Optimal GOP strategies for IPTV Quality of Service

William Lombard, Hugh Melvin

Discipline of Information Technology, National University of Ireland, Galway

email: w.lombard1@nuigalway.ie, hugh.melvin@nuigalway.ie

Abstract

Real-time video communication in the form of IPTV (Internet Protocol TV) and Internet TV over a WAN (Wide Area Network) is often subject to disruptive elements such as packet loss, excessive delay and delay variations called jitter. This research topic examines the quality of service impact that packet-loss has on video streams that use modern compression techniques. Specifically, the paper analyses the optimal 'Group of Picture' size and grouping strategies of the three distinct frame types used by MPEG-4 encoded video files, in the face of simulated packet loss. The paper also seeks to further define the relationship between optimal Group Of Picture size and the motion and texture content of an encoded video file. Results are based on industry-standard objective quality metrics such as PSNR and SSIM. Technologies used include FFmpeg, Evalvid and the network simulator NS-2.

Some secondary subjective tests are conducted using a streaming video session between two machines on a wired local area network running VLC Player with NetEm induced network impairments.

Keywords

IPTV, GOP, Objective Quality

1 Introduction

IPTV is a system for the delivery of video content over private IP network, characterised by continuous streams of professionally produced TV content and viewed on a TV via a set-top box (STB).

We are now entering an era where broadband access is quickly becoming available to masses. The recent improvements in communications architecture are merging the once totally distinct formats of TV and the Internet. This is also driven by current research into the optimisation of video compression codecs such as MPEG-2, MPEG-4 and more recently, H.264. These video compression technologies discard the redundant temporal and spatial data from the data stream in a motion picture and later reconstruct it so that the lost temporal and spatial data is not immediately apparent to observer.

Video compression research is not just confined to IPTV. Applications also exist in the field of ultra low bit-rate video recording and transmission such as those used by 3G-4G mobile phones and also in video conferencing, which can exploit the webcams and microphones built into even the cheapest of modern netbook and laptop computers.

Ongoing video compression research and development will further drive the sharing and delivery of this vitally important media type.

Compressed videos transmitted over networks such as the public Internet and private IPTV infrastructures are often subject to impairments such as packet loss, corruption and delay variations known as jitter. Packet loss and jitter can be caused for example by factors such as network congestion (over-utilisation and/or over-subscription), higher priority traffic blocking lower priority traffic, or network equipment problems such as failing routers or switches. These can all have the effect of causing video frames to be lost or disputed when receiver by the customer, which in turn decreases Quality of Experience (QoE). This paper will examine if there is an optimal GOP size and frame type ratio for video sequences of differing textural complexity and motion content when subjected to simulated packet loss and delay variations.

The remainder of this paper is structured as follows. Section 2 outlines the origins and functions of modern video compression codecs. Section 3 briefly examines how video quality can be subjectively and objectively measured, whilst Section 4 studies how video sequences can be decomposed into fragments known as 'Groups of Pictures'. Section 5 looks at how the testbed was established and what parameters were examined. Section 6 discusses results found and Section 7 concludes the paper.

2 Modern Codecs

Arguably, Internet Protocol TV would not exist today without the development of audio, image and video compression technologies. Compression technology has also been behind the development and widespread consumer demand for devices such as DVD players, digital photography and the ubiquitous MP3 player. Quite simply, the bandwidth available on the current communications architecture and devices could not simultaneously support demand for high quality uncompressed video as well as quick, low-cost delivery. To put this statement into context, an uncompressed standard definition TV broadcast would require more than 200 Mbit/s of transmission capacity , so a 2-hour movie would require almost 200 gigabytes of storage; something that would be simply impossible to stream in real time and wouldn't be feasible for even the most patient of customers to download and store on their devices for later viewing.

As such, compression is what allows IPTV services providers to broadcast several high quality audio and video streams simultaneously over a broadband IP network. It achieves this by exploiting the natural limitations in the human audio and visual perception systems, along with inherently redundant data in a given video or audio stream. The compression used in audio and transmission tends to be 'lossy', whereby less important information is stripped away at encoding and can never be recovered later. This contrasts with 'lossless' compression, the type used where absolutely no data is lost between compression and decompression. This is critically important in certain applications, such as compressing a computer program to make it quicker to download, but if even one bit of data is lost or corrupted the program may be rendered unusable or unstable when running. A further factor in the drive towards IPTV is the schedule in most developed nations for ceasing the transmission of analogue terrestrial television, after which all television receivers will need compression technology to decode and display TV images. There is absolutely no doubt that video compression is here to stay [1].

At the heart of the vast majority of video coding systems and standards is transform coding. Spatial image data (image samples or motion-estimated residual samples) are transformed into a different representation known as the transform domain. There are several good reason for transforming the data this way. Spatial image data is inherently difficult to compress: neighbouring samples are highly correlated (interrelated) and the energy tends to be evenly spread across an image making it difficult to discard data or reduce the precision of the data without adversely affecting image quality. With a suitable choice of transform, the data is 'easier' to compress in the transform domain. There are several desirable properties of a particular transform for compression. It should compact the energy in the image (concentrate the energy into a small number of significant values), it should also decorrelate the data so that discarding significant data has a minimal effect on image quality whilst also being suitable for implementation in hardware and software.

The two most widely used image compression transforms are the discrete cosine transform (DCT) and the discrete wavelet transform (DWT). The DCT is usually applied to small, regular blocks (*macroblocks*) of the image sample samples (e.g. 8 x 8 square), effectively separating the video block into parts of differing importance. Important parts of the block are retained for further processing while the remaining parts are discarded. This approach ensures that the human eye does not notice the removal of less important parts of the video block, whilst limiting the overall bit rate [2]. The DWT is usually applied to larger image selection (e.g tiles) or to complete images. The DCT has proven to be particularly durable and is at the core of most of the current generation of image and video coding standards. The DWT is however gaining popularity as it can outperform the DCT for still image coding (JPEG-2000) and for still texture coding in MPEG-4 [3].

It is also important to discuss *temporal* and *spatial* redundancy and how they can both contribute to video compression. Compression can be achieved by taking advantage of the fact consecutive frames in a video sequence are often almost identical. This redundancy has the potential for a major reduction of storage or bandwidth required over simply encoding each frame separately, but the effect can be lessened by the fact that video often contains frequent scene changes [4].

Spatial compression refers to the bit reduction achieved on the pixels available in a single frame. This is possible as pixels that are located side by side in frame often have similar luminance and chrominance values. Rather than encoding each individual pixel, the spatial redundancy technique encodes the difference between

neighbouring pixels. The numbers of bits required to represent the difference between neighbouring pixels is in most cases less that the amount of data required compressing each pixel individually [5].

