EQUITABLE QUALITY VIDEO STREAMING OVER DSL

Barry Crabtree, Mike Nilsson, Pat Mulroy and Steve Appleby

BT Innovate, Adastral Park, Martlesham Heath, Ipswich IP5 3RE, UK

ABSTRACT

Video streaming has frequently been deployed using constant bit rate video encoding and transmission as these are easily implemented and network provisioning, although requiring admission control, is straightforward. However, it is sub-optimal for both the user, who experiences time varying quality, and the network operator, who cannot fully utilize the network during periods of low usage and is faced with admission control and session refusal at busy times. Instead we propose an equitable quality scheme in which the available network bandwidth is divided between the concurrent sessions so that the same quality is delivered in each. We show that we can increase average quality (Mean Opinion Score) by 0.74, or alternatively increase the number of sessions that can be supported at the same average overall quality by 100%.

1 INTRODUCTION

THERE is growing interest around the world in delivering video services over IP networks, as video compression has improved while internet access speeds have increased, allowing high quality video to be delivered to the home. Many of these systems, and particularly those running over a bandwidth limited “last mile”, currently deliver video encoded at a constant bit rate and delivered at constant bit rate (CBR), because it has predictable network requirements which can be guaranteed with straightforward admission control.

1.1 Background

Encoded video naturally has variable bit rate (VBR), as the number of bits produced when encoding a picture depends on the picture content: how similar it is to previously encoded pictures and how much detail it contains. Some video scenes can be coded to a given quality with a small number of bits, whereas other scenes may require significantly more bits to achieve the same quality.

When CBR coding is used, a small bit buffer is used at the output of the encoder to smooth short term variations in the bit rate produced by the encoder, while long term variations can only be controlled by adjusting the quality of the encoding, resulting in some scenes that are coded well and some that are not.

The overall perceptual quality of a video sequence is not the mean of the quality of small chunks of the video, but is biased towards the worst quality observed [16]. Coding with VBR allows these variations in quality to be avoided, and better overall subjective quality to be achieved for a given total bit budget.

VBR coding with limited buffering at the client has been widely studied: Lakshman et al. [10] provide a good review of the literature. Typically this was studied in conjunction with admission control, but also with “blind” statistical multiplexing [12]. In all cases, to avoid significant amounts of packet loss in the network, the amount of bandwidth that needed to be allocated was more than the mean rate of the stream, often more than negating the benefits of VBR coding over CBR coding. In [10] our work is most closely characterised as ‘Feedback-VBR’ where we use network state information to manage the delivery rate.

1.2 Equitable Quality Video Streaming

In this paper we use the concept of equitable quality video streaming that we first introduced in [4]. We relax the constraint of limited buffering at the client and allow the client to have sufficient buffering to be able to store a whole video sequence. We also allow concurrent video streaming sessions to share the available backhaul network bandwidth such that the same instantaneous video quality is delivered on each of these sessions, regardless of the genre of content. The streaming application manages the quality and the delivery rate of each session.

When there are few users on the network, our approach allows constant higher quality video to be delivered than CBR encoding and delivery at some conservatively chosen bit rate. When more users join the network, equitable quality video streaming allows these users to receive video by reducing the quality of video received by each, whereas CBR coding and delivery may lead to session rejection by the network admission control system.

1.3 Scope and Structure

In this paper, we model the network as a video server connected to a number of client devices over a shared backhaul.

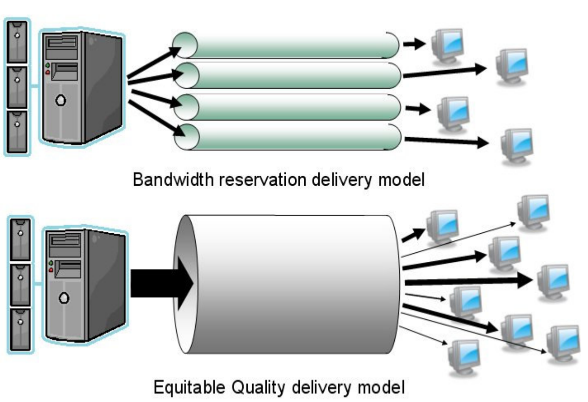

Fig 1. The two network models used for video delivery.

