

NINEVAH UNIVERSITY
COLLEGE OF ELECTRONICS ENGINEERING
COMPUTER AND INFORMATION ENG. DEPARTMENT



Implementation of a Random Linear Network Coding for 5G Video Delivery Using FPGA

By

Amenah Muafaq Younus Altaee

M.Sc. Thesis

In

Computer and Information Engineering

Supervised by

Assistant Prof. Mohammed Hazim Al-Jammas

2021 A.D.

1442 A.H.

Implementation of a Random Linear Network Coding for 5G Video Delivery Using FPGA

A Thesis Submitted

By

Amenah Muafaq Younus Altaee

To

The Council of the College of Electronic Engineering

Ninevah University

As a Partial Fulfillment of the Requirements

For the Degree of Master of Science

In

Computer and Information Engineering

Supervised by

Asst.Prof.Dr. Mohammed H.Al- Jammal

2021 A.D.

1442 A.H.

Supervisor's Certification

I certify that the dissertation entitled (**Implementation of a Random Linear Network Coding for 5G Video Delivery Using FPGA**) was prepared by **Amenah Muafaq Younus** under my supervision at the Department of Computer and Information Engineering, University of Ninevah, as a partial requirement for the Master of Science Degree in Computer and Information Engineering.

Signature:

Name: Asst.Prof.Dr. Mohammed H.Al- Jammas

Department of Computer and Information Engineering

Date: / /2021

Linguistic Advisor Certification

I certify that the linguistic evaluation of this thesis entitled "**Implementation of a Random Linear Network Coding for 5G Video Delivery Using FPGA**" was carried out by me and it is accepted linguistically and in expression.

Signature:

Name:

Date: / /2021

Post-Graduate Committee Certification

According to the recommendations presented by the supervisor of this dissertation and the linguistic reviewer, I nominate this dissertation to be forwarded to discussion.

Signature:

Name: Assistant Prof. Maan A. S. Al-Adwany

Department Head Certification

I certify that this dissertation was carried out in the Department of Computer and Information Engineering. I nominate it to be forwarded to discussion.

Signature:

Name: Assistant Prof. Maan A. S. Al-Adwany

Date: / /2021

Committee Certification

We the examining committee, certify that we have read this dissertation entitled **(Implementation of a Random Linear Network Coding for 5G Video Delivery Using FPGA)** and have examined the postgraduate student (**Amenah Muafaq Younus**) in its contents and that in our opinion; it meets the standards of a dissertation for the degree of Master of Science in Computer and Information Engineering.

Signature:

Name:

Head of committee

Date: / /2021

Signature:

Name:

Member

Date: / /2021

Signature:

Name:

Member

Date: / /2021

Signature:

Name:

Member and Supervisor

Date: / /2021

The college council, in its meeting on / /2020, has decided to award the degree of Master of Science in Computer and Information Engineering to the candidate.

Signature:

Name:

Dean of the College

Date: / /2021

ACKNOWLEDGEMENTS

Praise to ALLAH, Lord of the Worlds. Praise be to ALLAH who gave me strength the ability to complete my dissertation.

I would like to express my sincere gratitude and thanks to my supervisor, *Dr. Mohammed H. Al-Jammas* for the continuous guidance, useful suggestions and constant encouragement and assistance to me throughout the research period, so I thank him very much.

I also express my gratitude and thanks to the Head and all members of the Computer and Information Engineering Department who provided me with science and knowledge throughout the preparatory year.

And thanks to all those who gave me advice and provided me with the information that was useful to me during the period of research

I thank my dear parents for their continuous support, and I thank my brother Mohammed and my sisters Shahad, and Roua for standing with me and supporting me throughout my study.

And thanks all my friends who supported me.

Last but not the least, I thank ALLAH for everything.

Researcher

Amenah Muafaq Younus

2021

ABSTRACT

The rapid growth of video transmission and the increasing demand for high-definition, multi-display, and wide-area video services in recent years. This has led to the need for robust technologies to cover these requirements. The Random Linear Network Coding (RLNC) is one of the most promising technology for video delivery, improving throughput, utilizing bandwidth capacity, and saving High reliability and low latency. In this thesis, the RLNC algorithm has been applied to video transmission and implemented on FPGA.

A proposed model consisting of three algorithms for three stages: encoding algorithm performs in the source node, the recoding algorithm performs in the intermediate node, and the decoding algorithm performs in the destination node.

In Matlab, a simple network is designed to apply these three algorithms, where it consists: of the source node, intermediate node, and then destination nodes. The effect of two parameters on video transmission; is the degree of Galois field and video resolution were studied.

The simulation results revealed the effectiveness of selecting large numbers from the Galois field in increasing the reliability of the video coding and thus increased the reliability of the video transmission. Also, it is shown that the effect of video resolution on the time of each stage. When resolution increases, simulation time for each stage increases. The different resolution videos were delivered, and the Bit Error Rate (BER) between the video frames at the transmitter end and the video frames at the receiving end was equal to zero. Throughput is increased by 25% than traditional approaches that do not use RLNC by increasing bandwidth capacity utilization as more than one packet is transmitted in the transmission process, thus reducing the number of transmission times.

Xilinx Vivado 2018.2 has been in conjunction with Matlab in hardware Implementation. Design methodology (HDL verifier through (FPGA in the loop) and Modelsim) was used to design and implement three models; (Encoder, Recorder, and Decoder) on FPGA Xilinx Nexys A7 (XC7A100T-1CSG324C) kit board in FPGA implementation, the results to transmit H videos at 30 fps in real-time, the encoding time spends (104.14050 ms), and the decoding time (104.14084 ms).

TABLE OF CONTENTS

Subject		Page
ACKNOWLEDGEMENTS		I
ABSTRACT		II
TABLE OF CONTENTS		IV
LIST OF FIGURES		VII
LIST OF TABLES		X
LIST OF ABBREVIATIONS		XI
Chapter One- Introduction		1
1.1	Introduction	1
1.2	Video Delivery Problems Definitions	1
1.3	Literature Survey	2
1.4	Research Methodology	5
1.5	Aims of Thesis	5
1.6	Thesis Layout	5
Chapter Two- Introduction of 5G and RLNC		7
2.1	Introduction	7
2.2	Introduction of 5G	7
2.3	Applications of 5G	9
2.4	Video in 5G	10
2.5	Type of Video According to Resolution	11
2.5	Network Coding (NC)	11
2.5.1	Max-Flow Min-Cut Theorem	12
2.5.2	Network Coding in a Butterfly Network	13

2.5.3	NC in a Wireless Network	15
2.5.4	Benefits of Network Coding	16
2.6	Network Coding Classifications	18
2.6.1	Linear Network Coding (LNC)	19
2.6.2	Intra-Flow.	20
2.6.3	Inter-Flow.	20
2.6.4	Random Linear Network Coding (RLNC)	20
2.6.4.1	Encoding	23
2.6.4.2	Recoding	24
2.6.4.3	Decoding	25
Chapter Three - Simulate a Proposed Model of RLNC for Video Delivery		26
3.1	Introduction	26
3.2	Proposal Model	26
3.2.1	Encoding	27
3.2.2	Recoding	30
3.2.3	Decoding	32
3.3	Simulation of Proposal Model	34
3.3.1	The Parameters of the Simulation Model	35
3.3.2	Implemented the Proposal Model	35
3.4	Results	36
3.4.1	Video Resolution	36
3.4.2	Simulation Time	49
3.5	Discuss The Results	51
Chapter Fourth - FPGA Implementation of Proposal Model		53

4.1	Introduction	53
4.2	Field Programmable Gate Array (FPGA)	53
4.2.1	FPGA Basic Parts	53
4.2.2	Nexys A7	54
4.3	FPGA in The Loop	54
4.4	Implementation of The Three Stages for The Proposed Model	55
4.4.1	Encoder Model	55
4.4.1.1	Implementation of Encoder	56
4.4.1.2	Internal Details and Processing for Encoder	57
4.4.1.3	Time for Encoder	59
4.4.1.4	The Frame Resulting from The Encoding Process	60
4.4.2	Recoder Model	63
4.4.2.1	Implementation of Recoder	64
4.4.2.2	Internal Details and Processing for Recoder	65
4.4.2.3	The Frame Resulting from The Recoding Process	65
4.4.3	Decoder Model	66
4.4.3.1	Implementation of Decoder	67
4.4.3.2	Internal Details and Processing for Decoder	68
4.4.3.3	Time for Decoder	70
4.4.3.4	The Frame Resulting from The Decoding Process	71
Chapter Five - Conclusions and Suggestion for Future Work		74
5.1	Conclusions	74
5.2	Future Work	74
References		76

LIST OF FIGURES

Figure	Title	page
2.1	Usage Scenarios Of IMT for 2020 and Beyond	9
2.2	Butterfly Network with A Multicast Flow From S to Y	13
2.3	Butterfly Example	14
2.4	Example of NC in Wireless	15
2.5	Security Benefit Using NC	18
2.6	Network Coding Classifications	18
2.7	Linear Network Coding	19
2.8	RLNC Scheme	21
2.9	Structure of Transmitted Coded Packet	23
2.10	Recoding Process Types	24
3.1	General Block Diagram	26
3.2	Encoding Process in The Source Node	27
3.3	Encoded Packet Format Out of Source Node	29
3.4	Encoding Process	29
3.5	Flowchart Of Recoding in Intermediate Node	30
3.6	Recoding Process	31
3.7	Flowchart of Decoding Process	32
3.8	Network Model	36

3.9	SD Resolution video , GF=4	37
3.10	SD Resolution video , GF=8	38
3.11	SD Resolution video , GF=16	39
3.12	Comparison of SD Frames Encoded with Different Degrees of GF= 4, 8, and 16	40
3.13	HD Resolution Video , GF=4	41
3.14	HD Resolution Video , GF=8	42
3.15	HD Resolution Video , GF=16	43
3.16	Comparison of HD Frames encoded with Different Degrees of GF = 4, 8, and 16	44
3.17	4K Resolution Video , GF=4	45
3.18	4K Resolution Video , GF=8	46
3.19	4K Resolution Video , GF=16	47
3.20	Comparison of 4K Frames Encoded with Different Degrees of GF= 4, 8, and 16	48
3.21	Simulation Time for Each Stage When Use 4k Video with Different GF Values	49
3.22	Simulation Time When Use HD Video With Different GF Values	50
3.23	Simulation Time When Use SD Video With Different GF Values	50
3.24	Simulation Time for Different Videos with Different GF Values	51
3.25	Frames and Their Histograms at Source And Destination	52

4.1	Nexys A7 Overview	54
4.2	Encoder Model	56
4.3	RTL Internal Details For Encoder-Subsystem Block	58
4.4	Encoding Time with Different Number of Frames	60
4.5	Modelsim Timing Signals for Encoding Process (30 Frame)	61
4.6	Frames Before and After Encoding Process	62
4.7	Recoder Model	63
4.8	RTL Internal Details for Recoder	65
4.9	Frames Before and After Recoding Process	66
4.10	Decoder Model	67
4.11	Internal Details for Decoder	69
4.12	Decoding Time with Different Number of Frame	71
4.13	Modelsim Timing Signals for Decoding Process (30 Frame)	72
4.14	Frames Before and After Decoding Process	73

LIST OF TABLES

Table	Title	page
2.1	Various Formats of Video	11
4.1	Device Utilization Summary for Encoder	57
4.2	Encoding Time for SD and HD Videos	59
4.3	Device Utilization Summary for Recoder	64
4.4	Device Utilization Summary for Decoder	68
4.5	Decoding Time for SD and HD Videos	70

LIST OF ABBREVIATIONS

Abbreviation	Name
3GPP	Third Generation Partnership Project
4G	Fourth Generation
5G	Fifth-Generation
AR	Augmented Reality
ARQ	Automatic Repeat-Request
BER	Bit Error Rate
CLBs	Configurable Logic Blocks
E	Edges
eMBB	Enhanced Mobile Broadband
FEC	Forward-Error-Correction
FIL	FPGA in The Loop
FPGA	Field Programmable Gate Arrays
fps	Frame Per Second
Fq	Finite Fields
GF	Galois Field
GSMA	Group Special Mobile Association
HD	High Definition
HDL	Hardware Description Languages

IC	Integrated Circuit
IMT	International Mobile Telecommunications
IOBs	Input/Output Blocks
JTAG	Joint Test Action Group
LNC	Linear Network Coding
LTE	Long Term Evolution
LUTs	Lookup Tables
Mbps	Mega Bit Per Second
ms	Millisecond
NC	Network Coding
NCC	Network-Coded Cooperation
PLL	Phase-Locked Loop
QoS	Quality Of Service
RAM	Random Access Memory
RGB	Red, Green, Blue
RLNC	Random Linear Network Coding
S/F	Store-And-Forward
SD	Standard Definition
UHD	Ultra High Definition
V	Vertex

VHDL	(Very High Speed Integrated Circuit) Hardware Description Language
VR	Virtual Reality
YCbCr	Luminance; Chroma: Blue; Chroma: Red

Chapter One

Introduction

1.1 Introduction

The number of mobile wireless devices has grown in recent years and is predicted to rise rapidly in the next few years. The demands of consumers to be continuously connected are growing in such a way that it is predicted that the number of mobile devices will reach 5.7 billion by 2023 [1]. In addition, mobile data traffic also continues to rise in volume and the video delivery will approach an estimated 78% of overall mobile data traffic by 2021. Yet, technology as being used now may not be able to accommodate this volume of traffic. The new 5G standard will offer a range of new opportunities for improving video delivery around the 5G network. One of the promising approaches to video delivery RLNC [2].

1.2 Video Delivery Problems Definitions

Mobile video transmission is on the rise, with video traffic reaching approximately 78% of overall mobile data traffic by 2021, up from 60% in 2016. During the period (2016-2021), phone traffic expects to increase to 7 times [2], which needs high bitrates. High bitrates in video delivery systems inspire studies to optimize network throughput, as congestion originating from intolerable application data requirements will cause queue overload and packet drops. The consequences of packet loss on video transfers are further exacerbating when we consider wireless networks; thus, mitigating the effect of packet loss on the content of the received video at the end terminal is both an important and fascinating challenge. To fix packet losses, we first consider current approaches. For real-time video applications, the conventional ARQ (Automatic Repeat-

request) scheme may not be appropriate; because it imposes lengthy delays and may not satisfy the strict time criteria for streaming video. Alternatively, to secure video packets, one should follow forward-error-correction (FEC) codes and use several multicast trees to provide redundant paths to transmit the coded video bit stream [3][4][5]. The packets are conveyed via the store-and-forward S/F system at intermediate nodes in this scheme. This clear redundancy will introduce an unwanted burden to the network and reduce the rate at which it is possible to transfer data. However, if nodes generalize operations beyond basic replication and routing, it has been seen that where coding is permitted at intermediate nodes instead of the restrictive S/F network model as in the NC approach, enhanced throughput and inexplicit redundancy may be accomplished [4].

1.3 Literature Survey

- In 2000, Rudolf Ahlswede et al. [6] offered a new type of problem termed network information flow that is inspired, they supposed a point-to-point communication network where a set of information sources are transmitted to a certain number of destinations. They presume the sources of information are mutually independent. The issue is to define the area of the allowable coding rate. In this paper, they researched the problem with a single source of information, and they obtained a simple description of the admissible coding rate area. Their result can be considered as the Max-flow Min-cut Theorem for network information flow. Their work reveals that when using coding at the node, the bandwidth will, in general, be saved.
- In 2008, Sachin Katti et al. [7] suggested a new architecture for wireless mesh networks called COPE. In addition to packet forwarding, packets coming from various sources are combined by routers to increase the content of each transmission, they revealed the intelligently combining packets increase the network throughput.

- In 2009, Dong Nguyen et al. [8] suggested several network coding schemes for minimizing the number of broadcasts from one transmitter to multiple recipients. The key idea is to permit the sender to merge and transfer lost packets in a certain manner so that multiple recipients can retrieve their lost packets with one transmission.
- In 2010, Hui Wang et al. [4] offered a description of the implementation of Network Coding (NC) to robust video streaming in the erasure network. They also introduced the benefits and drawbacks of using NC for efficient video streaming. And they assessed the conventional way of video erasure protection in the store-and-forward (S/F) network against the NC approach.
- In 2011, Janus Heide et al. [9] studied the issue of effective decoding a random linear network coding (RLNC) over a finite field. They concentrated on the case where the code is random, relatively sparse, . The aim is to decode data using fewer actions to theoretically optimize coding efficiency and reduce energy consumption, they offer several simple enhancements to minimize the volume of decoding operations required. Their results demonstrate that the enhancements can decrease the number of operations used during decoding by (10-20) percent on average.
- In 2017, Mohammad Esmaeilzadeh et al. [10] researched the issue of broadcasting layered streaming video over heterogeneous wireless single-hop networks utilizing feedback-free (RLNC). They researched the impact of the number of layers and suggest a strategy that chooses the optimal number of layers to conform to the best efficiency, it is noted that the performance gap between the proposed feedback-free strategy and the ideals scheme is very minimal, and that the adaptive selection of a number of video layers significantly gap closes.

- In 2018, Dejan Vukobratovic et al. [2] conducted a thorough analysis of two interrelated issues in mobile cellular networks, RLNC for packet erasure security and mobile video distribution. Using a module-based approach, they presented RLNC, inspired by the search to discover both the need and the suitable place for RLNC sub-layers in 4G/5G mobile cellular network video delivery solutions.
- In 2018, Lars Nielsen et al. [11] analyzed the effect of generation and symbol size on latency for encoding in RLNC. They concentrated on estimating the impact of generation changes (number of novel packets) and symbol size (number of bytes per data packet) settings on the encoding latency on full vector and on-the-fly algorithms. They demonstrate that encoding latency increases to double when either the generation size or the symbol size increases to double and proves this by comprehensive simulations.
- In 2019, Diogo Goncalves et al. [12], suggested written RLNC by P4 language. for the first step towards a production-level platform. Their approach offers the ability to mix the payload of many packets and perform the required Galois field operations, and they claim to be realistic even under the rigid memory and processor constraints of switching hardware.
- In 2019, Evgeny Tsimbalo et al. [13] assumed a single source, several intermediates nodes, and a single destination loss network using RLNC for all transmission nodes. They discussed the issue of calculating the probability of efficient decoding at the destination node. They assumed an arbitrary field size and take into account the relationship between the intermediates nodes. They suggest a new maximum limit for an arbitrary number of intermediates nodes, that becomes accurate for a single intermediate node. Utilizing Monte Carlo simulations, they demonstrated that the suggested limit is very accurate.

- In 2020, I. Leyva-Mayorga et al. [14] offered network-coded cooperation (NCC) protocol for the efficient transfer of large content and an analytical model that explains its nature. The NCC protocol incorporates the advantages of collaborative architectures such as mobile clouds with RLNC. Their results demonstrated that the advantages of their NCC protocol are threefold relative to the development of multiple parallel unicast sessions: minimized power consumption by co-operating users and provided throughput benefits when cellular bandwidth is inadequate.

1.4 Research Methodology

In this project, the focus will be on RLNC is a scheme identified to create a potentially high impact on flexible, efficient, and reliable mobile video delivery. The proposed model will be simulated in Matlab and then will be implemented on FPGA. RLNC is a simple and effective packet-level erasure protection mechanism. It has many useful features including a rateless property, which means the ability to improve throughput in network scenarios than by re-encoding packets in intermediate network nodes.

1.5 Aims of Thesis

- Study the possibility of applying the RLNC algorithm in video delivery.
- Simulate the proposed model in Matlab to preserve the video sent over the network and deliver it to the destination without errors..
- Implement the proposed model using the (FPGA in the loop) tool is provided by Matlab and VIVADO to implement the model on FPGA.

1.6 Thesis Layout

Chapter two presents an introduction about 5G, applications, and video in 5G. Offers an introduction about NC, its principles, benefits, and classification.

Explains briefly about Liners Network Coding (LNC). Then explains RLNC and its stages.

Chapter three explains details of the proposed model and algorithms for each stage in the proposed model to apply the RLNC on video delivery, presents the implementation of the proposed model in Matlab, introduces the results then discusses them.

Chapter four includes the FPGA implementation of three processes for the proposed model; encoder, recoder, and decoder and offers results.

Chapter five includes conclusions and recommendations for future work.

Chapter Two

Introduction of 5G and RLNC

2.1 Introduction

In recent years, the number of connected phones has increased, and this trend is expected to continue in the next few years. Consumer's need for constant connectivity is increasing to the point where it is estimated that the number of connected phones will exceed 5.7 billion by 2023 [1]. Besides, mobile data traffic also continues to rise in volume and the video delivery will approach an estimated 78% of overall mobile data traffic by 2021. Yet, technology as being used now may not be able to accommodate this volume of traffic. The new 5G standard will offer a range of new opportunities for improving video delivery around the 5G network. One of the promising approaches to video delivery RLNC [2].

2.2 Introduction of 5G

The fifth-generation (5G) of wireless communications aims to offer incredibly high data rates, exceptionally low delay, and a substantial increase in base station capacity, and a dramatic change in the realized quality of service (QoS) of customers relative to the existing 4G LTE (Long Term Evolution) networks. The increased growth of mobile devices, the emergence of new multimedia applications, and the exponential growth in demand for wireless data (multimedia) place a huge strain on current cellular networks. 5G wireless networks with enhanced speeds of data, capacity, delay, and QoS are predicted to reduce existing mobile network troubles [15][16][17][18]. The Group Special Mobile Association (GSMA) partners with its members to eventually shape 5G communication. By combining the various research developments from industry and academia, eight key requirements of 5G systems are presented below [18][19][20]:

1. 10 Gbps data rate in actual networks: about 10 times the estimated peak data rate of the conventional LTE network of 150 Mbps.
2. 1 ms round trip latency: decrease to around 10 times the round trip time of 4G by 10 ms.
3. High unit area bandwidth: It must be capable of linking a wide range of users with higher bandwidths and longer duration in a given area.
4. A huge number of connected devices: 5G networks have to offer connectivity to thousands of devices in order to achieve the Internet of Things (IoT) vision.
5. 99.999 percent availability preparation: 5G illustrates that the network should always be available in reality.
6. Nearly 100% coverage with access everywhere at any time: 5G wireless networks offer maximum coverage regardless of the position of subscribers.
7. Almost 90 percent decrease in energy use: the production of green technologies is now being regarded by standard bodies. It is more important with the fast data speeds and massive connectivity of 5G wireless.
8. High battery life: In evolving 5G networks, the reduction of power consumption by devices is profoundly significant.

Wireless industries, universities, and academic institutions have begun working on various facets of 5G wireless networks to provide these eight listed specifications

2.3 Applications of 5G

5G is the fifth-generation wireless standard established by the third generation partnership project (3GPP) in Version 15 and is also the predecessor to the previous cellular standard, i.e., the LTE standard [21]. 5G Infrastructure redefines the conventional way the internet is wirelessly linked. The new generation of cellular technologies can accommodate evolving new cases involving high throughput, a vast number of connected devices, low latency, and stable connectivity that cannot be accomplished by the current cellular standard [22]. In the current standard, Mobile Network Operator used to be the sole player responsible for delivering connectivity to customers. However, 5G technology offers the requisite infrastructure and tools for different industry verticals so that they can now deliver new services to users [23]. The International Telecommunications Union has advised three major types of use cases for 5G in the form of International Mobile Telecommunications (IMT) 2020 and beyond [22]. Likewise figure 2.1, further illustrates the general application of 5G technology[24].

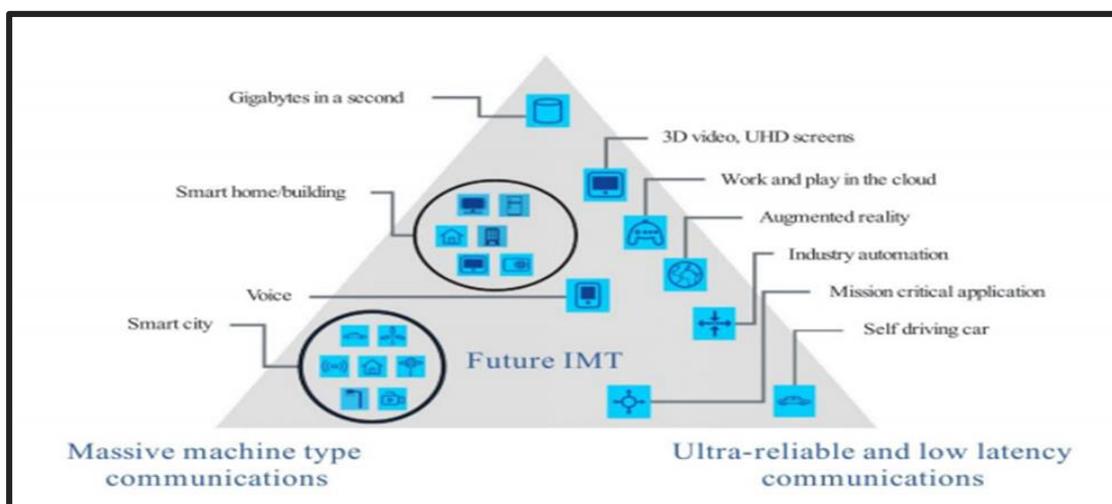


Figure 2.1. Usage Scenarios of IMT for 2020 and Beyond

- Massive Machine-Type Communication refers to services where billions of machines are linked to create a truly interconnected community. These instruments usually transfer low volumes of non-delay data with low throughput. These systems need to have a low cost and longer battery life, which can last for years [22].
- Ultra-Reliable Low Latency Communication utilizes case-based applications that have strict latency, efficiency, and availability requirements [25].
- Enhanced Mobile Broadband can be viewed as a clear expansion to the 4G technology usage case. Due to the prevalence of smart wireless devices along with digital content such as high definition (HD) video, online gaming, Augmented reality (AR), virtual reality (VR), etc., the need for ultra-broadband coverage is becoming increasingly unavoidable to provide higher performance and smooth user experience. As estimated, mobile data traffic is expected to rise seven-fold from 2017 to 2022 [26]. Consequently, the wireless network has to accommodate more data traffic than the current network capacity. The key objective of Enhanced mobile broadband (eMBB) is therefore to offer improved indoor and outdoor broadband services to consumers by offering better data rate, spectral quality, coverage, capacity, latency, and user density [24].

2.4 Video in 5G

Mobile video transmission keeps rising in volume, this growth is attributed to the mix of ever-increasing resolution of consumer handsets and the emergence of 4K/8K ultra high definition formats. Incentivized by the development of innovative video services that depend on multi-view and 360-degree video, improved broadcasting, and peer-to-peer video services [26][27]. In this thesis will be focused on video delivery in 5G.

2.5 Type of Video According to Resolution

Video resolution determines the amount of detail in the video, or how realistic and clear the video appears. It's measured by the number of pixels contained in the standard aspect ratio. A higher pixel number means a higher-resolution video, while a lower pixel number means a lower-resolution video. Table 2.1 shows the various formats of video (resolutions).

Table 2.1. Various Formats of Video [27][28]

Format Name	Format	Resolution(pixels)
Standard Definition	SD	640 × 480
Standard Definition	SD	720 × 576
High Definition	HD	1280 × 720
High Definition / 2K	FULL-HD	1920 × 1080
Ultra High Definition	UHD	3840 × 2160
4K UHD	4K	4096 × 2160
8K UHD	8K	8192 × 4320

2.5 Network Coding (NC)

From the beginning, different definitions have been given for the NC Theory. The basic concept behind NC is to perform simple mathematical operations on the contents of the packets that will be transmitted. The coding performed at a network node is referred to as NC [6]. So, NC is defined as an encoded packet consisting of many packets produced by a network node of the same length as the source packets. This approach will minimize the volume of data transmitted over network connections, thus reducing the risk of congestion over a network node. NC, rather than seeing a node in the network as a simple element to transfer the flow of information only as a router that merely stores and forwards information flow without processing it. NC lets the contract do the computation by combining and sending packets into it. Different theoretical and

experimental research indicates that substantial advances can be achieved by the use of NC in multi-hop wireless networks, as well as by the use of multi-cast sessions [29][30][31]. NC enhances throughput due to efficient packet transmission [6]. In NC the packet lost during transmission is retrieved, upon receiving a group of component packets. NC is characterized by security and adding complexity to the network [32].