The codecs under investigation in this particular research paper will be the MPEG-4 Part 2 Standard along with the more recently approved MPEG-4 Part 10 Standard, which is more commonly known nowadays as H.264. Both of these have their foundations in the earlier compression standard of MPEG-2. The MPEG-2 standard itself is still in use today and not confined to just IPTV, but also used in the transmission of terrestrial and satellite television as well as on storage media such the Digital Video Disc (DVD). This paper will not discuss the MPEG-2 standard in any great detail, as it did not make up any part of the testing procedures. Additionally, the H.264 standard is poised to take over as the dominant compression type owing to its excellent compression efficiency over older standards across a broad range of bandwidths, from low resolution and bitrate 3G mobile video transmission up to full 1080p video storage on Blu-ray discs.

3 Video Quality Assessment

H.264's higher quality rating at a given encoding bitrate, compared to older codecs, leads onto this discussion as to how video quality and codec performance can be objectively gauged. Lossy compressed video and still images by their inherent nature are subject to degradations in quality. In essence, compression removes redundant information from the video or image and later reconstructs so that an identical or nearly identical image is displayed in relation to the uncompressed source image. When an image is compressed to too great a degree, artefacts can be seen upon decoding. This can be achieved be specifying a high quantisation factor when encoding using a tool such as FFmpeg and/or specifying to use a very low target bitrate. Quantisation is the stage after the DCT transformation and before encoding that is responsible for the 'lossy' part of the compression process. An example of a quantisation-induced artefact is shown in Figure 1.

Fig.1 Compressed 'Akiyo' frame on left, encoded at 320 Kbps. On the right, heavily compressed version with loss of detail and pixilation. Encoded at 16 Kbps H264

In the sequence shown in Figure 1, there is comparatively little motion throughout, additionally many of the pixels displayed for a given frame possess a large degree similarity with their neighbours in terms of both colour and contrast, which make this quite easy to encode at a low average bitrate whilst retaining fidelity.

A video sequence possessing lots of motion (sports footage for example) with little or no inter- or intraframe redundancy will be much more difficult at a low bit rate whilst retaining fidelity to the uncompressed source. Network transmission impairments such as packet loss and jitter can also have a massive effect on video quality, introducing 'blocking' and smearing of the outputted video.

Assessing video quality itself and the also effects of encoding artefacts and transmission impairments falls into two distinct categories; subjective human-based testing of video footage and the objective computer based testing of values such as peak signal to noise ratios for each frame.

In the case of subjective video testing, sets of video frames are generated with varying encoding parameters. Observers are then invited to subjectively the visual quality of these frames. Alternatively observers are to provide some measurement of impairment to the pictures. A five-point scale rating system of the degree of impairment has been adopted for these types of objective tests. At the lowest end of the scale the impairment is considered to be "not noticeable". At the opposite end of the five-point scale the impairment is considered to be "extremely objectionable" [6].

However, rating video quality in this fashion has the potential to become very time consuming in terms of hiring observers, later analysing results and drawing conclusions. Other myriad factors such as viewing distance and ambient light settings can affect perception of video quality to a great extent. Furthermore, subjective quality ratings are typically higher when the test scenes are accompanied by high quality sound [7].

The alternative approach to gauging compressed image of video quality is employ computer based tests such as MSE, PSNR and SSIM. The MSE (mean square error) and PSNR (peak signal ratio) are popular metrics in image and video processing. The MSE is the square of the difference between the grey-level in two pictures or sequence I and $\tilde{I}$ (which is the uncompressed reference image or video.

$$\mathrm{MSE} = \frac{1}{TXY}\sum_{t}\sum_{x}\sum_{y}[I(t,x,y) - \tilde{I}(t,x,y)]^2$$

for pictures of size X * Y pixels and T frames in a sequence. Root mean square error is simply the square root of the MSE value.

Closely related to MSE, another formula used in determining image compression is Peak Signal to Noise Ratio (PSNR). PSNR is expressed in decibels as the ratio between the power of a video signal and the power generated by electromagnetic noise,

$$\mathrm{PSNR} = 10 \log \frac{m^2}{\mathrm{MSE}}$$

where m is the maximum value that a pixel can take (255 for 8-bit images). PSNR measures image fidelity i.e. how closely the processed or compressed sequence resembles the original, uncorrupted source. Both MSE and PSNR use only luminance as no agreement exists on how to apply the formulas or interpret results when chrominance is taken into account. PSNR values can be mapped to equivalent subjective Mean Opinion Scores, as described in Table 1.

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

Tab. 1 PSNR to MOS conversion

Whilst both metrics are quick and easy to calculate, both operate at the pixel-by-pixel level and can fail to take into account errors glaringly obvious to the human visual system. Encoding distortions are much more disturbing in relatively smooth areas than in textured or complex regions of frame, an effect not taken into account by these metrics. Therefore the human perceived quality of images or video sequence with the same PSNR can be quite different [7]. When these facts are taken into facts are taken into account, we can see than objective video based metrics such as MSE and PSNR do not always provide reliable picture quality assessment. However their implementation is much faster and easier than that of subjective based assessments. Furthermore, objective based metrics are repeatable. Owing to these merits, objective quality assessments are still used despite these drawbacks [8].

Some other video quality metrics include 'Structural Similarity' (SSIM) and 'Video Quality Metric' (VQM). These metrics offer more reliable results regarding video quality, but require higher computational complexity to calculate [9].

The Structural Similarity Index (which was used alongside PSNR in this paper) improves upon PSNR and MSE metrics, which are inconsistent with human visual characteristics. The SSIM metric is based on frame-to-frame measuring of three components (luminance similarity, contrast similarity and structural similarity) and combining them into a single value called an index. The SSIM index is a decimal value between 0 and 1, where 0 mean zero correlation with the original image, and 1 means the exact same image [10].

4 'Group of Picture' Fundamentals

One of more important concepts behind MPEG encoding is that of frame type. There are three defined type of frame. An 'I' frame is frame compressed solely on the information contained in the frame, no other reference is made to any other video frames before or after it. The "I" strands for "intra" coded. A 'P' frame is a frame that has been compressed using the data encoded in the frame itself and the data from the closest preceding P frame or I frame. The "P" stands for predicted. It is not actually an encoded image and contains motion information that allows the decoding device to rebuild the frame. P frames require less bandwidth than I frames. The final type of frame is known as the 'B' frames, which signifies that it a 'bidirectional' frame, composed of information from both I and P frames [11].