Fig 1 shows the comparison of the models used. Traditionally with IPTV services bandwidth is reserved to guarantee a particular level of service, show as the 'Bandwidth reservation delivery model' while the equitable quality video streaming does not rely on bandwidth reservation, but assumes that the backhaul bandwidth is shared as needed between the video streams.

In section 2 we describe our video encoding configuration. In section 3 we describe how we simulated video delivery over a network, and in section 4 we describe how we calculated the information necessary to perform the simulation. In section 5 we describe a metric to compare the performance of equitable quality video streaming with CBR encoding and delivery, and provide the results in section 7. We conclude in section 8.

2 VIDEO ENCODING

It is well known that the human visual system is more tolerant of distortion in regions of pictures with high spatial activity than in relatively flat areas [3][9]. Consequently it is possible to introduce more distortion in detailed or complex regions, by, for example, using coarser quantisation, while achieving the same perceptual quality.

Davis [6] developed a technique for measuring the perceptual quality of video compressed according to ITU-T Recommendation H.264 [2] where only the compressed video is available. This takes account of the quantisation index and a measure of the spatial masking present in the video signal, to produce estimates of perceptual quality, which have been shown to have good correlation with the results of subjective viewing tests.

To encode with constant perceptual quality, as measured by Davis, we incorporated his algorithm into the video encoder. We measured the spatial masking of the video signal for each video frame, and then selected a single value of quantisation index for coding that frame, so that if the algorithm were applied to the resulting coded video, the perceptual quality would be determined as approximately constant over the entire video sequence.

The BT H.264 software encoder [1] was used for encoding according to ITU-T Recommendation H.264 [2]. It was configured to encode with an IBBP group of pictures (GOP) structure, with I frames coded every 24 frames; two reference frames and macroblock and picture frame field adaptive coding were used, but to reduce the time taken for encoding, CABAC entropy coding was not used.

We used a total of 30 standard definition 25fps video sequences, with durations ranging from 30 minutes to two hours, and representing a wide range of genres from action and drama movies to television drama, news and sport to children's television and user generated content.

Firstly, we encoded each sequence at a CBR of 1Mbit/s with a bit rate smoothing buffer of size 1Mbit. Then, to support equitable quality video streaming, we encoded each video sequence at nine different fixed quality levels, uniformly spaced between 2.6 and 4.2 inclusive as measured on the continuous scale defined in ITU-R Recommendation BT.500 [11], in which the quality terms bad, poor, fair and good are associated with values, known as Mean Opinion Scores (MOS), between 1 and 5 inclusive. The number of bits used to encode each picture was recorded, and was used in the network simulation. There is a trade-off here for the equitable quality video streaming case in that the same content needs to be encoded and stored at the different quality levels. However, with the growth in storage capacities this is not seen to be a significant issue.

3 VIDEO DELIVERY NETWORK MODEL

We modeled the video delivery network as a server connected to client devices over a shared backhaul with a given capacity reserved solely for video streaming sessions. In the access part of the network, each DSL line is capped at between 17% and 33% of the backhaul network.

3.1 Session Initiation

We simulated the operation of the video streaming service over periods of one day. We created a relationship between TV viewing levels and time of day using the data available in [13] and [15], and shown in Fig 2.

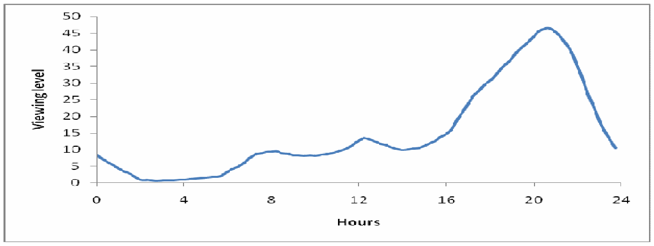

Fig 2. The relationship between viewing levels and time of day used to determine session initiation times.

