ESTIMATION OF LRD PRESENT IN H.264 VIDEO TRACES USING WAVELET ANALYSIS AND PROVING THE PARAMOUNT OF H.264 USING OPF TECHNIQUE IN WI-FI ENVIRONMENT

JOHN JAYASEELAN
(04019561)

Submitted for the degree of
Master of Philosophy

Department of Electronic Imaging and Media Communications

UNIVERSITY OF BRADFORD

2012

ABSTRACT

While there has always been a tremendous demand for streaming video over Wireless networks, the nature of the application still presents some challenging issues. These applications that transmit coded video sequence data over best-effort networks like the Internet, the application must cope with the changing network behaviour; especially, the source encoder rate should be controlled based on feedback from a channel estimator that explores the network intermittently. The arrival of powerful video compression techniques such as H.264, which advance in networking and telecommunications, opened up a whole new frontier for multimedia communications. The aim of this research is to transmit the H.264 coded video frames in the wireless network with maximum reliability and in a very efficient manner. When the H.264 encoded video sequences are to be transmitted through wireless network, it faces major difficulties in reaching the destination. The characteristics of H.264 video coded sequences are studied fully and their capability of transmitting in wireless networks are examined and a new approach called Optimal Packet Fragmentation (OPF) is framed and the H.264 coded sequences are tested in the wireless simulated environment. This research has three major studies involved in it. First part of the research has the study about Long Range Dependence (LRD) and the ways by which the self-similarity can be estimated. For estimating the LRD a few studies are carried out and Wavelet-based estimator is selected for the research because Wavelets incarcerate both time and frequency features in the data and regularly provides a more affluent picture than the classical Fourier analysis. The Wavelet used to estimate the self-similarity by using the variable called Hurst Parameter. Hurst Parameter tells the researcher about how a data can behave inside

the transmitted network. This Hurst Parameter should be calculated for a more reliable transmission in the wireless network. The second part of the research deals with MPEG-4 and H.264 encoder. The study is carried out to prove which encoder is superior to the other. We need to know which encoder can provide excellent Quality of Service (QoS) and reliability. This study proves with the help of Hurst parameter that H.264 is superior to MPEG-4. The third part of the study is the vital part in this research; it deals with the H.264 video coded frames that are segmented into optimal packet size in the MAC Layer for an efficient and more reliable transfer in the wireless network. Finally the H.264 encoded video frames incorporated with the Optimal Packet Fragmentation are tested in the NS-2 wireless simulated network. The research proves the superiority of H.264 video encoder and OPF's master class.

DEDICATION

This thesis is dedicated to my parents, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my relatives, who taught me that even the largest task can be accomplished if it is done one step at a time. It's also dedicated to my friends who showed me courage and helped me when I was low. Loving dedication of this thesis goes to my wife, who supported all through this thesis work.

ACKNOWLEDGEMENT

There are a number of people without whom this thesis might not have been written, and to whom I am greatly indebted.

To my Parents, who has been a source of encouragement and inspiration to me throughout my life, a very special thank you for providing a 'writing space' and for nurturing me through the months of writing. And also for the myriad of ways in which, they helped me throughout my life, and have actively supported me in my determination to find and realise my potential, and to make this contribution to our world.

Very special thanks to my Wife Suvila, for her practical and emotional support and for being so supportive and helping me with this thesis. This work is for you and all the generations to come. It is dedicated to all our journeys in learning to thrive.

It is also dedicated to my relatives who knowingly and unknowingly- led me to an understanding of some of the more subtle challenges to our ability to thrive. It is also written in honour of all our my Grandparents, who though not marked by history struggled so heroically to resist the definitions attributed to them by the dominant system and communicate messages of hope and expectation

I would like to give my appreciation to my friends for their kind friendship and support from back stage. There are so many other people whom I would like to mention. However, although I am unable to bring up all the names here, you all are a part of my success.

Lastly, I am not able to find the words to say thank you to my Vanderbilt advisors, Professor Jianmin Jiang and Dr.Geyong Min, who always kept my hopes alive. If I were a F1 driver, they would have been my chief engineers. I would never have had a chance to find the chequered flag without their guidance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

1. Introduction

The development of new and powerful video compression techniques such as H.264, MPEG-4 and the advancement in networking and telecommunications, opens up a whole new frontier in multimedia communication. Interactive TV, Video conferencing, telemedicine, access to interactive pre-recorded multimedia content which are stored in remote databases, video on-demand are a few of the innumerable exciting applications can be offered.

Before diving into the research work, let us know a few facts about the video transmission over wireless network. The video content will be of very high volume and will be very much bursty in nature. If the data is very bursty, there will be a lot of data loss in the physical layer. So the video content should be less bursty while transmitted through the network. If the video steam has a large amount of Long Range Dependence (LRD), the streaming will be bursty in nature. The video content should have LRD but the amount of LRD should be optimal. The LRD value can be statistically determined using Hurst Parameter (H). Hurst parameter determines the exact value of LRD. Hurst value should be 0.5 to 1 (0.5 > = H <= 1).In this thesis the optimal value of LRD is emphasized for better video transmission in accordance with Quality of Service, error resilience and minimal data loss.

The first part of this work aims at building the LRD estimator that can track the network changes accurately. A number of methods have been proposed to estimate the Hurst parameter. Some of the most popular estimation methods are aggregated variance, local Whittle and the wavelet based methods. We opt for wavelet based technique as this could capture both time and frequency features in the data and often provide a richer picture than any other analysis.

The Internet video traffic needs to be analyzed for the time domain which consists of the timestamp in which the video packets are reaching the destination. The size of the frames and internal behavior of the packets can be analyzed by frequency domain. The self-similar scaling which is often encountered in the presence of long range dependence is naturally captured by the wavelet spectrum of the data. This scaling is used to define the wavelet estimator of the Hurst parameter in terms of the slope of the wavelet spectrum. Wavelets will construct or deconstruct data without gaps or overlaps so that the deconstruction process is mathematically reversible, which is helpful in video encoding and decoding. Thus, sets of complementary wavelets are useful in wavelet based compression/decompression algorithms where it is desirable to recover the original information with minimal data loss.

The Second part of this research work relies on determining the LRD using Hurst Parameter for trace files generated from MPEG-4 and H.264 encoder. Video over internet traffic would have a characteristic burst length which would tend to be smoothed by averaging over a long enough time scale whereas measurements of real traffic indicate that significant traffic variance (burstiness) is present on a wide range of time scales. Almost there is no self-similarity if H>1. H defines the consistency of

a sample path and is known as the Hurst parameter. These processes are referred to as scale-variant as a characteristic scale for self-similar processes that cannot possibly be defined.

A self-similar process is said to exhibit long-range dependence if it has observable bursts on all time scales. Innately in the process, memory is built-in because the dependence among an LRD process's broadly separated values is very significant and even across large time shifts. H is called the Hurst exponent which is preserved irrespective of scaling in space or time. The nature of the congestion produced from self-similar network traffic models had considerable impact on queuing performance, due in large part to variability across various time scales. This demonstrated that self-similarity resulted in performance degradation by drastically increasing queuing delay and packet loss. Self-similarity in video traffic has originated primarily due to heavy-tailed distributions of file sizes. The Hurst component was associated with heavy tailed distribution in this research work that indicates that highly large transfer requests could occur with non-negligible probability, which makes us go for efficient video coder which produces optimal amount of LRD.

Now the MPEG-4 and H.264 encoders are used to generate the trace files which indeed produce the Hurst value for each encoded video sequence. The research work produces the evidence in chapter 3 and chapter 4 with the statistical data proving H.264 produces optimal Hurst value.

Third part is the black box of this research work. The video is transmitted as packets over wireless networks to undergo packet loss due to transmission error or inserted

fading bit errors. This kind of packet loss regularly occurs in bursts that may cause considerable amount of loss of transmitted video quality. Presence of LRD generates bursty traffic. The packet video transmission over wireless networks is likely to undergo packet loss due to large packet size or very small packet size, both of which could cause bursty packet losses and thus cause considerable quality deprivation to the transmitted video. So to avoid bursty traffic or data loss for video content which are long tailed and possess LRD a new technique has been introduced to transmit optimal packet size over the wireless network. The technique is called Optimal Packet Fragmentation (OPF).The large packet size produced higher Packet Loss Ratio (PLR) in Physical Layer (PHY) [14]. If the packets that are larger in size than the capacity of the wireless network then the packet loss will be more. The small packet size produces packet loss in Medium Access Layer (MAC) due to congestion in the packets and thus increases the PHY overhead. As a result, the need exists for video coding and transmission schemes, which provide not only efficient compression performance but also comparatively strong transport performance in the presence of optimal packet size, which is encoded in the MAC level. Here OPF technique plays a major role in segregating the packets from the encoded video stream as optimal packet size.

H.264 encoder has the functionality of generating NAL units in the Network Abstract Layer (NAL). NAL units are nothing but packets of video streams. NAL is very much friendly in delivering the video stream to the wireless network. An acknowledgement signal is got from the wireless network which returns the network capacity. NAL units are also formed according to the capacity of the wireless network by the H.264 encoder, so that the overhead in MAC Layer will be reduced.

The video stream is packetized according to the network capacity and transmitted over the network in the MAC level. The wireless network research was carried out using NS2 simulator. Results show that H.264 video coded traces using OPF technique provide a very prominent enhancement in wireless network performance with minimum transmission errors, less data loss and high quality of video.

CHAPTER II

2. OVERVIEW OF WAVELET-BASED ESTIMATION TECHNIQUE

2.1 INTRODUCTION

Wavelets are functions that basically break down a signal into different frequencies and time scales and then examine the signal at the resolution corresponding to a particular scale. They can also be linked to a Fourier transformation by observing the differences between Fourier functions, which are complex exponentials, and the Wavelet functions.

The analysis based on Wavelet studies gives performs a wide comparison of the successful methods of estimation by analyzs using simulated, synthetic and real Internet traffic data sets. This highlights the various critical challenges one faces when estimating the long-range dependence parameter in Internet data traffic traces. Wavelet normally provides superior picture quality than the classical Fourier analysis and incarcerates the features of both time and frequency in the data.

Since the seminal work of Abry and Veitch [1], wavelets have become a well-known tool for analyzing the long-range dependence properties of network traffic. The wavelet spectrum of the data captures the existence of long-range dependence by self- similar scaling. This scaling is used to define the wavelet estimator of the Hurst parameter in terms of the slope of the wavelet spectrum.

The wavelet spectrum contains additional valuable information on both medium and small scales about the dependence on the data. Other important traffic features such as periodic components, deterministic breaks in the mean traffic rate and the intricate non-stationary features in the data can also be represented by the wavelet spectrum. For example, as shown by Feldmann [14], at the small time scales, the wavelet spectrum can be significantly affected by the round-trip times (RTT). This information will be ignored when we use only the estimate of the long-range dependence in Hurst parameter to illustrate network traffic. Some unusual features are also demonstrated by the Wavelet spectra, which are inconsistent with the standard models of long-range dependence.

The complex exponentials exist for the whole time scale as they are 'global' and they also have a value different from zero. As wavelets are local and exist for only a certain interval, they can easily detect the signal at any specific moment in time but on the other hand, Fourier transformation can only observe a signal in its entire domain. The wavelet function could be scaled in time and order to cover the full real line, which will help the observation of the whole signal.

Wavelet function is a window function in time domain and cannot be window function in frequency domain. It permits effective evaluation which is widely used in practice and is adequately covered in the Fourier domain. The continuous wavelet transformation is defined analogously to continuous Fourier transformation.

If *X(t)* is signal observed and $\psi_{a,u}(t)$ is a wavelet function such that u is a shift factor and *a* is scaling factor such that [2],

$$\Psi_{a,u}(t) = \frac{1}{\sqrt{a}} \Psi_0 \left(\frac{t-u}{a} \right), a \in R^+, u \in R \qquad \ldots 1$$

The continuous wavelet decomposition consists of the coefficients:

$$T_X(a,u) = < X, \Psi_{a,u} >, a \in R^+, u \in R \qquad \ldots 2$$

This constitutes an inner product between signal X and set of functions: $\Psi a,u(t)$. To obtain set $\Psi a,u(t)$ we start from Ψ_0, which is called a mother wavelet and satisfies the admissibility condition whose weak form is:

$$\int_{-\infty}^{\infty} \psi_0(t) dt = 0 \qquad \ldots 3$$

Hence, the mother wavelet gets scaled in time and space, actually in time and frequency domain, and therefore is able to 'observe' the signal at different resolutions. The actual signal can be recovered using inverse wavelet transformation:

$$X(t) = C_\Psi \iint T_X(a,\tau) \psi_{a,\tau}(t) \frac{da d\tau}{a^2} \qquad \ldots 4$$

Where C_Ψ is a normalizing constant.

The Continuous Wavelet transformation is analogous to Continuous Fourier transformation. There are several different wavelet systems defined such as Haar, Daubechies and Meyer wavelets. The property of all wavelet systems is that they have *N* vanishing moments:

$$\int_{-\infty}^{\infty} t^k \psi_0(t) dt = 0 \ \ for \ \ k = 0,1,.....N-1 \qquad \ldots .5$$

This property is valuable for the purpose of estimation and enables wavelet to 'clean' the signal from the superimposed polynomial trends of the order N-1. It must be noted that this property is separate from the wavelet rank. For example, for

Daubechies wavelets, the rank is always 2 and the other parameter, called genus, determines the number of vanishing moments. Since the continuous wavelet transformation maps 1D signal into 2D space, it follows that the wavelet coefficients are highly redundant; hence, the next step is to select the sufficient subset of coefficients that capture the complete signal. The theory that establishes the criteria for sampling the coefficients is called *multi resolution analysis* (MRA), and the wavelet transformation used is called *discrete wavelet transformation* (DWT). In this sense the Discrete Wavelet transform is analogous to Discrete Fourier transform. The MRA defines two sets of functions, scaling functions and wavelet functions.

The scaling function is associated with the wavelet matrix $A = \left(a \frac{s}{k} \right)$. The matrix A consists of elements $a = \frac{s}{k}$ and has dimension m *x* mg where m is a rank of wavelet and g its genus. The scaling function is defined as:

$$\phi(x) = \sum_{k=0}^{mg-1} a \frac{s}{k} \ \phi(mx - k) \qquad \text{.... 6}$$

Where s is a row in matrix A and $s = 1$ always for scaling function (first row) and k is a column in matrix A, $1 = k = m \cdot g$.

If $f(x)$ exists such, that $f(x) \in L^2(R)$, then $f(x)$ is called *scaling function* associated with wavelet matrix A. The $L^2(R)$ is a space of square integrable functions. For the scaling functions one defines *wavelet functions* associated with f and A as:

$$\Psi^2(x) = \sum_{k=0}^{mg-1} a \frac{s}{k} \ \phi(mx - k) \qquad \text{.... 7}$$

Where $1 < s = m$ and $1 = k = m \cdot g$.

Now we can translate and rescale the functions (6, 7) in order to analyze the signal. The functions,

$$\phi_{jk}(x) = 2^{\frac{j}{2}} \ \phi(2^i x - k) \qquad \text{.... 8}$$

$$\psi^{s}_{j,k}(x) = 2^{\frac{j}{2}} \psi(2^{i}x - k) \qquad \ldots. 9$$

are rescaled and translated to scaling and wavelet functions. The coefficient *j* is called octave, and shows the level at which we want to observe the signal in detail. By the Lawton's theorem from Resnikoff and Wells Jr, [47], every signal f $\epsilon L^2(R)$ can be expanded as a sum:

$$f(x) = \sum_{k=-\infty}^{\infty} c_k \phi_{0,k}(x) \sum_{s=1}^{m-1} \sum_{j=0}^{\infty} \sum_{k=-x}^{x} d^{s}_{j,k} \psi^{s}_{j,k}(x) \qquad \ldots.. 10$$

Where coefficients are given as:

$$C_k = \int_{-\infty}^{\infty} f(x) \phi_{0,k}(x) dx \qquad \ldots. 11$$

$$d^{s}_{j,k} = \int_{-\infty}^{\infty} f(x) \phi^{s}_{j,k}(x) dx \qquad \ldots. 12$$

Since scaling and wavelet function are measured and transformed from the original function, they both appear self-similar in a way they are characterized. Therefore, they can be potentially used to analyze the self-similar signals. What remains is to establish that an arbitrary signal can be represented by using function (9) for various *j* and *k*.

According to Resnikoff and Wells Jr [47], each function $\phi_{j,k}(t), k \in Z$ constitutes a Riesz basis for some space V_i, where Riesz basis means that functions $\phi_{j,k}(t), k \in Z$ are not necessarily orthonormal or orthogonal, but they do span the space V_i. It follows that every function in space V_j can be expressed as this linear combination of functions $\phi_{j,k}(t), k \in Z$ although this expansion is not

necessarily unique. The multi-resolution analysis consists of the set of subspaces V_1 satisfying the following properties

1. $\cap_{jkz} V_j = 0$
2. $V_j \subset V_{j-1}$
3. $V_{j\in Z} V_j$ is dense in $L^2(R)$, where $L^2(R)$ is space of square integrable functions whose domain is R.
4. For each $X(t) \in V_j$, $X(2^j t) \in V_0$.

Therefore, the idea behind MRA is that signal X(t) is projected onto each of the subspaces V_j. The approximation is:

$approx_j(t) = (proj_{vj}X)(t)$

$$= \sum_k a_x(j,k)\phi_{j,k}(t) \quad \text{.... 13}$$

Starting from the set of spaces V_j, one can construct set of spaces W_j, such that $V_j = V_{j-1} \oplus Wj$. It is shown in Resnikoff and Wells Jr [47], that functions $\Psi_{j,k}$ span space W_j. In essence, space W_j consists of the signal details, which exist on time scale 2^j. For the details we have:

$Detail_j(t) = approx_{j-1}(t) - approxj_{(t)} \Rightarrow$

$Detail_j(t) = (Pro_{vj}X)(t)$

$$= \sum_k d_x(j,k)\psi_{j,k}(t) \quad \text{.... 14}$$

and note that $approx_j$ is coarser approximation than $approx_{j-1}$. So we can start from some reference space, label it as V_0, and conclude that for every signal $X(t)$, the following holds: Every signal $X(t)$ can be approximated (or exactly represented if $X(t)$ is in V_0) by:

$$approx_0(x(t)) = approx_j(t) + \sum_{j=1}^{j} detail_t(t)$$

$$\sum_{k} a\ (j,k)\phi_{j,k}(t) + \sum_{j=l}^{j} \sum_{k} dx(j,k)\psi_{j,k}(t) \qquad \text{.... 15}$$

The single sum in equation (15) is a projection of *X (t)* to the space V_j, and double sums are details of X(t). The coefficients $a_x\ (j,k) = (X,\phi_{j,k})$ are projection coefficients of the signal X onto base V_j. Therefore, the coefficients $d_x\ (j,k) = (X,\psi_{j,k})$ represent the details of the signal X(t) for time scale $2j$, which gets lost when signal X(t) is approximated on time scale 2^{j+1} .

The set of coefficients a_x and *dx represent* discrete (and non-redundant) wavelet transformation (DWT) [47]. The coefficients d_x belong to the subset of coefficients T_X of the CWT and:

$$d_x\ (j,k) = T_x(2^j, 2^{j,k}) \qquad \text{.... 16}$$

2.2 ESTIMATION TECHNIQUES

2.2.1 Hurst Parameter Estimation

The Hurst parameter estimation has been carried out with twenty-one video traffic traces using wavelet-based estimators. The Hurst parameter estimator uses Daubechies' wavelets of genus 3 (three vanishing moments). The results are summarized in Table 1. For each estimation of *H*, the range of octaves is reported in everyplace where the linear regression is performed. The visual analyses of the log scale diagrams and identification of the linear region form basis for the selection of these ranges. For the Hurst parameter estimator, we choose the order of the moment's $q = 3$ which are also shown in Table 1 and the estimates of *H* obtained from the periodogram-based estimator and R/S plots. Log scale diagrams of the

analyzed traces have similar shapes. A typical example is shown in Fig. 1. Log scale diagrams exhibit a linear relationship between $\log_2 \mathrm{E}\{d(j, k)^2\}$ ($\log_2 Sq(j)$) and j for the largest values of j (the coarsest octaves or time scales) [37]. The linear area typically begins at $j = 2$ or 3. The lack of linearity meant for the finer octaves may be attributed to artefacts of MPEG compression algorithms or to a transition between short-term and long-term scaling behaviour. In Hurst parameter estimators of H produce similar results, as indicated by Table 1. They are in good contrast with periodogram-based estimates. On the contrary, a majority of wavelet and periodogram-based estimates are greater than one and differ from R/S estimates. The linearity of the log scale diagrams for the coarsest octaves and the good match between wavelet-based and periodogram-based estimates indicate that PSDs of the traces exhibit power-law behaviour close to the origin, with exponents $\alpha > 1$. This contradicts the LRD assumption because for LRD processes $\alpha < 1$.