2.5.1 Max-Flow Min-Cut Theorem

Commonly, the volume of information flowing in a transmission network cannot exceed the specified value, which is determined by the maximum capacity that can be achieved in that network. This capacity is specified by the concept of max-flow min-cut. Using NC helps achieve maximum capacity. The problem with the max-flow min-cut in graph theory is to find the maximum flow that can be taken out from the source to the destination. In networks, this is about identifying the paths that the data transmitted from the source would take to provide the maximum flow capacity. As an example, the butterfly network is shown in figure 2.2 source S and destination Y assuming unit capacitance links. Where S has the ability to send at most two pieces of data to Y through: $S \rightarrow T \rightarrow Y$ and $S \rightarrow U \rightarrow W \rightarrow X \rightarrow Y$ [33].

The capacity as defined in the min-cut max-flow between S and Y has a value of 2 in this case (equivalent to the number of individual paths connecting the source to its destination in the case of unit capacity). Likewise, the maxflow min-cut capacity regardless of the flow from S to Y, between S and Z is (2). Now, let's assume that Source S would like to send two pieces of data to two Y and Z destinations. The maximum limit to be achieved is equal to the lowest of the maximum capacity for any of the destinations to be taken individually.

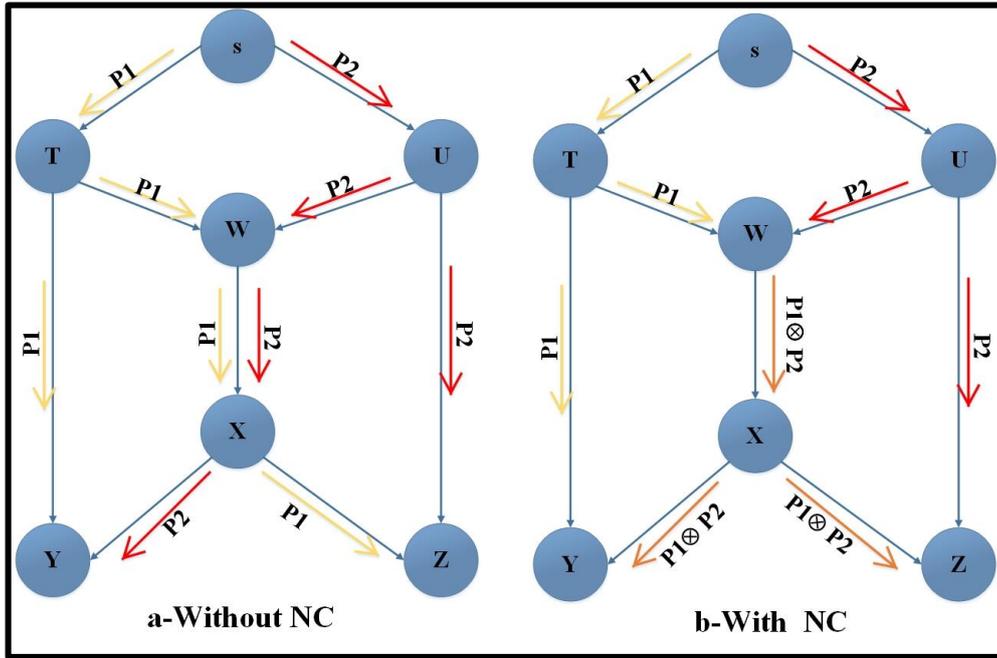


Figure 2.2. Butterfly Network with a Multicast Flow from S to Y

Therefore, this capacity is equal to 2. Even so, without NC, the unit capacity of the $W \rightarrow X$ Link does not allow us to concurrently route the data of the multicast flow coming from the T and U nodes. Therefore, it is not possible to achieve the full reachable capacity of S towards its destinations, even by doubling the capacity of the $W \rightarrow X$ link. On the other side, the use of NC allows the transmission of a data combination on a $W \rightarrow X$ unit capacity connection, allowing it possible to concurrently route data to destinations and thus to achieve a maximum capacity of (2) units. In what follows, we are focused on the viability of a coordination scenario that applies NC and allows us to achieve a max-flow min-cut boundary [33].

2.5.2 Network Coding in a Butterfly Network

They used what is known as a butterfly network to illustrate the principle of NC and to present a simple scenario where NC improves throughput, which then became extremely popular with the butterfly example name [6]. Give an example of a simple network consisting of a single source (S) with multicast

communication and two destination nodes (D1, D2) assuming erasure-free links, error-free, and carrying a single unit for each link. Figure 2.3 describes two cases in which the source node S sends two P1 and P2 packets to the target nodes D1 and D2 in both of them. These packets are received by Nodes R1 and R2 and forwarded to both R3 and D1 or D2. When (R3) node receives two different packets, it mixes packets by using an XOR operation, $P1 \oplus P2$, then sends it to node R4 that sends the same packet to the destination nodes D1 and D2. D1, which obtained P1 directly from node R1, but P2 obtain by executing $P1 \oplus (P1 \oplus P2)$. D2 obtained P2 directly from node R2, but P1 obtain by executing $P2 \oplus (P2 \oplus P1)$. It can easily be concluded that 10 transmissions (t) are needed by the

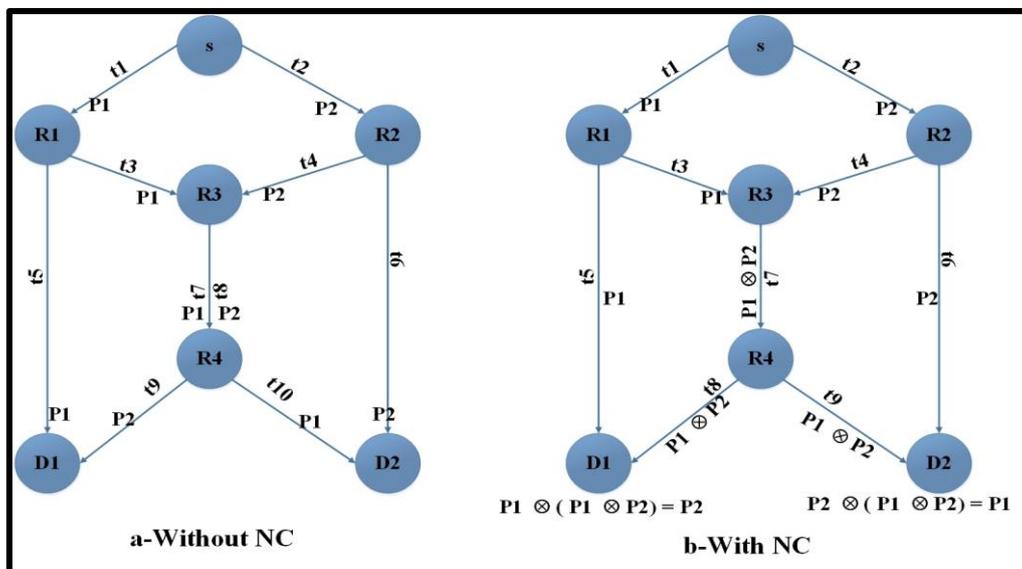


Figure 2.3. Butterfly Example

traditional approach in figure 2.3a, whereas 9t is required by the approach in figure 2.3b. The coding gain on the wired network is then $10t/9t$.

As seen in figure 2.3b, with NC, a boost of 25% of the throughput can be seen in a butterfly network. In this example, they revealed that NC would aid the sender and the recipient in transfer data traffic at the same rate as the min-cut theorem [6], observing that nodes transmit set of the original data and that senders and receivers do not need to track every packet to know if a packet has been lost,

rather, concentrate on collecting a sufficient number of individual linear combination packets to decode them into the originals [34].

2.5.3 NC in a Wireless Network

A scenario is seen in figure 2.4 to illustrate the advantages of NC in a wireless network. When Alice and Bob want to share their packets via a router. In the traditional approach, Alice sends the packet to the router that forwards it to Bob, and Bob sends the packet to the router that forwards it to Alice. This method takes four transmissions times. When the NC approach is used, Alice and Bob each send their packets to the router, which performs an XORing operation between them and broadcasts the encoded packets. Alice and Bob can decode

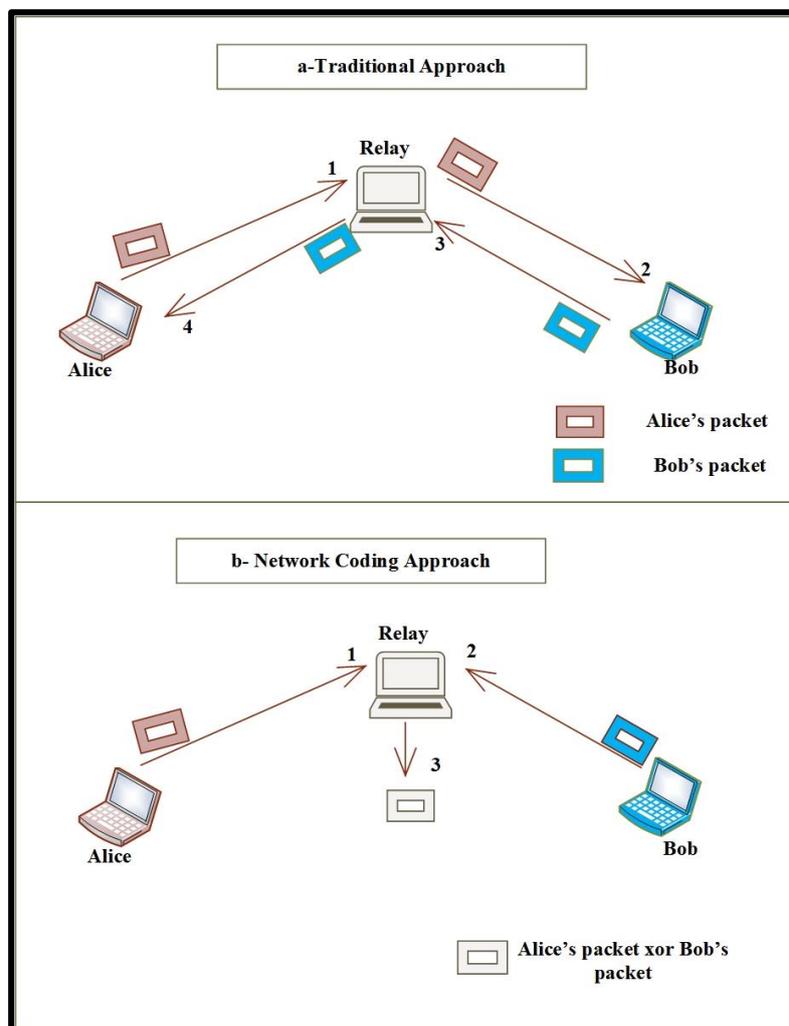


Figure 2.4. Example of NC in Wireless

encoded packets by performing XORing between encoded packets with their packets to get the required packets. Instead of using 4 transmissions (without NC), it takes 3 transmissions for the two nodes to exchange packets, the remaining transmission time can be utilized to transmit extra packets. This improves the wireless throughput wherewith using NC increases throughput by 25% over conventional forwarding scheme [35].

2.5.4 Benefits of Network Coding

Following are the main benefits of NC:

A. Throughput

One of the most distinctive features of NC is the increase in throughput, as this feature is used to send packets more efficiently, meaning more packets can be sent with fewer transmissions. To illustrate, take the previous example in figure 2.4 when Alex and Bob want to communicate with each other via an intermediate node. Using traditional methods, they will exchange information within four transmissions. But by using NC technology, instead of broadcasting each packet separately, the intermediate can get both packets from Bob and Alex, performs an XOR operation on them, and then broadcast the resulting packets. Since Bob and Alex each have their packets, they can calculate the incoming packet from the other using a simple XOR operation. Whereby NC, the number of transmissions has been reduced to three instead of four [36][37]. In truth, there are situations where NC offers a solution to a problem that is not solved by conventional routing. One of them is to reach the maxflow-min-cut limit of each of the nodes concurrently in the butterfly network [38][40].

B. Robustness

As (random) network coding is implemented to wireless networks, the network can become more robust against packet losses. Such packet losses

can occur from connection degradation, buffer overflow, and packet collisions on a medium in a shared media topology. Standard, erasure coding is implemented to the sender side and decoding to the recipient side. If instead, erasure coding is added to all connections, robustness would improve at the expense of the network latency. NC, however, provides this advantage without needing any decoding of any connection, and hence without adding any further delay to the network [38][39].

C. Complexity

In some cases, finding optimum routing is a difficult task. Finding Optimal routing utilizing NC is linear optimization [36]. In reality, NC can allow communication that cannot be accomplished by conventional routing[38][42].

D. Security

NC is capable of providing some security to the network. It does, however, pose some security issues. Owing to the nature of the linear mixture of packets, tampering can be detected [39]. This can be shown by the following example: assume a network of four nodes as seen in figure 2.5. The source node A sends two messages to node D via the intermediate nodes B and C. All links have the unit capacity. The conventional routing approach of doing this is illustrated in figure 2.5a. An intruder listening to one of the AB-BD or AC-CD routes will get this information. If alternatively, NC is used, the same volume of data can be moved from A to D. Figure 2.5b An intruder listening to one of the routes is unable to extract either of the messages a or b. Receiver D is capable of merging the two packets ($a+b$ and $a+2b$) to get both messages a and b. Redundancy in the message delivered is a necessity for this to be preserved. If not, NC helps the intruder to retrieve both data streams at a single place[32]. Identifies the protocol to defend against Byzantine attacks using NC from A to D.

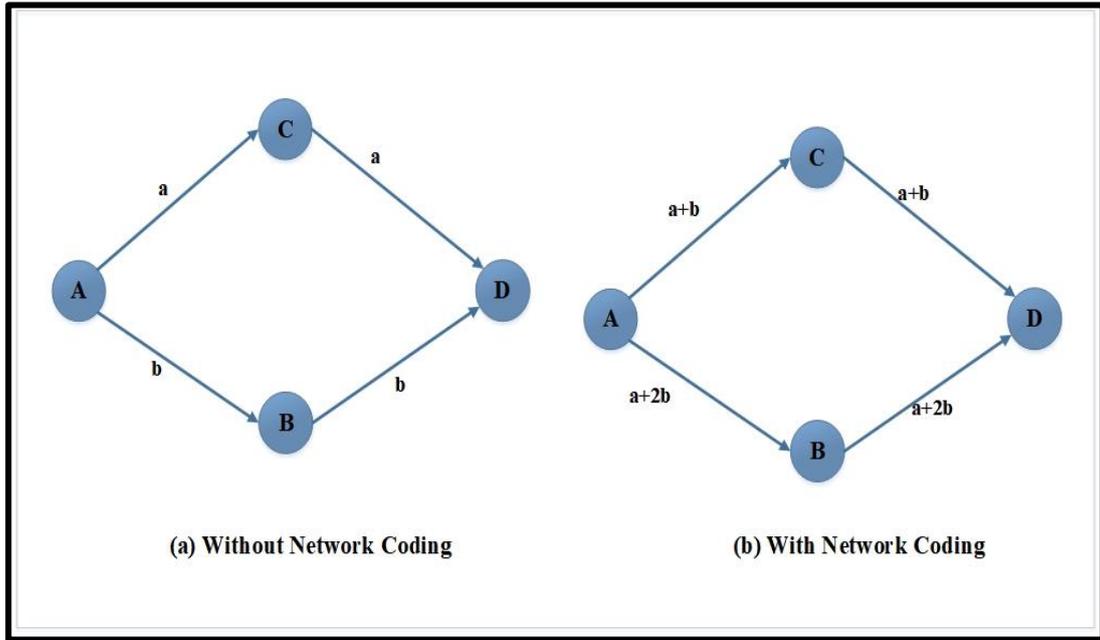


Figure 2.5. Security Benefit Using NC

2.6 Network Coding Classifications

NC is classified such as shown in figure 2.6 [40].

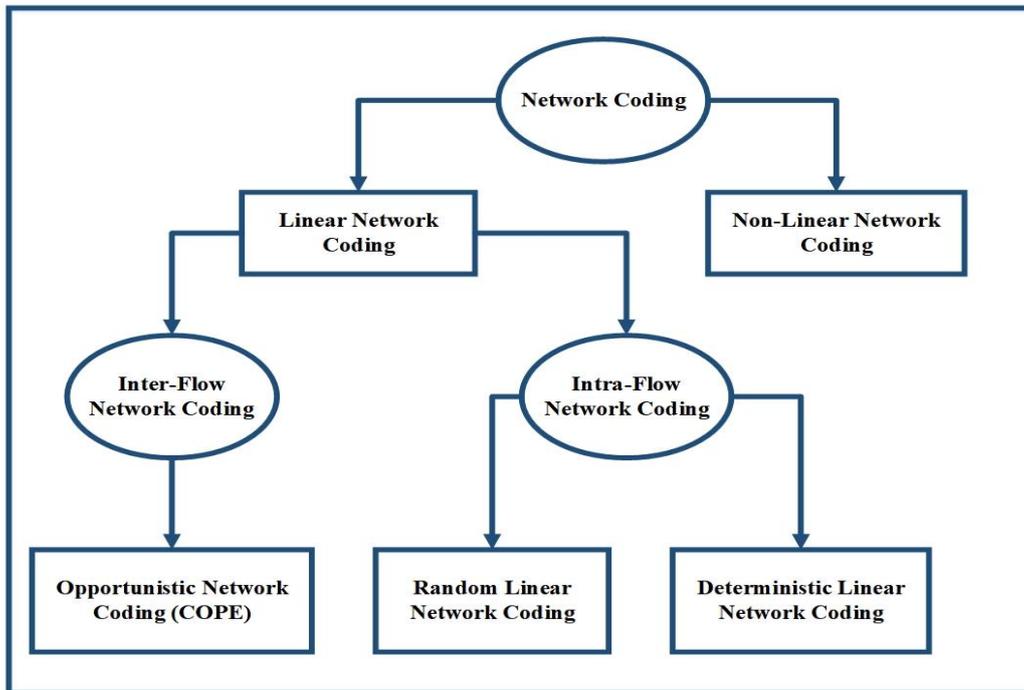


Figure 2.6. Network Coding Classifications

2.6.1 Linear Network Coding (LNC)

It has been shown that, by using NC, intermediate nodes in the network are capable of merging a number of packets received into one or more new outgoing packets. Linear Network Coding (LNC) [39][41], operates on fields larger than binary fields operations. So, it allows performing complex operations when mixing the packets received in the intermediate nodes. This makes LNC one of the most successful algorithms in multicasting because it achieves network capacity with low complexity. In the LNC, each data unit is processed utilizing finite fields F_q with a prime number q or, considering a Galois Field (GF), $q=2^m$ for some integer m , where F_2 refers to $[0, 2^{m-1}]$ having a graph $G=(V,E)$, where V symbolize vertex and E the edges through where data is transferred, The butterfly network is again the example selected to better understand LNC operations within the network, as shown in figure 2.7[42].

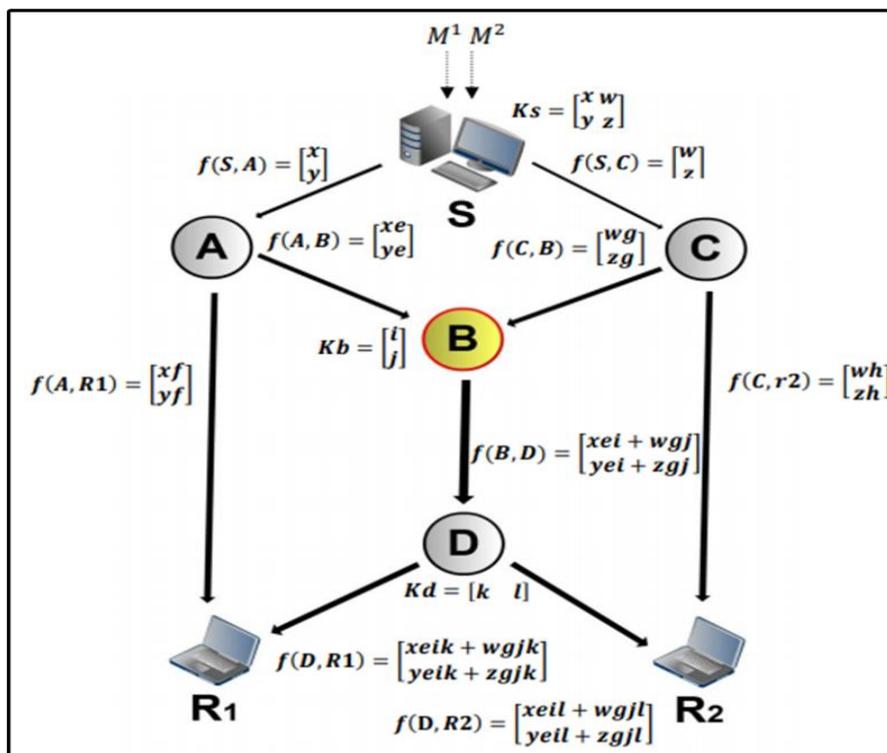


Figure 2.7. Linear Network Coding

2.6.2 Intra-Flow.

In this method, routers will merge packets sent to the same destination. Routers do not need to decode packets; the destination will ultimately decode them after a sufficient amount of encoded packets have been received [43] [44].

2.6.3 Inter-Flow.

In this method, routers merge or code packets relating to different streams of communications (already intra-session coded) that intersect at an intermediate point in the network [43][44].

2.6.4 Random Linear Network Coding (RLNC)

RLNC is a category of network coding that was originally suggested for multicast communication. It makes the transmission scheme easy and efficient. Because it does not need coordination between the network nodes. A coded packet is generated according to this approach by selecting and mixing the data packets randomly over finite fields. The random feature of RLNC provides the property of ratelessness, this allows generating an infinite number of encoded packets. In comparison to deterministic codes, the coding attribute of RLNC also reduces the need for signaling. It is possible to decode the original packets from any enough set of encoded packets. In comparison, unlike most conventional coding systems, the RLNC can adapt to any transmission rate on the fly. Due to these attributes, RLNC is simple to implement and is considered an appropriate technique for dynamic topologies and various links [45][46].

Consequently, RLNC is a useful technique of node cooperation, particularly for broadcast communication, and in distributed networks, where nodes cannot easily organize the routing of information across the network. Besides, it has been shown that due to its intrinsic randomness, RLNC achieves multicast capability in a distributed manner. The communicating nodes are usually battery-powered and have a restricted-energy budget in energy-constraint wireless networks, such

as sensor networks. It is a big challenge to enhance the network lifetime without reducing network efficiency. RLNC can reduce the number of independent packet transfers on the network and reduce or remove packet retransmission resulted from poor channel conditions [45]. Consequently, RLNC has the ability to both boost energy efficiency and reduce total network latency, which ultimately contributes to an improvement in the life of the network [47][48][49]. The RLNC is based on the concept of “a random linear mapping of input to output data symbols over a finite field”. Figure 2.8 illustrates the scheme of RLNC[50].

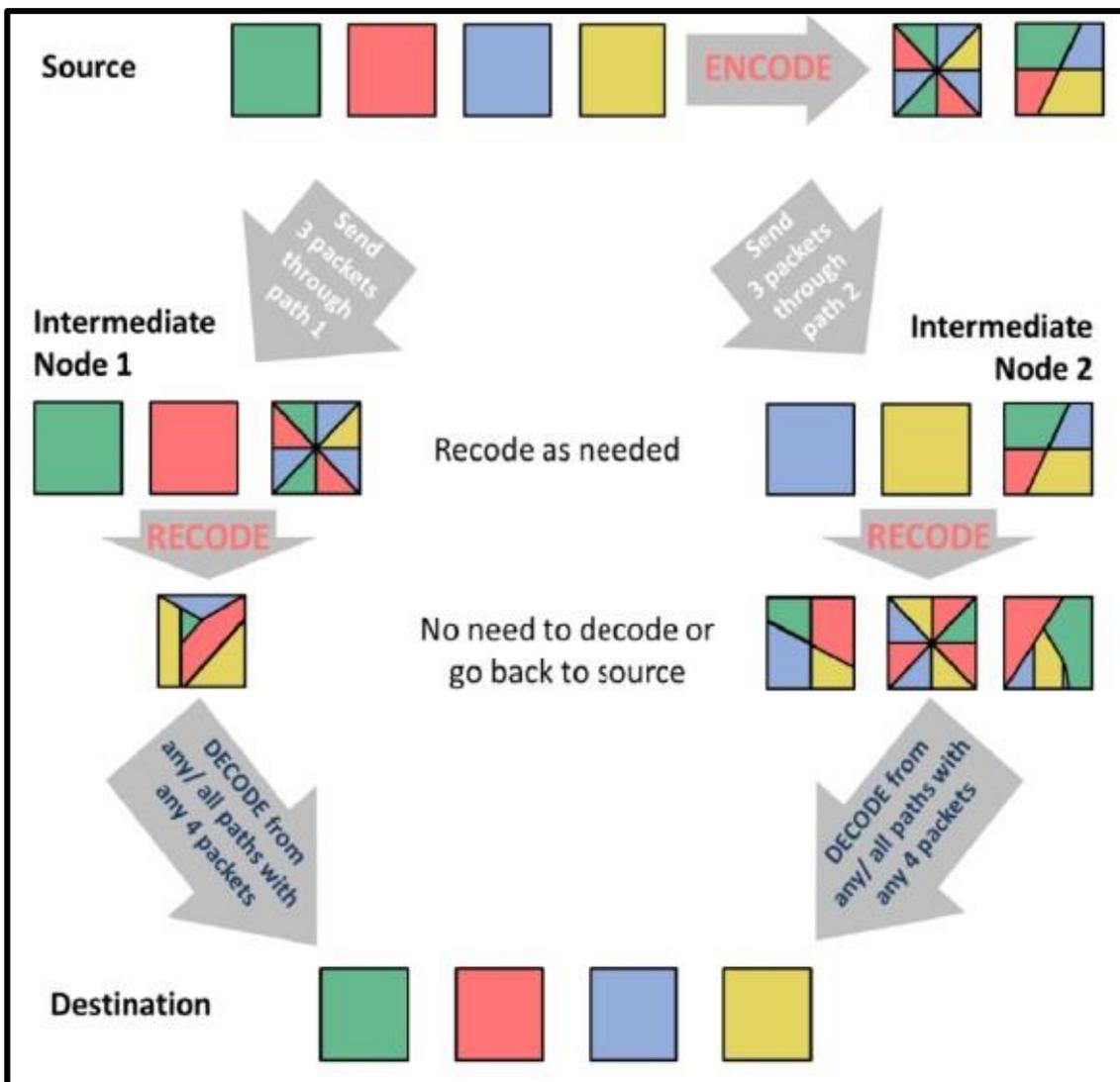


Figure 2.8. RLNC Scheme

The collecting receivables incoming packets and then encodes the originating packets into a network-encoded packet with random coefficients instead of predefined ones. It randomly produces a local coding vector and locates the coding information inside the packets to enable them to be decoded by the succeeding nodes. For each of the nodes executing the previous step of coding over randomized coefficients, the vector of a node's random coefficients is updated. These random coefficients allow distributed transmission and enhanced robustness against losses, and additional redundancy can be added to minimize packet loss.

Furthermore, RLNC offers to the intermediate node's capability to (recode) without decoding firstly. This attribute distinguishes RLNC from conventional coding schemes such as Reed-Solomon. Where intermediate nodes need to decode the obtained data until they can recode it. Key factor that characterizes the method of random coding: the field size of Galois field [51].

The field size of a Galois field is the maximum amount of field elements representing the coefficients used to code a field. Essentially, the field size indicates the number of field members. The source packet p of size s can be expressed binary as an integer value within the field domain. If the field size (q) is sufficiently large, the use of a random code will reach a multicast capacity with a high probability. In practice, the finite field of size $q = 2^m$ is used where m is the number of bits per field element. With this, the probability of not achieving multicast capabilities using RLNC decreases exponentially with the number of bits per field element. To ensure better independent linearity between encoded packets, a larger field size must be used. Alternatively, the larger size, the more likely it is to have linear independent encoding packets. However, the cost of having a large field size is seen in performance, as the field size rises, so does the size of the header since the coding coefficients become important. The effect of field size on the average decoding latency of the receiver is analyzed in [52][53].

