



UNIVERSITETET I AGDER

---

*Performance Evaluation of the GCR Block ACK Mechanism in  
IEEE 802.11aa Networks*

---

*Author: Qihao LI*

*Supervisors: Dr. Lei JIAO  
Prof. Frank Y. LI*

Master Thesis in  
**Information and Communication Technology**

FINAL VERSION

Faculty of Engineering and Science  
University of Agder

June 3, 2013

## **Abstract**

With the growing demand for multimedia services, video streams have become major traffic sources in the Internet. However, it is challenging to transmit multimedia streams over IEEE 802.11 Wireless Local Area Networks (WLANs) with high performance and reliability. As a solution to improve system efficiency, a new standard, 802.11aa, is introduced to provide much more reliable and robust transfer of video stream by introducing several new service features. In this thesis, we analyze the performance of the Groupcast with Retries (GCR) Block ACK scheme as one of the most important features in this standard based on Markov chain models. The properties of groupcast service and block acknowledgement will be merged together in our model. Besides, we take into account the memory feature of the wireless channel and extend it into multi-receivers in accordance with the standard. In particular, the proposed model is built on fixed block size, a type of communication technique in which the block size is kept as fixed during transmission. Furthermore, the lost position of a packet in the block is taken into consideration for in-depth analysis.

Numerical results show that under the groupcast with constant retry limit and fixed block size transmission mechanism, the network throughput will be reduced with the growing number of receiving terminals. Compared with the throughput result for the variable block size mechanism obtained from our previous study, we found that with the increasing number of stations, the performance of the fixed block size mechanism is much more stable with respect to the number of stations and channel memory property. However, the variable block size transmission scheme exhibits much better throughput performance when the number of stations is small.

# Preface

This thesis is the result of IKT 590 submitted to fulfil the requirements of the Degree of Master of Science, at the Faculty of Engineering and Science, University of Agder (UiA) in Grimstad, Norway. The research topic was started from 1 January 2013 and ended on 3 June 2013. The thesis constitutes 30 ECTS credits towards the degree of MSc ICT. The main objective of this thesis is to analyze the performance of GCR Block ACK mechanism in IEEE 802.11aa network.

The task of the completion of this thesis would not have been possible without the support of several individuals. Our sincere gratitude goes to Lei Jiao who is a Post Doctor of the ICT department in University of Agder, for his valuable comments on the thesis. Working with Lei has been very inspiring due to his dedication, innovation ideas an excellent organization. Moreover, we are greatly indebted to my supervisor, Professor Dr. Frank Y. Li, for his guidance, support and teaching us the process of good scientific research. Last, but certainly not the least, I would also like to thank other team members for their constructive suggestions during this work.

Qihao Li  
Grimstad  
June 3 2013

# Contents

<b>Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>4</b>
<b>List of Tables</b>	<b>6</b>
<b>1 Introduction</b>	<b>7</b>
1.1 Introduction of 802.11 Wireless LANs . . . . .	7
1.2 Motivation . . . . .	8
1.3 Problem Statement . . . . .	8
1.4 Thesis Approach . . . . .	9
1.5 Thesis Organization . . . . .	10
1.6 Chapter Summary . . . . .	11
<b>2 802.11 WLANs Principle, MAC and Performance Evaluation</b>	<b>12</b>
2.1 Background Information on 802.11 WLANs . . . . .	12
2.1.1 Architecture . . . . .	13
2.1.2 Frame Type . . . . .	14
2.1.3 Medium Access Control . . . . .	15
2.2 Block Ack Mechanism in 802.11e Amendment . . . . .	18
2.3 Previous Studies on MAC Performance in 802.11 WLANs . . . . .	20
2.3.1 Media Access Control . . . . .	20
2.3.2 Reliable Media Transmission . . . . .	22
2.4 Chapter Summary . . . . .	22
<b>3 GCR Block Transmission Mechanism in IEEE 802.11aa Networks</b>	<b>23</b>
3.1 GCR Block ACK Application Scenario . . . . .	23
3.2 GCR Block ACK Setup . . . . .	24
3.3 GCR Block ACK Frame Structure . . . . .	26
3.4 GCR Block ACK Operation . . . . .	28

3.5	Chapter Summary . . . . .	29
<b>4</b>	<b>Performance Analysis</b>	<b>30</b>
4.1	System Model, Assumption and Analysis Procedure . . . . .	30
4.1.1	System Model . . . . .	30
4.1.2	Assumptions . . . . .	31
4.1.3	Analysis Procedure . . . . .	31
4.2	Packet Transmission Probability . . . . .	32
4.2.1	Retransmission Transition Model for One Packet . . . . .	32
4.2.2	Analysis of Block ACK Transmission Channel Model . . . . .	33
4.2.3	Retransmission System Status Model . . . . .	37
4.3	Chapter Summary . . . . .	44
<b>5</b>	<b>Numerical Results and Discussion</b>	<b>45</b>
5.1	Throughput . . . . .	45
5.2	Retransmission Probability . . . . .	48
5.3	Throughput on Large Packet Size . . . . .	49
5.4	Chapter Summary . . . . .	49
<b>6</b>	<b>Conclusion and Future Work</b>	<b>51</b>
6.1	Conclusion . . . . .	51
6.2	Future Work . . . . .	52
	<b>Bibliography</b>	<b>53</b>
	<b>MATLAB Code</b>	<b>55</b>

# List of Figures

2.1	The Architecture of IEEE 802.11 Wireless LANs. . . . .	13
2.2	The General Frame Format in IEEE 802.11. [10] . . . . .	14
2.3	The MAC Architecture in IEEE 802.11. [10] . . . . .	15
2.4	The Distribution Coordination Function (DCF). [10] . . . . .	16
2.5	IFS Relationships. [10] . . . . .	16
2.6	The ACK Operation Mechanism in 802.11 WLANs. [9] . . . . .	18
2.7	Markov Chain for BE ( $\nu = 1$ ) and BK ( $\nu = 0$ ) ACs. [19] . . . . .	21
2.8	Markov Chain for VO ( $\nu = 3$ ) and VI ( $\nu = 2$ ) ACs. [19] . . . . .	21
3.1	Infrastructure Transmission Mode with Access Point. . . . .	24
3.2	Message Sequence Chart for GCR Block Ack. [7] . . . . .	24
3.3	The MRG Block Ack Request Frame Format. [7] . . . . .	26
3.4	The MRG Block Ack Request Information Field Format. [7] . . . . .	26
3.5	The GCR Block Ack Frame Format. [7] . . . . .	27
3.6	The BA Control Field Format. [7] . . . . .	27
3.7	GCR Block ACK with Fixed Block Size (Mechanism 1). . . . .	28
3.8	GCR Block ACK with Variable Block Size. . . . .	28
3.9	GCR Block ACK with Fixed Block Size (Mechanism 2). . . . .	29
4.1	System Model Scenario. . . . .	30
4.2	Retransmission State Transition Model . . . . .	33
4.3	System Model by Two State Markov Chain. . . . .	34
4.4	System Model by Four State Markov Chain. . . . .	34
4.5	Block Transmission Relation between AP and Stations. . . . .	35
4.6	Retransmission State Transition Mode 1 . . . . .	38
4.7	Transition Probability from State 10 to State 02. . . . .	39
4.8	Retransmission State Transition Mode 2 . . . . .	41
5.1	Throughput versus Number of Stations based on Variable Block Size. [1] . . . . .	46
5.2	Throughput versus Number of Stations based on Fixed Block Size. . . . .	47

5.3 Successful Transmission Probability versus Number of Stations. . . . . 48

5.4 System Throughput with Large Packet Size. . . . . 49

# List of Tables

2.1	User Priority to Access Categories in EDCA [9] . . . . .	17
2.2	Different Channel Access [9] . . . . .	18
4.1	Four Cases of Transition Probability $P_{A_j, A_{j+1}}$ . . . . .	37
4.2	One Frame Retransmission Transition States . . . . .	38
4.3	Transmission State for Each Packet and Corresponding Probability . . . . .	39
4.4	Transmission State for Each Packet and Corresponding Probability (Technical 2) . . . . .	41
4.5	Payload and Header Transmission Time . . . . .	43
5.1	System Parameters used to obtain Numerical Results . . . . .	45



# Chapter 1

## Introduction

In this chapter, along the traces of growing demand for multimedia service, a brief introduction about the shortcoming of original 802.11 standards on poorly supporting multimedia stream in WLAN will be given and the motivation for choosing this topic will be expressed. Moreover, we will indicate the problem we raise for our thesis and then briefly explain the method to solve it. Finally, we will present the structure of our thesis.

### 1.1 Introduction of 802.11 Wireless LANs

Wi-Fi is a popular technology that enables wireless data communication between electronic devices over a computer network. Convenience access function and much more unrestricted connection make wireless media connection become the dominant media access function.

As IEEE 802.11 standard for Wireless LAN (WLAN) continues to be major wireless access networks in residential buildings and offices, users tend to expect much more variable multi-media applications and services than simply data and media information exchange. However, higher data transmission restriction on original IEEE 802.11 wireless networks cannot present an efficient video stream transmission for this tendency. There are several reasons verifying the shortcomings of the original 802.11 standard on poorly supporting effective multimedia stream transmission [2], including: 1) in spite of the efficiency guarantee of packets delivery on MAC protocol, the basic rates, e.g., 6Mbps in 802.11g, remain the entire system transmission limitation; 2) the management overhead on access mechanism for different types of packets is too much; 3) inefficiency and unreliable transmission mechanism still provide unsatisfied video streaming performance to terminal users in accordance with current WLANs transmission techniques.

The first two limitations have been mitigated by subsequently ratifying amendments to the 802.11 standard. On the one hand, the maximum achievable rates have been raised from 11 Mbps by amendment of 802.11b to 54 Mbps by 802.11a/g, and finally to 600 Mbps modulation rates based on 802.11n. On the other hand, media stream access differentiation is introduced in 802.11e via the setting of the contention parameters to ensure the different packet categories prioritization. [23] Although the classification on different types of packets still cannot guarantee transmission

priority, this category function has proved its effectiveness by various research papers. Therefore, the only challenge remained is system transmission reliability and efficiency. In this thesis, we will study the new standard, 802.11aa, which provides several service features on a reliable and robust video stream transmission.

## 1.2 Motivation

Following the ratification of the 802.11aa standard in 2012, the motivation for this master thesis is to build an analytical model on one part of this standard in order to evaluate the performance of the new standard. There are new services introduced in the standard to provide much more reliable and robust multi-media transmission, which contains: Groupcast with Retries (GCR), intra-access category prioritization, Streaming Classification Service (SCS), Overlapping Basic Service Set (OBSS) management, and interworking with the IEEE 802.1Q Stream Reservation Protocol (SRP). [7]

Based on extensive literature studying, there does not exist analytical model for GCR Block Acknowledgment (GCR Block ACK), which is one of the MAC sublayer functions in the first service feature. In this thesis, we will firstly summarize the technical points and mechanism details of this mechanism, and then try to make a model to precisely analyze its performance. Moreover, most of previous Block ACK assumption on successive packets transmission is analyzed based on mutual independent relationship between each packet. However, in accordance with wireless channel experience, channel memory property, which means the last transmission packet has an impact on the current one, has not been taken into consideration in these models. This becomes one of our targets of this thesis. Furthermore, earlier analysis model on consecutive packets transmission is built based on single receivers in accordance with the related standard. As the new amendment proposes a new scenario that system transmits block of packets to multi-receivers simultaneously, a new precise model corresponding to this scenario is required. Motivated by these requirements for an efficient analyzing model which can optimize channel transmission mechanism and system operation scenario, we focus on modelling system in this thesis.

In addition, there are two different block size configurations on Block ACK transmission mechanism, which are variable block size and fixed block size. It denotes a type of communication technique in which block size varies or keep fixed during transmission. We will compare the advantage and disadvantage of these two mechanism from the numerical result of our model. Moreover, we will modify the system configuration, in order to analyze the working performance of the system in different situations. These issues become some other targets of our thesis and motivated factors of our current research work.

## 1.3 Problem Statement

The analysis of video and audio transmission performance in wireless communication between transmitter and receiver is always the interest of many researchers. The reason that the data frame cannot be delivered reliably on time due to the changing wireless channel, the lack of coordination on MAC layer, or out-dated multimedia process function, performance evaluation model is built

based on different assumptions. The ratification of the new standard 802.11aa produces Groupcast Retries Block ACK, one of the most important services to enhance the media stream reliability transmission. It has some new features different from those previous study. As we know, the Block ACK mechanism is firstly proposed in 802.11e standard to propose as system retransmission guarantee mechanism. However, the application scenario is proposed as transmitting block of packets to one receiver, simultaneously. Then the question naturally comes up, “how we can build a model to evaluate the performance on multi-receivers based on Block ACK mechanism?” Moreover, previous studies found the continuous packets transmission model based on the assumption of packets mutually independent from each other. However, wireless channel follows memory property in practice. How we can integrate this characteristics into our model? Furthermore, in order to describe the system behavior, a system model should be produced. Then here goes another question: how we can combine the channel model and system model together?

This thesis addresses the following questions on how to:

- combine the Groupcast Retry scenario with Block ACK mechanism together;
- take channel memory feature into account to precise our model;
- build up system model to indicate the system transmission state in ordinary mode;
- combine the channel model and system model together;
- try to evaluate the system throughput and successful transmission probability, and compare the performance of two different block ack mechanism.

## 1.4 Thesis Approach

In this thesis, the main objective is to evaluate system performance based on the transmission scenario described in the standard. As the initial task, we will give an introduction on original 802.11 WLANs mechanism. In this part, Block ACK mechanism in previous standard will be mentioned. Then, we target at the GCR Block ACK produced in the new standard 802.11aa. We will depict the setup process, frame format and mechanism operation of GCR Block ACK in details. Moreover, we will design our model by considering all the conditions we mentioned in previous section. Finally, we will evaluate our model and analyze the numerical result. Therefore, the main tasks of the thesis are summarized as follows.

1. Study the GCR Block ACK which is one of the important service features in the new standard 802.11aa.
2. Design a specific channel model by considering channel memory property. Extend this model to a multi-receiver channel model and arrange it from system’s point of view.
3. Design a system model based on Block ACK mechanism. Integrate the channel model into this system model and analyze the performance from system ordinary state.
4. Evaluate the system performance in terms of different parameters and validate those results by using MATLAB calculation.

## 1.5 Thesis Organization

The exposition in this thesis deals with the design of the combination between channel model and system model for the GCR Block ACK mechanism. A substantial section of this thesis has been concerned with the design of a Markov chain for channel model and a Markov chain for system model, respectively. The combination of these two models is another critical part in this thesis. The thesis is conceived in six chapters and the contents of each chapter are summarized below.

### **Chapter 1:**

In the first chapter, a brief introduction on high data rate requirement and original amendment on 802.11 standards are given. Then, the motivation for choosing this topic is exposed. Moreover, we indicate the problem we arise for our thesis and briefly explain our method to solve it. Finally the problem statement is presented with the thesis definition.

### **Chapter 2:**

Chapter 2 gives a general description on the structure of the 802.11 wireless LANs. Some fundamental concepts like media access control or frame format will be explained in this part. Moreover, in order to make better sense of the GCR Block ACK mentioned in new standard 802.11aa, we will focus on the explanation of the Block ACK operation in 802.11e. Additionally, some prior model and numerical analysis of the Block ACK operation and EDCA mechanism will be briefly depicted.

### **Chapter 3:**

The new amendment is characterized by several important services features: group addressed transmission service, stream classification service, management of overlapping networks and support for the IEEE 802.1QTM Stream Reservation Protocol. In this part, we will discuss the first service feature, GCR Block ACK mechanism. This chapter will be separated into four sub-chapters. In the first part of this chapter, we will figure out the new standard application scenario in order to overall explain the GCR Block ACK mechanism operation. The second part will indicate the setup function and third part will illustrate the system format. Finally, three different Block ACK operation will be explain in details which is also the foundation scenario of our analysis model.

### **Chapter 4:**

Chapter 4 contains a comprehensive analysis of the proposed model. We will analyze the saturated throughput of GCR Block ACK with fixed block size under an infrastructure Basic Service Set (BSS) in this part. The model will be built based on two important points in this part. Firstly, we will build up system channel model based on channel memory property. We will consider a scenario that system transmits a block of packets consecutively to multi-receivers. A system transmission model will also be made up to illustrate the transition between different system statuses. Moreover, we will combine the channel model and system model together to analyze the overall system transmission status.

### **Chapter 5:**

The numerical results are shown in Chapter 5. The performance of the system is illustrated in terms of system throughput and successful transmission probability.

**Chapter 6:**

Finally, chapter 6 summarizes the main contributions and conclusions of the thesis and points out some directions for future research.

## **1.6 Chapter Summary**

The chapter began with brief introduction on original shortage of multi-media stream transmission mechanism. We figure out some solutions in accordance with the amendments to the 802.11 standard. Motivate factors behind the thesis work is then explained by summarizing previous model and indicating our main designing concept. In the problem statement, some technical problems are referred in this part. The main objective of the thesis and activities are presented in the thesis approach. The chapter ended by thesis organization.

# Chapter 2

## 802.11 WLANs Principle, MAC and Performance Evaluation

In this chapter, the general introduction on original 802.11 standard will be given. The chapter will be separate into two parts, background information on 802.11 WLANs and introduction to previous study. The purpose to indicate the first part is that the new standard we want to analyze is designed based on the original 802.11 transmission environment. It is necessary to have some fundamental concepts on the basic standard in order to have a better understanding of the new amendment. And the objective for the second part is used to depict the previous studies on MAC performance and reliable transmission.

In the background information introduction part, we will indicate some important and basic techniques which is related to our research. At first, we will figure out the overall architecture of the 802.11 wireless LAN. The application scenario of new standard 802.11aa, which we will explain in the third chapter, is contained in this structure. Then, we will show the basic frame format. This part is the foundation of designing principle of all the frame format. Moreover, we will indicate the different Medium Access Control which is the foundation of wireless transmission organization. Some basic concepts used in the new standard which will be mentioned in chapter three, will be explained here. In addition, we will focus on the explanation of the Block ACK operation indicated in previous amendments to give us a better knowledge on the new standard and its implication.

In the introduction to previous study part, some prior models and numerical analyses of the Block ACK operation and EDCA mechanism will be briefly depicted.

### 2.1 Background Information on 802.11 WLANs

In this section, a brief introduction on the IEEE 802.11 wireless network standard will be described. There are various supplements and amendments concluded in the standard, such as the different physical layer implementations for 2.4 and 5 GHz frequency band or the Quality of Service (QoS) extensions.