All these different frame types are then combined in a specific repeating order to create what is known as a Group of Pictures (GOP). The first frame of the GOP is always an I frame, not only by definition, but also by virtue of the fact that the I frame is the only frame kind that can decode by itself without reference to other frame. This allows all the other frames in the GOP to use it as a reference. A typical GOP pattern might be represented as IBBPBBPBBPBB(I), where each letter represents viewing order and type. The size (or interval between each repeating I frame) of this GOP is 12. See Figure 2.

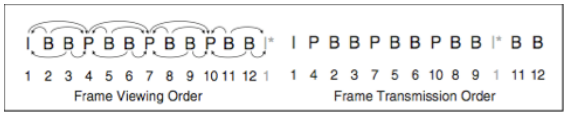

Fig. 2 Viewing Order vs. Transmission Order [12]

5 Testbed

5.1 Source Videos

As this paper aims to further establish the relationship between video content, encoding strategy and transmission impairments, video with varying degrees of motion and textural complexity were sought. These video sequences, similar to those already used in other video research papers were found online [13] in their native raw YUV format, each encoded with 4:2:0 chroma subsampling at CIF resolution (352 * 288 pixels) and displayed at 25 frames per second. Each sequence used for testing was 300 frames long (or 12 seconds in duration). Six video sequences were eventually decided upon and grouped according to increasing textural complexity or motion. See Table 2.

Rank	Motion Content	Texture Complexity
1 (least)	Suzie	Akiyo
2	Foreman	Container
3 (most)	Football	Mobile

Tab. 2. Final Testbed videos ranked in order of increasing motion and texture complexity

5.2 Encoding

FFmpeg, a complete cross platform solution to record, convert and stream audio and video [14], handled all the encoding of video sequences. The uncompressed source YUV was chosen as the input while destination and name is chosen for the output in fashion similar to the command line example below

$ ffmpeg –i ~/folder/input.yuv -sameq ~/folder/output.mp4

As the MPEG-4 videos, as well as the versions created whilst simulating network packet loss are evaluated using metrics such PSNR and SSIM, one should remove as much as possible the effects of compression quantisation during evaluation and focus just on the effect that packet loss have on video quality. This was achieved by using the –sameq switch while encoding from the command line. Doing so produced large files, but the typical average PSNR of the output video file (with default encoding parameters) was typically between 41 and 47 dB, depending on how easy the sequence was to encode (easier sequences such as 'Akiyo' with the highest average values). All generated video files had a mean SSIM index approaching 0.99, indicating a near perfect replication of the source YUV video, albeit at a fraction of the size. Additionally, as the motion or textural complexity of outputted file increased when using the -sameq encoding option so did the average bit rate and file size of the video in question. All videos were of fixed resolution, frame rate and length.

Using FFmpeg's encoding parameters it is possible to directly specify which GOP size to use and how many B frames to employ between successive P frames using the –g and –bf switches, respectively. See Table 3 for list of the GOP sizes and structures chosen .The number proceeding 'BF' indicates how many B frames are stored between each P frame within a GOP.

All Frame Types	I & P Frames Only
GOP16 BF2	Encoder Default - GOP 12
GOP16 BF3	Intra Frames Only
GOP16 BF4	Long - GOP 32
GOP16 BF5	Medium – GOP 16
	Short – GOP 6
	Super – GOP 250

Table. 3 Chosen GOP sizes and structures for test set

5.3 Hinting

Once a video of a sequence of a particular GOP size and composition has been prepared it is then necessary to prepare the file for transmission across a real or simulated network by 'hinting' it, which involves adding a 'hint' track. This hint track will then tell the server exactly how to packet the media data for the network e.g. MTU size etc.

All the video sequences in this experiment were hinted with MP4Box, an MP4 multiplexer [15]. Only once the video files have been prepared via MP4Box can the video quality experiment truly begin.

5.4 Evalvid & NS2

Evalvid represents a complete framework and toolset for the evaluation of the quality of video transmitted over a real or simulated communication network. Aside from measuring the QoS parameters of the underlying network such as loss rates, delay and jitter it also supports the supports the objective video quality of the received video based on frame-by-frame PSNR calculation as well as SSIM. The toolset is freely available to the public [16]. The version 2.7 used in this paper supported the MPEG-4, H.263 and H.264

standards for sequences at CIF and QCIF resolution. See Figure 3 for schematic diagram. Integrating it to work alongside the network simulator NS-2 can also enhance Evalvid's basic functionality, potentially allowing researchers to analyse video transmission performance over large simulated networks [17]. Evalvid and NS2 were used in this particular case to model the UDP transmission of video between 3 nodes on a network with 2 duplex links between them, each with a particular level of bandwidth and transmission delay.

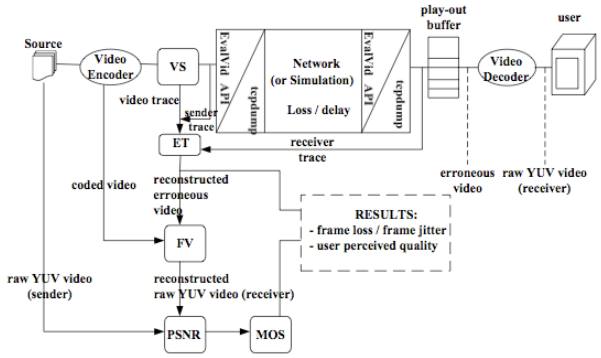

Fig. 3 Schematic illustration of the framework provided by Evalvid [16]

To start the process, a video 'trace' will be generated by streaming a hinted video to another machine on the network using the mp4trace program in the toolset. This will examine every frame of video and produce a log flle containing statistics such frame number, frame type, size in kilobytes, packets required to transmit and transmission time in relation to the overall duration of the video sequence. Sample shown below.

1	I	65232	45	0.071
2	P	39067	27	0.073
3	P	41142	29	0.081
4	P	41579	29	0.120
…	…	…	…	…

Once the trace file has been generated, an NS-2 tcl script file (see Figure 4) can then parse through the video trace file and then generate sender (sd_a01) and receiver (rd_a01) 'dump' files for the simulated network, each of which contain packet data such as transmission time, unique packet id and packet size on separate lines. Altering parameters within the NS tcl script such as throttling bandwidth or increasing transmission delay can cause packets to drop, which is reflected in real world conditions and in the resulting receiver dump file. Alternatively, NS-2 could have been omitted by using mp4trace to transmit video between two machines on a real ad-hoc network, whilst sender and receiver dump files could have been generated by running tcpdump file on each machine prior to transmission. This approach was originally adopted during early testbed development, however the receiver dump file always had to be copied back to the sending machine to be used in later analysis, which was very not only inconvenient, but exposed to tests to missing or duplicate data, due to the volume of simulations undertaken.