2.2.2. Investigating the Source of Unreliability of the Estimates

The most likely sources for the uncertainty of the wavelet estimators can be determined by carrying out the research test for the Gaussian of the traces and the time reliability of the scaling exponent α. We also address the performance of the monofractal wavelet-based estimator in the presence of strong SRD and LRD components.

2.2.2.1 *Testing the Gaussianity of the Wavelet Coefficients*

The wavelet estimator anticipates that the process being examined and its wavelet coefficients on various octaves are Gaussian [1]. Therefore, the *q-q plots (*Engineering statistics) are used to find out how near the traces and their wavelet

coefficients are to a Gaussian distribution. A sample set of q-q plots is shown in Fig. 1. The dashed line is the reference line with a slope of one. The vertical lines mark the 10% and 90% quartiles. The wavelet coefficients are approximately Gaussian in the range of octaves where *H* is estimated.

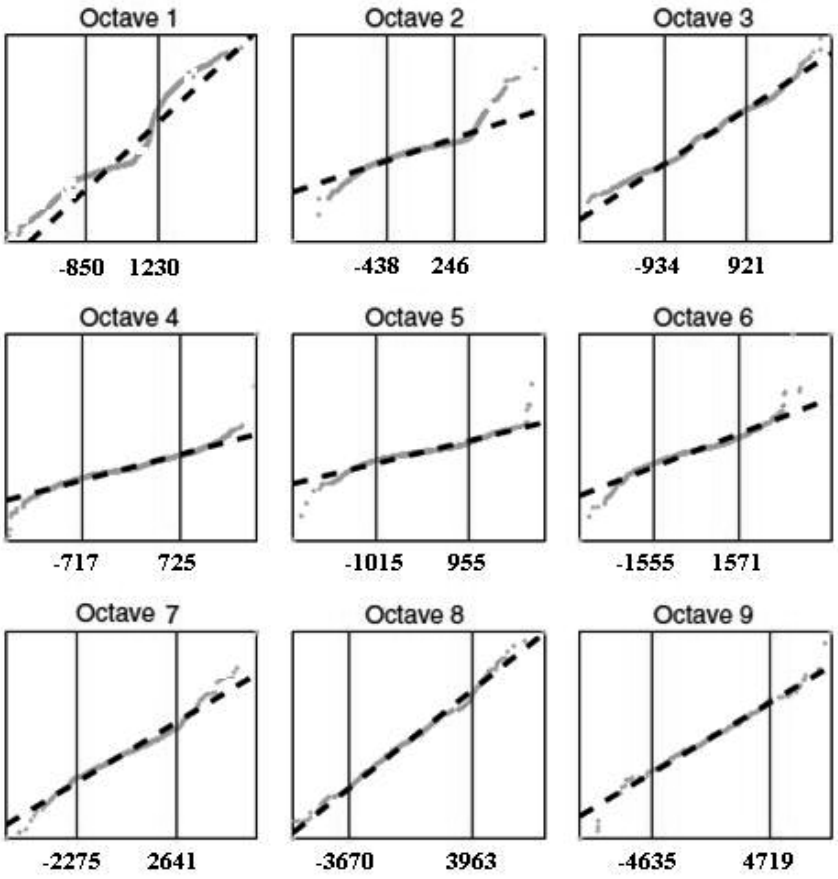


Fig 1: Q-Q plots of the wavelet coefficients for octaves 1–9 of the "Silent" video sequence.

2.2.2.2 *Testing the Time Constancy of α*

The time constancy of α for each trace can be observed. A set of tests was carried out by varying the number of sub-traces (m) between 3 and 15. The lower bound of the range of octaves where α is estimated is set to the value given in Table 1. It varies

between 2 and 4. The upper bound depends on *m*. For larger *m*, the sub-traces are shorter and there are fewer available octaves. In this experiment, the upper octave varies between 8 and 12. A sample graphical output of the test for $m = 12$ is shown in Fig. 2. The graph shows the overall value of α (solid horizontal line), the average of the 12 estimates of α (dashed horizontal line), and the confidence intervals of the estimates. In this example, the test shows that the probability of α being constant is 0%. Our findings indicate that ten traces fail the test used for all values of *m*, while others pass the test for certain values of *m*

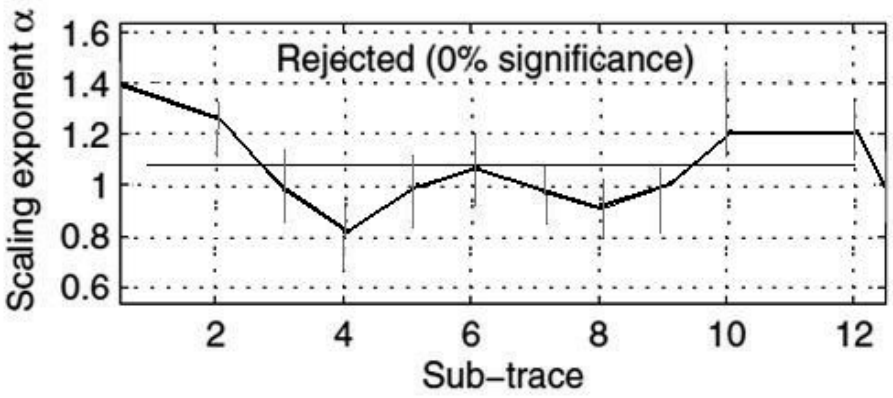


Fig. 2: Test for time constancy of α for the H.264 encoded "Silent" video sequence.

2.2 WAVELET-BASED ESTIMATION OF LONG-RANGE DEPENDENCE IN H.264 AND MPEG-4 VIDEO TRACES

The Hurst parameters, close to or greater than one, are frequently created by the Wavelet-based estimators. The varying performance of the wavelet-based estimators is the non-stationary of the scaling exponent and the existence of both short-range and long-range dependent components in the video traces. Long-range dependence (LRD) is often found in network traffic combined from many different sources [33]. The levels of LRD are characterized by the Hurst parameter (*H*), which is a significant parameter. Their precise and effective calculations are important in

statistical analysis. Known estimators for H are based on either time-domain characteristics of long-range Dependent processes (R/S plot and variance-time plot) or frequency spectrum characteristics (Whittle, period gram, and wavelet-based estimators) the purpose of the wavelet-based estimator of H to network traffic has been already reported [1]. The estimator often generates non-physical Values of $H >$ 1 and has been used for video traces [46] as well. The existence of LRD over the rough time scales is represented by the graphical outputs of the wavelet estimator. However, numerical estimates of H often lead to $H > 1$.

2.3.1. Long – Range Dependence

Let $X(n)$, $n = 0, 1, \ldots$, be a wide-sense stationary stochastic process with an autocorrelation function $r(k)$ with time lag k. The process $X(n)$ is called *long-range dependent* if the sum of $r(k)$ over all k's is infinite [38]. For large lags k, $r(k)$ is modelled as a hyperbolically (power-law) decay function:

$$r(k) = c_r k^{-2-2H)}, k \to \infty \qquad \ldots. \text{(i)}$$

where c_r is a positive constant and H ($0.5 \leq H < 1$) is the Hurst parameter. The power spectral density (PSD) $f(v)$ of $X(n)$ satisfies

$$f(\upsilon) = cf\,|\upsilon|^{-\alpha}, |\upsilon| \to 0\,. \qquad \ldots. \text{(ii)}$$

Where cf is a positive constant and α is the scaling exponent [5]. For LRD processes, $0 < \alpha < 1$. The relationship between H and α is linear:

$$H = 0.5(1+\alpha) \qquad \ldots. \text{(iii)}$$

The Hurst parameter measures the degree of LRD of a process. For SRD processes, $H = 0.5$. Values of $H \approx 1$ indicate a process with strong LRD. For example, busty network traffic has a large H [7].

2.3.2 Video Traffic Traces and Analysis Tools

Publicly available video traces were extracted from the University of Bradford [27], for this research. They comprised the sizes of video frames in bits they were examined. The duration of each frame is 40 ms (25 frames per second) and the length of the traces varies between 15 min (22,498 frames) and 60 min (89,998 frames). The characteristics of each trace in terms of their video sequence, encoding algorithm, number of frames and duration are given in Table – 1.

Video Clip	Clip Format	Range	H.264/AVC	Mpeg 4	Number of Frames
Akiyo	CIF	2-8	0.624	0.747	300
Akiyo	QCIF	2-8	0.662	0.775	300
Bridgefar	QCIF	2-8	0.420	0.793	2101
Carphone	QCIF	2-8	0.559	1.100	382
Claire	QCIF	2-8	0.630	0.829	494
Football	CIF	2-8	0.834	1.872	90
Foreman	CIF	2-8	0.383	1.069	300
Foreman	QCIF	2-8	0.638	0.985	400
Grandma	QCIF	2-8	0.523	0.906	870
Highway	CIF	2-8	0.764	1.098	2000
Highway	QCIF	2-8	0.764	1.170	2000
Mother	CIF	2-8	0.625	1.181	300
Mother	QCIF	2-8	0.662	0.837	961
News	CIF	2-8	0.326	0.602	300
News	QCIF	2-8	0.413	0.508	300
Paris	CIF	2-8	0.471	0.671	1065
Salesman	QCIF	2-8	0.721	0.959	449
Silent	CIF	2-8	0.684	1.231	300
Silent	QCIF	2-8	0.723	1.178	300
Walk	CIF	2-8	0.360	0.931	376
Walk	QCIF	2-8	0.368	1.002	376

Table – 1: Shows the Properties of the video traces and Estimates of the Hurst Parameter

A. Discrete Wavelet Transform

The discrete wavelet transform (DWT) represents a signal $X(t)$ as a weighted sum of basis functions called *wavelets* [1]:

$$X(t) = \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t), \qquad \ldots. 1$$

Where $d(j, k)$ is the wavelet coefficient at octave j and time k, and $\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t-k)$ is the wavelet obtained from an adequately chosen *mother wavelet* ψ. The wavelet $\psi_{j,k}$ is a scaled (by a factor of 2^{-j}) and shifted (by k time units) version of the mother wavelet.

The DWT captures a signal at different time scales (levels of aggregation). Due to the scale invariance of the basic functions, it is right for analyzing properties that are present across a range of time scales, such as LRD. The DWT may be implemented by a filter-bank-based pyramidal algorithm. Its small computational cost makes the DWT a popular tool for signal analysis [48].

B. Wavelet-based Hurst Parameter Estimator

The wavelet-based Hurst parameter estimator is based on the shape of the PSD function of the LRD signal $X(n)$. It has been shown that when the PSD has a power-law behaviour, the relationship between the variance of the wavelet coefficients on a given octave and the octave j [6] is

$$E\{d(j,k)^2\} = 2^{jk}\ fcC \qquad \ldots. 2$$

where the average is calculated for various k, and C is a constant that depends on the choice of the mother wavelet.

When a suitable mother wavelet is chosen [1], calculation of $E\{d(j,k)^2\}$ becomes a simple time average or sample mean for all k's:

$$E\{d(j,k)^2\} = \frac{1}{n}\sum_{k=1}^{n_j} d(j,k)^2 \qquad \ldots 3$$

where n_j is the number of wavelet coefficients available at octave *j*. Linear relationship with a slope α ($0 < \alpha < 1$) between $\log_2 E\{d(j,k)^2\}$ and *j* for a range of octaves, including the coarsest, indicates presence of LRD. Therefore, α is obtained by performing linear regression of $\log_2 E\{d(j,k)^2\}$ over *j* in a range of octaves. *H* is calculated by using equation 3. We employ publicly available MATLAB code [59] to compute the estimates of $\log_2 E\{d(j,k)^2\}$, variances of the estimates, and to perform weighted linear regression. The weights are inversely proportional to the variances of the estimates of $\log_2 E\{d(j,k)^2\}$. This estimator is called *monofractal wavelet estimator*.

The extension of the basic monofractal wavelet estimator is known as the *multifractal estimator*. It takes the moments of higher order into consideration, in addition to the second moments (variances) of the wavelet coefficients.

$$5a = \frac{1}{n}\sum_{k=1}^{n_j} d(j,k)^2 \qquad \ldots.4$$

It estimates the slope α_q by performing linear regression of $\log_2 Sq(j)$ for a range of *j*'s. *H* is calculated using an expression analogous to (eqn. 4), by taking into account the order of the moment:

$$H = 0.5 + \alpha_q/q \qquad \ldots 5$$

Both monofractal and multifractal estimators are used to produce diagrams of $\log_2 E\{d(j,k)^2\}$ or $\log_2 Sq(j)$ versus *j*, with the subsequent confidence intervals. They are called *log scale diagrams*. An example is shown in Fig. 3. The solid line connects

the estimates of $\log_2 E\{d(j,k)^2\}$ and the dashed line is the slope of the linear regression performed over the range of octaves. The vertical lines represent the confidence intervals of the estimates of $\log_2 E\{d(j,k)^2\}$.

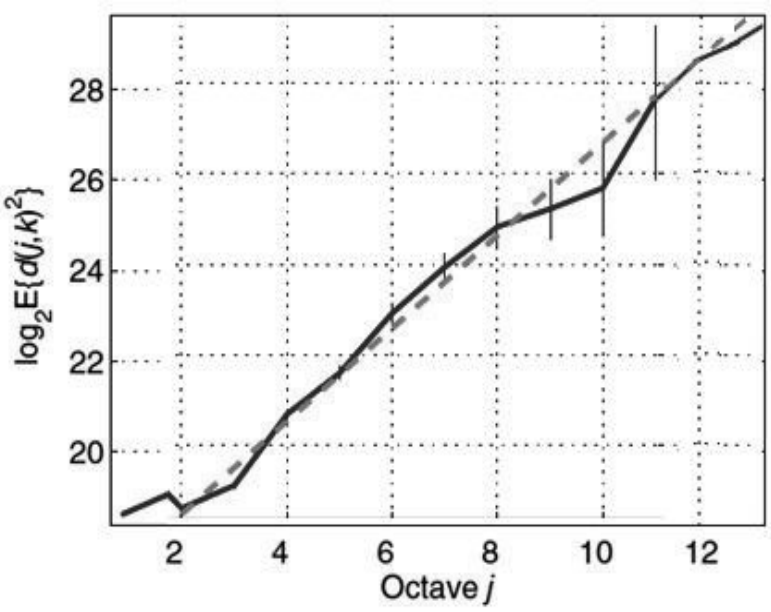


Fig -3: Logscale diagram of the H.264 encoded "Silent" video sequence.

C. Test for Time Constancy of the Scaling Exponent α

LRD processes are wide-sense stationary. However, they exhibit high burstiness and appear non-stationary [2]. In order to determine whether a process is wide-sense stationary with LRD or inherently non-stationary time constancy of the scaling exponent α has been tested. The definition and the statistical properties of the test for the constancy of α are given in D [2]. The trace is divided into m sub-traces and α is expected for each sub-trace. Therefore, the decisions are taken as to whether or not α may be considered constant over the duration of the entire trace after the comparison of estimates. The test which is carried out in MATLAB is available online [59].

2.4 CONCLUSION

In this chapter we report a wavelet based estimator for the analysis of LRD to consent to the estimation of Hurst parameter. Wavelet based estimator is applied the video traces which are generated by H.264 and MPEG-4 encoder for determining the Hurst value. As theoretically proved Wavelet based estimator is more efficient than any other Hurst estimator as Wavelet based estimator could capture both time and frequency features in the data and often provide more clarity in the derived results. By testing the gaussianity of the Wavelet coefficients and testing the time constancy of α, the Wavelets are used to determine the LRD in heavy tailed video traffic. The results in Table 1 show that the Hurst values for H.264 are more optimal than the MPEG-4 values.

CHAPTER III

3. OVERVIEW OF HURST PARAMETER

3.1 INTRODUCTION

The traffic behaviour in a network has serious implications for the design, control, and analysis of the network. It was established, by analyzing the collected data, that the Ethernet Local Area Network traffic is statistically self-similar [33]. Asynchronous Transfer mode, high speed, cell relay networks are mainly used as backbones for the interconnection of enterprise networks comprising of several LAN's, which result in self-similar behaviour of the traffic in ATM Networks [60]. Self-similarity is the property associated with fractals the object seems to be similar irrespective of the scale at which it is viewed. It is demonstrated in the absence of a natural length of a "burst "; at every time scale ranging from a few milliseconds to minutes and hours, bursts consisting of bursty sub-periods separated by less bursty sub periods.

A self-similar phenomenon appears to be the same way or behaves the same even when it is considered from different scales on a dimension or different degrees of "magnification". The dimension could be length, width or time [8]. The commonly adopted models for network traffic such as the Poisson distribution were not able to represent the fractal behaviour of the traffic due to which they could not fit the recorded traces.

Were traffic to follow a Poisson or Markovian arrival process, it would have a characteristic burst length which would tend to be smoothed by averaging over a

long enough time scale whereas measurements of real traffic indicate that significant traffic variance (burstiness) is present on a wide range of time scales.

According to [62], the degree of self-similarity as outlined by the Hurst parameter usually depends on the utilization level of the network. It can also be used to measure the traffic "burstiness". As the value of H increases, the degree of self-similarity also increases ($0.5 < H < 1$). Almost there is no self-similarity for H=0.5 and H>1. H measures the length of a long-range dependence of a stochastic process.

H defines the regularity of a sample path and is known as the *Hurst parameter*. These processes are referred to as scale-variant as a characteristic scale for self-similar processes which can not possibly be defined. In other words, *scaling* can be defined as a negative property of a time series, i.e., the absence of characteristics scales.

Rigorous constraints are imposed on a process by scaling or self-similarity. Relaxing the similarity requirement renders LRD processes in turn raising a weaker notion of *second- order self-similarity*, i.e., zooming out of the process' sample path will yield paths that are similar to the original in second-order statistics.

3.2 LONG-RANGE DEPENDENCE

The existence of long-range dependence was viewed from the heuristic plots such as Variance Time plot and Autocorrelation plot. A self-similar process is said to exhibit long-range dependence if it has observable bursts on all time scales; values at any to instant are normally related with value at all future instants [34].

A significant building block of Internet engineering and design is traffic modelling and analysis. We depend on a thorough examination of network measurements and their transformation into models to assist in explaining the functionality of the Internet and improve its performance as Internet can neither be replicated nor studied as a whole.

The introduction of long-range dependence (LRD) and self-similarity, approximately 10 years ago, modified our perceptive of network traffic. LRD means that the behaviour of a time-dependent process shows statistically significant correlations across large time scales and self-similarity describes the phenomenon in which the behaviour of a process is maintained regardless of scaling in space or time. Prior to that, Poisson processes were assumed by the researchers (that is, the packet arrival process is memory-less and inter arrival times follow the exponential distribution) to be a satisfactory representation for network traffic in real systems [67].

A stochastic process satisfying relation $r(k) \sim k^{-\beta} L1(k)$, as $k \rightarrow \infty$ $0<\beta<1$ demonstrates long-range dependence. Processes with long-range dependence are distinguished by an autocorrelation function that decomposes hyperbolically (as compared to the exponential decay exhibited by traditional traffic models). The exponential decay is faster than hyperbolic decay, and since $\beta<1$, the autocorrelation values of such series added together approach infinity. The implication of this non-sum-able autocorrelation is that, if we consider n samples from the series, then the variance decreases by a value of n^{-b} and not as a function of 'n' [7].

Long-range dependence moves in the phase of conventional wisdom by suggesting that network traffic exhibits long-term memory (its behaviour across widely separated times is correlated). This assertion challenged the validity of the Poisson assumption and changed the focus of the community away from assuming memory-less and smooth behaviour and towards long memory and bursty behaviour. In this research work, an overview has been provided on what the community has learned from 10 years of LRD research and we have identified the caveats and limitations of our ability in detecting the LRD. Particularly, we want to raise awareness on two significant issues: that identifying and estimating LRD is extremely complex, and that the large-scale aggregation of the Internet's core might have shifted packet-level behaviour toward being a Poisson process. Finally, in order to measure and model the Internet we have to constantly reinvent models and methods.

3.3 SELF-SIMILARITY IN INTERNET TRAFFIC

The existence of LRD, self-similarity and heavy-tailed distributions (meaning large values can exist with non-negligible probability) in different aspects of network behaviour is evoked by sufficient quantity of evidence gathered over the past decade. We must firstly describe LRD and self-similarity in the context of time-series analysis before looking into the major progress made in LRD research.