For each day that we simulated, we generated session initiation times according to this distribution, and for each session, selected one of the pieces of video content at random, with a uniform probability distribution. We continued to add new sessions until a limit was reached. In some experiments this was when the newest session would have been rejected by the network admission control in the CBR reference case; and in other experiments we continued until a given number of sessions had been generated.

3.2 Calculating Video Qualities and Delivery Rates

In the case of CBR coding and delivery, a video stream is simply modeled as being delivered at the encoded rate until it is completely delivered. The backhaul bandwidth determines how many concurrent sessions can be delivered.

In the case of equitable quality video streaming, the aim is to divide the backhaul bandwidth between the concurrent sessions to enable the same best quality to be delivered in each session. We assume the client device has sufficient buffering to be able to store up to a whole video stream, and in this method we are able to switch between a number of pre-encoded sequences at varying qualities. Streaming begins immediately and in our model play out will start within a second. Once the streaming has begin the client buffer allows the timing of delivery of the video data to be decoupled from the decoding of it, provided we ensure data is delivered before it is needed for decoding.

The delivery bit rate required to deliver a given video stream in timely fashion depends on how much data is already buffered at the client, and the bit rate profile of the bits yet to be decoded. By pre-analyzing all of the video streams, and by monitoring the amount of data buffered at the client for each concurrent video session, we can determine the bit rate required for timely delivery.

At the end of every GOP period, and whenever a session starts or ends, we calculate, for each concurrent video session and for each quality level, the bit rate required for the session, given the current amount of data already buffered at the client. We sum these rates for each quality level, and select the highest quality requiring less total bit rate than the network capacity, and, subject to DSL line constraints, deliver this quality on all sessions from this point in time, regardless of the quality level previously delivered. Thus, the (equitable) quality, Q, of all streams is selected according to (1), where $R(q,i)$ is the rate required to deliver stream i at quality q given the amount of data already buffered at the client for stream i, and C is the total backhaul network bit rate available for video delivery. However, this quality may not be possible for all streams, just those where the DSL rate is greater than $R(q,i)$.

$$Q = \arg\max(q) \left| \sum_i R(q,i) \le C \right| \qquad (1)$$

In the cases where the DSL constraint is broken the rates for the streams that do not break the DSL constraints are set to what we have calculated. This bandwidth is then removed, and the calculations redone at lower qualities with the remaining bandwidth until the DSL constraints are met.

4 VIDEO DELIVERY AND BUFFERING

In this section we describe how we analyzed the encoded video streams to allow us to estimate the information needed to solve (1) and hence determine the video quality and delivery rate required for each concurrent video session.

4.1 Video Stream Analysis

After encoding a video sequence with constant quality, we analyzed its bit rate characteristics. For each quality level at which a video sequence was encoded, we determine the mean bit rate over the whole sequence, and then for delivery rates that are 0.6, 0.7, 0.8, 0.9, 1.0, 1.2 and 1.4 times the mean bit rate, we determine the start-up delay that would be needed to start delivery at each GOP in the sequence to achieve continuous play-out. We also determine for each GOP the delivery rate that would be needed to deliver the remainder of the video sequence with no start-up delay and still achieve continuous play-out.

We initialized a hypothetical decoder buffer to be empty, and then filled it at the given rate. After each GOP period, one GOP of bits is removed. A non-zero start-up delay is required to start delivery at a given point if the hypothetical buffer level is ever lower than the level at that given point. The start-up delay required is this difference, normalized by the delivery rate. Fig. 3 shows an example buffer trajectory and the start-up delay required for starting delivery at each GOP of a sequence delivered at its mean rate.

