Scalable and Network Aware Video Coding for Advanced Communications over Heterogeneous Networks

A thesis submitted for the degree of

Doctor of Philosophy

in

Electronic & Computer Engineering

by

Sanusi MUHAMMAD

Department of Electronic & Computer Engineering

Brunel University

March 2013

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is the true copy of the thesis, including any required final versions as accepted by my examiners.

I understand that my thesis work may be made electronically available to the public and therefore I authorise Brunel University to make available electronically to individual or institutions for the purpose of scholarly research.

Signature…………………………………………………

Sanusi Muhammad

Date …………………………..

I further authorise Brunel University to reproduce this thesis by means of photocopying or by any other means at the request of other institutions or individuals for the purpose of scholarly research.

Signature………………………………………………...

Sanusi Muhammad

Date…………………………….

Abstract

This work addresses the issues concerned with the provision of scalable video services over heterogeneous networks particularly with regards to dynamic adaptation and user's acceptable quality of service.

In order to provide and sustain an adaptive and network friendly multimedia communication service, a suite of techniques that achieved automatic scalability and adaptation are developed. These techniques are evaluated objectively and subjectively to assess the Quality of Service (QoS) provided to diverse users with variable constraints and dynamic resources. The research ensured the consideration of various levels of user acceptable QoS The techniques are further evaluated with view to establish their performance against state of the art scalable and non-scalable techniques.

To further improve the adaptability of the designed techniques, several experiments and real time simulations are conducted with the aim of determining the optimum performance with various coding parameters and scenarios. The coding parameters and scenarios are evaluated and analyzed to determine their performance using various types of video content and formats. Several algorithms are developed to provide a dynamic adaptation of coding tools and parameters to specific video content type, format and bandwidth of transmission.

Due to the nature of heterogeneous networks where channel conditions, terminals, users capabilities and preferences etc are unpredictably changing, hence limiting the adaptability of a specific technique adopted, a Dynamic Scalability Decision Making Algorithm (SADMA) is developed. The algorithm autonomously selects one of the designed scalability techniques basing its decision on the monitored and reported channel conditions. Experiments were conducted using a purpose-built heterogeneous network simulator and the network-aware selection of the scalability techniques is based on real time simulation results. A technique with a minimum delay, low bit-rate, low frame rate and low quality is adopted as a reactive measure to a predicted bad channel condition. If the use of the techniques is not favoured due to deteriorating channel conditions reported, a reduced layered stream or base layer is used. If the network status does not allow the use of the base layer, then the stream uses parameter identifiers with high efficiency to improve the scalability and adaptation of the video service.

To further improve the flexibility and efficiency of the algorithm, a dynamic de-blocking filter and lambda value selection are analyzed and introduced in the algorithm. Various methods, interfaces and algorithms are defined for transcoding from one technique to another and extracting sub-streams when the network conditions do not allow for the transmission of the entire bit-stream.

Author's Contributions

1. *Development and Evaluation of Novel Scalability Techniques*. These techniques include *t+q*, *t+s+q*, *t+s* and *s+q*. The objective and subjective performance of the techniques are evaluated. The techniques revealed significant objective quality performance over non-combined and *SL* techniques. The techniques variability for objective and subjective quality and real time performance made them suitable for dynamic scalability adaptation. The *SL* and the techniques show the same subjective performance at bit-rates above 64kbits/s. At bitrates below 64kbits/s, there exists slight improvement over the techniques which are due to overheads on the scalable stream and their efficiency in compression which is used for better scalability adaptation. We also evaluated against some other existing techniques(MPEG-4) and based on real time performances.

2. *SVC design parameters and coding elements* are evaluated for objective, subjective and real time performances. The influences and effects of the design parameters and coding elements are analysed. We achieved a flexible and dynamic scalability adaptation scheme based on the design parameters performances through fixed parameter design or parameter exchange.

3. *Dynamic De-blocking Filtering Algorithm* is proposed and implemented for scalability adaptation over heterogeneous networks. Simulation results show that, scalability adaptation can be achieved from this algorithm. The algorithm supports scalability adaptation by adapting a filtering technique with better real time performance. The techniques have influences on r.d.o. due to the natural video content or the details of the video sequence which arises due to the spatial format used. The results (section 6.8.1 to 6.8.3) of this work are utilised for the support of *SADMA* algorithm.

4. *Novel Lambda Value Selection (LVS) Algorithm*: This algorithm supports a maximum gain for EL and the BL. In H.264/AVC the impact of selection parameter vector on the BL is done with no consideration of the ELs (section 6.9). The algorithm is operated based on current computed bit-rates prior considering an effective lambda value selection for BL/EL. The implementation of this algorithm provides bits reduction in kbit/s (section 6.9)

5. *Novel Constraint Minimum Qp Algorithm*: This algorithm considers the current transmission rates and restricts a minimum quantisation level. This supports achieving target rates and efficient scalability adaptation (section 5.13). The gain in bits reduction is in hundreds of bits from the results of the algorithm simulations (section 5.13). However, to maintain a picture quality the algorithm is limited at bit-rates up to 300kbits/s based on simulations output results.

6. ***Novel Improved t+s technique (Im-t+s):*** To further support BL scalability adaptation, an improved *t+s* technique is developed. This technique employed a reduction of the current BL thereby creating another layer with better adaptation characteristics than the active BL. An adaptation algorithm mentioned from the literature in section 7.2 uses only BL as a solution. Hence this new technique supports better flexible scalability by providing a more adaptable BL to improve the scalability adaptation especially when the networks conditions are highly deteriorated. The gain achieved with the new BL is kbit/s reduction depending on the target bit-rates. Experiments show that the quality of the new layer (BL) is good with tiny additional processing complexity incurred by the technique about 0.69secs using the simulating PC (Appendix A3).

An experimental work is conducted to look into the ***impact of additional layering structure*** on scalability level output performance and the BL decoding computational complexity from the hierarchy of different quality layers. An optimised layering structure has been determined to be from four quality layer structure in QCIF format. Additional layering increases the scalability levels and provides graceful degradation/robust bit-stream due to different discrete rate points and several temporal scalable layers. The performance of the layering structure has been evaluated based on objective and subjective quality output. There is no video quality deterioration observed from the subjective performance of the hierarchical quality layers (section 6.3.6). We also show and experimentally proved that G α T and G α kT. G refers to Group of Pictures, k is a natural number and T refers to temporal layers.

7. An experimental work is conducted to investigate the ***quality scalability influence on a scalable stream when different rate points*** are used. One quality scalability (*MGS*) which provides 25% rate points and the other (*CGS*) which supports 10% rate points are simulated. Both the low and high resolution formats are used in the experiment. The results (section 5.10) show that *MGS* has better scalability and rate distortion support with up to 10 kbit/s reductions and a dB gain. It is therefore most preferred in providing an adaptive scalability. ***Methods for avoiding picture drift at BL*** with SNR scalability are introduced in section 5.9. Other Methods for ***improving future SNR scalability*** are also proposed in section 5.9.

8. An experimental work is conducted to establish the ***impact of scalability adaptation and quality performance for encoding key pictures*** within the video sequence. Key Pictures (KPs) are inserted after a number of designed frames for synchronisation. Experimental work revealed the effects of scalability and quality performance at various hierarchical *GOP* structure size. Poor quality is shown at smaller *GOP* sizes and the quality is improved at large *GOP* sizes, when KPs are encoded. Although encoding the key pictures support more than 20000 bits reduction and provide better scalability, the *GOP* size allocation should be considered as it can affects the video quality (section 5.12).

9. *Two level scalability adaptation schemes* are proposed. The scheme is provided and is suitable to a number of applications based on their requirements. Some suitable applications include broadcasting where video streams are transmitted to multi-channel links and clients of dynamic resources and requirements. This scheme is illustrated in Figure 2 - 4. The scheme includes several developed techniques encoded at the same time and transmitted from one single processor. The requirement of this technique will involve the use of very high speed PC processor that will reduce the complexity of processing several scalable bit-streams at the same time. However the scalability adaptation provision will be higher and can be designed to suit any client or target resources. Experimental results are presented in section 5.14.

10. *Novel Scalability Decision Making Algorithm* (*SADMA*) is developed. This algorithm dynamically supports a flexible scalability adaptation over heterogeneous networks. It employed the developed scalability techniques (section 6.3 to section 6.7), the introduced adaptation algorithms (chapters five, six and seven) and evaluated parameter design and coding elements (chapter five) in its implementation. The algorithm ensures that a suitable scalable stream is passed into the network at a particular network condition. It does this by frequent processing of channel conditions and hence decides for a suitable scalable bit-stream. The processed network delay is compared with a pre-configured real systems acceptable threshold delay before passing a suitable bitstream. The algorithm executes its task using some sets of procedural adaptation procedures. It does not perform all the algorithm tasks at once except when there is need from the network report. This is design with this nature to minimise the processing complexity and supports high quality picture when the network is not fully swamped. The algorithm establishes a suitable technique for a specific network and when there is need for changes to other techniques (*Adaptation I or Adaptation II*) which are described in chapter seven.

The algorithm is represented as transcoder that switch from one bit-stream to another within the networks. This is illustrated as in Figure 7 - 17. The homogeneous transcoding is represented by the automatic scalability adaptation provided in the techniques bit-stream and the heterogeneous transcoding support switching from one technique to another.

The methods that can be used for switching among different techniques bit-streams and within the sub-bit-streams of a technique are discussed in section 7.12.

11. *Journal papers*

1. *S. Muhammad, A.H. Sadka*, "Design and Evaluation of Novel Scalability Techniques for Dynamic Video Adaptation over Heterogeneous Networks", IEEE Transactions on Multimedia – Submitted for internal review Feb, 2013 and to the journal on 17, April, 2013.

2. *S. Muhammad, A.H. Sadka*, "Analysis of Coding Parameters/Elements for their Effects on Video Scalability and Adaptation" – Elsevier, Signal Processing: Image Communication - Submitted for internal review Feb, 2013 and to the journal on 17, April, 2013.

3. *S. Muhammad, A.H. Sadka*, Design and Evaluation of Novel SNR Scalability Techniques and Quality layers Impact for Adaptation. – To be submitted in 3-4 months.

12. *Conference papers*

1. *S. Muhammad, A.H. Sadka*, "SVC Concept and its Performance Evaluation against Single Layer Coding", RESCON, Brunel University, Uxbridge, UK, 21st – 23rd June 2010.

2. *S. Muhammad, A.H. Sadka*, "Combined Scalability Techniques for Improved and Flexible Scalability with H.264/AVC Extension", RESCON, Uxbridge, UK, 18st – 21st June 2012.

13. *Seminars*

1. *S. Muhammad, A.H. Sadka*, "Scalable and Network Aware Video Coding for Advanced Communications over Heterogeneous Networks" CMCR Lab, Brunel University 2010.

2. *S. Muhammad, A.H. Sadka*, “Improved and Optimised Scalability Methods”, CMCR Lab, Brunel University, November, 2011.

Acknowledgements

Firstly, I would like to thank the Almighty God (Allah) for taking me through all these years successfully and keeping me in good health and guidance. It's quite a journey with full of experiences for the rest of my professional and personal life. I am also thanking the Almighty Allah for blessing me with the ability, favour and patience to conduct this research work to completion.

This achievement wouldn't have been possible without help, support and stimulation of my supervisors. I would like to express my sincere gratitude and appreciation to my supervisors. I had the honour to work with two of the well-recognised researchers in the field of multimedia, image processing and video communication. Professor A.H. Sadka who is my first supervisor and had the opportunity to work with him throughout my research period is a role model, talented and internationally respected researcher. Professor A.H. Sadka's comments, advices and suggestions on my work have always made my work easier and successful. His knowledge, profession, availability, patience and constructive criticism on my work and support are unique. My relationship with him from the beginning up to the end is exceptional, he has been very supportive especially when I get stacked on my work.

Dr. Ammar Aggoun who is my second supervisor whom I benefited from many of his advices, lectures and seminars is also a role model, talented and internationally respected researcher. His knowledge, suggestions and guidance during his visits to CMCR Lab and Studio Lab made good experience to me.

I would like to say my special thanks to Dr. N. Sani Gwarzo, Professor M. Abubakar Gusau, Abubakar Roko and Abdullahi Abbas.

Thanks go to Karen Thompson for bringing encouragement and joy to colour my life in the CMCR Lab and ECE of Brunel University. Also thanks go to Lisa McCarthy for her support during my stay in SED. Many thanks to them and all other staff for great help with some routine but important administrative work.

I also wish to thank my colleagues in CMCR Lab. Thanks Rafiq, Mohib, Ghaida, Nawaz, Sadiq, Obaid, Abdulkadir, Ibrahim and Amal. I appreciate the friendship and collaboration.

My brothers and sisters deserve special thanks and appreciations for always supporting me without which I would not have been able to face the challenges of this research work. I appreciate all your prayers and concerns.

I would like to finally say my thanks to my employer CSTD/NASRDA for giving me the opportunity to further my studies up to this level. I particularly wish to thank the Director General Dr. S.O. Muhammed and the Director CSTD Dr. S.O. Onuh who kindly accepted and dealt with all the bureaucratic procedures and approvals necessary to pursue my studies here in the UK for PhD degree. Many thanks for their federalism and appropriate actions to the staff of CSTD/NASRDA.

Dedication

I dedicated this thesis to my beloved parents Alhaji Muhammad Sakatare and Hajiya Fatima Muhammad for their parental guidance, love, advice that sees me through as a growing child and manhood.

This thesis work is also dedicated to my beloved wife Hauwau and my three children Fatima, Hafsat and Fadila. The patience and sacrifices you all have made during my thesis work for more than three years which supported my thesis work are highly appreciated. You have made it possible for me to complete this work and there is no way I would ever be able to have done it without your support. Your patience, understanding, love and all supports brought me through the challenges of long days and nights and provided the necessary support to finish this thesis work. I always love you all and forever. Great thanks for being there for me during the cause of my PhD studies here in the United Kingdom.

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

AVC	Advance Video Coding
ATM	Asynchronous Transfer Mode
BL	Base Layer
CGS	Course Grain Scalability
CABAC	Context Based Adaptive Binary Length Coding
CAVLC	Context Based Adaptive Variable Length Coding
CIF	Common Intermediate Format
DCS	Discrete Cosine Transform
DPB	Decoded Picture Buffer Size
EL	Enhancement Layer
ELa	Enhanced with a number of layers
GOP	Group of Pictures
HDTV	Higher Definition Television
HEVC	Higher Efficiency Video Coding
Im-t+s	Improved temporal + spatial scalability
IP	Internet Protocol
KP	Key Pictures
LDTV	Low Definition Television
Ln	Layer n (n = 0, 1…….N)
MB	Macro-block
MC	Motion Compensation
ME	Motion Estimation
MGS	Medium Granular Scalability
MV	Motion Vector
NAL	Network Abstraction Layer
SADMA	Scalability Decision Making Algorithm

SL	Single Layer
SITL	System In The Loop
SNR	Signal to Noise Ratio
SVC	Scalable Video Coding
r.d.o.	Rate Distortion Optimisation
t+s+q	Temporal+Spatial+Quality Scalability Technique
t+s	Temporal+Spatial Scalability Technique
t+q	Temporal+Quality Scalability Technique
t	Temporal Scalability
s	Spatial Scalability
q	Quality Scalability
QCIF	Quarter Common Intermediate Format
QL(s)	Quality Layer(s)
Qp	Quantisation Parameter
QoS	Quality of Service
VCL	Video Coding Layer

CHAPTER ONE

1. INTRODUCTION

1.1. The Context

The current trend in video and multimedia communication is towards ubiquitous availability of an application. Diverse versions of multimedia devices will have access to an application at any time and place. This feature of multimedia devices is the motivating challenge to develop efficient scalable and adaptive techniques for video communications.

The current and future video and multimedia communication devices and network terminals have a tendency to include varied capabilities. These capabilities include limitations in display resolution, memory capacity, processing power and bandwidth requirement. On the other side, every one application user will also require an acceptable quality. A good example here is a Higher Definition Television (HDTV) device using a standard screen resolution of 1280x720 or 1920x1080 pixels and Low Definition Television (LDTV) device using a standard resolution of 480x272 pixels [1]. Personal Computers and Desktops with resolution of 800x600, 1024x768 or 1280x1024 pixels and variable processing and memory capabilities etc [2] are also good examples. This is also applicable to network terminals namely routers, switches, bridges and other used decoders. With SVC video bit-stream, one single application can be available to all the set of variable capability devices mentioned above. Figure 1 below illustrates a Scalable Video Coder providing video services from a single source stream to multi-devices with wide-range of resource requirements.

A scalable video stream is that which can be available for a wide range of different devices and networks. [3] defined scalable stream as that which enables the transmission and decoding of partial bit-streams. Therefore, SVC provides graceful degradation functionalities in lossy transmission environments as well as power, format and bit-rate adaptation. These functionalities enhanced video transmissions, storage and devices portability. The adaptation functionality also reduces bandwidth requirement compared to implementing a non-scalable stream. The recent H.264/SVC extension has achieved major improvements in coding efficiency with an improved degree of supported scalability relative to scalable baselines of previous video coding standards. H.265/HEVC (High Efficiency Video Coding) is now the most recent standard to be released in the near future (July 2013), an upgrade of H.264. It is designed for the next generation networks/devices and applications ranging from high quality video streaming on mobile devices to ultra-high resolution displays. HEVC can reduce bandwidth requirement and bit-stream size by 50% providing equivalent or enhanced quality over current H.264/AVC [4].

In further broad terms, Scalable Video Stream can be expressed as a stream which can adapt to the network variable time conditions and can be reconstructed by decoders with variable resource capabilities. A better and flexible SVC stream is the one, which a higher number of clients of diverse variety of application resources and network conditions are adaptable to. This simply describes video services adaptation over heterogeneous and multi-channel environment. This research work mainly focused on scalability adaptation over heterogeneous networks.

Figure 1 - 1 illustrates scalability and adaptation concept. It demonstrates an application generating an SVC stream that is accessible by multiple devices and devices with variable bit-rates and other required resources. The scalable stream is the original video encoded and transmitted from a communicating device such as mobile hand set, satellite transmitter or receiver, a ground antenna, or a computer machine. It is a one single stream embedded with multiple layers of video with different priorities of bitrates, frame rates of the encoder and spatial resolution of the video content. The multiple layers of the stream might have dependencies as the stream can be inter-layer predicted to minimise the video content size and hence the bandwidth. The details of these are further discussed in chapter six.

Scalability is achieved through a layered coding method [5]. [5] further describes that streaming solutions usually resort to achieving scalability using one of a variable video coding methods. The structure of the technique used is based on the codec. The different methodologies used to embed a video stream with a layered coding in the past and existing standard codecs are discussed in chapter two.

Scalability is expressed or defined based on the codec technology or application it is specifically designed for. [6] defined scalable video as the one regarded as being synonymous with layered video coding which was originally proposed by the author to increase robustness of video codec against cell/loss in ATM networks. Our earlier definition is based on the recent codec *H.264/AVC Extension* which is designed to scale video stream within a heterogeneous network.

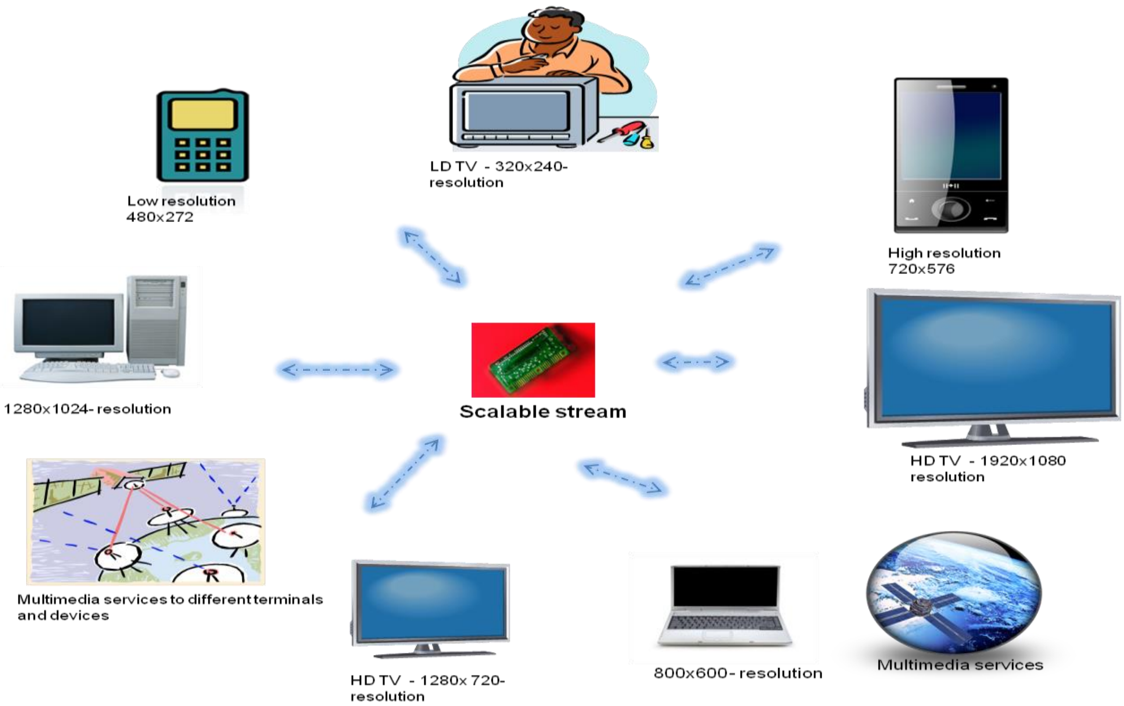


Figure 1 - 1 : Scalable Device Providing Video Services from a Source Single Scalable Stream to Wide Range of Multimedia Devices in a Heterogeneous Network Channels.

1.2. Highlights of Scalability Applications

Scalability applications are classified based on the type of scalability and codec [7]. All H.264/AVC and MPEG codecs support temporal, spatial and quality scalability techniques [6, 7]. These scalability techniques are described in section 2.4. It is obvious that particular requirements for scalable video coding may vary widely, according to the application being considered [11]. The demanded requirements for a family of one or more applications may not be the same requirements for other applications.

1.2.1. Quality (SNR) Scalability Applications

The bit-stream in this technique consists of at least two layers of same spatial and temporal resolution. One of the layers posses a higher picture quality than the other [6]. SNR can have a base picture quality layer and can be drift free. Hence suitable applications can be [6, 8]:

- Transmission of video at dissimilar quality of interest such as multi-quality video, video on demand, broadcasting of TV and enhanced TV.

- Video over heterogeneous networks with a high error or packet loss rates, such as internet or greatly congested network.

1.2.2. Spatial Scalability Applications

In this technique video stream is embedded with two layers of different spatial resolution. There can be loose dependency for each layer to use any codec [6]. Hence numerous applications [8, 6] are found with this technique as follows:

- Simulcasting of drift free good quality video and two spatial resolutions such as standard TV and HDTV.
- Distribution of video over computer networks. Computer networks utilise variable bandwidth and display resolutions.
- Video browsing with spatial technique might reduce time delay
- Reception of good quality low spatial resolution pictures over mobile networks.
- Transmission of error resilience video over packet networks.

1.2.3. Temporal Scalability Applications

This technique employs a moderately complex encoder where a single coder encodes both layers. The layers will operate at different temporal rates. The major applications [6, 8] of this technique include:

- Migration to progressive (HDTV) from the current interlaced broadcast.
- Internetworking between lower bit-rate mobile and higher bit-rate fixed networks. The variable temporal rate of the coder with variable bitrates will suit the application.
- Video over Local Area Network (LAN), internet and ATM for computer workstations.
- Video over packet (internet/ATM) networks for loss resilience.

In general, SVC allows splitting of the video bit-stream into a base and one or multiple enhancement layers. These partial bit-streams can now be transmitted via variable channels or networks with different resources. For example, with IP networks the dissimilar quality bit-stream classes can be assigned to one or more consumed network streams, thus a possible quality adaptation on the network stream is guaranteed [9].

Other benefits can be achieved from SVC when the protection of the SVC layer is treated independently. Forward Error Correction (FEC) is often utilised to protect data sent out via wireless broadcast channels [9]. Error protection Schemes such as Unequal Error Protection (UEP) and /or Unequal Erasure Protection (UXP) can be used to guarantee an error free transmission of priority layers, e.g. the base layer. UEP/UXP can be used on top of the existing FEC channel. With SVC layers, this scheme can guarantee a basic video quality over a wide range of channel error rates. This is illustrated in Figure 1 - 2.

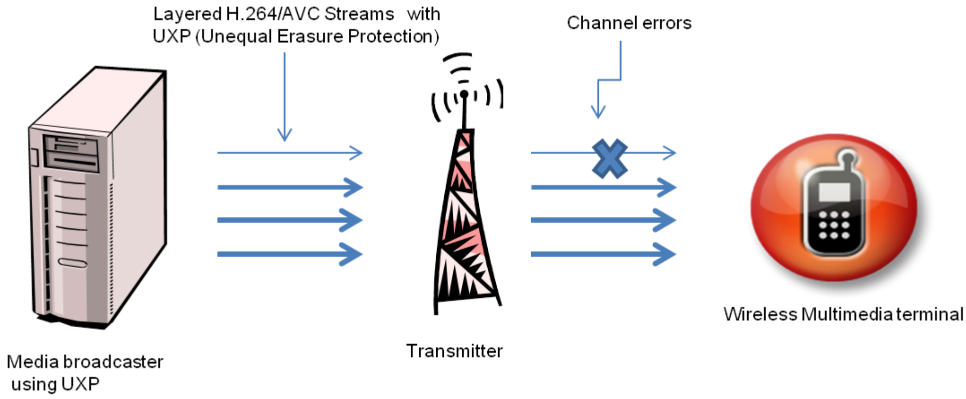


Figure 1 - 2 : Unequal Erasure Protection with SVC Quality Layers

1.3. The Problem

Bandwidth limitation is one of the important factors in multimedia communications. Video streams are transmitted over networks with time varying conditions and resource limitations [5]. To efficiently make use of available network resources at any instance and guarantee a required level of perceptual quality from user's perspective, adaptation mechanism is required. Network adaptation is a core application of scalability. An adaptable video stream can scale its input to reduce network congestion as well as provide access to variable terminals, devices and applications. A non-adaptable stream may lead to congested network, loss of data and hence quality degradation. In multimedia communications, network terminals, devices and end user applications exhibit variable conditions and limited resources.

This group of variable networks and devices environment is referred to as heterogeneous networks. Hence, the issue of serving a video stream to these environment elements with a guaranteed quality of service (QoS) is our focus in this thesis.

The main challenges in this thesis are scalability and adaptation techniques and algorithms that improve the video bit-stream scalability adaptation and ensure the provision of guaranteed quality of service (QoS).

1.4. Thesis Objectives

The aim and objectives are defined to solve the problem outlined in section 1.3. The main objectives are as follows:

1. Study and understanding of the state-of-the-art in scalable video coding
2. Design and development of a suite of suitable scalability and adaptation techniques that will improve scalability adaptation and guarantee quality of service(QoS) from the user's perspective over heterogeneous networks.
3. Design and development of algorithms and methodologies that will use video coding elements and parameter settings for an efficient scalability adaptation over heterogeneous networks
4. Analyse the impact of multiple layering on scalability outputs

1.5. The Solution

There is diverse variety of channel capacities, display resolutions, limited power and processing in the current and future multimedia communications. This can be resolved with effective scalability techniques and algorithms. The algorithms and techniques mostly consist of efficient and effective control mechanisms [5]. The swamping of the network with extreme delays and loss of information could be avoided by a preventive instead of reactive solution [5]. In preventive control, network reports are collected within a time interval and the most suitable identifiers of a stream are activated. The activated stream will be of low bit-rate, frame rate, resolution and quality that will provide suitable adaption to network predicted conditions. An activated stream will consist of several numbers of video layers. These layers are of different levels of priority in terms of bitrates, temporal rates, quantisation levels and other associated video coding tools and parameters. The activated stream can also be one layer with reduced complexity depending on the reported network conditions.

Other solutions considered in this thesis are using a combination of scalability techniques. Temporal + spatial technique (*t+s*), temporal + quality (*t+q*) and temporal + spatial +quality (*t+s+q*) techniques are considered. These techniques are simulated and results are compared with non-combined and non-scalable techniques. The combined techniques are found to be more scalable and adaptive to network variability. Other adaptation techniques are also introduced and evaluated (chapter five and six). In line with these techniques and schemes, a Dynamic Scalability Decision Algorithm (*SADMA*) is developed. This algorithm allows communicating into the network one of the techniques based on network condition reports.

A heterogeneous network simulator is purposely built and validated to allow testing and evaluating of the *SADMA* algorithm and other introduced schemes and techniques (chapter five and six).

Another methodology considered is a *standalone PC processor* hosting several encoders with variable scalability performances in adapting to network conditions. This system requires a high speed PC to simultaneously run the encoders. This algorithm is improved by defining a two-level scalability which includes *SADMA* in chapter seven and several scalable techniques discussed in chapter five and six.

The defined solutions above for network control and adaptation will be more desirable in real time video services. The effects of network congestion and delays could be more disastrous in real time video communications where the reconstructed quality is much less tolerant to data loss and delays [5].

1.6. Research Methodologies and Approaches

The design of the scalability techniques focused on achieving a preventive and network aware control. Preventive control takes reactive measures when network conditions are predicted to be dreadful in future. To meet this aim, the following approaches are considered (1) Decision Making Algorithm that frequently monitors network conditions (2) the usage of analysed and evaluated parameter identifiers (3) development of scalability techniques that improve the scalability and adaptation over single(*SL*)/other techniques (4) development of new algorithms and schemes that can support dynamic and flexible scalability adaptation (4) building of a purpose heterogeneous network simulator for real time simulations (6) verification and validation of the simulator and simulation environment set up (7) employment of a conceptual frame work that describes adaptation from source encoder to application user (8) the usage of standard ITU video sequences for research purposes (9) development of window scripts to process video sequences and configuration files with H.264/AVC Extension codec, (10) objective, subjective and real time analysis and evaluation of the algorithm output results, scheme and scenarios which allow evaluation among non-scalable algorithms and state-of-the-art scalability algorithms and methods.

The research involves extensive use of H.264/AVC Extension codec. The coder was studied and customized for the research aim. The usage of the H.264/AVC codec involves several experiments with several video sequences to establish the performance of the coder for a suite of developed scalability techniques, adaptation schemes/algorithms, coding elements and parameter design. Other standard tools used include YUVviewer [10] for subjective analysis and frames extraction. Opnet program is used to build the heterogeneous network simulator for the real time test of the performance of the designed scalability techniques, schemes and algorithms.

1.7. Thesis Main Achievements

The thesis achieved objectives are outlined as follows:

1. Literature survey for scalable video coding state of the art
2. Designed and evaluated scalability techniques that used combination of different types of techniques.
3. Built and validated Heterogeneous Network Simulator for real time simulations of the developed scalability techniques and several coding scenarios and algorithms.
4. Design of algorithms and schemes that are used to improve scalability adaptation. These include dynamic de-blocking filter algorithm, Lambda value selection algorithm, Constraint minimum Qp value selection based on transmission bit-rate, inter-layer prediction and key-picture encoding technique.

5. Evaluation and analysis of influences of coding parameters design on scalability outputs over multi-channel networks.
6. Design and evaluation of Scalability Decision Making Algorithm (*SADMA*) which supports a dynamic and flexible scalability adaptation.
7. Design of two levels Scalability Adaptation Scheme.
8. Definition of sub-stream extraction and transcoding methods for scalability control and adaptation.

1.8. Thesis Outline

This thesis is outlined based on the format acceptable by Brunel University. Apart from this **Chapter** which is introduction, the next **Chapter** describes Scalable Video Coding concepts and technologies. The state of the art in the field of scalable coding are discussed. We also describe bit-rate variability and control as one of the factors influencing scalability. Previous, existing and future scalability techniques are discussed. Adaptation as the core application of scalability concept is also described. The chapter concludes with discussions on SVC evaluation metrics that are generally used and stated in the literature. In **Chapter Three**, H.264/AVC Extension codec requirements and SVC features are described. Comparative analysis with other prior standards implementing scalability is also discussed.

In **Chapter Four**, the required network models for simulation and experimental works for this research work are built and described. The built model is a heterogeneous simulator consisting of WIMAX, Wireless LAN and LAN networks channels. The simulator is used to test the real time performances for the developed scalability techniques, schemes and algorithms. All built networks and experimental set-ups in the research work are described, verified and validated in this chapter.

Chapter Five covers the analysis and evaluation of influences of coding elements and parameter design on SVC output. Several schemes and algorithms that could be used to improve scalability adaptation are introduced and described in this chapter.

Chapter Six covers the development of novel scalability techniques. The performances of the techniques are evaluated against non-scalable and state-of-the-art techniques. Other adaptation algorithms are also introduced in this chapter.

Chapter Seven describes the design and the implementation of Scalability Decision Making Algorithm (*SADMA*). Several experiments and real time simulations are discussed and analysed for validation and verification of the algorithm implementations and procedures.

Chapter Eight summarises the thesis and the main achievements. Future works that could extend this project achievement are also described.

1.9. Summary and Conclusion

In this chapter, we introduce the context of the subject of scalable video coding. A scalable coding is used for the adaptation to network terminals and devices variable time conditions. We described that scalability as a scheme has different techniques that have specific applications for video services. New scalability techniques have the potential to support present and future multimedia communications with better and flexible scalability adaptation. An effective solution is to prevent the occurrence of network delays and loss of data by using network aware scalability schemes and algorithms.

Simulcast technique differs from SVC in terms of complexity and functionalities. All layers for a simulcast are independently coded whereas SVC layers are coded dependently. The dependent and interlayer coding will provide more efficient compression and reduce complexity over network channels. Furthermore, independent coding of each layer may incur higher costs of implementation. SVC is to be designed to produce a bit-stream that is accessible by multi-channels links and applications.

In this thesis, new techniques are developed and evaluated to provide adaptation solutions for present and future network links and applications. To provide improved flexibility with the scalability adaptation, a Scalability Decision Making Algorithm (*SADMA*) is developed. The algorithm provides scalability adaptation to suit the reported network channels conditions.

In the next chapter, a detailed overview of scalable video coding is provided and some major areas of this research work are highlighted.

References

[1] Holub, P., Matyska, L., Liška, M., Hejtmánek, L., Denemark, J., Rebok, T., Hutanu, A., Paruchuri, R., Radil, J. and Hladká, E. (2006) "High-definition multimedia for multiparty low-latency interactive communication", *Future Generation Computer Systems*, vol. 22, no. 8, pp. 856-861.

[2] Preuveneers, D., Vandewoude, Y., Rigole, P., Ayed, D. and Berbers, Y. (2006) "Context-aware adaptation for component-based pervasive computing systems", *Advances in Pervasive Computing,* , pp. 125-128.

[3] Schwarz, H., Marpe, D. and Wiegand, T. (2007) "Overview of the scalable video coding extension of the H. 264/AVC standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1103-1120.

[4] Sullivan, G.J. and Ohm, J. (2010) "Recent developments in standardization of high efficiency video coding (HEVC)", *Proc. SPIE*, pp. 77980V.

[5] Sadka, A.H. (2002) *Flow Control in Compressed Video Communications,* Wiley Online Library.

[6] Ghanbari, M. (2003) "Standard Codecs: Image compression to advanced video coding", Institution of Electrical Engineers, .

[7] Haskell, B.G., Puri, A. and Netravali, A.N. (1996) *Digital video: an introduction to MPEG-2,* Springer.

[8] Ohm, J.R., Koenen, R. and Chiariglione, L. "SVC requirements specified by MPEG (ISO/IEC JTC1 SC29 WG11)", .

[10] ftp.tnt.uni-hannover.de/pub/svc/testsequence Accessed January 2010.

[11] Richardson, I.E.G. "H. 264 and MPEG-4 video compression. 2003", *John Wiley&Sons Ltd., New Jersey,* .

[12] Segall, C.A. and Sullivan, G.J. (2007) "Spatial scalability within the H. 264/AVC scalable video coding extension", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1121-1135.

CHAPTER TWO

2. OVERVIEW OF SCALABLE VIDEO CODING

2.1. Introduction

In this chapter, the concept of scalability, adaptation and associated technological schemes are discussed. A framework structure for designing scalability and adaptation solutions is described from the literature. The discussions include a general overview on how to design and improve on an existing scheme or for environmental and applications requirements. A comparative and associative discussion between rate control and scalability are made. Also included in this chapter are discussions on the performance differences and compatibility of the various scalability and adaptation techniques including H.264/AVC extension and other state-of-the-art video coding standards.

A literature survey of the past and existing scalability and adaptation techniques is presented. The methodology of the techniques and how they are related or improved with this research work are also discussed. It is obvious that these technologies are improved or replaced from one age to another. This is due to rapid evolvement of new devices, network technologies e.g. 3G, and 4G networks. The new devices and structure posses more constraints in their bandwidth, power and therefore may have other user preferences and scalability and adaptation requirements. SVC evaluation metrics and techniques are also discussed. The metrics are used for the evaluation of the developed scalability techniques and algorithms in chapters five, six and seven.

2.2. Scalability and Bit-rate Control

The DCT-based coding algorithm bit-rate fluctuates according to the nature of the video sequence [1]. This variation is caused due to moving objects velocity, size, texture or camera pan and movement. Typically, an encoder with constant parameters for example $Q_{p,}$ and motion estimation search area will generate more bits when motion is high with more video details, and fewer bits when motion is low with less video details. The main objective of the rate controller is to achieve a constant bit-rate for transmission over a circuit switched network. This fluctuation in bitrates can be a problem for many practical delivery and storage mechanisms [1]. A packet switched network can support varying throughput rates but the average throughput at any point in time is limited by congestion and link rate factors. A number of existing control algorithms are used to implement rate control in video and multimedia communications. The type of algorithm used will depend on applications and their requirements.

Scalable coder presents an output that automates control of network variable time conditions. The output of a scalable coder bit-stream is embedded with multiple layers that can be extracted and reconstructed by different terminals and application. On the other hand, bit-rate control uses a particular mechanism to adjust the bit-rate when the need arises. The solutions used for the adjustment of the bit-rates vary and depend on specific application requirements. The technique used for the adjustment can be one of the scalability or the adaptation algorithms. Scalability and bit-rate control techniques adapt to network through different mechanisms and techniques.

In this research work, new scalability techniques are developed and evaluated for adaptation to diverse and pervasive environment. The techniques are operated for the functionality of a Scalability Decision Making Algorithm (*SADMA*) described in chapter seven. The algorithm decides based on the reported network status which of the techniques to pass into the network. The selected technique will either increase or decrease the bit-rate variation. Hence, the scalability algorithm embeds a rate control mechanism in addition to the defined automated rate control in the scalable bit-stream. In the automated rate control, a client can switch video reception from a high bit-rate to low bit-rate or from low bit-rate to high bit-rate when high throughput is allowed by the network and the end application. The switching mechanism is discussed in chapter seven.

In conclusion and based on the discussions above, a scalable coder produces an automated rate control video bit-stream, whereas a rate controller provides bit-stream management and control when the need arises. The bit-rate management can also be supported from scalability schemes and algorithm as the one developed and shown in chapter seven. The comparative relationship between Scalability Control Algorithm (S_a), Scalability (S_c), and Rate Control (R_c) can be represented mathematically with set theories. This is presented in (2-1 - 2-3).

$$R_c \in S_a \qquad 2\text{-}1$$

$$R_c \subseteq S_a, R_c \cup S_a = S_a \qquad 2\text{-}2$$

$$R_c \cap S_a = S_a \cap R_c \qquad 2\text{-}3$$

2.3. Scalability

Scalable coding enables video stream to adapt to variable time conditions of diverse multi-channel networks. A decoder therefore can selectively decode part of the reconstructed bit-stream [2]. Hence the stream is flexible to serve a wide range of networks and applications. The scalable bit-stream is structured in a number of layers. This includes a single Base Layer (BL) and a single or multiple Enhancement Layers (ELs). This is illustrated in Figure 2 - 1. From Figure 2 - 1, decoder X receives only the BL and can decode a basic quality version of the scene, whereas decoder Y receives all layers and reconstruct a high quality of the video scene [2]. A number of applications can be deduced from this layering arrangement, for instance, a decoder with low complexity may only be able to decode the base layer; a network segment with limited resources can extract the base layer with low rate; and an error-resilient BL layer may be transmitted with higher priority than enhancement layers.

In simulcasting coding technique, each layer representing a quality or resolution is coded independently [3]. Hence any layer can be decoded by a single layer non-scalable decoder. Also in simulcast coding, the total available bandwidth is partitioned depending on the quality desired for each layer. The assumption in simulcast is that independent decoders would be used to decode each layer [3]. This type of partition coding is a technique design in MPEG-2 codec [3, 4]. It is a means of dividing MPEG-2 single layer bit-stream in two parts or two layers. The critical parts will be in the first part layer which includes headers, motion vectors, and lower order DCT coefficients. The second layer of the bit-stream takes less critical data such as higher order DCT coefficients and is transmitted in channels with lower error performance [3]. This minimises degradation to channel errors since the critical parts are better protected [3]. However, coding of video layer independently induces complexity and in-efficiency.

In Scalable Video Coding, one layer of the bit-stream is coded independently whereas other layers are coded dependently [3]. A following layer is coded with respect to the previous layer. The independently coded layer can be reconstructed by one-layer decoders. Format compatibility can be achieved if this layer is to be coded with another video standard. For instance, an independent layer may be coded by H.264, H.261, MPEG-4, MPEG-1, thus providing compatibility with H.264/AVC, H.261, MPEG-4, and MPEG-1 respectively [3].

In general, scalable video coding is more efficient than simulcast coding. Each layer can reuse some of the bandwidth assigned to the previous layer except for the independently coded layer. The efficiency of the scalable coder is dependent on the technique used, the allocated bandwidth for each of the layers and the adaptation algorithm/scheme adopted. Also scalable coding functionalities are beyond simulcast coding especially with the recent H.264/AVC SVC. H.264/AVC is the recent SVC codec that provides most efficient coding.

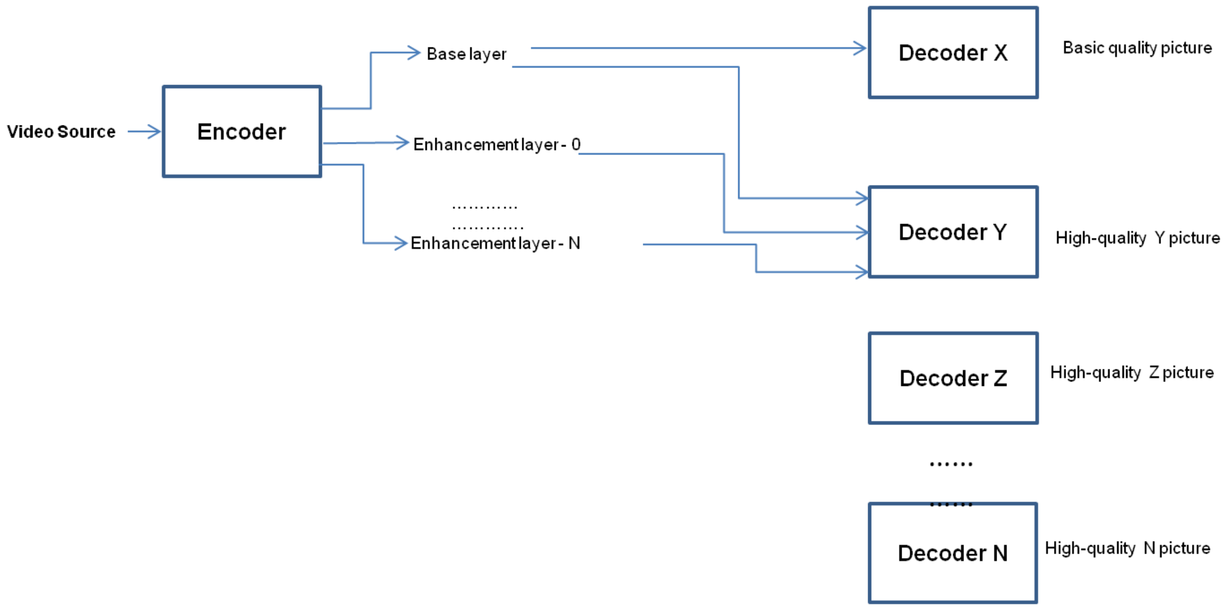


Figure 2 - 1: General Concept of a Scalable Coder

2.3.1. Base Layer (BL) and Enhancement Layers(ELs)

The composition and structure of the base and enhancement layers depend on codec design and intended applications. The current and the most recent codecs are all structured on a layering composition of a BL and an EL. The number of ELs depends on the codec implementation. BL hold the basic video data which all decoders should access. A decoder can then extract any of the ELs to improve the quality of the video. H.264/AVC Extension can implement scalability with up to eight ELs composition [5]. Initial Scalability implementations of H.261 and H.263 employed a BL and a single EL. The layering structure impacts on the re-construction complexity. In MPEG-4 and H.264, EL is reconstructed by adding the enhancement layers to the base layer. The reconstruction process is achieved as follows in MPEG-4 [1].

1. The base layer is decoded and up-sampled to the initial resolution
2. The enhancement layer is decoded
3. The output video is formed by adding the residual from the enhancement layer to the decoded base layer.

The above re-construction process is also achieved in H.261, H.263 codec [6]. The re-construction of the video sequence highly depends on the base layer, while the EL can improve the video sequence quality but its absence causes deterioration of the received video quality [6]. In Quality Scalability, the decoder optionally decodes one of the ELs. This is limited to decoder capability and quality requirement.

In MPEG-4 Visual, spatial scalability is encoded as follows [1]

- The input video frame is sub-sampled horizontally and vertically.
- The reduced resolution frame is encoded to form the base layer.
- The base layer is decoded and up-sampled to the original resolution to form a prediction frame.
- The full-resolution frame is subtracted from the prediction frame.
- Encode the difference which is the residual to form the enhancement layer.

With H.264/AVC Extension, in quality scalability, a single video sequence (QCIF or CIF) represents both the base and enhancement layers. Here, a virtual difference is embedded where quantisation, frame rates, bitrates of the two layers are altered. This will represent a varied quality.

In H.264/AVC Spatial Scalability which is implemented from two video sequences of dissimilar resolution, the two resolution layers are obtained by up-sampling or down-sampling to CIF (Higher) or QCIF (lower) resolutions format. The Lower/Higher resolution can include additional QCIF layers to provide quality scalability. The additional layers will be several enhancement layers for the particular resolution [5]. Figure 2 - 2 illustrates a spatial scalability of two layers QCIF and CIF. In each of the layers, an independent hierarchical coding structure with specific motion parameters for each layer is used [5]. The layers are coded at a frame rate of 15Hz and 30Hz for QCIF and CIF respectively. The prediction structures for the layers are aligned as shown in Figure 2 - 2. This is to allow a temporal scalable representation (section 2.4.2). The effective *GOP* size structures used as in Figure 2 - 2 are 4 and 8 for the QCIF and CIF layers respectively. Inter-layer prediction is employed in this structure as shown by the red arrows of Figure 2 - 2. Inter-layer prediction is used inside an access unit i.e. between the base and enhancement layer pictures at same time instance. The CIF layer frame rate is twice that of QCIF for the base layer, the enhancement layer frames of the highest temporal level do not use inter-layer prediction coding. These frames are only predicted by motion- compensated temporal prediction mode. At the decoder end, these layers are extracted using specific identifiers in the bit-stream header.

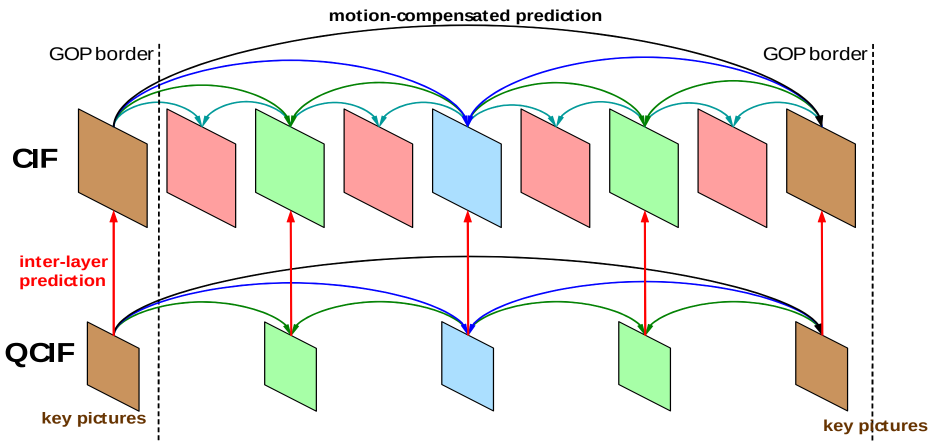


Figure 2 - 2 : H.264/AVC Extension Coding Structure with two spatial layers [6]

2.3.2. Factors influencing Scalability

The number of bits generated for a video stream will determine the video rate, quality and complexity of the control. Video encoder operates in two principles, it can either be in a variable or fixed bit rate [3, 6]. This affects the decoded quality and bit adaptation. In the first principle, the control parameters of the encoder are kept constant (e.g. motion estimation, search area, quantization step size, etc) and then the number of coded bits produced for each MB will change depending on the content of the video frame, causing the bitrates of the encoder output to vary. This generates a fixed decoded quality. The encoder produces more bits when there is high motion and/or detail in the input sequence and fewer bits when there is low motion and/or detail [3]. To guarantee a constant perceptual quality of the decoded sequence, it is necessary to keep a constant quantiser value during the encoding process [6]. With the Second principle, the control parameters of the encoder are varied (e.g. Quantisation Step Size etc) and then constant bits are produced for each MB. This principle might cause undesirable variation in the decoded quality of the video. The choice of one of the principles will highly depend on the channel and application characteristics, for example a constant bit-rate channel (such as circuit-switched channel) cannot transport a variable-bit-rate data stream and a packet-switched network can support varying throughput rates where the mean throughput at any point in time is limited by factors such as link rates and congestion [3].

In a bit rate regulation scheme, the video source might sometimes be required to decrease its output flow due to high traffic load across the network [6]. The reduction in bit rate could certainly lead to quality degradation since the quantisation distortion becomes more noticeable at lower bit rates. Although quality degradation from a congested network is much more unfavourable than that caused by a coarser quantisation process. The effects of network congestion are not tolerable in real time video services due to delay and data loss.

There are lots of research efforts exerted to establish efficient techniques for resolving congestion problems as described in [6]. A feedback control mechanism for flow control of video sources over the multicast backbone of the internet. In this rate control scheme, the rate control of a video encoder is regulated by modifying some encoding parameters as indicated by message that includes some statistics data such as average packet transmit time, average loss rate for multicast traffic, average packet delay etc. The sender collates this data and adjusts its output flow accordingly. Feedback mechanism which is a probing technique to solicit information and estimate the number of receivers in a multicast tree was then employed [6]. Some other research efforts produced reactive approaches such as error concealment and video data recovery schemes.

In this research, a Network Aware Scalability Decision Algorithm is employed in regulating the bit-rates inconsistency. This Algorithm passes on to the network a scalability stream that can be tolerated in conformity with the current reported network condition. It also passes into the network other bit-streams with reduced complexity. This includes a BL only, BL+ an EL, the usage of dynamic adaptation schemes and algorithms introduced in this research work (chapter five, six and seven), and most efficient designed parameters evaluated in chapter five. There are a number of bit-rate control methods which use different techniques in [6]. Methods for bitrate regulation are described in (section 2.3.3-2.3.7).

2.3.3. Variation of Quantisation-Step-Size Q_p

Output bit rate regulation can be achieved through a variable quantisation step size (Q_p) [6]. This is done by adjusting the Q_p of the next frame, *GOP* or MB based on the local buffer occupancy that is dictated by the status of the network. However, even though varying the Q_p affects the output bits, the average number of bits generated for each *GOP* or MB is not linearly dependent on Q_p. This is because the coefficient values change from one *GOP*, MB or frame to another.

In addition to that, the video content affects the number of bits required to code a video frame. Experiments showing a number of objective results produced from different sequences with the same coding technique are presented in chapter five. Figure 2 - 3 presents an experiment with two video sequences soccer and harbour encoded using the same technique and target bit-rate. A difference in the coding bits between the two videos is due to the natural contents of the videos. Hence the current produced bits should be considered before a quantisation value is adjusted. Traditional quantisation rate control techniques provide irregular and sometimes highly variable bit-rates thereby escalating the possibility of local buffer overflow that results in severe data losses in the case of network congestion. To realise a stable output, more efficient rate control algorithms need to be employed. In these algorithms, the buffer fullness and the video activity have to be considered to choose an appropriate Q_p so that the realised bit-rate is about the target bit-rate. The introduced Qp Constrain and Lambda Value Selection Algorithms (Chapter Five and Six) considered current computed bit-rate before applying bit-rate control on the bit-stream.

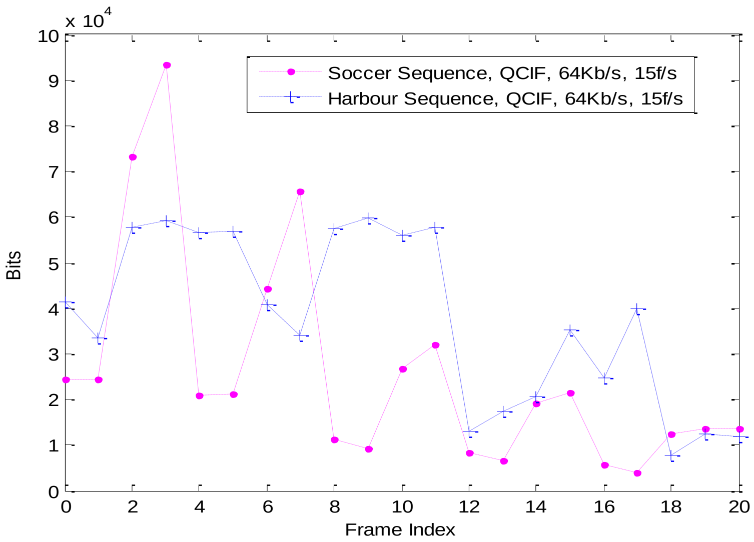


Figure 2 - 3 : Two video sequences with the same coding technique H.264/*t*+*q* showing variable scalability

2.3.4. Adjustment of Encoding Parameters

In parameter adjustment, a trade-off in quality and compression efficiency when controlling video rate is made [6]. Reducing the video rate could be done at the expense of degraded quality. There are four encoding parameters in block-transform coders which could be adjusted to control the output bit-rate. These parameters are described in [6] as follows:

Frame rate

Frame rate determines the number of encoded frames per second. This parameter can be modified to match the bit-rate requirements. It is generally used when the quality of pictures cannot be compromised since it targets the temporal and not the spatial redundancies of video signals.

The frame rate could also be referred or implemented as the time it takes the encoder to skip frame(s) to another frame. This has been used in the hierarchical temporal prediction used in this research work to generate several temporal layers for better scalability adaptation.

Encoding only a spatial portion

Another way of generating fewer bits at the cost of reduced quality is to encode only a spatial portion of each 8x8 or 16x16 pixels such as the diagonal coefficients (1x1), (2x2), (4x4), etc., or only the low-frequency coefficients of a macro-block. Lesser bits are then generated per block at the expense of reduced quality due to the subtraction of more video data. This could result in a compressed or thumbnail picture with a reduced size or detail.

The use of this technique can minimise the bandwidth requirement and hence support network adaptation. However applications might have preference or requirements for some other spatial formats and QoS. The implementation of a wider scalability in terms of user's wider demands will be more sufficient. A coded scalable video stream with multi-optional scalability adaptation which is an extended version of this technique will enhance the scalability system. Hence, this implementation will incur more complexity but will be more effective. For instance, the scalability technique *t+s* developed in chapter five supports low and high spatial formats and is adaptive to multi-channel environments. In a bandwidth limited situation, the technique (*t+s*) can include reduction of any of the formats to form another video layer that will deliver a reduced resolution picture (e.g. thumbnail). This will result in the reduction of bit-rates. However, it will demand extra processing although in a highly network deteriorated condition, it can be a preferred option. A similar method (Imp-*t+s*) is introduced in chapter six to enhance the BL adaptation. Experimental results show that, an equivalent picture quality can be obtained with a reduced bit-rate (chapter six).

Quantisation Parameter Q_p

This parameter controls the number of bits required to quantise output video code-words, such as transform coefficients. Increasing Q_p will result in encoding the DCT coefficients with fewer bits as more zero coefficients would be obtained (as a result of quantisation) prior to run-length coding. Applying lower Q_p values results in a wider encoding range and hence higher bit-rates. The operational adjustment of Q_p could be done on a frame, MB, *GOP* or coded layer basis. The Qp is adjusted frame by frame.

In adjusting quantisation parameter, careful consideration is required because coarser Qp values can result in degradation of the picture quality. Hence Qp values can be assigned on layer level requirements. This will ensure that, there is a provision for the bit-stream (video layers) to provide a maximum support for rate control during network congestion or for adaptation to diverse channels and application characteristics.

Motion Detection Threshold

This parameter is set to control the decision whether a *GOP* in a predicted frame (B or P frame) is coded with cod = 0 or skipped with cod = 1. If the threshold increases, the encoder becomes less sensitive to motion and thus the number of coded MBs decreases. Hence, the number of bits required for encoding a P-frame decrease at the expense of lower sensitivity to motion. Conversely, improved motion sensitivity with higher bit-rates is achieved from a lower motion threshold and larger number of coded MBs. In the same way, the intra and inter mode threshold could be used to control the output bit-rate of each coded MB in a frame. An improved bits generation is realised from intra frames with improved decoded quality at the expense of higher bit-rates. This is due to the absence of prediction in the coded intra frames.

The above discussed parameters could be used to adjust the bit-rates of an encoder for bit rate control and regulation. It is however observed that the design of a video coder includes design changes in the operational coding parameters and elements. Hence the parameter influences and the number of adopted parameters differ from one codec to another. This is due to added functionalities from one coder to another. The mentioned parameters in this section are used with earlier H.261, H.263, MPEG-4 standards and most with H.264/AVC codec. The usage of each parameter can change. The inclusion of more operational scalability functionalities in MPEG-4 and H.264/AVC has broadened the number of parameters used in coding process. Most of the additional parameters in H.264/AVC are designed or re-designed for better efficiency and adaptation to multi-channel environments.

Some of the parameters included in H.264/AVC codec which are not implemented in the prior standards include changes in deblocking filter (chapter three). In H.264/AVC, the design is brought within the motion-compensated prediction loop so that this can support inter-picture prediction to enhance the ability to predict other pictures) [7]. Inter-layer-Loop-Filter, inter-layer predictions for better flexible predictions to improve the adaptation in quality layers is also supported within H.264/AVC standard.

GOP structure flexibility to allow applications and multi-channel environments with higher demand of adaptation is supported in H.264/AVC. The flexibility is higher than prior standards such as MPEG-4.

Entropy mode (*CABAC*/*CAVLC*) is supported in H.264/AVC design to support context adaptation for enhance performance relative to prior standards. *CABAC* algorithm supports 5-15% bitrate reduction than *CAVLC*, uses low complexity methods using only shifts and look-ups [7]. Simulations and experiments are provided in chapter five for the evaluation of this feature.

The use of smaller size transform (4x4) in H.264/AVC improves both inter and intra predictions. This characteristic allows obtaining residual signal with less spatial correlation and provides efficiency in removing statistical correlation. Other distinct characteristics of H.264/AVC relative to prior standards are discussed in chapter three. These parameters are simulated, evaluated for objective and subjective quality assessment and real time performances in chapter five to establish their influence in scalable bit-stream.

Other introduced elements in H.264/AVC include Flexible Slice Structure which increases coding efficiency. Rigid slice structure is adopted in prior standards (MPEG-2). H.264/AVC design also supports the ability to partition the picture into slice groups (Flexible Macroblock Ordering (FMO)) so that each slice becomes an independently-decodable subset of a slice group [7]. This feature when effectively adopted can significantly enhance the robustness to data losses and scalability adaptation by managing the spatial relationship among the regions that are coded in each slice. This has been simulated in chapter five.

The most efficient parameter bit-stream that is simulated to perform better on real time can be used to adapt to a deteriorated network condition. Several methods in this research work are proposed and can be adopted depending on an application requirement as follows:

1. **Multi-Bit-streams-to-Multi-Channels Scalability Adaptation**:

A number of encoders are generated from one processor. These encoders are differentiated with different performance parameter bit-streams and then broadcast into the network. Hence, an automatic adaptation can be supported for heterogeneous networks and channels. Each of the encoder output unique scalability adaptation characteristics. Several network conditions can be tolerated from any of the bit-streams. Also variable decoders with variable resources can switch from one layer/stream to another layer/stream. This is because all the streams are encoded from one source. The methods of switching that can be used are discussed in chapter seven. This technique implementation is experimented in chapter five. Figure 2 - 4 illustrates this concept.

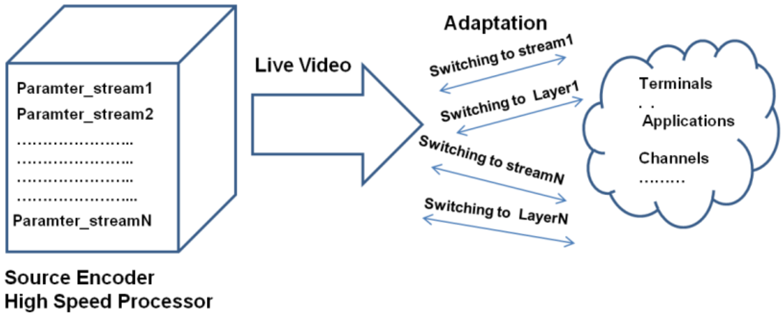


Figure 2 - 4 : A source encoder for multiple parameter scalability streams

In H.264/AVC Extension, each of the Video Coding Layer (VCL) Network Abstraction Layer unit consists of an identifier that refers to the content of the associated picture parameter set and each picture parameter set contains an identifier which refers to the content of the relevant sequence parameter set [7]. In this design approach, a small quantity of data (identifier) can be used to refer to a larger quantity of information (parameter set) without repeating that information within each one of the VCL NAL components. In this sense, the parameters set of a picture can be sent ahead of the VCL NAL components that they refer to and can be exchanged to provide robustness against loss of data.

The parameter sets can be hosted within the same channels that convey the VCL NAL components. This is called *in-band transmission*. In some other applications, it can be preferred to carry the parameter sets *out of band* using a more reliable transport mechanism than the video channel itself as illustrated in Figure 2 - 5.

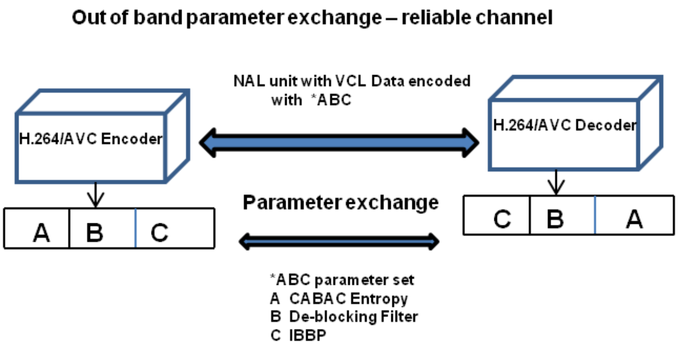


Figure 2 - 5 : Out of band parameter set exchange using reliable channels

2. A Dedicated Server

Dedicated server can be used to host the several parameter streams with different scalability and adaptation levels. An algorithm that allows monitoring and processing of network conditions is linked to the server (*SADMA* algorithm developed in chapter seven). At any condition of the network, a suitable stream can be retrieved from the server and then broadcast into the network. This method reduces the bandwidth and processor computational complexity obtained in the above method (*multi-bit-stream-to-multi-channels*). The transcoder is represented as *SADMA* (chapter seven) as illustrated in Figure 2 - 6. This method can be applied to broadcasting, videoconferencing and other real time applications.

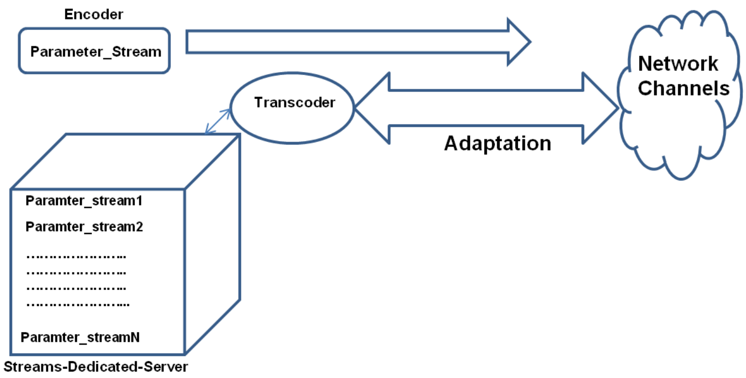


Figure 2 - 6 : A dedicated server hosting scalability streams

The parameter sets can be *hard-coded* or wired in a decoder implementation. Every coded slice can call up the relevant picture and sequence of parameters using one *CAVLC* or *CABAC* parameter_id for instance.

2.3.5. Multi- Layer Coding Rate Control

The output stream of a multi-layer coder consists of several layers of variable bit-rates and frame rates and optionally with different layer format resolution. This is done to achieve a scalable output in the case of diverse channel conditions. The output stream consists of a base layer (BL) and one or more enhancement layers (ELs). The ELs have different contributions to the decoded quality. The BL contains a minimum quality of the video and is mainly for the reconstruction of the video sequence, while the EL supports the improvement of video quality. Various layering and scalability techniques are discussed in sections 2.4.

Multi-layer stream for temporal and quality scalabilities are supported in MPEG-4 and H.264/AVC video coding standards. The video output of the codec consists of a BL and one or several ELs for every visual object plane in the sequence. These layers can be decoded at different bit-rates and frame rates. The multiple frame rates define the temporal activity within the video sequence.

In support for different types of spatial and temporal resolutions, MPEG-4 and H.264/AVC provide a special type of scalability that is due to the object-oriented coding of video frames. The coding of each object is decided at different quantisation levels and spatial resolution and frame rate. These values can be assigned or used according to the channel time varying bandwidth requirements. Table II - I shows the PSNR values and bit-rates for the *SL*, BL and EL coded at 3/6 f/s, 75/10 f/s and 5/10 f/s with H.264/AVC(t+q) codec. This experiment is conducted with foreman sequence using 150 frames. The results indicate that coding only the BL at 5/10 f/s achieved a reduction in bit-rate better than coding the sequence at 7/10 f/s. This is because the high frame rate coding requires and generates more output bits.

Table II - II shows a multi-layer scalability with MPEG-4 sequence. The objective quality values for *SL*, BL and EL are shown on the table. There are differences in bit-rates between the two codecs. H.264/AVC(t+q) produces better bit-rates because a quality scalability is used which employs better predictions since the video sequences used for the BL and EL are similar and are from the same video source. Also, the quantisation levels used followed an adaptive principle in which H.264/AVC usually yield better PSNR values. However, in other developed techniques in chapter six, MPEG-4 is better in bit-rates while H.264/AVC generates higher PSNR values.

It can also be observed that, from both MPEG-4 and H.264/AVC codecs, BL generates fewer bits than EL. The BL is independent while all other ELs depend on the BL. Hence, decoding the EL (BL+EL) will require additional bits. Real time simulations in chapter seven reveal the performance of BL and BL+EL. This explains why in *SADMA* Algorithm (chapter seven), the BL is allowed into the network for the reported condition where the BL+EL is not tolerated (experimentally proved in chapter seven). However, in some techniques like the one used for H.264/AVC($t+q$) in Table II - I, the bits difference between BL and EL is not large although for subsequent high level ELs (EL2, EL3…ELn) the bits difference could be larger. The reduction in bits between BL and EL is due to the efficient inter-prediction between the BL and the EL.

The H.264/AVC codec, which is the existing and most recent efficient scalable codec, is unique compared to the other standard video coding standards in a number of coding tools and usage efficiency in a multi-channel environment. These features are discussed in section 2.6.

Table II - I : *t+q* Scalability in the multi-layer H.264/AVC, foreman sequence

Frame rate (f/s)	H.264/***t+q* technique**			
	Bitrates(kbit/s)/Luminance(γ)dB			**Qp**
	SL	BL	EL	*SL* / BL/ EL
3/6	15.05/44.93	10.9/35.38	11.78/40.01	16.61 / 16.05/ 23.84
7.5	35.39/35.92	35.71/35.78	38.76/35.80	31.51/ 31.08/ 31.73
5/10	10.8/49.41	6.07/39.29	6.77/39.29	11.11/10.55/ 20.30

Table II - II : Multi- layer MPEG-4 scalability, foreman sequence

Frame rate (f/s)	**MPEG-4**			
	Bitrates(kbit/s)/Luminance(γ)dB			
	SL	BL	EL	Qp
3/6	18/30.96	28/30.89	33/30.92	13
7.5	47/31.01	32/30.97	57/31.04	13
5/10	37.5/31.0	25/30.94	44/31.00	13

2.3.6. Buffer Based Rate Control [BBRC]

BBRC [6] algorithm is a buffer based rate control technique. It was designed for MPEG-4 real time multimedia transmissions. The algorithm requirement was to achieve scalability at variable bitrates from 10 kbit/s up to 1 Mb/s as well as spatial and temporal resolutions. This technique includes, encoding of I, P and B frames, but its application is only limited to single visual object (VO) for rate control purposes. The algorithm assumes that the encoder rate distortion function can be modelled as follows:

$$R_b = A_1 \, x \, S \, x \, Q^{-1} + A_2 \, x \, S \, x \, Q^{-2} \qquad 2\text{-}4$$

Where R_b is the encoding bit count, S is the mean absolute difference (encoding complexity), Q is the quantisation parameter and A_1 and A_2 are the modelling parameters.

The BBRC algorithm executes its main processes in four stages which are initialisation, computation of the target bit-rate before encoding, computation of Q_p prior encoding, and updating the model parameters based on the results generated from coding the current frame. In the first stage, the algorithm checks the current frame prediction type is an INTRA (I) or INTER (P, B) frame. For INTRA coded frames, the initialisation part extracts the first and second order coefficients and Q_p is set to the initially configured parameter (from the user or application). Stages 2 and 3 are then skipped and the rate distortion model parameters are updated based on the current frame encoding results. The bits allocated to the header and the motion vectors are removed as they are not related to $Q_{p.}$ In the last stage, the algorithm checks the occupancy status of the current buffer. If it is below eighty percent, it continues to next frame, otherwise it skips the next frame and updates the buffer occupancy. In the case where the current frame is an INTER coded picture, then initialisation will be discarded and the algorithm continues to the bit-rate computation stage. At this step, the target bit-rate is calculated based on the bits available and the last frame encoded bits. To guarantee a minimal quality, a lower bound of target rate (R_b/30) is used. The target rate is tuned according to the buffer status to avoid overflow or underflow. The Q_p computation stage is activated after the target bit-rate has been computed. Quantisation parameter is computed based on the model parameters A_1 and A_2. Q_p is limited within the interval (1, 31) and can vary only twenty-five percent of the previous Q_p value to keep dissimilarity of quality under control. The result of the Q_p computation is used to calculate the new model parameters A1 and A2, and the procedure is continued. These procedures are summarised in the flow chart presented in Figure 2 - 7. The two modes intra and inter may not guarantee a good quality. The skipping of frames in the intra mode may result in bad quality especially where the sequence involves a high-speed motion. A method of minimising number of intra frames to decrease the bit-rate to a close value of the target bit rate will improve the video adaptation. The use of large *GOP* with intra frames at the beginning of every picture for synchronisation is a suitable method to decrease the bit-rates though quality can be degraded. Similarly, continuous adjustment of the bits with the inter mode may also result in unguaranteed picture quality due to delays and loss of bits. However, imposing a threshold (twenty five percent of the previous values) for the Q_p values as well as increasing the buffer size to meet application requirements will guarantee better quality.

The BBRC algorithm adopts skipping of frames and waiting until the buffer occupancy is less than 80%. This may not work with real time applications requiring a negligible amount of delay. Delays can occur in the case of a large data transfer and hence violating some applications requirements. Also the technique did not employ a frequent check-up for channel status before the frames are re-processed and the model is updated. Channel conditions can sometimes tolerate high bit rates which can improve the reconstructed video quality.

The BBRC algorithm employs skipping of frames when the buffer is more than 80% in the intra mode. In a situation where a large amount of frames are skipped due to a low bit-rate, the picture quality can be distorted. The adjustment of the target bit rates should have been preferred since skipping of frames does not account for which frames are discarded and how relevant they are to the overall video importance.

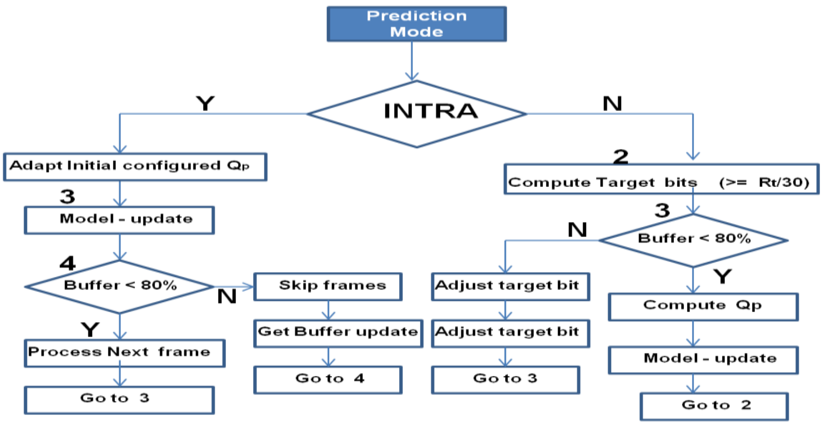


Figure 2 - 7: Scalable Rate Control Algorithm Procedures for intra and inter prediction modes

2.3.7. Feed-Forward Rate Control (FFRC)

Customarily, the quantisation parameter of the next frame is determined based on the number of bits produced by the previous frame. This operation is used for many variable-quantisation rate control algorithms. In feed-forward rate control schemes [6], the Q_p is obtained based on the number of bits required to code the prediction error of the present frame, MB or GOB. Transform coefficients and motion vectors hold most of the bits produced by block-transform coding algorithms with number of bits for transform coefficients being the most un-predictable. The number of bits necessary to code the transform coefficients depends on the resulting prediction error (residual matrix) and the Q_p. The prediction error per block for the present frame, which is vital for estimating the bit required to code an equivalent video data, can be obtained during the motion estimation process. The Q_p can be exploited to estimate the bits required to code a given prediction error value.