2.6.4.1 Encoding

The encoding method is used for packets/symbols, where a packet may be made up of several symbols. These packets could either be produced after splitting the information at the source node or could be packets with various information flows received at intermediate nodes. In order to explain the encoding process, let us presume that there are m packets $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ and must be encoding using RLNC. A coded packet \mathbf{Y}_i can be obtained by simple vector multiplication, as following equation (2.1) [43][49].

$$\mathbf{Y}_i = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} [\mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3] \quad (2.1)$$

Where, $[\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \dots, \mathbf{c}_m]$ is a coding vector whose elements (coding coefficients) are chosen randomly over the finite field \mathbf{F}_q with size q . In this manner, we can produce $(\mathbf{m}+\mathbf{k})$ encoded packets and encoding vectors, where \mathbf{k} is any number of redundancy packets. Thus, the encoding method will, in theory, produce an infinite number of encoded packets. However, there is a non-zero probability that some of the coding vectors are linearly dependent and the resulting coded packets are unable to participate to the decoding process due to the random selection of coding coefficients. The coding vector is added to the related encoded packet before transmitting the encoded packet to the network, figure 2.9 shows the header may comprise information or data related to other layers of the protocol stack needed for the packet to reach its desired destination [43][49].



Figure 2.9. Structure of Transmitted Coded Packet

2.6.4.2 Recoding

Recoding is a feature that is unique to the RLNC. Coded packets can be recombined without being decoded using recoding. Figure 2.10 illustrated the recoding process[50].

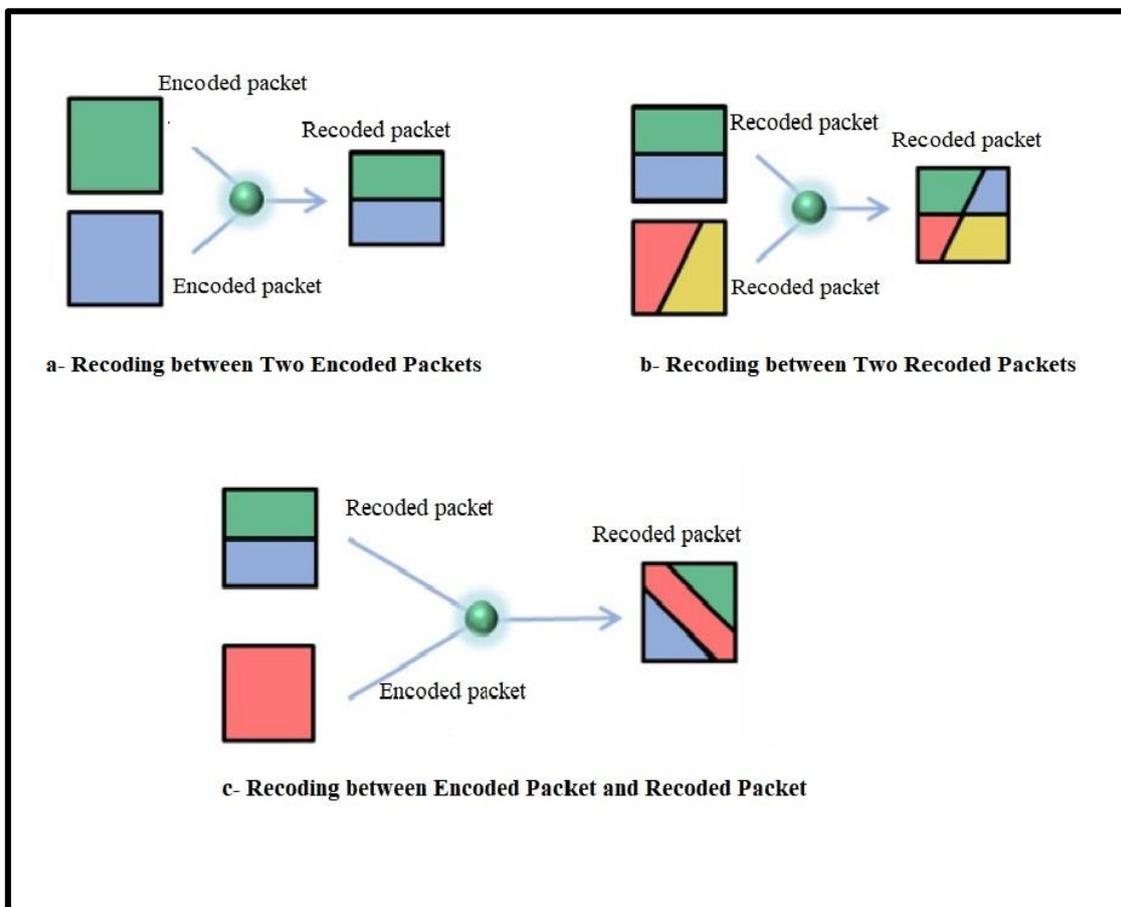


Figure 2.10. Recoding Process Types

The principle of recoding is that the coefficients produced in the recoder are multiplied by the coded packets and the coefficient vector. The encoded packets represented by the \mathbf{Y}_i column vector comprise the matrix \mathbf{C} result coefficient used to construct these encoded packets. From the received packets, these coefficients are obtained. \mathbf{G} is the coefficients produced in the recoder, \mathbf{H} is the resulting coefficients of previous and new coefficients, and the coded packets \mathbf{Z} are calculated in the below equations [43][54].

$$\mathbf{H} = \mathbf{G} * \mathbf{C} \quad (2.2)$$

$$\mathbf{Z} = \mathbf{G} * \mathbf{Y} \quad (2.3)$$

2.6.4.3 Decoding

Once the decoder node has sufficiently linearly independent coded packets, the decoding can be performed. When the decoder receives \mathbf{n} creative packets, it does not have to insert further packets into the matrix because it can decode the \mathbf{n} source symbols, by Gauss-Jordan elimination. As the destination node \mathbf{t} receives encoded packets \mathbf{Z} , which can be described by a linear matrix equation [43]:

$$\mathbf{Z} = \mathbf{H} * \mathbf{X} \quad (2.4)$$

\mathbf{H} is the transfer matrix corresponding to the product of all coefficients of all nodes, from the source node to the destination node. \mathbf{X} is the original packet. The decoding node (destination) puts the received encoded packets \mathbf{Z} in a matrix. Because the received encoded data represents the result of all encoded transmission coefficients within the packets, the original data can be retrieved by equation (2.5) [43].

$$\mathbf{X} = \mathbf{H}^{-1} * \mathbf{Z} \quad (2.5)$$

The original data cannot be retrieved unless the packets received are linearly independent. Coefficient vectors must also be independent in order to be able to invert \mathbf{H} , their determinant must not be zero [43][55].

Chapter Three

Simulate a Proposed Model of RLNC for Video Delivery

3.1 Introduction

After introducing the theoretical background of NC and offering an overview of 5G in the previous chapter. Our model is proposed based on the theoretical foundation of the NC approach. This model will apply the RLNC algorithm to video delivery. The proposed model and algorithms for each stage will be explained in detail in this chapter. Then the model will be implemented on a simple network and it will be simulated in Matlab based on certain parameters. finally, the results obtained will be presented and discussed.

3.2 Proposal Model

The block diagram of the proposed model is shown in figure 3.1, apply the RLNC algorithm to video delivery. This model consists of three stages. The first stage is the encoding process in the source node, the second stage is the recoding process in the intermediate node, and the last is the decoding process in the destination node.

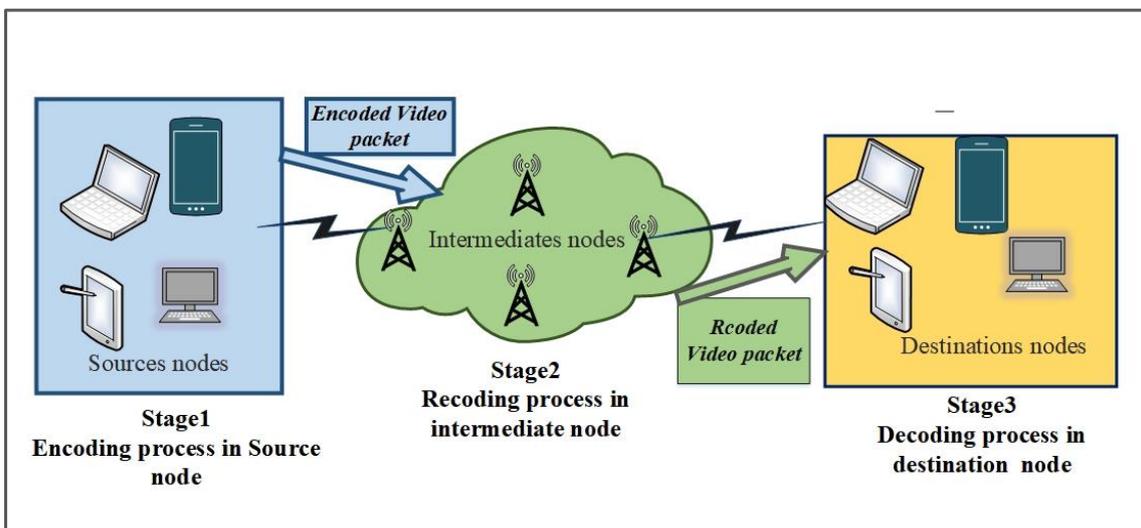


Figure 3.1.General Block Diagram

3.2.1 Encoding

The encoding process happens in the source node explained by Flowchart in figure 3.2.

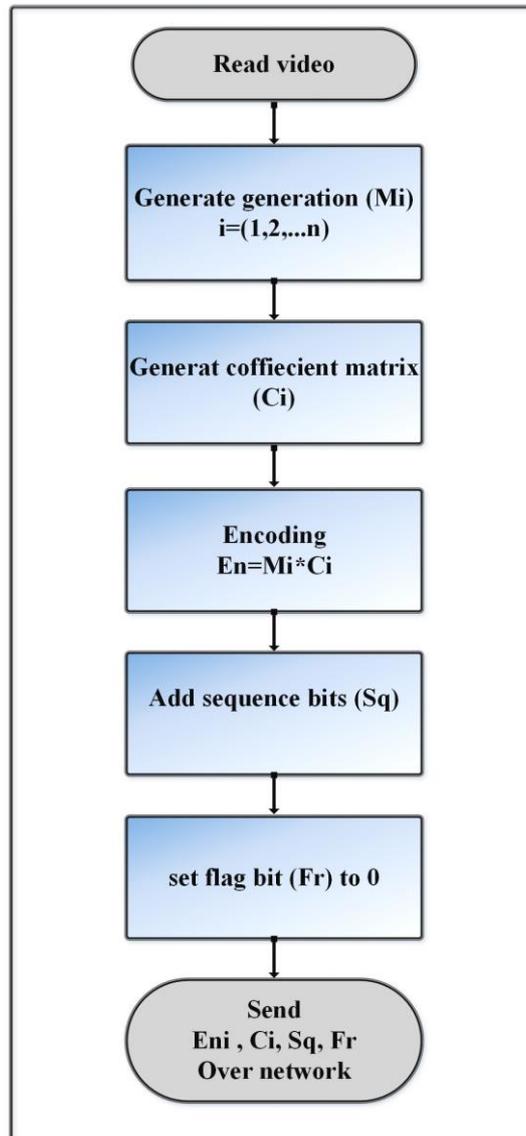


Figure 3.2. Flowchart of Encoding Process in The Source Node

The following steps explaining the algorithm of the encoding process:

Step 1:

Divide the video into groups; these groups are called generations. Each generation (**g**) contains several frames (**f**) called symbols or data packets. Then, represent the generation by a matrix called the symbol matrix (**M**) as shown in equation (3.1).

$$\mathbf{M} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \cdot \\ \cdot \\ \mathbf{f}_n \end{bmatrix} \quad (3.1)$$

Step 2:

Generate a matrix of coefficients (**C**); these coefficients are selected randomly from Galois Field. The number of matrices of coefficients (**C**) is equal to the number of generations that will be transmitted, and each generation has a coefficients matrix. Equation (3.2) shows the matrix of coefficients.

$$\mathbf{C} = [\mathbf{c}_1 \ \mathbf{c}_2 \ \dots \ \mathbf{c}_n] \quad (3.2)$$

Step 3:

Encoding operation is performed for each generation through symbols matrix (**M**) multiply by coefficients matrix (**C**) and produce the encoded packet (**En**) as shown in equation (3.3).

$$\mathbf{En}_i = \mathbf{M}_i * \mathbf{C}_i \quad (3.3)$$

$i = (1, 2, \dots, m)$, where (m) is the number of generations.

Step 4:

Add extra bits (it is called sequence bits (**Sq**)). The (**Sq**) bits indicate the index of the packet in the original video. The number of extra bits depends on the size or length of the generation as in equation (3.4)

$$\mathbf{Sq} = \log_2 l \quad (3.4)$$

Where l represents generation length or size.

Step 5:

Also, another extra bit will be add called flag bit (**Fr**).

- Initially, its value is (0) at the encoding stage.

- When $Fr=1$, indicates that the encoded packet is the result of the recoding process.
- While $Fr=0$, indicates that the encoded packet did not result from the recoding process.

Step 6:

Finally, the encoded packets with their coefficients and extra bits are sent over the network. Figure 3.3 shows the encoded packet format.



Figure 3.3. Encoded Packet Format out of Source Node

Figure 3.4 shows the shape of the frames generated after encoding process.

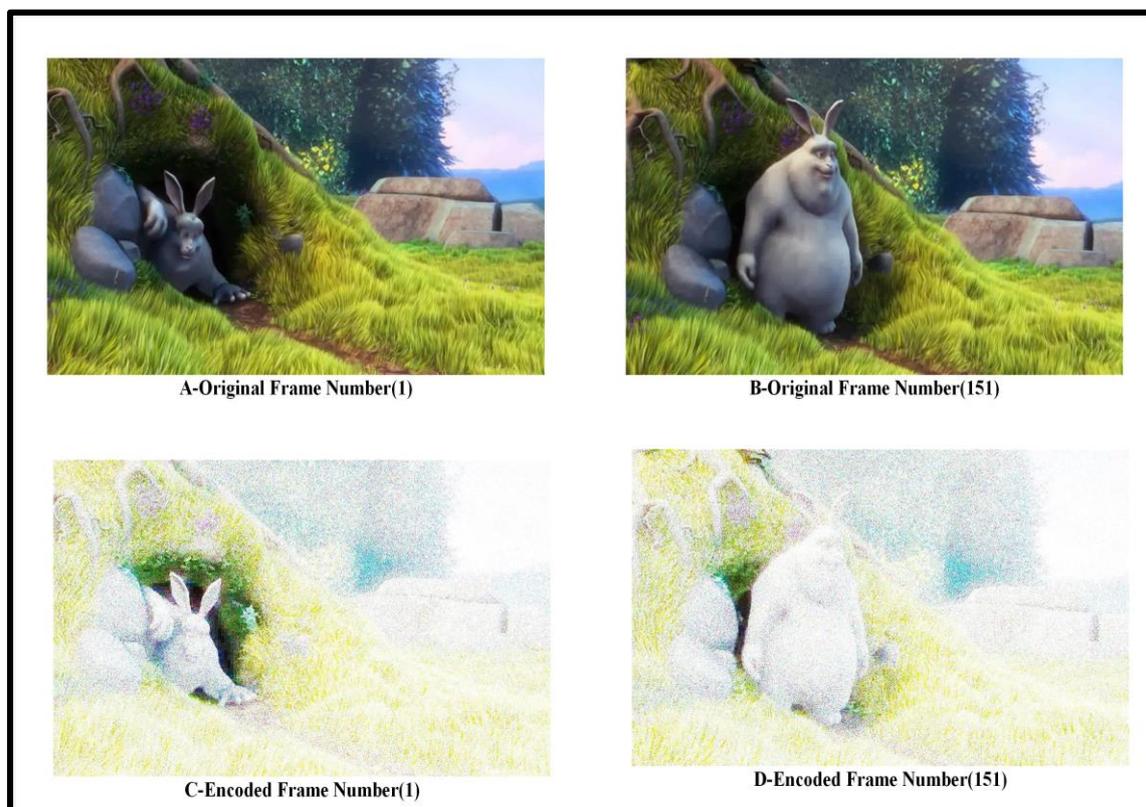


Figure 3.4. Encoding Process

3.2.2 Recoding

The recoding process is an important feature of RLNC. This process occurs in intermediate nodes. The flowchart in figure 3.5 illustrates the recoding process.

In the intermediate node, encode packets \mathbf{En} with their coefficients \mathbf{C} , the sequence bits \mathbf{Sq} , and flag bit \mathbf{Fr} are received from different nodes. The following steps explain the recoding process that is performed on encode packets.

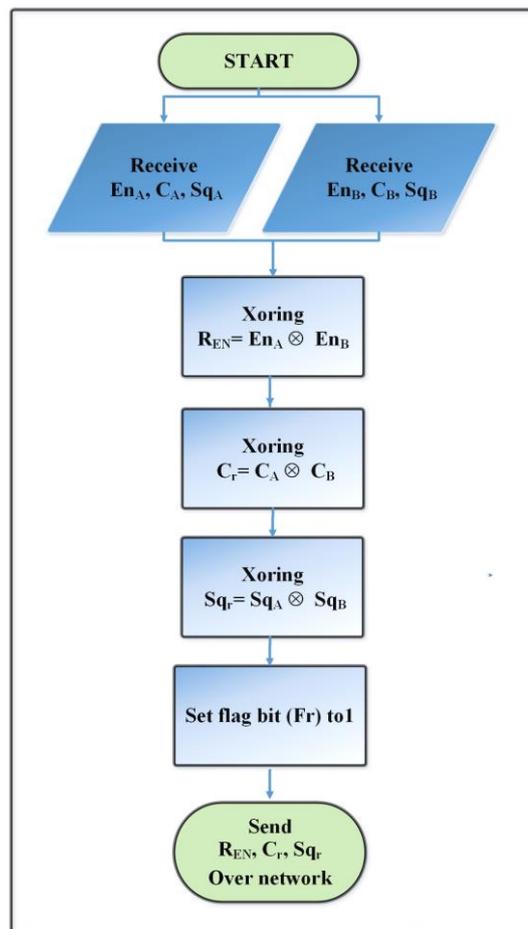


Figure 3.5. Flowchart of Recoding Process in Intermediate Node

Step 1:

The encoded packets are mixed to produce a new packet called a recoding packet, denoted by (\mathbf{R}_{En}) . To implement the merging process between encoded packets, XOR function is used. Suppose that two encoded packets come from two different nodes (A) and (B) to the intermediate node (X).

- XOR operation is executed between encoded packet \mathbf{En}_A and encoded packet \mathbf{En}_B to produce \mathbf{R}_{En} as in the below equation(3.5).

$$\mathbf{R}_{En} = \mathbf{En}_A \otimes \mathbf{En}_B \quad (3.5)$$

- XOR operation is performed between coefficients \mathbf{C}_A and coefficients \mathbf{C}_B to produce \mathbf{C}_r as in the below equation(3.6).

$$\mathbf{C}_r = \mathbf{C}_A \otimes \mathbf{C}_B \quad (3.6)$$

- XOR operation is performed between sequence bits \mathbf{Sq}_A and sequence bits \mathbf{Sq}_B to produce \mathbf{Sq}_r as in the below equation(3.7).

$$\mathbf{Sq}_r = \mathbf{Sq}_A \otimes \mathbf{Sq}_B \quad (3.7)$$

Step 2:

Set the flag (\mathbf{Fr}) to indicted to recoding process.

Step 3:

After recoded operation completion intermediate node sends \mathbf{R}_{En} , \mathbf{C}_r , \mathbf{Sq}_r , and \mathbf{Fr} over the network. Figure 3.6 shows the shape of the frames generated after recoding process.

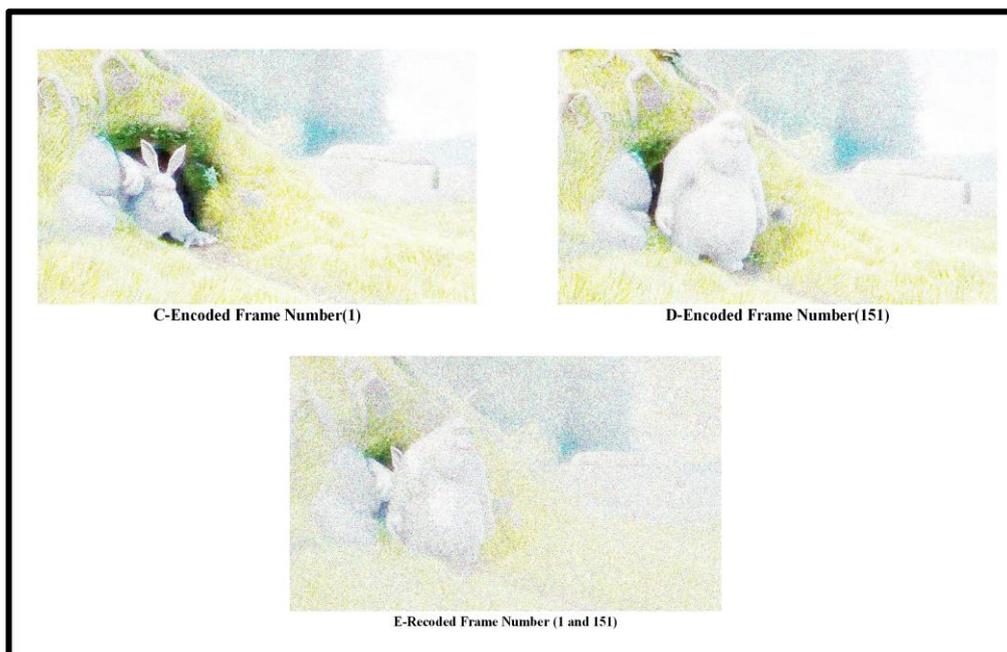


Figure 3.6. Recoding Process

3.2.3 Decoding

The decoding process occurs in the destination node at the recipient end as shown in figure 3.7. In this stage, all encoded and recoded packets will be decoded to retrieve original video packets. The decoding process begins by collecting all encoded and recoded packets that come from different nodes linked with the destination node. Then check the value of the flag **Fr** for all packets:

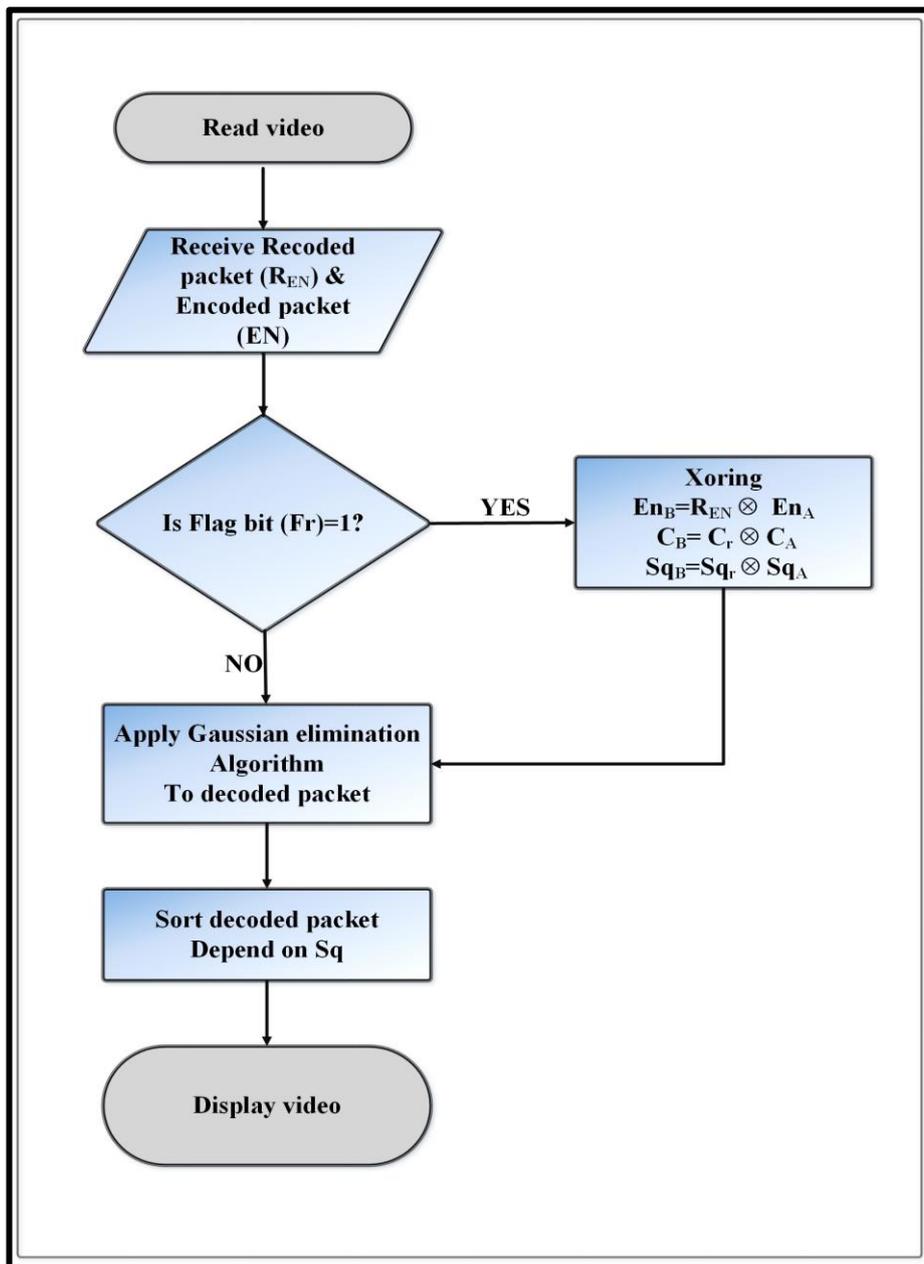


Figure 3.7. Flowchart of Decoding Process

- If the value of **Fr** bit equals to “1”, which means the packet is the recoded packet coming from an intermediate node, it has resulted from the recoding process.
- If the **Fr** bit equals to “0” that means the packet is not pass through the intermediate node and it's just the result of an encoding process.

To illustrate the decoding process, for example, **R_{En}** & **En_A** are received and the **En_B** packet is not received or missing. To retrieve lost packets and decoding them with the following steps:

Step 1:

The **Fr** bit of packets is checked, indicating that:

- **En_A** packet is only an encoded packet.
- **R_{En}** packet is coming from the intermediate node and results from the recoding process.