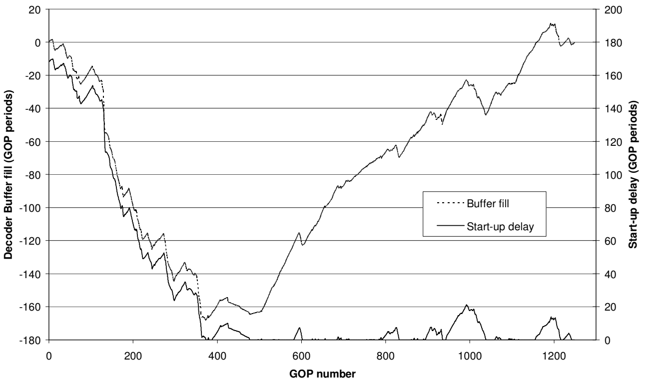

Fig 3. Buffer fill and start-up delay for a sequence delivered at its mean rate.

The bit rate required for delivery with zero start-up delay is calculated as the maximum of, for each subsequent GOP in the sequence, the ratio of cumulative bit count to time, both being measured from the current GOP.

4.2 Delivery Rate Interpolation

To solve (1) we need to know the delivery rate required given the amount of data already buffered at the client, which if measured in terms of the play-out time it represents, corresponds to the start-up delay allowed for the next data to be delivered. As it is not feasible to pre-calculate for all possible values of start-up delay, we estimate the delivery rate from the information that we have pre-calculated. Fig. 4 shows a hypothetical bit consumption trajectory, and bit delivery trajectories for three delivery bit rates, *R1*, *R2* and *R3*. The optimal start-up delay, *S1* and *S3*, for the delivery rates *R1* and *R3* respectively has been pre-calculated. The problem then is to estimate a value for the delivery rate *R2* required for start-up delay *S2*, where *S2* is between *S1* and *S3*.

To ensure timely delivery of bits, the delivery schedule *R2* must be on or to the left of the cumulative bit curve. This is ensured if the delivery schedule *R2* passes through the intersection of the delivery schedules, *R1* and *R3*, which occurs at time *T*, given in (2). An acceptable delivery schedule for rate *R2* is the straight line of slope *R2* passing through this point, given by (3). As shown in Fig. 4, this may not be optimal.

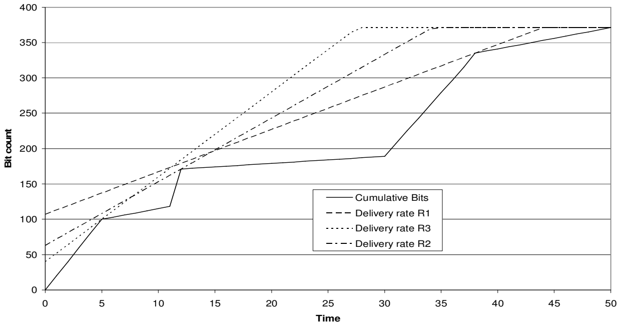

Fig 4. A hypothetical bit consumption trajectory, and three bit delivery trajectories for different delivery bit rates, *R1*, R2, and R3. 'T' is where R1 and R3 intersect (at about 11 on the Time axis)

$$T = (S3 \cdot R3 - S1 \cdot R1) / (R1 - R3) \qquad (2)$$

$$R2 = (S1 + T) \cdot R1 / (S2 + T) \qquad (3)$$

By calculating the start-up delay for a set of delivery rates and the delivery rate for zero start-up delay, we can estimate the delivery rate required for a wide range of start-up delays.

5 COMPARISON METRIC

To compare the performance of equitable quality video streaming with CBR, we need a metric to measure the user experience, focusing on how the video quality varies with time during a streaming session.

It has been widely asserted that users prefer constant quality to time varying quality [7]. The overall perceptual quality of a video sequence is not equal to the average perceptual quality of each individual frame, but is dominated by periods of below average quality [16]. Fig 5 belows a typical CBR coded sequence and how the quality varies over the duration of the video.