3.4 STOCHASTIC TIME SERIES

Let $X(t)$ be a stochastic process. In some cases, X can take the form of a discrete time series $\{Xt\}$, $t = 0, 1, ..., N$, either through periodic sampling or by averaging its value

across a series of fixed length intervals. We say that $X(t)$ is stationary if its joint distribution across a collection of times $t1$, ..., tN is invariant to time shifting. Thus, we can characterize the dependence between the process's values at different times by evaluating the process's *autocorrelation function* (ACF), which is $\rho(k)$. The ACF measures similarity between a series Xt and a shifted version of itself X_{t+k} [39]:

$$\rho(k) = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2}$$

Where μ and σ are the mean and standard deviations, respectively, for X.

Also of interest is a time series' aggregated process $X_k^{(m)}$:

$$X_k^{(m)} = \frac{1}{m} \sum_{i=km}^{(k+1)m-1} X_i, k = 0,1,2,...,\left[\frac{N}{m}\right] - 1$$

Intuitively, $\{X_k(m)\}$ describes the average value of the time series across "windows" of m consecutive values from the original time series. If $\{X_k(m)\}$ were independent and identically distributed, then $\mathrm{Var}[X(m)] = \sigma^2/m$. However, if the sequence possesses long memory, then the aggregated process' variance converges to zero at a much slower rate than $1/m^2$.

3.5 SELF-SIMILARITY AND LRD

A stationary process X is long-range dependent if its autocorrelations decay to zero so slowly that their sum doesn't converge – that is, $c(\mu) = 1 for \mu = 1, 2.....7$. Intuitively in the process, memory is built-in because the dependence among an LRD process' broadly separated values is very significant and even across large time shifts.

A stochastic process X is self-similar if

$$X(at) = a^H X(t),\ a > 0,$$

where the equality refers to equality in distributions, a is a scaling factor, and the self-similarity parameter H is called the *Hurst exponent*. Intuitively, self-similarity describes the fact in which convinced process properties are preserved irrespective of scaling in space or time.

The self-similarity in second-order depicts the property that a time series' correlation structure (ACF) regardless of time aggregation is preserved. A second-order self-similar time series' ACF is similar to both coarse of fine time scales. According to the reference [39], a stationary process Xt is second-order self-similar if

$$\rho(k) = 1/2\ [(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}],$$
$$0.5 < H < 1$$

and asymptotically exactly self-similar if

$$\lim_{k->\infty} \rho(k) = 1/2[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}],$$
$$0.5 < H < 1$$

Second-order self-similar processes are characterized by a hyperbolically decaying ACF and used extensively to model LRD processes. Conversely, quickly decaying

correlations characterize short-range dependence. From these definitions, we can infer that LRD characterizes a time series if $0.5 < H < 1$. As $H \rightarrow 1$, the dependence is stronger.

In network measurement processes, *X* refers to the number of bytes and packets which arrive at a meticulous destination at consecutive time intervals, meaning that *X* describes the volume of packets or bytes observed in a link every time interval *t*.

3.6 SELF-SIMILARITY IN INTERNET TRAFFIC

Leland and colleagues' pioneering work provided the first empirical evidence of self-similar characteristics in LAN traffic [34]. They executed a stringent statistical analysis of Ethernet traffic measurements and confirmed its self-similar nature.

Specifically, they observed that Internet traffic variability was consistent to the observed time scale which means that as indicated by the Poisson traffic model, traffic didn't become smooth quickly with aggregation.

Subsequently, the failure of applying Poisson modelling in wide-area Internet traffic was described by Paxson and Floyd [41]. They exhibited that packet inter arrival times for Telnet and FTP traffic were described by heavy-tailed distributions and characterized by burstiness, thereby suggesting that the Poisson process underestimated both burstiness and variability. In addition, they proved that large-scale correlations characterized wide-area traffic traces, concluding, "Poisson modelling in wide-area traffic for all but user session arrivals should be neglected."

Both these landmark studies nudged the researchers away from traditional Poisson modelling and independence assumptions, which were discarded as being extremely simplistic and unrealistic. The nature of the congestion produced from self-similar network traffic models had a considerable impact on queuing performance [12], due in large part to variability across various time scales. Further studies emphasized that the Poisson-based models undervalued performance measures substantially. This demonstrated that self-similarity resulted in performance degradation by drastically increasing queuing delay and packet loss [40].

Self-similarity in Internet traffic has originated primarily due to heavy-tailed distributions of file sizes [9] [63]. The Hurst component was associated with heavy-tailed distribution by several studies, indicating that highly large transfer requests could occur with non-negligible probability.

Other than LRD, Internet traffic presents very versatile scaling and multifractal characteristics. It has been illustrated by many simulations and empirical studies how scaling behaviour and the intensity of the observed dependence are related to the scale of observation. Especially, strong versus loose dependence exists in larger versus smaller time scales, correspondingly. The change point is usually associated with either the round-trip time (RTT) or intrusive “fast” flows with small inter-arrival times [15] [67].

It has been suggested by some research studies that the traditional Poisson models and independence could still be relevant, in spite of the overwhelming evidence of the existence of LRD, as the number of sources increases in fast backbone links that

carry vast numbers of distinct flows, which results in large volumes of traffic multiplexing [9]. Furthermore, it has been pointed out by Y. Zhang [66] that several end-to-end network properties appear to be in line with the assumptions of independence when the non-stationarity is present (that is, statistical properties vary with time).

The previous studies conducted showed this importance of the long range dependence. They proved by using the real traffic data rather than traditional network models, that the packet loss and delay behaviour was very different in simulations [34].

Long-Range Dependence also has a critical impact on cell loss in Wide Area ATM Networks. From studies conducted by Attila Vidacs, Sandor Molnar and Geza Gordos the following results were obtained:

Let {Xi, i = 1, 2…} be a covariance stationary process (eg: the sequence of cell inter- arrival times) with mean μ , and variance σ^2 and the autocorrelation function

$$\rho (i - j) = E [(X i - \mu) (X j - \mu)] / \sigma^2$$

According to Attila Vidacs, Sandor Molnar and Geza Gordos, the long-range dependant process can be defined as:

$\{X_i\}$ is called long-range dependent, if there exists a real number $\alpha \in (0,1)$ and a constant $C_1 > 0$ such that

$$\lim_{k \to \infty} \rho(k) / [C_1 k^{-\alpha}] = 1$$

If the parameter $H = 1 - \alpha / 2$ is used instead of α, long-range dependence occurs for $0.5 < H < 1$. The higher the value of H is, the stronger the long-range dependence. The above is an asymptotic definition, it determines only the rate of convergence as the lag tends to infinity, and it does not specify the correlations for any fixed finite lags. This definition cannot be applied in practice.

3.7 LRD ESTIMATION AND ITS LIMITATIONS

The predominant way to quantify LRD is through the Hurst exponent, which is a scalar, but the calculations of this exponent are to quite complex. Firstly, it can only be estimated and can't be calculated definitively. Secondly, although the Hurst component can be estimated by different methods, the results are often contradictory and it is unclear as to which is the most accurate estimation.

The estimators of Hurst component can be categorised into two general categories: those estimators operating in the time domain and those operating in the frequency or wavelet domain. Due to time and space constraints, a complete description of all available estimators is unavailable, only an overview appears elsewhere [57].

The power-law relationship between a specific statistical property in a time series and the time-aggregation block size *m* is examined by the time-domain estimators: LRD exists if the specific property versus *m* is a straight line when plotted in log-log scale. The slope of this line is an estimate of the Hurst exponent, so time-domain estimators involve two presuppositions for LRD to exist: statistically significant evidence that a straight line is represented by the relevant points, and the slope of the

line is such that $0.5 < H < 1$ (the Hurst exponent H depends on this slope). Several methodologies such as *R/S* (rescaled range statistic), absolute value, variance, and variance of residuals are used by these estimators.

Naturally, frequency-domain and wavelet-domain estimators operate in the frequency or wavelet domain. They also investigate if a time series spectrum or energy follows power-law behaviour like the time-domain methodologies. The Periodogram, the Whittle, and the wavelet Abry-Veitch (AV) estimators [2] are included in these frequency or wavelet estimators.

3.8 CONCLUSION

In this chapter we do a detailed study of how Hurst parameter helps in finding out the LRD in a variable bit rate (VBR) video stream. It also deals with the self-similar processes that could depict the LRD. Self-similar models were analyzed. Mostly the self-similar processes could be modelled by heavy-tailed distribution such as video transmission.

Theoretically worked-out properties of the self-similar traffic were observed; for example, if the number of wireless traffic sources increases the self-similar nature increased; the long range dependence of the network traffic increased as the Hurst parameter increased, and the buffer sizes used for normal models could not be used for self-similar traffic. This particular property of long range dependence of self-similar traffic could affect the loss ratio and buffer allocation. Even today a lot of research continues on the self-similar nature of Network traffic and its effect on various factors such as data loss, buffer size and peak signal to noise ratio.

CHAPTER IV

4. ANALYSIS OF VIDEO TRACES USING MPEG-4 VIDEO COMPRESSION STANDARD

4.1 INTRODUCTION

Moving Picture Experts Group (MPEG) is a collection of standards, which are used for coding audio-visual information (e.g. video, movies, and music) in a digital compressed format. MPEG uses sophisticated compression techniques due to which the sizes of their files are very much smaller with quality equal to that of other video and audio coding formats.

Electronic products such as digital television set-top boxes, DSS, HDTV decoders, video conferencing, DVD players, Internet video and many others use MPEG techniques and treat it as a heart of all these goods. The MPEG video compression is used in these products mainly because it requires less storage for the same amount of data used by other techniques and also less bandwidth to transmit the video signals [52].

The MPEG worked to generate the specifications under the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). MPEG-1 and MPEG-2 are the two finalized standards within MPEG video. Both these standards have a similar concept as they both are based on motion-compensated block-based transform coding techniques. MPEG-4 on the other hand is different in its usage of software image construct descriptors from these more traditional approaches, for target bit-rates in the very low range, < 64Kb/sec. Presently a large number of applications utilize both MPEG-1 and MPEG -2.

However, there is no reference of MPEG-3 as initially it was expected that this standard would refer to HDTV applications, but it was later found that minor extensions to the MPEG-2 standard would suffice for this higher bit-rate, higher resolution application, so work on a separate MPEG-3 standard was neglected.

MPEG-1 was originally optimized to work at video resolutions of 352x240 pixels at 30 frames/sec (NTSC based) or 352x288 pixels at 25 frames/sec (PAL based), commonly referred to as Source Input Format (SIF) video, and it was finalized in 1991. The MPEG-1 resolution can go as high as 4095x4095 at 60 frames per second and it is not restricted only to the above sizes. The bit-rate is optimized for applications of around 1.5 Mb/sec, but if required higher rates are possible. MPEG-1 has no direct provision for interlaced video applications, such as in television applications and is defined only for progressive frames [30].

MPEG-4 was finalized in year 2000 and addressed issues such as the efficient coding of field-interlaced video and scalability which are directly related to digital television broadcasting. It has a very high quality video as the target bit rate was raised between 4 and 9 Mb/sec. MPEG-4 consists of *profiles* and *levels*. Profile defines the bit stream scalability and the color space resolution, whereas the image resolution and the maximum bit-rate per profile are defined by Levels. Presently Main Profile and Main Levels (MP@ML) perhaps the most common descriptor in use, refer to 720x480 resolution video at 30 frames per second and at bit-rates up to 15 Mb/sec for NTSC video. One more example is HDTV picture resolution of 1920x1080 pixels at 30 frames per second and bit-rate that ranges up to 80 Mb/sec. This is an example of High Level (MP@HL) and the Main Profile descriptor.

4.2 PRINCIPLES OF NON-SCALABLE VIDEO ENCODING

In this section a brief overview of the main principles of non-scalable (single-layer) video encoding (compression) with forward and backward reference, is given Fig 4. Further details are available at [52], [20].

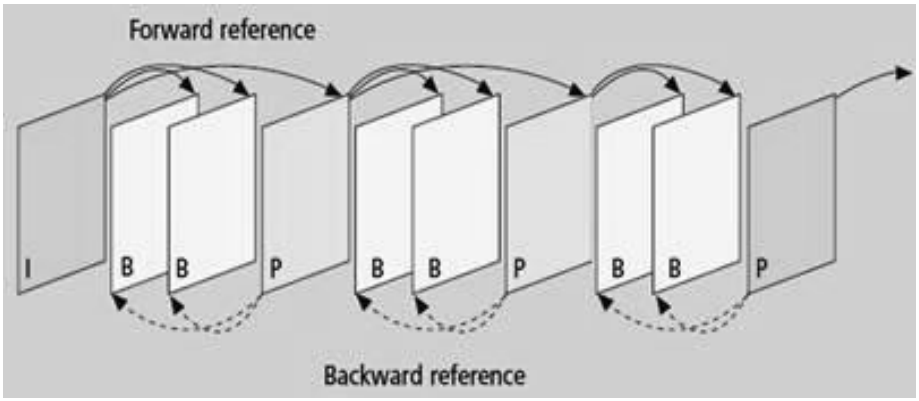


Fig. 4: Non-Scalable Forward and Backward reference video coding

In this overview a focus is made on the principles employed in the MPEG-4 standards. It is to be noted Also note that the most commercial codec's, such as Real Video and Windows Media, are derived from these standards. Intra-frame coding using the discrete cosine transform (DCT), and inter-frame coding using motion estimation and compensation between successive video frames are the two most important principles of MPEG-4 and also H.263, H.264 video coding.

In intra-frame coding each video frame is divided into blocks of 8 × 8 samples of Y samples, U samples, and V samples. Each block is transformed, using the DCT, into a block of 8 × 8 transform coefficients, which represent the spatial frequency components in the original block. These transform coefficients are then quantized by an 8 × 8 quantization matrix that contains the quantization step size for each coefficient. The quantization step sizes in the quantization matrix are obtained by

multiplying a base matrix by a quantization scale. This quantization scale is typically used to control the video encoding. A larger quantization scale gives a coarser quantization, resulting in a smaller size (in bits) of the encoded video frame as well as a lower quality. The quantized coefficients are then zigzag scanned, run-level coded, and variable-length coded to achieve further compression.

MPEG introduced frame types such as intra-coded (I), inter-coded (P), and bidirectional coded (B) in inter-frame coding and even similar frame types exist in H.26x video coding. These different frame types are organized into so-called groups of pictures (GoPs). One GoP is referred as the sequence of frames from a known I frame until and including the frame preceding the next I frame. *GoP structure* or *GoP pattern* is nothing but a pattern of I, P, and B frames that make up a GoP. A typical GoP pattern with three P frames in a GoP and two B frames before and after each P frame is illustrated in Fig. 4. The different frame types are encoded as follows. In an I frame, all blocks are intra-coded as outlined above. In a P frame the macro blocks (whereby a macro block consists of four blocks of 8×8 samples) are inter-coded (as explained shortly) with reference to the preceding I or P frame, that is, the preceding I or P frame serves as a forward reference, as illustrated by the solid arrows in Fig. 4. In a B frame the macro blocks are interceded with reference to the preceding I or P frame, which serves as forward reference, and the succeeding I or P frame, which serves as backward reference, as illustrated by the dashed arrows in Fig. 4. To inter-code a given macro block the best matching macro block in the reference frame(s) is determined and identified by a motion vector; this process is commonly referred to as *motion estimation*. Any typically small difference between the block to be encoded and the best matching block is transformed using the DCT,

quantized, and coded as outlined above; this process is commonly referred to as *motion compensation*.

The macro block is intra-coded if a good match cannot be found in the reference frame(s). (In the optional 4MV mode the above processes are applied to blocks instead of macro blocks.)

4.3 PRINCIPLE OF SCALABLE VIDEO ENCODING

The video is encoded hierarchically into a base layer and one or more enhancement layer(s) with conventional layered encoding, A basic video quality is obtained by decoding the base layer, while an enhanced video quality can be obtained by decoding the base layer together with the enhancement layer(s).

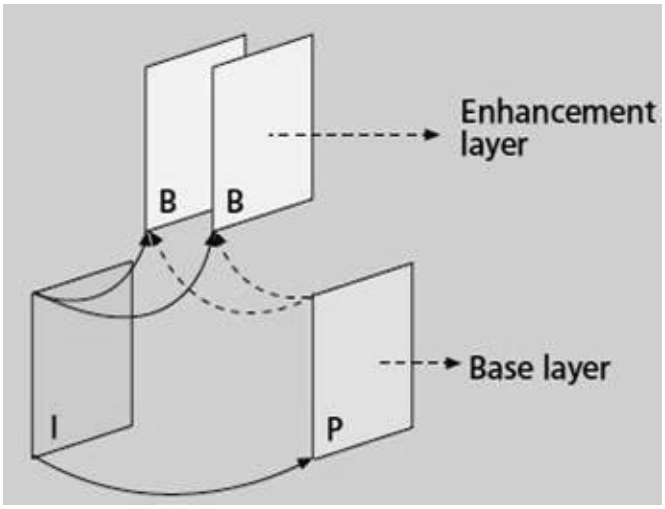


Fig 5: *Example for temporal scalable encoding: I and P frames form the base layer and B-frames form the enhancement layer.*

The scalability modes such as data partitioning, temporal, spatial, and signal-to-noise (SNR) has been standardized by MPEG. We have briefly reviewed the temporal and

spatial scalability modes as they are considered in the later discussion of the trace statistics [35]. With temporal scalable encoding, the enhancement layer frames are interleaved between base layer frames. Each enhancement layer frame is inter-coded with reference to the immediately preceding base layer frame and the immediately succeeding base layer frame as shown in Fig. 5 for a scenario where I and P frames form the base layer and B frames form the enhancement layer.

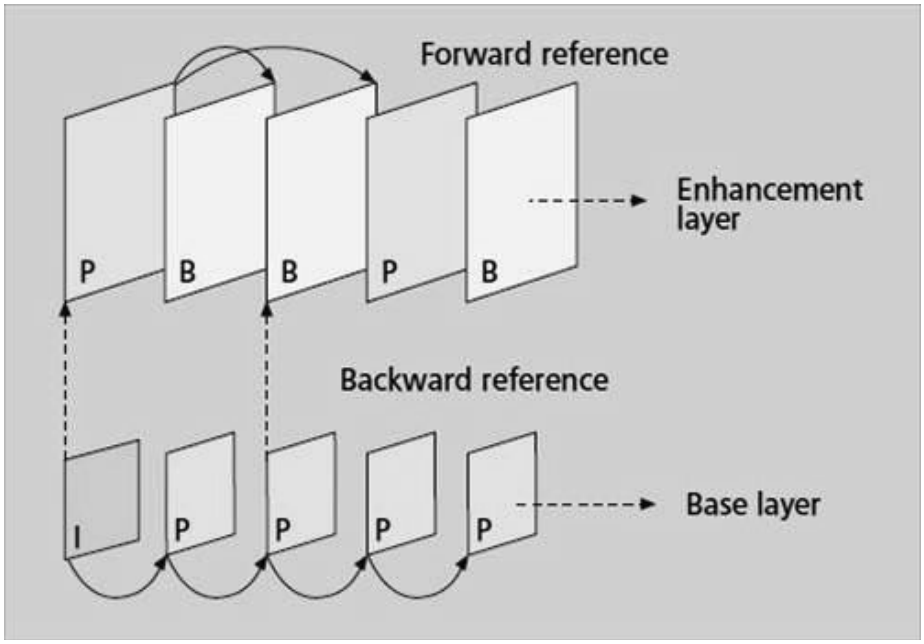


Fig 6. Example for spatial scalable encoding.

The down sampled video is encoded into a base layer stream consisting of I and P frames. The difference between the decoded and up-sampled base layer and the original video is encoded into the P and B frames in the enhancement layer.

The base layer of the temporal scalable encoding provides a basic video quality with a low frame rate. The frame rate is increased by adding the enhancement layer to the base layer. It should be noted that since each base layer frame is only encoded with reference to another base layer frame, the base layer can be decoded independently

of the enhancement layer. On the other hand, the enhancement layer requires the base layer for decoding since the enhancement layer frames are encoded with reference to base layer frames. With spatial scalability the base layer provides a small video format (e.g. QCIF) and the video format increases by adding the enhancement layer (e.g. to CIF). The base layer of the spatial scalable encoding can be up-sampled to give a coarse video at the larger format. To generate a spatial scalable encoding, the original (uncompressed) video is first down-sampled to the smaller base layer format and the down-sampled video is encoded employing the intra and inter coding techniques described above. A base layer consisting of only I and P frames is illustrated in Fig. 6. Subsequently the encoded base layer is decoded and up-sampled. The difference between a decoded and up-sampled base layer frame and the corresponding uncompressed frame is then encoded using the DCT transforms coding (and possibly inter-coding within the enhancement layer). More specifically, a given enhancement layer frame can be encoded with reference to the corresponding base layer frame, which is referred to as backward reference in this context, and with respect to a preceding frame in the enhancement layer, which serves as forward reference. In the example illustrated in Fig. 6 the enhancement layer frames are coded as either P or B frames. A P frame in the enhancement layer is coded with reference to the corresponding I frame in the base layer. A B frame in the enhancement layer is coded with reference to the corresponding P frame in the base layer and the preceding P frame in the enhancement layer.

