QATAR UNIVERSITY

COLLEGE OF ENGINEERING

PERFORMANCE EVALUATION OF CROSS-LAYER DESIGN WITH DISTRIBUTED AND SEQUENTIAL MAPPING SCHEME FOR VIDEO APPLICATION OVER IEEE 802.11E

BY

HEBA D. DAWOUD

A Thesis Submitted to the Faculty of

College of Engineering

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

January, 2016

COMMITTEE PAGE

The members of the Committee approve the dissertation of Heba D. Dawoud defended on 10th February 2016.

_____

Dr. Mohamad Saleh

Thesis Supervisor

_____

Dr. Amr Mohamad

Committee Member

_____

Dr. Abdulla Khalid Al-Ali

Committee Member

_____

Dr. Khaled Shaban

Committee Chair

Approved:

_____

Dr. Rashid Alammari, Dean, College of Engineering

ABSTRACT

The rapid development of wireless communication imposes several challenges to support QoS for real-time multimedia applications such as video stream applications. Researches tackled these challenges from different points of view including the semantics of the video to achieve better QoS requirements. The main goal of this research is to design a UDP protocol to realize a distributed sequential mapping scheme (DSM) with a cross-layer design and evaluate its accuracy under different network conditions. In DSM, the perceived quality of a multi-layer video is addressed by mapping each video layer into channel resources represented as queues or access categories (ACs) existing in IEEE 802.11e MAC layer. This research work further investigates the efficiency of this scheme with actual implementation and thorough simulation experiments. The experiments reported the efficiency of this scheme with the presence of different composite traffic models covering most known traffic scenarios using Expected Reconstructed Video Layers (ERVL) and packet loss rate as accuracy measures. This research work also investigates the accuracy of calculating the ERVL compared to its value using actual readings of layers drop rate. The effect of changing the ACs queue size on the ERVL is studied. The use of this scheme shows zero-drop in the base layer in almost all scenarios where no ongoing traffic is presented except that the testing video sessions between nodes. In these experiments, the ERVL continuously reported high values for the number of expected reconstructed video layers. While these values dramatically vary when introducing ongoing different composite traffic models together with the testing video sessions between nodes. Finally, a 40% increase in the ACs queue size shows significant improvement on ERVL while an increase of the queue size beyond this value has very little significance on ERVL.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AC	Access Category
ACK	Acknowledgement
AIFS	Arbitrary IFS
AM	Adaptive Mapping
AODV	Ad-hoc On-demand Distance Vector
AP	Access Point
ARP	Address Resolution Protocol
CBR	Constant Bit Rate
CFP	Contention Free Period
CP	Contention Period
CQM	Comb-shaped Quadratic Mapping
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CTS	Clear to Send
CWmax	Maximum Contention Windows size
CWmin	Minimum Contention Windows size
DCF	Distributed Coordination Function
DIFS	Distributed IFS
DM	Dynamic Mapping
DSM	Distributed Sequential Mapping
DVFI	Dispersive Video Frame Importance
EDCA	Enhanced Distributed Channel Access
ERVL	Expected Reconstructed Video Layers
FIFO	First in – First out
FTP	File Transfer Protocol
GOP	Group of Pictures
HC	Hybrid Coordinator
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HD	High Definition
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IFq	Interface Queue
LL	Link Layer
MAC	Media Access Control
MDC	Multi-Description Coding
MRC	Multi-Resolution Coding/Multi-layered video
MPEG	Moving Picture Experts Group
NAM	Network Animator
NetIF	Network Interface
OSI	Open Systems Interconnection
OTcl	Object oriented Tool Command Language
PC	Point Coordinator
PCF	Point Coordination Function
PHY	Physical

PIFS	Point Coordination IFS
PSNR	Peak Signal-to-Noise Ratio
QoS	Quality-of-Service
R-D	Rate-Distortion
RED	Randomly Early Detection
RTS	Request to Send
SBED	Significance Based Early Detection
SCS	Stream Classification Service
SIFS	Short Interframe Spaces
SM	Static Mapping
SNR	Signal to Noise Ratio
SSH	Secure Shell
SSL	Secure Sockets Layer
SVC	Scalable Video Coding
TCP	Transmission Control Protocol
TXOPlimit	Transmission Opportunity limit
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
VoIP	Voice-over-IP

Acknowledgements

‘**All Praise is for Allah by whose favor good works are accomplished.**’

I would like to express my heartiest and sincere gratitude towards my supervisor, Dr. Mohammad Saleh, who provided me with excellent help and support, invaluable advices, contribution of ideas and great encouragement through thesis work. Without his guidance and continually following up with me, this work would not have been achieved.

Sincerely, I would like also to thank my dear friend Fatmaa Amir for her support, helpful discussion, valuable hints, and for all her efforts during this thesis. In addition, I would like to thank my friend Eman Rizk for her help in collecting the results.

Moreover, I would like to thank Dr. Amr Mohamed and Dr. Abdulla al-Ali for interesting discussions and their help in broadening my knowledge in NS2 simulator.

I wish to extend my heartfelt gratitude to all my Qatar University professors and instructors for their help and collaboration during the master program.

Finally, thanks for my family and friends for their help, their endless love and for their prayers to make it worthwhile.

Chapter 1

1 Introduction

1.1 Research Overview and Motivation

The development of wireless networks has enabled several existing and emerging multimedia-streaming applications. These multimedia applications include products based on video information, such as interactive video conferencing and video streaming, which have become a significant part of daily life, especially with the rapid growth of information technology and the internet. Video applications are widely used in numerous fields, including military, education, medical, and entertainment spheres. Moreover, every mobile phone and computer device comes with an integrated video player and camera. Therefore, to support end users, several service providers have attempted to offer ideal infrastructure for universal video content and communications deployed with IEEE 802.11 wireless local area networks (WLANs).

Although video applications have deeply penetrated daily life, weaknesses remain in the quality of the video transmitted over wireless channels. Thus, delivering real-time multimedia video traffic over WLANs is a significant issue that has been addressed previously [1]. Moreover, due to the large consumption of available video data bit rate compared to other media, enhancing video quality must be considered to attain successful deployment of practical systems [1]. Hence, providing improved quality-of-service (QoS) in terms of bandwidth intensity, throughput, delay, and jitter is a challenging task in the efficient transmission of multimedia over a wireless network.

Video quality has improved significantly using different WLAN standards: the first WLAN standard, published in 1997, offered a data bit rate of only 2 Mb/s, while the most recent development, IEEE 802.11n published in 2009, offers a data bit rate of 600 Mb/s [2].

The IEEE 802.11e standard is an enhanced distribution of IEEE 802.11 that provides improved QoS for the transmitted data in the media access control (MAC) layer.

In addition to WLAN standard solutions for streaming real-time video, several non-standard solutions have been developed in recent years to optimize video transmission over WLANs, such as admission control, application-layer rate control, and cross-layer optimization [2]. The cross-layer design is a well-known approach designed to enhance video quality in different layers, such as the physical layer, link layer; transport layer, and application layer [3]. These layers are joined carefully in cross-layer solution to bridge the gap between different Open Systems Interconnection (OSI) layers. Cross-layer design has been used to address challenges such as existing network conditions and the QoS requirements imposed by applications [3].

Adaptive real-time applications offer the capability to transmit multimedia data over heterogeneous networks. However, wireless packet networks suffer from limited bandwidth transmissions, excessive delays, and congestion control, which degrade the video quality. Most real-time multimedia applications and services utilize User Datagram Protocol (UDP) as their transport protocol rather than Transmission Control Protocol (TCP). TCP is connection oriented which guarantees packet delivery in the same sending order and requires flow control before sending any user data. In addition, TCP overhead is significantly higher than UDP overhead, as it transmits more data per packet than UDP. Therefore, TCP is neither suitable for wireless networks nor for video applications due to its features that affect the goodness of the video quality. Conversely, UDP is a connectionless protocol that does not guarantee packet ordering, does not require flow control, and guarantees less overhead and low retransmission delay over a wireless network. Nevertheless, the UDP transport layer alone does not consider the varying channel conditions or the significance of video information generated from the application layer. These examples highlight challenges

necessitating a special type of transport layer control over wireless channels. Due to UDP challenges, a cross-layer design is required to take into account the significance of video data and varying channel conditions.

The development of the use of high-quality video applications over wireless networks by enormous number of mobile phone and tablet applications raised questions about the performance in quality issues, which led to a vast amount of research. Thus, there is a need for research that focuses on the creation of numerous solutions to obtain the required level of QoS for customers and end users. Some researchers proposed enhancing system performance by fine-tuning some of the controlled parameters in WLANs standards, such as data transmission rate [4], [5], contention window (CW) size [6], transmission opportunity limit (TXOPlimit) [7], [8], and other QoS requirements and resources [9]. Nevertheless, those mechanisms do not consider the significance of video traffic type, thus limiting the perceived video quality that can be obtained [1]. In contrast, other researchers examined the cross-layer solution of the transmitted video over wireless channels by exploiting the characteristics of different OSI layer architecture to optimize video quality [1], [10], [11]. Most of the existing research studies have been proposed for IEEE 802.11e mechanisms; therefore, multiple cross-layer solutions have been adopted to support QoS satisfaction for video applications.

1.2 Research Problem

Supporting QoS requirements for real-time multimedia traffic in WLAN is an active open research field. Many researchers have suggested different mechanisms to achieve QoS in this area. One branch of research in this field considers cross-layer solutions based on the IEEE 802.11e enhanced distributed channel access (EDCA) schemes model, which leverages MAC layer resources and improves video streaming performance [12], [13], [14], [15]. In a recent study, Romdhani et. al. [16], [17] proposed a distributed and adaptive cross-layer solution that involves an adaptive mapping of video layers into channel resources represented

as queues existing in the MAC layer. The mapping considers dependency between layers and adopts canonical mapping [16]. Their algorithm is known as distributed sequential mapping (DSM) strategy, which reportedly achieve improvements in the QoS of the delivered video.

The goal of this research is to implement DSM using a cross-layer design. Thus, the aim is to implement it at various OSI layers in a network model and thoroughly study it using various experiments, for instance, by reporting the performance of DSM when ongoing traffic of various types are present in the network.

The proposed implementation of the DSM strategy is simulated using a specialized network development environment namely NS2. These environments provide greater opportunities to conduct experiments in a controlled environment that adheres to the conditions for precise implementation, such as IEEE 802.11e, EDCA queues, and other entities in the system, thus ensuring accuracy when implementing the proposed model. The simulation model can be used in conjunction with other network models developed by other researchers in this integrated development environment (IDE). In addition, it can be easily extended or modified to produce other variations.

1.3 Research Scope and Objectives

The scope of this research is performing a cross-layer implementation of DSM strategy over IEEE 802.11 EDCA access schemes and experimenting it using simulations. The key objectives for this thesis can be summarized as follows:

1. To implement DSM over IEEE 802.11e EDCA access schemes using a cross-layer design that provides the best-effort service differentiation mechanism over IEEE 802.11e wireless networks for real-time multimedia applications based on a recently proposed mathematical model;
2. To use the implemented model developed by this work to conduct intensive simulation experiments on cross-layer design for video application over wireless

packet networks. These experiments include introducing various ongoing traffic and experimental settings;

3. To verify the validation of the DSM proposed by [16], [17] via determining whether it aligns with the results of the implemented model; and
4. To produce a simulation model that can be used in conjunction with other network models developed by other researchers in this IDE as well as the model can be easily extended or modified to produce other variations.

1.4 Thesis Overview

This chapter contains an overview of the research and presents the research problem and objectives. In Chapter 2, an overview of the main concepts used in this research and related studies regarding cross-layer designs with different mappings are provided. In Chapter 3, the network topology is described and the algorithm is summarized for the adopted model used in this thesis. Chapter 4 contains an explanation of the implementation of the adopted model using network simulator NS2. In Chapter 5, the simulation and analysis results for different simulation scenarios are described, and a detailed description of each simulation scenario is provided. Chapter 6 contains the conclusion of this thesis study and direction for future work.

Chapter 2

2 Background and Literature Review

This chapter contains background information about the main concepts related to this thesis and research in the study area.

2.1 Background

This section contains a necessary and sufficient review of related material required for an improved understanding of the research topic. The subsequent subsections introduce the required knowledge of these topics.

2.1.1 Video Coding

The demand for high coding efficiency in video compression is growing due to the prevalence of video content sharing, the growing number of streaming video providers, and high-definition (HD) TVs, which provide digital audio and video transmission over wired and wireless channels. In addition, WLANs or universal mobile telecommunications system (UMTS) suffers greater leaks from higher data rate transmissions than wired links do; thus, providing efficient coding enables the transmission of more video channels, which provides better video quality representations [2]. In simple terms, video coding can be referred to as the process of video compression (encoding) and decompression (decoding) [18]. Encoding is the process of converting a sequence of characters for file or a raw video format such as RGB or YUV into a particular format for efficient transmission or storage at the sender's side. Conversely, decoding is the process of converting the encoded video file back into its original format at the receiver's side. Original video (uncompressed video) requires a large amount of storage space before sending, while compressed video at the sender's side allows for a smaller version of a video format to be transferred to the receiver for decompression before displaying it to the end user [18]. Efficient coding or compression of video applications helps to reduce the amount of resources consumed over WLANs, such as transmission channel

capacity and data storage space [2]. Therefore, different video coding standards have been adopted in literature to compress video signals. These video coding standards can be classified into three types, namely non-scalable video coding, scalable video coding (SVC), and multiple description video coding (MDC), as indicated by Ke [19].

2.1.1.1 Non-scalable video coding

When using non-scalable video coding, the video content is encoded independently of actual channel characteristics [20]; Moving Picture Experts Group (MPEG) video frames is an example of this type of coding. It is composed of three basic frame types: I-frame (intra-coded frame), P-frame (predictive-frame), and B-frame (bi-directionally predictive–coded frame) [19]. I-frame is encoded independently of other frames and decoded by itself, while P-frame is encoded based on preceding I- or P-frames in video sequences. Finally, B-frame is encoded from previous and successive I- or P-frames [19]. Generally, efficient coding schemes can be obtained based on how each video frame is encoded. In non-scalable video coding, multiple small units called group of pictures (GOP) form the structure of the video frame sequence, as shown in Figure 1 such that a GOP is divided into a sequence of packets for delivery over a network used to prevent error propagation [19], [21].

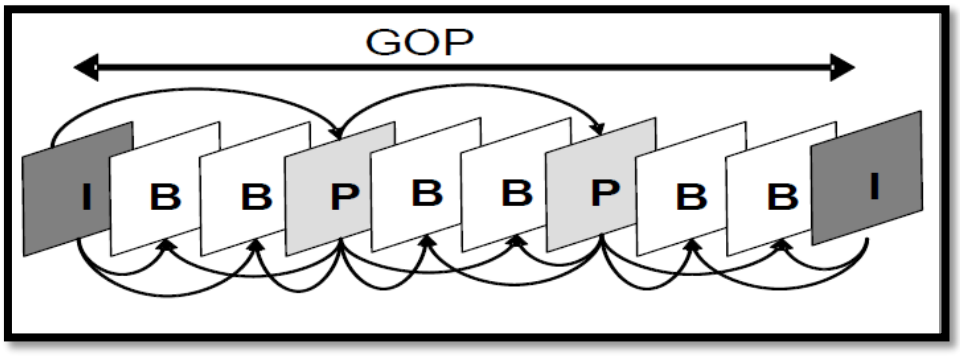


Figure 1: An example of MPEG coding with GOP [19]

However, this coding method cannot efficiently adapt non-scalable video stream content over heterogeneous network terminals with time-varying communication channels, especially for wireless applications [20].

2.1.1.2 Scalable video coding

In SVC, the video content is encoded into a base layer and several enhancement layers, a process referred to as layered video coding, such as multi-resolution coding (MRC). The base layer in MRC can be decoded to provide basic video quality, while the enhancement layers can be decoded with a basic layer to refine the quality of the video [19][20][22], as shown in Figure 2. Thus, all enhancement layers become useless if the basic layer is corrupted, even if they have been received perfectly. The important scalability features of this video coding include spatial, temporal and SNR scalability. Spatial scalability enables adaption to video resolution, temporal scalability enables adaption to frame rate, and SNR scalability enables adaption to video quality [19].

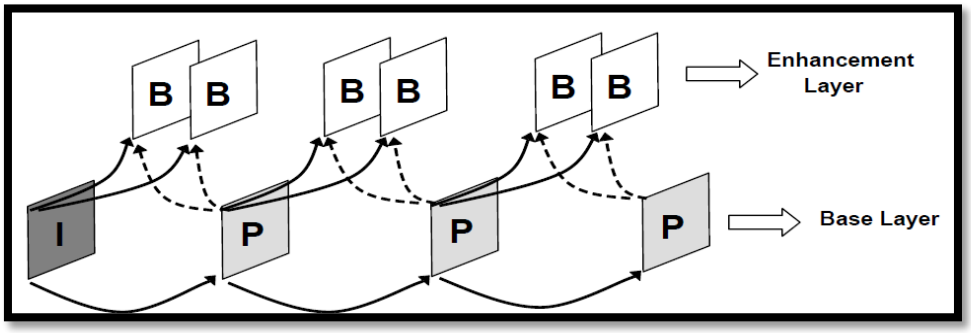


Figure 2: An example of temporal video coding [19]

2.1.1.3 Multiple description video coding

MDC is used mainly for speech communication over telephony networks. Its basic concept is to divide video signal into multiple decodable video sub-streams (n sub-streams) referred to as descriptors, where each descriptor or sub-stream is a composite of multiple packets (see Figure 3). The more sub-streams are received, the more information from the

original source can be restored [19], unlike in the case of scalable or layered video, where the base layer should be received almost error free to guarantee a basic level of quality otherwise the enhancement layers are useless [22]. MDC commonly is used to improve the error rate of a video delivery system because no retransmission of lost packets is required unless a high packet-loss rate occurs. Therefore, all packets are treated equally in MDC, and acceptable video quality can be maintained [22].

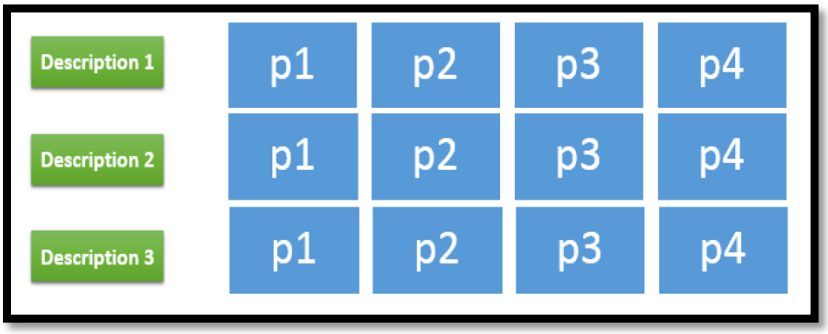


Figure 3: An example of MDC

2.1.2 Overview of IEEE 802.11 Standards for Wireless Networks

In past years, the main technology for the deployment of numerous wireless infrastructures is the IEEE 802.11 family of standards due to ease of communication and low cost [2]. The increase in popularity of WLANs led to the development of new applications that enable the delivery of multimedia services, including voice-over-IP (VoIP) applications, video streaming and conferencing, and online gaming [2]. To address this issue, the IEEE 802.11 standards have been enhanced continuously.

The IEEE 802.11 family of [23] WLAN standards covers MAC layer and physical (PHY) layer. The base WLAN standard was released in 1997 and can achieve a data rate of only 2 Mb/s, while the most recent standard, the IEEE 802.11n [24] released in 2009, can achieve up to 600 Mb/s. The IEEE 802.11ac [25] and IEEE 802.11ad [26] standards are presently under development and will provide higher data transfer rates—IEEE 802.11ad is

dedicated to supporting high-definition video streams with a data rate that can reach up to 7 Gb/s. The two currently used standards, IEEE 802.11e [27] and IEEE 802.11aa [28], were implemented to support improved QoS efficiency of video transmission over WLANs in the MAC layer. IEEE 802.11e was proposed in 2005 to offer higher priority and a differentiation access channel method based on the concept of multiple prioritized access category (AC) queues to reduce transmission delay [29]. Nevertheless, this standard does not consider unicast and multicast transmission and thus fails to fulfil video requirements [29], as only one of the AC queues is dedicated to video traffic regardless of the significance of video frames.

In contrast, IEEE 802.11aa [28] was proposed in 2012 to provide robust streaming of video over WLANs via a group of new error recovery mechanisms. It supports both unicast and multicast transmission and offers a stream classification service (SCS) based on the prioritized access mechanism [29]. Conventional IEEE 802.11 standards, such as 802.11a and 802.11b, provide higher data rates than the base WLAN standard released in 1997; however, the MAC layer in these standards does not support QoS [2]. Moreover, they are bandwidth restricted and delay sensitive, whereas IEEE 802.11e is the first amendment to enhance the QoS performance of 802.11 WLAN [2]. In the subsection that follows, IEEE 802.11e and the differences between IEEE 802.11 and IEEE 802.11e are illustrated in detail.

2.1.3 IEEE 802.11 MAC layer

The MAC located in the data link layer is responsible for providing QoS support to end users. The MAC layer is composed of two transmission modes for transmitting data packets: the distributed coordination function (DCF) and the point coordination function (PCF) [30]. Both modes use a collision-free period-repetition interval formed by a PCF contention-free period (CFP) followed by a DCF contention period (CP) [30], [31], as shown in Figure 4.

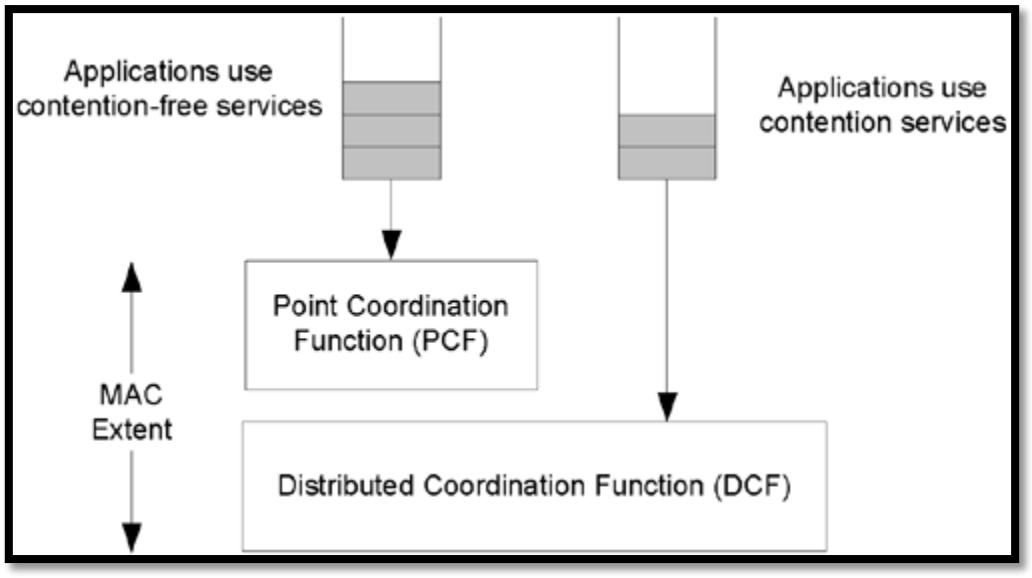


Figure 4: IEEE 802.11 MAC architecture [30]

DCF is the fundamental access mechanism of IEEE 802.11 that provides asynchronous transmission and uses carrier sense multiple access with collision avoidance (CSMA/CA) as a medium access mechanism. It allows the sharing of wireless resources between radio transmitters. Moreover, it allows collision and contention between stations, thus it can be used for applications that do not require QoS. In contrast, PCF provides synchronous transmission and uses a polling-based access mechanism as the point coordinator (PC), for example, access point (AP), which is responsible for controlling all transmissions [31].

Generally, the IEEE 802.11 standard identifies four main types of interframe spaces to describe different priorities, namely short-interframe spaces (SIFS), point-coordination IFS (PIFS), distributed IFS (DIFS), and arbitrary IFS (AIFS) [27], as shown in Figure 5. The SIFS is utilized to transfer small frames, as it is the smallest IFS and has the highest priority, for instance, Acknowledgment (ACK) , Request to Send (RTS), and Clear to Send (CTS)

[27]. The PIFS is the second smallest IFS, which is utilized by the hybrid coordinator (HC) to acquire the medium before any other stations [27]. The DIFS has the lowest priority and is utilized for asynchronous data services, which are used by other stations to wait after sensing an idle medium. Finally, AIFS is the IFS utilized by different access categories (ACs) in EDCA to wait after sensing an idle medium [27].

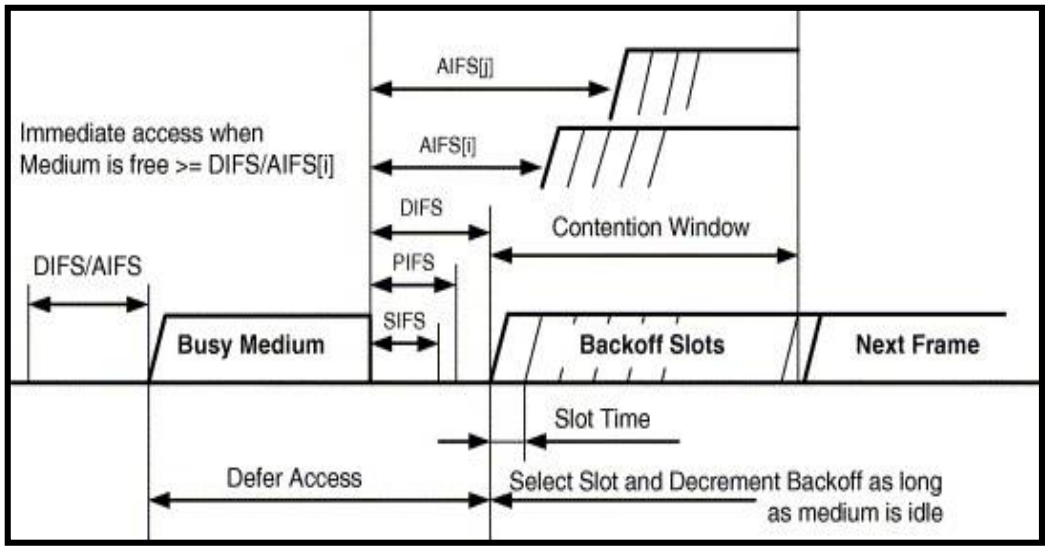


Figure 5: IEEE 802.11e EDCA operations [27]

IEEE 802.11 standards work well with data traffic but lack the consideration for the QoS requirements of multimedia applications. Therefore, IEEE 802.11e offers QoS for multimedia traffic applications.

2.1.4 IEEE 802.11e (EDCA)

The IEEE 802.11e standard consists of a MAC layer coordination function in the data link layer called HCF (hybrid coordination function) to support QoS [27]. HCF is classified as two main channel access methods, which are enhanced distributed channel access (EDCA) and a centralized polling-based channel access mechanism, called HCF-controlled channel access (HCCA) [27]. EDCA improves the DCF by introducing traffic prioritization, while

HCCA enhances the PCF polling scheme with a parameterized traffic classification [32]. All the details of the HCCA are beyond the scope of this thesis, as it is restricted to the EDCA.

The EDCA is an enhancement of IEEE 802.11. The distributed coordination function (DCF) mechanism defines four prioritized AC queues to support service differentiation for different traffic resources for channel access. These access queues, denoted by {AC_n, where n=priority}, are used to improve the delivery of multimedia traffic [21], as shown in Figure 6. In standard MAC layer coordination function, a single FIFO queue is used; hence, collisions may occur if any two workstations are attempting to access the medium simultaneously [21]. Thus, the service differentiation among these ACs in EDCA scheme is achieved by assigning different prioritized traffic flows for each AC_n. Such that AC_0 or AC_BK is used for background traffic, AC_1 or AC_BE is used for best effort traffic, AC_2 or AC_VI is used for video traffic, and AC_3 or AC_VO is used for voice traffic [27].

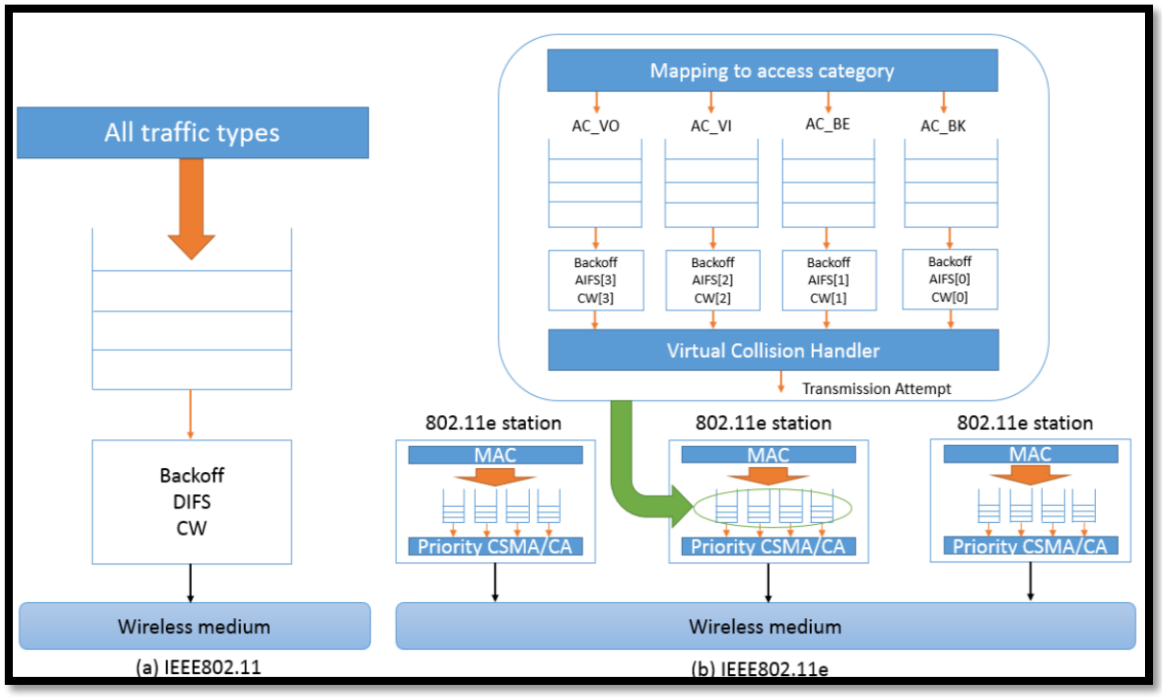


Figure 6: (a) FIFO queue in conventional IEEE 802.11 (b) Four access categories in IEEE 802.11e

Each AC has its own buffered queue and is determined by specific AC operational parameters, called EDCA parameter sets, including arbitration interframe space (AIFS), minimum contention window size (CWmin), maximum contention windows size (CWmax), and transmission opportunity limit (TXOPlimit) [27]. CWmin and CWmax are sizes used for selecting random backoff counters [33]. The backoff counter can be determined as a random integer value selected from the interval [0, CW]; the CW is initialized with CWmin. When a transmission fails, the CW value increases. If there are still any unsuccessful transmission attempts, another backoff counter is set with a new CW value as 2 * (CW + 1) – 1, with an upper bound of CWmax [32]. Therefore, if multiple stations are attempting to access the channel medium, collision probability is reduced. After successful transmission, the CW value is reset to CWmin, and the station performs another DIFS and another random backoff, even if there is no pending frames in the queue [32]. TXOPlimit can be defined as the time interval used to send more than one data frame continuously per each AC when a particular station initiates its transmission such that higher priority AC guarantees longer TXOPlimit [34].

Each station represents an individual AC queue, which has its own CWmin, CWmax, AIFS, and TXOP. Table 1 shows the recommended values of parameter sets for all ACs in EDCA to support service differentiation among those ACs. The AC with the highest priority is assigned by the smallest values of the EDCA parameter set (AIFS, CWmin, CWmax, and TXOP) to acquire the channel medium first [27]. Although service differentiation can be guaranteed among the EDCA parameter set in EDCA, the obtained performance is not optimal, as those parameters can be adapted with varying network conditions.

Table 1: IEEE 802.11e EDCA parameter set [21]

Priority	AC	Designation	AIFS	CWmin	CWmax	TXOP limit
3	AC_VO	Voice	2	7	15	0.003008
2	AC_VI	Video	2	15	31	0.006016
1	AC_BE	Best effort	3	31	1023	0
0	AC_BK	Background	7	31	1023	0

Before beginning of transmission over the channel medium, each AC within a station senses the medium and starts its backoff timer if the medium is idle for at least the AIFS period, where the backoff timer is determined by CW sizes. However, the station defers its access to the medium and initiates a backoff timer to avoid collision if the medium is busy [35]. Collisions in EDCA can be classified according to two types of collision that can occur in the wireless channel: internal collision and external collision [35]. Internal collision occurs at the same station when more than one AC attempts to transmit simultaneously and their backoff has been completed and counted to zero [35]. In this case, the AC with the highest priority acquires the medium, and the lowest priority ACs wait and their backoff restarts [34]. Conversely, external collision occurs when the EDCF backoff timer of more than one station counts to zero at same time and those stations acquire access to the medium [35]. These ACs use a special congestion control or avoidance mechanism to determine an earlier stage of collision within a single wireless station before sending the packet out to the wireless channel; then a virtual collision controller grants access to the AC with the highest priority to resolve collisions [21].

2.1.5 Cross-layer Solution

The cross-layer solution is one of the non-standard mechanisms that has been used widely to overcome problems caused by video transmission over wireless channels. Numerous researchers have investigated various cross-layer solutions.

The OSI model is an example of layer architecture that deals with the overall network operations and network services as layers [36]. The cross-layer approach is inter-layer communication that can be defined as a protocol design that allows communication between different OSI layers by sharing information among all layers and not restricting it to only adjacent layers. Sharing information has been accomplished by permitting one layer to access information from another layer that is not necessarily adjacent to it, thus exploiting the dependency between protocol layers to improve performance [3], [36], [37].

Fu et al. in [36] categorizes the goals of cross-layer design as QoS, security, and mobility. First, QoS targets the improvement of video quality at the application layer by exploiting the characteristics of both the physical layer and data link layer over a wireless network. Several issues may affect QoS in WLANs, including transmission error due to packet losses. Second, security is one of the most important goals that, when applied, guarantees better system performance with secure communication. Thus, security includes protocols that deal with security factors, including encryption and decryption methods such as secure shell (SSH) and secure sockets layer (SSL) at the application layer, which are used for end-to-end communication encryption. Last, mobility aims at sustained communication in wireless networking without experiencing interruptions caused by node movement, which requires changing routes or channel resources.

2.1.5.1 Categories of cross-layer designs

Many previous studies on cross-layer design over wireless networks list several classifications for cross-layer designs, and numerous surveys have been conducted to

summarize some of these works, as shown in [3], [36], [37], and others. In one of the recent surveys conducted by Fu et al. [36], the researchers classified the cross layer into two main categories: either by how information can be shared across OSI layers, which occurs inside one node, or by the network organization, which occurs among multiple nodes, as described in Table 2.