Last coded Q_p and bit-rate are used to decide the Q_p value of a current frame in FFRC algorithm. This is done to reduce the number of iterations required in the BBRC technique. Using the selected Q_p, the required bits to code the transform coefficients of the present difference frame is estimated using the prediction error for a block and the bit for error curves. The estimated bits are then utilised to estimate the bits required to code the motion vectors and the luminance coded block pattern. Let $Q_{p\text{-frame}}$ be the selected Q_p for the present coding frame, B_{tag} be the target bit-rate per frame and B_{tot} be the total bits allocated until the present MB + total required predicted bits to code the remaining MB [6].

If $(B_{tot}/B_{tag}) > T_1$ *and* $Q_P \leq Q_{p\text{-frame}} + 2$,

// Target bit-rate is high, decrease produced bit

Increase Q_p, *where* $T_1 > 1$.

If $(B_{tot}/B_{tag}) < T_1$ *and* $Q_P \geq Q_{p\text{-frame}} - 2$,

// Target bit-rate is low, increase produced bit

decrease Q_p, *where* $T2 < 1$.

else

$Q_p = Q_{p\text{-frame}}$

This algorithm procedure and definition yielded a better rate control technique compared to traditional variable Q_p techniques. To evaluate the performance of this control technique, a comparative study is conducted with a traditional rate control method implemented in the H.263 model. The frame based technique is modified to achieve Q_p regulation on a MB level in order to establish a fair comparison. Table II - III presents the regulated bit-rates and PSNR values of 4- ITU test sequences encoded with the default mode H.263 coder at a target bit-rate of 20Kb/s, a frame rate of 7.5 f/s and also using both FFRC and TMN5. The achieved bit-rates of the FFRC technique are closer to the defined target rates and the quality degradation is minimal.

Table II - III : Performance evaluation of conventional TMN5 and FFRC MB –based control algorithm for four sequences coded at 20Kb/s and 7.5f/s [6]

	Original TMN5		FFRC algorithm	
Sequence/frames	Actual bit-rate (kbit/s)	PSNR(dB)	Actual bit-rate(kbit/s)	PSNR(dB)
Foreman/240	20.33	28.27	20.01	28.14
Carphone/200	23.06	31.29	20.16	30.86
Suzie/149	19.91	32.65	20.13	32.81
Salesman/200	20.86	31.64	19.99	31.57

Further experiments are conducted with H.264 developed techniques foreman sequence at the same frame rate of 7.5 f/s and target rate of 20kbit/s and 240 frames. The experiment is conducted with the technique developed in chapter six with the H.264/AVC codec (*t+q*, *t+s* and *t+s+q*). The results of the experiments are shown in Table II - IV. The *t+q* technique has better PSNR values compared to FFRC and original TMN5 with a difference up to 6.13dB. Although FFRC and original TMN5 have better bit-rates with a difference of about 20.33kbit/s, this is because FFRC adopted a procedure to achieve a defined target rate before transmitting the video bit streams. With H.264, an algorithm is introduced in chapter five which supports bit regulation, achievement of target rate and adaptation to limited and bandwidth-hungry conditions. The algorithm used the current coded bits to decide the minimum Qp to be employed during rate control. However, the algorithm considers the bits regulation when the coded bits exceed 3000Kbs. This is because, at lower rates high reduction of bits may not favour the reconstruction of good picture quality. The algorithm yielded better results as presented for high bit rates (section 5.13). Also better quality values are achieved with *t+s* and *t+s+q* techniques of H.264 although the target bits are not achieved. Hence, it can be concluded that H.264 scalability techniques favour better PSNR quality values rather than reduced bit-rates which is a reverse case with respect to the FFRC algorithm whose main aim was to achieve the target bit rates. Carphone, Suzie and Salesman are not available in H.264 filter and therefore could not be used for a fair comparison with the FFRC algorithm.

Table II - IV : H.264/*t*+*s*, *t*+*q* and *t*+*s*+*q* performance at 20kbit/s and 7.5 f/s

	H.264/*t*+*q*-cif	H.264/*t*+*s*	H.264/*t*+*s*+*q*
Sequence/Frames/Layer	Actual bit-rate(Kb/s) / PSNR(dB)	Actual bit-rate(Kb/s) / PSNR(dB)	Actual bit-rate(Kb/s) / PSNR(dB)
Foreman/240/L0	35.10 / 34.40	36.41 / 30.81	26.32 / 34.01
Foreman/240/L1	40.07 / 34.41	74.45 / 28.90	28.75 / 34.04
Foreman/240/L2	NIL	NIL	145.80 / 38.00
Foreman/240/L3	NIL	NIL	152.05 / 37.63

It can be concluded that changing Q_p at an MB level achieves the best output bit-rate regulation. A disadvantage of the FFRC algorithm is a less stable perceptual quality as expected. The fluctuations in quality occur most drastically in periods of high video scene motion or activity when a higher number of bits are required for motion and error predictions. The limitation of achieving the target rate as the only scheme for bit regulation and adaptation is not adequate. The process requires an additional complexity when it is not needed as the network condition is not monitored.

2.4. Previous and existing scalability techniques

In this section, the state of the art scalability techniques are discussed. The past techniques from which the recent codec evolved are also described. The three main techniques of temporal, spatial and quality scalabilities that are used from MPEG codec up to the present technologies are described.

2.4.1. Temporal Scalability

Temporal scalability technique allows the coding of a video sequence at different frame rates. A bit-stream provides temporal scalability when the set of corresponding access units can be partitioned into a temporal Base Layer and one or more temporal enhancement layers with the following property [4]. Let the temporal layers be identified by a temporal layer identifier T, which starts from 0 for the base layer and is increased by 1 from one temporal layer to the next. Then, for each natural number k, the bit stream that is obtained by removing all access units of all temporal layers with a temporal layer identifier T greater than k forms another valid bit-stream for the given decoder. This explains how temporal scalability involves using a number of frame rates to achieve multiple scalability support.

The prior video coding standards MPEG-1 [7], H.262/MPEG-2 Video, H.263 [8], and MPEG-4 Visual [9] all support temporal scalability to some degree. H.264/AVC Extension [10] provides significant increased flexibility for temporal scalability because of its reference picture memory control [4]. It allows the coding of picture sequences with arbitrary temporal dependencies, which are only restricted by the maximum usable DPB size. Hence, for supporting temporal scalability with a reasonable number of temporal layers, no changes to the design of H.264/AVC were required.

Figure 2 - 8 illustrates the concept of temporal scalability for a QCIF video sequence. It is a similar process for high resolution format video sequence (CIF). A number of frame rates (F_0 – F_N) are used to code the video sequence. Hence, the coder produces a unique bit-rate (b_0 – b_N) for each of the frame rates. Each of the coded rates will stand for a unique scalability support within the video service environment. In our research work, we investigate and evaluate the effects of combining temporal scalability with another scalability technique. Combining the temporal scalability with another technique will improve the scalability and adaptation characteristics of the technique. This is because each of the technique delivers a specific behaviour and applications benefits as highlighted in chapter one.

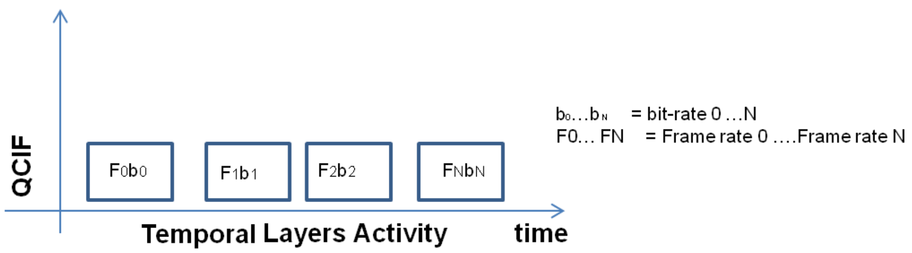


Figure 2 - 8 : Illustration of temporal scalability

Figure 2 - 9 is the result of the experiment with harbour sequence for temporal scalability. The result reveals the objective quality performance of H.264 codec temporal scalability. The comparison of the temporal scalability with temporal and spatial (*t+s*) or quality (*t+q*) reveal the additional benefits of the combination where more scalability layers are produced with a better objective quality. This is shown in Figure 2 - 10.

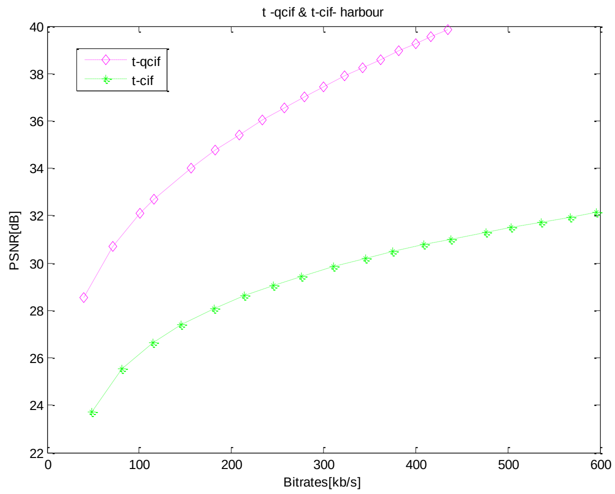


Figure 2 - 9 : Temporal scalability for QCIF and CIF (Frame rates = 0.9375Hz, 1.8750Hz, 3.7500Hz, 7.5000Hz, 15.0000Hz for QCIF/CIF with additional 30.0000Hz for CIF)

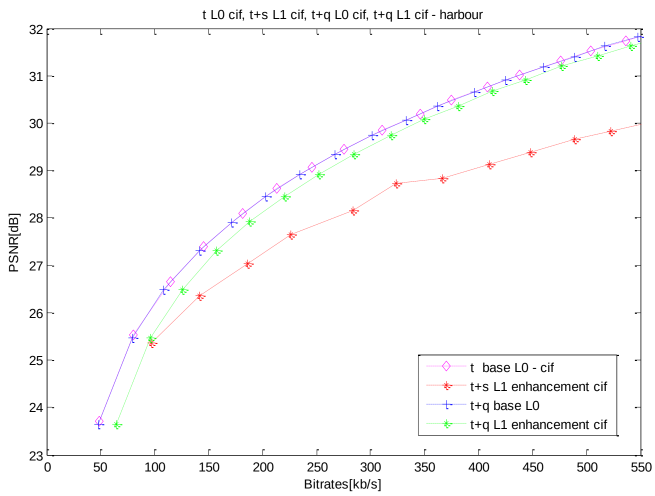


Figure 2 - 10 : Temporal scalability and its combination with spatial (*t+s*) or quality (*t+q*) scalability

2.4.2. Hierarchical Prediction Structures

Figure 2 - 11 shows temporal scalability with dyadic temporal enhancement layers. This structure can be efficiently provided with the concept of hierarchical B-pictures. From the Figure 2 - 11 illustration, the enhancement layer pictures are coded as B-pictures, where the reference picture lists 0 and 1 are restricted to the temporally preceding and succeeding picture respectively with a temporal layer identifier less than the temporal layer identifier of the predicted picture. Each set of temporal layers ($T_0,\ldots\ldots,T_k$) can be decoded independently of all layers with a temporal layer identifier T > k [4]. In the Figure 2 - 11, the set of pictures

between two successive pictures of the temporal layers group of pictures (*GOP*). For example, the frames representing the temporal layers from 4, 3, to 1 constitute a *GOP*. This *GOP* structure can be re-defined to take other number of frames 2,4,16, 32 0r 64. The size of the *GOP* will determine the number of temporal layers and number of key pictures numbered 0 and 1 in *GOP* 1 and *GOP* 2 respectively. Hence, the *GOP* structure will determine the temporal scalable layers. Key picture 1 (KP1) which is assigned at the beginning of every *GOP* is an inter-frame and therefore if an encoder is designed for a small *GOP* size, there will be large number of intra-frames generating excessive bits. However this might result in a good picture quality but the network scalability adaptation will be limited. In the case of excessive network delays, there may be errors and loss of packets in the bit-stream which can result in an un-guaranteed picture quality. The selection of the *GOP* structure is a trade-off between scalability adaptation and picture quality. This selection should therefore depend on the intended applications and the network environment.

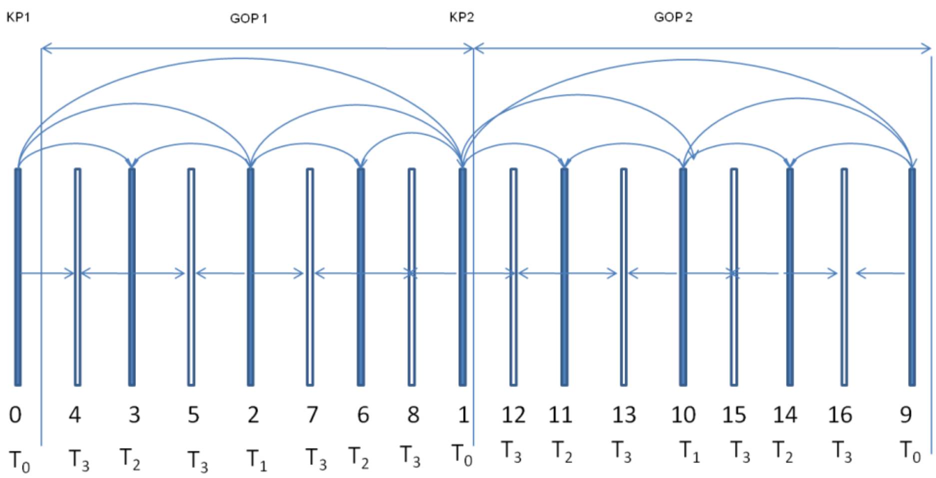


Figure 2 - 11 : Hierarchical prediction structure for enabling temporal scalability with B-pictures.

2.4.3. Spatial Scalability

Spatial scalability technique is implemented by coding a video sequence into two layers at the same frame rate but different spatial resolution formats. The BL is coded at a lower resolution format. The reconstructed BL picture is up-sampled to form the prediction for the high-resolution picture in the EL. Figure 2 - 12 illustrate the concept of spatial scalability.

To support spatial scalable coding, SVC follows the conventional approach of multilayer coding which is also used in H.262, MPEG-2 video, H.263 and MPEG-4 Visual [4]. Each layer corresponds to a supported spatial resolution format (QCIF, CIF, 4CIF etc) and is referred to as spatial layer or dependency identifier D. The identifier D for the base layer equals zero and is increased by one for subsequent identifiers. Motion-compensated prediction and intra-prediction are employed in each of the spatial layer identifier. Additional so-called inter-layer prediction mechanisms can be incorporated in order to improve the coding efficiency in comparison with simulcasting with independent layers [4]. In order to restrict the memory requirements and decoder complexity, SVC specifies that the same coding order is used for all supported spatial layers. The representations with different spatial resolutions for a given time instant form an access unit and have to be transmitted successively in increasing order of their corresponding spatial layer identifier D. Lower layer pictures do not need to be present in all access units as illustrated in Figure 2 - 13.

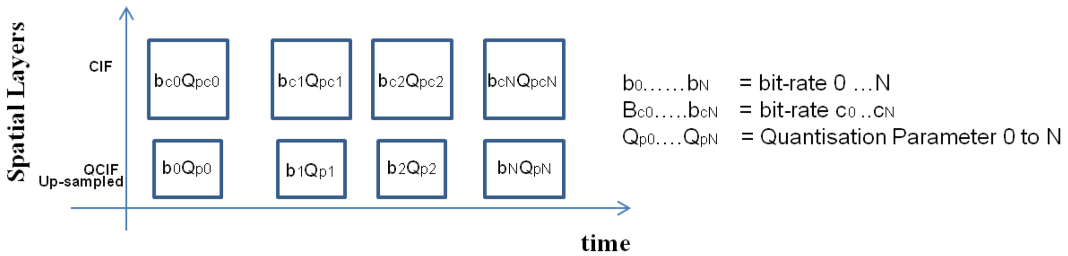


Figure 2 - 12 : Illustration of Spatial Scalability

The support of inter-layer prediction within the spatial pictures can allow a combination of temporal and spatial scalability. This will enhance the scalability level and adaptation to diverse terminals and users' preferences and requirements. With the introduction of spatial scalability in the temporal scalability discussed above, a significant benefit is derived including a range of spatial resolution format access. Each spatial layer can generate multiple scalable layers with a unique adaptation and scalability features. This technique is experimented and evaluated in chapter six.

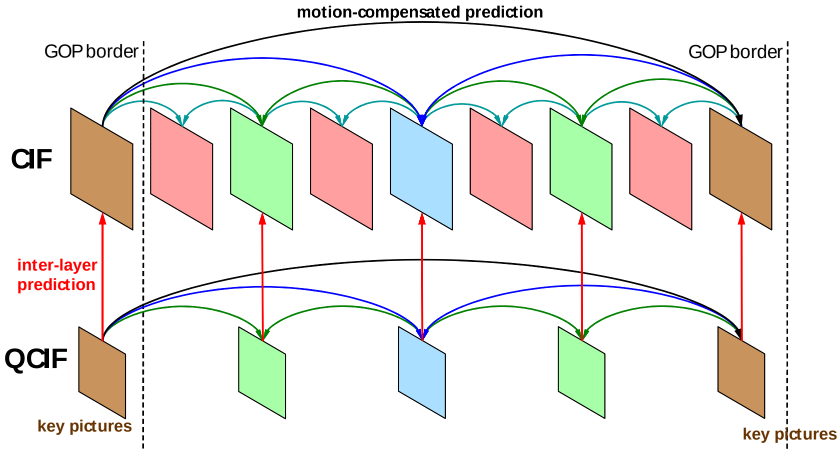


Figure 2 - 13 : Coding structure with two spatial layers [6]

Figure 2 - 15 represents an experiment conducted with H.264 codec for spatial (s) scalability. The results present both QCIF and CIF formats. Hence, the bit-stream supports adaptation to low and high resolution users' preferences.

Another experiment is conducted with the results in Figure 2 - 16 showing an additional temporal scalability. The inclusion of temporal scalability has improved the adaptation of the bit-stream with better rate distortions and scalability levels. This scalability technique is discussed and evaluated in detail in chapter six. Another possible combination is spatial and quality scalability which is also discussed in chapter six. Figure 2 - 14 illustrates spatial scalability with two video sequence formats QCIF and CIF. Each of the formats is encoded with different quantisations and bit-rates.

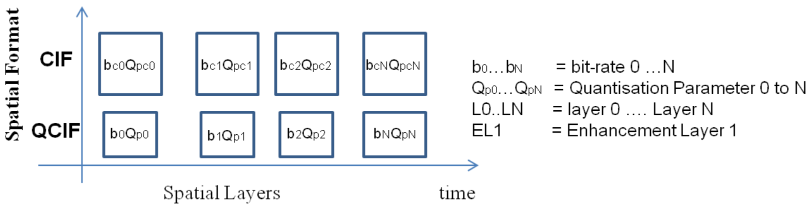


Figure 2 - 14 : Illustration of Spatial Scalability

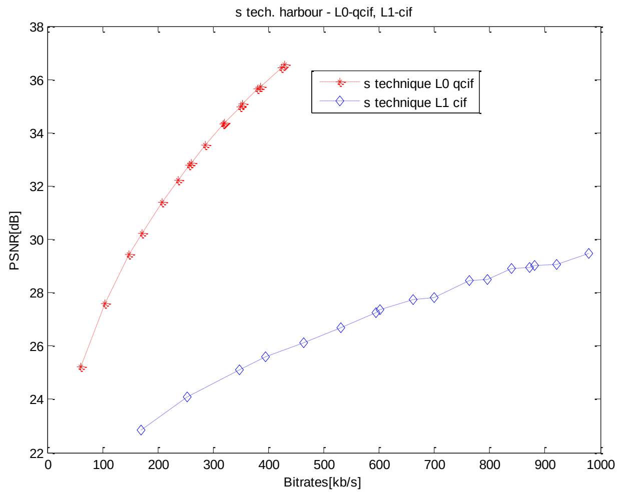


Figure 2 - 15 : Spatial Scalability with QCIF and CIF resolution format

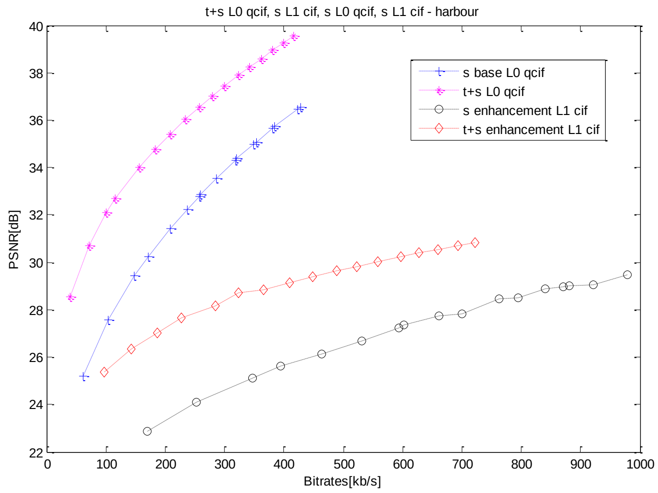


Figure 2 - 16 : combined temporal and spatial (*t+s*) scalability and spatial (s) only scalability.

Figure 2 - 17 shows two spatial scalability experiments output with mobile and harbour ITU standard sequences. The output for each experiment shows two spatial layers with CIF and QCIF format resolution. This has given decoders with variable display characteristics access to the spatial scalability stream.

a. Mobile, Frame 100, 544Kb/s, CIF b. Mobile, Frame 100, 512Kb/s, QCIF

c. Harbour, Frame 100, 576Kb/s, CIF format d. Harbour, Frame 100, 544Kb/s, QCIF

Figure 2 - 17 : Spatial Scalability showing two spatial formats for mobile and harbour sequence experiments

2.4.4. Quality Scalability

Quality or SNR scalability is a special case of spatial scalability with identical picture sizes for base and enhancement layers [4]. It is similar to the spatial scalable coding with the exception that all layers have an identical spatial resolution with different quantisations and bit-rates. Figure 2 - 18 illustrates the concept of SNR scalability.

In SNR scalability, a limited number of extractable rate points are included in the bit-stream [5]. There are two different possibilities providing SNR scalable bit-stream namely course-grain scalability (*CGS*) and Medium Grain Scalability (*MGS*). *MGS* Coding supports many more rate points in addition since an arbitrary set of *MGS* layers packets may be extracted. Furthermore, the encoder may choose to partition the transformed coefficients of a layer into and up to 16 *MGS* layers which increases the number of packets and thus the number of subsets of packets and a finer granularity is achieved [5]. *MGS* provides improvements in flexibility, bit-stream adaptation and error robustness. *MGS* is a modified high level signalling which allows a switching between different *MGS* layers in any access unit and the so called key picture concept which allows the adjustment of a suitable trade-off between drift and enhancement layer coding efficiency for hierarchical prediction structures. With the *MGS* concept, any enhancement layer NAL unit can be discarded from a quality scalable bit-stream, and thus packet-based quality scalable coding is provided [4].

SNR scalable video can employ inter-layer prediction mechanisms as employed for spatial scalable video, but without using the corresponding up-sampling operations and the inter-layer de-blocking for intra-coded reference layer macro-blocks. Furthermore, the inter-layer intra- and residual-prediction are directly performed in the transformed domain. When utilizing inter-layer prediction for coarse-grain quality scalability in SVC, a refinement of texture information is typically achieved by re-quantizing the residual texture signal in the enhancement layer with a smaller quantization step size relative to that used for the preceding *CGS* layer [4].

Conclusively, with the adoption of SNR scalability, there is support for providing multiple levels of QoS in the bit-stream. Applications and channels availability will tolerate a level of the quality at a particular instance. The categorisation of quality levels is achieved by encoding the temporal layers with a several levels of quantisation, bit-rate and frame rate. Figure 2 - 20 and Figure 2 - 19 show Quality (SNR) scalability experimented with the “BUS” standard ITU sequence producing several number of quality levels. Users with different resources can access and handle the scalable stream.

SNR scalability can be combined with other types of scalability, for example SNR+temporal, SNR+spatial or SNR+temporal+spatial. The latter combination of scalability will provide full combination of the available scalabilities. The performance of the various and possible combinations are developed, discussed and evaluated in chapter six. The evaluation involves both objective and subjective assessment of the scalability techniques products and real time simulations to establish the performance of the techniques over heterogeneous networks.

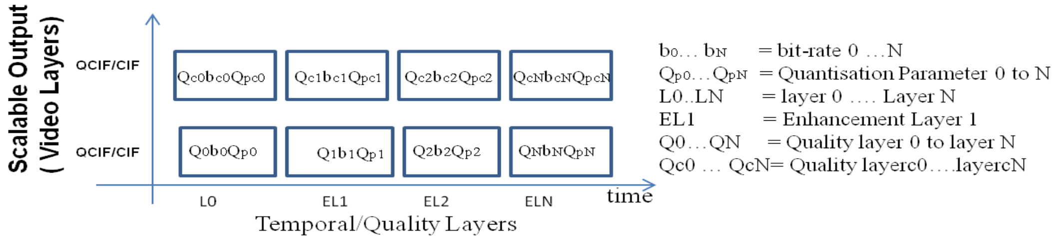


Figure 2 - 18 : Quality or SNR Scalability

a. 75.27Kb/s, 23.96dB, Qp=46.08 b. 121.23Kb/s, 26.19dB, Qp=41.92 c. 193.58Kb/s, 28.52dB, Qp=37.76

Figure 2 - 19 : Quality level products for quality scalability (QCIF resolution format)

a. 123.48Kb/s, 22.06dB, Qp=51 b. 203.53Kb/s, 24.53dB, Qp=46.15

c. 259.41Kb/s, 25.60dB, Qp=25.60

Figure 2 - 20 : Quality level products for quality scalability (CIF resolution format)

2.4.5. Fine Granularity Scalability

For a decoder or user's application and decoder to adapt to the fluctuations in the bit-rate of the connecting channels, a unit in a video bit-stream such as a frame or a MB is divided into small items [11]. A measure of the quantity of items comprising a unit is referred to as its granularity. The initial item of each division contains the basic and coarsest part of the data and the remaining items contain refinements to the base item. The technique of gradual refining and improving the granularity of a division is known as Fine Granularity Scalability (FGS). The multi-layer scalability operating on multiple techniques introduced in the developed techniques in chapter six supports Fine Granular Scalability (FGS). The availability of several frame rates, bit-rates, picture sizes allows gradual improvement and refinement of the scalability. Hence, a higher video quality is obtained through increasing the number of layers decoded at the receiver end. However, every decoder has limitations on the number of layers that it can decode and reconstruct.

2.5. Other Scalability techniques and methods

In this section, earlier implemented scalability and adaptation techniques and methods are described. Scalability originated from the use of multiple layers coding to produce layers of different bit-rates, resolutions and frame rates. The coded video stream consists of a BL and one or more enhancement layers (ELs). Each of the layers supports different decoded quality. The BL provides the basic quality required for the reconstruction of the video sequence while the enhancement layer improves the quality of the BL. The following sections discuss H.261, H.263 and H.264/AVC (H.264L) regarding the concept of scalability and adaptation techniques implementation in these standard video codecs.

2.5.1. H.261 Adaptive two layer technique

H.261 was the first commonly used standard codec for videoconferencing [3]. The standard was developed by ITU to support videoconferencing and video-telephony over ISDN circuit switched channels. Circuit switched channels operate at multiples of 64kbit/s and H.261 was designed to provide computationally simple video coding for these bit-rates.

Two-layer video coding technique was proposed and applied to H.261 for transmission over ATM networks [3, 6]. Low quality picture is produced from the BL at low bit-rate while the second layer (EL) is the coded difference between the original video and the output of the BL. In order to sustain graceful degradation, data from the BL is transmitted with high priority using the guaranteed channels of an ATM network. The generated packets from the second layer are transmitted through the non-guaranteed channels and vulnerable to bits loss if congestion occurs. However, if the second layer is received the decoded quality of the picture will be improved. Figure 2 - 21 illustrates the concept of this technique.

The adopted technique in H.261 is inefficient when it operates over a pervasive network environment. The adaptation of this technique is very limited and constrained to specific channels only. The coding separation between the BL and EL can render a bad picture quality. This cannot support many real time and non-real time applications where the picture quality is a preference or requirement. Although it is applied to videoconferencing, it is not effective. There is a need to broaden the technique to support multi-channels and applications. This has been effective with the techniques developed and evaluated in chapter six and algorithm in chapter seven. The evaluated techniques over heterogeneous channels show robust scalability and adaptation.

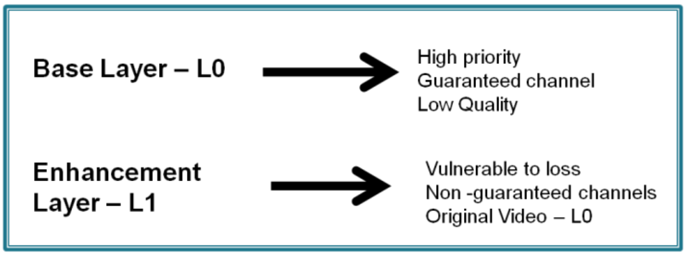


Figure 2 - 21 : H.261 two layer adaptation technique

`

2.5.2. H.263 Adaptation and Scalability

The ITU-T group invented H.263 as an attempt to improve on the compression performance of H.261. This provides basic quality at low bit-rates less than 32kbit/s. H.263 is a part of a suite of standards designed to operate over a wide range of circuit and packet-switched networks [3, 6]. The baseline of H.263 standard model was adopted as core of MPEG-4 Visual Simple Profile.

The application of Video Scalability on the H.263 standard consists of a BL and an EL. The EL is presented with different temporal and spatial scalabilities. The BL consists of low resolution QCIF format and coded at a frame rate F1 while the EL consists of high resolution CIF format coded at a frame rate F2. The BL video data is protected with a more powerful error protection scheme than that of the EL. This ensured a guaranteed minimal perceptual quality in the case of packet loss. This feature of error protection scheme has given it wide flexibility to operate over many channels, rather than just a single channel as in H.261.

However, there is still a need to improve the scalability as the EL video data is not protected. Hence if EL data is not received, picture quality may not be guaranteed. Figure 2 - 22 illustrates the concept of H.263 Scalability.

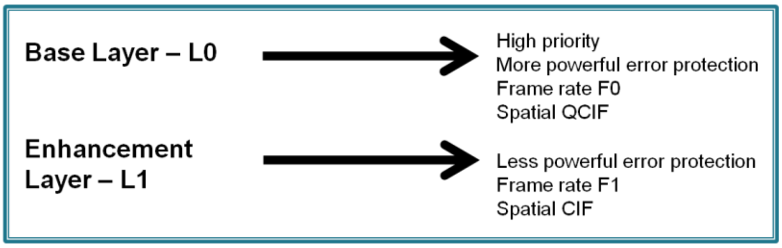


Figure 2 - 22 : H.263 Adaptation and Scalability

2.5.3. H.263 Extensions.

In late 1990s, the video coding Experts Group (VCEG) of the ITU formed two groups of activities [1, 6]. The main objective was to develop very low video coding standard at bit-rates lower than 64Kb/s and specifically at less than 24kbit/s. One group was to look at a coding standard for low bit-rates which is named H.263+. Work on the H.263+ activity continued for further improvement on H.263+ under the name H.263++. The other group of activities that is more related to MPEG-4, is the work on advanced low bit-rate video coding under the name H.26L/H.264. The H.263+/H.263++ work is intended for short term standardisation of EL of the H.263 video coding model for conversational, telecommunication and non-conversational services. The H.26L is aimed at producing of new video coding technology beyond the capabilities of improvements made to H.263 by the H.263+/H.263++ coding model.

The enhancements made on H.263 are in two categories (1) enhancing quality within existing applications (2) broadening the current range of applications. Specific examples involve improving perceptual compression efficiency, reduction of video coding delay and provision of greater resilience to bit error and data losses. In addition, H.263+ has all the features of H.263.

2.6. H.264/AVC and Extended Features

The long term objectives of the ITU Video Expert Group is to provide a video coding recommendation which at very low bit rates can perform substantially better than that achievable with the existing standards (H.263+ etc). The current and most recent adopted technology is H.264/AVC standard [3, 1].

The design requirements of H.264/AVC Extension include the adaptation of a generally simple straight forward design using well known building blocks such as use of a minimal number of VLC tables for all parameters to be coded. High compression performance with 50% or greater bit rate savings from H.263 V2 or MPEG-4 advanced simple profile at all bit rates, Flexible application to delay constraints appropriate to a variety of services such as low delay for real time conversational services (e.g. no B-pictures), network friendliness, error resilience, ease of packetizing information priority control, higher delay usage appropriate for storage or server based streaming application, packet loss resilience and mobile channel corruption resilience were all design objectives of the H.264\AVC Extension requirement [12]. Most of these requirements are included in the current version except rate control for an SVC stream which control and improve network friendliness. The design of high quality applications characterised by improved bit rate performance, file storage support, support of multiple streams with transitions, random access, simple stream exchange and HTTP streaming service are also achieved goals of the H.264\AVC Extension. Some other techniques which aid in supporting network friendliness are developed in chapter five and six (*Constrain Qp Algorithm, Lambda Value Selection Algorithm* etc).

As one of the requirements for H.264\AVC is to adopt well known building blocks from the standard codec, some of its features are similar to that of a generic MPEG encoder. These features include coding of 4:2:0 pictures, division of pictures into slices and macro-blocks, macro-block is 16x16 pixels for luminance and 8x8 pixels for each colour difference signal, block-based transform coding, quantization and scanning of transform coefficients, spatial predictive coding of some values, temporal prediction using motion estimation and motion vectors, I-, P- and B- Pictures, lossless coding of coefficients, motion vectors etc. [13]. Some of the concepts that are new but extended based on extensive experience of their use in MPEG-2 and H.263 include pixels and residuals transformed in 4x4 blocks in which the previous standards used 8x8 blocks, new integer transform, additional 2x2 transform for DC coefficients of a macro-block, logarithmic quantisation, de-blocking filter, extended use of spatial prediction, tree structured segmentation for motion estimation, more accurate motion vectors where quarter pixel is used, multiple reference pictures especially in temporal predictions, weighted averaging, more efficient macro-block-adaptive frame/field coding and improved lossless coding.

H.264\AVC Extension has improved in many of the features mentioned earlier. In spatial prediction, MPEG-2 uses spatial predictive coding for the DC coefficients within a slice and prediction is always from the previous macro-block horizontally. MPEG-4 added prediction of the first row and column of the AC coefficients. H.264/AVC engineers made extensive use of adaptive quantization techniques and provide a complex array of prediction mechanisms. Complex 16x16 macro-blocks may be predicted in one of four modes, including a plane predictor. Alternatively, each of the 16 4x4 blocks within a macro-block may be predicted in 10 different modes, with the ability to use any previously decoded nearby block as the predictor. One of the obvious artefacts for DCT-based systems such as JPEG and MPEG is blocking. H2.64 introduced a de-blocking filter. De-blocking filter has been proposed from many decoder designers to reduce the visibility of blocking. However, this is the first time that such filter has been a normative part of the compression standard which is incorporated in a compliant decoder. In H.264/AVC, the filter is part of the decoder within the prediction loop of the encoder. So, any changes in the output that result from the de-blocking filter are included in the comparison for motion prediction, and in the calculation of prediction residuals. The filter is highly adaptive to the image content, and to the compression processes being used for the particular area of an image. H.264/AVC made an excellent extended modification in specifying quarter pixel resolution for motion vectors which was also used with MPEG-4 but does not provide improved result. In H.264, the half pixel interpolation is performed with a six-point filter providing excellent accuracy, and then the second interpolation to quarter-pixel values uses two point interpolations. This approach seems to offer the best improved result. The most obvious change in H.264 temporal prediction is that, it permits multiple pictures reference. The syntax permits up to 15 and the reference pictures are identified by a reference picture index. H.264 provides more efficient lossless compression which is the final compression step in any encoder. H.264 adopted Context-Based Adaptive Variable Length Coding (*CABAC*) algorithm which is an improved version of variable length coding technique used in earlier standards. H.264 was extended to provide 4:2:0, 4:2:2, 4:4:4 formats and handle sample precision up to 12 bits [12].

2.7. Scalable Video Adaptation Concept

Video adaptation is one main application of scalability. It employs scalability and manipulations at various levels to transform video inputs to a video output in augmented multimedia form. This is done in order to meet diverse resource constraints and user preferences while optimising the overall subjective quality of the video. In diverse multimedia environments, clients may access and interact with the media content on variable types of terminals and networks. Such environment is made up of a heterogeneous network comprising TV channels, mobile phones, computer processors and many other clients and devices. There is a critical need in such a ubiquitous environment to handle the huge variation of resource constraints such as display characteristics, memory capacity, CPU speed power and bitrates.

Video adaptation is an emerging field that includes various techniques such as scalability techniques responding to the above challenges [14]. A video adaptation system adapts one or more video components to generate a new presentation with a video format that meets diverse client's needs while scalability adaptation is a system that supports the adaptation of various scalability techniques through a defined mechanism (e.g. through channel monitoring and selection of suitable techniques). The original video format is customised so that multiple numbers of decodable bit-streams are embedded in the video output. Figure 2 - 23 illustrates the concept of video adaptation in a diverse-resources media environment. It takes into account information about content characteristics, usage environments, user preferences and digital rights situations. The objective is to maximize the QoS demands of the final terminal and client presentation while satisfying a range of constraints and conditions.

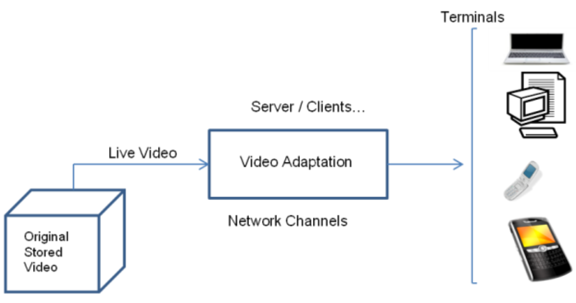


Figure 2 - 23 : Video adaptation in a diverse media environment supporting heterogeneous networks and terminals

The most challenging and required solution in response to the pervasive media environments characteristics is the need to establish a dynamic adaptation technique. This has become the last solution due to the fact that terminals are not predictable. Terminals might dynamically lose or change resources, programs run on different servers which can be induced with errors and therefore the processing capacity may be affected and new applications can introduce new demands on the network. In view of these points, a dynamic scalability decision making algorithm is invented in this research work. The algorithm is discussed and described in chapter seven and six.

2.7.1. Unified Adaptation Conceptual Frame Work

Many complex issues are involved in the design of video adaptation systems. The conceptual framework for adaption design clarified and unified various interrelated issues. This is illustrated in Figure 2 - 24 where the framework consists of a systematic design procedure and taxonomy for classifying dissimilar adaptation techniques. The components and issues related to the design framework are discussed as follows in the following bulleted points [14]. In [15], an encryption based technique is used to authorise the consumption of the video content to the clients.

- **Entity:**

Entity referred to the basic unit of video that undergoes the adaptation process. The form of entities may exist at different levels, such as pixel, object, frame, shot, and scene, syntactic and semantic components. In the case of this research work, entity is represented as a video layer which is a set of video frames that is made up of a particular video sequence with its own quality and adaptation characteristics.

Different adaptation operators can be defined for different types of entities. For instance, a video can be reduced in resolution, spatial quality, or skipped in order to reduce the overall bandwidth. In the case of this research work, we present a number of resolutions in the video stream by presenting several layers with specific formats. Each layer will contain a spatial resolution format where the higher format is predicted from low resolution format layer in order to reduce the bandwidth requirement. Hence variable terminals with different spatial characteristics can decode from the video stream. A semantic component such as a text from a newspaper in a visual or textual and a one subset of shots such as highlights in a sequence can be removed in order to generate a reduced version of the video.

- **Adaptation Space:**

Adaptation space is defined as the space of feasible adaptations for a given video entity. The framework in each of the dimensions represents particular adaptation operations and a point in the space represents a combination of operations from different dimensions. For example, a popular technique for transcoding inter-frame transform coded video includes two dimensions: (1) Dropping a subset of transform coefficients in each frame and (2) Skipping a subset of frames in the video sequence. In the design algorithm in chapter seven, an example can be (1) discarding all the higher layers, delivering only the base layer (2) switching from *t+s+q* technique to *t+q* for network channels adaptation.

- **Resource Space**

Each entity is associated with certain resource requirements and utility values. The operation for any adaptation transforms the entity into a new one and thus changes the associated resources and utility values. There are many multiple dimensions that exist in the resource and utility spaces. Resources examples include bit-rate, display characteristic (spatial resolution and colour depth for example), processor speed, power and memory. In this case, the concentration is on the available resources in the usage environment or the delivering network channels. The information detailing the usage environment limitations can be used to derive implicit constraints and limitations for determining suitable adaptation operations. In this case, the developed and evaluated scalability techniques in chapter six can be adopted for a particular application or network channels environment due to their embedded adaptation control.

- **Utility Space**

Utility Space represents the QoS or users satisfaction of the video content. In this research work, utility values are measured in different levels, namely the objective level and subjective level of quality. The objective level is represented by PSNR (peak-to-signal-noise-ratio) and the subjective level is represented by scores on the users' visual perception of the videos.

The utility value of a video sequence is not fixed and depends on the coding values and method. This value is heavily affected by the user preferences. In our scalable techniques, different quality levels are embedded in one video bit-stream so that users can extract their preferences. The users' preference is particularly true for the subjective and semantic level utilities. The subjective utility value of a video sequence depends on the users' needs for their current task. Hence, dynamic adaptation is required for effective channel control and video services since users' tasks and clients' demands vary. In addition to the implicit constraints set by the resource limitations, the user preferences may also be used to set explicit constraints on the feasible adaptation operations. For instance, if the user may prefer to view videos within a window no larger than a fraction of the screen size, though the actual resolution is not a limiting factor for the full-sized video. This can be achieved either at the decoder end or embedded in the bit-stream where a particular layer characterised by the user preference can be identified and extracted. In most current codec technologies, this particular type of preference is achieved and processed at the decoder end. However, this and similar operations can be achieved from the encoder end although this might not be a preference for most users or terminals.

For a given video sequence, the interrelationship among the adaptation space, the resource space and the utility space represents critical information for designing effective adaptation techniques. In Figure 2 - 24, the cube in the resource space represents the resource constraints imposed by the usage environment. There exist several adaptation techniques and solutions that satisfy the constraints – referred to as *resource-constrained permissible adaptation set*. For instance, each of the developed techniques in this research work can effectively satisfy a specific application better than the others. Similarly, different adaptation operators may fall in the same utility space value. The main objective is to choose the optimal adaptation technique with the highest utility value or the minimal resource while satisfying the resource constraints.

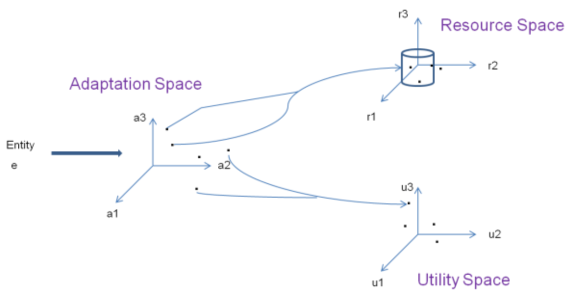


Figure 2 - 24 : A Conceptual Frame Work for Video Adaptation Design and Interrelated Resources

2.7.2. Designing Video Adaptation Technologies Systematic Procedures

The described conceptual framework in Figure 2 - 24 can be used for guidance to the design process of efficient practical adaptation techniques or solutions. The procedures discussed below utilize the concepts and interrelationship among adaptation, utility and resource spaces [14].

- **Identification of the required and available entities for adaptation** e.g. frame, sequence of shots etc. In the design of developed techniques in this research work, we consider bit-rate and its associated delivered objective and subjective quality, number of video sequence layers, number and level of achievable frame rates, level of scalability performance etc.

- **Identification of possible adaptation operators**. For example re-quantisation, frame dropping, shot dropping, replacement etc and their associated parameters. In this research work, we employ dropping of higher layers, effective re-quantisation value during rate control, use of most efficient coding parameters, use of inter-layer prediction to reduce the required bandwidth, use of dynamic de-blocking filter algorithm and choice of an optimum value with respect to current coded bits for lambda for an improved rate distortion optimisation. Other operators used to improve the scalability are the choice of *GOP* hierarchy to support adequate adaptation in a pervasive network environment.
- **Development of models for measuring and estimation of the resource and utility values** that are related to the video sequence and processed or coded using the identified and evaluated operators. In this case, the research work employs H.264/AVC Extension simulator as the primary codec to run the proposed and identified operators. The objective and subjective quality performance is measured. A heterogeneous network simulator is also built and validated. The simulator is used to characterise and evaluate the proposed scalability and adaptation operators, techniques and algorithms for real time performance. The detailed discussions about these designs are provided in chapters five, six and seven.

- **Development of mechanisms and strategies** to deduce the optimal adaptation operator(s) satisfying given user preferences and constraints. The strategies employed include several experiments for a given operator to identify its effectiveness and scalability performance. The scalability performance can reveal the suitability of the operator or technique for a given application or user given the constraints and preferences. In terms of QoS levels in which different users and applications will have different preferences, each given range of operator value, for example bit-rate, is objectively and subjectively evaluated. This is to enable the establishment of the quality performance of the given operator, algorithm or technique.

From the details of the above adaptation design procedure, many video adaptation design solutions can be formulated as follows. Given a content entity (e_1), user preference, and resource constraints (C_t), deduce the optimal adaptation, $a_{opm,}$ within the feasible adaptation region so that the utility of the adapted entity e_1 is maximized. Comparable to this, we can formulate other problems in a symmetric way exploring the *utility-constrained allowable set* to discover the optimal adaptation operator to satisfy utility constraints while requiring minimal resources.

2.8. Video Adaptation Solutions

There exist several numbers of adaptations, operations and manipulations in the literature defined and implemented as adaptation solutions. In this section, most of the existing and past adaptation manipulations are presented as follows [14].

2.8.1. Format transcoding

Transcoding from one format to another is one of the basic adaptation processes in order to achieve compatibility with a new usage environment. This happens due to different formats prevailing in different application sectors such as consumer electronics, internet streaming and broadcasting. However, this implementation may sometimes not be feasible due to the cost of computation which increases system complexity or quality degradation.

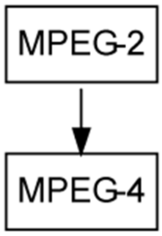


Figure 2 - 25 : Format transcoding

2.8.2. Selection and Reduction

In a highly bandwidth constrained environment, the most popular technique is to trade some components of the video to save the available resources. Such types of methods are implemented by selection and reduction of some of the video elements, for example frames, pixels in image plane, bit-plane in pixels, frequency components in transformed representation, etc. The considered techniques will result in changing the bit-rate, frame-rate or resolution format of the video stream. Figure 2 - 26 represents selection and reduction technique by reducing the image to a thumbnail.

However, this method may not satisfy the preferences of all users and applications. Also, reduction of some part of the video items may degrade the picture quality or details as may be required from some other applications.

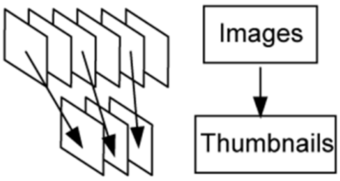


Figure 2 - 26 : Selection and reduction

Another method of adaptation is the one introduced in this research work. The adaptation reduces the complexity of the current video stream by extracting a BL, or BL+EL to suit particular network characteristics. The method can also switch to another technique of less or high complexity based on the predicted stability of the network environment. The transcoding processor generates a less complex bit-stream based on the predicted network conditions. This technique concept is illustrated in Figure 2 - 27.

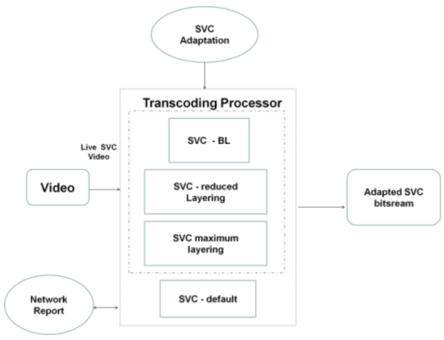


Figure 2 - 27 : Adaptation through complexity reduction

Figure 2 - 28 shows two video streams (low/high bit-rates) where a transcoder can switch to the low bit-rate producing bit-stream to adapt to a current channel condition. The methods of transcoding that can be used for switching among bit-streams are discussed in chapter seven. Also real time simulations that demonstrate the functionality of the developed algorithm *SADMA* are provided in chapter seven.

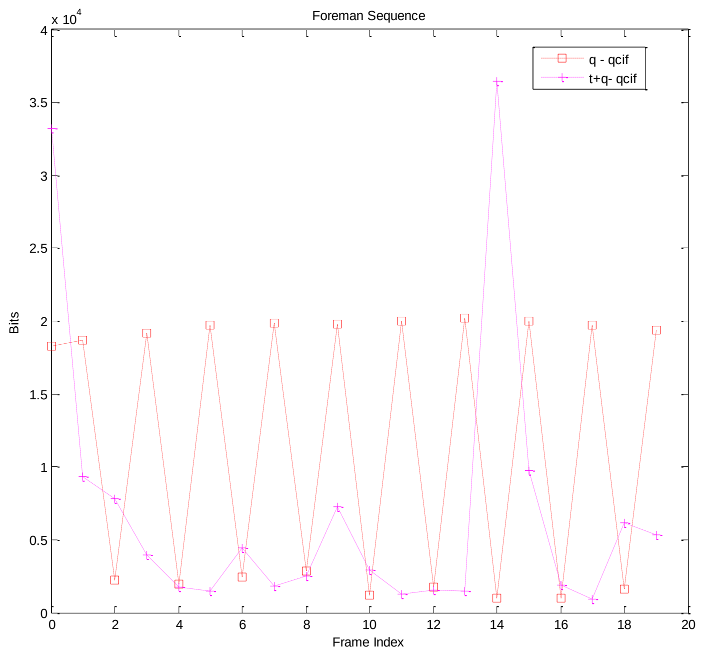


Figure 2 - 28 : Adaptation from High bits video stream (q-qcif) to low bits-stream (*t*+*q*-qcif)

2.8.3. Replacement

This method of adaptation replaces selected items in a video stream with cheaper counterparts, while aiming at maintaining the video quality. For example, a video stream can be replaced with still frames (key frames or visuals representative) and associated narratives to generate a slide show presentation. Hence, the bandwidth requirement can dramatically be reduced. This adaptation scheme can be used to support efficient browsing aids in which still visuals can be used as visual summaries as well as efficient indexes to important points in the original video. However, this adaptation method is efficient and unique to just an application. A better scheme is required to efficiently serve other applications and users' requirements.

2.8.4. Frame and coefficient dropping

Dropping of some regions of frames and/or coefficients is another approach to reduce the bandwidth requirement. This procedure will support adaptation to the desired environment and application. However, dropping of bits from the I- frame coefficients, dropping of B and P frames can result in degrading the picture quality. It might also initiate problems in decoding since some of the frames or coefficients might be needed for motion compensation. Identification of the part of the frames that is less important, for example reducing the header bytes which hold less important information for intended applications, or extraction of the only preferred part of the video sequence for particular user requirements can be employed to achieve certain adaptation. Increasing the data size which will result in decreasing the number of header bytes in the overall video stream can be an approach to useful adaptation. The method of decreasing the header size has been adopted and implemented in *Multi-Bit-Streams-to-Multi-Channels adaptation* method (introduced in this thesis) in chapter five. Experimental results show that, a significant gain can be achieved by employing this approach. Figure 2 - 29 illustrates the concept of this adaptation solution.

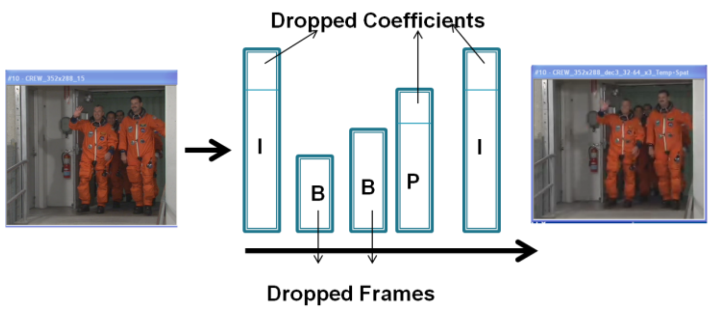


Figure 2 - 29 : Adaptation Solution by frame or coefficients dropping

2.9. Metrics and Techniques for SVC Bit-stream Evaluation

In order to validate and evaluate the various conducted experiments and achieved results, it is necessary to quantify the video quality for a given developed technique or algorithm. The techniques and algorithms produced streams with variable temporal rates, bit-rates and quantisation levels. With SNR scalability, several quality levels are achieved from multiple layers of dissimilar priorities, quality and access or retrieval. These properties in the multi-layer stream result in generating various level of perceived quality depending on the adopted technique or algorithm.

A reliable metric need to be employed to properly assess and evaluate the performance of the techniques and algorithms. In the literature [16], various standard metrics are used to evaluate scalable video performance. Some of these techniques include objective and subjective assessments. Other evaluation approaches evaluating SVC video against non-scalable video as well as a simulcast video stream. Also with improved techniques achieved in this project, a comparison is made with the current JSVM performance and other state of the art scalability techniques available in the literature.

2.9.1. Objective Evaluation Metric

Objective quality assessment metric using Peak Signal to Noise Ratio (PSNR) and mean square error have been used to estimate scalable video quality [17]. PSNR (equation 2-6) is measured on a logarithmic scale and depends on the mean squared error (MSE) of between an original and an impaired video frame or image relative to square(2^n – 1) which is the square of the highest possible signal value in the frame, picture or image where n is the number of bits per image sample [3].

$$PSNR_{dB} = 10log10\ (2^n - 1)^2 \div MSE \qquad 2\text{-}6$$

However, this metric cannot properly reflect human or user perception. Therefore, in order to satisfy the demand of quality and to guarantee an optimal end-to-end QoS of video or multimedia communication systems, subjective metric need to be adopted [16].

Conventional studies mainly focus on the analysis of codec differences [13, 18] and the relationship between frame rate and quality for given bitrates budgets, and the influence of disturbing artefacts and distortions on perceptual quality. Other several quality metrics have also been proposed that directly make use of scalable bit-stream characteristics, such as bit and frame rates instead of relying on properties of the human visual system [19, 20].

However, several of the above mentioned metrics do not consider spatial and temporal variations of the video bit-stream. This is critical when measuring the quality of video bit-stream, because subjective quality highly depends on the amount of spatial and temporal variations in the video bit-stream [16].

A full reference quality metric for scalable video bit-stream has been discussed in [18]. The proposed quality metric in [18] considered frame rate, information regarding PSNR and spatial variation. The subjective assessments were realised using Double Stimulus Continuous Quality Scale (DSCQS) as recommended by ITU-RBT.500 [21].

2.9.2. Subjective Assessment Metric Technique

In order to establish a video quality assessment that will reflect the subjective quality in a reliable way, Double Stimulus Continuous Quality Scale (DSCQS) method with the following ITU-T recommendation BT.500-11 [1] is used.

- A number of participants are involved to view and give a quality score
- A 17 inch monitor is used
- Viewing distance between the each person and video frames is six times the video sequence height.
- Each viewer views the original and reconstructed video sequence twice. This is illustrated in Figure 2 - 30.
- The order of the two sequences original (reference) and reconstructed (impaired) is randomised during the assessment session so that the assessor does not know which is the original and which is the impaired sequence. This method prevents the assessor from pre-judging the impaired sequence compared with the reference sequence [3]. At the end of the assessment, a mean opinion score that indicates relative quality of the reconstructed and reference sequences.

Each assessment is graded on a five-point scale (Bad, Poor, Fair, Good and Excellent) as in Figure 2 - 30. The score is assigned to both the original and reconstructed sequences. The final assessment score SR for the subjective quality of the reconstructed frame R is obtained by subtracting the original frame score O from the reconstructed frame score as shown in (2-7). The lower the value for SR the more similar the reconstructed to the original video.

$$SR = R - O \qquad 2\text{-}7$$

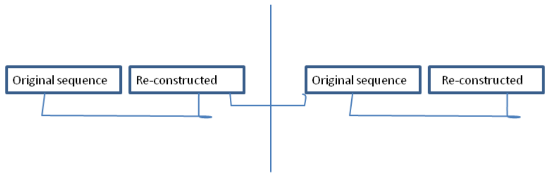


Figure 2 - 30: Double Stimulus Continuous Quality Scale (DSCQS) Subjective Quality Assessment.

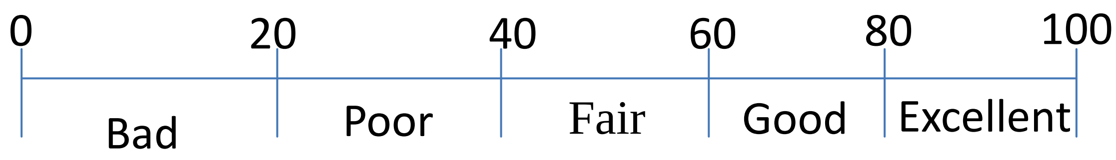


Figure 2 - 31: Subjective Assessment rating Scale

The subjective assessment tests such as DSCQS are accepted to be realistic processes of subjective visual quality. However, this type of assessment test experiences some practical problems. The results can vary significantly depending on the assessor and the video sequence under test. This variation is compensated for by repeating the test with several sequences and several assessors. An expert assessor or evaluator who is regarded to be familiar with the nature of video compression distortions or artefacts may give a biased score and it is preferable to use 'non-expert' assessors [2]. This means that a large pool of assessors is required because a non-expert evaluator will quickly learn to identify characteristic artefacts in the video sequences and so act like an expert viewer.

2.9.3. Comparative Evaluation of SVC and Non-Scalable Stream

SVC is designed to over perform non-scalable video stream and simulcast video coding. A comparative evaluation is made to reveal the profit gain of an SVC stream over single layer or simulcast video stream in terms of scalability and adaptation performance. Objective and subjective quality performance are used as the criteria for the evaluation and comparison. The level of scalability/adaptation and real time performances over heterogeneous networks are also used to conclude scalability flexibility and performance.

2.9.4. Complexity of SVC Stream

Complexity of a video stream indicates how many bits are to be spent for coding the bit-stream. [1] provides a formula for evaluating Scene Complexity Index complexity of a video stream. This has been used in chapter five to evaluate the complexity differences among scalable video streams. The number of spent seconds for encoding an SVC stream also describes the complexity of the stream. The spent time complexity is computed by the amount of processing time required to encode a video sequence using a specific technique or algorithm [22]. This type of complexity differs from one technique to another. This method is used in chapter five and six to evaluate the performances of different coding parameters and techniques.

In [23], a video trace which analyses the frame size and quality or offset distortion characteristics of a given video is described. The earlier methods (frame size and quality) are employed in the analysis and evaluation in chapter-five/six.

2.9.5. Evaluation challenges

Some careful measures need to be adopted in evaluating scalable and non-scalable video streams. These measures or guidelines are followed so that a fair assessment of the video streams is made. The guidelines are highlighted as follows:

- Employing the same sequences provided from one source for any simulation or experimental assessment. Standard sequences are provided by ITU [24, 25] for research and SVC evaluation purposes. The original sequences should be generated from one camera specification at the same instance. This is because camera functions, targets, camera handling vary from one person to another and this can alter the captured image and hence may generate in-accurate results.

- To compare a produced result or sequence from algorithm-1 with algorithm-2, the same frame rates and target rates should be used for both algorithms. Therefore, if algorithm-2 codec is not available to perform the same experiment as done in algorithm-1, a result from the same experimental requirements should be obtained from the established results in the literature.
- A unique coder configuration and design is required for two sets of results to have a fair comparison.

2.9.6. Summary and Conclusion

In this chapter, the concept of scalability and adaptation are discussed. Several existing and previous scalability and network adaptation techniques are discussed. The existing temporal, spatial and quality scalability are described and possible improvements to the techniques are explained. The development and evaluation of the scalability techniques are discussed in detail in chapter six.

The relationship between scalability and rate control are highlighted. From the discussions, it is concluded that rate control is a sub-set of scalability. An effective scalability solution embeds network bit-rate regulation and control. Various techniques and methods for bit-rate regulation and how they are related or can be improved for future scalability techniques are discussed.

Network adaptation as one core application of scalability is described. The frame work and procedures for designing a network adaptation solution which is one main achievement of this work is discussed.

The metrics used in the literature for evaluating scalability streams are discussed. Objective and subjective quality evaluation metrics and complexity analysis are described. These metrics are employed in evaluating the products of the developed scalability techniques, algorithms and coding scenarios.

In conclusion, the existing codecs (H.264/AVC and other standards like MPEG-4) do not include a network aware control for multi-channel network adaptation. Some of the described algorithms (BBRC and FFRC) do not efficiently manage the network resources in other to achieve better bitrates and adaptation. In view of this, an effective scalability and adaptation solution should be implemented that satisfies the described adaptation framework. The solution is also based on the intended application and users' requirements. In this research work, we implement for a diverse network environment (heterogeneous network) where adaptive scalability techniques and schemes to provide an effective and flexible network control at any instance of a predicted network condition are introduced.

References

[1] Ghanbari, M. (2003) "Standard Codecs: Image compression to advanced video coding", Institution of Electrical Engineers, .

[2] Segall, C.A. and Sullivan, G.J. (2007) "Spatial scalability within the H. 264/AVC scalable video coding extension", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1121-1135.

[3] Richardson, I.E.G. "H. 264 and MPEG-4 video compression. 2003", *John Wiley&Sons Ltd., New Jersey,* .

[4] Schwarz, H., Marpe, D. and Wiegand, T. (2007) "Overview of the scalable video coding extension of the H. 264/AVC standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1103-1120.

[5] Team, J.V. (2009) "H. 264/SVC reference software (JSVM 9.18) and manual", *CVS sever at garcon.ient.rwth-aachen.de,* .

[6] Sadka, A.H. (2002) *Flow Control in Compressed Video Communications,* Wiley Online Library.

[7] Wiegand, T., Sullivan, G.J., Bjontegaard, G. and Luthra, A. (2003) "Overview of the H. 264/AVC video coding standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, no. 7, pp. 560-576.

[8] Nack, F. and Lindsay, A.T. (1999) "Everything you wanted to know about MPEG-7. 1", *Multimedia, IEEE,* vol. 6, no. 3, pp. 65-77.

[9] Rijkse, K. (1996) "H. 263: Video coding for low-bit-rate communication", *Communications Magazine, IEEE,* vol. 34, no. 12, pp. 42-45.

[10] Oelbaum, T., Schwarz, H., Wien, M. and Wiegand, T. (2008) "Subjective performance evaluation of the SVC extension of H. 264/AVC", *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*IEEE, , pp. 2772.

[11] Team, J.V. (2003) "Advanced video coding for generic audiovisual services", *ITU-T Rec.H,* vol. 264, pp. 14496-14410.

[12] Choupani, R., Wong, S. and Tolun, M. (2007) *Scalable Video Coding: A Technical Report,* .

[13] Symes, P.D. (2003) *Digital video compression,* McGraw-Hill/TAB Electronics.

[14] Winkler, S. and Dufaux, F. (2003) "Video quality evaluation for mobile applications", *Proc. of SPIE Conference on Visual Communications and Image Processing, Lugano, Switzerland*, pp. 593.

[15] Arachchi, H.K., Perramon, X., Dogan, S. and Kondoz, A. (2009) "Adaptation–aware encryption of scalable H. 264/AVC video for content security", *Signal Processing: Image Communication*, vol. 24, no. 6, pp. 468-483.

[16] Chang, S.F. and Vetro, A. (2005) "Video adaptation: concepts, technologies, and open issues", *Proceedings of the IEEE*, vol. 93, no. 1, pp. 148-158.

[17] Sohn, H., Yoo, H., Kim, C.S., De Neve, W. and Ro, Y.M. (2009) "Perceptual quality measurement for scalable video at low spatial resolution in mobile environments", *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 20.

[18] Jumisko-Pyykkö, S. and Häkkinen, J. (2005) "Evaluation of subjective video quality of mobile devices", *Proceedings of the 13th annual ACM international conference on Multimedia*ACM, , pp. 535.

[19] Feghali, R., Speranza, F., Wang, D. and Vincent, A. (2007) "Video quality metric for bit rate control via joint adjustment of quantization and frame rate", *Broadcasting, IEEE Transactions on,* vol. 53, no. 1, pp. 441-446.

[20] Kim, C.S., Jin, S.H., Seo, D.J. and Ro, Y.M. (2008) "Measuring video quality on full scalability of H. 264/AVC scalable video coding", *IEICE Transactions on Communications,* vol. 91, no. 5, pp. 1269-1278.

[21] Seshadrinathan, K., Soundararajan, R., Bovik, A.C. and Cormack, L.K. (2010) "A subjective study to evaluate video quality assessment algorithms", *SPIE Proceedings Human Vision and Electronic Imaging*Citeseer, .

[22] Maarif, H.A.Q., Gunawan, T.S. and Priantoro, A.U. "Complexity Evaluation in Scalable Video Coding", *Advances in Multimedia-An International Journal (AMIJ) Volume (1),* , no. 1.

[23] Seeling, P. and Reisslein, M. (2011) "Video transport evaluation with H. 264 video traces", .

[24] http://ftp3.itu.chlar-arch/jvt-site/bitstream-exchange/ - Accessed December, 2009.

[25] http://ftp3.itu.ch/av-arch/jvt-site/draft_conformance/ - Accessed December, 2009

CHAPTER THREE

3. OVERVIEW OF H.264/AVC EXTENSION

3.1. Introduction

In this chapter, the design requirements for H.264/AVC Extension are discussed. H.264/AVC scalability techniques are designed for the achievement of specific requirements against prior video coding standards. The Coder structure and processes are discussed. The different types of predictions in H.264/AVC, compression algorithms and methods, de-blocking filtering, slicing syntax, and their main differences with the earlier standards are discussed.

The SVC Network Abstraction Layer and Video Coding Layer of H.264/AVC, the set of three H.264/AVC profiles (Baseline, Main and Extended profiles) with their encoding specifications are described. This chapter presents the features of H.264/AVC extension codec highlighting the elements used with this research work and where differences are made from the earlier and existing standards.

3.2. H.264/AVC Encoder Description

H.264/AVC Extension coder does not explicitly define a codec but rather defines the syntax of an encoded video bit-stream together with the process of decoding this bit-stream [1]. In practice, a compliant encoder and decoder are likely to include the functional components shown in Figure 3 - 1. Most of the basic essential components (motion estimation, transformation, quantisation, compression) of the coder are present in the previous standards (MPEG-1, MPEG-2, MPEG-4, H.261, H.263) but the significant changes in H.264 occur in the details of each functional block. These functional blocks include quantisation where H.264 introduces adaptive quantisation which can be applied within the frames and video layers and the introduction of better and more efficient compression algorithm *CABAC* which is described in section 3.11. are examples where changes occurred in relation to the existing H.264 standard. Some other elements of significant changes are discussed in chapter two. In [2] H.264/AVC traffic characteristics are described as the most recent and efficient codec.

The Encoder in Figure 3 - 1 includes two dataflow paths, a forward path (left to right) and a decoding path (right to left). The dataflow path in the decoder in Figure 3 - 2 is shown from right to left to illustrate the similarities between Encoder and Decoder.

3.3. Encoder Path

A source video frame Fn is processed in units of a macroblock (MB). A MB is encoded in intra or inter-prediction mode. These modes are discussed in section 3.6. A prediction frame P is formed based on the reconstructed picture samples for each block in the MB. With intra mode, P is formed from samples (unfiltered) in the current slice that have previously been encoded, decoded and reconstructed. In inter mode, P is formed by motion-compensated prediction from one or two reference picture(s) selected from the set of list 0 and/or list 1….N reference frames. In Figure 3 - 1, the reference frame is shown as the previous encoded picture $F1_{n-1}$ but the prediction reference for each MB partition in inter-mode may be chosen from a selection of past or future frames that have already been decoded, reconstructed and filtered. The Prediction frame P is subtracted from the current block to produce a residual block R_n. The Rn is then transformed (block transform) and quantised to produce a set of quantised transformed coefficients X which are re-ordered and compressed using *CABAC* or *CAVLC* entropy algorithm. The encoded coefficients produced from the entropy process together with the side information required to decode each block within the MB (mode of predictions, quantisation parameter, motion vector information etc) form the final Video Coding Layer (VCL) bit-stream. This bit-stream is encapsulated in the Network Abstraction Layer (NAL) to form the full video bit-stream for transmission into the network or storage in the local picture memory.

The above encoding process is applied to each scalable encoder described in chapter two, five and six. However, each encoder consists of a layered structure. Each of the layers is also encoded through predicting its coefficients from other layers. The overall layered structure is then encoded and predicted as one single bit-stream.

3.4. Encoder Reconstruction Path

The Encoder after encoding macro-blocks (transform and quantise), decodes them to provide a reference for future predictions. This is why the encoder can use long and short term pictures for prediction. The coefficients ‘x’ from Figure 3 - 1 are scaled (de-quantised Q^{-1}) and inverse transformed (T^{-1}) to generate a difference block $R^1_{n.}$ The prediction block is added to R^1_n to create a reconstructed block uF^1_n (unfiltered frame which is a decoded version of the original block). The function of the provided filter is to reduce the effects of blocking distortion and the reconstructed reference picture is created from a series of blocks F^1_n.

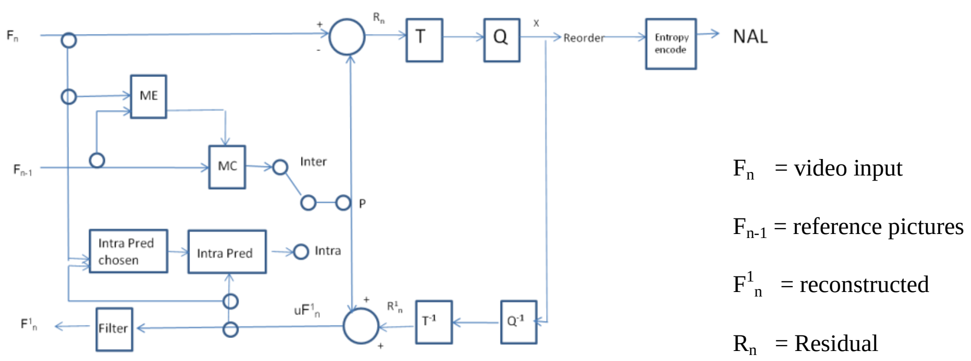


Figure 3 - 1 : H.264/AVC Encoder

3.5. Decoder

The first function of the decoder is to receive a compressed bit-stream encapsulated in the NAL component. The compressed bit-stream is decompressed and decoded to generate a set of quantised coefficients. The processes of de-quantisation and inverse transformation are applied after decompression to generate R^1_n. The header information is also decoded from the bit-stream and a prediction block P is created which is identical to the initial prediction frame P created in the encoder end. P is now added to R^1_n to generate unfiltered samples which are filtered to produce the decoded blocks F^1_n.

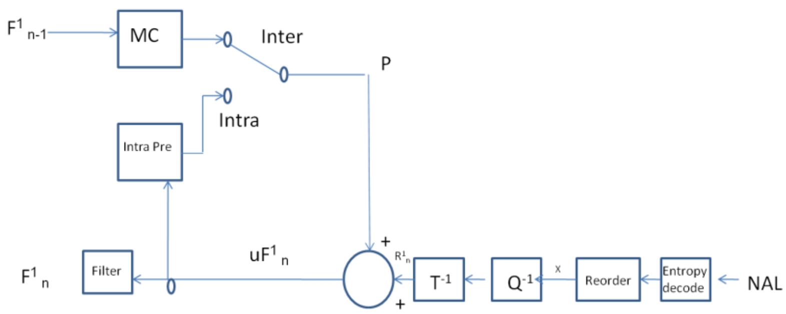


Figure 3 - 2 : H.264/AVC Decoder

3.6. H.264/AVC Prediction methods

The facilitation and simplification of efficient SVC bit-stream production is supported through efficient predictions between the video stream frames and layers. There are two major types of prediction namely inter and intra prediction modes. These modes are discussed in the following sections [3].

3.6.1. Inter-Layer Prediction

Inter-prediction produces a prediction model from one or more previously encoded video frames using block-based motion compensation. The essential differences from earlier codecs include the support for a choice of block-sizes (16x16 to 4x4) and fine sub-sampled motion vectors (quarter-sample resolution in the luma element) [1]. The main objective when designing inter-layer prediction tools is to support the usage of the lower layer information. This is to gain in rate distortion (RD) efficiency of the enhancement layers (ELs). Simulations are conducted (chapter six, section 6.10) to evaluate the QoS performance of this technique. In other previous and existing codecs (H.262, MPEG-2, H.263, MPEG-4 Visual), the only provided inter-layer prediction mode employs the decoded samples of the base layer (BL) signal [3]. In H.264/AVC, the prediction signal is either generated by motion-compensated prediction in the EL, by up-sampling the reconstructed lower layer signal or by averaging the up-sampled signal with a signal from temporal prediction. This method supports better r.d.o. (section 6.10).

The samples of the base layer which actually represent the complete information of the layer are not always suitable samples for the inter-layer prediction. Usually, the inter-layer predictor competes with the temporal predictor especially for sequences with slow motion and high spatial detail. The temporal prediction sample mostly provides a better approximation of the original signal than the up-sampled reconstruction of the lower layer.

