A Performance Study of Adaptive Video Coding Algorithms for High Speed Networks

Sanjeev Gupta

Pacific Bell
2600 Camino Ramon
San Ramon, CA, USA 94583
sanjeev@cs.usask.ca

Carey L. Williamson

Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada S7N 5A9
carey@cs.usask.ca

Abstract

Adaptive video coding algorithms are digital video compression algorithms that can adapt the encoding of a video stream dynamically based on the amount of bandwidth available on the network. While such algorithms are more complicated than traditional video coding algorithms, they are attractive because of their inherent robustness to changes in network load (i.e., network congestion). Adaptive video coding algorithms seem particularly suitable for high speed network environments, such as B-ISDN/ATM, that offer Available Bit Rate (ABR) services.

The goal of this paper is to assess the role that adaptive video coding algorithms will play in future high speed networks. The paper presents a simple mathematical model and analysis of several hypothetical video coding algorithms for high speed networks, and a simulation study of one such adaptive video coding algorithm that we have implemented in a local area network environment.

The results show that adaptive video coding algorithms are indeed robust across a wide range of network loads. More importantly, however, the results suggest that the domain of adaptive video coding algorithms is quite narrow: moderately to heavily loaded networks with speeds on the order of 10 Mbps and 100 Mbps. As a result, adaptive video coding algorithms will likely play only a limited role in future high speed networks.

1 Introduction

Networks such as the Broadband Integrated Services Digital Network (B-ISDN) provide the raw network technology required for a wide range of video applications and multimedia services [2, 6, 11]. The last ten years have witnessed the proliferation of these applications and services, and the next ten years promise even more such applications.

Video compression algorithms (or video coding algorithms) are key to the success of many of these applications, since the video component in a multimedia application typically has (by far) the largest bandwidth requirement. For example, an NTSC-quality uncompressed video stream, with 640×480 pixels per frame, 8 bits of colour information per pixel, and a display rate of 30 frames per second, requires over 70 Mbps of network bandwidth.

Video coding algorithms can reduce the amount of bandwidth required by video applications by an order of magnitude or more, depending on the coding algorithm. The video stream is compressed by detecting and encoding the patterns and redundancies that occur both within single frames (*intraframe coding*), and between successive frames in a video sequence (*interframe coding*). Some video coding algorithms are *lossless*: the reconstructed images are exact pixel-by-pixel replicas of the original images. Some video coding algorithms are *lossy*: the reconstructed images are only approximations of the original images. Lossy algorithms usually offer better compression than lossless algorithms.

Video coding algorithms can also be classified as Constant Bit Rate (CBR) or Variable Bit Rate (VBR) algorithms. In their simplest form, CBR algorithms use the same number of bits to represent each frame in the video sequence, so that the compressed video stream is as smooth as possible, with no burstiness. The result is a constant bit rate stream with variable quality video (e.g., the quality may degrade during scene changes or during video sequences with lots of fast motion). VBR algorithms, on the other hand, provide constant quality video, by using as many bits as needed to represent each video frame. Since different frames in a video sequence inherently compress by different amounts, a variable bit rate compressed video stream results.

Of particular interest in this paper are *adaptive* VBR video coding algorithms. Adaptive video coding algorithms are able to adapt the encoding of a video stream (and thus its bandwidth requirements) dynamically based on the amount of bandwidth available in the network. For example, when the network is congested, the algorithm compresses the video images as much as possible to use the lowest possible bit rate; when extra capacity is available in the network, the algorithm spends less effort on compressing images, or may even transmit images uncompressed.

The choice of a video coding algorithm depends somewhat on the services offered by the underlying network. For example, if the network only provides performance guarantees (or billing) using peak bandwidth allocation, then either Constant Bit Rate or simple Variable Bit Rate coding algorithms may suffice. On the other hand, if the network offers a service class for Available Bit Rate (ABR) traffic[1], then adaptive video coding algorithms are attractive. Rather than reserve bandwidth based on the peak bit rate of the video stream, which may be inefficient, or the mean bit rate, which may be inadequate, adaptive video coding algorithms can dynamically adjust the encoding of a video stream, and thus its bandwidth requirements, to match the available capacity in the network.