We close this overview of scalable encoding by noting that apart from the layered coding considered here, a number of other methods have been developed in order to achieve scalable encoding. Fine granular scalability (FGS) encodes the video into a base layer and one enhancement layer [35]. The special property of the FGS

enhancement layer is that it can be cut anywhere at the granularity of bits allowing the video stream to finely adapt to changing network bandwidths. On the other hand, with conventional layered coding, the video stream can only adapt at the granularity of complete enhancement layers. The video is encoded into several streams (descriptions) with Multiple Description Coding (MDC) [23]. Each of the descriptions contributes to the decoded video quality. High video quality can be achieved by decoding all the descriptions whereas decoding an arbitrary subset of the descriptions will result in a lower quality. This is in contrast to conventional hierarchical layered videos where a received enhancement layer is useless if the corresponding base layer is missing. With wavelet transform coding, a video frame is not divided into blocks, as with the DCT-based MPEG coding [58]. Instead, by the use of wavelet transform, the entire frame is coded into several sub-bands. These methods used to achieve scalable video coding are beyond the scope of this article. This article is focused on the network performance evaluation for conventional non-scalable (single-layer) and layered (hierarchical) encoded video, for which traces are currently publicly available [27].

4.4 MPEG-4 VIDEO LAYERS

MPEG-4 video is broken up into a hierarchy of layers to help with error handling, random search and editing, and synchronization, for example with an audio bit stream. The first layer, from the top level, is known as the video sequence layer, and is any self-contained bit stream, for example a coded movie or advertisement. The second layer down consists of group of pictures that is composed of 1 or more groups of non-intra (P and/or B) pictures and/or intra (I) frames that will be

explained later in the dissertation and the third layer down is the picture layer itself. Finally, the next layer beneath is called the slice layer. Each slice is a contiguous sequence of roster ordered macro blocks, most often on a row basis in typical video applications, but not limited to this by the specification. Each slice consists of macro blocks, which are 16x16 arrays of luminance pixels, or picture data elements, with 2 8x8 arrays of associated chrominance pixels [10]. For transform coding the macroblocks can be further divided into separate 8x8 blocks. Each layer has its own unique 32 bit start code defined in the syntax to consist of 23 zero bits followed by one, then followed by 8 bits for the actual start code. These start codes can have as many zero bits as preferred preceding them.

4.4.1 Discrete Cosine Transform

In general, neighboring pixels within an image tend to be highly correlated. As such, it is desirable to use an invertible transform to concentrate randomness into fewer, de-correlated parameters. The Discrete Cosine Transform (DCT) has been shown to be near optimal for a large class of images in energy concentration and de-correlating. The DCT separates the signal into fundamental spatial frequencies that allow further processing techniques to reduce the accuracy of the DCT coefficients, which is more reliable and consistent with the Human Visual System (HVS) model.

The DCT/IDCT transform operations are described with Equations 1 & 2 respectively [18]:

$$f(\mu,\nu) = \frac{1}{4} c(\mu)c(\nu) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left[\frac{(2x+1)\mu\pi}{16}\right] \cos\left[\frac{(2y+1)\nu\pi}{16}\right]$$

$$c(\mu) = \frac{1}{\sqrt{2}} for \mu = 0$$

$$c(\mu) = 1 for \mu = 1, 2.....7$$

Equation 1: Forward Discrete Cosine Transform

$$f(x, y) = \frac{1}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} c(\mu) c(\nu) F(\mu, \nu) \cos\left[\frac{(2x+1)\mu\pi}{16}\right] \cos\left[\frac{(2y+1)\nu\pi}{16}\right]$$

Equation 2: Inverse Discrete Cosine Transform

The above equations are based on data blocks of 8x8 sizes. It is definitely possible to compute the DCT for other block sizes like 4x4 or 16x16 pixels, but the 8x8 size has become the standard as it represents an ideal compromise between adequate data decorrelation and reasonable computability. If the equations were solved directly each would normally require 1024 multiplies and 896 additions, but providentially with Fast Fourier Transform, various fast algorithms exist that make the calculations considerably faster.

4.5 MPEG-4 VIDEO STRUCTURE

The modeling approach in this thesis is intended to describe the common properties of all of the widely used *hybrid coding* standards, rather than any specific coding standard. Hybrid coding, like MPEG-4 or H.263, comprises *lossy intra frame coding* and *motion compensation* to exploit both spatial and temporal redundancy. The subject of the actual investigation is a set of VBR MPEG-4 traces taken from [16] and most of the video traces are taken from [27].

Three types of compressed frames are generated by a standard MPEG-4 encoder. I frames are compressed using intra-frame information only. P-frames are coded similarly to I-frames but with the addition of motion compensation in respect of the previous I-frames or P-frame. B-frames are similar to P-frames except that the prediction is bidirectional. Typically I-frames are the largest followed by P-frames and B-frames on the other hand require the lowest bandwidth. After coding, the frames are arranged in a deterministic order, which is called group of pictures (GOP, e.g., 'IBBPBBPBBPBB'). The GOP pattern is not specified by the standard and coders may use different patterns for subsequent GOPs. However, since many sequences are being coded with regular GOP patterns (often to simplify the codec design); developing a traffic model for such sequences has its merits.

There are two main levels of modeling: *frame level* and *GOP level*. The former attempt to catch the size of each frame while the latter takes a GOP as a whole and is not interested in the individual frame sizes. This research work is concerned with the GOP level thereby making the model independent of the MPEG-4 coding scheme in order to eliminate the deterministic alternation of frame types. The model can then be further refined to frame level.

4.6 MPEG–4 VIDEO COMPRESSION

MPEG–4 provides very efficient video coding covering the range from the very low bit rates of wireless communication to bit rates and quality levels beyond high-definition television (HDTV). In contrast to the frame-based video coding of MPEG–1 and H.263, MPEG–4 is object-based. Each scene is composed of video

objects (VOs) that are coded individually. (If scene segmentation is not available or useful, e.g., in very simple wireless video communication, the standard defines the entire scene as one VO.) Each VO may have several scalability layers (i.e., one base layer and one or several enhancement layers), which are referred to as video object layers (VOLs) in MPEG-4 terminology [5]. Each VOL in turn consists of an ordered sequence of snapshots in time, referred to as video object planes (VOPs). For each VOP the encoder processes the shape, motion and texture characteristics.

The shape information is encoded by bounding the VO with a rectangular box and then dividing the bounding box into MBs. Each MB is classified as lying:

• Inside the object

• On the object's border

• Outside the object (but inside the bounding box)

The border MBs are then shape coded. The texture coding is done on a per-block basis similar to the "frame-based standards (e.g. MPEG–4 and H.263). In an intra-coded (I) VOP the absolute texture values in each MB are DCT coded. The DCT coefficients are then quantized and variable-length coded length. In forward predicted (P) VOPs, each MB is predicted from the closest match in the preceding I (or P) VOP using motion vectors. In bi-directionally predicted (B) VOPs, each MB is predicted from the preceding I (or P) VOP and the succeeding P (or I) VOP. The prediction errors are DCT coded, quantized, and variable-length-coded. The I, P, and B VOPs are arranged in a periodic pattern referred to as a group of pictures (GoP). A typical GoP structure is IBBPBBPBBPBBPBB. For the transmission the shape,

motion and texture information is multiplexed at the MB level; that is, for a given MB the shape information is transmitted first, then the motion information, and then the texture information, then the shape information of the next MB, and so on. To combat the frequent transmission errors typical in wireless communication, MPEG–4 provides a number of error resilience and error concealment features [25], [42], [30], [53].

Error resilience and error concealment features are to be taken care of in error prone environment such as wireless networks.

4.7 STATISTICAL STUDY OF MPEG-4 ENCODED VIDEO TRACES

4.7.1 Video Trace Generation

In this section the generation of the video frame size traces is described. This process is illustrated in Fig. 7, which we refer throughout this section.

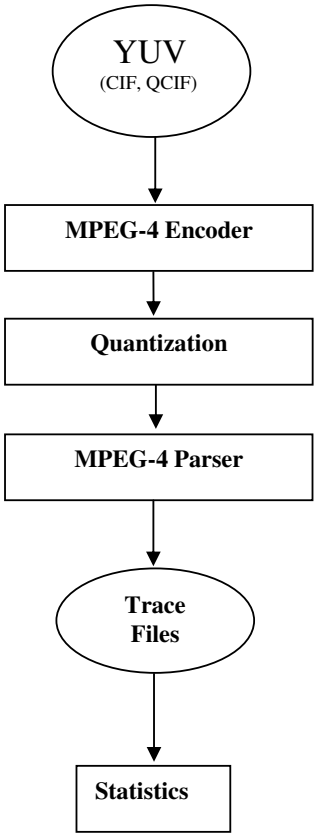


Fig 7: Generation of Frame Size Traces

The Video sequences (YUV) used for trace generation were taken from the Internet researcher lots of raw video sequences can be found in different research websites. The researcher chose chose the QCIF and CIF format because he was particularly interested in generating traces for the evaluation of wireless networking systems. It is expected that handheld wireless devices of next-generation wireless systems will typically have a screen size that corresponds to the QCIF and CIF video formats. The YUV video sequences for QCIF and CIF formats were created at a frame rate of 25 frames per second with 4:2:0 chrominance subsampling and quantization into 8 bits.

Random selection of 21 video sequences both from QCIF and CIF formats was made. The stored YUV frame sequences were used as input for the MPEG-4 encoder.

MPEG-4 Encoding Approach: All the raw Video sequences were encoded using MPEG-4 reference software [68]. The YUV information was encoded and converted into MPEG-4 bit streams using MPEG-4 reference software. We set the number of video objects to one (i.e., the entire scene is one video object). The width of the display is set to 176 pels, the height to 144 pels for QCIF format and 352 as width and 288 as height for CIF format. We used a pel depth of 8 b/pel. The single video object was encoded into a single video object layer. We set the video object layer frame rate (i.e., the rate at which video object planes are generated) to 25 frames/sec. The GoP pattern was set to IBBPBBPBBPBB. We encoded each video sequence with quantization parameters fixed at 20 for all three frame types so that, we can get the same quality for all video sequences.

Extracting Frame Sizes: The frame sizes (in bytes) of the individual encoded video frames by directly parsing the encoded MPEG-4 bit streams. Obtaining the frames sizes with some software would be less accurate. And therefore it was, directly parsed inside the source code for both the encoders and the frames sizes which were obtained directly from the actual encoded bit streams, that were stored in a temp buffer.

4.7.2 Statistical Analysis of MPEG-4 Traces

In this section a thorough statistical analysis of the generated MPEG-4 frame size traces was made for the analysis we introduced the following notation. Let N denote the number of video frames in a given trace. Let t denote the frame period (display time) of a given frame. Note that for almost all our MPEG–4 traces approximately $N = 300$ and $t = 40$ ms, which corresponds to a video runtime of about less than a minute. Let Xn, $n = 1, \ldots, N$, denote the number of bits in frame n (i.e., the frame size of frame n). Let G denote the number of frames per GoP. Let Ym, $m = 1, \ldots, N/G$, denote the number of bits in GoP m (i.e., the size of GoP m). Clearly,

$$Ym = \sum_{n}^{mG} = (m-1)\mathrm{G}+1Xn.$$

Frame Sizes for MPEG-4: Table 2 gives an overview of the statistical properties of the generated MPEG-4 traces. The mean $\bar{x}$ gives the average frame size. The coefficient of variation (defined as the standard deviation SX of the frame size divided by the average frame size $\bar{x}$) is a typical metric for the variability of the frame sizes; the larger the coefficient of variation, the more variable are the frame sizes.

Video Clip	Mean $\bar{x}$	Variance	SD Sx	CoV Sx/X	Number of Frames
Akiyo Cif	7821	44211	1202	0.153	300
Akiyo Qcif	3283	64516	803.2	0.224	300
Bridgefar Qcif	501.0	35514	188.4	0.376	2101
Carphone Qcif	5323	40555	1023	0.192	382
Claire Qcif	3244	35677	597.3	0.184	494
Football Cif	34900	1.395	11811	0.338	90

Foreman Cif	20187	2.997	5474	0.271	300
Foreman Qcif	6537	29940	1730	0.264	400
Grandma Qcif	1216	56902	754.3	0.620	870
Highway Cif	8014	25569	1599	0.199	2000
Highway Qcif	2387	33992	583.0	0.244	2000
Mother Cif	6132	63626	2522	0.411	300
Mother Qcif	2815	10228	1011	0.359	961
News Cif	22702	1.050	3241	0.142	300
News Qcif	8791	16152	1270	0.144	300
Paris Cif	49097	1.158	3404	0.069	1065
Salesman Qcif	3696	16511	1284	0.347	449
Silent Cif	11339	1.539	3923	0.345	300
Silent Qcif	4278	17881	1337	0.312	300
Walk Cif	32629	5.745	7579	0.232	376
Walk Qcif	10743	70200	2649	0.246	376

Table 2. An Overview of MPEG4 frame Statistics

In this MPEG-4 statistics the mean value is higher and so the covariance value is lower.

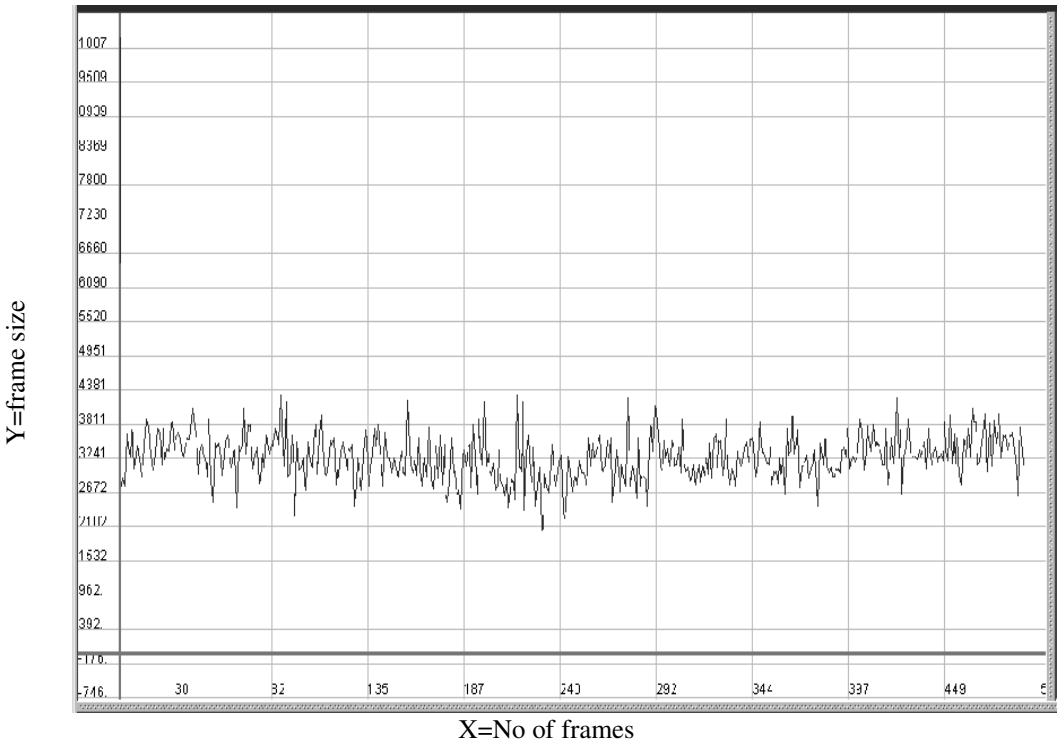


Fig 8. MPEG-4 Claire QCIF frame size.

Inspecting this trace closely, we are able to identify periods during which the frame sizes stay roughly at a fixed level; these periods appear to correspond to distinct scenes in the movie.

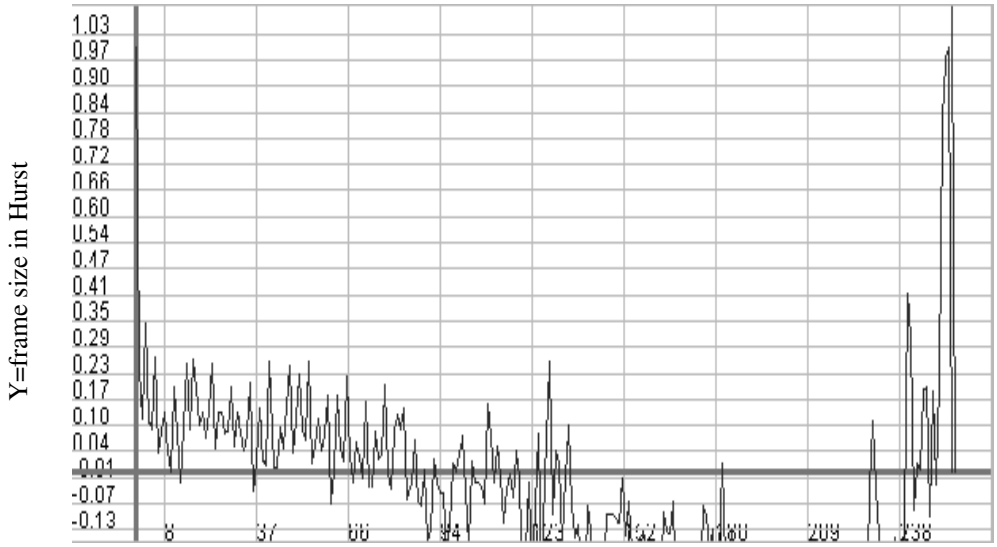


Fig 9: Autocorrelation function for Claire QCIF MPEG-4

Correlations and Long Range Dependence — Fig 9, gives the autocorrelation coefficient $\rho X(k)$ of the frame size sequence Xn, $n = 1, \ldots, N$, as a function of the lag k (in frames). The frame size correlations exhibit a periodic spike pattern that is superimposed on a decaying slope. The periodic spike pattern reflects the repetitive GoP pattern. The large positive spikes are due to (the typically large) I frames. An I frame is followed by a (typically midsized) p frame, which appears as a small positive spike. The subsequent B frames (typically small) show up as negative spikes. The decaying slope is characteristic of the long-term correlations in the encoded video.

The time-dependent statistics are important for network and traffic engineering since correlations in the video traffic can have a significant impact on the performance of

packet-switched networks. Several studies [10] have found that the losses and/or delays of queuing systems are considerably larger for positively correlated input traffic than uncorrelated input traffic. Carefully designed Variable Bit Rate (VBR) traffic models are able to capture the relevant range of correlations and predict the system performance accurately. For these reasons it is important to analyze the long-range correlations of the video traces. These long-range correlations are formally characterized as self-similarity or long-range dependence (LRD) [36]. Naturally, long-range dependent traffic is busty (highly variable) over a wide range of timescales. The cumulative effect of the correlations for large lags is significant and gives rise to the large losses and/or delays found in long-range dependent traffic (even though the correlations for large lags may be individually small). The Hurst parameter is a succinct metric for long-range dependence (i.e., the degree of self-similarity). Generally speaking, time series without long-range dependence have a Hurst parameter of 0.5. Hurst parameters between 0.5 and 1 indicate long-range dependence. Additionally, larger Hurst parameters indicate a higher degree of long-range dependence.

MPEG-4 encoder is used to extract the trace files for estimating the Hurst. The trace files contain video frame sequence and frame size. We estimate the Hurst parameter for Twenty One video traffic traces using monofractal wavelet-based estimators. We employ Daubechies' wavelets of genus 3 (three vanishing moments). The results are summarized in Table 3. For each estimate of *H*, we report the range of octaves where the linear regression is performed. These ranges are chosen by visual inspection of the Logscale diagrams and identification of the linear region.

Video Clip	Clip Format	Range	Mpeg 4	Number of Frames
Akiyo	CIF	2-8	0.747	300
Akiyo	QCIF	2-8	0.775	300
Bridgefar	QCIF	2-8	0.793	2101
Carphone	QCIF	2-8	1.100	382
Claire	QCIF	2-8	0.829	494
Football	CIF	2-8	1.872	90
Foreman	CIF	2-8	1.069	300
Foreman	QCIF	2-8	0.985	400
Grandma	QCIF	2-8	0.906	870
Highway	CIF	2-8	1.098	2000
Highway	QCIF	2-8	1.170	2000
Mother	CIF	2-8	1.181	300
Mother	QCIF	2-8	0.837	961
News	CIF	2-8	0.602	300
News	QCIF	2-8	0.508	300
Paris	CIF	2-8	0.671	1065
Salesman	QCIF	2-8	0.959	449
Silent	CIF	2-8	1.231	300
Silent	QCIF	2-8	1.178	300
Walk	CIF	2-8	0.931	376
Walk	QCIF	2-8	1.002	376

Table 3. With Hurst Parameters for MPEG-4 using Wavelet estimation

The linear region in Fig. 10 shows the visual indication of the range from which the value of Hurst parameter is calculated.