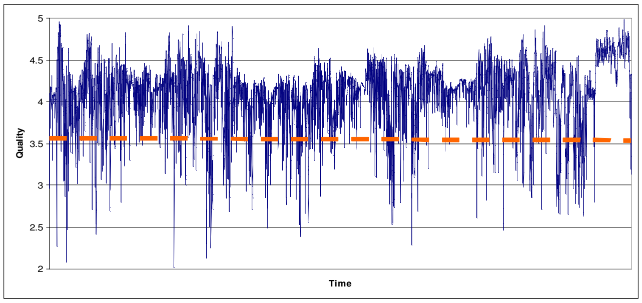

Fig 5. Plot of video quality over time for a CBR coded sequence.

The process to calculate an overall quality score for a video sequence is as follows: Firstly, we calculate the quality of each picture using the technique of Davis [6], in which the quality of a coded picture is a combination of two functions, one being a function of the set of quantisation parameters used to encode the picture and the other being a function of the spatial complexity of the decoded picture. We then calculate the average quality of each GOP as quality variations over this timescale (about one second) are not considered to be perceptible [5]. This results in one quality score for each GOP, which must be combined to give an overall quality measurement.

We used three methods to achieve this: mean, percentile, and weighted mean. The mean is calculated simply as the arithmetic mean of the individual quality measurements.

In the percentile method, the individual quality measurements were sorted from lowest to highest as in the lexicographic approach [8], and the overall quality is set to be the quality at a given position in the sorted list. We selected the position 10% through the list, indicating that 10% of groups of pictures were coded with lower quality than the overall measure, and 90% with higher quality.

The weighted mean is calculated by (4) and (5), where a larger weighting is given to groups of pictures with lower quality. *Q(i)* is the average quality of GOP *i*, *N* is the number of groups of pictures in the video sequence, and *W* is a constant.

$$OverallQuality = \sum_{i=1}^{N} w(i) \cdot Q(i) \Big/ \sum_{i=1}^{N} w(i) \qquad (4)$$

$$w(i) = 2^{(-Q(i)/W)} \qquad (5)$$

When *W* is large, the overall quality tends towards the mean; when *W* approaches zero, it becomes the lowest individual quality score, equivalent to the MINMAX method [14]; and when *W* is quite close to zero, it is very similar to the lexicographic approach [8]. We set *W* to 0.35 to achieve an overall quality measure between these extremes. This typically gives a very similar overall quality score to the percentile method described at the 10% level. The dashed line in Fig 5 shows the calculated overall quality score (the weighted mean) for that particular sequence, with a value of 3.53. For comparison, the percentile method gives a very similar value of 3.55, while the mean gives a value of 4.08.

6 THE SIMULATION

The simulation takes a sequence of videos randomly selected from the 30 available, timed to start with a frequency distribution based on Fig 2. Videos are added based on this frequency distribution until a specified concurrency is reached. That is, if the videos were played out based on their real-time requirements (as CBR videos), there would be some point during the day where there would be an overlap in play out that matched the specified concurrency. Each video event is given a maximum DSL line rate that it can use, randomly chosen between one and two Mbits/sec. In all cases the line rate chosen would be able to sustain real-time play out of all the video sequences used (at some, but not necessarily the highest quality).

A simulation run takes the video schedule, and performs the following operations:

- For each new or existing video playing:
  - determine for each quality level, the transmission rate needed, knowing how much data is already buffered at the client
  - determine whether the quality level is feasible given the DSL line rate
- Find the ‘equitable’ quality level
  - the highest that can be transmitted over the backhaul
- Share out any spare backhaul bandwidth.
- ‘Stream’ the videos for the next GOP period and recalculate amount buffered
- Repeat until all videos have been delivered

7 RESULTS

The results here compare the EQVS with standard CBR coding and delivery at 1MBit/s where the DSL network is a constraining factor. The results are split into three main areas. Comparing EQVS and CBR delivery where neither methods are constrained by the network. Comparing the two where there are constraints, and finally looking in more detail on the impact of the DSL network in a loaded network.