Table 2: Classifications of cross-layer design [36]

	Sharing information among OSI layers in one node		**The organization of a network among multiple nodes**	
Methods	**Non-manager method**	**Manager method**	**Centralized method**	**Distributed method**
Explanation	The communication is allowed between any pair of layers directly	There is a vertical plane as a manager to share data between layers	Uses a centralized node (i.e., base station or router) to manage data exchange of OSI layers between nodes	Does not use any centralized node to share information where multipath communication from one node to another is possible during information sharing
Difference	Affects the waterfall structure of the OSI layer because any two non-adjacent layers become adjacent	Does not affect waterfall structure of OSI layer because it introduces a vertical plan for managing data exchange, but the functionality of protocols in layers could be changed	This method is used in cellular networks.	This method is used in ad-hoc networks.

Another commonly used classification has been presented by [3], [37] and [38]. They classified a cross-layer solution based on the direction of information flow between layers, which is composed of five cross-layer approaches, including a downward approach, upward

approach, hybrid approach, MAC-centric approach, and joint adaption approach. Each of these approaches defines interaction between layers based on the path of information flow. For example, the information in the downward approach (top-down) is from the top down to allow the higher layer to obtain the information required to perform optimization or adaption from lower layers. However, the information flow in the upward approach (bottom-up) is from the bottom to the top, which allows lower layers to pass the information required for optimization to a higher layer. In contrast, the hybrid approach joins both the top-down and bottom-up approaches. For example, a system can feature the top-down approach where a higher layer can exploit some features from network or MAC layers while lower layers simultaneously can adapt some parameters from the application layer based on user preference. In the MAC-centric approach, the application layer passes on the required information, such as traffic flow and QoS requirements, to adjust MAC layer parameters. Finally, the joint adaption approach involves adjusting different strategies or schemes at different layers to enhance overall performance.

There are many benefits of using a cross-layer design that provides improved system performance, such as it allows direct inter-layer communication at any non-adjacent layers [37]. In addition, it allows the sharing of information between all OSI layers, including internal parameters, status, and other required information [36]. Thus, sharing information between layer boundaries enables the visibility of hidden information and compensates for network performance and reliability through controlling input parameters from one layer to another, including channel state information. Furthermore, it permits ongoing communication via the dynamic adaption of network changes and time-varying conditions [36].

Nevertheless, there are also some drawbacks involved in using an approach like the one presented by Fu et al. [36]. The main disadvantage of the cross-layer design is the destruction of the layered architecture of the computer network. In addition, the

characteristics of each cross-layer design differ, and they have their own specific communication manners and challenges, including coexistence, signaling, and overhead aspects. Although sharing information between layers is the main advantage of cross-layer design, this sharing of information may cause extra overhead when exchanging information across layers. Finally, there is no universal cross-layer design that can be adapted automatically to different applications, as each cross-layer design is dedicated to a special purpose or application.

2.2 NS2 Network Simulator

NS2 is one of the most widely used network discrete-event simulators for research purposes. This well-known open source simulator has been adopted as the simulation tool for this research. NS2 [39] is a packet-level simulator where packets, nodes, and access points (APs) are implemented accurately to reflect their physical form and to realize the implemented theories before actual implementing them, thus saving time and cost.

2.2.1 NS2 Overview

The NS2 project [39] was started in 1989 as a variant of REAL network simulators. Between 1995 and the present day, it has been supported by DARPA through several projects, including the VINT project and SAMAN and NSF with CONSER project. NS2 [39] is a discrete event simulator that consists of two languages, C++ and object-oriented tool command language (OTcl), which are linked together using TclCL language, as shown in Figure 7. C++, a rapidly compiled and robust language, defines the structure of the simulator, while OTcl is an interpreted high-level programming language that defines the simulation set up by specifying scenarios and configuring objects as discrete events. Examples of events include enqueueing a data packet and receiving an acknowledgement packet.

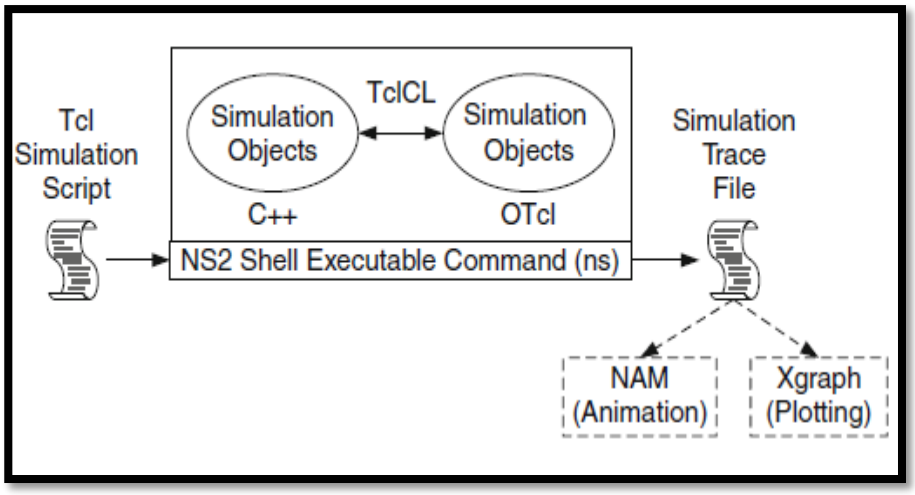


Figure 7: Basic architecture of NS2 [39]

NS2 [39] simulates both wired and wireless networks, which allows the creation of different nodes with different types of links between them and the addition of different types of traffic to these links. Furthermore, it simulates additional network types, such as sensor networks and satellite networks.

To observe network behavior, NS2 [39] comes with additional tools such as network animator (NAM), which is used to visualize the network topology; i.e., it shows packet flow in the network. The output of running OTcl script is in the form of a trace file that is used later to analyze results. Trace files include data such as sender, receiver, packet type, and packet size according to the created network topology. Moreover, with the NS2 trace file, one can use the Xgraph tool to create graphic representation of simulation results as well as plots and figures.

2.2.2 Why NS2?

As stated in subsection 1.3, this research is focused on the practical application of theories and assumptions recently published by Romdhani et al. [16], [17] using the NS2, a real-time simulation environment. NS2 is the open-source network simulator most widely

used for ad-hoc wireless simulations. Although a significant amount of research has been conducted in the field of video delivery across ad-hoc networks, in most new research and studies, researchers have used MATLAB for simulations to validate their mathematical algorithms. However, this system has numerous limitations. Although MATLAB offers fast simulation, the results produced may suffer significant accuracy loss. Therefore, it is possible to produce more accurate simulation results with these mathematical models using NS2. The following points explore what the NS2 simulator allows one to evaluate in terms of practical feasibility:

- Any protocol can easily be developed and extended in NS2; therefore, an extension of UDP protocol has been created to implement the cross-layer design and DSM model proposed by [16].
- The packet, node, and AP entities have not been considered in [16], while NS2 offers a real network packet level structure in which nodes and AP can be tested physically.
- The dependency between video layers has not been examined practically, thus the real behavior of the DSM algorithm in mapping video layers to appropriate ACs has not been considered either.
- In the algorithm developed in [16], the access delay model, dropping probability values are estimated using the Markov chain [40]. However, the NS2 simulator allows one to determine the packet loss due to collisions and congestion and to evaluate the effects of delay times.
- The data model used could be restricted to fewer video types, but NS2 features numerous video types that can be plugged-in to evaluate the effect of delivering multi-layered video over wireless network.

Hence, the most important components of the system were not presented properly. As a result, the values obtained, such as drop rate, delay, jitter, and other performance measures, may not be as accurate as desired.

Such concerns and interest in the outcome of this research have been addressed by further investigating this research work using NS2, which provides the opportunity to conduct more intensive studies using various types of video traffic patterns with different experimental settings. Nevertheless, NS2 is a complicated structure that suffers from large memory overhead and slow speeds because of its long time simulation for enormous network simulations. Further details about cross-layer and DSM implementation are discussed in the next chapters.

2.3 Literature Review

In this section, the related work carried out on previous cross-layer solutions of mapping algorithms for video transmission over wireless networks based on the EDCA scheme. The IEEE 802.11e standard offers three types of mapping mechanisms, which have been used in studies to allocate packets to prioritized queues (ACs). These mapping techniques are described as the following [41]:

- **IEEE 802.11e mapping (default mapping):** allocating channels to each type of traffic is fixed;
- **Static mapping:** allocating channels to packet types is priority based. As discussed in section 2.1.1, each GOP consists of three types of frames denoted by I, P, and B; and
- **Dynamic mapping:** video packets are mapped to appropriate ACs based on network traffic load and the significance of video data.

Transmitting video applications over wireless channels suffers from large delays, limited bandwidth, and inevitable burst losses caused by interference, fading, shadowing, multipath, and so forth [42]. IEEE 802.11e uses four ACs, as described in section 2.1.4, to

provide service differentiation across video packets, thus providing improved QoS for video delivery. The major problem in the EDCA scheme is that it maps all video packets to AC_2 by default (i.e., AC_VO), which is reserved for video traffic. Mapping all video packets to AC_2 can cause a problem when there is a large number of video packets and the corresponding queue (AC_2) is full, thus degrading the reconstructed video quality [21], [42]. As the other three AC queues are not used, it will not distinguish the significance of variant video flows [34]. Moreover, AC_2 is FIFO based and does not support differentiation between the importance levels of video packets themselves, therefore limiting the performance of video quality over IEEE 802.11e [21].

Consequently, numerous advanced mechanisms have been proposed to improve the quality of video delivery over wireless networks. Some of the proposed mechanisms enhance the performance of MAC layer operations by adjusting some of the controlled parameters, such as data transmission rate [4], [5], CW size [6], and TXOP limit [7], [8]. Nevertheless, those mechanisms did not take into account the significance of video traffic types, thus limiting the perceived video quality that can be obtained [21].

To support service differentiation among video packets and to consider the importance levels of different video traffic types, more efficient mechanisms using a cross-layer design have been adopted to overcome the problem of the EDCA default mapping scheme by using other types of mapping, such as static mapping (SM) and dynamic mapping (DM).

Ksentini et al. [43] and Chen et al. [15] proposed a static mapping (SM) algorithm that exploits the significance of video layers from the application layer and the features of the EDCA MAC layer to support QoS transmission and to improve video quality over an IEEE 802.11e network using a cross-layer design architecture for H.264-based video streaming. The proposed algorithm allocates the prioritized H.264 video data packets to different ACs based on their video coding importance levels, which are assigned statically in the application

layer using a different marking algorithm in both studies, depending on the temporal level in the application layer [15], [43]. For instance, all H.264 video streams including I-frame, P-frame, and B-frame are mapped to AC_2, AC_1 and AC_0 respectively. The metrics used in both studies are the packet loss rate [15], [43] and peak signal-to-noise ratio (PSNR) [15]. The results of their performance analysis indicated that, in the case of heavy network congestion, this SM algorithm outperforms the default-mapping algorithm in EDCA, which maps all video packets to AC_2. However, during light network congestion, the results showed that EDCA provides higher quality than SM because video packets are mapped only to lower prioritized ACs (i.e., AC_1 and AC_0), causing unnecessary delays and packet losses. In addition, it does not change the fact that AC queues are FIFO based. Hence, this SM algorithm cannot adapt varying network conditions.

Mai et al. [1] proposed another cross-layer design to improve video quality by developing an adaptive MAC layer prioritization mechanism called MAP, which estimates waiting time for each AC and maps each video packet to the prioritized ACs with the shortest waiting time. This prevents packets being sent to ACs with long waiting times in the queue. The importance of the video packet level depends on the PSNR value such that the lower the PSNR, the more important the packets in the application layer. Because important packets are sent first and low-priority packets are dropped based on their PSNR values, the researcher used the PSNR as a metric to evaluate video quality. This study was conducted on a real testbed, and the performance analysis showed that this cross-layer design enhances video quality even in a congested wireless network. Although the MAP mechanism avoids the extra video packets, such selection in the mechanism results in complicated waiting time calculations for all ACs because they may affect each other.

All the above-mentioned mapping mechanisms are static and not adaptive. Furthermore, SM performs worse than EDCA default mapping in some cases, such as in light

network congestion where all packets are mapped to low priority queues suffering from high packet losses, which prevents these techniques from adapting network traffic variations. In contrast to previous SM algorithms, the adaptive mapping (AM) algorithm proposed by Lin et al. [44] features a cross-layer design that dynamically maps video data packets to appropriate AC queues according to their importance levels. This AM algorithm is based on I/P/B video frame types and on real-time network traffic. This technique considers the queue length of each AC in the EDCA MAC layer as an indication of current network traffic load. It can probabilistically decide whether prioritized video packets can enter a higher priority queue (i.e., AC_2) or lower priority queues (i.e., AC_1 and AC_0). Additionally, it adopted two queue length thresholds, denoted as *threshold_low* and *threshold_high*, to determine network congestion level.

To illustrate this point, AC_2 in the AM algorithm is not a FIFO based queue but adheres to the random early detection (RED) principle. Such that each associated, I/P/B video frame follows a specific RED packet mapping function that dynamically obtains the probability of downward video packet mapping to lower priority queues depending on the queue length of AC_2. The downward video packet mapping used by [44] states that when a packet is sent and if the queue length of AC_2 is less than *threshold_low* (i.e., traffic load is light), all video packets are mapped to AC_2 (i.e., AC_VI). However, if the queue length of AC_2 is greater than *threshold_high* (i.e., traffic load is heavy), video packets are mapped to AC_1 and AC_0. If the queue length of AC_2 is between *threshold_low* and *threshold_high*, video packets are mapped to AC_2 or AC_1 based on the result of a new probability formula that is compared to random number ranges between 0 and 1 to identify video frame queue's location. The results of this algorithm have been tested against both EDCA and SM, and the results showed that AM outperforms both EDCA and SM in different network traffic conditions.

Based on the work of Lin et al. [44], several studies achieve improved video QoS over IEEE 802.11e networks. For example, Li et al. [42] proposed a cross-layer design based on a novel significance-based early detection (SBED) algorithm, which dynamically maps SVC packets to prioritized ACs based on the significance measurement of packets and considering network traffic load. Thus, they have used the characteristics of both application layer and MAC layer to enhance video quality in the same way as Lin et al. [44]. The only difference is that they measured the significance of packets using a predefined formula that considers the importance level of I/P/B frames to identify the location of the video frame queue instead of randomly estimating it. The experiments for this cross-layer design have been evaluated against the conventional EDCA scheme, SM implemented by [15], and AM adopted by [44]. PSNR was used as an evaluation metric where the results showed that the SBED algorithm outperforms the three algorithms in case of transmitting the base layer, second layer, and third layer. However, the last high level layer in the SBED algorithm does not guarantee the best video packet transmission because enhancement layers have little significance; thus, they have a small effect on video quality and can be dropped when the queue fills.

In addition, Shi et al. [34] improved the work of Li et al. [42] by including the quality layers method in the previously proposed cross layer using the SBED algorithm. The quality layers method includes a rate-distortion (R-D) optimization mechanism, which assigns a prioritization order for each SVC stream in the application layer such that for each video picture, the rate and distortion must be calculated to obtain slope, which is measured by dividing distortion with rate. Then, all pictures are sorted in descending order according to their predicted R-D slopes such that the larger the R-D slope is, the better the video quality efficiency is. Moreover, to guarantee successful transmission of all packets in the base layer, all packets are assigned a value from 0 to 63 according to different levels of priority; a 0 value means video packets have the highest priority (base layer). Experiments conducted by

applying this quality layers method with the SBED algorithm and comparing it to the three previous approaches. Results showed that their proposed design achieved the best video quality and improved the received video packet efficiency.

Furthermore, Lai [21] proposed a cross-layer design called the DVFI+CQM framework, which is composed of two parts: a dispersive video frame importance (DVFI) scheme in the application layer and a comb-shaped quadratic mapping (CQM) algorithm in the MAC layer. DVFI scheme is responsible for assigning video frame importance levels precisely based on PSNR degradation value, while CQM algorithm is responsible for mapping prioritized video packets to appropriate AC queues. Each video frame obtains the DVFI value using PSNR degradation caused by its transmission loss, using the impact of temporal coding dependency for other video frames. Accordingly, each video frame is allocated to a reserved AC queue based on the CQM algorithm, which dynamically maps less important video frames to lower priority queues using a multi-branched service differentiation similar to that of the approach of Lin et al. [44]. However, the proposed cross layer uses the DVFI scheme to label the importance of video packets and to keep the most important video packets in AC_2. In addition, it maps the less significant video packets via a minimum delay time function that selects between AC_1 and AC_0 as early as possible, depending on which one has the shortest delay time considering queue length and corresponding resources. Comprehensive experimental tests have been conducted to ensure the novelty of this cross-layer design for video quality delivery over IEEE 802.11e. These exhaustive tests include a PSNR metric to evaluate performance of the proposed cross-layer design against congestion levels of ACs, the queuing effect of ACs (i.e., most important video packets to AC_2 and less important video packets to AC_1 and AC_0), and the queue length. The results obtained were compared with the findings of previous studies, and the

results indicated that DFVI+CQM outperforms existing approaches, including conventional EDCA mapping, SM, and AM.

Similarly, Chen et al. [45] recently published a study regarding a cross-layer design based on the AM algorithm [44]. The difference between the two algorithms is that in the study of Chen et al., the researchers considered the significance of I/P/B frames to identify the location of the video frame queue. For example, when a packet arrives at the MAC layer, the type of frame has been checked. If it belongs to the I-frame, it is appropriately assigned to one of the empty queues (i.e., AC_2, AC_1, or AC_0), and if one of them is full, it is delivered to the next empty AC queue. However, if the video frame belongs to either the P-frame or B-frame, it must be checked against two formulas containing thresholds previously adopted by [44], *threshold_low* and *threshold_high*. The important issue in this study is the appropriate allocation of the AC queue. Such as efficient usage of queue length is ensured by delivering video frames to the current queue or next empty queue, thus decreasing the probability of dropping packets due to full queues. The performance evaluation for this study was completed using PSNR as a metric to show packet loss number for plugged-in multimedia highway video sources in all algorithms. Then results have been compared against EDCA, SM, and AM results. The results indicate that EDCA provides a better PSNR value when the traffic load is light; when the traffic load is heavy, this cross-layer design achieves the best performance among all studies conducted previously.

Romdhani et al. [16], [17] proposed the distributed sequential mapping (DSM) algorithm and claim that it improves the QoS of transferred video layers over EDCA priority queues in wireless channels. This technique uses a dynamic algorithm that takes into account different layer rates and the channel conditions to estimate the optimal mapping strategy between the perceived video layers and the EDCA priority queues. This mapping technique performance was confirmed using validations carried out using MATLAB. Moreover, all

recently developed video measurement approaches consider throughput, delay, drop rate, PSNR and numerous other parameters to evaluate video quality. However, because none of them considers dependency between layers, Romdhani et al. [16], [17] proposed a new evaluation metric that considers the dependency nature of layered video delivery called expected number of reconstructed video layers (ERVL). For instance, high throughput can be obtained, but with more packets from higher layers than lower layers, which reduces video quality because higher layer packets depend on lower layer packets.

The simulation results showed significant trade-offs between the complexity and the delivered video quality of the mapping schemes used, including canonical mapping and rate-based mapping. In canonical mapping, the number of layers assigned to each queue increases with the queue priority level. Whereas in rate-based mapping, the layers have different layer bit rate. In addition, it has been shown that canonical mapping can provide better QoS than other mapping strategies and that the efficiency of the DSM helps to maximize the average number of perceived layers.

The next chapter includes a description about the adopted system model in this study. Further details of DSM algorithm and cross-layer design are presented.

Chapter 3

3 System Model

Video delivery over wireless network can be classified into two main delivery models, namely distributed and centralized delivery models. The centralized delivery model consists of a wireless controller device, such as a router or access point, where the controller device is responsible for coordinating centrally between users' devices and PCs to track wireless devices and to troubleshoot problems easily [46]. The main vulnerability of this model is that if there is a single point of failure in the network controller node, it will cause a cut off in the entire network, which wastes time in locating and fixing the problem [46]. In contrast, the distributed delivery model considers that each user in the network performs tasks without the aid of any central controller. On decentralized networks, if one node or AP fails, other nodes keep working.

Hence, several approaches have adapted the decentralized approaches or distributed approaches. Similarly, the aim of this study, as explained in subsection 1.3, is to improve video quality through a practical evaluation of the performance of an analytical model proposed by Romdhani et al. [16], [17]. This chapter contains a detailed presentation of the video delivery model adopted in this study, and the main assumptions and theories of the implemented model are also examined. The next chapter includes an explanation of how to realize the adopted model using a cross-layer design via the NS2 simulator.

3.1 Network Topology

The network model is composed of a group of heterogonous nodes or users presented by n = 1... N. Each node is located at a fixed position and has a video with a number of layers denoted by L, L = L1…L_l. For example, a network topology made up of five users and three video layers, as shown in Figure 8, is considered. It is equipped with WLAN 802.11e and radio antennas to allow communication with one another. Thus, nodes can send, receive, and

forward data to other nodes via one-hop wireless links. All nodes send to and receive from other nodes via APs, so all nodes are sharing the same wireless channel and the same ACs. In this network, every node is trying to transmit the video layers from the source to its destination using different types of permutations “mappings” (for more details about how to find these mappings or permutations, see section 4.2.2.2). If every node is trying to work on its video layers in parallel with other nodes in the network, the AP should be informed about every node’s changes and about the traffic loads that are causing huge delays, overhead, and congestion problems, which is impractical and unreasonable.

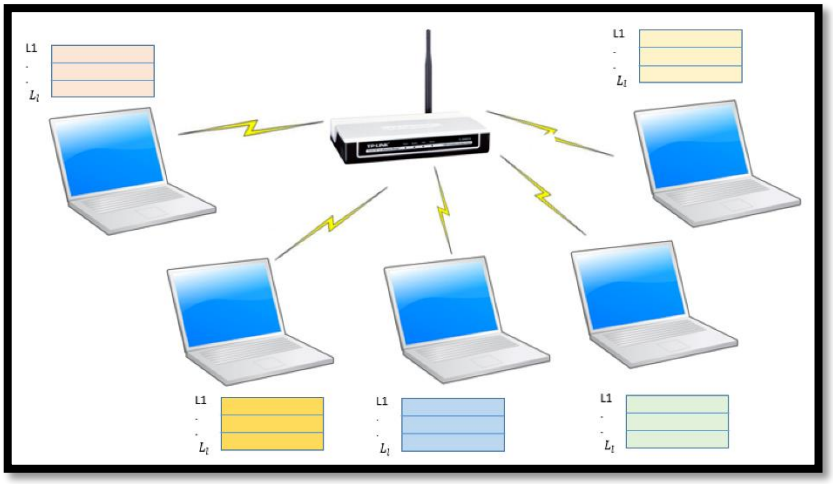


Figure 8: Network topology structure

Therefore, it is required to use a distributed sequential method to allow the AP to be aware of all changes that occur in the network node by node. As mentioned earlier in this section, in distributed systems, each node is permitted to communicate, coordinate, and do all assigned work only by passing messages without referring back to any centralized device. Hence, it allows resource sharing among all nodes in the network, manages fault tolerance, improves network performance, and ensures scalability and flexibility [47]. In distributed

systems, signaling overhead is almost null, as the AP attempts to signal one node at a time. If the node performed all the required work, it sends confirmation that the work has been completed to the AP, so the AP signals the next node in the network and so on. Thus, synchronization is important because the physical medium is shared.

3.2 Principles of Distributed Sequential Mapping (DSM) Video Delivery Model

The solution suggested by Romdhani et al. [16], [17] addresses an analytical model for a new distributed sequential mapping (DSM) strategy of video layers that improves the QoS of the transferred video over EDCA priority queues in the wireless channel. The main idea of the proposed model is to map arriving video layers into different EDCA ACs to adjust video quality delivery by maximizing the number of perceived video layers. Assumptions and theories of this analytical model are discussed in detail in the subsections that follow.

A simple EDCA model that estimates internal and external collision probabilities and interface queue dropping probability has been considered. Interface queue dropping probability includes the computation of packet drops due to queue overflow. The simple EDCA further features a delay model to contribute all events to access delay. The delay model for this EDCA can be summarized as follows:

Assume that U denotes video users and n_{AC_i} denotes number of ACs for all users contending for the channel medium, such that each layer is assigned to one AC_i per one video user. Authors in their work have considered the Markov chain mathematical model [48], [40] in their system model and extended the probability formulas to support a differential TXOPlimit parameter in the different computed performance metrics [14]. Suppose τ_i denotes probability when a node in AC_i transmits during a generic slot time, and P_i is the probability that AC_i senses a busy medium. The τ_i takes into account both internal and external collision, thus, probability τ that a node transmits in a given time slot is:

$$\tau = 1 - \left(\prod_{i=1}^{4} (1 - \tau_i) \right) \tag{3.1}$$

The collision probability for both internal and external collisions for an AC_i, is defined as follows:

$$P_{coll,i} = 1 - (1 - \tau)^{U-1} \prod_{h<i} (1 - \tau_h) \tag{3.2}$$

The probability $P_{succ,i}$ that a slot contains a successful transmission of frame of AC_i is given by:

$$P_{succ,i} = U * \tau_i (1 - P_{coll,i}) \tag{3.3}$$

The dropping probability $P_{i,drop}$ of each AC_i is:

$$P_{i,drop} = 1 - \left((1 - P_{i,drop,coll}) - (1 - P_{queue\ drop,i}) \right) \tag{3.4}$$

where $P_{queue\ drop,i}$ represents the probability that a packet is dropped due to queue overflow, and $P_{i,drop,coll}$ represents the probability that a frame is dropped due to reaching the maximum retry limit equals to 7 times.

3.2.1 Adaptive Video Mapping Techniques (Permutations)

Delivering multi-layered video over a wireless channel is not an easy task. IEEE 802.11e or EDCA provides discrimination of ACs via the EDCA parameter set, which is used to create discrimination between video layers. In practical terms, the number of video layers exceeds the number of ACs (i.e., the default number of ACs is equal to 4 ACs), so more than one layer should be mapped to each AC. Higher priority ACs should always have more layers than lower priority ACs. Accordingly, numerous permutations from the exhaustive search have been eliminated, thus the optimal solution should be one of the permutations called "canonical mapping" [16], [17]. To achieve QoS, it is important to have all those

permutations for wireless. The permutation differs based on the number of ACs assigned to each mapping.

The main system model includes exhaustive mapping, which defines all possibilities of mapping vector (MV). Let N be the maximum number of ACs, which equals 4 in the case of basic EDCA; thus, $\Delta MV = \{MV(L,n): 1 \leq n \leq N\}$ where each user has to execute an exhaustive search locally to define the best mapping vector among ΔMV. Exhaustive key search or brute force [49] consists of systematically enumerating all possibilities of the mapping vector ΔMV of all video layers to four ACs and then checking whether each permutation can obtain the best video quality. For example, when using exhaustive mapping, 10 layers distributed across 4 ACs can produce up to 210 combinations, according to the following formula:

$$\frac{n!}{(n-r)!\ r!} \tag{3.5}$$

This number was reduced to 23 ordered permutation-mapping techniques as shown in Table 3 below. The reduced mapping refers to an adaptive mapping strategy called canonical mapping or permutations, which is defined as "a subset of mapping strategies to enhance performance of searching for optimal video layers mapping" [16], [17] as shown in Figure 9.

As mentioned in subsection 2.1.1.2, the base layer is the most important video layer used to provide the basic video quality. In canonical mapping, the base layer should be always mapped to highest priority AC (AC_VO or AC_3) while other layers should be mapped based on their permutation taking into account that all layers should be mapped in a non-decreasing order in which $L_i > L_{i+1}$ for a layered video composed of $L = L_1 \ldots L_l$.

Table 3: All possible permutations

Layers	Mapping/Permutation	AC3	AC2	AC1	AC0
3	map1	3			
	map2	2	1		
	map3	1	1	1	
5	map1	5			
	map2	4	1		
	map3	3	2		
	map4	3	1	1	
	map5	2	2	1	
	map6	2	1	1	1
8	map1	8			
	map2	6	2		
	map3	5	3		
	map4	4	4		
	map5	6	1	1	
	map6	5	2	1	
	map7	4	3	1	
	map8	4	2	2	
	map9	3	3	2	
	map10	5	1	1	1
	map11	4	2	1	1
	map12	3	3	1	1
	map13	3	2	2	1
	map14	2	2	2	2
10	map1	10			
	map2	9	1		
	map3	8	2		
	map4	7	3		
	map5	6	4		
	map6	5	5		
	map7	8	1	1	
	map8	7	2	1	
	map9	6	3	1	
	map10	6	2	2	
	map11	5	4	1	
	map12	5	3	2	
	map13	4	4	2	
	map14	4	3	3	
	map15	7	1	1	1
	map16	6	2	1	1
	map17	5	3	1	1
	map18	5	2	2	1
	map19	4	4	1	1
	map20	4	3	2	1
	map21	4	2	2	2
	map22	3	3	3	1
	map23	3	3	2	2

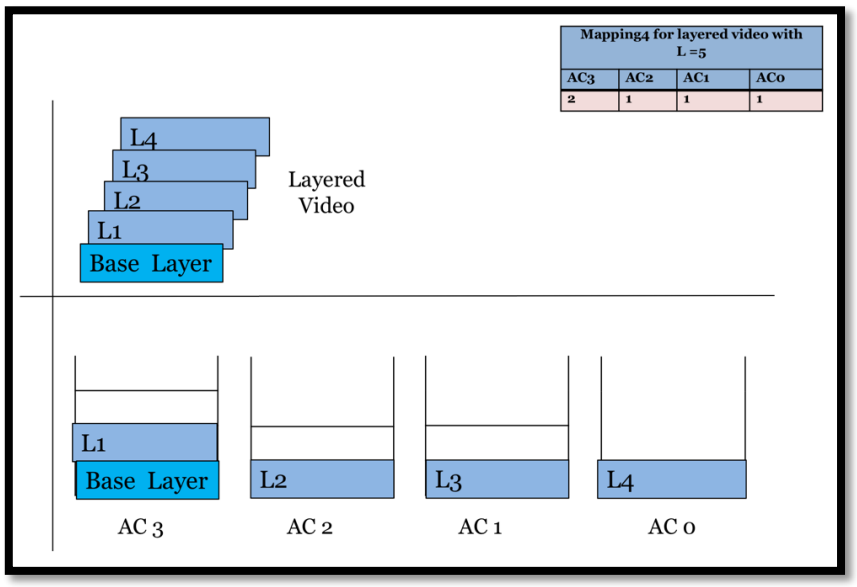


Figure 9: Canonical mapping [42]

The canonical mapping strategy can be defined as an arbitrary vector such that $MV_c(L,n) = \{x_1, x_2, x_3, \dots x_L\}$, where L layers are mapped into n ACs considering the following:

- **Ordered mapping**: the x_i values are always non-increasing, that is $x_i \geq x_j$; ∀ i ≥ j; and
- **Rate-based mapping**: the aggregate bit rate assigned to any AC where AC_i is greater than or equal to the aggregate bit rate assigned to AC_j, ∀ i ≥ j.

3.2.2 DSM Algorithm Formulation

An adaptive mapping algorithm used to map each video layer into channel resources represented, as queues existed in MAC layer, addresses the perceived quality of a multi-layer video. The performance of these mapped video layers is calculated using a new metric derived by them called expected number of reconstructed video layers (ERVL) used to maximize the average number of reconstructed video layers.

To leverage the resource allocation problem for video applications, a fully distributed algorithm called distributed sequential mapping (DSM) has been used [16]. This algorithm is based on EDCA access schemes (ACs) that offer differentiated access to the medium using different priorities for different traffic types. In addition, it presents a distributed and an adaptive cross-layer design that dynamically maps arriving video packets from the layered video into different ACs. Therefore, the QoS of wireless video transmission over IEEE 802.11 EDCA ACs can be optimized by maximizing the number of received video layers for each user running a sequential algorithm. Furthermore, this algorithm considers sequential mapping between ACs to make it more practical and scalable in terms of variant numbers of layers while requiring no synchronization between users. To illustrate this point, the complete scenario of DSM [16] considers two iterations:

- In the first iteration, initial arbitrary mapping is unnecessary be the optimal mapping must be set to all nodes, so every node in the network has to go through all possibilities over different ACs. Each node has to calculate the expected number of reconstructed video layers (ERVL) metric regarding its new mapping. Each possibility represents the user arbitrary mapping vector (MV) that maps L layers into n ACs, described as follows:

$$MV(L,n) = \{x_1, x_2, x_3, \dots, x_l, \dots, x_L\} \tag{3.6}$$

where L ≥ n and 1≤ x_l≤ n define the index of the AC layer l is mapped into it. The ERVL metric can be defined based on the individual dropping probability of each layer given that layer l_i can be reconstructed if and only if all layers $l_1$to l_{i-1} can be received successfully. Consequently, the ERVL metric can be calculated for all layers using the formula

$$ERVL\big(MV(L,n)\big) = \sum_{r=1}^{L} r * \prod_{i=1}^{r}(1 - Pr_{li}) * \ Pr_{li+1} \tag{3.7}$$

where Pr_{li} represents the dropping probability of layer l_i, and L is the maximum number of coded video layers. The Pr_{li} is computed according to which AC the layer l_i is

assigned to. Thus, all video layers mapped to the same AC have the same dropping probability, assuming dropping is statistically independent across all layers. Therefore, equation 3.7 can be written to represent the number of active ACs as:

$$ERVL\big(MV(L,n)\big) = ERVL\left(MV\big(L - y_{AC_n}, n-1\big)\right) * P_{n,drop}^{y_{AC_n}} + \prod_{h=1}^{n-1}\left(\left(1 - P_{h,drop}\right)^{y_{AC_h}}\right) * \left\{\sum_{i=1}^{y_{AC_n}} \left(i + Y_{AC_n-1}\right) * \left(1 - P_{n,drop}\right)^{i} * P_{n,drop}^{y_{AC_n} - i}\right\} \quad (3.8)$$

Where y_{AC_i} represents the number of layers assigned to AC_i in canonical mapping strategies. For instance y_{AC_1} represent the number of layers assigned to AC_1, y_{AC_2} represents the layers assigned to AC_2, in which $1 \leq y_{AC_1} \leq N$ and $\sum_{i=1}^{n} y_{AC_i} = L$. Therefore, canonical MV can be represented in term of ACs using another notation:

$$MV'^{(L,n)} = \{y_{AC_1}, y_{AC_2}, y_{AC_3}, \dots, \dots, y_{AC_n}\} \quad (3.9)$$

One can notice that the size of new MV' vector is of size n which represents the number of active ACs while the size of original MV is of size L which represents the number of video layers.