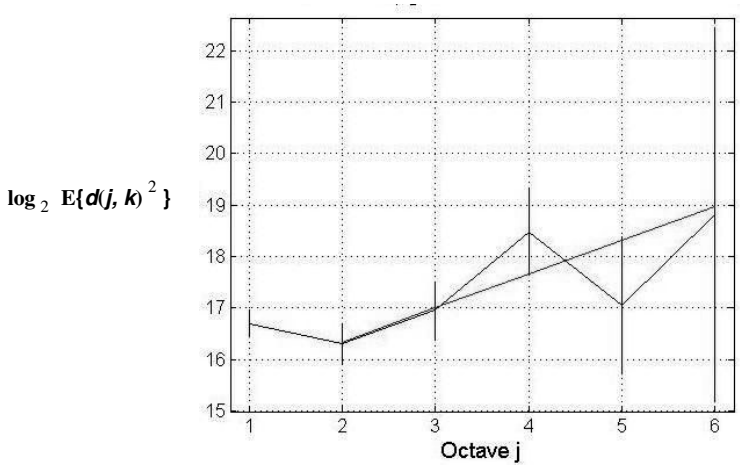


Fig. 10: Logscale diagram of encoded MPEG4 "Claire QCIF" sequence

Logscale diagrams of the analyzed traces have similar shapes. A typical example is shown in Fig. 10 for MPEG4 encoded "Claire QCIF" video sequence. Logscale diagrams exhibit a linear relationship between $\log_2 \mathrm{E}\{d(j, k)^2\}$ ($\log_2 Sq(j)$) and j for the largest values of j (the coarsest octaves or time scales). The linear region typically begins at $j = 2$.

The monofractal estimator of H produces different results for MPEG4, as indicated by Table 3.

Results shown in Table 3, gives the Hurst Parameters for the video sequences encoded by MPEG-4. The Video sequence Claire QCIF gives 0.829 as the Hurst value for the whole video sequence. Hurst parameter is around 0.8 for MPEG-4 encoded video sequence.

4.8 CONCLUSION

In this chapter we have studied and presented a publically available library of frame size traces of MPEG-4 encoded videos. We have encoded over twenty-one videos of more than sixty minutes length each. For each video we have generated MPEG-4 encoded video traces at single quality level so that the results will be precise.

A detailed statistical analysis of the generated MPEG-4 video traces was made as well as an analysis of MPEG-4 encodings, which have variable frame periods. A

statistical study of rate and frame size trace was made. The traces facilitate the analysis of MPEG-4 frame sizes in conjunction with their associated frame periods. The study has found that the traces are highly variable in their frame sizes and bit rates. Many of the traces show clear indications of long range dependence properties. The cumulative effect of the correlations for large lags is significant in large frame sizes MPEG-4 encoded trace and gives rise to the large losses and/or delays found for long-range dependent traffic.

CHAPTER V

5. ANALYSIS OF VIDEO TRACES USING H.264 VIDEO COMPRESSION STANDARD

5.1 INTRODUCTION

H.264/AVC is the most recent and most efficient video coding standard (Joint Video Team of ITU-T and ISO/IEC JTC 1, [28]). The MPEG-2 video coding standard was developed during late nineties as an extension of MPEG-1 video capability with additional support of interlaced video coding which was an innovative and enabled technology for digital television systems throughout the world. It is extensively applied for the transmission of High Definition (HD) and Standard Definition (SD) TV signals over satellites, cables, and terrestrial emission and storage of high-quality SD video signals onto DVDs.

The rapid growth of services and the popularity of high definition TV are creating greater requirements for higher coding efficiency. Nevertheless, other transmission media such as Cable Modem or xDSL offer very lower data rates than broadcast channels, and advanced coding efficiency can allow the transmission of more video channels or higher quality video representations within the existing capability of the digital transmission.

New applications may be used over existing and future networks. This raises the question of how to handle this variety of networks and applications. To deal with this

need for flexibility and customizability, the H.264/AVC design covers a Video Coding Layer (VCL), which is designed to proficiently represent the video content, and a Network Abstraction Layer (NAL) [31], which formats the VCL representation of the video and provides header information in a manner appropriate for conveyance by a variety of transport layers or storage media (Fig. 11).

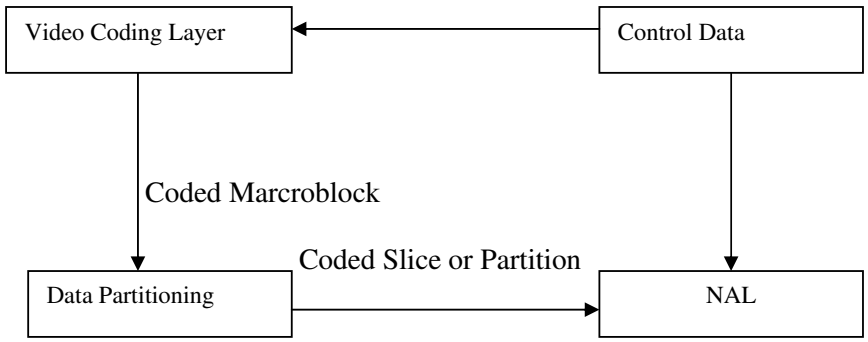


Fig 11: Model of H.264/AVC Video Encoder

5.2 OVERVIEW OF H.264/AVC VIDEO CODING STANDARD

In this session the whole working and design structure of H.264 encoder is explained. It gives an overall idea of how H.264 encoder works in encoding the videos.

5.2.1 Enhanced Motion Prediction

Improvement in MCP has been one of the underlying reasons for the increase in coding efficiency achieved by modern standards and H.264 is no different. Some of the enhancements that found their way into H.264 standard are as listed below [59]

Variable block size MCP: Various coding modes are specified for P macroblocks based on their partitioning. A P macroblock of size 16x16 can be

segmented into smaller regions for MCP with luma block sizes of 8x16 , 16x 8 and 8x 8 samples and each 8x8 region can further be partitioned into 4x8, 8x4 or 4x4 regions of luma samples (and corresponding chroma samples). The prediction signal for each *NxM* region is specified by a translational MV and a picture reference index that points to a reference picture from the decoded picture buffer. MCP for smaller regions than 8x8· uses the same reference index for predicting all sub-blocks, as the index for 8x8 region.

Multipicture MCP: MCP in H.264 uses more than just one or two previously decoded pictures, allowing the exploitation of long-term statistical dependencies as is the case with backgrounds, scene cuts etc.

Fractional-sample accuracy: To obtain a better motion representation, MC is performed in units of one-quarter of the horizontal and vertical distance between luma samples and with one-eighth sample accuracy for chroma.

MVs over picture boundaries: The H.264 syntax allows MVs to point over picture boundaries, solving the problem of motion representation for samples at the boundary of a picture.

Weighted prediction in P and B slices: Biprediction has typically been performed with a simple (½, ½) averaging of the two MCP signals and the prediction for P slices has not used weighting. However H.264 encoder can specify either temporally derived or explicitly chosen weights and offsets for P and B slices.

5.2.2 Use of Small Block-Size Integer Transform

One of the most significant improvements in H.264 is the improved Intra and Inter prediction processes, as a result of which, the spatial correlation among the residual

coefficients is small. This means that a transform as large as an 8x8 block transform is not needed. Also the visual benefits (lesser 18 mosquito noise), smaller processing word length and fewer computations that result from a smaller transform motivated the use of a 4x4 spatial transform [26]. In addition, the transform is a simple separable integer transform that has properties similar to DCT. This provides the advantage of smaller decoding complexity because the inverse transform is now defined by exact integer operations, avoiding any mismatches. The integer transform matrix is given by

$$H=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Since the inverse transform is defined by exact integer operations, inverse-transform mismatches are avoided. The basic transform coding process is very similar to that of previous standards. At the encoder the process includes a forward transform, zig-zag scanning, scaling and rounding as the quantization process followed by entropy coding. At the decoder, the inverse of the encoding process is performed except for the rounding. More details on the specific aspects of the transform in H.264/AVC can be found in [23].

It has been mentioned before that Intra modes are intended for coding of smooth areas. For that reason the DC coefficients undergo a second transform with the result that we have transform coefficients covering the whole macroblock. An additional 2x2 transform is also applied to the DC coefficients of the four 4x4 blocks of each chroma component. The procedure for a chroma block is illustrated in Fig. 12. The small blocks inside the larger blocks represent DC coefficients of each of the four

4x4 chroma blocks of a chroma component of a macroblock numbered as 0, 1, 2, and 3. The two indices correspond to the indices of the 2x2 inverse Hadamard transform.

To clarify the idea behind these repeated transforms, let us point to a general property of a two-dimensional transform of very smooth content (where sample correlation approaches 1). In that situation the reconstruction accuracy is proportional to the inverse of the one dimensional size of the transform. Hence, for a very smooth area, the reconstruction error with a transform covering the complete 8x8 block is halved compared to using only 4x4 transform. A similar rationale can be used for the second transform connected to the INTRA-16 x 16 modes [29].

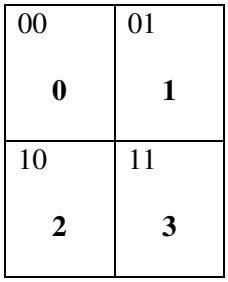


Fig. 12: Repatative transform for chroma blocks. The four blocks numbered 0 to 3 indicate the four chroma blocks of a chroma component of a macroblock.

There are several reasons for using a smaller size transform:

• One of the main improvements of the present standard is the improved prediction process for both inter and intra. Consequently the residual signal has less spatial correlation. This generally means that the transform has less to offer concerning decorrelation. This also means that a 4x4 transform is essentially as efficient in removing statistical correlation as a larger transform

• With similar objective compression capability, the smaller 4x4 transform has visual benefits resulting in less noise around edges (referred to as “mosquito noise” or "ringing" artifacts).

• The smaller transform requires less computations and a smaller processing word length. Since the transformation process for H.264/AVC involves only adds and shifts, it is also specified such that mismatch between encoder and decoder is avoided (this has been a problem with earlier 8x8 DCT standards)

5.2.3 Enhanced Entropy Coding

H.264 supports two methods for entropy coding viz. CAVLC and CABAC. Context-adaptive VLC (CAVLC) uses multiple VLC tables that are selected based on the context of source symbols. Since the VLC tables are context dependent, coding efficiency is higher than when using a single VLC table with “run + level” or “run + level + last” coding, as found in previous standards. The VLC tables for various syntax elements are switched depending on already transmitted syntax elements. Since the VLC tables are designed to match the corresponding conditioned statistics, the entropy coding performance is improved in comparison to schemes using a single VLC table.

In the CAVLC entropy coding, the number of nonzero quantized coefficients (N) and the original size and position of the coefficients are separately coded. After zigzag scanning of transform coefficients, their statistical distribution typically shows large values for the low frequency part decreasing to small values later in the scan for the high-frequency part. Context-adaptive binary arithmetic coding (CABAC) improves the efficiency further because it not only uses context

conditional probability estimates but also tries to adapt to non-stationary statistical behavior, besides offering the usual advantages of arithmetic coding (e.g. non-integer number of bits for encoding). The usage of adaptive codes permits adaptation to non-stationary symbol statistics. Another important property of CABAC is its context modeling. To estimate conditional probabilities, previously coded syntax elements are used. Conditional probabilities are used for switching several expected probability models [28]. In H.264/AVC, the arithmetic coding engine and its linked probability estimation are specified as multiplication-free low-complexity methods using only shifts and table hook-ups. Compared to CAVLC, CABAC typically provides a reduction in bit rate between 5-15% and the highest gains are typically obtained when coding interlaced TV signals.

5.2.4 Use of In-loop Deblocking Filter

Blocking is one of the most unpleasant artifacts commonly found in block-based codec's. The poorer MCP for samples at the edges (compared to interior samples) and the edge discontinuities introduced by block transforms give rise to such visible blocking artifacts. Block edges are typically reconstructed with less accuracy than interior pixels and "blocking" is generally considered to be one of the most visible artifacts with the present compression methods. For this reason, H.264 defines an adaptive in-loop deblocking filter that reduces blockiness while retaining the true sharp edges in the scene, improving the subjective quality of the video considerably by the strength of the deblocking filter that is controlled by the values of several syntax elements.

If a comparatively large difference between samples near a block edge is measured, it is quite possible to form a blocking artifact and should therefore be reduced.

Moreover, if the scale of that difference is so much more that it cannot be explained by the coarseness of the quantization used in the encoding, the edge is more likely to show the original behavior of the source picture and should not be smoothed over.

The sharpness of the content is mostly unchanged while reducing the blockiness. As a result, the subjective quality is considerably improved. The filter reduces bit rate typically by 5-10% while producing the same objective quality as the non-filtered video. Fig. 13 illustrates the performance of the deblocking filter [11].

Fig. 13: Performance of the deblocking filter for highly compressed pictures. Left: Deblocking filter used, right:Deblocking filter not used

5.2.5 Network Abstraction Layer

The network abstraction layer (NAL) is designed to provide *"network friendliness"* to enable simple and effective customization of the use of the VCL for a broad variety of systems.

The NAL facilitates the ability to map H.264/AVC VCL data to transport layers such as

• RTP/IP for any kind of real-time wire-line and wireless Internet services

• File formats, e.g. ISO MP4 for storage and MMS

• H.32X for wired and wireless conversational services

• MPEG-2 systems for broadcasting services, etc.

The full range of customization of the video content to satisfy the needs of each particular application is beyond the scope of the H.264/AVC standardization, but the design of the NAL anticipates a variety of such mappings. Some key concepts of the NAL are NAL units, byte stream, and packet format uses of NAL units, parameter sets, and access units.

NAL units are nothing but coded video data, which is ordered into NAL units, each of which is effectively a packet that contains an integer number of bytes. First byte of each NAL unit is a header byte, which contains an indication of the type of data in the NAL unit. Other remaining bytes consist of payload data of the type indicated by the header. A sequence of NAL units that are generated by an encoder is referred to as a NAL unit stream. Each NAL unit is prefixed by a particular pattern of three bytes called a start code prefix in the byte stream format. Searching the coded data for the unique start code prefix pattern can then identify the boundaries of the NAL unit. In IP /RTP systems, the coded data is carried in packets that are framed by the system transport protocol, and identification of the boundaries of NAL units within the packets can be established without use of start code prefix patterns. In such systems, the data-carrying capacity would be wasted when the start code prefixes are added, so as an alternative the NAL units can be carried in data packets without start code prefixes.

A parameter set contains the information that is expected to hardly change and offers the decoding of a large number of VCL NAL units. There are two types of parameter sets-sequence and Picture. Sequence parameter sets, apply to a series of consecutive coded video pictures called a coded video sequence, and Picture parameter sets,

apply to the decoding of one or more individual pictures within a coded video sequence.

A set of NAL units in a specific form is called an access unit. The decoding of each access unit gives out one decoded picture.

5.2.6 H.264 Video Coding Layer

In the block-based hybrid video coding approach, each coded picture is represented in block-shaped units of related luma and chroma blocks called *macro blocks*. The basic source-coding algorithm is a hybrid of inter-picture prediction to exploit temporal statistical dependencies and transform coding of the prediction residual to exploit spatial statistical dependencies. There is no coding element in the VCL that provides major improvements in compression efficiency in relation to prior video coding standards. It is rather a plurality of smaller enhancement that adds up to the significant gain.

H.264/AVC coded video consists of a sequence of *coded pictures*. The coded picture corresponds to either an entire *frame* or a single *field.* A frame of video can be regarded to contain two interleaved fields, a top field and a bottom field. The top field contains even-numbered rows 0, 2, 4, etc, H/2-1 with H being the number of rows of the frame. The bottom field consists of odd-numbered rows that start with the second line of the frame. If the two fields of a frame are captured at different time instants, the frame is referred to as an interlaced frame, and otherwise it is referred to as a progressive frame. The coding representation in H.264/AVC is primarily agnostic with respect to this video characteristic, i.e., the underlying interlaced or progressive timing of the original captured pictures [21].

A picture is partitioned into fixed-size macroblocks each of which cover a rectangular picture area of 16x16 samples of the luma component and 8x8 samples of each of the two chroma components. As the human visual system is more sensitive to luma than chroma, H.264/AVC uses a sampling structure in which the chroma component has one-fourth of the number of samples than the luma component, half the number of samples of both the horizontal and vertical dimensions.

Slices are a sequence of macroblocks which are processed in the order of a roster scan when not using flexible macroblock ordering (FMO).A picture is divided into several sizes. A picture is therefore a collection of one or more slices in H.264/AVC. Slices are self-sufficient in the sense that given the current sequence and picture parameter sets, their syntax elements that can be parsed from the bitstream and the samples in the area of the picture that the slice represents can be precisely decoded without use of data from other slices provided that used reference pictures are alike at encoder and decoder. Some more information from other slices may be required to apply the deblocking filter across slice boundaries. By introducing the concept of *slice groups,* FMO transforms the way in which the pictures are partitioned into slices and macroblocks. Every slice group is a set of macroblocks that are defined by a *macroblock to slice group map*, which is specified by the content of the picture parameter set and some information from slice headers. The macroblock to slice group map contains a slice group identification number for each macroblock in the picture, denoting which slice group the associated macroblock belongs to. Every slice group can be partitioned into one or more slices; a slice is a sequence of macroblocks within the same slice group that is processed in the order of a raster scan within the set of macroblocks of a particular slice group. With FMO, a picture

can be split into many macroblock scanning patterns such as a dispersed macroblock allocation, interleaved slices, one or more “foreground” slice groups and a “leftover” slice group, or a checker-board type of mapping.

Whether FMO is in use or not, each slice can be coded with different coding types as follows:

I slice: All the macroblocks of the slice are coded using intra-prediction.

P slice: With I slice, some macroblocks of the P slice can also be coded using inter-prediction with at most *one* motion compensated prediction signal per prediction block.

B slice: With coding types available in a P slice, some macroblocks of the B slice can also be coded using inter-prediction with *two* motion-compensated prediction signals per prediction block. Previous three coding types are the same as those in previous standards with the exclusion of the use of reference pictures as given below. The following two coding types for slices are new:

SP slice: Switching P slice that is coded such that efficient switching between different precoded pictures becomes possible.

SI slice: Switching I slice that allows an exact match of a macroblock in an SP slice for random access and error recovery purposes.

In encoding and decoding process of macroblocks all luma and chroma samples of a macroblock are either spatially or temporally predicted and the resulting prediction residual is encoded using transform coding [28]. For transform coding purposes, each color component of the prediction residual signal is subdivided into smaller 4x4 blocks. Each block is transformed using an integer transform, and the transform coefficients are quantized and encoded using entropy coding methods. Fig. 8 shows a block diagram of the video coding layer for a macroblock. The input video signal is

split into macroblocks, the association of macroblocks to slice groups and slices is selected, and then each macroblock of each slice is processed as shown. A proficient parallel processing of macroblocks will be feasible when there are various slices in the picture.

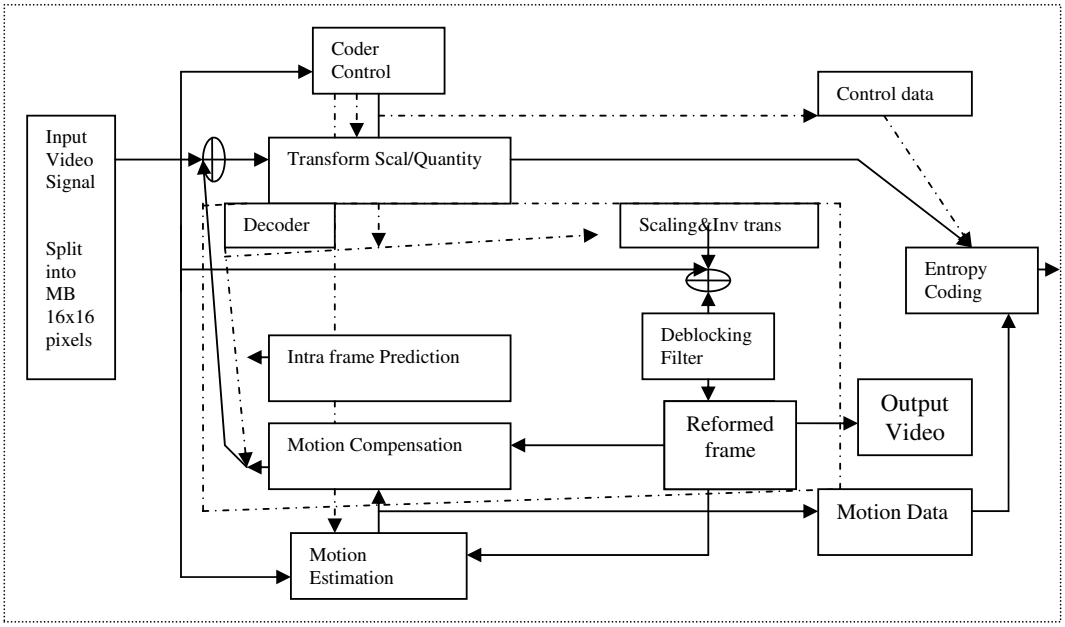


Fig. 14: Block diagram for H.264/AVC for a macroblock.