7.1 No session rejection by CBR admission control

Video sessions were generated for each day simulated as described in section 3 until the new session would have been rejected by admission control if CBR coding and delivery were used. We then compare the overall quality we can achieve with equitable quality video streaming compared to CBR, for identical session initiation patterns. We simulated 5 days of video delivery, and present average results when the backhaul bandwidth is set to 6 Mbit/s.

On average, the CBR case could support 25 video sessions per day, with the day profile used. This may sound like a low number but it is purely down to the frequency distribution of the day profile which meant that over 50% of the videos started playing in the four hour period between 7 and 10pm.

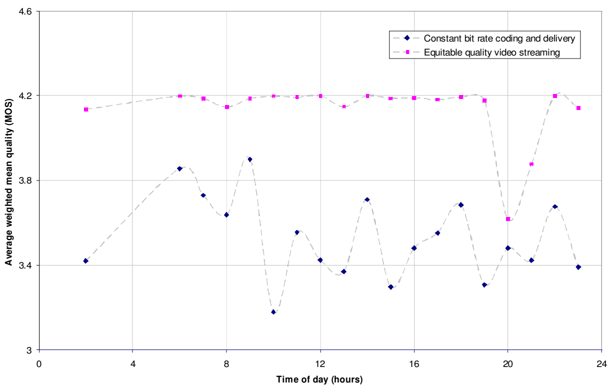

Fig. 6. Average weighted mean quality versus time of day for equitable quality video streaming and CBR coding and delivery. These results are the average of five simulation runs.

Fig. 6 shows how the average of the weighted mean quality scores varies with the time at which a session started, for both CBR and equitable quality video streaming. With CBR, the quality is independent of the time of day, except for variations due to the random selection of content. Equitable quality video streaming achieves high quality during off-peak times of the day, and slightly reduced quality at peak times; but at all times it achieves significantly higher average overall quality than CBR coding and delivery.

Table 1 shows the average overall quality for equitable quality video streaming and CBR, for both the whole day and peak times (where sessions would reach maximum concurrency – in this case six concurrent sessions), for each of the quality measures.

Table 1.
Average overall quality for equitable quality video streaming and CBR

		EQVS Quality	CBR Quality
Weighted Mean MOS	All Day	4.13	3.53
	Peak Time	4.08	3.52

7.2 Session rejection by CBR admission control

A fixed number of video sessions were generated for each day simulated as described in section 3. In the case of CBR coding and delivery, some sessions may be rejected by the network admission control. Fig. 7 shows how the average overall quality achieved with equitable quality video streaming varies as we increase the number of concurrent sessions. Note that the average overall quality achieved with CBR is independent of the number of sessions, being equal to 3.40. In the figure the quality does vary around the 3.4 mark and is due to the collection of content selected for that particular set of simulation runs. From the network configuration with a backhaul of 6Mbits/sec, a total of six concurrent sessions can be supported with CBR, and after that point it would lead to session rejection. In Fig 7 below we show the CBR session qualities above a concurrency of 6 as a dashed line to emphasize that session rejection would have taken place. We can also see that between 12 and 13 concurrent sessions there is a crossover of the EQVS plot with the CBR plot, indicating that we can deliver over twice as many concurrent video sessions with the same overall quality that can be delivered with CBR.

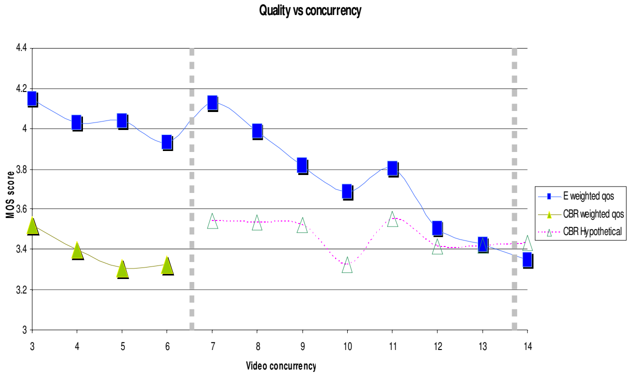

Fig. 7. Overall quality for equitable quality video streaming and CBR coding and delivery plotted against concurrency.