Research on adaptive video coding algorithms is now beginning to appear in the literature [4, 7]. For example, Garrett *et al.* [4] developed a video coder that changed its encoding scheme to maximize the quality of the received image based on feedback about expected loss rates from the network. Kanakia *et al.* [7] studied an adaptive congestion control mechanism designed to interoperate with VBR video traffic. Although such research efforts show the potential of adaptive video coding, there has not been much work in identifying the general conditions (e.g., network speeds, load characteristics) under which adaptive video coding algorithms are preferable to non-adaptive video coding algorithms. It is this latter issue that this paper addresses. In other words, the goal of this paper is to assess the role that adaptive video coding algorithms will play in future high speed networks.

The remainder of the paper is organized as follows. Section 2 provides more background on adaptive video coding: its definition, its motivation, and how it might be done. Section 3 presents a simple mathematical model of adaptive video coding, and an analysis of six hypothetical adaptive video coding algorithms. Section 4 presents a simulation study of one adaptive video coding algorithm that has been implemented and tested in a local area network environment. Section 5 summarizes the results from our analyses, and Section 6 presents the conclusions from this research.

2 Adaptive Video Coding

2.1 Definition

Adaptivity of a video coding algorithm is defined as the ability of the algorithm to adapt the encoding of a video stream (and thus its bandwidth requirements) dynamically based on the amount of bandwidth available in the network. Regardless of the network load, the video coding algorithm should be able to compress the image such that a perceptually satisfying image is received by the end-user in a reasonable amount of time.

[1] Also called Free Bit Rate [9] in the literature. In the traffic classes proposed by the ATM Forum, ABR traffic is guaranteed only a certain minimum bit rate, but is allowed to make use of any additional bandwidth that is available on the network.

2.2 Assumptions

While studying the need for adaptive video coding algorithms it is essential to understand that in networks with bandwidth reservation for guaranteed QOS (Quality of Service) there is no need for adaptive video coding algorithms. If peak bit rate allocation is always used, then the network can always guarantee the QOS, regardless of the network load. However, peak bandwidth allocation is often inefficient, especially for VBR traffic. Furthermore, users of high speed networks would rather pay for the bandwidth that they actually *use* than the bandwidth that they reserve. Thus one of the basic assumptions while studying adaptive video coding algorithms is that the network does not provide simple peak bandwidth reservation for guaranteed QOS. Instead, the network offers Available Bit Rate (ABR) services.

A second assumption critical to the analyses in this paper is that adaptive video coding can be done quickly (say, 10 milliseconds or less for the encoding and decoding of each frame). Encoding and decoding times much larger than this would clearly make the approach difficult or impossible to use with either real-time or stored video applications. Whether adaptive video coding can be done entirely in software or not is unclear, though we completely ignore this issue in this paper.

2.3 How to Adapt?

There are several ways in which video coding algorithms can adapt to changes in the network load. For example:

- **Reduce Colour Information**: The video coding algorithm can reduce the number of bits used to represent the colour information of each pixel when the network load increases. For example, if the colour information is dropped from 24 bits (full colour) to 8 bits (grayscale) for an image, then a bandwidth saving of a factor of 3 is obtained, though the decoded image will be less perceptually satisfying than the full colour image.

- **Reduce Image Size**: The video coding algorithm can reduce the size of the image when the network load increases. For example, an image of 640×480 pixels could be reduced to 320×240 pixels, obtaining a bandwidth saving of a factor of 4. Again, the decoded image will be less perceptually satisfying than the full size image.

- **Reduce Frame Rate**: The display rate of a video stream could be reduced under conditions of high network load. For example, if the frame rate of a video stream was reduced from 30 frames per second to 15 frames per second, then a bandwidth saving of a factor of 2 is obtained. The received image sequence may be jerky and lack continuity.

- **Change Coding Algorithms**: The video encoder could switch between video coding algorithms that offer low (or no) compression, moderate compression, and high compression as the network load changes. While this approach can trade off bandwidth and image quality like the above approaches, it makes decoding the video stream much more complicated for the receiver.

- **Change Algorithm Parameters**: Within a specific video coding algorithm, the parameters of the video coding algorithm (e.g., quantization) could be changed depending on the network load to provide higher or lower compression. Each setting of the parameter results in a different compression ratio for the algorithm, and may change the image quality. Thus, when the network is lightly loaded, the parameters of the video coding algorithm could be set to provide low compression ratios and a good quality image. When the network load increases, the parameter setting could be changed to provide greater compression (and lower image quality).