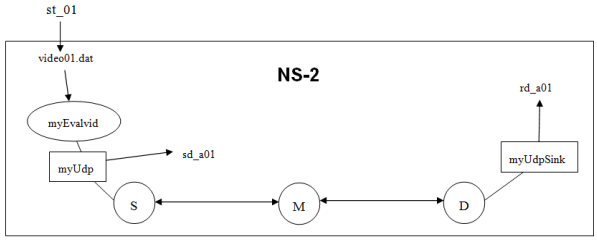

Fig. 4 NS-2 Transmission Schema

Evalvid uses the difference between the data contained in the sender and receiver dump file to compute the degree of data loss in the regenerated video (which can be demonstrated when bandwidth is throttled, within the tcl script). As such, in order to simulate loss, a Python script was developed [18] that removed a set number of lines (each representing a packet) randomly from the receiver dump file. In this case the bandwidth parameters in NS-2 tcl scrpt were set at a high level so that no packet loss was simulated initially and the resultant sender and receiver dump files contained an equal number of lines (packets sent being equal to packets received). For a given receiver dump file, the number of packets or lines were counted and the Python script then deleted 1% and 5% of the lines/packets contained, outputting to two separate text files.

As I frames are fully specified frames that don't depend on other frames to decode, an I frame damaged due to packet loss has a far greater negative effect on visual quality than the loss of other frame types, as P and B frames derive and convey changes in motion etc. from this frame type. In effect, a damaged I frame perpetuates the transmission error across the GOP. The packet loss model used in experiments was totally random, but as I frames require more packets to transmit data than P and B frames, there is with increasing

packet loss rates a proportionally greater random chance that an I frame will be affect by packet loss.

Once the sender dump, receiver dump (with random packet loss induced) and video trace files have been generated they can be used along with the originally encoded file to regenerate a corrupted video file that will reflect the effects of simulated packet loss.

The 'etmp4' (or 'evaluate trace') program in the toolset handled this duty. The video file generated by etmp4 can be then easily converted to YUV using FFmpeg, whereby tests for objective quality tests SSIM and/or PSNR can be carried out against the original source YUV files using the included 'psnr' program. This essentially completes the loop of video quality assessment when using Evalvid.

As a secondary test, video sequences were transmitted between a client and server machine on LAN, both running VLC Player initially with perfect network conditions and then with introduced packet loss. The network emulator for Linux, *Netem* was used to introduce set levels of random packet loss. The video received whilst packet loss was being emulated very closely matched the video generated by Evalvid when packet loss was being *simulated*, in terms of transmission artefacts generated and their relative levels, for each video sequence.

When video (and other data) is transmitted over a IP network such as the Internet, each frame is broken down in a series of packets by the sending device. These packets then travel across the infrastructure of the network, passing through devices such as routers, hubs and switches before being received by the device that is intended to display the video. Here the packets are then reassembled to form the video frames. The number of packets each individual frame needs for transmission is determined primarily by the bit rate of that frame. The larger the bit rate of a frame the more packets that will be required for transmission. All packets used by Evalvid and NS2 in experiments had a fixed maximum packet size of 1052 bytes, including a UDP header of 8 bytes and an IP header of 20 bytes.

The results presented focus primarily on the effects of packet loss on transmitted video sequences, each with differing GOP sizes and compositions.

Results tables to follow on next page.

Results Tables – Motion Content Videos (MPEG-4 Encoded)

Note that 'GOP' signifies 'Group of Pictures'. 'BF' denotes how many B frames are stored between successive P frames in a given GOP

	Encoded		Transmitted – 1% Packet Loss						Transmitted – 5% Packet Loss					
	Bitrate Kbps	**Mean SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**
GOP 16 BF2	882	0.99	33.89	5.95	-12.95	0.07	0.07	-0.08	29.01	7.59	-17.83	0.84	0.12	-0.16
GOP 16 BF3	914	0.99	34.59	6.89		0.92	0.09	-0.0	30.79	8.07		0.84	0.13	-0.16
GOP 16 BF4	956	0.99	35.24	7.63		0.92	0.18	-0.08	30.73	6.78		0.86	0.12	-0.14
GOP 16 BF 5	1015	0.99	34.93	8.95		0.90	0.12	-0.1	30.48	7.7		0.84	0.14	-0.16
I & P Only														
Enc. Default	870	0.99	37.75	7.9		0.94	0.07	-0.06	31.52	8.76		0.85	0.16	-0.15
Intra Only	3039	0.99	35.13	6.17		0.93	0.07	-0.07	33.43	6.52		0.91	0.09	-0.09
Long GOP 32	679	0.99	35.26	7.31		0.93	0.07	-0.07	32.12	7.88		0.88	0.12	-0.12
Med. GOP 16	839	0.99	36.33	7.15		0.94	0.07	-0.06	32.59	8.44		0.87	0.14	-0.13
Short GOP 6	1063	0.99	36.58	7.53		0.94	0.07	-0.06	32.97	7		0.90	0.11	-0.10
Super GOP 250	714	0.99	34.36	6.3		0.93	0.07	-0.07	19.16	5.27		0.69	0.10	-0.31

Tab. 4 Results for 'Suzie' video sequence. Least motion throughout

	Encoded		Transmitted – 1% Packet Loss						Transmitted – 5% Packet Loss					
	Bitrate Kbps	**Mean SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**
GOP 16 BF2	2278	0.99	23.50	4.97	-18.75	0.66	0.21	-0.34	17.41	4.37	-24.94	0.45	0.16	-0.55
GOP 16 BF3	2791	0.99	24.01	6.34		0.66	0.21	-0.34	16.59	4.43		0.43	0.18	-0.57
GOP 16 BF4	2832	0.99	23.13	4.9		0.62	0.19	-0.38	18.85	4.24		0.48	0.17	-0.52
GOP 16 BF 5	2929	0.99	22.26	5.69		0.64	0.23	-0.36	19.1	5.16		0.50	0.2	-0.50
I & P Only														
Enc. Default	2968	0.99	24.9	4.93		0.74	0.17	-0.26	20.81	5.19		0.59	0.2	-0.41
Intra Only	6121	0.99	25.24	4.49		0.73	0.17	-0.27	22.28	4.24		0.62	0.19	-0.38
Long GOP 32	2798	0.99	23.42	5.17		0.71	0.17	-0.29	17.49	3.34		0.48	0.13	-0.52
Med. GOP 16	2892	0.99	25.73	4.29		0.76	0.16	-0.24	19.49	4.87		0.54	0.18	-0.46
Short GOP 6	3242	0.99	22.56	5.55		0.76	0.17	-0.24	21.06	5.09		0.60	0.20	-0.40
Super GOP 250	2717	0.99	22.15	6.49		0.65	0.18	-0.35	13.46	0.7		0.37	0.1	-0.63

Tab. 5 Results for 'Foreman' video sequence.