We will place our focus on the MAC sublayer functionality which is required for the comprehension of the IEEE 802.11aa amendment. Besides, we will give a brief introduction to the physical medium dependent sublayer which is defined in original IEEE 802.11 standard.

### 2.1.1 Architecture

There are two topologies for wireless LAN. The first one is Ad Hoc networking in which there is no central wireless station connecting others. Therefore, data frames are sent directly from the sender to the receiver. Due to the impact of channel interference or new station access, the data information may be lost during transmission. This will result in a complete loss on the information, and it can only be solved by requesting the sender to retransmit it. The system cannot recover the data through retransmission by itself. Protocols that use intermediate wireless stations as relay stations to forward such transmissions are called mesh-networks. This mode is called Independent Basic Service Set (IBSS) in the standard. And a brief architecture of IBSS is shown in Fig. 2.1.

The second one is known as Basic Service Set (BSS). It is an infrastructure mode or managed mode in which a central wireless station called access point (AP) connects other wireless stations together. All data frames are sent from and to this AP. A wireless LAN with multiple access points connected by a wired backbone is called an Extended Service Set (ESS). It is able to cover larger areas than a single AP with wireless LAN does. Besides, in IEEE 802.11, there is a service always ensuring that a mobile wireless station is connected with the next AP. Fig. 2.1 gives a brief exhibition on BSS and ESS, respectively.

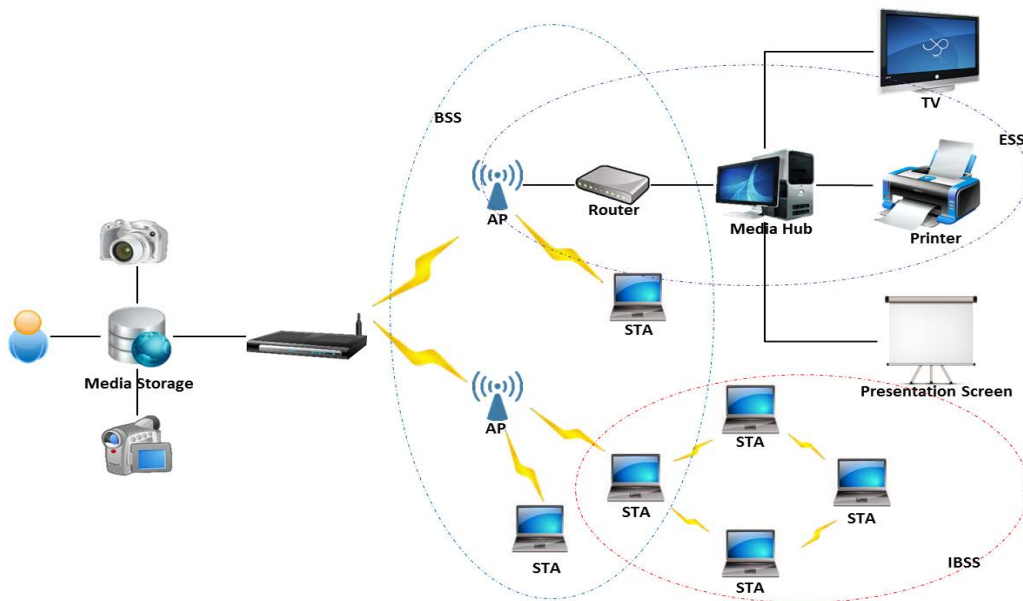


Figure 2.1: The Architecture of IEEE 802.11 Wireless LANs.

### 2.1.2 Frame Type

Obviously, the data should be organized in the same function to promise the possibility of decoding during different transmission process. In order to set up diverse transmission function, construct and decode particular operation, all stations shall be able to validate every received frame using the frame check sequence (FCS) and to interpret the information in certain fields.

Following the basic standard 802.11, the components of general MAC frame are:

- MAC header: Indicate frame control information, address, duration ID, and other basic knowledge to tell receiver how to decode and install own parameter.
- Frame Body: Primary data information.
- FCS: Contains an IEEE 32-bit CRC

A general frame format is shown in Fig. 2.2. In this figure, Frame Control, Duration and Address 1 establish the minimal frame and they should be contained in every generated frame. The first eight fields in frame format are described as MAC header to mention the system parameters of associated group members. The decoding function is also explained in this place. In order to provide reliable transmission, this part should be precisely clarified. The Frame Body field can be offered in various sizes among which the maximum size is 2304 octets. FCS part is used to correct the bit error in MAC layer. Depending on respective purposes, IEEE 802.11 defines three types of

Frame Control	Duration/ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control	Frame Body	FCS
2	2	6	6	6	2	6	2	0-2304	4

Figure 2.2: The General Frame Format in IEEE 802.11. [10]

frames to satisfy the users’ requirements: the *management frames*, the *control frames* and the *data frame*. In order to distinguish different types of frame, Type field and Subtype Field contained in the Frame Control to specify individual frame. Control Frame, which sets Type Field as 00, is used to guarantee reliability of data transmission. Data Frame, which sets Type Field as 10, provides the certain data which is necessary for receivers. Management Frame, which sets Type Field as 01, is used to build up a transmission procedure.

Moreover, the management frames are frames that associate, authenticate and setup agreements between stations. Like Acknowledgment (ACK), Block ACK Request, Block ACK Response and the RTS/CTS frames, the control frames are also used for transmission control. And the structure of management frame will used in the Block ACK mechanism discussion part. In addition, the data frames are frames that deliver transmitted data. It has some subtypes for QoS and HCCA support. More in details, the data forwarded to or from the upper layer is referred to as MAC Service Data Unit (MSDU), while data forwarded to or from the lower layer as MAC Protocol Data Unit (MPDU). A-MSDU can be divided into up to 16 MPDUs. [8]



### 2.1.3 Medium Access Control

Being a shared medium, the wireless channel only allows one station to transmit successfully at one time. [23] Different methods given in the IEEE 802.11 Medium Access Control (MAC) have to coordinate this access to the channel.

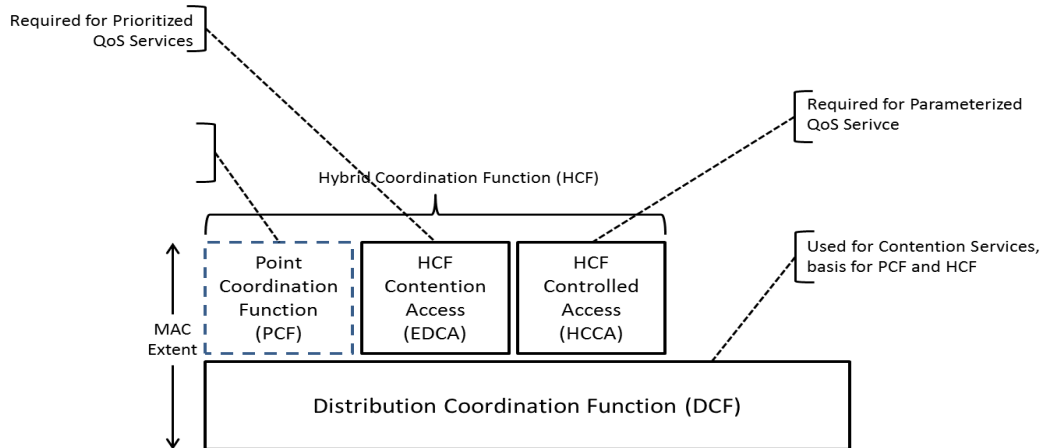


Figure 2.3: The MAC Architecture in IEEE 802.11. [10]

The data received from upper layer is transmitted in frames, each of which consists of a header, a frame body with adjustable length for the data and a check sequence frame. The header contains four parts: (a) frame control, determining the type of the frame; (b) duration, the time needed to transmit the frame and get a possible reply; (c) up to four addresses, the intended receiver or receivers of the frame and the source; (d) sequence control, 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. [21]

The duration and addresses contained in the header can help wireless stations protect transmission from interference and save power. The latter is done through virtual carrier sensing. When a station receives a header not addressed to it, it can turn off wireless radio for the provided duration to save energy. [21] If the transmission of a frame invokes as reply or it is intended to transmit more than one frame in a row, the value of the first frame's duration can be set to cover the complete transmission sequence, which will make other stations aware of the ongoing transmission in order to avoid interference.

#### DCF

Distributed coordination function (DCF), also known as carrier sense multiple access with collision avoidance (CSMA/CA) [10], is the fundamental access method of the IEEE 802.11 MAC. Every station can sense the wireless channel and will start to transmit until the channel is free. This is characterized by whether the received power on the channel is under the threshold for a given period of time. When this value is above the threshold, there are two suggestions: one is that another transmission is occupying the channel, the other one is that the channel is too noisy to transmit data successfully. [10]

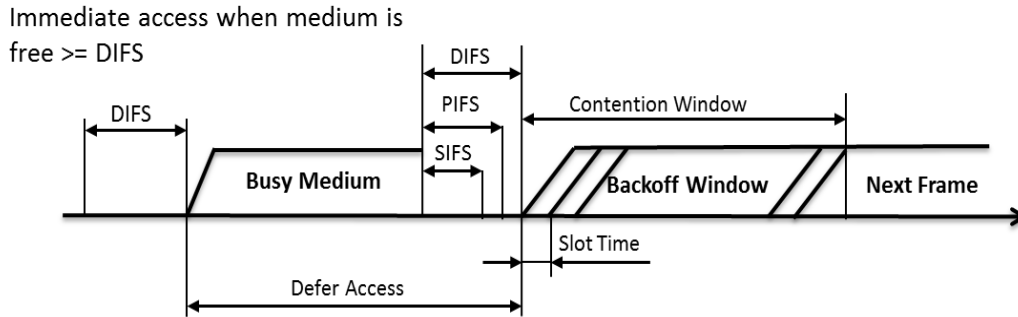


Figure 2.4: The Distribution Coordination Function (DCF). [10]

The channel must be free during the interval of two consecutive frames. This interval is called inter-frame space (IFS), and its relation between other IFSs is shown in Fig. 2.5. In DCF, such interval is called DIFS. When the channel is not free after one DIFS, the station will run a random back-off algorithm by drawing a random number from the contention window. [10] After the back-off number turns to zero, and the transmitter find the system channel is free. It will transmit during this system duration. On the contrary, if the transmitter sense the channel is busy after back-off counting. The transmitter will follow the procedure mentioned above until it senses the channel is free.

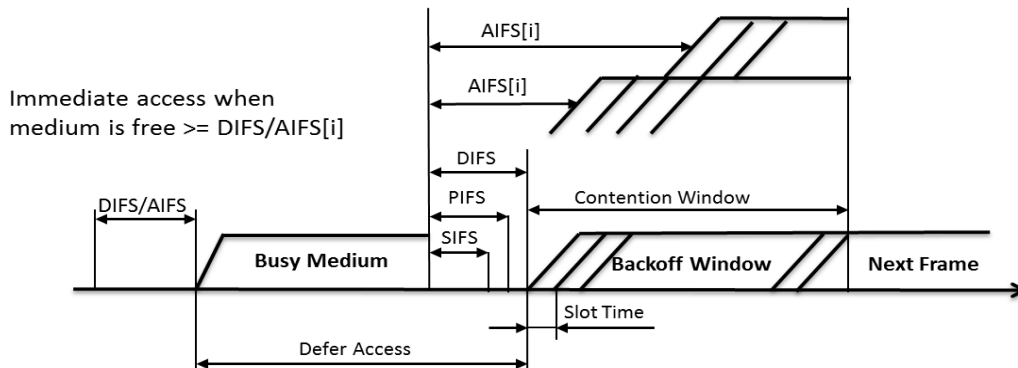


Figure 2.5: IFS Relationships. [10]

## PCF

Point coordination function (PCF) is proposed as that the IEEE 802.11 MAC may incorporate an optional access method on the top of the DCF. It is only applicable to infrastructure network configurations, in which circumstance the access point coordinates the channel access. [10] The period that media access to the wireless channel under the control of the PCF is called contention free period (CFP), while that under the control of the DCF is called contention period (CP).

When the wireless channel is free during one PIFS, the access point can poll the stations configured for PCF. The polled station sends out one data frame immediately. In the contention-free

period, the channel can only be idle for a SIFS. PIFS and SIFS duration are shorter than DIFS duration, as is shown in Fig. 2.5. Stations that only use DCF are unable to gain access to the wireless channel when it is occupied by an access point. To prevent this, CFP duration is limited and must be followed by a CP.

### HCF contention-based channel access (EDCA)

The EDCA, an extension of the DCF, defines four access categories (AC): voice, video, best effort and background traffic. Voice traffic enjoys the highest priority while background traffic enjoys the lowest. The IEEE 802.1D defines a user priority flag that can be set in the header of each frame. Table 2.1 shows how this user priority is mapped to the EDCA access categories.

Table 2.1: User Priority to Access Categories in EDCA [9]

UP	Designation	AC	Designation
1	BK	AC_BK	Background
2	–	AC_BK	Background
0	BE	AC_BE	Best Effort
3	BE	AC_BE	Best Effort
4	CL	AC_VI	Video
5	VI	AC_VI	Video
6	VO	AC_VO	Voice
7	NC	AC_VO	Voice

Beacon frame which is send by access point will announce the maximum duration of a TXOP. [23]. In the beacon frame structure, there is an element provided EDCA Parameter Set information. A wireless station can retransmit frames with more robust rates within a TXOP, if it is received incorrectly at the receiver part. Following the EDCA mechanism, it is responsible for stations to ensure that in its access duration, it won't occupy wireless channel longer than the maximum TXOP duration. [21]

For every access category, an arbitration IFS (AIFS) is defined along with a maximum size of the contention window. [23]  $AIFS[AC]$  denotes the AIFS duration for AC. The formula of AIFS is given by [9]

$$AIFS[AC] = AIFSN[AC] \cdot aSlotTime + aSIFSTime \quad (2.1)$$

where  $AIFSN[AC]$  is announced by the access point in the EDCA Parameter Set.

From Fig. 2.5, we can obtain the different access period for different access method. Moreover, in accordance the standard, EDCA access method follows the function of DCF. The size of the contention window in DCF and EDCA is doubled after each transmission error. The unsuccessful access will increase the retry counter one by one until the number of the retry exceed the system max retry limit. Then transmitter will drop this packet, and reset the contention window back to system configuration. When the transmitter successfully access the channel, the contention window of the transmitter is reset to its minimal size.

The contention window size in EDCA is defined as  $CW[AC]$  for each access category. Initially

Table 2.2: Different Channel Access [9]

AC	AIFSN	$CW_{min}$	$CW_{max}$	$Default_{min/max}$
BG	7	$aCW_{min}$	$aCW_{max}$	15/1023
BE	3	$aCW_{min}$	$aCW_{max}$	15/1023
VI	2	$(aCW_{min} + 1)/2 - 1$	$aCW_{min}$	7/15
VO	2	$(aCW_{min} + 1)/4 - 1$	$(aCW_{min} + 1)/2 - 1$	3/7

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. [23] In 802.11a the external and internal collisions are handled in the same way. However, the same two parts worked in EDCA which is called the EDCAF, runs independently for each access category in accordance with the back off algorithm. [21] Table 2.2 shows how the contention windows are defined for each access category. In the EDCA Parameter Set, the  $CW_{min}$  and  $CW_{max}$  is determined by  $ECW_{min}$  and  $ECW_{max}$ . The relationship between them can be expressed as:

$$\begin{aligned}
 CW_{min} &= 2^{ECW_{min}} - 1 \\
 CW_{max} &= 2^{ECW_{max}} - 1
 \end{aligned}
 \tag{2.2}$$

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  $CW_{max} = 1023$ , respectively. [23]

## 2.2 Block Ack Mechanism in 802.11e Amendment

In the legacy IEEE 802.11 standard every transmitted data frame is acknowledged with an ACK frame. This is illustrated in Fig. 2.6.

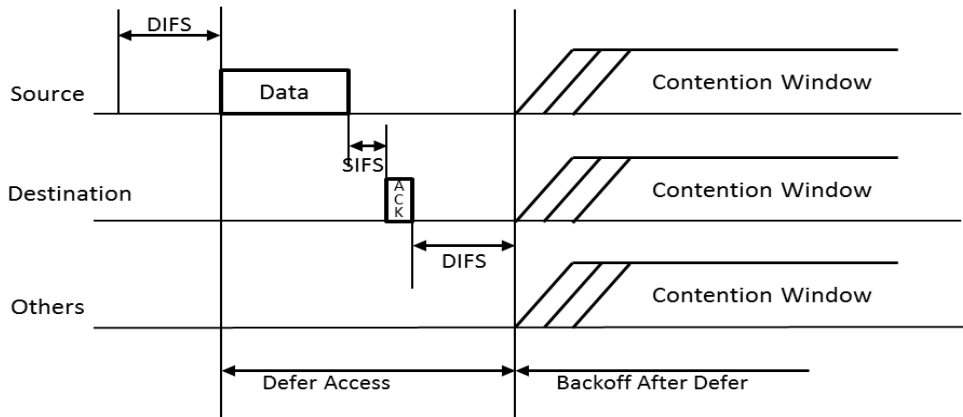


Figure 2.6: The ACK Operation Mechanism in 802.11 WLANs. [9]

This form of transmission is synchronous transmission. The originator will first transmit a frame to receiver and wait for an acknowledgment to ensure the connection. According to the

basic access method we provide in previous discussion, the duration for the originator to wait the ACK is SIFS. If after one SIFS, no ACK is feedback from the recipient. The originator will retransmit this packet based on the assumption that there is a failure reception at the receiver part.

With the TXOP duration mechanism mentioned in standard 802.11e, the channel utilization presents much better performance. 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, the duration between two continuous packets is one SIFS idle time.

The Block Ack mechanism also provides a more efficient channel usage by aggregating several acknowledgments into one frame. The transmitter need first initial the Block ACK mechanism from handshaking with receiver. Then, it can transmit a group of packets, sequentially, without ACK for any transmitting packet to 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 group of packets is transmitted by the originator without waiting for an ACK to each of packets. A buffer will be allocated for these packets in order to retransmit the failure reception packets. On the other hand, the recipient also arrange space for ordering the packets received from the originator. For the reason of disordered packets which is retransmitted based on the failure reception, a cache will be configured in each packet to keep track of the received packets. [22]

The Block ACK mechanism is used only if the transmitter and receivers both provide QoS service feature. The reason is that the sequence number in frame format which is provided in QoS is one necessary part for mechanism operation. In original 802.11 standard, system configures one sequence number for each station. However, in QoS operation mechanism, the sequence number is allocated to each data packet. Traffic identifier (TID) is used for this sequence number. Not only the data packet is labelled by TID, but also the user priority is also indicated by it.