7.3 The impact of DSL constraints.

It was important to understand the impact on the delivery and quality of limited bandwidth DSL connections. That is, DSL line rates that are significantly lower than the overall backhaul network bandwidth. Earlier experiments [4] had been performed with networks that assumed no constraints on the DSL part of the network.

Fig 8 shows the percentage time at each quality level as we increase the concurrency. Each band in the graph shows the amount of time spent at a particular quality level. At low concurrency the majority of the time is spent delivering the videos at the highest quality level, as one would expect. As the concurrency increases, more and more time is spent delivering at the lower qualities.

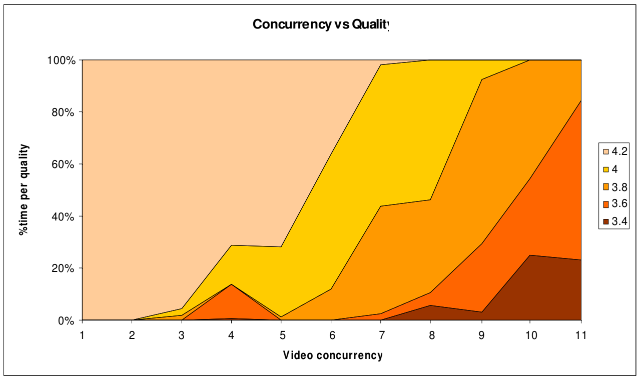

Fig 8. The effect of video concurrency on the quality of videos delivered with no DSL limitations.

Fig 8 shows the case where there is no limitation on DSL rates, that is the DSL rate was set to at least 6Mbits/sec, so the DSL portion of the network could accept traffic from a fully loaded backhaul network.

Fig 9 below, shows the same scenario (with the same video data set), but with each DSL rate limited to between one and two Mbits/sec. At low concurrencies, from one to four, we can see that some time is spent delivering videos at lower than the best quality, purely because of the bandwidth demands for delivering at the highest qualities are capped by the DSL rates. However, as we move to higher concurrencies, the Fig 8 above looks very similar in shape and proportion to Fig 9 below.

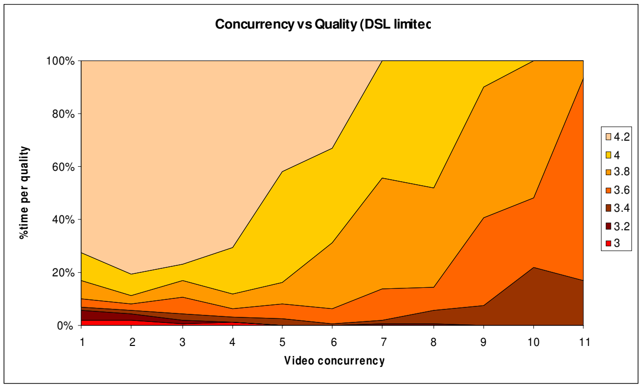

Fig 9. The effect of video concurrency on the quality delivered with DSL constraints.

We can safely say that DSL capping is not an important factor when we reach saturation in the backhaul network. This is also borne out in Fig 10 that shows the relative utilisation of the backhaul network and DSL network as the video concurrency is increased. Up to a concurrency of four, the DSL network is fully utilised, from a concurrency of six and above the backhaul network is fully utilised.