Step 2:

To decode **R_{En}** and retrieve **En_B** as follows:

- XOR operation is executed between Recoded packet **R_{En}** and encoded packet **En_A** to retrieve **En_B** as in the below equation(3.8).

$$\mathbf{En}_B = \mathbf{R}_{En} \otimes \mathbf{En}_A \quad (3.8)$$

- XOR operation is performed between coefficients **C_r** and coefficients **C_A** to retrieve **C_B** as in the below equation (3.9).

$$\mathbf{C}_B = \mathbf{C}_r \otimes \mathbf{C}_A \quad (3.9)$$

- XOR operation is performed between sequence bits **Sq_A** and sequence bits **Sq_B** to retrieve **Sq_r** as in the below equation(3.10).

$$\mathbf{Sq}_B = \mathbf{Sq}_r \otimes \mathbf{Sq}_A \quad (3.10)$$

Step 3:

After retrieving the lost packets \mathbf{En}_B , encoded packets are decoded by using the Gaussian Elimination Algorithm. To explain Gaussian Elimination Algorithm by flowing :

- To implement the Gaussian elimination, the equations system is established as shown in equation (3.11).

$$\begin{bmatrix} \mathbf{c}_{11} & \mathbf{c}_{12} & \dots & \mathbf{c}_{1i} \\ \mathbf{c}_{21} & \mathbf{c}_{22} & \dots & \mathbf{c}_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{i1} & \mathbf{c}_{i2} & \dots & \mathbf{c}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_i \end{bmatrix} \quad (3.11)$$

- Create an augmented matrix equation shown in equation (3.12)

$$\begin{bmatrix} \mathbf{c}_{11} & \mathbf{c}_{12} & \dots & \mathbf{c}_{1i} \\ \mathbf{c}_{21} & \mathbf{c}_{22} & \dots & \mathbf{c}_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{i1} & \mathbf{c}_{i2} & \dots & \mathbf{c}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_i \end{bmatrix} \quad (3.12)$$

- In X variables, the column vector is moved to name the rows of the matrix. The original row operations are now performed to place the augmented matrix in the form of the upper triangle as shown in equation (3.13).

$$\left[\begin{array}{cccc|c} \mathbf{c}_{11} & \mathbf{c}_{12} & \dots & \mathbf{c}_{1i} & \mathbf{a}_1 \\ \mathbf{c}_{21} & \mathbf{c}_{22} & \dots & \mathbf{c}_{2i} & \mathbf{a}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{c}_{i1} & \mathbf{c}_{i2} & \dots & \mathbf{c}_{ii} & \mathbf{a}_i \end{array} \right] \quad (3.13)$$

- Solve the equation of row (i) for variable x_i , then substitute it into the equation of the previous row (i-1) to solve the equation of the previous row ..etc.

Step 4:

Then, decoded packets are arranged depended on the Sq bit for each packet.

Finally, the video is displayed.

3.3 Simulation of Proposal Model

Simulation of the proposal model above-mentioned use PC with CPU core i9, 9900k 3.8 GHz, RAM 32 GB, Windows 10, and Matlab 2019.

3.3.1 The Parameters of The Simulation Model

There is more than one parameter that affects processing algorithms, but for our proposal model, we will rely on two parameters Galois field and Video resolution.

A. Galois Field

The random numbers of coefficients are selected from the Galois field. The Galois field is a finite field with p^n Element, where p is random prime numbers and n called degree is a positive number. When it implements the model in Matlab, the GF function is used $gf = GF(p, n)$. GF generates a Galois field array from matrix p , where p random matrix. The Galois field has 2^n elements, where n is an integer from 1 through 16. Different values are taken for n to see their effect on encoding, recoding, and decoding.

B. Video Resolution

The proposed model works on various display resolutions mentioned in Chapter Two. But, we chose three types of video resolutions:

- 4K=3840*2160
- HD=1280*720
- SD=640*480

3.3.2 Implemented the Proposal Model

The proposed model is implemented on the simple network as shown in figure 3.8. (S) represents source node implements encoding process, node (X)

represents intermediate node implements recoding process, and (d_1 , d_2 , d_3 , and d_4) nodes represent destinations nodes implement decoding process.

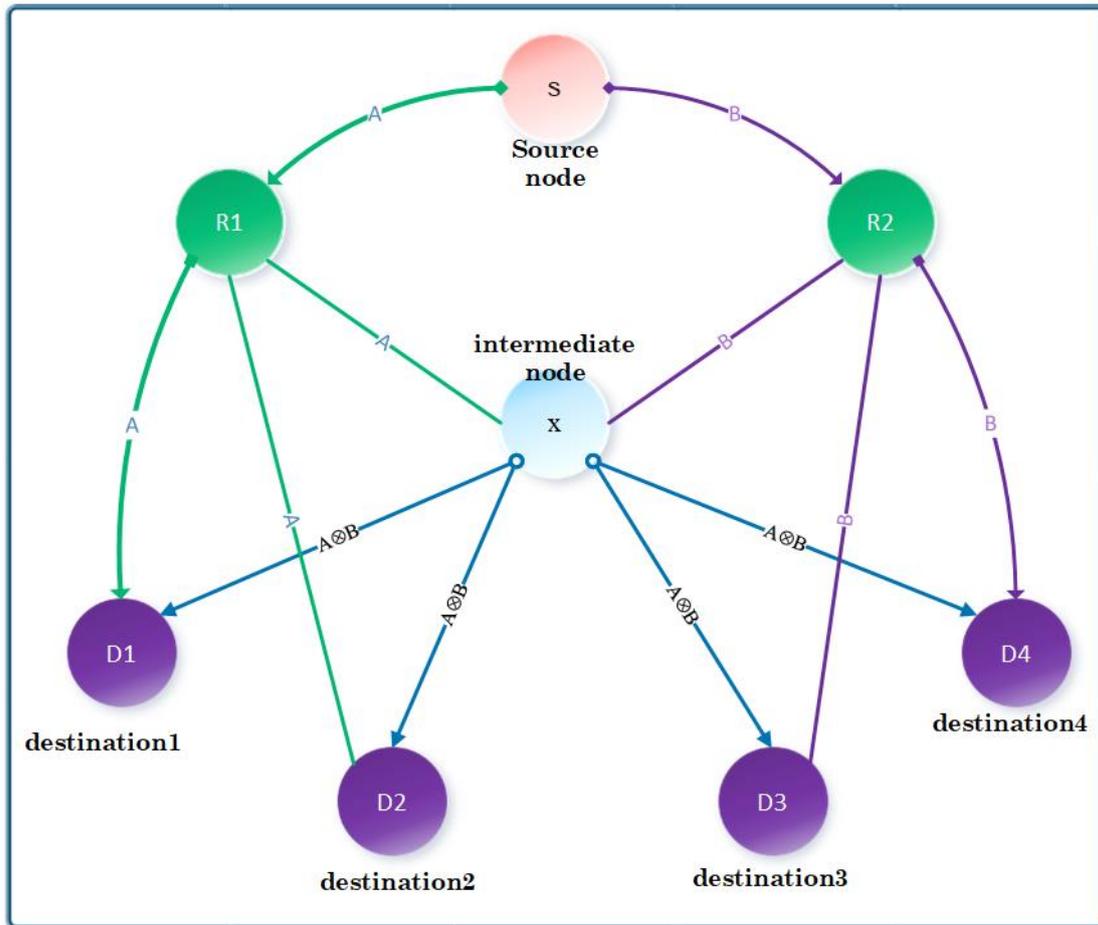


Figure 3.8. Network Model

3.4 Results

In this section, the results of two parameters affecting video transmission are presented; video resolution, and G_f .

3.4.1 Video Resolution

A. SD Resolution Video

- 1- The simulation results when applying the proposed algorithm use SD video with $G_f=4$ is shown in figure 3.9.

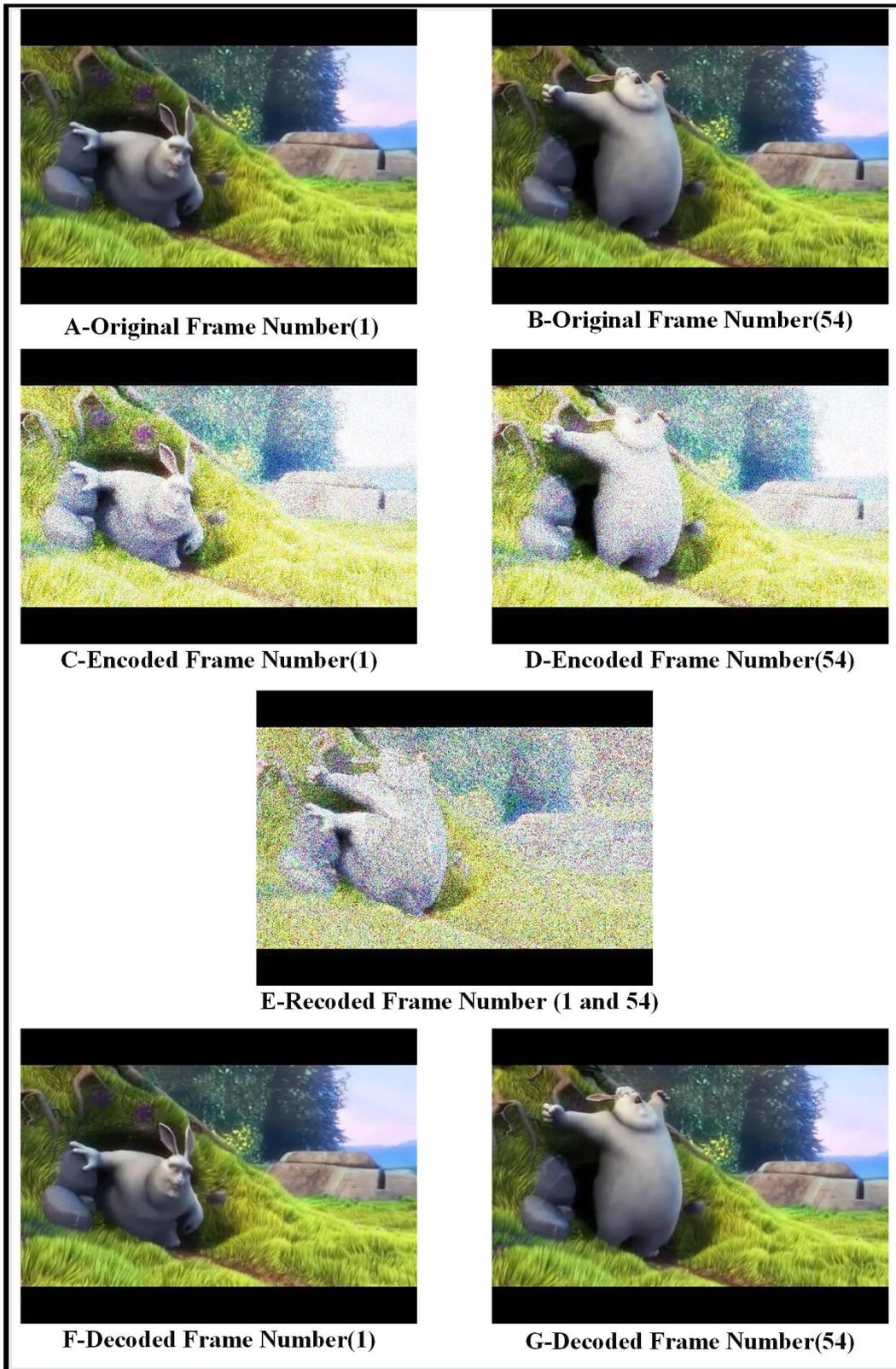


Figure 3.9. SD Resolution Video , GF=4.

2. At GF= 8, the simulation results is shown in figure 3.10.

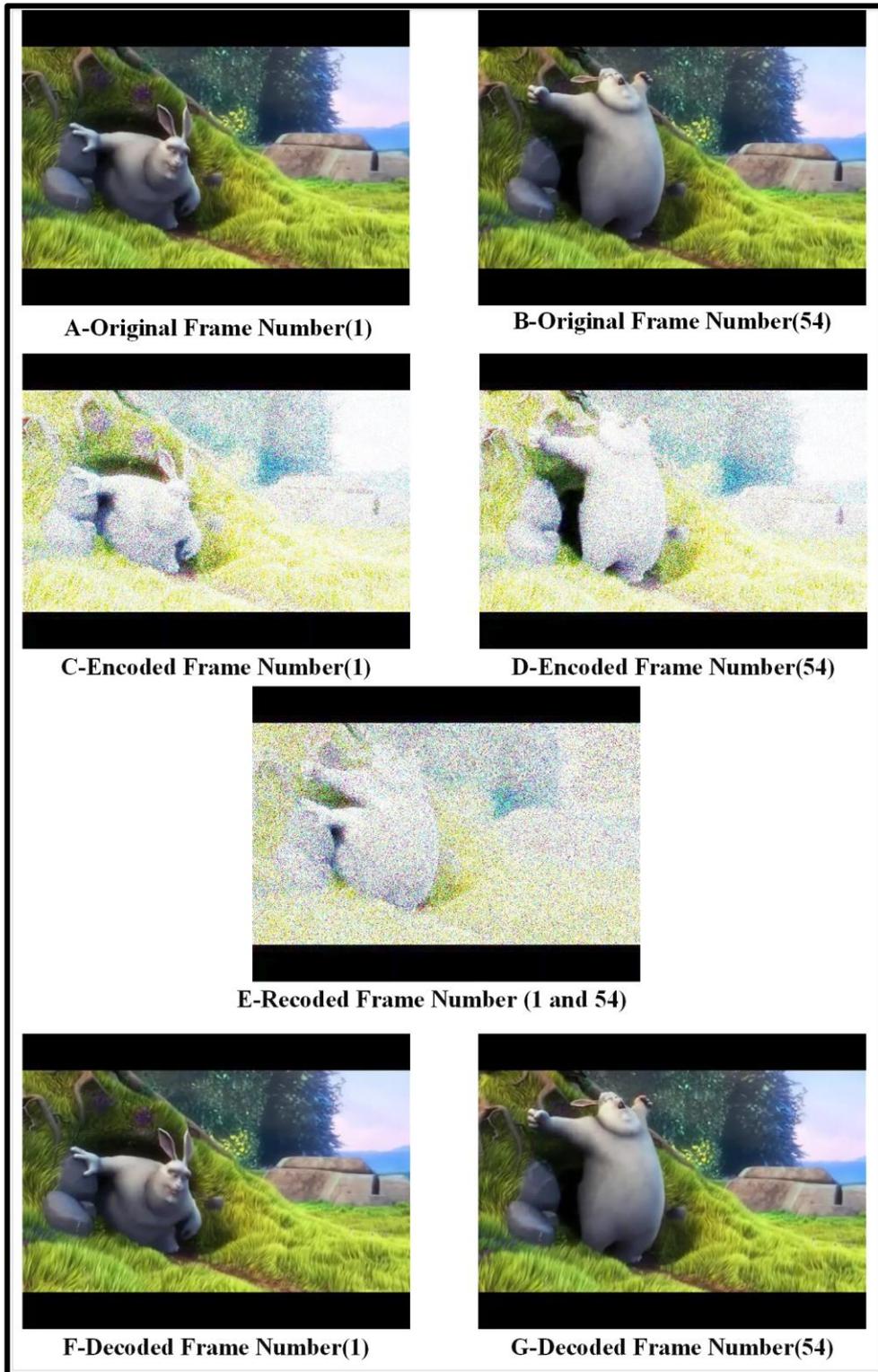


Figure 3.10. SD Resolution Video , GF=8.

3. At GF= 16, the simulation results is shown in figure 3.11.

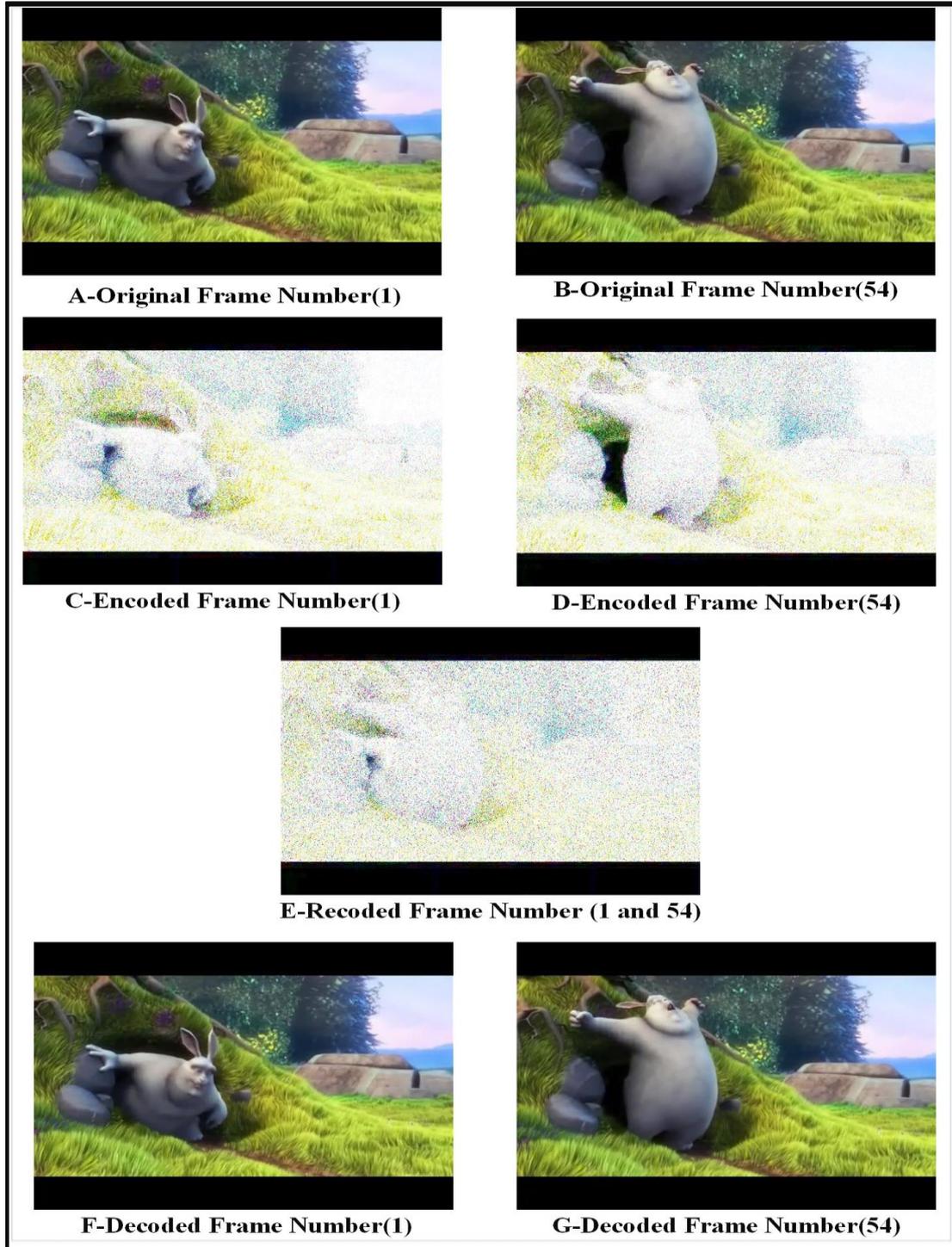


Figure 3.11. SD Resolution Video , GF=16

4. For comparison of SD video encoding with $GF = 4, 8, 16$, histogram is used to illustrate the difference between frames as shown in figure 3.12.

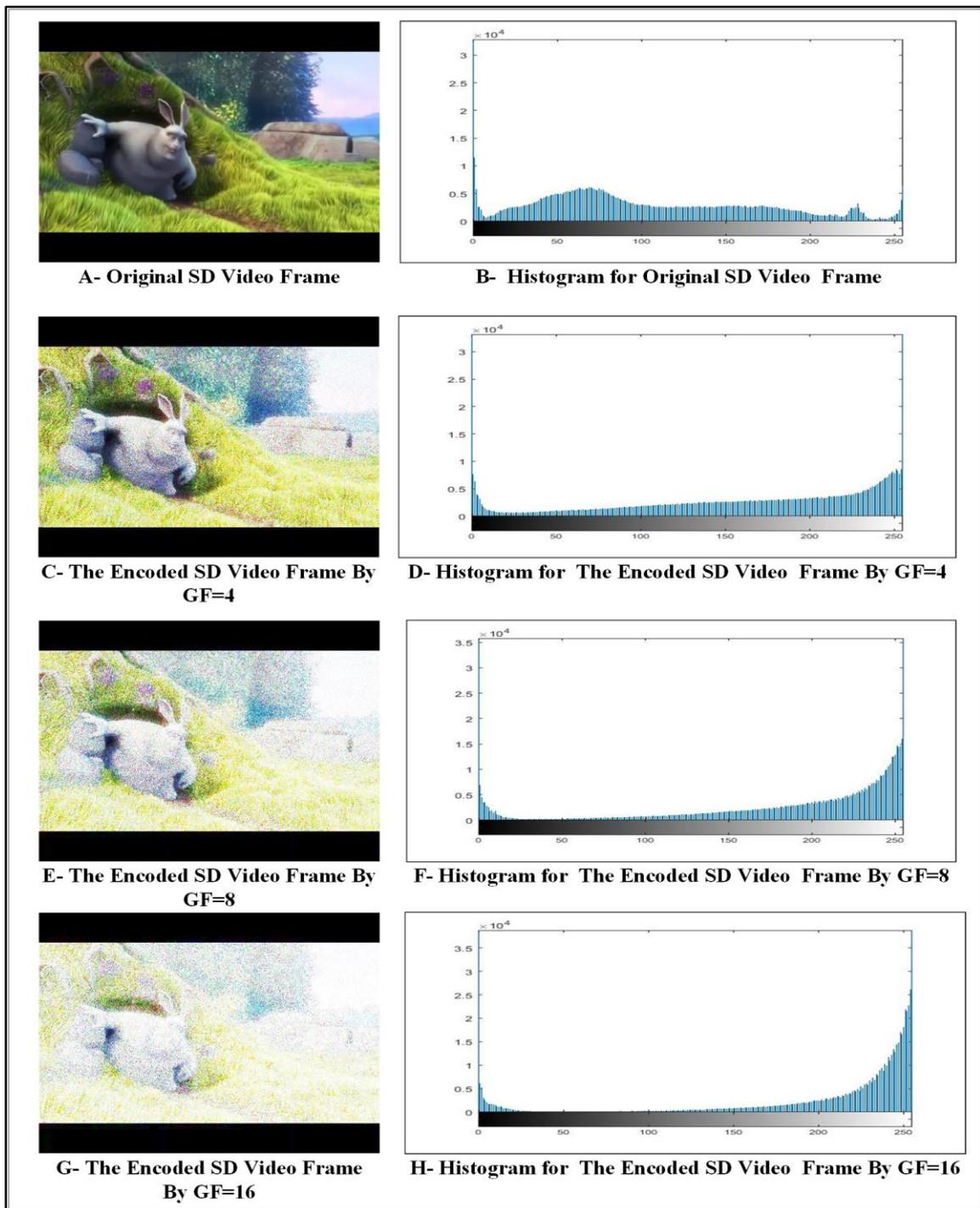


Figure 3.12. Comparison of SD Frames Encoded with Different Degrees of $GF = 4, 8,$ and 16 .

B. HD Resolution Video

1. At GF= 4, the simulation result is shown in figure 3.13.

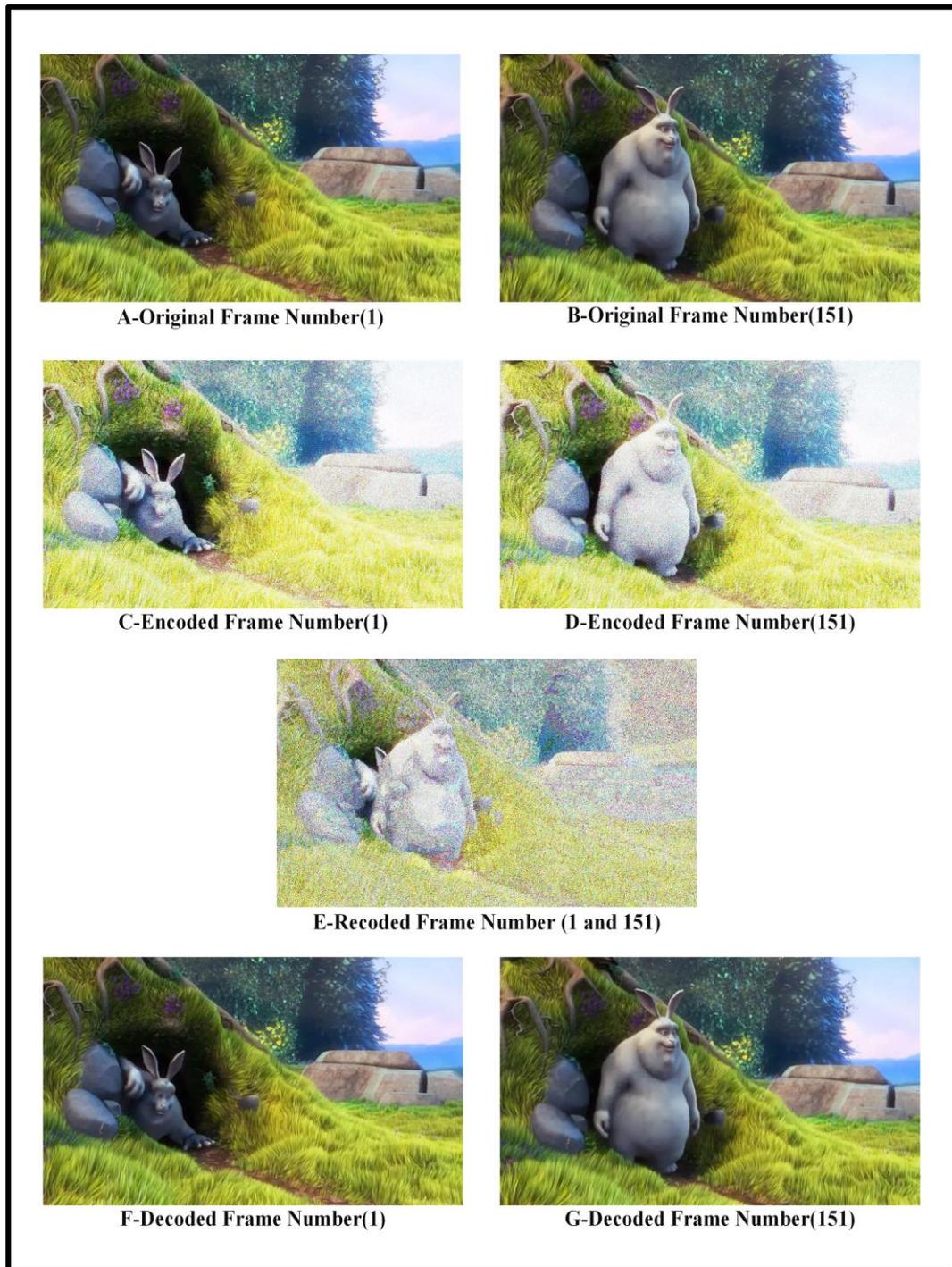


Figure 3.13. HD Resolution Video , GF=4.

2. At GF= 8, the result is shown in figure 3.14.

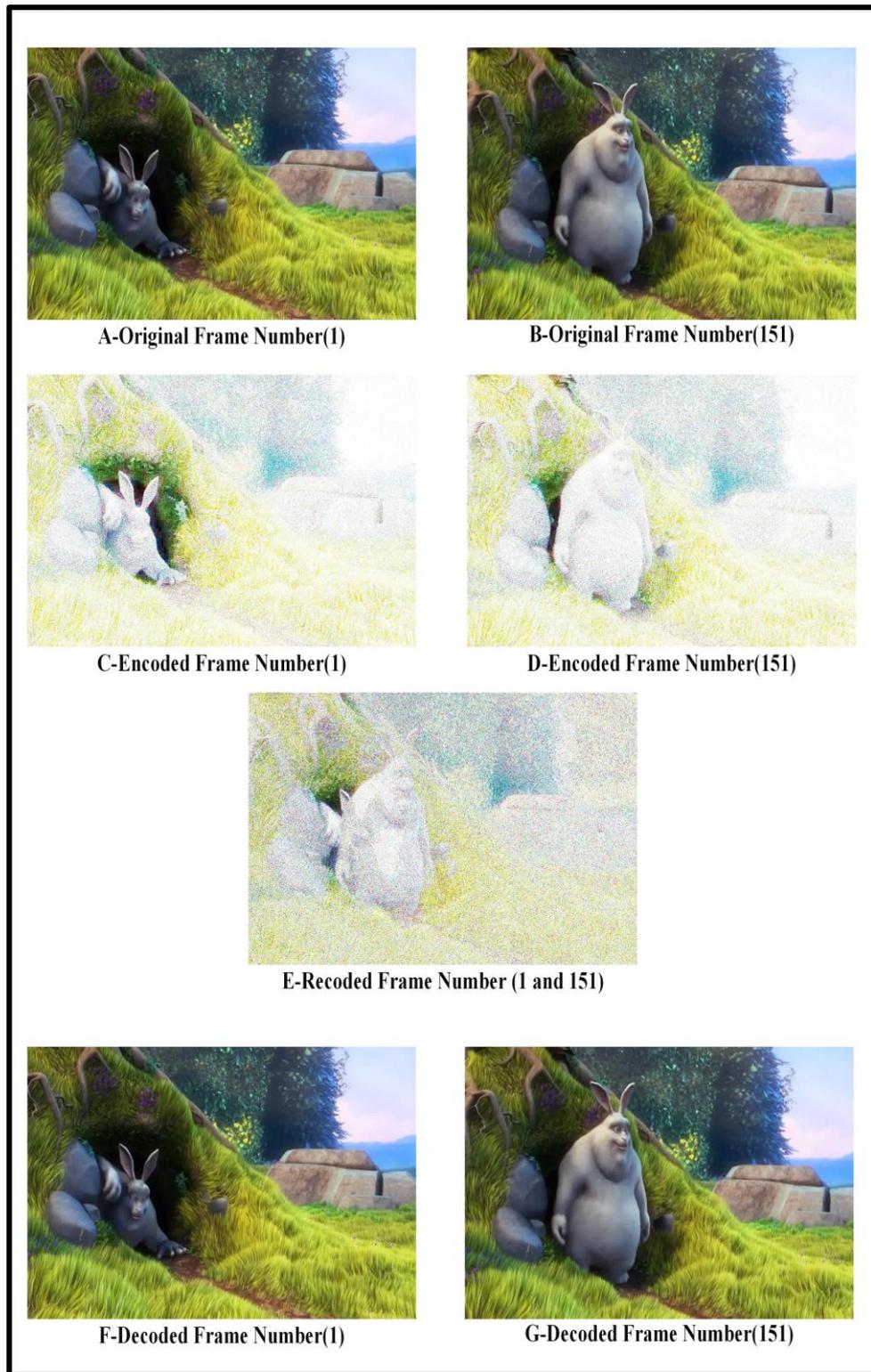


Figure 3.14. HD Resolution Video , GF=8.