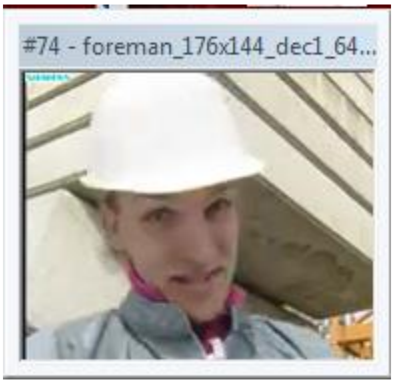
In the case of spatial ELs, if the minor syntax overhead is not considered, the coding efficiency of spatial scalable coding should never become worse than that of simulcast, since in SVC, all inter-layer prediction mechanisms are switchable. An SVC conforming encoder can select among the inter or intra mode prediction base on the generated local sample characteristics. Inter-layer prediction can only proceed in a given access unit using a layer with a spatial identifier D less than the spatial layer identifier of the layer to be predicted. The layer that is used for inter-layer prediction is termed as the reference layer, and it is signalled in the slice header of the EL slices.

3.6.2. Inter-Layer Motion Prediction

SVC includes a new MB type, which is signalled by a syntax element referred to as base mode flag. With this type of MB, it is only the residual signal that is transmitted. There is no additional side information such as intra-prediction modes or motion parameters that is transmitted. When the base mode flag = 1 and the corresponding 8x8 block in the reference layer lies inside an intra-coded MB, the MB is predicted by inter-layer intra-prediction. When the reference layer MB is inter-coded, the EL macroblock and the associated reference indexes and motion vectors are derived from the corresponding data of the co-located 8x8 block in the reference layer by the so called inter-layer motion prediction. Experiments are conducted to evaluate the performance of inter-layer motion prediction in chapter six, section 6.10. Better rate distortion optimisation is gained from employing inter-layer motion prediction.

3.6.3. Inter-Layer Residual Prediction

Inter-layer residual prediction can be employed for all inter-coded MBs regardless whether they are coded using the newly introduced SVC MB type which is signalled by the base mode flag or by using any of the standard MB type. A flag is added to the MB syntax for spatial ELs, which signals the usage of inter-layer residual prediction. When this residual prediction flag = 1, the residual signal of the corresponding 8x8 sub-macro-block in the reference layer is up-sampled using a bilinear filter and used as prediction for the residual signal of the EL MB, so that only the corresponding difference signal needs to be coded in the EL. Figure 3 - 3 illustrates a visual example for inter-layer-residual prediction. The filtered residual shows less quality although it is not blurred while the non-residual prediction is favoured with better quality. The residual prediction is up-sampled and filtered, processes that impact on the picture quality.



a. residual prediction, 75.45kbit/s, 35.76dB

b. non-residual prediction, 63.36kbit/s, 36.51dB

Figure 3 - 3 : Visual illustration for Interlayer-residual prediction at 64kbit/s target rate

3.7. H.264/AVC Extension Design Requirements

It is obvious that particular requirements for scalable video coding may vary widely, according to the application in consideration. The demanded requirements for a family of one or more applications may not be requirements for other applications. It is highly desirable that specific scalable coders meeting the requirements for a family of applications be achievable from a common procedural framework. In [4] the Joint Video Team (JVT) of ISO defined the requirements meant to be achieved in the H.264/AVC Extension standard.

3.8. H.264/AVC Video Coding and Network Abstraction Layers

3.8.1. Video Coding Layer (VCL)

The VCL contains the signal processing mechanism such as transform, quantization, entropy coding, intra prediction, motion-compensated inter-layer prediction. A coded picture of a base or enhancement layer consists of one or more slices. The NAL encapsulates each slice generated by the VCL into typical NAL units [5]. Each SVC Layer is formed by NAL units, representing the coded video bits of the layer. The VCL creates coded representations of the source content, whilst the NAL formats this data. This provides header information in a way that enables simple and effective customization of the use of VCL data for a broad variety of systems.

3.8.2. Network Abstraction Layer (NAL)

SVC (H.264/AVC Extension) retained the H.264\AVC network abstraction layer key properties. NAL is particularly described as follows:

- NAL units form the basic structure of an SVC bit stream. They are independently processed entities, and in most cases should be carried as a single transport unit packet by the underlying transport infrastructure.

- The parameter set concept is used to carry most information pertaining to more than one NAL unit. Enhancement layer may share the same picture or sequence parameter sets as other base or enhancement layers, or may refer to different ones.

- An SVC NAL unit consists of a header of one, two or three bytes and the payload byte string. The header indicates the type of the NAL unit, the potential presence of bit errors or syntax violations in the NAL unit payload, and information on the relative importance of the NAL unit for the decoding process. Optionally, when the header is three bytes in size, scalable layer decoding dependency information is also included [4].

3.9. H.264\AVC Slice Syntax

Figure 3 - 4 shows a simplified illustration of the syntax of a coded slice. The slice header defines the slice type and the coded picture that the slice belongs to and may contain instructions related to reference picture management [1, 6]. The Slice data consists of a series of coded macro-blocks and/or an indication of skipped (not coded) macro-blocks. Each Macro-block consists of header elements and coded residual.

The design feature of H.264/AVC that supports the ability to partition the picture into slice group regions is referred to as Flexible Macroblock Ordering (FMO) [1, 6]. Each of the slices then becomes an independently–decodable subset of a slice group. When this feature is effectively used, can significantly improves the robustness to data loss by managing the spatial relationship among the regions that are not coded in each slice. It can also be utilised for different application gains. For instance two slice groups (slice group-1 and slice-group-2) can be used for two separates packets or layers to provide flexible scalability and defence against data loss. Experimental demonstration shows the performance of different slice groups as provided in chapter five, section 5.7 to 5.7.3. This feature allows the implementation of several techniques as discussed in the next paragraph. Real time simulations are conducted in chapter five, section 5.8 to evaluate the performance of this feature over heterogeneous networks.

There are different techniques of mapping MBs to slices. A slice type takes a code between 0 and 6 to indicate the mapping technique (interleave, dispersed, foreground with left over, box-out, raster scan, wipe and explicit types). Also the MB to slice group map contain a slice group identification number for each MB in the picture, identifying which slice group the related MB belongs to. These techniques are described in Table III - I [1]. Experiments and real time simulations are conducted to evaluate the performance of MB to slice mapping techniques (section 5.8). Figure 3 - 5 illustrates FMO.

The reduction of header bytes is considered in the slices in order to reduce the bit-stream size and hence better scalability and adaptation efficiency. Several experiments are conducted (chapter five, section 5.7 to 5.7.3) and a slice argument of 200 is found to be the optimal slice structure yielding better bits reduction. This is due to the reduction of the header bytes size in the bit-stream slices. A redundant slice with a code = 1 can also be added to the bit-stream at the cost of additional bandwidth overhead. This can be adopted for error control in some channels.

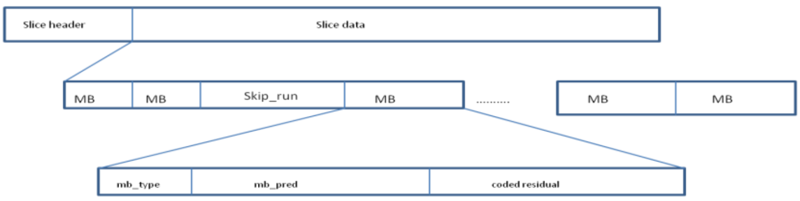
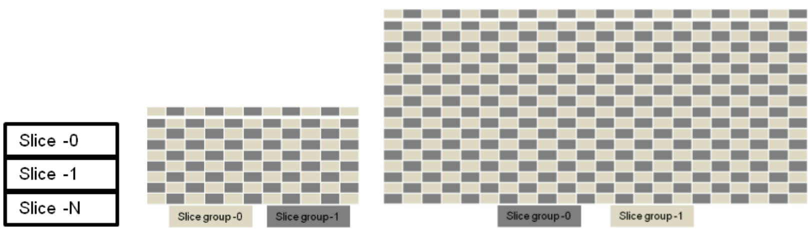


Figure 3 - 4 : H.264/AVC Extension Slice header Syntax



a. b. c.

Figure 3 - 5 : a. FMO not used b. Subdivision of QCIF frames (FMO used) c. Subdivision of CIF frames (FMO used)

Table III - I : Description of MB to slice group map types

Identifier	Name	Notation	Description
0	Interleaved	int	Run length MBs are assigned to each Slice group in turn
1	Dispersed	dis	MBs in each slice group are dispersed throughout the picture
2	Foreground With left over	fore	All slice group are defined as rectangular regions except the last slice group within the picture. The last slice group is defined with all MBs that are not contained in any other slice group.
3	Box out	box	A box is created starting from the centre of the frame containing group 0. All other MBs are in group 1. The size of the box is controlled by pre-configured parameters.
4	Raster Scan	ras	Group 0 contains MBs in raster scan order from the top-left and all Other MBs are in group .1
5	Wipe	wipe	Group 0 contains MBs in vertical scan order from the top-left and all other MBs are in group 1
6	Explicit	exp	A parameter, slice_group_id, is assigned for each MB to indicate its slice group. The macro-block map is entirely defined by the user.

In comparison to prior standard, a rigid slice structure is adopted in MPEG-2 and hence coding efficiency is reduced by increasing the quantity of header data and decreasing the prediction effectiveness [6].

3.10. H.264/Extension Deblocking Filter

One basic characteristic of block-based coding is the accidental production of visible block structures. Block edges are primarily reconstructed with a smaller amount of accuracy than the inner pixels and blocking is normally considered to be one of the most visible artefacts with the existing codecs. To improve this method, H.264/AVC devised an adaptive in-loop de-blocking filter, where the power of the filtering is guarded by the values of several essential elements. The H.264/AVC design has built further on the concept of H.263+ deblocking filter in which the H.264/AVC design is brought within the motion-compensated prediction loop, so that this additional enhancement in quality can be adopted in inter-picture prediction to provide the ability to predict other pictures [6].

The principle of the H.264/AVC de-blocking filter principle is illustrated in Figure 3 - 6 using a visualisation of one-dimensional edge. The samples of R_1 and q_0 and that of R1 and q_1 are filtering determined based on the quantisation parameter (Q_p) with dependent thresholds of $\alpha(Q_p)$ and $\beta(Q_p)$. Hence, the filtering of the samples only takes place if each of the following conditions is satisfied:

$$|R_0 - q_0| < \alpha\,(Q_P)$$
$$|R_1 - R_0| < \beta\,(Q_P)$$
$$|q_1 - q_0| < \beta\,(Q_P)$$

Where the β (Q_p) value is considerably smaller than that of α (Q_p). Accordingly, the filtering of R1 or q1 takes place if the corresponding condition below is satisfied:

$$|R_2 - R_0| < \beta\,(Q_P) \quad \text{or} \quad |q_2 - q_0| < \beta\,(Q_P)$$

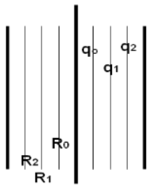


Figure 3 - 6 : Illustration of De-blocking Filter Principle

The main idea of de-blocking filtering is that if a relatively large absolute difference is measured among samples close to a block edge, it is quite expected a blocking artefact and therefore be reduced. However, if the extent of the difference is so large that cannot be deduced by the coarseness of the quantisation used in the encoding, the edge is more likely to reflect the actual characteristics of the source picture and should not be smoothed over. Figure 3 - 7 illustrates the subjective performance of the de-blocking filter. The filtering of all edges improved the subjective quality. In section 0 to 6.8.3, simulations are conducted to evaluate the effect of filtering across different edges and components of the picture. In view of this, a dynamic filtering algorithm is introduced based on the real time performances of several types of filtering across the edges and elements of the video (chapter six and seven).

a. Filter not applied b. Filter applied to all edges

Figure 3 - 7 : Performance of deblocking Filter

3.11. Compatibility and efficiency of *CABAC* and *CAVLC* compression methods on a scalable stream.

The basic mission of a scalable design is twofold: (1) To minimize the coding inefficiency relative to single layer coding and (2) to minimize the decoding complexity especially in single-layer decoding [7]. By single-layer coding, we refer to the type of coding that does not provide scalability. It is therefore important to employ an efficient compression method for a scalable coder especially where the encoder is amended to support multiple scalability streams. In this case, the multi scalable encoder will have a wide range of adaptation including spatial, temporal and SNR scalability. Specific compression methods will be suitable for a particular network situation and targeted applications. This means that by measuring the complexity of each method, scalability and bit-rate regulation can be improved for better adaptation and QoS. Two methods of compression exist in H.264/AVC Extension *CABAC* (Context Based Adaptive Binary Arithmetic Coding) and *CAVLC* (Context based Adaptive Variable length Coding).

A number of experiments and real time simulations are conducted in section 5.4 to evaluate and determine the scalability performances of the two compression algorithms *CABAC* and *CAVLC*. *CABAC* revealed better efficiency and scalability adaptation.

3.11.1. Context Based Adaptive Binary Arithmetic Coding (*CABAC*)

CABAC achieves better compression performance than *CAVLC*. This is achieved through (1) selecting probability models for each syntax element according to the element's context (2) adapting probability estimates based on local statistics and (3) using arithmetic coding rather than variable-length coding [1]. The arithmetic coding core processor and its connected probability estimation are defined as multiplication-free low complexity methods adopting only shifts and table-look-ups. The highest gains are typically achieved when coding interlaced TV signals [6].

3.11.2. Context based Adaptive Variable length Coding (*CAVLC*)

This method used to encode residual, zig-zag ordered 4x4 and 2x2 blocks of transform coefficients. The following summarised the processes involved in *CAVLC* [1]:

- Blocks are typically sparse containing mostly zeros after prediction, transformation and quantisation. *CAVLC* uses run-level coding to represent strings of zeros compactly.

- The highest non-zero coefficients after the zig-zag scan are often sequences of ±1 and *CAVLC* signals the number of higher-frequency ±1 coefficients (training ones) in a compact way.

- The number of non-zero coefficients in neighbouring blocks is correlated. The number of coefficients is encoded using a look-up table and the choice of look-up table depends on the number of nonzero coefficients in neighbouring blocks.

- The magnitude (level) of non-zero coefficients tends to be larger at the start of the reordered array (near the DC coefficient) and smaller towards the higher frequencies. *CAVLC* takes advantage of this by adapting the choice of VLC look-up table for the level parameter depending on the recently-coded level magnitudes.

Conclusively, *CABAC* provides improved coding efficiency over *CAVLC* as demonstrated through the several objective and subjective evaluations and real time simulations in chapter five. This efficiency has facilitated scalable stream adaptation and better QoS especially for a large swamp of network and bandwidth. The coding process of *CABAC* in H.264/AVC Extension is designed to also facilitate low-complexity implementations of arithmetic encoding and decoding [6]. Both *CABAC* and *CAVLC* are designed to be context adaptive algorithms. In low bitrates communication *CAVLC* revealed to produce low bit-rates as compared to *CABAC* from experimental investigations in section 5.4.

3.12. H.264/AVC Transform and Quantisation

The purpose of transform in the coding stages is to convert image or motion-compensated residual data into another domain (transform domain). The choice of transform domain depends on a number of criteria (1) data in the transform domain should be de-correlated (split into components with minimum inter-dependency) and compacted (most of the energy in the transform domain data should be converted into tiny number values) (2) the transform process should be reversible (3) the transform should be computationally tractable (low memory requirement, possible using limited-precision arithmetic and small number of arithmetic computations). The most popular transforms fall into either block-based or image based transform. In H.264/AVC and most existing codecs, a block based transform is implemented.

In H.264/AVC, smaller size transform are used for the following reasons (1) to improve prediction process for inter and intra. Hence the residual signal will posses less spatial correlation which means the transform has less to offer with regard correlation. This also indicated that a 4x4 transform is essentially as efficient in removing statistical correlation as a larger transform. (2) The smaller transform contain visual benefits resulting in less noise around edges known as ringing artifacts. (3) Less computations and word length are required. The H.264/AVC standard involves only adds and shifts specification such that encoder and decoder mismatch is avoided. This has been a problem with earlier 8x8 DCT standards [6]. In [8] an improved transformed module using high performance hardware is introduced. The transform module considers specific memory demands to support the generation of an optimised solution with respect to the encoder complexity and performance.

The function of a quantiser is to map a signal with a range of values J to a quantised signal with a reduced range of values Z. In the quantisation process, the quantised signal can be represented with lesser bits than the original since the range of possible values is smaller. A scalar quantiser maps one sample of the input signal to one quantised output value and a vector quantiser maps a group of input samples (vector) to a group of quantised values. An example of a scalar quantisation is the process of rounding a fractional number to the nearest integer (mapping is from J to Z). The procedure is lossy and not reversible since it is not achievable to determine the exact value of the original fractional number from the rounded integer. H.264/AVC implements a scalar quantisation procedure. The mechanisms of the forward and inverse quantisers are complicated by the requirements to (1) avoid division and/or floating-point arithmetic and (2) incorporate the post and pre-scaling matrices [1]. The basic forward quantiser operation is given as:

$$Y_{ij} = round\ (Z_{ij}/Q_{step}) \qquad (3\text{-}1)$$

Where Z_{ij} is a coefficient of the transform process discussed above, Q_{step} is the quantisation step size and Y_{ij} is a quantised coefficient. The rounding operation need not round to the nearest integer, for instance biasing the rounding operation towards smaller integers can give perceptual quality improvement. H.264/AVC Extension supports a total of 52 quantisation steps values. The quantisation step doubles for every increment of 6 in Qp. The wide range of allowable quantiser steps (0-51) makes it possible for an encoder to control the trade-off between bit-rate and video quality for maximum efficiency. This characteristic supports several quality productions for a *t+q* scalability technique developed in chapter six.

These Qp values (0-51) are arranged so that an increase in 1 in Qp means an increase of quantisation step size by approximately 12% and an increase of 6 means an increase of quantisation step size by exactly a factor of 2. In comparison with prior standards, the Qp takes values between 0 and 32 (MPEG-4). The wide range of values in H.264/AVC supports adaptable and flexible quantisation level for the transform coefficients.

However, the evaluation of the produced quality for every amended operation in Qp is required to ensure or establish the level of QoS of the algorithm. It is often that the use of smaller Qps can produce better objective quality but not the subjective quality. Hence, intensive evaluation of the end products is required.

3.13. Summary and Conclusion

In this chapter, the standard codec employed for this SVC research work has been explored. The coding tools and procedures used in the standard (H.264/AVC Extension) have been discussed highlighting where significant differences exist from the earlier and existing standard codecs.

In the discussion, descriptions are given where this research work is tailored to referring to specific elements and the work required. The next chapter will describe the built heterogeneous simulator for the purpose of this research work which is followed by chapters describing the developed techniques and algorithms which are simulated over the built simulator to assess the performances of the new techniques and algorithms.

References

[1]] Richardson, I.E.G. "H. 264 and MPEG-4 video compression. 2003", *John Wiley&Sons Ltd., New Jersey,* .

[2] Van der Auwera, G., David, P. and Reisslein, M. (2008) "Traffic characteristics of H. 264/AVC variable bit rate video", *Communications Magazine, IEEE,* vol. 46, no. 11, pp. 164-174.

[3] Schwarz, H., Marpe, D. and Wiegand, T. (2007) "Overview of the scalable video coding extension of the H. 264/AVC standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1103-1120.

[4] Ohm, J.R., Koenen, R. and Chiariglione, L. "SVC requirements specified by MPEG (ISO/IEC JTC1 SC29 WG11)", .

[5] Wenger, S., Wang, Y. and Hannuksela, M.M. (2006) "RTP payload format for H.264/SVC scalable video coding", *Journal of Zhejiang University-SCIENCE A,* vol. 7, no. 5, pp. 657-667.

[6] Wiegand, T., Sullivan, G.J., Bjontegaard, G. and Luthra, A. (2003) "Overview of the H. 264/AVC video coding standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, no. 7, pp. 560-576.

[7] Segall, C.A. and Sullivan, G.J. (2007) "Spatial scalability within the H. 264/AVC scalable video coding extension", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1121-1135.

[8] Husemann, R., Majolo, M., Susin, A., Roesler, V. and Valdeni de Lima, J. (2010) "Highly efficient transforms module solution for a H. 264/SVC encoder", *VLSI (ISVLSI), 2010 IEEE Computer Society Annual Symposium on*IEEE, , pp. 86.

CHAPTER FOUR

4. DESIGN OF NETWORK SIMULATOR, SIMULATION SETUP AND VALIDATION

4.1. Introduction

In this chapter, a heterogeneous network simulator is built. The model is purposely built for real time simulations of the developed and evaluated scalability techniques and algorithms described in chapters five, six and seven. Opnet Version 17.1 is used as the development and simulation tool. The model is a multi-channel simulator consisting of different networks including *WIMAX*, *WirelessLAN* (WILAN) and *LAN* networks. The number of included networks is limited due to limited memory with the simulation and development PC. However, the developed algorithm in chapter seven is made intelligent to be aware of all networks conditions before transmitting a matching bit-stream to the predicted network status.

The built model is validated using validation methods in [1]. The required set-up for the simulations and tools used for the real time simulations are also discussed. The set-up and used tools were experimentally verified. This is to ensure correct, expected and valid results are obtained. It is also provided in this chapter, a description of the developed scripts which are used in processing various methods and programs, coder configurations and managing various levels of video sequences for analysis and evaluation.

Wireless networks, WIMAX, Wireless LAN and LAN networks are discussed from the literature to equip the readers with the knowledge of the network structures necessary to understand the model used for simulating the provision of visual media services over contemporary heterogeneous networks.

4.2. Simulation Platform

Simulation modelling has become a popular method for evaluating a network performance. The first method is by mathematical analysis that is used to characterise a network as a set of equations. The main disadvantage is its over simplistic view of the network and inability to simulate the dynamic nature of the network. Thus, the study of a complex system always requires a discrete event simulation package which can compute the time that would be associated with real events in real world situations [2]. Opnet Version 17.1 is used as the simulation platform. Opnet is an event driven dynamic system level simulator which accurately and efficiently simulate the behaviour of real systems networks. It is the product of Opnet Technologies Inc widely used by industries, researchers and professionals. The tool is available to be used under purchased license from the providers. This tool will be used to simulate WIMAX, WILAN and LAN models.

4.3. Introduction to Wireless Network Model

In order to build, deploy and simulate a validated and functional Network structure, a reference is made from the literature [3]. There are four types of Wireless Network Structures based on 802.11 family namely *Independent Basic Service Set* (IBSS), *Infrastructure Basic Service Set*, *Extended Service Areas* and *Multi-BSS environments* (Virtual APs).

IBSS communicates directly with each other and thus must be within direct communication stage. The smallest possible 802.11 network is an IBSS with two stations. IBSS networks are set up for a specific purpose and for a short period of time. One common example is to create a short-lived network to support a single meeting in a conference room.

Infrastructure BSS are distinguished by the use of access points. Access points are used for all communications in infrastructure networks, including communication between mobile nodes in the same service area. One mobile station communicates with another within two hops. The originating mobile transfers the frame to the access point and the access point transfers the frame to the destination station.

Extended Service Areas (ESS) provides network coverage to larger areas. The 802.11 technology allows wireless networks of arbitrarily large sizes to be created by linking BSSs into an Extended Service Set. An ESS is created by chaining BSSs together with a backbone network. All the access points in an ESS are given the service set identifier (SSID) which serves as a network name for the users. 802.11 does not specify a particular backbone technology; it requires only that the backbone provides a specified set of services.

The algorithm and techniques for this thesis will be implemented using the extended service set. However, other network technologies will also be used to characterise the models.

4.4. Introduction to WIMAX

WIMAX (Worldwide Interoperability Microwave Access) is in the category of wireless transmission known as broadband wireless access (BWA). WIMAX does provide characteristic possessed by WIFI including high speed of broadband and coverage footprint that is similar to that provided by the cellular network. It is defined by the Institute of Electrical and Electronic Engineers (IEEE) under standard 802.16 that WIMAX is the best solution to the table today for broadband wireless access [4].

Basically, WIMAX has two key components: (1) A tower mounted antenna similar to the cell towers sprouting all over the landscape and (2) some form of receiver. A WIMAX can provide service coverage over an area as large as 2500 square miles. The frequency band allocation for WIMAX is up to 66GHz and the operational bandwidth is up to 30-40Mbps data access as for 2011, 1Gbit/s for fixed station as for the 2011 update. The WIMAX model consists of the following entities: (1) The subscriber station (SS) (2) The access service network which is made up of the base station (BS) and the Access Service Network (ASN) gateway and the connectivity service network (CSN) which consists of the visited network service provider and home network service provider (ASN) [5].

WIMAX operates in a similar procedure as WIFI. A computer equipped with a WIMAX radio card connects to the WIMAX-enabled network via a remote tower and then exchanges data with that remote device at very high data rates. This procedure is seen in the built WIMAX simulator (Figure 4 - 1) where computer or devices can connect to any of the mobile nodes.

4.5. Wireless Model (WIMAX)

The built wireless model uses WIMAX technology. It is simulated to cover European regions. It has seven base stations with five subscriber stations for each. This made thirty five mobile nodes for subscriber stations in the network. It also has an IP backbone link to the base stations. The IP backbone supports communication to other networks, links and channels. Other network models are built and linked with the WIMAX simulator. The Configuration used is provided in Appendix-A1.

4.6. Implemented Features not available in Opnet tool.

Some features associated with diverse network and channel processes are not implemented in the WIMAX model. Hence, these features will not be functional within WIMAX channels and connected networks. The implementation is related to the Opnet tool package which is being improved by the Opnet technologies. However, the design is improved to include the missing functionality to support a data link among the networks. The following features are not implemented in the core opnet frame work design:

Network ranging during handover

Ranging is a process in the new WIMAX standard. Timing offset estimation and synchronisation between a base station and all other users are done through the ranging process or more specifically initial ranging. Handover occurs when users roam between different access networks. To implement this so that WIMAX channels can transfer data to another network, a direct link is created between LAN and WIMAX. This supports video data trafficking from LAN to WIMAX or vice versa through a connected LAN switch (node_24) to a WIMAX subnet (Base_Station_3) and the WLAN_router (wlan_ethernet_router_137_upgrade/node_27). This is shown in Figure 4 - 1. The link connecting WIMAX base station_3 and the wlan_ wlan_ethernet_router_137_upgrade/ node_27 is Ethernet100BaseT. The wlan_ethernet_router_137_upgrade/node_27 is also linked to WIMAX Base Station_7.

Base Station-initiated periodic ranging

This feature is as discussed in the first bullet point. To implement this feature, a LAN switch is directly linked to a subnet in the WIMAX model as illustrated in Figure 4 - 1. This is discussed in the above immediate bullet point.

Frequency Division Duplex (FDD)

FDD is a technique where separate frequency bands are used at the transmitter and receiver side. The sending and receiving signals do not cause any interference due to different frequencies used. However, this does not affect the implementation of *SADMA* algorithm in chapter seven. *SADMA* implementation requires mainly the network condition which can be predicted regardless of the presence or absence of FDD [6].

4.7. Introduction to LAN and Wireless LAN networks

The requirements for LAN differ from those in a wide area or a public network. There exists a short distance between computers which imply that low-cost, high speed, reliable communication is possible. LAN structure is made up of a number of computers and network devices such as printers. All are connected by a shared transmission medium typically a cabling system or via a switch [7]. In LAN structure of the Network simulator in Figure 4 - 1, four computers are used connected via a switch.

In the context of WLAN, user mobility is particularly significant in situations where users carry portable computers or devices that need to communicate to a server or with each other. In Figure 4 - 1 for the simulator, a number of computers are used that communicate via a server.

4.8. Simulation Set up and Methodology

The Simulation set-up consists of hardware and software tools that are configured together to provide the suitable simulation environment. Two Computers are provided one (*Computer A*) for the transmitting of output video for the proposed scalability techniques and the other (*Computer B*) for hosting the built network model. The model is configured with System In The Loop (*SITL*) which receives packets from *computer A* and prepares them for simulation within the model system. The sent video from computer A is also received on *computer B*. The simulation results are collected at the end of the simulation. The simulation is set-up to run ten different seeds, in which it is changed to five different seeds due to low computer memory. This allows the simulation to successfully run without reporting any memory bugs, problems or leaving some of the seeds to run at a lesser time than others.

4.9. *SITL* (System In The Loop)

This component allows the proposed SVC stream to interact with the network simulator hosted by the OPNET software environment. Packets coming from the simulated SVC Video as shown in Figure 4 - 1 and Figure 4 - 2 are captured by a program called WinpCap [8] and then *SITL* broadcasts these packets into the simulation network channels.

SITL is an OPNET standard tool which is activated through license permission. WinpCap is standard open source software for link layer network access within the windows environment. It allows applications (*SITL* application in this case) to capture and transmit network packets by passing the protocol stack. WinpCap has additional useful features including kernel-level packet filtering, network statistics and support for remote packet capture. WinpCap consists of a driver that extends the operating system to provide low-level network access and a library that is used to easily access the low-level network layers. This library also contains the windows version of well-known libpcap Unix API.

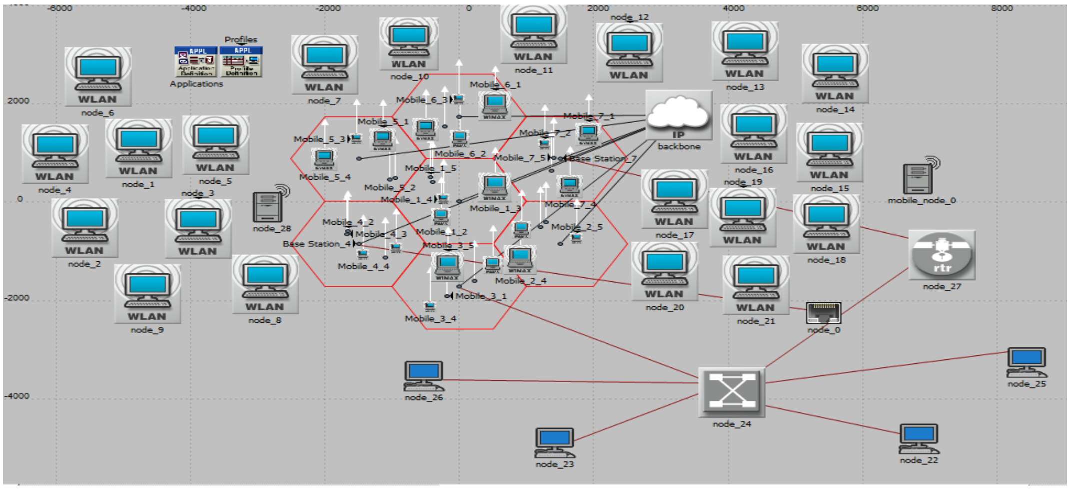


Figure 4 - 1 : Heterogeneous Network Simulator: WIMAX, Wired LAN and Ethernet

4.10. OPNET Version 17.1

This component hosts and supports the building of the Network Simulator. The built heterogeneous network simulator presented in Figure 4 - 1 takes as input the SVC stream from *SITL*. The *SITL* component receives its video packets inputs from Simulation '*Computer A'*.

4.11. VLC (VideoLAN Client)

VLC takes as input the proposed SVC stream and streamed to the Simulated Network via Local Area Network. *VLC* is hosted by both the *Computer A* and *B* as in Figure 4 - 2. It is free and standard multimedia player which plays most multimedia files and has various streaming protocols. [9] was visited for the details of the VLC package documentation and how it is utilised for streaming.

The set-up in Figure 4 - 2 used the various components within a local area network to broadcast a proposed video stream into the channels of the built heterogeneous network simulator which is presented Figure 4 - 1. The simulation set-up is verified to be functional in section 4.12 and the models are validated in section 4.13.

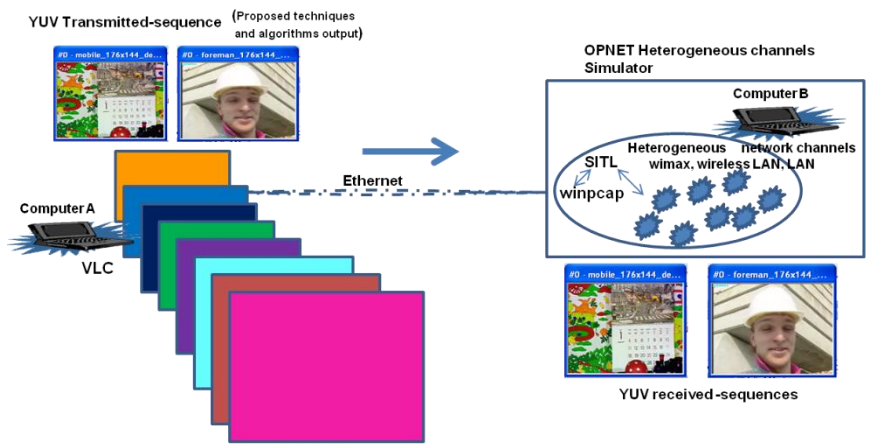


Figure 4 - 2 : Proposed SVC Stream Simulation Set-up Over Heterogeneous Networks

4.12. Simulation Set-up Verification

The simulation set-up is required to be checked and to verify that all functional modules are receiving the data and processing the data that is required. To do this, the communication links, model packets capturing and processing, *Computer A* transmission end and *Computer B* receiving end. This has been verified as follows:

4.12.1. Communication Link

Initially, the established communication between *Computer A* and *Computer B* components is confirmed through a widely used tool for this purpose (wireshark). A wireshark program is used to verify and validate that there is a communication link (Ethernet link in this case) between *Computer A* and *Computer B*. Two IP addresses are used for the *Computer A* and *B*, 192.168.2.1 and 192.168.2.2 respectively. This is reported instantly when the communication is initiated from *Computer A* to *Computer B*. The wireshark screen shows and identifies information regarding the current network including IP addresses of the source and destination of the incoming video data. The wireshark captured interface during the communication between the two computers via the local area network is presented in Figure 4 - 3.

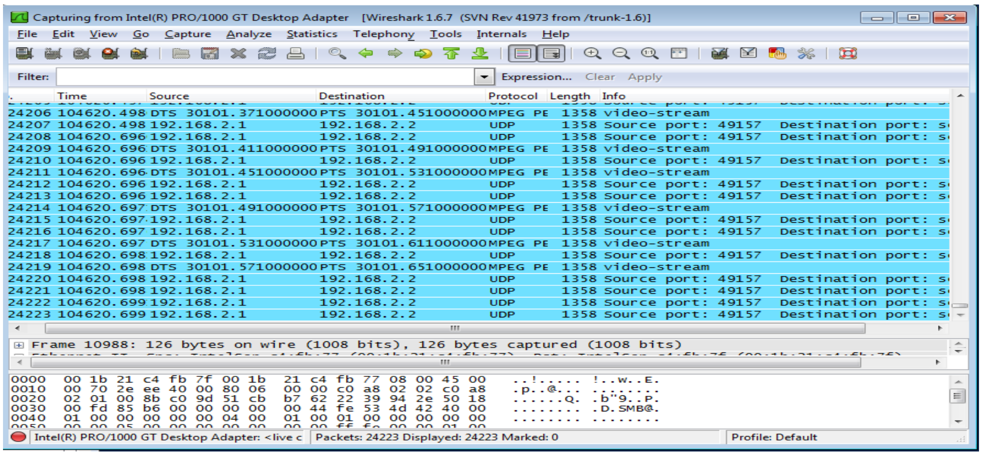
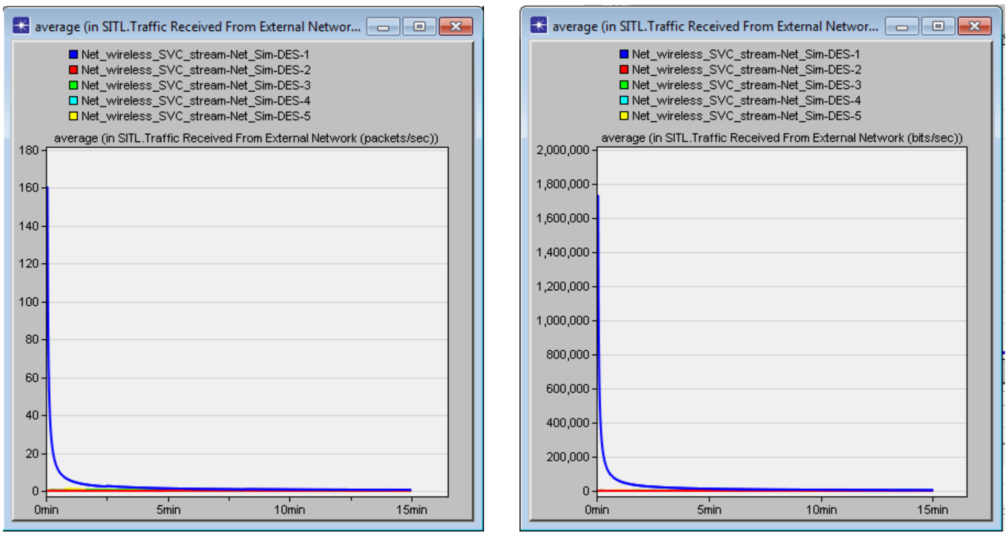


Figure 4 - 3 : Wireshark Interface for bit-stream coming from computer A to computer B

4.12.2. *SITL*: Packets Capturing and Broadcasting

To verify and confirm that *SITL* captures the streamed video from *Computer A* and the packets are broadcast into the network channels, an animation is set-up. The animation is a set of tools which allow the presentation of occurring processes and events within the network simulator to visually appear at the time of the event. The detail of *SITL* set-up documentation is obtained from Opnet Technologies Inc. To further verify the set-up, the *SITL* traffic, WIMAX, WILAN and LAN data is collected at the end of the running test.

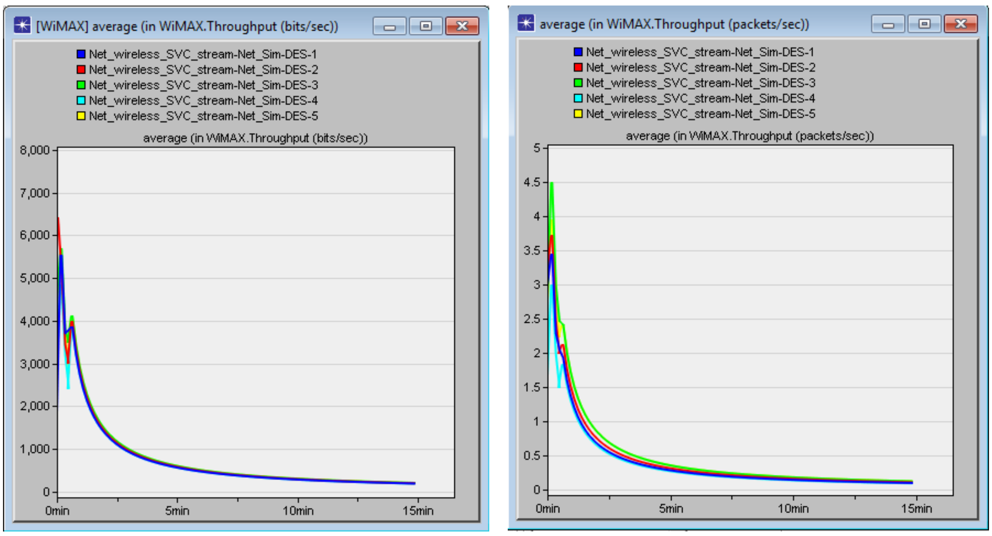
To identify that video packets from the external network (*Computer A*) are being received by the *SITL* component, statistic collection for *SITL* traffic is set-up. The result is collected after the simulation and is captured and shown in Figure 4 - 4. From the result, a maximum of 2 million bits are received per second, which amount to 180 packets. Each packet is a maximum of 1388 bytes.



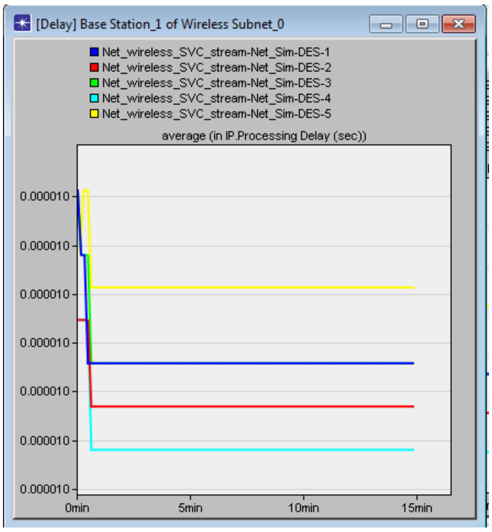
a. packets/sec b. bits/sec

Figure 4 - 4 : Traffic received from computer by the *SITL* component

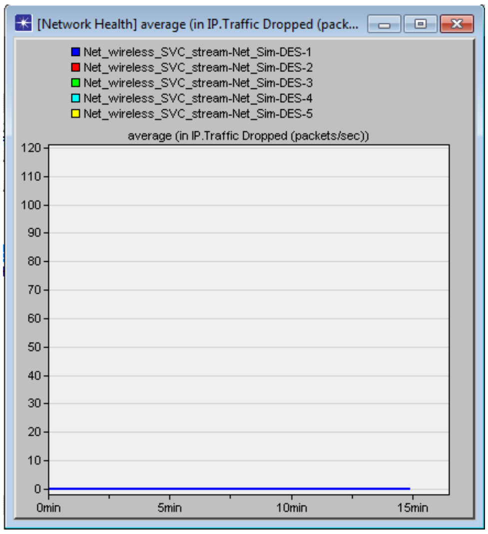
Figure 4 - 5 shows the WIMAX model throughput data in bit/sec and packet/sec. The Figure 4 - 5 also indicated *IP processing delay*, *IP traffic sent/received* and *IP dropped traffic*. The IP dropped traffic is used to verify the health of the network which indicated that the network is one hundred percent healthy since no traffic is dropped as shown in this test. This test has imposed confidence that the video packets are actually broadcast through the network channels successfully.



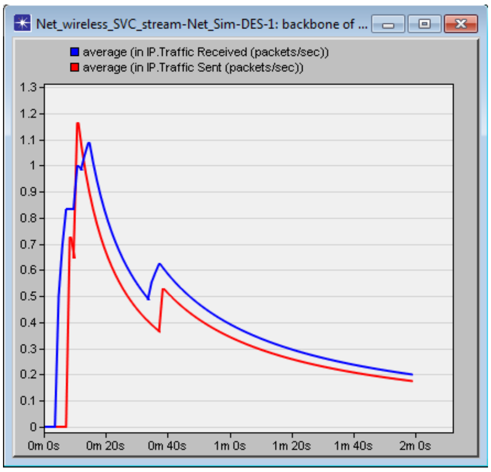
a. b.



c.



d.



e.

Figure 4 - 5 : (a) WIMAX throughput (bits/sec) (b) WIMAX throughput (packets/sec) (c) IP processing delay (sec) (d) IP Traffic dropped (packets/sec) (e) IP traffic received/sent (packet/sec)

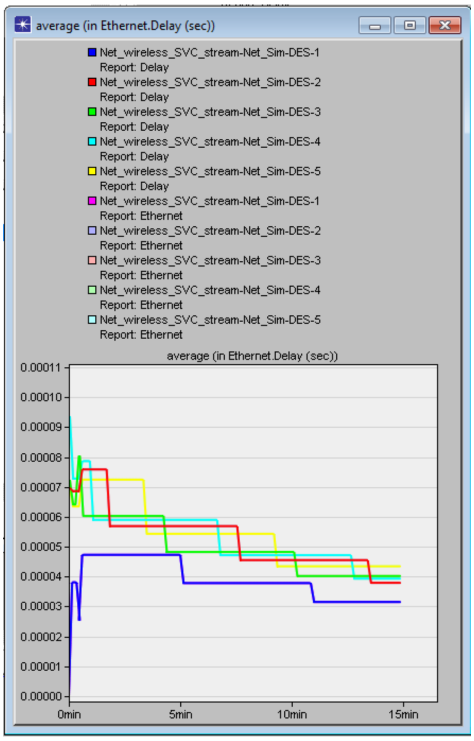
4.13. Model Validation and Verification Method

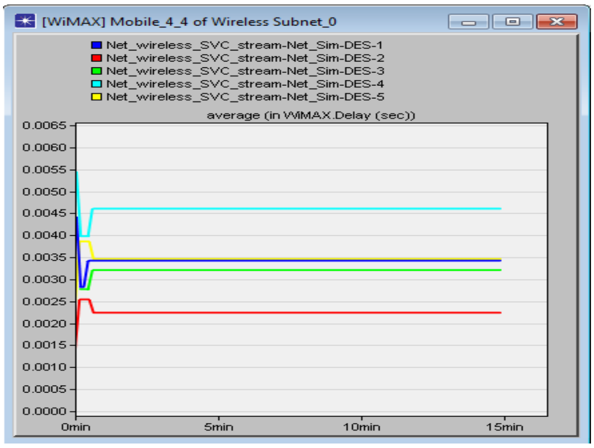
Validation is a key step before using final results to draw conclusions. Validation is generally performed repeatedly during the course of model building and development. Each step of the development is validated before it is enhanced. This will give a high degree of confidence and can easily identify which particular changes are responsible for new un-explained behaviour. Validation is therefore the process of building and maintaining confidence in the equivalency of the developed model to the real system. A validated model therefore has the capability to produce useful or real results as compared to the real system. There are several methods used in model validation provided by Opnet Technologies documentation [1] which are implemented for the validation and discussed in section 4.13.1 to 4.13.4.

4.13.1. Common Sense and Intuition

The developed model described in section 4.3 has been verified and validated. This is to ensure that the simulator that implements the model is debugged and the model produces results close to the one observed in real systems. The methodology used consists of the following techniques (i) Comparing the results of the simulation model with results historically produced by the real system operating under similar conditions, (ii) expert intuition (iii) another simulation model.

To validate this model, Figure 4 - 6(a, b) shows similar results compared with the experimental results provided in [10]. Figure 4 - 6(a) represents average end-to-end delay for all stations of the network while Figure 4 - 6(b) represents average WIMAX delay.





a. Average End to End Delay

b. Average WIMAX delay

Figure 4 - 6 : Heterogeneous Model Simulation results

4.13.2. Measurement

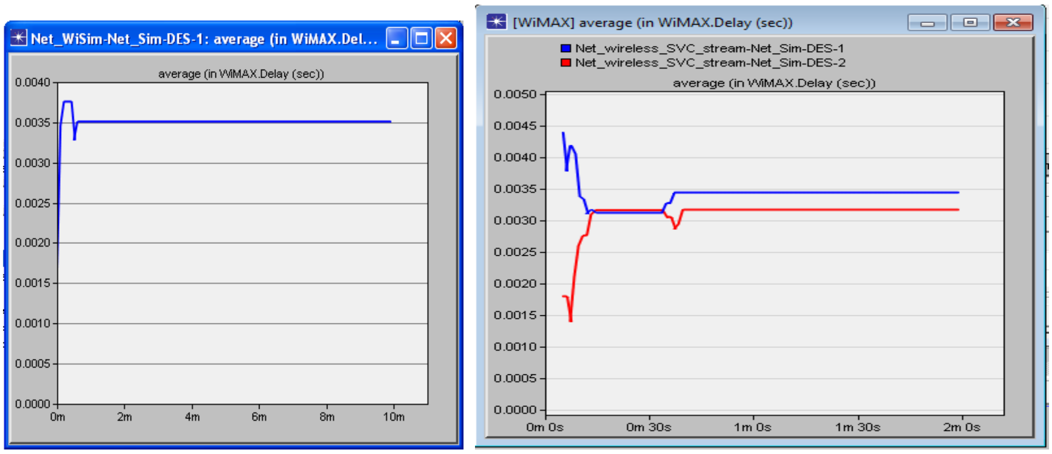
This is another validation approach which most people employ. This can be referred to as baseline against the real system. However, this can only be used if the real system is accessible. Many designers think of measurement as a validation tool.

4.13.3. Alternative Models

Building an alternative model using a different approach can give an insight into how both models behave and which one is the best. The alternative method can be built by you or another designer. The model may have been already provided with results as discussed in 4.13.1.

A similar model of WIMAX is built (Figure 4 - 7(a)) and obtained similar results as in Figure 4 - 6 (b) of the initial model. Near similar values are obtained in the results from seed five of Figure 4 - 6 (b).

Figure 4 - 7(b) is also a result of the initial model which generates near similar parameters with the second built model (Figure 4 - 7(b)). Figure 4 - 7 (b) slightly contains different bitrates parameters to those shown in the Figure 4 - 7(a).



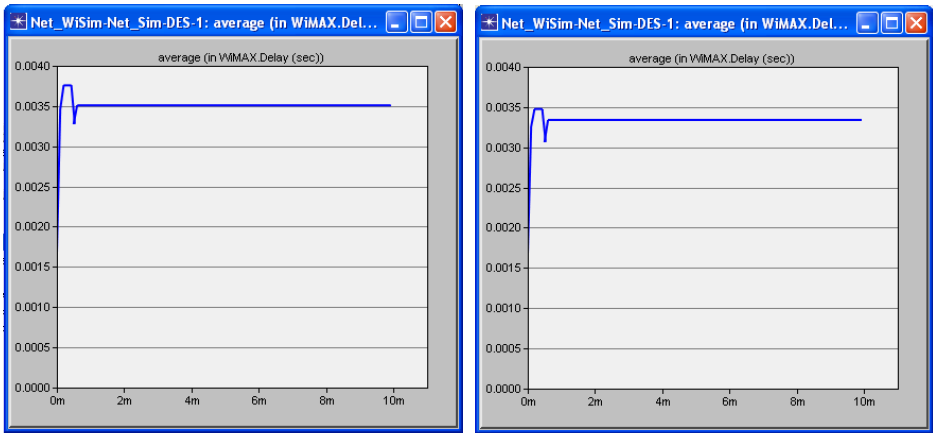
a. b.

Figure 4 - 7 : Alternative WIMAX model (average WIMAX delay)

4.13.4. Incremental Analysis

It is a sound validation technique to make individual changes and therefore gain confidence with the behaviour of individual features. Two simulations are conducted with different model parameters as shown in Table IV - I. The expected result is a datagram which requires more packets before switching and will incur more delay time with additional need of more buffers. This is shown from the results of delay in Figure 4 - 8 (a, b). In Figure 4 - 9, more bits are loaded into the WIMAX network due to a reduced queue with Simulation B. The two simulations have proved to us the expected results which are obtained from the model. For a system with a high number of packets (as in Simulation A) before switching is expected to experience high delay than that with low number of packets (as in Simulation B). The delay differences are shown in Figure 4 - 8 (a, b). Also low packets load is expected where the queue is delayed as in Simulation A. This is shown in Figure 4 - 9.

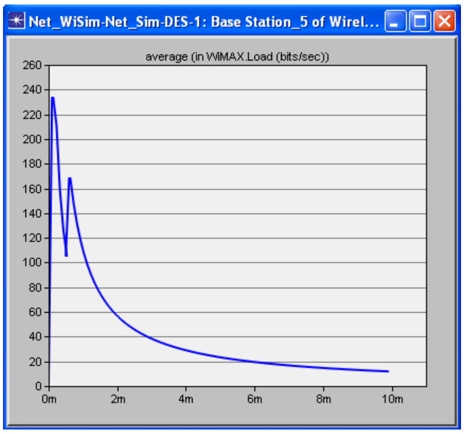
These two experiments (Simulation A and B) laid confidence in the validity of the model with regards to the obtained experimental values and implementations with the model.



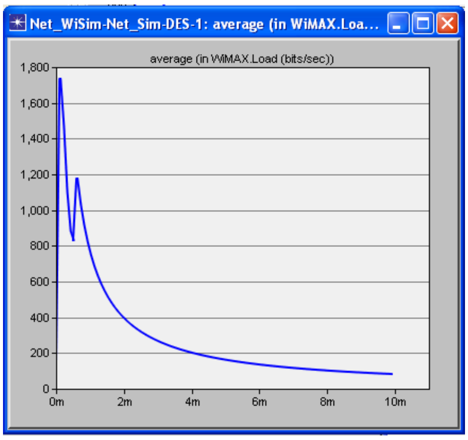
a. Simulation (A)

b. Simulation (B)

Figure 4 - 8 : Simulation 1 (a)



a. Simulation A



b. Simulation B

Figure 4 - 9 : Simulation 1 (b)

Table IV - I : Two Simulations with two different parameters-I

Parameter	**Simulation A Value**	**Simulation B Value**
Datagram Switching rate	500000 packets/sec	250000 packets/sec
Datagram forwarding rate	100000 packets/sec	50000 packets/sec
Memory	16MB	8MB

Another set of simulations is conducted named Simulation A1 and Simulation B1. Table IV - II shows the parameters used for the two simulations over heterogeneous networks. The Ethernet switch node_24 from the Figure 4 - 1 is configured with the two sets of BPDU and packet service rate parameters. The switch with the minimum value for packet service rate is expected to experience a minimum delay within the network. This is shown in Figure 4 - 10. The maximum delay incurred in simulation A1 is ~80/1000000 seconds and that incurred in simulation B1 is ~70/1000000 seconds. These expected responses from the network which are theoretically real have given more confidence in the validity of the model.

Table IV - II: Two simulations with two different parameters-II

Parameter	**Simulation A1 Value**	**Simulation B1 Value**
BPDU service rate (packet/sec)	100000 packets/sec	200000 packets/sec
Packet Service Rate (packet/sec)	5000 packets/sec	10000 packets/sec
IGMP Snooping (Member Time out Duration)	300 secs	400 secs

a. Simulation A1 b. Simulation B1

Figure 4 - 10 : Simulation A1 and B1

4.14. Developed Windows Scripts

A large number of set of scripts are developed. A script is developed for each sequence used for a particular proposed method, technique or algorithm. A script consists of a set of windows commands which are used in processing various functions and programs. The script is also used to process coder design configurations and managing various levels of video sequences for analysis and evaluation. The script allows access and processing of particular program methods which include PSNR-values extraction, and a particular layer extraction. The script allows access to a given coder configuration file to be used in the coding process. The video files are accessed and processed through a pre-set configuration file.

The scripts also access decoder processes and output products. These products are accessed at various levels and stored for analysis and evaluation. One script is created and run automatically to generate a designed encoder output. Similarly, another set of script commands are created to run the decoder proposed functionalities.

To ensure each script is executed correctly, various levels of the processing are paused to allow viewing of the process for verification. A large number of errors are encountered which are successfully debugged. Some of the errors appeared due to coder designs which are provided in a pre-set configuration and are not tolerated by the frame design of the codec.

The concept of developing windows scripts command and executable programs are available in many open source books and can be found in [11, 12]. An example and explanation of a coded and implemented script is provided in Appendix-A2.

4.15. Summary and Conclusion

In this chapter, a heterogeneous network simulator is built for the purpose of conducting simulations to evaluate the real time characteristics of scalability techniques and algorithms. The simulator includes a WIMAX model, LAN and WILAN networks. To validate the built simulator, methodologies adopted from Opnet technologies are used. These methods discussed in the chapter include validation through incremental analysis, alternative models, measurement and common sense and intuition. The measurement technique is not used since the real system is not accessible.

The components, tools and setup involved in the simulations and tests are described and discussed. The complete setup is verified through experimental tests to ensure that various components are receiving valid data all through.

The literature guidance to the network structure and components is also discussed. This is to ensure that the required knowledge of the network structure and the main components of the models are well-known.

The description of the developed windows scripts that are used to experiment and evaluate the developed scalability techniques and algorithm is given. The scripts are used in processing various methods and programs, coder configurations and managing various levels of video sequences for analysis and evaluation.

One limitation encountered is the limited memory capacity of the simulation computer. This does not allow the work to include a higher number of networks. However, the scalability and adaptation algorithm discussed in chapter seven is designed to intelligently predict all involved network conditions.

The next chapters will discuss the simulations work done on the built simulator to evaluate several developed techniques, algorithms and other coding elements.

References

[1] Modeler, O. (2009) "OPNET Technologies Inc", *Bethesda, MD.URL: http://www.opnet.com,* .

[2] Farrington, P., Nembhard, H., Sturrock, D. and Evans, G. (1999) "Network Simulations with Opnet", *Network Technology Research Center School of EEE, Nanyang Technological University Proceedings of the 1999 Winter Simulation Conference,* pp. 308.

[3] Gast, M. (2005) *802.11 wireless networks: the definitive guide,* O'Reilly Media, Incorporated.

[4] Shepard, S. (2006) *WIMAX Crash course,* McGraw-Hill.

[5] Torad, M., El Qassas, A. and Al Henawi, H. (2011) "Comparison between LTE and WiMAX based on system level simulation using OPNET modeler (release 16)", *Radio Science Conference (NRSC), 2011 28th National*IEEE, , pp. 1.

[6] http://www.techopedia.com/definition/27018/frequency-division-duplex-fdd. Accessed 3rd December 2012.

[7] Leon-Garcia, A. and Widjaja, I. "Communication Networks: Fundamental Concepts and Key Architectures, 2000", .

[8] Varenni, G., Degioanni, L., Risso, F. and Bruno, J. (2008) "WinPcap: The Windows Packet Capture Library", *2007-07-23)[2009-11-10].http://www.winpcap.org,*

[9] Fallon, H., de Lattre, A., Bilien, J., Daoud, A., Gautier, M. and Stenac, C. (2002) "VLC user guide", .

[10] Kundu, A. (2010) *Performance Evaluation of Integrated WiMAX and WLAN Networks for Voice over IP Application,* .

[11] http://en.wikipedia.org/wiki/Batch_file. Accessed February 2010.

[12] Shammas, N. (1993) *Windows batch file programming,* Windcrest/McGraw-Hill.

CHAPTER FIVE

5. ANALYSIS OF CODING PARAMETERS FOR THEIR EFFECTS ON SCALABLE VIDEO BIT-STREAM OUTPUT.

5.1. Introduction

One of the objectives of H.264/AVC Extension is to achieve an efficient coding that will minimise bandwidth requirements and therefore generates an optimised and flexible scalable stream [1, 2]. All codecs operate within an environment that places certain constraints on their operation [3]. The rate at which the encoder produces encoded data is the most important constraint in a scalable stream. This is because the scaled stream is required to be adapted to variable network conditions. The video data source usually supplies data at a constant rate and the video encoder processes this high constant rate source to produce a compressed stream of bits at a reduced bit-rate. It therefore implies that robustness of a scalable stream depends on the number of low and high rate layers provided in the bit-stream and a mechanism for their flexibility and adaptation to multi-channel networks [4]. The amount of compressed bitrates and scalable flexibility depends on a number of factors which are described in section 5.2. In this chapter, the influence of coding parameters on scalability adaptation performance are evaluated and analysed from a number of experimental and simulation results. The evaluation and analysis are considered from source encoder to decoder processes and performances.

It remains an important challenge in efforts to improve flexibility in scalability, to employ efficient and effective coding tools and parameters for better and optimised SVC stream. For instance, the recent improvements in H.264/AVC (H.264L) to Higher efficiency Video Coding (HEVC/H.265) rely on some best performance parameters supported, including entropy parameters where *CABAC* or Low Complexity Entropy Coding (LCEC) were made as baseline. *CABAC* and *CAVLC* have been compression algorithms options to adopt in H.264/AVC Extension. *CABAC* and *CAVLC* procedures are described in chapter two. The new Standard HEVC (H.265) whose draft international standard will be released by July 2013 will provide 2x better performance than H.264 standard. This means about half bitrates for a similar quality level at the expense of significantly higher computational complexity compared to the previous standard.

The natural content of a particular video data, its used resolution format, its operational bandwidth and any required spent bit-rates at a time can significantly affects its scalability and quality performance. This is shown in section 5.3 where harbour and mobile sequences are experimented using same encoding and decoding parameters in the same format. In view of this point, selection of coding and operational parameters can dynamically be based on video content, format and network time conditions. Hence, dynamic algorithms from experimental facts are invented for better scalability adaptation and video quality performance.

5.2. Coding tools and parameters

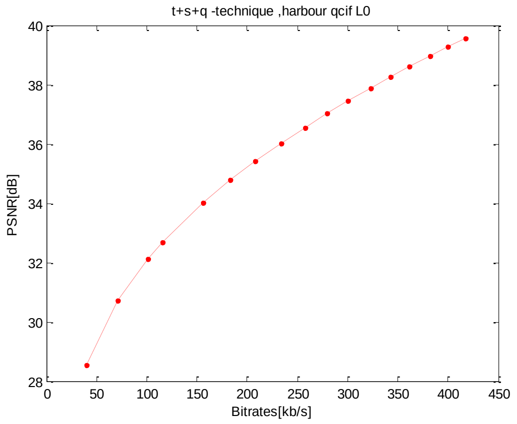
Encoding parameters and decision: These are the coding tools parameters which determine the efficiency of the output and flexibility of a scalable stream. The tools include quantiser step-size, number of intra-pictures, macro-block mode selection and control, motion vector search area [3]. Coding mode control (inter or intra frames), coding sequence structure (*IBBP*, *IPPP* etc), coded units which include slices, *GOP* (Group of Pictures) and Quantisation level (Qp).

Several experiments and tests are conducted with the most influencing coding tools and parameters described above. The experiments meant to find out an optimum operational output for a particular coding tool or parameter where better scalability is achieved. The results are analysed to evaluate the effect of a particular parameter on video perceptual quality. The analysis involves both subjective and objective evaluation and the complexity involved in coding the bit-stream.

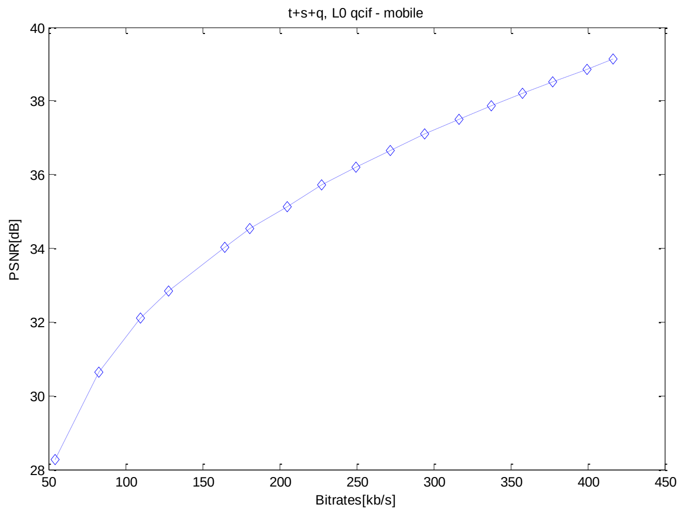
In chapter two, different methods are introduced which can be used for dynamic parameter adaptation. Such methods are integrated in the *SADMA* algorithm proposed in chapter seven.

5.3. Video sequence variable performances

Video sequence with lots of spatial detail and/or rapid movement generally generates more bits than the sequence containing little detail and/or motion. The variation in bits within the picture frames will determine how the whole picture will be scaled within the network. For example, Figure 5 - 1 (a) and (b) of harbour and mobile QCIF sequences show variation in the scale of bitrates where mobile performs better than harbour sequence on scalability adaptation and objective picture quality.



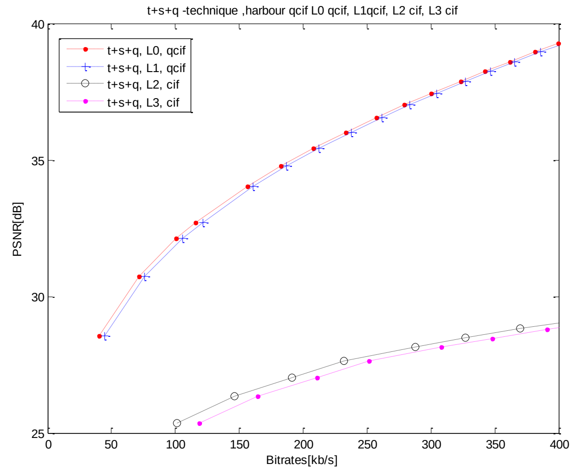
a.



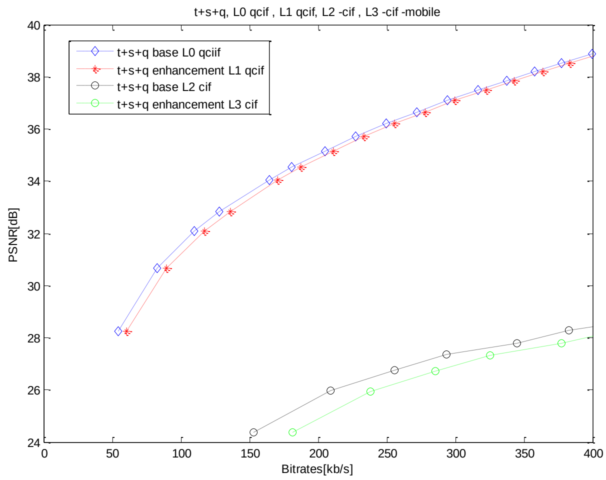
b.

Figure 5 - 1: (a) and (b) represent scalability performance for Layer 0 QCIF on (a) harbour and (b) mobile sequences respectively.

Also in Figure 5 - 2, where high resolution of CIF format is experimented; mobile sequence has better performance than harbour. The two sequences are encoded with the same coding parameters. This meant that, in case of lower bit-rates, the scalability adaptation will perform better on mobile sequence. With these experiments, it can be concluded that video scalability can be influenced due to picture motion as well as the speed of the motion. In order to meet certain criteria of bandwidth requirement, the parameters may need to be adjusted. The quantisation step size parameter can influence both the target bitrates and PSNR required to meet the minimum demanded picture quality. This has been one of the purposes why we evaluated the designed techniques for this research work in chapter six with several experiments and sequences.



a



. b.

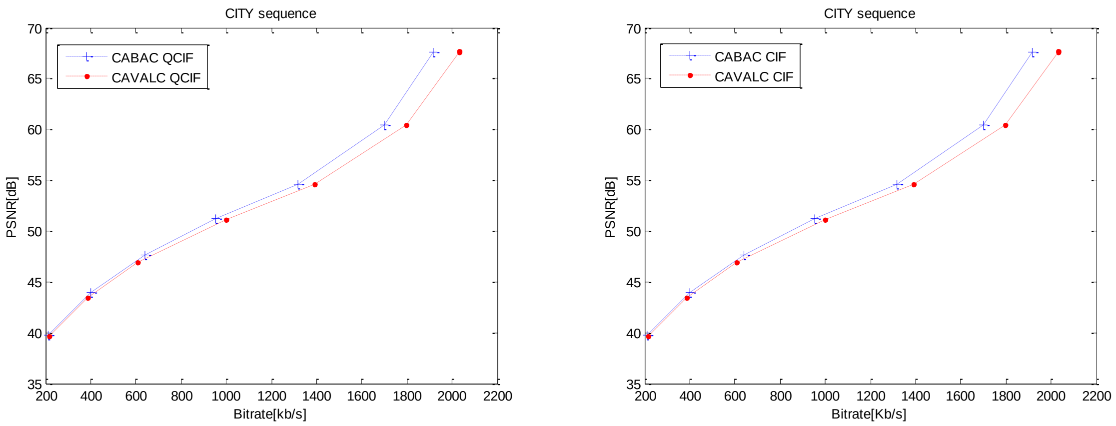
Figure 5 - 2 : (a) and (b) represent scalable performance for *t+s+q* scalability with 2 –Layers of QCIF and CIF on harbour (a) and mobile (b) sequences respectively

5.4. Performance and Evaluation of Compression Algorithms

The entropy coding algorithm used has influence in the adaptation of the scalable bit-stream and the encoding complexity. Two algorithms namely *CABAC* and *CAVLC* are experimented with quite a few video sequences to assess and evaluate their scalability performance. H.264 is configured for experiments to evaluate the efficiency and effectiveness for scalability and complexity of the two compression algorithms for an SVC bit-stream. *CABAC* is set with SymbolMode code 1 and *CAVLC* with code 0. From the results obtained, *CABAC* has best performance with good bit rate saving and at the same time achieving same or better quality than *CAVLC* with about 0.5dB difference. It uses arithmetic coding rather than variable length used in *CAVLC*. *CAVLC* algorithm shows better performance with low bit-rates and is less complex compared to *CABAC* algorithm. BUS, CITY, SOCCER and MOBILE sequences are used in these experiments. These are standard sequences provided from ITU to support SVC research.

5.4.1. Objective Evaluation and Analysis (*CABAC* & *CAVLC*)

It can be concluded that the performance of the two algorithms depends on the sequence as different performances are shown with CITY and CREW in Figure 5 - 3 and Figure 5 - 4. Also, *CABAC* and *CAVLC* tend to give the same performance at low bitrates as in Figure 5 - 3 (a, b) and Figure 5 - 4 (a, b). The choice of these algorithms will depend on application requirements as well as current network conditions.



a. b.

Figure 5 - 3: *t+s+q* bit-stream (a) *CABAC* and *CAVLC*, CITY sequence, QCIF, 75 frames (b) *CABAC* and *CAVLC* CITY sequence CIF, 150 frames.

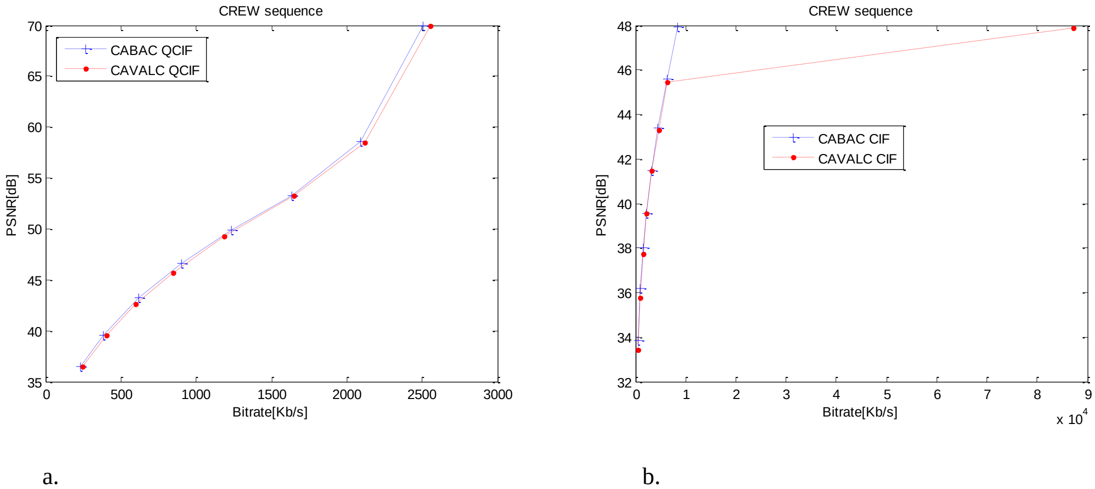


Figure 5 - 4: *t+s+q* (a) *CABAC* and *CAVLC* performance CREW sequence, 75 frames QCIF. (b) *CABAC* and *CAVLC* CREW sequence, 150 frames, CIF

Figure 5 - 5 describes how the bits change from one frame to another during the encoding process. The graph indicates both *CABAC* and *CAVLC* scalability characteristics. *CABAC* shows fewer bits generated. This proved the bit rate saving gain from the use of *CABAC* algorithm. It is more efficient in scalable coding stream as the frames are highly compressed. The experiments were run with the same encoding parameters and target rates.

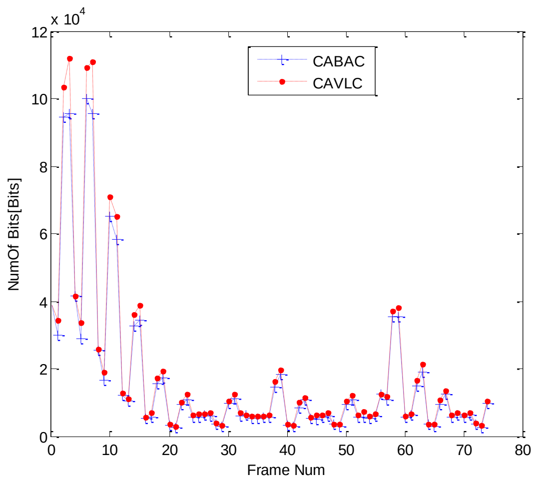


Figure 5 - 5 : Frame Bits Variation in a H.264/AVC Scalable Bit-stream with *CABAC* = 245kbit/s, 34.5dB and *CAVLC*=262kbit/s, 34.5dB – BUS Sequence, 75 frames.

5.4.2. Subjective Evaluation and Analysis (*CABAC* & *CAVLC*)

Experiments are conducted with BUS and CREW sequences to determine the subjective quality performance of *CABAC* and *CAVLC* algorithms in a scalable stream. Both QCIF and CIF resolution formats are used in the experiment.

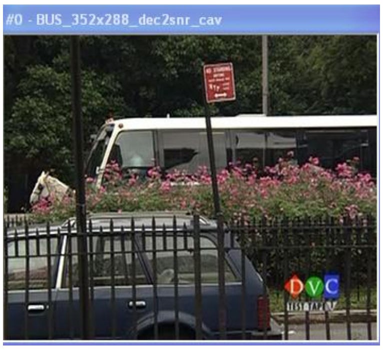
a. *CABAC*, 245kbit/s

b. *CAVLC*, 262kbit/s

Figure 5 - 6 : (a) *CABAC* entropy with BUS QCIF sequence, 75 frames, Achieved 245kbit/s, 34dB. (b) *CAVLC* entropy with BUS QCIF sequence, 75 frames, Achieved 262kbit/s, 34dB.

Figure 5 - 6 (a, b) are reconstructed BUS sequence QCIF videos showing frame 0. The videos are encoded with *CABAC* and *CAVLC* algorithm. A good quality is achieved from both algorithms and there is no distortion observed from any of the outputs.

a. *CABAC*, 1138kbit/s



b. *CAVLC*, 1254kbit/s

Figure 5 - 7: (a) *CABAC* entropy with BUS CIF sequence, 150 frames, 1138kbit/s, 34dB. (b) *CAVLC* entropy with BUS CIF sequence, 150 frames, 1254kbit/s, 34dB.

Figure 5 - 7 (a, b) are the CIF resolution videos, frame 0 for *CABAC* and *CAVLC* algorithms respectively. Although there are slight differences in the objective quality, this has not made any significant impact on the observed subjective picture quality.

a. *CABAC*, 518kbit/s b. *CAVLC*, 575kbit/s

Figure 5 - 8 : (a) *CABAC* technique with crew CIF sequence frame 0, 150 frames, 518kbit/s, 36dB. (b) *CAVLC* technique with crew CIF sequence frame 0, 150 frames, 575kbit/s, 36dB

Figure 5 - 8 (a) and (b) are decoded CIF resolution video sequences. A good and similar quality is observed for both videos. Figure 5 - 9 (a) and (b) represent the QCIF sequences and do not show any variable quality.

a. *CABAC* b. *CAVLC*

Figure 5 - 9 : (a) *CABAC* technique with crew QCIF sequence frame 0. (b) *CAVLC* technique with crew QCIF sequence frame 0.