### **Block Ack Operation**

The number of MPDUs is limited by originator transmission buffer. Moreover, for each MPDU, there is a replica corresponding to it in order to deal with the failure reception problem. The total number of MPDUs is configured during system handshake based on the buffer size of recipient. In accordance with the sequence number of each MSDU, the recipient should arrange the receiving packets to correct sequence in order to forward them up the higher layer. In addition, the receiver will keep track of each MPDU in a separate cache in order to forward the old sequence number packet to the upper layer as soon as possible. It is because a MPDU with old sequence number cannot be kept in the buffer

The sequence number space is of size  $2^{12}$ . And the space can be divided into the two parts, “new” and “old”. The “new” range begins with the start of the receiving buffer and ends with the place where plus  $2^{11}$  modulo  $2^{12}$  from the start. The rest is the “old” range.

After originator sequentially transmitting a block of packets, the Block ACK Request packet will be transmitted to ensure the reception state of the recipient. A starting sequence number is involved in the packet. This number indicates the start position of the window and was set to the oldest MPDU based on the ACK. Then, the receiver can feedback the reception status by one Block ACK packet.

In accordance with receiving status of each MPDU, the recipient will set the bits in corresponding position to 1 or 0 to label the lost packet. Then, the originator will determine the retransmission table based on the information got from the Block ACK.

In conclusion, the Block ACK mechanism is aimed at decreasing transmission overhead for unicast application scenario. The new amendment 802.11aa is published with proposed a new service feature in groupcast feedback mechanism. The Groupcast Retry Block ACK mechanism is introduced as one of the most important service features in this amendment. In the following chapter, we will first give a brief introduction on this mechanism, and then try to make a model to analyze the performance of it.

## 2.3 Previous Studies on MAC Performance in 802.11 WLANs

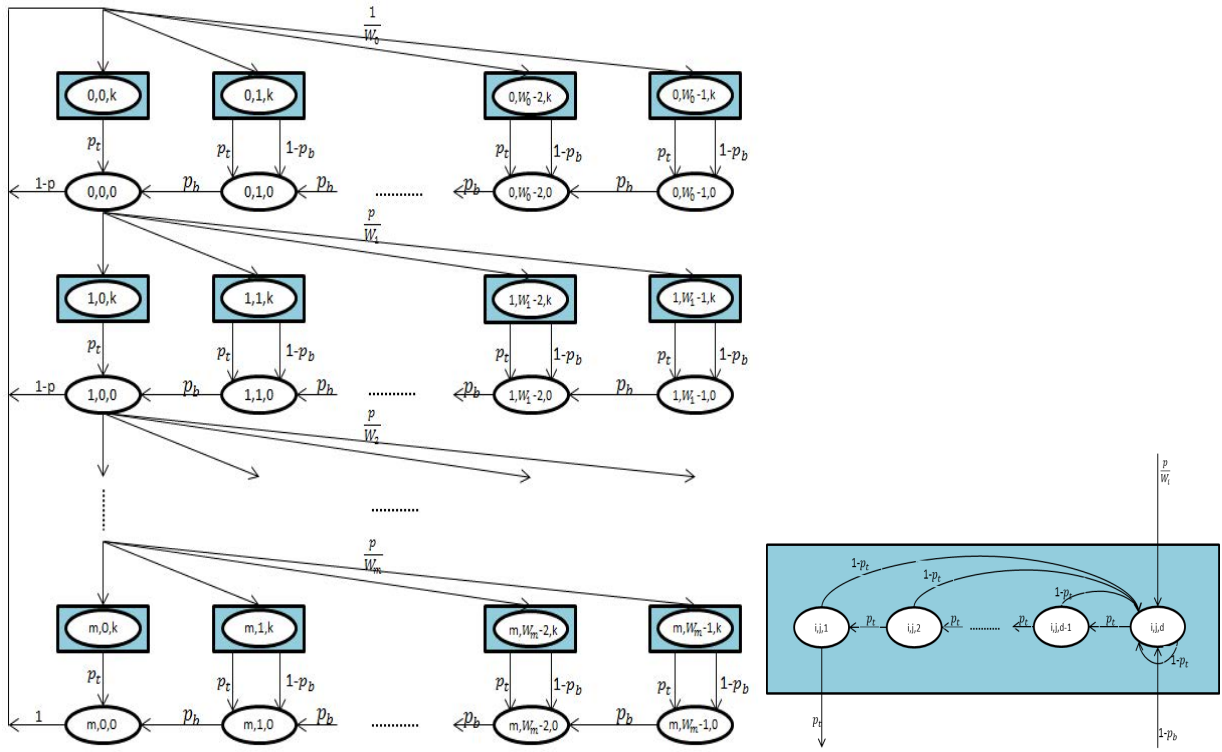
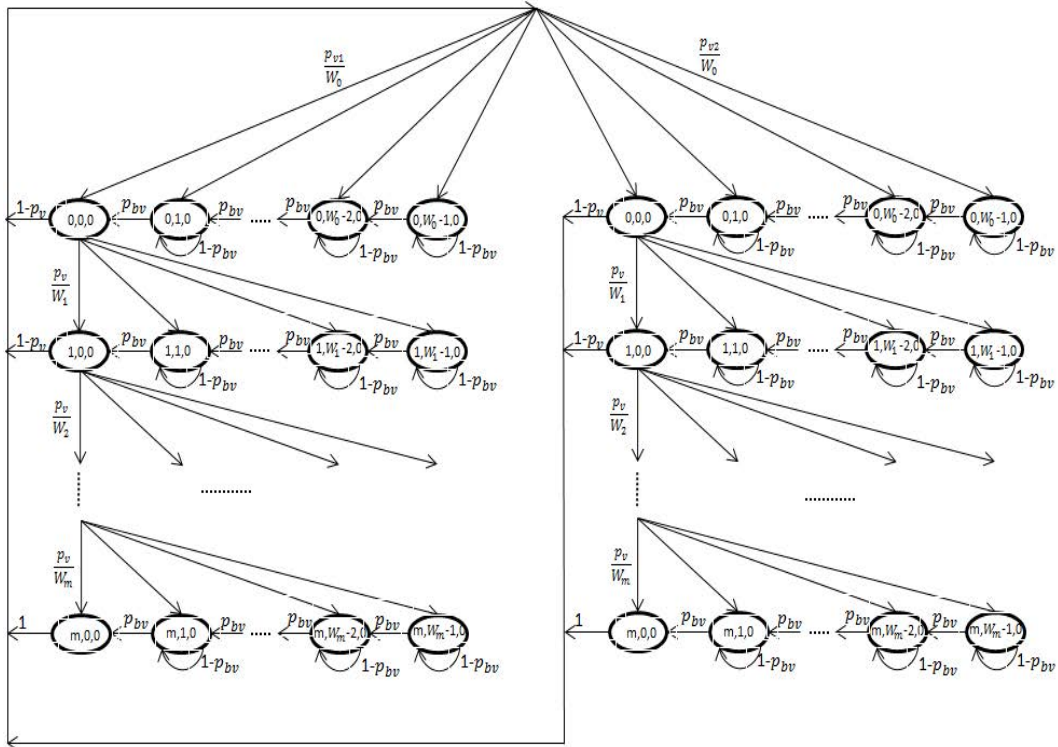
Having a high overhead, the IEEE 802.11 Medium Access Control (MAC) is unable to provide higher-speed WLANs required by customers. The solution to improve system efficiency is typically separated into two directions. One of them is improved from the media access control method, and the other one is enhanced from the system transmission reliability after successful channel access. In this part, we will introduce some related previous discussion on these two parts.

### 2.3.1 Media Access Control

Classical media access analysis model is Bianchi Model mentioned in [4]. In order to solve the problem of inefficient support on high Quality of Service (QoS) stream transmission, 802.11e [9] proposed a new method through differentiating traffic services. In 802.11aa [7], intra-access category prioritization is improved by adding two more access category to support finer grained prioritization of audio video streams [19]. [19] presents a new analytical model, based on the existing comprehensive EDCA model, to show the investigation on this part. For each AC, let  $s(t)$ ,  $b(t)$  and  $c(t)$  denote the stochastic processes representing the backoff stage, the backoff counter, and remaining number of time slots during the deferring period at time  $t$ , respectively. The 3-D process  $\{s(t), b(t), c(t)\}$  can be modelled as discrete-time Markov chains shown in Fig. 2.7 and Fig. 2.8. [19] We use  $\nu \in \{0, 1, 2, 3\}$  represent the  $\nu$ th AC in this intra-access category prioritization, which corresponds to BK, BE, VI, and VO, respectively.

Let  $b_{i,j,k}$  be the stationary distribution of the proposed Markov chains. Following the analysis process mentioned in [3], [19] and [5], we can get the steady state  $b_{0,0,0}$  for BE and BK:

$$b_{0,0,0} = \left[ \frac{(1 - p_{t\nu}^{d\nu})}{(1 - p_{t\nu})p_{t\nu}^{d\nu}} \left( (1 - p_{b\nu}) \sum_{i=0}^m \frac{W_{i\nu} - 1}{2p_{b\nu}} p_{\nu}^i + \sum_{i=0}^m \frac{p_{\nu}^i}{W_{i\nu}} \right) + \sum_{i=0}^m \frac{W_{i\nu} - 1}{2p_{b\nu}} p_{\nu}^i + \frac{1 - p_{\nu}^{m+1}}{1 - p_{\nu}} \right]^{-1} \quad (2.3)$$


 Figure 2.7: Markov Chain for BE ( $\nu = 1$ ) and BK ( $\nu = 0$ ) ACs. [19]

 Figure 2.8: Markov Chain for VO ( $\nu = 3$ ) and VI ( $\nu = 2$ ) ACs. [19]

And  $b_{0,0,0}$  for VO and VI as:

$$b_{0,0,0} = \left[ \sum_{i=0}^m \frac{W_{i\nu} - 1}{2p_{bv}} p_{\nu}^i + \frac{1 - p_{\nu}^{m+1}}{1 - p_{\nu}} \right]^{-1} \quad (2.4)$$

where  $m$  is the frame retransmission limit,  $W_{iv} - 1$  is the size of the contention window after the  $i$ th retransmission,  $p_v$  is the probability of collision, and  $p_{bv}$  is the probability of an idle medium in a time slot after AIFS. [19]

Then, we can get the overall transmission probability for the  $\tau$ th AC:

$$\tau_v = \frac{1 - p_v^{m+1}}{1 - p_v} b_{0,0,0} \quad (2.5)$$

In order to make our model precise, we should take MAC performance in our consideration. This part will be used in the future discussion.

### 2.3.2 Reliable Media Transmission

Block ACK, as one of the most important retransmission mechanism, is first mentioned in 802.11e [9] to improve the system efficiency. In [6] and [16], an analysis model based on packet level is proposed in accordance with this retransmission mechanism. The average transmission duration in the system interval and the average successful transmission payload are used to be the research foundation in this model. And the system throughput,  $S$ , can be figured out as:

$$S = \frac{E\{successful\ transmission\ payload\}}{E\{transmission\ duration\ in\ the\ system\ interval\}} \quad (2.6)$$

However, previous model in [6], [13] and [18], on this part tend to consider the transmission frame as an independent single frame. These analyses ignore the memory characteristic of wireless channel. In our model, we will take into account the memory feature of wireless channel to enhance the analysis model in this part.

## 2.4 Chapter Summary

In this chapter, we introduce the background information, basic technical concepts, and previous studies on original 802.11 WLANs, in order to lay the foundation for our research on new amendment. In the background information part, we exhibit the architecture of the whole 802.11 WLANs. We explain the basic frame format and describe the different medium access control method. Moreover, we demonstrate the Block ACK mechanism in previous amendment. And finally, we finished this chapter with previous studies on MAC performance. We present the extended model on EDCA mechanism which is mentioned in the new amendment which we will discuss in the next chapter. The access probability obtained from this model can be used directly in future discussion. In addition, we discuss about the previous study on Block ACK mechanism. In chapter three, the extended Block ACK mechanism, which is called GCR Block ACK, will be introduced.



## Chapter 3

# GCR Block Transmission Mechanism in IEEE 802.11aa Networks

The IEEE 802.11aa is published as a new amendment on MAC enhancements for robust audio and video streaming transmitting. This amendment tries to realize reliable and quality transmission on audio and video stream. Moreover, it is used to be the fundamental technique standard to facilitate the implementation of Wi-Fi display. It is characterized by several important services features: group addressed transmission service, stream classification service, management of overlapping networks and support for the IEEE 802.11QTM Stream Reservation Protocol. [2] [11] [12] In this part, we will focus on the Groupcast with Retries Block ACK, which is one of the most important system retransmission technique in the standard. We will firstly figure out the application scenario of our discussion. Then, present the new definition of this mechanism on system setup procedure, frame format and transmission operation, respectively, in the new standard.

### 3.1 GCR Block ACK Application Scenario

Figure 3.1 is an illustration of the infrastructure mode for 802.11aa networks. The transmitter sends a block of frames which pass through the access point using the 802.11aa mechanism to a group of receivers. As defined in this amendment, the groupcast originator (AP or non AP that provides the GCR service) should win the contention before its frame transmission starts and then declare whether its group members are capable enough to initiate this transmission function to decide which service to use. In addition, the system parameters should be configured during handshaking. In this study, since our focus is on the data block transmission and acknowledgment procedure, we skip the declaration and handshake procedure which initiates the GCR Block ACK. Moreover, we make an assumption that the originator can obtain the transmission permission whenever it needs to so as to simplify the analysis procedure.

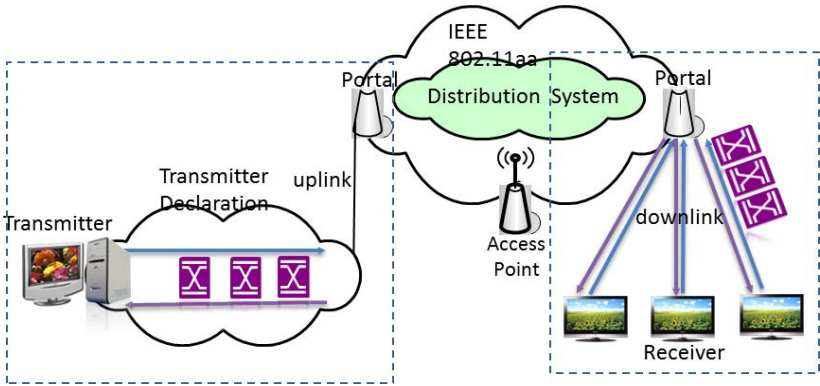


Figure 3.1: Infrastructure Transmission Mode with Access Point.

### 3.2 GCR Block ACK Setup

Groupcast transmission techniques have been widely applied to TV and radio broadcasting, gaming, video conferencing etc., as groupcast can reduce network traffic by transmitting the same data stream to multiple recipients simultaneously. Although the current 802.11 standard gives two solutions to group addressed frames transmission: broadcast and directed multicast (group addressed frames are converted to individually addressed frames), there is still shortcoming in this mechanism. The first is not reliable while the second is unscalable.

To avoid these shortcomings, 802.11aa proposes a new mechanism of providing feedback and admission control for multi-cast traffic, which is called Groupcast with Retries (GCR). GCR can make groupcast transmissions more reliable to overcome the problem of lacking efficient and robust support for transmitting audio/video contents to multiple destinations, which has resulted in the Project IEEE 802.11aa Task Group 1. In this section, we will explain the setup procedure of this mechanism.

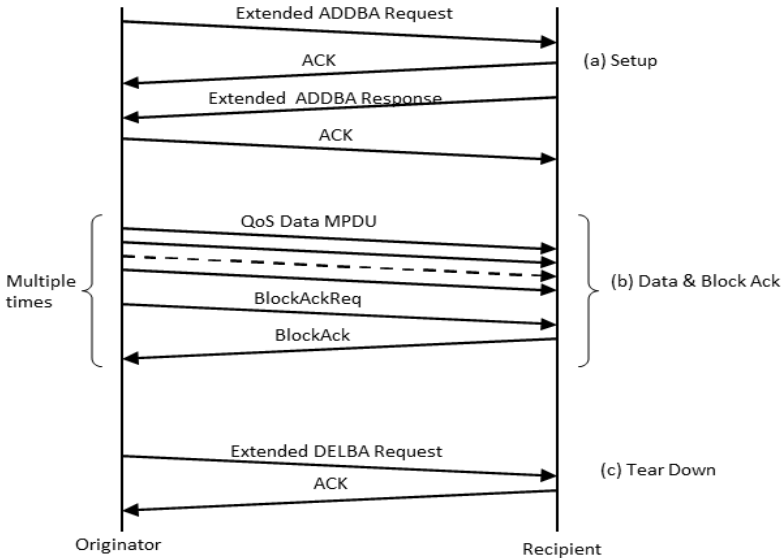


Figure 3.2: Message Sequence Chart for GCR Block Ack. [7]

Following the standard, after an access point has a GCR-Block-Ack agreement with a non-AP station, it initiates the setup of an GCR Block Ack agreement. A much precise message interaction is present in Fig. 3.2 and a description of this procedure is illustrated below.

New ADDBA frames, professionally called Extended ADDBA, are proposed which are similar to those for the unicast Block Ack. But they do have differences. For example, they differ with each other in the presence of an optional group address and the presence of an Extended Block Ack Parameter Set. 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 one flag is defined to indicate the presence of a group address, and this flag always has to be set for the Extended ADDBA Request and Response frames. New frame types are necessary for backward compatibility with legacy Block Ack, which makes the new mechanism capable to the legacy station or system. [21]

There are more important differences between them. First, the delayed Block Ack is not supported in GCR Block Ack. Then, in the GCR Block ACK system the access point is always the originator and the non-AP stations are always the recipients. The reason is that in BSS only the access point is permitted to broadcast to multiple receivers while non-AP stations address their transmissions to the access point.

A recipient reassembles the MSDUs from received MPDUs and then forwards up complete MSDUs required by sequence numbers to higher layer. During this process, a cache which stores the just received MPDUs' sequence number is maintained by the recipient, whose size and that of the buffer are the same with those for Block Ack. Therefore, the buffer size sets limitations on the number of MPDUs able to be sent without acknowledgement, and the cache size should be large enough to trace all these MPDUs. For instance, all the MSDUs are sent up to a higher layer when they are all correctly received.