3. At GF= 16, the result is shown in figure 3.15.

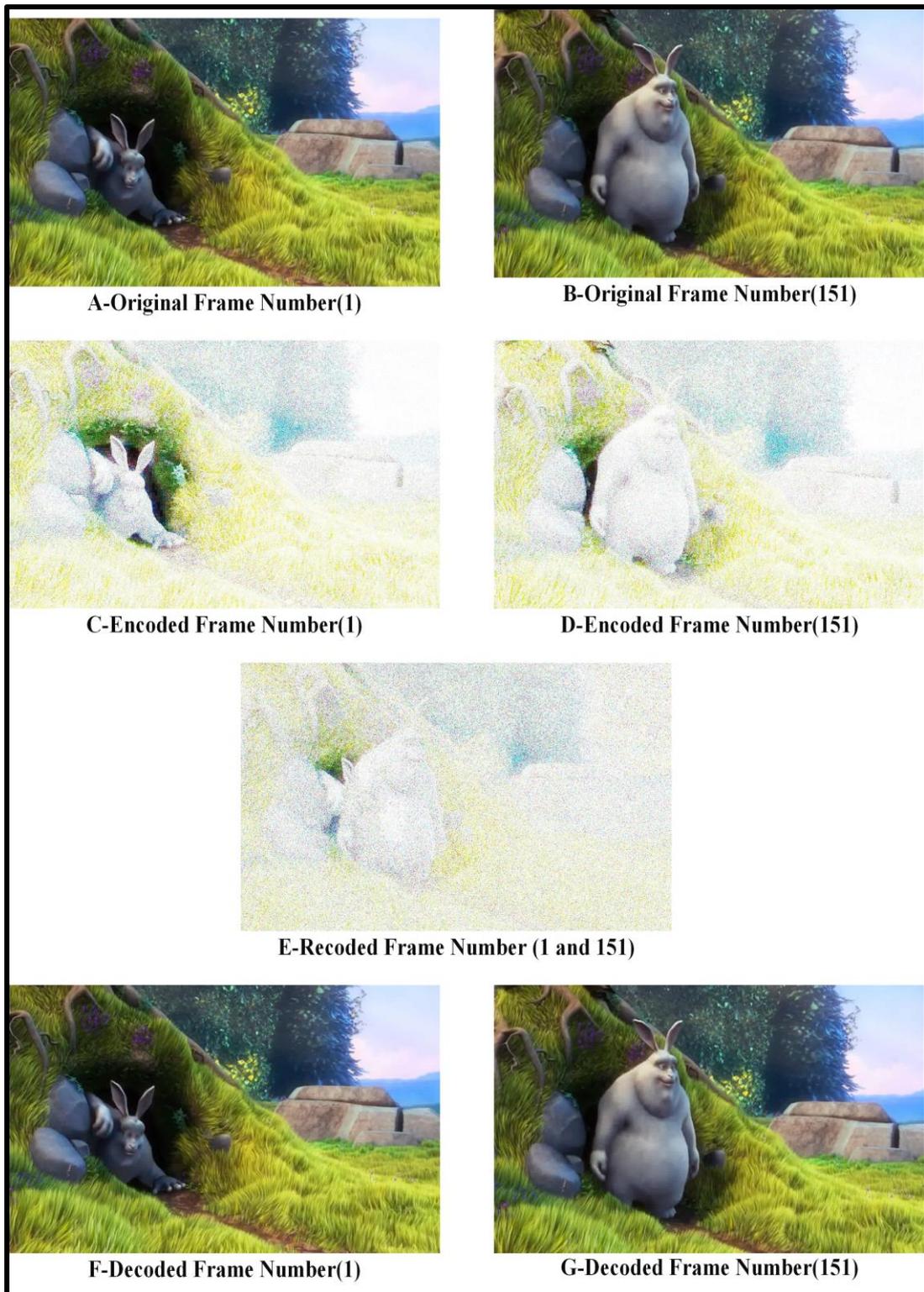


Figure 3.15. HD Resolution Video , GF=16.

4. Comparison of HD video encoding with difference GF =4, GF= 8, GF= 16 is shown in figure 3.16

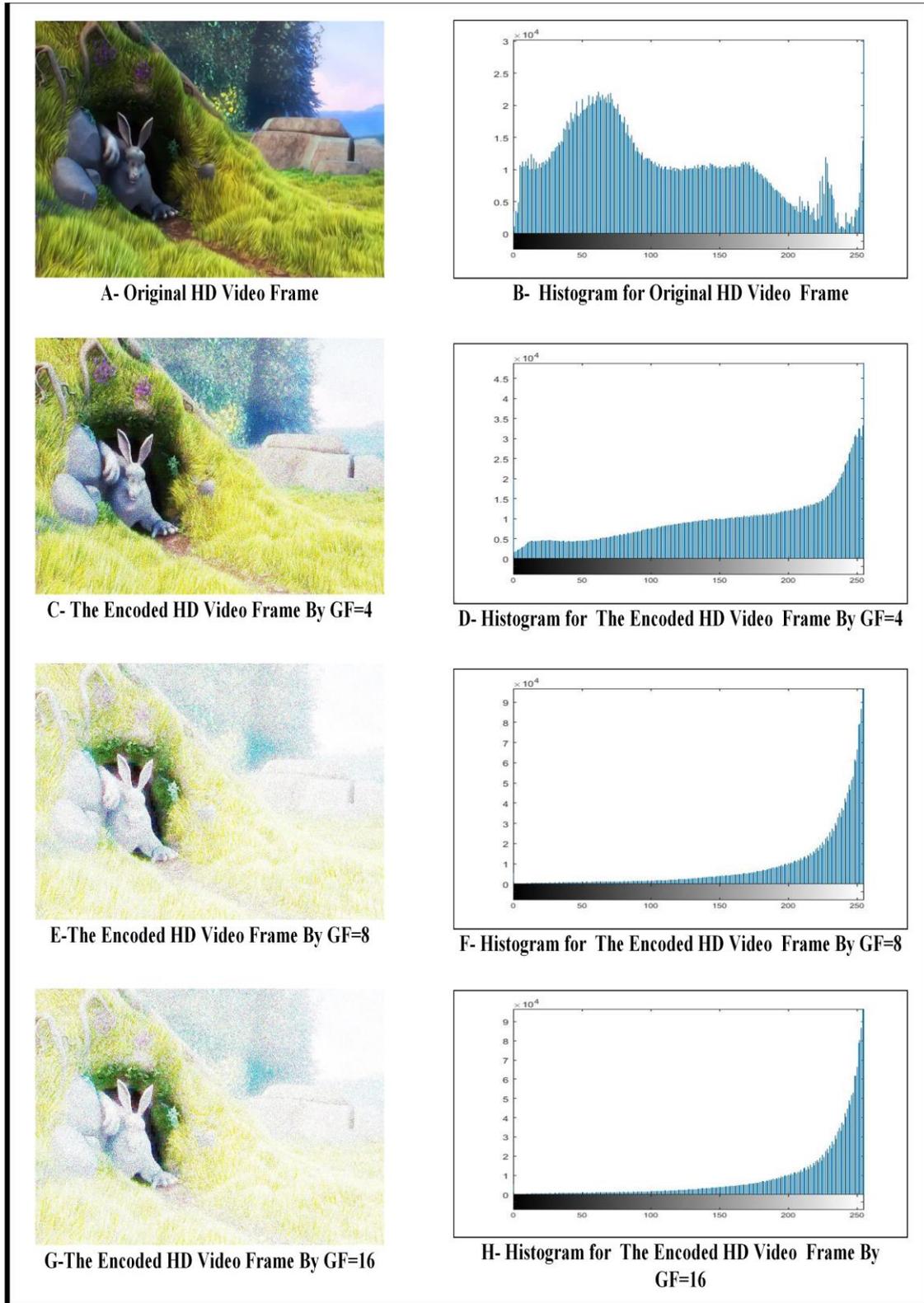


Figure 3.16. Comparison of HD Frames Encoded with Different Degrees of GF = 4, 8, and 16.

C. 4K Resolution Video

1. GF= 4, the simulation is shown in Figure 3.17.

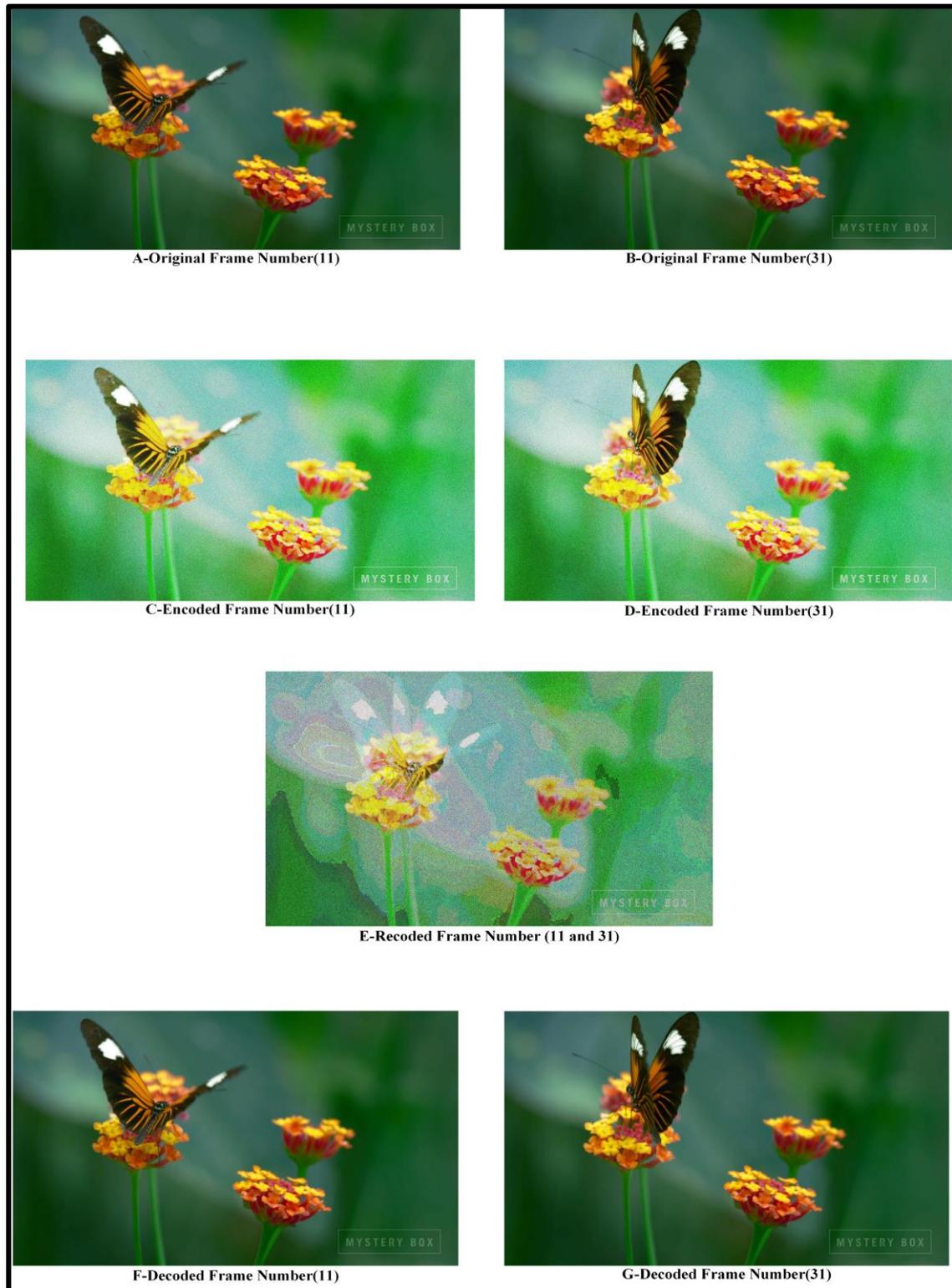


Figure 3.17. 4K Resolution Video , GF=4.

2. $GF=8$, the result is shown in figure 3.18.



A-Original Frame Number(11)



B-Original Frame Number(31)



C-Encoded Frame Number(11)



D-Encoded Frame Number(31)



E-Recoded Frame Number (11 and 31)



F-Decoded Frame Number(11)



G-Decoded Frame Number(31)

Figure 3.18. 4K Resolution Video , $GF=8$.

3. At GF= 16, the simulation is shown in figure 3.19.

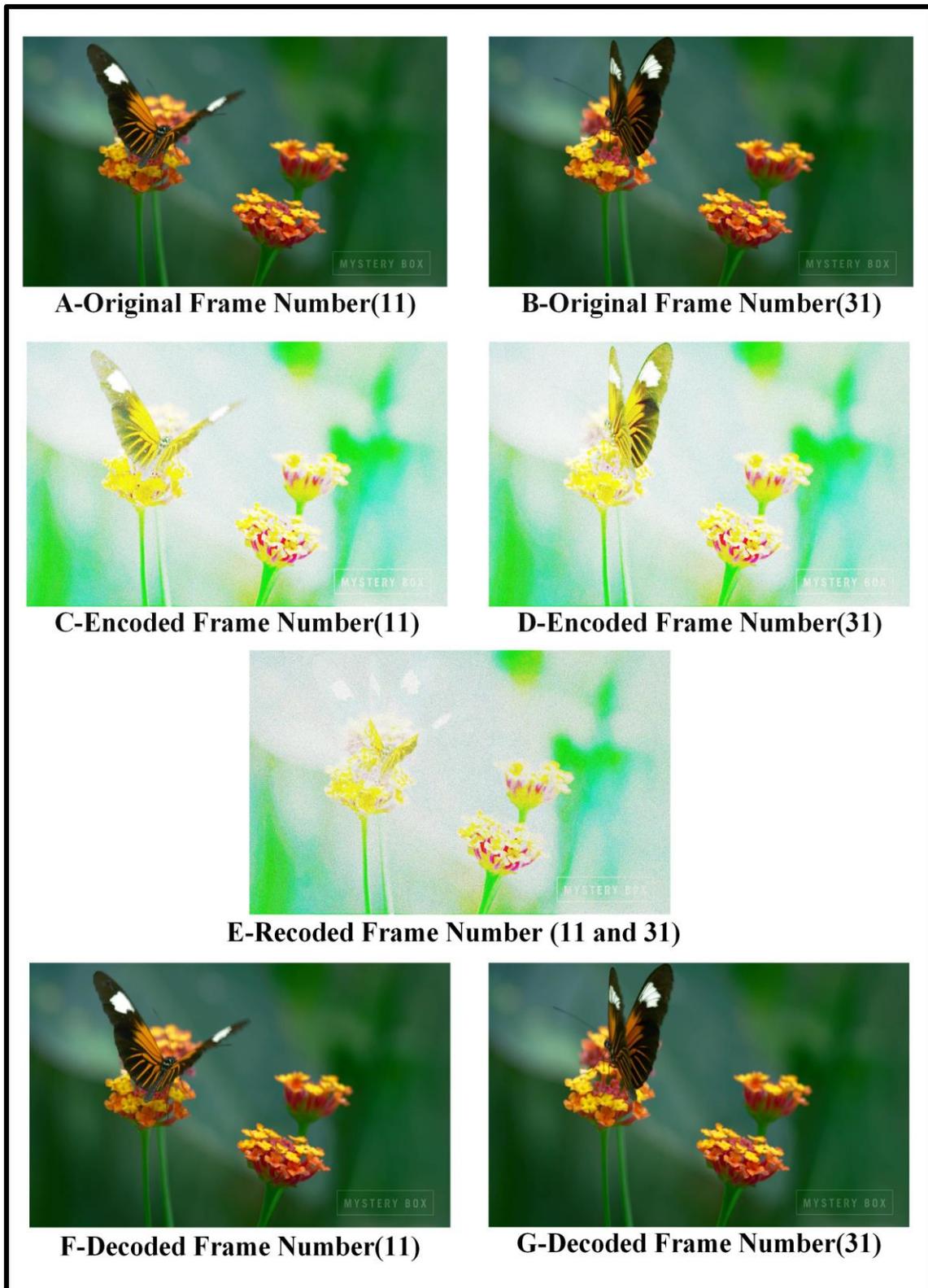


Figure 3.19. 4K Resolution Video , GF=16.

4. Comparison of HD video encoding with difference GF =4, 8, and 16 is shown in figure 3.20

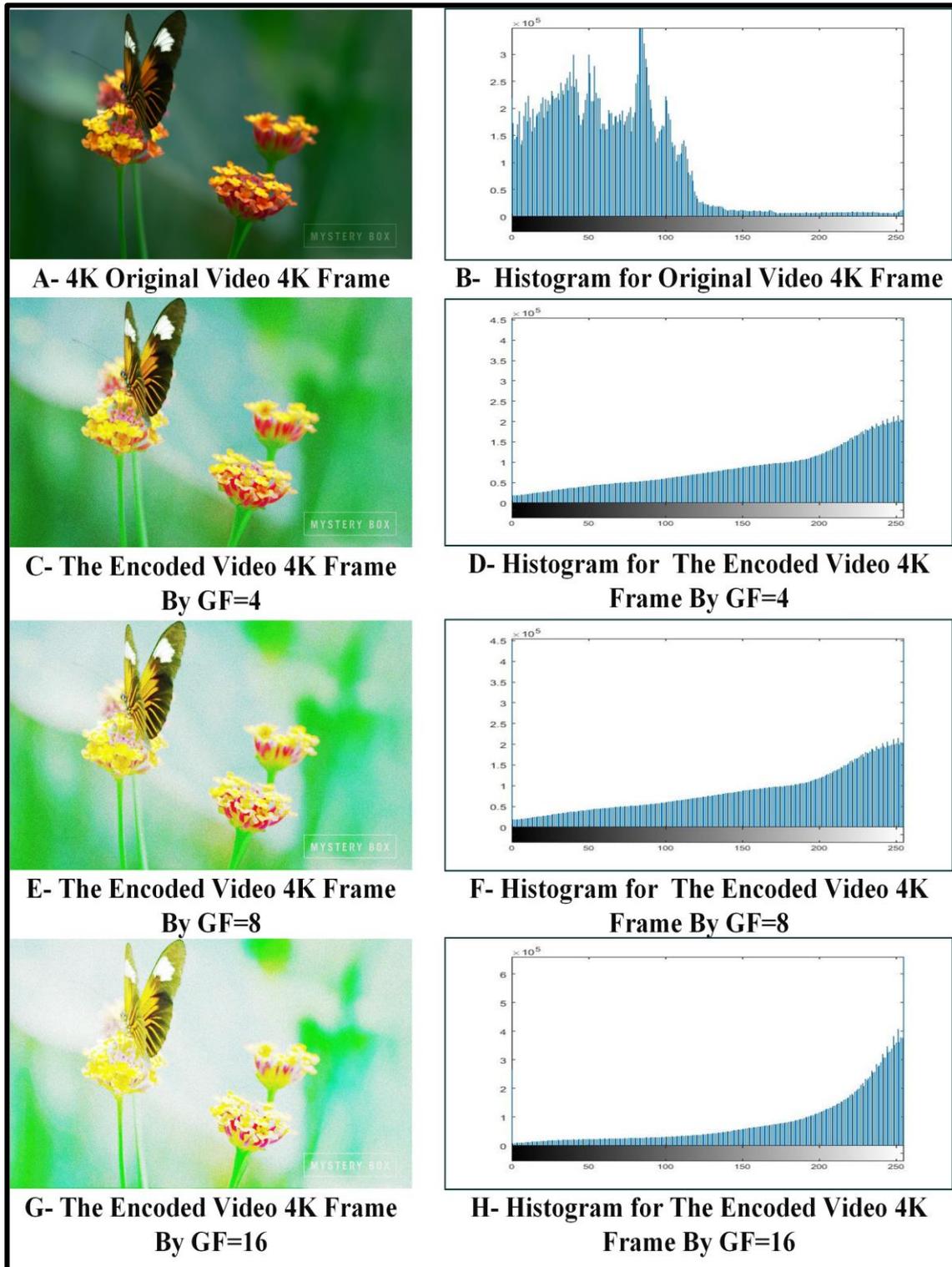


Figure 3.20. Comparison of 4K Frames Encoded with Different Degrees of GF = 4, 8, and 16.

3.4.2 Simulation Time.

Simulation Time for Different Values of GF.

1. Simulation time for each stage when use 4K video, GF=4, 8, and 16 is shown in figure 3.21.

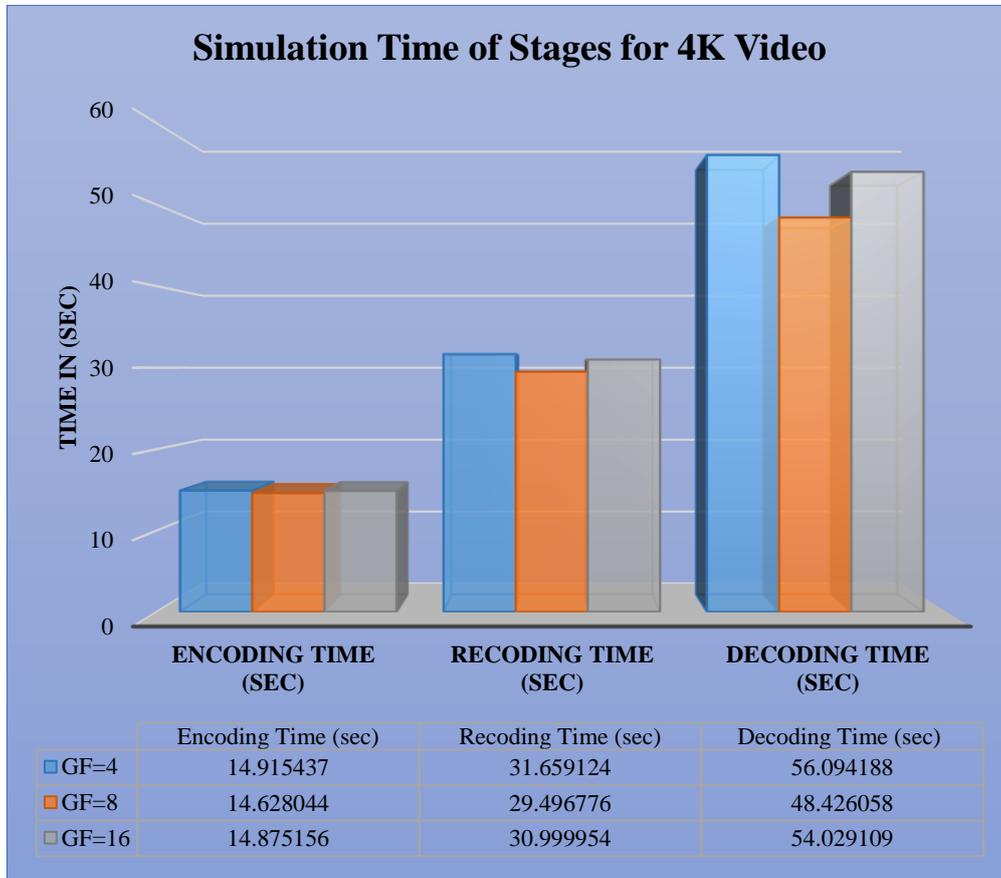


Figure 3.21. Simulation Time for Each Stage When Use 4K Video with Different GF Values.

2. Simulation time for each stage when use HD Video, and GF=4,8, and 16 is shown in figure 3.22.

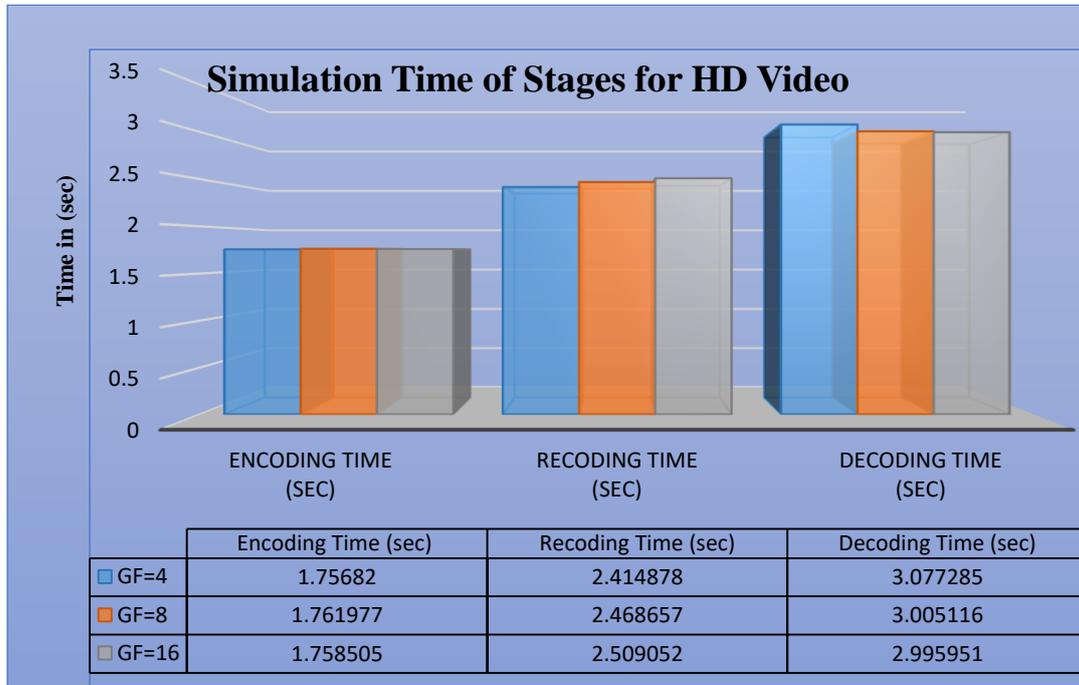


Figure 3.22. Simulation Time When Use HD Video with Different GF Values.

- Simulation time for each stage when use SD Video, GF=4, 8, and 16 is shown in figure 3.23.