- For each node, the distributed sequential algorithm selects a specific mapping strategy while considering fixed mapping strategy for other users in the first iteration.
- After obtaining the optimal mapping for each node, the algorithm has to run through several iterations until the system converges to stable optimal mapping.

3.2.3 DSM Algorithm Features and Assumptions

The DSM algorithm considers that each node should work sequentially and in a distributed manner. As determined in section 2.1.4, ACs are common for all nodes. They represent different logical queues (AC_3, AC_2, AC_1, and AC_0), whereas they can be represented physically as one shared physical queue (AC) where packets should be buffered. Therefore, for these logical queues the video load is global such that all nodes share the same channel. To clarify, there is only one wireless channel, which is shared among all nodes. This wireless channel is divided into logical queues and the AIFS required to access the channel medium differs. Thus, AIFS represents the time each logical AC must wait before accessing an idle medium (i.e., AC_3 has the smallest AIFS value, thus higher priority for sending, so it will attempt to access and gain the channel first). If each node in the network is attempting to assign packets to this shared physical queue/AC simultaneously, the load for this AC changes each time without identifying which node is responsible for making these changes, causing calculation errors due to its poor performance. Therefore, the AP handles this task.

In addition, if there is no synchronization between the nodes, each node performs its permutation without informing other nodes. As a result, network behavior changes dramatically, making it impossible to determine the optimal solution correctly. Moreover, if the AP synchronizes every single permutation each time by every node where each node has to inform the AP about every single permutation, it would cause higher network resource waste and higher overhead. Thus, AP should handle only the task of informing all nodes about changes that have occurred in the network and signaling every single permutation sequentially in a distributed manner.

As the nodes are located at fixed positions, mobility is not considered in this model. This model cannot be applied in dynamic channel systems because if one of the mobile nodes

changes its location and becomes out of the area coverage from other nodes, all permutations must be performed again. Therefore, this model is less efficient for the mobile environment.

The next chapter contains a presentation of the implementation of the DSM algorithm and a discussion of the structure and settings of a node as well as the structure of the network and DSM mechanism are presented.

Chapter 4

4 Cross-Layer Design and Implementation

This chapter includes the implementation of the DSM scheme in the IEEE 802.11e EDCA module [16] using the NS2 simulation tool. For an improved understanding of this implementation and the resulting modifications to different OSI layers, the structure of a simulation node that will host this implementation is presented in the first subsection. In the subsequent subsections, the structure of the affected layers is illustrated; these effects were caused by the implementation of the DSM algorithm and the cross-layer design. In these sections, the modifications required to each of these layers to enable such implementation are also discussed. The researcher intends to make the implementation of the model available for public use for research purposes.

In the two subsections below, the basic components of wireless nodes and the basic functionality of EDCA module in MAC layer in the NS2 simulator are introduced to facilitate describing the implementation of this thesis work

4.1 Wireless Mobile Node

In NS2 [39], the wireless node is extended from the regular node. The difference between the regular node and wireless mobile node is that the wireless node can connect to the network without the need to use any physical wired channel. In addition, it supports the mobility of the node object. However, in this study, wireless nodes are fixed and not moving, as described in subsection 0.

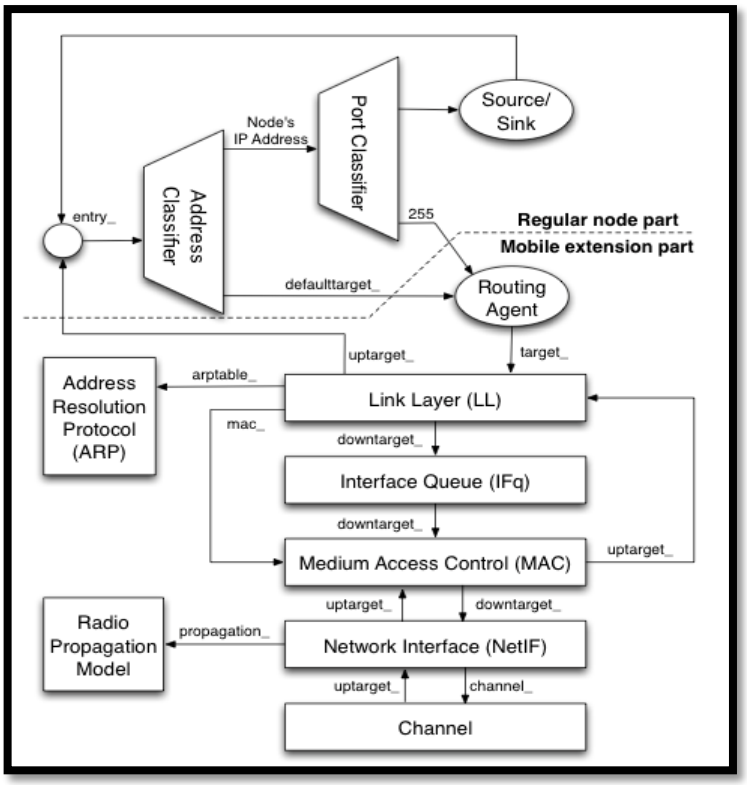


Figure 10: The architecture of a wireless mobile node [50]

As can be seen from Figure 10, the structure of a wireless mobile part [39], [50] is composed of

- A routing agent, responsible for forwarding the packet to the next hop node or its destination node (i.e., ad-hoc on-demand distance vector [AODV] routing protocol);
- A link layer (LL), responsible for managing packet framing and packet transmission time, delay, and bandwidth;
- An interface queue (IFq), responsible for buffer management;
- Address resolution protocol (ARP), responsible for translating the hardware address to the network IP address;

- MAC, responsible for interacting with the physical channel by forwarding packets downward to or upward from the medium channel after framing and de-framing the packet with a corresponding MAC address; and
- The network interface (NetIF), responsible for modulating the actual packet transmission, which works with a radio propagation model to simulate packet transmission error

The channel is shared among all nodes in the network and therefore is not a part of the mobile node; for any destination node, it picks the corresponding packet from the channel [39], [50].

4.2 Proposed Cross-Layer Design

According to NS2 default wireless node settings, each AC has a buffer capacity equivalent to 50 packets for queuing incoming packets. The adopted cross layer used in this study is composed mainly of three layers, namely the application layer, transport layer, and MAC layer. In addition, the interface queue exists in the data link layer, as shown in the yellow boxes in Figure 11.

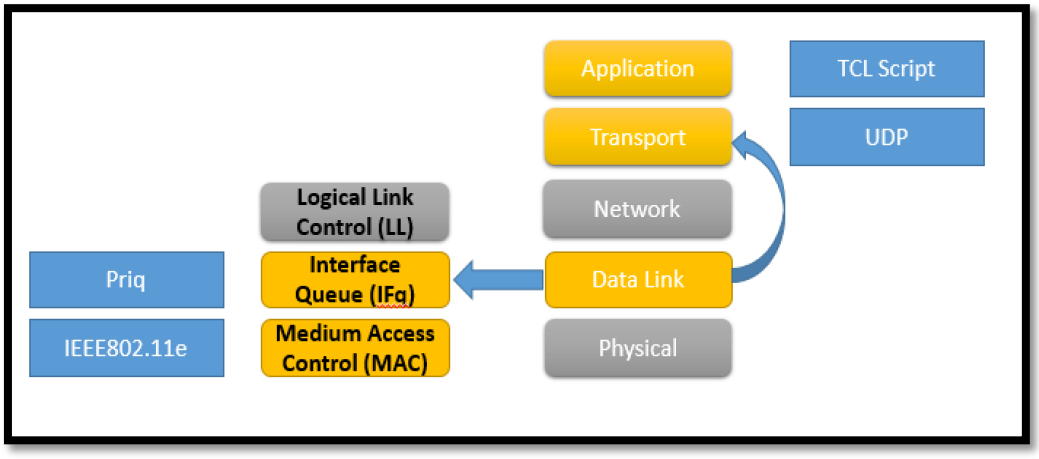


Figure 11: Cross-layer design in NS2

This cross layer design is needed to send information related to dropped packets from the MAC layer to the transport layer to calculate optimal ERVL based on varying channel conditions. Therefore, the cross layer used in this study can be classified as non-manager and a distributed cross-layer system, according to Fu et al. [36] (for more details, refer to section 2.1.5.1). Moreover, it can be classified as a top-down approach, according to [3], [37], and [38]. The functionality of each layer is discussed in detail in the next three subsections, and a summary for each layer is shown in Figure 12.

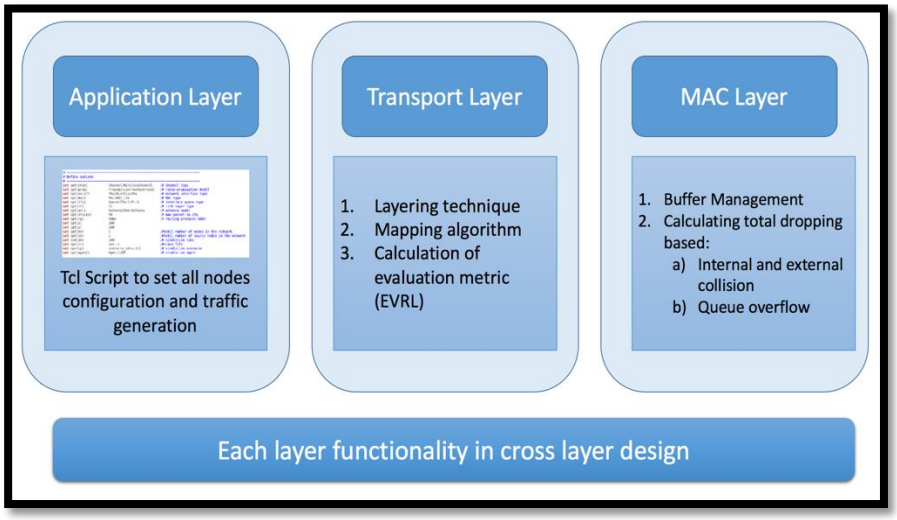


Figure 12: Cross-layer functionality

4.2.1 Application Layer

To run a simulation script, a TCL script should be created with an application traffic and mobility model. The TCL script sets up the wireless simulation components, including network component types such as channel type, antenna type, transmission range, radio propagation model, ad-hoc routing type, and interface queue (IFq) type. Furthermore, TCL script defines the total number of nodes and packets inside the queue as well as simulation time. Simulation time can be maintained and set from the application layer.

4.2.2 Transport Layer

The core work of this study has been done in the transport layer in the *UDPAgent()* class; because UDP guarantees low delay transmission and does not require packet acknowledgment for lost packets, it is effective when used with video traffic. Nevertheless, current UDP protocol alone does not consider varying video channel conditions or the significance of video information. Therefore, the DSM algorithm has been implemented in the *UDPAgent()* class to consider video information and varying channel conditions. The transport layer has two main functions, *UdpAgent::sendmsg(,,)* and *UdpAgent::recv(),* in which sendmsg() is responsible for sending packets continually with priority to lower layer protocols. If no priority is assigned to these packets from the transport layer, all packets belonging to video traffic have the same priority as assigned in TCL. Hence, for the DSM model, the *UdpAgent()* class handles three main processes in the *UdpAgent::sendmsg(,,)* function, summarized as labelling mechanism, mapping algorithm, and calculating ERVL value as a performance metric, as shown in Figure 12. More details about the implementation of the simulation model of the DSM algorithm are discussed in section 4.3.

4.2.2.1 Modeling the Video Layers

This subsection illustrates how the the video traffic has been structured into video layers. Initially, in NS2 all packets are ordered based on their sequence number (seqno_), as shown in Figure 13, which implies that they are ordered based on their layer number. UDP packets are generated with a fixed rate, and those packets are organized into layers depending on their sequence number. The layer the packet is assigned to can be determined according to the H.264 SVC encoder by using the formula:

$$Layer\ Number = sequence\ number\ of\ packet\ \%\ GOP\ Size$$

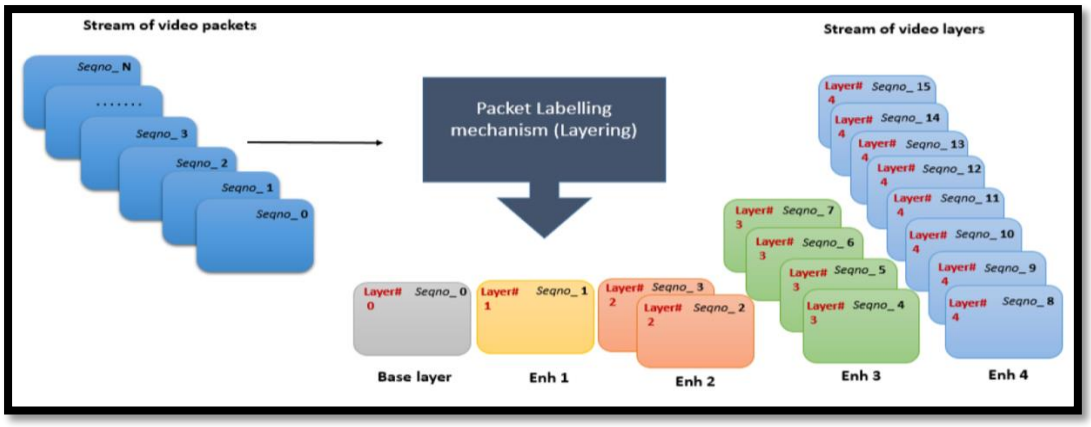


Figure 13: Layering mechanism for a layered video with five layers and a GOP contains 16 packets

To deliver H.264 scalable video coding quality, the video layers can be obtained based on GOP size which is 2^n, n = 0, 1, 2, 3… (i.e., 1, 2, 4, 8, 16, 32, and 64; default: 1 layer) must be a power of 2. Packets belonging to the base layer are presented by I-frames in SVC. Those corresponding to base layer (L0) have always a seqno_ equal to zero after performing the previous formula. For instance, to have 1 video layer with no enhancement layers (base layer only), GOP size should equal 1, which is 2^0; to have 2 layered video composed of base layer and 1 enhancement layer (Enh1), GOP size should equal 2, which is 2^1; to have 3 layers composed of base layer and 2 enhancement layers (i.e. Enh1 and Enh2), GOP should equal 4, which is 2^2. In case of 3 layered video, the first and second packets belong to the base layer and enhancement layer 1 (Enh1) respectively, while the last two packets belong to enhancement layer 2 (Enh2) and so on for any video with different layers. More illustrations regarding layering and the mapping algorithm appear in the next subsection.

4.2.2.2 Mapping Algorithm

An adaptive mapping algorithm is used to map each video layer into different ACs to address the perceived quality of a multi-layer video. Video layers are mapped into different ACs based on one of the assigned DSM canonical permutations as describe in section 3.2.1 and in section 3.2.2. Each one of the ACs has different priority number that varies between 0 – 3 depending on the EDCA parameter set that control the priority of transmission. Therefore, each layer is assigned a priority number based on the AC that is going to be mapped into it. For example, layers assigned priority number 3 indicate that they are mapped into AC_3 while layers assigned priority number 2 indicate that they are mapped into AC_2 and so on.

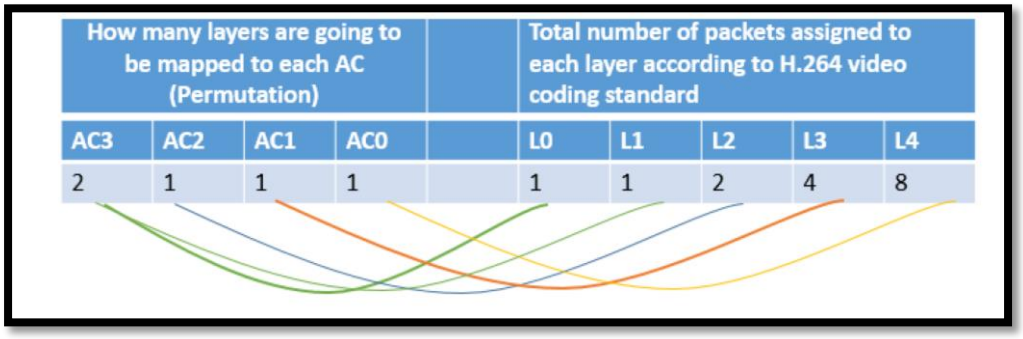


Figure 14: Distribution of packets across 4 ACs based on different mapping

Therefore, if there is one AC (i.e. AC_3) and a layered video composed of five layers in which a GOP contains 16 packets/frames, all layers of 16 packets are assigned priority number 3 and are mapped to AC_3 only. However, if there are four ACs and five layers to be considered, there will be one possible permutation over all these ACs (2, 1, 1, 1). As can be seen from Figure 14, all packets from layers (L0) and (L1) are assigned priority number 3 and will be mapped to AC_3, the highest priority AC, so the total number of packets will equal 2, as each one has only one packet to be mapped at a time to AC_3. While an L2 composed of 2 packets are assigned priority number 2 and will be mapped to AC_2, an L3 composed of 4

packets are assigned priority number 1 and will be mapped to AC_1, and the L4 composed of 8 packets are assigned priority number 0 and will be mapped to the last AC, which is AC_0.

Table 4: Distribution of 5 layers over different ACs based on 6 different mapping permutations

Perm	Total packets in AC_3	Total packets in AC_2	Total packets AC_1	Total packets in AC_0
5	16			
41	8	8		
32	4	12		
311	4	4	8	
221	2	6	8	
2111	2	2	4	8

Furthermore, if there are 2 ACs and 5 layers to be considered, there will be two different mappings (4, 1) and (3, 2) so that the higher number of layers 4 and 3 always will be assigned to the AC with the highest priority to be served first based on the method described in Figure 14 and Table 4. For additional details, all possible permutations considered for all video layers used in this study can be found in Table 3 in section 3.2.1. Those permutations referred as canonical mappings [16] were implemented using the twelvefold technique [51], which describes the problem of distributing n balls into m boxes—in this case, distributing L layers into n ACs.

4.2.2.3 Calculation of Evaluation Metric (ERVL)

Formula 3.8 represents a new metric to aid in the measurement of QoS for video delivery over wireless channel. It depends mainly on dropping probability counters, whether due to queue overflow or internal or external collision. It further depends on the total number of packets assigned to each AC. Dropping probability is proportional to the number of sent packets assigned to ACs, thus it is necessary to inform each node about the number of

packets sent, as only one node at a time is allowed to work on its all permutations and is allowed to change the traffic over the network. Having a maximum ERVL metric ensures improved QoS.

The metric ERVL has been calculated by [16] with the assumption that the dropping probability for each of the video layers mapped to AC_i are equal and identical to the dropping probability of the AC_i itself. In practice, this is an approximation because there is no guarantee that the packets dropped in AC_i belong to the layers mapped to this AC. In the worst-case scenario, packets in AC_i could have belonged to only one of the layers mapped to AC_i. In this study, calculation of ERVL is exact and measured based on readings of dropping probabilities per layer rather than AC dropping probability. The experiments in this research work are not constrained by restricting the size of the layers to be equal. SVC [20] is used in these experiments, which uses various sizes of video layers transmitted with different rates. Similar research work in this area considered variable sizes and rates of video layers [15], [21], [37], [40]-[43] and [52].

4.2.3 MAC Layer

The MAC layer represents a set of functions that allows nodes to access the wireless medium in a certain way. The MAC layer handles two main processes for the DSM model, namely ACs/buffer management and internal and external collision.

4.2.3.1 ACs/Buffer Management

Each layer is mapped to a specific AC depending on the priority field assigned previously in the transport layer using the permutation-mapping technique presented in Table 3 in section 3.2.1. If more than one AC is to be considered, the AC with the highest priority must gain the medium first, and layers inside each buffer/AC are served based on the FIFO scheduling mechanism. Thus, the first packet buffered inside this AC has the highest priority to be served first. For example, if there is a video composed of 5 layers (L is from 0 to 4), and

2 ACs, the possible ways to transfer this video information equal (4, 1) and (3, 2). In case of (4, 1) permutation, the layers that have priority equal to 0, 1, 2, and 3 will be mapped to access category AC_3, whereas layers with priority equal to 4 will be mapped to AC_2 in the MAC layer. After mapping layers into corresponding ACs, the AC may become full if the number of packets inside it exceeds the queue limit, which equals 50 packets per queue. If the AC becomes full, the incoming packet is dropped based on queue overflow. For each AC, the total and dropped packets per AC are stored in counters called totalPktsACNodesPri and QdropNodePri respectively, which are transferred to the UDP transport protocol to be used for calculating the ERVL metric.

4.2.3.2 Internal and External Collision

Packets that have been dropped due to internal or external collision are recorded in dedicated counters called macdropPerNodes and are transferred to the UDP agent to be used for calculating the ERVL metric as well. (For additional details about internal and external collision, please refer back to section 2.1.4.)

4.2.4 IEE802.11e MAC layer: EDCA Module Functionalities in NS2

The MAC layer defines the communication rules used to access the medium shared between all nodes. In NS2, the standard IEEE 802.11 MAC layer implementation is provided. However, to use IEEE 802.11e in the NS2 simulator, the MAC layer has been extended by the Telecommunication Networks Group at the Technical University of Berlin (TKN) [53] to provide all EDCA functionalities and mechanisms mentioned in section 2.1.4 to researchers. All underlying functionalities between the extended MAC layer and all other layers in the OSI protocol stack have been preserved and maintained. According to the TKN Telecommunication Networks Group [53], the implementation of the EDCA module is open source; therefore, it has been used in this study to implement the cross-layer design and the

DSM [16] algorithm based on ACs existing in the IEEE 802.11e MAC layer. Generally, NS2 implements a mac() class working as an abstract or as a parent for all other MAC types, thus the functionalities of this abstract class has been overridden by inherited classes such as IEEE 802.11 and IEEE 802.11e.

4.2.4.1 MAC802.11e() class

Because the MAC layer located in class *Mac802.11e()* includes all functionalities for communication between nodes, buffering, and timing issues, this sub-section offers a description of how the communication between sender and receiver takes place in the MAC layer. The flow control of sending a data packet can be summarized as follows [39], [54]:

- Upper object (i.e., queue) calls recv(), which calls send().
- send() calls sendData(packet), which sets the data packet using the pktTx_ object and calls sendRTS(), which sets the defer timer using the mhDefer_ object with a delay DIFS and a generated random CW slot.
- When the defer timer expires, it calls deferHandler(), which calls check_pktRTS(), which calls transmit(), responsible for sending RTS packets.
- If the receiver, which is in an idle state, receives an RTS packet, it sends back a CTS packet and waits for data. Alternatively, the sender keeps retransmitting the RTS packet through RetransmitRTS() if send time expires.
- If the CTS packet is received by the sender using recvCTS(), it deletes the pktRTS_ and calls tx_resume(), which sends the data packet using pktTx_.
- After successful transmission of the data packet, the sender waits for an acknowledgment using recvACK() before returning to an idle state. If no acknowledgment is received, it sends the RTS packet again using RetransmitRTS() until it reaches the retry counter limit, which equals 7 times. After that, it drops the pktTx_, resets the retry counter and CW time, and starts another backoff timer.

In contrast, the control flow of receiving a packet can be summarized as follows:

- The lower object (i.e., NetIF) calls recv(), which sets the pktRx_ to received packet and calls txtime() to set the receiver time using mhRecv_.
- Once the receiver time expires and an RTS packet is received via recvRTS(), it calls sendCTS(), which calls recv_timer().
- The recv_timer() function checks the received packet type held in pktRx_ and calls recvDATA(), which calls sendACK() to send an acknowledgment if the data packet was received correctly. If the packet was not received, it returns to an idle state until it receives another RTS packet.

4.2.4.2 Queueing/Buffering

The *Queue()* class in NS2 models the buffering mechanism in a node that stores the incoming packets from upper layers in buffer and then forwards them to lower layers. By default, the buffer size is set to 50 packets using variable qlim_ inside *Queue()* class in each wireless node [39]. Similar to *mac()* class, the *Queue()* works as an abstract class for all inherited classes, for example, PacketQueue(), which models main operations of buffering such as enqueueing using enqueue(packet) and dequeueing using dequeue(). Additional queue types are derived from the base *Queue()* class, including drop-tail, priority queue, fair queuing, deficit round robin, random early detection, and class-based queue objects [39].

The *DropTail()* class overrides the enqueue(packet) and dequeue() functions of the *Queue()* class, maintains a single FIFO scheduling queue, and allows drop-on-overflow buffer management where the packets are dropped from the tail rather than the head of the buffer [39]. The enqueue(packet) function in the *DropTail()* class first checks whether an empty space is available. If there is an empty space (the number of packets inside the buffer is less than 50), it enqueues packet (p) inside its buffer labelled q_. Otherwise, it drops the incoming packet from the tail using the drop(packet) function if the limit is exceeded in the

buffer size and the packet will cause a buffer overflow [39]. The dequeue() function implements a simple FIFO scheduling mechanism that allows the first buffered packet in the queue to be served and sent first to the medium [39]. The priority queue, or *Priqueue()* class, prioritizes routing packets, with packets with these payload types having the highest priority: PT_DSR, PT_TORA, PT_AODV, PT_AOMDV, and PT_MDART. This class is derived from the *DropTail()* class, but it enqueues packets based on their routing priority. This queue is called the interface queue (IFq), as the queue is installed in each wireless physical interface.

As described in section 2.1.4, the EDCA model maintains four queues called access categories (ACs) to support QoS, and each queue has a different priority (i.e., 0 to 3), where AC_3 has the highest priority for gaining the channel medium. The default *prioqueue()* class implements only one queue, while in EDCA four queues are implemented inside *priq()*, which is derived from the *d-tail()* class. The priority field is identified in each packet header using prio_ field where each queue has one priority from 0 to 3, as shown in Figure 15.

priority.tcl

```
# 802.11b parameters (default EDCA parameter set), aCWmin=31, aCWmax=1023
proc priority { ifq_name } {
   upvar $ifq_name ifq

    #parameters for Queue 0
    $ifq Prio 0 PF 2
    $ifq Prio 0 AIFS 7
    $ifq Prio 0 CW_MIN 31          ;# aCWmin
    $ifq Prio 0 CW_MAX 1023        ;# aCWmax
    $ifq Prio 0 TXOPLimit 0

    #parameters for Queue 1
    $ifq Prio 1 PF 2
    $ifq Prio 1 AIFS 3
    $ifq Prio 1 CW_MIN 31          ;# aCWmin
    $ifq Prio 1 CW_MAX 1023        ;# aCWmax
    $ifq Prio 1 TXOPLimit 0

    #parameters for Queue 2
    $ifq Prio 2 PF 2
    $ifq Prio 2 AIFS 2
    $ifq Prio 2 CW_MIN 15           ;# (aCWmin+1)/2 - 1
    $ifq Prio 2 CW_MAX 31           ;# aCWmin
    $ifq Prio 2 TXOPLimit 0.006016

    # parameters for Queue 3
    $ifq Prio 3 PF 2
    $ifq Prio 3 AIFS 2
    $ifq Prio 3 CW_MIN 7            ;# (aCWmin+1)/4 - 1
    $ifq Prio 3 CW_MAX 15           ;# (aCWmin+1)/2 - 1
    $ifq Prio 3 TXOPLimit 0.003264
```

Figure 15: Default EDCA parameter set

4.3 The Simulation of the DSM Algorithm

This section contains a description about how the network model of the DSM algorithm was implemented in NS2 (additional detail about the DSM algorithm was illustrated in Chapter 3). Further, the main DSM algorithm features and assumptions are examined.

Referring to Table 3 in section 3.2.1, if there are 5 layers, 5 nodes, and 1 AP, then, considering all possible canonical mapping strategies, one guarantees that maximum ERVL can be obtained from one of different mappings equal to 6 distributed over 4 ACs. Such that each mapping is assigned specific window time equal to **m** seconds, as shown in Figure 16. Every node is sending its video content in turn, which means multiple video streams are to be sent where the channel is allocated to all nodes. The aim for each node is to find the most effective way to adjust the best permutation or mapping strategy to drop the least number of packets, thus obtaining the maximum ERVL value.

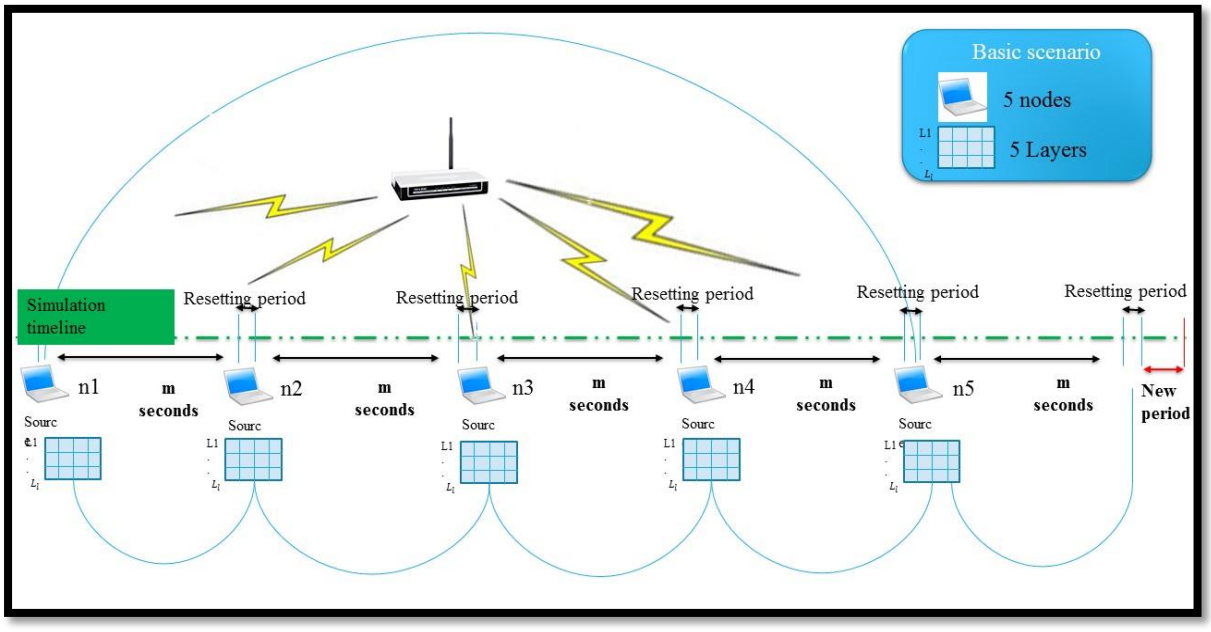


Figure 16: Basic network topology

At the outset, all nodes have to work on an initial mapping where each node in the network is unaware of the other nodes. Hence, to begin a new experiment with each node,

this node should inform the AP about its current mapping until it finishes all six assigned mapping techniques. Then the AP prompts the next node to begin working on its mapping strategies. For example, all nodes in Figure 16 have to work for 0.5 seconds on their initial mapping, and then the AP has to signal all nodes sequentially and in turns. Therefore, it begins by signaling the first node to start working on its six mapping techniques, while other nodes should complete working on their initial mapping. This node is considered the source, and it sends its video stream by performing all assigned mapping strategies in turn. Based on dropping counters and the total number of packets recorded by the destination node, those counters are sent back to the source node to calculate the ERVL value for each mapping, as the source node is responsible for the layering and mapping algorithm, as shown in Figure 17.

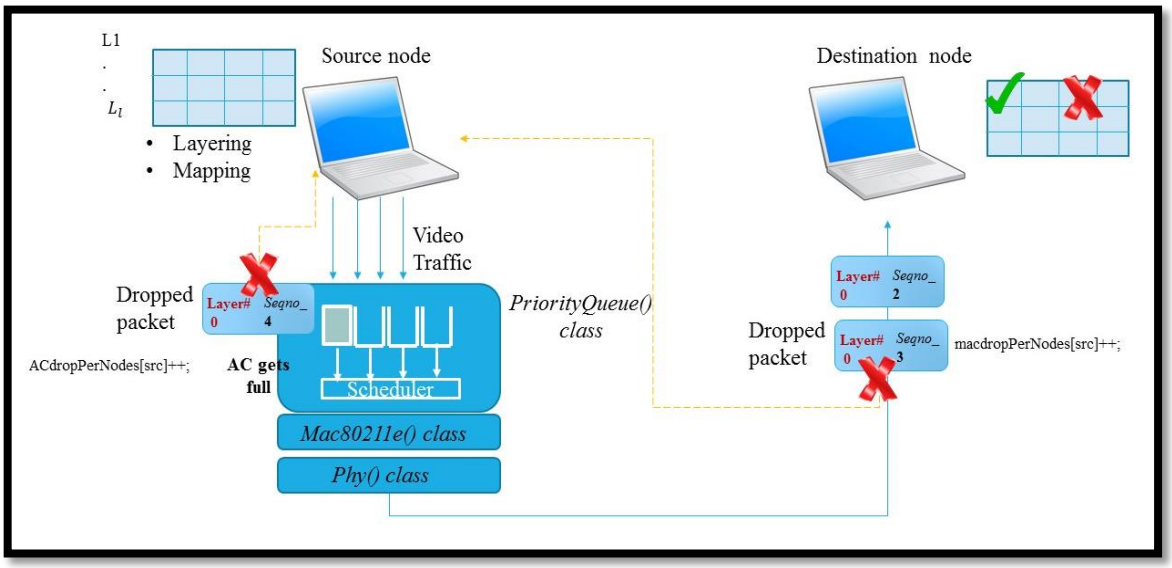


Figure 17: Communication between any two nodes in network topology

After obtaining the best ERVL value from the six mapping techniques completed by the first node, all experiments and queues/ACs with all dedicated counters should be reset in a window time equal to 0.5 seconds to begin a new experiment in the second node. Hence,

the second node should also begin working on its six mapping techniques, and the first node has to keep working on the mapping that guarantees the best ERVL rather than initial mapping, while others (i.e., node 3, node 4, and node 5) should keep working on their initial mapping technique. Once the second node finishes working on its mapping and obtains the maximum ERVL value, the experiments, counters, and queues are reset, and all other nodes continue to work on in the same way as node 1 and node 2. After all nodes obtain their best ERVL values, all of them should work for a second turn to obtain the optimal mapping until the ERVL converges to a certain value. As can be seen from the above scenario, the AP is responsible for synchronizing and defining turns between nodes (i.e., which node should work during which time slot).

The implementation steps that show the entire functionality for DSM in each node are provided below and illustrated in Figure 18.