The IEEE 802.11aa contains some other extensions to the Block Ack mechanisms from the Project IEEE 802.11n Task Group as well. The Block Ack mechanism distinguishes Basic Block Ack from Compressed Block Ack. The former provides support for fragmenting a MSDU into up to 16 MPDUs and acknowledgment for each single MPDU individually. The latter offers no support for fragmentation but is in much smaller size.

According to the standard, the buffer size announced in the ADDBA frames ranges from 0 to 127, but its actual size is in the range of 1 to 128 MPDUs. The maximum length of a MPDU is limited to 2034 octets, which results in a maximal buffer size of 288 *kBytes* for a single agreement in one station. Apart from that, the number of the MSDUs is limited since the recipient has to store cache. This is due to the fact that to offer fragmentation support, space for 16 MPDUs per MSDU must be provided. When A-MSDU, which consists of many MSDUs assemble into one MPDU and whose length can be up to 7935 octets, are used, it will needs a storage of 991 *kBytes*. An A-MSDU is made up of multiple MSDUs aggregated into one MPDU. It is introduced in the amendment 802.11n.

### 3.3 GCR Block ACK Frame Structure

A GCR Block Ack Request may be sent by the originator at any time. Its starting sequence number, sent upon request, points out the oldest MPDU with obvious acknowledgment. If a GCR Block Ack Request is received, the recipient can forward up all complete MSDUs which are received with a sequence number older than the starting sequence number.

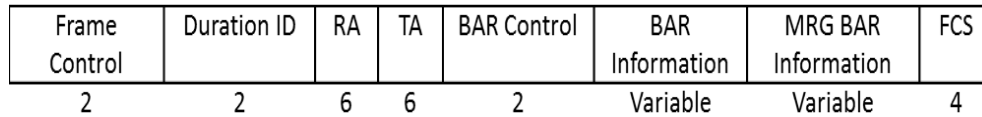


Figure 3.3: The MRG Block Ack Request Frame Format. [7]

GCR Block Ack request frame is sent to the group address. As shown in Fig. 3.3, the BAR Control field is denoted to as the BA Control field, while only the GCR flag is defined in this field of BA but not BAR. The flag for Compressed Bitmap switches over the use of the Compressed and the Basic Block Ack variant. [7] [21] It is the only utility of the Compressed which is useful to the GCR Block Ack, as fragmentation is not suitable to group addresses traffic. Therefore, there is no necessity for the support of fragmentation in the Basic Block Ack variant.

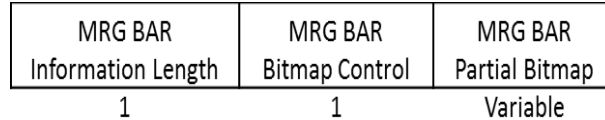


Figure 3.4: The MRG Block Ack Request Information Field Format. [7]

The GCR Block Ack Request frame includes a GCR BAR Information field in which stations are encoded so that they can reply to the GCR Block Ack Request. A recipient is addressed by its association identifier (AID), which is given to the non-AP stations by the access point in the association. When a recipient is addressed, its AID is encoded in a bitmap that contains up to 2008 bits and is organized to have up to 253 octets. [7] [21] [12] An AID corresponds to bit number  $(AID \bmod 8)$  in octet number floor  $(AID/8)$ . The GCR BAR Partial Bitmap starts with this octet and ends with the last octet with a bit set to 1. The GCR BAR Information Length is set to the length of the virtual bitmap plus 2 to express the length of the complete GCR BAR Information field.

The recipient addressed in a GCR Block Ack Request calculates the point-in-time so that it can reply with GCR Block ACK. The calculation formula is shown below:

$$(N + 1) \times SIFS + N \times TXTIME(MRGBlockAck) \tag{3.1}$$

where  $N$  is the position the recipients AID in the list of addressed AID. The smallest addressed AID is at the position when  $N = 0$ .  $TXTIME$  is the transmission time of the GCR Block Ack

frame at the given data rate. Two typical frames exchange mechanisms that use the GCR Block Ack are shown in Fig. 3.7 and Fig. 3.8

The recipient shall transmit the GCR Block Ack frame at the same data rate with that used in the transmission of the GCR Block Ack Request frame, because it is more probable to lose the GCR Block Ack if using a less robust data rate with a higher data rate. Transmission at a more robust data rate takes more time and is the most likely to collide with the GCR Block Ack transmission of another recipient.

The recipient addressed in the GCR Block Ack frame sets the GCR flag in the BA Control field to indicate the presence of a GCR Group Address field and stores group address for the feedback in this field. Detailed structure is shown in Fig. 3.5 and Fig. 3.6. The GCR Block Ack frame is transmitted to the originator. In the Block Ack Information field the recipient indicates the reception of lost for up to 64 MSDUs.

The size of BA Information is either in 130 or 10 octets. Among these octets, the first two are always the starting sequence control. The starting sequence control corresponds to the first bit in the octets, and the rest of the MPDUs sequence numbers correspond to all other bits. In other variant, the Compressed Block Ack, one bit is a MSDU which is completed with 8 octets. Hence, this kind of variant is appropriate only when there is no fragmentation.

Frame Control	Duration ID	RA	TA	BAR Control	BAR Information	MRG BAR Information	FCS
2	2	6	6	2	Variable	6	4

Figure 3.5: The GCR Block Ack Frame Format. [7]

BA Policy	Multi TID	Compressed Bitmap	MRG	Reserved	TID INFO
1	1	1	1	6	4

Figure 3.6: The BA Control Field Format. [7]

The recipient reports that the MPDU is received if the GCR Block Ack send requests for feedback on a MPDU older than the oldest stored in the recipients cache. In the Basic Block Ack variant the recipient only acknowledges the MPDUs beginning from the starting sequence control until the MPDU with the highest sequence number has been received.

If the GCR Block Ack Request frame or one of the GCR Block Ack frames is lost, the originator will send another GCR Block Ack Request to ask for acknowledgment for the same sequence numbers. However, the originator will not address the stations that has successfully sent a GCR Block Ack for the original request in this request. But it will repeat this request unless all the addressed stations give feedback or the lifetime of the MPDU is reached.

### 3.4 GCR Block ACK Operation

In the figure 3.7 and 3.8, two transmission functions are initiated with a block size of three. Fig. 3.7 presents fixed block size and Fig. 3.8 presents variable block size, which the configuration for block size is requested during handshaking. Following the standard definition on continuous packets transmission, the interval between two consecutive packets is SIFS. Then, the system will enter the control frame transmission phase after waiting another SIFS when all the frames are delivered. In this phase, the transmitter send a Block Ack Request frame to each receiver to ask for the receiving status of all the recipients who give feedback by sending Block Ack frame. When a frame is lost, its information will be integrated by the transmitter and then will be retransmitted in the next transmission cycle.

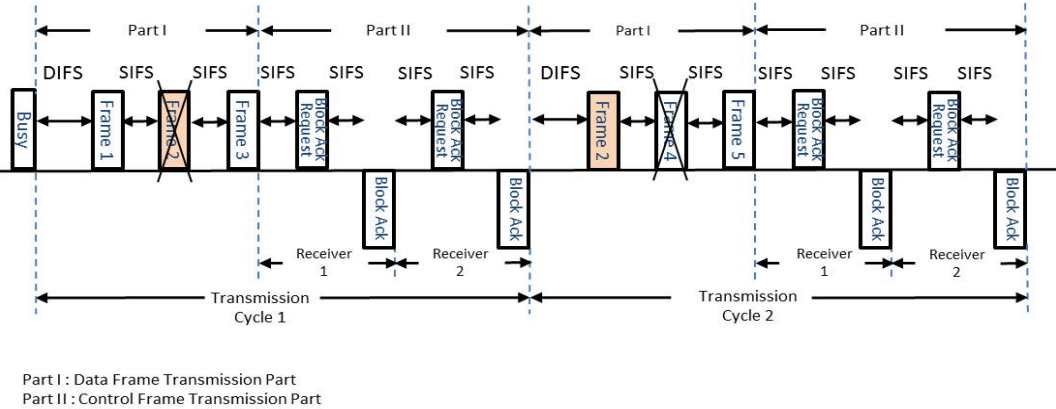


Figure 3.7: GCR Block ACK with Fixed Block Size (Mechanism 1).

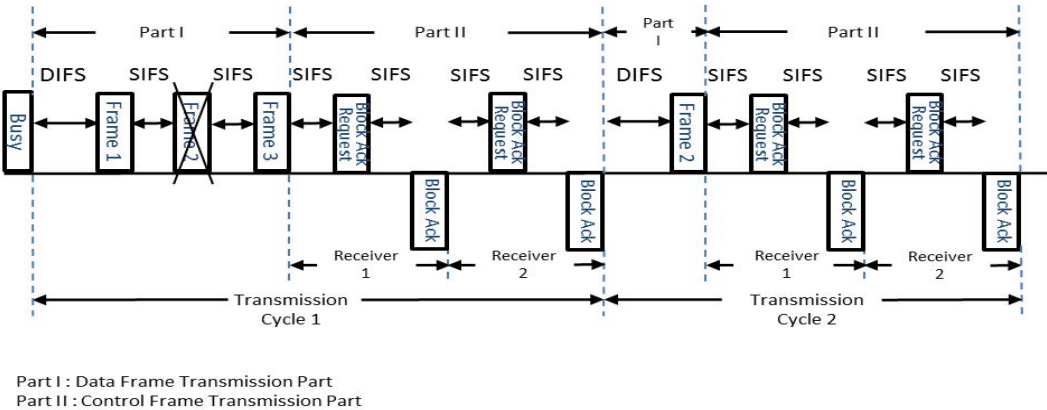


Figure 3.8: GCR Block ACK with Variable Block Size.

The difference between these two mechanism is compared below. In Fig. 3.7, the second packet is lost in the first transmission cycle. When the AP gets the information that the second packet is lost in the first attempt, it will retransmit the old frame lost in the previous cycle (here is packet 2). It will also transmit new packets filled into the whole block size in this transmission cycle in line with the size of the initial block size. While in Fig. 3.8, the lost second frame is retransmitted alone in the next transmission cycle. The block size will be kept in variable based on the lost information we get from the last system cycle.

Moreover, when the fixed block size mechanism is used, the position of retransmitted packet is not defined in the standard. Then, according to previous mechanism, we have another transmission function which is shown in Fig. 3.9. Compared with Fig. 3.7, we can get that, in Fig. 3.7, the lost packet is retransmitted without position changed in the block. The new packet will be filled into the position where the packet is discarded or successfully transmitted in the last system cycle. While in Fig. 3.9, retransmitted packets will replace the position of packets which is discarded or successfully transmitted in the previous transmission cycle, sequentially. The new packet will fill into the left position in the block. Even though, it has not impact on the system operation, system throughput and system probability. But it indicates a result that system will not go into some previous transmission stats mentioned in Fig. 3.7. Much detailed illustration will be given in next chapter.

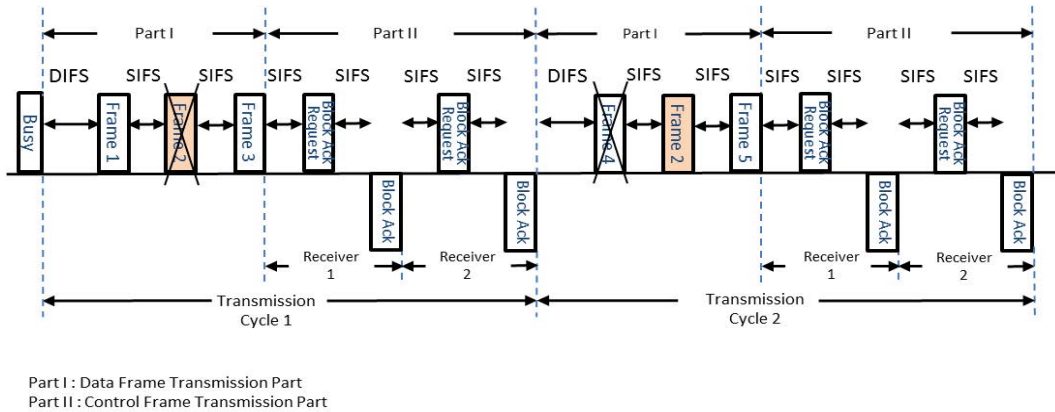


Figure 3.9: GCR Block ACK with Fixed Block Size (Mechanism 2).

### 3.5 Chapter Summary

In this chapter, we provide the description of GCR Block ACK mechanism in the new amendment. We firstly present the application scenario of GCR Block ACK to entirely describe this new mechanism service feature. Then, we discuss the setup procedure and show the different configuration between the new service and old one. Even though we consider that the setup procedure of GCR Block ACK is always successful in the model mentioned in chapter four, we still need to show the system operation function here. In addition, we present the frame format which is used in new mechanism in order to precisely calculate the transmission duration for each packet during model analysis. Finally, we show the GCR Block ACK operation procedure. The model in chapter four is build based on the operation we mentioned here.

# Chapter 4

## Performance Analysis

In the following, we will present our model of GCR Block ACK mechanism. The model assumption and analysis procedure will be given firstly to help understand our model. Then we will describe our model step by step. Finally, the conclusion will be given.

### 4.1 System Model, Assumption and Analysis Procedure

#### 4.1.1 System Model

According to the GCR Block ACK application scenario we described in Fig. 3.1 in Ch. 3, our model is build based on an abstracted part of this scenario which is shown in Fig. 4.1. The sys-

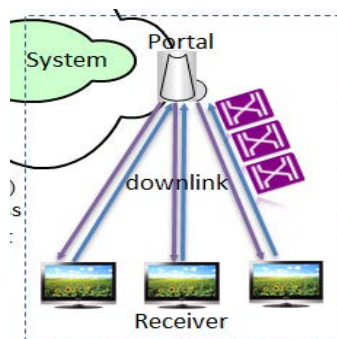


Figure 4.1: System Model Scenario.

tem model contains one access point which will control the whole system transmission procedure. Besides, there are more than one stations exchanging information with access point in the system. The stations will request the transmission first. Then, access point will groupcast a block of packets to the stations which are successfully connected with it, simultaneously. The stations will feedback the reception status one by one after receiving the packets. Moreover, the access point will retransmit the lost packet in accordance with the feedback from stations. In every transmission cycle, the block size is kept in fixed.



### 4.1.2 Assumptions

In this thesis, we will analyze the saturated throughput of GCR Block ACK mechanism. The model will be built based on two important aspects for packet transmission. One is that we take into consideration the memory feature of the channel and extend it into wireless group-cast operation in line with the standard. The other one is that we will study different system status from the system's point of view during the entire transmission operation cycles. The impact of each packet in the whole block on the system retransmission will be considered.

We conclude the assumption above as:

- channel is provided with memory property;
- the stations share identical and independent channel conditions, which is relevant in the time domain and independent in the space domain [17];
- the number of station keeps fixed during transmission, no other station will ask connection during model operation, each generates the traffic with the same priority, and always have packets queued in the transmission buffer.
- access point is the originator in this model which can obtain the transmission permission whenever it needs;
- the handshake between station and access point is always successful.

### 4.1.3 Analysis Procedure

As we mentioned in previous studies in chapter two, the system throughput,  $S$  can be obtained from the average successful transmission payload divide average transmission duration. Because of the reason that the proposed model is based on fixed block size transmission mechanism, the duration in each transmission cycle is the same. Then average successful transmission payload in the system steady state during one transmission cycle becomes the core point. The system throughput can be expressed as :

$$S = \frac{P \times L}{T} \quad (4.1)$$

where  $T$  is the fixed system transmission duration in one transmission cycle,  $P$  is the probability of successful system transmission and  $L$  is the payload during transmission.

According to the 802.11 standard, the reason for successful system transmission can be divided into two parts: one is the successful channel access, and the other is the successful packets delivery in this accessed channel. We mainly focus on the second part. A short introduction to the channel access model which has been indicated in existing literature has been introduced in previous studies in Ch. 2.

In the first part we will discuss the transition model from the system's point of view under the condition of successful channel access, which is made by retransmission status of each packet in the block. In this part, we will first analyze the model for one packet retransmission transition

model, in which the system will broadcast one packet to  $N$  stations in each transmission cycle. We will take this model as an introduction to our following research. Then, based on the 802.11aa block ACK mechanism, a Markov model will be made to analyze the transmission channel feature with multi-receivers. [1]

In the second part, a system model will be built upon the system retransmission states. Continuous packets property will be take into account. Finally, we will combine the system model and channel model together to make a retransmission model for a block of packets.

## 4.2 Packet Transmission Probability

In this section, we will begin with the analysis of a simple transition model for one packet retransmission. Then, a markov model based on GCR Block ACK will be made. And finally, a retransmission system model will be obtained according to the two previously mentioned parts.

### 4.2.1 Retransmission Transition Model for One Packet

Let the transmitter broadcast only one packet to  $N$  stations in each transmission cycle. We assume each station is independent from each other. If the packet is not successfully received by all the stations, it will be retransmitted in the next cycles until the max system retry limit is reached. The retry counter of the system will be increased with each retransmission. However, the packet will be discarded when the system comes to the max retry limit and still does not successfully transmit this packet.

In order to simplify the analysis, as an example, we set the max retry limit of the system as 1. Let  $i$  represent the different state of this system. According to the previous system process description, there are three states in this transmission model:

1. Packet discard state: after the first retransmission. If the packet is still not successfully transmitted, it will be discarded directly.
2. Packet first time retransmission state: After the packet's first time transmission. If it is not successfully transmitted, the retry counter of the system will be increased by one each time. The system will come to the first time retransmission state and retransmit this packet again.
3. Packet successful transmission state: the packet is successfully received by all the stations in this transmission cycle.