		Encoded			Transmitted – 1% Packet Loss						Transmitted – 5% Packet Loss					
		Size MB	Bitrate Kbps	Mean SSIM	Mean PSNR	SD PSNR	Delta PSNR	Mean SSIM	SD SSIM	Delta SSIM	Mean PSNR	SD PSNR	Delta PSNR	Mean SSIM	SD SSIM	Delta SSIM
	GOP 16 BF2	5.94	4150	0.99	20.16	6.29	-23.92	0.56	0.18	-0.44	16.05	2.69	-28.03	0.31	0.16	-0.69
	GOP 16 BF3	6.14	4921	0.99	20.83	6.62		0.59	0.20	-0.41	16.25	2.16		0.33	0.13	-0.67
	GOP 16 BF4	6.27	4383	0.99	18.39	2.86		0.51	0.16	-0.49	16.18	2.49		0.31	0.21	-0.69
	GOP 16 BF 5	6.65	4625	0.99	19.95	4.78		0.57	0.21	-0.43	17.35	2.76		0.43	0.15	-0.57
	I & P Only															
	Encdr. Default	5.68	3970	0.99	21.68	6.21		0.62	0.20	-0.38	17.54	5.77		0.37	0.22	-0.63
	Intra Only	9.79	6841	0.99	20.47	4.47		0.61	0.19	-0.39	18.40	3.20		0.5	0.19	-0.50
	Long GOP 32	5.47	3823	0.99	18.77	4.31		0.53	0.16	-0.47	16.11	2.91		0.31	0.13	-0.69
	Med. GOP 16	5.60	3912	0.99	21.67	7.06		0.63	0.21	-0.37	17.74	3.85		0.43	0.2	-0.57
	Short GOP 6	6.04	4221	0.99	20.83	4.56		0.62	0.18	-0.38	17.38	2.99		0.45	0.14	-0.55
	Super GOP 250	5.34	3730	0.99	17.68	4.50		0.46	0.13	-0.54	15.86	2.82		0.3	0.12	-0.70

Tab. 6 Results for 'Football' video sequence. Most motion throughout

Results Tables – Textural Complexity Videos (MPEG-4 encoded)

		Encoded		Transmitted – 1% Packet Loss						Transmitted – 5% Packet Loss					
		Bitrate Kbps	Mean SSIM	Mean PSNR	SD PSNR	Delta PSNR	Mean SSIM	SD SSIM	Delta SSIM	Mean PSNR	SD PSNR	Delta PSNR	Mean SSIM	SD SSIM	Delta SSIM
	GOP 16 BF2	348	0.99	44.03	3.03	-3.5	0.99	0.01	-0.01	36.80	9.11	-10.73	0.94	0.13	-0.06
	GOP 16 BF3	354	0.99	41.52	4.24		0.99	0.01	-0.01	37.09	7.5		0.95	0.09	-0.05
	GOP 16 BF4	372	0.99	42.72	3.33		0.99	0.00	-0.01	37.34	6.37		0.97	0.02	-0.03
	GOP 16 BF 5	610	0.99	42.02	3.64		0.99	0.01	-0.01	36.14	8.26		0.95	0.09	-0.05
	I & P Only														
	Enc. Default	350	0.99	44.11	3.21		0.99	0.01	-0.01	38.94	6.94		0.96	0.08	-0.04
	Intra Only	/	/	/	/		/	/	/	/	/		/	/	/
	Long GOP 32	421	0.99	42.26	0.99		0.99	0.01	-0.01	37.61	9.14		0.94	0.13	-0.06
	Med. GOP 16	302	0.99	42.50	3.35		0.99	0.01	-0.01	41.47	3.27		0.99	0.01	-0.01
	Short GOP 6	390	0.99	42.52	3.20		0.99	0.00	-0.01	41.55	2.85		0.98	0.00	-0.02
	Super GOP 250														

Tab. 8 Results for 'Akiyo' video sequence. Least amount of textural complexity throughout

	Encoded		Transmitted – 1% Packet Loss						Transmitted – 5% Packet Loss					
	Bitrate Kbps	**Mean SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**
GOP 16 BF2	1552	0.99	36.92	2.59	-5.97	0.96	0.02	-0.04	28.67	6.31	-10.73	0.88	0.11	-0.12
GOP 16 BF3	1505	0.99	34.34	5.72		0.94	0.08	-0.06	28.67	6.31		0.88	0.11	-0.12
GOP 16 BF4	1572	0.99	33.25	6.43		0.92	0.10	-0.08	23.88	6.11		0.79	0.10	-0.21
GOP 16 BF 5	1618	0.99	33.34	6.29		0.92	0.09	-0.08	26.27	6.4		0.83	0.12	-0.17
I & P Only														
Enc. Default	1753	0.99	37.63	2.19		0.97	0.01	-0.03	29.33	4.60		0.88	0.05	-0.12
Intra Only	/	/	/	/		/	/	/	/	/		/	/	/
Long GOP 32	1536	0.99	36.28	2.41		0.96	0.01	-0.04	28.75	6.20		0.87	0.11	-0.13
Med. GOP 16	1665	0.99	37.42	2.93		0.97	0.01	-0.03	28.78	6.44		0.87	0.10	-0.13
Short GOP 6	2111	0.99	37.10	3.76		0.96	0.06	-0.04	31.89	6.21		0.90	0.10	-0.10
Super GOP 250	1424	0.99	18.19	7.06		0.70	0.12	-0.3	27.11	4.37		0.85	0.05	-0.15

Tab. 9 Results table for the 'Container' video sequence . Increased textural complexity relative to 'Akiyo'