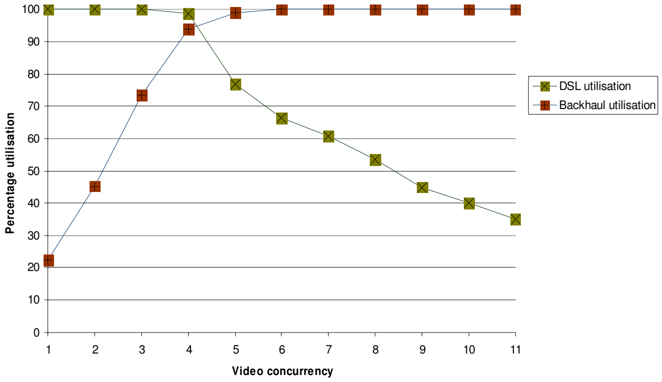

Fig 10. Network utilisation in the backhaul and DSL networks as we increase video concurrency.

8 CONCLUSION

CBR encoding and transmission are easily implemented and network provisioning, although requiring admission control, is straightforward. But the result is variable video quality and session refusal. We have introduced the concept of equitable quality video streaming, where the network bit rate available for video streaming is divided between the concurrent users such that they receive equal video quality subject to their local DSL constraints.

We have shown that equitable quality video streaming provides significantly better quality to the end user, improving the quality (MOS) by 0.60 overall and by 0.56 at peak times. We have also shown that DSL constraints do not have a significant effect with EQVS streaming where the ratio of backhaul network to DSL line rate is of the order of 4.5:1.

We have also shown that the number of sessions that can be delivered at the same overall perceptual quality as CBR can be increased by 100%.

ACKNOWLEDGMENT

The authors thank Andy Davis, Damien Bayart and David Hands for the use of their technology for measuring the perceptual video quality of a compressed video bitstream.

REFERENCES

[1] "About BT Group: Video technology and IPTV solutions", online at: http://www.btplc.com/Thegroup/Technologylicensing/Multimedia.htm.

[2] "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4) AVC, 2003.

[3] G. L. Anderson and A. N. Netravali, "Image Restoration Based on a Subjective Criterion," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-6, No. 12, December 1976.

[4] B. Crabtree, M. Nilsson, P. Mulroy and S. Appleby "Equitable quality video streaming" Accepted at Computer Communications and Networking Conference, Las Vegas, January 2009.

[5] I. Dalgic and F. A. Tobagi, "Characterization of quality and traffic. for various video encoding schemes and various encoder control schemes," Stanford CSL-TR-96-701, August 1996.

[6] A.G. Davis, "Video Quality Measurement," International patent application, WO2007066066, June 2007.

[7] D. S. Hands, "Subjective responses to constant and variable quality video," in Proc. SPIE Visual Communications Image Processing 2008.

[8] D. T. Hoang, E. L. Linzer, and J. S. Vitter, "Lexicographic bit allocation for MPEG video," Journal of Visual Communication and Image Representation, vol. 8, no. 4, December 1997.

[9] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based on models of human perception," Proc. IEEE, October 1993.

[10] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR Video: Tradeoffs and potentials," in Proc. IEEE, May 1998, vol. 86, pp. 952-973.

[11] "Methodology for the subjective assessment of the quality of television pictures," ITU-R Recommendation BT.500-11, 2002.

[12] M. E. Nilsson, P. Key, and R. Macfadyen, "VBR video, policing and dimensioning," Proc. International Workshop on Audio-Visual Services over Packet Networks, Aberdeen, September 1997.

[13] P.A. Pellegrini, Listen without Prejudice, VUE (June 2005).

[14] G.M. Schuster and A.K. Katsaggelos, "The minimum average and minimum-maximum criteria in lossy compression," Vistas in Astronomy, vol. 41, no. 3, pp. 427-437, 1997.

[15] Television 2005. International Key Facts, 12th Edition, October 2005, IP International Marketing Committee (CMI), Copyright IP & RTL Group.

[16] P. H. Westerink, R. Rajagopalan, and C. A. Gonzales, "Two-pass MPEG-2 variable-bit-rate encoding," IBM Journal of Research and Development, Vol. 43, No. 4, pp. 471-488, July 1999