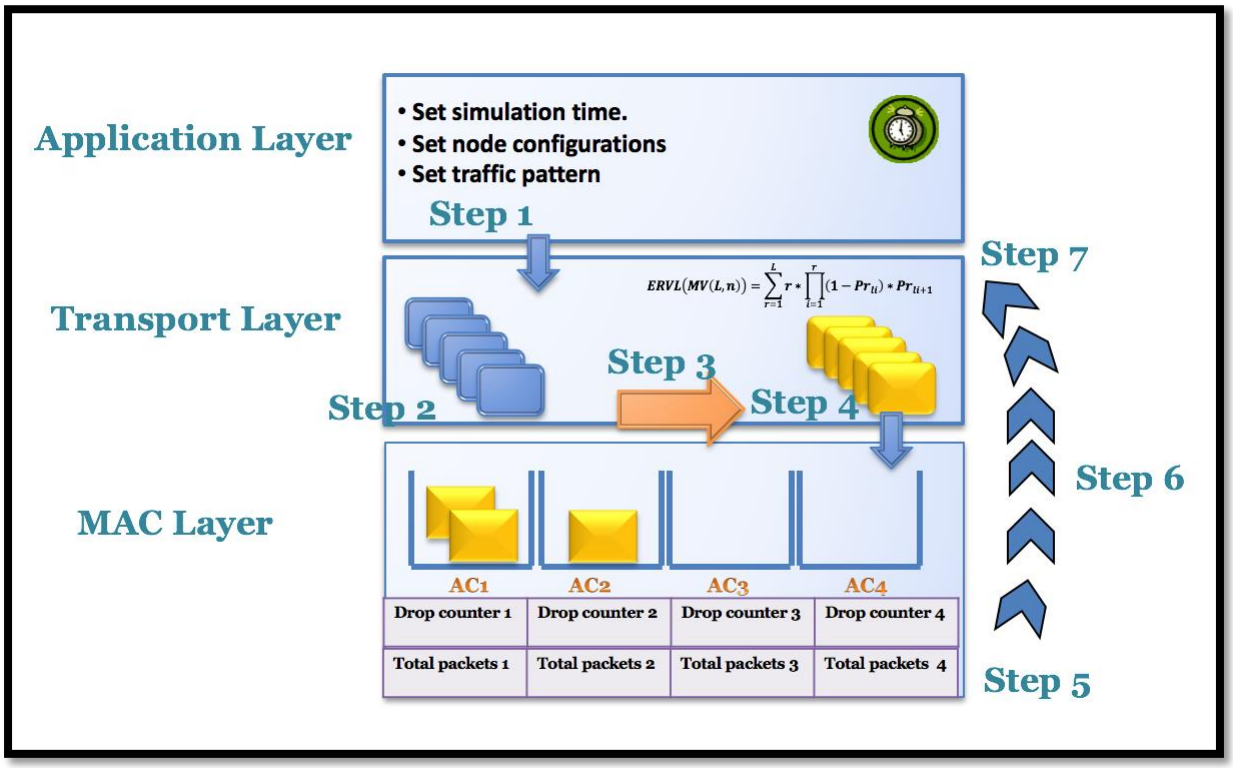


Figure 18: Experimental design of the cross-layer scenario

Step 1: Configure network parameters such as simulation time, number of nodes, routing type, and other parameters. This is achieved using TCL scripting language and by running different wireless scenarios.

Step 2: Implement the packet labelling mechanism to assign each packet a corresponding layer number, where each layer is composed of several packets.

Step 3: Implement the calculation of the possible ordered mapping strategies (permutations) based on the twelvefold technique described by Stanley in [51].

Step 4: Implement the mapping mechanism depending on each mapping strategy where each layer is going to be assigned a number belongs to one of the AC's priority numbers that is going to be assigned into it. Each layer is mapped to the proper AC based on the current ordered mapping strategy implemented in step 3.

Step 5: Implement where and when the necessary parameters, drop counters, and total packets in the IEEE 802.11e MAC layer will be calculated. These calculations are used later to calculate the average number of perceived layers (ERVL).

Step 6: Send the parameters calculated in Step 5 to the transport layer, UdpAgent class, using the cross-layer design.

Step 7: Finally, ERVL metric has been calculated based on the formula 3.8 proposed in [16], [17].

The next chapter contains a presentation of the experiments conducted to evaluate the proposed DSM strategy. The details of simulation settings as well as the purpose of each experiment and its findings are discussed.

5 Performance Evaluation

The purpose of this chapter is to present the findings and the discussions on the results obtained by evaluating DSM strategy using various simulation experiments of the implemented strategy.

5.1 Simulation Setup

The experiments in this research work are conducted using network environment (ns2 version 2.8). The simulation utilizes the network topology in Figure 16 and uses UDP transport protocol and AODV network routing protocol in all of the studied experiments. The detailed settings of simulation's parameters are shown in Figure 19 below.

```
# ======================================================================
# Define options
# ======================================================================
set opt(chan)           Channel/WirelessChannel    ;# channel type
set opt(prop)           Propagation/TwoRayGround   ;# radio-propagation model
set opt(netif)          Phy/WirelessPhy            ;# network interface type
set opt(mac)            Mac/802_11e                ;# MAC type
set opt(ifq)            Queue/DTail/PriQ           ;# interface queue type
set opt(ll)             LL                         ;# link layer type
set opt(ant)            Antenna/OmniAntenna        ;# antenna model
set opt(ifqlen)         50                         ;# max packet in ifq
set opt(rp)             AODV                       ;# routing protocol DSDV
set opt(x)              100
set opt(y)              100
set opt(nn)             11                         ;#total number of nodes in the network
set opt(sn)             1                          ;#total number of source nodes in the network
set opt(nn2)            10                         ;#total number of nodes in the network
set endtime             4700                       ;# simulation time
set opt(tr)             /dev/null                  ; #out.tr  ;#trace file
set opt(cp)             scenario_edca.tcl          ;# simulation scenario
```

Figure 19: Simulation parameters

Two main sets of experiments are considered (Set A and Set B). In each set, different scenarios are obtained by measuring the effect on nodes for different number of layers or visa versa. In Set A experiments, no additional traffic is introduced other than the DSM tested video sessions between nodes. While in Set B, five different composite traffic patterns (5 Groups) are introduced in three different load levels (light "**SL**", intermediate "**SI**" and heavy

"**SH**") hence, having 15 different experimental scenarios repeated for various combination of (number of nodes, number of layers) in Set B. The number of nodes vary as (N=5, 10, 15, and 25) while the number of layers vary as (L=3, 5, 8 and 10). Every individual experiment is repeated 50 times reporting averaged results to enhance the results' confidence interval.

Table 5: Background scenarios

	Scenario Name	**AC_3**	**AC_2**	**AC_1**	**AC_0**
Group 1	SL1	5 Voice	1 Video		
	SI1	10 Voice	1 Video		
	SH1	20 Voice	1 Video		
Group 2	SL2	1 Voice	1 Video	5 HTTP	
	SI2	1 Voice	1 Video	10 HTTP	
	SH2	1 Voice	1 Video	20 HTTP	
Group 3	SL3	1 Voice	1 Video	1 HTTP	5 FTP
	SI3	1 Voice	1 Video	1 HTTP	10 FTP
	SH3	1 Voice	1 Video	1 HTTP	20 FTP
Group 4	SL4	5 Voice	1 Video	5 HTTP	5 FTP
	SI4	10 Voice	1 Video	10 HTTP	10 FTP
	SH4	20 Voice	1 Video	20 HTTP	20 FTP
Group 5	SL5	1 Voice	1 Video	1 HTTP	1 FTP
	SI5	1 Voice	2 Video	1 HTTP	1 FTP
	SH5	1 Voice	3 Video	1 HTTP	1 FTP

The composite traffic patterns are formed using Voice, Video, HTTP and FTP traffics generated by traffic models supported by NS2 such as Exponential, Pareto, CBR and Trace

traffic patterns [55]. In these experiments foreground traffic refers to the data that needed to be examined while background traffic refers to the composite traffic patterns with light, intermediate, and heavy loads as shown in Table 5. Foreground traffic can be extremely affected by background traffic since both traffics compete to utilize available network resources thus background traffic has an important impact over foreground traffic.

The foreground traffic which represents the multi-layer video is a Constant Bit Rate (CBR) flow generating packets at constant bit rate. The background traffic is a composite traffic of exponential voice traffic, CBR video traffic, Pareto HTTP traffic, and best effort FTP traffic. Voice traffic has been simulated with the aid of an exponential ON/OFF distribution protocol built in NS2 simulator according to a standard PCM codec [56] with mean ON equals to 350 seconds (talk spur), and mean OFF equals to 640ms over UDP protocol. All set of parameters can be found in Figure 20.

```
set Voice [new Application/Traffic/Exponential]

$Voice set packetSize_   84
$Voice set burst_time_   350s
$Voice set idle_time_    640ms
$Voice set rate_         64k
$Voice set random_       0
$Voice set interval_     0.05
$Voice use-rng $rng
```

Figure 20: Voice traffic represented by exponential distribution protocol

While Video traffic has been simulated with the aid of CBR application over UDP protocol which guarantees synchronized time between the sender and the receiver. CBR is designed for any type of data that requires real time variation like Voice and Video

Conferencing. More details about the set of parameters used to construct CBR traffic is shown in Figure 21.

```
set udp [new Agent/UDP]
$ns_ attach-agent $src $udp

set Video [new Application/Traffic/CBR]

$Video set packetSize_  2000
$Video set random_      0
$Video set interval_    0.002
$Video set prio_        2
$Video set fid_         0
$Video attach-agent $udp

set sink [new Agent/UDP]
$ns_ attach-agent $dest $sink
```

Figure 21: CBR application traffic over UDP protocol

HTTP traffic has been simulated with the aid of Pareto distribution protocol according to [57] with mean ON time is set to 350ms and the mean OFF time is set to 650ms over UDP source as in Figure 22.

```
set HTTP [new Application/Traffic/Pareto]

$HTTP set packetSize_   280
$HTTP set burst_time_   350ms
$HTTP set idle_time_    650ms
$HTTP set rate_         400k
$HTTP set random_       0
$HTTP set interval_     0.02
$HTTP set shape_ 1.11
$HTTP use-rng $rng
```

Figure 22: HTTP traffic represented by Pareto distribution protocol

Finally FTP traffic has been simulated over TCP which has the following set of parameters according to [56] as shown in Figure 23.

```
set tcp [new Agent/TCP]
set snk [new Agent/TCPSink]

$tcp set packetSize_     512
$tcp set window_         20
$tcp set prio_           0
$ns_ attach-agent $src $tcp
$ns_ attach-agent $dest $snk

set ftp [new Source/FTP]
$ftp set agent_ $tcp
```

Figure 23: FTP application traffic over TCP protocol

Lastly, the values of the parameters CWmin, CWmax, AIFS and TXOP of each AC are the default EDCA parameter in IEEE802.11e shown in Table 1 and the maximum queue size for each AC is 50 packets.

5.2 Performance Metrics

The performance metrics to be considered can be listed as follows:

- **Packet loss rate:** This metric can be defined as the percent of packets lost with respect to total number of packets sent.
- **ERVL metric:** This metric guarantees the significance of video information; thus it considers the dependency between video layers. It depends on the packet drop rate such that whenever the packet loss rate is high, the ERVL value is low. Thus degrading video quality.

5.3 Experiments' Sets on Evaluating DSM Algorithm

In addition to Set A and Set B experiments, a number of initial experiments are simulated referred to as initial set. The purpose of the experiments in the initial sets is to find out the best packet size and data rate that can be used for transmitting video traffic over a

network topology as described in section 4.3. Details of these three sets of experiments are discussed in the subsequent subsections. The reported results of each individual instance of an experiment are averaged over 50 repetitions of the same instance of that individual experiment.

5.3.1 Initial Set

For these initial experiments, a video traffic composed of five layers has been examined over five nodes. The first set of initial experiments test the ERVL performance metric that is calculated per each mapping across various packet sizes composed of S = 800, 1200, 1600, 2000, 2400 and 2800. As shown in Figure 24, ERVL is not affected by the packet size but it is affected by the mappings or permutations that are assigned to each node. It is clear that map6 gives the best results among all packet sizes but the maximum ERVL can be obtained when packet size = 800 and 2000. Therefore, the adopted packets size for this study is 2000 bytes since video data payload usually contains big data sizes.

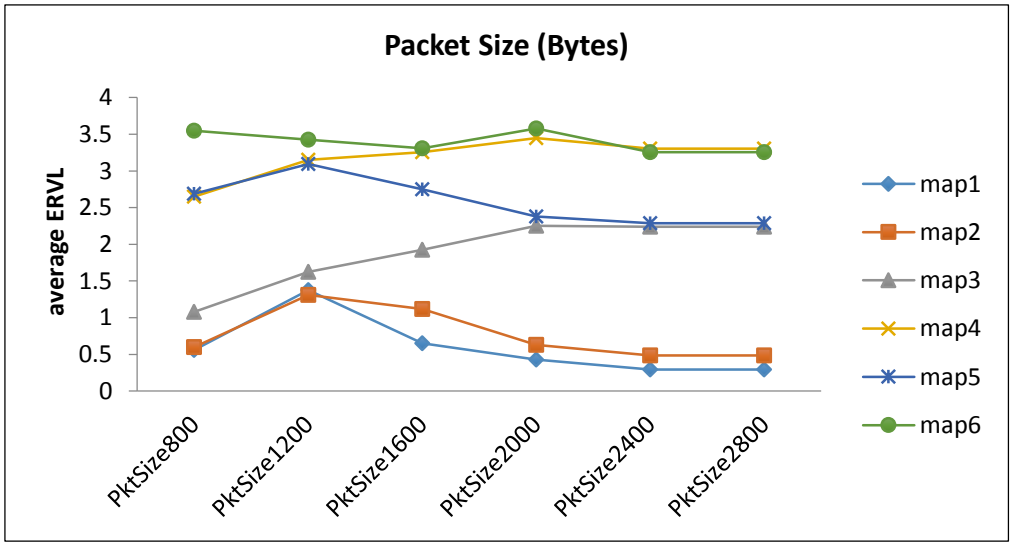


Figure 24: ERVL for different packet size Vs. different mapping over 5 nodes

The adopted data rate of the wireless link has been examined with different values to obtain optimal data rate to transfer HD video quality (i.e. H.264, MPEG 4…) over WLAN in the second set of initial experiments.

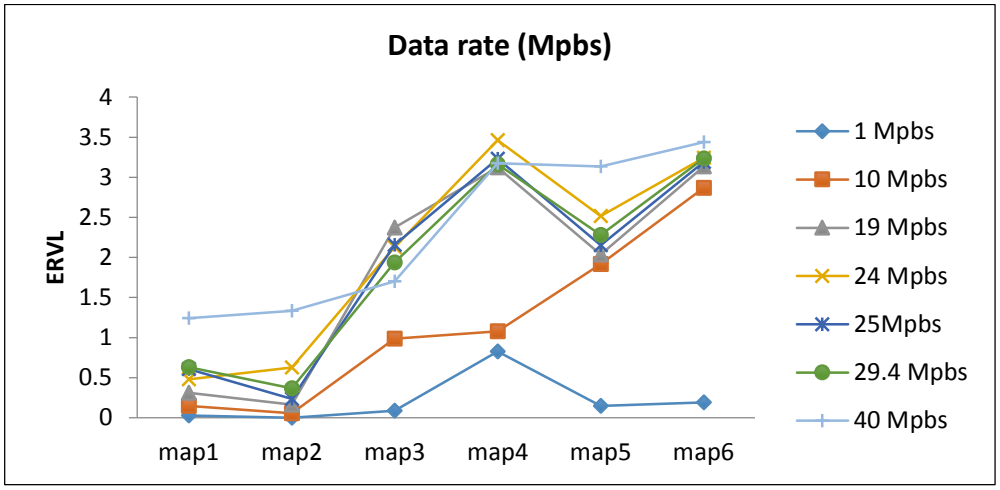


Figure 25: Data rate Vs. different mapping over 5 nodes

Based on Figure 25, using high data rates to transfer video traffic guarantees best video quality compared to lower data rate. Using 24 Mpbs and 40 Mpbs obtain maximum ERVL value across map4, map5 and map6. Therefore, 24 Mpbs data rate have been considered for all experiments to transfer a video traffic over wireless network and the basic rate used transferring control packet (i.e. ACK, RTC, CTS…) is 36 Mbps.

5.3.2 Experiment Set A: Initial Evaluation - Utilizing main network resources for foreground traffic (video traffic) only

The corresponding subsections report the performance of the cross layer design for H.264 structured video delivery with different set of video layers (L=3, 5, 8 and 10), with no other ongoing traffics to investigate optimal solution over the desired network topology while varying the number of nodes (N=5, 10, 15 and 25). This video traffic is allocated to diverse ACs based on its permutation. Since all the nodes have been examined with the same set of

parameters and without any type of congestion, they have achieved the same results. In this scenario, the loss rate for each layer, AC and ERVL values have been analyzed.

5.3.2.1 Scenario 1: Impact of Different Number of Nodes on Video Layers

The network performance metric, i.e. packet loss rate for mapping 5 layers over 6 different permutations per node is shown in Figure 26. For example, map1 represents mapping all video layers to AC_3, map2 and map3 represent mapping layers across two ACs which are AC_3 and AC_2 in which map2 has (4,1) permutation and map3 has (3,2) permutation according to Table 3. Since base layer frames are the most important video frames to decode higher video layers' frames, they have been always mapped to highest priority queue AC_3 as mentioned previously, thus the number of lost frames must end with zero drop frames.

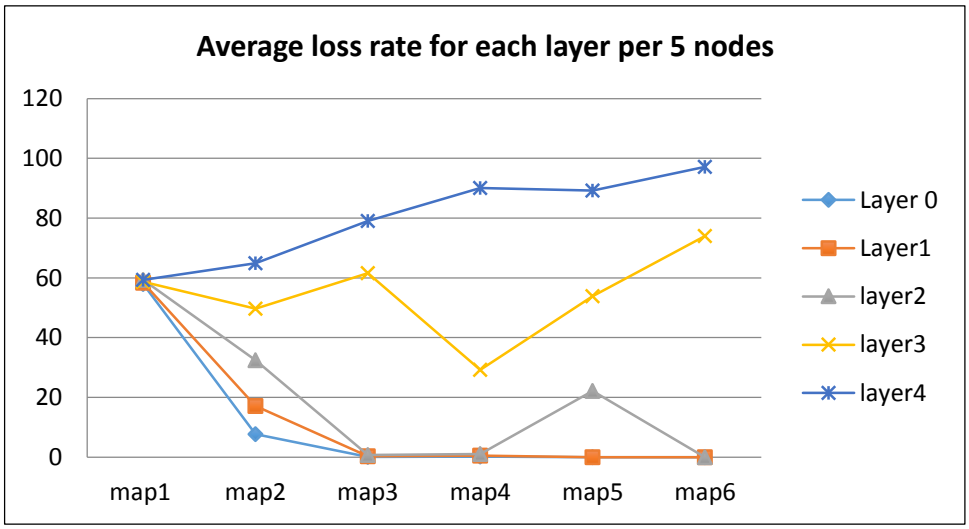


Figure 26: Loss rate for all 5 layers per 5 nodes

Table 6: Packet loss rate for each layer (%) in each mapping

	L0	L1	L2	L3	L4	Total loss rate
map1	48.95	50.38	53.10	49.56	54.68	52.58
map2	6.97	13.86	27.46	46.64	63.48	48.14
map3	0.00	0.10	0.50	53.96	75.72	51.44
map4	0.39	0.59	1.77	29.86	90.77	53.13
map5	0.00	0.00	23.49	61.34	91.20	63.87
map6	0.00	0.00	0.00	74.65	97.55	67.41

It is clearly seen from Table 6, when the DSM algorithm started with no adaption in which all layers are mapped to AC_3, half of the packets have been dropped in each layer since all of them will be mapped to the same AC causing a queue overflow. However, when DSM algorithm used the three alternate priority queues AC_2, AC_1 and AC_0 and started to adapt video layers between them, resulting in a lower packet loss among all of them. In addition, map6 guarantees zero dropped frames across L0, L1 and L2 since they will be mapped to AC_3 and AC_2 and will not contend with other ACs to access the medium. In contrast, L3 and L4 packets will be always dropped because they are mapped to lowest priority ACs in every permutation or mapping.

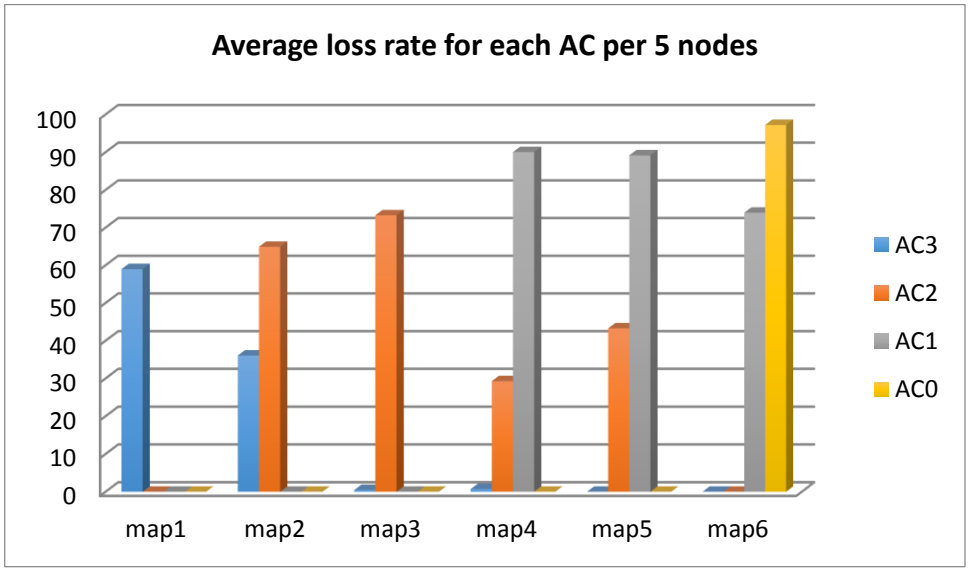


Figure 27: Loss rate for 4 ACs per 5 nodes

Table 7: Loss rate for each AC (%) per each mapping across 5 nodes

	AC_3	AC_2	AC_1	AC_0
map1	52.58	0.00	0.00	0.00
map2	32.79	63.48	0.00	0.00
map3	0.28	68.47	0.00	0.00
map4	1.13	29.87	90.77	0.00
map5	0.00	48.72	91.20	0.00
map6	0.00	0.00	74.64	97.55

The bar chart shown in Figure 27 demonstrates the loss rate for video layers mapped in each AC within the same node. When all video layers are mapped to the same AC, (i.e. AC3) as in "map1", which has a limited size equals to 50, half of the video layers' packets will be dropped regardless of its importance yielding high loss rate in all layers. In this case, since there was no ongoing traffic on other ACs, all ACs (i.e. AC2, AC1 and AC0) have zero packet loss as shown in Table 7.

Conversely, when these layers are distributed over four ACs, the loss rate in highest priority queues AC3 and AC2 started to decrease until it reaches zero as in "map6" while most of layer's packets delivered over AC1 and AC0 have been dropped as shown in Table 7 since they are going to be mapped to lower priority ACs.

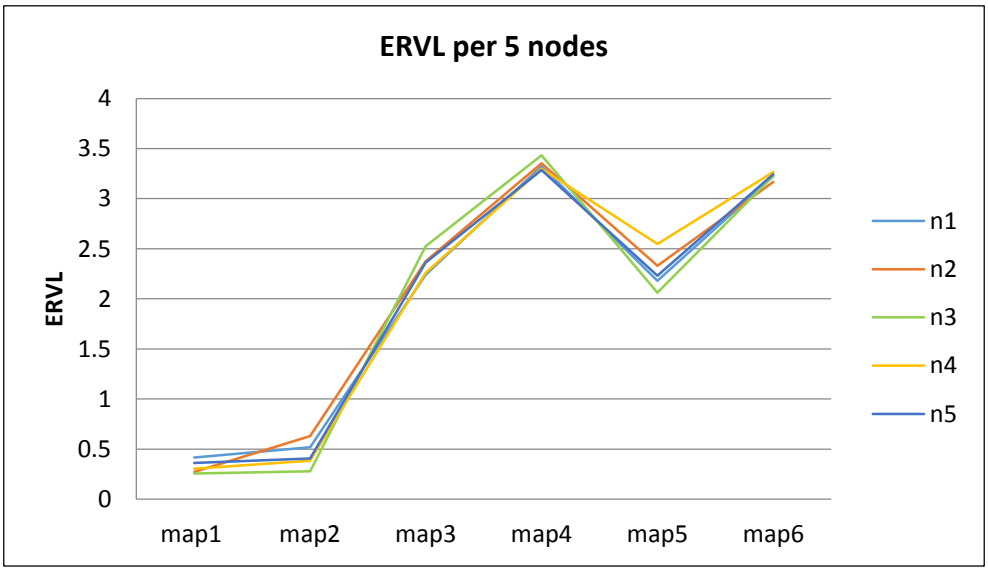


Figure 28: ERVL for different mapping for 5 layers per 5 nodes

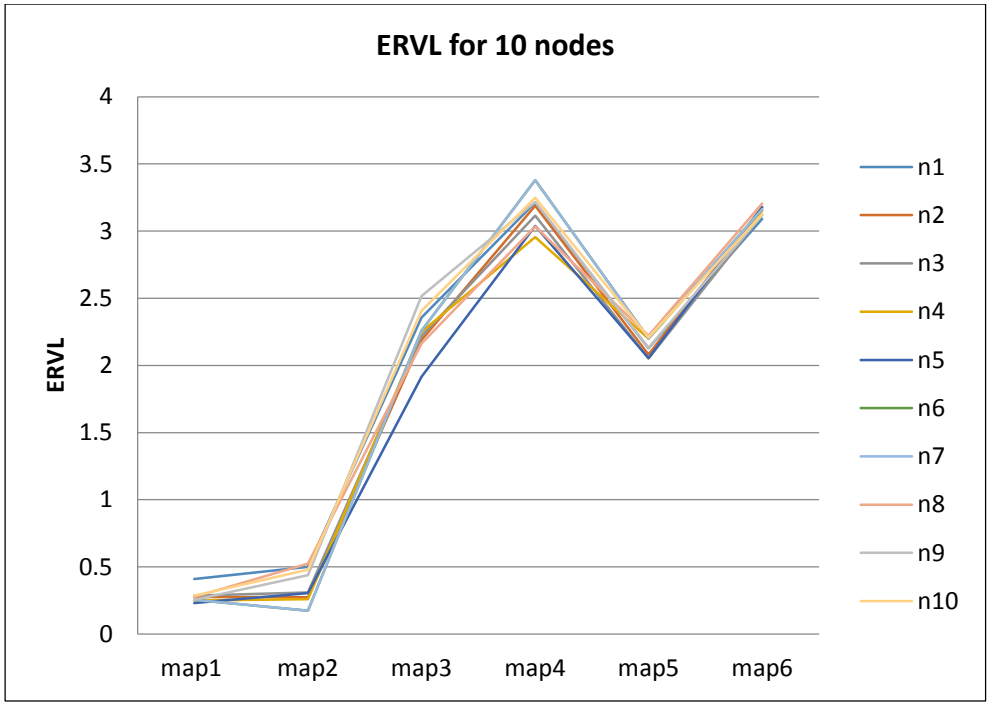


Figure 29: ERVL for different mapping for 5 layers per 10 nodes

Figure 28 and Figure 29 show the results for the ERVL performance metric, for 5 layers with 6 different mapping techniques, according to Table 3 in section 3.2.1, between 5 nodes and 10 nodes. Generally, both figures illustrate that mapping five layers to more than one AC ensures maximizing ERVL value thus having better video quality. As can be seen, there is a clear drop among all nodes in map1 and map2 in both figures due to heavy load of AC_3 since five layers in map1 and 4 layers in map2 are going to be mapped to AC_3. The last layer (one layer) is going to be mapped to AC_2 in map2 causing no difference when mapping all five layers in map1. In addition, the ERVL value in both figures indicates that there is a considerable decrease in number of dropped packets when mapping all layers to 3 ACs and 4 ACs as in map4, map5 and map6 respectively. Although map4 has the largest ERVL value between all other mappings, few packets in the base layer have been dropped due to queue overflow with a small percent equals to 0.39% according to Table 6 which may cause a slight

degradation of video quality. Consequently, having the largest ERVL value does not always guarantee delivering the base layer with zero drop, which is very important to provide basic video quality.

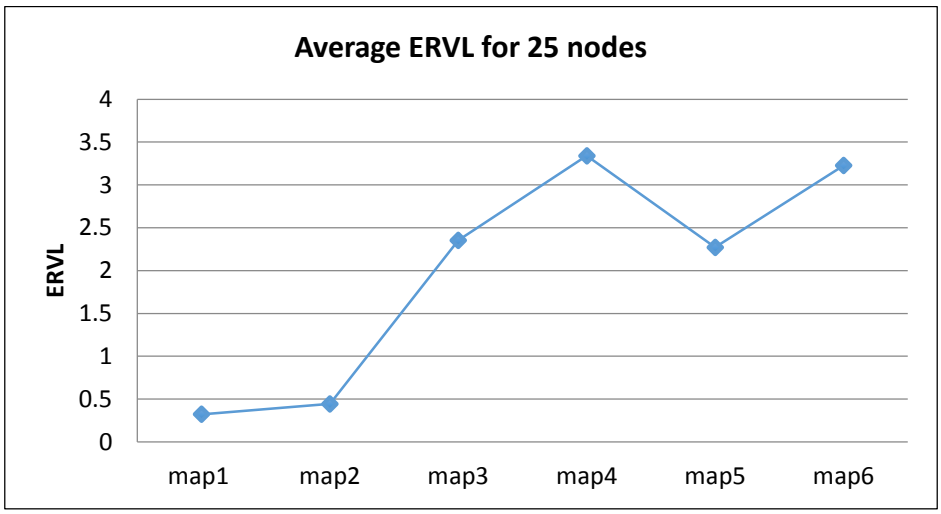


Figure 30: Average ERVL for 5 layers per 25 nodes

Although there is a minimal difference of the ERVL value between map4 and map6, map6 guarantees that most important video packets corresponding to L0 will be delivered with zero-drop. Exploiting all available four ACs guarantees the best ERVL value across all nodes even when the number of nodes increased as in Figure 30. Consequently, distributing all video layers across four ACs (regardless the number of nodes) delivers better video quality.

5.3.2.2 Scenario 2: Impact of Different Video Layers

The results shown in Figure 31 illustrates the loss rate results for a layered video composed of 8 layers distributed over four ACs with different sets of mappings/permutations found in Table 3. As can be observed from the results shown in Table 8, the last two layers L6 and L7 have larger drop rate than other layers, which can affect the video quality as well.

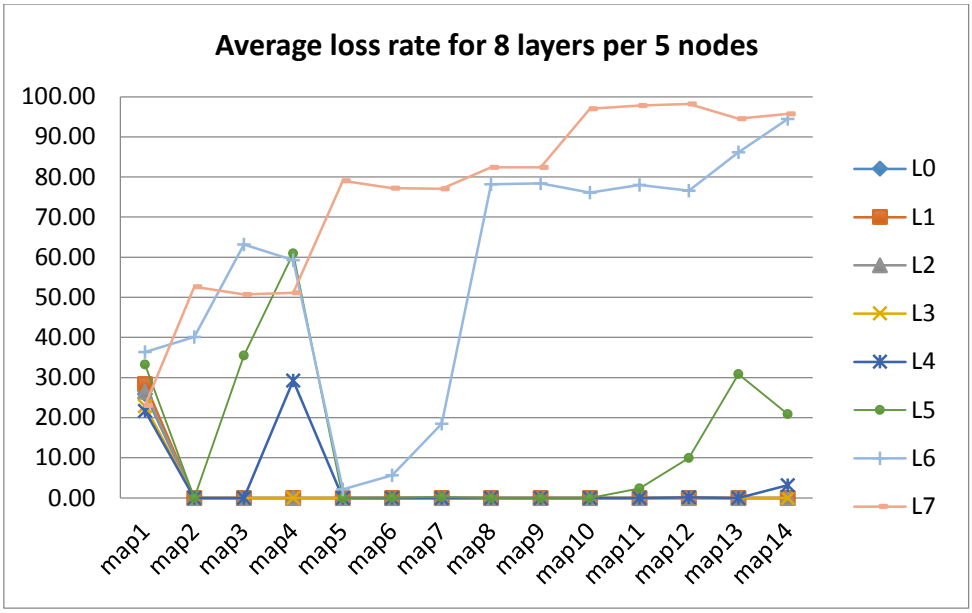


Figure 31: Loss rate for 8 layers for each mapping per 5 nodes

Table 8: Packet loss rate for each layer (%) per each mapping

	L0	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**
map1	25.93	28.36	26.42	22.87	21.67	33.25	36.35	22.99
map2	0.00	0.00	0.00	0.00	0.00	0.00	40.09	52.64
map3	0.00	0.00	0.00	0.00	0.00	35.46	63.20	50.74
map4	0.00	0.00	0.00	0.00	29.26	60.94	59.30	51.15
map5	0.00	0.00	0.00	0.00	0.00	0.00	2.11	79.05
map6	0.00	0.00	0.00	0.00	0.00	0.04	5.65	77.25
map7	0.00	0.00	0.00	0.00	0.00	0.23	18.46	77.03
map8	0.00	0.00	0.00	0.00	0.00	0.00	78.18	82.42
map9	0.00	0.00	0.00	0.00	0.00	0.00	78.44	82.42
map10	0.00	0.00	0.00	0.00	0.00	0.00	76.13	97.04
map11	0.00	0.00	0.00	0.00	0.00	2.33	78.02	97.81
map12	0.00	0.00	0.00	0.00	0.10	9.96	76.54	98.19
map13	0.00	0.00	0.00	0.00	0.00	30.86	86.16	94.56
map14	0.00	0.00	0.00	0.00	3.17	20.90	94.44	95.73

From the results, it can be seen that L0 has no loss rate in all experiments except that in map1 as illustrated in Figure 32 and in Table 9 in which AC_3 has a loss rate below 30%.