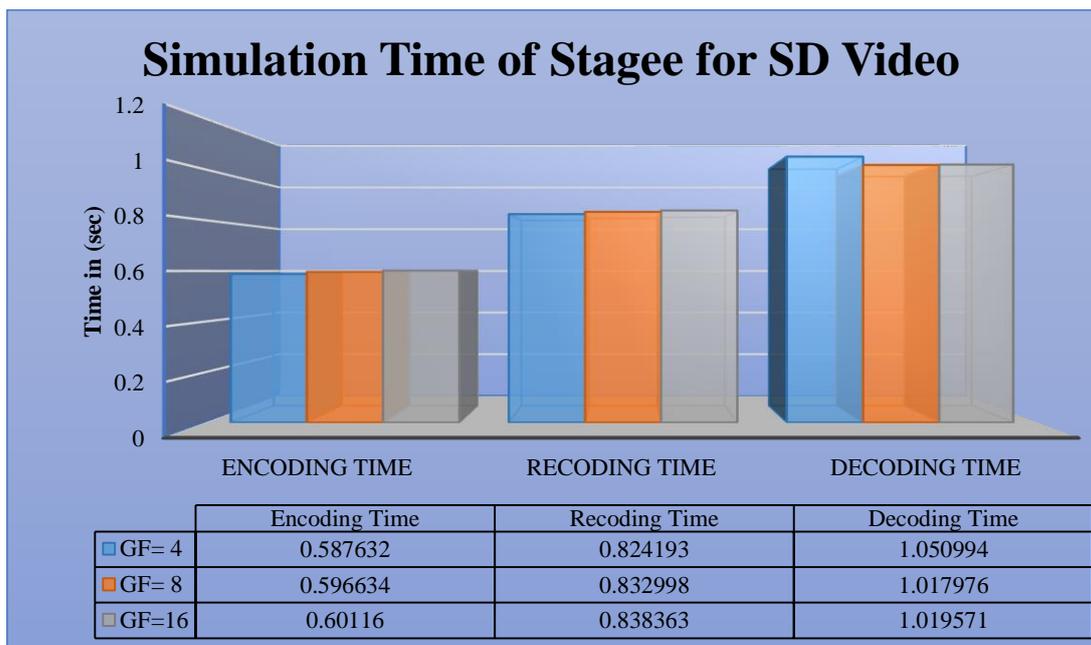


Figure 3.23. Simulation Time When Use SD Video with Different GF Values

4. The simulation time for different Videos resolution with GF=4, 8, and 16 shown in figure 3.24.

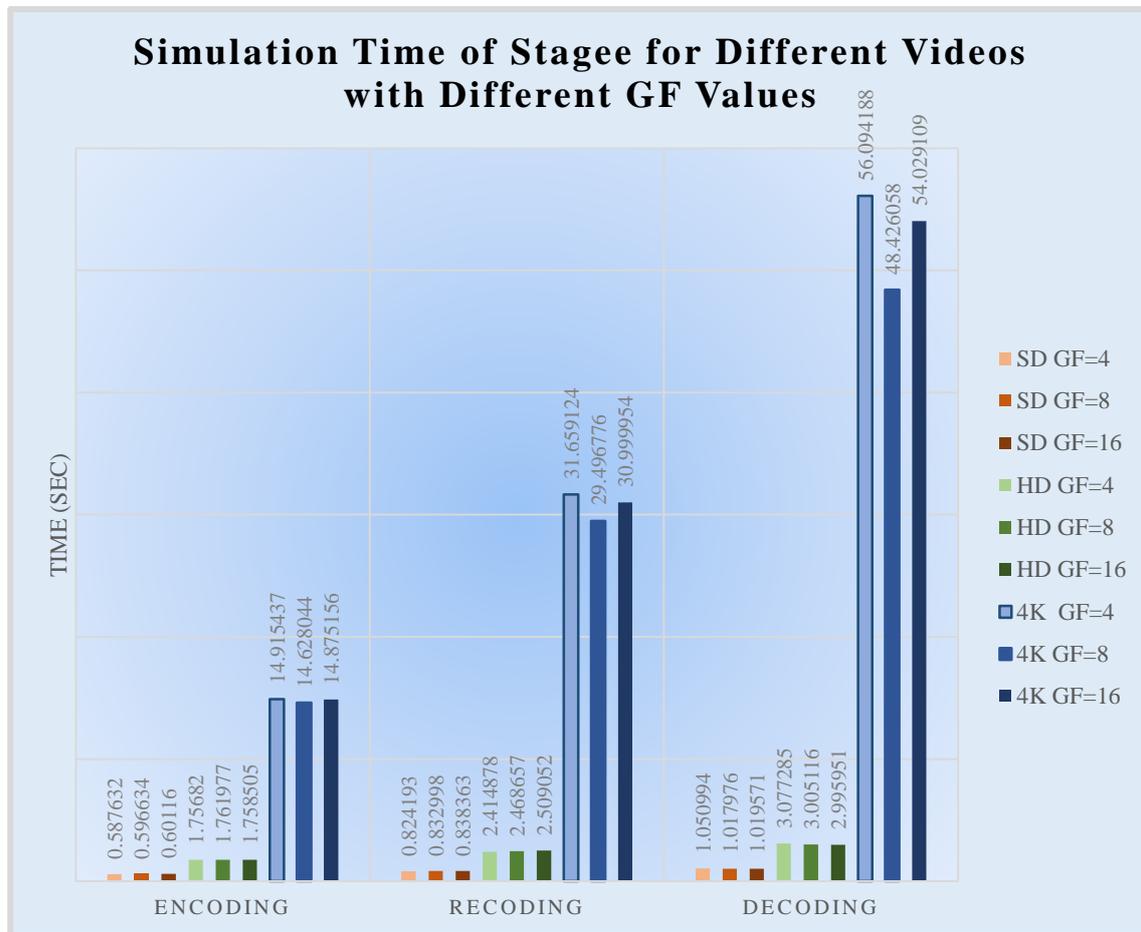


Figure 3.24. Simulation Time for Different Videos with Different GF Values

3.5 Discuss The Results

- AS the GF degree increases, the length of the Galois field increases, thus increasing the variety of random numbers used in coding. Figures (3.12,3.16,3.20) show a comparison between the original frame histogram and the histogram of the frame encoded with different degrees of GF. when the GF degree increase, the difference between the original frame histogram and the encoded frame histogram increases. In turn, leads to coding reliability and thus video transmission reliability.

- In figures 3.24, it is clear that the effect of changing degree of GF does not noticeably affect the simulation time for each stage within the chosen gf values. While effect of changing the video resolution noticeably affects on the execution time for each stage. when the video resolution increases, the simulation time for each operation increases.
- Throughput is increased by 25% than traditional approaches that do not use RLNC by increasing bandwidth capacity utilization as more than one packet is transmitted in the transmission process, thus reducing the number of transmission times.
- Finally, the different resolution videos were delivered, and the BER between the video frames at the transmitter end and the video frames at the receiving end was equal to zero. We can observe this through a histogram of the original and decoded frame in the figure 3.25 for 4K video.

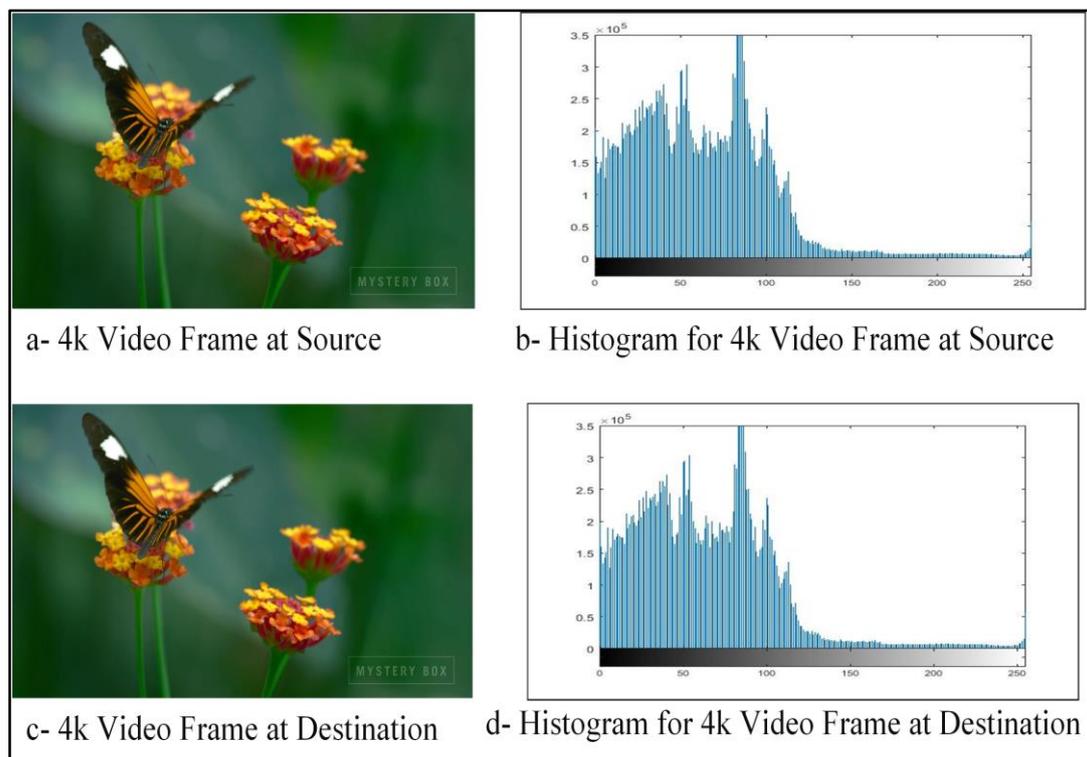


Figure 3.25. Frames and Their Histograms at Source and Destination

Chapter Fourth

FPGA Implementation of Proposal Model

4.1 Introduction

The proposed model was explained for the application of the RLNC algorithm on the delivery of the video and its three stages; encoder, recoder, and decoder, then simulated in the Matlab and achieved the main objective of the project which is to deliver the video. In the current chapter, an overview of the (Field Programmable Gate Arrays) FPGA will be presented. The three stages for the proposed model are implemented using one of the Xilinx FPGA devices, The Nexys7 kit board (Xilinx part number XC7A100T-1CSG324C). by Using Matlab and Vivado programs. Synthesis of two types of video files (SD, and HD), the synthesis produces an output number of reports. The most common output files are the timing report.

4.2 Field Programmable Gate Array (FPGA)

FPGA is a digital integrated circuit (IC) that includes configurable (programmable) logic blocks alongside configurable interconnects between those blocks. They are built in a way that engineers can configure the design in the field and perform a huge range of tasks [56].

4.2.1 FPGA Basic Parts

Xilinx FPGAs compose of arrays of configurable components, the three main configurable components being. Configurable logic blocks (CLBs), Input/output blocks (IOBs), and Programmable interconnect. The CLBs offer functional components for constructing the logic of the user. IOBs offer an interface between the package pins and the internal signal lines. Programmable interconnect offers routing paths for connecting inputs and outputs of CLBs and IOBs [57].

4.2.2 Nexys A7

The three stages for the proposed model are implemented using one of the Xilinx FPGA device, The Nexys7 kit board (Xilinx part number XC7A100T-1CSG324C). This device has features including [58]:

- 15,850 Programmable logic slices, each with four 6-input LUTs and 8 flip-flops.
- 1,188 Kbits of fast block RAM.
- Six clock management tiles, each with a phase-locked loop (PLL).
- 240 DSP slices.
- Internal clock speeds exceeding 450 MHz. A

Figure 4.1 shows an overview of the Nexys7 kit board

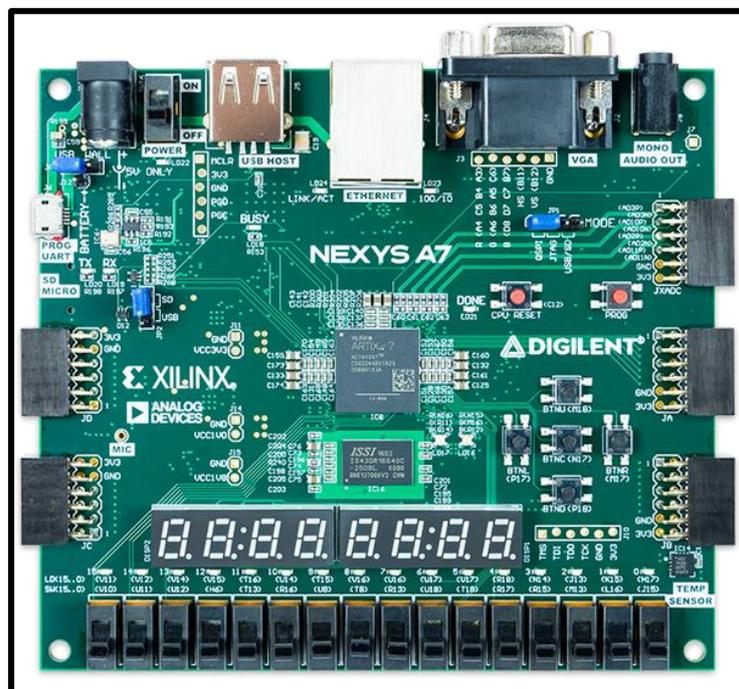


Figure 4.1. Nexys A7 Overview

4.3 FPGA in the Loop

FPGA-in-the-Loop (FIL) is one of the (Hardware Description Languages) HDL Verification methods to test the behavior of the designed algorithm for FPGA hardware stand-alone implementation [59]. This method is a combination

of software and hardware methods and inherits the advantages of both. In the FIL approach, the design is deployed to hardware and runs in real-time. However, the surrounding components are simulated in a software environment. This approach facilitates software flexibility with real-world precision and hardware speed performance [60]. FIL accelerates the simulation of video processing utilizing Simulink and FPGA. During the simulation in Matlab, video data is transmitted from the host PC where Matlab is installed to the FPGA board via a Joint Test Action Group (JTAG) link. The input video data will then be processed by the FPGA board and the resulting output data will be returned to the Host PC. The data received will then be analyzed by Matlab [61].

4.4 Implementation of The Three Stages for The Proposed Model

The three stages of video delivery by using RLNC: encoder, recoder, and decoder are designed in Matlab Simulink and are implemented on FPGA by using FIL.

4.4.1 Encoder Model

The encoder model is designed in Matlab Simulink as shown in figure 4.2, where it comprises 'From Multimedia File block' to read the video file to be sent, 'Chroma Resampling Block' color space conversion has been an important aspect of the processing and transfer of image and video. Real-time images and videos are preserved in RGB Color Space, the processing of an image in the RGB color space with a range of RGB values for each pixel is not an efficient method. To accelerate some of the processing steps, many transmissions, video, and imaging standards utilize various video signals, such as YCbCr, to make a method for conversion between formats necessary. 'Frame to Stream block' to convert a frame to stream. 'Random Number Generator block' to generate random numbers. 'Encoder-Subsystem block' performs encoding RLNC operation on streams of frames. 'Encoded_Y block', 'Encoded_Cb block', and 'Encoded_Cr block' to output the encoded streams of frames to the network.

4.4.1.1 Implementation of Encoder

The ‘Encoder block subsystem’ is doing the encoding process, so it will be implemented on FPGA by the following steps:

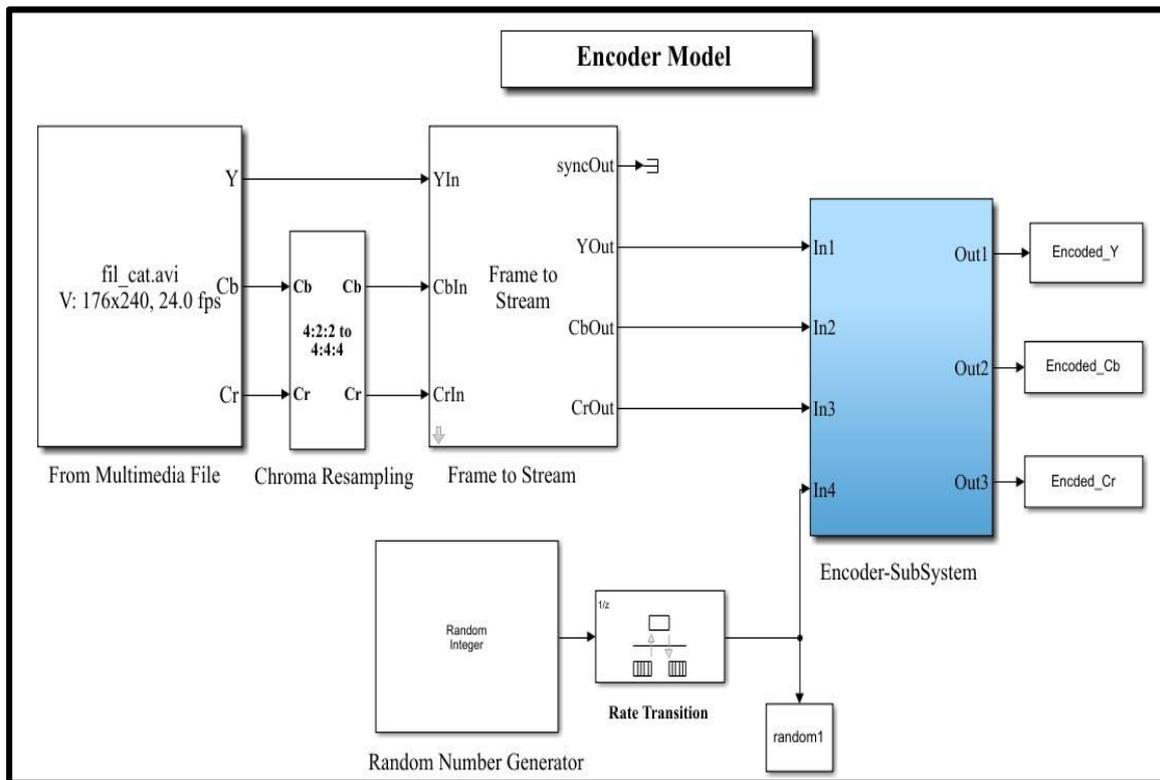


Figure 4.2. Encoder Model

- Encoder-Subsystem block is the block for which VHDL is generated.
- Using the FPGA-in-the-Loop Wizard in Matlab the FPGA programming file for the Encoder-Subsystem block is generated
- The Encoder-subsystem FIL is generated.
- The FPGA programming file for Encoder-Subsystem is loaded into FPGA.

Then the synthesis has been implemented on the ‘Encoder Subsystem block’. The summary of the utilization report is given in Table 4.1.

Table 4.1. Device Utilization Summary for Encoder

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	80	126800	1%
Number Used as Flip Flops	80		
Number of Slice LUTs	4	63400	1%
Number Used Exclusively as Route-Thrus	4		
Number with Same-Slice Register Load	4		
Number of Occupied Slices	19	15850	1%
Number of LUT Flip Flop Pairs Used	76		
Number with An Unused LUT	72	76	94%
Number of Fully Used LUT-FF Pair	4	76	5%
Number of Unique Control Sets	1		
Number Of Bonded IOBS	84	210	40%
Number of BUFG/BUFGCTRLs	1	32	3%
Number Used as BUFGs	1		
Number Of DSP48E1s	3	240	1%
Average Fanout of Non-clock Nets	2.07		

4.4.1.2 Internal Details and Processing for Encoder

Encoding processes on video take some time. To reduce the time spent in the encoding process, it takes advantage of converting video into three streams where these streams are encoded in parallel simultaneously. The Encoder-Subsystem block contains three sub-blocks as shown in figure.4.3. These sub-blocks work in parallel to encode these streams simultaneously, which reduces encoding time.

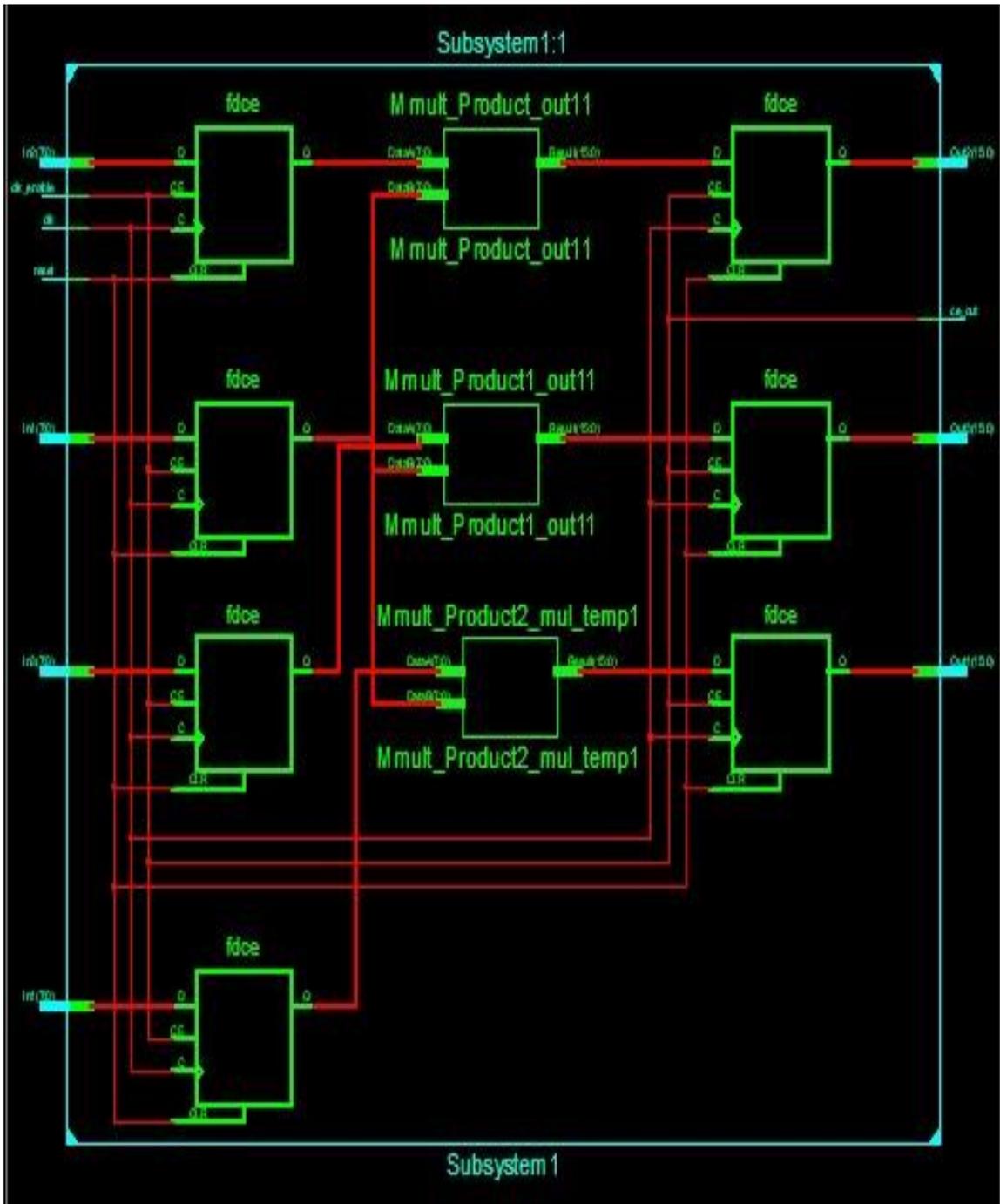


Figure 4.3. RTL Internal Details for Encoder-Subsystem Block

4.4.1.3 Time for Encoder

The encoding time for two different videos (HD and SD) was calculated by ModelSim several times and each time on a different number of frames as shown in table 4.2.

Table 4.2. Encoding Time for SD and HD Videos

Number of frames	Encoding time (ms) SD (240*176) video	Encoding time (ms) HD (1280*720) video
7	0.633595	13.823855
13	1.689550	36.863995
19	2.745595	59.903855
25	3.801585	82.943995
30	4.773095	104.140505

From the chart shown in figure 4.4, it is noticed that the encoding process on videos occurs in real-time, where HD video with frame rate 30 fps is processed in less than one second.

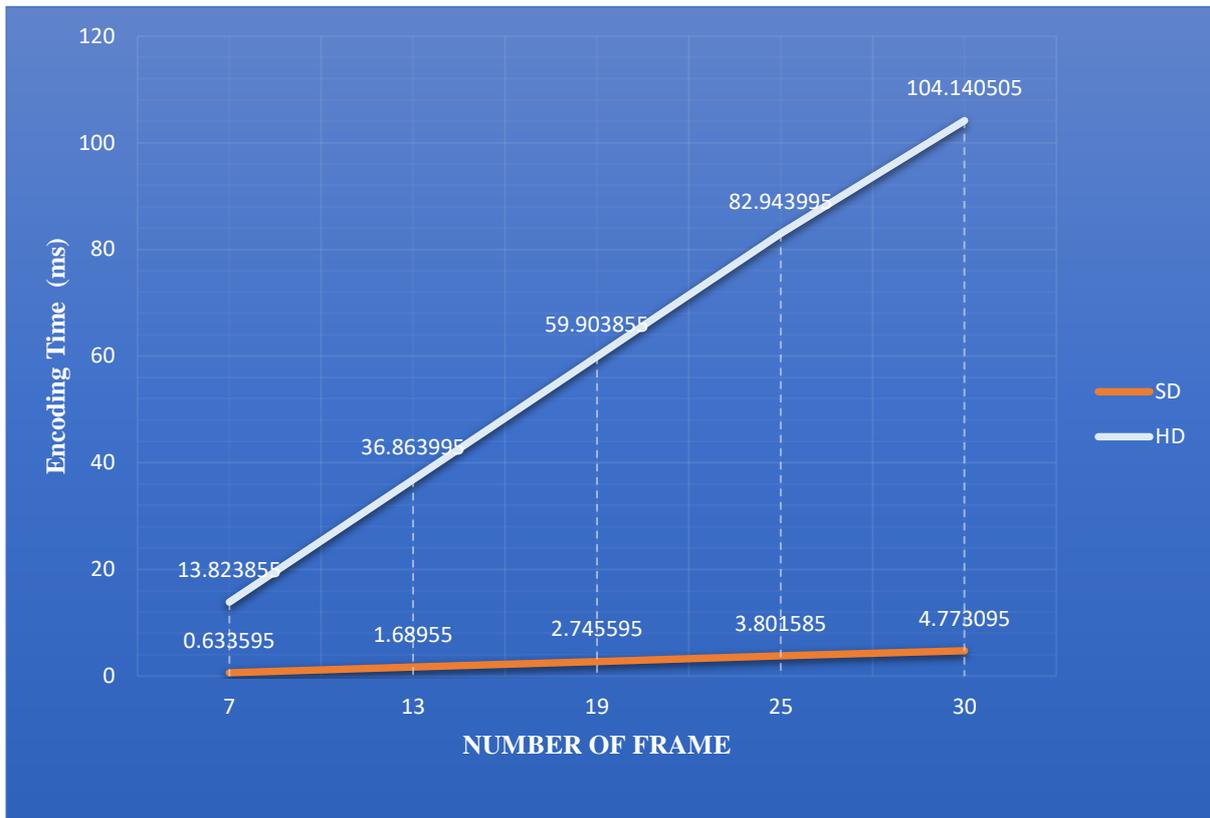


Figure 4.4. Encoding Time with Different Number of Frames

Figure 4.5 show the timing signals measured by ModelSim for encoding process for 30 frame HD video.

4.4.1.4 The Frame Resulting from The Encoding Process

The encoding process takes place in the source node, it encodes the original frames to produce the encoded frame. Figure 4.6 shows the video frame before and after the Encoding process.