We use 0, 1 and 2 to represent *packet discard*, *packet first time retransmission* and *packet successfully transmitted* state, respectively. And a transition model is present in Fig. 4.2:

According to the previous assumption, we assume the stations experience identical independent channel condition. Let  $p$  represent the probability of each station's packet failure reception. Provided a system with  $N$  stations, the probability that there is at least one station failing to receive

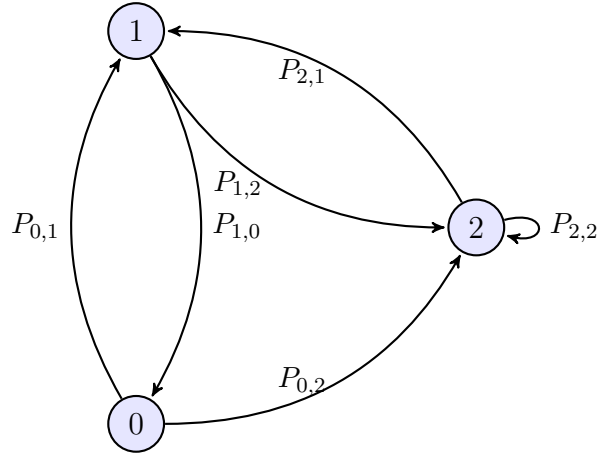


Figure 4.2: Retransmission State Transition Model

the packet,  $P_e$ , can be expressed as:

$$P_e = \sum_{n=1}^N \binom{n}{N} p^n (1-p)^{N-n}. \quad (4.2)$$

$P_e$  also states that the system unsuccessfully transmits this packet during this transmission cycle. We use  $P_s$  to denote the success transmission of a packet from the system's point of view:

$$P_s = (1-p)^N. \quad (4.3)$$

The transition matrix for the retransmission model in Fig. 4.2 is given by:

$$\mathbf{P} = \begin{matrix} & \begin{matrix} 0 & 1 & 2 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} & \left\| \begin{array}{ccc} 0 & P_e & P_s \\ P_e & 0 & P_s \\ 0 & P_e & P_s \end{array} \right\| \end{matrix} \quad (4.4)$$

### 4.2.2 Analysis of Block ACK Transmission Channel Model

In this part, we discuss the system model for a block of packets. To analyze the system's performance at the receiving packet level with time correlated non-ideal channel, a 0 – 1 two state transition Markov chain should be indicated firstly. In Fig. 4.3, state 1 means that the packet has been received correctly by the receiver while 0 means not. The transition probability  $p_{ij}$ , where  $i, j \in \{0, 1\}$  is obtained from experimental results. This value indicates the probability of another failure or success reception based on the status of the previous transmission.

In order to combine the relevance property of continuous transmission packet in the time domain and the independent property of different stations in the space domain, we present a four state Markov transition model for two receivers as an example of multi-receiver transition situation in Fig. 4.4. Following the same principle of 0 – 1 transition Markov model, each state in our extend-

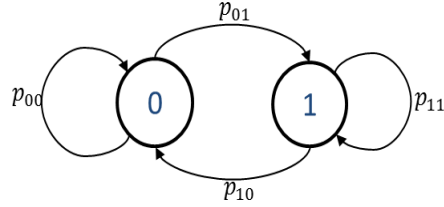


Figure 4.3: System Model by Two State Markov Chain.

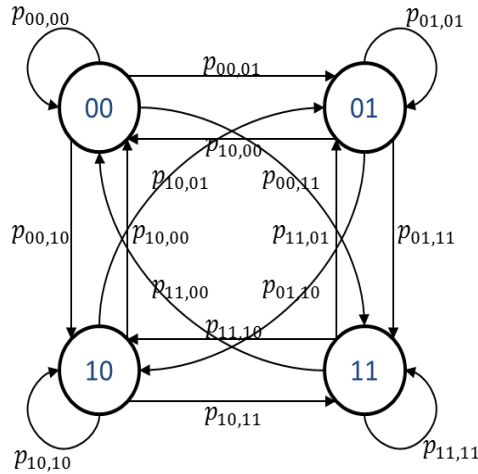


Figure 4.4: System Model by Four State Markov Chain.

ed model also represents the receiving status of the current transmission frame for two stations. Moreover, the first element in the state represents the status of the first station, while the second element represents that of the second station.

When we consider the packet index in a block, we have some assumptions below. Let  $a_j$  represent the status of  $j$ th frame in the block received by the group members, which is also the state in Fig. 4.4. Use  $c_n^j$  to express the receiving state of the same frame for different receivers, where  $n$  represents the number of station.  $c_n^j$  can be expressed as

$$c_n^j = \begin{cases} 1 & \text{frame received} \\ 0 & \text{frame lost} \end{cases} \quad (4.11)$$

Moreover,  $a_j$  can be expressed as  $a_j = (c_1^j, c_2^j, \dots, c_N^j)$ , where  $N$  stands for the maximum number of receivers. In order to illustrate this transition process, we define  $a_{j+1}$  as the next transition state in this model. It can be denoted as  $a_{j+1} = (c_1^{j+1}, c_2^{j+1}, \dots, c_N^{j+1})$ . According to the definition of  $a_j$  and  $c_n^j$ , the transmission status relationship between access point and stations from the system point of view is indicated in Fig. 4.5. In this figure, we have block packets transmitted from access point as  $\{a_1, a_2, \dots, a_j, \dots, a_{B_{max}}\}$ , and have  $\{c_1^j, c_2^j, \dots, c_i^j, \dots, c_N^j\}$  for each  $a_j$ . The figure still describe a channel model example, where  $B_{max}$  can be set as infinite.

Furthermore, each  $a_j$ , which broadcast from the access point, has a group of corresponding receptions status for each station. The reception state of each station for this packet is obtained based on each station's channel condition. Moreover, the continuous packets,  $c_n^j$ , for the same

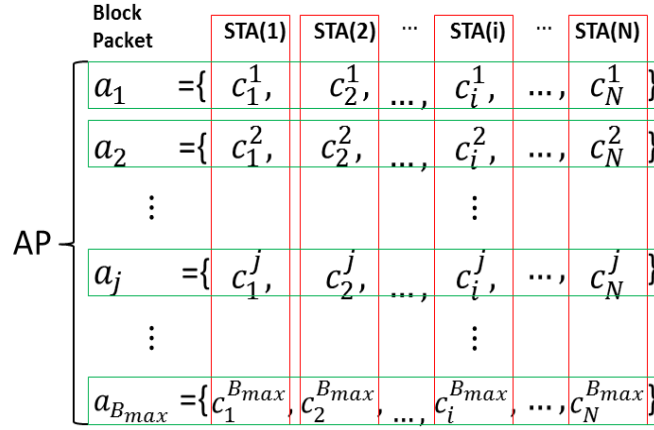


Figure 4.5: Block Transmission Relation between AP and Stations.

station follow channel memory characteristics. The receiving station in the system is  $N$  and the transmission block size is  $B_{max}$ . Since it is assumed that transmission channels of receivers are spatially independent of each other, the transmission status between each station does not affect with each other. As the channel memory property is taken into account, the reception status of another packet for one station in the system is based on the status of the previous transmission. From the perspective of system, transition probability of  $p_{a_1, a_2}$  can be obtained by multiplying transition probability for the corresponding receiving packet of each single receiver together. We can get  $p_{a_1, a_2} = \prod_{i=1}^N p_{c_i^1, c_i^2}$ . Then, we can calculate any transmission status of two continuous packets in the block for the system as:

$$p_{a_j, a_{j+1}} = \prod_{i=1}^N p_{c_i^j, c_i^{j+1}} \quad (4.12)$$

where  $p_{c_i^j, c_i^{j+1}}$  is the transition probability in the two state Markov transition matrix of each station. Correspondingly, each station has its own independent transmission channel characteristics,  $p_{c_i^j, c_i^{j+1}}$  of each channel is different. In order to simply the example of system transmission channel we mentioned in Fig. 4.5, we assume each station channel follow the independent and identical distribution property. Then, we set the transition probability for the two state markov transition model mentioned in Fig. 4.3 as:  $p_{10} = 0.001, p_{11} = 0.999, p_{00} = 0.97$  and  $p_{01} = 0.03$ , which is obtained from the experience. [15] Then we can get the transition Markov matrix for the model in Fig. 4.4, which describe the transmission channel property with two stations under channel

memory characteristics, given by:

$$\mathbf{P} = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{cccc} 00 & 01 & 10 & 11 \\ \left\| \begin{array}{cccc} 0.94090 & 0.02910 & 0.02910 & 0.00090 \\ 0.00097 & 0.96903 & 3.000e - 05 & 0.02997 \\ 0.00097 & 3.000e - 05 & 0.96903 & 0.02997 \\ 1.000e - 06 & 0.00099 & 0.00099 & 0.99800 \end{array} \right\| \end{array} \quad (4.13)$$

Moreover, let  $\mathbf{e}_N = [1, 1, \dots, 1]_N$ , where  $N$  means the length of vector  $\mathbf{e}$ . Based on the transition matrix  $\mathbf{P}$ , we can calculate the transition steady state in Fig. 4.4 following the equation below:

$$\begin{cases} \boldsymbol{\pi} \cdot \mathbf{P} = \boldsymbol{\pi} \\ \boldsymbol{\pi} \cdot \mathbf{e}_N = 1 \end{cases} \quad (4.14)$$

According to the Block ACK mechanism, the system will determine the current packet retransmission status based on the reception from each station. If any receivers replies with reception failure on corresponding packet, the system will record the unsuccessfully transmitting of this packet and retransmit it in next transmission cycle. Let  $A_j$  represent the retransmission determination from the system, where  $j$  denotes this packet is at the  $j$ th position in the block from the system point of view. Then, we use  $A_j = 0$  to denote that the packet at the  $j$ th position in the current transmission block should be retransmitted in the next system cycle. And use  $A_j = 1$  to denote the successful transmission of this packet. Correspondingly,  $A_j$  is the intersection of reception result of  $j$ th packet in the block from all the stations, and it can be expressed as:

$$A_j = \bigcap_n c_n^j = \begin{cases} 1, & \text{packet successful transmission} \\ 0, & \text{packet should be retransmission} \end{cases} \quad (4.15)$$

Therefore, we can obtain the system transition status from the receiving states of different stations under corresponding channel model. Apparently, from the example of four state markov channel model mentioned in Fig. 4.4, we can achieve the same result of  $A_j$  from different transition states, such as  $a_j = \{0, 0\}$ ,  $a_j = \{0, 1\}$  and  $a_j = \{1, 0\}$ . Then, the transition probability from  $A_j$  to  $A_{j+1}$  should be the summation of all different receiving status from  $a_j$  to  $a_{j+1}$ .

Let  $P_{A_j, A_{j+1}}$  be the transition probability for receiving status from system's point of view. According to the definition of  $A_j$ , we can separate this transition states into four cases, as shown in Table. 4.1.

Take  $A_j = 0, A_{j+1} = 1$  as an example. The system state under this condition is that there is at least one station failing to receive the first packets from the system but all stations successfully receive the second packet. As we consider an identical and independent channel transition model for each station, the transition probability from  $A_j$  to  $A_{j+1}$  can be obtained by the summation of the probability corresponding to all the possibilities that results in  $A_j = 0$  and  $A_{j+1} = 1$ . We can

Table 4.1: Four Cases of Transition Probability  $P_{A_j, A_{j+1}}$ 

$A_j = 0, A_{j+1} = 1$	$P_{A_j, A_{j+1}} = \sum_{n=1}^N \binom{N}{n} (p_{0,1})^n (p_{1,1})^{N-n}$
$A_j = 1, A_{j+1} = 0$	$P_{A_j, A_{j+1}} = \sum_{n=1}^N \binom{N}{n} (p_{1,0})^n (p_{1,1})^{N-n}$
$A_j = 0, A_{j+1} = 0$	$P_{A_j, A_{j+1}} = \sum_{n_1} \sum_{n_2} \sum_{n_3} \sum_{n_4} \binom{N}{n_1, n_2, n_3, n_4} (p_{0,0})^{n_1} (p_{0,1})^{n_2} (p_{1,0})^{n_3} (p_{1,1})^{n_4} \text{ where } \begin{cases} n_2 + n_4 < N \\ n_3 + n_4 < N \\ n_1 + n_2 + n_3 + n_4 = N \end{cases}$
$A_j = 1, A_{j+1} = 1$	$P_{A_j, A_{j+1}} = (p_{1,1})^N$

$A_j = 0$  represents this frame has to be retransmitted by the system.

$A_j = 1$  represents this frame has been successfully delivered.

$p_{*,*}$  represents transition probability in a two state Markov chain.

get the other transition probability of system status in accordance with the same function.

Since the number of frames in a block at the each system transmission cycle is  $B_{max}$ , let  $X_j$  represent the random variable of the transmission status of the  $j$ th packet in the block from the system's point of view. Then, we have  $A_j$  as the value of each corresponding random variable. Following the definitions, the probability of transmitting this block of frames can be calculated through  $B_{max}$  steps transition probability, and it can be expressed as:

$$\begin{aligned}
 P_{A_1, A_{B_{max}}}^{(B_{max})} &= Pr\{X_1 = A_1, X_2 = A_2, \dots, X_{B_{max}} = A_{B_{max}}\} \\
 &= \pi \times \prod_{j=1}^{B_{max}} P_{A_j, A_{j+1}}
 \end{aligned} \tag{4.16}$$

where  $\pi$  is the initial state of this  $B_{max}$  step transition process. What's more, a  $\pi$  of a four state transition Markov matrix can be obtained from the matrix in Eq. (4.13).

### 4.2.3 Retransmission System Status Model

In this part, we discuss the system retransmission model based on that mentioned in the Fig. 4.2. According to our previous discussion, we establish a system channel transmission model based on GCR Block ACK operation. In this model, we not only study the transmission characteristics of a group of consecutive packets transmitting in a channel with memory property, but also get the corresponding transmission probability expression. In the following, we will combine the system retransmission model and transmission channel model together to present GCR Block ACK mechanism with fixed block size.

Let  $b_j^i$  stand for the retransmission state of the  $j$ th packet in the block which is transmitted in the  $i$ th system cycle, where parameter  $j$  represents the position of this packet in the block and  $i$  represents the number of transmission cycle. In Fig. 4.2, we give the  $Retry\_limits = 1$ , and use  $Retry\_limits + 1 = 2$  to stand for the system successful transmission status while 0 means the packet is discarded by the system. As the  $B_{max}$ , which is the length of the block size, is considered as 1 in Fig. 4.2, we can get a normal retransmission status table as follows:

From the system point of view, each packets in the block which is transmitted in the previous transmission cycle has impact on the current packets' retransmission time and position. However,

Table 4.2: One Frame Retransmission Transition States

Condition	Source	Transmission Status	Destination	Transmission Probability
$b_j^i \in \{0,1,2,\dots, \text{Retry\_limits} - 1\}$	$b_j^i$	0	$b_j^i+1$	$P_e$
$b_j^i = \text{Retry\_limits}$	$\text{Retry\_limits}$	0	0	$P_e$
$b_j^i = \text{Retry\_limits} + 1$	$\text{Retry\_limits} + 1$	0	1	$P_e$
$b_j^i \in \{0,1,2,\dots, \text{Retry\_limits} + 1\}$	$b_j^i$	1	$\text{Retry\_limits}+1$	$P_s$

the transmission status of packets in different system cycle is independent with each other, and it has nothing relation with channel memory property. Then, we can get the transmission probability is only depend on the transmission status which happens in current transmission cycle. Based on the one packet retransmission model in Fig. 4.2, state 0 also stands for a packet is discarded. Since *Max Retry Limit* is indicated as 1, we use 1 to represent the packet is in the first retransmission and let 2 stand for that the packet is successfully transmitted in the current round. As previously discussed, we mentioned two different GCR Block ACK operations based fixed block size. One of them is transmitted without packet position changed in the block. The new packet will be filled into the position where the packet is discarded or successfully transmitted in the last system cycle. The other mechanism is that the block transmission process is the same as transmitting in a queue. Retransmission packets will replace the position of packets which is discarded or successfully transmitted in the previous transmission cycle, sequentially. Here, we can show this system retransmission transition model with block size of two in Fig. 4.6 with the first transmission mechanism.

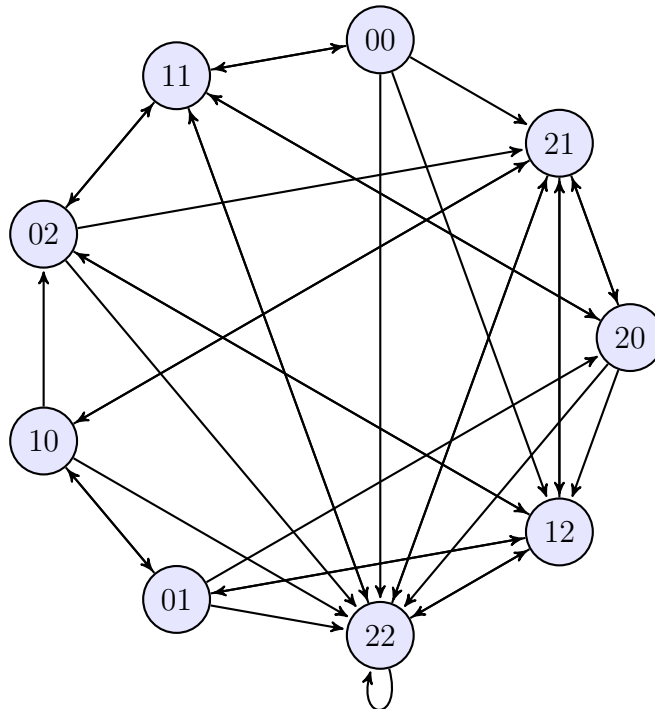


Figure 4.6: Retransmission State Transition Mode 1

In this model, each state represents the transmission status of the current transmission block for N stations. Moreover, the first element in each state stands for the retransmission status of



the packet at the first position in the block, while the second element represents that of the second packet. For example, in Fig. 4.6, state 00 means during this block of packets transmission, both packets are discarded by the access point; 01 means that the packet at the first place in this block is discarded and the one at the second place is retransmitted in the first time during block transmission.

The transition probability can be obtained from Eq. (4.22), and one example is shown in Fig. 4.7.

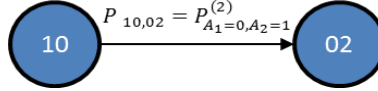


Figure 4.7: Transition Probability from State 10 to State 02.

From the transition model, following different transmission status, we have the same transition result as shown in Table. 4.2 from source to destination for each packet in the block. However, we consider a two consecutive packets transmission process based on the channel memory characteristics. The corresponding transmission probability can be obtained from the system channel model mentioned in previous section. The transition result and transition probability is illustrated in Table. 4.3, respectively.

Table 4.3: Transmission State for Each Packet and Corresponding Probability

Source	Transmission Status	Destination	Transmission Status	Transition Probability
$b_j^i \in \{0,1,2,\dots, \text{Retry\_limits} - 1\}$	0	$b_j^i+1$	0,0	$P_{A_1=0, A_2=0}^{(2)}$
$b_j^i = \text{Retry\_limits}$	0	0	0,1	$P_{A_1=0, A_2=1}^{(2)}$
$b_j^i = \text{Retry\_limits} + 1$	0	1	1,0	$P_{A_1=1, A_2=0}^{(2)}$
$b_j^i \in \{0,1,2,\dots, \text{Retry\_limits}\}$	1	$\text{Retry\_limits}+1$	1,1	$P_{A_1=1, A_2=1}^{(2)}$

In Table. 4.3, based on the previous system configuration that the transmission block size is two, we separate transmission status into four cases from the access point of view. Apparently, we analyze the transition model in Fig. 4.6 from two points of view. One of them is from the transition result of each packet in the block. And the other one is transmission status of these continuous packets which is contributed to this transition result based on system channel model. Therefore, take transition model in Fig. 4.6 as an example, we have four different transmission status for each  $b_1^i, b_2^i$ .