This paper focuses on the latter approach: changing algorithm parameters. However, this approach could certainly be combined with one or more of the other approaches to improve the overall adaptivity of a video coding algorithm.

2.4 When to Adapt?

Adaptive video coding algorithms can adapt on either an intraframe (i.e., several times per individual frame transmitted) or interframe (i.e., once every N frames, where $N \geq 1$) basis. This paper only considers interframe adaptivity[2] with $N = 1$. For NTSC standard video (30 frames per second), adapting on a per-frame basis means adapting the video coding parameters every 33 milliseconds based on load changes or feedback from the network.

3 Mathematical Model and Analysis

This section presents a simple mathematical model for adaptive video coding, several metrics for the performance evaluation of adaptive video coding algorithms, and a brief analysis of the domain of adaptive video coding algorithms.

3.1 A Simple Mathematical Model

The time T it takes for the end-user to receive a single image in a video sequence depends on the compression time T_c for the image, the transmission time T_t of the compressed image, and the decompression time T_d. The transmission time is the time from when the sender starts transmitting the first bit of the compressed image to the time the last bit is received by the receiver. By definition, this time includes any retransmission, queueing and channel access time on the network. Thus,

$$T = T_c + T_t + T_d \tag{1}$$

The compression time T_c depends on the compression algorithm used, the size of the uncompressed image S_u, the contents of the uncompressed image, and the speed of the machine used. For the sake of simplicity, T_c (a function of these four variables) is represented by a fixed value F_c depending only on the compression algorithm. That is, the uncompressed image size S_u is assumed to be constant, and the effect of image contents and machine speed are ignored.

The decompression time T_d depends on the compression algorithm used, the size of the compressed image S_c and the speed of the machine used. For the sake of simplicity, T_d (a multi-variable function) is represented by a value F_d depending only on the compression algorithm used. Image size, image contents and machine speed are ignored.

The expected transmission time T_t depends on three main factors: the transmission capacity of the network (B bits per second), the average utilization ρ of the network ($0 \leq \rho \leq 1$), and the size S_c of the image to be transmitted. Thus,

$$T_t = \frac{S_c}{(1-\rho)B} \tag{2}$$

For example, a compressed image of 20 Kilobytes transmitted over a 10 Mbps network link that has a network utilization of 0.3 requires

$$T_t = \frac{20 \times 1024 \text{ bytes} \times 8 \text{ bits/byte}}{(1-0.3) \times 10 \times 10^6 \text{ bits/sec}} = 0.0234 \text{ sec}$$

The size of the compressed image depends on the size of the original image S_u, and the compression ratio C_r offered by the compression algorithm, where the compression ratio is the ratio of the size of the original image to the size of the compressed image. Mathematically,

$$S_c = \frac{S_u}{C_r} \tag{3}$$

Therefore Equation 1 becomes

$$T = \frac{S_u}{(1-\rho)BC_r} + F_c + F_d \tag{4}$$

For example, consider a video compression algorithm with $F_c = 0.010$ and $F_d = 0.005$, compressing an image with 128×128 pixels and 8 bits of colour information per pixel, with a compression ratio of 10:1. If the image is transmitted on a 10 Mbps Ethernet LAN with a utilization of 0.3, the predicted end-to-end transfer time is

$$T = \frac{128 \times 128 \text{ pixels} \times 8 \text{ bits/pixel}}{(1-0.3) \times 10 \times 10^6 \text{ bits/sec} \times 10} + 0.010 + 0.005$$
$$= 0.0169 \text{ sec}$$

Since compression time and decompression time are assumed to be fixed (for a given algorithm), the transmission time T_t is the primary performance metric for adaptive video coding algorithms. Compression ratio and image quality are closely related, but secondary, performance metrics.

[2] Experiments reported in [5] have found that the additional complexity and overhead of intraframe adaptivity make it of very limited benefit, and that interframe adaptivity for $N > 1$ is very sluggish to respond to changes in network load.

3.2 Six Hypothetical Video Coding Algorithms

This section introduces six hypothetical video coding algorithms as possible candidates for adaptive video coding. The algorithms are characterized by their *compression profile*: the way in which the compression ratio of the algorithm changes as its parameters are varied. Each hypothetical algorithm is assumed to have a single generic parameter, which can range from a minimum setting of 0 (offering the least compression) to a maximum setting of 100 (offering the most compression).