	Encoded		Transmitted – 1% Packet Loss						Transmitted – 5% Packet Loss					
	Bitrate Kbps	**Mean SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**	**Mean PSNR**	**SD PSNR**	**Delta PSNR**	**Mean SSIM**	**SD SSIM**	**Delta SSIM**
GOP 16 BF2	7170	0.99	19.31	3.35	-21.43	0.80	0.12	-0.20	13.69	2.25	-27.65	0.17	0.19	-083
GOP 16 BF3	7091	0.99	19.47	4.08		0.75	0.17	-0.25	13.52	2.33		0.16	0.15	-0.84
GOP 16 BF4**	7197	0.99	19.30	5.74		0.47	0.24	-0.53	25.27**	9.69		0.78	0.18	-0.22
GOP 16 BF 5	7279	0.99	18.13	5.65		0.52	0.26	-0.48	11.82	0.26		0.12	0.05	-0.88
Enc. Default	8250	0.99	18.50	3.44		0.62	0.19	-0.38	13.12	1.63		0.21	0.18	-0.79
Intra Only	/	/	/	/		/	/	/	/	/		/	/	/
Long GOP 32	7984	0.99	18.01	4.27		0.58	0.20	-0.42	12.19	2.12		0.17	0.15	-0.83
Med. GOP 16	8140	0.99	19.35	5.37		0.62	0.20	-0.38	11.33	0.43		0.12	0.88	-0.88
Short GOP 6	8984	0.99	20.31	3.41		0.70	0.19	-0.30	12.68	2.05		0.18	0.16	-0.82
Super GOP 250	7846	0.99	15.05	3.15		0.46	0.16	-0.54	11.30	0.44		0.12	0.07	-0.88

*Tab. 10 Results Table for 'Mobile Video' sequence. Contains the most textural complexity throughout. i.e the least spatial redundancy all test sequences. ** File regenerated using different model in etmp4, where frame loss is based on only the first packet belonging to a given frame as missing.*

6 Results Discussion

Examining the results table for each of the six MPEG-4 encoded videos, some trends emerge in regard to how packet loss can affect video quality as well as how some GOP types perform better than others. This is best shown in the summarised table forms below. As outlined earlier, packet loss was introduced randomly across frame types, which adds a certain degree of uncertainty to results. To negate this would require extensive generation of many alternate receiver dump files at each level of packet loss and regeneration of video files, as well as YUV files to make SSIM/PSNR comparisons. The data presented in the results tables required 7 gigabytes of hard drive space alone to generate, due to large size of each YUV file created.

The performance of a given GOP type was determined by finding the PSNR and SSIM values for each of the regenerated videos, and subtracting that value from the SSIM or PSNR of the original video. The video with a particular GOP type whose change or 'delta' was the least was considered the optimal GOP - some videos were encoded using intra frames only and are thus while the most resilient to packet loss artefacts, they are also the least bandwidth efficient and cannot be considered as having an optimal GOP. Additionally, if a particular GOP type was found to have the smallest delta SSIM value, but had a fellow cohort member with a similarly small delta SSIM that required significantly less bandwidth to transmit, that was instead chosen to be the optimal GOP type.

SSIM was chosen as the qualifying metric as all the *original* hinted and FFmpeg encoded videos had an SSIM index at or approaching 0.99 and the standard deviation of values for each frame was comparatively narrow. The mean PSNR values for original encoded video sequences with a GOP size of 16 and B frame value of 2 (i.e. IBBPBBP…) ranged from between 41.34 dB to 47.53 dB; additionally, individual values for I frames could peak dramatically, reinforcing the choice of SSIM as the qualifying metric as it had a standard 'base' value for each video sequence being tested. PSNR values were still recorded nevertheless; so as to draw some comparisons between the values arrived at by each.

[In the tables below, note that the first set of values listed for each sequence represents testing when all three frame type were employed; the second set of values was for when testing only examined the performance of GOPs with I & P frames only.]

	1% Packet Loss	5% Packet Loss
Suzie	GOP 16 BF2	GOP 16 BF4
	Default (GOP 12)	Short GOP 6
Foreman	GOP 16 BF2	GOP16 BF5
	Medium GOP 16	Default (GOP 12)
Football	GOP 16 BF3	GOP 16 BF 5
	Medium GOP 16	Short GOP 6

Tab. 11 Optimal GOP sizes and types for increasing motion content video sequences.

	1% Packet Loss	5% Packet Loss
Akiyo	GOP 16 BF2	GOP 16 BF4
	Medium GOP 16	Medium GOP 16
Container	GOP 16 BF 2	GOP 16 BF 4
	Medium GOP 16	Short GOP 6
Mobile	GOP16 BF2	n/a
	Short GOP 6	n/a

Tab. 12. Optimal GOP sizes and types for video sequences of increasing textural complexity. Results for Mobile @ 5% packet loss have been omitted. Videos too corrupted to generate any meaningful results; SSIM ≈ 0.15

One particular trend that emerges is that the default FFmpeg encoding scheme (for MPEG-4; GOP size 12, no B frames) seldom offers optimal results for video to be transmitted over networks where packet loss might be encountered. Additionally, for increasing random packets loss, having an increased concentration of B frames within a GOP tends to result in higher objective quality. This trend appears to replicate itself for all test sequences.

When B frames are not employed within a given GOP, the best performance for simulated 1% packet loss tends to be a GOP size of either 12 or 16. The texturally complex 'Mobile' sequence seems to be the one exception to this trend. Invariably, shortening the GOP size to just 6 frames for periods of heavy random packet loss offers an improvement in objective SSIM test scores. This is by virtue of the fact that the next I frame to 'correct' the damage caused by random packet loss is always much closer than when the GOP size is 12 or 16, in the test set. However videos encoded in this fashion have a much higher 'overhead' as there are far more I frames to transmit, with each I frame carrying a much

greater payload of data as these frames are not compressed to the same extent as P or B frames.

Videos encoded with GOP sizes of 'Long' (32 frames) or 'Super' (250 frames), never featured as having the optimal GOP size, in tests. In fact, video encoded with a GOP size of 250 displayed the worst performance for both levels of random packet loss in all tests.

Examining the results tables as a whole also shows that a predefined level of random packet loss doesn't always translate into equal delta PSNR and SSIM values for each video sequence. Owing to the wealth of data presented in the tables, this is best represented in a series of charts, the first of which is shown in Figure 5. The charts represent a sample of one of the GOP configurations used in testing (GOP 16 and BF 2). As the induced packet loss is totally random, thereby not expressing a preference for any particular frame type, these results are prone to a degree of variability i.e. if proportionally more packets constituting I frames were lost or corrupted instead of B frames then the decrease in visual quality would be greater.

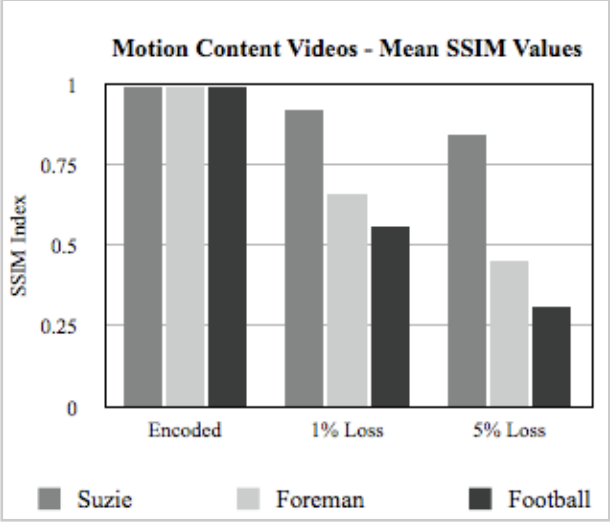

Fig. 5 Mean SSIM values for increasing motion content video sequences (listed left to right), grouped according to increasing random packet loss rate.