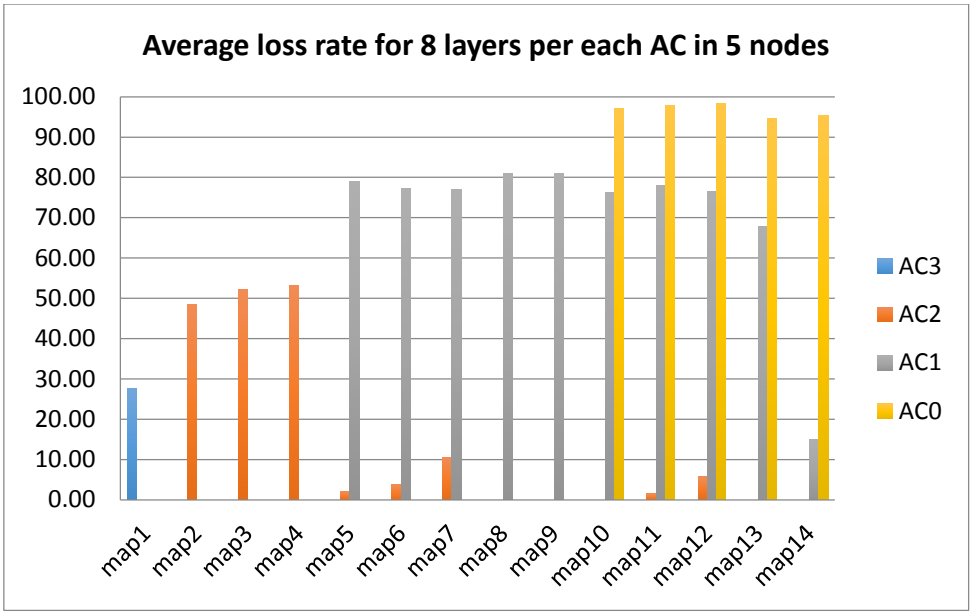


Figure 32: Loss rate for all 8 layers per ACs in node 1

Table 9: Packet loss rate for each AC (%) per each mapping

	AC_3	AC_2	AC_1	AC_0
map1	27.64	0.00	0.00	0.00
map2	0.00	48.45	0.00	0.00
map3	0.00	52.11	0.00	0.00
map4	0.00	53.17	0.00	0.00
map5	0.00	2.11	79.05	0.00
map6	0.00	3.77	77.25	0.00
map7	0.00	10.61	77.02	0.00
map8	0.00	0.00	81.01	0.00
map9	0.00	0.00	81.10	0.00
map10	0.00	0.00	76.13	97.04
map11	0.00	1.56	78.02	97.81
map12	0.00	5.73	76.54	98.19
map13	0.00	0.00	67.73	94.56
map14	0.00	0.00	14.99	95.30

Figure 33 and Figure 34 show a decline in ERVL performance metric for a layered video composed of 8 layers over 5 nodes and 10 nodes respectively. At the beginning, there was a slight change between all nodes in ERVL value for each mapping in Figure 33.

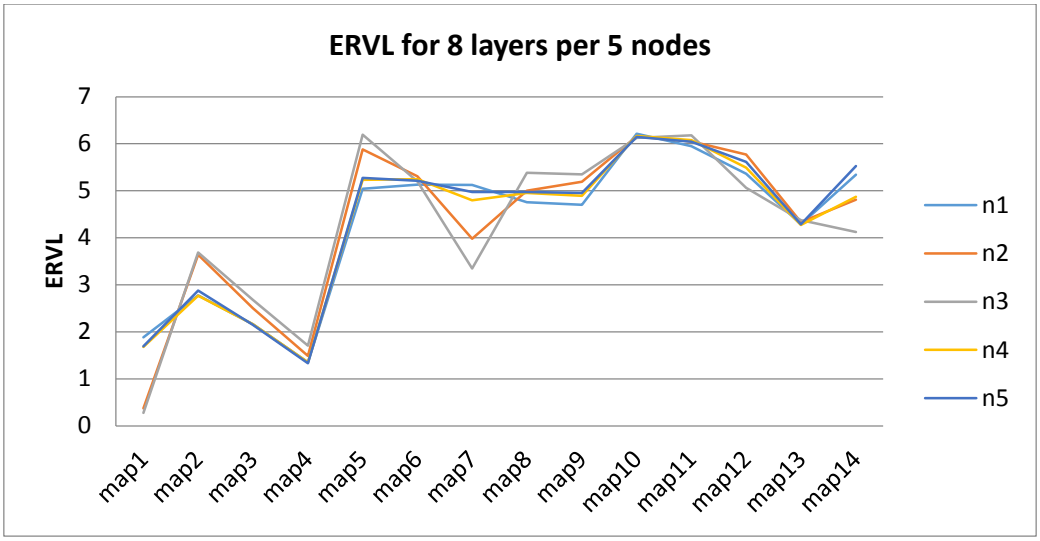


Figure 33: ERVL for different mapping across 8 layers per 5 nodes

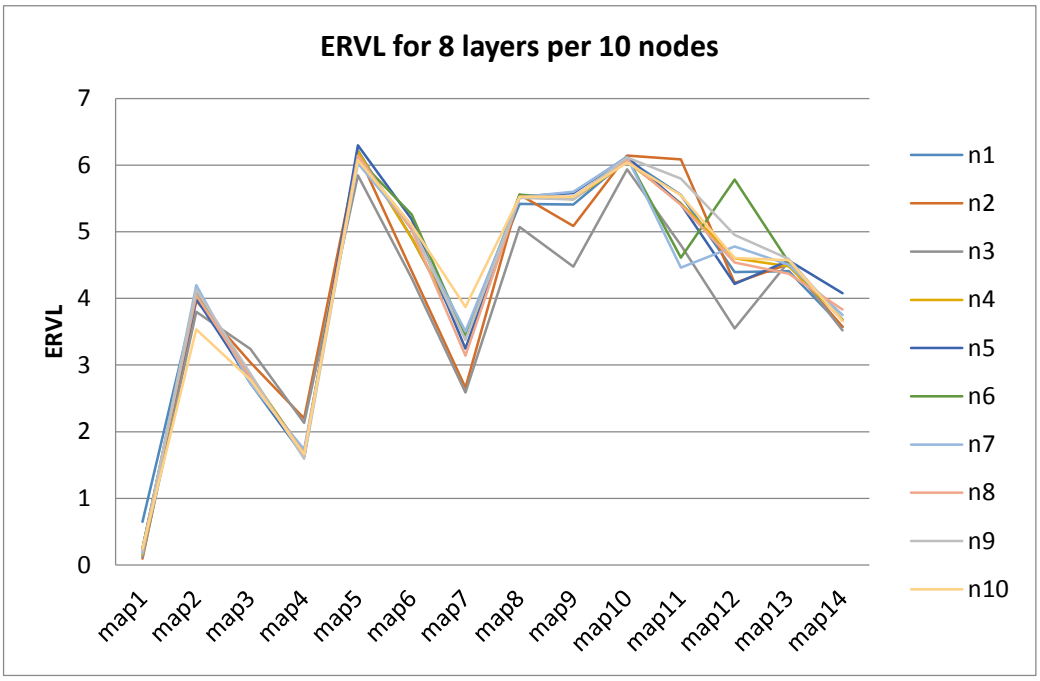


Figure 34: ERVL for different mapping across 8 layers per 10 nodes

Moreover, node 3 and 4 have the largest ERVL value as in “map5” while node 1, 2 and 5 have largest value as in “map10”. Overall, it can be seen that all nodes started to

converge when the number of nodes increased to 10 nodes where “map5” guarantees the best ERVL value as shown in Figure 34.

Figure 35 and Figure 36 show the loss rate for each layer and for each AC for a layered video composed of 10 layers transmitted by 5 nodes respectively. As can be seen, increasing the number of video layers leads to an increase of the possible permutations of mapping video layers to obtain the best video quality. It is clearly seen that the last three layers (i.e. L7, L8 and L9) are the most layers which have high loss rate since they will be always mapped to lowest priority ACs. However, these video layers are used only to refine the video quality not to provide the basic video quality.

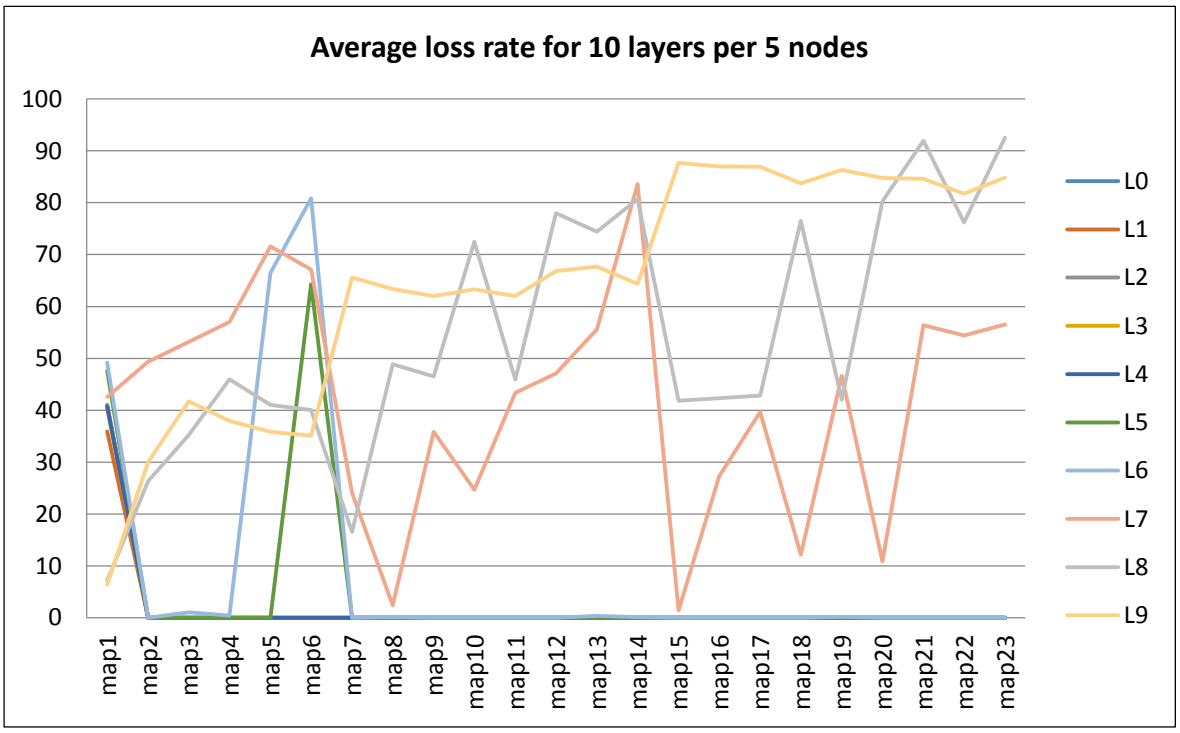


Figure 35:Loss rate for all 10 layers per 5 nodes

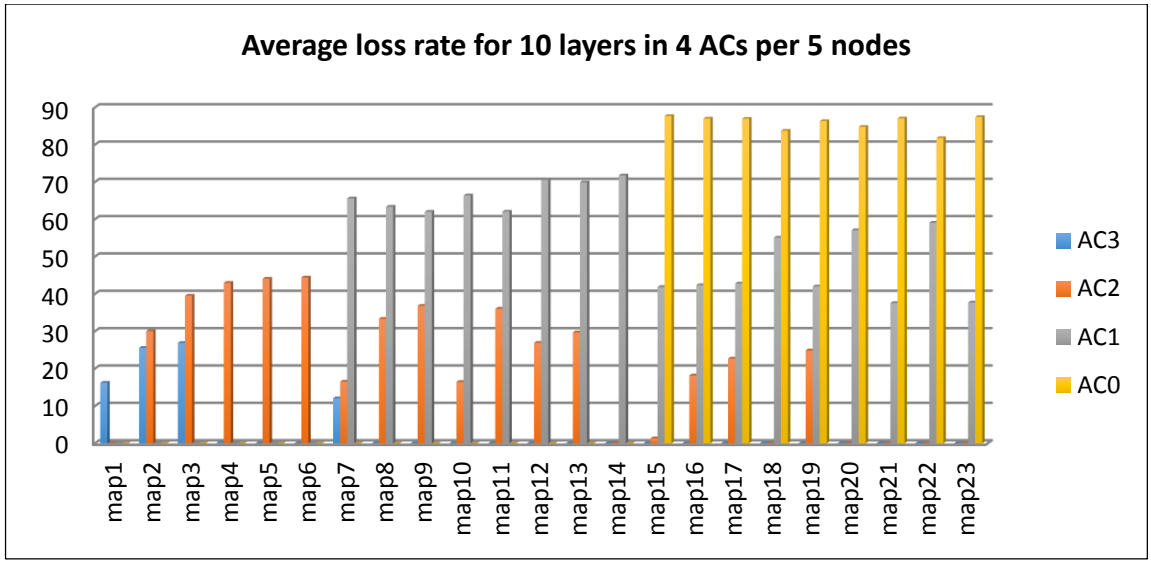


Figure 36: Loss rate for 10 layers over 5 nodes per mapping

Table 10: Packet loss rate for each AC (%) per each mapping over 5 nodes

	AC_3	AC_2	AC_1	AC_0
map1	16.20	0.00	0.00	0.00
map2	25.53	30.04	0.00	0.00
map3	26.87	39.52	0.00	0.00
map4	0.16	42.95	0.00	0.00
map5	0.00	44.05	0.00	0.00
map6	0.00	44.39	0.00	0.00
map7	12.04	16.50	65.57	0.00
map8	0.04	33.35	63.35	0.00
map9	0.00	36.82	62.00	0.00
map10	0.00	16.43	66.39	0.00
map11	0.00	36.03	62.03	0.00
map12	0.00	26.91	70.55	0.00
map13	0.00	29.71	69.91	0.00
map14	0.00	0.05	71.75	0.00
map15	0.00	1.35	41.85	87.67
map16	0.00	18.17	42.36	87.00
map17	0.00	22.69	42.78	86.94
map18	0.00	0.00	55.08	83.72
map19	0.00	24.86	41.99	86.31
map20	0.00	0.00	57.09	84.76
map21	0.00	0.00	37.56	87.03
map22	0.00	0.00	59.08	81.77
map23	0.00	0.00	37.70	87.41

Figure 37 presents the results for ERVL for a multi-layered video with 10 layers over 5 nodes. As can be shown, all nodes have the highest ERVL when mapping video layers using "map15" with permutation (7,1,1,1) where 7 layers are mapped to AC_3 with zero dropped packets as shown in Figure 36 and in Table 10. While 1 layer is mapped to AC_2 with 1.3% packet drop, 1 layer is mapped to AC_1 with 41.8% packet drop and the last layer is mapped to AC_0 with 87.6% packet drop thus maximizing ERVL value.

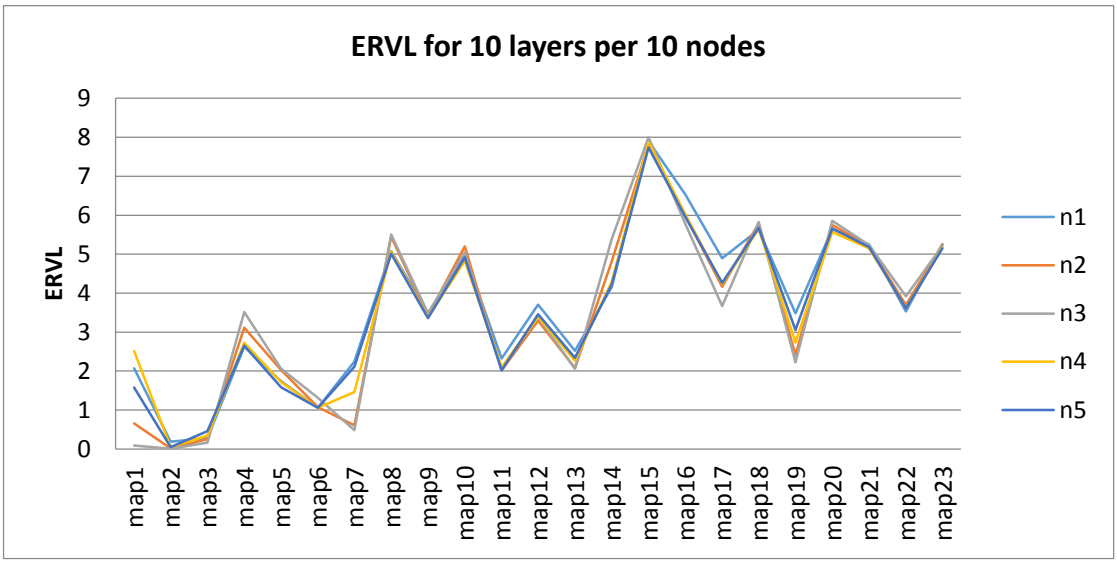


Figure 37: ERVL for different mapping across 10 layers per 10 nodes

Based on the obtained results, the ERVL value can be affected directly by the loss rate for each AC. The lowest the loss rate per AC, the highest ERVL value and the best video quality can be achieved. Regardless the number of nodes in all previous experiments, the best ERVL can be achieved when lower layers are mapped to AC_3 with zero dropped packet including base layer as well. However, when the number of packets per layers is increased specially in the last layers according to H.264 coding scheme, most of these packets will be dropped because they are going to be mapped to lowest priority ACs (AC_1 and AC_0). This leads to higher loss rate due to queue overflow congestion.

5.3.3 Experiment Set B: Utilizing Network Resources for Foreground Traffic and Background Traffic

The previous section addressed DSM algorithm performance for a video traffic over a fixed network condition. As discussed previously that DSM algorithm leverages MAC layer resources when considering mapping all video layers over four ACs. The experiment assumed the absence of other traffic types running in the background. This section particularly studies the effect of DSM algorithm for video traffic under varying directed AC traffic. This traffic is produced by composing Voice traffic mapped to AC_3, Video traffic mapped to AC_2, HTTP traffic (Best Effort) mapped to AC_1 and finally FTP traffic mapped to AC_0. This resulted in five groups of experiments as shown in Table 5, section 5.1. Each experiment group is further tested with three levels of network loads namely, light, intermediate and heavy. Each experiment has been carried out 50 times. Similar experimental work has also been conducted while varying number of layers (L = 3, 5, 8 and 10), each repeated varying number of nodes in the network (N=5, 10 and 15). The subsequent section presents the results and discussions for each of these experiment sets with the best mapping being agreed upon by all nodes on the network.

Group 1 Experiments

In this group of experiments, light, intermediate and heavy voice traffic mapped to AC_3 have been introduced together with the ongoing video traffic. Five, ten, and twenty concurrent flows of voice traffic are activated in light, intermediate and heavy traffic load levels (i.e. "**SL1**", "**SI1**" and "**SH1**") respectively. Detailed settings of a voice flow and a video flow are discussed in section 5.1. The average loss rate for each layer and the average ERVL are reported by these experiments. Consequently, the experiments are repeated with a variation of L layers and N nodes.

Generally in case of layered video composed of L = 3, 5 and 8 layers, the results show that when the voice background load increases, the ERVL decreases among all stations due to high delay as shown in Figure 38: (a), (c) and (e). The effect of the high delay is caused by the transmission of multiple voice flows mapped to AC_3 (AC_VO) that are trying to gain access of the channel. These flows have the same time window to gain access to the medium competing with the collision window of the video traffic that is also mapped by the DSM algorithm permutations to the same AC. In case of layered video with L = 3, the base layer (L0) has the most important video information that is mapped to AC_3. However, 50% of its packets have been dropped as shown in Figure 39: (b). While layers mapped to AC_2 do not suffer from high loss rate because there is no other background traffic competing with them and they are mapped to the second highest priority AC to access channel resources. For example, L1 shown in Figure 39: (b) is only mapped to AC_2 having less amount of dropping packets than other layers. Nevertheless, since L2 is mapped to lower priority AC (AC_1), it has the highest dropping rate among all layers.

Therefore, when there are other traffics mapped to AC_3 beside video traffic, the video layers suffer severely from high degradation in video quality. Thus the best ERVL value can be obtained considering the permutation (1, 1, and 1) “map3” when there are light voice traffics transmitted over AC_3 and exploiting all ACs as N = 5, 10 and 15 shown in Figure 38: (a). However, when the traffic load increases as in SI1 and SH1, the best ERVL value can be obtained when mapping all video layers into AC_3 only considering “map1” as N = 10 and 15 as illustrated in Appendix A. Appendix A shows a summary of the optimal mapping results for delivering a video with different layers over different number of nodes. This is expected and is mostly due to the drop rate in the AC_3 queue since all voice traffic is directed to this queue. Furthermore, the results show that when the number of nodes increases, the highest

video quality can be achieved with an increase of ERVL value from 0.45 when N = 5 and 10 to 0.66 when N = 15.

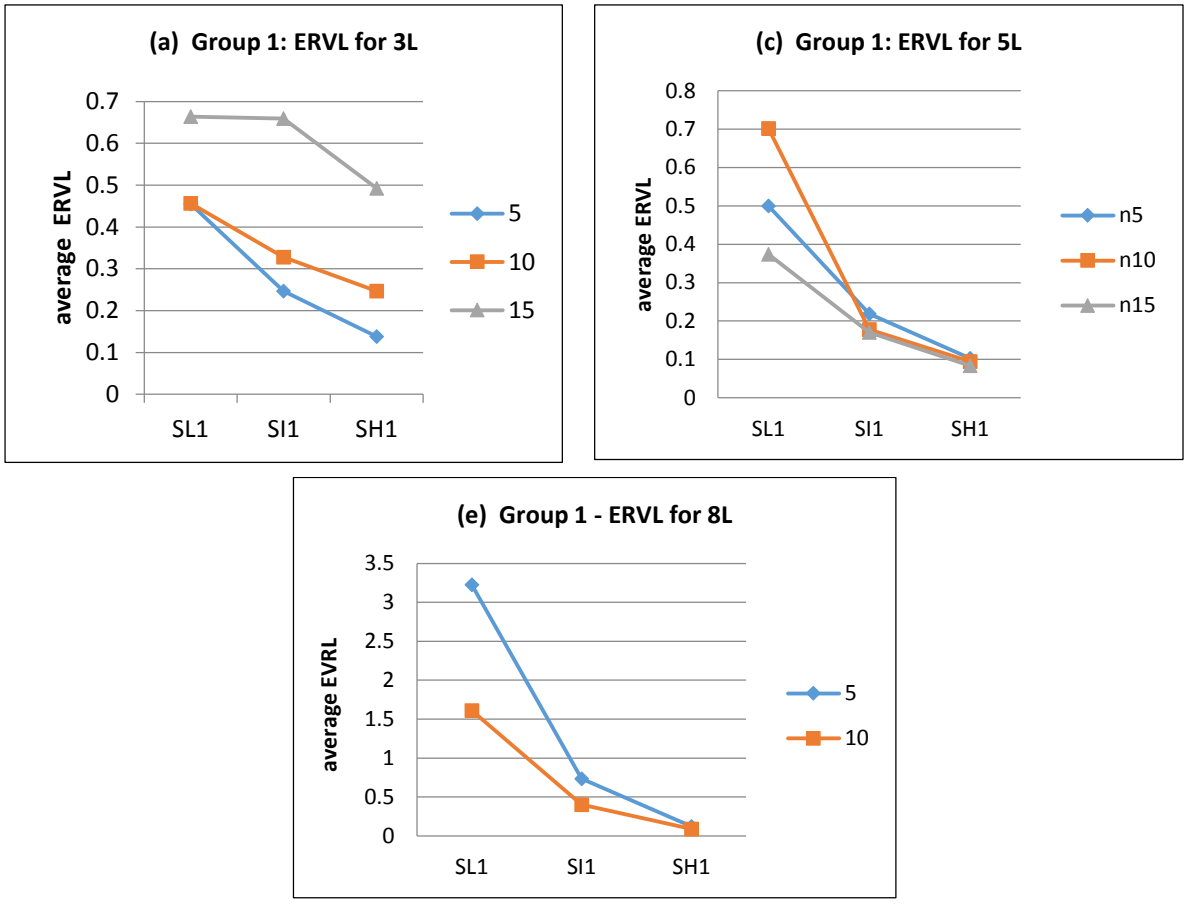
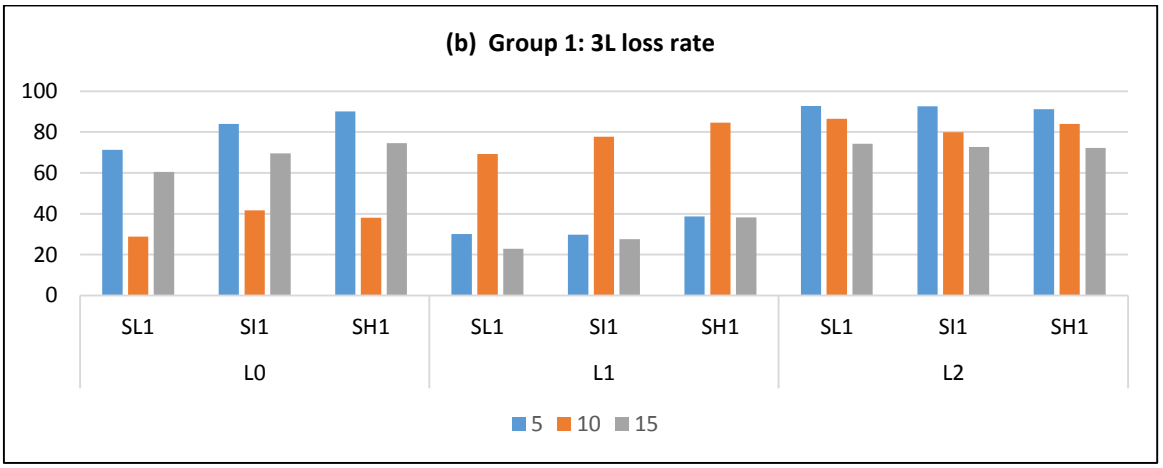


Figure 38: Average ERVL for Group1 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL1, SI1 and SH1) for a layered video with: (a) L = 3, (c) L = 5, and (e) L = 8 layers.



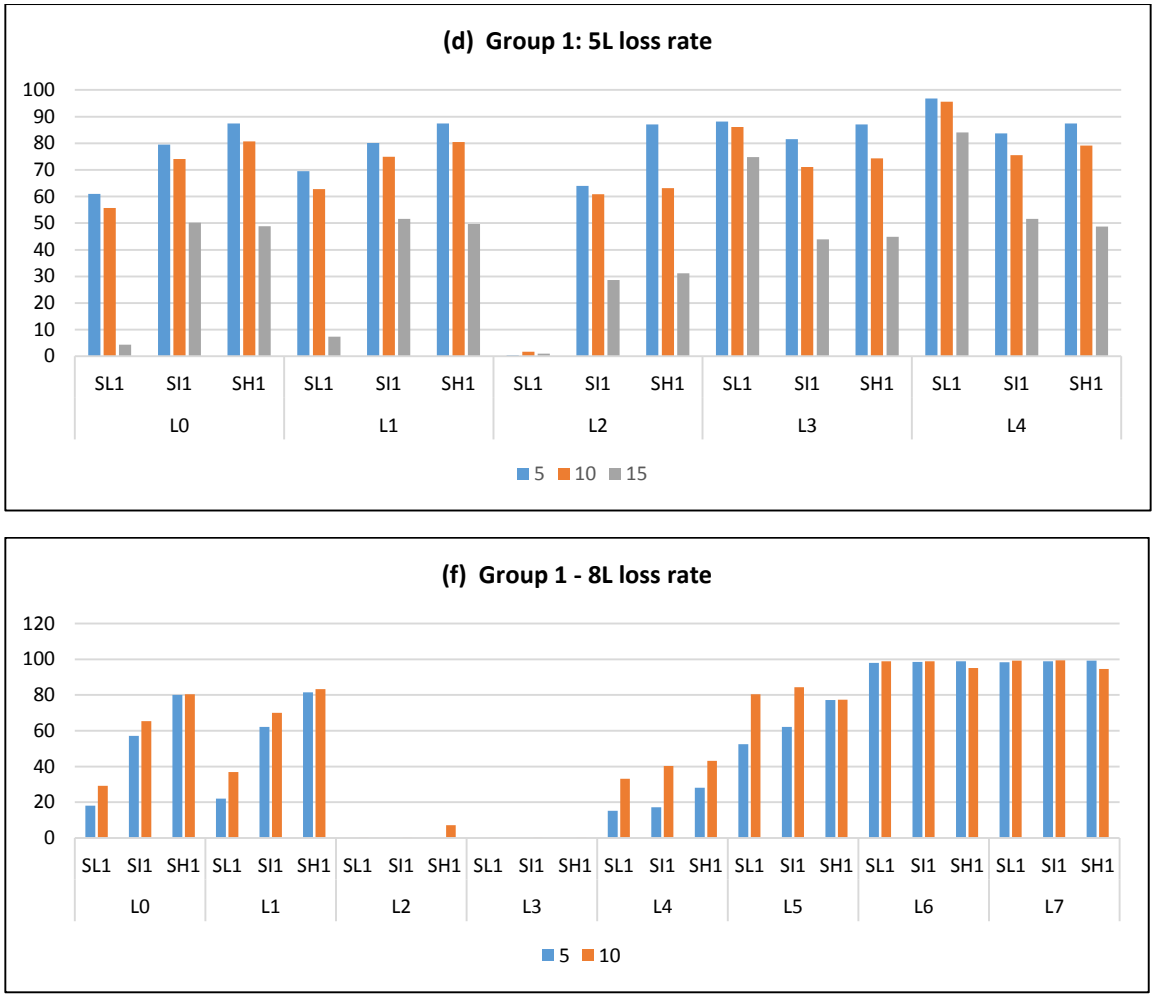


Figure 39: Average loss rate per each layer for Group1 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL1, SI1 and SH1) for a layered video with: (b) L = 3, (d) L = 5, and (f) L = 8 layers.

The group of experiments in case of layered video composed of L = 5 and 8 are to be carried out in the same way as the previous ones in case of L = 3. The ERVL performance metric is measured considering a video traffic composed of L = 5 as shown in Figure 38: (c) and Figure 39: (d), L = 8 layers as shown in Figure 38: (e) and Figure 39: (f) with different scenarios as described in Table 5. In case of 5 layers, the ERVL can be obtained considering the permutation (2, 1, 1 and 1) "map6" when there are light and intermediate traffic transmitted to AC_3. Nevertheless, when the voice traffic load rises as in SH1 scenario the

mapping is changed to “map1” having all five layers mapped to AC_3 only besides voice traffic according to Appendix A. The same conclusion applied in case of 8 layered video where the best ERVL value can be obtained considering the permutation (2, 2, 2 and 2) “map14” when there are light voice traffics transmitted over AC_3 and exploiting all other ACs as N = 5. However, when the voice load traffic increases as in SI1 and SH1, the best ERVL value can be obtained when mapping all video layers into AC_3 only considering “map1” as N = 10 as illustrated in Appendix A.

Group 2 Experiments

In this group of experiments, five, ten, and twenty concurrent flows of HTTP traffic are activated in light, intermediate and heavy traffic load levels (i.e. “**SL2**”, “**SI2**” and “**SH2**”) respectively and mapped to AC_1 together with the ongoing video traffic. Detailed settings of HTTP flow are discussed in section 5.1. The average loss rate for each layer as well as the average ERVL are reported by these experiments. Consequently, experiments are repeated with different configuration of L layers and N nodes.

In case of 3 layers, results for Group2 experiments show that best ERVL can be achieved when mapping all video layers across 3 ACs using permutation (1, 1, and 1) “map3” as in Appendix A and having different HTTP traffic load levels mapped to AC_1. The number of expected video layers to be reconstructed is 1.14 when N = 5 as in Figure 40: (a). Nevertheless, ERVL can be affected slightly when there are heavy traffics transmitted concurrently with video traffic as in SI2 and SH2 and when the number of nodes increased as in N = 10 and 15 such that the number of expected video layers to be constructed is still above 1 layer.

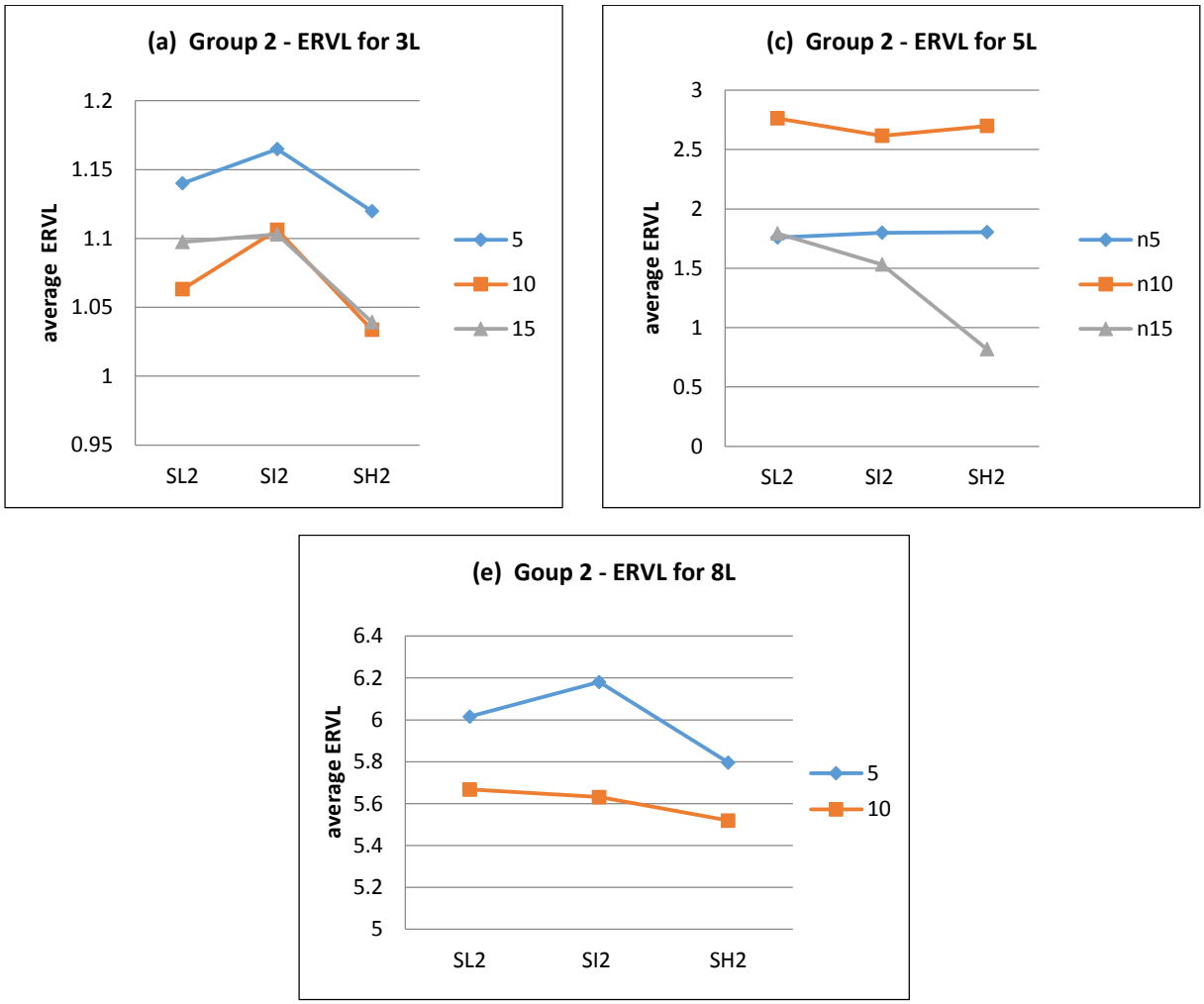
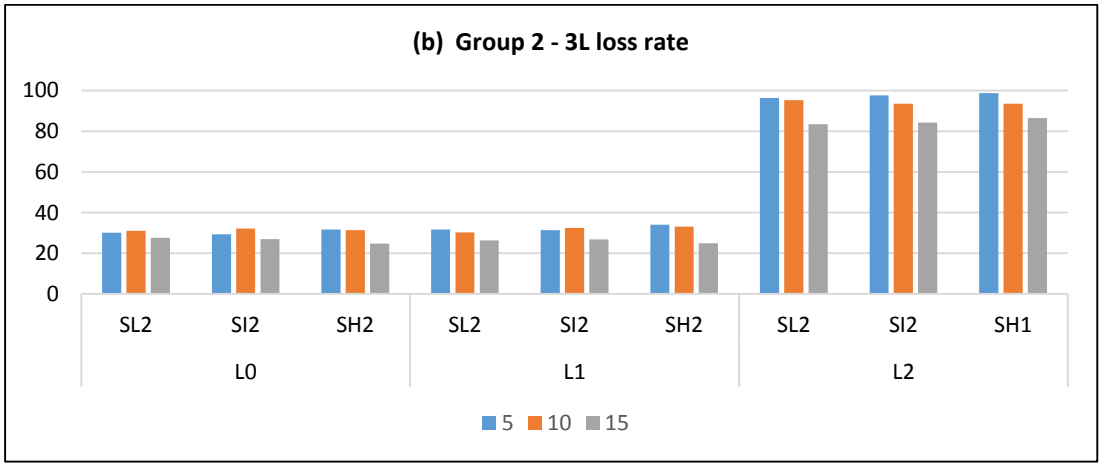


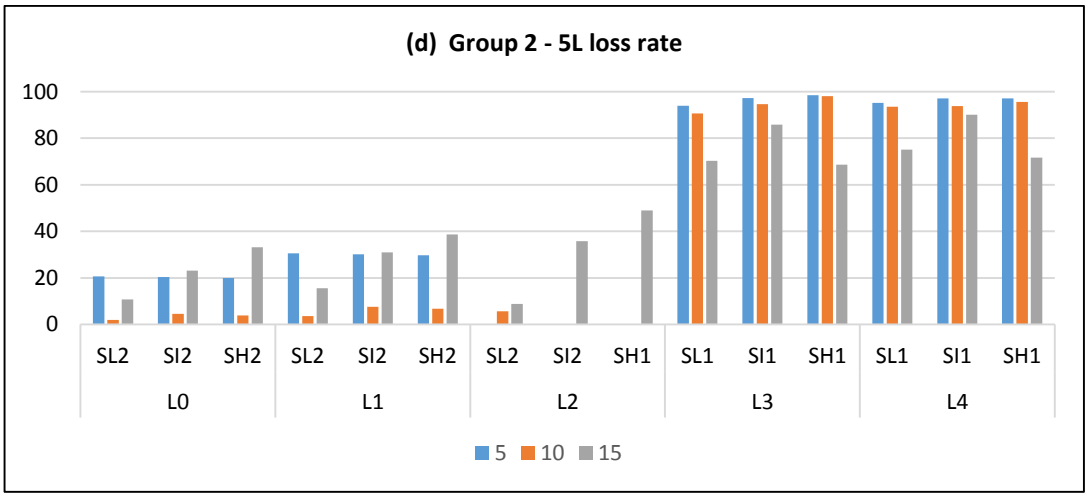
Figure 40: Average ERVL for Group2 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL2, SI2 and SH2) for a layered video with: (a) L = 3, (c) L = 5, and (e) L = 8 layers.