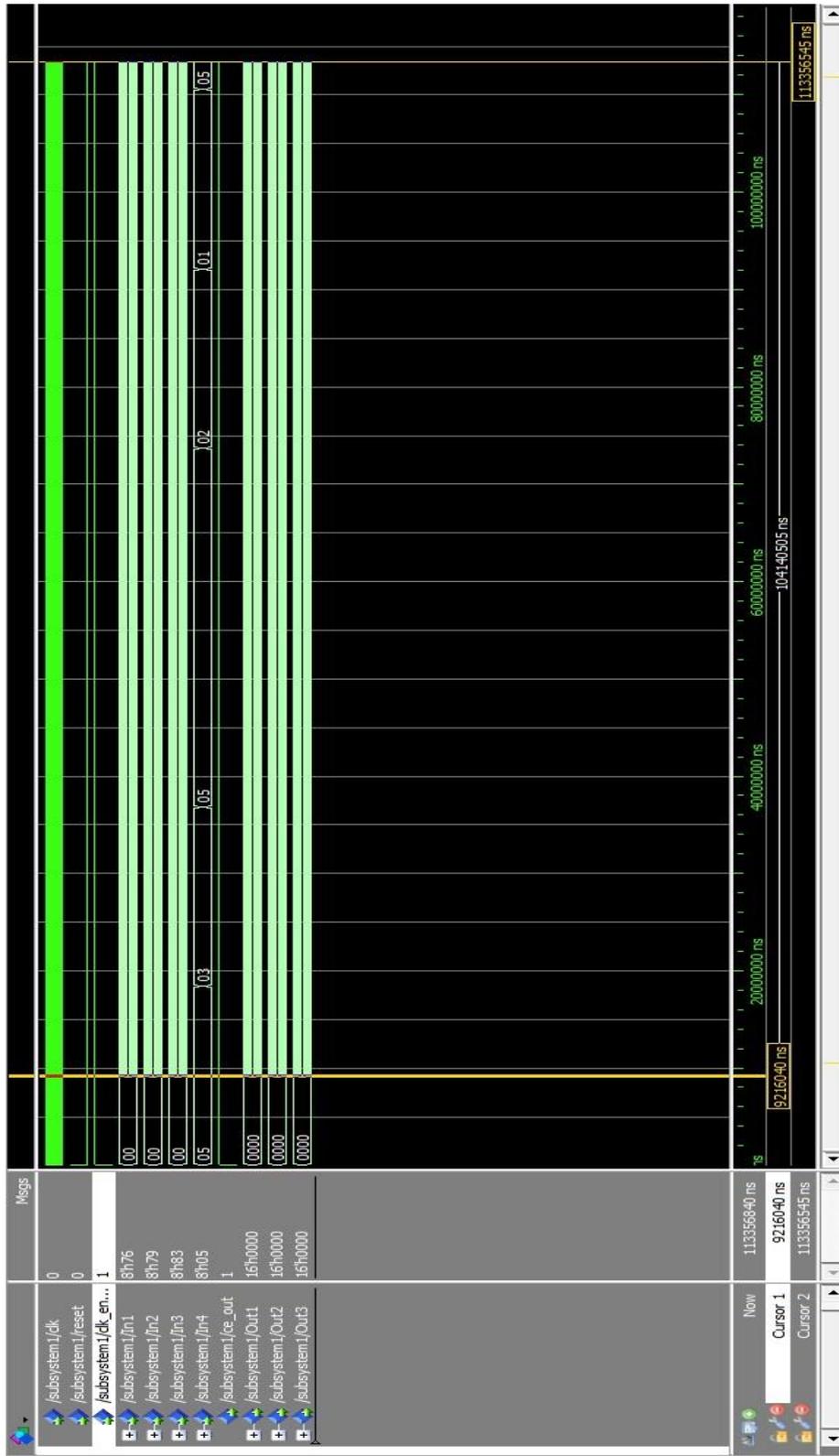


Figure 4.5. ModelSim Timing Signals for Encoding Process (30 Frame)



a- Original Frame



b- Encoded Frame

Figure 4.6. Frames Before and After Encoding Process

4.4.2 Recoder Model

The recoder model is designed in Matlab Simulink as seen in figure 4.7. The model includes signals input blocks for entering signals coming from the network, rate transition blocks to match the frequency between signals input

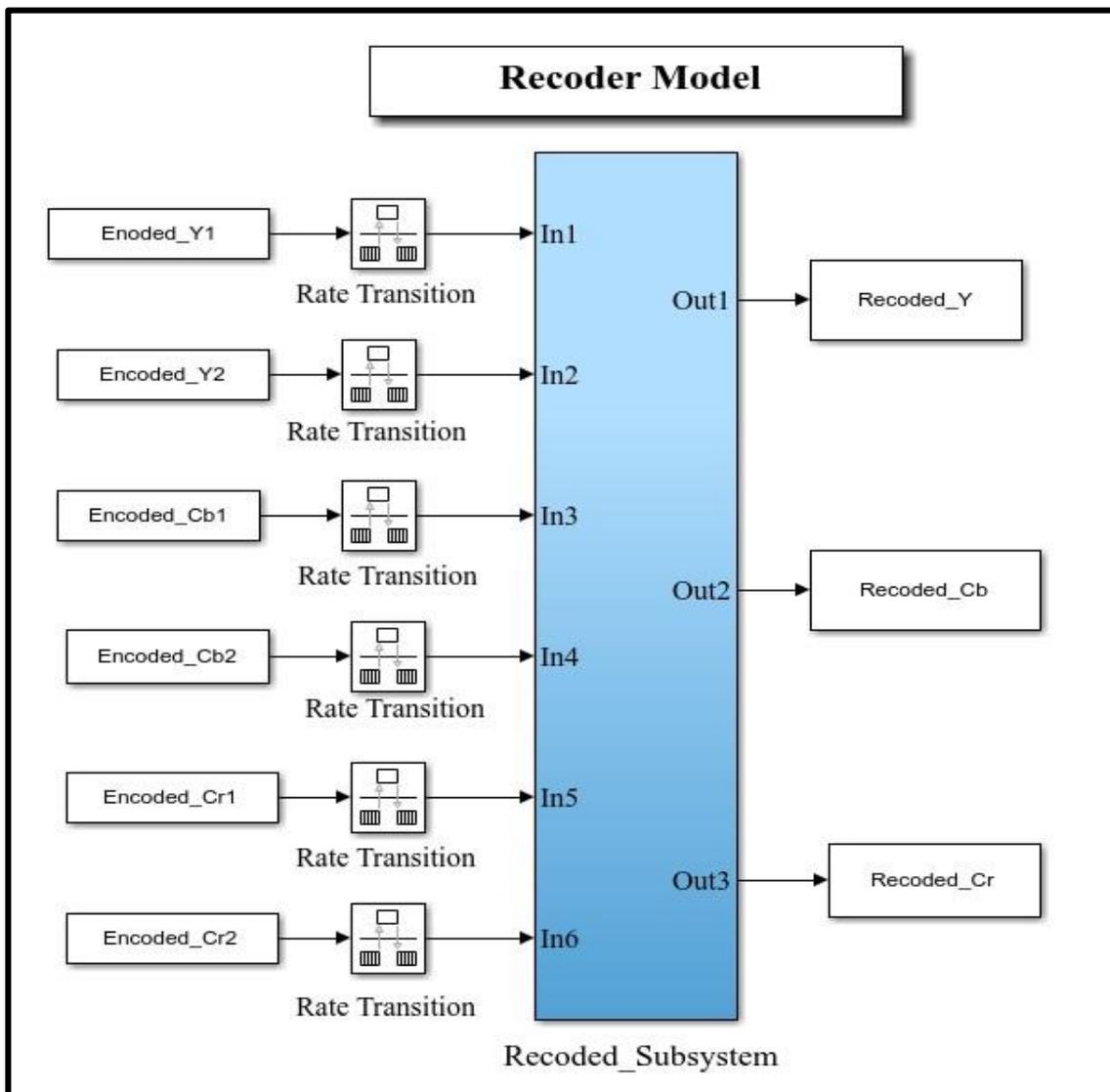


Figure 4.7. Recoder Model

blocks, and recoder blocks. The most important block in this model is the 'Recoder Subsystem block' performs recoding operation on streams of frames. 'Recoded_Y block', 'Recoded_Cb block' and 'Recoded_Cr block' to output the recoded streams of frames to the network

4.4.2.1 Implementation of Recoder

'Recoder_Subsystem block' performs the recoding process so, it will be implemented on FPGA by the following steps: -

- 'Recoder_Subsystem block' is the block for which VHDL is generated.
- by the FPGA-in-the-Loop Wizard in MATLAB, the FPGA programming file for 'Recoder_Subsystem block' is generated.
- The 'Recoder-subsystem FIL' is created.
- Finally, the 'Recoder_Subsystem block' programming file is loaded to FPGA.

Then, the synthesis has been implemented on the 'Recoder-Subsystem block'. The summary of the utilization report is given in Table 4.3.

Table 4.3. Device Utilization Summary for Recoder

Slice Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	24	63400	1%
Number Used as Logic	24	63400	1%
Number of Occupied Slices	21	15850	1%
Number of LUT Flip Flop Pairs Used	24		
Number with An Unused Flip Flop	24	24	100%
Number of Bonded IOBS	144	210	68%
Average Fanout of Non-Clock Nets	1.00		

4.4.2.2 Internal Details and Processing for Recoder

'Recoder_Subsystem block' contains three sub-blocks, as shown in figure 4.8. These sub-blocks work parallel simultaneously, where each sub-block merges the two incoming streams from the network into one stream and sends them again.

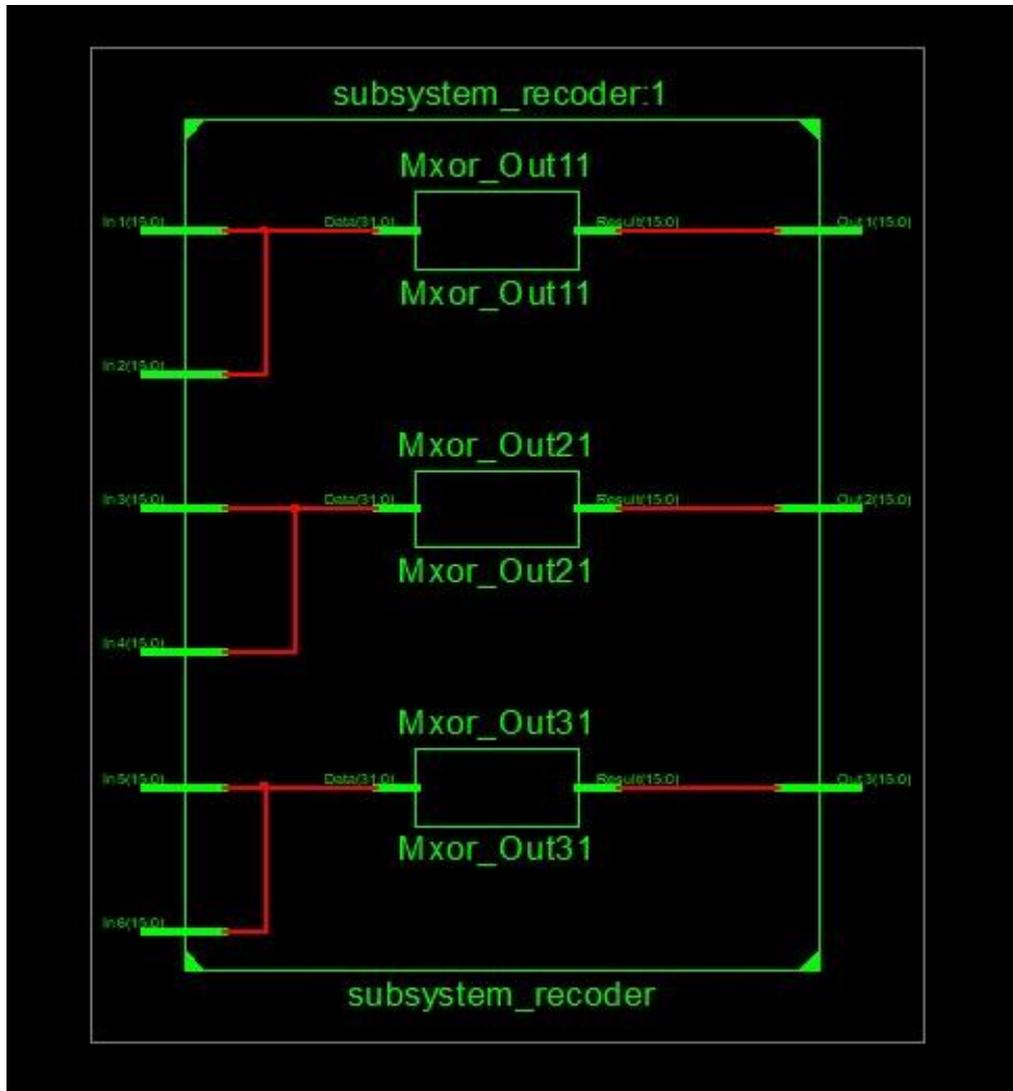


Figure 4.8. RTL Internal Details for Recoder

4.4.2.3 The Frame Resulting from The Recoding Process

The recoding process takes place in the intermediate node, it recodes the encoded frames to produce the recoded frame. Figure 4.9 shows the video frame before and after the recoding process.

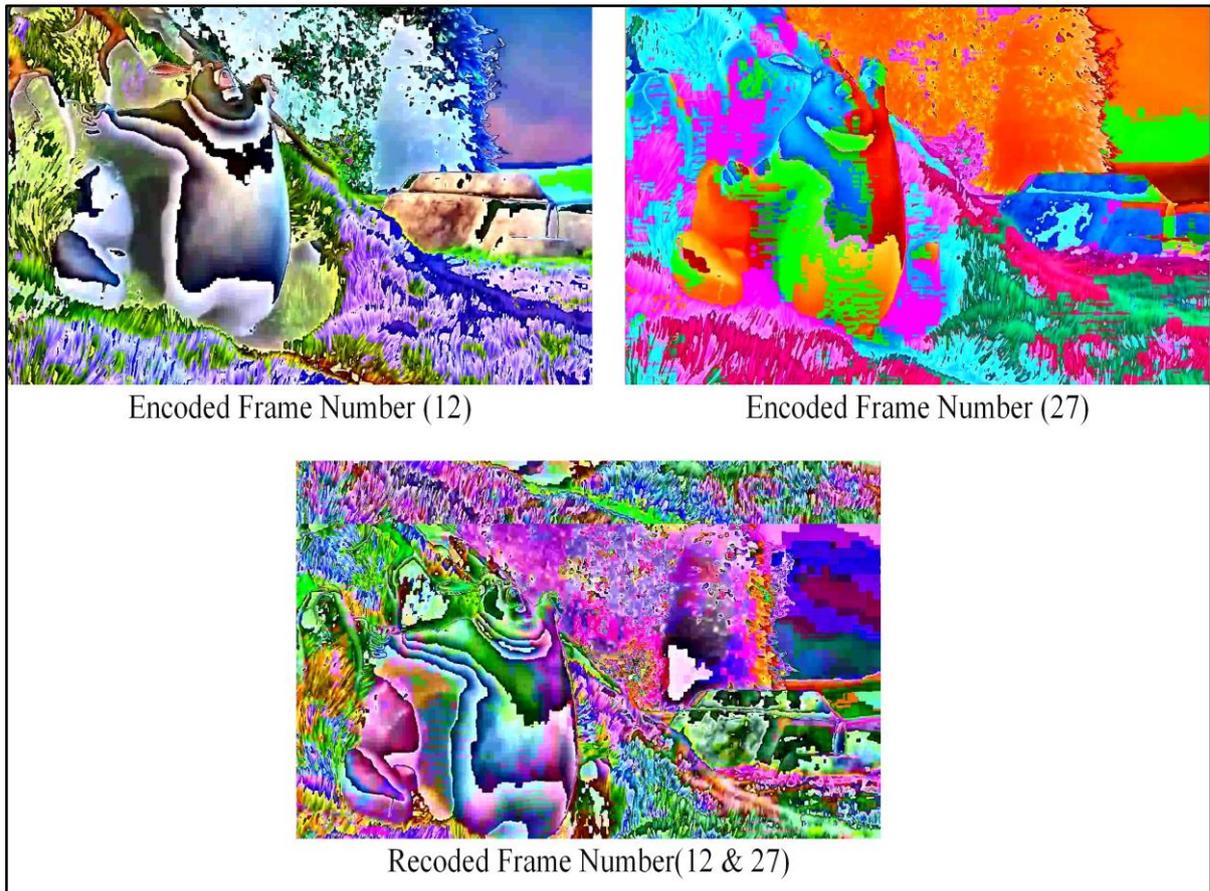


Figure 4.9. Frames Before and After Recoding Process

4.4.3 Decoder Model

In Matlab Simulink, the decoder model is designed as shown in figure 4.10. Components included in the model are signal input blocks to entering signals for encoded and recoded streams of frames coming from the network. ‘Decoder-Subsystem block’ performs the decoding process on recoded and encoded frames to recover the original frames. ‘Stream to Frame’ to convert from stream to frame, ‘Color Space Conversion’ The conversions from Y'CbCr to R'G'B' color spaces, ‘Matrix Concatenate block’ to concatenate frames, ‘Video Viewer block’ to display video.

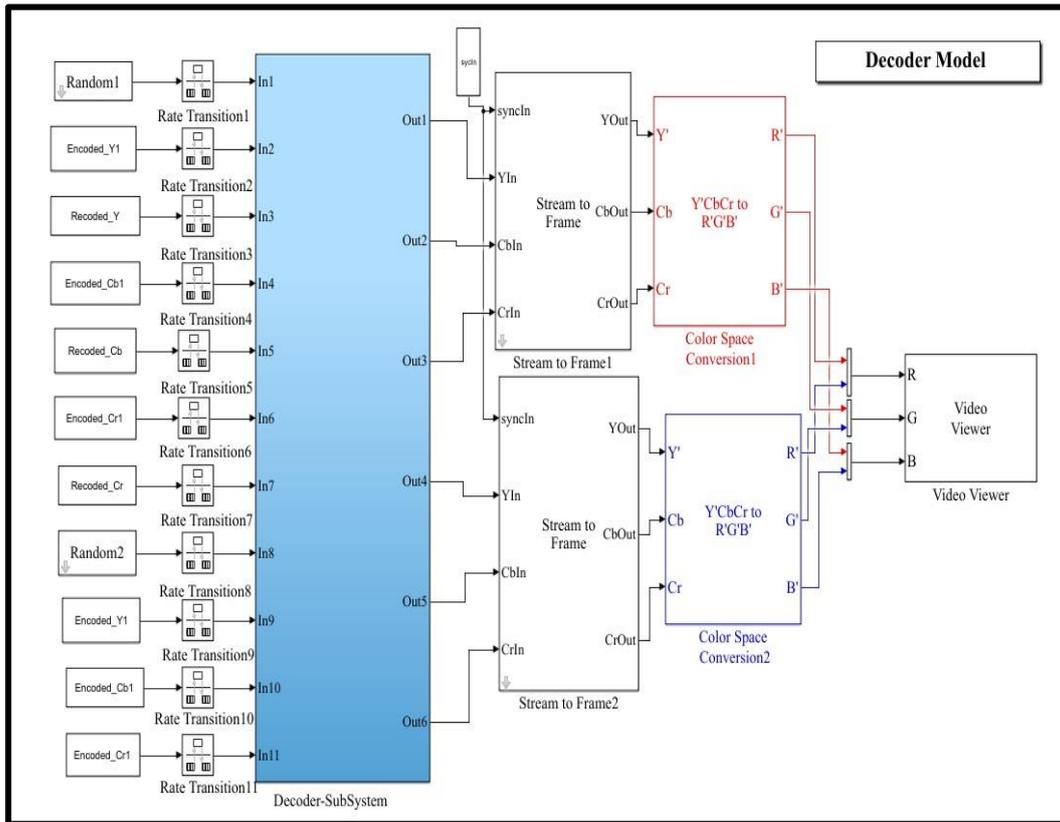


Figure 4.10. Decoder Model

4.4.3.1 Implementation of Decoder

‘Decoder_Subsystem block’ performs the decoding process so, it will be implemented on FPGA by the following steps: -

- The VHDL code for the ‘Decoder-Subsystem block’ is generated
- By using the FPGA-in-the-Loop Wizard in Matlab, the FPGA programming file for ‘Decoder-Subsystem block’ is generated
- The ‘Decoder-Subsystem FIL block’ is created.
- The programming file of Decoder-Subsystem is loaded into FPGA.

Then, the synthesis has been implemented on the ‘Decoder-Subsystem block’. The summary of the utilization report is given in table 4.4.

Table 4.4. Device Utilization Summary for Decoder

Slice Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	2046	63,400	3%
Number Used as Logic	2046	63,400	3%
Number Using O6 Output Only	1,498		
Number Using O5 Output Only	6		
Number Using O5 and O6	542		
Number of Occupied Slices	772	15850	4%
Number of LUT Flip Flop Pairs Used	2046		
Number with An Unused Flip Flop	2046	2046	100%
Number of Bonded IOBS	208	210	99%
Average Fanout of Non-Clock Nets	4.32		

4.4.3.2 Internal Details and Processing for Decoder

Video decoding takes some time. The ‘Decoder-Subsystem block’ is designed to decode streams simultaneously, which reduces decoding time. The figure 4.11 shows The ‘Decoder-Subsystem block’, where includes two units, one to decode the recoded frames and the other to decode encoded frames, each of which has three blocks operating in parallel simultaneously to decode the flows that come to it.

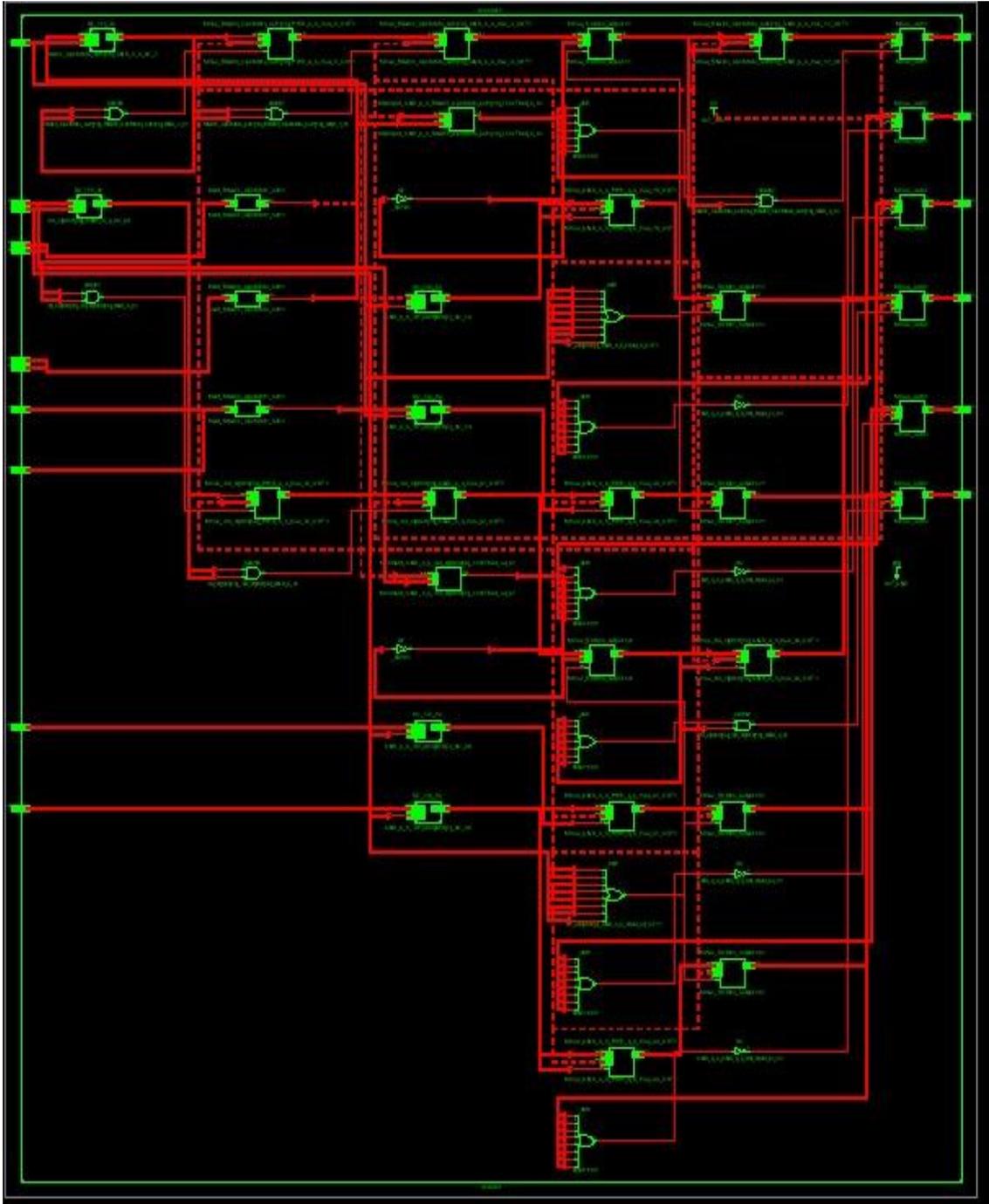


Figure 4.11. Internal Details for Decoder

4.4.3.3 Time for Decoder

The decoding time for two different videos (HD and SD) was calculated by ModelSim several times and each time on a different number of frames as shown in table 4.5.

Table 4.5. Decoding Time for SD and HD Videos

Number of frames	Decoding time (ms) SD video (240*176)	Decoding time (ms) HD video (1280*720)
7	0.634	13.824
13	1.6896	36.864
19	2.746	59.904
25	3.802	82.944
30	4.773	104.14084

The figure 4.12, noticed that the decoding process on videos occurs in real time, where HD video with frame rate 30 fps are processed in less than one second.

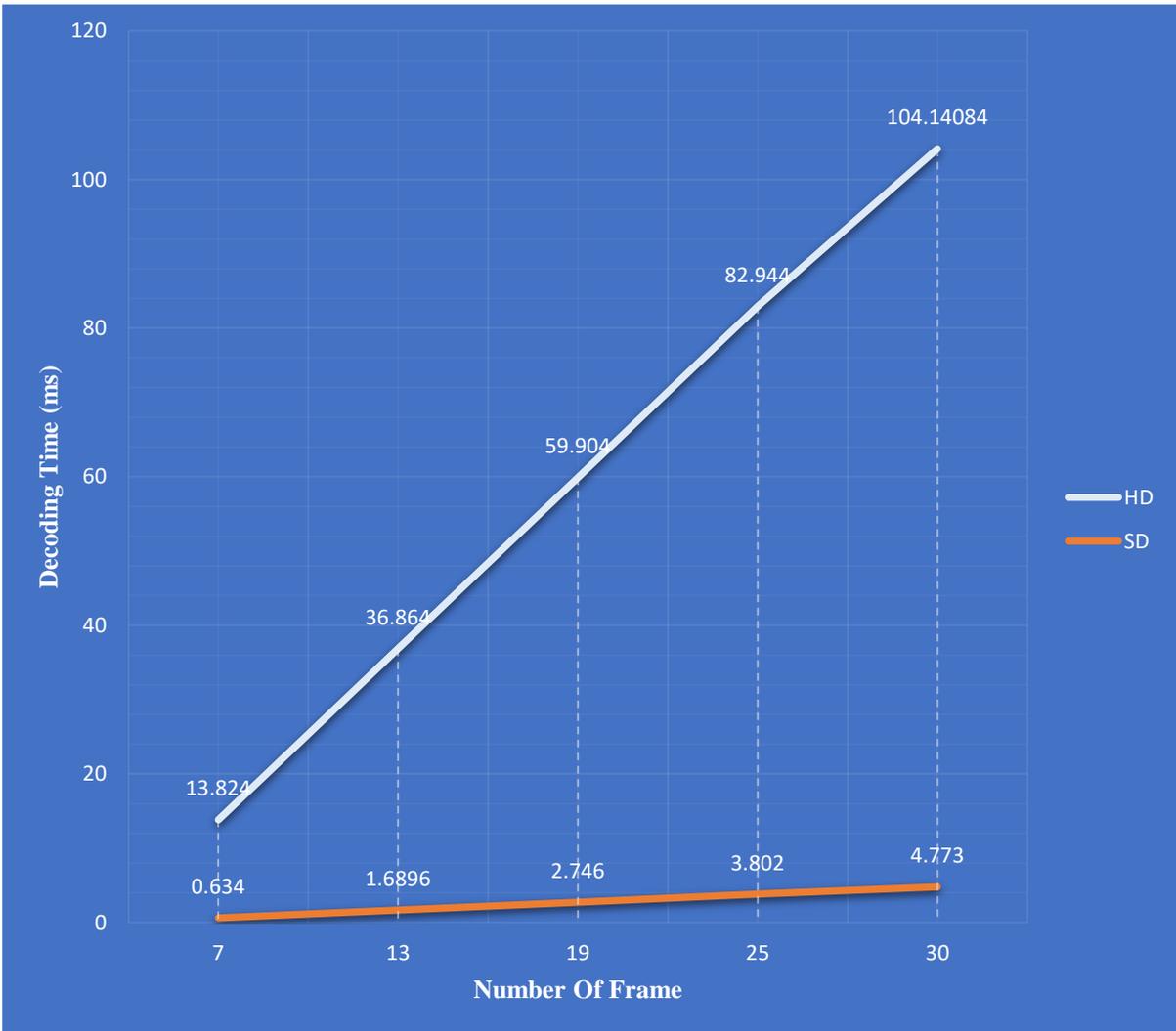


Figure 4.12. Decoding Time with Different Number of Frame

Figure 4.13 shows the timing signals for decoding process for 30 frame HD video measured by Modelsim.