Based on the objective and subjective results obtained, we can conclude that, *CABAC* is better in scalability adaptation as it is encoded with better and more efficient prediction than *CAVLC*. However, both *CABAC* and *CAVLC* ensured a guaranteed quality picture with *CABAC* characterised with better adaptation.

5.4.3. *CABAC* & *CAVLC* algorithms complexity performance and analysis

Two methods are employed to assess the complexity of *CABAC* and *CAVLC* algorithms. One method is to automatically record the start and end times of the coding process [5]. The other method used to evaluate their bit-streams is Scene Complexity Index (SCI) [3]. The complexity index of a picture is the product of its bitrates and average quantiser step size in the corresponding frame. This is defined in (5-1), section 5.5.3.

From the results shown in Table V - I, *CABAC* experienced more complexity than *CAVLC* in most of the sequences. This is because *CABAC* employed bi-directional prediction coding for efficiency. This contributes a slight amount of complexity against the *CAVLC*. The *CABAC* and *CAVLC* in Table V - I were experimented with different parameters.

Table V - I : *CABAC*/*CAVLC* coding time and SCI

Algorithm	**Encoding Time**	**SCI**	**Sequence**
CABAC	5min, 11.83sec	271	Bus
CAVLC	4min, 15.24 secs	289	
CABAC	10min, 5.9sec	335	Football
CAVLC	10min, 3.63 secs	338	
CABAC	10.79 min	333	Crew
CAVLC	5m, 25.06 secs	335	
CABAC	7min, 35.40 secs	286	Mobile
CAVLC	8min, 4.94 secs	292	
CABAC	3.54 min	168	City
CAVLC	3min, 22.25 secs	175	

The results shown on Table V - I do not represent the time expected for running real world applications. The computer where the experiments are conducted is of the specification provided in Appendix-A3. Also, in the experiments several functions and scripts are executed to allow extraction of test and experimental data. This process of execution (extraction of experimental and test data) costs an amount of time which may not be included in the normal video encoding. However, all the experiments experienced the same conditions and configurations for a fair comparison.

Figure 5 - 10 presented a real time simulation for *CABAC* and *CAVLC* compression algorithms. The simulation is conducted with BUS sequence for 150 frames, encoded using $t+s+q$ technique. *CABAC* algorithm experienced less amounts of delay than *CAVLC*. *CABAC* has shown better compression efficiency than the *CAVLC* algorithm in several conducted experiments in this chapter.

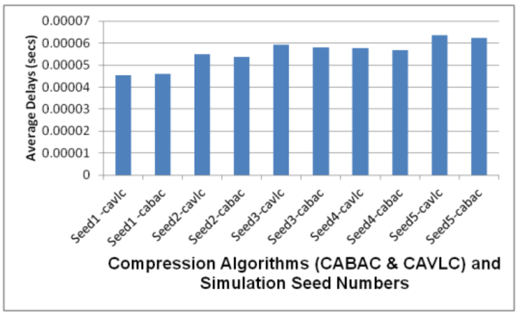


Figure 5 - 10 : Real Time Performance for *CABAC*/ *CAVLC* Algorithms over Heterogeneous Channels

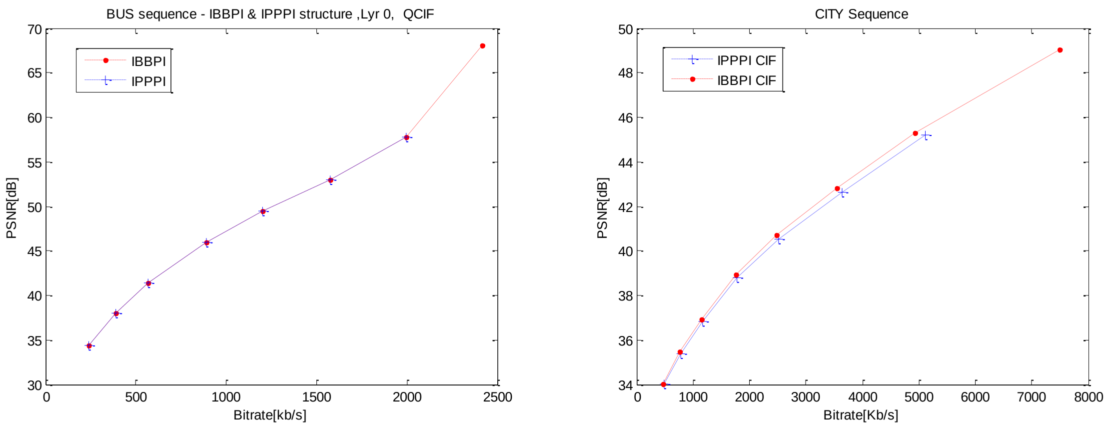
5.5. Performance and Evaluation of Sequence Structure

Video frames are predicted and compensated through a bi-directional prediction mode or one-direction prediction mode. The bi-directional mode which uses past and future stored frames to predict for current frame is more computationally demanding than one-direction which uses only the past or future frame as a reference. Bi-directional frames are referred to as B-frames and the others as P-frames. Experiments are conducted to analyse the influence of these modes on scalable stream layers. Sequence structures *IBBP*B, *IPPP*P are used in the video frames. Also, real time performances of the two streams are examined over heterogeneous networks.

5.5.1. Objective evaluation of sequence structure (*IBBP* and *IPPP*)

Several experiments for *IBBP* and *IPPP* structures compression algorithms are conducted to evaluate their performances in scalable stream adaptation. *IBBP* sequence shows better quality performance with about 0.1dB gain. It also shows better bit-rates saving than *IPPP*. However, this varies from one sequence to another for example in Figure 5 - 11 where low resolution and low bit-rate are used, both structures show the same performance. This result is not the same at high rates from 2Mb/s. At high bit-rates, the *IPPP* performance begins to drop. This is due to adjusted quantisation employed when *IBBP* structure is used as the bits start to reduce. Figure 5 - 11 (b) and Figure 5 - 13 represent the performance in CIF high resolution while Figure 5 - 11 (a) and Figure 5 - 12 (a, b) represent the performance in QCIF resolution format. *IBBP* structure performed better than *IPPP* with a difference of 0.02dB and a better bits reduction than *IPPP* structure.

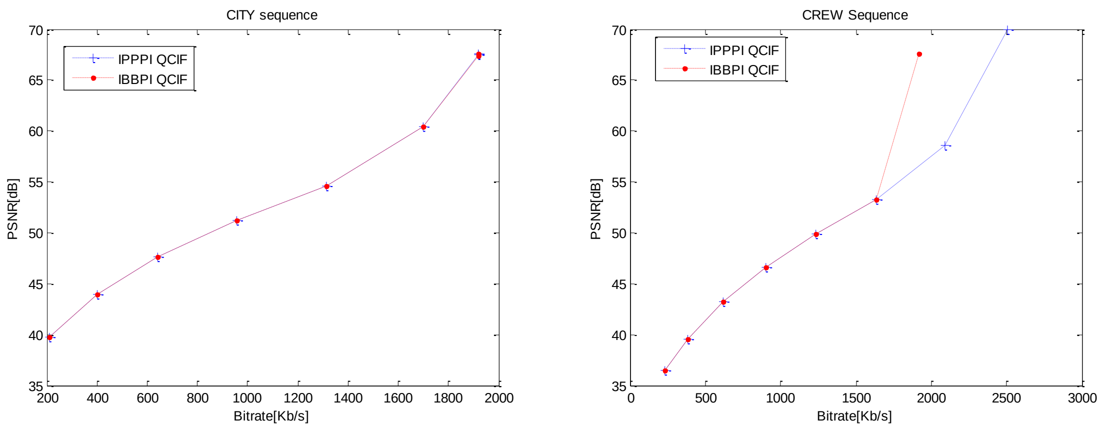
One other reason that allows *IBBP* sequence to generate low bit rates is that most of the frames are highly compressed using bi-directional mode of prediction and therefore fewer bits are spent although there is an additional complexity in the coding compared to the one directional mode of prediction. The quantisation parameter is decreased in bi-directional mode due to the low number of bits in the spatial domain. Table V - II and Table V - III show the complexity of *IBBP* and *IPPP* sequence usage based on coding time. This information gives an idea of computational complexity for different sequence structure used. A scalable stream can then employ a structure based on network conditions and application demands.



a. BUS, QCIF

b. CITY, CIF

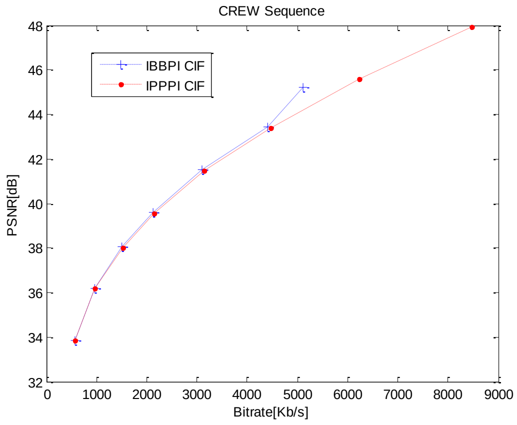
Figure 5 - 11: (a) *IBBP*/*IPPP*- PSNR against bit-rates performances, 75 frames –L0. (b) *IBBP*/*IPPP*- PSNR against bit-rates performances, 150 frames-L2.



a. CITY, QCIF

b. CREW, QCIF

Figure 5 - 12: a. *IBBP*/*IPPP* PSNR against bit-rates performances, with 75 frames-L0. b. *IBBP*/*IPPP* PSNR against bit-rates performances, with 75 frames-L0.



CREW CIF

Figure 5 - 13 : *IBBP*/*IPPP*- PSNR against bit-rates performance, with 150 frames-L3

5.5.2. Subjective Analysis and Evaluation of Sequence structure

Figure 5 - 14 and Figure 5 - 16 (a, b) are the subjective performance results for *IPPP* and *IBBP* sequence structures from BUS and SOCCER video sequences. This represents a lower resolution QCIF format. Although there is difference in bit-rates obtained, un-distorted picture is recovered from each of the structures. Figure 5 - 15 and Figure 5 - 17 show the subjective performance for the CIF resolution format from the same sequence. There is no distortion or degradation observed from both the videos.

a. *IPPP*, 255kbit/s

b. *IBBP*, 241.241.9kbit/s

Figure 5 - 14 (a) *IPPP* structure, QCIF, 75 frames, 255kbit/s, 34.52dB (b) *IBBP* structure, QCIF, 75 frames, 241.91kbit/s,34.65dB

a. *IPPP*, CIF, 1213kbit/s

b. *IBBP*, 1119kbit/s

Figure 5 - 15 (a): *IPPP* structure, 150 frames, 1213kbit/s, 33.98dB (b) *IBBP* structure CIF BUS sequence, 150 frames, 1119kbit/s, 34.10dB

a. *IBBP*

b. *IPPP*

Figure 5 - 16 (a): *IBBP* structure QCIF SOCCER sequence, 75 frames (b): *IPPP* structure QCIF SOCCER sequence, 75 frames,

a. *IBBP*

b. *IPPP*

Figure 5 - 17 : (a) *IBBP* structure CIF SOCCER sequence, 150 frames (b) *IPPP* structure CIF SOCCER sequence, 150 frames

5.5.3. Sequence structure complexity performance and analysis

To evaluate a scalable bit-stream, it is important that its complexity is known. Computational complexity is a trade-off between coded bits reduction and coding delay. Additional computations in bi-directional sequence will derive lesser bandwidth than a one-directional sequence prediction.

Two methods are employed to derive the complexity of *IBBP* and *IPPP* sequences. One method is to automatically record the start and end times of the encoder process. The other method uses a Scene Complexity Index (SCI) [3]. The complexity index of a picture is the product of its bitrates and average quantiser step size in the corresponding frame. This is expressed in (equation 5-1).

$$SCI = 1/GOP\ [IQ_I + \sum_{j=1}^{m} Pj\ Qpj + \sum_{j=1}^{m} BjQBj] \qquad 5\text{-}1$$

Where I, P and B are the target bit-rates for the I, P, B frames, and Q_I, Q_P, Q_B are their average quantiser step sizes respectively. Applications and channel conditions will allow a suitable structure based on calculated complexities.

Table V - II : *t+s+q* Stream (*IBBP*/*IPPP*) - coding time and scene complexity index

Structure	**Encoding Time**	**SCI**	**Coding Parameters**	**Sequence**
IBBP	3.57 sec	261.5	QCIF = 133.07kbit/s, 37.26dB, 150 Frames – *IBBP*	Foreman
IPPP	3.82 sec	321.5	CIF=254.40kbit/s, 36.80dB, 300 frames – *IBBP*	Foreman
IBBP	3.80 sec			Foreman
IPPP	3.87 sec		QCIF=139.25kbit/s, 37.19dB,150 frames – *IPPP*	Foreman
IBBP	3.35 sec		CIF=276.01kbit/s, 36.76dB, 3000 frames – *IPPP*	Foreman
IPPP	3.83 sec			Foreman
IBBP	10.27 min	275	QCIF= 326.28kbit/s, 35.21dB, 75 frames – *IBBP*	Football
IPPP	8 .40 min	275	CIF=439.01kbit/s, 32.94dB,150 frames – *IBBP*	Football
IBBP	10.23 min			Football
IPPP	8.12 min		QCIF=334kbit/s, 35.30dB, 75 frames – *IPPP* CIF=453kbit/s, 32.89dB, 150 frames	Football
IBBP	3.47min	203	QCIF=234.51kbit/s, 34.15dB, 150frames - *IBBP*	Harbour
IPPP	4.44min	205.3	CIF=655.41kbit/s, 33dB, 300 frames –*IBBP*	Harbour
IBBP	3.99min			Harbour
IPPP	4.13min		QCIF=258.01kbit/s, 33.97dB, 150 frames –*IPPP* CIF=149.91kbit/s, 33.17dB, 300 frames – -*IPPP*	Harbour
IBBP	4.4 min	201.5	QCIF= 184.18kbit/s, 36.79dB, 75 frames – *IBBP*	Soccer
IPPP	4.28min	138.9	CIF= 670.41kbit/s, 36.38dB, 150 frames – *IBBP*	Soccer
IBBP	4.06min			Soccer
IPPP	4.31min		QCIF= 182.58kbit/s, 36.53dB, 75 frames – *IPPP*	Soccer
IBBP	4.06min		CIF= 773.50kb/s, 36.18dB, 150 frames – *IBBP*	Soccer
IPPP	4.28min			Soccer

Table V - II presents results obtained for the encoding time of *IBBP* and *IPPP* structures. *IPPP* shows less encoding time than *IBBP*. This is due to IBBP bi-directional coding which result in bits reduction and hence better efficient prediction and compression.

It can therefore be concluded from Table V - II results, the employment of sequence structure should be an application and network status requirement during the video transmission. The best structure is chosen depending on the available resources and need.

In Table V - III, decoding time for *IBBP* is higher than that of the *IPPP* structure. This is because the decoder processes a larger number of compensation vectors to reconstruct the sequence. *IBBP* structure will be desirable where bandwidth is limited. This is because it supports low bit-rates, providing better r.d.o. than *IPPP*. Internet applications and similar ones where channels are not guaranteed, the *IBBP* structure will be appropriate. This is further proved from the real time simulations presented in Figure 5 - 18.

Table V - III : Decoding time for *IBBP*/*IPPP* structure (*t+s+q* technique)

Structure	**Layer**	**Decoding time(secs)**	**Sequence**
IBBP	0	0.86	Harbour
IPPP	0	0.78	Harbour
IBBP	1	5.28	Harbour
IPPP	1	5.02	Harbour
IBBP	2	5.34	Harbour
IPPP	2	5.09	Harbour
IBBP	3	5.29	Harbour
IPPP	3	5.21	Harbour
IBBP	0	0.77	Soccer
IPPP	0	0.75	Soccer
IBBP	1	4.56	Soccer
IPPP	1	4.53	Soccer
IBBP	2	4.61	Soccer
IPPP	2	4.53	Soccer
IBBP	3	4.58	Soccer
IPPP	3	4.52	Soccer
IBBP	0	0.74	Foreman
IPPP	0	0.74	Foreman
IBBP	1	4.7	Foreman
IPPP	1	4.5	Foreman
IBBP	2	4.61	Foreman
IPPP	2	4.55	Foreman
IBBP	3	4.63	Foreman
IPPP	3	4.55	Foreman

5.5.4. Real time performance evaluation for sequence structure

To evaluate the performance of *IBBP* and *IPPP* coding structure in real time scenarios, simulations are conducted using the set-up described in chapter four. The main set-up distinguisher here is that, two sequences are encoded to use *IBBP* and *IPPP* respectively. Their bit-streams are then streamed into the built heterogeneous network simulator. Delay statistics are collected and averages are computed. In all of the five seeds used for the simulations as in Figure 5 - 18, *IBBP* structure yields lesser delay times than *IPPP* structure. This shows that, the complexity and bit-rates of *IBBP* are reduced over the network although it is more complex to encode. The produced encoder video and network layer packets are more efficiently compressed and hence better adaptation is supported.

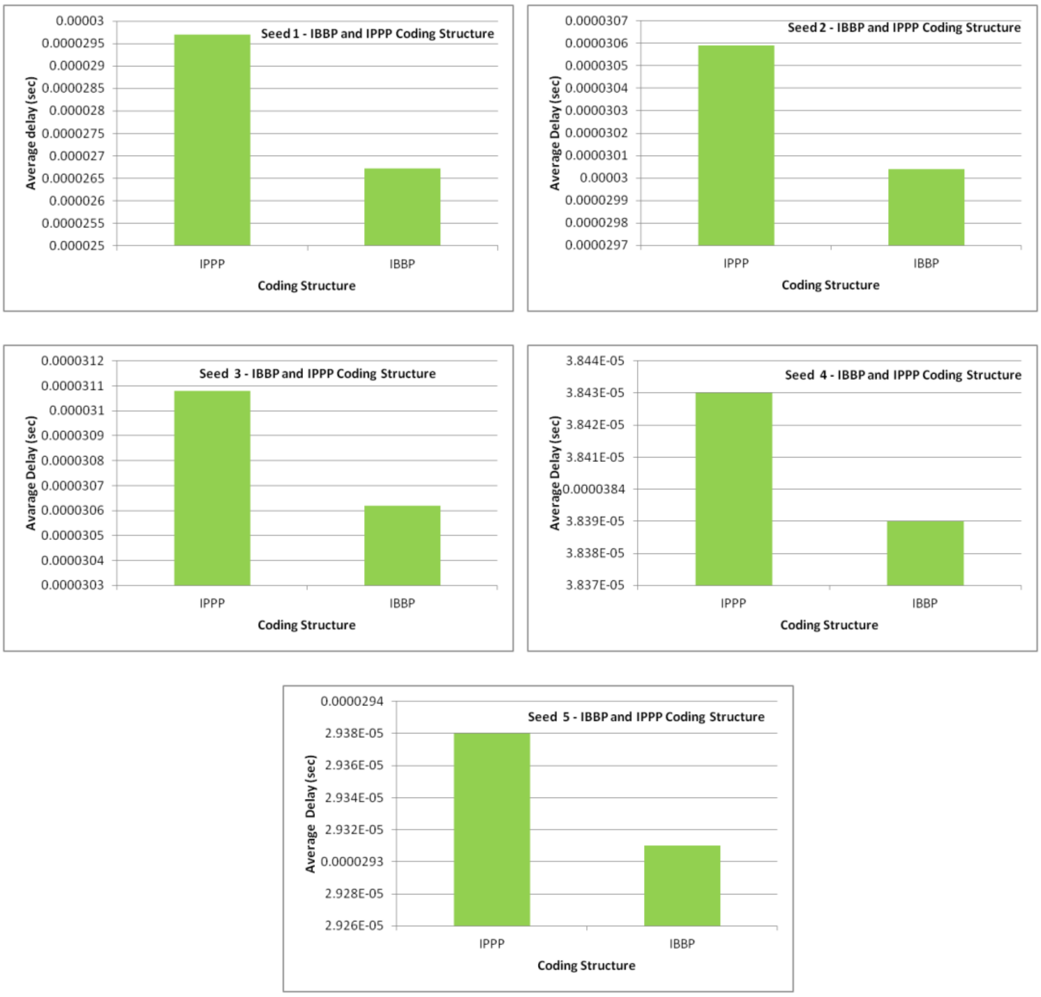


Figure 5 - 18 : Real Time Performance of *IBBP* and *IPPP* Coding Structures over Heterogeneous Networks

5.6. Video Coded units

The partition of SVC bit-stream into units is an important syntax in a scalable stream. The partitions determined the number of coded independent pictures representing the entire video sequence and therefore influence the scalable outputs. The coded units include *GOP* (Group of Pictures) which constitutes groups of frames or pictures whereby each is an entity that generates one instance of the video with specific video performance or characteristics. Slices are sections of coded video and this is described in chapter two.

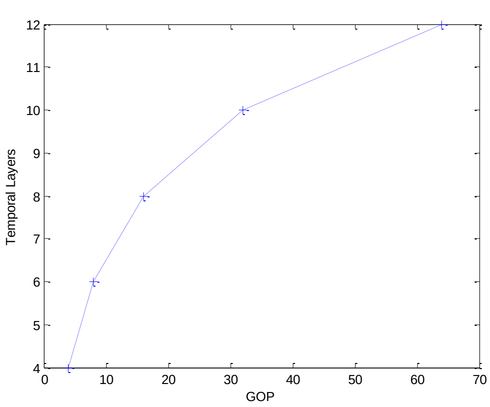
Experiments and simulations are conducted to examine the output variations resulting from variable *GOP* usage on the stream. Sections 5.6.1 to 5.6.8 described these experiments and the results obtained.

5.6.1. Performance and Evaluation of Group Of Blocks (*GOP*) units

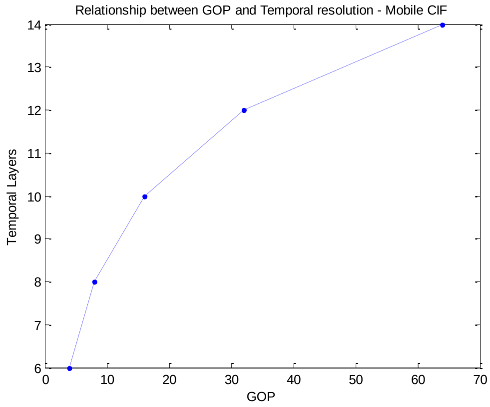
Several numbers of sequences are used in these experiments to determine how the subjective and objective performances are varied from different *GOP* unit's structure in the bit-stream. Figure 5 - 19 results show the scalability layers produced from variable *GOP* unit structure for QCIF resolution of the mobile sequence. A large number of *GOP* units increase the number of scalability layers. This is because each frame entity of *GOP* unit represents the whole picture in that unit. Therefore, there is room to represent the unit with more layers having different frame and bit-rates. Increasing the number of frame entities in a *GOP* unit allows better prediction and this is proved from Figure 5 - 31 for the bit-stream sizes and *GOP* relationship experiments. Although this has improved the scalable stream, however it added some computational complexity. It is therefore a trade-off between complexity and effective scalability. The trade-off between EL coding efficiency and drift can be adjusted by the choice of number of hierarchy stages or the *GOP* size. In this view, we can conclude that ineffective scalability will yield an ineffective adaptation to the network and effective scalability might experience computational complexity from the source encoder [6]. However, this can be improved by the used of high-speed processors, especially where scalability and adaptation are prioritised for a guaranteed quality picture.

5.6.2. *GOP* and Temporal Layers Analysis

An encoded video with the highest *GOP* number provides much efficient compression and bit-rates saving on the scalable stream. This is due to better prediction within the frames of a *GOP* unit. A higher level of scalability is also indicated in the experiments presented in Figure 5 - 19 (a, b). A CIF resolution sequence in Figure 5 - 19 (b) shows that, a *GOP* structure with 64 pictures generates 14 temporal scalable quality layers with variable frame rates, bitrates on to the network and *GOP* structure with 32 pictures produces 11-12 scalable quality layers. *GOP* from Figure 5 - 19 and Figure 5 - 20 represents the number of pictures (frames) allocated in a *GOP*.

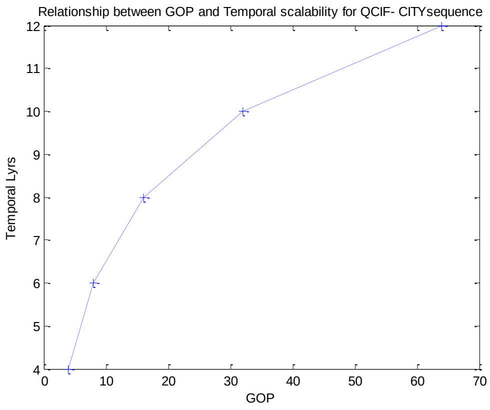


a. MOBILE, QCIF

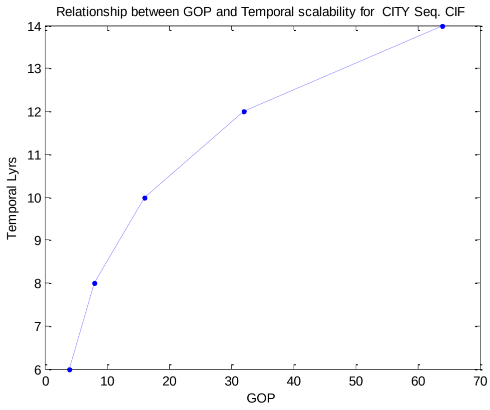


b. MOBILE, CIF

Figure 5 - 19 : (a) Relationship between *GOP* and Temporal Scalability – L0 (b) Relationship between *GOP* and Temporal Scalability L3.



a. CITY, QCIF



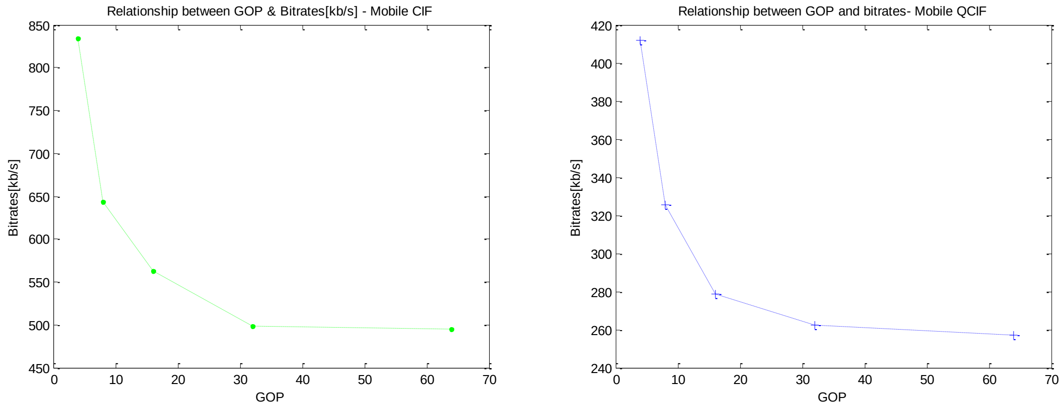
b. CITY, CIF

Figure 5 - 20 : (a) Relationship between *GOP* and Temporal Scalability – L0 (b) Relationship between *GOP* and Temporal Scalability – L3

5.6.3. *GOP* and Bit-rates Analysis

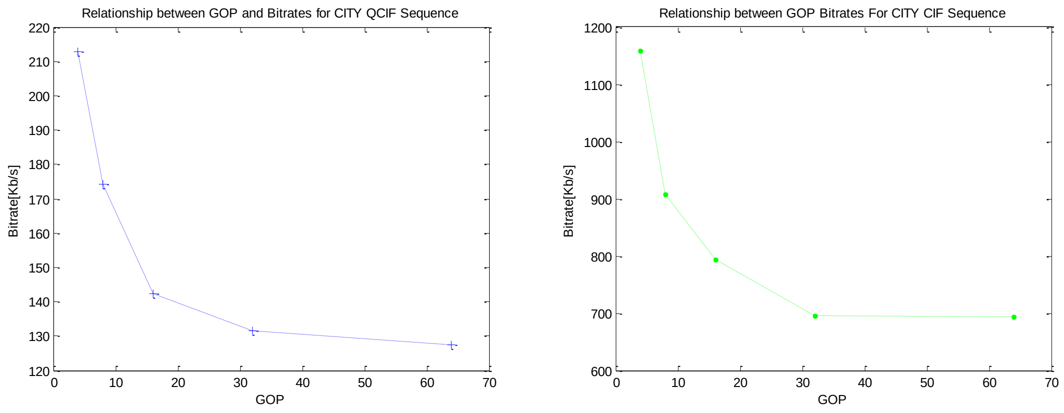
Figure 5 - 21 (a) is CIF resolution mobile video sequence experiment which represents the bitrates performance for scalable layer 2 embedded in a bit-stream of spatial, quality and temporal scalability. The *GOP* units of 4, 8, 16, 32 and 64 values are used in the experiment. A *GOP* within 32 and 64 units produced lower bits and thereby making the scalable stream much more adaptable within the network. The lower bits are achieved by reducing the number of intra-frames within the stream as indicated in Figure 5 - 26. This is due to lower bit-rates which can reduce congestion within the network and can support lower rates

applications to decode from the bit-stream. These variable *GOP* structures support variable scalability adaptation and therefore a *GOP* structure is chosen on the basis of network or application requirements.



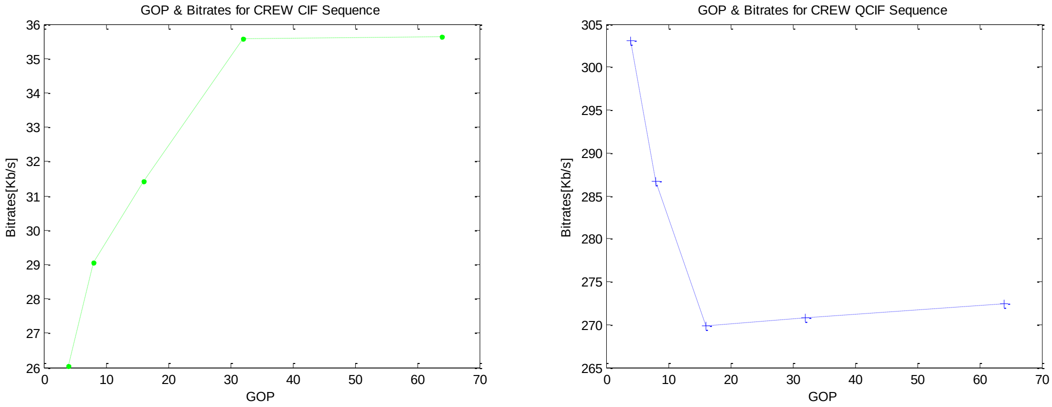
a. MOBILE, CIF   b. MOBILE, QCIF

Figure 5 - 21 : (a) Relationship between *GOP* and Bitrates –L3 (b) Relationship between *GOP* and bitrates –L0



a. CITY, QCIF   b. CITY, CIF

Figure 5 - 22 : (a) Relationship between *GOP* and Bitrates – L0 (b) Relationship between *GOP* and Bitrates -L3



a. CREW, CIF

b. CREW, QCIF

Figure 5 - 23 : (a) Relationship between *GOP* & Bitrates – L3 (b) Relationship between *GOP* and Bitrates –L0

Figure 5 - 24 shows the bits characterisation for several video *GOP* sizes. The experiment is conducted with football standard sequence at 224kbit/s. The results revealed that low *GOP* sizes can consume large amounts of bits due to extra bits generated from the allowable intra-frames within the picture. This can reduce scalability efficiency and adaptation. An aligned and optimised *GOP* should be considered for different applications. This is because applications are characterised as real and non-real time and various delay requirements. For instance, videoconferencing applications can employ *GOP* of size 8 or 16 for efficiency and effectiveness, while storage, Internet and some streaming applications might employ *GOP* of size 64 for better scalability and adaptation.

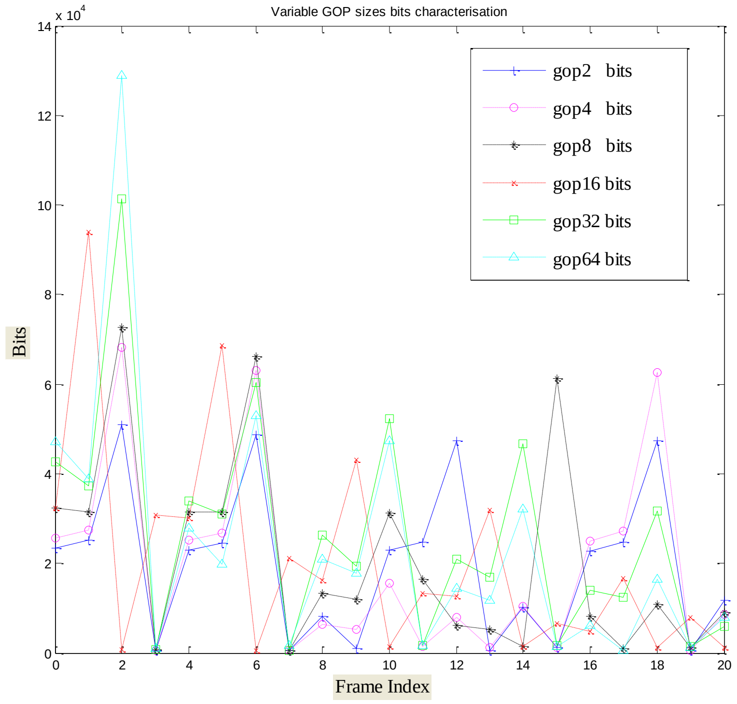


Figure 5 - 24 : Variable *GOP* structure scalability characterisation- football sequence at 224kbit/s, 130frames QCIF.

In order to subjectively evaluate the performance of the variable *GOP* sizes, different *GOP* structure sizes (2, 4, 8, 16, 32 and 64) experiments are conducted with football sequence (standard sequence provided from ITU). The result is presented in Figure 5 - 25. All the *GOP* sizes are coded at the same bit-rate of 224kbit/s and the same other coder configurations. A good picture quality is generated from all the *GOP* structures.

a. *GOP* = 2

b. *GOP* = 4

c. *GOP* = 8

d. *GOP* = 16

e. *GOP* = 32 f. *GOP* = 64

Figure 5 - 25: Frame 120, 224Kb/s, 130 frames football sequence, QCIF- Subjective assessment of variable *GOP* sizes coding (a) 2, (b), 4, (c) 8, (d) 16, (e) 32, (f) 64

Figure 5 - 26 illustrates the *GOP* structure within a video stream coding presentation. Pic1 is the key picture for each *GOP* size which is an I-frame. An I-frame is inserted at the start of a *GOP* for synchronisation purposes.

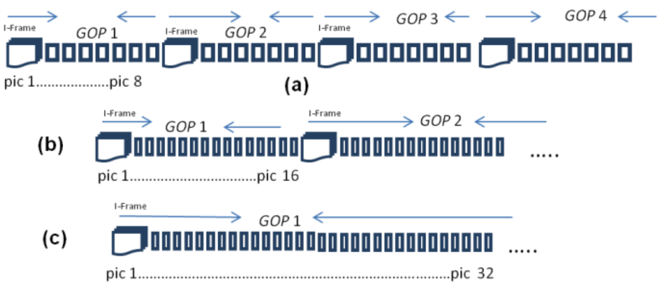
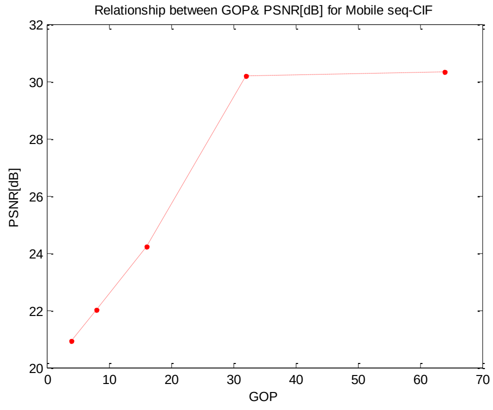


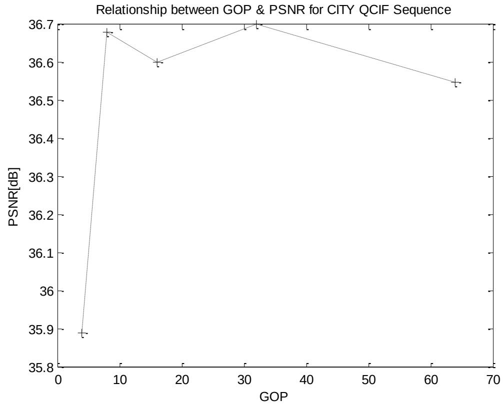
Figure 5 - 26 : (a) 8 -*GOP* Units with 4-intrafames (key pictures) and 32 pictures (b) 16- *GOP* units with 2-intraframes and 32 pictures (c) 32-*GOP* units with 1-intraframe and 32 pictures.

5.6.4. *GOP* and PSNR Analysis and Evaluation

In Figure 5 - 27, *GOP* units with higher number of pictures generate higher PSNR values even though there are fewer bits generated than for the case of a *GOP* with a smaller number of pictures. This is because a higher number of *GOP* unit pictures achieved better prediction and therefore produce a smaller number of coded bits.

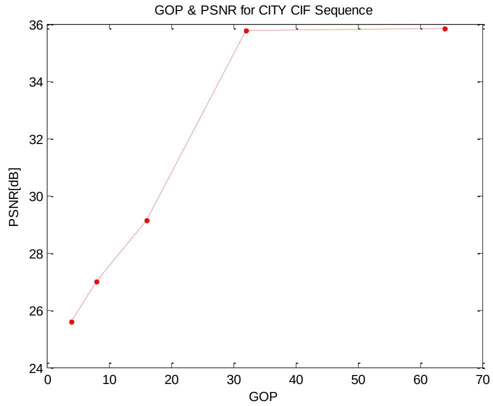


a. MOBILE, CIF

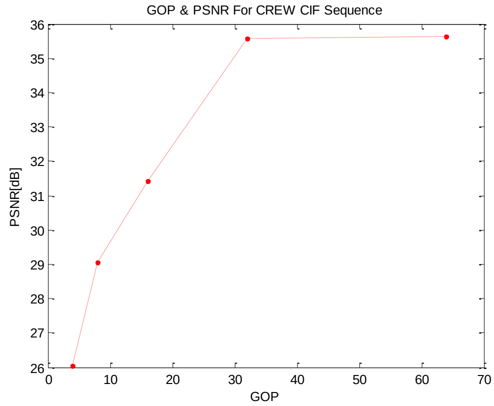


b. CITY, QCIF

Figure 5 - 27 : (a) Relationship between *GOP* and PSNR [dB] –L3 (b) Relationship between *GOP* and PSNR [dB] -L0

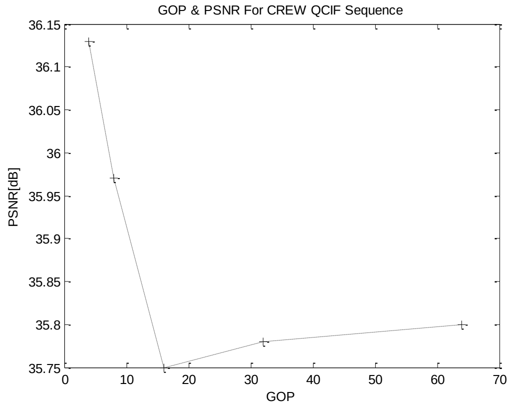


a. CITY, CIF



b. CREW, CIF

Figure 5 - 28 : (a) Relationship between *GOP* & PSNR – L3 (b) Relationship between *GOP* and PSNR –L3



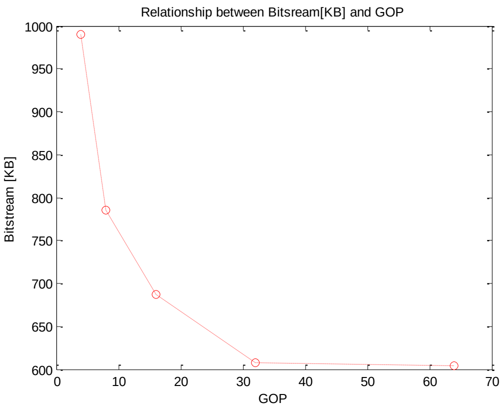
CREW, QCIF

Figure 5 - 29 : Relationship between *GOP* & PSNR –L0

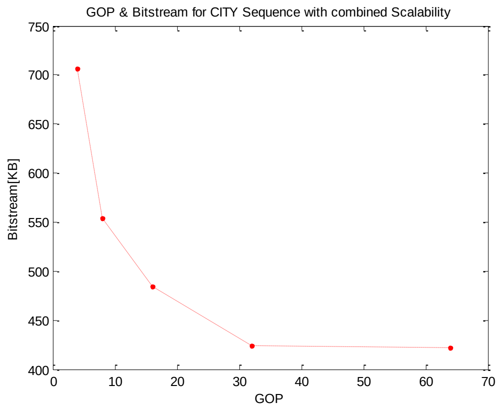
5.6.5. *GOP* and Bit-stream size Analysis and Evaluation

Experimental results in Figure 5 - 30 and Figure 5 - 31 show that a large number of temporal hierarchical stages generate a highly compressed video data. This meant that spatial and temporal redundancies are efficiently removed. Hence network performance and scalability adaptation are improved.

A *GOP* size of 32 and 64 indicate an optimum hierarchical structure with efficient compression. It shows a difference of ~100KB of reduced bit-stream size compared with *GOP* size of 16 and up to ~400KB with *GOP* size of 2. However, there is an increase in complexity which can be significant for some applications. The complexity analysis is discussed in section 5.6.7.

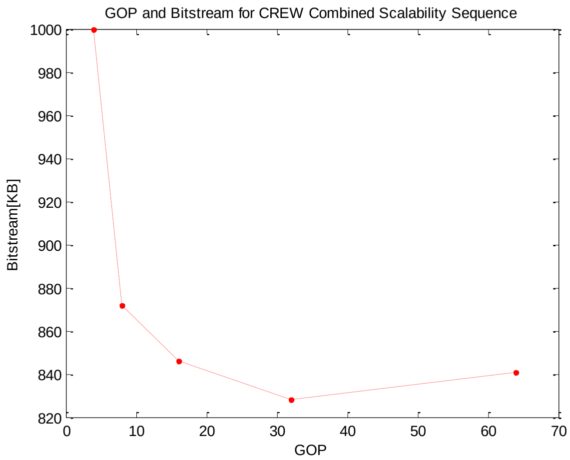


a. MOBILE, CIF



b. CITY, QCIF

Figure 5 - 30 : (a) Relationship between *GOP* and Bit-stream, 300 frames -L3 (b) Relationship between *GOP* & Bit-stream for CITY Sequence with *t*+*s*+*q* 150 frames –L0



CREW

Figure 5 - 31 : Relationship between *GOP* and Bit-stream with $t+s+q$ scalability.

5.6.6. Subjective Evaluation and Analysis of *GOP* structure

Since the variation of hierarchical stages or *GOP* size impacts on the coding efficiency and required bits to be spent, picture quality can also be influenced. To evaluate and assess these variations and influences, experiments are conducted with the standard ITU sequence soccer at variable video coding hierarchical stages of *GOP* sizes equal 2, 4, 8, 16, 32 and 64. The lower *GOP* sizes pictures show slightly better brightness at the fore and background. This can be noticed around the wall, fence and the playing ground. The quality difference resulted from additional bits used to code pictures with small *GOP* sizes. The bits large size is due to the large number of unpredicted frames in their *GOP* frames. However, a good picture quality is achieved within all the *GOP* sizes. The results are presented in Figure 5 - 32.

a. *GOP* = 2

b. *GOP* = 4

c. *GOP* = 8

d. *GOP* = 16

e. *GOP* = 32

f. *GOP* = 64

Figure 5 - 32 : Variable Hierarchical stages encoded at 128Kb/s, frame 70 of *GOP* sizes (a) 2 (b) 4 (c) 8 (d) 16 (e) 32 and (f) 64

5.6.7. Complexity Analysis and Evaluation for *GOP* structure

Experiments are conducted using a number of *GOP* sizes and the complexity for each of the *GOP* structure in the bit-stream is computed as in Table V - III. The encoding time is obtained by automatically recording the start and end times of the process. This time is obtained from encoding four multiple layer stream using combined temporal, spatial and quality (t+s+q) scalability.

Table V - IV: *GOP* Structure: Encoding Time and SCI- foreman/harbour videos.

***GOP* size**	**Encoding time (min)** Foreman/harbour	**SCI** Foreman/harbour	**Achieved Scalable layers** Foreman/harbour
2	1.73/1.68	2128/2128	6/6
4	2.58/2.74	1872/1872	10/10
8	3.17/3.2	1038/1038	13/13
16	3.57/3.78	549/549	18/18
32	4.05/4.06	255/255	22/22
64	4.15/4.34	222/222	26/26

In the experiments from Table V - IV, it is shown that when a *GOP* size is doubled there is an additional encoding time of about 3%. This is because increasing a *GOP* size results for additional prediction cost for every *GOP* unit within the total video sequence. This has increased the computational task and processes. However, the number of pictures available for predicting each *GOP* unit is increased and therefore additional gain for more temporal layers and scalability adaptation is obtained. This is evidenced in Figure 5 - 26. The results in Table V - V describe the scalability provided for a *GOP*=16 and *GOP*= 32. Frame rate and bit-rates are shown for each of the structure. Also dependency_id, temporal_id and quality_id are also shown. A quality_id equal 1 is better than that equal 0. The temporal_id represents the associated temporal scalable layer.

Table V - V : Scalability stream performance for *GOP* = 32 and *GOP*=16

Layer	***GOP*-size/ Identifier**	**Resolution**	**Frame-rate(Hz)**	**Bitrates (Kb/s)**	**Dependency_id (D)**	**Temporal_id (T)**	**Quality_id (Q)**
0	16	QCIF	1.8750	35.50	0	0	0
0	32	QCIF	0.9375	21.70	0	0	0
1	16	QCIF	3.7500	47.30	0	1	0
1	32	QCIF	1.8750	28.60	0	1	0
2	16	QCIF	7.5000	59.30	0	2	0
2	32	QCIF	3.7500	40.00	0	2	0
3	16	QCIF	15.0000	73.30	0	3	0
3	32	QCIF	7.5000	52.70	0	3	0
4	16	QCIF	1.8750	71.60	0	0	1
4	32	QCIF	15.0000	66.70	0	4	0
5	16	QCIF	3.7500	93.10	0	1	1
5	32	QCIF	0.9375	43.10	0	0	1
6	16	QCIF	7.5000	115.00	0	2	1
6	32	QCIF	1.8750	55.90	0	1	1
7	16	QCIF	15.0000	137.30	0	3	1
7	32	QCIF	3.7500	76.80	0	2	1
8	16	CIF	1.8750	119.20	1	0	0
8	32	QCIF	7.5000	99.80	0	3	1
9	16	CIF	3.7500	152.00	1	1	0
9	32	QCIF	15.0000	121.80	0	4	1
10	16	CIF	7.5000	188.20	1	2	0
10	32	CIF	0.9375	67.30	1	0	0
11	16	CIF	15.0000	232.80	1	3	0
11	32	CIF	1.8750	88.10	1	1	0
12	16	CIF	30.0000	272.20	1	4	0
12	32	CIF	3.7500	121.20	1	2	0
13	16	CIF	1.8750	249.30	1	0	1
13	32	CIF	7.5000	158.50	1	3	0
14	16	CIF	3.7500	312.00	1	1	1
14	32	CIF	15.0000	203.30	1	4	0
15	16	CIF	7.5000	376.00	1	2	1
15	32	CIF	30.0000	242.60	1	5	0
16	16	CIF	15.0000	452.20	1	3	1
16	32	CIF	0.9375	13.80	1	0	1
17	16	CIF	1.8750	518.20	1	4	1

17	32	CIF	30.0000	177.80	1	1	1
18	32	CIF	3.7500	240.00	1	2	1
19	32	CIF	7.5000	306.30	1	3	1
20	32	CIF	15.0000	382.70	1	4	1
21	32	CIF	30.0000	448.70	1	5	1

The coder achieves temporal scalability by skipping frames. It utilises different frame rates to achieve this as in Table V - V. The *GOP*=32 coder started with 0.9375 frame per second against 1.8750 frames per second for *GOP*=16. This enables the coder to achieve a minimum bit-rate of 21.70Kb/s against 35.50Kb/s for a stream of *GOP*=16. This characteristic improved its scalability adaptation than the lower *GOP* structure. In addition *GOP*=32 generates 22 scalable layers against 18 layers for *GOP*=16 stream.

Table V - VI describes the decoding timings for base and enhancement layers for *GOP* structures. The reconstruction is obtained from $t+s+q$ scalable bit-stream. From the results obtained on foreman sequence, a *GOP* size = 2, 4, 8, 16, 32 and 64 from the base layer is reconstructed within 0.86sec, 0.79sec, 0.76sec, 0.75sec, 0.73sec and 0.77sec respectively. This result implies that, a *GOP* with a higher number of frames does not necessarily take much time in the reconstruction process. The decoder extracts each layer using header information received which provides layering information embedded in the bit-stream.

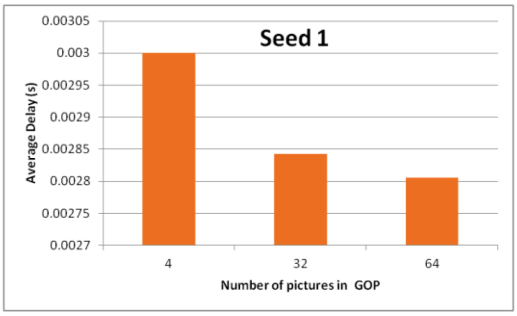
Table V - VI : Decoding time for variable *GOP* structure –foreman and harbour sequence

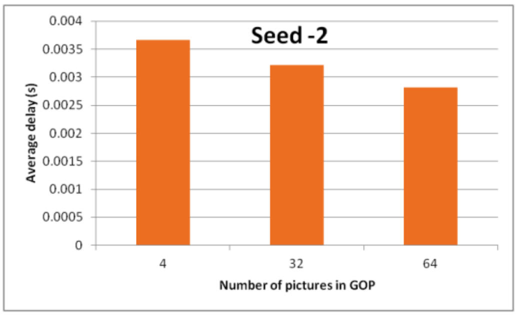
Layer	***GOP*-unit Foreman/Harbour**	**Decoding time(sec) Foreman/Harbour**
Base	2	0.86/0.96
Enhancement	2	5.0/5.57
Base	4	0.79/0.84
Enhancement	4	4.75/5.2
Base	8	0.76/0.82
Enhancement	8	4.47/5.09
Base	16	0.75/0.77
Enhancement	16	4.68/5.0
Base	32	0.73/0.76
Enhancement	32	4.62/5.03
Base	64	0.77/0.78
Enhancement	64	4.63/4.97

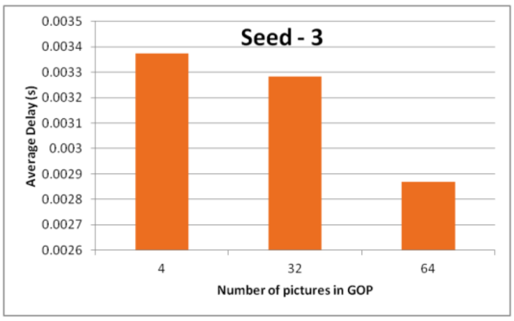
A layer with a large *GOP* size will possess sufficient information that facilitates the reconstruction process. This is because improved and more efficient prediction is achieved as earlier described. In this process, the decoder stores layers of information in a different memory position for each of the used *GOP*. For example for *GOP*=16, a T_o layer is predicted from frame 22 and for *GOP*=32 T_4 layer is predicted from this frame at this time. Hence, these frames values are stored in different positions in the array and therefore might have slight differences in search or access. This and the *GOP* size difference accounted for variable decoding time of the base and enhancement layers.

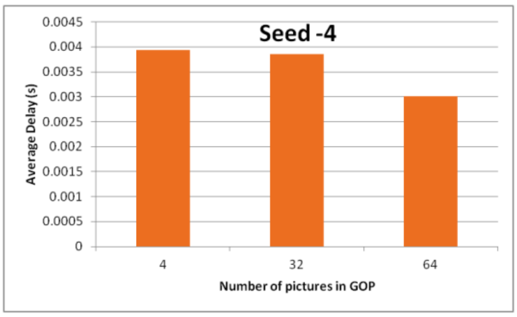
5.6.8. Real Time performance Analysis and Evaluation for variable *GOP* structure

An experiment is set-up to evaluate the real time performance of different *GOP* structures. An ITU standard sequence harbour is used. Three encoder outputs are generated with *GOP* = 4, *GOP*=32 and *GOP*=64 respectively. Each of the encoder output is streamed into a live heterogeneous network simulator where delay statistics are collected at the end. The statistics results indicated that higher delays are experienced with *GOP*s with lower number of pictures. This is due to less-efficient compression and high number of intra-frames. However, the *GOP* with small number of pictures will be less complex in encoding but will not be efficient in scalability as the large *GOP* size supports large hierarchical picture predictions. This became a trade-off in efficiency of scalability and complexity. Applications with higher processing capacity will perform efficiently with *GOP*s predicting large number of pictures. Hence, large numbers of scalability layers are produced for better adaptation and network degradation control.









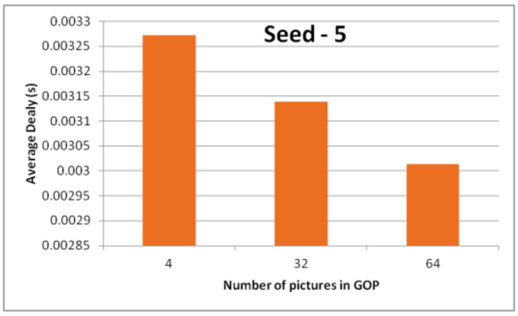


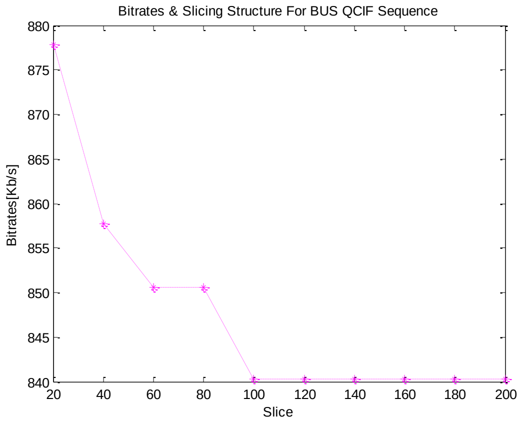
Figure 5 - 33 : *GOP* structures: Average delay statistics from real time simulations.

5.7. Performance and Evaluation of Slices and Slice Groups

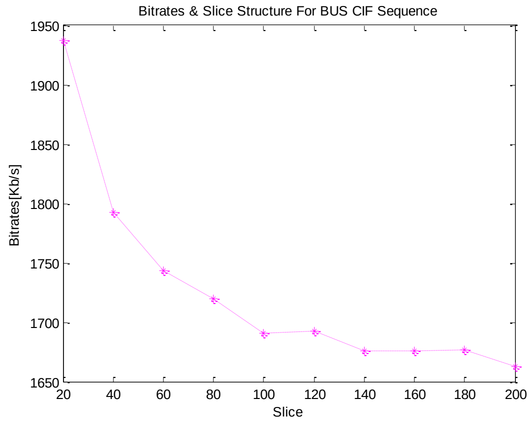
Slices are structured as a sequence of MBs which are processed in the order of a raster scan when not using Flexible MB Ordering (FMO). FMO is a recent coding ability that partitions a picture into regions referred to as slice group [7]. When this feature is effectively used, it can significantly enhance robustness to data losses and scalability adaptation. This is achieved by managing the spatial relationship among regions coded in each slice. A video sequence can be split into several slices and therefore is a collection of one or more slices. Slices are self-contained in the view that slice representation can be correctly decoded without the use of data from other slices provided that utilised reference pictures are identical at encoder and decoder. With FMO, video can be split into many MB scanning patterns such as interleave slices, foreground and slice groups as described in section 5.8 and in chapter three. The effects of these different scanning patterns on scalability adaptation are evaluated.

5.7.1. Slice Structure and Bit-rates Analysis

The experimental results in Figure 5 - 34 to Figure 5 - 36 show that, the number of MBs coded as a slice has impact on video adaptation. The coding efficiency is improved for a large number of macro-blocks forming a slice. However, this might reduce the error resiliency of the video. From Figure 5 - 35(b), a bit-rate of 450kbit/s, 470kbit/s, and 490kbit/s are spent from 200, 100 and 80 macro-block slices. As a general observation from the results of various sequences and resolution, the required bit-rate is incremented highly when the macro-blocks for a slice are less than a hundred. This revealed that the slice structure above hundred MBs will be an optimum for scalability adaptation and for robust error resiliency applications.

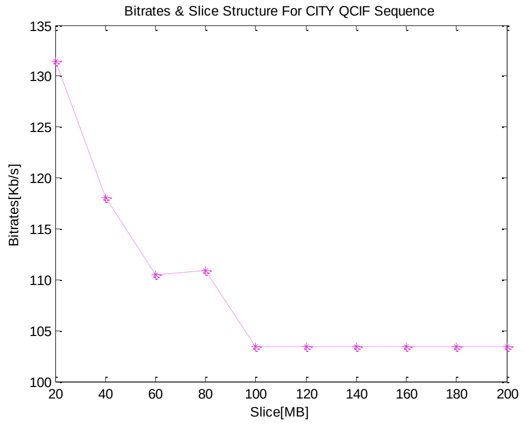


a. BUS, QCIF

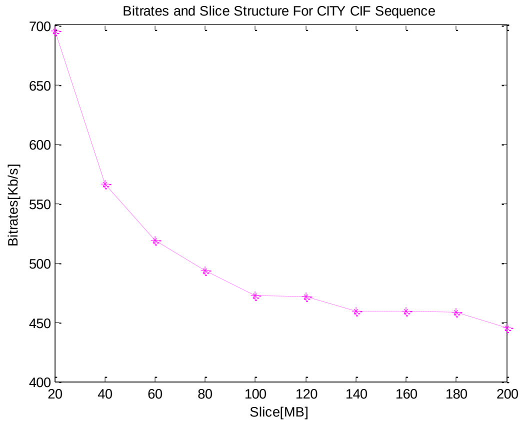


b. BUS, CIF

Figure 5 - 34 : (a) Relationship between Bitrates and Slicing Structure –L0 (b) Relationship between Bitrates and Slicing Structure –L3

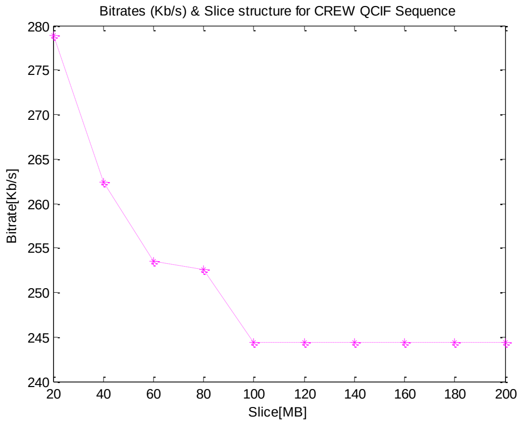


a. CITY, QCIF

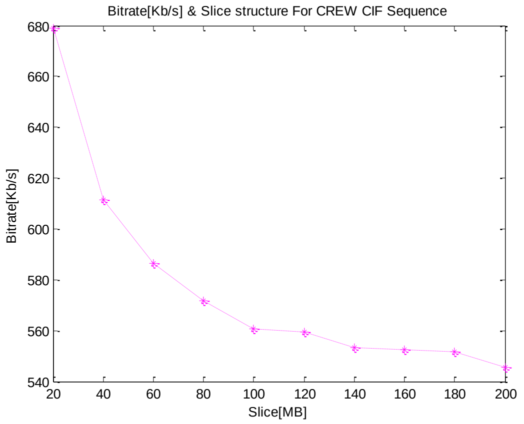


b. CITY, CIF

Figure 5 - 35 : (a) Relationship between Slice Structure and Bit-stream with *t+s+q* Scalability –L0. (b) Relationship between Bitrates and Slice Structure –L3



a. CREW, QCIF

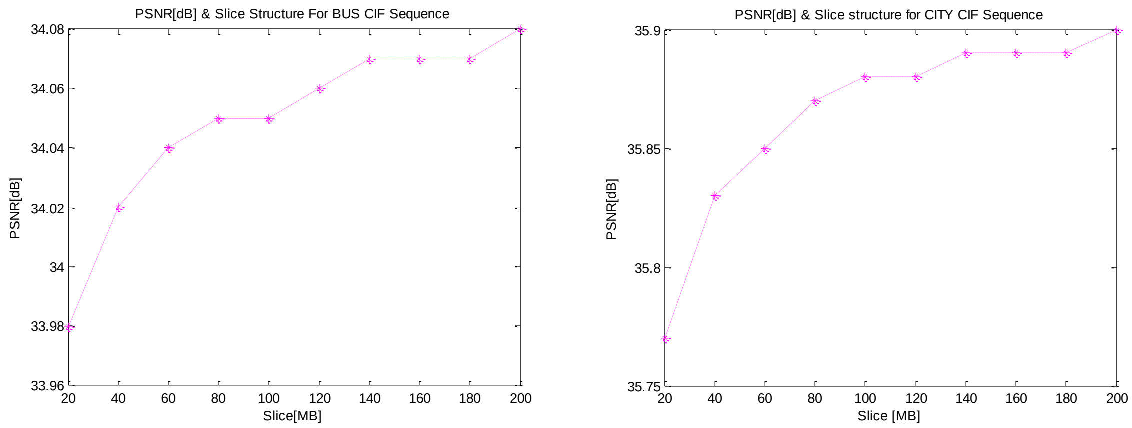


b. CREW, CIF

Figure 5 - 36 : (a) Relationship between Bitrates & Slice structure -L0 (b) Relationship between Bitrates & Slice structure –L3

5.7.2. Slice Structure and PSNR Analysis and Evaluation

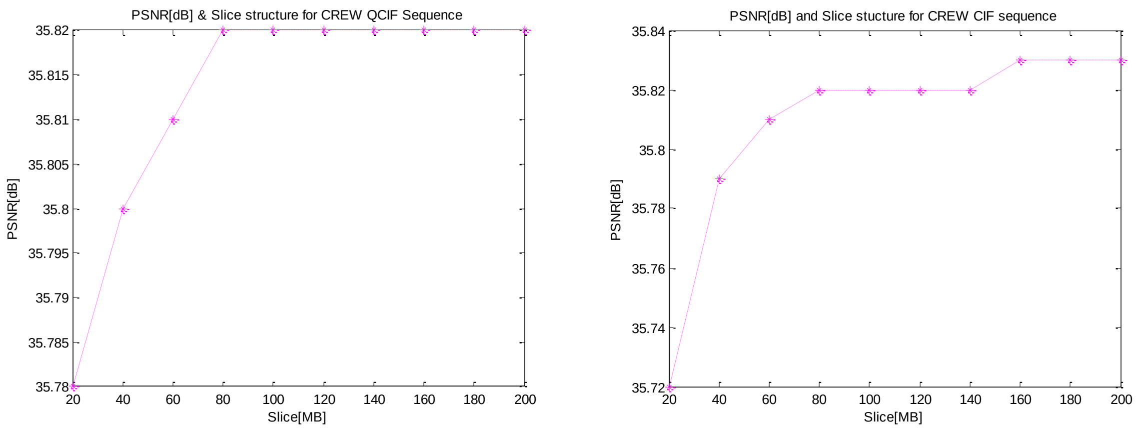
Experimental results presented in Figure 5 - 37 to Figure 5 - 38 show that high values of γ-luminance are generated from slices with a large number of blocks. This shows better objective quality performance for the slice groups. The PSNR values [dB] tend to be approximately constant at slice groups, equal to hundred and above. This is a general observation from the results for several sequences and resolution formats. The H.264 Qp adaptation favours the achievement of quality rather than bit reduction. In view of this, even though the increase of slice groups supports bit-rate reduction and coding efficiency, it might not be supported (bit-reduction) at some level where quality degradation is anticipated.



a. BUS, CIF

b. CITY, CIF

Figure 5 - 37: (a) Relationship between PSNR [dB] and Slice Structure –L3 (b) Relationship between PSNR [dB] and Slice structure –L3.



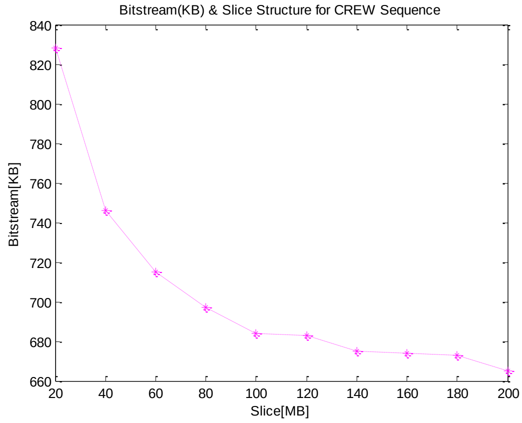
a. CREW, QCIF

b. CREW, CIF

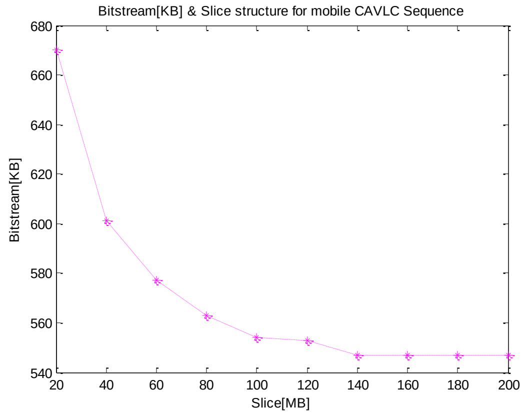
Figure 5 - 38 : (a) Relationship between PSNR & Slice structure –L0 (b) Relationship between PSNR & Slice structure –L3.

5.7.3. Slice Structure and Bit-stream size Analysis and Evaluation

Experimental results in Figure 5 - 39 to Figure 5 - 41 show the compression efficiency of slice group sizes. The output video packets for slice group sizes from hundred to two–hundred has a reduced size with more efficient compression. This range of slice groups sizes (100 -200) has shown better performance in rate distortion optimisation and compression efficiency. Hence, it is a suitable slice group size for better rate control and scalability adaptation performance.

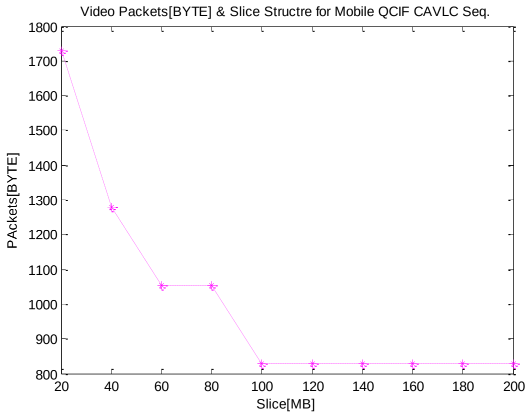


a. CREW

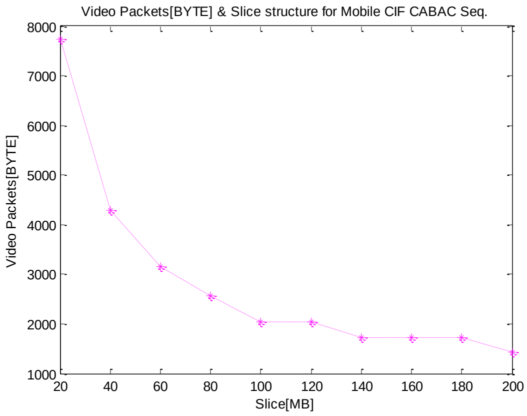


b. MOBILE

Figure 5 - 39 : (a) Relationship between Slice Structure and bit-stream size with *t+s+q* Scalability –L3. (b) Relationship between Slice Structure and bit-stream size with *t+s+q* – L3.

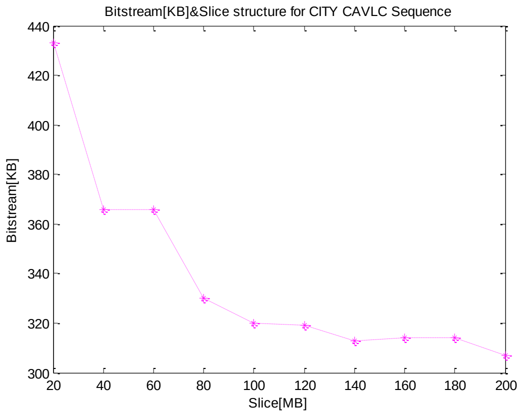


a. MOBILE, QCIF

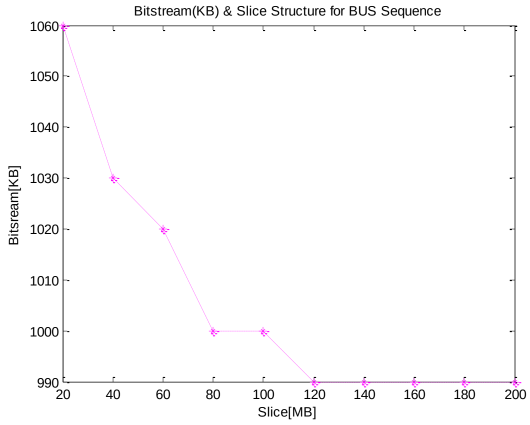


b. MOBILE, CIF

Figure 5 - 40 : (a) Relationship between Video packets output and Slice Structure –L0 (b) Relationship between Video Packets & Slice structure – L3.



a. CITY



b. BUS

Figure 5 - 41: (a) Relationship between Slice Structure and bit-stream size with *t+s+q* scalability – L3 (b) Relationship between video packets & slice structure with *t+s+q* scalability –L0

5.8. Performance and Evaluation of Macro-block to slice group mapping

In every coded picture, there exists a slice group as a subset of macro-blocks. The macro-block may contain one or more slices. Macro-blocks are coded in raster order within each slice in a group [8]. The encoding of macro-blocks in raster order is adopted where only one slice group is used per picture unless arbitrary slice order (ASO) is used. ASO which is supported for Baseline Profile allows slices in a coded frame to follow any decoding order. This is defined to be used if the first MB in any slice in a decoded frame exists with a smaller MB address than the first MB in a previously decoded slice within the same picture.

Descriptions and used notations for the slice group map types are presented in section 3.9. To evaluate the characteristics of the slice group map types, real time simulations are conducted. This is to assess which of the mapping methods adapts best to the reported network conditions. Figure 5 - 42 presents the real time performance of the mapping types.

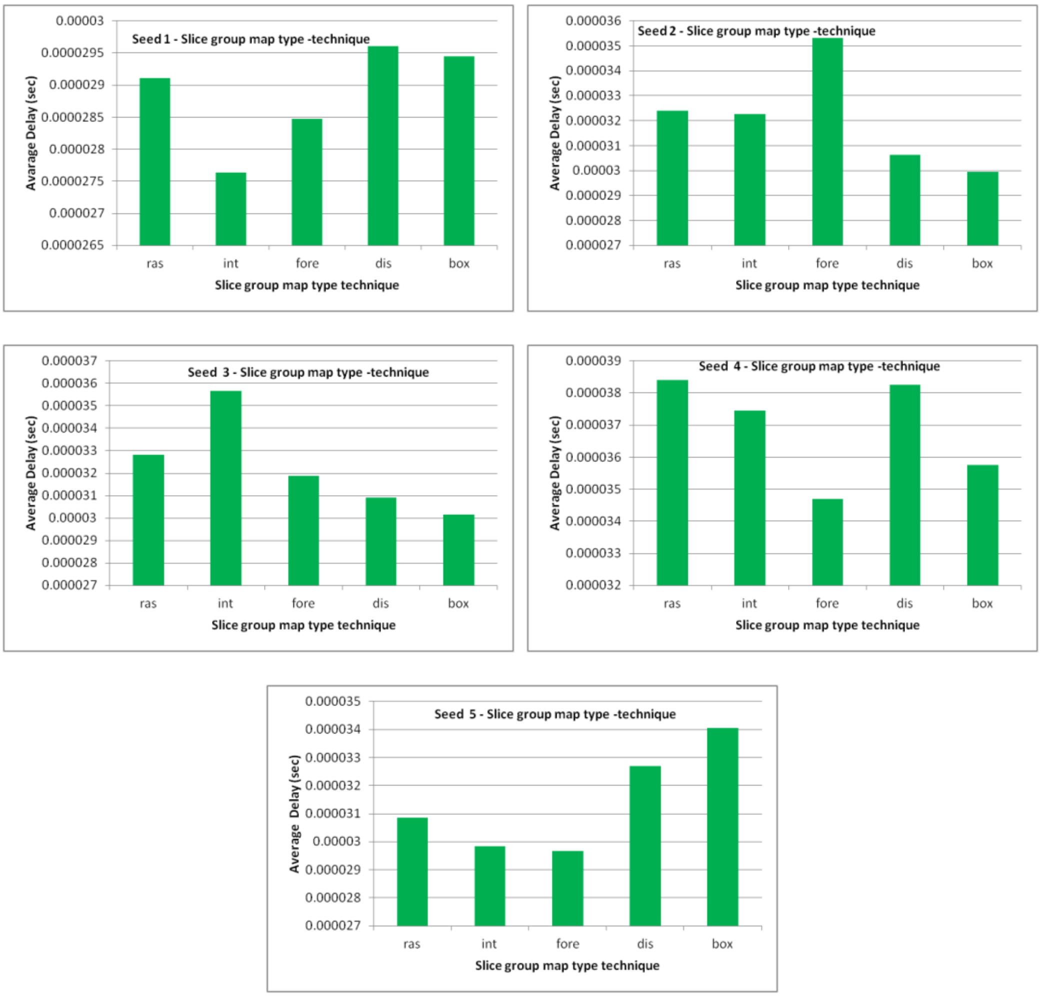


Figure 5 - 42 : Macro-block to Slice Group Map Types- Real Time Performances over Heterogeneous Networks

Table V - VII – Real time simulations over heterogeneous networks for MB to slice-group map type

Map-type	**Average delay (secs)**
Ras	0.01636
Inter	0.016286
Fore	0.016001
Dis	0.01621
Box	0.015933

5.9. *SNR* Scalability

As discussed in section 6.3.1, a minimum of two video sequences (two layers) are required to achieve *SNR* (Quality) scalability. The main intended applications for *SNR scalability* include telecommunication and video services that require several distinct qualities e.g. standard TV and HDTV video services.