Whereas in case of 5 layers, the number of expected video layers to be reconstructed is above 2.5 when N = 10 and is above 1.5 when N = 5 in all different traffic load scenarios as seen in Figure 40: (c). However, when N = 15, ERVL started to decrease gradually when the traffic load increases from 1.7 in case of light load traffic to 0.8 in case of heavy load traffic.

In case of 8 layers, 6 layers are expected to be reconstructed in case of SL2 and SI2 scenarios while in SH2, ERVL is equal to 5.8 which is almost near to 6 when N = 5 as shown in Figure 40: (e). When N = 10, ERVL suffers from minimal drop reaches up to 5.5 in SH2

case. As can be observed from Appendix A that all nodes do not agree on certain mapping in case of SL2 and SH2 in which 50% of nodes agreed on “map5” to be their best mapping in SL2 scenario and 50% of nodes agreed on “map12” to be their best mapping in SH2 scenario.





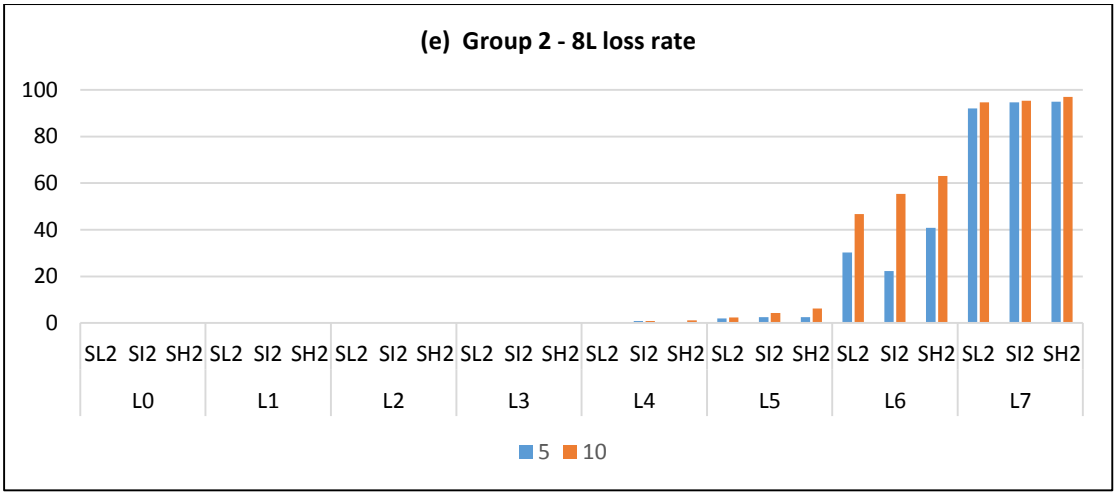


Figure 41: Average loss rate per each layer for Group2 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL2, SI2 and SH2) for a layered video with: (b) L = 3, (d) L = 5, and (f) L = 8 layers.

According to the ACs utilization, results show that when there are lower priority traffics (i.e. HTTP) mapped to AC_1 besides video traffic that is mapped using DSM algorithm permutations, the lower priority (AC_1) does not affect transmission delay of higher priority ACs (AC_3 and AC_2). Since HTTP traffic is mapped to an AC that has different time window from AC_3 and AC_2 to gain access of the medium. AC_1 has higher CWmin, CWmax values and AIFS with a smaller TXOPlimit than AC_3 and AC_2 to guarantee successful transmission, thus it has a lower priority to access channel medium and to compete with video layers mapped to those ACs.

Therefore, video layers mapped to AC_1 will be more affected by background traffic. To make it clear, lower layers (i.e. last layers) are mapped to lower priority ACs, and the important video layers are mapped to AC_3 and AC_2. In this case, the base layer loss ratio has been decreased below 40% as shown in Figure 41: (b) and (d) in case of 3 and 5 video layers respectively. While in 8 video layers, all first 5 layers have zero-drop as shown in Figure 41:

(f), because all these layers are transmitted over AC_3 and AC_2 and are not competing with other traffics.

Group 3 Experiments

In this group of experiment, five, ten and twenty FTP traffic flows are delivered over AC_0 in term of light, intermediate and heavy traffic load levels (i.e. "**SL3**", "**SI3**" and "**SH3**") together with the ongoing video traffic. Detailed settings of FTP flow are discussed in section 5.1. The average loss rate for each layer as well as the average ERVL are reported by these experiments. Consequently, the experiments are repeated with a variation of L layers and N nodes.

In case of L = 3, It can be observed from Figure 42: (a) that the additional best effort traffic has no effect on the video quality since the number of possible ACs to be exploited cannot be more than 3 ACs as shown in Table 3. FTP flow has been mapped to AC_0 resulting in having an AC that has not been used by video traffic permutations, therefore ERVL has not been affected by the concurrent best effort traffic transmitted into AC_0. Thus, all nodes agreed on "map3" to deliver video traffic as in **SL3**, **SI3** and **SH3** traffic scenarios. However, ERVL increases even when network load level becomes more congested since ERVL is not affected by the FTP flow that is directed to an AC that has no video layers mapped by the DSM permutations.

Whereas in case of L = 5 and 8, the number of layers are more than number of ACs, therefore AC_0 is going to be utilized for both FTP best effort traffic and the ongoing video traffic. According to Table 3, when mapping video traffic with L = 5 layers, best ERVL value is greater than 2.5 can be obtained using the permutation (2, 1, 1 and 1) "map6" in case of N = 10 and 15. But when N = 5, ERVL decreases to 1.7 in all traffic load levels as shown in Figure 42: (c). While mapping video traffic with L = 8, best ERVL above 5.5 can be obtained

using the permutation (4, 2, 1, and 1) "map11" when N = 5 and 10 and in all traffic load levels as shown in Figure 42: (e).

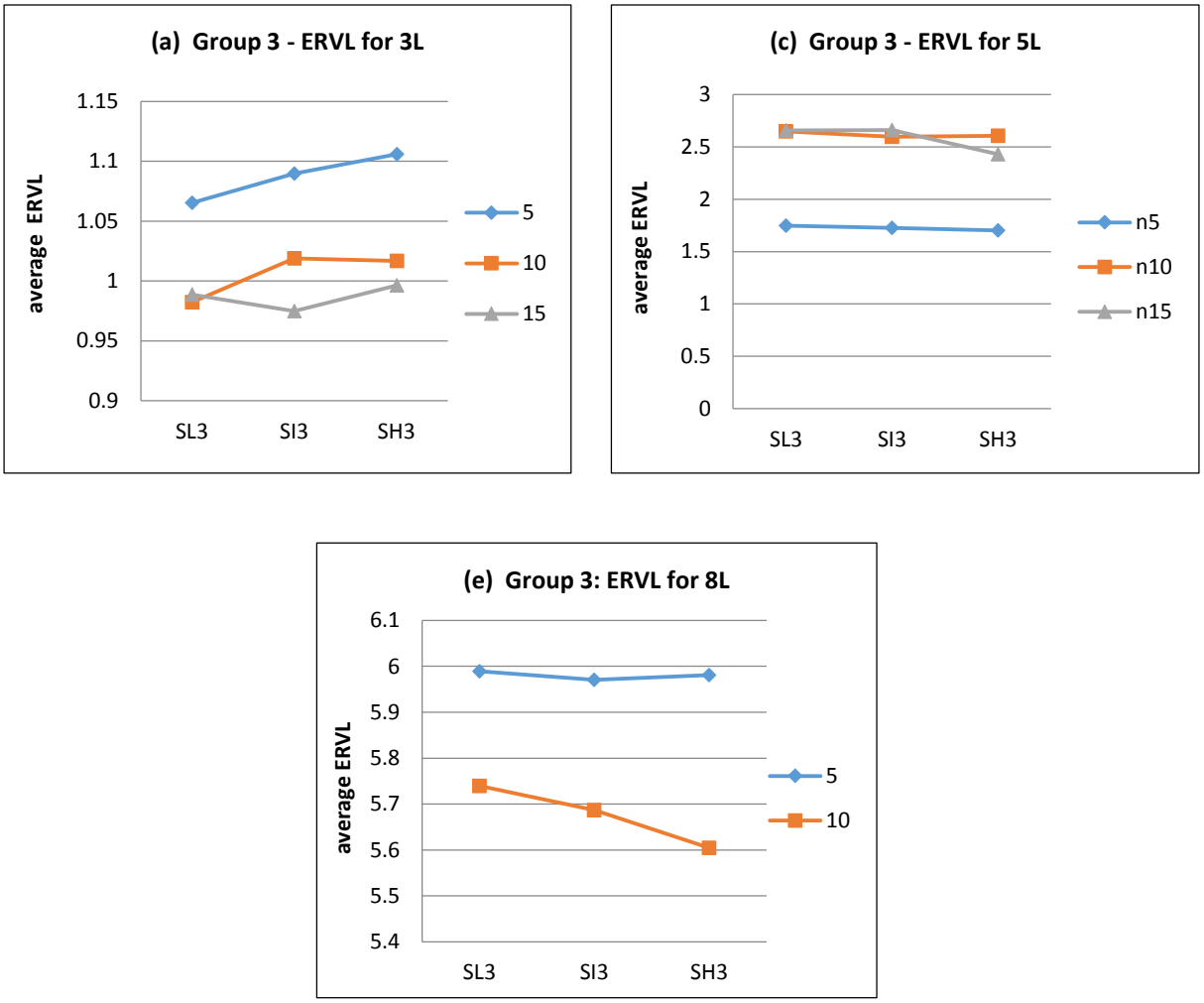
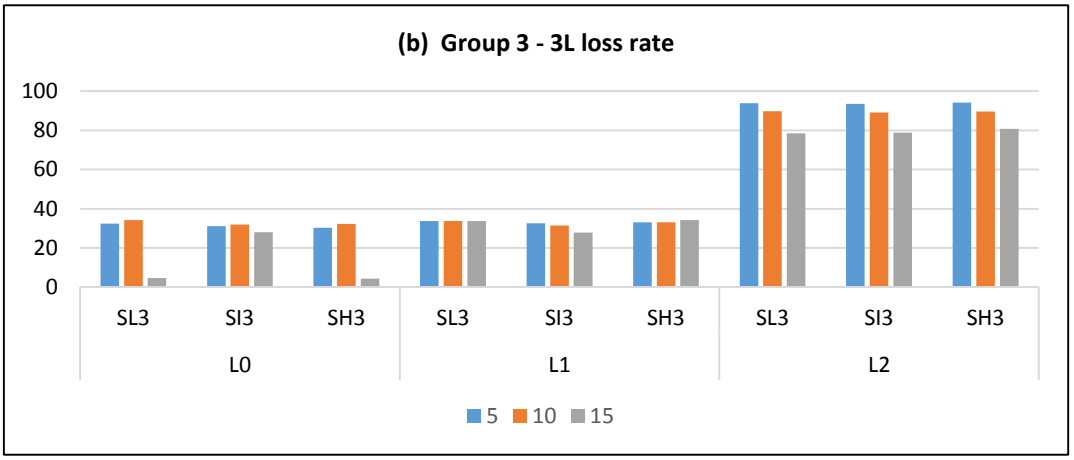
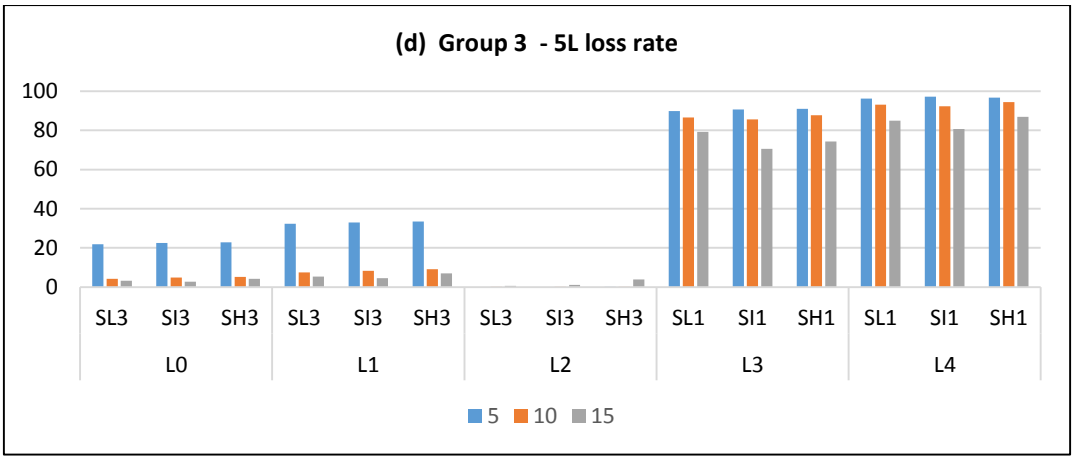


Figure 42: Average ERVL for Group3 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL3, SI3 and SH3) for a layered video with: (a) L = 3, (c) L = 5, and (e) L = 8 layers





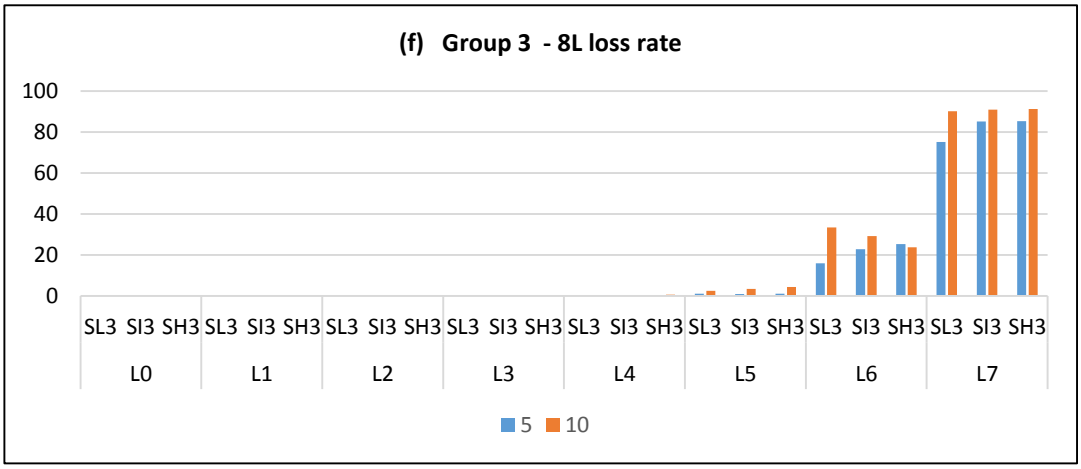


Figure 43: Average loss rate per each layer for Group3 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL3, SI3 and SH3) for a layered video with: (b) L = 3, (d) L = 5, and (f) L = 8 layers.

Figure 45: (b), (d) and (f) illustrate the loss rate for layered video with L = 3, 5 and 8 layers respectively. It can be observed that when video layers equal to L = 3, lower layers: base layer (L0) and Enhancement layer (Enh1) have lower drop rate below 40% than Enh2. This happened due to L0 and Enh1 prioritized over Enh2 packets and mapped to higher priority ACs (AC_3 and AC_2). This drop rate for lower layers also decreases when the number of video layers increased as in Figure 45: (d) and (f) which ends with zero drop rate when L = 8 for the first 5 layers.

Group 4 Experiments

This group of experiments examine the ERVL performance when there are three kinds of concurrent flows delivered over all ACs besides the ongoing video flow traffic. For example, five, ten and twenty concurrent traffic flows are activated in light, intermediate and heavy load levels (i.e. "**SL4**", "**SI4**" and "**SH4**") respectively as described in Table 5. Those kinds of flows include voice flows mapped to AC_3, HTTP flows mapped to AC_1 and FTP flows mapped to AC_0. The average loss rate for each layer as well as the average ERVL are presented by these experiments. Consequently, experiments are repeated with different settings of L layers and N nodes.

In case of 3 layers, the results in Figure 44: (a) show that ERVL decreases gradually when the number of background load level increases among all nodes. Based on Appendix A, best ERVL value can achieved when using "map3" when N = 5 in all load levels. While best ERVL value when N = 10 and 15, can be obtained using permutation (1, 1, and 1) "map3" in SL4 scenario only but in case of SI4 and SH4 best ERVL can be obtained using permutation (3,0,0) "map1". This happened since there are other traffics mapped to highest priority ACs besides the video traffic resulting a degradation of ERVL video quality metric. Moreover, results moved to the same trend when L = 5 as in Figure 44: (c) and when L = 8 as in Figure 44: (e) where best ERVL can be obtained using "map6" in case of 5 layers having SL4 and

using “map1” having SI4 and SH4. The best ERVL can be obtained using “map14” in case of SL4 and SI4 and using “map1” in case of SH4.

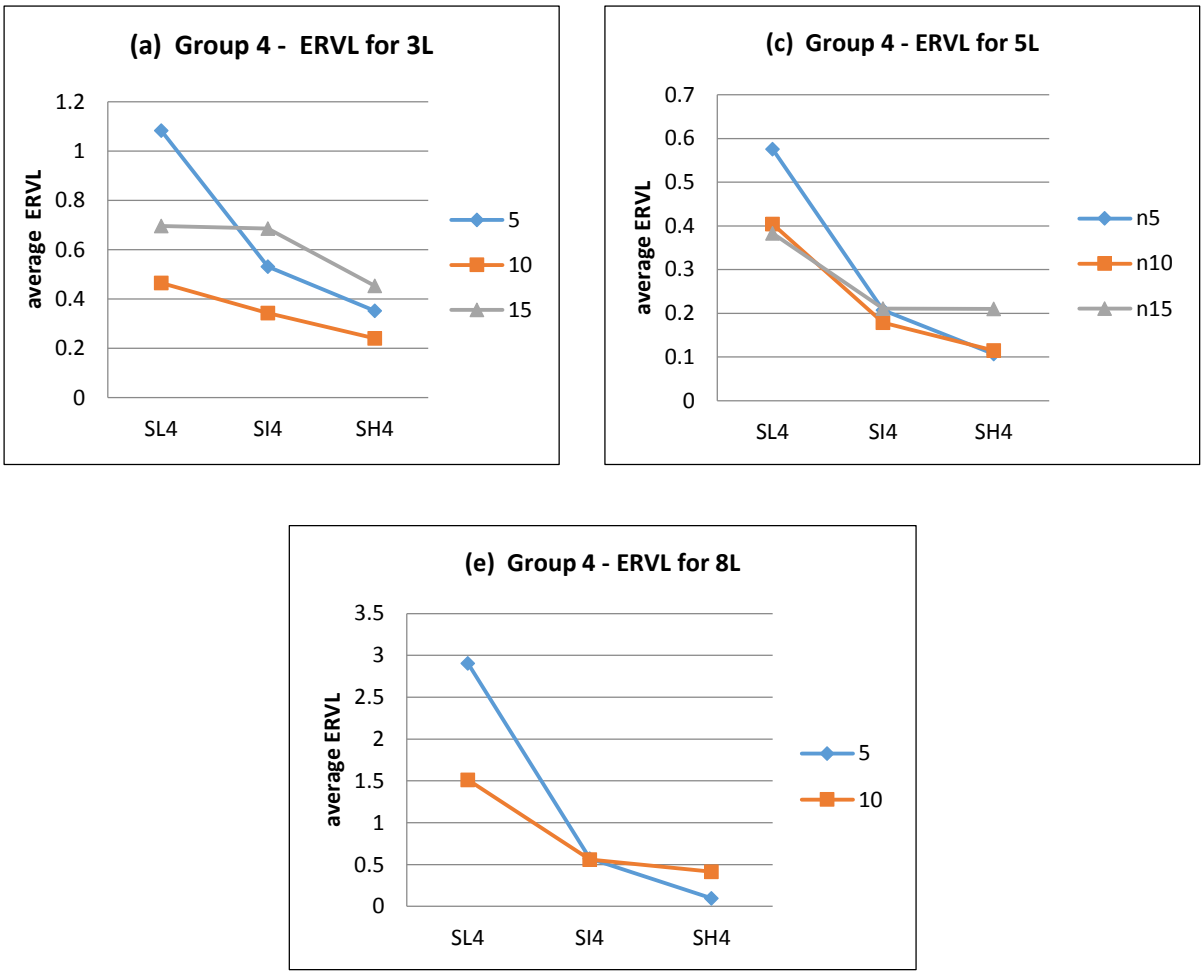
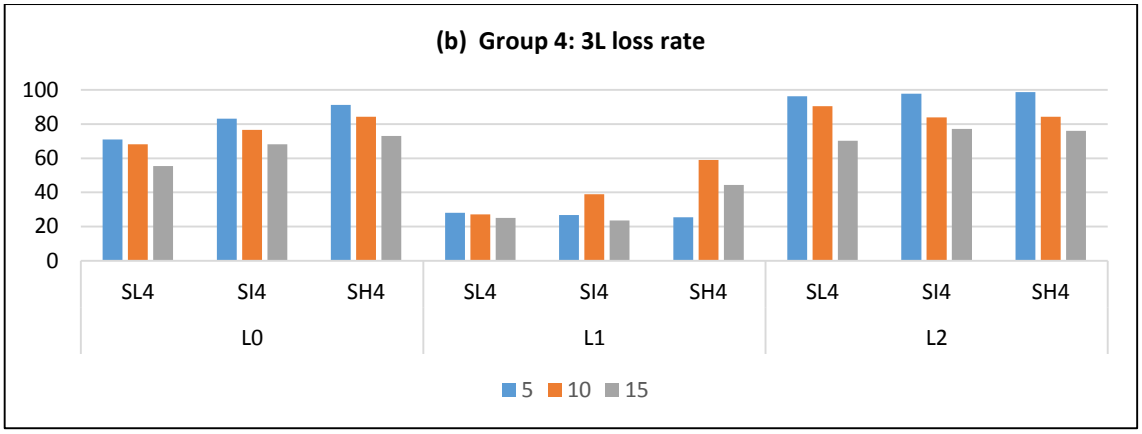
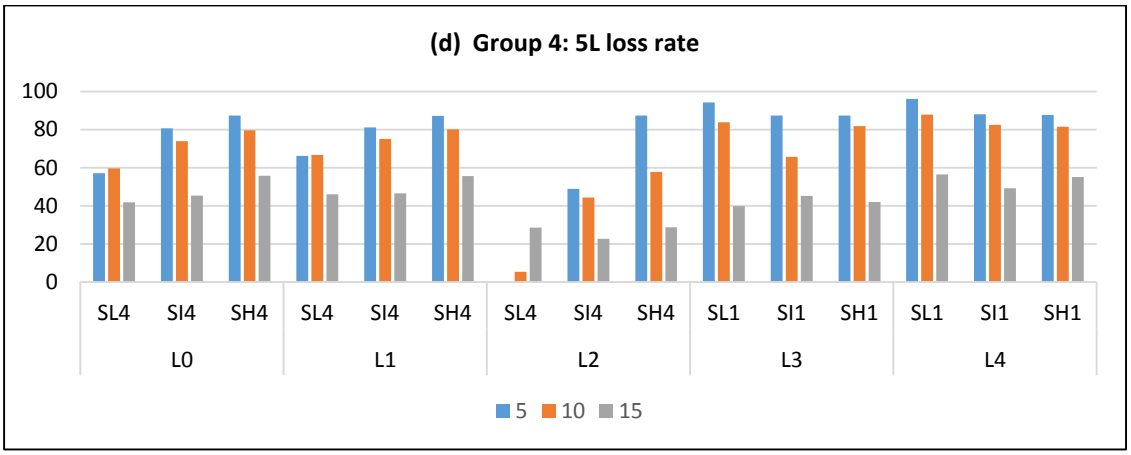


Figure 44: Average ERVL for Group4 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL4, SI4 and SH4) for a layered video with: (a) L = 3, (c) L = 5, and (e) L = 8 layers





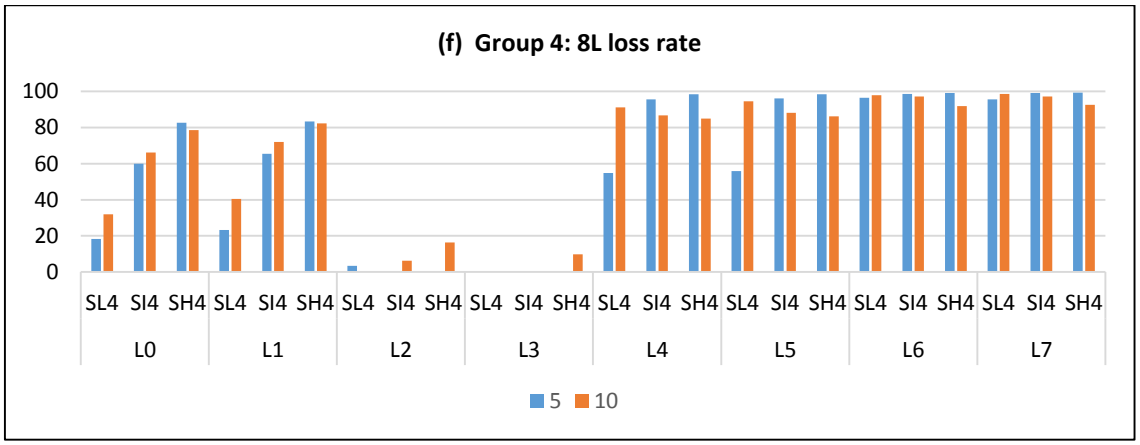


Figure 45: Average loss rate per each layer for Group4 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL4, SI4 and SH4) for a layered video with: (b) L = 3, (d) L = 5, and (f) L = 8 layers.

The amount of lost packets in L0 is very high compared to other layers as shown in Figure 45: (b), (d) and (f). There are multiple traffics from all ACs competing with video traffic to gain access of the medium causing larger drop rates due to queue overflow. As a result, more than 50% of base layer packets have been dropped causing a severe degradation of ERVL value resulting in bad video quality. However, video layers mapped to AC_2 do not suffer from great amount of packets being lost because there are no other traffics contending with those video layers into this AC. As an example, L1 in case of layered video composed of 3 layers, L2 in case of layered video composed of 5 layers and (L 2, L3) in case of layered video composed of 8 layers.

Group 5 Experiments

Finally, this group of experiments illustrate ERVL performance when there are other video traffics mapped to AC_2 (AC_VI) besides the main video traffic contending to gain access of the channel resources. One and two concurrent flows of video traffic are activated in intermediate and heavy traffic load levels (i.e. "**SI5**" and "**SH5**") respectively while in light load traffic (i.e. "**SL5**"), only one flow from each kind of traffic is delivered over ACs as shown in Table 5. The average loss rate for each layer as well as the average ERVL are reported by these experiments. Consequently, experiments are repeated with a variation of L layers and N nodes.

The results show that ERVL value decreased gradually when the background traffics increased among all traffic load levels when L = 3, 5 and 8 as illustrated in Figure 46: (a), (c) and (e). It indicates that other video flows are delay constrained and having same window size competing against each other to access the wireless medium resources. This happened due to there are no priorities between the transmitted video flows so every video flow contends each other to acquire available bandwidth. Best ERVL can be achieved in a layered

video with L = 3 using “map3” while in a layered video with L = 5 best ERVL can be found using “map6”. However, in case of 8 layered video all nodes do not agree on certain mapping to obtain optimal ERVL value as shown in Appendix A.

The loss rate for each layer in different number of video layers is shown in Figure 47: (b), (d) and (f) accordingly.