Each macroblock can be transmitted in one of several coding types depending on the slice-coding types; there are three types of frame predictions. Intra, Inter P slice and Inter B slice. . In all slice-coding types, the following types of intra-coding are supported, which are denoted as Intra 4x4 or Intra 16x16 together with chroma prediction and I PCM prediction modes.

The Intra 4x4 mode is based on predicting each 4x4 luma block separately and is well suited for coding of parts of a picture with significant detail. The Intra 16x16

mode, on the other hand, does prediction of the whole 16x16 luma block and is more suited for coding very smooth areas of a picture. The smooth areas of the luma block are nothing but the contrast part of the video frames. Together with these two types of luma prediction another separate chroma prediction is carried out. As an alternative to Intra 4x4 and Intra 16x16, the I PCM coding type allows the encoder to simply bypass the prediction and transform coding processes and instead directly send the values of the encoded samples.

In Inter P slice prediction, together with the intra-macroblock coding types, various *predictive* or motion-compensated coding types are specified as P macroblock types. Every P type macroblock corresponds to a particular partition of the macroblock into the block shapes used for motion-compensated prediction. Marcoblock Partitions that are with luma block sizes of 16x16, 16x8, 8x16, and 8x8 samples are supported by the syntax. Sometimes partitions with 8x8 samples are selected and an additional syntax element for each 8x8 partition is transmitted. This syntax element states whether the resulting 8x8 partition is further partitioned into partitions of 8x4, 4x8, or 4x4 luma samples and corresponding chroma samples.

In Inter-frame Prediction of B Slices the concept of B slices is generalized in H.264/AVC. Other pictures can be reference pictures containing B slices for motion-compensated prediction, depending on the memory management control operation of the multi-picture buffering. Thus, the substantial difference between B and P slices is that B slices are coded in a manner in which some macroblocks or blocks may use a weighted average of two distinct motion-compensated prediction values for building the prediction signal. B slices utilize two distinct lists of reference pictures, which are referred to as the first (list 0) and second (list 1) reference picture lists, respectively. Which pictures are actually located in each reference picture list is an

issue of the multi-picture buffer control and an operation very similar to the conventional MPEG-2 B pictures can be performed if desired by the encoder.

In B slices, four different types of inter-picture prediction are supported: list 0, list 1, bi-predictive, and direct prediction. For the bi-predictive mode, the prediction signal is formed by a weighted average of motion-compensated list 0 and list 1 prediction signals. The direct prediction mode is inferred from previously transmitted syntax elements and can be either list 0 or list 1 prediction or bipredictive [33].

B slices use the same macroblock partitioning as P slices. Other than the P 16x16, P 16x8, P 8x16, P 8x8, and the intra-coding types, bi-predictive prediction and another type of prediction called direct prediction, are provided. For each 16x16, 16x8, 8x16, and 8x8 partition, the prediction method (list 0, list 1, bi-predictive) can be chosen separately. An 8x8 partition of a B macroblock can also be coded in direct mode. If no prediction error signal is transmitted for a direct macroblock mode, it is also referred to as B Skip mode and can be coded very efficiently similar to the P_Skip mode in P slices. The motion vector coding is similar to that of P slices with appropriate modifications because neighboring blocks may be coded using different prediction modes.

The new standard H.264/AVC is designed for technical solutions to include the following areas of applications:

- Video-on-demand or multimedia streaming services over ISDN, Cable Modem, DSL, LAN, wireless networks, etc.
- Broadcasting over DSL, cable, Modem, satellite, terrestrial, etc.
- Serial or Interactive storage on magnetic and optical devices, DVD, etc.
- Colloquial services over Ethernet, wireless, ISDN, LAN, DSL, mobile networks, modems, etc.

Multimedia Messaging Services (MMS) over Ethernet, wireless, ISDN, LAN, DSL, mobile networks, modems, etc.

5.3 STATISTICAL STUDY OF H.264 ENCODED VIDEO TRACES

5.3.1 Video Trace Generation

The video frame size trace generation for H.264 encoder has been achieved in this section. This process is illustrated in Fig. 3, which is referred throughout this section.

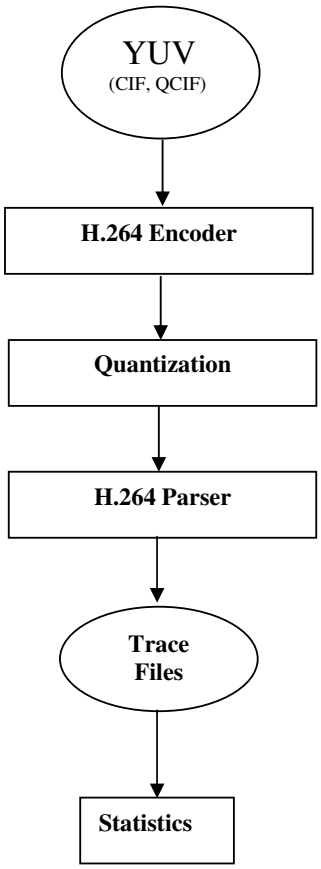


Fig 15: Generation of Frame Size Traces

The same process which is used to generate the trace files for MPEG-4 is used here in H.264 trace generation. The only difference is H.264 encoder is used instead of MPEG-4 encoder. The YUV video sequences for QCIF and CIF formats were created at a frame rate of 25 frames per second with 4:2:0 chrominance sub sampling and quantization into 8 bits. Twenty one video sequences were selected randomly both from QCIF and CIF formats. The stored YUV frame sequences were used as input for the H.264 encoder.

H.264 Encoding Approach: The uncompressed YUV information was encoded into an H.264 bitstream with the tml encoder (v.10.0) and surely not encoded in real time; thus, there was no encoder bottleneck. The tml encoder parameters were set to encode in the QCIF (176 × 144 pel) and CIF (352 × 288) video formats at a fixed reference frame rate of 25 frames/sec to which Inter Block search was enabled from 16×16 till 4×4 was used to improve Loop filter for improved video quality. Inter-motion search range was set to 16 and the number for previous frames used for Inter-motion search was set to 5. Weighted Prediction was disabled for both P and B frames. Bi Predictive Motion Estimation was also disabled. Each video sequence was encoded at 25 frames per second with quantization parameter set to 20 for all three frame types so as to get the same quality for all video sequences.

Extracting Frame Sizes: Frame sizes (in bytes) of the individual encoded video frames were obtained by directly parsing the encoded H.264 bitstreams. Obtaining the frames sizes with some software would be less accurate and so the researcher, we directly parsed inside the source code for both the encoders and the frames sizes were obtained directly from the actual encoded bitstreams, which were stored in a temp buffer.

5.3.2 Statistical Analysis of H.264 Traces

In this section a thorough statistical analysis of the generated H.264 frame size traces are made.. Let N denote the number of frames in a given video trace. Let Xn; $n = 1, \ldots, N$, denote the number of bits in frame n (i.e., the frame size of frame n). Let tn; $n = 1, \ldots, N$, denote the frame period (display time) of frame n in milliseconds. Let Tn; $n = 1, \ldots, N$, denote the cumulative display time up to (and including) frame n, that is, $Tn = \sum^{n}_{k=1} tk$ (define $T0 = 0$). The trace gives on line n; $n = 1, \ldots, N$, the cumulative display time Tn-1 (up to frame $n - 1$), the type (I, P or PB) of frame n, and the frame size Xn in bytes. As illustrated by the trace file, the Tn's are integer multiples of the basic (reference) frame period $\Delta = 40$ ms of the H.264 encoder. For VBR encodings (i.e., without a specified target bit rate), the H.264 encoder typically does not skip any frames. Most encoded frames have a frame period of 2Δ. This is because the encoder produces mostly PB frames (i.e., two consecutive frames are encoded as one entity).

Video Clip	Mean X	Variance	SD Sx	CoV Sx/X	Number of Frames
Akiyo Cif	1961	1.055	3248	1.65	300
Akiyo Qcif	645.9	16311	1277	1.97	300
Bridgefar Qcif	207.9	16013	126.5	0.608	2101
Carphone Qcif	3314	1.076	3280	0.989	382
Claire Qcif	773.1	11848	1088	1.40	494
Football Cif	29408	4.575	21391	0.727	90
Foreman Cif	10967	1.190	10909	0.994	300
Foreman Qcif	3029	92603	3043	1.004	400
Grandma Qcif	828.9	17413	1319	1.591	870
Highway Cif	5006	1.614	1324	0.264	2000
Highway Qcif	1674	17554	1324	0.790	2000
Mother Cif	3384	1.750	4184	1.236	300
Mother Qcif	1250	32603	1805	1.444	961

News Cif	5685	3.628	6023	1.059	300
News Qcif	2076	56890	2385	1.148	300
Paris Cif	9470	7.180	8473	0.861	1065
Salesman Qcif	1376	50486	2246	1.632	449
Silent Cif	6343	4.801	6929	1.092	300
Silent Qcif	2233	51978	2279	1.02	300
Walk Cif	23597	2.771	16646	0.705	376
Walk Qcif	6829	9.767	5562	0.814	376

Table 4:An Overview of H.264 frame Statistics

If no frame is skipped, the PB frame has a frame period of 2Δ when emitted by the encoder; at the decoder, however, the B frame is displayed first for a period of Δ and then the P frame for a period of Δ.

Frame Sizes of H.264: Tables 4 gives an overview of the statistical properties of the generated H.264 traces. The mean *X* gives the average frame size. The coefficient of variation (defined as the standard deviation *SX* of the frame size divided by the average frame size *X*) is a typical metric for the variability of the frame sizes.

The Claire QCIF encoding has an average frame size of 773 bytes, while Claire QCIF encoded by MPEG-4 has an average frame size of 3244 bytes. The more efficient VBR encoding, however, has a larger variability of frame sizes. It is important to note that for variable frame period H.264 encoded video, the frame sizes are only one component of the video stream statistics. For the complete picture we need to consider the frame sizes in conjunction with their associated frame periods. Clearly, if the larger frame sizes of the VBR H.264 encodings were associated with larger frame periods, and vice versa, the larger frame periods could be used to smooth out the larger frames. Figure 6 gives the frame size *Xn* for Claire

QCIF video trace. a pronounced bimodal distribution of the frame sizes. This is because the encoder typically produces:

• P frames with an average size of roughly 3 kbytes

• PB frames with an average size of 6 kbytes

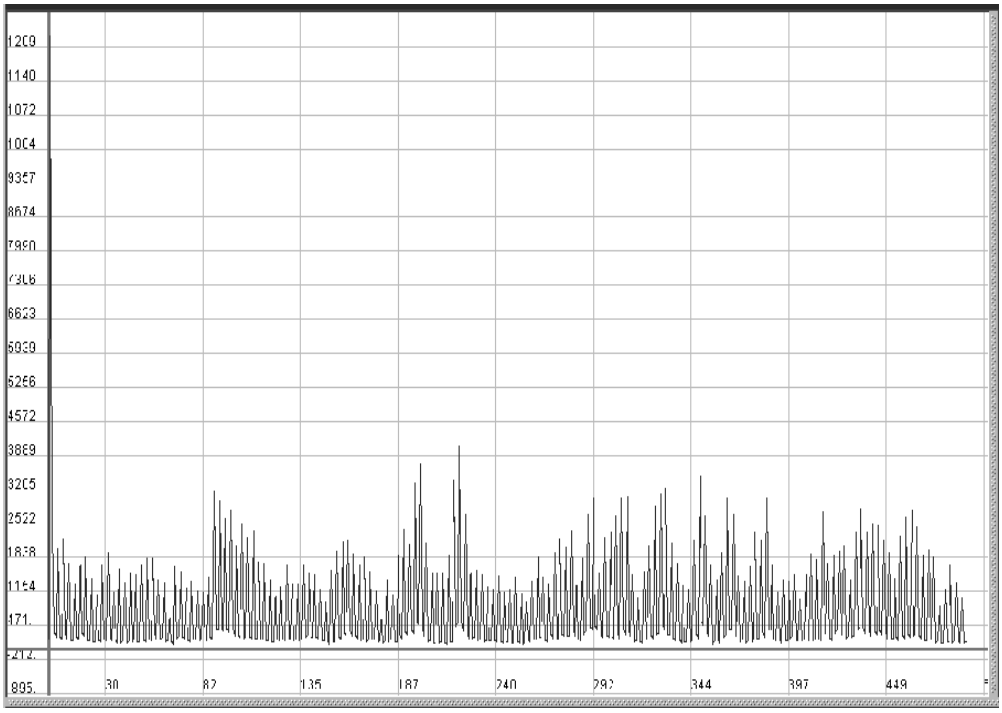
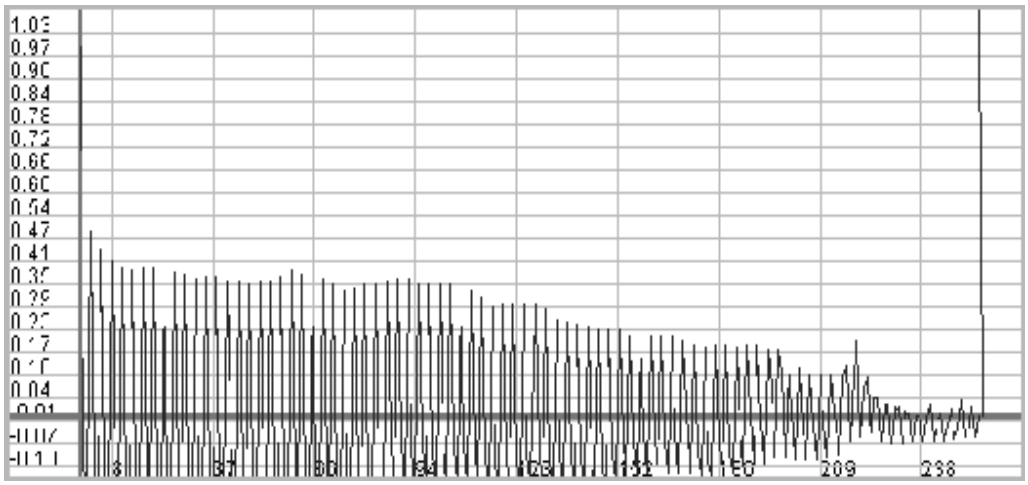


Fig 16: H.264 encoded Claire QCIF frame size

It is observed from Fig. 16 that the rate trace of the VBR encoding of *Claire QCIF* exhibits fast timescale fluctuations but not underlying slow timescale fluctuations. However, there are occasional “spikes” in the frame trace that are roughly four to six times higher than the average rate. These spikes can be effectively smoothed out by averaging the trace over (non-overlapping) blocks of roughly 10 or more reference frame periods. Smoothing is also highly effective in reducing the fast timescale fluctuations.

Y=frame size in Hurst



X=No of frames

Fig 17: H.264 Autocorrelation for Claire QCIF

Correlations and Long-Range Dependence: Fig 17, gives the autocorrelation coefficient of the frame size trace $\rho F(k)$ and the autocorrelation coefficient of the sampled rate trace $\rho R(k)$ as a function of the lag k (in reference frame periods of length Δ) for the generated H.264 traces over 494 reference frame periods of length Δ. It is observed that the autocorrelation function for the encoding of Claire QCIF decays very slowly.

Video Clip	Clip Format	Range	H.264/AVC	Number of Frames
Akiyo	CIF	2-8	0.624	300
Akiyo	QCIF	2-8	0.662	300
Bridgefar	QCIF	2-8	0.420	2101
Carphone	QCIF	2-8	0.559	382
Claire	QCIF	2-8	0.630	494
Football	CIF	2-8	0.834	90
Foreman	CIF	2-8	0.383	300
Foreman	QCIF	2-8	0.638	400
Grandma	QCIF	2-8	0.523	870
Highway	CIF	2-8	0.764	2000
Highway	QCIF	2-8	0.764	2000

Mother	CIF	2-8	0.625	300
Mother	QCIF	2-8	0.662	961
News	CIF	2-8	0.326	300
News	QCIF	2-8	0.413	300
Paris	CIF	2-8	0.471	1065
Salesman	QCIF	2-8	0.721	449
Silent	CIF	2-8	0.684	300
Silent	QCIF	2-8	0.723	300
Walk	CIF	2-8	0.360	376
Walk	QCIF	2-8	0.368	376

Table 5: With Hurst Parameters for H.264 using Wavelet estimation

Table 5 gives the Hurst Parameters for the video sequences encoded by H.264. The Video sequence Claire QCIF gives 0.630 as the Hurst value for the whole video sequence. For the aggregation level of a=100(frames) it gives the Hurst value 0.836. This indicates high level of Long-Range Dependence for a short period; however all the encoded video sequence exhibits Hurst Parameter which is typically around 0.6 or smaller. It is observed that H.264 encoded video sequence settles down and smoothens over long intervals. When Table 3 and 5 are compared, the Hurst parameter is around 0.8 for MPEG-4 encoded sequence and Hurst parameter around 0.6 for H.264 encoded video sequence. This result proves that H.264 is superior to MPEG-4 video coding standard.

5.4 CONCLUSION

This chapter analyses with the superiority of H.264 encoder, with the help of the same video libraries as used in MPEG-4 to compare the traces which were generated using H.264 encoder with video of single quality level as same as MPEG-4. The detailed statistical analysis has been carried out in H.264 video traces, as we known,

if the Hurst value is high, LRD will be large which in turn produce bursty traffic leading to data loss and transmission error. As we know that only H.264 encoder has Network Abstract Layer which can produce welcoming traffic stream to the wireless network, we made use of it very extensively. A new technique (optimal NAL allocation) is introduced to reduce the LRD effect for H.264 video stream. The stream is divided into optimal NAL units which are calculated by knowing the wireless network capacity. Table 3 and Table 5 statistical results of both encoded video traces, are compared. It shows that H.264 exhibits optimal Hurst value which is more or less equal to 0.6. This proves H.264 is more efficient and friendly in video stream transmission over wireless network.

CHAPTER VI

6. ANALYSIS OF VIDEO TRACES USING OPTIMAL PACKET FRAGMENTATION APPROACH

6.1 INTRODUCTION

The packet video transmission over wireless networks is likely to undergo packet loss due to temporary link outages or inserted fading bit errors. This kind of packet loss regularly occurs in bursts that may cause considerable deprivation to the transmitted video quality. Therefore, the Quality of Service (QoS) is reduced substantially. With the materialization of 3G-enabled broadband wireless IP networks, the predictable evolution of wireless mobile services includes data and multimedia transmission. The unpredictable and error-prone character of the wireless links is a major obstruction to dependable and efficient wireless video services. Third-generation (3G) mobile wireless systems supporting video services have to adjust with this lack of QoS assurance. Moreover, the wireless channel is extremely varying in time due to fading and obstruction effects. The packet video transmission over wireless networks is prone to undergo packet loss due to large packet size or very small packet size, both of which could cause bursty packet losses and thus cause considerable quality deprivation to the transmitted video. The large packet size produces higher Packet Loss Ratio (PLR) in Physical Layer (PHY) [14]. The small packet size produces packet loss in Medium Access Layer (MAC) due to congestion in the packets and thus increases the PHY overhead. As a result, there is a need for video coding and transmission schemes, which provide not only efficient compression performance but also comparatively strong transport performance in the presence of optimal packet size, which is encoded in the MAC level.

The purpose of this chapter is to study the transfer of video sequences over wireless ad hoc networks using the 802.11 technologies and check the characteristics of video frames after the transmission. The H.264 video encoder is configured to match the ad hoc network scenario as well as to accustom to varying channel conditions. The original video test sequences are packetized according to the H.264 Network Adaptation Layer (NAL) for transmission using the RTP/UDP/IP protocol stack. Wireless network conditions are simulated using the *ns-2* network simulator. The designs of the system involving different layers are analyzed to optimize the overall performance of the system. In this thesis, the researcher aims at globally optimizing the parameters involved in a real-time video transmission, ranging from video encoding and packetization to the 802.11 MAC interface parameters. Accurate and objective results, obtained through network simulations and video quality evaluation, are provided, showing the system performance under wireless network environment.