We attribute these four transmission results as  $\{0, 0\}$ ,  $\{0, 1\}$ ,  $\{1, 0\}$  and  $\{1, 1\}$ . The transmission status  $\{0, 0\}$  not only represents that for each packet in the block there are at least one station unsuccessfully receiving the packet; but also indicates that the next state of each packet should be discarded or retransmission.

Then through the expression Eq. (4.22) and Table. 4.1, we can get each transmission probabil-



retransmission transition model in Fig. 4.8.

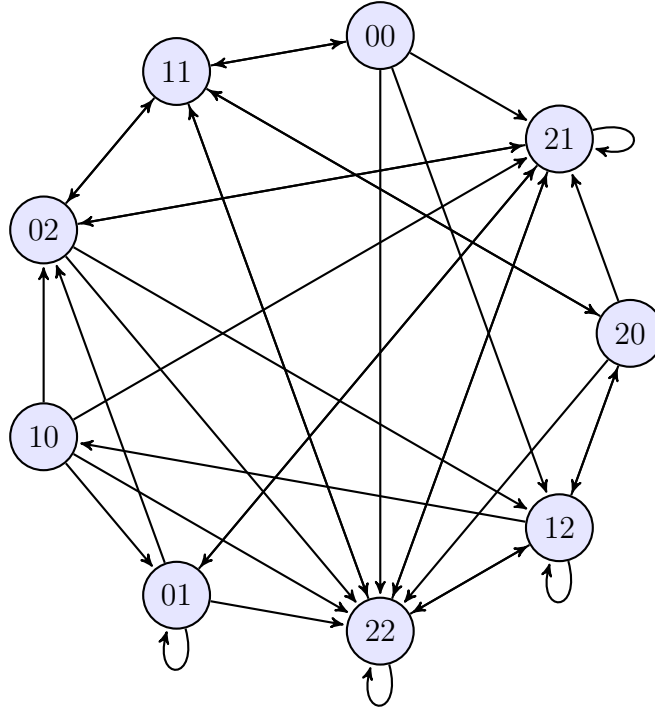


Figure 4.8: Retransmission State Transition Mode 2

According to the transition model, we find that each packet follow the same transition schedule mentioned in previous discussion. Then, in Tab. 4.4, we only figure out the different transition situation from the first transmission technique. Then, we can get the transition table in Table. 4.4.

Table 4.4: Transmission State for Each Packet and Corresponding Probability (Technical 2)

Source	Transmission Status	Destination
$b_j^i \in \{0,1,2, \dots, Retry\_limits - 1\}$	0	$b_j^{i+1}$
$b_j^i = Retry\_limits$	0	0
$b_j^i = Retry\_limits + 1$	0	1
$b_j^i \in \{0,1,2, \dots, Retry\_limits\}$	1	$Retry\_limits+1$

Source	Transmission Status	Destination
$b_j^i \in \{0, R\_l + 1\},$ $b_{j+1}^i \in \{1, 2, \dots, R\_l - 1\}$	0,0	$b_j^i + 1, 1$
	0,1	$b_j^i + 1, R\_l+1$
	1,0	$R\_l+1, 1$
	1,1	$R\_l+1, R\_l+1$
$b_j^i \in \{0, R\_l + 1\},$ $b_{j+1}^i = R\_l$	0,0	0,1
	0,1	0, $R\_l+1$
	1,0	$R\_l+1, 1$
	1,1	$R\_l+1, R\_l+1$

Transmission Status	Transition Probability
0,0	$P_{A_1=0, A_2=0}^{(2)}$
0,1	$P_{A_1=0, A_2=1}^{(2)}$
1,0	$P_{A_1=1, A_2=0}^{(2)}$
1,1	$P_{A_1=1, A_2=1}^{(2)}$

Furthermore, we achieve the transition matrix in Eq. (4.19)



where  $P$  represents the transition matrix of system model. One example of  $P$  is matrix 4.19

Let  $S$  be the normalized network throughput defined as the ration between the average packet payload size and the length of total time used for one transmission. Let  $P_{ca}$  represent the probability of having successful channel access. Due to the complexity of overlapping BSS, we neglect the contention between different AP. Then the network throughput  $S$  can be expressed as

$$S = \frac{P_{ca} \times E\{payload\}}{T\{slot\}} \quad (4.23)$$

As shown in Fig. 3.7, a transmission cycle is roughly divided into two parts : the data frame transmission phase and the control frame transmission phase. Following this pattern, the total length of one transmission time  $T\{slot\}$  can be represented by the sum of two components: the transmission time of delivered the packets in one block (Part I) and the control period when group members exchange BAR and BA (Part II). According to the standard, based on the GCR Block ACK with fixed block size transmission working with  $N$  receiving station, the  $T\{slot\}$  can be expressed as

$$T\{slot\} = 4 \times (SIFS + T_F + T_{plcp}) + N \times (2 \times T_{plcp} + T_{BAR} + T_{BA}) + (2 \times N - 1) \times SIFS + DIFS \quad (4.24)$$

where the frame transmission duration is separated into the summation of physical header and frame body these two parts (more details in Table. 4.5).

Table 4.5: Payload and Header Transmission Time

$SIFS$	$DIFS$	$Payload$	$T_{plcp}$	$T_{BAR}$	$T_{BA}$	$T_F$
16	34	$Frame\_size \cdot B_{max}$	$PHY\_Header/r_c$	$(BAR\_CF + BAR\_IF)/r_c$	$(BA\_CF + BA\_IF)/r_c$	$Frame\_size/r_d$

$SIFS$ : the duration between different step in one system cycle	$B_{max}$ : the initial block size
$DIFS$ : the duration between different system cycles	$Frame\_size$ : the size of one data unit
$T_{plcp}$ : time spends on physical header for each frame	$r_c$ : the transmission rate of control frame
$T_F$ : time spends on each frame	$r_d$ : the transmission rate of data frame
* $CF$ : the control part in a <i>Block_Ack_Request</i> or <i>Block_Ack</i> frame	
* $IF$ : the information part in a <i>Block_Ack_Request</i> or <i>Block_Ack</i> frame	
$T_{BA}$ : transmission time delay on <i>Block_Ack</i> by receivers	
$T_{BAR}$ : transmission time delay on <i>Block_Ack_Request</i> by originator (e.g. Access point)	

We use  $E\{payload\}$  to represent average transmission payload. Let  $F$  be a random variable equalling to the number of frames that system successfully transmit in one system cycle. Then, the  $E\{payload\}$  can be obtained from the multiplying between system payload and average total number of  $F$  in one system cycle.

$$E\{payload\} = payload \times E\{F\} \quad (4.25)$$

The average total number of  $F$  is given as:

$$E\{F\} = \sum_{n=1}^{B_{max}} n \times P_c(n) \quad (4.26)$$

where  $n$  represents the number of packet successfully transmitted in current system cycle. The maximum value of this summation should be  $B_{max}$  which is defined through the system configuration.  $P_c(n)$  represents the probability that  $n$  packets are successfully transmitted in current system cycle.

As we get the transmission steady state  $\pi_{B \cup_n^{k_n}}$ , the probability of successfully transmission  $n$  packets in the block can be obtained from the summation of these packets appeared in  $n$  different position in this block. Eq. (4.27) shows how  $P_c(n)$  can be calculated.

$$P_c(n) = \sum_{k_1=1}^{B_{max}-(n-1)} \sum_{k_2=k_1+1}^{B_{max}-(n-2)} \cdots \sum_{k_n=k_{n-1}+1}^{B_{max}-(n-n)} \pi_{B \cup_n^{k_n}} \quad (4.27)$$

### 4.3 Chapter Summary

In this chapter, a Markov model for throughput analysis is proposed for the GCR Block ACK, which is one of the service features in the new amendment 802.11aa, based on fixed block size transmission mechanism. This model is made up with two models. One of them is denoted as channel model which describes the transmission characteristics of the block from transmitter's point of view. We not only design a model for the new system scenario of GCR Block ACK which proposed in this amendment, but also take channel memory property into consideration to extend our model. The other one is indicated as system model which mentions the status transition between different system states. Finally, we combine these two models together to get overall GCR Block ACK model. In addition, an expression is derived to calculate the performance of system throughput in the last part of the section.

# Chapter 5

## Numerical Results and Discussion

In this section, the performance of 802.11aa is evaluated through numerical results based on our analytical model. Then, the factors that influence the performance of the GCR Block ACK scheme on fixed block size is investigated. These factors include the initial transmission block size, re-transmission cycles and the number of receivers. We will show our previous numerical results [1] on Block ACK with variable block size and compare the results obtained from the fixed block size model in this thesis. Our purpose is to compare these two different techniques and try to indicate the advantage and disadvantage of each mechanism based on our results. The application scenario we consider here is infrastructure based which has been mentioned in previous section. The size of MSDU is fixed, which is 2034 bytes. Other parameters are configured as shown in Tab. 4.5 and Tab. 5.1.

Table 5.1: System Parameters used to obtain Numerical Results

Frame_Size	2304bit
MAC Header	16 bit* $B_{max}$
PHY Header	128bit
BAR_CF	16bit
BA_CF	16bit
BAR_IF	64bit
BA_IF	16* $B_{max}$
$r_c$	6Mbit/s
$r_d$	54Mbit/s

### 5.1 Throughput

In this first section, the network throughput for various number of stations and initial block size is investigated when the system maximum transmission retry limit is configured as four. In Fig. 5.1, the system throughput result is obtained from variable block size transmission mechanism. As the figure shows, the throughput decreases with the increasing of station numbers. This result is because that, overhead for handshaking and information exchanging on block receiving status

in the receiver part will increase. Moreover, the Block ACK mechanism we modeled is based on the scenario that the system has to guarantee that the packets can be retransmitted successfully to the receivers who lost the packet until the packets' retry limitation. For an increasing number of stations, this retransmission operation will need much more time to complete this objective. Furthermore, the decreasing rate of system throughput is obviously shown in Fig. 5.1. However, the changing on block size does not have a great impact on the curves. The reason is that, even though the initial block size is different, system will tend to transmit a statistical fixed length of block size after several number of transmission cycles because of the identical channel property. On the contrary, in accordance with the memory property of transmission channels, larger number of stations make this reason much likely to be achieved. The most important point we get from this figure is the case of the affection of smaller initial block size on the throughput with larger number of stations. The reason is that, shorter initial block size will reduce the probability of losing frame with the memory characteristics of transmission channel.

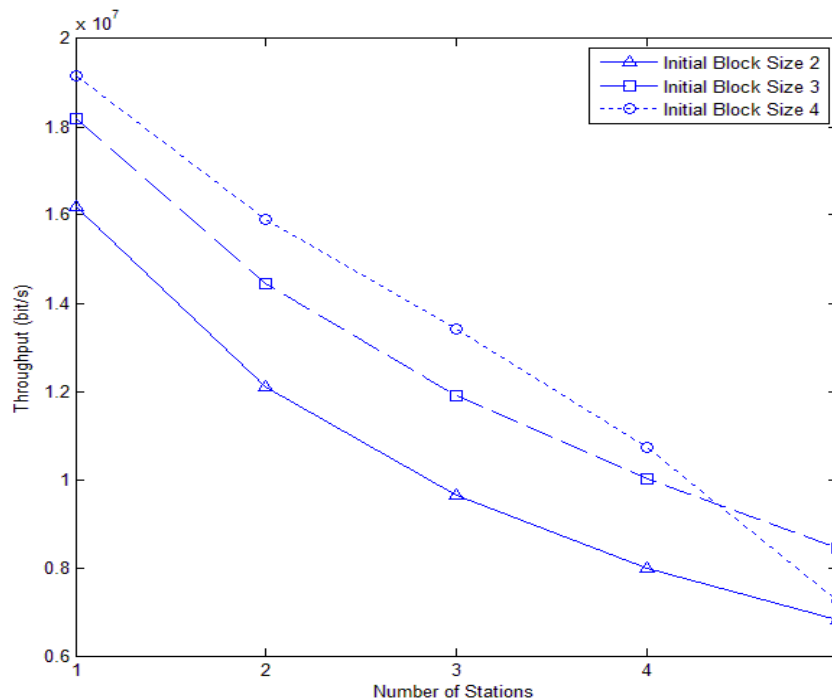


Figure 5.1: Throughput versus Number of Stations based on Variable Block Size. [1]

In Fig. 5.2, the system throughput result is achieved from a fixed block size transmission mechanism. From the figure, the throughput also decreases with growing number of receivers. This result is acquired with the same reason as which mentioned in the throughput analysis with variable block size. It is interesting that the decreasing rate of the system throughput is not as obvious as the one shown in Fig. 5.1. However, the changing on block size has a great impact on the curves which is totally different from variable block transmission mechanism. The reason is that for fixed block transmission mechanism, the system will keep transmitting the same length block of packets to the receivers. Even though channel memory property will influence the reception status of each station, the system will still keep transmitting the block of packets in the same size in the next transmission cycle until the packet attains its own retry limit. However, it is different from the variable one that the throughput with different initial block size have a great distinction with each other with growing



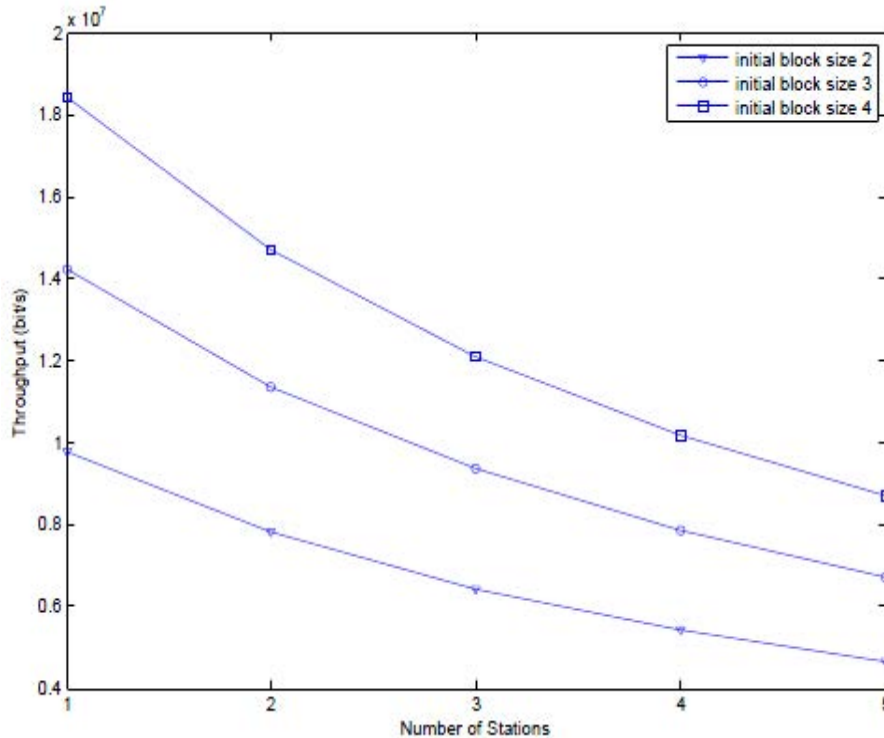


Figure 5.2: Throughput versus Number of Stations based on Fixed Block Size.

number of stations. Smaller initial block size offers a poorly efficiency on throughput, no matter the number of station is one or much larger. The reason is that even though the character of channel memory will lead to a much easier failure reception on larger block of packets, its property also indicates that when the first packet is successfully received by the receiver, the other packets in the block will also be received successfully following corresponding channel feature. The most important point of comparison is that variable block size mechanism starts with a higher throughput that fixed one under small number of station. With the growing number of stations, fixed block size mechanism tends to beyond the throughput performance of variable one. It is because that under small number of stations, channel memory feature, which is considered as an important and unique (from some points of view) factor, make an impact on the system performance. This factor makes fixed block size mechanism transmit with an higher probability on failure packets reception. With the increasing number of stations, the number of stations gradually becomes a much important factor which influences the system throughput performance. The reason is that the network with more stations have much more packets reception statuses. Instead of efficiently transmitting more packets in valid block space, variable block size mechanism cannot improve the system throughput performance.

Finally, we can get that the advantage of fixed block size mechanism in larger station transmission scenario is obviously better than variable one. However, when the number of station is small, variable mechanism will be a much better choice.

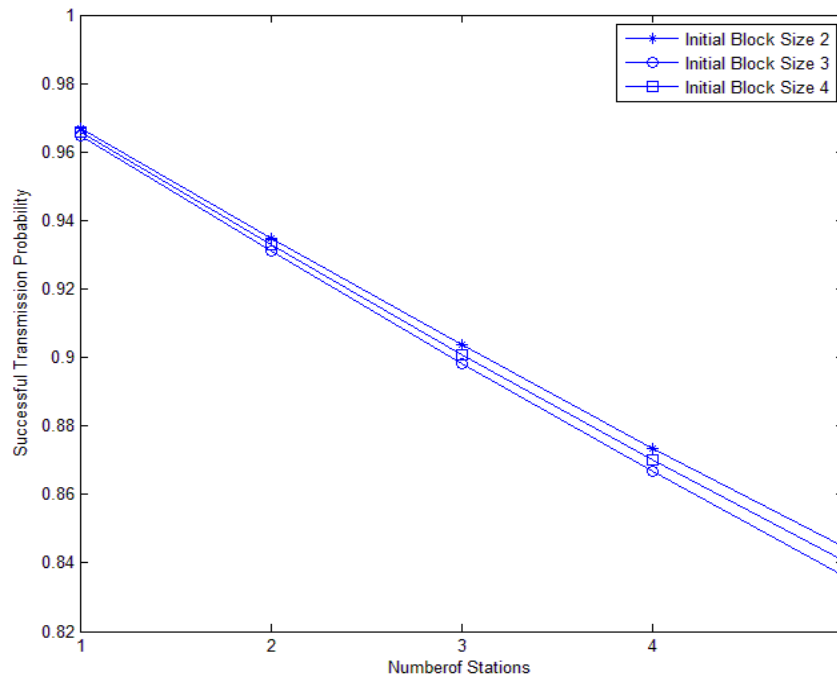


Figure 5.3: Successful Transmission Probability versus Number of Stations.