The two required minimum layers for *SNR* scalability implementation are of the same spatial and temporal resolutions coded from one video source. One of the layer is the BL coded to provide the basic picture quality, and the second layer (EL) is coded to support the enhancement of the BL. When the BL is added to the EL, an enhanced picture quality is achieved. Since the EL is said to improve the signal-to-noise-ratio of the BL, this type of scalability is termed *SNR* (*signal-to-noise-ratio*). Alternatively, *SNR* scalability could have been called *Coefficient Amplitude Scalability* (CAS) or *Quantisation Noise Scalability* (QNS) [3].

Figure 5 - 43 represents a block diagram for two layers Quality Scalable Encoder. The input video is coded at low bit-rate to produce the base layer bit-stream with lower image quality. The Enhancement Layer bit-stream is generated by coding the difference between the input video and the decoded output of the base layer. The BL and EL bit-streams are then multiplexed for transmission over the network.

In the above discussed *SNR* scalability implementation (MPEG-1 and MPEG-2), it can be observed that two non-scalable encoders are required which introduces additional complexity as compared to data partitioning which uses only a single video sequence. In *H.264/AVC* which is the most recent codec, interlayer prediction and a flexible temporal scalability can be implemented to improve in the scalability. Since, the two *SNR* layers are of the same spatial resolution, there is a big room to remove spatial redundancies and therefore reduce the bit-stream size. The inter-layer prediction has been implemented with the results in section 6.10 while temporal scalability has been employed in the developed techniques in section 6.3.

In *SNR* scalable coder where the quantisation of the BL and the EL is tightly couple together as shown in Figure 5 - 44, picture drift is likely to be generated in the BL bit-stream. This is because BL is decoded independently. The drift will not be affecting the second layer (EL) since both the two layers are normally decoded together for better picture quality. Decoders that are not capable to decode both the EL and BL, usually reconstruct the video sequence from the base layer stream.

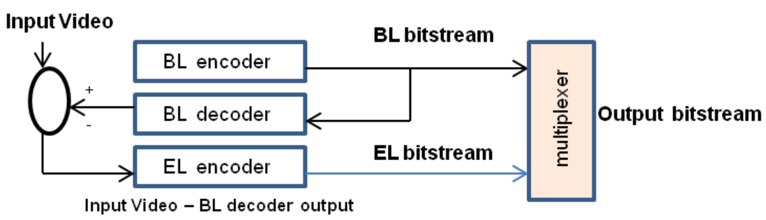


Figure 5 - 43 : Illustration of Two Layer SNR scalable encoder.

To reduce the predictable drift in the BL picture and improve on the performance of *SNR* scalability, the following measures and ideas are proposed:

1. The prediction loop of the BL should not be fed with EL data.

2. Each layer should be coded with its own quantiser with consideration that it is a BL, lower or higher EL. This implementation will support an *enhanced multi-quality SNR* scalability. The experimental implementation of this idea is discussed in section 6.3.6 and section 2.4.4.

3. *SNR* scalability is intended for multi quality applications or multi-channel environment. An independent consideration of each layer quantiser can provide an adaptable *multi-quality bit-stream* in a heterogeneous environment. The modelling of each layer quantiser can be obtained based on the following factors (1) *number of SNR quality layers used* (2) *intended applications* (3) *scalability level (4) target bit-rate* and (5) the *coder quantiser-step-size influence* on video quality. Several distinct models of *SNR* scalability performances can be achieved with this modelling parameters and approach. These models can be utilised on the basis of applications and channels characteristics.

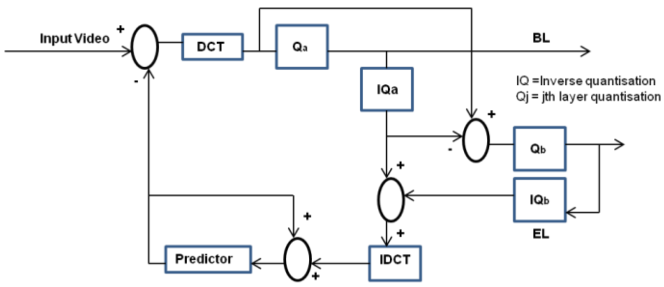


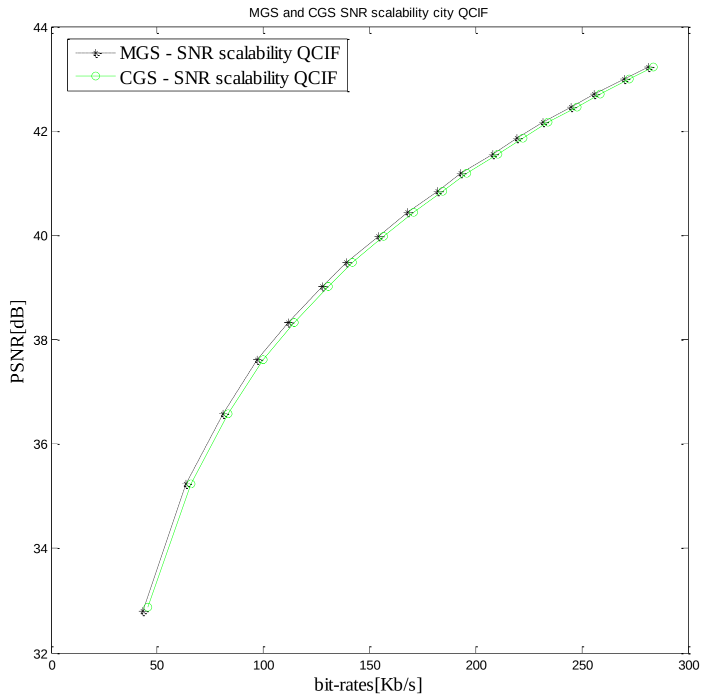
Figure 5 - 44: Two-Layer *SNR* encoder with drift at the Base Layer [3]

5.10. Performance Evaluation of *SNR* Scalability Methods

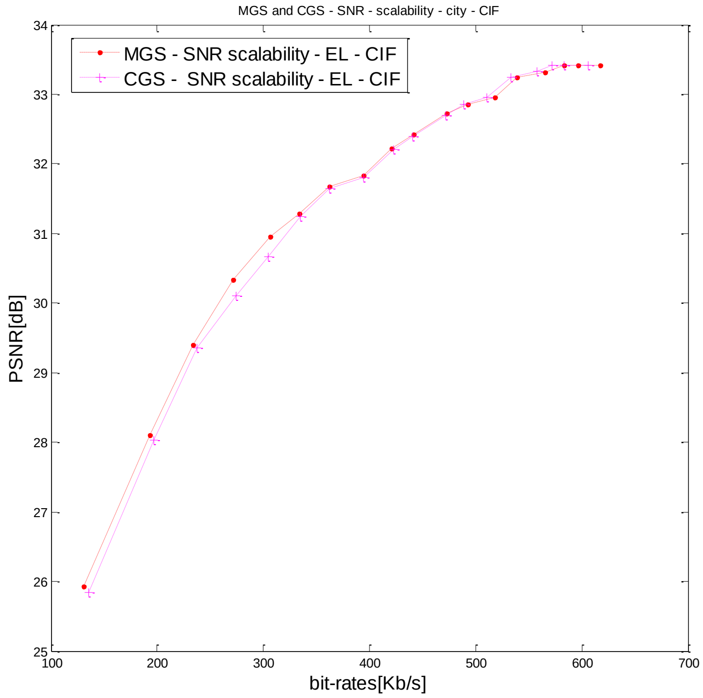
In Scalable video coding, there exist two possibilities for the provision of SNR scalable bit-streams. One method is referred to as *CGS* (Coarse Grain Scalability) and the second method is known as *MGS* (Medium Grain Scalability). In standard algorithms, *MGS* and *CGS* are identical to spatial scalable coding with the only exception that all layers have a similar spatial resolution [9].

In comparison between the two methods, *MGS* coding supports many rate points up to 25% where *CGS* supports less rate points up to 10% only [10]. In *MGS*, coding an arbitrary subset of *MGS* layers packets may be extracted (subset of *MGS* layers packets). In addition to this, *MGS* encoder may choose to partition the transformed coefficients of a layer up to 16 *MGS* layers. This increases the number of packets and the number of subsets of packets. In this procedure, a finer granularity (section 2.4.5) is achieved.

To further evaluate and assess the performance of *MGS* and *CGS* scalable bit-streams, an experiment is set-up with city video sequence. The sequence is encoded and configured for *MGS* and *CGS* bit-stream outputs respectively. The experiment is conducted for both high and low resolutions formats. There is no difference observed in the base layer (BL) of *MGS* and *CGS*. This is because BL is a reference layer and all the subsequent subset layers are enhancement layers of BL where *MGS* rate points are generated. BL is an independent layer which all other subset layers and packets depend on and reference from in both *MGS/CGS* implementations. Figure 5 - 45 (a, b) shows the results obtained for QCIF and CIF formats.



a.



b.

Figure 5 - 45 : SNR bit-stream- *MGS*/*CGS* bit-stream performance (a) QCIF resolution (b) CIF resolution

MGS Enhancement Layer (EL) generated 32.80dB, 35.24dB, 36.58dB and 37.61dB from 43000bits, 63540bit, 80940bits and 97290bits per second respectively. Similarly, *CGS* produced 32.88dB, 35.24dB, 36.58dB and 37.61dB from 45560bits, 66120bits, 83520, and 99880bis per second respectively. Both the *MGS* and *CGS* results are obtained from QCIF resolution video.

In high resolution CIF format, *MGS* generated 25.93dB, 28.11dB, 29.40dB, and 30.33dB from 130330bits, 193330bits, 233670bits, and 233670bits per second respectively. *MGS* shows better scalability performance in both CIF and QCIF resolutions. In QCIF, it is observed from the experimental results in Figure 5 - 45 (a) that in the two methods there is a slight difference in performance with *MGS* showing better adaptation especially at bit-rates less than 100kbit/s.

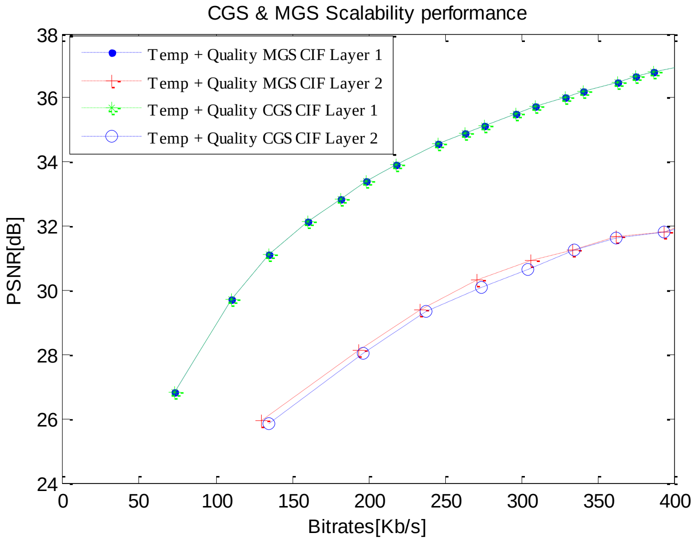


Figure 5 - 46 : *MGS* and *CGS* layers in a multi-layer *t+q* technique

5.11. Performance Evaluation of H.264/AVC Quantiser-Step-Size

The transformation of pixel domain does not actually produce any compression [3]. A block of 32 pixels is transformed into 32 coefficients. The energy in both the pixel and transform domains is equal due to orthonomality. There is no compression achieved then. However, the transformation process causes a significant part of the image energy to be concentrated at the lower frequency components with most of the coefficients having less energy. It is actually the quantisation and variable length or arithmetic coding of the DCT coefficients that lead to bit rate reduction. Table V - IX shows the effect of picture quality and scalability on quantisation step size of 2 units. This explains that as we increase the quantisation unit, lesser bits are generated as in (equation 5-2).

$$Z = X_j/Q_i \qquad 5\text{-}2$$

Where Z is the quantised bits, X_j is the transformed coefficient and Q_i is the Quantisation unit.

There should be caution in adjusting the quantisation unit parameter. This is because yielding a smaller number of bits may deteriorate the picture quality. The experiment conducted and shown in Table V - VIII proves that 2 units increment may not degrade picture quality but improves the scalability adaptation.

Table V - IX : Effect of Quantiser Step Size on Objective quality- *t+s+q* scalability, city, QCIF sequence

Quantisation Step	Bit-rates [Kb/s]	PSNR[dB]
22	181.70	40.50
24	143.70	39.23
26	115.63	38.08
28	95.38	36.94
30	78.25	35.75
32	65.14	34.55
34	54.88	33.37
36	46.14	32.14
38	39.01	30.88
40	33.86	29.68

5.12. Performance of *Encoded* & *Non-Encoded Key* Pictures

The default design of H.264 allows the key pictures of each video sequence to be streamed un-encoded. This results in producing a high number of bits and hence increasing the spent bandwidth. With the encoding of the pictures, this problem is reduced and less number of bits will be required to be spent. Experiments with foreman video sequence show that a minimum quality is still realised at the reconstruction end. In this procedure, error robustness is improved as well as bit-stream scalability and adaptation. Figure 5 - 47 shows a better rate distortion performance efficiency of the encoded over the non-encoded key picture.

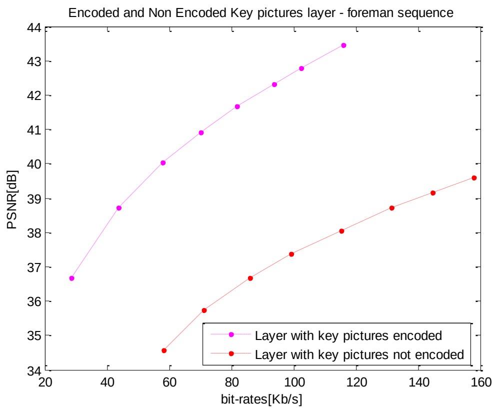


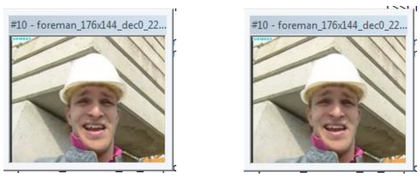
Figure 5 - 47 : Encoded and non-encoded key pictures in scalable stream

Figure 5 - 48 (a) and (b) shows the subjective quality of frame number ten for both the non-key encoded and encoded key pictures respectively. A *GOP* size of 64 is used to structure the

sequence frames. There is no significant difference between the two frames due to high bit-rate used.

Figure 5 - 49 (a) and (b) shows frame seventy non-encoded key picture and encoded key picture with *GOP* size 4 respectively. Figure 5 - 49 (a) shows better quality pictures than the encoded in Figure 5 - 49 (b). This manifests itself when the bitrates are below 64Kb/s and the *GOP* size is small. Hence when a coder is employed to use small *GOP* sizes, the coding efficiency as well as the subjective quality might be reduced for encoding the key picture technique.

Figure 5 - 50 where a *GOP* size of 2 is used for encoding key picture technique shows less bright picture although the picture quality for both encoded and non-encoded key pictures is good. This is because high rates are used up to 224kbit/s. Generally, for the encoded key picture, there is coding bits reduction since the pictures are predicted and motion compensated. Hence, for large number of coded pictures, a large number of bits might be lost and this can result in picture drift especially where bit-rate is low as proved with Figure 5 - 49 experimental results. This became a trade-off when applications must employ a coder with small *GOP* sizes to reduce complexity. The limitation of bandwidth and the user or application requirements are considered. In application such as storage, video highlights etc, bandwidth limitation can be a priority while in applications such as TV broadcast etc, a guaranteed quality picture may be a major concern.



a. b.

Figure 5 - 48: Frame 10 of foreman sequence coded at 224kbit/s, 15f/s, *GOP* size of 64, 75 frames (a).with no encoded key picture and (b) with encoded key picture.

a. b.

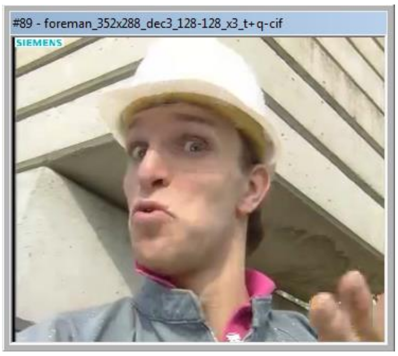
Figure 5 - 49: Frame 70 of foreman sequence coded at 32kbit/s, 15f/s, *GOP* size of 4 (a) with no encoded key picture and (b) with encoded key picture.

Figure 5 - 50: Encoded Key Picture, Frame 10 of foreman sequence coded at 224kbit/s, 15f/s, *GOP* size 2, 150 frames

Table V - X : Performance of sequence with encoded and non-encoded key pictures

Key-Picture	Bit-rates(Kb/s)	PSNR(dB)	Quantisation
Encoded	28.62	36.67	28.60
	43.82	38.71	24.44
	57.91	40.04	22.01
Non-Encoded	58.30	34.57	35.19
	71.00	35.73	31.03
	85.86	36.67	28.60

Table V - X shows that there is also a gain in PSNR values for the encoded pictures up to 4dB. The smaller number of bits used for generating the encoded picture influences the selection of a quantisation parameter to achieve lesser bits and high energy. In the case of non-encoded picture which generates more bits, the quantisation parameter needs to be adjusted within the sequence to compensate production of un-required bits for the key pictures. The three experimental values for bit-rates and PSNR are obtained from experimenting with the foreman sequence. The corresponding values for the un-encoded picture shows that more bits are spent which increased the bit-rates. However, in applications where bandwidth is not limited and there is a high demand for video quality, non-encoded key pictures can produce a guaranteed and better quality reconstructed pictures. Hence, the use of these techniques will depend on bandwidth availability and video quality requirement or user demand.



a.

b.

c.

d.

Figure 5 - 51 : Frame 89 of foreman sequence coded at 128kbit/s, 30f/s, *GOP* size 2, 300 frames (a) (c) with key pictures encoded (b) (d) with no key pictures encoded.

Figure 5 - 51 is a high resolution CIF format for encoded (a, c) and non-encoded (d, b) key pictures. Frame 89 in (a) shows a slight loss of details around the eye and frame 299 in (c) is slightly less bright than frame 299 in (d) where the key pictures are not encoded. The slight quality reduction in the encoded key pictures occurs due to the large number of predicted frames producing fewer bits and details. The slight differences can be noticed around the eyes and level of brightness within the vegetation areas.

5.13. *Q_p Constraint algorithm* for dynamic adaptation

H.264/AVC adapts the quantisation parameter within the video sequence frames and slice units. The Qp parameter automatically changes for rate control with a reference to a minimum value. This happens at rate-control coding to regulate the bit-stream rates. However, constant values for the minimum Qp do not achieve the set target rates at most and especially for higher current bit-rates. This situation does also affect the ELs in achieving an optimised coding performance due to their reference from BL. The BL uses this set minimum Qp and does not consider the effect of this on the higher layers (ELs) performance. The higher layers that reference from the BL will perform better if considered in this Qp setting. To achieve the target bit-rates and optimum performance, the current computed bits need to be considered prior to assigning the minimum Qp for BL control. Experiments are conducted which proved that the Qp value can be restricted to a certain value to support the achievement of the target rate and result for a better scalability control. Experimental results in Table V - XI, Table V - XII and Table V - XIII show the application of this technique for *t+q*, *t+s* and *t+s+q* scalability techniques. The experiments reveal that the Q_p restriction is required at higher rates > 32kbit/s. At target rates >= 300000bits, a maximum Q_p Delta of 8 is required to achieve the target rates.

The pseudo-code below defines an improved and dynamic algorithm for the selection and restriction of minimum Q_p value during the BL rate control.

If $R_t <= minRt$ {

$Q_p = Q_{pcurrent} + MinDeltaQ_p$

} *else*

{

$Q_p = Q_p + MaxDeltaQ_p$}

Where R_t is the computed target rate, $MaxDeltaQ_p$, $MinDeltaQ_p$ are the maximum (8) and minimum (2) quantisation levels respectively, minRt defines a target rate of 32000bits/sec and $Q_{p\text{-}current}$ is the last Q_p used.

In Table V - XIII, the higher ELs do not improve the bits reduction. This is because ELs below EL0 do not reference from the BL only. They reference from the immediate above ELs. Hence in a multi-layer bit-stream, this algorithm is implemented where the BL is needed or the next Enhancement Layer (EL0).

Table V - XI : Qp Constraint Algorithm Performance (H.264/*t*+*q*-qcif) foreman sequence

Constraint Q_p	Qp Constraint Algorithm		non-usage of Qp Constraint Algorithm
	Target bits (kbit/s)	**Actual Bits(kbit/s) / PSNR(dB) / Q_p used**	**Actual Bits(kbit/s) / PSNR(dB) / Q_p used**
(L0)	32	31.26 / 30.37 / 29	47.10 / 33.35 / 35.76
(L1)	32	69.52 / 33.72 / 36.67	51.65 / 33.36 / 36.36
(L0)	400	176.69 / 39.62 / 40.25	226.67 / 42.35 / 20.69
(L1)	400	182.67 / 39.61 / 40.44	230.86 / 42.35 / 26.95

Table V - XII : Qp Constraint Algorithm Performance (H.264/*t*+*s*) city sequence

Constraint Q_p	Qp Constraint Algorithm		non-usage of Qp Constraint Algorithm
	Target bits(kbit/s)	**Actual Bits(kbit/s) / PSNR(dB) / Q_p used**	**Actual Bits(kbit/s) / PSNR(dB) / Q_p used**
(L0)	318	46.13 / 33.21 / 21.42	173.80 / 40.63 / 22.09
(L1)	350	302.95 / 34.88 / 33.16	349.06 / 33.88 / 35.00
(L0)	350	46.13 / 35.39 / 20.84	186.84 / 41.00 / 21.52
(L1)	382	327.86 / 35.39 / 32.63	388.82 / 34.49 / 34.00
(L0)	414	46.13 / 33.21 / 19.83	212.13 / 41.66 /20.51
(L1)	446	354.64 / 35.81 / 31.71	413.16 /34.51 /34
(L0)	446	46.13 / 33.21 / 19.39	223.52/41.95/20.6
(L1)	478	363.35 / 35.94 / 31.29	424.93 / 34.54 /33.94

Table V - XIII : Qp Constraint Algorithm Performance (H.264/*t*+*s*+*q*) harbour sequence

Constraint Q_p	Qp Constraint Algorithm		non-usage of Qp Constraint Algorithm
	Target bits (kbit/s)	**Actual Bits(kbit/s) / PSNR(dB) / Q_p used**	**Actual Bits(kbit/s) / PSNR(dB) / Q_p used**
(L0)	320	252.40 / 35.14 / 35.14	279.15 / 37.02 / 24.76
(L1)	320	272.71 / 35.45 / 28.81	283.35 / 37.02 /29.00
(L2)	352	555.28 / 29.25 /38.01	492.00 / 29.63 / 37.40
(L3)	352	590.93 / 29.23 / 38.72	513.06 / 29.62 /37.83
(L0)	384	252.40 / 35.14 / 27.48	323.27 / 37.89 / 23.66
(L1)	384	287.88 / 35.79 / 27.72	327.26 / 37.88 / 29.00
(L2)	416	605.77 / 29.69 / 37.09	560.77 / 30.00 / 36.62
(L3)	416	642.03 / 29.67 / 37.81	582.26 / 29.97 / 37.05

5.14. Scalability Adaptation technique using Multi-encoder States

In section 2.3.4, methods of scalability adaptation with several coding elements and parameter design influences are introduced. One of the methods requires a very fast processor that is capable of encoding several encoders at the same time. In this section, the method is experimented with a set of three encoders embedded with three sets of parameters capabilities. Multiple encoders can be designed and run from the same processor. However, this will depend on the speed and memory capacity of the encoder and the intended applications. This method of multi-encoder processing will be efficient and effective for broadcasting applications since the video is transmitted from one source to multi-billion users of variable constrained resources. Streaming as well as a large group of videoconferencing and internet applications will also become more effective when using this technique. The result of the simulation reveals different scalability adaptation capability from each of the encoders. Figure 5 - 52 presents encoder state X1, X2 and X3. The definitions of the elements and their usage in the encoders are defined in Table V - XIV. The Header sizing is altered for the three sets of encoders, the layer quantisation, Hierarchical *GOP* structure, the quality layers and the coding sequence structure. These elements influence the encoder performance and efficiency and hence the encoders produce variable scalability outputs. These elements are all simulated.

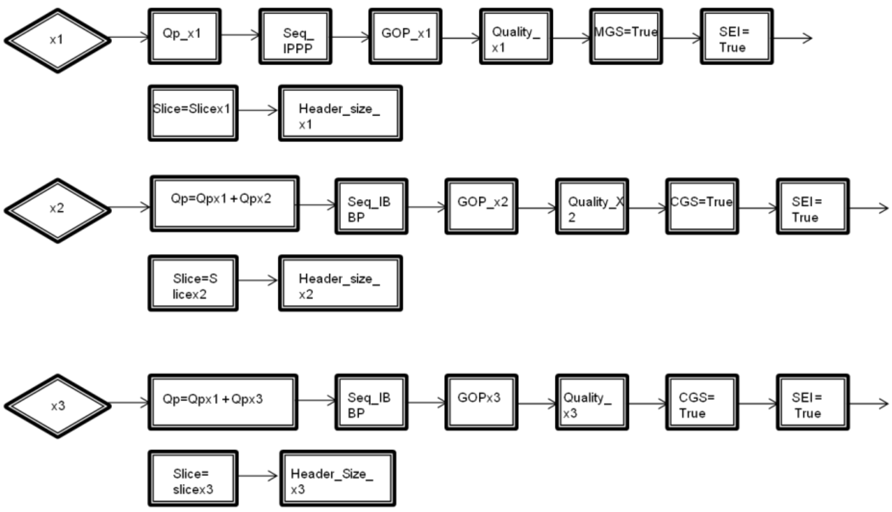


Figure 5 - 52 : Coder state X1, X2 and X3 running from same processor at the same time.

Table V - XIV: Used Notations and their Definitions for Figure 5 - 52

Notation	**Definition**
Qp	Quantisation parameter value (incremented by 2 for X2 and by 3 for X3)
Seq	Type of sequence structure used (X1 = *IPPP*, X2= *IBBP*, X3 = *IBBP*)
Quality	Number of quality layers used (X1 = 2, X2 = 3, X3 =3)
SEI	Scalable Enhancement Information(True for X1,X2 and X3)
Map	Type of MB map to slice group
Header_size	Reduced number of headers (X1 = 7%, X2 = 40%, X3 = 67%)
CGS	Coarse Grain Scalability (X2 and X3)
MGS	Medium Grain Scalability (X1)
GOP	Group of Pictures (X1=4, X2= 16 and X3=64)

5.15. Quality and scalability adaptation performance for X1, X2 and X3 encoder states.

The three encoder states X1, X2 and X3 are encoded from a single processor source transmitting their scalable outputs into the network at the same time. The main differences exhibited by the three states are defined in Table V - XIV. The header size reduction, the number of quality layers, the coding structure etc differentiates the three encoder states. The *MGS* and *CGS* scalability usage are experimented in section 5.10. The scalability performance differences are what is required to be achieved from the three states. This feature will enhance scalability adaptation performance in a multi-channel environment. Figure 5 - 53 shows the objective quality performance for the X1, X2 and X3 states from city sequence QCIF format. The X3 state shows better quality and scalability performance. For instance, 32dB quality value required a bit-rate value of 270kbit/s, 340kbit/s and >600kbit/s for X3, X2 and X1 states respectively. This shows a bit-reduction >70kbit/s for X3 state. The objective quality performance differences for the set of encoder states will support a wide range of adaptation to heterogeneous networks. For example, a decoder or client application decoding from state X2 or X1 can switch to state X3 when the bandwidth is limited or there is congestion. The technique employed for switching among bit-streams and within bit-stream is described in chapter seven.

Another scalability adaptation support is that, each of the states provides a different number of scalable and quality layers as shown in chapter six. Figure 5 - 55 presents frames and their corresponding bits characterisation for the three-encoder states.

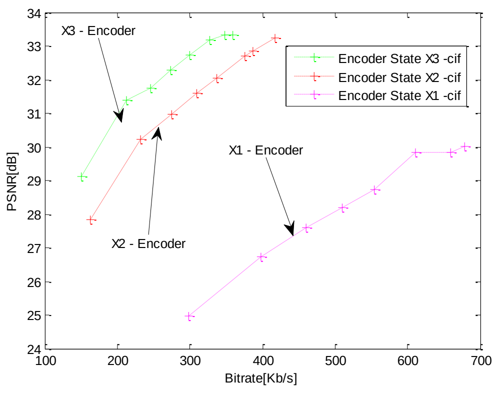


Figure 5 - 53 : Encoder States X1, X2 and X3 objective quality performance for cif – city sequence

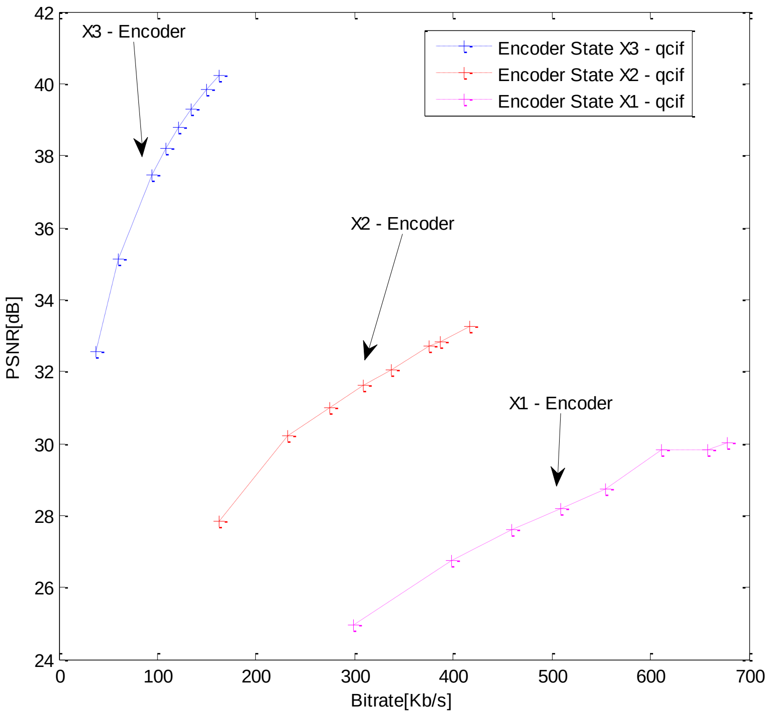


Figure 5 - 54 : Encoder States X1, X2 and X3 objective quality performance for qcif – city sequence

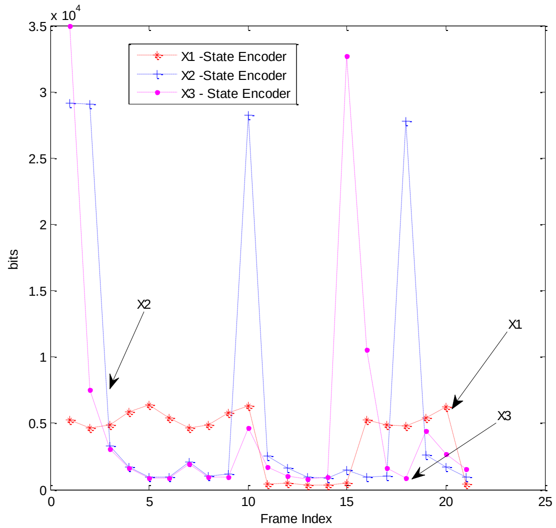


Figure 5 - 55 : Encoded output bits characterisation for X1, X2 and X3 encoder states scalability

5.16. Summary and Conclusion

In this chapter, coding parameters and elements are evaluated. These elements and parameters include coding sequence performance, compression algorithms, different types of slicing, different types of rate points in quality scalability, encoding and non-encoding of synchronisation pictures, layer quantisation step-size (step-up or step-down), different types of hierarchical structures etc. The evaluation includes objective, subjective and real time performance assessment. The experiments involved several video sequences of different formats and natural content. This has provided a clear revelation of the H.264/AVC Extension coding tools and parameters performance with regard to scalability and adaptation.

A number of methods are proposed in chapter two that could be employed for parameters adaptation support. A *multi-encoder technique* is experimented and this has revealed the possibility of adopting the technique for various applications demands and requirements. The implementation of Slice group flexibility in a video sequence has shown that, video stream protection against data loss and graceful degradation can be supported. The inputs of this chapter can be used for scalability adaptation depending on the intended application and channels requirements. In this chapter, a *Qp Constraint algorithm* is proposed and experimented. The algorithm supports scalability adaptation for BL and the followed layer EL0. The algorithm can be adopted where network is highly deteriorated since it improves the BL adaptation.

The evaluated parameters/elements and other adaptation algorithms in this chapter are integrated into the functionality of *Scalability Decision Making Algorithm* (*SADMA*) in chapter seven to provide better flexibility and adaptation.

References

[1] Segall, C.A. and Sullivan, G.J. (2007) "Spatial scalability within the H. 264/AVC scalable video coding extension", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1121-1135.

[2] Team, J.V. (2004) "of ISO/IEC MPEG and ITU-T VCEG, reference software to committee draft", *JVT-F100 JM10,* vol. 2.

[3] Ghanbari, M. (2003) "Standard Codecs: Image compression to advanced video coding", Institution of Electrical Engineers, .

[4] http://www.vcodex.com Accessed April 2012.

[5] Maarif, H.A.Q., Gunawan, T.S. and Priantoro, A.U. "Complexity Evaluation in Scalable Video Coding", *Advances in Multimedia-An International Journal (AMIJ) Volume (1),* , no. 1

[6] Schwarz, H., Marpe, D. and Wiegand, T. (2007) "Overview of the scalable video coding extension of the H. 264/AVC standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1103-1120.

[7] Wiegand, T., Sullivan, G.J., Bjontegaard, G. and Luthra, A. (2003) "Overview of the H. 264/AVC video coding standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, no. 7, pp. 560-576.

[8] Richardson, I.E.G. "H. 264 and MPEG-4 video compression. 2003", *John Wiley&Sons Ltd., New Jersey,* .

[9] Team, J.V. (2009) "H. 264/SVC reference software (JSVM 9.18) and manual", *CVS sever at garcon.ient.rwth-aachen.de,* .

[10] Ohm, J.R., Koenen, R. and Chiariglione, L. "SVC requirements specified by MPEG (ISO/IEC JTC1 SC29 WG11)", .

CHAPTER SIX

6. DESIGN AND EVALUATION OF NOVEL SCALABILITY TECHNIQUES

6.1. Introduction

The emergence of new applications and the rapid change of technological devices has demanded for an improved multimedia adaptation techniques and algorithms in the area of conversational and non-conversational applications. H.264/AVC is the current video codec standard developed by the ITU-T video Coding Experts Group (VCEG) and the ISO/IEC Moving Experts Group (MPEG). H.264 design has achieved an enhanced compression performance and supports a network user friendly coding techniques namely Temporal, Spatial and Quality (SNR) scalability techniques [1]. Temporal refers to the smoothness of picture, Spatial is the size or detail in the picture and Quality is the fidelity of the video signal. H.264/AVC was initially designed to be a codec with long term objectives (H.26L) as its tool are everyday experimented and redesigned for better efficiency [2]. Hence, there is a big room for further research and improvement since in most cases the gap in r-d-o performance to single layer is still significant [3]. H.265 (HEVC) is the recent codec which provides 2x efficiency of the existing H.264 at the same video quality [4].

In this chapter, combined scalability techniques are developed and evaluated. The techniques are designed to provide different adaptable performances. Hence, the techniques can be adopted for a specific set of applications and channel links. A transcoding algorithm (*SADMA*) is designed in chapter seven which utilises the developed techniques and other algorithms to provide a flexible adaptation for multi-channel links. Several experiments are conducted to evaluate the objective and subjective performances of the techniques. Also, comparative analysis is made to evaluate the techniques against non-scalable bit-stream and other state-of-the art established scalable techniques. The real time performance of the developed techniques is also established over the built heterogeneous network simulator discussed in chapter four.

6.2. Previous and Related Work

A number of related works to improve the H.264/AVC scalability are available in the literature. A multi-layer rate distortion optimization algorithm was proposed in [3]. The algorithm considered both base and enhancement layers where motion vectors, macro-block modes and quantized residues are well chosen to derive gain in the inter-layer prediction. Simulations revealed the efficiency of this algorithm. However, a very high computational cost is involved because a multi-pass process is required to be employed to achieve the multi-layer optimization. Also, fidelity scalability was not achieved by multiple layers but by a flexible sub-stream extracting technique where un-predicted fluctuations may occur [5].

New techniques for enhancing the efficiency of inter-layer prediction in spatial scalability were proposed in [5-7]. Basically, the author's main ideas are to improve the accuracy of the prediction by further exploring the correlation within SVC layers using Lagrange Multiplier (LM) Selection. Averages of 0.22dB and 0.35dB for four layer fidelity scalability and three layers spatial scalability are achieved. This algorithm incurred a tiny computational cost compared to the ones described earlier. With the proposed techniques in this chapter, *t+s+q*, *t+q* and *t+s* realised a gain value of 3.97dB, 3.97dB and 3.94dB for BL over q scalability technique respectively. Similarly with the EL, 3.11dB, 3.34dB and 2.65dB are achieved. The *t+s+q* technique was experimented with four layers while *t+q* and *t+s* achieved this profit using two layers.

In our proposed techniques, inter-layer prediction is enhanced by enabling all layers of a multi-layer stream dependent on the sub-layers. Hence the spatial, temporal redundancy predictions are improved and thereby provide better efficiency and scalability. A proper selection of *GOP* picture sizes further improved the prediction with a small computational complexity cost. The complexity will become negligible if the benefits of the techniques on channel performances are considered compared with those techniques incurring less cost in the coding processes. The use of the most efficient technique at a particular network condition supported by *SADMA* algorithm (chapter seven) will compensate a part or all of the complexity cost.

The developed techniques are combined scalabilities to allow for a better and more flexible scalability stream within a wide range of applications and variable network conditions. The developed techniques are temporal + spatial + quality (*t+s+q*), temporal + quality (*t+q*), temporal + spatial (*t+s*) and spatial + quality (*s+q*) scalability techniques. Several experiments are conducted with several sequences to cover different characteristics in the sequences. The developed techniques show up to 50% gain in scalability adaptation against single layer and non-combined techniques which made them highly adaptable in variable network environments. Several simulations have shown that the proposed techniques outperformed the default conventional techniques. Further experiments to determine the impact of additional layers have achieved some amount of gain in scalability adaptation as defined in section 2.7. To improve the scalability adaptation, an improved *t+s* technique is developed and evaluated (section 6.4.5).

The developed techniques proved to be providing an automatic adaptation while in video communication transition over variable networks. However, the nature of network channels is unpredictable and can change with time. To further improve the scalability and achieve better scalability adaptation against network time varying requirements, a *Dynamic Scalability Decision Making Algorithm* is invented and evaluated in chapter seven. The *Scalability Decision Making Algorithm* (*SADMA*) transcodes to a suitable adaptable stream based on network condition reports. The algorithm therefore prevents network congestion happening by frequently monitoring the network state. Network congestion can cause excessive delays, high bit-rate errors and packet loss ratio and thereby resulting in an un-guaranteed picture quality. Several experiments conducted in this chapter and chapter seven revealed that the algorithm can provide a robust and flexible scalability and adaptation within a heterogeneous network. The experimental set up, verification and validation are discussed in section 4.12-4.13.

In general, the major technological idea distinguishing SVC from single and simulcast coding is the inter-layer dependency. Interlayer dependency or prediction technique is employed to efficiently remove the spatial and temporal redundancies within a layered SVC bit-stream. Conclusively, research should be focused on inter-layer prediction methods and algorithms in order to enhance the coding efficiency of H.264/SVC Extension in comparison with single layer and simulcast coding.

6.3. Scalability Techniques

The developed techniques are a combination of various scalability methods. The layer dependency and multiple layers embedded in the combination distinguished them from the other techniques. The evaluated combinations are listed as follows:

- Temporal + Quality (*t*+*q*)
- Temporal + Spatial (*t*+*s*)
- Temporal + Spatial + Quality (*t*+*s*+*q*)
- Spatial + Quality (*s*+*q*)

Other techniques that are evaluated are a scenario where the layering is improved to provide a high level of scalability, and t + s technique where a spatial layer is further disseminated. The combination of the above defined techniques is derived within the available techniques where various experiments and evaluations revealed the performance of a particular technique. The performance of a technique will determine its suitability for particular applications or network requirements. Hence, a technique can be adopted to provide services for an application where suitable, for example video conference, storage, internet and other applications. The techniques are discussed in the following sections:

6.3.1. Temporal + Quality (*t*+*q*) technique

In this technique, a temporal multiplexer partition the original video sequence into two sets of image sequences. The sets of image sequences can be more than two depending on the intended applications for a robust error free bit-stream as shown in Figure 6 - 1. Each image sequence is fed to an individual encoder. This means that every one encoder will produce a quality level unlike the others. As shown on Figure 6 - 1, *t*+*q* scalability is achieved between the base and enhancement layers. The base layer which is the reference layer produced a minimum quality picture with Q_p greater than EL Q_p and a bit-rate less than EL-bit-rate. The embedded temporal scalability allows the generation of temporal scalable layers which are of variable quality levels and priority. The temporal prediction of each picture frame produces the temporal layers. The number of temporal layers depends on the number of pictures that are defined as a *GOP*. The smaller the *GOP* size, the less the layers and coding efficiency. This is because each *GOP* is defined by an initial intra frame which produces a larger number of bits than an inter-frame. The insertion of an initial I-frame at the start and end of a *GOP* unit is provided for synchronisation.

In Figure 6 - 2, the quality scalability is performed between the BL and the first EL1. The temporal scalability is implemented between the second EL2 and the locally decoded picture of the quality scalability coder. The encoding and decoding is as described in section 3.2-3.5. The input video (V) is partitioned into two sets of video sequences and these are fed to the individual encoders (V-0 and V-1).

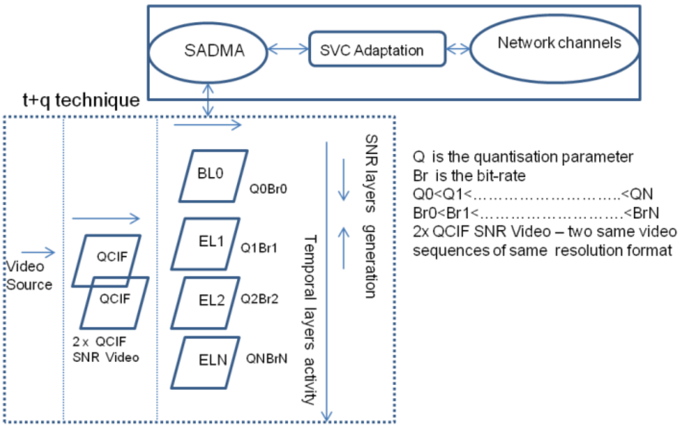


Figure 6 - 1 : *t+q* technique architecture and *SADMA* (Scalability Decision Making Algorithm)

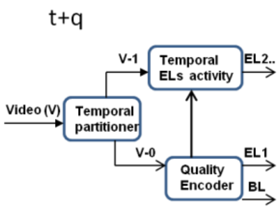


Figure 6 - 2 : *t+q* technique encoder

6.3.2. Temporal Scalable Layers Analysis (*t+q*)

An experiment is conducted with the city video sequence, one of the standard ITU sequences available for scalability research. A CIF resolution format is used with a maximum of 150 frames. 2 x CIF video sequences are used as illustrated in Figure 6 - 1 and a number of fourteen scalable layers are formed. Each one of the layers takes a unique frame and bit-rates. This feature allows the generation of multiple quality layers embedded in the video stream. Also, different quantisation levels are associated with the quality levels of the layers. For example, L0 and L1 are encoded with Q_p values 31.51 and 29.90 respectively. This enabled the bit-stream layers to generate variable coded bits and QoS levels.

The video stream embeds a minimum of 0.4688Hz and a maximum of 30.00Hz frame rates. The variable frame rates were encoded with variable bit-rates having a minimum of 102.70kbit/s and a maximum of 611.70kbit/s discrete levels. The details of the results are summarised in Table VI - I. For a similar experiment with 2xQCIF resolution, a minimum of 29.0kbit/s and a maximum of 109kbit/s bit-rates are produced.

Table VI - I: *t*+*q* technique: Temporal Scalability Layers with their associated quality EL- CITY CIF

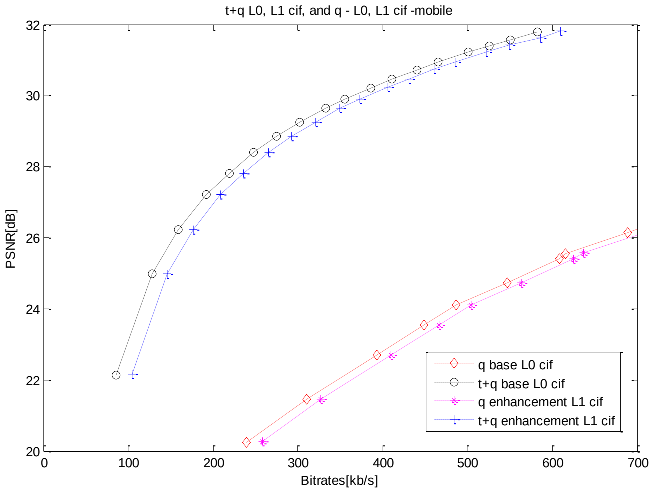
Layer	Resolution	Frame rate (Hz)	bit-rates (kbit/s)	Dependency_id	Temporal_id	Quality_id
0	352x288	0.4688	102.70	0	0	0
1	352x288	0.9375	160.20	0	1	0
2	352x288	1.8750	214.20	0	2	0
3	352x288	3.7500	272.20	0	3	0
4	352x288	7.5000	314.40	0	4	0
5	352x288	15.0000	364.50	0	5	0
6	352x288	30.0000	423.80	0	6	0
7	352x288	0.4688	157.80	0	0	1
8	352x288	0.9375	241.60	0	1	1
9	352x288	1.8750	315.30	0	2	1
10	352x288	3.7500	394.70	0	3	1
11	352x288	7.5000	454.40	0	4	1
12	352x288	15.0000	527.10	0	5	1
13	352x288	30.0000	611.70	0	6	1

6.3.3. Objective Evaluation and Analysis (*t*+*q*)

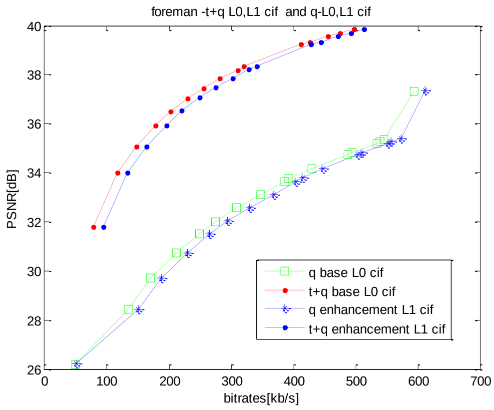
In order to objectively evaluate the performance of *t*+*q* technique, a number of ITU standard sequences are used. Experiments are conducted with mobile, foreman, city and harbour video sequences. More than one sequence is used to assess the performance of the technique from dissimilar video motion and natural content. The experiment is set-up as illustrated in Figure 6 - 1 and the results are presented in Figure 6 -3 (a), (b), (c) and (d).

Figure 6 -3 (a) shows that *t*+*q*-L0 has a better r.d.o performance over the q-L0 technique which does not employ temporal scalability and hence is coded with less efficient predictions. To obtain a γ-luminance value of 22.1dB, 25dB and 28dB, a bit-rate of 99.8kbit/s, 140kbit/s and 210kbit/s is respectively required for *t*+*q*-L0. For the q-L0 technique, 89kbit/s, 130kbit/s and 200kbit/s bit-rates are correspondingly required. This gives a reduction of 10kbit/s for *t*+*q*-L0 over the q-L0 technique.

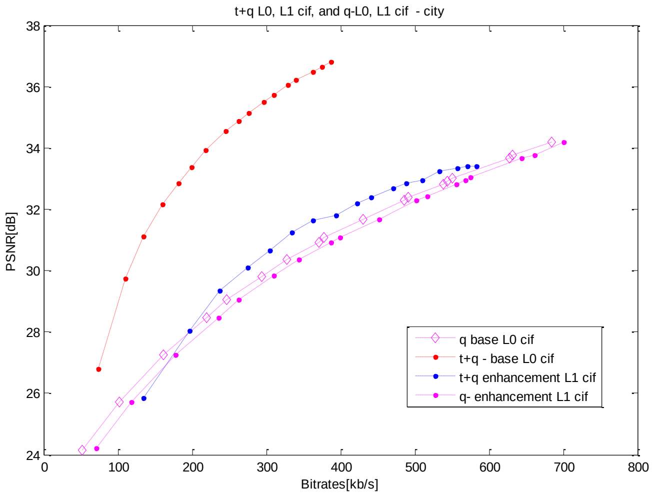
On the side of scalability adaptation layers support, q-technique generates four (CIF and QCIF) scalable layers and *t+q* technique provides twenty six (CIF and QCIF) scalable layers. Every one layer is designed to include variable bit-rates, frame-rates and a hierarchy of QoS levels.



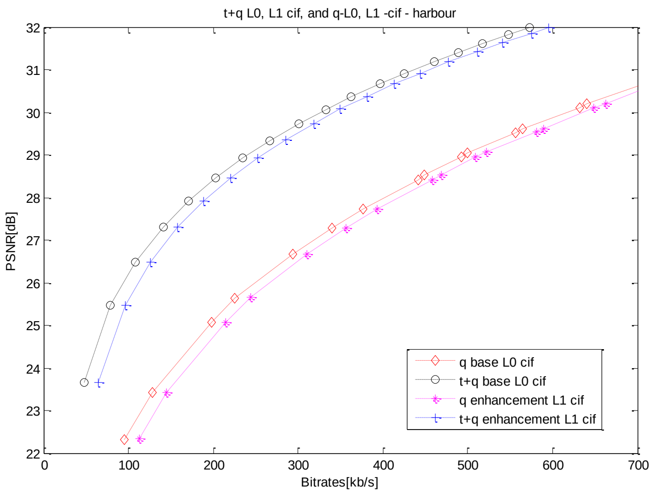
a.



b.



c.



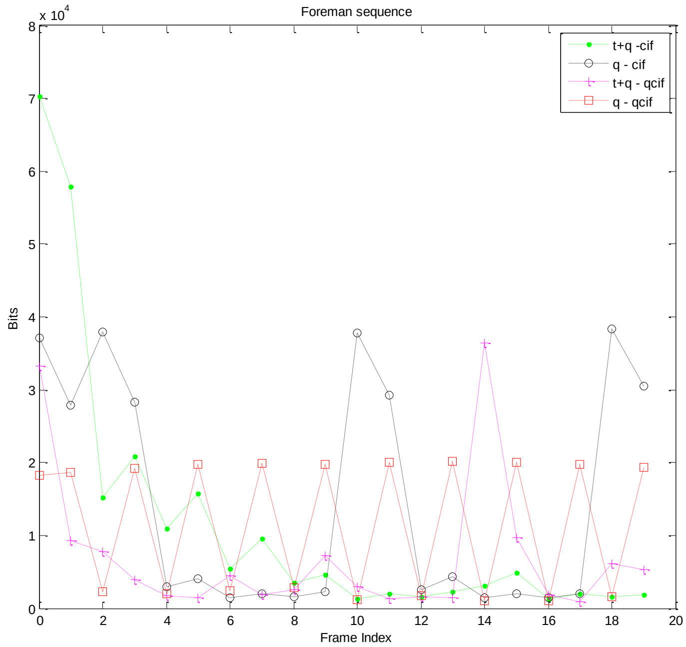
d.

Figure 6 -3 : *t+q* technique: Objective Performance for (a) mobile sequence (b) foreman sequence (c) city sequence (d) harbour sequence

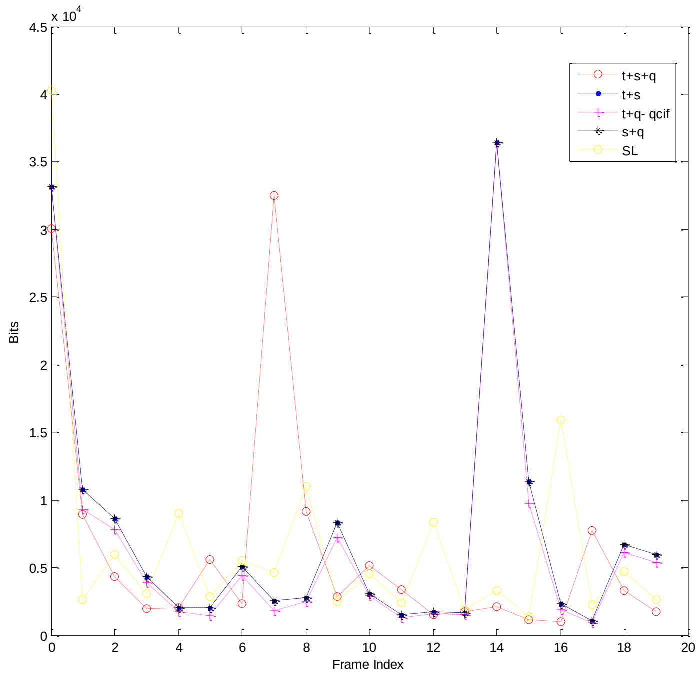
Figure 6 - 4 presents coding bits characterisation for *t+q*, *t+s*, *t+s+q*, *s+q*, q and *SL* techniques. Smaller number and smoother bits characteristics are shown with *t+q* technique over q, *SL*, *t+s+q* and *t+s* techniques. This is due to the fact that two equal video sequences are used in *t+q* technique and therefore spatial and temporal redundancies are efficiently predicted and removed.

Figure 6 - 5 shows that *t+q*-L0 achieves a reduction of 5000 bits per second over *SL* objective quality performance. This implies that *t+q*-L0 is less expensive than a non-scalable layer and can be preferred for video adaptation.

From the results and experimental analysis discussed, it can be concluded that *t+q* technique



a.



b.

Figure 6 - 4 : *t+q*, q, *t+s*, *s+q*, *t+s+q* and *SL* techniques at 64kbit/s bits characterisation for foreman sequence.

can be preferred when channel links are highly deteriorated and there is scarcity of bandwidth with the video service clients and network terminals. This thought is further evaluated and established from the conducted real time simulations over heterogeneous networks. These simulations are presented in section 6.7.

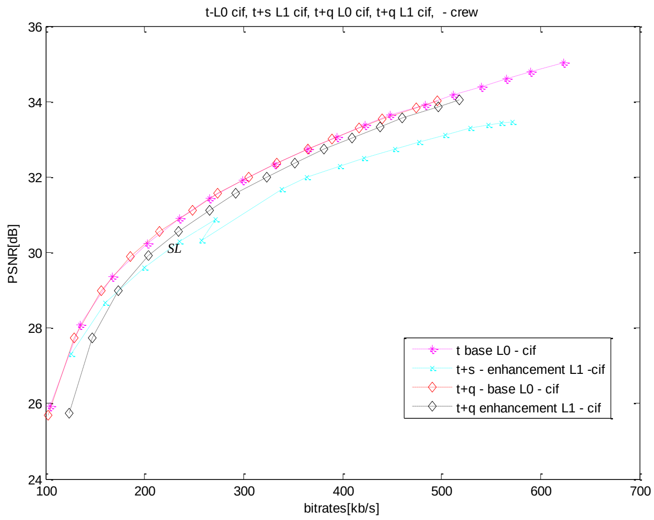


Figure 6 - 5 : *t+q* and *SL* performance- crew sequence

6.3.4. Subjective Evaluation and Analysis (*t+q*)

To subjectively evaluate and assess the developed techniques, Double Stimulus Continuous Quality Scale (DSCQS) is employed as recommended by ITU-R BT.500 [8]. The DSCQS subjective assessment method is widely used to standardize the objective quality assessment method. The main idea of DSCQS method is described in chapter two. A total number of fourteen participants were involved in the conducted evaluation exercise. The individual participants include experts in video coding and assessments and non-experts in the field. A 42 inch and 21 inch Standard monitor types (INTEL® Q45/Q43 EXPRESS CHIPSET) were used throughout the exercise. The viewing distance between the participants and the monitor was set to six times the height of the video in view as recommended by ITU [8, 9].

A score assessment sheet is provided to each participant and the video sequences are run for more than one time and as many times as the participant requested. Some participants requested more than two times view of the videos before recording their perceived scores. In each of the technique, a left and right view video is run at a time. One of the videos is the reference video in spatial domain as obtained from ITU. This video is used as a reference to obtain the second video using the developed technique and algorithm. No participant was informed which of the two videos represents a reference or an impaired sequence. The consequence of revealing this is a biased or wrong score in one of the sequence and hence can result in invalid results and conclusions of the work.

Each assessment is graded on a five point scale (Bad, Poor, Fair, Good and Excellent) as in chapter two. The score is assigned to both the original and reconstructed sequences from the developed technique. The final score SR for the subjective quality of the reconstructed frame R is obtained by subtracting the original frame score O from the reconstructed frame score as in chapter two (equation 2-1). The lower the value for SR the more similar the technique output picture to the original video. The results of the evaluation exercise for *t+q* technique are provided in Figure 6 - 12. The sample participants' questionnaire is provided in Appendix-A4.

Figure 6 - 6 and Figure 6 - 8 represent some of the videos which were used for subjective assessment exercise. The impaired videos which are the output of the developed techniques revealed almost the same quality to the quality of the reference videos in the pixel domain. A slight brightness difference within the two videos is due to the absence of de-blocking filter applied to improve the quality of the distorted edges of the impaired videos. Other reasons include adoption of a small number of intra-frames to achieve better adaptation. Scalability adaptation is one main aim of this research work and thus a small number of un-predicted frames are preferred.

a. *SL*, 32kbit/s, QCIF b. *t+q*, 32kbit/s, QCIF

c. *SL*, 64kbit/s, QCIF d. *t+q*, 64Kb/, QCIF

e. *t+q*, 32kbit/s, QCIF f. *SL*, 32kbit/s, QCIF

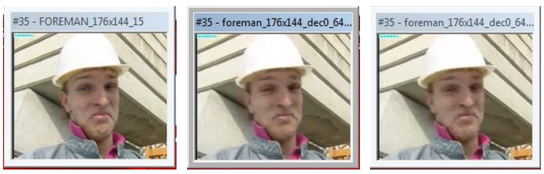
Figure 6 - 6 : *t+q* - Subjective quality evaluation (QCIF)

The use of inter-frames might cause bits reduction in the entire picture and this might result in the loss of brightness in some regions of the picture.

Figure 6 - 8 shows frame 280 of the mobile sequence for *SL* and the *t+q* technique picture output. The *SL* and *t+q* are run at 320kbit/s bit-rate. A similar and good quality picture is obtained from the two techniques. Figure 6 - 7 provides *t+q* technique products at 64kbit/s.

At this bit-rate, there is a slight difference on the brightness and this is due to bits reduction to achieve better adaptive scalability. The bits reduction is realised by reducing the number of intra-frames in the video sequence so that a large *GOP* size is obtained providing several temporal adaptation layers.

a. Pixel domain, frame 141 b. *SL*, 64kbit/s, Frame 141 c. *s+q*, 64kbit/s, Frame 141



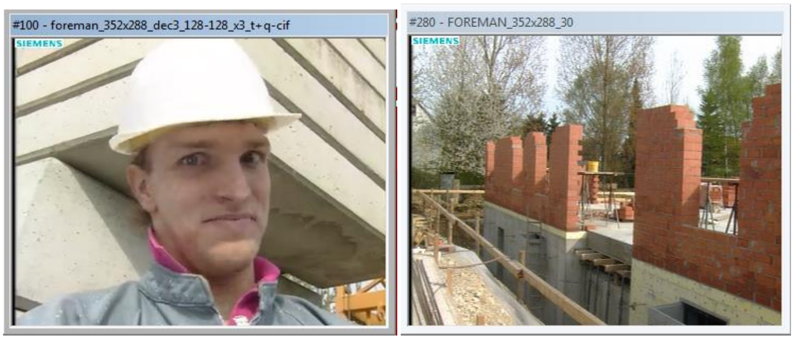
a. Pixel domain, Frame 35 b. *SL*, 64kbit/s, Frame 35 c. *s+q*, 64kbit/s, Frame 35

Figure 6 - 7 : *t+q*- subjective quality evaluation (QCIF)

a. *SL*, 320kbit/s b. *t+q*, 320kbit/s

Figure 6 - 8 : Subjective Performance, CIF format, 300frames, 320kbit/s (a) *SL* (b) *t+q*

a. Pixel domain, frame 100, CIF b. *SL*, 128kbit/s, frame 100, CIF



c. *t+q*, 128kbit/s, frame 100, CIF d. pixel domain, frame 280, CIF

d. *SL*, 128kbit/s, frame 280, CIF e. *t+q*, 128kbit/s, Frame 280, CIF

Figure 6 - 9 : *t+q* - subjective quality evaluation

The subjective performance of the technique is assessed to be of good quality even at low bit-rates of 64kbit/s. A de-blocking filter is required to smooth and remove distortion effects at bit-rates lower than 64kbit/s. This filter can be adopted for an application or can be used dynamically during video services.

A comparison is made with non-scalable stream and the BL output pictures quality. The *Dynamic Scalability Decision Making Algorithm* (chapter seven) might decide as a result of predicted channels deterioration concluded from the network report to switch video services using BL bit-stream only. The comparison made in Figure 6 - 6 and Figure 6 - 10 for bit-rates of BL, 32kbit/s shows that, a good quality picture for QCIF format and a fairly acceptable quality for CIF resolution format are guaranteed.

However, a guaranteed quality is achieved for bit-rates greater than 64kbit/s with BL. Figure 6 - 10 shows subjective quality performance for *t+q* and *SL* at 64kbit/s and 32kbit/s respectively. Figure 6 - 10 (e) depicts the subjective quality performance for q-technique which revealed a distorted video compared to *t+q* and *SL*. This is due to the associated encoded key pictures in the coding design. The technique contains a large number of Key pictures (I-frames) which are encoded and this resulted in the removal of a large amount of bits and hence the distorted product. Detailed discussions are shown in chapter five for encoding the key pictures.

a. *t+q*, 32kbit/s

b. *SL*, 32kbit/s

c. *t+q*, 64kbit/s

d. *SL*, 64kbit/s

e. q, 64kbit/s

Figure 6 - 10 : Subjective Performance, CIF format, frame 100, 150 frames (a) *t+q*, 32kbit/s (b) *SL*, 32kbit/s (c) *t+q*, 64kbit/s (d) *SL*, 64kbit/s (e) q , 64kbit/s

Figure 6 - 12 represents the visual perception score opinions obtained from the subjective assessment exercise which included both experts and non-experts in the field. The assessment was made from video images of more than 64kbit/s and the participants viewed that the visual perception between the *t+q* technique and the original video is the same. This gave S─R scores to be all zero for both QCIF and CIF formats. However, some slight QoS levels are observed from sequences run at bit-rates less than 64kbit/s in CIF format. This is shown in Figure 6 - 11. In the case of QCIF resolution format, both the expert and non-expert in the field have recommended that both the original and *t+q* product pictures are the same in visual quality at low and high bit-rates. This is shown in Figure 6 - 6 subjective performance frames.

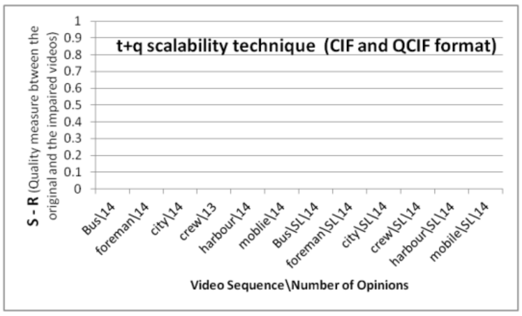


Figure 6 - 12 : *t+q* pair opinion scores for original, *t+q* and *SL* videos

6.3.5. *t+q* technique comparison with other established scalability

To evaluate the quality and performance differences of *t+q* technique and a state of the art scalable codec MPEG-4 [3], several experiments are conducted with a number of different frame rates. Foreman sequence, a standard ITU sequence, is used in the experiments. Experimental results for 3/6f/s, 75/10f/s and 5/10f/s frame rates are presented in Table VI - II.

From Table VI - II, coding the two techniques at 3/6 f/s, *t+q* technique gained a reduction of 300bits, 189000bits and 21000bits per second over *SL*, BL and EL respectively. Similarly, with the *t+q* technique profits gain of 14dB, 5dB and 9dB for *SL*, BL and EL correspondingly are realised. These achieved gains are possible due to adaptive and variable Qp used for *t+q* technique while MPEG-4 used a constant Qp at all coding bits and frame rates.

At 75/10 f/s coding frame rate, 12000bits, 3700bits and 19800bits per second reductions are achieved with *t+q* technique against MPEG-4 scalability for *SL*, BL and EL respectively. Also 5dB, 5.12dB and 4.76dB gains are realised with *t+q* technique. Bits reduction of 27300bits, 19070bits and 38000bits is achieved for *SL*, BL and EL when coding with 5/10 f/s rates. Similarly, at this frame rate 19.41dB, 9.2dB and 8.29dB gain is achieved for *SL*, BL and EL respectively.

Table VI - II : *t+q* technique and MPEG-4 quality and adaptation performance

Frame rate (f/s)	MPEG-4				*t+q*/H.264 technique			
	Bitrates(kbit/s)/Luminance(γ)dB				Bitrates(kbit/s)/Luminance(γ)dB			Qp
	SL	BL	EL	Qp	*SL*	BL	EL	*SL* / BL/ EL
3/6	18/30.96	28/30.89	33/30.92	13	15.05/44.93	10.9/35.38	11.78/40.01	16.61 / 16.05/ 23.84
7.5	47/31.01	32/30.97	57/31.04	13	35.39/35.92	35.71/35.78	38.76/35.80	31.51/ 31.08/ 31.73
5/10	37.5/31.0	25/30.94	44/31.00	13	10.8/49.41	6.07/39.29	6.77/39.29	11.11/10.55/ 20.30

6.3.6. Performance Evaluation of an improved Layering technique

A scalable bit-stream is implemented from a number of layers which is a minimum of BL and an EL which supports the enhancement of the video quality. In this section, experiments are conducted to assess the level of an increased performance when the layering structure is enhanced with additional ELs. An experiment is conducted with standard ITU sequence city using QCIF and CIF formats. The experiment is aimed to evaluate the level of complexity, flexibility and scalability level of different layering structures. Hence applications can implement an optimised layering structure based on the available resources and requirements.

Table VI - III shows scalability levels for different hierarchy of used quality layers. The video quality tends to be the same at a hierarchy of four encoded quality layers up to eight quality layers with PSNR value equal 37.92. This meant that, for low bandwidth required applications or network environment, the bit-stream can be limited with four encoded quality layers. The main advantage of the additional layers is that, some of their temporal rate points generate entirely different bit-rates and therefore is an additional adaptation support.

Table VI - III : Scalability levels of different numbers of encoded quality layers- city QCIF

Quality Layers	Produced Scalable layers	Avg. bitrates(kbit/s) / PSNR(dB)	Bit-stream-size(KB)	Encoding time / BL-decoding time(sec)
1	7	289.14 / 32.52	64	17.07 / 0.09
2	14	93.65 / 36.11	58	31.84 / 0.49
3	21	76.54 / 35.46	47	19.76 / 0.7
4	28	122.17 / 37.92	75	20.97 / 0.83
5	35	126.91 / 37.92	78	22 / 0.99
6	42	131.77 / 37.92	81	23.55 / 1.1
7	49	136.5 / 37.92	84	24.99 / 1.87
8	56	139.92 / 37.92	86	26.27 / 1.37

Figure 6 - 13 and Figure 6 - 14 show the coded bits characterisation for the usage of a different number of quality layers in the scalable bit-stream for QCIF and CIF formats respectively. The coding bits efficiency is dependent on the prediction type used within the layer and also among the layers. For instance, the use of one quality layer does not produce better efficiency due to the lack of inter-layer prediction. In the case of quality layers above two, the bits show less expensive characteristics. The discrete bit-rates within the temporal layers are low for the bit-stream with high number of coded quality layers and therefore lower rates are supported in the stream of these layers. The minimum and maximum bit and frame rates of the temporal layers are shown in Table VI - IV.

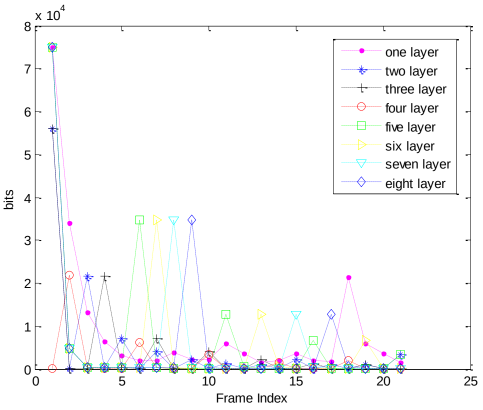


Figure 6 - 13 : Coding bits characterisation for variable quality layers-city sequence -qcif

Table VI - IV : Quality Layers minimum and maximum bit and frame rates

Quality Layers	Minimum bit-rate(kbit/s)	Maximum bit-rate(kbit/s)	Minimum frame rate (Hz)	Maximum frame rate(Hz)
1	34.40	205.60	0.4688	30
2	64.20	92.20	1.8750	15
3	25.50	149.30	0.4688	30
4	34.50	240.10	0.4688	30
5	34.50	249.10	0.4688	30
6	34.60	258.40	0.4688	30
7	34.60	267.40	0.4688	30
8	34.60	273.40	0.4688	30

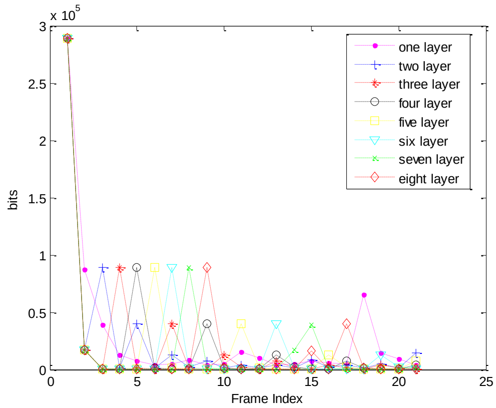


Figure 6 - 14: Coding bits characterisation for variable quality layers-city sequence, cif

The same experiment is repeated for CIF resolution format and the results revealed similar characteristics as shown in Table VI - V. The number of quality layers used impacted also on the decoding complexity. It will cost a decoder decoding from a bit-stream coded with 4–layers and 5–layers quality layers 6.04 secs and 7.52 secs respectively. This fact is according to the experimental results in Table VI - V. Therefore, although the utilisation of several quality layers supports graceful degradation of the stream quality, it can also increase decoding complexity. This happened due to more overhead and several temporal layers in the bit-stream whereby its size increases with the increase in the number of used layers. The number of produced layers is equal to the *produced layers x number of quality layers*. For instance, the coded quality layers 7 generates 7x49 temporal scalable layers. This has enhanced the scalability and adaptation of the bit-stream.

Table VI - V : Scalability levels of different number of encoded quality layers- city -CIF

Quality Layers	Produced Scalable layers	Avg. bitrates(kbit/s) / PSNR(dB)	Bit-stream-size(KB)	Encoding time / BL-decoding time(sec)
1	7	263.01 / 38.26	321	91.9 / 2.06
2	14	294.21 / 38.56	359	269.79 / 3.49
3	21	303.25 / 38.56	371	557.8 / 4.85
4	28	313.74 / 38.56	383	658 / 6.04
5	35	324.24 / 38.56	396	1043.26 / 7.52
6	42	357.19 / 38.56	409	1224 / 8.66
7	49	407.92 / 38.57	440	1441.22 / 10.12
8	56	444.65 / 38.57	453	1960.93 / 11.34

Figure 6 - 15 shows the subjective quality performance of a different number of quality layers (frame1). The experiment utilised the standard city sequence. The Quality layers (QLs) are

experimented from one layer up to eight layers. The videos are run using *YUVViewer* and all show a good quality picture. The trade-off in the use of more layers remains a better scalability adaptation and guaranteed video quality or a reduced complexity. Hence, this can be decided based on applications and requirements. Figure 6 - 16 shows frame 19 of the QLs.

QL = 1 QL = 2 QL=3

QL=4 QL=5 QL=6

QL=7 QL=8

Figure 6 - 15 : BL, Frame-1, same target rate -Subjective Quality Performance for different number of Quality Layers (QL),

QL = 1 QL = 2 QL=3

QL=4 QL=5 QL=6

QL=7 QL=8

Figure 6 - 16 : BL, Frame-19, same target rate -Subjective Quality Performance for different number of QLs.

Let us define a temporal layer with identifier T. T starts from 0 and is increased by 1. Then for each number k, the bit-stream obtained by removing all access units of all temporal layers, each with an identifier T greater than k, a valid bit-stream is formed for a given decoder. The allocated *GOP* size can be represented by identifier G. G starts from 2 to any number multiple of 16 and less than 64. The identifier G defines the number of pictures which constitutes the number of access points in a given interval of video sequence synchronisation points. Then, the relationship between G and T can be defined as follows:

$$G \propto T \qquad 6\text{-}1$$

$$G \propto kT \qquad 6\text{-}2$$

Where k is a constant referring to a natural number at each access unit. The results obtained in section 5.6.7 present an experimental proof of equation 6-2. However, the high *GOP* sizes increase the encoding time with better efficiencies and scalability adaptation. Therefore, *GOP* sizes are allocated on the basis of application requirements. The Heterogeneous environment will require a large *GOP* size for better adaptation and therefore a flexible quality scalability should be provided in the bit-stream to support several quality demands.