4.4.3.4 The Frame Resulting from The Decoding Process

In the destination node, the decoding process takes place on the recoded and decoded frames to retrieve original frames. Figure 4.14 shows the shape of the video frame before and after the decoding process.

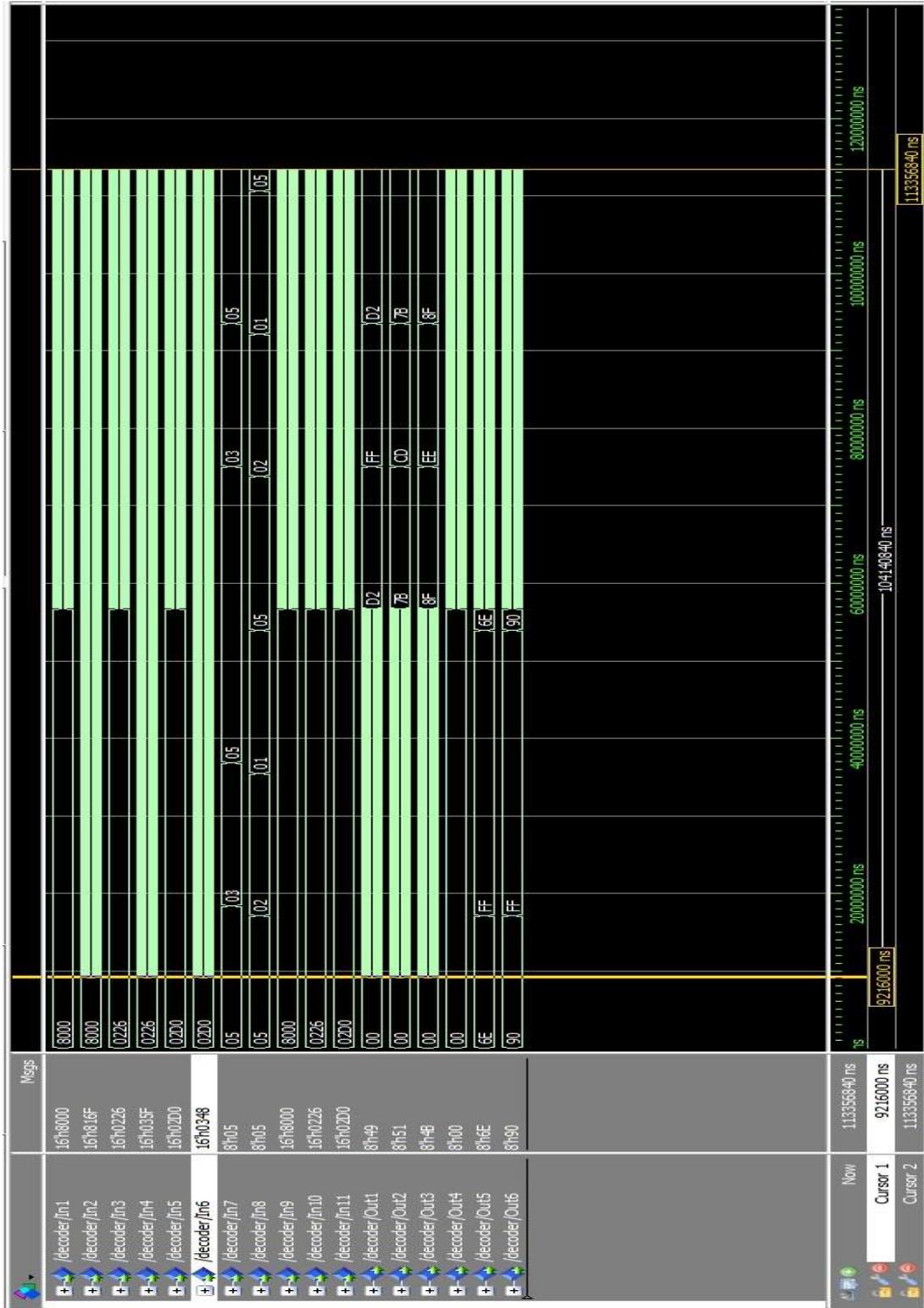


Figure 4.13. ModelSim Timing Signals for Decoding Process (30 Frame)

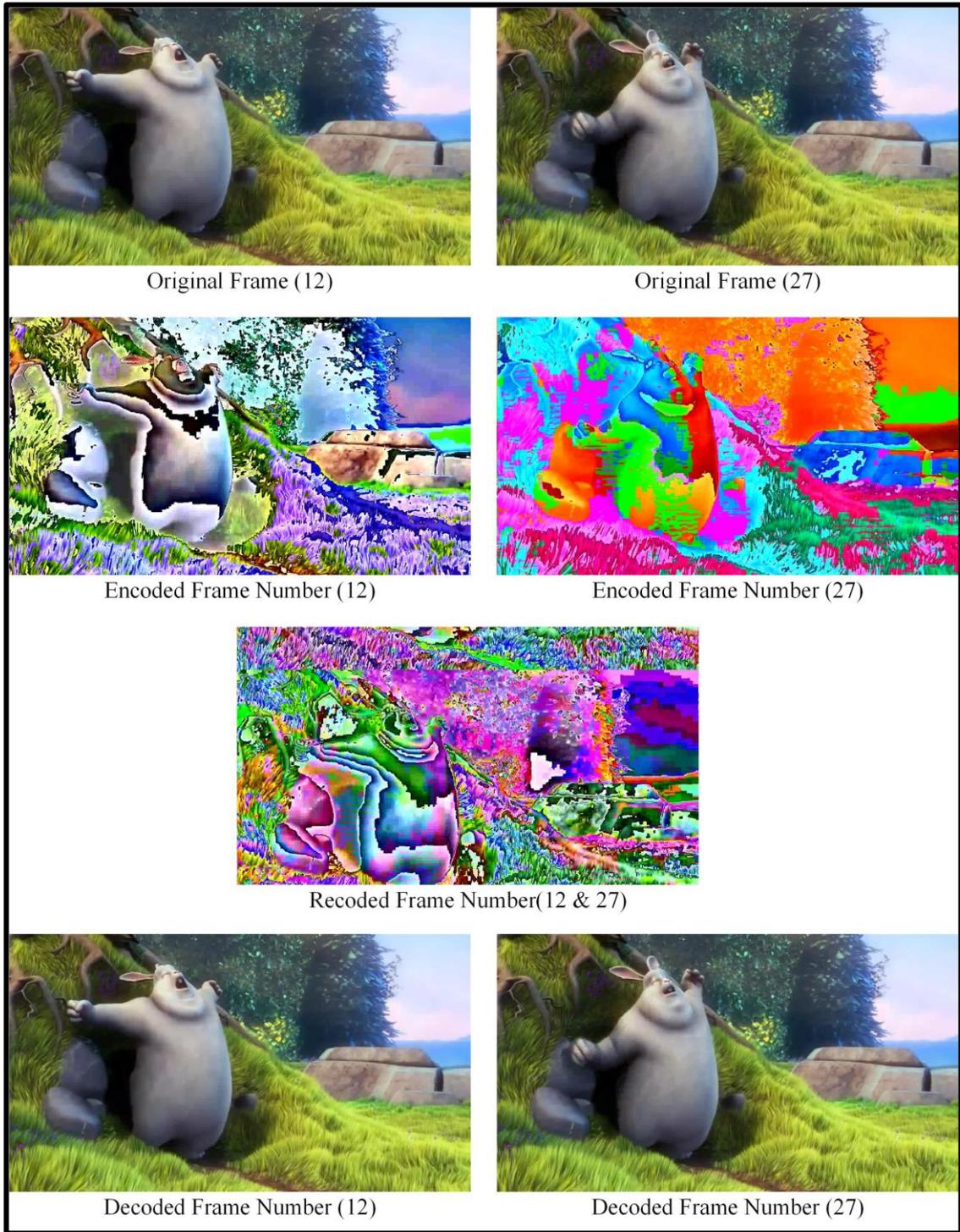


Figure 4.14. Frames Before and after Decoding Process

Chapter Five

Conclusions and Suggestion for Future Work

5.1 Conclusions

- Zero BER is obtained between the video frames at the transmitter end and the video frames at the receiving end when the video is delivered in different formats.
- Throughput has been improved by 25% over the methods that do not use RLNC and increasing bandwidth capacity utilization by sending more than one packet in the transmission process, thus reducing the number of transmission times.
- The reliability of the video transmission is improved by increasing the efficiency and reliability of the encoding when a large GF value is selected.
- The execution time of each process increases as the video's resolution increases, while it is affected slightly by increasing the degree of GF within the chosen gf values.
- HD video 30fps in real time is encoded at the source node and is decoded in the destination node.
- The time obtained in the Encoding and decoding phase was within real-time limits due to the use of parallel implementation on the FPGA board.
- The linear relationship of time in the encoding and decoding stage gives an indication of the real-time transferability of HD video files with a frame rate up to 240 fps.

5.2 Future Work

1. One of the challenges which we encountered during the FPGA implementation was dealing with video files. So, we recommend implemented

video delivery by applying RLNC using system on chip based FPGA, On Zynq-7000 evaluation kit.

2. Secure video delivery with random Linearly Network Coding Scheme.
3. Construction of a RLNC algorithm based on the neural network.

References

- [1] “Cisco Annual Internet Report (2018–2023) White Paper.” [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. [Accessed: 29-Jan-2021].
- [2] D. Vukobratovic, A. Tassi, S. Delic, and C. Khirallah, “Random linear network coding for 5G mobile video delivery,” *Information (Switzerland)*, vol. 9, no. 4, pp. 1–20, Mar. 2018.
- [3] T. Schierl, K. Ganger, C. Hellge, T. Wiegand, and T. Stockhammer, “SVC-based multisource streaming for robust video transmission in mobile Ad Hoc Networks,” *IEEE Wireless Communications*, vol. 13, no. 5, pp. 96–103, 2006.
- [4] H. Wang, J. Liang, and C. C. Jay Kuo, “Overview of robust video streaming with network coding,” *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 1, pp. 36–50, Jan. 2010.
- [5] A. Zakhor and W. Wei, “Multiple tree video multicast over wireless ad hoc networks,” *Proceedings - International Conference on Image Processing, ICIP*, pp. 1665–1668, 2006.
- [6] R. Ahlswede, N. Cai, and R. W. Yeung, “Network information flow theory,” *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [7] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the Air: Practical Wireless Network Coding,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, Jun. 2008.
- [8] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, “Wireless broadcast using network coding,” *IEEE Transactions on Vehicular Technology*, vol. 58, no.

- 2, pp. 914–925, Feb. 2009.
- [9] J. Heide, M. V. Pedersen, and F. H. P. Fitzek, “Decoding Algorithms for Random Linear Network Codes,” in *NETWORKING 2011 Workshops. Lecture Notes in Computer Science*, vol. 6827 LNCS, 2011, pp. 129–136.
 - [10] M. Esmaeilzadeh, P. Sadeghi, and N. Aboutorab, “Random Linear Network Coding for Wireless Layered Video Broadcast: General Design Methods for Adaptive Feedback-Free Transmission,” *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 790–805, Feb. 2017.
 - [11] L. Nielsen, R. R. Hansen, and D. E. Lucani, “Latency performance of encoding with random linear network coding,” in *European Wireless 2018; 24th European Wireless Conference*, 2018, pp. 1–5.
 - [12] D. Goncalves, S. Signorello, F. M. V. Ramos, and M. Medard, “Random linear network coding on programmable switches,” in *2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS 2019*, 2019, pp. 1–6.
 - [13] E. Tsimbalo and M. Sandell, “Reliability of Relay Networks Under Random Linear Network Coding,” *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5230–5240, Aug. 2019.
 - [14] I. Leyva-Mayorga *et al.*, “Network-Coded Cooperation and Multi-Connectivity for Massive Content Delivery,” *IEEE Access*, vol. 8, pp. 15656–15672, 2020.
 - [15] K. Fouli, M. Maier, and M. Medard, “Network coding in next-generation passive optical networks,” *IEEE Communications Magazine*, vol. 49, no. 9, pp. 38–46, Sep. 2011.
 - [16] X. Liu, K. Fouli, R. Kang, and M. Maier, “Network-coding-based energy

- management for next-generation passive optical networks,” *Journal of Lightwave Technology*, vol. 30, no. 6, pp. 864–875, 2012.
- [17] N. Panwar, S. Sharma, and A. K. Singh, “A survey on 5G: The next generation of mobile communication,” *Physical Communication*, vol. 18, pp. 64–84, 2016.
- [18] Amir Zarei, P. Pahlevani, and Mansoor Davoodi Monfared, “Sparse Network Coding for 5G Network and Real-time Wireless Communication,” M. S. thesis, Institute for Advanced Studies in Basic Sciences, Zanzan, Iran, 2018.
- [19] J. K. Edward J. Oughton, William Lehr, Konstantinos Katsaros, Ioannis Selinis, Dean Bublely, “Revisiting Wireless Internet Connectivity: 5G vs Wi-Fi 6,” *Telecommunications Policy*, vol. 45, no. 5, 2021.
- [20] J. G. Andrews *et al.*, “What will 5G be?,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [21] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, “5G Evolution: A View on 5G Cellular Technology beyond 3GPP Release 15,” *IEEE Access*, vol. 7, pp. 127639–127651, 2019.
- [22] M. Series, “IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond,” *Recommendation ITU*, vol. 2083, 2015.
- [23] 5G PPP, “5G Empowering Vertical Industries,” *5G-PPP White Papers*, 2016.
- [24] Prashant Kumar Shah, J. Haapola, A. Laghrissi, and V. Halonen, “ENABLING SEAMLESS APPLICATION MIGRATION OVER MULTI-CORE NETWORK ENVIRONMENTS,” University of Oulu, 2020.

- [25] S. Vannithamby, Rath and Talwar, *Towards 5G: Applications, Requirements and Candidate Technologies*. John Wiley & Sons, 2017.
- [26] C. V. N. Index, “Global mobile data traffic forecast update 2017–2022,” *Cisco White paper*, 2019.
- [27] A. O. Adeyemi-Ejeye, M. Alreshoodi, L. Al-Jobouri, M. Fleury, and J. Woods, “Packet loss visibility across SD, HD, 3D, and UHD video streams,” *Journal of Visual Communication and Image Representation*, vol. 45, pp. 95–106, May 2017.
- [28] “Separating SD, HD, Full HD, 4K and 8K,” 05-Mar-2018. [Online]. Available: <https://www.image-engineering.de/library/technotes/991-separating-sd-hd-full-hd-4k-and-8k>. [Accessed: 29-Jan-2021].
- [29] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, “Achieving minimum-cost multicast: A decentralized approach based on network coding,” in *In Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005, vol. 3, pp. 1608–1617.
- [30] D. S. Lun *et al.*, “Minimum-cost multicast over coded packet networks,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2608–2623, 2006.
- [31] S. Jaggi *et al.*, “Polynomial time algorithms for multicast network code construction,” *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
- [32] C. Fragouli and E. Soljanin, “Network coding fundamentals,” *Foundations and Trends in Networking*, vol. 2, no. 1, pp. 1–133, 2007.
- [33] Y. Benfattoum, S. Martin, and K. Al Agha, “Network Coding: From Theory to Practice,” in *Network Coding*, K. Al Agha, Ed. Hoboken, NJ USA: John

- Wiley & Sons, Inc., 2013, pp. 1–26.
- [34] R. Ghrist and S. Krishnan, “A topological max-flow-min-cut theorem,” in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 815–818.
- [35] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the Air: Practical Wireless Network Coding,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, Jun. 2008.
- [36] T. Ho and D. Lun, *Network coding: An introduction*, vol. 9780521873. Cambridge: Cambridge University Press, 2008.
- [37] P. Karimian, “Random Linear Network Coding for Non-Multicast and Multi-Resolution Multicast Problems,” M.S., University of Alberta, Edmonton, Canada, 2017.
- [38] R. Stenvoll, “Network Coding in Bluetooth Networks,” Masters Thesis, University of Bergen, Bergen, Norway, 2009.
- [39] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, “Network coding,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, Jan. 2006.
- [40] T. Van Vu, “Application of network coding in wireless networks: coding conditions and adaptive redundancy control,” Ph.D. dissertation, University Pierre et Marie Curie-Paris VI, Paris, France, 2014.
- [41] R. W. Y. and N. C. S. -. R. Li, “Linear Network Coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [42] M. Tan, “A unified framework for linear network coding,” M.S. Thesis, University of Hong Kong, Hong Kong, China.
- [43] M. Médard and A. Sprintson, *NETWORK CODING Fundamentals and Applications*, First. Waltham, USA: Academic Press is an imprint of

- Elsevier, 2012.
- [44] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, “I2NC: Intra- and inter-session network coding for unicast flows in wireless networks,” in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 1035–1043.
 - [45] T. Ho *et al.*, “A random linear network coding approach to multicast,” *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.
 - [46] P. A. Chou and Y. Wu, “Network Coding for the Internet and Wireless Networks,” *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77–85, Sep. 2007.
 - [47] S. Deb *et al.*, “Network Coding for Wireless Applications: A Brief Tutorial,” in *Proc. Int. Workshop on Wireless Ad Hoc Networks*, 2005.
 - [48] D. Szabó, A. Gulyás, F. H. P. Fitzek, and D. E. Lucani, “Towards the tactile Internet: Decreasing communication latency with network coding and Software Defined Networking,” in *Proceedings of 21st European Wireless Conference, European Wireless 2015*, 2015.
 - [49] A. S. Khan, “Characterisation and Performance Analysis of Random Linear Network Coding for Reliable and Secure Communication Declaration of Authorship,” Ph.D thesis, Lancaster University, Bailrigg, Lancaster, United Kingdom, 2018.
 - [50] K. Fouli, M. Médard, and K. Shroff, “Random Linear Network Coding A Technical Feature Overview,” *Code On*, 2018.
 - [51] M. Luby, “LT codes,” in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, 2002, pp. 271–280.
 - [52] D. E. Lucani, M. Stojanovic, and M. Médard, “Random linear network coding for time division duplexing: When to stop talking and start

- listening,” *IEEE INFOCOM 2009*, pp. 1800–1808, 2009.
- [53] D. E. Lucani, M. Medard, and M. Stojanovic, “On Coding for Delay—Network Coding for Time-Division Duplexing,” *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2330–2348, Apr. 2012.
- [54] M. V. Pedersen, J. Heide, F. H. P. Fitzek, and T. Larsen, “A mobile application prototype using network coding,” *European Transactions on Telecommunications*, vol. 21, no. 8, pp. 738–749, 2010.
- [55] V. Johansson, “Enhancing user satisfaction in 5G networks using Network Coding,” M. S. thesis, KTH Royal Institute of Technology School of Information and Communication Technology (ICT), Sweden, Stockholm, 2015.
- [56] Clive “Max” Maxfield, *The Design Warrior’s Guide to FPGAs*. Burlington: Newnes is an imprint of Elsevier, 2004.
- [57] C. H. Roth, *Digital Systems Design Using VHDL*. USA: Wadsworth Publ. Co., 1998.
- [58] “Nexys A7™ FPGA Board Reference Manual,” 10-Jul-2019. [Online]. Available:
https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-a7/nexys-a7_rm.pdf. [Accessed: 20-Mar-2021].
- [59] N. Othman, M. H. Jabbar, A. K. Mahamad, and F. Mahmud, “FPGA-in-the-Loop Simulation of Cardiac Excitation Modeling towards Real-Time Simulation,” in *5th International Conference on Biomedical Engineering in Vietnam*, vol. 46, Springer, Cham, 2015, pp. 203–206.
- [60] M. Alareqi, R. Elgouri, K. Mateur, A. Zemmouri, A. Mezouari, and L. Hlou, “Optimization of high-level design edge detect filter for video processing system on FPGA,” in *2017 Intelligent Systems and Computer*

Vision, ISCV, 2017, pp. 1–8.

- [61] J. C. T. Hai, O. C. Pun, and T. W. Haw, “Accelerating video and image processing design for FPGA using HDL coder and simulink,” in *2015 IEEE Conference on Sustainable Utilization And Development In Engineering and Technology (CSUDET)*, 2015, pp. 1–

الخلاصة

النمو السريع لنقل الفيديو والطلب المتزايد على خدمات الفيديو عالية الدقة على نطاق واسع في السنوات الأخيرة ، وقد أدى ذلك إلى الحاجة إلى تقنيات قوية لتغطية هذه المتطلبات حيث تعد تقنية ترميز الشبكة الخطية العشوائية واحدة من أكثر التقنيات الواعدة لتسليم الفيديو ، وتحسين الإنتاجية ، والاستفادة من سعة النطاق الترددي ، وتوفير الموثوقية العالية ، وزمن الانتقال المنخفض. في هذه الرسالة ، تم تطبيق هذه الخوارزمية في نقل الفيديو وتنفيذها على FPGA .

النموذج المقترح يتكون من ثلاث خوارزميات : خوارزمية الترميز تعمل في عقدة المصدر ، إعادة الترميز تعمل في العقدة الوسيطة ، وفك الترميز تعمل في عقدة المستقبل .

تم استخدام Matlab في تصميم شبكة بسيطة لتطبيق هذه الخوارزميات الثلاث ، حيث تتكون هذه الشبكة : من عقدة تمثل عقدة المصدر ، وعقدة وسيطة ، وعقدة للمستلمين . وتمت دراسة تأثير معاملين على نقل الفيديو ؛ هما درجة مجال Galois ودقة الفيديو . وكشفت نتائج المحاكاة عن فعالية إختيار أعداد كبيرة من Galois field في زيادة موثوقية ترميز الفيديو وبالتالي زيادة موثوقية نقل الفيديو . يتضح تأثير دقة الفيديو بشكل كبير على زمن كل مرحلة . حيث يزداد زمن المحاكاة عند ازدياد دقة الفيديو . تم نقل مقاطع فيديو ذات دقات عرض مختلفة ، وكان معدل الخطأ في البت بين إطارات الفيديو في طرف الإرسال وإطارات الفيديو في طرف الاستقبال يساوي صفرًا . إضافة الى زيادة الإنتاجية بنسبة 25% عن الأساليب التقليدية التي لا تستخدم ترميز الشبكة الخطية العشوائية عن طريق زيادة استخدام سعة النطاق الترددي حيث يتم إرسال أكثر من حزمة واحدة في عملية الإرسال ، وبالتالي تقليل عدد مرات الإرسال.

تم استخدام برمجيات شركة Xilinx (Vivado 2018.2) وربطها مع الماتلاب (Matlab) في بناء الكيان المادي للخوارزميات. حيث تم استخدام (FPGA in the loop) و (Modelsim) لتصميم ثلاث خوارزميات (المرمز ، معيد الترميز ، مفكك الترميز) ثم تنفيذهم على FPGA . حيث تم إرسال فيديو عالي الدقة (HD) بمعدل 30 اطار في الثانية ، وقد استغرق وقت الترميز (104.14050 مللي ثانية) ، بينما وقت فك الترميز استغرق (104.14084 مللي ثانية).

اقرار لجنة المناقشة

نشهد بأننا أعضاء لجنة التقويم والمناقشة قد اطلعنا على هذه الرسالة الموسومة (تنفيذ الترميز الشبكي الخطي العشوائي لتسليم الفيديو في الجيل الخامس باستخدام مصفوفة البوابات المبرمجة حقليا) وناقشنا الطالبة (امنة موفق يونس) في محتوياتها وفيما له علاقة بها بتاريخ // / 2018 وقد وجدناها جدير بنيل شهادة الماجستير-علوم في اختصاص هندسة الحاسوب والمعلوماتية.

التوقيع:

التوقيع:

رئيس اللجنة:

عضو اللجنة(المشرف):

التاريخ: // / 2021

التاريخ: // / 2021

التوقيع:

التوقيع:

عضو اللجنة:

عضو اللجنة:

التاريخ: // / 2021

التاريخ: // / 2021

قرار مجلس الكلية

اجتمع مجلس كلية هندسة الالكترونيات بجلسته المنعقدة بتاريخ : // / 2021 وقرر المجلس منح الطالب شهادة الماجستير علوم في اختصاص هندسة الحاسوب والمعلوماتية.

مقرر المجلس: د.

رئيس مجلس الكلية:

التاريخ: // / 2021

التاريخ: // / 2021

إقرار المشرف

اشهد بان الرسالة الموسومة ب " تنفيذ الترميز الشبكي الخطي العشوائي لتسليم الفيديو في الجيل الخامس باستخدام مصفوفة البوابات المبرمجة حقليا " تمت تحت اشرافي وهي جزء من متطلبات نيل شهادة الماجستير في هندسة الحاسوب والمعلوماتية

التوقيع:

المشرف : أ . م . محمد حازم الجماس

التاريخ: / / 2021

إقرار المقيم اللغوي

اشهد بانني قمت بمراجعة الرسالة الموسومة ب " تنفيذ الترميز الشبكي الخطي العشوائي لتسليم الفيديو في الجيل الخامس باستخدام مصفوفة البوابات المبرمجة حقليا " من الناحية اللغوية وتصحيح ما ورد فيها من أخطاء لغوية وتعبيرية وبذلك أصبحت الرسالة مؤهلة للمناقشة بقدر تعلق الامر بسلامة الأسلوب وصحة التعبير.

التوقيع:

المقوم اللغوي:

التاريخ: / / 2021

إقرار رئيس لجنة الدراسات العليا

بناء على التوصيات المقدمة من قبل المشرف والمقوم اللغوي أشرح هذه الرسالة للمناقشة.

التوقيع:

الاسم: أ . م . معن أحمد شحادة العدوانى

التاريخ: / / 2021

إقرار رئيس القسم

بناء على التوصيات المقدمة من قبل المشرف والمقوم اللغوي ورئيس لجنة الدراسات العليا أشرح هذه الرسالة للمناقشة.

التوقيع:

الاسم: أ . م . معن أحمد شحادة العدوانى

التاريخ: / / 2021

**تنفيذ الترميز الشبكي الخطي العشوائي لتسليم الفيديو في الجيل الخامس
باستخدام مصفوفة البوابات المبرمجة حقليا**

رسالة تقدم بها

امنة موفق يونس الطائي

إلى

مجلس كلية هندسة الالكترونيات

جامعة نينوى

كجزء من متطلبات نيل شهادة الماجستير

في

هندسة الحاسوب والمعلوماتية

ياشرف

الأستاذ المساعد الدكتور

محمد حازم الجماس



جامعة نينوى

كلية هندسة الالكترونيات

قسم هندسة الحاسوب والمعلوماتية

تنفيذ الترميز الشبكي الخطي العشوائي لتسليم الفيديو في الجيل الخامس باستخدام مصفوفة البوابات المبرمجة حقليا

امنة موفق يونس الطائي

رسالة ماجستير علوم في هندسة الحاسوب والمعلوماتية

بإشراف

الأستاذ المساعد الدكتور
محمد حازم الجماس

م 2021

هـ 1442