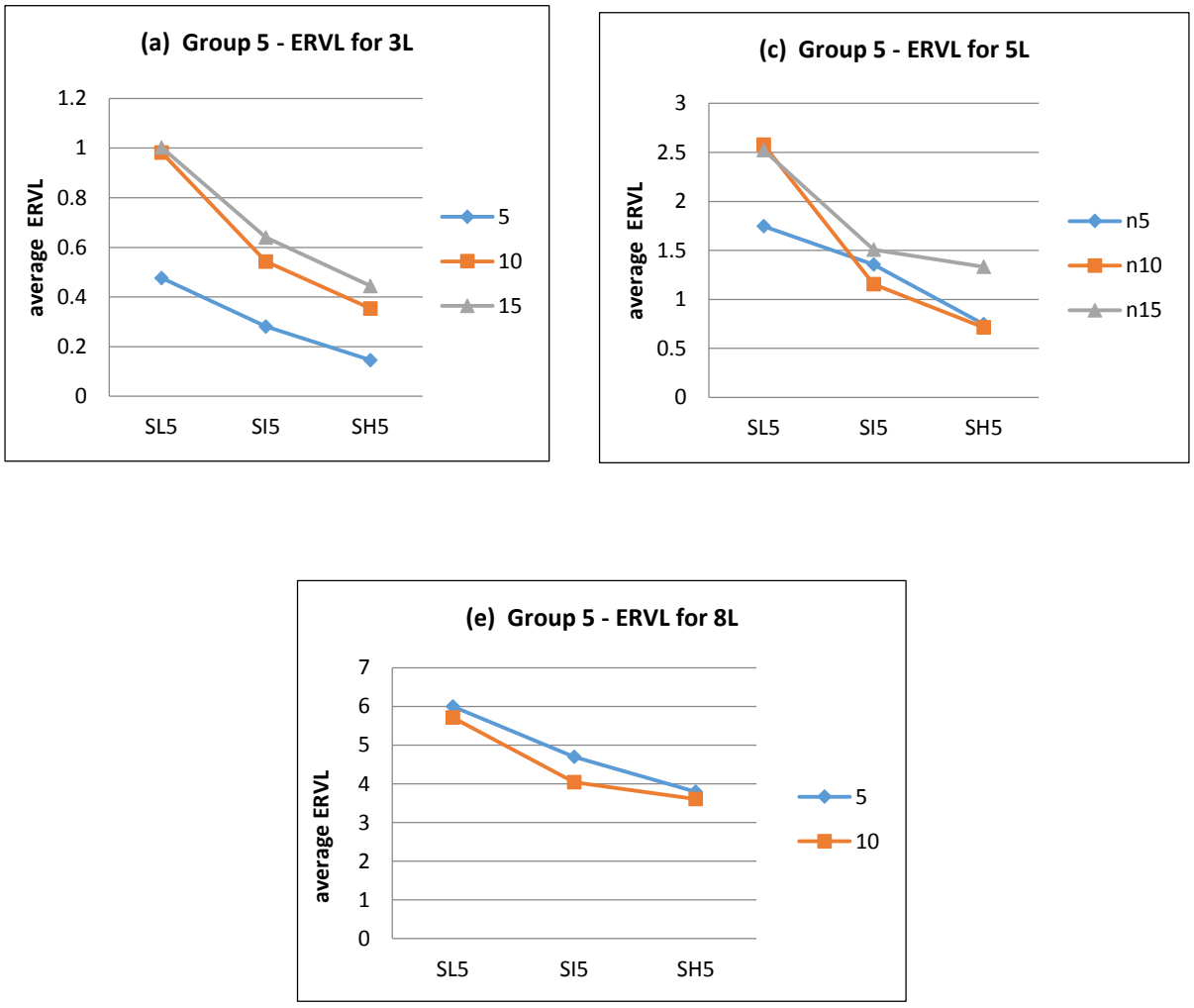
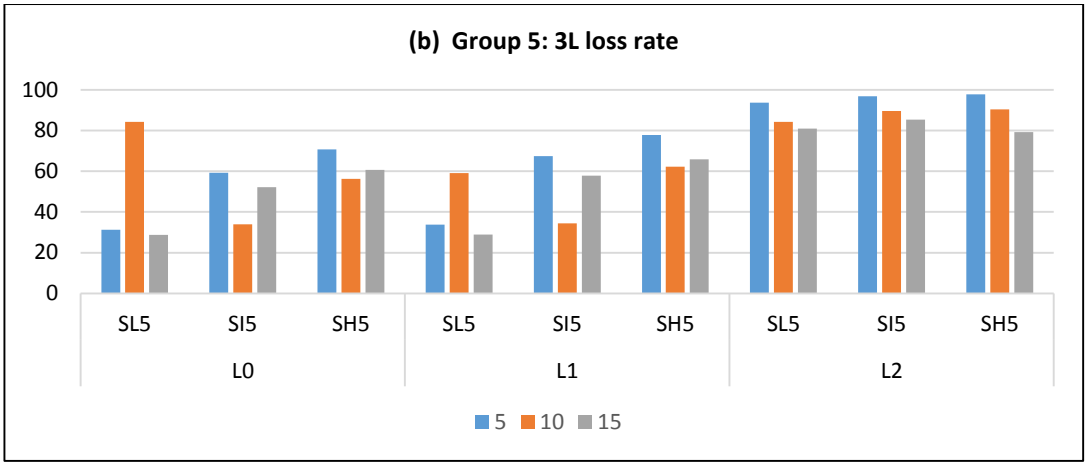
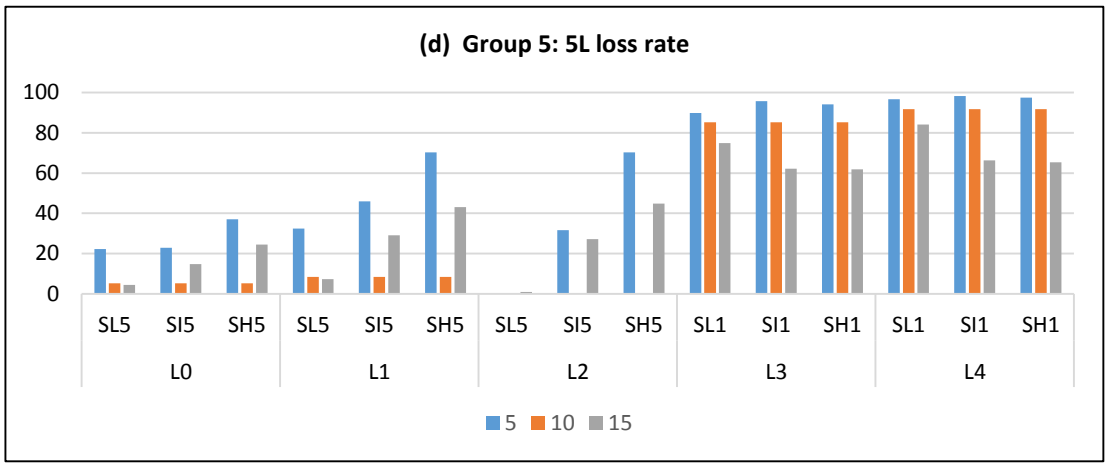


Figure 46: Average ERVL for Group5 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL5, SI5 and SH5) for a layered video with: (a) L = 3, (c) L = 5, and (e) L = 8 layers





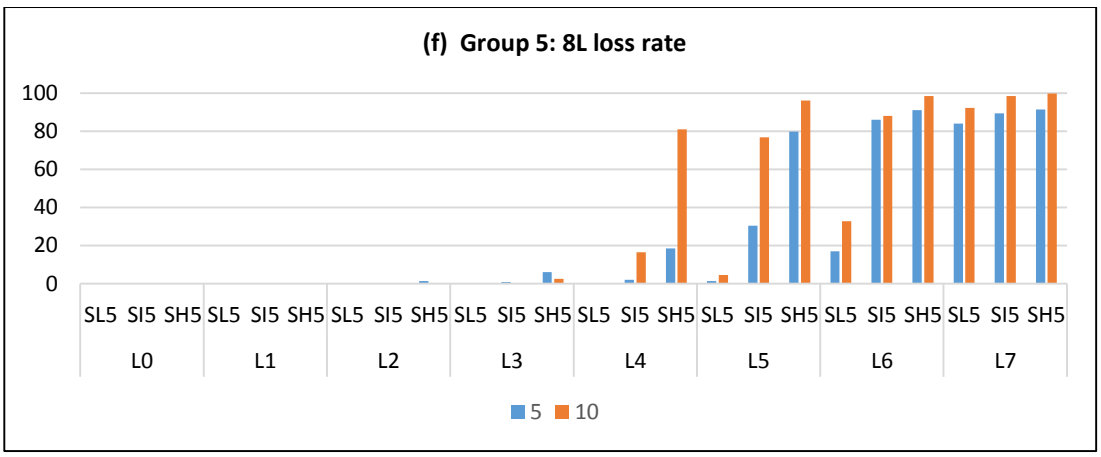
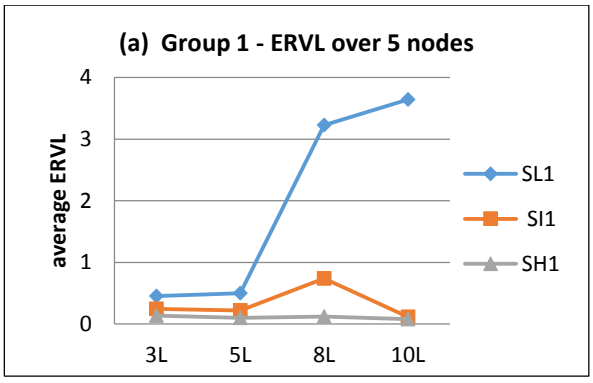
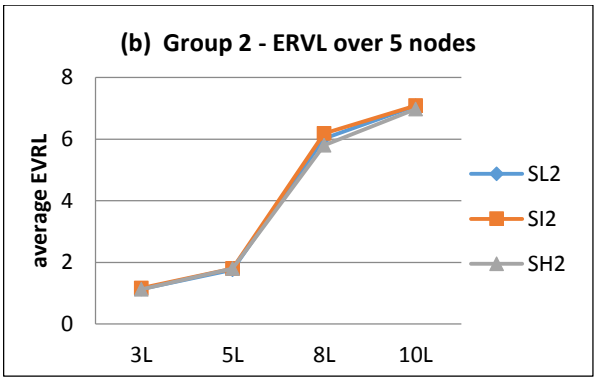


Figure 47: Average loss rate per each layer for Group5 experiment over 5, 10 and 15 nodes and with different background traffic scenarios (SL5, SI5 and SH5) for a layered video with: (b) L = 3, (d) L = 5, and (f) L = 8 layers.

Impact of Varying Different Layers over Fixed Number of Nodes.

The following set of results shown in Figure 48 demonstrates the ERVL performance metric considering layered video composed of L = 3, 5, 8 and 10 over five nodes when there are different concurrent background traffic loads transmitted in the wireless network as light, intermediate and heavy. In Group 1 and Group 4 experiments shown in Figure 48 (a) and (d), it can be clearly seen that ERVL increases when the number of video layers increased in case of light traffic load level in which 4 layers can be reconstructed when L=10. However, when there are intermediate and heavy traffic load levels delivered over ACs, the number of layers to be reconstructed is less than 1 layer in case of all video layers. While in Group2 and Group3 shown in Figure 48 (b) and (c), ERVL has the same value when varying number of traffic loads since the best effort traffic distributed over AC_1 and AC_0 does not affect video quality. Therefore, 7 layers can be expected to be reconstructed when L = 10, 6 layers are expected to be reconstructed when L = 8, 2 layers are expected to be reconstructed when L = 5 and 1 layer can be reconstructed when L = 3 layers. This could be well reasoned by the different distributions of video packets over the corresponding layers.





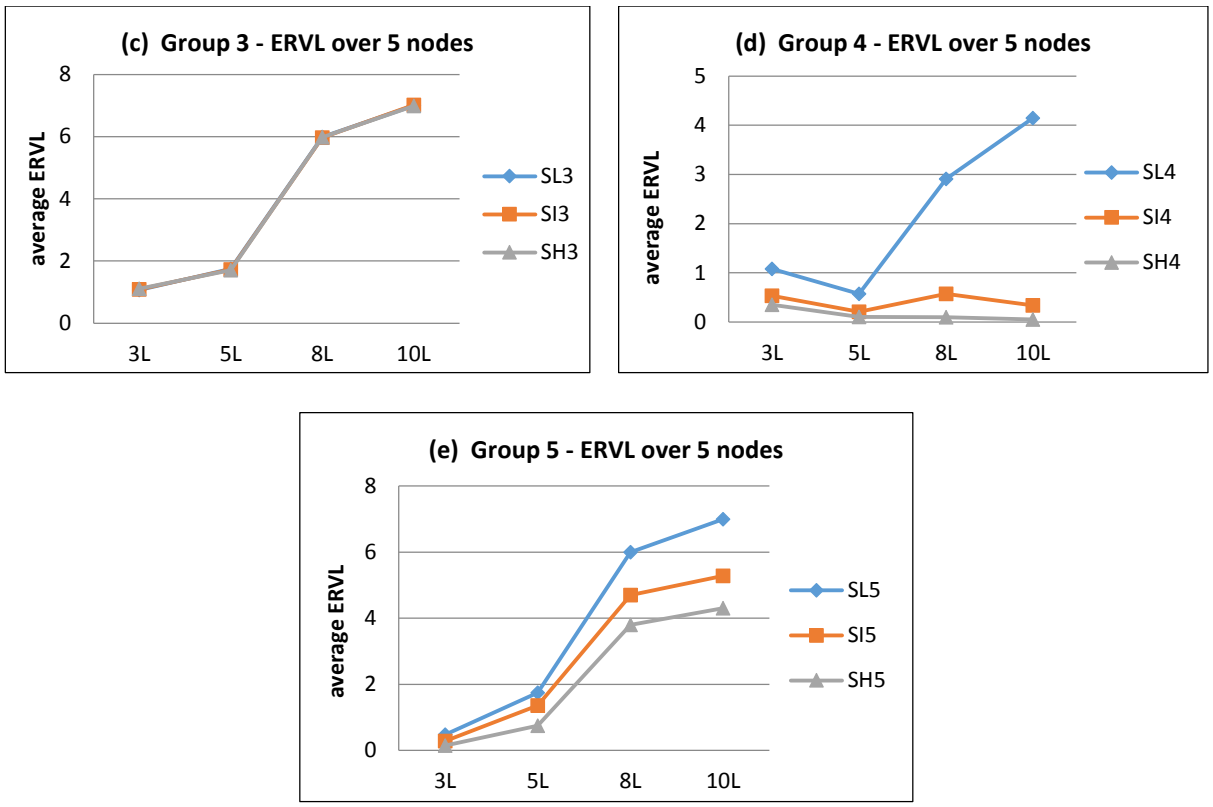


Figure 48: Average ERVL for various video layers (L = 3, 5, 8, and 10) over 5 nodes across different groups (a) Group 1, (b) Group 2, (C) Group 3, (d) Group 4, and (e) Group 5

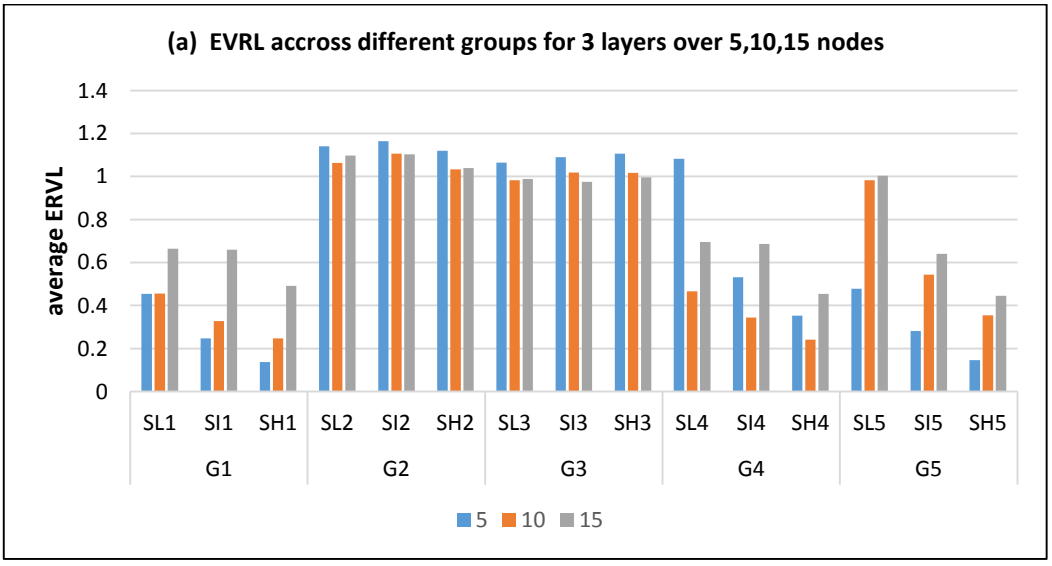
ERVL can be affected dramatically depending on layers' loss rate mapped into different ACs. Thus, heavy congestion in the AC level can result into high loss rate in video layers. For example, in Group1 and Group4 any layers mapped to AC_3, which has another ongoing background traffic, ERVL value is decreased severely. However, when there is no background traffic going into AC_3, ERVL has higher values as in Group2 and Group3 experiments in which the expected number of layers to be reconstructed is equal to more than 1 layer. In Group5 experiments, the ERVL increased as the number of video layers increased. Indeed, ERVL achieved higher values in SL5 than in SI5 and SH5.

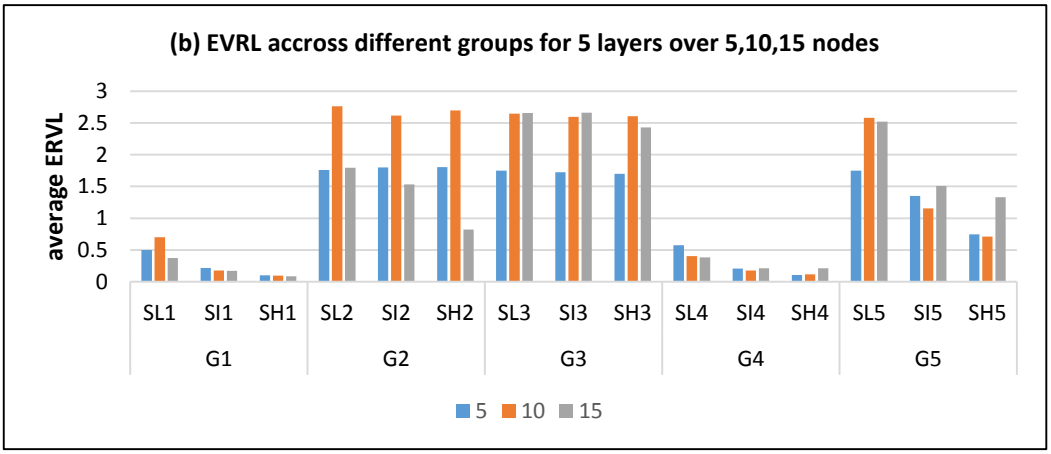
- **Summary**

Based on previously obtained results, the congestion level of ACs affects the video quality since all ACs are contending for the same wireless medium. The background traffic has a greater effect on overall ERVL value and layers loss rate ratio. The results show a variation in congestion level of ACs considering different number of cross traffic flows in term of high, intermediate and heavy level loads transmitted concurrently with video traffic distributed over those ACs.

The results shown in Figure 49 summarize the overall ERVL performance as described in the previous set of experiments that measure the performance of video delivery when there are other ongoing background traffics.

It is obvious that when all background traffic loads in any levels delivered into AC_3 besides the ongoing video traffic as in Group1 and Group4 experiments, the packet loss rate is high. This happened because most important video traffic (i.e. lower video layers) are going to the same AC which has a limited size. Therefore, the AC started to drop packets when it gets full causing a degradation of the video quality.





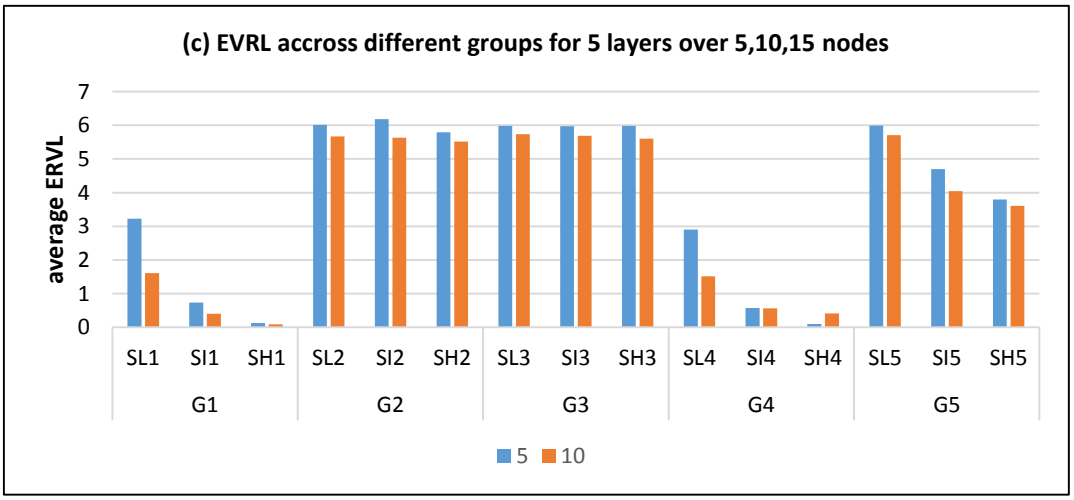


Figure 49: Average ERVL for all groups of experiments for layered video with (a) 3 layers, (b) 5 layers and (c) 8 layers

The effect of delivering a layered video composed of L layers over N nodes guarantees better QoS when there are different background load levels mapped to AC_1 and AC_0 as shown in Group2 and Group3. Since higher priority ACs are not congested in serving other traffics so important video layers distributed over those ACs have higher priority to be buffered and served first than lower priority ACs. Furthermore, lower priority ACs do not have the most important layers (base layer L0) since base layer should be always mapped to highest priority AC. Therefore, when the highest priority queue (i.e. AC_3) is extremely congested transmitting other types of data (i.e. Voice), no more video layers should be

mapped into it and the DSM algorithm can be started from AC_2 thus exploiting three ACs instead of four ACs.

As discussed previously, the optimal ERVL value can be obtained having layered video composed of L = 3, 5, and 8 layers in light load traffic level using:

- ✓ "map3" with permutation (1, 1, and 1) in case of 3 layers;
- ✓ "map6" with permutation (2, 1, 1 and 1) in case of 5 layers;
- ✓ "map14" with permutation (2, 2, 2, and 2) in case of 8 layers

Thus, best video delivery can be shown when exploiting all ACs. However, in Group1 and Group4, it has been shown that when there are other background traffics contending to gain access of channel resources, the optimal mapping that guarantees best ERVL value can be achieved when mapping all video layers into AC_3 only using "map1" with all video layers (i.e. L= 3, 5, and 8).

Overall, utilization of EDCA priority queues (ACs) is extremely efficient since when the DSM algorithm started with no adaption in which all layers are mapped to AC_3, half of the packets have been dropped in each layer caused by queue overflow. Nevertheless, when DSM made use of other alternative priority queues (i.e. AC_2, AC_1 and AC_0) and started to adapt video layers between them, the dropping rate of each video layer started decrease gradually as well as the ERVL video quality metric started to increase dramatically. In subsequent sections, the effect of changing the size of queue on value of EVRL is studied.

5.3.3.1 The Effect of Actual vs. AC-based layers Drop Rate on ERVL

According to Romdhani et al. [16] work, the dropping probability of each layer is calculated based on the dropping probability of the AC that it is assigned to. Their assumption is that the distribution of layers into ACs is statistically independent across all layers and therefore, drop rates of layers on ACs are equal to the drop rate of AC. In this research work, the actual number of dropped packets per layer is collected throughout

experiments. Hence, the drop rates for each layer is calculated based on these numbers. These readings in the case of multiple layers assigned to a specific AC, show differences with the drop rate of the AC itself. These differences are insignificant in some experiments while the differences are noticeable and therefore significant in other experiments. The overall value of the ERVL measure is influence by these differences in measuring the layers drop rates. Table 11 illustrates insignificant differences found in measuring the layers drop rate using the assigned AC drop rate and that of actual reading for different scenarios of experiments. While Figure 50 illustrates the effect demonstrated in Table 11 on ERVL.

Table 11: (a) The loss rate (%) for each layer and AC in two different scenarios, and (b) Actual difference between the measures of layers drop rate using the assigned AC drop rate

(a)

		L0	**L1**	**L2**	**L3**	**L4**	**AC_3**	**AC_2**	**AC_1**	**AC_0**
5n5L	SI1	0.795	0.801	0.64	0.816	0.837	0.797	0	0.178	0.196
10n5L	SH3	0.045	0.078	0	0.983	0.963	0.062	0	0.983	0.963

(b)

		Drop/AC_3	**Drop/ layers**	**L0 in AC_3**	**L1 in AC_3**
5n5L	SI1	9628 packets	9627 packets	0.57%	0.79%
10n5L	SH3	55 packets	55 packets	1.84%	1.82%

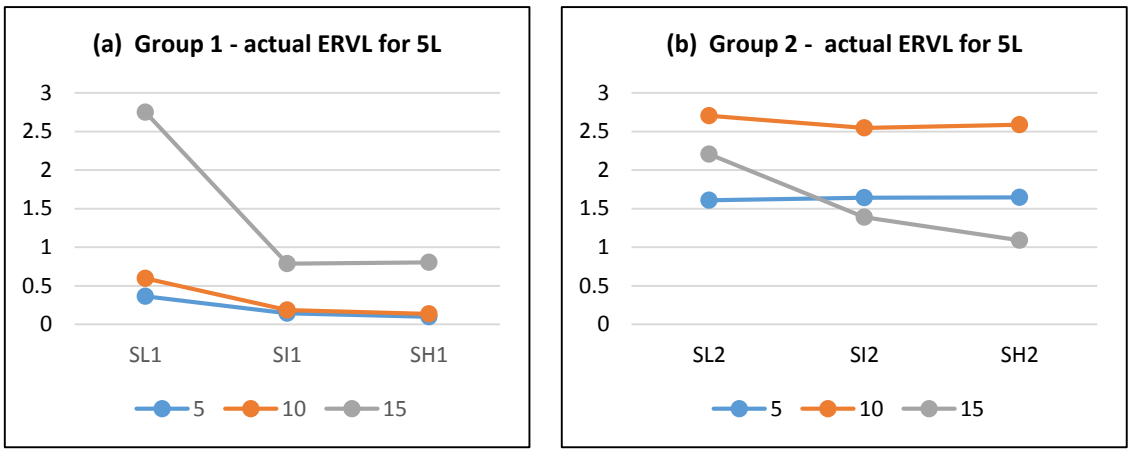


Figure 50: The actual ERVL measurement per layers for (a) Group1, and (b) Group2 experiments

On the other hand, Table 12 illustrates differences that are significant to measuring the layers drop rate based on AC drop rate and that found through experimental readings. The effect of these differences on ERVL is shown in Figure 51.

Table 12: (a) The loss rate of each layer, (b) The loss rate of each AC, (c) and (d) Actual difference between the measurement of layers drop rate using the assigned AC drop rate and the Layers drop rate

(a)

		L0	L1	L2	L3	L4	L5	L6	L7
5N8L	**SI1**	0.57	0.62	0	0	0.17	0.62	0.99	0.99

(b)

AC_3	AC_2	AC_1	AC_0
0.60	0	0.47	0.99

(c)

Drop/ AC_3	Drop/ Layers	L0 in AC_3	L1 in AC_3	Drop/ AC_2	Drop/ Layers	L2 in AC_2	L3 in AC_2
175	174	3.15%	1.89%	0	0	0.01%	0.01%

(d)

Drop/ AC_1	Drop/ Layers	L4 in AC_1	L5 in AC_1	Drop/ AC_0	Drop/ Layers	L6 in AC_0	L7 in AC_0
1642	1642	29.95%	14.97%	13779	13779	0.29%	0.15%

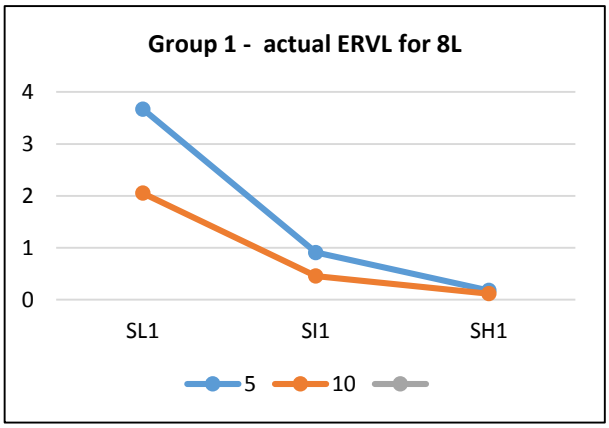


Figure 51: Actual ERVL measurement for 8L

There is a difference between the actual layers drop rate and the drop rate of an AC since the drop rate of AC that has multiple layers is not necessarily belongs to the same layer.

5.3.4 Additional Experiments

The main objective of DSM algorithm is to find the optimal mapping that guarantees the best delivery of a layered video over wireless network via exploiting all available IEEE 802.11e EDCA resources, which are represented as queues or ACs. The layered video is composed of base layer, which has the most important video packets to provide the basic quality of the video stream and multiple enhancement layers that are used to refine the video quality. Therefore, important video layers should always map to higher priority ACs (i.e. AC_3 and AC_2) to make the best use of channel resources while taking care of less significant video layers by mapping them into lower priority ACs (i.e. AC_1 and AC_0). Thus, it is interesting to study the effect of varying the size of each AC on DSM algorithm.

The purpose of this experiment is to examine the impact of having different AC sizes on a layered video composed of 5 layers. These layers are mapped into different ACs over 5 nodes. Therefore, set of scenarios with different AC sizes have been examined without ongoing background traffic and when there is an intermediate background traffic mapped to different ACs. These scenarios are represented as **SI1**, **SI2**, **SI3**, **SI4** and **SI5** that belong to different groups as shown in Table 5.

The results showed that changing each AC size does not affect the ERVL value when there is a video traffic transmitted over different ACs without having any background traffic as illustrated in Figure 52. Moreover, it does not affect choosing the best ERVL permutation since in all cases the ERVL value can be obtained using “map6” as shown in section 5.3.2.1.

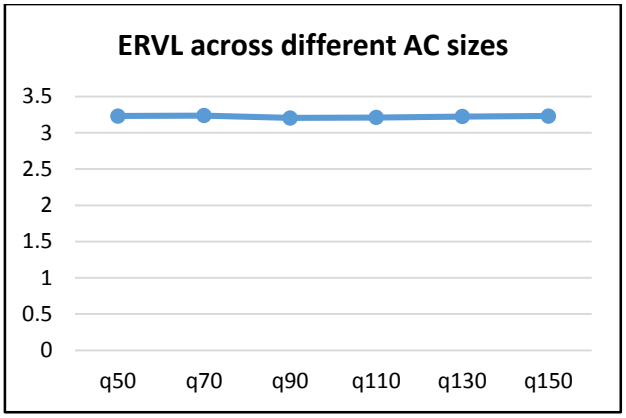
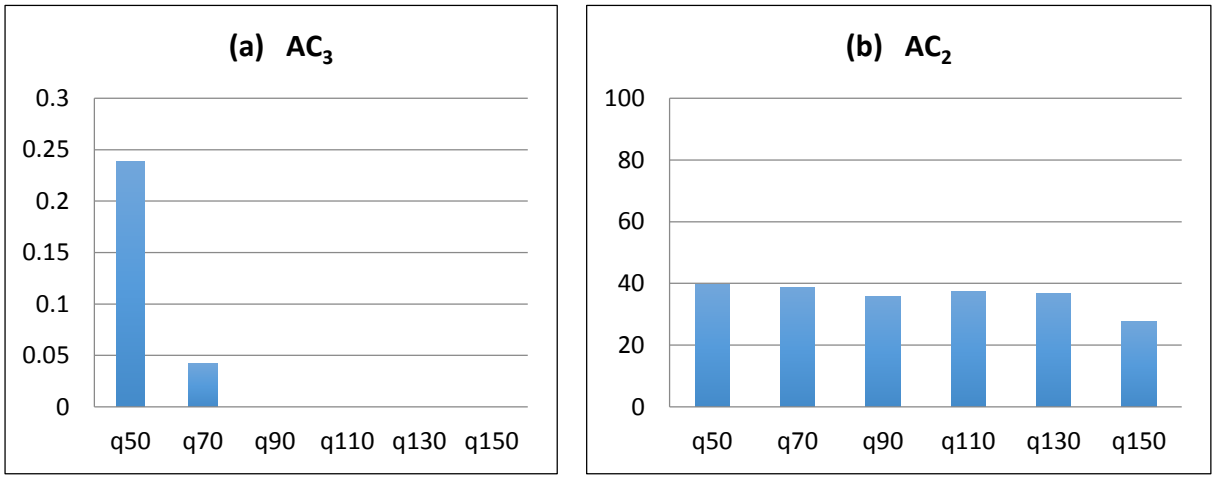


Figure 52: ERVL for a layered video with L = 5 over 5nodes across different AC sizes

Since the ERVL is not affected by the queue size when there is no ongoing background traffic, the loss rate for each AC is not affected as shown in Figure 53.



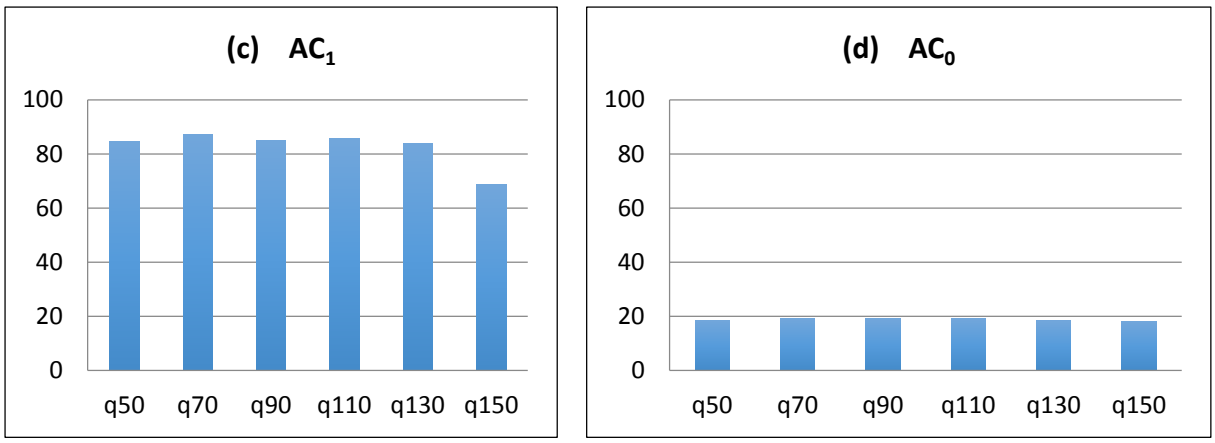


Figure 53: ACs space utilization of 802.11e EDCA (a) AC_3, (b) AC_2, (c) AC_1, and (d) AC_0

On the other hand, the same queue sizes have been adopted but when there are other intermediate traffics introduced together with the ongoing video traffic. The average loss rate of each AC and the average ERVL have been presented for each scenario. In the first scenario, ten concurrent flows of voice traffic are activated in intermediate traffic load level (i.e. "**SI1**").

It is clearly seen from Figure 54: (a) that changing the size of each AC does not affect the ERVL metric when there is an intermediate voice traffic mapped to AC_3. All voice traffic levels together with the video traffic have the same priority since all of them mapped to the same AC. Therefore, all traffics are contending to have access of the available bandwidth resulting high loss rate in each layer mapped to each AC caused by the high delay of each one. However, the highest priority queue (AC_2) has a zero-drop since it has the second priority to access the channel resources after AC_3 according to EDCA parameter set illustrated in Table 1. Moreover, since the optimal mapping is obtained using "map6" with permutation (2, 1, 1 and 1), therefore, only 1 layer is mapped to this AC according to its

permutation which results in reducing the level of congestion over this AC and ending with zero-drop rate.

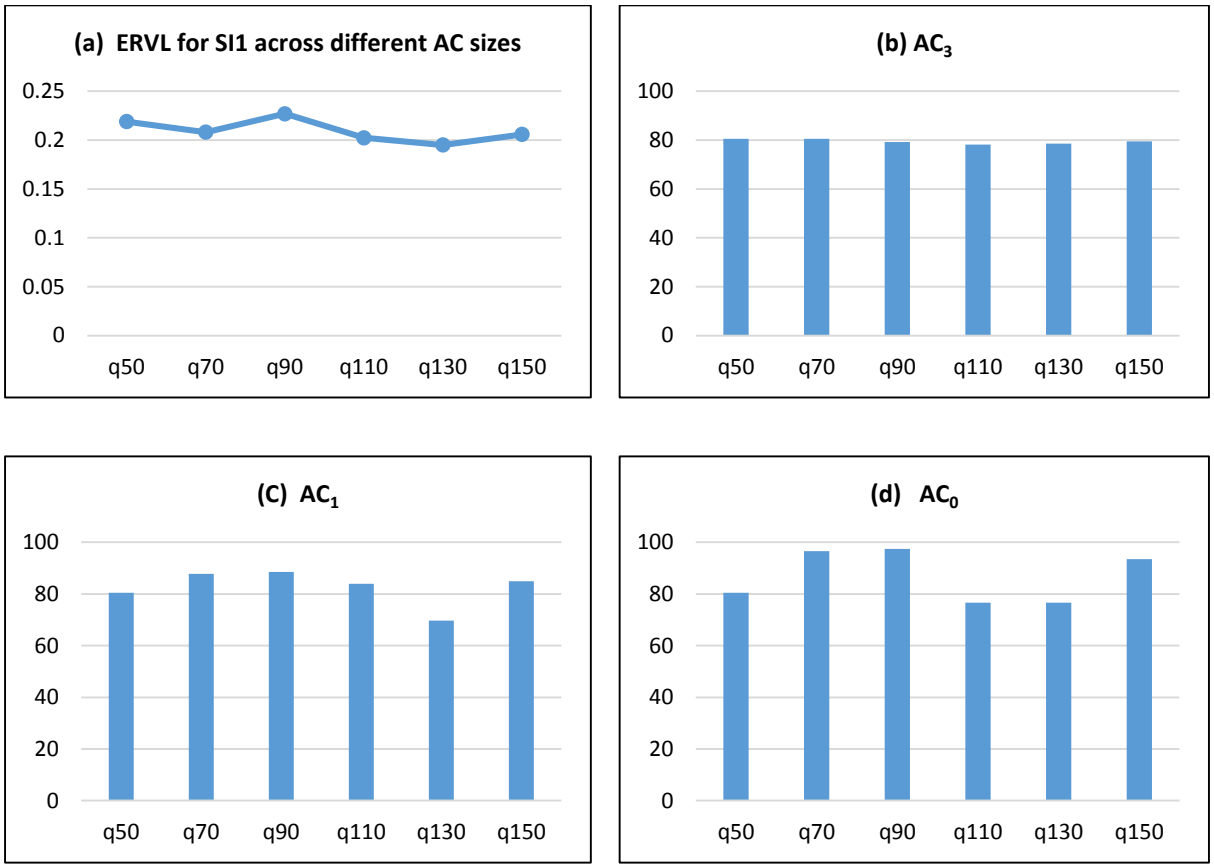


Figure 54: (a) average ERVL, (b) AC_3, (c) AC_1 and (d) AC_0 for SI1 scenario across different AC sizes over 5 nodes

It is also shown from the average ERVL and loss rate of each AC in Figure 55 that a similar congestion level of each AC is affecting both measured thus not affecting the change of AC size. In addition, changing the AC size does not affect the selection of optimal mapping since in both scenario cases, **SI1** and **SI4**, the optimal mapping is obtained using "map6".