## 5.2 Retransmission Probability

Figure 5.3 shows the successful transmission probability as a function of the number of stations considering the fixed block size mechanism. It illustrates the same result in Fig. 5.2 that increasing number of stations can affect the transmission probability of the network. Following the channel model we indicated in Chapter IV, when the number of station is increasing, the successful transmission probability will decrease.

In Fig. 5.3, the successful transmission probability is subject to the system model. It describes that when the system maximum transmission retry limits is also configured as four, on the condition of the same initial block size, the successful transmission probability, which is indicated as transition probability in system model in Chapter IV, will decrease with the growing number of stations. The reason is as follows. In channel model, with the increasing number of stations, successful reception status of each station cannot be guarantee. More number of the stations, lower probability the block is successfully received by the stations. Then we discuss system performance on the condition of the same number of stations and different number of initial block sizes. From the Fig. 5.3, we can get that a larger block size obtains a smaller successful transmission probability. It is because that when we take channel memory property into account during our model analysis, we consider the transmission influence between each packet in a block. The larger the block is, the higher the influence will be. We can observe from this result that the larger block size we initialize, the higher impact on the success transmission probability the system model will have. On the other hand, it shows that smaller initial block size can require better successful transmission probability.

### 5.3 Throughput on Large Packet Size

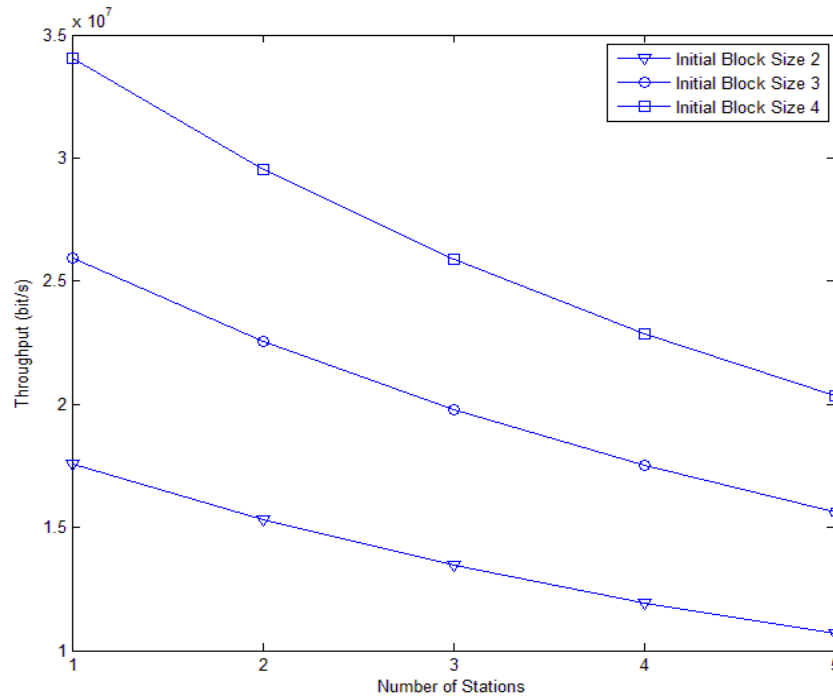


Figure 5.4: System Throughput with Large Packet Size.

In Fig. 5.4, we increase the size of data unit to 7935 bits, which is the maximum packet size of aggregation MAC service data unit. From the figure we can get that the feature of each curve is the same as the one in the Fig. 5.2. The throughput decreases with the growing number of stations for the same initial block size, and increase with the growing size of initial block for the same number of stations. Moreover, compared with Fig. 5.2 which the size of data unit is configured as 2304 bits, system presents a better performance on throughput. The reason is that we propose our model on packet level which means the reception status of the packets is depended on the channel memory property, but not the size of each packet itself. However, we still can get the result that if we increase the size of each packet, the system throughput performance can be increased.

### 5.4 Chapter Summary

This chapter analyze our model from four perspective. One of them is throughput, the other one is successful retransmission probability. In the first part, we find that system throughput of fixed block mechanism decrease with the growing number of stations. And larger initial block size offer a better performance on this part. Compared with previous result we got from variable block size mechanism, we find that fixed block size mechanism shows a much better stable transmission appearance with increasing number of stations. Even though the initial block size in increased, the fixed block size mechanism still exhibit a higher system stability.

In the second part, the system successful transmission probability is discussed. We find that

the numerical result is followed our channel assumption. It decrease with the growing number of stations. Moreover, with lager initial block size, the successful transmission probability for the system exhibits worse than smaller block size. It illustrates the channel memory characteristics we indicated in channel model design.

Finally, we increase the size of transmission packet to 7935 bits, which is used as in aggregation MAC service data unit. We find that GCR Block ACK on fixed block size mechanism has a good support on large data transmission.

# Chapter 6

## Conclusion and Future Work

In this chapter we conclude this thesis by summarizing main conclusion. We also suggest some future research directions that could provide the next steps to extend our model towards a more realistic scenario for 802.11aa.

### 6.1 Conclusion

In this thesis, we expressed the main objective as the performance evaluation of 802.11aa GCR Block ACK mechanism with Markov model. In this final chapter, we will conclude by describing the progress made towards this objective fulfillment in terms of the development of the Markov model referred to as channel model and system model, respectively. Firstly, we build up the channel model based on channel memory property. We consider a scenario that system transmits a block of packets consequently to multi-receivers. Then, a system model is made up to illustrate the transition between different system statuses. We combine these two models together to analyze the overall system transmission status.

Based on the mathematical analysis, the performance of the system is illustrated in terms of system throughput and successful transmission probability. We found that, under the groupcast with constant retry limit under fixed block size transmission mechanism, the network throughput will be reduced with the growing number of receiving terminals. Comparing the throughput results we get from variable block size mechanism in previous study, we found that with the increasing number of station, the fixed block size mechanism performs much stable transmission appearance, which is influenced by the number of station and the channel memory property. Moreover, in analysis of successful transmission probability for various number of stations and initial block size, we can find that the result shows us the system status has only relationship with transmission status, which is also be inferred in the system transmission model mentioned in chapter four.

## 6.2 Future Work

The result of this thesis point to several interesting directions for future work.

The entire analysis presented in this thesis is based on the assumption that the media channel access is always successful. In practice, the channel access method is configured as PCF and EDCA in the standard. Thus it is needed to develop a much more complete model to study the performance of the GCR Block ACK.

Another important part is that we consider the stations work in saturation conditions, i.e., data always queued in the transmission buffer. In reality, many non-saturation model is built for 802.11e or original 802.11 model. We can extend these model in GCR Block ACK mechanism scenario to enhance our research area.

# Bibliography

- [1] Q. Li, L. Jiao and F. Y. Li, "Performance Evaluation of the GCR Block ACK Mechanism in IEEE 802.11aa Networks," *European Wireless*, Guildford, UK, April 2013, pp.7.
- [2] L. E. Barranco, "Performance Evaluation of IEEE 802.11aa MAC Enhancements for Robust Audio Video Streaming," M.S. Thesis, Dept. Eng. in Telecommunication, Univ. Carlos III of Madrid, Leganés, Sep. 2011.
- [3] X. J. Bai X., Mao Y.M., "Performance Investigation of IEEE 802.11e EDCA under Non-saturation Condition based on the M/G/1/K Model," *IEEE Int. Conf. on Industrial Electronics and Applications*, 2007.
- [4] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function" *IEEE J. Sel. Areas Commun (JSAC)*, Vol. 18, No. 3, pp. 535–547, Mar. 2000.
- [5] M. Gas, K. Kosek-Szott, M. Natkaniec and A.R. Pach "3D Markov chain-based saturation throughput model of IEEE 802.11 EDCA," *Electronics Letters*, pp. 826-827. 2011.
- [6] H. Lee, I. Tinnirello, J. Yu, and S. Choi. "Throughput and Delay Analysis of IEEE 802.11e Block ACK with Channel Errors," *Int. Conf. on Commun. Syst. Software and Middleware (COMSWARE)*, Jan. 2007, pp. 1–7.
- [7] IEEE Standard for Local and Metropolitan Area Networks - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Enhancements for Robust Audio Video Streaming, IEEE Std 802.11aa, May 2012.
- [8] IEEE Standard for Local and Metropolitan Area Networks Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, IEEE Std. 802.11n, Oct. 2009.
- [9] IEEE Standard for Local and Metropolitan Area Networks - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service Enhancements, IEEE Std 802.11e, Nov. 2005.
- [10] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std P802.11, Nov. 1997.
- [11] K. Kosek-Szott, A. Krasilov, A. Lyakhov, M. Natkaniec, A. Safonov, S. Szott, I. Tinnirello, (2012) "What' s New for QoS in IEEE 802.11," [Online.] Available: <http://home.agh.edu.pl/szott/What' s New for QoS.pdf>.

- [12] M.A. Santos, J. Villalon and L. Orozco-Barbosa, "Evaluation of the IEEE 802.11aa Group Addressed Service for Robust Audio-Video Streaming," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Ottawa, Canada, Jun. 2012, pp. 6879-6884.
- [13] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of Packet Loss For Compressed Video: Does Burst-Length Matter," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, Hong Kong, China, Apr. 2003, pp. 684-687.
- [14] J. Apostolopoulos, W. Tan, S. Wee, and G. W. Wornell, "Modelling Path Diversity for Multiple Description Video Communication," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, Florida, USA, May 2002, pp. III-2161 – III-2164.
- [15] X. He, and F. Y. Li. "An Automatic Cooperative Retransmission MAC Protocol in Wireless Local Area Networks," in *Proc. European Wireless*, Aalborg, Denmark, May 2009, pp. 228-233.
- [16] L.U. Choi, M.T. Ivrlac, E. Steinbach, and J.A. Nossek, "Analysis of Distortion due to Packet Loss in Streaming Video Transmission over Wireless Communication Links," in *Proc. IEEE Int. Conf. on Image Proc. (ICIP)*, Genoa, Italy, Sep. 2005, pp. I-189 – I-92.
- [17] Y. Kwok, and V. Lau, "The Mobile Radio Propagation Channel" *Wireless Internet and Mobile Computing: Interoperability and Performance*, 1st ed., Chichester, UK: Wiley, 2007, pp. 117.
- [18] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of Video Transmission over Lossy Channels," *IEEE J. Sel. Areas Commun (JSAC)*, vol. 18, no. 6, June 2000, pp. 1012-1032.
- [19] K. Kosek-Szott, "A Throughput Model of IEEE 802.11aa Intra-Access Category Prioritization," *Wireless Pers Commun.*, 2012.no. pp. 1-8.
- [20] K. Maraslis, P. Chatzimisios, and A. Boucouvalas, "IEEE 802.11 aa: Improvements on Video Transmission over Wireless LANs," *IEEE International Conference on Communications (ICC)*, Ottawa, Canada, June 10, 2012, pp. 115-119.
- [21] B. Ranieri, "Combined Link Adaptation and Admission Control for More Reliable Groupcast according to 802.11aa," M.S.thesis, Dept. of Comput. Sci., Saarbruecken German, Univ. of Saarland, June 2010.
- [22] S. Chiochan, and E. Hossain, "Wireless fountain coding with IEEE 802.11e Block ACK for Media Streaming in Wireline-cum-WiFi Networks: a Performance Study," *IEEE Trans. on Mobile Computing*, vol. 10, no. Oct. 2011, pp. 1416-1433.
- [23] J. Farooq, B. Rauf, "Introduction to IEEE 802.11e," *An Overview of Wireless LAN Standards IEEE 802.11 and IEEE 802.11e*, Jahanzeb Farooq Ed, Sweden, University of Umea, 2006 pp. 11-12.



# Appendix

## MATLAB Code

### System Throughput

```
1  clc
2  clear
3  Retry_limit = 7;%dot11shortRetrylimit = 4 and dot11longRetrylimit = 7
4  B_Max_Max = 4;
5  N_station_MAX = 5;%5;
6  state_tranfer_matrix = [0.97 0.03; 0.001 0.999];
7
8  num_just_for_this = 1;
9
10 for B_Max = 2:B_Max_Max
11
12     SIFS = 16e-6;
13     DIFS = 34e-6;
14     PHY_Header = 128;
15     MAC_Header = 16*B_Max;
16     BAR_CF = 16;
17     BAR_IF = 64;
18     BAR = BAR_CF + BAR_IF ;
19     BA_CF = 16;
20     BA_IF = 16*B_Max;
21     BA = BA_CF + BA_IF ;
22     L = 2304;
23     Rc = 6e6;
24     Rd = 54e6;
25
26     T_PLCP = PHY_Header / Rc;
27
28     T_Frame = T_PLCP +SIFS + MAC_Header/Rc + L/Rd;
```

```

29
30 for N_station = 1: N_station_MAX
31 Payload_Broadcast = L*B_Max;
32
33 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
34 % In the following , we will first cacculate out the transition matrix
35 % which is established by two-state-transition matrix.
36
37 Number_station = 1:N_station;
38 lose_data_base = [];
39 lose_data_base(1,:) = ones(1,N_station);
40 for x_x = 1:N_station
41     ones_matrix_station = combntns(Number_station,x_x);
42     XX_1 = size(ones_matrix_station);
43     station_transfer_matrix = ones(XX_1(1,1),N_station);
44     for i = 1:XX_1(1,1)
45         station_transfer_matrix(i,ones_matrix_station(i,:)) = 0;
46         lose_data_base = [lose_data_base; station_transfer_matrix(i,:)];
47     end
48 end
49
50 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
52 markov_matrix = ones(length(lose_data_base),length(lose_data_base));
53 XX_lose = size(lose_data_base);
54
55 for ss_1 = 1:XX_lose(1,1)
56     for ss_2 = 1:XX_lose(1,1)
57         for ss_3 = 1: XX_lose(1,2)
58             markov_matrix(ss_1,ss_2) = markov_matrix(ss_1,ss_2) * ...
                    state_tranfer_matrix(lose_data_base(ss_1,ss_3)+1,
                    lose_data_base(ss_2,ss_3)+1);
59         end
60     end
61 end
62
63 original_markov_chain = markov_matrix;
64 for ss_4 = 1:15
65     markov_matrix = markov_matrix*markov_matrix;
66 end
67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
69 \\
70 steady_vetor_markov = markov_matrix(1,:);
71 %here we assume there are Retry_limit + 2 states in the whole system, which

```

## APPENDIX . MATLAB CODE

```

72 %contains that 0 1 2 3 ... Retry_limit+1. Here we assume state 0 as the
    initial
73 %state of the whole system, sate Retry_limit+1 as the successful state of
74 %the system, and each other states represent the retransmission cycle of
75 %the access point in the whole cycles.
76 original_matrix = zeros(1,Retry_limit+2);
77
78 %here we
79 for i=1:length(original_matrix)
80     original_matrix(i) = i-1;
81 end
82 %original_matrix = [0 1 2 3 4 ... Retry_limit+1];
83 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
84 %
85 %initiate the original state matrix
86 %
87 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88 original_state_matrix = original_matrix';
89 xx_1 = 1;
90 original_state_matrix= for_original_state(original_state_matrix,
    original_matrix,xx_1,B_Max);
91
92 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
93 %
94 %     Here we initial the state that can occurs durring transfer
95 %
96 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
97 Number_Bmax = 1:B_Max;
98 current_transfer_base = [];
99 current_transfer_base(1,:) = ones(1,B_Max);
100 for x_x = 1:B_Max
101     ones_matrix_station = combntns(Number_Bmax,x_x);
102     XX_1 = size(ones_matrix_station);
103     FF_transfer_matrix = ones(XX_1(1,1),B_Max);
104     for i = 1:XX_1(1,1)
105         FF_transfer_matrix(i,ones_matrix_station(i,:)) = 0;
106         current_transfer_base = [current_transfer_base; FF_transfer_matrix(i
            ,:)]);
107     end
108 end
109 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
110 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
111 %     here we initial our transfer matrix
112 %
113 n = length(original_state_matrix);

```

```

114 transfer_matrix = zeros(n,n);
115
116 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
117 %
118 %     As we have got the the current transfer base, we need to make the
119 %     transfer matrix based on this base.
120 %
121 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
122
123 %[A_0,B_0] = find(current_transfer_base == 0);
124 %[A_1,B_1] = find(current_transfer_base == 1);
125 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
126 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
127 %
128 %     we will product the transfer state based on each
129 %     original_state_matrix itself
130 %
131 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
132 %current_transfer_base
133 transfer_base_memory = current_transfer_base;
134
135 base_memory_probability = zeros(length(current_transfer_base),1);
136 base_memory_probability(1) = markov_matrix(1,1)*original_markov_chain(1,1)^(
    B_Max-1);
137
138 XX_1 = size(current_transfer_base);
139 for xx_1 = 2:XX_1(1,1)
140
141     vector_1 = find(current_transfer_base(xx_1,:) == 0);
142     byte_length = length(vector_1);
143     lose_data_transfer_matrix = ones(N_station,B_Max);
144     s = 1;
145     PPP = 0;
146
147
148     for kk_1 = 2:length(lose_data_base)
149         [KKKK,PPP] = for_inner_cycle(s,vector_1,lose_data_base, ... kk_1,
            lose_data_transfer_matrix,original_markov_chain,PPP,
            steady_vetor_markov);%markov_matrix(1,1));
150     end
151
152     base_memory_probability(xx_1) = PPP;
153 end
154
155 sum(base_memory_probability)