6.4. Temporal +Spatial technique (*t+s*)

Spatial scalability enables video adaptability for display characteristics of the client device without any re-encoding requirements. The temporal scalability allows decoding of moving pictures with multiple frame rates up to 60Hz [10]. In this technique, both spatial and temporal scalabilities are enabled in the bit-stream to increase the adaptation of the spatial layers.

A minimum of two dissimilar spatial resolution formats are required to enable *t+s* technique. Hence, the source video must be up-sampled or down-sampled to provide second spatial picture scalability. The inclusion of temporal activity in the encoding operation allows the production of several spatial picture enhancement layers for each of the two required video resolution formats. The number of layers generated is limited from the definition of *GOP* as discussed in section 6.3.1. Figure 6 -17 described the structure and functional components of *t+s* technique.

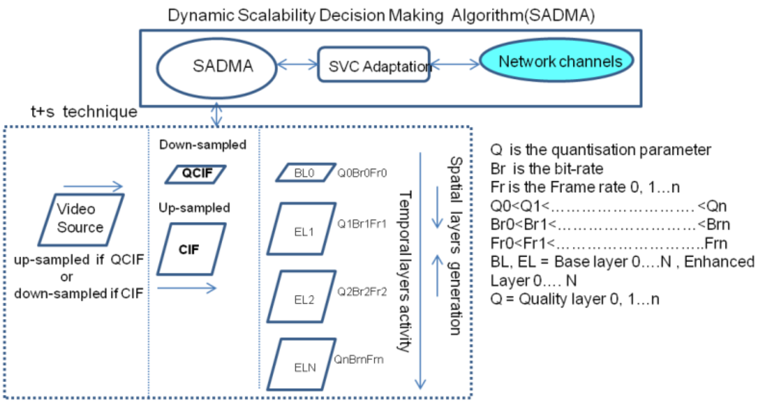


Figure 6 -17 : *t+s* technique architecture and *SADMA* (Scalability Decision Making Algorithm)

In Figure 6 - 18, the original input video is partitioned into two image sequences V-0 and V-1. The image sequence is fed to the scalable spatial encoder where it is compressed version V-0 is the input to the BL encoder. The spatial encoder then produces two bit-streams for the BL and EL1. The EL2 image sequence is fed to the temporal enhancement encoder to produce the third EL2 bit-stream. The temporal Enhancement encoder uses the locally decoded frames of the spatial encoder as prediction frames. The spatial scalability in this encoder is formed from then BL and EL1 while the temporal scalability is realised between EL2 and from both the BL and EL1.

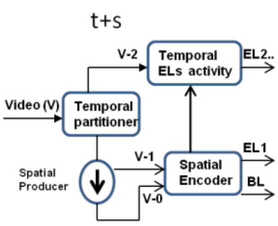


Figure 6 - 18 : *t+s* technique encoder

6.4.1. Temporal Scalable Layers Analysis (*t+s*)

An experiment is conducted with city video sequence, one of the standard ITU sequences captured to support research on H.264/AVC Extension scalability. A CIF resolution format is used with a maximum of 150 frames. 1 x CIF and 1 x QCIF video sequences are used as described in Figure 6 -17 and a number of thirteen scalable layers are generated. Seven layers are for CIF resolution scalability and six layers for QCIF resolution scalability. All of the layers take a unique frame and bit-rate. This feature allows the generation of multiple quality layers embedded in the video stream. Also, altered quantisation levels for every layer and bit-rates are associated with the quality levels of the layers. For example, L0 and L1 are encoded with Q_p values 35.87 and 43.12 to generate a bit-rate of 38.25kbit/s and 115.73kbit/s respectively.

The video stream embeds a minimum of 0.4688Hz and a maximum of 30.00Hz frame rates. The variable frame rates were encoded with variable bit-rates having a minimum of 47.60kbit/s and a maximum of 474.20kbit/s discrete levels. The details of the results are summarised in Table VI - VI.

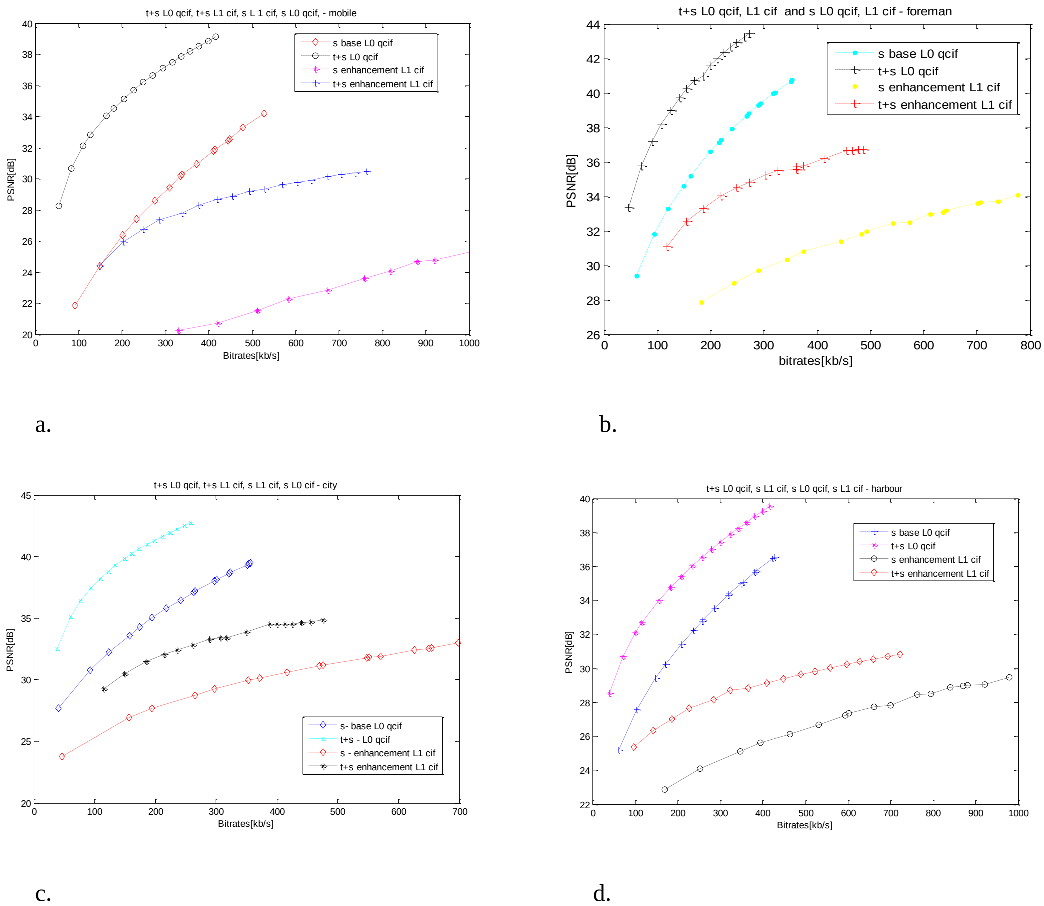
Table VI - VI : *t+s* technique: Temporal Scalability Layers with their associated quality EL- CITY sequence

Layer	Resolution	Frame rate (Hz)	Bitrates kb/s	Dependency_id	Temporal_id	Quality_id
0	176x144	0.4688	47.60	0	0	0
1	176x144	0.9375	77.10	0	1	0
2	176x144	1.8750	115.90	0	2	0
3	176x144	3.7500	159.70	0	3	0
4	176x144	7.5000	203.60	0	4	0
5	176x144	15.0000	257.20	0	5	0
6	352x288	0.4688	107.40	1	0	0
7	352x288	0.9375	163.60	1	1	0
8	352x288	1.8750	225.90	1	2	0
9	352x288	3.7500	294.20	1	3	0
10	352x288	7.5000	357.40	1	4	0
11	352x288	15.0000	437.10	1	5	0
12	352x288	30.0000	474.20	1	6	0

6.4.2. Objective Evaluation and Analysis (*t+s*)

In order to objectively evaluate the performance of the *t+s* technique, a number of ITU standard sequences are used. Experiments are conducted with mobile, foreman, city and harbour video sequences. More than one sequence is used to assess difference performance of the technique from dissimilar video natural content. The experiment is set-up as illustrated in Figure 6 -17 and the results are presented in Figure 6 - 19 (a), (b), (c) and (d).

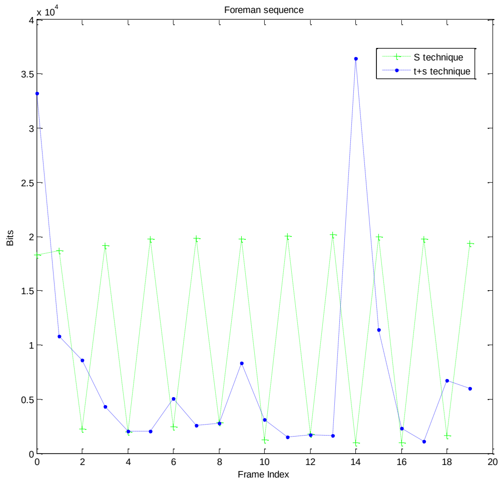
Experimental results for *t+s* and s techniques both with L0 and L1 are presented in Figure 6 - 19. From Figure 6 - 19 (a), it is shown that to generate the video quality of 32dB, 28dB and 34dB, 100kbit/s, 50kbit/s and 150kbit/s bandwidths are respectively required for *t+s* technique. To achieve similar quality pictures with s-technique, 400kbit/s, 300kbit/s and 500kbit/s of bandwidths are required. This gives an approximate bits reduction of 200000 to 400000 per second for *t+s* against s technique. These profitable characteristics of *t+s* technique over s-technique are achieved due to the efficient prediction employed in *t+s* by inclusion of temporal scalability. Temporal redundancies are removed and therefore better compressed coded bits are obtained. Similar results are shown in the other sequences in Figure 6 - 19 (b), (c) and (d) favouring the use of *t+s* for better scalability adaptation.



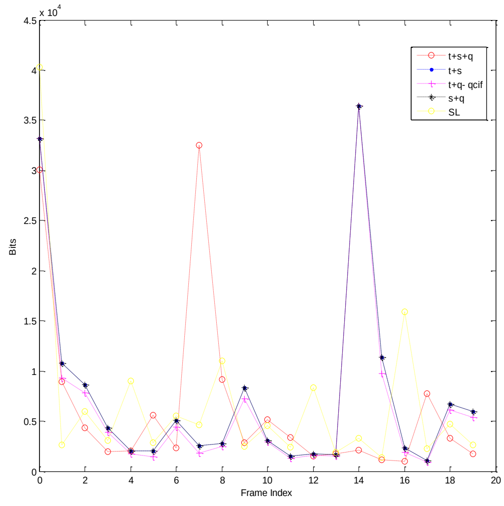
a. b. c. d.

Figure 6 - 19 : *t+s* and s techniques: Objective Quality Performance for (a) mobile (b) foreman (c) city and (d) harbour sequences

Figure 6 - 20 (a) and (b) presents *t+s*, *SL*, *t+s+q* and *s+q* coding bits characterisation. *t+s* technique shows smoother and smaller number of coding bits than *SL* and *s+q* technique.



a.



b.

Figure 6 - 20 : *t+s*, *t+s+q*, *t+q* and *SL* techniques all at 64kbit/s bits characterisation for foreman sequence

The bits coded for *t+q* and *t+s+q* show less expensive cost than *t+s*. Figure 6 - 21 presented experiments for *t+s*, *t+q* and *SL* (t base L0-cif) in CIF format resolution. *t+q*-cif and *SL*-cif made-up better quality and adaptation performance than *t+s*-cif.

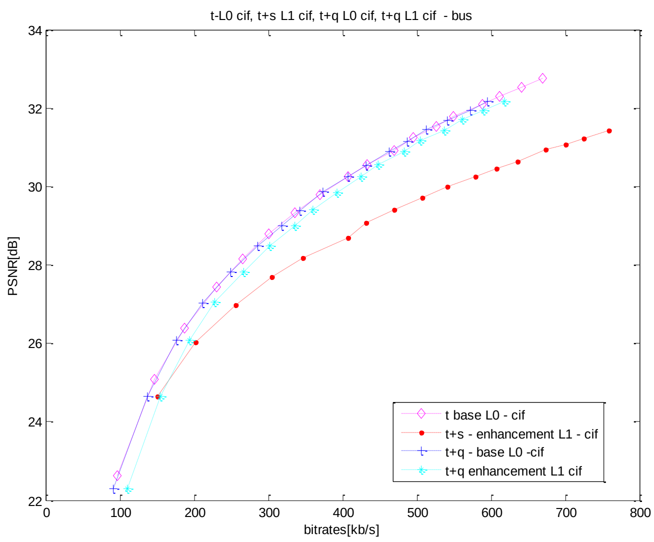


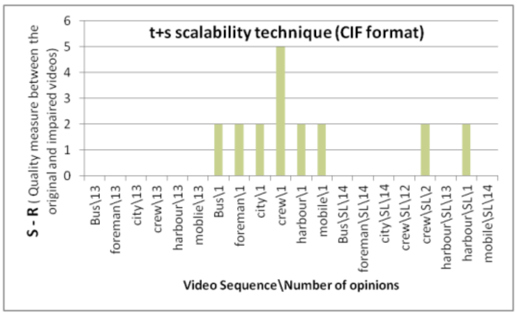
Figure 6 - 21 : *t+s* technique: Objective Quality Performance for *SL* and *t+s* technique for bus sequence

6.4.3. Subjective Evaluation and Analysis (*t+s*)

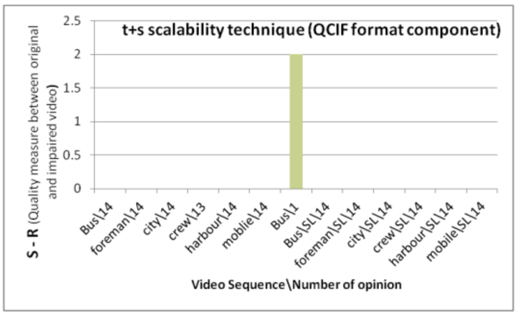
The method and procedures described for subjective evaluation of *t+q* in section 6.3.4 are obeyed in *t+s* subjective evaluation and assessment. Majority of the participants numbered thirteen out of the total fourteen participants perceived *t+s* –cif format with excellent and very good picture quality. The results of the evaluation exercise are presented on Figure 6 - 22 (a) and (b).

On the QCIF format of the *t+s* technique, approximately hundred percent of the participants recommended that, there is no difference between the *t+s* output picture and the original video in the pixel domain in their visual perception quality. This recommendation is a given view from many of the experts and non-experts in the field. The results of the assessment exercise for QCIF are presented in Figure 6 - 22 (b).

On CIF format evaluation, a higher number of participants recommended that most of the sequences are similar in quality and visual perception. However, a small number of participants observed a slight difference in brightness with the original sequence being brighter. This is due to the inter-prediction coding employed which usually reduces picture details because a large amount of bits might be removed. In this research case, this is purposely admitted to provide the maximum possible adaptation level for heterogeneous networks in situations where network channels are highly deteriorated and crammed.



a.



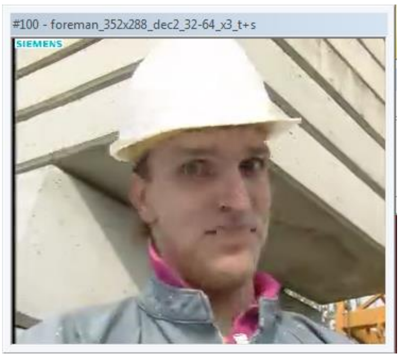
b.

Figure 6 - 22 : *t+s*, Subjective performance evaluation and pair opinion views among the original, impaired and *SL* videos.

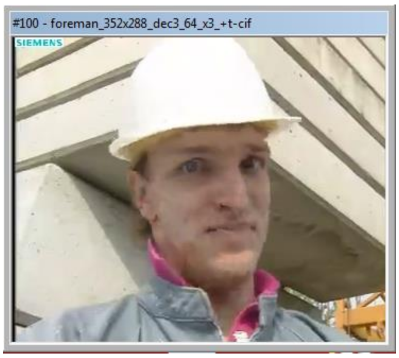
Figure 6 - 23 presents the *t+s* technique and *SL* outputs pictures at 64kbit/s. The technique provides a good quality picture at this bit-rate compared to the non-scalable technique. At bit-rate higher than 64kbit/s, the quality is as that of *SL* coder output picture.

a. *t+s*, 64kbit/s

b. *SL*, 64kbit/s



c. *t+s*, 64kbit/s



d. *SL*, 64kbit/s

e. s, 320kbit/s

Figure 6 - 23 : *t+s*, s and *SL* subjective quality performance, CIF format, 300 frames (a) *t+s*, frame 100, 64kbit/s (b)*SL*, frame 100, 64kbit/s (c) *t+s*, frame 100, 64kbit/s (d) *SL*, frame 100, 64kbit/s (e) s, frame 280, 320kbit/s

In QCIF format, the quality performance is guaranteed even at low bit-rate of 32kbit/s. Figure 6 - 24 presented *t+s* and *SL* techniques subjective quality performance for city sequence frame 74, QCIF resolution format.

a. *t+s*, 32kbit/s b. *SL*, 32kbit/s

Figure 6 - 24 : *t+s* and *SL* subjective quality performance, QCIF format, frame 74, 75 frames (a) *t+s*, 32kbit/s (b) *SL*, 32kbit/s

6.4.4. Comparison with other established scalability (*t+s*)

A number of experiments are conducted using foreman sequence to evaluate quality and adaptation performance of MPEG-4 scalability and *t+s* technique. Frame rates of 75/10 f/s and 15f/s are used for the experiments. The experimental results are presented in Table VI - VII.

In coding with 75/10f/s frame rate, *t+s* technique achieved bits reduction of 7000bits, 10590bits and 550bits per second for *SL*, BL and EL respectively. Similarly, the *t+s* technique gained a quality difference of 4.95dB, 10dB and 1.91dB for *SL*, BL and EL respectively. At coding frame rate of 15f/s, *t+s* technique gained 1160bits, -3190bits and -6790bits per second for *SL*, BL and EL respectively. Similarly *t+s* technique quality is bettered by 4.18dB, 4.78dB and 0.48dB. It is observed in the 15f/s, MPEG-4 technique gained over *t+s* technique on bits reduction while the *t+s* quality values are higher. The adaptive quantisation used with *t+s* favoured the use of high Qp values of 31.44 and 41.61 to adjust the picture quality.

Table VI - VII : *t+s* technique and MPEG-4 scalability adaptation and quality performance

Frame Rate (f/s)	**MPEG-4**				**H.264/*t+s* technique**			
	Bit-rates(kbit/s)/Luminance(γ)dB				**Bit-rates(kbit/s)/Luminance(γ)dB**			**Qp**
	SL	BL	EL	Qp	SL	BL	EL	SL/BL/ EL
75/10	47/31.01	32/30.97	57/31.04	13	35.39/35.92	21.41/40.97	56.45/32.95	31.51/23.69/38.36
15	47/31.01	32/30.97	57/31.04	13	45.84/33.41	35.19/35.75	63.79/31.52	33.41/31.44/41.61

6.4.5. Improved Temporal + Spatial Scalability (*Im-t+s*)

The *t+s* technique discussed above supports two layers of spatial resolution formats. To increase the scalability adaptation of the technique, one lower or higher spatial layer can be added. The added lower layer will then become the lower BL of the stream and the higher layer will represent the highest spatial layer. To implement the lower layer, the current BL is reduced to a layer of 128x96 format. The reduction is a process which costs an additional computational complexity of 0.69 sec with the available simulation computer (Appendix-A3). The gain achieved in this technique compared to the original BL is presented in Table VI - VIII. The modified *t+s* coder to *Im-t+s* coder is shown in Figure 6 - 25. The BL0 is the reduced version of BL1 from v-0 sequence.

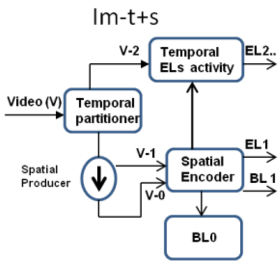


Figure 6 - 25 : *Im-t+s* technique encoder

The additional reduced spatial layer has improved the r.d.o. against the original spatial layer as shown in Table VI - VIII. This technique can be applied for scalability adaptation support especially where bandwidth and other decoding resources are limited.

Table VI - VIII : Gain of additional reduced spatial layer in *t+s* technique

Original BL Actual Bit-rates(kbit/s) / PSNR (dB)	New-BL Actual Bit-rates(kbit/s) / PSNR (dB)
38.25 / 32.55	30.30 / 35.16
60.65 / 35.11	44.20 / 37.68
77.25 / 36.43	56.46 / 39.24
93.85 / 37.45	67.49 / 40.34
108.81 / 38.21	77.44 / 41.19

Figure 6 - 26 shows the subjective performance result for the original and new created BL to support better scalability adaptation. The reduced BL can be up-sampled to the size of the

original BL at the same video quality. It can also be extracted to form the reduced version as shown in Figure 6 - 27 with good picture quality.

a. b.

c. d.

Figure 6 - 26 : **(a and b)** - New BL, 30.30kbit/s, 35.16dB. **(c and d)-** Original BL, 38.25kbit/s, 32.55dB

a. b.

Figure 6 - 27 : **(a and b)** - 30.30kbit/s, 35.16dB

Figure 6 - 28 shows the coded bits for two scalable BLs, the initial and the reduced version BL which indicated better scalability and adaptation.

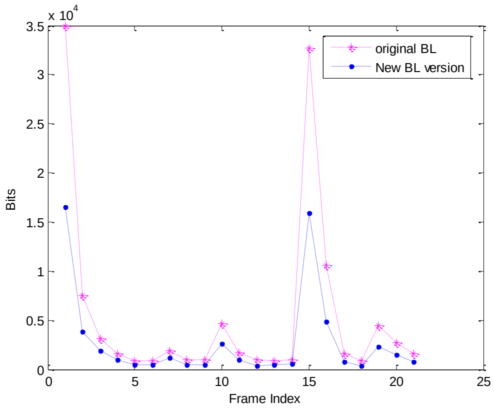


Figure 6 - 28 : *Im-t+s* – showing coded bits for Initial BL and reduced BL version

6.5. Temporal + Spatial + Quality Technique (*t+s+q*)

The combination of temporal, quality and spatial techniques may result in better adaptation and scalability. This technique therefore provides wider flexibility to multi-channel links and clients. A minimum of three original layers will be required to enable this technique. A minimum of two similar layers which enable quality scalability and one dissimilar layer to provide spatial picture scalability are required. In this technique, spatial scalability is included which is not presented in *t+q* scalability method.

With the increased number of layers, the inter-layer dependency is increased in this technique to provide lesser number of bits to be spent. From experiments with city sequence, it is shown that at bit-stream size of 100KB, *SL*, *t+q*, *t+s* and *t+s+q* techniques will be spending 170000bits, 169000bits, 169000bits and 135000bits per second respectively. The fully combined technique (*t+s+q*) spent fewer amounts of bits due to the large amount of inter-layer dependency and better prediction within the layers and their frames. This is discussed in section 6.7, and the experimental results are presented in Figure 6 - 43. The structure and functional procedures of this technique are made known in Figure 6 -29.

In Figure 6 - 30**,** the temporal partitioner partitions the original input video (V) into the video sequences V-0 and V-1. The video sequence V-2 is coded at the highest EL3 using the predictions from the lower layers. The video sequence V-1 is initially down sampled to generate a lower resolution video sequence V-0. This image sequence is then coded to provide Quality scalable coded video producing the BL and the lower EL1. The Quality (SNR) scalable coded is then up-sampled and interpolated to form the spatial enhancement encoder (EL2) predictions. The spatial encoder produces EL2 bit-stream.

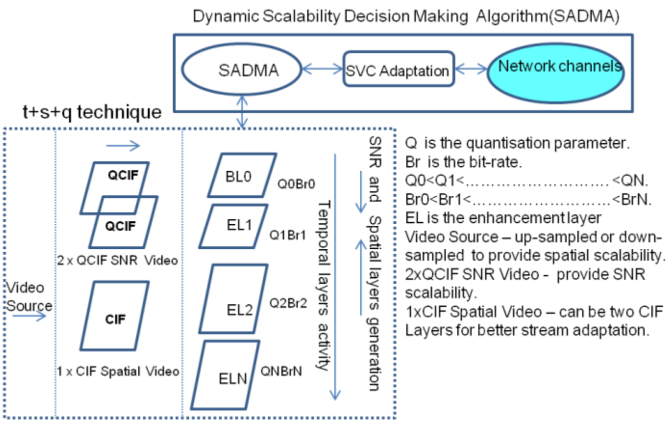


Figure 6 -29 : *t+s+q* technique architecture and *SADMA* (Scalability Decision Making Algorithm)

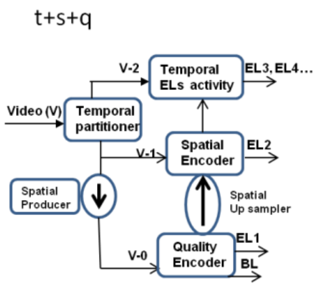


Figure 6 - 30 : *t+s+q* technique encoder

6.5.1. Temporal Scalable Layers Analysis (*t+s+q*)

An experiment is conducted with city video sequence, one of the standard ITU sequences. A CIF resolution format is used with a maximum of 150 frames. 2 x CIF and 2 x QCIF video sequences are used to enable a full scalability (*t+s+q*) as described in Figure 6 -29. A number of twenty two scalable layers are generated. Twelve layers are for CIF resolution scalability and ten layers for QCIF resolution scalability. Each of the layers takes a unique frame and bit-rate. This feature allows the generation of multiple quality layers embedded in the video stream. Also different quantisation levels are associated with the quality levels of the layers. For example, L0, L1, L2 and L3 are encoded with Q_p values 32.55, 32.57, 31.47 and 31.40 to generate a bit-rate of 38.25kbit/s, 44.80kbit/s, 181.20kbit/s and 212.74kbit/s respectively. The corresponding PSNR values of 32.55dB, 32.57dB, 31.47dB and 31.40dB are generated from these rates respectively.

The video stream embeds a minimum of 0.9375Hz and a maximum of 30.00Hz frame rates. The variable frame rates were encoded with variable bit-rates having a minimum of 36.80kbit/s and a maximum of 1131.40kbit/s discrete levels. The details of the results are summarised in Table VI - IX.

In a similar experiment with mobile sequence of maximum 300 frames, 12xQCIF layers and 14xCIF layers are generated for a total of twenty six scalable layers. This has improved the flexibility of scalability adaptation in the video stream. The minimum bit-rate for this sequence is 33.70kbit/s and maximum of 864.20kbit/s.

Table VI - IX : *t+s+q* technique: Temporal Scalability Layers with their associated quality EL - CITY sequence

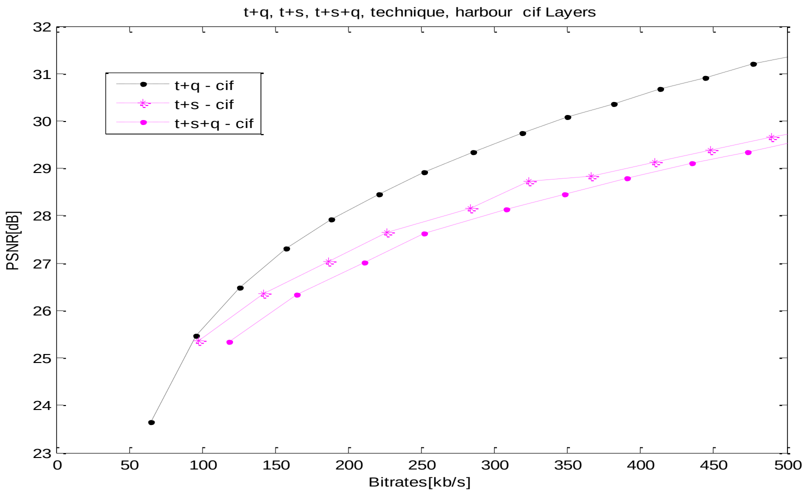
Lyr	Res.	Frame rate (Hz)	Bit rate(kbit/s)	Dependency_id	Temporal_id	Quality_id
0	176x144	0.9375	36.80	0	0	0
1	176x144	1.8750	55.90	0	1	0
2	176x144	3.7500	78.40	0	2	0
3	176x144	7.5000	103.80	0	3	0
4	176x144	15.000	132.50	0	4	0
5	176x144	0.9375	67.70	0	0	1
6	176x144	1.8750	102.00	0	1	1
7	176x144	3.7500	145.60	0	2	1
8	176x144	7.5000	197.30	0	3	1
9	176x144	15.000	250.90	0	4	1
0	352x288	0.9375	137.70	1	0	0
1	352x288	1.8750	212.00	1	1	0
2	352x288	3.7500	300.30	1	2	0
3	352x288	7.5000	393.10	1	3	0
4	352x288	15.000	505.20	1	4	0
5	352x288	30.0000	607.70	1	5	0
6	352x288	0.9375	241.40	1	0	1
7	352x288	1.8750	369.60	1	1	1
8	352x288	3.7500	529.10	1	2	1
9	352x288	7.5000	704.00	1	3	1
10	352x288	15.0000	919.10	11	4	1
11	352x288	30.0000	1131.40	1	5	1

6.5.2. Objective Evaluation and Analysis (*t+s+q*)

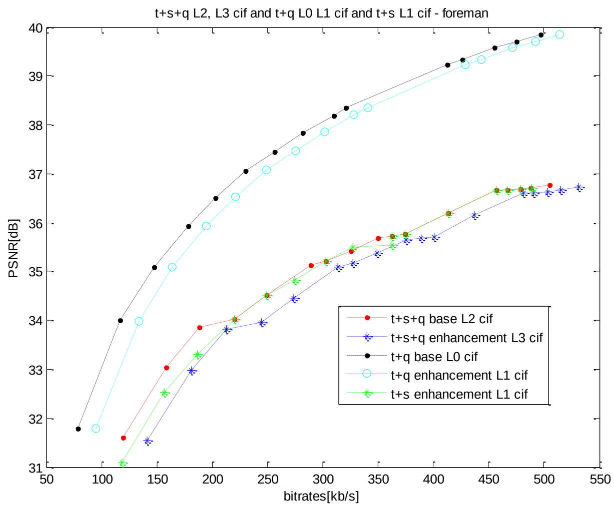
Several experiments are conducted with a number of sequences namely foreman, harbour and bus images. The purpose is to objectively evaluate quality and adaptation performance of *t+s+q*. The experimental results are presented in Figure 6 - 31.

From Figure 6 - 31 (a), 150kbit/s, 200kbit/s and 250kbit/s are required to generate 27dB quality value for *t+q*, *t+s* and *t+s+q* techniques respectively. The spent bits are higher with the *t+s+q* in this layer production. Since the *t+s+q* technique generates more layers than *t+s*, other layers might be formed with enhanced adaptation characteristics. The adaptation and r.d.o. performance for *t+s+q* (L1) technique shows better performance from the experiment conducted with bus sequence. This is shown in Figure 6 - 31 (c).

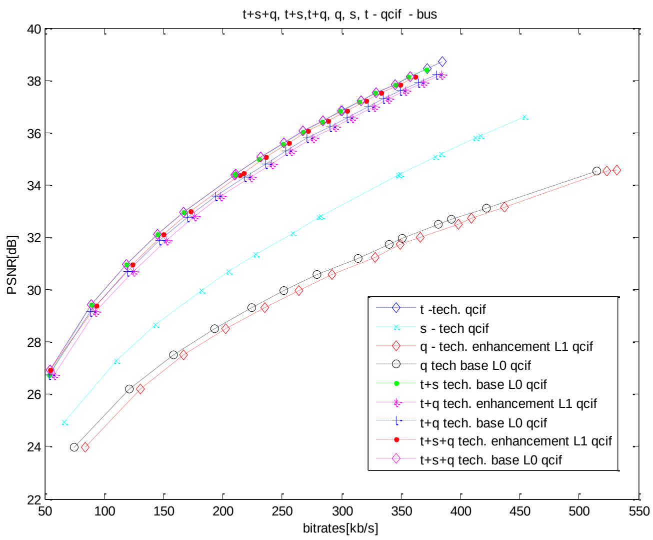
However, bits representation may not necessary satisfy the achievement of guaranteed video quality. Real time simulations over heterogeneous networks are conducted as presented on Figure 6 -44.


t+q, t+s, t+s+q, technique, harbour cif Layers
PSNR[dB]
Bitrates[kb/s]
t+q - cif
t+s - cif
t+s+q - cif

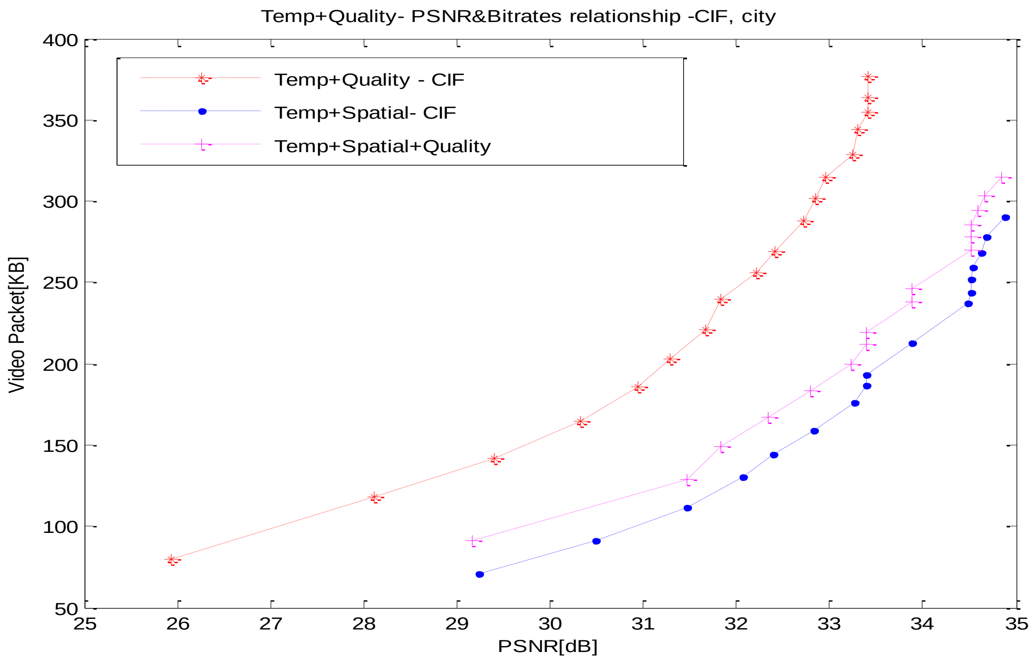
a.


t+s+q L2, L3 cif and t+q L0 L1 cif and t+s L1 cif - foreman
PSNR[dB]
bitrates[kb/s]
t+s+q base L2 cif
t+s+q enhancement L3 cif
t+q base L0 cif
t+q enhancement L1 cif
t+s enhancement L1 cif

b.


t+s+q, t+s,t+q, q, s, t - qcif - bus
PSNR[dB]
bitrates[kb/s]
t -tech. qcif
s - tech qcif
q - tech. enhancement L1 qcif
q tech base L0 qcif
t+s tech. base L0 qcif
t+q tech. enhancement L1 qcif
t+q tech. base L0 qcif
t+s+q tech. enhancement L1 qcif
t+s+q tech. base L0 qcif

c.



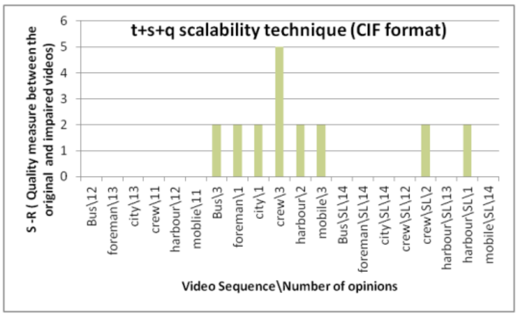
d.

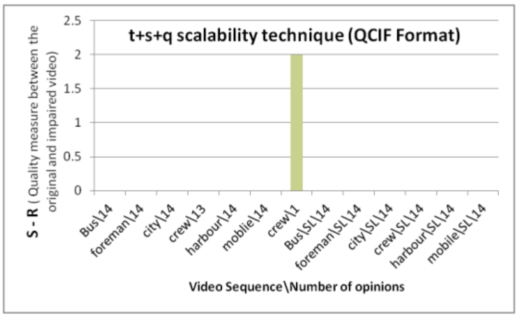
Figure 6 - 31: *t+s+q* –Quality and Adaptation performance for *t+s+q*, *t+s*, *t+q*, *SL* (t), s and q techniques (a) harbour (b) foreman, (c) bus and (d) city sequences

6.5.3. Subjective Evaluation and Analysis (*t+s+q*)

Similar procedures and guidelines employed for *t+q* and *t+s* techniques are use for subjective evaluation of *t+s+q* output pictures. The evaluation results are presented on Figure 6 - 32 (a) and (b).

The gathered opinions for QCIF format revealed that, QCIF products of *t+s+q* technique are approximately hundred percent similar to the original videos in the spatial domain. This result is shown in Figure 6 - 32 (b). However, with CIF resolution format, there are few opinions indicating a slight difference among the original and *t+s+q* pictures. The CIF results are presented in Figure 6 - 32 (a). Reasons for this slight difference in brightness are discussed in the previous section 6.4.3. The differences in the feature and natural content of the original video data may influence the end quality of the product. This is because, different backgrounds and foregrounds may hold different bits values and this is influenced by the prediction technique that is used for coding the video frames.





a.

b.

Figure 6 - 32 : *t+s+q* subjective evaluation, pair opinion scores among the original video, *SL* and *t+s+q* techniques.

Figure 6 - 33(a, b, c, d) present pictures with subjective quality performance for *t+s+q* and non-scalable techniques. The quality presented is achieved from 96kbit/s and 64kbit/s. Both the two techniques achieved almost the same decoded quality with the original video sequences.

a. *SL*, 96kbit/s

b. *t+s+q*, 96kbit/s

c. *SL*, 64kbit/s

d. *t+s+q*, 64kbit/s

Figure 6 - 33 : *t+s+q* and *SL* subjective quality performance, frame 100, 300 frames (a) *SL*, 96kbit/s (b) *t+s+q*, 96kbit/s (c) *SL*, 64kbit/s (d) *t+s+q*, 64kbit/s

Figure 6 - 34 presents the QCIF resolution format for *t+s+q* and a non-scalable technique with the decoded quality at 32kbit/s. Observation shows that these two pictures produce almost the same visual quality.

a. *SL*, 32kbit/s b. *t+s+q*, 32kbit/s

Figure 6 - 34 : *t+s* and *SL* techniques, frame 30, 75 frames (a) *SL*, 32kbit/s (b) *t+s+q*, 32kbit/s

Figure 6 - 35 shows frames 14 and 34 for *t+s+q*, *SL* and the original video sequences. At this bit-rate of 64kbit/s, a good quality is achieved which can be improved by applying the de-blocking filter.

a. Pixel domain, frame 14 b. *SL*, 64kbit/s, Frame 14 c. *t+s+q*, 64kbit/s, Frame 14

d. Pixel domain, Frame 34 e. *SL*, 64kbit/s, Frame 35 f. *t+s+q*, 64kbit/s, Frame 34

Figure 6 - 35 : *t+s+q* – subjective quality evaluation

6.5.4. Comparison with other established scalability (*t+s+q*)

Experiments are conducted with foreman sequence to evaluate the quality and adaptation performance of *t+s+q* against MPEG-4 scalability techniques. The results are presented in Table VI - X.

In coding the stream at 75/10f/s, a bit reduction of 11100bits, -4280bits and -8840 bits is achieved for *SL*, BL and EL for *t+s+q* scalability against MPEG-4 scalability. Correspondingly, a delta gain of 4.91dB, 4.51dB and 0.61dB is obtained for *SL*, BL and EL respectively for *t+s+q* method against that of MPEG-4.

It is observed that, *t+s+q* technique has gained an increased bit-rate while maintaining a higher quality than MPEG-4 at these rates, although a better quality is achieved from *t+s+q* technique. Another distinctive characteristic of *t+s+q* is that, a larger number of temporal layers are generated than that of MPEG-4 which has improved the bit-stream adaptation to the network.

At 1.5/10 f/s coding frame rate, a bit reduction of 1160bits, -15100bits, and -5350bits is obtained for *SL*, BL and EL respectively for *t+s+q* against MPEG-4. In the same way, a quality gain of 2.4dB, 2.38dB and 2.32dB for these rates is realised for *t+s+q* technique against the MPEG-4 technique.

It can be concluded from the above results that H.264/AVC favours Quality (PSNR values rather than bit-rates in *t+s+q* scalability technique. This is so because, at lower bit-rates, poor quality is more likely to be achieved unlike in higher bit-rates video communication.

Table VI - X : *t+s+q* technique and MPEG-4 scalability, quality and adaptation performance

Frame Rate (f/s)	**MPEG-4**				***t+s+q* technique**			
	Bit-rates(kbit/s)/Luminance(γ)dB				**Bit-rates(kbit/s)/Luminance(γ)dB**			**Qp**
	SL	BL	EL	Qp	SL	BL	EL	SL /BL/ EL
75/10	47/31.01	32/30.97	57/31.04	13	35.39/35.92	36.28/35.48	65.84/31.65	31.51/32.01/42.57
15	47/31.01	32/30.97	57/31.04	13	45.84/33.41	47.10/33.35	51.65/33.36	33.41/35.76/36.36

6.6. Spatial + Quality technique (*s+q*)

Temporal scalability is reduced in this technique implementation. This has increased the number of intra-frames within the video and the breakdown of the video into small groups of pictures. The increase in the I-frames increases the number of bits generated and hence escalates the bandwidth requirements. However, this will guarantee a high quality picture in no-delay channels and if the I-frames (key-pictures) are not encoded. Key pictures are usually encoded for improved adaptation and rate control. In error prone and delay channels, if the key pictures are encoded this might result in poor video quality. The enhancement layers production is as described in the other techniques, though a reduced number of temporal layers are realised due to the disabling of temporal scalability in the technique. This is discussed in section 6.4.1. Figure 6 - 36 describes the functional and procedural structures of this technique.

One other distinguishing feature of this technique compared to others is the ability to use only high frame rates. Experiments show that the minimum frame rate that can be employed with this technique is 15Hz. This is because a large number of I-frames are used at the beginning of each *GOP* hierarchy. The *GOP* size is designed to be small and hence large frame skipping is not allowed and the frame rate becomes faster. It is also observed that encoding of the key pictures at the start of every *GOP* may result in poor quality product of this technique.

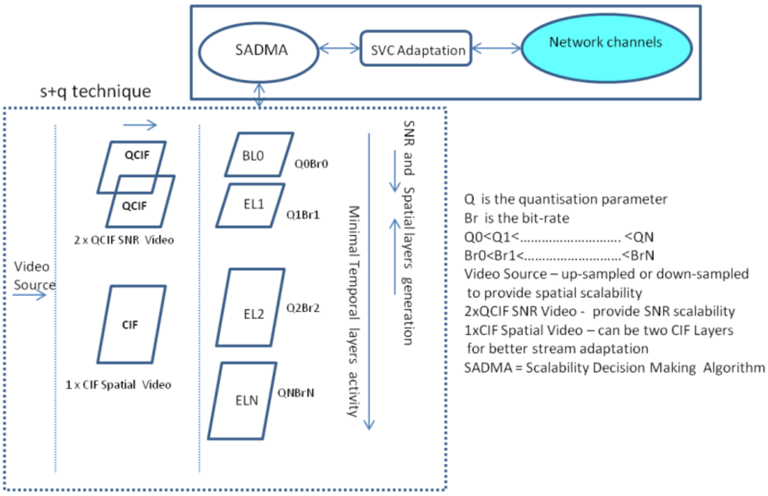


Figure 6 - 36 : *s+q* technique architecture and *SADMA* (Scalability Decision Making Algorithm)

In Figure 6 - 37, the original video (V) is down-sampled to a lower resolution V-0. This is fed into the Quality (SNR) encoder which forms the BL and the lower EL1. The locally decoded frames from the Quality encoder are up-sampled to a higher resolution to form the prediction pictures for the spatial encoder. The spatial encoder produces the predictions for the next ELs (EL2 to ELN).

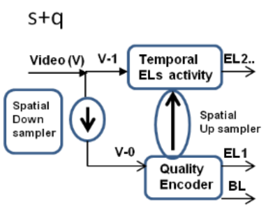


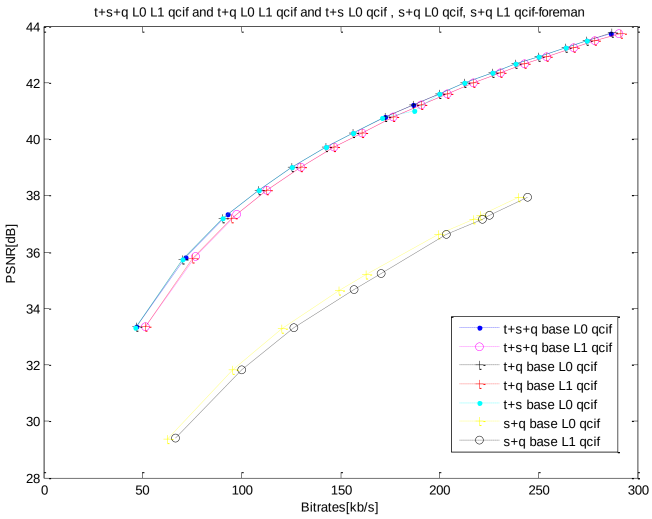
Figure 6 - 37 : *s+q* technique encoder

6.6.1. Temporal Scalable Layers Analysis (*s+q*)

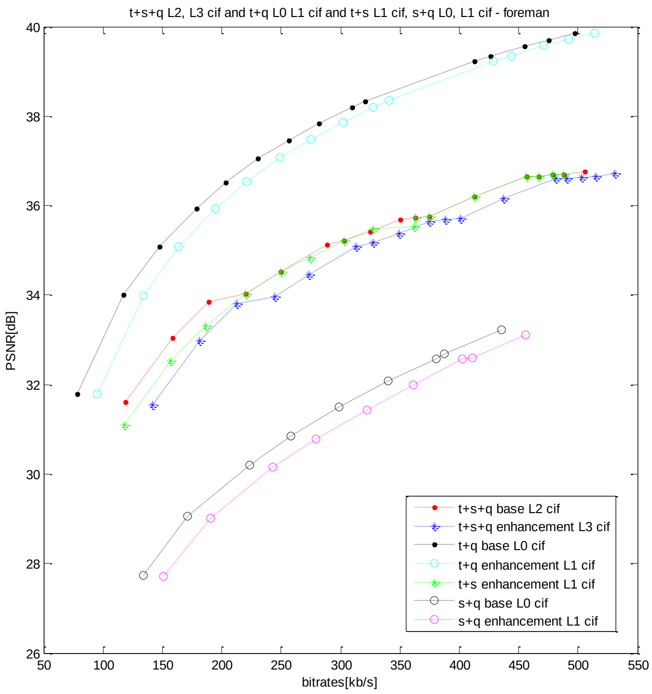
An experiment is conducted with foreman sequence to assess the scalability performance of *t+s* technique. The experimental functional components are set-up as illustrated in Figure 6 - 36. A total of six scalability layers are produced consisting of 2x layers for QCIF and 4x layers for CIF format resolutions. A minimum and maximum frame rate of 15f/s and 30f/s are used. This is because shorter frame skipping is adopted in *s+q* technique and hence the temporal layers production is minimal and only six layers are created. Also, this technique employs small *GOP* sizes which resulted in a large number of bits lessening network adaptation. In this experiment, the generated bit-stream is embedded with sub-layers of minimum and maximum bit-rates of 69.50kbit/s and 140.90kbit/s.

6.6.2. Objective Evaluation and Analysis (*s+q*)

Experiments have revealed that, the decoded picture quality and adaptation is low compared with the other techniques. From Figure 6 - 38 (a, b), *s+q* layers are under the curve where low quality values and higher bit-rates are observed as compared to other techniques producing better r.d.o. curves. The two layers of *s+q* technique L0 and L1 for both QCIF and CIF formats show poor objective quality from the results as compared to *t+s+q*, *t+q* and *t+s* scalability methods.



a. QCIF



b. CIF

Figure 6 - 38 : *s+q* objective quality performance, foreman sequence

6.6.3. Subjective Evaluation and Analysis (*s+q*)

Figure 6 - 39 and Figure 6 - 41 represent the decoded frame 30 picture quality for QCIF and decoded frame 100 picture quality for CIF format respectively. The QCIF frame is decoded at

32kbit/s for both the *s+q* and *SL* techniques. *s+q* technique decoder produced poor subjective quality picture compared to *SL* technique.

a. *SL*, 32kbit/s, frame 30 b. *s+q*, 32kbit/s, frame 30

Figure 6 - 39 : *s+q* and *SL* techniques, frame 30, 150 frames

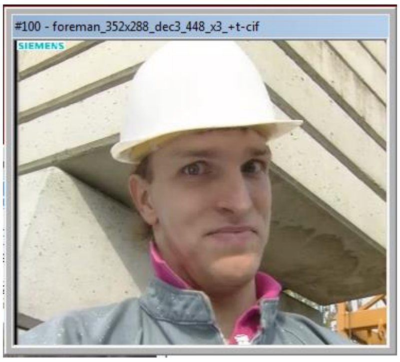
Figure 6 - 40 shows a 64kbit/s frames for original, *SL* and *s+q* video quality. *s+q* technique provides the lowest picture quality as compared to the other developed techniques products.

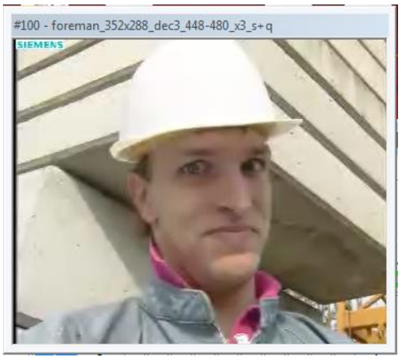
a. Pixel domain, frame 141 b. *SL*, 64kbit/s, Frame 141 c. *s+q*, 64kbit/s, Frame 14

a. Pixel domain, frame 35 b. *SL*, 64kbit/s, Frame 35 c. *s+q*, 64kbit/s, Frame 35

Figure 6 - 40 : *s+q*, subjective quality evaluation

The CIF frames in Figure 6 - 41 (a) and (b) are decoded at bit-rates of 448kbit/s and 480kbit/s for *SL* and *s+q* techniques. The smaller *GOP* sizes employed for these techniques where the key pictures are encoded resulted in bad quality. Encoding the key pictures for small *GOP* sizes removes a large amount of details from the video content. The details of key pictures encoding and simulations are discussed in section 5.12.





a. *SL*, 448kbit/s b. *s+q*, 480kbit/s

Figure 6 - 41 : *s+q* technique, frame 100, 300 frames (a) *SL*, 448kbit/s (b) *s+q*, 480kbit/s

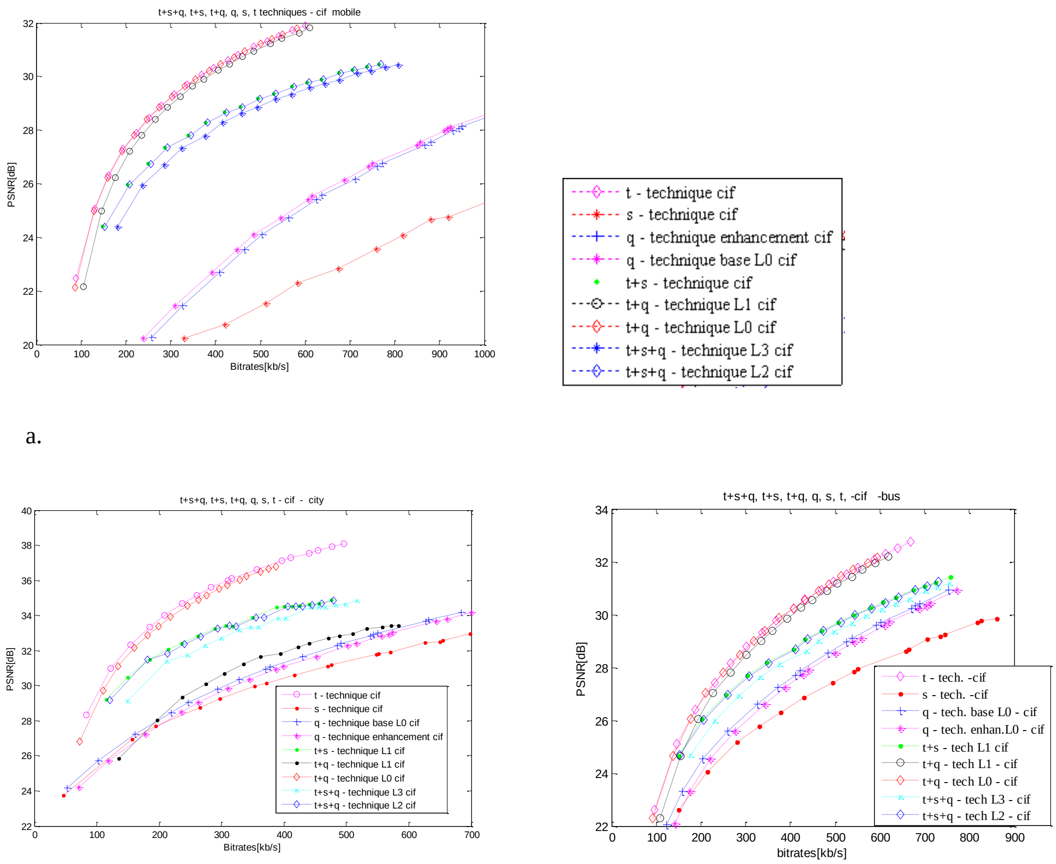
6.7. Comparative Analysis of the Developed Scalability Techniques

To evaluate the developed techniques, several ITU-standard sequences are used in the simulations. Three sequences with the highest number of frames (300 frames) namely foreman, mobile and harbour are used. These sequences are obtained from ITU-standard generally used video sequences for JSVM H.264/AVC Extension research [9-11]. The selected sequences are characterised by clear fore and back-grounds as well as obvious video motion. Other standard sequences used with lower number of frames (150 frames) are bus, crew and city video sequences. To conduct a fair comparison between the techniques and *SL* evaluation, all experiments are run under the same encoder and decoder configurations and target bit-rates.

It is observed from Table VI - XII that L0 bit-rates and luminance values are similar for *t+s+q*, *t+q* and *t+s*. One reason for this is the dependency of BL, and each of the techniques use the same syntax for the BL as it stands as a reference to all other layers. The initial frame bits are not inter-coded and may not contain any other predicted value. The subsequent inter-coded frames will contain predicted values that depend on the number of factors including layering and employed scalability technique and layer dependency used.

The sequences used in the entire experiments are in the pixel domain and both lower and high resolution formats QCIF and CIF respectively are experimented with. The luminance (γ) components in dB and bit-rate values in kbit/s are extracted for each of the techniques and sequences used. From Table VI - XII, L0 stands for a base layer with a QCIF format while L1 represents the enhancement layer with CIF video format. Single Layer (*SL*) CIF and QCIF are compared with L1 and L0 layer components of other techniques respectively. This is because *SL* generates only a layer. Hence, for the user requiring a resolution other than in the transmitted *SL* video, the reconstructed video format must be up or down converted to a higher or lower resolution respectively. This results in extra processing cost for the decoder. This cost is avoided when the scalability techniques are used and hence the scalability techniques are more beneficial for resolution adaptation and real time video services.

The technique *t+q* has a better efficiency with CIF resolution than the other techniques. The technique embeds several scalability and QoS levels ranging from low to high quality videos. The nature of this technique has given it a powerful flexible scalability which eases the bandwidth burden of service delivery and can open market for service providers to new subscribers. The combination of both temporal and quality scalability (*t+q*) provides a better prediction than just quality scalability (q) within the bit-stream. From the experiments conducted, with a bandwidth of 100kbit/s, q achieved < 28dB while *t+q* achieved ~32dB. These results are shown in Figure 6 - 42 for mobile, crew and bus video sequences.



a.

b.

c.

Figure 6 - 42 : Comparative Analysis and Evaluation for the Developed Scalability Techniques (a) mobile (b) crew (c) bus

It can be concluded from the results which show the objective performances of the techniques and scalable layers (Figure 6 - 42 and Table VI - XI) that the designed techniques are better in terms of performance, adaptability and high flexible scalability. The experiment in Table VI - XI is conducted with foreman sequence, 300 frames at a target bit-rate of 32kbit/s. For example, *t+q* produced fourteen scalable layers with 87.20kbit/s, 122.60kbit/s, 182.20kbit/s, 258.30kbit/s, up to 506kbit/s discrete levels of bit-rates for CIF and other scalable layers for QCIF resolution formats. This is a different adaptation performance from q technique that produced only four scalable layers.

The techniques that do not provide a high level of temporal scalability will have a large number of I-frames within the pictures and therefore a large number of bits is spent. This has however ensured a guaranteed picture quality at the expense of poor adaptability in a non congested network environment. In a situation where network channels experience excessive congestion, picture quality can deteriorate and may not be effective for real time applications. Table VI - XII presents the experimental results from six dissimilar ITU sequences to assess the quality and adaptation characteristics for *t+q*, *t+s+q*, *t+s*, *SL*, q, and s techniques for both high-resolution format CIF and low-resolution format QCIF. *t+s+q*, *t+q*, *t+s* techniques show better adaptation characteristics compared to *s+q*, q, and s scalability techniques respectively.

Table VI - XI : Developed and Current Scalable Techniques- Scalability Layers and Adaptation Characteristics

TECHNIQUE	SCALABILITY AND QUALITY LAYERS	MINIMUM BITRATE(kb/s)	MINIMUMFRAME RATE (Hz)
SINGLE SCALABILITY TECHNIQUES			
Quality	4	215.00	15.0000
Spatial	3	215.00	15.0000
DEVELOPED COMBINED SCALABILITY TECHNIQUES			
Temp+Quality	26	29.80	0.4688
Temp+Spatial	13	47.60	0.4688
Temp+Quality+Spatial	22	27.10	0.9375
Quality+Spatial	6	215.10	15.0000

The developed techniques also show higher bitrates than the *SL* technique from 1- 20kbit/s for CIF for some video sequences. This is due to additional administrative overheads included in the scalable bit-streams. With scalable bit-stream there are overheads termed as *Supplemental Enhancement Information (SEI)* messages for each layer. These include picture format, bit-rate and many others that allow sub-stream extraction, decoding and control. Also identifiers for specific elements types such as slice groups, macroblocks and many others are additional overheads compared to SL technique. *t+q* scalability method shows better scalability adaptation among the other developed techniques from the experimental results gathered in Table VI - XII. This is followed by *t+s* method and then *t+s+q* scalability method. However, the actual real time adaptation is simulated and the result is presented in Figure 6 -44.

Figure 6 - 43 (a, b) presents experimental results evaluating the bit-stream size that the techniques generate against variable bit-rates and quality values. *t+q* technique shows better compression efficiency followed by *t+s+q* and then *t+s* and *SL* techniques. *t+q* achieved this efficiency due to efficient redundancy removal in the spatial and temporal domains. Same video layers are used to form and construct *t+q* technique as discussed in section 6.3.1.

In Table VI - XII, to run the *SL*-coder, the flag SVCMode is set code 0 and is set to code 1 to run the scalable mode.

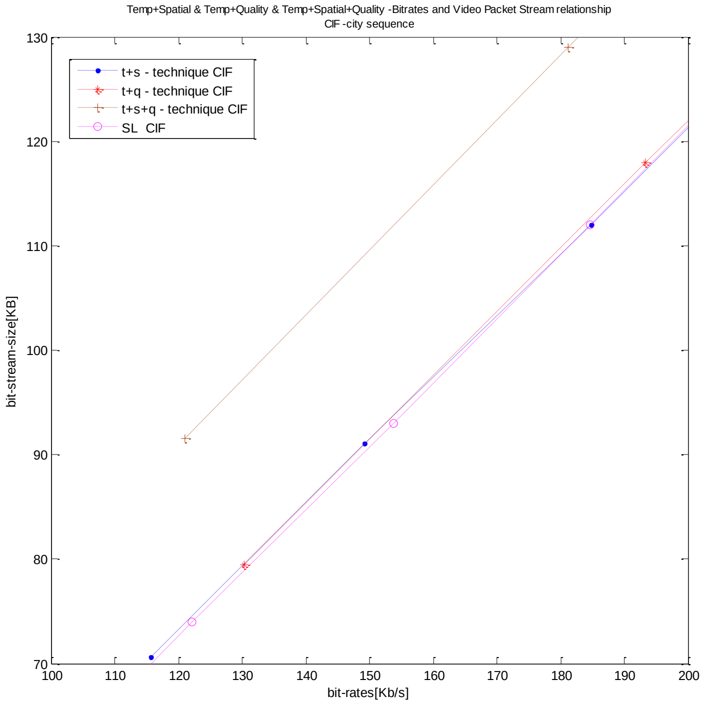
Table VI - XII : Performance Evaluation for the Developed Scalable Video Coding Techniques

Video Seq.		**Scalability Techniques Evaluation**						
		Bit-rates(kbit/s)/Luminance(Y)dB						
		t+s+q	***t+q***	***t+s***	**q**	**s**	***SL*-cif**	***SL*- qcif**
Foreman	L0	47.10/33.35	47.10/33.35	46.55/ 33.32	62.35/29.38	62.35/29.38	117.10/31.84	46.55/33.32
	L1	119.02/31.55	78.26/31.78	118.08/31.09	134.21/28.44	183.11/27.89		
Crew	L0	46.08/28.64	43.99/28.88	46.08/28.64	51.74/29.17	51.74/29.17	134.99/28.08	47.59/29.23
	L1	131.58/27.22	102.29/25.70	126.09/27.39	142.18/28.16	131.88/27.72		
City	L0	38.25/32.55	38.59/32.88	38.25/32.55	56.05/27.59	40.82/46.73	122.12/30.96	42.93/33.61
	L1	121.07/29.16	73.06/26.81	115.73/29.23	102.33/25.72	92.30/30.77		
Mobile	L0	54.25/28.26	54.25/28.26	54.25/28.26	92.83/21.88	92.83/21.88	130.69/25.07	54.25/28.26
	L1	153.26/24.39	86.16/22.14	147.51/24.41	239.24/20.24	66.39/24.93		
Bus	L0	54.97/26.94	53.80/26.74	53.80/26.74	75.27/23.96	66.39/24.93	146.59/25.09	54.97/26.94
	L1	153.26/24.67	91.84/22.30	150.50/24.65	159.16/23.30	150.80/22.61		
Harbour	L0	40.13/28.56	40.13/28.56	40.13/28.56	61.48/25.21	61.48/25.21	80.60/25.53	40.13/28.56
	L1	101.21/25.36	48.01/23.65	97.36/25.37	127.73/23.43	169.74/22.86		

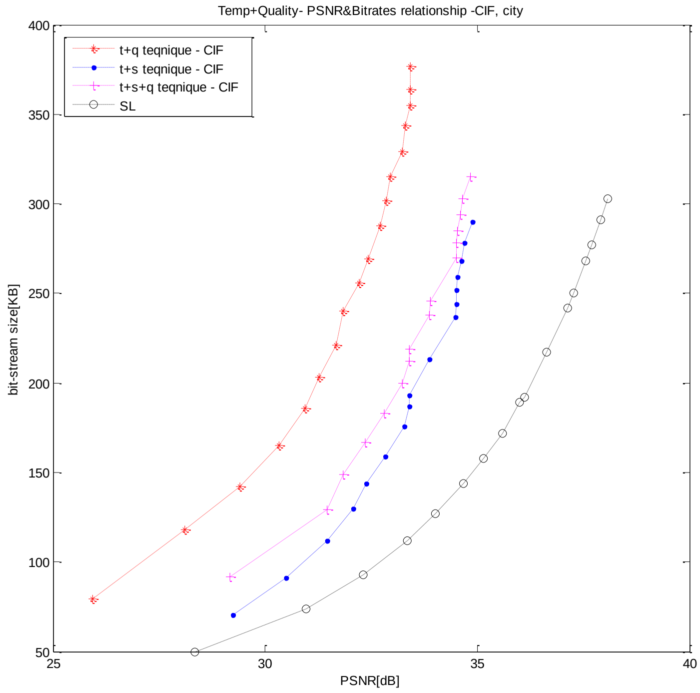
The complexity of the techniques is also evaluated based on the time it takes the encoder to encode a given number of frames for a video sequence using one of the scalability techniques as shown in Table VI - XIII. The techniques layering structure, layer dependency and level of scalability is varied as discussed in their respective sections. Therefore, it takes a different time to encode each of the techniques. The adoption of a high speed processor can improve the performance of a technique. For instance, *t+s+q* technique takes more time than others but is more scalable and adaptable to bandwidths, resolution and other resources. This is shown from the real time simulations discussed in this section. It should also be noted that the time shown in Table VI - XIII includes the time for extracting the experimental data which takes a large percentage of the total time. The experimental PC specification is shown in Appendix-A3.

Table VI - XIII : Developed techniques complexity based on encoding time

Technique	Encoding Time (mins)
t+q-qcif	2.68
t+q-cif	19.86
t+s	17.29
t+s+q	51.72
SL	0.07



a.



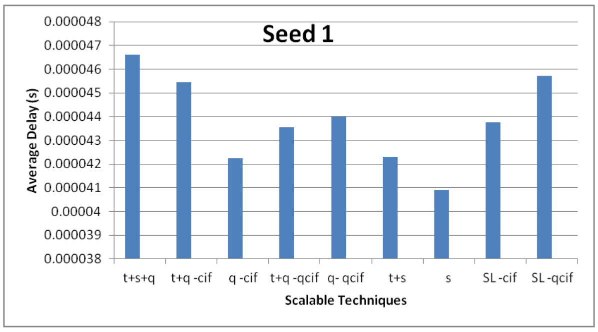
b.

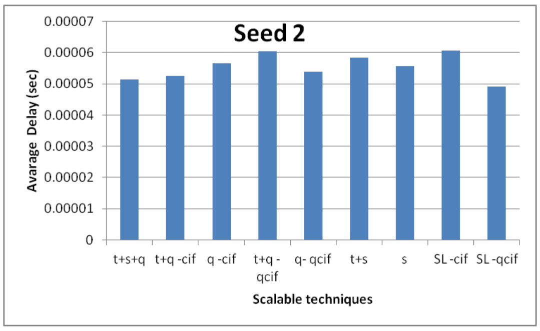
Figure 6 - 43 : Developed techniques and *SL*: (a) bit-rates [kbit/s] and bit-stream size relationship (b) PSNR [dB] and bit-stream size relationship

Real time simulations are conducted to measure the characteristics of the developed techniques over heterogeneous channel links. The simulation collects 100 values of samples per statistic. The collected statistic represents the end to end delay of all packets received by all the stations. The method of collection is referred to as bucket mode sample mean. Five different seeds are used for each of the techniques. The number of seeds and channels used are limited due to the available memory resource of the simulation computer (Appendix A3). The simulation results are presented in Figure 6 -44. Both high and low resolution video sequences are used in the simulation. *t+q*-qcif and *t+s+q* performed with better average low delays over the links compared to *SL*. The video natural content type can influence the obtainable delay. This is because the number of spent coded bits will vary from one video to another even when the same technique is used. Also the level of flexibility in the scalable stream is one of the affecting factors. Table VI - XIV shows different channels resources set-up to simulate the level of scalability performance of the techniques within limited multi-channel resources. *Configuration-A* and *Configuration-B* are set-up. The techniques performed within the two configurations with a slight delay difference for some seed values.

Table VI - XIV : Resources Configurations for the Heterogeneous network Simulator

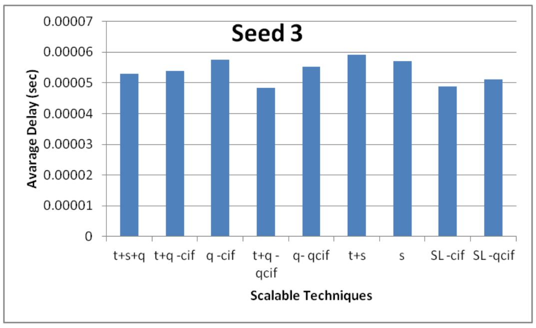
Resource	**Configuration-A**	**Configuration-B**
1. Node_27 (router –LAN and WIMAX) IP packets processing	500,000 packet/sec	100,000 packet/sec
2. Memory (node_27)	16MB	8MB
3. Datagram Forwarding rate (node_27)	100kbit/sec	96kbit/s
4. WILAN- data rate	18Mbps	24Mbps
5. WIMAX – mobile_4-4_node	ARQ- disabled	ARQ-enabled
6. WIMAX – Datagram –forwarding rate	Infinity	5kbit/s
7. Memory(WIMAX)	16MB	32MB
8. Datagram forwarding rate node_18 (WILAN)	32kbit/s	64kbit/s
9. Datagram forwarding rate LAN (node_23)	500kbit/s	32kbit/s

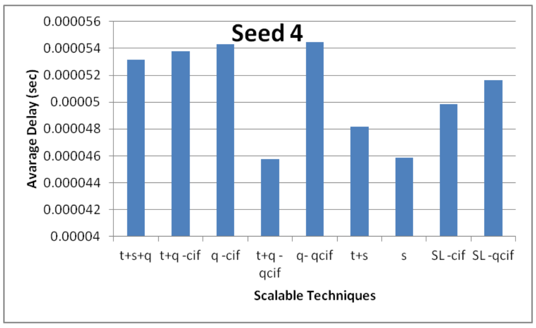




a.

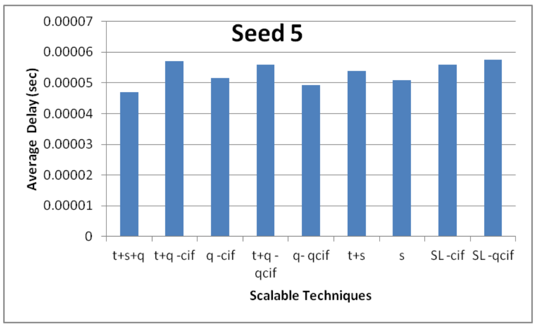
b.





c.

d.



e.

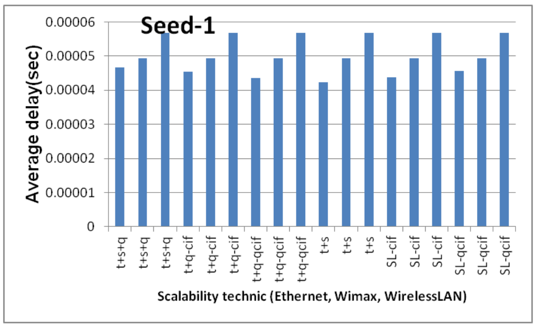
Figure 6 -44 : *SADMA*: Real Time Performances of Developed Scalability Techniques over Heterogeneous Networks Configuration-A

Further simulation results showed the robustness, functional scalability and flexible adaptation of all the techniques. A real time simulation is conducted with mobile sequence. The sequence output from each of the techniques and *SL* are used to determine the performance of each technique over heterogeneous networks consisting of WIMAX, LAN and wireless LAN. The simulation results are presented in Figure 6 - 45. The simulation was conducted using five different seeds in the algorithms.

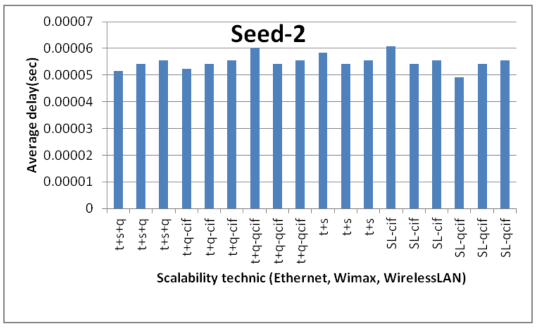
In the presented result in Figure 6 - 45, each of the techniques shows results for the average delay encountered within the video communication transition. The Ethernet delay is the average delay encountered through all stations within the video communication time. WIMAX is the queue delay experienced within the WIMAX network and the Wireless LAN is the average delay collected within the video session. An equal amount of delay is experienced in much of the simulation time. This shows the robustness and flexibility of the techniques for scalability adaptation. Each of the techniques is designed to adapt to numerous clients and target resources including bit-rates, frame rates and resolution.

The simulation used a purposely built *heterogeneous simulator* discussed in chapter four to conduct the simulation events and to test the developed techniques and the *Scalability Decision Making Algorithm* (Chapter Seven). The built network covered some of the common and available networks. Other networks such as UMTS, LTE and many others should have been included but due to limited computer system processor and memory capacities, this is limited to only three networks. UMTS was included and memory error is reported and the simulation is limited to WIMAX, LAN and wireless LAN. However, the number of conventional and available networks could not be exhausted in a simulation, the

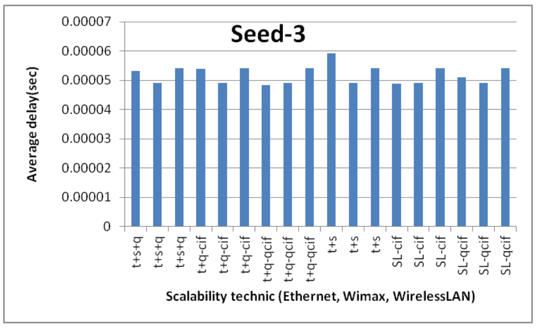
SADMA algorithm in chapter seven intelligently understands when a bit-stream needs to be switched to another. The algorithm (*SADMA*) does this by constantly monitoring the network condition when a video stream has been pass into the network channels.



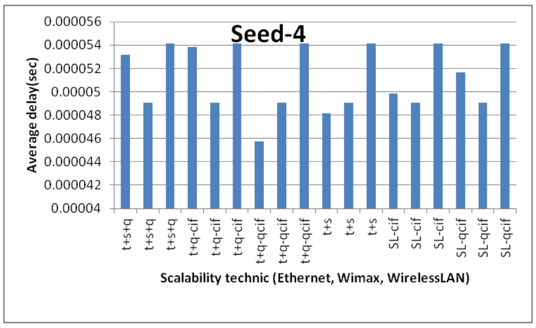
a.