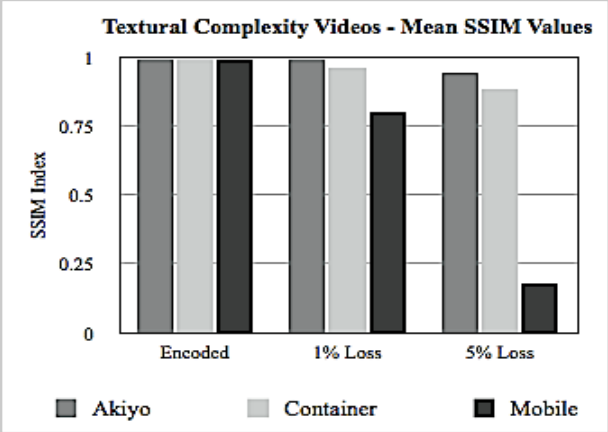

Fig 6. Mean SSIM values for increasing textural complexity videos, grouped according to increasing random packet loss rate.

Video sequences that required much higher bitrates to maintain a high degree of fidelity in relation to their raw, uncompressed YUV counterparts (such as 'Football' and 'Mobile'), all tended to display packet loss artefacts to much greater degree at a given packet loss percentage than video such as easier to encode, lower bitrate video such as 'Akiyo' and 'Suzie'.

Playback of 'Football' at just 1% random packet loss resulted in artefacts such as noise and blockiness (a.k.a macroblocking) appearing sporadically, as demonstrated in Figure 7. Playback at 5% random packet loss resulted in a stuttering, jerking effect throughout the sequence, rendering the sequence barely watchable.

Likewise, the slow moving but spatially complex 'Mobile' sequence displayed jerking, noise and trailing of colours when replayed at 1% packet loss. At 5% packet loss the sequence bore no similarity whatsoever to its uncompressed, untransmitted version so bad were the artefacts.

Packet loss at either level was scarcely evident for playback of the 'talking head' type of sequence 'Akiyo'.

Transmission of these encoded sequences over a LAN with Netem induced packet loss displayed similar levels of artefacts (measured subjectively), in proportion to level of motion or textural complexity contained throughout. The aim of this step was to benchmark Evalvid generated videos against those transmitted over 'real' networks with set levels of packet loss.

Fig. 7'Football' sequence with 1% simulated packet loss

Examining the 'delay' log files Evalvid generates along with each video it creates, details whether or not a given frame was successfully received and decoded (as well as statistics for each frame such as end-to-end transmission delay and cumulative jitter), confirmed the relationship between the number of frames lost and the change in the objective quality at a given degree of packet loss. See Table 13 along with Figures 8 & 9.

Sequence name	Frames Lost	Percentage Frames Lost	Δ PSNR	Δ SSIM
Suzie	47	15	-17.83	-0.16
Foreman	121	40	-29.94	-0.55
Football	162	55	-28.03	-0.69
Akiyo	24	8	-10.73	-0.06
Container	69	23	-14.42	-0.12
Mobile	213	70	-27.65	-0.83

Tab. 13 Relationship between lost frames and changes in objective quality. Data taken from GOP 16 BF2 @ 5% packet loss for each

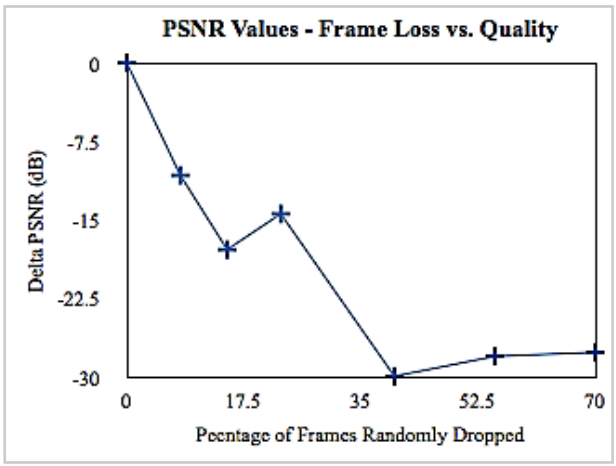

Fig 8. Plot of random frame loss rate against mean PSNR

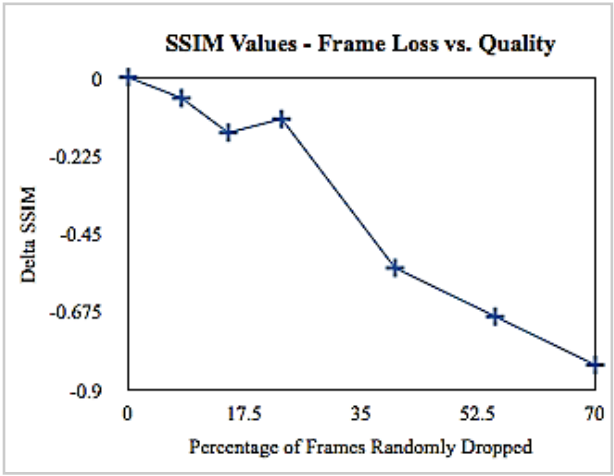

Fig 9. Plot of random frame loss rate against mean SSIM

Looking at Table 13, whilst the same numbers of proportional packets were lost (in this case 5% of the total), there is a massive range in the number of frames not received or decoded properly. The least demanding sequence to encode at an SSIM index of 0.99, 'Akiyo', lost only 8% percent of its *frames*, whilst the most demanding to encode 'Mobile', at an SSIM of 0.99, and also the sequence which required the greatest level of transmission bandwidth, suffered a catastrophic level of *frame* loss at 70%.

Low complexity sequences such as 'Akiyo' and 'Suzie' required the least bit rate (and thus number of packets), whilst 'Football' and 'Mobile' required the greatest. The particular assessment model used when regenerating videos based on data contained in sender and receiver dump files as well as video trace files, was to regard a frame as missing or corrupt if any packet that it contained at transmission time was not present at receive time. This model was found to have the greatest correlation with tests involving video transmission with induced packet loss using Netem.