```

```
156
157 %for i =1:length(original_matrix)
158 for xx_2 = 1:length(original_state_matrix)
159     Vector_current = [];
160     Vector_current = original_state_matrix(xx_2, :);
161
162     for xx_3 = 1:B_Max
163         i = Vector_current(xx_3);
164         if i<original_matrix(length(original_matrix)-1)
165             transfer_base_memory(current_transfer_base(:,xx_3) == 0,xx_3) = i
166                 +1;
167             transfer_base_memory(current_transfer_base(:,xx_3) == 1,xx_3) =
168                 original_matrix(length(original_matrix));
169         elseif i == original_matrix(length(original_matrix)-1)
170             transfer_base_memory(current_transfer_base(:,xx_3) == 0,xx_3) =
171                 0;
172             transfer_base_memory(current_transfer_base(:,xx_3) == 1,xx_3) =
173                 original_matrix(length(original_matrix));
174         else
175             transfer_base_memory(current_transfer_base(:,xx_3) == 0,xx_3) =
176                 1;
177             transfer_base_memory(current_transfer_base(:,xx_3) == 1,xx_3) =
178                 original_matrix(length(original_matrix));
179         end
180     end
181 end
182
183 vector_A = find((Vector_current ~= 0)\&(Vector_current ~= Retry_limit +1))
184 ;
185 vector_B = find((Vector_current == 0)|(Vector_current == Retry_limit +1));
186 vector_change_position = [];
187 if length(vector_A)
188     if length(vector_B)
189         vector_change_position = [vector_A,vector_B];
190     else
191         vector_change_position = [vector_A];
192     end
193 end
194 if length(vector_change_position)
195     transfer_base_memory_new = transfer_base_memory;
196     YE = size(transfer_base_memory);
197     for xx_5 = 1:YE(1,1)
198         for xx_6 = 1:YE(1,2)
199             transfer_base_memory_new(xx_5,xx_6) = transfer_base_memory(
200                 xx_5,vector_change_position(1,xx_6));
201         end
202     end
```

```

193     end
194 end
195 %transfer_base_memory;
196 for xx_4 = 1:length(transfer_base_memory)
197     SS = [];
198     if length(vector_change_position)
199         SS = transfer_base_memory_new(xx_4, :);
200     else
201         SS = transfer_base_memory(xx_4, :);
202     end
203     %original_state_matrix(find(ismember(original_state_matrix, SS, 'rows')
204         ==1), :)
205     transfer_matrix(xx_2, find(ismember(original_state_matrix, SS, 'rows')
206         ==1)) = base_memory_probability(xx_4);
207 end
208 new_transfer_matrix = transfer_matrix;
209
210 for ss_1 = 1:7
211     new_transfer_matrix = new_transfer_matrix*new_transfer_matrix;
212 end
213 NTmatrix_steady_state = new_transfer_matrix(1, :);
214 XX_2 = size(original_state_matrix);
215
216 SPFrame = 0;
217 for yy_1 = XX_2(1,1)
218     success_frame = find(original_state_matrix(yy_1, :) == (length(
219         original_matrix)-1));
220     if success_frame
221         SP= NTmatrix_steady_state(yy_1)*length(success_frame);
222     end
223     SPFrame = SPFrame + SP;
224 end
225
226 TMT_Payload_fix(N_station) = L* SPFrame;
227 T_BA = N_station*(2*T_PLCP + BAR/Rc + BA/Rc) + (2*N_station-1)*SIFS + DIFS;
228 slot_fix(N_station) = T_Frame*4 + T_BA;
229 end
230 N_station = 1:N_station_MAX;
231
232 TMT_Broadcast_fix = TMT_Payload_fix./slot_fix;
233 TMT_Broadcast_frame_fix(num_just_for_this, :) = TMT_Broadcast_fix;
234

```

```

235 num_just_for_this = num_just_for_this+1;
236 end
237
238 wowo_2=size(TMT_Broadcast_frame_fix);
239 for wow = 1:wowo_2(1,1)
240     plot(N_station,TMT_Broadcast_frame_fix(wow,:))
241     hold on
242 end
243
244
245 function [loss_data_transfer_matrix,PPP] = for_inner_cycle(s,XX_2,
    lose_data_base,kk_1,loss_data_transfer_matrix,...
246                                     original_markov_chain,PPP,
    steady_state_pro)
247
248     loss_data_transfer_matrix(:,XX_2(s)) = lose_data_base(kk_1,:);
249     if s == length(XX_2)
250         KK = [];
251         SS_1 = size(loss_data_transfer_matrix);
252         %0111
253         %1111
254         %for ss_1 = 1:SS_1(1,2)
255             %KK = loss_data_transfer_matrix(:,ss_1);
256             ss_2 = 1;
257             while sum(abs(lose_data_base(ss_2,:)-(loss_data_transfer_matrix
   (:,1))')) ~= 0
258                 ss_2 = ss_2 +1;
259             end
260             PP = steady_state_pro(1,ss_2);%* original_markov_chain(1,ss_2);
261             % lose_data_base(ss_2,:)
262             for ss_3 = 2:SS_1(1,2)
263                 ss_4 = 1;
264                 while sum(abs(lose_data_base(ss_4,:)-(
    loss_data_transfer_matrix(:,ss_3-1))')) ~= 0
265                     ss_4 = ss_4 +1;
266                 end
267                 ss_5 = 1;
268                 while sum(abs(lose_data_base(ss_5,:)-(
    loss_data_transfer_matrix(:,ss_3))')) ~= 0
269                     ss_5 = ss_5 +1;
270                 end
271                 PP = PP*original_markov_chain(ss_4,ss_5);
272             end
273         %end
274     %     for ss_1 = 1:SS_1(1,1)

```

```

275 %           KK(1,:) = [1 loss_data_transfer_matrix(ss_1,1)];
276 %           P(ss_1) = steady_state_pro*state_tranfer_matrix(KK(1,1)+1,KK(1,2)
+1);
277 %           for ss_2 = 2:length(loss_data_transfer_matrix(ss_1,:))
278 %               KK(ss_2,:) = [ loss_data_transfer_matrix(ss_1,ss_2-1)
loss_data_transfer_matrix(ss_1,ss_2)];
279 %               P(ss_1) = P(ss_1)* state_tranfer_matrix(KK(ss_2,1)+1,KK(ss_2
,2)+1);
280 %           end
281 %       end
282 %       PP = 1;
283 %       for ss_3 = 1:length(P)
284 %           PP = PP * P(ss_3);
285 %       end
286 %       %PP
287 %       PPP = PPP +PP;
288 %   end
289 %   s = s+1;
290 %   if s<=length(XX_2)
291 %       for kk_2 = 2:length(lose_data_base)
292 %           [loss_data_transfer_matrix,PPP] = for_inner_cycle(s,XX_2,
lose_data_base,kk_2,... loss_data_transfer_matrix,
original_markov_chain,PPP,steady_state_pro);
293 %       end
294 %   end
295 % end
296
297
298 %
299 %     Here u can get the result of original state matrix sample
300 %
301 % function [original_state_matrix]=for_original_state(original_state_matrix,
original_matrix, ... i,B_Max)
302
303 % KK = [];
304 % SS = [];
305 % xx_3 = 1;
306 % for xx_1 = 1:length(original_state_matrix)
307 %     for xx_2 = 1:length(original_matrix)
308 %         KK = [original_state_matrix(xx_1,:),original_matrix(xx_2)];
309 %         SS(xx_3,:) = KK;
310 %         xx_3 = xx_3+1;
311 %     end
312 % end
313

```



```

314 i = i+1;
315 original_state_matrix = [];
316 original_state_matrix = SS;
317
318 if i<B_Max
319     [original_state_matrix]=for_original_state(original_state_matrix,
320     ... original_matrix,i,B_Max);
321 end
322 end

```

### Successful Transmission Probability

```

1 function trofixba()
2
3 clc
4 clear
5
6 Retry_limit = 4;%dot11shortRetrylimit = 4 and dot11longRetrylimit = 7
7 B_Max_Max = 4;
8 N_station_MAX = 5;%5;
9 state_tranfer_matrix = [0.97 0.03; 0.001 0.999];
10 ssss = 1;
11
12 for B_Max = 2:B_Max_Max
13
14     SIFS = 16e-6;
15     DIFS = 34e-6;
16     PHY_Header = 128;
17     MAC_Header = 16*B_Max;
18     BAR_CF = 16;
19     BAR_IF = 64;
20     BAR = BAR_CF + BAR_IF ;
21     BA_CF = 16;
22     BA_IF = 16*B_Max;
23     BA = BA_CF + BA_IF ;
24     L = 2304;
25     Rc = 6e6;
26     Rd = 54e6;
27
28     T_PLCP = PHY_Header / Rc;
29
30
31     T_Frame = T_PLCP +SIFS + MAC_Header/Rc + L/Rd;
32
33 for N_station = 1: N_station_MAX

```

```

34 Payload_Broadcast = L*B_Max;
35
36 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
37 % In the following , we will first cacculate out the transition matrix
38 % which is established by two-state-transition matrix.
39
40 Number_station = 1:N_station;
41 lose_data_base = [];
42 lose_data_base(1,:) = ones(1,N_station);
43 for x_x = 1:N_station
44     ones_matrix_station = combntns(Number_station,x_x);
45     XX_1 = size(ones_matrix_station);
46     station_transfer_matrix = ones(XX_1(1,1),N_station);
47     for i = 1:XX_1(1,1)
48         station_transfer_matrix(i,ones_matrix_station(i,:)) = 0;
49         lose_data_base = [lose_data_base; station_transfer_matrix(i,:)];
50     end
51 end
52
53 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
54 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
55 markov_matrix = ones(length(lose_data_base),length(lose_data_base));
56 XX_lose = size(lose_data_base);
57
58 for ss_1 = 1:XX_lose(1,1)
59     for ss_2 = 1:XX_lose(1,1)
60         for ss_3 = 1: XX_lose(1,2)
61             markov_matrix(ss_1,ss_2) = markov_matrix(ss_1,ss_2) * ...
62                 state_tranfer_matrix(lose_data_base(ss_1,ss_3)+1,
63                 lose_data_base(ss_2,ss_3)+1);
64         end
65     end
66 end
67
68 original_markov_chain = markov_matrix;
69 for ss_4 = 1:15
70     markov_matrix = markov_matrix*markov_matrix;
71 end
72
73 steady_vetor_markov = markov_matrix(1,:);
74 %here we assume there are Retry_limit + 2 states in the whole system, which
75 %contains that 0 1 2 3 ... Retry_limit+1. Here we assume state 0 as the
    initial

```

```

76 %state of the whole system, sate Retry_limit+1 as the successful state of
77 %the system, and each other states represent the retransmission cycle of
78 %the access point in the whole cycles.
79 original_matrix = zeros(1,Retry_limit+2);
80
81 %here we
82 for i=1:length(original_matrix)
83     original_matrix(i) = i-1;
84 end
85 %original_matrix = [0 1 2 3 4 ... Retry_limit+1];
86 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87 %
88 %initiate the original state matrix
89 %
90 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
91 original_state_matrix = original_matrix';
92 xx_1 = 1;
93 original_state_matrix= for_original_state(original_state_matrix,
    original_matrix,xx_1,B_Max);
94
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96 %
97 %     Here we initial the state that can occurs durring transfer
98 %
99 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
100 Number_Bmax = 1:B_Max;
101 current_transfer_base = [];
102 current_transfer_base(1,:) = ones(1,B_Max);
103 for x_x = 1:B_Max
104     ones_matrix_station = combntns(Number_Bmax,x_x);
105     XX_1 = size(ones_matrix_station);
106     FF_transfer_matrix = ones(XX_1(1,1),B_Max);
107     for i = 1:XX_1(1,1)
108         FF_transfer_matrix(i,ones_matrix_station(i,:)) = 0;
109         current_transfer_base = [current_transfer_base; FF_transfer_matrix(i
    ,:)]];
110     end
111 end
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
114 %     here we initial our transfer matrix
115 %
116 n = length(original_state_matrix);
117 transfer_matrix = zeros(n,n);
118

```

```

119 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
120 %
121 %     As we have got the the current transfer base, we need to make the
122 %     transfer matrix based on this base.
123 %
124 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
125
126 %[A_0,B_0] = find(current_transfer_base == 0);
127 %[A_1,B_1] = find(current_transfer_base == 1);
128 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
129 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
130 %
131 %     we will product the transfer state based on each
132 %     original_state_matrix itself
133 %
134 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
135 %current_transfer_base
136 transfer_base_memory = current_transfer_base;
137
138 base_memory_probability = zeros(length(current_transfer_base),1);
139 base_memory_probability(1) = markov_matrix(1,1)*original_markov_chain(1,1)^(
    B_Max-1);
140
141 XX_1 = size(current_transfer_base);
142 for xx_1 = 2:XX_1(1,1)
143
144     vector_1 = find(current_transfer_base(xx_1,:) == 0);
145     byte_length = length(vector_1);
146     loss_data_transfer_matrix = ones(N_station,B_Max);
147     s= 1;
148     PPP = 0;
149
150
151     for kk_1 = 2:length(lose_data_base)
152         [KKKK,PPP] = for_inner_cycle(s,vector_1,lose_data_base, ... kk_1,
            loss_data_transfer_matrix,original_markov_chain,PPP,
            steady_vetor_markov);%markov_matrix(1,1));
153     end
154
155     base_memory_probability(xx_1) = PPP;
156 end
157
158 sum(base_memory_probability)
159
160 %for i =1:length(original_matrix)

```

```

161 for xx_2 = 1:length(original_state_matrix)
162     Vector_current = [];
163     Vector_current = original_state_matrix(xx_2,:);
164
165     for xx_3 = 1:B_Max
166         i = Vector_current(xx_3);
167         if i < original_matrix(length(original_matrix)-1)
168             transfer_base_memory(current_transfer_base(:,xx_3) == 0,xx_3) = i
169                 +1;
170             transfer_base_memory(current_transfer_base(:,xx_3) == 1,xx_3) =
171                 original_matrix(length(original_matrix));
172         elseif i == original_matrix(length(original_matrix)-1)
173             transfer_base_memory(current_transfer_base(:,xx_3) == 0,xx_3) =
174                 0;
175             transfer_base_memory(current_transfer_base(:,xx_3) == 1,xx_3) =
176                 original_matrix(length(original_matrix));
177         else
178             transfer_base_memory(current_transfer_base(:,xx_3) == 0,xx_3) =
179                 1;
180             transfer_base_memory(current_transfer_base(:,xx_3) == 1,xx_3) =
181                 original_matrix(length(original_matrix));
182         end
183     end
184
185     vector_A = find((Vector_current ~= 0) & (Vector_current ~= Retry_limit +1));
186     vector_B = find((Vector_current == 0) | (Vector_current == Retry_limit +1));
187     vector_change_position = [];
188     if length(vector_A)
189         if length(vector_B)
190             vector_change_position = [vector_A,vector_B];
191         else
192             vector_change_position = [vector_A];
193         end
194     end
195
196     if length(vector_change_position)
197         transfer_base_memory_new = transfer_base_memory;
198         YE = size(transfer_base_memory);
199         for xx_5 = 1:YE(1,1)
200             for xx_6 = 1:YE(1,2)
201                 transfer_base_memory_new(xx_5,xx_6) = transfer_base_memory(
202                     xx_5,vector_change_position(1,xx_6));
203             end
204         end
205     end
206
207     %transfer_base_memory;

```

```
199     for xx_4 = 1:length(transfer_base_memory)
200         SS = [];
201         if length(vector_change_position)
202             SS = transfer_base_memory_new(xx_4,:);
203         else
204             SS = transfer_base_memory(xx_4,:);
205         end
206         %original_state_matrix(find(ismember(original_state_matrix,SS,'rows')
207             ==1),:)
208         transfer_matrix(xx_2,find(ismember(original_state_matrix,SS,'rows')
209             ==1)) = base_memory_probability(xx_4);
210     end
211 end
212 new_transfer_matrix = transfer_matrix;
213
214 for ss_1 = 1:7
215     new_transfer_matrix = new_transfer_matrix*new_transfer_matrix;
216 end
217 NTmatrix_steady_state = new_transfer_matrix(1,:);
218 XX_2 = size(original_state_matrix);
219
220 SP = 0;
221
222 for yy_1 = XX_2(1,1)
223     success_frame = find(original_state_matrix(yy_1,:) == (length(
224         original_matrix)-1));
225     if success_frame
226         SP= NTmatrix_steady_state(yy_1)+SP;
227     end
228 end
229 TMT_Payload_fix(N_station) = SP;
230 end
231
232
233 HHHH(ssss,:) = TMT_Payload_fix;
234
235 ssss = ssss+1;
236 end
237
238
239
240 wow_n = size(HHHH);
```

```

241 %n=1:wow_n(1,2);
242 retr =1:wow_n(1,2);
243 figure
244 for wow = 1:wow_n(1,1)
245     plot(retr,HHHH(wow,:))
246     hold on
247 end
248
249
250 function [loss_data_transfer_matrix,PPP] = for_inner_cycle(s,XX_2,
    lose_data_base,kk_1,loss_data_transfer_matrix,... original_markov_chain,
    PPP,steady_state_pro)
251
252     loss_data_transfer_matrix(:,XX_2(s)) = lose_data_base(kk_1,:);
253     if s == length(XX_2)
254         KK = [];
255         SS_1 = size(loss_data_transfer_matrix);
256         %0111
257         %1111
258         for ss_1 = 1:SS_1(1,2)
259             %KK = loss_data_transfer_matrix(:,ss_1);
260             ss_2 = 1;
261             while sum(abs(lose_data_base(ss_2,:)-(loss_data_transfer_matrix
                (:,1))')) ~= 0
262                 ss_2 = ss_2 +1;
263             end
264             PP = steady_state_pro(1,ss_2);%* original_markov_chain(1,ss_2);
265             % lose_data_base(ss_2,:)
266             for ss_3 = 2:SS_1(1,2)
267                 ss_4 = 1;
268                 while sum(abs(lose_data_base(ss_4,:)-(
                    loss_data_transfer_matrix(:,ss_3-1))')) ~= 0
269                     ss_4 = ss_4 +1;
270                 end
271                 ss_5 = 1;
272                 while sum(abs(lose_data_base(ss_5,:)-(
                    loss_data_transfer_matrix(:,ss_3))')) ~= 0
273                     ss_5 = ss_5 +1;
274                 end
275                 PP = PP*original_markov_chain(ss_4,ss_5);
276             end
277
278             PPP = PPP +PP;
279     end
280     s = s+1;

```

```
281     if s<=length(XX_2)
282         for kk_2 = 2:length(lose_data_base)
283             [loss_data_transfer_matrix,PPP] = for_inner_cycle(s,XX_2,
                lose_data_base,kk_2,... loss_data_transfer_matrix,
                original_markov_chain,PPP,steady_state_pro);
284         end
285     end
286
287
288 end
289
290
291 %
292 %     Here u can get the result of original state matrix sample
293 %
294 function [original_state_matrix]=for_original_state(original_state_matrix,
                original_matrix, ... i,B_Max)
295
296
297 KK = [];
298 SS = [];
299 xx_3 = 1;
300 for xx_1 = 1:length(original_state_matrix)
301     for xx_2 = 1:length(original_matrix)
302         KK = [original_state_matrix(xx_1,:),original_matrix(xx_2)];
303         SS(xx_3,:) = KK;
304         xx_3 = xx_3+1;
305     end
306 end
307
308 i = i+1;
309 original_state_matrix = [];
310 original_state_matrix = SS;
311
312 if i<B_Max
313     [original_state_matrix]=for_original_state(original_state_matrix,
                original_matrix, ... i,B_Max);
314 end
315 end
```