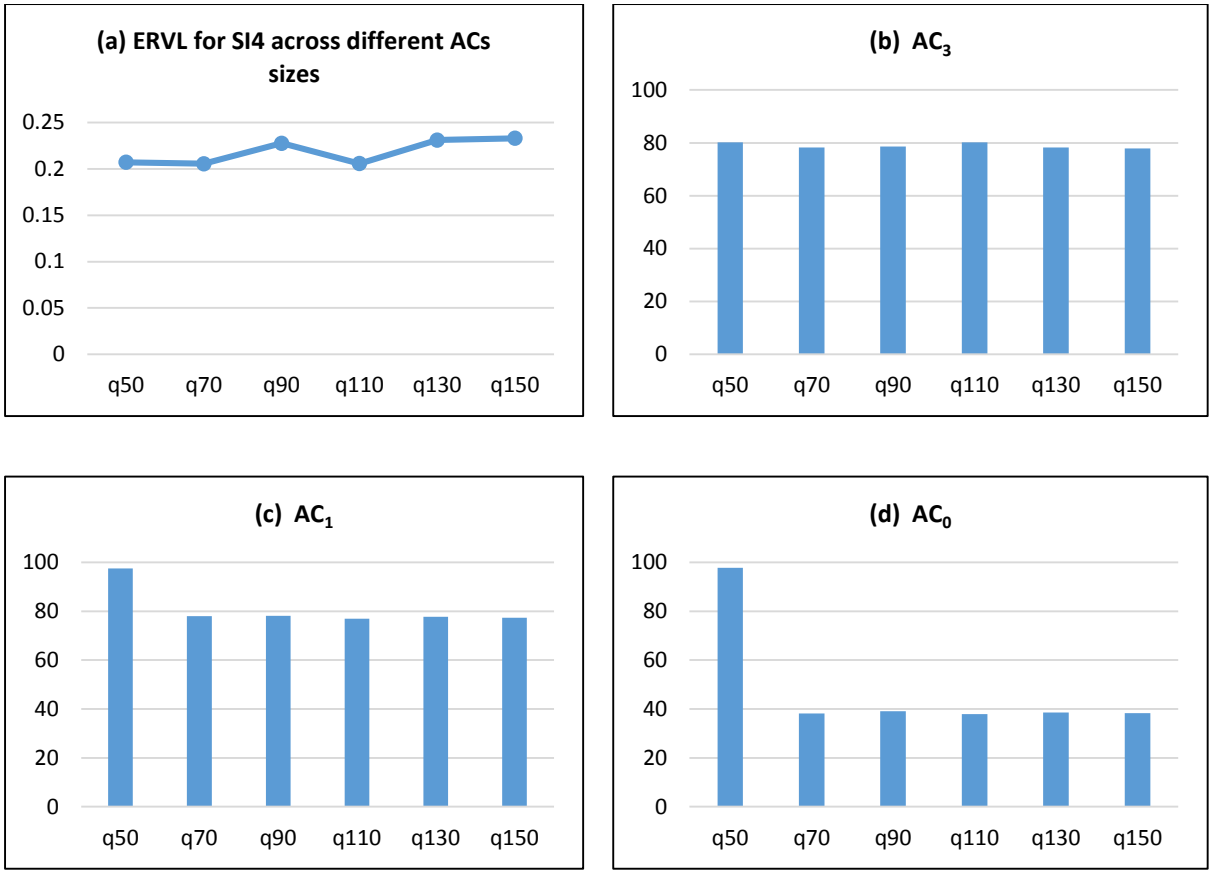


Figure 55: (a) average ERVL, (b) AC_3, (c) AC_1 and (d) AC_0 for SI4 scenario across different AC sizes over 5 nodes

In case of SI2 scenario, ten concurrent flows of HTTP traffic are activated in intermediate traffic load level together with the ongoing video traffic. These flows are mapped to the best effort AC (i.e. AC_1). Based on the obtained results, when the AC size is 50 the expected reconstructed video layers is 1.2 out of 5 layers as shown in Figure 56. When there is a 40% increase of the queue size, the ERVL increases sharply from 1.6 to 3, it is in fact an increase of approximately 50% of ERVL value. However, there was no significance increase of ERVL value while increasing the AC size after the capacity 70 of the AC size as it remains steady over all other AC sizes. It is obvious that an improvement in the loss ratio of bas layer has been achieved due to light load distributed over AC_3 and AC_2. Therefore,

increasing the AC size affects the overall performance of ERVL measurements as it decreases the loss rate of each layers. Moreover, since the intermediate background traffic is mapped to AC_1 besides the video layers that are assigned to this AC according to their permutations as in Table 3, this AC suffers severely from the highest loss rate among all ACs.

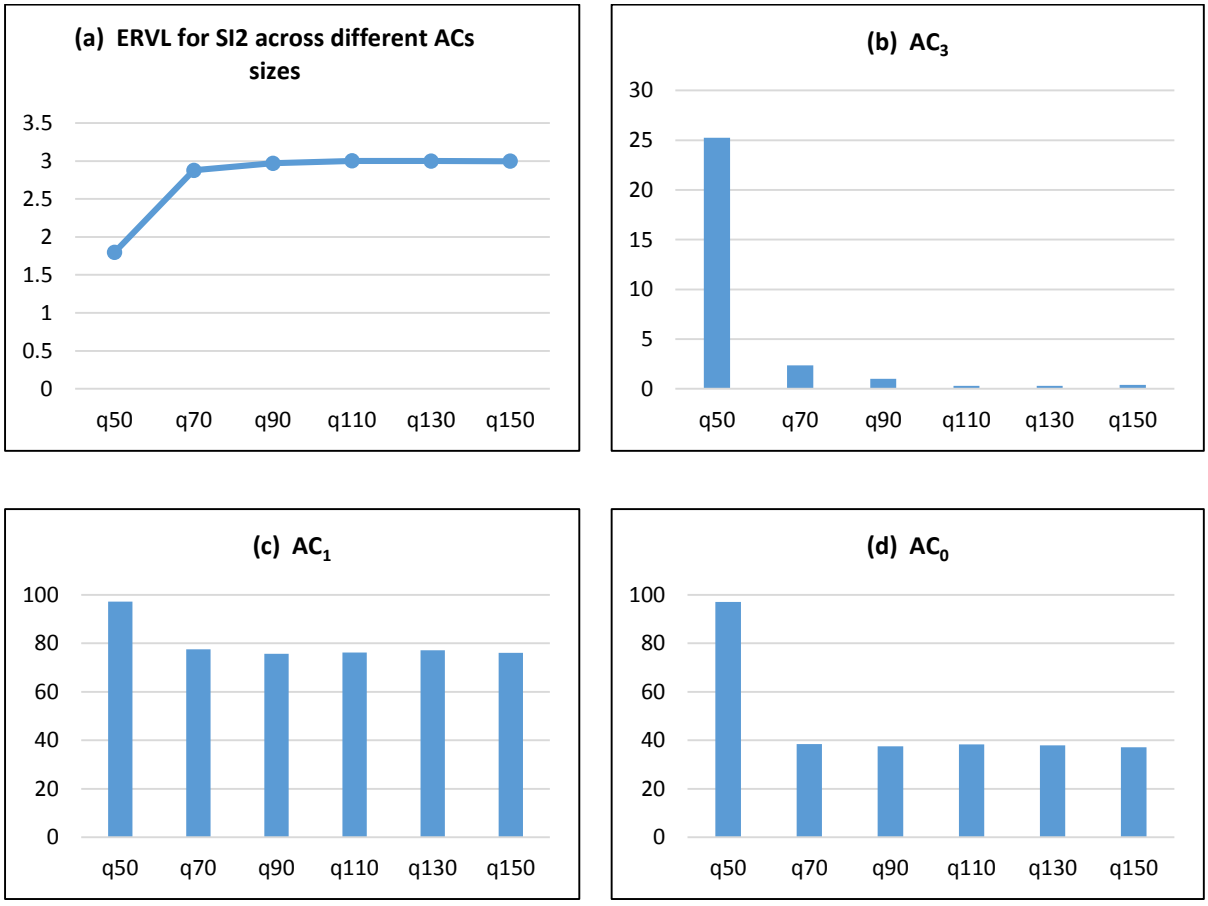


Figure 56: (a) average ERVL, (b) AC_3, (c) AC_1 and (d) AC_0 for SI2 scenario across different AC sizes over 5 nodes

The obtained average ERVL and loss rate of each AC in Figure 57 look quite similar to that in SI2 scenario since both AC_3 and AC_2 are the least congested in these scenarios. For example, AC_3 has loss rate equals to 25% and AC_2 has zero loss rate since it has only 1 layer assigned to it. As a result, changing the AC size in each scenario does not affect choosing the

ERVL value. However, it reduces the loss rate of each AC. The optimal ERVL in both scenarios can be obtained using "map6" which guarantees maximum ERVL value.

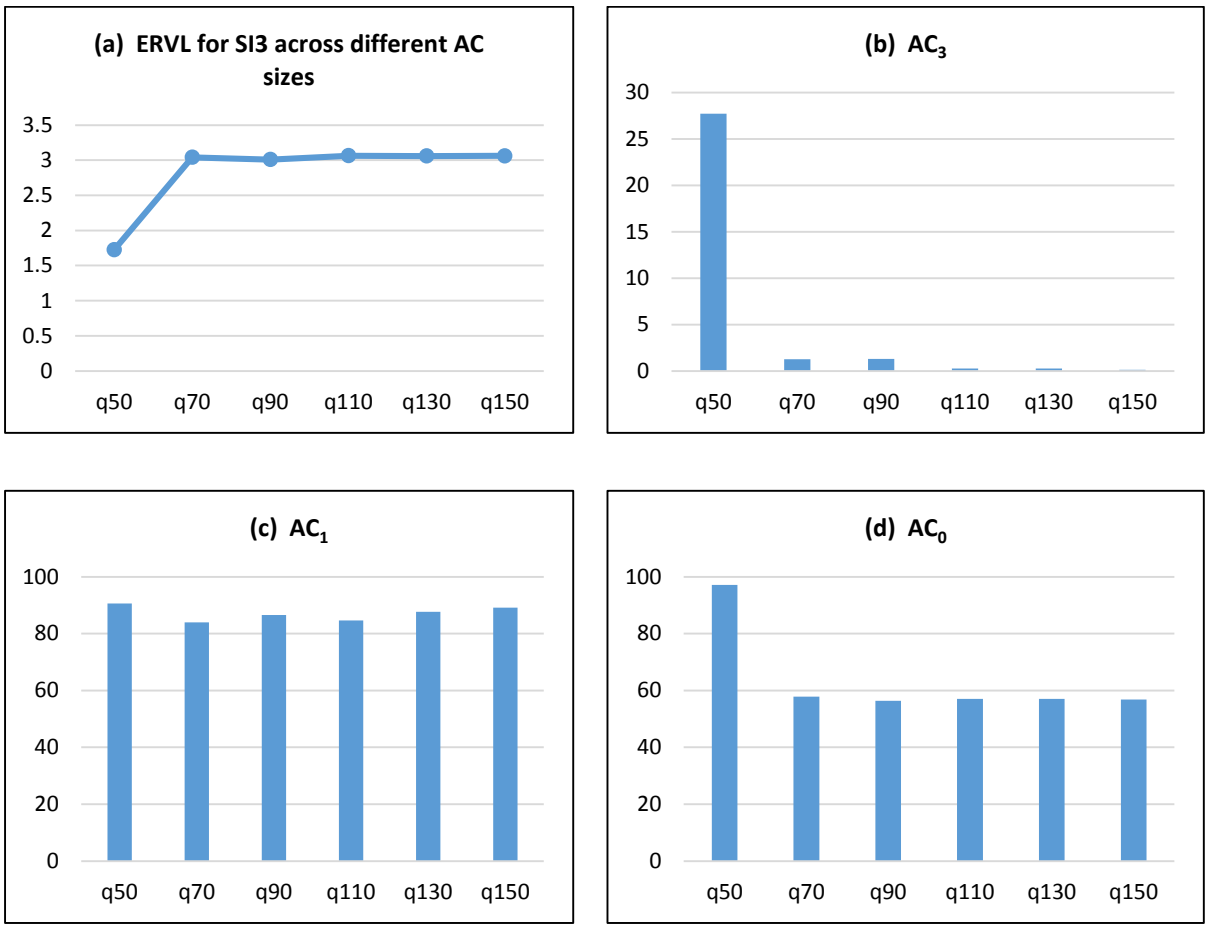


Figure 57: (a) average ERVL, (b) AC_3, (c) AC_1 and (d) AC_0 for SI3 scenario across different AC sizes over 5 nodes

Finally, varying the queue size of each AC has been tested when there is another ongoing video traffic mapped to AC_2 as in SI5 scenario. Based on the obtained results shown in Figure 58, the ERVL increases as the AC size increases in which it starts with 1.3 when the queue size equals to 50 and reaches 2 when the queue size equals 150. Nevertheless, this change of ERVL value is considered minimal while having a big queue size difference.

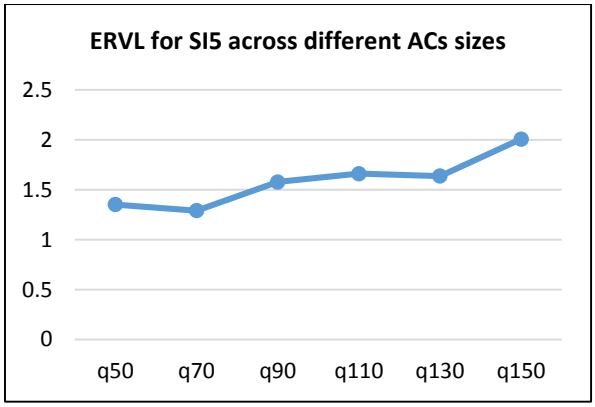


Figure 58: ERVL over 5nodes across different AC sizes for SI5 scenario

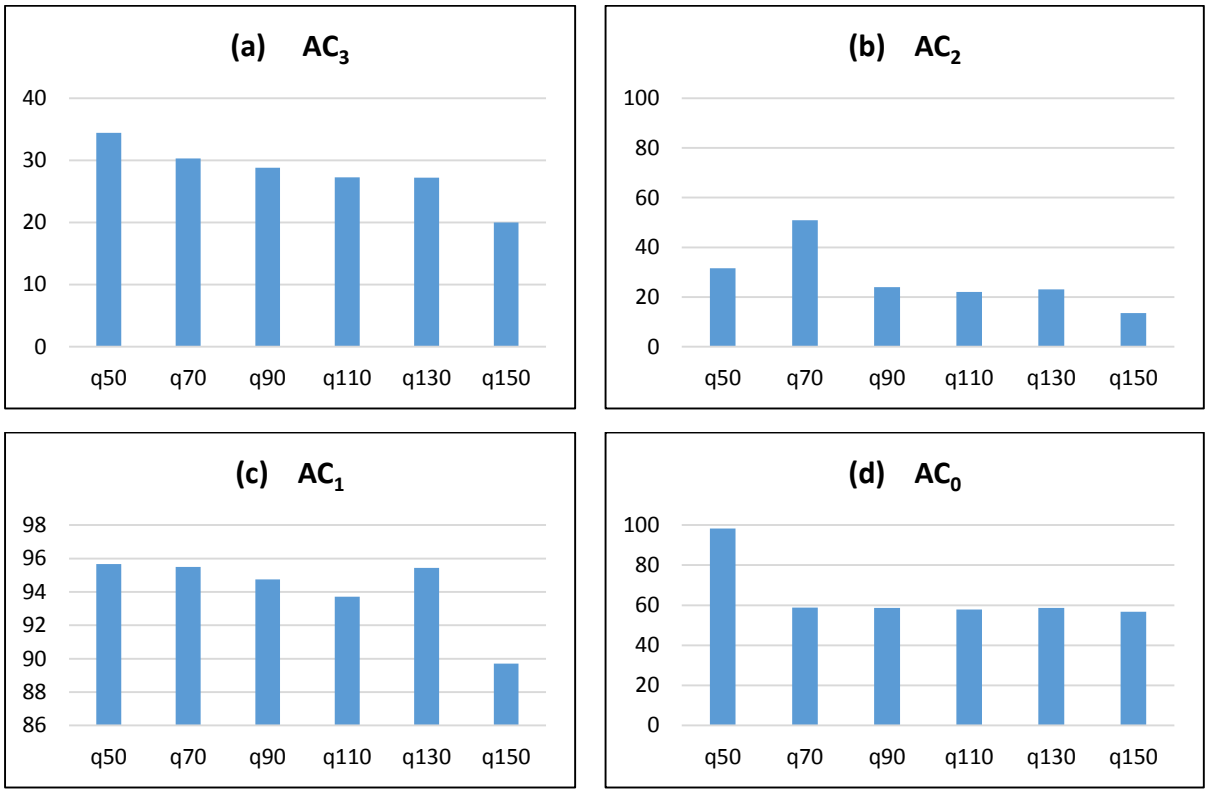


Figure 59: (a) average ERVL, (b) AC_3, (c) AC_1 and (d) AC_0 for SI5 scenario across different AC sizes over 5 nodes

Consequently, the loss rate of each AC with variant sizes is decreased slightly when the queue size increased as shown in Figure 59.

6 Conclusion, Challenges and Future Work

This chapter provides a summary for thesis core work, obtained findings, main challenges considering this work and finally a direction for future work.

6.1 Conclusion

A wireless medium is a multi-access medium in which different traffic types compete to have access to available resources. This research work realized a precise DSM algorithm implementation in NS2 with cross-layer design. Such implementation adheres to detailed specifications of network standards addressed by NS2 implementations. The goal of this step is to provide a research-oriented implementation model to facilitate further research on this area with minimal needed effort by researches. In the conducted research experiments, nodes in turn map its video layers dynamically into appropriate EDCA ACs considering different possible mappings. The performance of utilizing DSM algorithm over EDCA access schemes has been studied with a variety of ongoing traffic patterns and network load conditions. Exact ERVL values are measured based on readings of dropping probabilities per layer rather than AC dropping probability. The obtained values are compared to those obtained using AC dropping probability instead. Additional experiments studying the effect of changing the queue size for all AC on the ERVL metric are also reported.

Trivially, the obtained results show that the ERVL value is affected directly by the loss rate for each AC. The lowest the loss rate per AC, the highest ERVL value and the best video quality is achieved. DSM performs well in the case of no background traffic being distributed over ACs besides the ongoing layered video traffic. DSM dynamically allocated all video layers to the most appropriate ACs based on their best chosen permutations chosen from all possible permutations shown in Table 3. However, the performance of DSM is inconsistent when different traffic patterns are introduced into the experiments. This effect is highly noticeable when any of these traffic patterns is mapped to the ACs that are expected to

have video layers based on the best chosen mapping that is currently adopted by nodes. For example, the best values of ERVL are those when lower layers are mapped to AC_3 and no other traffic is mapped to this AC. In these cases, as in Group2 and Group3 experiments, AC_3 is having almost zero dropped packets. However, ERVL is strongly affected when other traffic is mapped to higher priority ACs. This is clearly shown in Group1, Group4 and Group5 experiments. More details about experiment's results are shown in the summary of section 5.3.3. Generally speaking, the use of DSM shows that the expected number of layers to be reconstructed has a high value when all alternate priority queues are utilized. Finally, a 40% increase in the ACs queue size shows significant improvement on ERVL while an increase of the queue size beyond this value has very little significance on ERVL.

6.2 Challenges

Implementing a cross-layer design model with the rigid network stack structure of NS2 is a challenging task. Many alternate solutions are attempted to provide a structured, bugs-free, extendable, and realistic model to be conveniently used by researches in this area. The model defines structured communications between different layers that suits the implementation of a cross-layer design. Intensive tests with debugging scenarios have been conducted with professional debugging tools to validate the accuracy and quality of the implemented model.

The second challenge is related to the design of the different scenarios for the experiment set adopted by this thesis research work. This includes selecting appropriate network traffic models that realistically represent a variety of traffic types such as Voice, Video and internet traffic.

Collecting intensive simulation readings and analyzing them is the third challenge of this research work.

The forth challenge of this research work is related to reporting simulation results of highest possible accuracy which is achieved by reporting average of obtained results for each conducted experiments iterated for large number of times. Finally, exploring the derivation of the DSM algorithm was a challenging task, which benefited this work and directed to evaluating the accuracy of measuring the ERVL based on the exact drop rate of layers instead of the drop rate of the ACs. A lot of future work is also inspired this derivation.

6.3 Future work

Having such extendable and flexible implementation of this model is convenient to conduct additional experimentations as well as introduce extension to DSM model or even implement in a similar way other algorithms by utilizing this model as a framework to do so. Some of the intended research work to be conducted in the nearest future is shown below:

- Measuring the effect of varying EDCA parameters on the overall performance of the network.
- Injecting noise and interruption of service in other abnormal events during the run time of the experiments.
- Experimenting with different kind of video encoding techniques other than SVC.
- Conducting experiments on a large-scale network by utilizing powerful computational servers.
- Include other performance metrics of interest such as throughput, delay and SNR.

References

[1] C. Mai, “Cross-layer Adaptive H . 264 / AVC Streaming over IEEE 802 . 11e Experimental Testbed,” no. Vcl, 2010.

[2] C. W. Chen and F. Zhengyong, “Video over IEEE802 . 11 Wireless LAN : A Brief Survey,” pp. 1–19, 2013.

[3] A. A. Bin-salem and T. C. Wan, “Survey of Cross‐layer Designs for Video Transmission Over Wireless Networks,” pp. 229–247, 2012.

[4] J. Kim, H. Tode, K. Murakami, and A. Q. M. A. C. Protocol, “Service-based Rate Adaptation Architecture for IEEE 802 . 11e QoS Networks,” pp. 3341–3345, 2005.

[5] V. A. Siris, “Resource Control for the EDCA Mechanism in Multi-Rate IEEE 802 . 11e,” 2006.

[6] S. Takeuchi, K. Sezaki, and Y. Yasuhiko, “Dynamic adaption of Contention Windows Sizes in IEEE 802.11e Wireless LAN,” pp. 659–663, 2005.

[7] J. Majkowski and F. C. Palacio, “Enhanced TXOP scheme for efficiency improvement of WLAN IEEE 802 . 11e,” pp. 1–5, 2006.

[8] J. Majkowski and F. C. Palacio, “Dynamic TXOP configuration for Qos enhancement in IEEE 802 . 11e wireless LAN,” pp. 1–5.

[9] A. S. M. Tariq and K. Perveen, “Analysis of Internal Collision and Dropping Packets Characteristics of EDCA IEEE802 . 11e Using NS-2 . 34 Simulator,” vol. I, pp. 20–23, 2010.

[10] E. H. Putra, E. Supriyanto, J. Din, and H. Satria, “Cross Layer Design of Wireless LAN,” no. July, 2011.

[11] J.-L. H. J.-L. Hsu and M. van der Schaar, “Cross Layer Design and Analysis of Multiuser Wireless Video Streaming Over 802.11e EDCA,” *IEEE Signal Process. Lett.*, vol. 16, 2009.

[12] C. H. F. C. H. Foh, Y. Z. Y. Zhang, Z. N. Z. Ni, J. C. J. Cai, and K. N. N. K. N. Ngan, "Optimized Cross-Layer Design for Scalable Video Transmission Over the IEEE 802.11e Networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, 2007.

[13] Y. X. Y. Xiao, X. D. X. Du, F. H. F. Hu, and J. Z. J. Zhang, "A Cross-Layer Approach for Frame Transmissions of MPEG-4 over the IEEE 802.11e Wireless Local Area Networks," *2008 IEEE Wirel. Commun. Netw. Conf.*, 2008.

[14] H. Z. H. Zhang, Y. Z. Y. Zheng, M. Khojastepour, and S. Rangarajan, "Cross-layer optimization for streaming scalable video over fading wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 28, 2010.

[15] H.-L. C. H.-L. Chen, P.-C. L. P.-C. Lee, and S.-H. H. S.-H. Hu, "Improving Scalable Video Transmission over IEEE 802.11e through a Cross-Layer Architecture," *2008 Fourth Int. Conf. Wirel. Mob. Commun.*, 2008.

[16] L. Romdhani and A. Mohamed, "Sequential QoS Mapping Model for Multi-layer Video Delivery in 802 . 11e Wireless Networks," *Int. Conf. Adv. Comput. Sci. Eng.*, vol. 2nd, pp. 1–5, 2013.

[17] L. Romdhani and A. Mohamed, "An analytic study of a distributed EDCA-based QoS mapping for layered video delivery in WLAN," 2013.

[18] I. Richardson, "Introduction to Video Coding | Vcodex," 2013. [Online]. Available: http://www.vcodex.com/introduction.asp. [Accessed: 06-Aug-2015].

[19] C.-H. Ke, "myEvalSVC: an Integrated Simulation Framework for Evaluation of H.264/SVC Transmission," *KSII Trans. Internet Inf. Syst.*, vol. 6, no. 1, pp. 379–394, 2012.

[20] H. Sun, A. Vetro, and J. Xin, "An overview of scalable video streaming," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 2, pp. 159–172, Feb. 2007.

[21] W.-P. Lai and E.-C. Liou, "A novel cross-layer design using comb-shaped quadratic

packet mapping for video delivery over 802.11e wireless ad hoc networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2012, no. 1, p. 59, 2012.

[22] Y. Wang, a R. Reibman, and S. Lin, "Multiple Description Coding for Video Delivery," *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, 2005.

[23] I. Standard, *Supplement to IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Speci fi c requirements — Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer*, vol. 1999. 1999.

[24] I. Standard, *IEEE Standard 802.11n, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throupt*. 2009.

[25] L. A. N. Man, S. Committee, and I. Computer, *IEEE Standard 802.11ac, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4 : Enhancements for Very High Throughput for*. 2012.

[26] L. A. N. Man, S. Committee, and I. Computer, *IEEE Standard 802.11ad, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3 : Enhancements for Very High Throughput in the 60 GHz Band IEEE Computer Society*, vol. 2012, no. December. 2012.

[27] I. Standard, *IEEE Standard 802.11e, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, vol. 2005, no. Reaff 2003. 2005.

[28] L. A. N. Man, S. Committee, and I. Computer, *IEEE Standard 802.11aa, Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2 : MAC Enhancements for Robust Audio Video Streaming IEEE Computer Society*, vol. 2012, no. May. 2012.

[29] E. Charfi, L. Chaari, and L. Kamoun, “PHY/MAC Enhancements and QoS Mechanisms for Very High Throughput WLANs: A Survey,” *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 1714–1735, 2013.

[30] A. Ganz, Z. Ganz, and K. Wongthavarawat, *Multimedia Wireless Networks: Technologies, Standards, and QoS*. Prentice Hall PTR, 2003.

[31] Q. Ni, L. Romdhani, and T. Turletti, “A survey of QoS enhancements for IEEE 802.11 wireless LAN,” *Wirel. Commun. Mob. Comput.*, vol. 4, no. 5, pp. 547–566, 2004.

[32] S. Choi, S. S. N, J. Prado, and S. Mangold, “IEEE 802 . 11e Contention-Based Channel Access (EDCF) Performance Evaluation,” *Proc. IEEE ICC*, vol. 2, pp. 1–6.

[33] N. Based, S. A. Tariq, and F. Granelli, “Performance Analysis of Wireless Ad-Hoc Network Based on EDCA IEEE802.11e,” pp. 33–36, 2010.

[34] Y. Shi, N. Zhou, H. Du, and J. Xu, “Scalable video transmission with quality layers over IEEE 802.11e WLAN through a cross-layer design,” *2011 Int. Conf. Comput. Probl.*, pp. 93–96, 2011.

[35] S. M. Padle and V. Mendre, “CROSS LAYER (APPLICATION AND MEDIUM ACCESS CONTROL) DESIGN AND OPTIMIZATION OF WIRELESS NETWORK FOR,” 2014.

[36] B. Fu, Y. Xiao, S. Member, H. J. Deng, and H. Zeng, “A Survey of Cross-Layer Designs in Wireless Networks,” vol. 16, no. 1, pp. 110–126, 2014.

[37] S. Mantzouratos, G. Gardikis, H. Koumaras, and A. Kourtis, “Survey of cross-layer proposals for video streaming over Mobile Ad hoc Networks (MANETs),” *2012 Int. Conf. Telecommun. Multimed.*, pp. 101–106, Jul. 2012.

[38] T. Sutinen, J. Vehkaperä, E. Piri, and M. Uitto, “Towards ubiquitous video services through scalable video coding and cross-layer optimization,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2012, no. 1, p. 25, 2012.

[39] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*, 2nd ed., vol. 2. 2009.

[40] Y. Xiao, "Performance Analysis of Priority Schemes for IEEE 802.11 and IEEE 802.11e Wireless LANs," vol. 4, no. 4, pp. 1506–1515, 2005.

[41] D. Singh and H. Rathore, "Dynamic Adaptive Cross layer mapping mechanism for video transmission over wireless networks," *2011 Int. Conf. Emerg. Trends Networks Comput. Commun.*, pp. 205–208, Apr. 2011.

[42] X. Li, T. Ren, and J. Xu, "A cross-layer design for transmission of scalable H.264 video over IEEE 802.11e networks," *Int. Conf. Comput. Probl.*, pp. 306–309, 2010.

[43] A. Ksentini, M. Naimi, and A. Gueroui, "Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture," *IEEE Commun. Mag.*, vol. 44, 2006.

[44] C.-H. Lin, C.-K. Shieh, C.-H. Ke, N. K. Chilamkurti, and S. Zeadally, "An adaptive cross-layer mapping algorithm for MPEG-4 video transmission over IEEE 802.11e WLAN," *Telecommun. Syst.*, vol. 42, no. 3–4, pp. 223–234, Jul. 2009.

[45] H. C. CHEN, D. H. GUO, and H. M. FENG, "DYNAMIC CROSS-LAYER ADAPTIVE MAPPING ALGORITHM DESIGN FOR MPEG-4 VIDEO APPLICATIONS IN WIRELESS NETWORK," 2014.

[46] F. Sherman, "The Disadvantages of Centralized Internet | eHow." [Online]. Available: http://www.ehow.com/info_12129489_disadvantages-centralized-internet.html. [Accessed: 19-Aug-2014].

[47] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *DISTRIBUTED SYSTEMS Concepts and Design*, 5th ed. USA: Pearson, 2011.

[48] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," vol. 18, no. 3, pp. 535–547, 2000.

[49] T. L. Christian.Charras, “Brute Force algorithm,” 2014. [Online]. Available: http://www-igm.univ-mlv.fr/~lecroq/string/node3.html. [Accessed: 01-Jan-2016].

[50] Teerawat Issariyakul, “Mobile Networking: Regular Nodes and Mobile Nodes,” 2013. [Online]. Available: http://ns2ultimate.tumblr.com/. [Accessed: 18-Aug-2015].

[51] R. Stanley, “Enumerative Combinatorics 2,” vol. 62, no. March, pp. 1–4, 1999.

[52] M. Li, Z. Chen, and Y. P. Tan, “Cross-layer optimization for SVC video delivery over the IEEE 802.11e wireless networks,” *J. Vis. Commun. Image Represent.*, vol. 22, pp. 284–296, 2011.

[53] S. Wietholter and C. Hoene, “Design and Verification of an IEEE 802.11e EDCF simulation model in ns-2.26,” *Telecommun. Netw. Gr.*, p. 44, 2003.

[54] J. Schiller, *Mobile Communications*, 2nd ed. New York, 2003.

[55] T. Henderson, “Applications objects,” *Information Science Institue (isi)*, 2011. [Online]. Available: http://www.isi.edu/nsnam/ns/doc/node516.html. [Accessed: 19-May-2014].

[56] A. K. Karapantelakis, “A study of video and voice traffic over an 802 . 11e wireless network .,” University of CRETE, 2005.

[57] P. Antoniou, V. Vassiliou, and A. Pitsillides, “Delivering Adaptive Scalable Video over the Wireless Internet,” *Components*, 2007.

Appendix A: Best mapping for video of different vide layers over 5, 10 and 15 nodes

	Scenario Name	AC3	AC2	AC1	AC0	3L 5N	3L 10N	3L 15N	5L 5N	5L 10N	5L 15N	8L 5N	8L 10N
Group 1	SL1	5 Voice	1 Video			map3	map3	map1	map6	map6	map6	map14	map14
	SI1	10 Voice	1 Video			map3	map1	map1	map6	map6	map6	map14	map14
	SH1	20 Voice	1 Video			map3	map1	map1	map1	map1	map1	map14	map1
Group 2	SL2	1 Voice	1 Video	5 HTTP		map3	map3	map3	map6	map6	map6	1:1 M5: M11	map8
	SI2	1 Voice	1 Video	10 HTTP		map3	map3	map3	map6	map6	map6	M5	map8
	SH2	1 Voice	1 Video	20 HTTP		map3	map3	map3	map6	map6	map6	1:1 M8: M12	map8
Group 3	SL3	1 Voice	1 Video	1 HTTP	5 FTP	map3	map3	map3	map6	map6	map6	map11	map11
	SI3	1 Voice	1 Video	1 HTTP	10 FTP	map3	map3	map3	map6	map6	map6	map11	map11
	SH3	1 Voice	1 Video	1 HTTP	20 FTP	map3	map3	map3	map6	map6	map6	map11	map11
Group 4	SL4	5 Voice	1 Video	5 HTTP	5 FTP	map3	map3	map1	map6	map6	map6	map14	map14
	SI4	10 Voice	1 Video	10 HTTP	10 FTP	map3	map1	map1	map6	map6	map6	map14	map14
	SH4	20 Voice	1 Video	20 HTTP	20 FTP	map3	map1	map1	map1	map1	map1	map14	map1
Group 5	SL5	1 Voice	1 Video	1 HTTP	1 FTP	map3	map3	map3	map6	map6	map6	map11	map11
	SI5	1 Voice	2 Video	1 HTTP	1 FTP	map3	map3	1:1 M3: M1	map6	map6	map6	map8	map13
	SH5	1 Voice	3 Video	1 HTTP	1 FTP	map3	map3	map3	map6	map6	map6	map13	map14