6.2 THE OPF 802.11 WIRELESS NETOWRK MAC LAYER APPROACH

In the video traffic source, typical video test sequences are packetized according to the IP Network Adaptation Layer (NAL) requirements of the H.264 standard and transmitted using the RTP/UDP/IP protocol stack. Using the TCP/IP protocol suite which is the object of this transmits data traffic. At the MAC and physical layers, all nodes provide the 802.11 functions. A MAC Protocol Data Unit (MPDU) is received correctly if the channel is in state good for the whole duration of the MPDU transmission; if not, it is received as an error . The MAC Layer protocol has a few limitations such as excessive packet collusion and data loss in the nodes while transmitting which leads to the poor performance [33]. This can be avoided by implementing routing algorithms in the transport layer, which leads to more

complicated and bad performance in the MAC layer. Next approach would be made by implementing new protocols with the MAC Layer, which is very difficult to achieve as we have to make changes in the hardware some in the firmware and a few in the drivers of the network adapter (NIC) [22]. For avoiding these overheads we use a new approach for better performance of the MAC Layer by introducing a method in which the packets are divided into optimal sizes according to the network capacity. A new approach called Optimal Packet Fragmentation (OPF) is used in which the exact packet size is segmented and transmitted over wireless network as shown in Fig 18

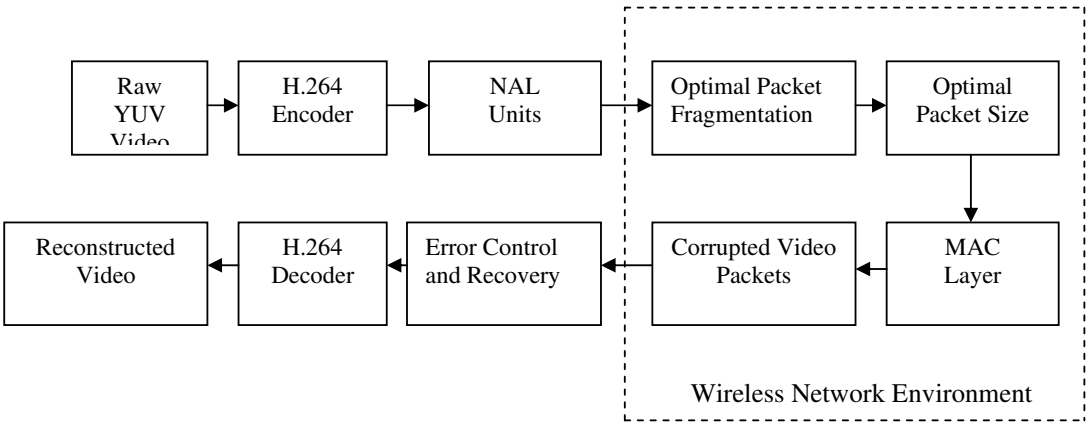


Fig 18: Optimal Packet Fragmentation Technique.

An H.264 source coder encodes the input video, which is considered with typical bandwidth limitations of wireless channels, in this work, which utilize relatively low resolution QCIF-format (176x144) video sequences. In order to transmit H.264 video over wireless IP networks, the H.264 encoded bit-stream has to be packetized. Based on the RTP-H.264 payload format condition, the H.264 encoded bit-stream is packetized and then transmitted as RTP/UDP/IP packets. Usually, the associated RTP/UDP/IP header is 40 bytes; however, in order to ease real-time applications over bandwidth-limited wireless networks, the researcher uses robust header

compression (RoHC) to compress the RTP/UDP/IP header resulting in decreased packet header overhead. Especially, the compressed RTP/UDP/IP header is 3 bytes. The RTP/UDP/IP packets can go through additional fragmentation and packetization procedures in the UMTS protocol stack before being transmitted over the network channel. In the video applications, it is possible that the RTP/UDP/IP packet may be larger than the capable network and physical-layer frame size resulting in fragmentation. When the packet size exceeds the maximum reliable transmission that a network is capable of, we use the approach Optimal Packet Segmentation, in which the video frames are sliced into smaller decodable data units (NAL units) as shown in Fig 19.

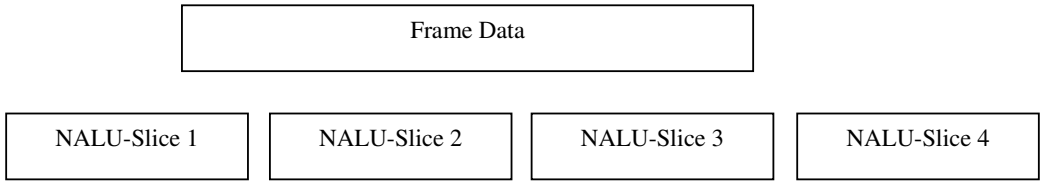


Fig 19: Frames are Sliced into Decodable data.

Slices are self-contained with all the information for decoding the slice available within the slice. Each Slice has encapsulated a NAL unit.

The Optimal Packet Fragmentation is carried out in the MAC Layer to achieve lowest Packet Loss Ratio (PLR). The OPF method approaches the overhead, which lies in the MAC Layer by accruing the network capacity. The OPF method checks the capacity of the network by calculating the speed and verifies whether the data has reached the destination without any data loss. The acknowledgement data gives the OPF approach the network capacity. According to the capacity the packets are fragmented. For instance, the wireless network is capable of transmitting 600 bytes.

The OPF segments the packets from the frame data near to 500 bytes. Then the packets are transmitted through wireless network to the destination node.

The error control and recovery system is also used for better QoS purpose. A forward error correction approach aims to develop the quality of the transmitted video in the occurrence of packet loss. Before transmitting the video clip, it is encoded into two versions with different quality levels. One is encoded with high quality and the other with low quality. During transmission, each high quality frame is piggybacked with a low quality frame version of the previously seen high quality frame. For instance, a high quality version of frame F1 will be piggybacked with a low quality version of frame F0. At the receiver, the decoder primarily decodes the high quality frame versions if they are present. If a high quality frame version cannot be decoded due to corruption or packet loss, the low quality version of the frame is used. To evaluate the scheme, video clips were transmitted over RTP using this scheme and the resulting clips were evaluated by a group of 42 viewers. The results showed that using redundant data improved the perceptual quality of the video. For example at 1% and 20% data loss, users reported an improvement of 20% and 65% respectively in video quality.

The wireless simulation environment used is as follows: Video sequences are encoded using the JM 9 codec of the newly developed H.264 video coding standard. A typical QCIF test video sequences, Foreman is used. The frame rate is 30 fps for the QCIF Foreman sequence. The Foreman sequence at 30 fps is regarded as a low-motion level. The sequence was coded at constant bit rates specified by using the associated rate control scheme. The first frame of the sequence is intra-coded and the rest of the frames are inter-coded as *P* frames with or without slice-based intra-

updating. In our packetization scheme, each slice is packetized into one RTP packet, thus every QCIF frame is packetized into RTP packets.

6.3 OPF IN NS 2 WIRELESS SIMULATION ENVIRONMENT

The *NS-2* network simulator [37] is used for protocol performance estimation and to provide a simulated network environment for coded video transmission. Wide range of simulation in transport protocols, wireless networks, routing protocols and queuing algorithms is provided by *NS-2*. One of the main features that *NS-2* offers is the wireless network implementation. It is based on the CMU wireless extensions, which have now been integrated into the main *NS-2* distribution. The implementation supports all layers of the Internet. The Physical layer simulates propagation models (such as TwoRayGround for longer distances and Friss-space attenuation for shorter distances), radio interfaces and the antennas. At the Link layer, *NS-2* supports the 802.11 MAC Layer protocols and ARP protocols. For the network layer, it supports routing protocols, such as, Ad hoc On Demand Distance Vector (AODV) and Dynamic Source Routing (DSR). The MAC used is the generic 802.11, propagation model is TwoRayGround, interface queues are drop-tail and the routing protocol used is Destination-Sequenced Distance Vector (DSDV). The simulator is used to conduct a trace-driven simulation. Before an experiment can be conducted in a trace-driven simulation environment, the trace must be produced. In this study, the video evaluation tool produces the trace. The trace is then fed into the simulator for processing. After processing the video trace, *NS-2* produces a packet trace of received packets that can be used to analyze the video.

The Optimal Packet Fragmentation technique is tested in NS2 wireless network simulated environment. The process includes mainly three steps.

1) Video Trace Files Generation from H.264 Encoder

2) Fragmentation process in NS2 Simulation.

3) Video Streaming Evaluation.

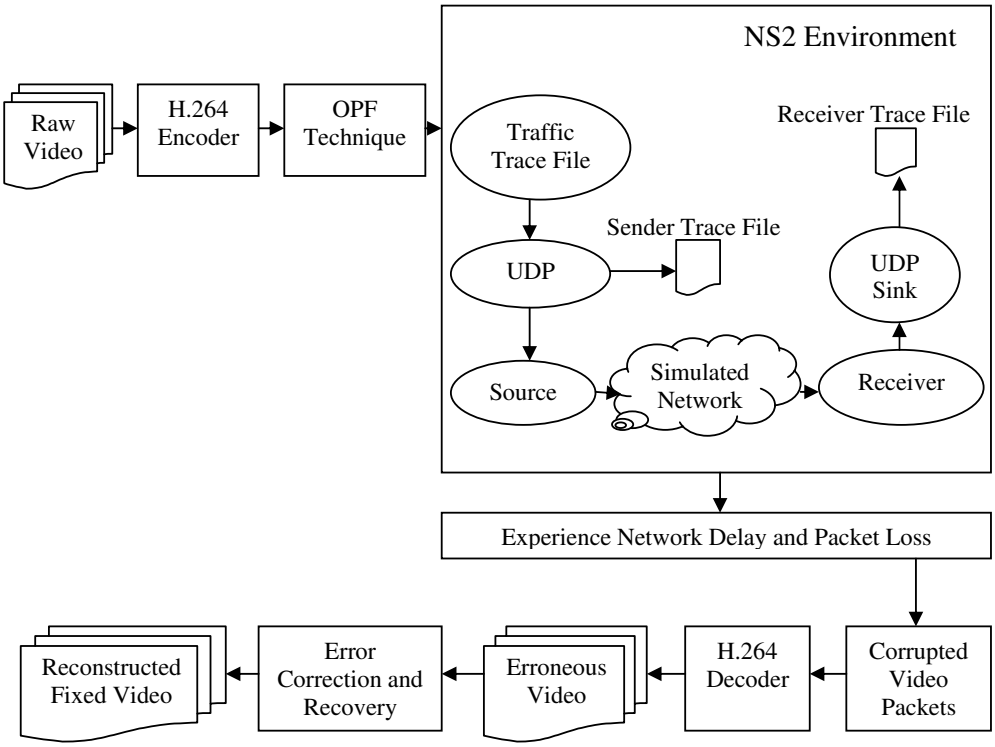


Fig 20: OPF Technique in Wireless Environment.

Figure 20, shows the working process of OPF in wireless environment. The target of the first step is to generate video trace files. In order to make the video transmission adaptively, in this step we generate lots of trace files in N different scales. The second step is network simulation. It plays the role of black box in our system. We have used network structure of RTP/UDP/IP. We generate sender trace file and receiver trace file to record the packets details during the simulation process in this step. Finally, by using the trace files generated during simulation and the

encoded videos in step one, we can regenerate received video. After decoding the constructed video, the transmission can be evaluated by calculating Peak Signal Noise Ratio (PSNR).

6.3.1 Video Trace Files Generation Using OPF

In this part we generate the video trace files which describes the details of the compressed video and can be used as the input for the next step. This part includes H.264 Encoder and Video Trace Files Generator. The input of this part is raw video sequence generated by a video source. They can be either in the YUV QCIF which has 176*144 pixels or in the YUV CIF which has 352*288 pixels. These two formats are commonly used in network related studies. The output of this part is video trace files.

Our target is to have a real-time video rate controller in the network simulator, but without having to do the media encoding itself. That's why the media encoding must before the network simulation, In H.264, the valid quantizer scale values are in the range from 1 to 31, with 1 producing the best quality and the highest bit rate while 31 producing the worst quality and the lowest bit rate.

The video trace file is in the format of frame size. However, this is not enough as an input of NS2 because we have to consider the timestamp for each frame/packet in the NS2. For this, the Video Traffic Trace Generator is used to produce a trace file containing the timestamp and size.

6.3.2 OPF in NS2 Simulation Environment

This part plays the role as a black box in our system. It is an object-oriented, discrete-event-driven network simulator As RTP/UDP/IP is becoming the standard for multimedia Streaming, As the RTP is becoming more and more popular for multimedia streaming because of its stable performance, the target of the extended RTP/RTCP includes:

1) Add the feedback (Sender Report and Receiver Report) into RTP/RTCP parts to transmit the QoS parameters timely.

2) Generate the sender trace file and the receiver trace file for recording the transmission details. Sender trace file records the sending time, packet sequence number, protocol name, packet size (in bytes), quantization scale, frame type. The video generator we choose support the unit of GoP, so you find in the figure that the quantization of one GoP is fixed. The receiver trace file is in the similar format but if it finds the sequence number is not consecutive it will symbol the lost packets as lost. By using these two files, metrics which are related to network topology such as packet loss, jitter and frame loss can be calculated directly.

6.3.3 Video Streaming Evaluation Using OPF Technique

Tools in this part include video generator, H.264 decoder, Error Concealment Generator and PSNR&MOS calculation. With the generated NS2 sender trace file and receiver trace file and RTP-traffic trace file and the compressed video files from step one, the following metrics for evaluating can be got: Packet/Frame loss rate and Jitter Constructed received video PSNR (between original video and constructed received video) MOS (Mean Opinion Score).

The Packet/Frame loss rate and Jitter can be got within the work in NS2. Here we should point out that in multimedia transmission frame loss rate performs important role in evaluation compared with packet loss. The main part of this step is the reconstruction of received video with the help of generated trace files. The process of reconstruction can be regarded as a process of copying the original compressed video file packet by packet, omitting packets that are lost during network simulation. Then the Error Concealment Generator is used by simply inserts missing frames by copying the former frame so that sent and received video consists of equal number of frames.

Up to now the full process of video transmission is finished. We can use VLC Player to open the reconstructed encoded video. Some blur can be found during playback because of the congestion occurred in network simulator. Using the reconstructed video and the original video, we can evaluate the whole process between them. Nowadays the most widespread method is the calculation of peak signal to noise ratio (PSNR) image by image. It is a derivative of the well-known signal to noise ratio (SNR), which compares the signal energy to the error energy. The PSNR compares the maximum possible signal energy to the noise energy, which has shown to result in a higher correlation with the subjective quality perception than the conventional SNR. We use the PSNR to get the results, between the luminance component Y of source image S and destination image D.

6.3.4 OPF Coding Used in NS2 Environment

In this section researcher has included few main parts of the codes used in coding the NS2 simulator. NS2 simulator is coded using Tool Command Language (TCL) script.

Few header files which are used in NS2 simulator are modified according to our requirements. Modified the file packet.h in the common folder. The variables frametype_ and sendtime_ field in the hdr_cmn header are modified. The frametype_ field is to indicate which frame type the packet belongs to. I frame type is defined to 1, P is defined to 2, and B is defined to 3. The sendtime_ field is to record the packet sending time. It can be used to measure end-to-end delay.

```
//added the following three lines
intframetype_;              // frame type for H.264 video transmission
doublesendtime_;        // send time
unsigned long intframe_pkt_id_;
```

Modifed the fileagent.h file in the common folder

```
class Agent : public Connector {
public:
Agent(packet_tpktType);
virtual ~Agent();
voidrecv(Packet*, Handler*);
inlinepacket_tget_pkttype() { return type_; }
// add the following two lines
inline void set_frametype(int type) { frametype_ = type; }
inline void set_prio(intprio) { prio_ = prio; }
```

Modifed the file agent.cc file in the common folder

Agent::Agent(packet_tpkttype) :

size_(0), type_(pkttype), frametype_(0)

channel_(0), traceName_(NULL),

oldValueList_(NULL), app_(0), et_(0)

Modifed the ns-allinone-2.28/ns-2.28/tcl/lib/ns-default.tcl

Added the following two lines

Agent/myUDP set packetSize_ 1000

Tracefile set debug_ 0

Recompiled NS2

./configure ; make clean ; make

We run this on cygwin environment.

Raw YUV video sequence with 30 frames per second, a GOP length of 30 frames with no B-frames is encoded using H.264 encoder.

$./xvid_encraw -i akiyo_cif.yuv -w 352 -h 288 -framerate 30 -max_key_interval 30 -o a01.m4v

The command lines create ISO MP4 files containing the video samples (frames) and a hint track which describes how to packetize the frames for the transport with RTP.

$./MP4Box -hint -mtu 1024 -fps 30 -add a01.m4v a01.mp4

The OPF Technique is able to send a hinted mp4-file per RTP/UDP to a specified destination host. The output of h264 trace will be needed later, so it should be redirected to a file.

$./mp4trace -f -s 192.168.0.2 12346 a01.mp4 > st_a01

First, we used a simple topology to test the H.264 video delivery over a best-effort service network. The simulation environment is shown in Fig 20. S1 will use the data in the trace file to transmit packets to D1. The script file is myh264.tcl

S1 → R1 → R2 → D1

After simulation, ns2 will create two files, sd_h264 and rd_h264. The file sd_h264 is to record the sending time of each packet while the file rd_h264 is used to record the received time of each packet.

The next step is the reconstruction of the transmitted video as it is seen by the receiver. For this, the video and trace files are processed by etmp4 (Evaluated Traces of MP4-file transmission).

Decoded the received video to YUV format. Used ffmpeg to decode the compressed file. It won't cause any error in most cases. If we use other codec to decode, it may cause errors in most cases.

$./etmp4 sd_a01 rd_a01 st_a01 a01.mp4 a01e

This generates a (possibly corrupted) video file, where all frames that got lost or were corrupted are deleted from the original video track.

$./ffmpeg -i a01e.mp4 a01e.yuv

Finally the PSNR is computed using the below command.

$./psnr.exe 352 288 420 akiyo_cif.yuv a01e.yuv

6.4 EXPERIMENTAL RESULTS

Optimal Packet Fragmentation (OPF) is used in which the exact or optimal packet size is segmented according to the capability of the network and transmitted over wireless network. If we use large packet sizes in a network, which has very less capability for transmitting the packets, it has few advantages and disadvantages. The advantages such as it reduce the overheads of RTP/UDP/IP and also reduce the overheads of the Physical Layer (PHY). It reduces the contentions and collusions the overheads in MAC Layer. But the main disadvantages is that it has higher Packet Loss Ratio (PLR) in the Physical Layer which is denoted by,

$PLR=1-(1-BER)^{pkt\_len} \sim BER * pkt\_len$

If smaller packets are transmitted in the large capable wireless network then the Packet Loss Ratio is lesser in Physical Layer that adds as an advantage. There will be higher packet loss in the MAC due to congestion because there will be huge amount of small packets travelling to the same destination. As there will be a lot of small packets transmitted in the wireless network there will be too much overhead in the Physical Layer. Because of larger and smaller packet sizes has too much trouble in the wireless network; the optimal sizes are fragmented from the frame data. So, there exists an optimum value for the packet length.

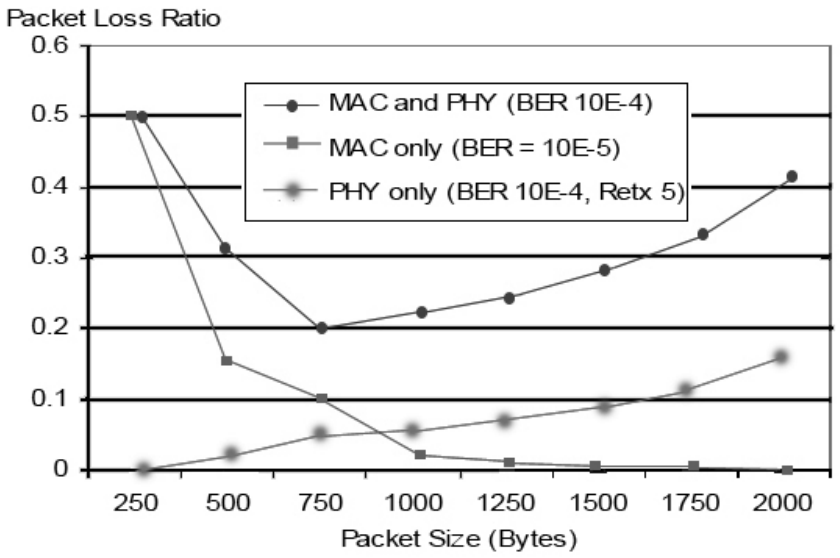


Fig 21: Packet Size and Packet Loss Ratio

The new Optimal Packet Fragmentation carries out the packet size adjustment technique in which the optimal packet that is optimum in the network is transmitted. This is explained by transmitting 2200 bytes packet that is larger for the network capability of 800 bytes that is transmitted. The large packet size output video frame with a large Packet Loss Ratio is given in the Fig 21. The smaller packet sizes of 400 bytes are transmitted in the 800 bytes capacity wireless network. The packet losses due to congestion for the smaller size packets are given in Fig 23. Finally, the Optimal Packet of size 800 byes is transmitted in the wireless network. The output frame of optimal value frame length is given by comparing the maximum possible signal energy to the noise energy (PSNR) shown in Fig 22.