The alternative model was to only regard a frame as dropped/corrupted if only its *first* packet wasn't properly transmitted. Tests carried out with this model didn't correlate well with Netem transmissions with set packet loss.

To put frames and packets into context, the 'Akiyo' sequence required an average of 1.6 packets for each frame, whilst at the opposite end of the scale, 'Mobile' required an average of 25 packets per frame, in tests, which correlates well with the fact that for 5% packet loss on average, you will lose one packet in 20, or nearly one packet per frame of video (at 25 frames per second), thus 70% frame loss.

As complex sequences such as 'Football' and 'Mobile' have much higher bit rates and thus more packets/frame, this makes them much more vulnerable to the effects of network packet loss, as it potentially takes only 1 lost or corrupt packet to disrupt an entire frame. Tests showed a linear relationship between sequence bit rate/file size and *frame* loss percentage at a fixed packet loss percentage of 5%. See Figure 10.

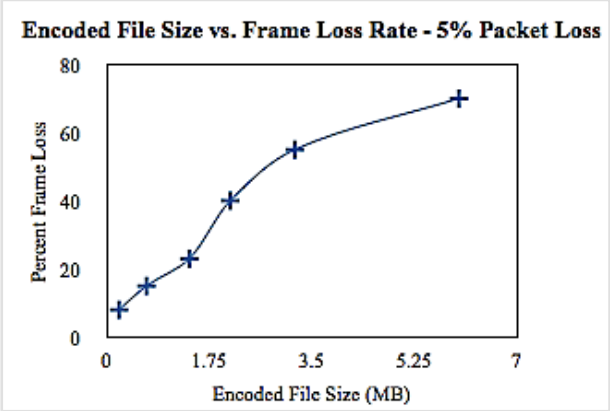

Fig. 10. Plot of file size against frame loss @ 5% packet loss. Packet size 1052 bytes

6.1 SSIM vs. PSNR

Research by Pocta *et al.* [19] suggests that PSNR as a metric is much less sensitive to transmitted content changing, which is one of the key characteristics of the 'Football' video sequence. Additionally, when one examines the declining trend line for delta mean PSNR against frame loss percentage plotted in Figure 8, the line actually starts to *recover* upwards when it encounters the complex 'Football' and 'Mobile' sequences. For the SSIM plot of the same GOP data in Figure 9, the trend line continues to dive for increasing frame loss percentage versus the decrease in SSIM, as one would expect. This validates the work of many other researchers who dismiss as PSNR as an reliable objective visual quality metric, and instead propose using SSIM and VQM, as these both have a far greater correlation with human subjective testing scores.

7 Conclusions

The objective quality tests performed reasonably well in further establishing the relationship between the content displayed in a sequence and the optimal encoding strategy in terms of Group of Picture size and the concentration the bidirectional ('B') frames contained within. Tests also demonstrated that better objective quality results if the GOP size is shortened and/or concentration of B frames is increased whenever packet loss rates start to increase. Whilst sequences such as 'Akiyo' and 'Suzie' were forgiving of high levels of packet loss, more demanding sequences such as the sports footage depicted in 'Football' were not. In cases like this, packet losses of even 0.1% can have a noticeable effect on user Quality of Experience (QoE), suggesting that network management should be an even greater priority for service providers, when transmitting this content type.

Objective tests also demonstrated some deviations in terms of how well PSNR and SSIM assess video quality. The PSNR metric appeared to perform at its best when the transmitted content was changing slowly, as was the case for 'Akiyo', 'Suzie' and 'Container'. Sequences such as 'Football' and 'Mobile' tended to give strange PSNR values when heavy packet losses were introduced. Again this is illustrated in Figures 8 & 9 for each sequences mean delta PSNR & SSIM values, respectively. The two points nearest the right of each plot represent these two particular sequences.

To try and remove some of 'noise' present in results, it would be desirable to run simulations multiple times for each GOP type and sequence and observe if the trends shown in this paper repeat consistently. Alternatively, longer video sequences far greater than the 300 frames used in the these particular tests could have been employed. This will be investigated in later work.

Although all the results here were generated using network simulations and used none of the complex equipment found in an IPTV environment such as streaming servers, firewalls, multiplexers and set-top boxes, the principals of video compression and transmission remain unchanged throughout, thereby making a research contribution to the state-of-the-art.

8 References

[1] I. Richardson, *Video Codec Design:* Wiley, 2002

[2] G O'Driscoll, *Next Generation IPTV Services and Technologies:* Wiley, 2008

[3] I. Richardson, *Video Codec Design:* Wiley, 2002, pp. 35

[4] Gringeri et al, "Robust Compression and Transmission of MPEG-4 Video", Proceedings of the 7th ACM international conference on Multimedia, 1999

[5] G. O'Driscoll, *Next Generation IPTV Services and Technologies:* Wiley, 2008

[6] Y. Shi, H. Sun, *Image & Video Compression for Multimedia Engineering*: CRC Press, 2008

[7] S. Winkler, *Digital Video Quality – Vision Models and Metrics*; Wiley, 2005

[8] Y. Shi, H. Sun, *Image & Video Compression for Multimedia Engineering*: CRC Press, 2008

[9] D. Li, J. Pan, "Evaluating MPEG-4 AVC Video Streaming Over IEEE Wireless Distribution System", Proc. IEEE ICIP, Jan 2008

[10] M. Ghanbari, *Standard Codecs – Image Compression to Advanced Video Coding*: IEEE, 2003

[11] G O'Driscoll, *Next Generation IPTV Services and Technologies:* Wiley, 2008

[12] W. Simpson, *Video Over IP – IPTV, Intetnet Video, H.264, P2P, Web TV and Streaming: A Complete Guide to Understanding The Technology*: Simpson, 2008

[13] Video Research Library - *http://www.tkn.tu-berlin.de/research/evalvid/cif.html*

[14] FFmpeg Website – *www.ffmpeg.org*

[15] MP4Box Website – *www.videohelp.com/tools/mp4box*

[16] Jirka Klaue, Berthold Rathke and Adam Wolisz. "Evalvid - a framework for video transmission and quality evaluation". In *Computer Performance Evaluation/Tools*, pages 255–272, 2003.

[17] C.Ke, C. Shieh, W. Hwang, A. Ziviani, "An Evaluation Framework
for More Realistic Simulations of MPEG Video Transmission", Journal of Information Science and Engineering

[18] John Conroy, MSc Software Design & Development II, IT Department, NUI Galway, Ireland

[19] P. Pocta, R. Hudec, M. Bojmir, "Performance Evaluation of video Transmission over IEEE 802.11b WLANs from the Video Quality Perspective", The Mediterranean Journal of Computers and Networks, Vol. 5, No. 2, 2009