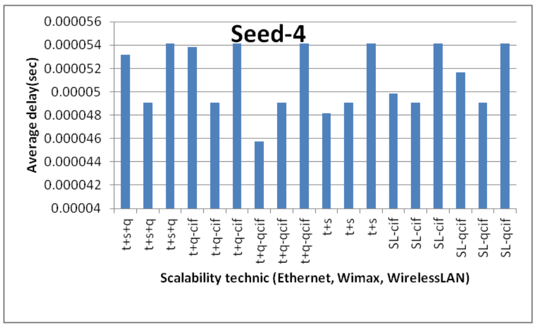
b.



c.



d.

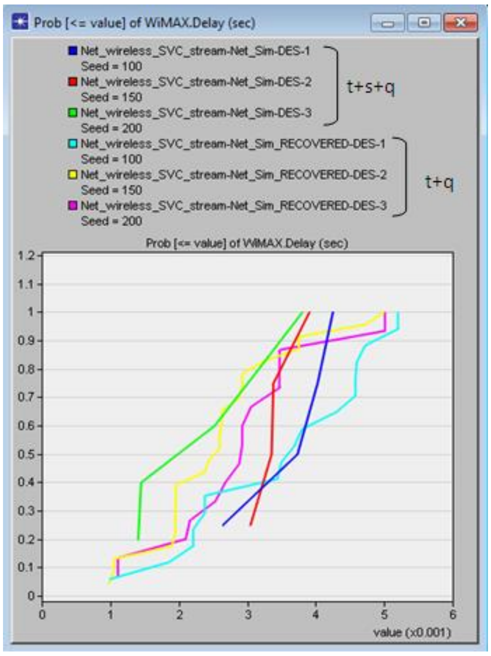


e.

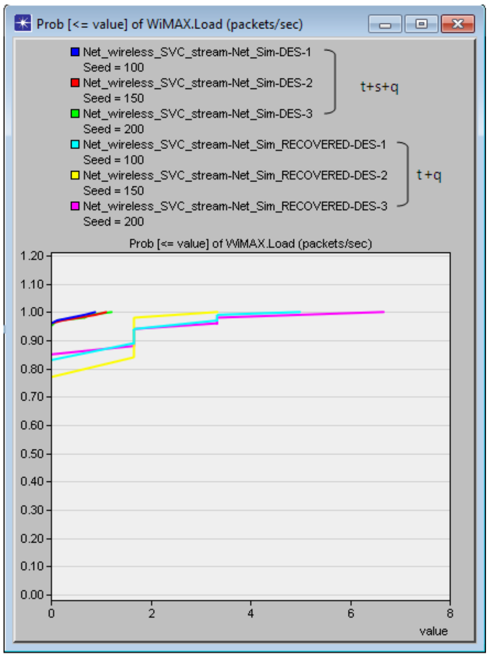
Figure 6 - 45 : Real time performances for scalability techniques and single layer over heterogeneous networks (Ethernet, WIMAX and Wireless LAN) – Configuration-B.

A similar simulation is conducted at maximum bit-rate of 250packets/sec and all the techniques show similar average delay values across the WIMAX channels. However, the average Ethernet delay experienced for each technique across all stations differed as shown in Figure 6 - 45. The *SL* technique experiences slightly low delay due to fewer overheads and the current network configuration compared to the scalable techniques. The performance of the BL over the network channels compared to other ELs is presented in section 7.10. The procedure for prioritising bit-stream and layers is implemented with the *SADMA* algorithm discussed in chapter seven.

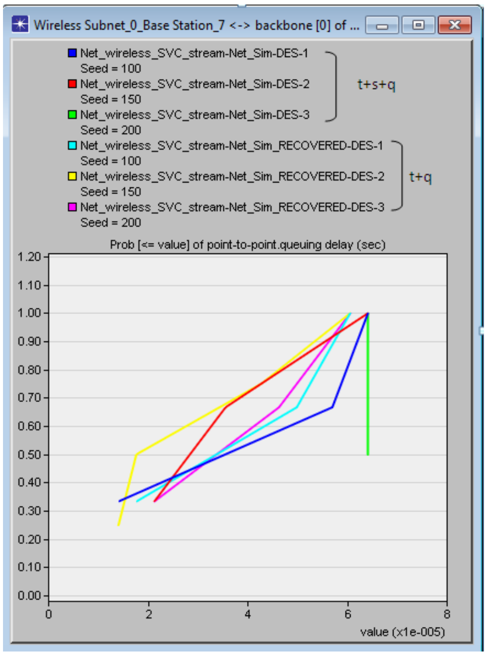
The simulation results presented in Figure 6 - 46 illustrates the probability of delay in seconds with a value <= the WIMAX delay. The probability extracted shows slightly different performance of the techniques. However, the queuing delay is revealed to be the same for *t+q* and *t+s+q* in Figure 6 - 46 (d). Hence, this shows the robustness of the scalability techniques where adaptation layers are embedded in each of the techniques. From Figure 6 - 46 (a-d), different seeds give different values where a technique is better in one seed and is inferior in the other.



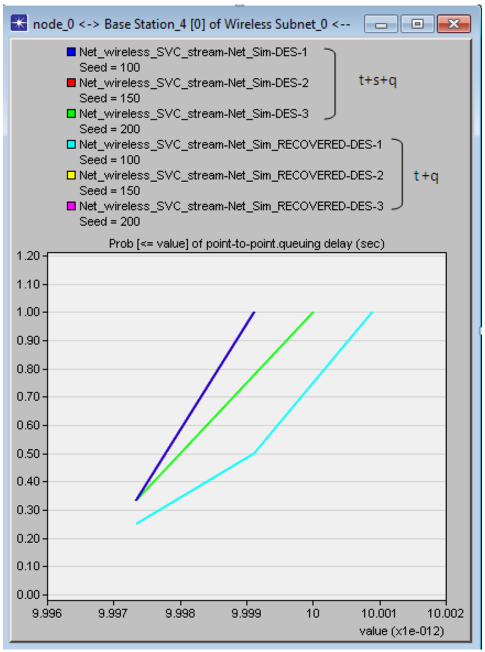
a.



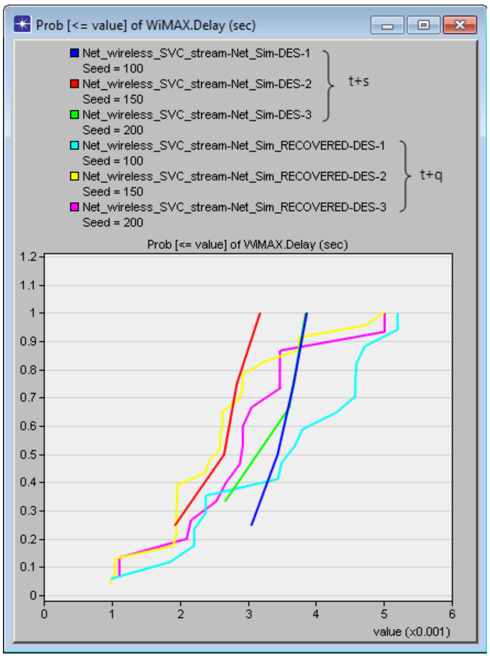
b.



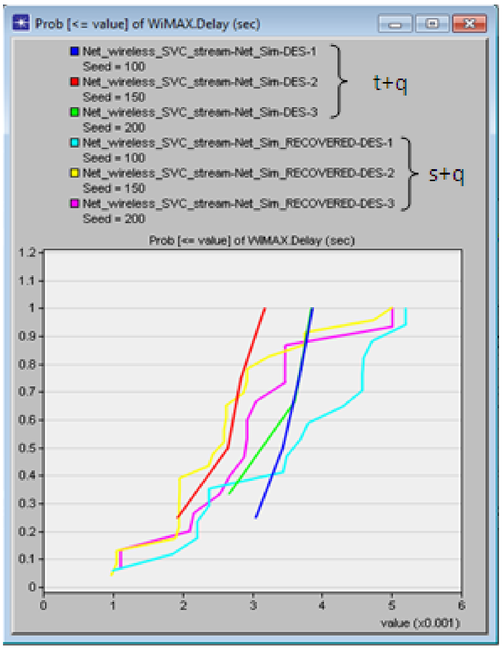
c.



d.



e.



f.

Figure 6 - 46 : Real Time Simulations for *t+q* and *t+s+q* techniques over heterogeneous networks at 250packets/sec

Table VI - XV shows average delay values taking into account different seed values. The WIMAX represents an average delay encountered within the WIMAX channels connection. The Ethernet is the average delay encountered in all stations and the WLAN is within the wireless-LAN connections. Due to the scalability adaptation support of all the techniques, there is little or no delay difference among the techniques. However, the priority and importance of a technique will be prioritised on a particular station. For instance in WIMAX, *t+s+q* can be prioritised since it can support all the temporal, quality and spatial scalability and the technique experienced lesser delay over all stations. Any of the techniques can support WILAN. The high delay encountered is due to the lack of hand-over support from Opnet tool for the WIMAX simulator. However, the link between WIMAX and WILAN has been created in the design as discussed in chapter four.

Table VI - XV : Average delay values for WIMAX, Ethernet and WILAN seeds for the scalability techniques over heterogeneous network- Resources Configuration-A

Technique	**WIMAX average delay (secs)**	**Ethernet average delay (secs)**	**WILAN delay average (secs)**
t+s+q	0.3261738	0.005023	0.5309553
t+q –cif	0.3255976	0.052518	0.5309553
t+q –qcif	0.3255976	0.0050822	0.5309553
t+s	0.3255976	0.0052338	0.5309553
SL-cif	0.3255976	0.0051814	0.5309553
SL-qcif	0.3255976	0.0051062	0.5309553

Table VI - XVI : Average delay values for WIMAX, Ethernet and WILAN seeds for the scalability techniques over heterogeneous network- Resources Configuration-B

Technique	**WIMAX average delay (secs)**	**Ethernet average delay (secs)**	**WILAN delay average (secs)**
t+s+q	0.2591484	0.0053206	0.4309553
t+q –cif	0.342392	0.0049672	0.4909655
t+q –qcif	0.342392	0.0053326	0.4808454
t+s	0.342392	0.005323	0.4889563
SL-cif	0.342392	0.0051566	0. 4889563
SL-qcif	0.342392	0.0053438	0.4889563

6.8. Dynamic De-blocking Filter Algorithm (DDFA)

The multi-level structure of channels and applications that are enabled within a heterogeneous network cannot be limited. Hence, the scalability adaptation for video and multimedia communication needs to be dynamic in order to control and adapt to the variation of channel conditions with time. Channels performances and applications are not precisely predictable with time, and therefore algorithms that improve the adaptation based on reported the channel conditions are required.

The H.264/AVC codec is designed to operate with a de-blocking filter technology described in chapter three. The idea of the filter applications is to reduce blocking distortion [9]. Inverse transform in the encoder is performed before filter application and then macro-blocks are reconstructed and stored for subsequent inter-frame predictions. The filter is also used in the decoder before reconstructing and displaying the macro-block. The aim here is to smooth the block edges and improve the appearance of the reconstructed pictures. The filtered frames are used to generate motion-compensation vectors for subsequent video frames which can improve the compression performance as the filtered picture is better than the blocky un-filtered picture. The Filter operation techniques are defined in Table VI - XVII [4]. These techniques show variable performances depending on the operational mode as in Figure 6 - 48. In view of this, a dynamic filtering technique algorithm is designed to achieve a better flexible scalability with the time-varying network conditions. Experiments are conducted with foreman video sequence transported over heterogeneous channels. The results are shown in Figure 6 - 51 of section 6.8.3.

Table VI - XVII : Filtering Techniques Definitions, Identifiers and Used Notations.

Notation	Parameter Identifier	Definition
NFL	1	No filter is applied to blocks edges
FL	0	Filter is applied to all blocks edges
FNSL-1	2	Filter is applied to all blocks edges except slice boundaries
2-FLT	3	2 – stages de-blocking process, slice boundaries filtered in 2nd stage
LUNC	4	Luma blocks are filtered, chroma not filtered
LUNCNSL	5	Luma blocks are filtered, slice boundaries not filtered
2-LUNC	6	2 – stages de-blocking process, slice boundaries filtered in 2nd stage , no chroma

The dynamic deblocking algorithm (DDFA) is developed on the basis of real time performance of the deblocking-filter techniques. This algorithm is one component of *Scalability Decision Making Algorithm* (*SADMA*) discussed in chapter seven. It is employed in the *SADMA* functionality so that on excessive delays, a low complexity deblocking-filter stream or technique is used.

6.8.1. DDFA Procedures and Operation

Video length or resolution format representation is checked for an initial suitable filtering, for a QCIF or lower resolution video filtering is not applied. Experimental results as in Table VI - XVIII and Table VI - XIX show that a lower resolution video performed better when no filtering is applied to block edges. This is because the filter application tends to increase the number of bits in the video frames especially when the initial video has portions that are highly distorted. One way to correct this is by re-adjusting the Qp values for the filter applied regions although this might originate a small amount of complexity.

The algorithm also considers target bit-rates of an application in the selection of a suitable filtering technique. At low bit-rates, the encoder may favours the mode generating fewer bits and pays less attention to the distortion that one mode might produce. In Figure 6 - 48, the presented result shows that at low bit-rates less than 64kbit/s, all techniques are inclined to generate the same bits.

If V_{width} *&&* $V_{Len} = V_{low}$

Filter = n_filter;

Else

If V_{width} *&& VLen* $= V_{High;}$

Select_Filter;

Where V=video, n_filter= no filter applied, VLen = video vertical pixels and Select_Filter is the algorithm in Figure 6 - 47.

With a high resolution video sequence CIF or higher, filtering techniques produce bits with variation. Conditional procedures are invoked before applying a suitable filtering technique at a particular instance of the network conditions. The initial procedure of the algorithm is that a particular filter is used for a particular network based on real time simulations. If the filter is not allowed due to a change of network conditions, the algorithm is re-directed to *Select_Filter* part of the algorithm. The selection of a filter (*Select_Filter)* is based on the filter that shows better adaptation characteristics from the experimental and simulation results. A filter with low delay and better rate distortion performance characteristics is selected to improve the adaptation to bad channel reports. However, the smoothing effects are not only dependent on the filtering technique but also on the image content.

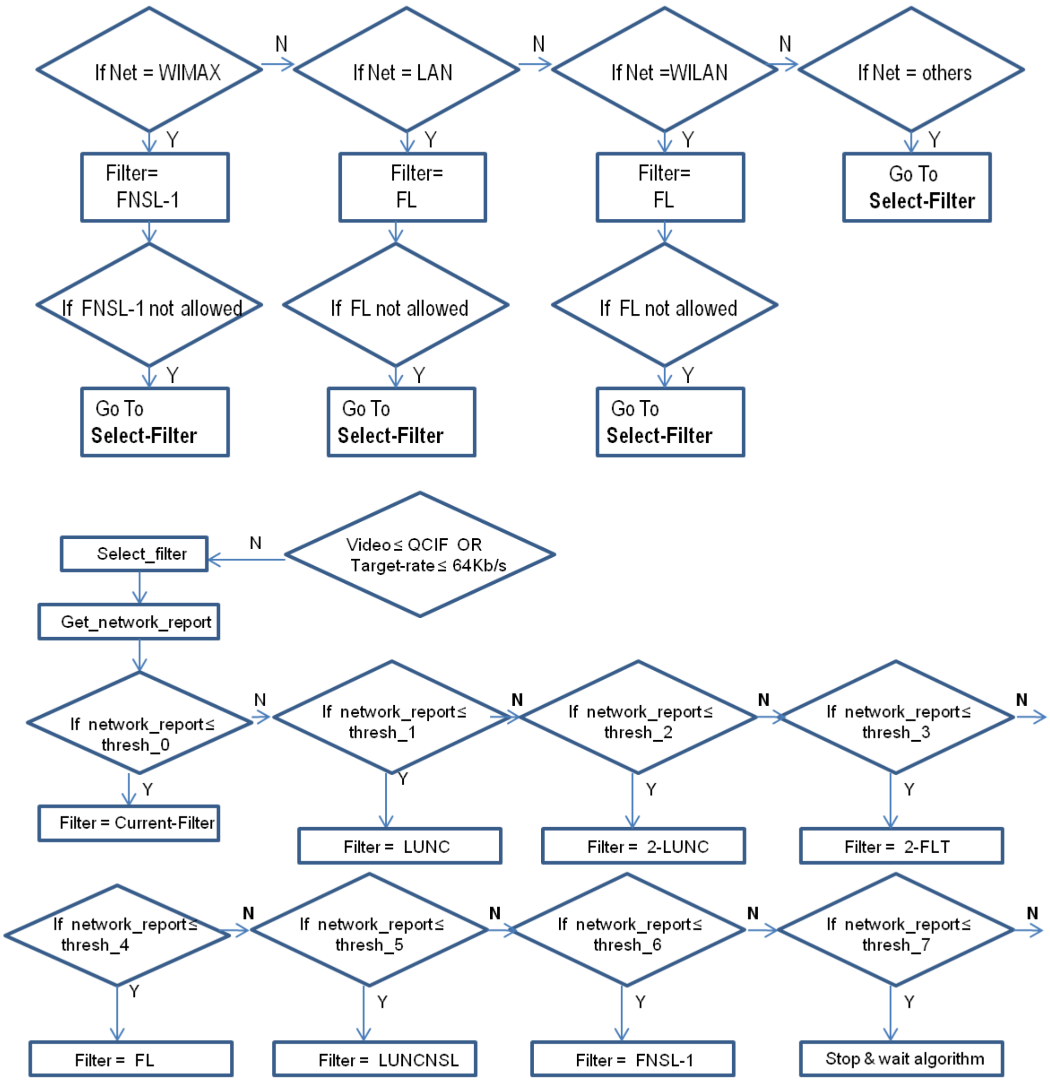


Figure 6 - 47 : Dynamic De-blocking Filtering Algorithm Flow-chart

Table VI - XVIII : De-blocking Filter Techniques: Objective Quality and Adaptation Performances- foreman QCIF

Filter-Application Technique	Bit-rates(kbit/s)	PSNR(dB)	Quantisation	Bit-stream size(KB)
Applied to all block Edges(1)	44.29 55.98	35.23 36.46	32.78 28.62	70 98
Filter not used (2)	35.53 49.84	31.06 37.25	35.53 26.90	47.7 64.7
Filter is applied Except slice boundaries (3)	35.53 49.62	35.73 37.37	31.04 26.88	47.5 64.5
Slice boundaries Filtered in 2nd stage (2-stage de-blocking Process)(4)	58.32 71.00	34.57 35.73	35.19 31.03	39.2 47.5
Chroma is not Filtered(all luma Blocks filtered)(5)	58.32 71.00	34.57 35.73	35.19 31.03	39.4 47.6
Luma blocks edges Filtered except slice Boundaries and chroma(6)	58.38 71.07	34.57 35.73	35.20 31.04	39.4 47.7
Slice boundaries are Filtered in 2nd stage, Chroma is not filtered (2 –stage de-blocking process)(7)	58.32 71.00	34.57 35.73	35.19 31.03	39.4 47.6

Table VI - XIX : De-blocking Filter Techniques: Objective Quality and Adaptation Performances- foreman CIF sequence

Filter-Application Technique	**Bit-rates(kbit/s)**	**PSNR(dB)**	**Quantisation**	**Bit-stream size(KB)**
Applied to all block Edges (1)	78.23 117.10	31.78 33.99	44.57 40.41	117 163
Filter not used (2)	79.59 118.08	31.55 33.79	44.60 40.44	118 164
Filter is applied Except slice boundaries (3)	78.64 117.03	31.77 33.99	44.58 40.42	117 163
Slice boundaries Filtered in 2nd stage (2-stage de-blocking Process)(4)	78.34 117.20	31.78 34.00	44.57 40.41	116 163
Chroma is not Filtered(all luma Blocks filtered)(5)	78.34 117.20	31.78 33.99	44.57 40.41	117 164
Luma blocks edges Filtered except slice Boundaries and chroma (6)	78.64 117.03	31.77 33.99	44.58 40.42	117 163
Slice boundaries are Filtered in 2nd stage, Chroma is not filtered (2 –stage de-blocking process)(7)	78.34 117.20	31.78 33.99	44.57 40.41	117 164

6.8.2. Objective Evaluation of De-blocking Filter Techniques

Experimental results from Figure 6 - 48 show that, the application of a filter leads to a variable performance gain depending on the applied techniques. This gain or performance effect is realised where a decoder reconstructs the image from both base and enhancement layers of the bit-stream. In cases where only the base layer is decoded, a low bit-rate is encountered and all edges around the picture show no influence over each other. The base layer is not referenced from any other layer and there is no influence from the filtering algorithm. The more number of layers decoded the more bandwidth cost required as well as block edges differences. Hence, the type of filtering technique will influence the picture distortion performance. This explains why the results obtained from Figure 6 - 48 show individual differences against base layers presented in Table VI - XVIII and Table VI - XIX.

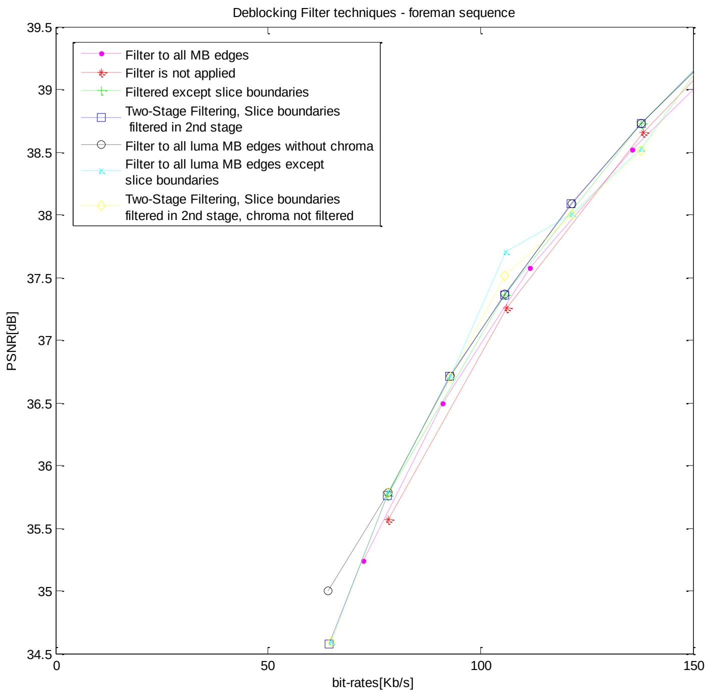


Figure 6 - 48 : Filtering Techniques Performances

Table VI - XX and Figure 6 - 49 show de-blocking filtering techniques bits characterisation. This is extracted from an experiment with foreman sequence using various deblocking filter techniques. The spent bits for frame 0–9 are extracted. The FN*SL*-1 technique shows a cheaper operation where less bits are used for most of the frames. The techniques reveal a variable performance at different bit-rates.

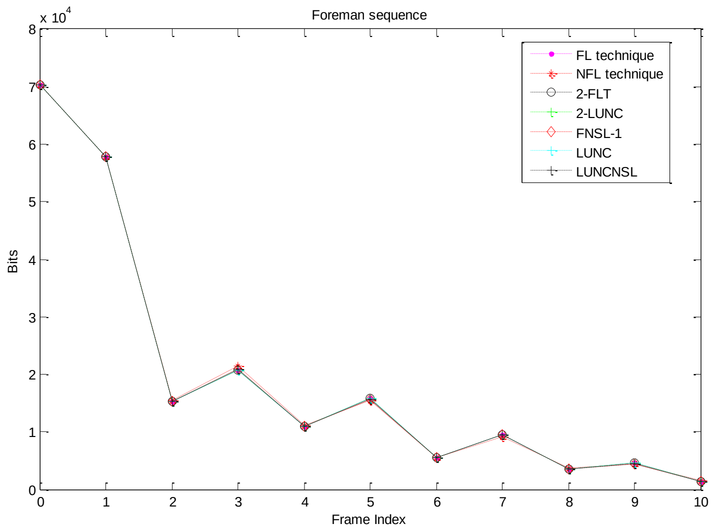


Figure 6 - 49 : De-blocking Filter Techniques Bits Characterisation

Table VI - XX : Summary of Filtering Techniques Bits Characterisation

Frame Index	FL (bits)	NFL (bits)	2-FLT (bits)	2-LUNC (bits)	FN*SL*-1 (bits)	LUNC (bits)	LUNCN*SL* (bits)
0	70264	70264	70264	70264	70272	70264	70272
1	57864	57864	57864	57864	57864	57864	57864
2	15248	15480	15256	15256	15232	15256	15232
3	20800	21460	20800	20800	20920	20800	20920
4	10992	11046	10992	10992	10992	10992	10992
5	15776	15552	15776	15776	15632	15776	15632
6	5448	5464	5448	5448	5440	5448	5440
7	9512	9120	9512	9512	9472	9512	9472
8	3464	3664	3464	3464	3456	3464	3456
9	4544	4416	4552	4552	4448	4552	4448

6.8.3. Subjective Evaluation of De-blocking Filter Techniques

An experiment to simulate a subjective variation between the techniques is conducted with foreman sequence. Figure 6 - 50 presents the subjective quality performance results for the dissimilar filtering techniques. These results show different smoothing effects of the filtering techniques. The techniques LUNC, FN*SL*-1, FL and 2FLT show a better smoothing effect. Since the research aim is to achieve efficient scalability and adaptation with a minimum quality degradation and none of the techniques shows a significant damage to the picture quality, a better performance in the objective result will be a priority.

a.

b.

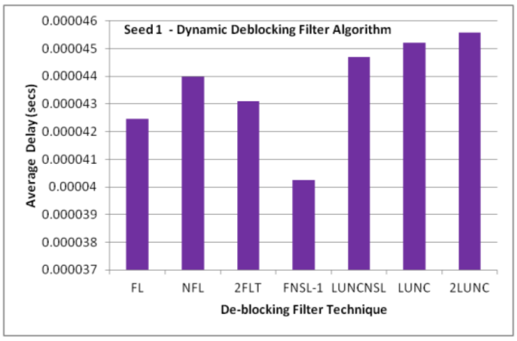
c. d.

e. f.

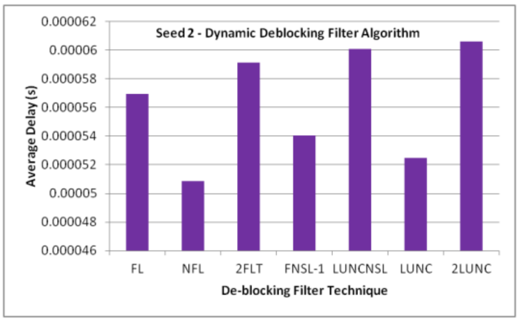
g.

Figure 6 - 50 : Subjective results for different filter techniques, frame 70 (a) 2- stage de-blocking process, slice boundaries filtered in 2nd stage (b) Luma blocks filtered, chroma not filtered (c) Deb-locking filter applied to all blocks edges except slice boundaries (d) Deb-locking filter applied to all block edges without the slice boundaries (e) deb-blocking filter is applied to all luma block edges, slice boundaries is not filtered. (f) Deb-locking filter is not applied to any of block edges (g) De-blocking filter is applied to all block edges

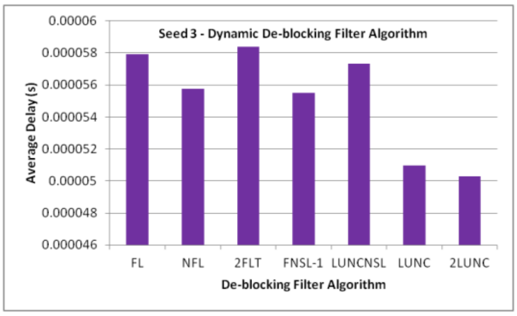
The choice of a suitable filtering technique will depend on application requirements and its resources. A limited bandwidth channel and application will utilise a technique that may not incur additional delay to image reconstruction and display. The variable filtering techniques are experimented with to simulate their real performance over heterogeneous channels and the results are presented in Figure 6 - 51.



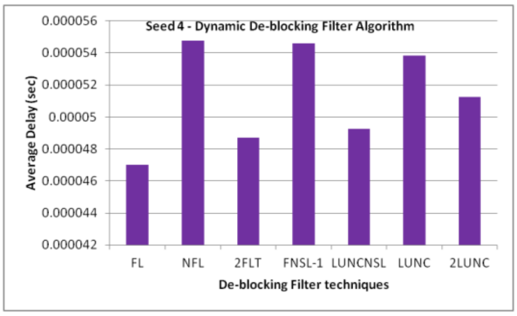
a.



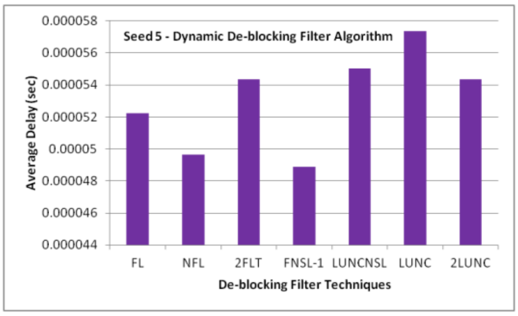
b.



c.



d.



e.

Figure 6 - 51 : De-blocking Filter Techniques: Real Time Performance Evaluation over Heterogeneous Network –for five different seeds a-e.

Table VI - XXII shows the result from real time simulation of de-blocking filters over heterogeneous channels. Alternative filters are switched on when a certain amount of delay is reported, and the new filters show an improvement in the network condition by reporting a low delay time. This shows an example of the implementation of the filtering algorithm as part of *SADMA* algorithm in chapter seven. This is used to improve the scalability adaptation over multiple networks and channel links.

Table VI - XXI shows the average delay values taking into account all seeds for WIMAX, Ethernet and WILAN networks. FN*SL*-1 technique shows lesser delay value within the WIMAX channels and FL technique within the Ethernet links followed by the FN*SL*-1 technique.

Table VI - XXI : Delay average values for WIMAX, Ethernet and WILAN seeds for the Filtering Techniques over heterogeneous networks

Filter Technique	WIMAX Average delay (secs)	Ethernet average delay (secs)	WILAN average delay (secs)
FL	0.3261738	0.0040868	0.5309553
NFL	0.3261738	0.0142586	0.5309553
2FLT	0.3261738	0.0052738	0.5309553
2LUNC	0.3261738	0.064274	0.5309553
LUNC	0.3261738	0.005196	0.5309553
LUNCN*SL*	0.3261738	0.0053284	0.5309553
FN*SL*-1	0.256892	0.0050652	0.5309553

Table VI - XXII : Simulation Results for Dynamic De-blocking Filtering Algorithm

Dynamic De-blocking Filtering Algorithm Simulation results

Current-Filter	Current-bits (kbit/s)	net_dly_report (secs)	new_filter_applied	new_bit_rate (kbit/s)	new_dly_rep (secs)	Delta-Gain_with_new _filter (kbit/s-dB)
2-FLT	58.32	0.005436×10^{-2}	FN*SL*-1	35.53	0.004888×10^{-2}	22.79-1.16
2-FLT	164.08	0.005914×10^{-2}	FN*SL*-1	131.06	0.005404×10^{-2}	33.02-2.43
LUNCN*SL*	86.05	0.005734×10^{-2}	2-LUNC	85.88	0.005031×10^{-2}	0.17-0.01
FN*SL*-1	79.00	0.005404×10^{-2}	NFL	77.76	0.00487×10^{-2}	1.24-0.17

6.9. Lambda Selection Algorithm

The encoding process described in the H.264/AVC Extension model from JVT team specifies a bottom-up procedure in which the first base layer is encoded followed by the enhancement layer [12]. Hence, the base layer is encoded without considering the impact of enhancement layer which limits the coding efficiency in the higher layers. For every access unit, the BL coding parameters are determined with the use of the widely known Lagrangian approach [13] as in (equation 6-3).

$$P_{ro} = \arg\min_{\{pro\}} D_r(p_r) + \lambda_r \, . \, R_r(p_r) \qquad (6\text{-}3)$$

However, the impact of this is not considered on the enhancement layers. $D_r(p_r)$ and $R_r(p_r)$ are the rate distortion and rate associated with selecting parameter vector P_{ro} . λ_r represents the Lagrange multiplier, which is obtained based on the chosen Q_p [14].

$$\lambda_r = c \, . \, Q_p^2 \qquad (6\text{-}4)$$

where c is a constant with a value of 0.8. Q_p represents the quantisation step size.

The lambda selection algorithm allows the value of lambda to be selected and improves the adaptation of the scalable stream. In this algorithm, the lambda value is selected using two methods or techniques. One method is to select the value of new lambda as the current-single-layer value. The second method is to improve lambda value by multiplying the current-single layer value by an integer number [14]. Table VI - XXIII shows the application of the algorithm with two layers *t+q* scalability technique. The experiment is conducted with foreman sequence. The results reveal that better adaptability and performance in the EL with lambda-new value as 0.8*lambda-current with gain up to 6 kbit/s reductions are obtained. The selection of lambda-new value as lambda-current in the BL yields a better performance with a gain up to 13 kbit/s reductions where the target bit rate exceeds 64kbit/s.

From the experimental results shown in Table VI - XXIII, the selection of λcurrent-single-layer value as the new value performed better where the spending bits equate between 32000bits and 64000 bits. Hence, the target bits generated should be considered so that for the target and operational bits less than 64000bits, lambda value should be selected as lambda-current value. The following pseudo-code summarised this conditional algorithm as follows.

If bits ≤ 64000 {

$$\lambda_{new} = \lambda_{current\text{-}single\text{-}layer\text{-}selection}$$

} else

$$\lambda_{new} = 0.8\,\lambda_{current\text{-}single\text{-}layer\text{-}selection;}$$

The experimental results presented in Figure 6 - 52 show the quality and adaptation performance for Lambda Selection Algorithm. It is shown in the results that non-usage of the algorithm results in poor performance.

Table VI - XXIII : Two methods for Lambda Value Selection (LVS) with Forman sequence

λ_{new}= 0.8 $\lambda_{current\ –single-layer-selection}$		λ_{new}=$\lambda_{current-single-layer-selection}$		λ_{new}= No selection algorithm applied	
BL	EL	BL	EL	BL	EL
bit-rates(kbit/s)/PSNR(dB)	bit-rates(kbit/s)/PSNR(dB)	bit-rates(kbit/s)/PSNR(dB)	bit-rates(kbit/s)/PSNR(dB)	bit-rates(kbit/s)/PSNR(dB)	bit-rates(kbit/s)/PSNR(dB)
37.47 / 36.08	46.52 / 36.11	35.50 / 35.73	38.80 / 35.76	47.10/33.35	51.65/33.36
50.14 / 37.51	53.06 / 37.52	49.58 / 37.37	52.76 / 37.36	70.38/35.75	75.45/35.76
63.68 / 38.69	66.41 / 38.69	65.73 / 38.71	68.79 / 38.73	90.49/37.18	94.86/37.18
76.57 / 39.57	79.63 / 39.59	78.94 / 39.61	81.97 / 39.66	108.68/38.20	112.98/38.20
89.43 / 40.35	92.54 / 40.40	93.02 / 40.39	96.28 / 40.51	125.63/39.00	129.87/39.00
100.77 / 40.92	104.12 / 41.02	105.08 / 40.9	108.94 / 41.14	142.49/39.71	146.66/39.71
113.23 / 41.47	117.01 / 41.63	119.03 / 41.49	123.33 / 41.77	156.58/40.23	160.91/40.23
123.65 / 41.91	127.63 / 42.12	126.33 / 41.80	130.97 / 42.09	172.31/40.78	176.71/40.77

The two curves under Figure 6 - 52 represent two layers generated where the algorithm is not used for CIF resolution format. This shows poor quality performance and r.d.o. From Figure 6 - 52, 36dB quality is achieved from 185kbit/s bit-rate for L2-cif with lambda-new = current-lambda*0.8 method. The same quality is achieved with the same L2-cif from 200kbit/s bit-rate and lambda-new=current-single-layer-lambda method. This quality value is obtained when the algorithm is not used at a bit-rate greater than 250kbit/s. This gives a bit-rate reduction of 15000bits for lambda-new=current-lambda*0.8 method against lambda-new=current-single-layer-lambda method. For the non-usage of the either algorithm, a bit-reduction of 75000bits and 50000bits for lambda-new=current-lambda*0.8 and lambda-new=current-single-layer-lambda is achieved against the non-usage of the algorithm respectively.

With QCIF, L0, the lambda-new=current-lambda*0.8 technique produced 36dB from a required bit-rate of 48kbit/s while lambda-new=current-single-layer-lambda and non-usage of the algorithm techniques generated this quality value from a required bit-rate of 52kbit/s and 56kbit/s respectively. Hence, the usage of lambda-new=current-lambda*0.8 is more valuable.

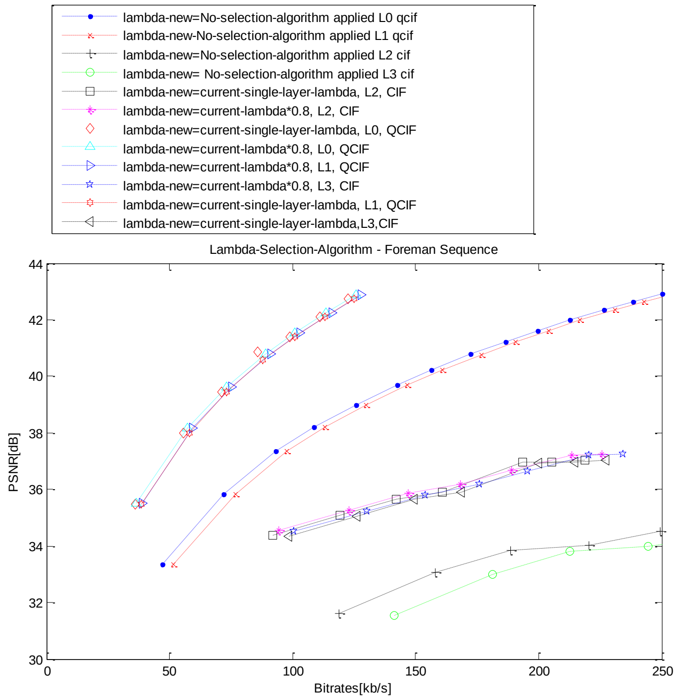


Figure 6 - 52 : Multi-Layers SVC performance with Lambda Selection Algorithm using two lambda selection methods and non-usage of the methods- Foreman sequence

6.10. Inter-Layer Prediction

With the SVC having a layered structure, the existing layers could be used to reference the motion values for other layers. This technique generates a higher gain with SNR scalability where the layers show same similarity. An increased gain in rate-distortion optimisation is obtained from predicting the higher layers from the lower layers. The base layer with id equal LayerId0 is set with no prediction but as a reference layer for the higher layers. An experiment is conducted with foreman standard sequence using *t+q* scalability and the result is shown in Figure 6 - 53. A bit-rate of 47.10kbit/s and 28.63kbit/s generates 33.35dB and 36.67dB with no-interlayer prediction and with interlayer prediction respectively. Similarly, a bit-rate of 70.38kbit/s and 43.83kbit/s produces 35.75dB and 38.71dB with no-interlayer prediction and with interlayer prediction respectively.

With the inter-layer motion prediction, all MBs of the video sequence are predicted from the corresponding base layer. This means the MBs modes, the reference indices and motion compensation vectors are always copied from the base layer being the reference layer. Experiments revealed that a reduction of about 11% of bit-stream size is obtained with inter-layer prediction technique. Another technique is to arbitrarily select inter-layer prediction through a rate-distortion optimisation algorithm. This shows slightly better efficiency as shown in Figure 6 - 53 results. An efficiency of 0.01kbit/s is shown in the lower rates values of the experiment with foreman sequence. However, the usage of this method is limited due to the constraint of the maximum number of motion vectors that could be used for a picture. In this case, where a maximum number of required motion vectors are exceeded, this technique will not be used for inter-layer prediction without MBs adaptation.

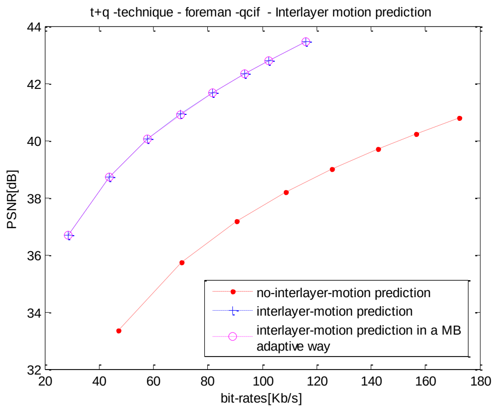


Figure 6 - 53: Rate-distortion performance of interlayer-motion prediction technique -foreman

6.11. Summary and Conclusion

In this chapter, new scalability techniques are developed and evaluated. The evaluation includes objective and subjective performance assessment. Real time simulations over the built heterogeneous network simulator discussed in chapter four are also conducted to evaluate the techniques real time performances. The level of performance of the techniques against existing and non-scalable techniques is also described.

As part of scalability adaptation improvement, three algorithmic schemes are introduced. These schemes include dynamic de-blocking filtering, the adaptation of absolute possible inter-layer dependency and prediction and a scheme which supports lambda value selection based on current computed bit-rate and two methods. An improved *t+s* technique is also introduced which supports high scalability and adaptation level especially where bandwidth and applications resources are extremely limited.

In the next chapter, the proposed *Scalability Decision Making Algorithm* (*SADMA*) is discussed. The implementation of the inputs in chapter six, five and four within the algorithm are described. The *SADMA* algorithm employed an absolute usage of the scalability adaptation techniques and algorithms introduced in this work to support a dynamic and flexible scalability within a heterogeneous network environment.

References

[1] Schwarz, H., Marpe, D. and Wiegand, T. (2007) "Overview of the scalable video coding extension of the H. 264/AVC standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1103-1120.

[2] Ghanbari, M. (2003) "Standard Codecs: Image compression to advanced video coding", Institution of Electrical Engineers, .

[3] Li, X., Amon, P., Hutter, A. and Kaup, A. (2009) "One-pass multi-layer rate-distortion optimization for quality scalable video coding", *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*IEEE, , pp. 637.

[4] Wiegand, T., Sullivan, G.J., Bjontegaard, G. and Luthra, A. (2003) "Overview of the H. 264/AVC video coding standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, no. 7, pp. 560-576.

[5] Symes, P.D. (2003) *Digital video compression,* McGraw-Hill/TAB Electronics.

[6] Chang, S.F. and Vetro, A. (2005) "Video adaptation: concepts, technologies, and open issues", *Proceedings of the IEEE,* vol. 93, no. 1, pp. 148-158.

[7] Liu, Y., Li, Z.G. and Soh, Y.C. (2008) "Rate control of H. 264/AVC scalable extension", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 18, no. 1, pp. 116-121.

[8] Sohn, H., Yoo, H., Kim, C.S., De Neve, W. and Ro, Y.M. (2009) "Perceptual quality measurement for scalable video at low spatial resolution in mobile environments", *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 20.

[9] Richardson, I.E.G. "H. 264 and MPEG-4 video compression. 2003", *John Wiley&Sons Ltd., New Jersey,* .

[10] Ohm, J.R., Koenen, R. and Chiariglione, L. "SVC requirements specified by MPEG (ISO/IEC JTC1 SC29 WG11)", .

[11] http://ftp3.itu.chlar-arch/jvt-site/bitstream-exchange/ - Accessed December, 2009.

[12] Schwarz, H. and Wiegand, T. (2007) "RD optimized multi-layer encoder control for SVC", *Image Processing, 2007. ICIP 2007. IEEE International Conference on*IEEE, , pp. II.

[13] Li, X., Amon, P., Hutter, A. and Kaup, A. (2009) "Lagrange multiplier selection for rate-distortion optimization in SVC", *Picture Coding Symposium, 2009. PCS 2009*IEEE, , pp. 1.

[14] Pang, C., Au, O.C., Dai, J. and Zou, F. (2011) "Frame Complexity Guided Lagrange Multiplier Selection for H. 264 Intra-Frame Coding", *Signal Processing Letters, IEEE,* vol. 18, no. 12, pp. 733-736.

[15] ftp://vqeg.its.bldrdoc.gov/SDTV/VQEG_Phase1/TestSequences/ALL_625/ - Access December,2009.

[16] http://ftp3.itu.ch/av-arch/jvt-site/draft_conformance/ - Accessed December, 2009.

[17] Team, J.V. (2009) "H. 264/SVC reference software (JSVM 9.18) and manual", *CVS sever at garcon.ient.rwth-aachen.de,*

CHAPTER SEVEN

7. DYNAMIC SCALABILITY AND ADAPTATION ALGORITHMS AND SCHEMES

7.1. Introduction

The current technological advancements in multimedia communication services pose many challenges to provide a guaranteed Quality of Service (QoS) due to limited resources and dynamic network and applications requirements. The fluctuation of network channels including wireless, LAN network resources, internet which originates errors, data loss and poor quality picture necessitated a dynamic and congestion preventive scalability approach.

In this chapter, dynamic adaptation and scalability algorithms are proposed. The algorithms and schemes are a combination of procedural tasks that are executed based on reported channel conditions. The algorithms used various techniques and experimental facts in chapter five and six to provide a network-aware video communication service. The transcoding techniques, interfaces and methods for switching within the techniques/schemes and extracting of sub-streams are described.

7.2. Network-Aware Scheme and Previous Work

The implementation of the network-aware scheme is stimulated by the following facts (1) the network channels delay can be very high in a heterogeneous channel and thereby causing excessive bit-error rate (BER) and packet loss (2) In excessive BER, packet loss is unavoidable which results in poor picture quality (3) real time applications have limited tolerable delays (4) The loss of video packets is unavoidable when the available bandwidth is less than required (5) rapid advancement in new technologies demands better approach for effective video adaptation. If video streams are transmitted with no awareness of network condition, transmitted layers may get corrupted resulting in poor image quality. Hence, we proposed *network-aware scheme* to address these issues which intelligently select a suitable bit-stream or sub-stream based on the reported network status. A suitable bit-stream can be one of the set of scalable techniques developed and evaluated in chapter five, a reduced layered stream of the techniques, a technique with better efficient parameter identifiers, a base layer (high priority layer) or/and lower enhancement layer. The algorithm can drop all the enhancement layers EL passing into the network a base layer only. A base layer BL is the highest priority layer which all other layers reference. It can deliver the video packet at a cheaper rate with less quality. A BL and one of the EL's can also be used depending on the reported network condition. The dropping order is from the highest enhancement layer down to the base layer.

Our proposed algorithm (*SADMA*) is unique from other scalability and adaptation work conducted in [1] and [2] for adaptive service and adaptive framework respectively. Research work has also been conducted in adaptive services for wireless networks [3] and scalable video over wireless IP networks in [4]. The authors achieved significant contributions in approaches for adaptation services and their framework. [4] has described scalable adaptation over wireless IP networks. However, the flexibility of the adaptation is limited to only one scalable stream(BL) as a solution to network deterioration and the system is functional within single network medium only. In providing remedies to bad channel conditions, the use of base layer is limited as last solution. Also, the adopted scalability is not evaluated for user quality satisfaction. Conclusively, *SADMA* is unique from the previous works in the following aspects (1) It is designed and evaluated to provide scalability within heterogeneous networks (2) The flexibility of the scalability involves usage of variable scalable techniques and bit-streams based on network tolerance. (3) The Algorithm is designed to be network-aware and therefore provides a preventive defence against the consequences of excessive network congestion (4) An end to end evaluation and assessment of the scalability techniques and algorithm for guaranteed QoS is conducted.

7.3. Adaptation Algorithm (*SADMA*) Service Requirements

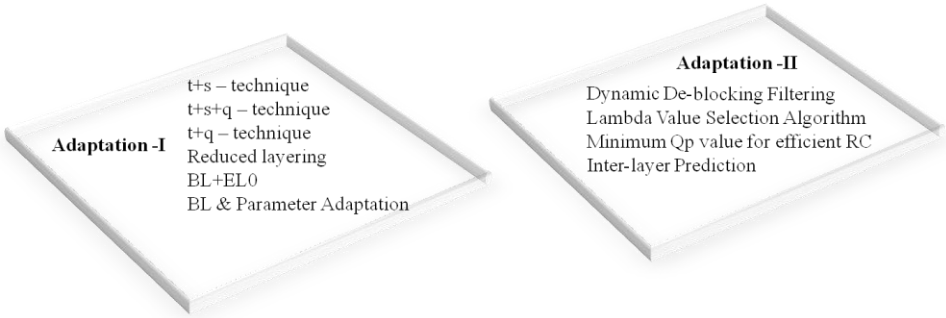
The algorithm will operate based on a *network-aware service* delivering preventive services against predicted network fluctuations. The algorithm usage of a particular video stream is decided from several experimental analysis and real time simulations. The algorithm procedures are executed at all instances of new network conditions. These procedures are shown in Figure 7 - 4 in a flowchart. The established conditions for switching between techniques are described in TABLE VII - 1.

At a video source encoder, variable bit-streams of different bit-rates, frame rates and quality are produced. These streams hold different priorities into the network. For example, the BL which has the highest priority is used when the network conditions do not tolerate the usage of the BL + lowest enhancement layer (EL0). If the network status does not accept the usage of the base layer, then the stream uses parameter identifiers with high efficiency to improve the scalability and adaptation of the video service. TABLE VII - 1 defined the requirement conditions for transcoding from a bit-stream to a different one. The conditions are established from experiments and real time simulations. These experiments and simulations are covered in chapters five and six.

TABLE VII - 1 : Summary of transcoding conditions for various techniques and bit-streams (*Adaptation I*)

Transcoded-to-technique	**Transcoding conditions**
$t+s+q$	WIMAX network high rate
$t+s+q$	LAN network high rate
$t+s+q$	WILAN all rates
$t+s$	WIMAX network low rate
$t+s$	current delay is high (defined value)
$t+s+q$	delay ≥ $t+s$ can tolerate
$t+s+q$+ parameter adaptation	delay ≥ $t+s+q$ can tolerate
$t+s+q$	delay ≥ $t+q$ can tolerate
$t+q$+ parameter adaptation	delay ≥ $t+q$ can tolerate
BL+EL0	delay ≥ Use of parameter adaptation on techniques can tolerate
BL+EL0+ parameter adaptation	delay ≥ BL+EL0 can tolerate
BL	delay ≥ BL+EL0 can tolerate
BL+ parameter. adaptation	delay≥ BL can tolerate
Im-$t+s$-BL	delay ≥ BL+ parameter. Adaptation can tolerate

The algorithm responds to reported network conditions in two defined stages namely *Adaptation I* and *Adaptation II*. Each of the Adaptations consists of a number of algorithms and schemes as presented in Figure 7 - 1. The algorithm (*SADMA*) invokes *Adaptation II* option when all the techniques and schemes in *Adaptation I* presented in TABLE VII - 1 cannot adapt to the load of the current network conditions. The introduction of *Adaptation II* option is to further improve the robustness of the algorithm. This is in the situation when network conditions are highly deteriorated. The un-predictable nature of networks also motivated for introducing *Adaptation II* option in Figure 7 - 1 (b). The implementation of *Adaptation I* prior to *Adaptation II* will reduce the complexity of the system. The *Adaptation II* schemes additionally and intelligently make decisions based on the video format and current bandwidth after considering network status report. In addition to this, when the network conditions can be controlled from *Adaptation I* algorithm, it is important that bandwidth is utilised efficiently to achieve a maximum perceptual quality.



a. b.

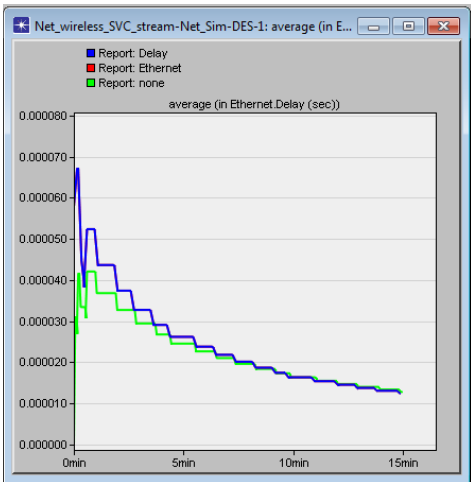
Figure 7 - 1 : (a) *Adaptation I* techniques and scheme. (b) *Adaptation II* techniques and scheme.

7.4. Monitoring of Channels Conditions

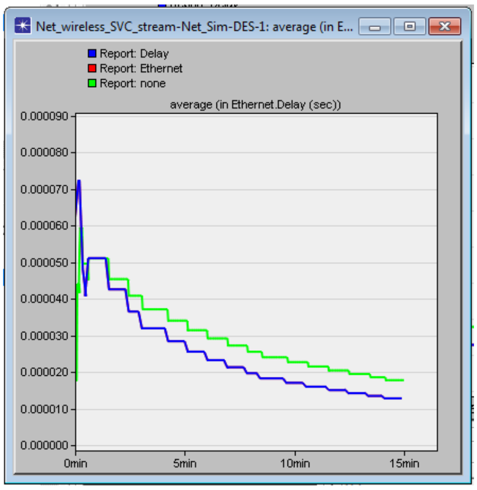
The designed algorithm (*SADMA*) uses delay as a major factor to determine the condition of the network channels. Delay is a one induced feature within networks that is inevitable [5,6] and is measured in seconds. This amount of time (delay) may have a tolerable limit for certain applications. The variation of delay value depends on the operational channel factors and characteristics. These factors can be external or surrounding conditions, applications characteristics or new demands. Other factors that could be used to predict channel conditions include bit-error rate and packet loss ratio. However, for a scalable stream this may not be observed for all stations at most times. Delay is inevitable even in error free communications. The amount of incurred delay during the video services could be acceptable or inacceptable depending on the application requirements. Generally, longer delays are not acceptable for real time video services.

To provide flexibility in the algorithm operations, *pre-configured values* are provided by the user. This will allow the algorithm to decide the level of scalability that will be required for a particular predicted network condition.

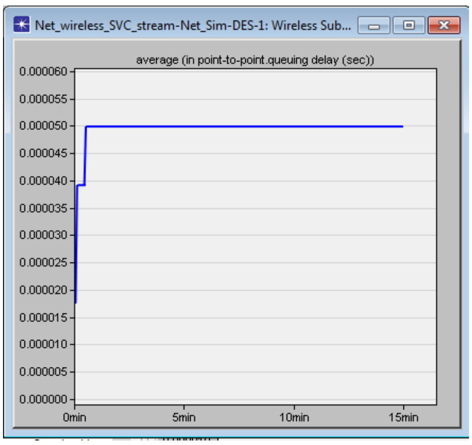
The simulation collects 100 values of sample mean per statistic. The collected statistic represents the end-to-end delay of all packets received by all the stations. The method of collection is referred to as bucket mode sample mean. The frequency of collection when the packets are being transmitted on the network impact on the bandwidth requirement. When the network report is collected or processed within a shorter time interval, the algorithm will generate a suitable scalable stream based on the timely network report. Hence, it will be more efficient for the cost of additional bandwidth. However, a longer time of report collection requires less bandwidth with the scalability adaptation being less efficient. It is therefore a trade-off between efficiency and bandwidth utilisation. Two experiments were conducted where two and four seconds of report collection interval are used. The result in Figure 7 - 2 shows that the shorter interval will spend more bandwidth than the longer interval method of channel condition processing. Figure 7 - 2 (a) and (c) shows a maximum delay of < 0.000070 second and 0.000050 second while Figure 7 - 2 (b) and (d) shows a maximum value of > 0.000070 second and 0.000055 second. This is achieved for four seconds and two seconds report collection method respectively. The experiments were done for the same network and channel configurations in both cases within fifteen minutes of the communication time.



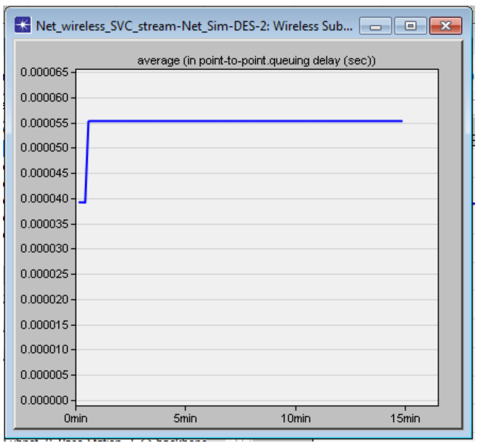
a.



b.



c.



d.

Figure 7 - 2 : Channels conditions collection methods (a) Longer: End to End delay for 4 seconds and (c) Longer: Queuing delays for 4 seconds interval. (b)Shorter End to End delay for 2 seconds and (d) Shorter: Queuing delays for 2 seconds interval.

However, the decision for a suitable method can be selected based on the known channel performance, available bandwidth and specific applications requirements.

7.5. Dynamic Scalability Decision Making Algorithm (*SADMA*)

The techniques designed in chapter six provide an automatic scalability in real and non-real time video services for variable channels characteristics. However, network channels possess unpredictable variations with time. Network terminals and applications decoding bit-rates and memory capacity dynamically do changes and demands can be brought from new clients etc. Hence, this can limit the performance adaptation for a particular adopted technique. The consequences will be delay which leads to bit-errors and packet loss and then poor quality picture delivery. To avoid and control these situations, a *Dynamic Scalability Decision Making Algorithm* (*SADMA*) is developed. The *SADMA* performs adaptation functions from two adaptation scenarios namely *Adaptation I* and *Adaptation II* as shown in Figure 7 - 1 (a) and (b). The provision of the two schemes and their procedural implementation is to reduce complexity and maximise perceptual quality when the network condition is improved as described in section 7.3. TABLE VII - 1 shows the set conditions required for transcoding from a bit-stream to another with *Adaptation I* schemes. With *Adaptation II*, the schemes are all applied to improve the highly deteriorated network conditions.

Figure 7 - 3 is a block diagram showing *SADMA* algorithm architecture. The Algorithm regularly monitors network conditions fluctuations. It processes the predicted channel status on a set frequency. The frequency is defined based on application requirements as described in section 7.4. The Algorithm then compares sets of pre-defined delay thresholds values with the reported one.

The algorithm autonomously selects one of the designed scalability techniques basing its decision on the monitored and reported channel conditions. Experiments were conducted using a purpose-built heterogeneous network simulator and the network-aware selection of the scalability techniques is based on real time simulation results. A technique with a minimum delay, low bit-rate, low frame rate and low quality is adopted as a reactive measure to a predicted bad channel condition. If the use of the techniques is not favoured due to deteriorating channel conditions reported, a reduced layered stream (BL+EL)) or base layer is used. If the network status does not allow the use of the base layer, then the stream uses parameter identifiers with high efficiency to improve the scalability and adaptation of the video service. In all the deteriorated channel conditions requiring transcoding between bit-streams, the selected technique should be of low bit-rate, frame rate and/or quality.

In a situation where excessive network deterioration is predicted and *Adaptation I* algorithms and schemes are not tolerated, *Adaptation II* is invoked. *Adaptation II* in Figure 7 - 1 (b) supports the *Dynamic Filtering*, *Constraint Qp Algorithm*, *Lambda Value Selection Algorithm* for better r.d.o. performance and adaptation. *Interlayer-Prediction* is also provided to reduce the spatial and temporal redundancies for more effective scalability. The sequential hierarchy for executing the *Adaptation II* algorithms depends on the schemes and algorithms performance. The performances of the algorithms and schemes are discussed and presented in this chapter and chapter five and six. Since the adaptation is re-directed to *Adaptation II* from *Adaptation I*, it is most likely that the network condition is worsened and therefore high performance algorithm is first used from *Adaptation II*. The conditions for the algorithms usage are summarised in TABLE VII - 1.

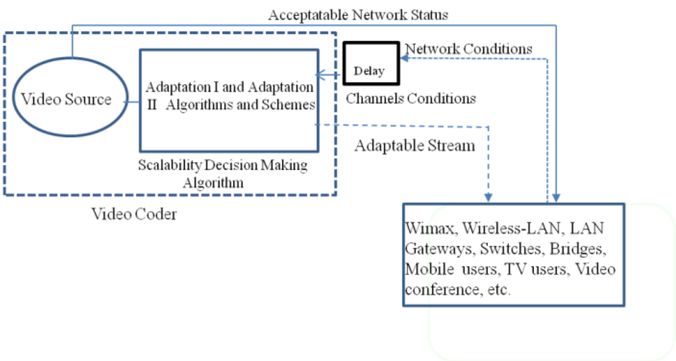
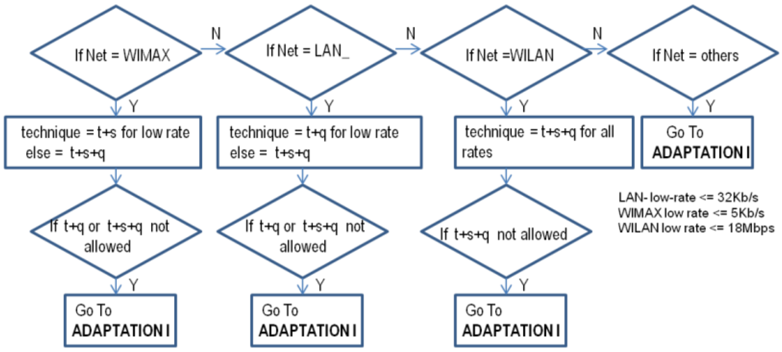


Figure 7 - 3 : Scalability Decision Making Algorithm (*SADMA*) Block Diagram

Figure 7 - 4 is a flowchart showing the algorithm procedures for the selection of a suitable scaling technique. The selection is made after processing the network status report. The status report is predicted by recording the time packets are sent and the time response is received. Delay is determined from the difference of the two times. The status report could also be processed from BER of the video packets being sent and a packet loss ratio can then

be calculated. A threshold is set which can act as a measure to predict a future bad network condition. An acceptable threshold is normally predetermined for various applications. The value of the threshold is designed to be a parameter that could be re-configured in the codec system for a dynamic and flexible design.

The Flowchart is divided in three procedural parts. In the first procedural part, the algorithm uses the best technique for the current network. This is determined from real time simulations presented in section 6.7. If the technique is not allowed in the current network due to additional demands then the algorithm switches to *Adaptation I* procedure (second procedure). The channel condition is monitored and determined in all the procedures. The second procedure (*Adaptation I*) supports the selection of techniques and bit-stream based on the predicted network condition. This is shown in the flow of *Adaptation I* schemes procedures (Figure 7 - 4). The third procedural part is where the algorithm re-directs for *Adaptation II* scheme to be processed when the channel conditions are extremely bad. The *Adaptation II* scheme is described in Figure 7 - 9. Parameter adaptation is described in Figure 7 - 11 as part of *Adaptation I*.



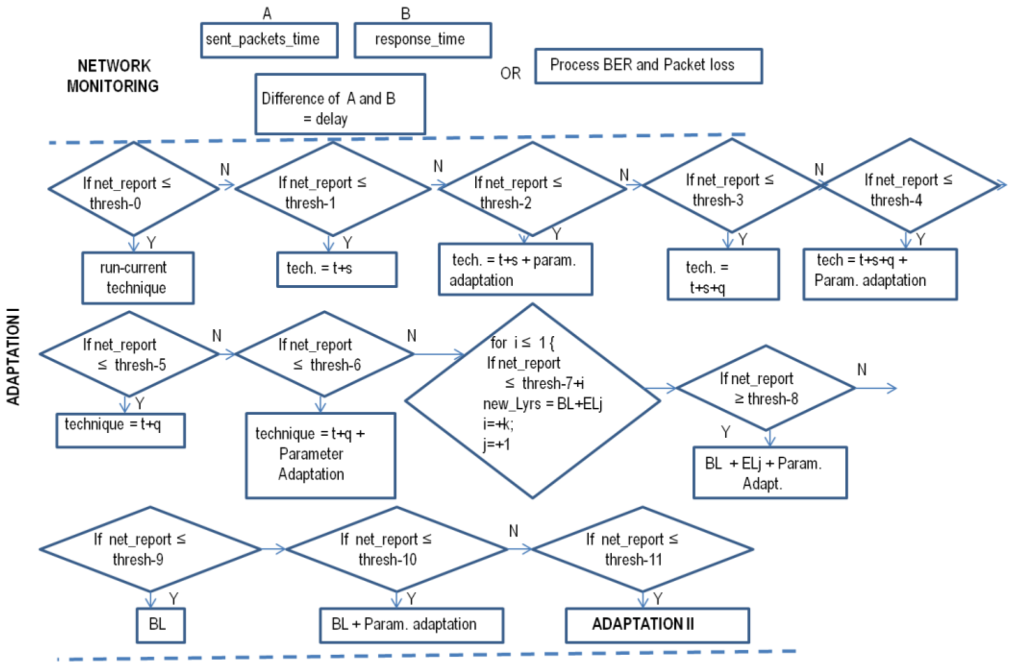


Figure 7 - 4 : Scalability Decision Making Algorithm (*SADMA*) Flow Chart illustrating the Algorithm Logical and Asserted Procedures

7.5.1. Different Priority Layers Extraction

The developed scalability techniques in chapter six produce a scalable bit-stream consisting of variable layers of different bit-rates, frame rates and quality. The layers exhibit different tolerance characteristics by the network. Figure 7 - 5 shows 4 scalable layers generated from $t+s+q$ technique using standard city video Sequence. The layers present different scalability adaptation to network channels. For instance, L0, L1, L2 and LN can be allowed for a network report R0, R1, R2 and RN. An interface for sub-stream (Layer) extraction is defined in section 7.10.

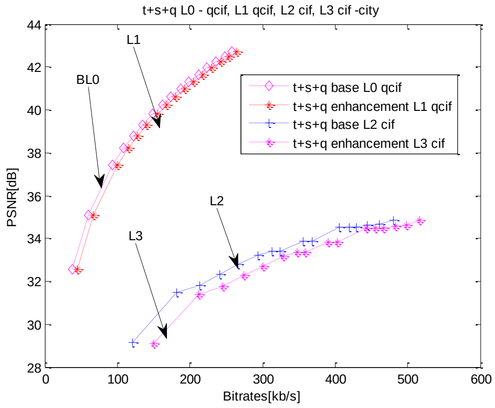
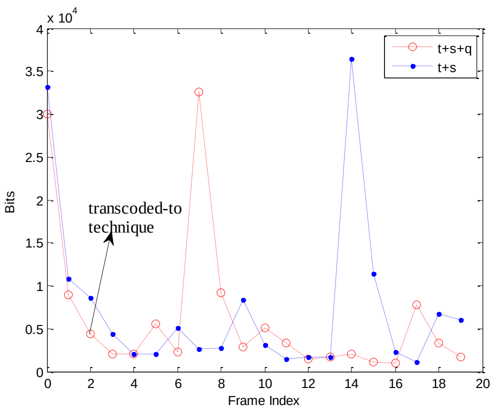


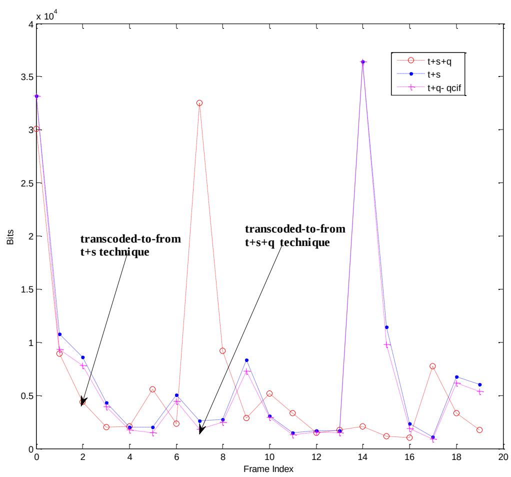
Figure 7 - 5 : *t+s+q* - four scalability layers generated with variable adaptation capability

7.5.2. Results and Discussions for *Adaptation I* option

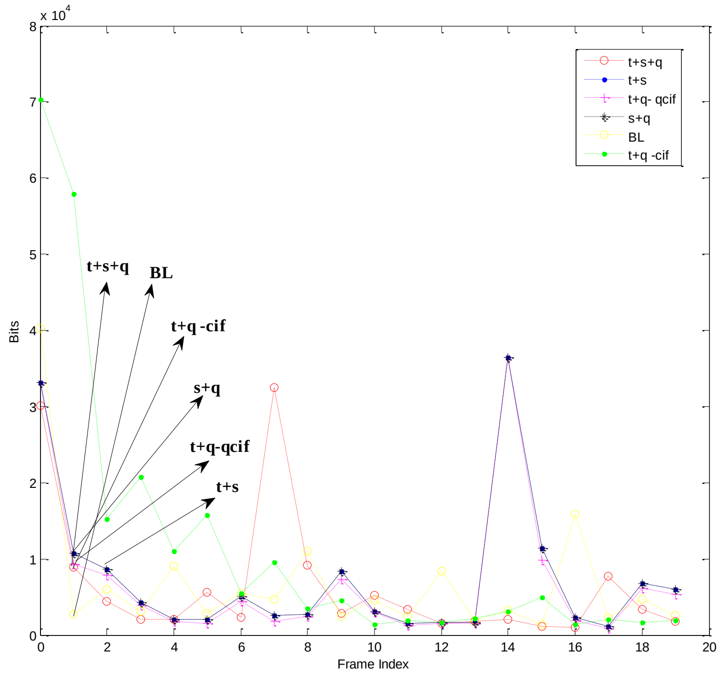
The usage of *Adaptation I* option involve the developed techniques in chapter six, parameter adaptation algorithm, BL usage or BL+ any of the EL. Figure 7 - 6 (a, b, c) show the bits characterisation for *t+s+q*, *t+q* and *t+s* scalability techniques. There is a variation of coded and spent bits from one technique to another. In Figure 7 - 6 (a), the produced *t+s+q* bits from *t+s+q* technique are less than those of *t+s* technique in most of the frames. This indicated a higher bandwidth requirement for *t+s* than *t+s+q*. Although it is also noticed that some *t+s* frames produce less bits but experiments reveal that *t+s* objective result does not actually reflect the subjective result. The *t+s+q* technique shows better picture quality at the same target bit-rate.



a.



b.

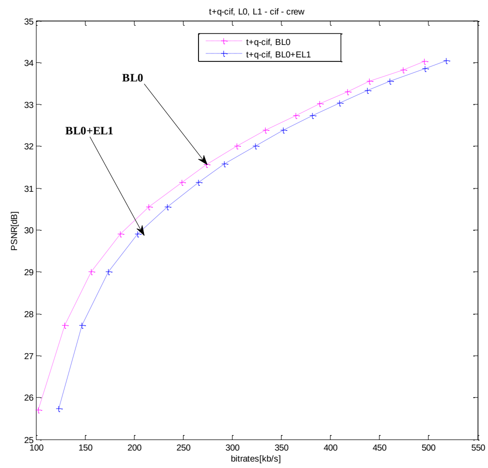


c.

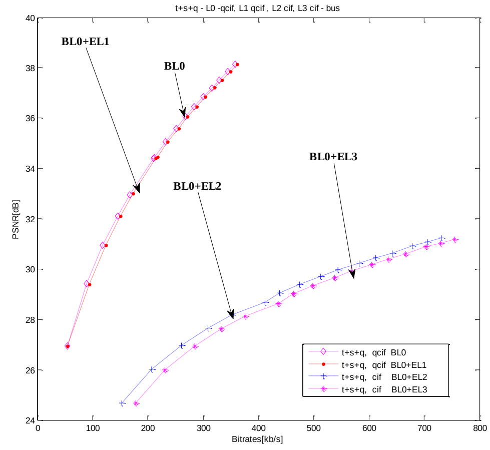
Figure 7 - 6 : Bits characterisation with the transcoded bit-stream and techniques

Hence, less bandwidth will be required for *t+s+q* than *t+s* to produce a good quality picture. In Figure 7 - 6 (b), *t+q* technique is cheaper than *t+s+q* technique and therefore is broadcast into the network when the current network condition cannot tolerate *t+s+q*.

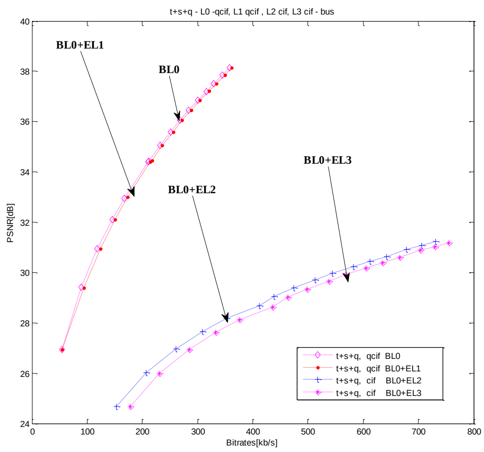
Figure 7 - 7 (a-f) shows the experimental results conducted with a number of ITU standard sequences. The graphical representation of the results shows the purpose of transcoding a BL+EL to a BL. The transcoding provides a better adaptable stream.



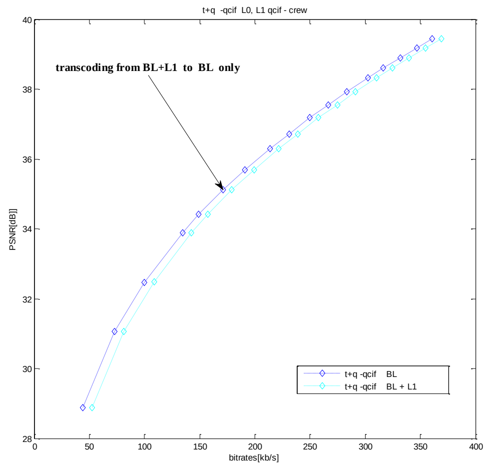
a.



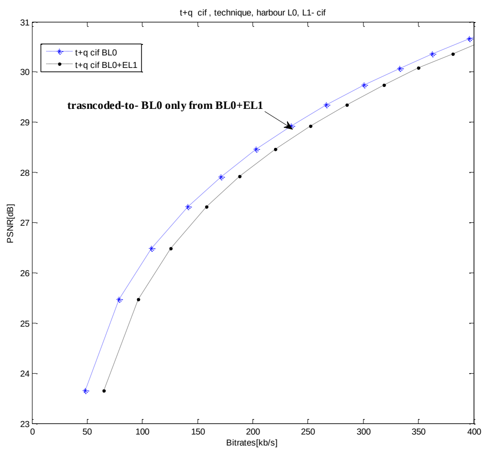
b.