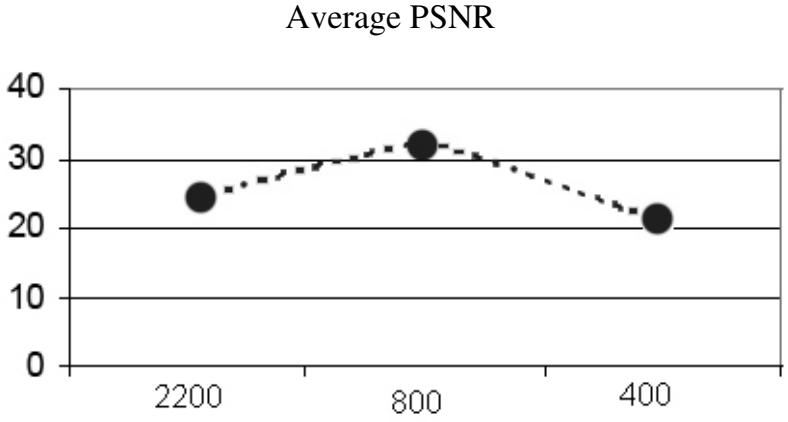


Fig 22: Packet size in Wireless Network

Having the PSNR value, another metric MOS (Mean Option Score) can be calculated. This method has the advantage of showing clearly the distortion caused by the network at a glance. The determination of MOS is shown in the table 6 below:

PSNR [dB]	MOS
> 37	5 (Excellent)
31-37	4 (Good)
25-31	3 (Fair)
20-25	2 (Poor)
< 20	1 (Bad)

Table 6: PSNR to MOS Conversion

Fig 23: 2200 Bytes Large Packet Size

Fig 24: 800 Bytes Optimal Packet Size

Fig 25: 400 Bytes Smaller Packet Size.

The frame packets are combined to form frames. Each frame (I/P/B) size is stored in the trace file after transmitting through the wireless network simulator environment. After storing all the coded video sequence frame sizes in the trace file, the file is used for checking the LRD in the wireless network. The trace file is used to acquire the Hurst parameter, which denotes Long-Range Dependence. This method shows the phenomena of LRD character in the video frames. As discussed before in chapter 2, the LRD is calculated for the video sequence, which uses the OPF (Optional Packet Fragmentation) approach, and the LRD is calculated for the video sequence without using the OPF method.

The Hurst Parameter is calculated by using wavelet estimation for the video sequence trace file, which is produced from the network simulation environment without using OPF approach. The Hurst Value for this video trace is given in Table 7. For this study, 6 video sequences were used. Three QCIF and 3 CIF video sequences were also used.

Video Clip	Clip Format	Range	H.264/AVC	Number of Frames
Akiyo	CIF	2-8	0.847	300
Akiyo	QCIF	2-8	0.892	300
Bridgefar	QCIF	2-8	0.745	2101
Carphone	QCIF	2-8	0.775	382
Football	CIF	2-8	0.934	90
Foreman	CIF	2-8	0.634	300

Table 7: Hurst Parameter without using OPF

After using the OPF approach the trace files are generated for each H.264 video coded sequence that are transmitted through wireless network environment. The Hurst values for this video sequence are given in Table 8.

Video Clip	Clip Format	Range	H.264/AVC	Number of Frames
Akiyo	CIF	2-8	0.728	300
Akiyo	QCIF	2-8	0.782	300
Bridgefar	QCIF	2-8	0.704	2101
Carphone	QCIF	2-8	0.720	382
Football	CIF	2-8	0.898	90
Foreman	CIF	2-8	0.567	300

Table 8: Hurst Parameter using OPF

The value in the above tables shows the efficiency of Optimal Packet Fragmentation. The efficiency of OPF approach increases from 25% to 30% as shown in Fig 22. This result proves that with the OPF approach the H.264 video coded sequence performs extremely reliably and more efficiently in robust wireless environment.

6.5 CONCLUSION

This chapter deals with the core content of this research work. If there is packet loss while transferring video streams over wireless network the Quality of Service will be reduced substantially and to avoid the researchers has introduced a new technique

called Optimal Packet Fragmentation (OPT). This OPF technique segregates the H.264 video stream into optimal packets. Network behaviour is simulated using NS2 simulator. The network capacity is obtained and packets are prepared accordingly. The simulation is carried out by getting the trace files as input and video is regenerated according to the output trace from the simulator. Some blur can be found during playback because of the congestion that occurred in network simulator. The PSNR was used to get the results, of the luminance component Y of source image S and destination image D. The study carried out using PSNR clearly proves that OPF technique is really worth a shot. The result shown in Table 7 depicts the efficiency of OPF technique in H.264 encoded video traces.

Furthermore contribution of this chapter is twofold. First, we have presented the integration of Optimal Packet Fragmentation (OPF) and NS2 to provide a novel generalized and comprehensive tool-set for evaluating the video quality performance of network designs in a simulated environment. It enables OPF to link seamlessly with NS2 in such a way that researcher have greater freedom to analyze their proposed network designs for video transmission without being obliged to consider an appropriate tool-set for video quality evaluation. Simulations of real video streams are enabled over a large set of network scenarios, including relatively large topologies, node mobility, different kinds of concurrent traffic, or any other functionality available by the network simulator. Secondly, in an analysis enabled by the new tool-set, we have shown that the fraction of decodable frames can adequately reflect the behaviour of the PSNR QoS video assessment metric with reasonable accuracy and while being less time consuming by at least one order of magnitude. Therefore, when researchers want to encode their own test video

sequences or adopt well-known ones in order to evaluate the delivered video quality in a simulated network environment, our proposed QoS assessment framework would be a good choice.

CHAPTER VII

CONCLUSION

In this thesis the researcher presents from his studies, details on a publicly available library of frame size traces of MPEG–4 and H.264 encoded videos (http://www.inf.brad.ac.uk/~jjayasee/). He conducted a detailed statistical analysis of the generated traces. He has found that the traces are typically variable in their frame sizes, especially the traces of H.264 encoding. Also, many of the traces show clear indications of long-range dependence properties in MPEG-4 encoding. He also observed from the study that H.264 produces Hurst value less than MPEG-4. So, H.264 has negligible amount of LRD. In this thesis, Optimal Packet Fragmentation is applied to the emerging video coding standard H.264. Video coded sequence of H.264 proves the dominance to MPEG-4 encoder. Further the thesis shows the power of Optimal Packet Size Fragmentation approach when the H.264 encoded videos are transmitted through wireless network in the MAC Layer.

As a part of the future work, we are expanding our video trace study by producing and analyzing MPEG-4 and H.264 encoding of more videos. Moreover, we are encoding videos using the H.264 encoder, which incorporates more advanced motion prediction and enhanced PB frames. As video transmissions over wireless network gain more and more prominence, simple, robust and network-adaptive optimal packet size becomes important. Further studies are being carried out by using Optimal Packet Size Fragmentation approach in mobile networks.

REFERENCES

[1] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch, “Wavelets for the analysis, estimation, and synthesis of scaling data in Self-Similar Network Traffic and Performance Evaluation”, pp. 39–88, edited by K. Park and W. Willinger. New York, NY: Wiley, 2000.

[2] P. Abry and D. Veitch, “Wavelet Analysis of Long-Range Dependence Traffic,” IEEE Trans. Information Theory, vol. 44, no. 1, pp. 2–15, 1998.

[3] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, “Adaptive unicast video streaming with rateless codes and feedback,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 2, pp. 275–285, 2010.

[4] P. Ansel, Q. Ni, and T. Turletti, “An efficient scheduling scheme for IEEE 802.11e,” in Proceedings of IEEE Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt ’04), Cambridge, UK, March 2004.

[5] Anthony Vetro, Huifang Sun, "An Overview of MPEG-4 Object-Based Encoding Algorithms," itcc, p. 0366, International Conference on Information Technology Coding and Computing (ITCC '06), 2006.

[6] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. “Long–range dependence in variable–bit–rate video traffic. “IEEE Transactions on Communications, 43(2/3/4):1566–579, February/March/April 1995.

[7] Beshan Kulapala, Patrick Seeling, Martin Reisslein, “Characteristics of Traffic and Quality Traces of Wavelet Encoded Video”, Electrical Engineering, Arizona State University, Goldwater Center, December 8, 2003.

[8] Boris Tsybakov and Nicolas D. Georganas, "On Self-Similar Traffic in ATM Queues: Definitions , Overflow Probability Bound, and Cell Delay Distribution , " IEEE / ACM Trans. Networking , Vol. 5 , No. 3 , pp. 397-409, 1997

[9] J. Cao, “On the Nonstationarity of Internet Traffic,” Sigmetrics/Performance, ACM Press, pp. 102–112, 2001.

[10] M. Dai and D. Loguinov, “Analysis and Modeling of MPEG-4 and H.264 Multi-Layer Video Traffic,” IEEE INFOCOM, Mar, 2005.

[11] K. Dovstam, “Video Coding in H.26L Video Coding in H.26L,” Ph.D. Dissertation, 2000.

[12] A. Erramilli, O. Narayan and W. Willinger, “Experimental Queuing Analysis with Long-Range Dependent Packet Traffic,” IEEE/ACM Trans. Networking, vol. 4, no. 2, pp. 209–223, 2002.

[13] Y. P. Fallah, P. Nasiopoulos, and H. Alnuweiri, “Scheduled and contention access transmission of partitioned H.264 video over WLANs,” in Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM ’07), pp. 2134–2139,Washington, DC, USA, November 2007.

[14] Y. P. Fallah, D. Koskinen, A. Shahabi, F. Karim, and P.Nasiopoulos, “A cross layer optimization mechanism to improve H.264 video transmission over WLANs,” in Proceedings of the 4th IEEE Consumer Communications and Networking Conference (CCNC ’07), pp. 875–879, Las Vegas, Nev, USA, January 2007.

[15] A Feldmann., Gilbert, A., Huang, P. & Willinger, W., "Dynamics of IP traffic: A study of the role of variability and the impact of control", Computer Communication Review 29(4), 301 - 313. Proceedings of the ACM/SIGCOMM'99, Cambridge, MA 1999.

[16] F. Fitzek, P. Seeling, and M. Reisslein, “H.26L Pre-Standard Evaluation,” acticom–mobile networks, Germany, Tech. Rep. acticom-02-002, November 2002.

[17] F. Fitzek., “Video and Audio Trace Files of Pre-Encoded Video Content for Network Performance Measurements,“ Proc.IEEE Consumer Commun. and Net. Conf. (CCNC), Las Vegas, NV, pp. 245–50, Jan. 2004.

[18] F. H. P. Fitzek and M. Reisslein, “MPEG-4 and H.263video traces for network performance evaluation”. Technical Report TKN-00-06, Technical University of Berlin, Telecommunication Networks Group, 2002.

[19] M. Freytes, C.E. Rodriguez, and C.A. Marques, “Real-Time H.263+ Video Transmission on 802.11 Wireless LANs,” in Proc. Int. Conf. on Information Technology: Coding and Computing, Las Vegas, NV, April 2001.

[20] M. Ghanbari, Standard Codecs: Image Compression to Advanced Video Coding, IEE, 2003.

[21] M. M. Ghandi, B. Barmada, E. V. Jones, and M. Ghanbari, “H.264 layered coded video over wireless networks: channel coding and modulation constraints,” EURASIP Journal on Applied Signal Processing, vol. 2006, Article ID 85870, 8 pages, 2006.

[22] M. M. Ghandi, B. Barmada, E. V. Jones, and M. Ghanbari, “Wireless video transmission using feedback-controlled adaptive H.264 source and channel coding,” IET Communications, vol. 3, no. 2, pp. 172–184, 2009.

[23] V. K. Goyal, “Multiple Description Coding: Compression Meets the Network”, IEEE Signal Processing Mag., vol. 18, no. 5, pp. 74–93, Sept. 2001.

[24] A. Grilo, M. Macedo, and M. Nunes, “A scheduling algorithm for QoS support in IEEE802.11E networks,” IEEE Wireless Communications, vol. 10, no. 3, pp. 36–43, 2003.

[25] F. Halsall, “Multimedia Communications: Applications, Networks, Protocols, and Standards”, Addison-Wesley, 2001.

[26] Irini Reljin,Andreja Samcovic and Branimir Reljin "H.264/AVC Video Compressed Traces: Multifractal and Fractal Analysis" EURASIP Journal on Applied Signal Processing Volume 2006, Article ID 75217, Pages 1–13, 2006

[27] John Jayaseelan “Video traces of H.264 and MPEG-4 for network performance evaluation”, University Of Bradford, [Online]. Available: http://www.inf.brad.ac.uk/~jjayasee/.

[28] Joint Video Team, “Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC),” Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, March 2003.

[29] Jun Xin, Anthony Vetro, Huifang Sun, “Efficient Macroblock Coding-Mode Decision for H.264/AVC Video Coding”, Mitsubishi Electric research Lab, Dec 2004.

[30] R. Koenen, “Overview of the MPEG-4 Standard,” ISO/IEC 14496, May/June 2000.

[31] H. Koumaras, C. Skianis, G. Gardikis, A. Kourtis, “Analysis of H.264 video encoded traffic”, Institute of Informatics and Telecommunications, Athens Greece, April 2005.

[32] A. Ksentini, M. Naimi, and A. Gueroui, “Toward an improvement of H.264 video transmission over IEEE 802.11e through a cross-layer architecture” IEEE Communications Magazine, vol. 44, no. 1, pp. 107–114, 2006.

[33] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, “Error resiliency schemes in H.264/AVC standard,” Journal of Visual Communication and Image Representation, vol. 17, no. 2, pp. 425–450, 2006.

[34] W.E. Leland, M.S. Taqqu, D.V. Wilson and Walter Willinger, " On the Self-Similar Nature of Ethernet Traffic (extended version), " IEEE / ACM Trans. Networking , vol. 2 , no. 1, Revised, 2005.

[35] W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard”, IEEE Trans. Circuits and Systems for Video Tech., vol. 11, no. 3, pp. 301–17, Mar. 2001.

[36] Lucantoni D.M., Neuts M.F. and Reibman A.R, ‘‘Methods for performance evaluation of VBR video traffic models’’, IEEE/ACM Transactions on Networking, Vol.2, No.2, pp176–180, 1994.

[37] “The Network Simulator — NS–2.” [Online]. Available: www.isi.edu/nsnam/ns/index.html.

[38] Nikola Cackov, Zelimir Luˇci´c, Momˇcilo Bogdanov, Ljiljana Trajkovi´c, “Wavelet-Based Estimation of Long Range Dependence in MPEG Video Traces”.

[39] K. Park and W. Willinger, (2000) “Self-similar network traffic: An overview in Self-similar Network Traffic and Performance Evaluation pp. 1–38, edited by K. Park and W. Willinger. New York, NY: Wiley, 2000.

[40] K. Park, G. Kim, and M.E. Crovella, “On the Relationship Between File Sizes Transport Protocols, and Self-Similar Network Traffic,” Int’l Conf. Network Protocols, IEEE CS Press pp. 171–180, 2002.

[41] V. Paxson and S. Floyd, “Wide Area Traffic: The Failure of Poisson Modeling,” IEEE/ACM Trans. Networking, vol. 3, no. 3, 1995, pp. 226–244, 1995.

[42] A. Puri and T. Chen, “Multimedia Systems, Standards, and Networks”, Marcel Dekker, 2000.

[43] Richard G.Clegg, “A Practical Guide to Measuring the Hurst Parameter”, June 28 2005.

[44] M. Reisslein, F. Fitzek, “Traffic and Quality Characterization of Scalable Encoded Video: A Large–Scale Trace–Based Study,” Arizona State University, Telecommunications Research Center, Tech. Rep., 2002.

[45] M. Reisslein, “Traffic and Quality Characterization of Scalable Encoded Video: A Large-Scale Trace-based Study, Part 1: Overview and Definitions,” Arizona State Univ., Telecommunications Research Center, Tech. Rep., Dec. 2002.

[46] M. Reisslein, “Traffic and Quality Characterization of Scalable Encoded Video: A Large-Scale Trace-Based Study, Part 2: Statistical Analysis of Single Layer Encoded Video,” Arizona State University, Dept. of Electrical Engineering, Tech. Rep., Dec. 2002.

[47] M. Reisslein, “Traffic and Quality Characterization of Scalable Encoded Video: A Large-Scale Trace-Based Study, Part 3: Statistical Analysis of Temporal Scalable Encoded Video,” Arizona State University, Dept. of Electrical Engineering, Tech. Rep., Dec. 2002.

[48] M. Reisslein, “Traffic and Quality Characterization of Scalable Encoded Video: A Large-Scale Trace-Based Study, Part 4: Statistical Analysis of Spatial Scalable Encoded Video,” Arizona State University, Dept. of Electrical Engineering, Tech. Rep., Aug. 2003.

[49] H. L. Resnikoff, Raymond O'Neil Wells, "Wavelet analysis: the scalable structure of information", Rice University, Houston, 1998.

[50] O. Rose, “Statistical properties of MPEG video traffic and their Impact on traffic modeling in ATM systems,” University of Wurzburg, Institute of Computing Science, Report No. 101, Feb. 1995.

[51] V. Sgardoni, M. Sarafianou, P. Ferr´e, A. Nix, and D. Bull, “Robust video broadcasting over 802.11a/g in time-correlated fading channels,” IEEE Transactions on Consumer Electronics, vol. 55, no. 1, pp. 69–76, 2009.

[52] T. Sikora, “MPEG Digital Video Coding Standards,” Digital Electronics Consumer Handbook, McGraw Hill, 1997.

[53] L. D. Soares and F. Pereira, “MPEG–4 : A Flexible Coding Standard for the Emerging Mobile Multimedia Applications,” Tech. rep., MOMUSYS Project, 2003.

[54] T. Stockhammer, M.Westerlund, D. Singer, S.Wenger, and M.M. Hannuksela, “RTP Payload Format for H.264 Video,” RFC 3984, February 2005.

[55] T. Stockhammer and M. Bystrom, “H.264/AVC data partitioning for mobile video communication,” in Proceedings of the International Conference on Image Processing (ICIP ’04), vol. 1, pp. 545–548, Singapore, October 2009.

[56] Stoev, S., Taqqu, M., Park, C., Michailidis, G. & Marron, J. S., “LASS: a tool for the local analysis of self-similarity”, Computational Statistics and Data Analysis ,2004.

[57] M. S. Taqqu, Statistical methods for long-range dependence [Online].Available: http://math.bu.edu/people/murad/methods/index.html.

[58] D. S. Taubman and M. W. Marcellin, “Image Compression Fundamentals, Standards and Practice”, Kluwer Academic Publishers, 2001.

[59] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra,“Overview of the H.264 / AVC Video Coding Standard”, IEEE Transactions on Circuits and System For Video Technology, July 2003.

[60] D. Veitch and P. Abry, (2001) “A statistical test for the time constancy of scaling exponents,” IEEE Trans. Signal Processing, vol. 49, no. 10, pp. 2325– 2334, Oct. 2001.

[61] D. Veitch, MATLAB code for estimation of scaling exponents [Online]. Available: http://www.cubinlab.ee.mu.oz.au/~darryl/secondorder code.html.

[62] Vern Paxon and Sally Floyd, " Wide Area Traffic : The failure of Poisson Modeling , " IEEE / ACM Trans. Networking , Vol. 3 , pp. 226-244 , 1995.

[63] W. Willinger, “Self-Similarity through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level,” IEEE/ACM Trans. Networking, vol. 5, no. 1, pp. 71–86, 2001.

[64] J. Xin, A. Vetro and H. Sun, “Converting DCT coefficients to H.264/AVC transform coefficients,” (submitted to) Pacific-rim Conference on Multimedia (PCM), 2004.

[65] F. Xue and Lj. Trajkovi´c, “Performance analysis of a wavelet-based Hurst parameter estimator for self-similar network traffic,” in Proc.SPECTS, Vancouver, BC, pp. 294–298, May 2000.

[66] Y. Zhang, “On the Constancy of Internet Path Properties,” Proc. ACM Sigcomm Internet Measurement Workshop, ACM Press, pp. 197–212, 2001.

[67] Z.L. Zhang, “Small-Time Scaling Behaviors of Internet Backbone Traffic: An Empirical Study,” Proc. IEEE Infocom, IEEE CS Press, pp. 1826–1836, 2003.

[68] Zip file of MPEG-4 Reference Software for Encoding and Decoding of videos [Online]. Available: http://perso.telecom-paristech.fr/~dufourd/mpeg-4/systems-streams.zip.