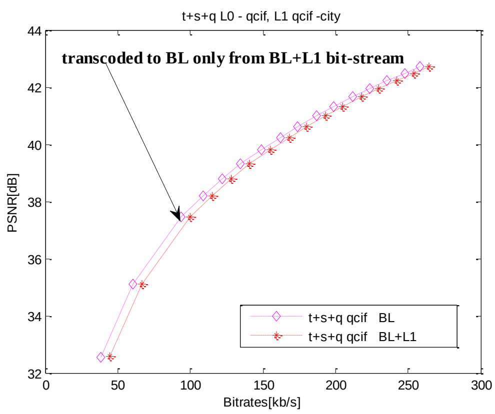
c.



d.



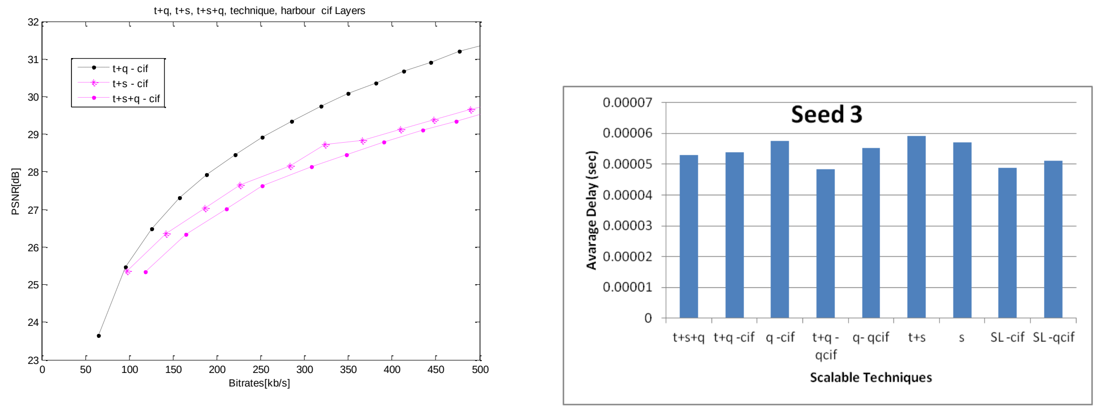
e.



f.

Figure 7 - 7 : Performance of SVC techniques for transcoding between bit-stream

Figure 7 - 8 (a) shows the experimental result for the bit-stream objective performance of the svc techniques (*t+s+q*, *t+q* and *t+s*). *t+q* indicated a better rate distortion performance. Hence, it has better adaptation characteristics. For example, to generate PSNR value of 27dB, 221kbit/s, 200kbit/s and 150kbit/s are required for *t+s+q*, *t+s* and *t+q* respectively. Figure 7 - 8 (b) is a result of the real time simulation of the algorithm over heterogeneous network. The simulation includes the developed techniques, current techniques and non-scalable layer. The delay values are sampled after every 4secs of transmitting with a bit-rate of 250kbit/s. A reduced delay is experienced with *t+q*, *t+s+q* better than *t+s*. The *SL* average delay is low which made it a last option when the techniques are not allowed due to deteriorated or high delay channels.



a. b.

Figure 7 - 8 : **Left-** bit-stream objective quality performance **Righ**t- SVC techniques real time simulation performance

The real time simulations results for the scalability techniques are presented in section 6.7

7.6. Efficient Parameter Identifiers for Dynamic Scalability and Video Service Adaptation.

Network Abstraction Layer (NAL) units are grouped into Video Coding Layer (VCL) and non-VCL units. The VCL units contain data that represents the values of the samples in the video frames, and the non-VCL NAL units include any associated extra information such as parameter sets (important header data that can be relevant to a large number of VCL NAL units) and supplemental enhancement information (timing information and other supplemental data that may enhance the usability of the decoded video signal but are not necessary for decoding the values of the samples in the video image [7].

7.6.1. Parameter Sets Design and Structure

The design of the parameter sets of H.264/AVC provides for robust and efficient conveyance for header information. The decoding and reconstruction process could have a severe impact when few key bits are lost (e.g. picture header information or sequence header) when using prior standards. H.264/AVC offers a more flexible manner of handling this information [8]. This supports and offers the decoding of a large number of VCL NAL units. There are two designed types of parameter sets:

1. Sequence parameters sets, which apply to a series of consecutive coded video pictures referred to as a coded video sequence.

2. Picture parameter sets, which apply to the decoding of one or more individual pictures within a coded video sequence.

Each VCL NAL unit contains an identifier that refers to the content of the relevant sequence parameter set. In this design, a small amount of data (the parameter identifier) can be used to refer to a larger amount of information (the parameter set) without repeating that information within each VCL NAL unit.

Sequence and picture parameter sets can be sent well ahead of the VCL units that they are relevant to, and can be repeated to offer a robust scalability and adaptation against data loss. In some applications, parameter sets may be sent within the channel that carries the VCL NAL units termed in-band transmission. In other applications, it can be profitable to convey the parameter sets out-of-bound using a more reliable transport technology than the video channel itself as discussed in section 2.3.4.

In our proposed application, the designed algorithm will present the service for the exchange of parameters when this is allowed from the reported network conditions and the defined logical conditions in Figure 7 - 4 and TABLE VII - 1. The selection of parameter identifiers is determined from the number of experiments and real time simulations presented in chapter five. The algorithm exchanges parameter identifiers to improve current bit-stream for better adaptation to the network channels. The parameters are a set of identifiers which can refer to the content of an applicable sequence parameter set. The presented algorithm in *Adaptation II* does not only provide adaptation based on network condition but also on the current image format and bandwidth of transmission. The arguments for the parameter identifier algorithm are facts from several experimental and real time simulation results. The experiments were conducted with many standard ITU video sequences. All the experimental results and simulations are documented and described in chapters five and six. Several sequences are used in the experiments to reflect the video content different characteristics.

When using earlier standards design, the loss of information (for example sequence header or picture header bits from the prior standards) could create a severe negative impact on the reconstruction process. This key information was separated for handling in a more flexible and specialised manner in the design of H.264/AVC [8]. The parameter set is designed to support for a robust and efficient transportation of header bits.

```
//employ inter-layer prediction
If  net_rep ≤ thresh-12  {
        If   inter-layer = 0 {
         Inter-layer = 1
}
Elseif       net_report ≤ thresh-13 {
// Constrained QP minimum  value for rate control   RC) - described in chapter 5
     If   Rt  ≤ minRt {                        (12)//determine the bandwidth
        Qp = Qpcurrent + MinDeltaQp            (13)//set Qp for better rate control
              } else                           (14)// otherwise
              {                                (15)// start
         Qp = Qp + MaxDeltaQp}                  (16)// for rate > 32kbit/s
        } elseif net_report ≤ thresh-14{
// setting lambda value for better rate distortion optimization (rdo)
     If bits ≤ 64000 {                         (18)//select lambda value if bit-rate ≤ 64kbit/s
    λnew = λcurrent-single-layer-selection      (19)// lambda based on experiment in chapter 6
              } else                           (20)// bit-rate ≥ 64kbit/s
   λnew = 0.8λcurrent-single-layer-selection
// filtering algorithm
Start processing filtering algorithm in chapter 6
                    }
```

Figure 7 - 9 : *Adaptation II* algorithms for flexible scalability and adaptation

Where R_t is the computed target rate, $MaxDeltaQ_p$, $MinDeltaQ_p$ are the maximum (8) and minimum (2) quantisation levels respectively, minRt defined a target rate of 32000 bits/sec and $Q_{p\text{-}current}$ is the last Q_p used.

7.7. Results and Discussions for *Adaptation II*

Dynamic Filtering, *Constraint Qp Algorithm* value in rate control and *Lambda Value Selection* for rate distortion are the main components for this adaptation option II. A foreman sequence is used to evaluate the techniques for de-blocking filter on a heterogeneous network simulator. FN*SL*-1 filtering revealed better performance than FL, NFL, 2FLT, LUNCN*SL*, LUNCL and 2LUNC. These notations are defined in Table VI - XVII. FN*SL*-1 shows a reduction of average delay values of 0.0000381sec, 0.0000004sec, 0.000003sec, 0.00000047sec, 0.0000053sec and 0.00000054sec against FL, NFL, 2FLT, LUNCN*SL*, LUNCL and 2LUNC respectively. The delay values are sampled at a frequency of 4secs with 250 kbit/s bit-rates.

Table VI - XXII shows results obtained for transcoding from a bit-stream with filter technique say B to a bit-stream with filter technique say A. Objective and subjective evaluations of the different filter techniques are presented in section 0 and section 6.8.3 respectively.

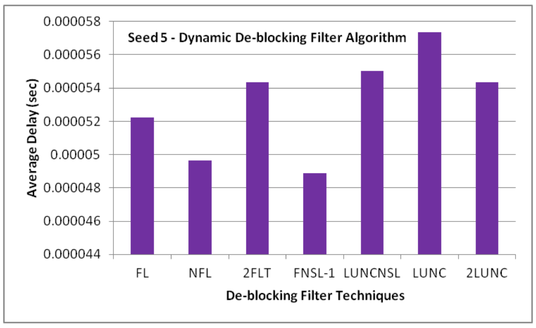


Figure 7 - 10 : Real time Simulations for dynamic de-blocking filter algorithm over heterogeneous networks (seed5)

Table VI - XXI summarised the average delay values for de-blocking filtering techniques. Table VI - XXIII contained detailed experimental results where the developed lambda Selection Algorithm is used. The algorithm considered different setup values to be used for lambda based on the video current bit-rates. This has been introduced based on the facts obtained from several experimental results. The experimental results are documented in section 6.9.

In experiments with *Constraint Qp Algorithm*, a significant gain is realised. The results obtained in Table V - XI, Table V - XII and Table V - XIII show that a proper choice and constraining of the Qp value during rate control will support achieving the target bit-rate with up to 50% bits saving. The experimental results in Figure 6 - 53 indicate that more than 50% bits reduction can be obtained from the maximum use of possible inter-layer predictions.

// Collect network report and determine best adaptable stream

If net-report ≤ thresh-7 (1) //check immediate net status

technique = current; (2) // stay on current if net status improved

Else if (3) //otherwise improve the stream adaptation

Net_report >thresh-7 { (4) // net-report does not improve

new-Q_p = old-Q_p +2; (5) // reduce produced bits

Seq = Seq_B; (6) // IBBP – proved better in chapter five

SnrLyrs = snrMinLy; (7)// reduce Number of quality Lyrs

*new-GOP = old-GOP * 2; (8)// increase GOP pictures- better prediction*

quality = MGS; (9)// MGS proved better r.d.o. in chapter five

entropy = cabac; (10)//cabac has better bit-rate saving-chapter5

slice-struct = 200 (11)// max. slice better r.d.o. in chapter five

KeyPic_Enc = true (17)// Encode KeyPics for better r.d.o.

Slice_to_MB_map_type = box (18)// select slice map type (simulations)

Figure 7 - 11: Parameter Adaptation Algorithm

7.8. Results and Discussions for Parameter Adaptation

Experiments and real time simulations show the gains of using the parameter adaptation algorithm. In real time simulations over the built heterogeneous network in section 5.5.4, exchanging *IBBP* identifier for that of *IPPP* reduced the average delay by 0.0000027sec, 0.000006sec, 0.00000005sec, 0.00000004sec and 0.00000008sec for seed 1, 2, 3, 4, and 5 respectively.

Similarly in section 5.6.8, when *GOP* of 64 identifiers is exchanged with that of *GOP* of 32, a reduction of 0.000030sec, 0.0005sec, 0.0005sec, 0.0007sec and 0.00014sec for seed values 1, 2, 3, 4, and 5 respectively is achieved. Figure 7 - 12 shows a bit-stream on *GOP* 32 which identifier is changed to that of *GOP* 64. There is a bit reduction in this process which improves the adaptation of the bit-stream to the network channels. Figure 7 - 12 represent scalability characterisation for *GOP* 64 and *GOP* 32 where bitstream decoding is switched to *GOP* 64 bitstream because it has low complexity bitstream (low bitrates).

An experiment is conducted with BUS sequence for *CABAC* and *CAVLC* identifiers performance in section 5.4-5.4.3. A reduction of 10kbit/s, 12kbit/s and 13kbit/s for L0, L2 and L3 respectively is obtained with the *CABAC* algorithm identifier exchanged to substitute the *CAVLC* identifier. Similarly, 40-50% bit-rate saving is achieved for encoding key pictures as presented from the experimental results in Table V - X. Detailed discussions of the results are available in chapter five, sections 5.4.1, 5.4.2 and 5.4.3. Other results for Qp value changes in section 5.11, *MGS* and *SGS* scalability methods evaluation in section 5.10, and slice structure implementations in section 5.7 are discussed and presented in chapter five.

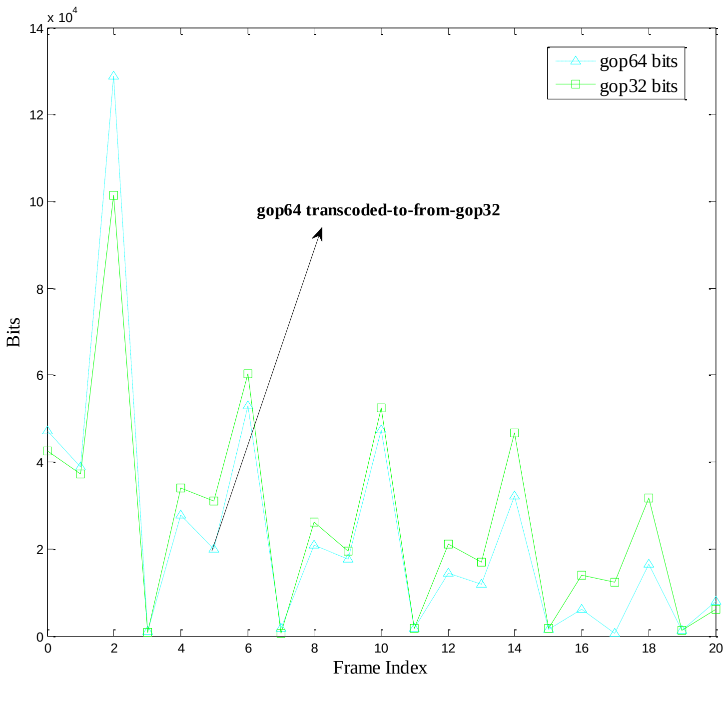


Figure 7 - 12 : Transcoding from a bit-stream on *GOP* 32 to bit-stream on *GOP* 64

7.9. Transcoding for Scalability Adaptation

The challenge in the delivery of guaranteed quality video to consumers is the constantly changing bandwidth of the network channels. This is noticeable over the Internet channel links especially where real time communication needs to be established. The frequent and unpredicted channel conditions fluctuations and their constraints require reactive measures to be taken at different time instances and terminal locations. This action is performed through a method called video transcoding. Several techniques and algorithms are developed in chapter five to seven. An efficient operational functionality of the techniques and algorithms to provide a robust and flexible scalability adaptation within a multi-channel links remains a challenge. One main challenge here is the efficiency of the system implementation with a reduced complexity.

Video transcoding consists of the necessary procedures for the conversion of a compressed video stream from one syntax to one alternative syntax for inter-network communications. Thus, the tool that makes use of this algorithm to perform the conversion from one syntax to another is referred to as a video transcoder [9, 10]. In the transcoding process, an incoming video stream is converted to a different video format, bit-rate or QoS level. Hence, our proposed algorithm (*SADMA*) is the video transcoder tool that acts with the necessary remedies during the video services at different network terminals and conditions. Figure 7 - 16 and Figure 7 - 17 show *SADMA* as a video transcoder that converts from one scalability technique or sub-stream to another one of less or more complexity.

There are four major types of video transcoding algorithms that are usually discussed in the literature. The most commonly referred transcoding is the homogeneous video transcoding. In this transcoding algorithm, the main aim is to reduce the bit-rate, frame rate and/or resolution of the pre-encoded video stream. The method does not involve any kind of syntax modifications to the coded video data. Therefore, the incoming compressed video stream preserves its format and compression characteristics after conversion to a lower bit-rate or resolution. This is illustrated in Figure 7 - 13. In our developed techniques presented in chapter five, homogeneous transcoding is one feature which happens automatically. With the technique, it is possible to provide video services with communication links that have different bandwidth requirements at various frame rates and output bit-rates. This is proved to have more gain than the state-of-the-art-techniques as discussed and evaluated in chapter six. Multi-point real time applications such as video conferencing can be supported with the use of the techniques. $t+s$ technique can support various display formats and bit-rates. The popularity of heterogeneous video transcoding has increased due to the increase in the diversity of multimedia networks. The developed techniques in chapter five output an encoded bit-stream with a wide coverage of bandwidth requirements, QoS levels and several frame rates for adaptation to multi-channels and applications. This has allowed heterogeneous transcoding to be operational from the use of the techniques. *SADMA* is employed as a heterogeneous transcoder to switch between the techniques when there is a need from current report of network channel conditions during the video services. Error resilience systems and multimedia traffic planning purposes are the main motivations for the third and fourth types of transcoding [9].

For the purpose of this research work, the use of Homogeneous and Heterogeneous transcoding are adopted[10]. In the homogeneous transcoding, necessary resources are naturally built in the developed techniques while in the heterogeneous method, the diversity of heterogeneous networks requirements is accommodated by switching between the bit-streams of several developed techniques. This is possible with a condition that all the bit-streams originated from the same source of video sequence. However, methods that can be used to switch to a different video source are discussed in section 7.12.1. The details of homogeneous and heterogeneous transcoding for the purpose of this research work are discussed in section 7.9.1 and 7.11.

7.9.1. Scalability Adaptation using Homogeneous Transcoding

In this transcoding algorithm, the main aim is to decrease the bit-rate, frame rate and/or resolution of the pre-encoded video stream. The technique does not involve any kind of syntax modifications to the coded video data. Therefore, the incoming compressed video stream preserves its format after conversion to a lower bit-rate or resolution.

In a scalable stream, the same video sequence is coded with multiple bit-rates for video transmission and adaptation across multiple channels and decoders. These channels and decoders vary in bit-rate, quality and display resolution requirements. In the developed scalability techniques of chapter six, the bit-stream of each of the techniques shows different performances of scalability and adaptation. This has been evaluated and proved to provide better scalability than state-of-the-art and non-scalable techniques. One single bit-stream of a technique provides a range of scalability levels, quality and adaptation. Each level of scalability is a sub-stream which can switch to another lower sub-stream in a situation where the reported network condition predicts a future degrading of the channels. The interface for the extraction of BL, EL0 and other sub-streams is discussed in section 7.10.

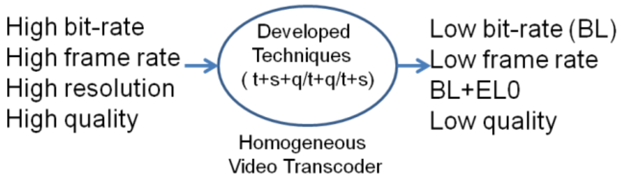


Figure 7 - 13 : Illustration of Homogeneous transcoding for *t+s+q*/*t+s*/*t+q* techniques

7.10. Interface for sub-stream extraction

The design of SVC standard supports easy bit-stream manipulations and extraction. In a process of extracting a sub-stream for example BL or and EL0, or any part of the bit-stream with a reduced spatio-temporal resolution and/or bit-rate, all NAL units that are not required for decoding the target sub-stream should be discarded. For this process to occur, two important pieces of information need to be known. The parameters like dependency identifier D, the quality identifier Q, and the temporal identifier T, should be known for each coded slice NAL unit. In addition, it needs to be known what NAL units are required for inter-layer prediction of higher layers [7].

In order to simplify bit-stream manipulations, the 1-byte header of H.264/AVC Extension is extended by additional three bytes for SVC NAL unit types. The identifiers D, Q and T include these extended header bytes. SVC also specifies additional Supplemental Enhancement Information (SEI) messages which contain information such as resolution format and bit-rate of the layers that are included in the SVC bit-stream. This can further support the bit-stream adaptation process. For example, to extract BL0 from the bit-stream for one of the techniques, T, D and Q identifiers need to be known from each of the slice units. Other parameters such as layer bit-rate, spatial resolution also need to be known for the processing of *Constraint Qp Algorithm*, *Lambda Value Selection Algorithm*, etc. Figure 7 - 14 shows an example of SEI messages that need to be known from the bit-stream header prior to layer extraction and the developed algorithm processing.

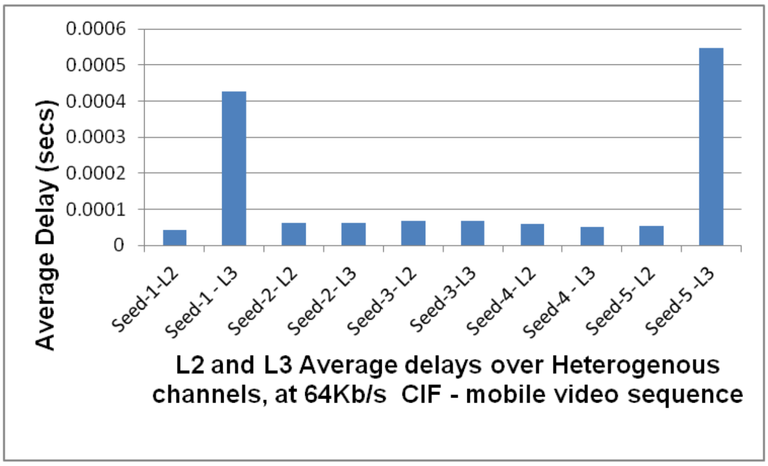
Slice-Number	slice0	slice1	slice2 …………………	sliceN
Temporal-ID	T0T4T3T5	T5T3T4T0	T0T4T3T5	TN0TN3TN4TN5
Dependency-ID	D0D2D1D1	D0D2D1D1	D0D2D1D1	D0D2D1D1
Quality-ID	Q0Q2Q1Q2	Q0Q2Q1Q2	Q0Q2Q1Q2	Q0Q2Q1Q2
Bit-rate(Kb/s)	10,32,300,64	10,32,300,64	10,32,300,64	10,32,300,64
Format		CIF/QCIF		

Figure 7 - 14 : SEI messages required to be known prior layer extraction and processing of *Adaptation I* and II

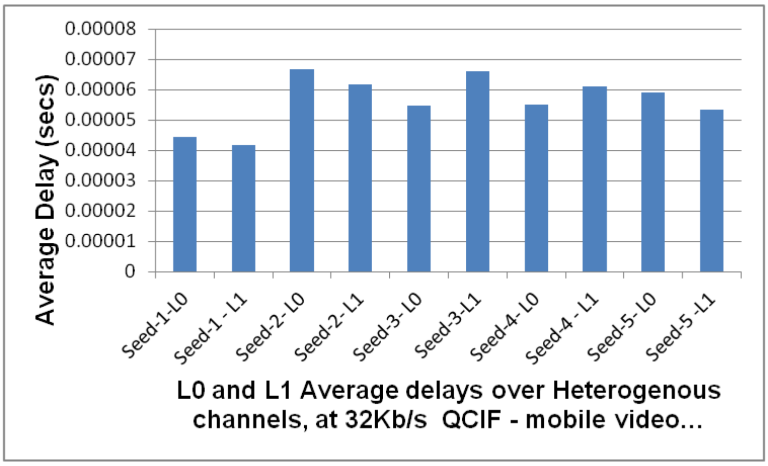
Figure 7 - 15 (a, b) presented a real time simulation for four extracted layers from t+s+q scalability technique bit-stream. The layers are L0, L1 in (b) and L2, L3 in (a). L0 is the BL in the bit-stream which all other layers reference from. L1 is the next layer in the hierarchy and to decode this layer BL + L1 should be decoded. L1 contains highly compressed and predicted frames from L0. The packets sizes of L0 and L1 are experimentally verified to be the same. However, a decoder decoding from L1 usually requires higher bandwidth because it reconstructs the picture from L0 and other layers.

Figure 7 - 15 (a) indicates that L2 which is lower layer than L3 performs better in the multi-channel network video services. L3 is referenced from all the lower layers L2, L1 and L0 and therefore a decoder should use higher bandwidth than the one that decodes L2 or L1.

From the results of this experiment, it can be concluded that *SADMA* algorithm can switch transmission to BL only or BL plus any of the available layers for a satisfactory scalability adaptation at a network reported condition.



a. L2 and L3



b. L0 and L1

Figure 7 - 15 : Real Time Performances for sub-layers in a *t+s+q* mobile bit-stream (L0, L1, L2 and L3)

7.11. Scalability Adaptation using Heterogeneous Transcoding

In Heterogeneous encoding, some different formats might be involved. The formats will possess variable syntax or complexity. Figure 7 - 16 illustrates this technique where one scalability technique is switched to another technique for adaptation of video services to the network. Although the scalable streams of the techniques are all coded from the same video sequences but are differentiated based on bit-rate, frame rate and quality level. In some cases, both Heterogeneous and Homogeneous transcoding may be involved as shown in Figure 7 - 17.

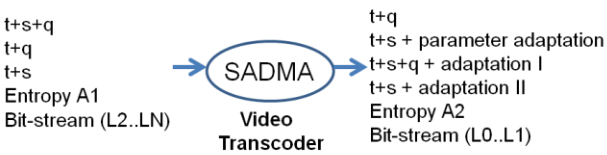


Figure 7 - 16: Illustration of Heterogeneous transcoding switching between *t+s+q*, *t+s*, and *t+q* techniques

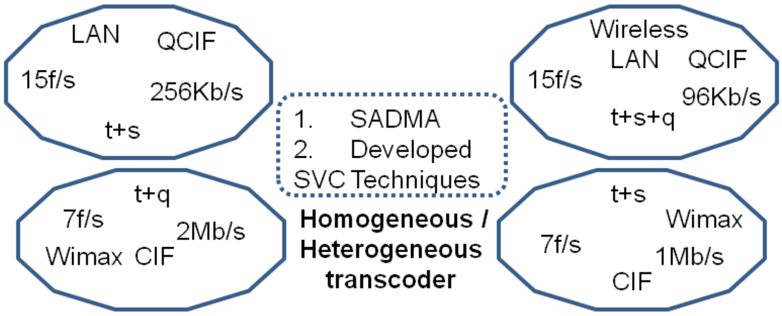


Figure 7 - 17 : Heterogeneous and Homogeneous transcoding for multi-channel video services

7.12. Switching between bit-streams

Several scalability levels are embedded in *t+s*, *t+s+q* and *t+q* techniques. These techniques posses dependent scalability levels that provide alternative scalability and adaptation when there is need emanating from the reported network conditions. This switching is done as soon as forthcoming channel congestion is predicted. SI (Secondary Intra-Frame) and SP (Secondary Pictures) pictures are used to switch between bit-streams [11, 12].

However, picture drift is encountered since the frame prediction type in one bit-stream is unlike the other when using SP-frame (P-frame has reference to previous frame). These pictures (SP/SI) can be produced at the time of switching from one bit-stream to another and are transmitted as the first picture after switching. Secondary pictures are coded slices that allow efficient switching between the video bit-streams. These pictures also allow the video decoders for efficient random access. For instance, a decoder starts decoding *t+s+q* bit-stream from the receiving packets. At a time during the decoding process, decoding or processing capability of the application becomes low and can no longer afford to decode at the bit-rate provided by *t+s+q* bit-stream. This will require switching to another low bit-rate and low complexity bit-stream such as *t+q* bit-stream. The *SADMA* algorithm does the switching based on the processed report about network conditions. Now, two levels of scalability are provided with *SADMA* algorithm. One is the embedded scalability in *t+q* bit-stream and the other is distinction of scalability levels of *t+s+q* and *t+q* techniques. In chapter six, experimental results show that the video stream embedded a minimum of 29kbit/s and 33.7kbit/s for *t+q* and *t+s+q* techniques respectively for the same anticipated target rates. Section 7.12.1 discussed a scenario describing how the bit-stream *t+s+q* is switched to the bit-stream *t+q*. The process of resolving encountered picture drift due to the switching between bit-streams is also discussed.

7.12.1. Description of a scenario for switching between bit-streams using I-frames and SP-frames.

To present an example scenario for switching between bit-streams, an active decoder is used as an illustration. Considering an active decoder, decoding stream A(*t*+*s*+*q*) and for limitation of resources (bandwidth, memory etc.) needs to switch to decoding stream B(*t*+*q*). In here, we have made an assumption that each frame is encoded as a single slice and predicted from one previous decoded frame. After the decoder might have decoded P slices tsqA0 and tsqA1, it now wants to switch to decode stream B(*t*+*q*) and decode tqB2, tqB3….tqBN. If all slices in stream A(*t*+*s*+*q*) are originally coded as P-slices, then the decoder will not have the correct decoded reference frame(s) required to reconstruct tqB2. This is because tqB2 is predicted from the decoded picture tqB1 which does not exist in bit-stream A(*t*+*s*+*q*). This might result in a picture drift consequence.

One solution to avoid picture drift when switching between bit-streams is to assign tqB2 using intra-coding as an I-frame. Because it is coded without prediction from any slice/frame, it can be independently decoded without reference to stream B(*t*+*q*). The decoder can now switch between bit-stream A(*t*+*s*+*q*) and B(*t*+*q*). It can be concluded that switching can be accommodated by inserting an I-frame at regular intervals in the coded sequence to initiate switching points. However, an I-frame may contain large amounts of coded bits than a P-slice and the output is an unwanted peak in the coded bit-rate at all points of switching. Figure 7 - 18 illustrates the switching process using I-frame insertion at switch points.

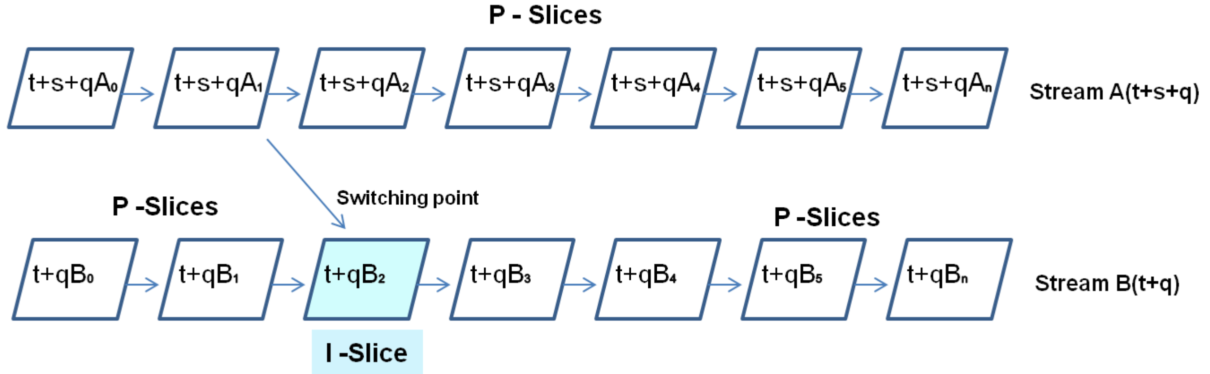


Figure 7 - 18 : Illustration of switching between bit-streams using I-frame

Figure 7 - 19 demonstrates the use of SP slices to support switching between bit-streams A(t+s+q) and B(t+q). The A(t+s+q) and B(t+q) bit-streams are coded from the same source sequence. The use of SP-slice is more efficient than the use of I-slice as the increase in bit-rate initiated from using the I-slices is avoided. At the switching point (t+s+$q$$A_{1,}$ frame number two), three SP slices (t+s+$q$$A_2$, t+s+$q$$AB_2$, t+$q$$B_2$) are made, each one of the SP slices is encoded using motion compensated prediction making them more efficient than the I-slice. The main key to the switching process is t+s+$q$$AB_2$ slice created in such a way that it can be decoded using motion compensated reference picture t+s+$q$$A_1$ to generate decoded frame t+$q$$B_{2.}$ This meant that the decoder output frame t+$q$$B_2$ is identical whether decoding t+$q$$B_1$ followed by t+$q$$B_2$ or t+s+$q$$A_1$ followed by t+s+q.tqAB_2.

In this process, an additional SP-slice is required at each switching point. However, the extra SP-slice is likely to be more efficient than encoding frames t+s+$q$$A_2$ and t+$q$$B_2$ as intra-frames.

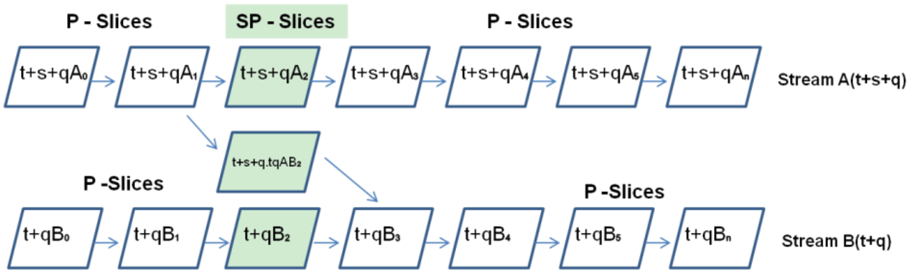


Figure 7 - 19 : Illustration of switching between bit-streams using SP- frames

Figure 7 - 20 shows how SP-slice t+s+$q$$A_2$ is produced. The decoded frame t+s+$q$$A_1$ (motion compensated slice of t+s+$q$$A1^1$) is subtracted from frame t+s+$q$$A_2$ and the residual is encoded. This is not similar to usual P-slice, the subtraction occurs in the transformed domain. This is because the subtracted frame t+s+$q$$A1^1$ is not quantised.

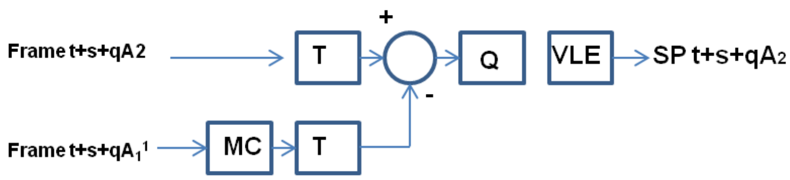


Figure 7 - 20 : Encoding SP-slice t+s+$q$$A_2$

SP-slice $t+qB_2$ in Figure 7 - 21 is coded with the same procedure. From Figure 7 - 22, a decoder that has previously decoded frame $t+s+qA_1$ can decode SP-slice $t+s+qA_2$. In practice, further quantisation and re-scaling steps are required to avoid the mismatch between encoder and decoder.

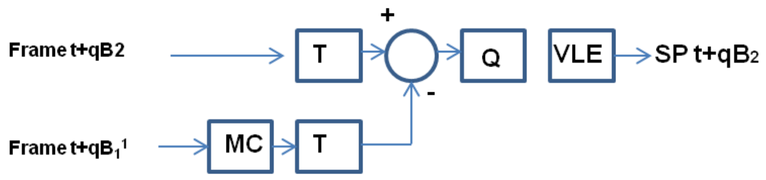


Figure 7 - 21 : Encoding SP-slice $t+qB_2$

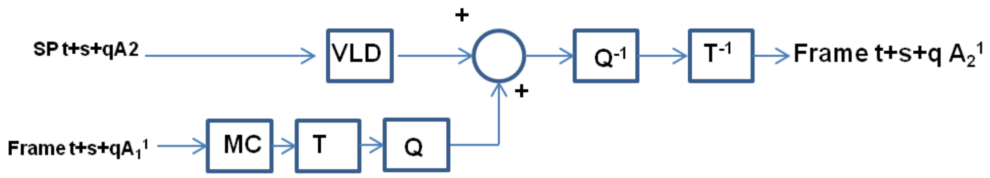


Figure 7 - 22 : Decoding SP-slice $t+s+qA_2$

In Figure 7 - 23 SP-slice $t+s+q.t+qAB_2$ is coded. $t+q$B2 frame which we are switching to is transformed and a motion-compensated prediction is formed from $t+s+qA1^1$ (the decoded frame which we are switching from). The Motion Compensated (MC) block in this Figure attempts to find the best match for each macro-block of frame $t+qB_2$ using decoded picture $t+s+qA_1$ as a reference. The MC frame $t+s+qA1^1$ is transformed and then subtracted from the transformed $t+q$B2 frame. The residual is obtained and quantised, encoded and transmitted into the network.

Figure 7 - 23 shows how SP-slice $t+s+q.t+qAB_2$ is generated from encoding frame $t+qAB_2$. Also Figure 7 - 24 shows how frame $t+qB_2{}^1$ is decoded from SP-slice $t+s+q.t+qAB_2$ [11, 12]. It can be concluded that, given a decoder that has previously decoded $t+s+qA1^1$ frame can decode SP-slice $t+s+q.t+qAB_2$ to generate $t+qB_2{}^1$. $t+s+qA1^1$ is motion compensated (with the motion vector data coded as part of $t+s+q.t+qAB_2$), transformed and added to the decoded and scaled (inverse quantised) residual, the result is inverse transformed to produce $t+qB_2{}^1$.

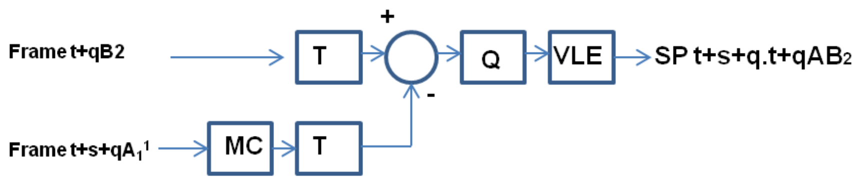


Figure 7 - 23 : Encoding SP-slice $t+s+q.t+q\mathrm{AB}_2$

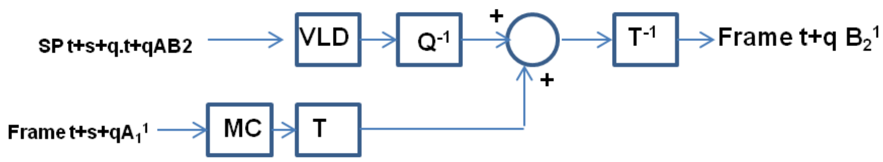


Figure 7 - 24 : Decoding SP-slice $t+s+q.t+q\mathrm{AB}_2$

To conclude the discussions for bitstream switching, if the bit-stream *t+s+q* and *t+q* are versions of the same source of original video and are coded at different bit-rates, frame rates and quality levels, the motion compensated prediction of *t+q*B from *t+s+q*A (with SP-slice $t+s+q.t+q\mathrm{AB}_2$) should be satisfactory.

The Extended profile of H.264 (chapter three) supports another type of switching picture referred to as SI-slice. This is implemented in a similar way described in this section for switching SP-slice. The only exception is that, the prediction is produced using the 4 x 4 intra prediction method from the previously decoded samples of the reconstructed frame. This slice mode can be used to switch from one sequence to a totally dissimilar video sequence. In this case motion compensation prediction is not desirable since there is no correlation among the two video sequences.

7.13. Methods for presenting scalability techniques.

The Scalability techniques can be presented in one of the following proposed methods:

Multi-bitstream-to-Multi-Channels

All available techniques are pre-coded and configured before video transmission. The techniques are then transmitted from one source computer or equipment. In this case, two-levels of scalability are provided. The embedded scalability in each of the techniques *SADMA* and the scalability provided when transmission is switched to another bit-stream. Figure 7 - 25 illustrates that multiple bit-streams are broadcast from one video source. In this situation, the bit-stream switching described in this section will be efficient. One negative effect is that this technique will require an expensive PC to reduce the complexity of processing multiple encoders at the same time.

This technique might be particularly appropriate and efficient for video services providing efficient scalability adaptation to multi-channel links and applications with several requirements for bit-rate, quality and display resolutions. Other specific applications will include video broadcasting and streaming, Internet, video surveillance and other real time applications. Experiments are conducted in chapter five section 5.14 for similar method with distinct encoders prioritised based on coding parameter influence.

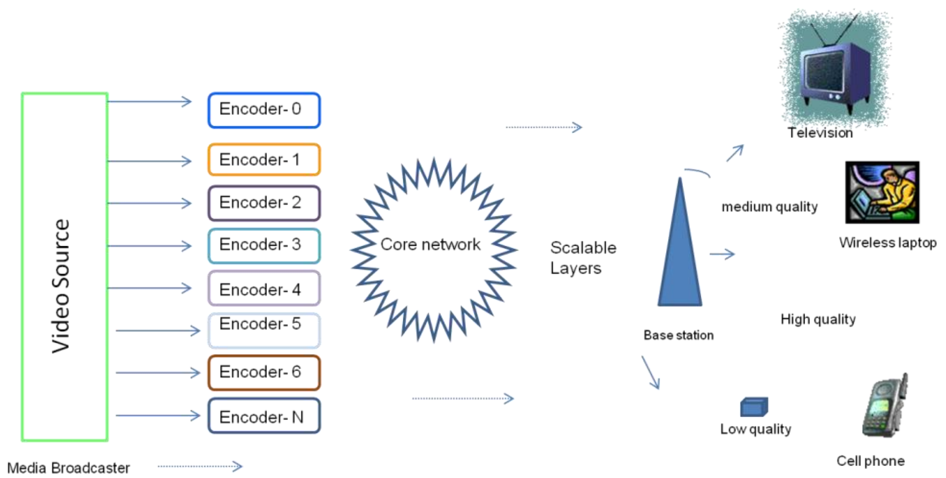


Figure 7 - 25 : Multiple scalable and adaptable video streams broadcasted from one source

The P-picture transcoding method proposed in [13] can be used as one adaptation technique. However an additional computational complexity is incurred compare to *multi-bitstream* to *multi-channel* scalability adaptation proposed here.

Transform domain-to-Pixel domain bit-stream

Techniques are presented when the reported network condition can no longer allow the current technique. In this case, the current bit-stream (transform domain) is converted to pixel domain and coded for the desired technique. Figure 7 - 26 illustrates this concept. The Algorithm (*SADMA*) decides which technique or adaptation method to be used to improve the current channel condition. The current technique is converted to pixel domain format. It is then coded to the desired adaptable bit-stream. However, although this technique can improve the adaptation to network, it also incurs complexity of coding the video sequence to another technique.

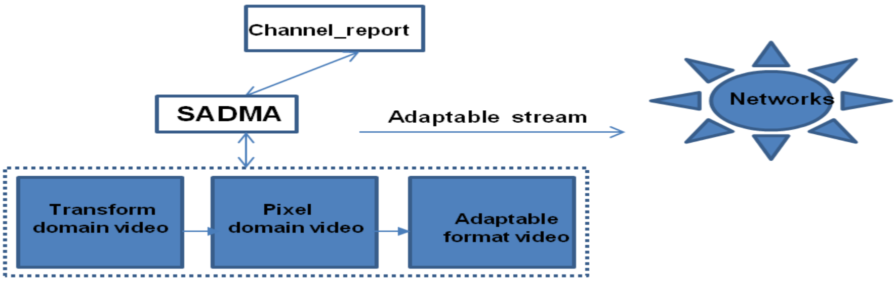


Figure 7 - 26 : The Link between *SADMA* algorithm and encoder transformation from transform to pixel domain

Use of dedicated server

In this technique, a server can be dedicated where scalable bit-streams are hosted. The *SADMA* algorithm makes one of the bit-streams available to the network when this is desired. This technique can be used since there is frequent monitoring of the network conditions. This will reduce the complexity of re-encoding bit-streams along the network terminals and the complexity of processing the bit-streams at same time. Figure 7 -27 illustrates the concept of this technique.

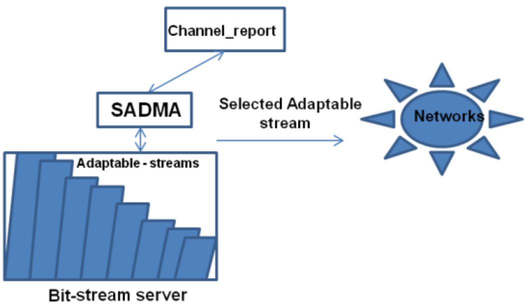


Figure 7 -27 : The Link between *SADMA* algorithm and selection of adaptable streams

7.14. Summary and Conclusion

In this chapter, Scalability Decision Making Algorithm (*SADMA*) is proposed. The algorithm frequently monitors channel conditions and takes a decision to switch to another bit-stream based on the predicted channel conditions. The switching to bit-stream can be of lower or higher bit-rate than the former depending on the reported network conditions.

The algorithm is executed in two alternative options. This is to minimise complexity and maximise QoS level when this is allowed from the network conditions. One option is *Adaptation I* which involves switching between developed scalability techniques and coding parameter adaptation in chapters six and five respectively. *Adaptation II* option is invoked when none of the scalability is tolerated in *Adaptation I*. It includes *Lambda Value Selection Algorithm*, *Inter-Layer Prediction*, *Constraint Qp Algorithm* for rate control and *Dynamic De-blocking Filtering Algorithm.*

Several real time simulations are presented from a built heterogeneous networks simulator (chapter seven, five and six) which revealed the performance of the algorithm and efficiency for adaptation to heterogeneous networks. The scalability performance of the developed techniques (chapter six) and other schemes (chapter seven, five and six) show improvement over the existing H.264/AVC Extension, non-scalable and state-of-the-art techniques.

A detailed discussion on interfaces for sub-streams extraction and transcoding among various techniques are presented. With the presented and proposed transcoding and switching techniques, it is proved that switching from one bit-stream to another is efficient. The use of SP-slice to switch to another bit-stream is demonstrated to be more efficient than the employment of I-slices since this will introduce undesirable bits causing additional load into the network.

In a situation where the *Adaptation I* and *Adaptation II* options are not allowed into the network, a stop and wait algorithm can be used. In this method, the video transmission is halted and the algorithm continues to monitor channel conditions until any of adaptations I or II conditions is allowed. However, this is a trade-off between quality and real time performance. Hence, this method functionality and implementation is limited and is not suitable for real time and low delay tolerance applications.

In conclusion, this chapter adopted an absolute usage of the developed scalability techniques, the algorithms and schemes introduced in chapters five and six and the built heterogeneous simulator in chapter four to support the simulation of a robust and *dynamic scalability adaptation algorithm* over heterogeneous networks.

References

[1] Naghshineh, M. and Willebeek-Lemair, M. (1997) "End to end QoS provisioning multimedia wireless/mobile networks using an adaptive framework", *Communications Magazine, IEEE,* vol. 35, no. 11, pp. 72-81.

[2] Lu, S., Lee, K.W. and Bharghavan, V. (1997) "Adaptive service in mobile computing environments", *Building QoS into Distributed Systems,(A.Campbell, K.Nahrstedt, Eds), pp25-36,[Chapman & Hall],* .

[3] Bianchi, G., Campbell, A.T. and Liao, R.R.F. (1998) "On utility-fair adaptive services in wireless networks", *Quality of Service, 1998.(IWQoS 98) 1998 Sixth International Workshop on* IEEE, , pp. 256.

[4] Wu, D., Hou, Y.T. and Zhang, Y.Q. (2000) "Scalable video transport over wireless IP networks", *Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on* IEEE, , pp. 1185.

[5] Zhang, W., Branicky, M.S. and Phillips, S.M. (2001) "Stability of networked control systems", *Control Systems, IEEE,* vol. 21, no. 1, pp. 84-99.

[6] www.ehow.com/info_8595771_types-delays-digital-system.html. Accessed June 2012.

[7] Schwarz, H., Marpe, D. and Wiegand, T. (2007) "Overview of the scalable video coding extension of the H. 264/AVC standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, no. 9, pp. 1103-1120.

[8] Wiegand, T., Sullivan, G.J., Bjontegaard, G. and Luthra, A. (2003) "Overview of the H. 264/AVC video coding standard", *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 13, no. 7, pp. 560-576.

[9] Sadka, A.H. (2002) *Flow Control in Compressed Video Communications,* Wiley Online Library.

[10] Amon, P., Li, H., Hutter, A., Renzi, D. and Battista, S. (2008) "Scalable video coding and transcoding", *Automation, Quality and Testing, Robotics, 2008. AQTR 2008. IEEE International Conference on*IEEE, , pp. 336.

[11] Richardson, I.E.G. "H. 264 and MPEG-4 video compression. 2003", *John Wiley&Sons Ltd., New Jersey,* .

[12] Ghanbari, M. (2003) "Standard Codecs: Image compression to advanced video coding", Institution of Electrical Engineers, .

[13] Dziri, A., Diallo, A., Kieffer, M. and Duhamel, P. (2008) "P-picture based H. 264 AVC to H. 264 SVC temporal transcoding", *Wireless Communications and Mobile Computing Conference, 2008. IWCMC'08. International*IEEE, , pp. 425.

CHAPTER EIGHT

8. Conclusion

8.1. Introduction

In this chapter, the achievements obtained in this thesis are summarised. These achievements are in line with the thesis objectives outlined in chapter one. The relevant future works are also discussed.

8.2. Literature Survey

A literature survey for scalable video coding state of the art is conducted. An overview of scalable video coding and the existing and previous techniques are discussed. Scalability techniques are improved due to the dynamic nature of the current and future applications and requirements. Scalability adaptation design concept and existing solutions are described. Rate control techniques as one major scalability components are also discussed. In line with these, the major works tailored to this research work are highlighted in the literature survey.

The concept of H.264/AVC and its major components are discussed highlighting their relevance to this research work. The major comparisons of H.264/AVC with the previous and existing techniques are also provided from the literature. The major elements where work has been done in this thesis which include layer and inter-layer predictions techniques, quantisation levels, spatial prediction and reduction, new scalability techniques with specific scalability combination and layering structure and etc are discussed.

8.3. Development and Evaluation of Novel Scalability Techniques

Scalability techniques are developed and evaluated. The default scalability techniques do not support a wide range of adaptation due to their limitations of applications usage. The design of these techniques involves combination of different scalability techniques and specific layering structure which revealed better adaptation. In the developed techniques design, more number of access units are provided and the most efficient parameter identifier values are used. These techniques include *t+q*, *t+s+q*, *t+s* and *s+q*. The objective and subjective performance of the techniques are evaluated. The techniques revealed significant objective quality performance over non-combined techniques. The techniques show variable objective performance and they are adopted as suitable for dynamic scalability adaptation. The *SL* and the techniques show the same subjective performance at bit-rates above 64kbit/s. At bitrates below 64kbit/s, there exists slight improvement over the techniques which is due to overheads on the scalable stream and their efficiency in compression which is used for better scalability adaptation. The techniques are also evaluated against some other existing techniques and based on real time performances. Section 8.3.1 to 8.3.4 summarised individual performance of the techniques:

8.3.1. *t+q*

In this scalability technique, the temporal and quality scalabilities are implemented. A reduction of bits up to 28kbit is obtained over q scalability and a gain of 3dB for qcif low resolution video sequences. In CIF format bits reduction up to 40kbits is obtained.

8.3.2. *t+s+q*

This technique combined all the three techniques of t, q and s scalability. It is therefore more adaptive in multi-channel environment. Real time simulations (section 6.7) show that *t+s+q* experienced less delay on simulations over built simulator (chapter four) which hosts WIMAX, WILAN and LAN network channels. This behaviour is observed on low and high transmission bit-rates.

8.3.3. *t+s*

This technique supports both temporal and spatial scalability**.** A bit reduction up to 15kbits is realised over s technique in QCIF while in CIF the reduction is up to 60kbits.

8.3.4. *s+q*

In this technique temporal scalability is not implemented and therefore temporal prediction efficiency is reduced escalating the transmission bit-rates. The other developed techniques outperformed this technique in the objective and subjective quality performance.

8.4. Development of Heterogeneous Network Simulator and Validation

A heterogeneous network simulator is built. The simulator is purposely built to support real time simulation experiments for the developed scalability techniques and proposed algorithms and schemes. The simulator hosts three different network channels comprising WIMAX, LAN and WILAN networks. The number of networks is limited to three due to the available capacity of the simulating PC. The Simulator is validated in accordance with the validating techniques provided by Opnet technologies. These techniques include comparing the results with another validated model, incremental analysis etc. This simulator can be re-used for any other relevant project. The simulation environment set-up which is linked together with the simulator is also verified and validated as provided in chapter four.

8.5. Analysis and Evaluation of Coding Tools and Parameters

The ***Evaluation of coding elements and parameters design*** is done for their effects on scalability outputs. The evaluation involves both objective and subjective performance and real time simulations. The design and implementation of coding elements and parameters design are change from one codec design to another. This evaluation has supported for a dynamic parameter exchange for scalability adaptation. It has also supports designing different levels of bit-streams with variable complexities. From the results obtained in this experimental and real time simulation work, a ***parameter adaptation algorithm*** is developed (section 7.6 and section 7.8).

8.6. Dynamic De-blocking-Filtering Algorithm for adaptation

Dynamic De-blocking-Filtering Algorithm supports scalability adaptation by adapting a filtering technique with better real time and adaptation performance. The filtering techniques have influences on r.d.o. due to the natural video content or the details of the video sequence which arises due to the spatial format used. Objective, subjective and real time simulations over the built heterogeneous simulator to determine the impacts of different filtering techniques defined in section 6.8 are conducted. The results (section 6.8.1 to section 6.8.3) of this work are utilised for the support of *SADMA* algorithm.

8.7. Dynamic and Novel Lambda Value Selection Algorithm

***Lambda Value Selection (LVS) Algorithm*:** This algorithm supports a maximum gain for enhancement layer and the BL. In H.264/AVC the impact of selection parameter vector on the BL is done with no consideration of the ELs (section 6.9). The H.264/AVC coding is initiated from the BL to EL in which the EL reference from the BL. This algorithm treats the usage of Lagrange parameter (Lambda) dynamically based on the layer and the bitrates. The algorithm is operated based on current computed bit-rates prior considering an effective lambda value selection. The implementation of this algorithm provides bits reduction of more than 20000bits and this depends on the current transmitting bit-rates (section 6.9).

8.8. Novel Constraint QP Adaptation Algorithm

Constraint Minimum Qp Algorithm considers the current transmission rates and employs a minimum quantisation level. This supports achieving transmission target rates and efficient scalability adaptation (section 5.13). The gain in bits reduction is in hundreds from the results of the algorithm simulations (section 5.13). However, to maintain a picture quality the algorithm is limited at bit-rates up to 300kbit/s. If the algorithm is considered at lower bit-rates, this may not favour the subjective picture quality.

8.9. Novel Improved t+s technique (Im-t+s)

Improved t+s technique (Im-t+s) is introduced to further support BL scalability adaptation, an improved *t+s* technique is developed. This technique employed a reduction of the current BL thereby creating another layer with better adaptation characteristics than the active BL. The use of the BL is one of the last solutions *SADMA* employed to resolve bad network problems. An adaptation algorithm mentioned in section 6.2 uses only BL as a solution. Hence this new technique supports better flexible scalability by providing a more adaptable BL to improve the scalability adaptation especially when the networks conditions are highly deteriorated. The gain achieved with the new BL is up to 20000 bits depending on the target bit-rates. Experiments show that the quality of the new layer (BL) is good with tiny processing complexity incurred about 0.69secs using the simulating PC (Appendix A3).

8.10. Experimental investigations on encoding key pictures techniques

An experimental work is conducted to establish the ***impact of scalability adaptation and quality performance for encoding key pictures*** within the video sequence. Key pictures are inserted after a number of designed frames for synchronisation. Experimental work revealed the effects of scalability and quality performance at various hierarchical *GOP* sizes. Poor quality is shown at smaller *GOP* sizes and the quality is improved at large *GOP* sizes. Although encoding the key pictures supports more than 20000 bits reduction and provides better scalability, the *GOP* size allocation should be considered as it can affect the video quality. Hence, the decision to encode the key pictures can be a trade-off between adaptation and quality. This is then determined based on application demand.

8.11. Quality Scalability Performance with different rates points.

An experimental work is conducted to establish the ***quality scalability influence on a scalable stream when different rate points*** are used. One quality scalability (*MGS*) which provides 25% rate points and the other (*CGS*) which supports 10% rate points are simulated. Both the low and high resolution formats are used in the experiment. The results (section 5.10) show that *MGS* has better scalability and rate distortion support with up to 10kbits reduction and a dB gain. It is therefore most preferred in providing an adaptive scalability. ***Methods for avoiding picture drift at BL*** with SNR scalability are introduced in section 5.9. Other Methods for ***improving future SNR scalability*** are also proposed in section 5.9.

8.12. Introduction and Implementation of Two level Scalability adaptation

Two level scalability adaptation scheme is proposed. The scheme is provided and is suitable to a number of applications based on the facts of their requirements. The applications include broadcasting where video streams are transmitted to multi-channel links and clients of dynamic resources and requirements. This scheme is illustrated in Figure 2 - 4. The scheme includes several developed techniques encoded at the same time and transmitted from one single processor. The requirement of this technique will involve the use of very high speed PC processor that will reduce the complexity of processing several scalable bit-streams at the same time. However the scalability adaptation provision will be higher and can be designed to suit any client or target resources.

A similar multiple bit-stream scalability adaptation is also proposed. The scheme consists of several bit-streams whereby the complexity and scalability outputs are measured based on parameter influences. This scheme is experimented in section 5.14.

8.13. A Novel Scalability Decision Making Algorithm

A Scalability Decision Making Algorithm **(*SADMA*)** is proposed and implemented. This algorithm dynamically supports a flexible scalability adaptation over heterogeneous networks. It employed the developed scalability techniques (chapter six), the introduced adaptation algorithms (chapter five, six and seven) and evaluated parameter design and coding elements (chapter five) in its implementation. The algorithm ensures that a suitable scalability stream is passed into the network at a particular network condition. It does this by frequent processing of channel conditions and hence decides for a suitable scalable bit-stream. The network report is compared with a pre-configured threshold delay allowable for a real system before a suitable bitstream is pass on to the network. The algorithm executes its task using some sets of procedural adaptation procedures. It does not perform all the algorithm tasks at once except when there is need from the network report. This is design with this nature to minimise the processing complexity and supports high quality picture when the network is not fully swamped. The algorithm establishes a suitable technique for a specific network and when there is need changes to other techniques (*Adaptation I* or *Adaptation II*) which are described in chapter seven.

The algorithm is represented as transcoder that switch from one bit-stream to another within the networks. This is illustrated as in Figure 8 - 1. The homogeneous transcoding is represented by the automatic scalability adaptation provided in the techniques bit-stream and the heterogeneous transcoding support switching from one technique to another.

The methods that can be used for switching among different techniques bit-streams and within the sub-bit-streams of a technique are discussed in section 7.12.

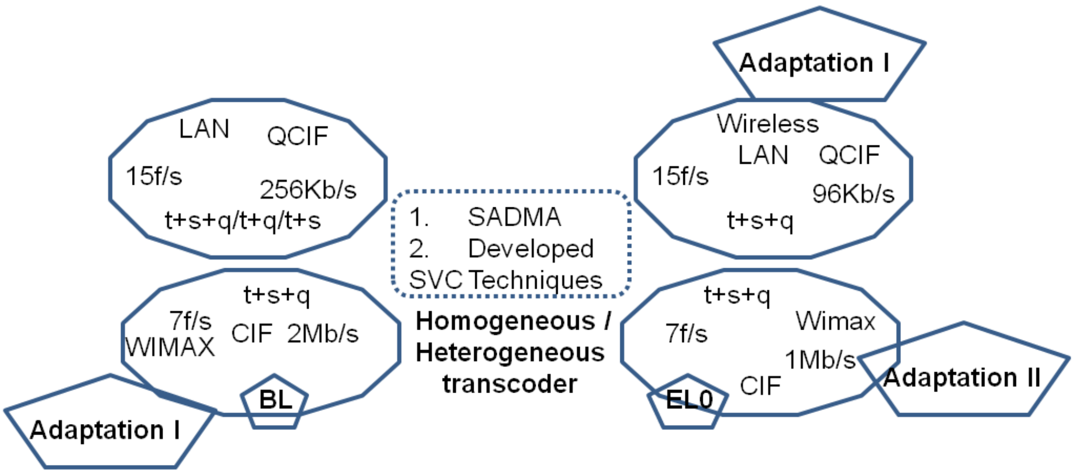


Figure 8 - 1 : Heterogeneous and Homogeneous transcoding for multi-channel video services

8.14. Other Conducted Experimental work

Inter-layer Motion Prediction: The SVC bit-stream is built and structured based on a hierarchy of layers. A maximum possible dependency among the layers will yield a maximum removal of spatial and temporal redundancies and therefore minimise the impacts of the high SVC overheads against *SL*. Hence the complexity of the bit-stream will be reduced on the transmission state. Experiments conducted revealed a gain up to 20kbits reductions over non-inter-layered sequence (section 6.10).

An experimental work is conducted to look into the ***impact of additional layering structure*** on scalability level output performance. An optimised layering structure has been determined. A layering structure encoded with four similar video sequences (quality layers) in QCIF format produces an optimised bit-rate and quality. Additional layering increases the scalability levels and provides graceful degradation due to different discrete rate points and several temporal scalable layers produced. The performance of the layering structure has been evaluated based on objective and subjective quality output. There is no video quality deterioration observed from the subjective performance of the hierarchical quality layers (section 6.3.6). The ***BL decoding computational complexity*** has also been experimented from different hierarchy of quality layers. The hierarchy of layers take variable weighing of inter-layer reference.

8.15. Future Work

There exist a number of works that can be done to extend on some areas of the thesis in a relevant further research work. The following discussed issues include some of the future projects that can be conducted.

1. The developed scalability techniques can be re-designed and evaluated to provide alternative supports for specific intended applications. For example other BL structure and different hierarchy of ELs can be defined to suit certain requirements.

2. The developed techniques, the proposed and tested algorithms and schemes can be hard-coded and evaluated to establish their performances on a system. For example, the parameter exchange during transmission can be hard-coded.

3. The built *heterogeneous network simulator* can be up-graded to include other existing and future network channels. However the proposed *SADMA* algorithm responses to all networks conditions as it does considers the reported delay on the channels to take a decision for an appropriate stream to pass into the network. In other to adapt one of the techniques for a new network channel or application specifically, the techniques should be evaluated. Hence each one of the technique can be adopted for video transmission for a particular suitable requirements. The combination of all the techniques is to suit the dynamic nature of heterogeneous network environment.

APPENDIX-A1

WIMAX NETWORK CONFIGURATION INFORMATION

1. Cell Radius = 7 celled WIMAX

2. Base Stations (BS) = 7

3. Simulation Time = 10800 secs and 600 secs used for validation purposes

4. Number of subscriber stations per BS = 5

5. Number of mobile nodes in the model = 35

6. Base station model = wimax_bs_ethernet4_slip4_router

7. Subscriber station Model = WIMAX Workstation.

8. IP Backbone Model = router_slip64_dc

9. Link Model (BS-Backbone) = ppp_adv

10. Link Model (*SITL* to WIMAX) = *SITL*_virtual_gateway_to_real_world

11. Service Class Definitions

12. Gold, Silver and Bronze

13. Efficiency Mode

14. Physical Layer Enabled

Physical Layer (PHY) Profiles

SC PHY Profiles attribute:

Frame Duration (milli sec) = 1.0

Symbol Duration (Ω sec) = 0.05

Allocation Quantum (1PS)

Frame Structure = All values promoted

UL/DL Boundary = UL Sub-frame size (Symbols) = 5000

15. Associate Subscriber station with base stations

BSMAC Address = Distance based

16. Define Service Flows

Downlink Service Flows

Service Class Name = Gold

Modulation/coding = Adaptive

Average SDU size (bytes)=1500

Activity Idle Timer (Sec) = 60

Buffer Size (bytes)= 256

ARQ = Enabled

PDU Dropping Prob. = Disabled

CRC Overhead = Disabled

HARQ Enabled = Disabled

Uplink Service

Service Class Name = Gold

Modulation/coding = Adaptive

Average SDU size (bytes)=1500

Activity Idle Timer (Sec) = 60

Buffer Size (bytes) = 256

ARQ = Enabled

PDU Dropping Prob.= Disabled

CRC Overhead = Disabled

HARQ Enabled = Disabled

17. Assign Traffic to Service Flows

Type of SAP = IP

Traffic Characteristics = Match Value Promoted

Service Class Name= Gold

18. Configuring Physical Layer Parameters

Path loss parameters = free space

Terrain Type = Terrain Type A

Shadow fading= Disable shadow fading

APPENDIX-A2

EXAMPLE IMPLEMENTED SCRIPT FOR *t+s+q* TECHNIQUE EXPERIMENT

1. echo %date%, %time% "*IBBP*I seq Struc enc start time *CABAC*" >> time_mobile_*CABAC*.txt

2. @..\bin\H264AVCEncoderLibTestStatic.exe -pf cfg\main_bsptest1_MOBILE.cfg

3. echo %date%, %time% "*IBBP*I seq Struc enc end time *CABAC*" >> time_mobile_*CABAC*.txt

4.pause

5. @..\bin\BitStreamExtractorStatic str\MOBILE_stream.264 str\Bit-extr\MOBILE_QCIFbsp0.264 -l 0 -t 80

6. pause

7. @..\bin\BitStreamExtractorStatic str\MOBILE_stream.264 str\Bit-extr\MOBILE_QCIFbsp1.264 -l 1 -t 80

8. pause

9. @..\bin\BitStreamExtractorStatic str\MOBILE_stream.264 str\Bit-extr\MOBILE_CIFbsp2.264 -l 2 -t 80

10. pause

11. @..\bin\BitStreamExtractorStatic str\MOBILE_stream.264 str\Bit-extr\MOBILE_CIFbsp3.264 -l 3 -t 80

12. pause

13. echo %date%, %time% "*IBBP*I seq Struc dec start time for 4 Lyrs QCIF L0, CIF L1, CIF L2 and CIF L3 with *CABAC* " >> dectime_mobile_*CABAC*.txt

14. echo %date%, %time% "*IBBP*I seq Struc dec start time QCIF L0 *CABAC* " >> dectime_mobile_*CABAC*.txt

15. @..\bin\H264AVCDecoderLibTestStatic str\Bit-extr\MOBILE_QCIFbsp0.264 str\dec\MOBILE_176x144_dec0snr.yuv -ec 2

16. echo %date%, %time% "*IBBP*I seq Struc dec end time QCIF L0 *CABAC*" >> dectime_mobile_*CABAC*.txt

17. echo %date%, %time% " seq Struc dec start time QCIF L1 *CABAC*" >> dectime_mobile_*CABAC*.txt

18. @..\bin\H264AVCDecoderLibTestStatic str\Bit-extr\MOBILE_QCIFbsp1.264 str\dec\MOBILE_176x144_dec1snr.yuv -ec 2

19. echo %date%, %time% "*IBBP*I seq Struc dec end time QCIF L1 *CABAC*" >> dectime_mobile_*CABAC*.txt

16. echo %date%, %time% "*IBBP*I seq Struc dec start time CIF L2 *CABAC*" >> dectime_mobile_*CABAC*.txt

20. @..\bin\H264AVCDecoderLibTestStatic str\Bit-extr\MOBILE_CIFbsp2.264 str\dec\MOBILE_352x288_dec2snr.yuv -ec 2

21. echo %date%, %time% "*IBBP*I seq Struc dec end time CIF L2 *CABAC* " >> dectime_mobile_*CABAC*.txt

22. echo %date%, %time% "*IBBP*Iseq Struc dec start time CIF L3 *CABAC*" >> dectime_mobile_*CABAC*.txt

23. @..\bin\H264AVCDecoderLibTestStatic str\Bit-extr\MOBILE_CIFbsp3.264 str\dec\MOBILE_352x288_dec3snr.yuv -ec 2

24. echo %date%, %time% "*IBBP*I seq Struc dec end time CIF L3 *CABAC*" >> dectime_mobile_*CABAC*.txt

25. echo %date%, %time% "*IBBP*I seq Struc dec end time for 4 Lyrs QCIF L0, CIF L1, CIF L2 and CIF L3 with *CABAC*" >> dectime_mobile_*CABAC*.txt

26. pause

```
27. @..\bin\PSNRStatic 176 144 .\orig\MOBILE_176x144_15.yuv str\dec\CITY_176x144_dec0snr.yuv 0 0 str\Bit-extr\MOBILE_QCIFbsp0.264 15 >PSNRdat\MOBILE_176x144_psnr0.txt

28. pause



29. @..\bin\PSNRStatic 176 144 .\orig\MOBILE_176x144_15.yuv str\dec\MOBILE_176x144_dec1snr.yuv 0 0 str\Bit-extr\MOBILE_QCIFbsp1.264 15 >PSNRdat\MOBILE_176x144_psnr1.txt

30. pause

31. @..\bin\PSNRStatic 352 288 .\orig\MOBILE_352x288_30.yuv str\dec\MOBILE_352x288_dec2snr.yuv 0 0 str\Bit-extr\MOBILE_CIFbsp2.264 15 >PSNRdat\MOBILE_352x288_psnr2.txt

32. pause

33. @..\bin\PSNRStatic 352 288 .\orig\MOBILE_352x288_30.yuv str\dec\MOBILE_352x288_dec3snr.yuv 0 0 str\Bit-extr\MOBILE_CIFbsp3.264 15 >PSNRdat\MOBILE_352x288_psnr3.txt

34 pause
```

Figure - A2 1

Figure - A2 1 shows lines of scripts codes used to implement the developed *t+s+q* scalability technique. The scripts contain standard windows command codes. It simplified the implementation of the proposed techniques. The codes are linked with the H.264/AVC codec to access various executables and implemented changes in the codec. The script is modified to work for different sequences and techniques. Codes are provided for each of the techniques and algorithms. The lines of the codes in Figure - A2 1 are described as follows:

Line 1 : Allow reading of the system time and writing to a text file (start time of encoding).

Line 2 : starts encoder execution with a reference from a defined design and defined configuration file main_bsptest1_MOBILE.cfg

Line 3 : write the end of encoding time to a defined file

Line 4 : stops the process execution

Line 5 – Line 11 : allows the bit-stream for L0, L1, L2 and L3 to be extracted.

All lines with pause are intentionally inserted for error checks and verification.

All lines with ‘echo’ word at the beginning either read the start or the end of an encoding/decoding time.

Line 15 – Line 23 allows the various encoded layers to be decoded and stored in a defined file with reference to a defined design and configuration

Line 27 – Line 33 allows the extraction of Layers bit-rates, PNSR and quantisation values, and then stores this information in a defined file.

APPENDIX-A3

PCs SPECIFICATIONS FOR SIMULATIONS AND EXPERIMENTS

Four PCs are involved in the experiments and simulations conducted in this thesis. The PCs specifications are described as follows:

EXPERIMENTAL PC-1 SPECIFICATION (desktop)

Processor : Intel® Core™ 2 Quad CPU, Q9550 @ 2.83 GHz

RAM : 2.92 GB RAM

System type : 32-bits OS, Windows XP

EXPERIMENTAL PC-2 SPECIFICATION (Laptop)

This laptop specification is used in many of the experiments including the experimental recorded time with the developed techniques.

Processor : Intel(R) Core (TM) i3 CPU M370 @2.40GHz

Installed Memory : 6GB (5.68 GB Usable)

System type : 64-bit OS Windows 7 Home Premium, Service Park 1

SIMULATION PC - A

This PC hosts the bit-stream of the developed techniques and algorithms which is communicated to simulation PC – B where the network simulator is hosted.

Processor :Intel(R) Core (TM) 2 Quad, CPU Q 9550 @ 2.83GHz

Installed Memory :512MB

System type : 64-bit OS, Windows 7 Enterprise – Service Park 1

SIMULATION PC - B

This PC hosts the OPNET environment where the Heterogeneous Network Model is built.

Processor : Intel(R) Core (TM) 2 Quad, CPU Q 9550 @ 2.83GHz

Installed Memory : 512MB

System type : 64-bit OS, Windows 7 Enterprise – Service Park 1

H.264/AVC Coder Version

JSVM 9.19.4

APPENDIX-A4

Sample Questionnaire used for subjective evaluation

1. Name:

2. Profession:

3. Age: 18 – 25

26 – 30

31 – 35

36 – 40

Over 40

4. Do you have any eye problem? Please state if yes

……

5. General Score: What is your opinion on the quality of reference and impaired videos?

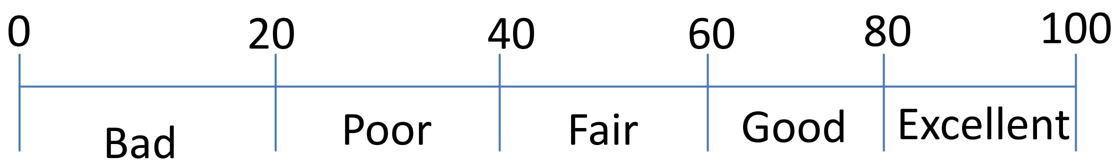
1. The same

2. Almost the same

3. Slightly different

4. Completely different

Rating Scale:



t+*s*+*q* - cif

	Sequence	Rating left	Bad	Poor	Fair	Good	Excellent	Rating right	Bad	Poor	Fair	Good	Excellent
1	bus												
2	foreman												
3	city												
4	crew												
5	harbour												
6	mobile												

t+*s*+*q* - qcif

	Sequence	Rating left	Bad	Poor	Fair	Good	Excellent	Rating right	Bad	Poor	Fair	Good	Excellent
1	bus												
2	foreman												
3	city												
4	crew												
5	harbour												
6	mobile												

t+*q* - cif

	Sequence	Rating left	Bad	Poor	Fair	Good	Excellent	Rating right	Bad	Poor	Fair	Good	Excellent
1	bus												
2	foreman												
3	city												
4	crew												
5	harbour												
6	mobile												

t+*q* - qcif

	Sequence	Rating left	Bad	Poor	Fair	Good	Excellent	Rating right	Bad	Poor	Fair	Good	Excellent
1	bus												
2	foreman												
3	city												
4	crew												
5	harbour												
6	mobile												

t+*s* - cif

	Sequence	Rating left	Bad	Poor	Fair	Good	Excellent	Rating right	Bad	Poor	Fair	Good	Excellent
1	bus												
2	foreman												
3	city												
4	crew												
5	harbour												
6	mobile												

t+*s* - qcif

	Sequence	Rating left	Bad	Poor	Fair	Good	Excellent	Rating right	Bad	Poor	Fair	Good	Excellent
1	bus												
2	foreman												
3	city												
4	crew												
5	harbour												
6	mobile												