As seen in Figure 1, video coding algorithms with several different compression profiles were considered. The compression profiles of the hypothetical cases of video coding algorithms were obtained by fitting values to some arbitrarily chosen curves.

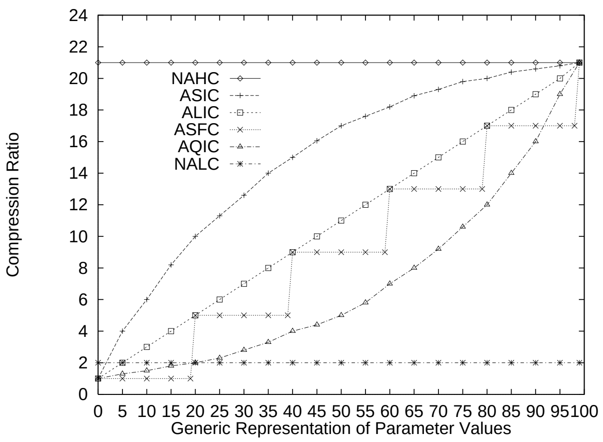

Figure 1: Compression Profiles of Hypothetical Video Coding Algorithms

The hypothetical video coding algorithms under consideration are:

- Non-Adaptive Low Compression (NALC): This represents a non-adaptive video coding algorithm with a low compression ratio. The compression profile is flat (horizontal). Studying an NALC video coding algorithm is useful in order to show that even though non-adaptive video coding algorithms with low compression ratios provide images of very high quality, adaptive video coding algorithms may be more robust than non-adaptive video coding algorithms.

- Non-Adaptive High Compression (NAHC): This represents a non-adaptive video coding algorithm with a high compression ratio. The compression profile is again flat, but at a much higher value than the NALC algorithm. There is a need to study NAHC algorithms in order to evaluate if adaptive video coding algorithms provide any performance gains over non-adaptive video coding algorithm with a high compression ratio.

- Adaptive Linearly Improving Compression (ALIC): This represents an adaptive video compression algorithm with a linear increase in the compression ratio as its parameters are changed. Thus, as the network load increases, the video coding algorithm can adapt to provide higher compression ratios with a deterioration in the image quality. An example of an ALIC algorithm would be the linear reduction in frame rate for a video sequence as the network load increases. It is interesting to study a video coding algorithm with an ALIC profile in order to see if it outperforms a non-adaptive video coding algorithm operating at the maximum and minimum compression ratios of the ALIC profile and if video coding algorithms with ALIC profiles are suitable for adaptivity.

- Adaptive Step Function Compression (ASFC): This represents a video coding algorithm with a small number of discrete levels in its compression ratios. The algorithm has a step function for its compression profile. An example of an ASFC algorithm would be the reduction of a colour image to a black and white image or the reduction in the size of the image (through a progression of predetermined sizes) as the parameter settings are changed. A study of a video coding algorithm with an ASFC profile would help in determining if a video coding algorithm with very few levels can be considered for adaptivity. The study will also show how a video coding algorithm with an ASFC profile compares to a video coding algorithm with an ALIC profile having the same average slope (i.e., the same compression values at the endpoints in the compression profile, where the parameter settings are at their lowest and highest values).

- Adaptive Quadratically Improving Compression (AQIC): This represents an adaptive video compression algorithm with a quadratic increase in its compression ratio as its parameters are changed. Compression improves slowly at first, but then improves faster and faster as the maximum parameter setting is approached. Even though a video coding algorithm with an AQIC profile provides lower compression ratios than a video coding algorithm with an ALIC profile (with the same average slope), it would be interesting to see the relative performance in terms of image quality and transmission time.

- Adaptive Sublinearly Improving Compression (ASIC): This represents an adaptive video compression algorithm with a logarithmic increase in its compression ratio as its parameters are changed. Compression improves quickly at first, but then improves less and less (sublinearly) as the maximum parameter setting is approached. While no specific example of a video coding algorithm with an ASIC profile is known, the ASIC profile is useful as a hypothetical case. The ASIC profile provides higher compression ratios than the ALIC profile over most of the range, even though both have the same average slope. Again, an algorithm with an ASIC profile is of interest because it will help in determining the characteristics that an adaptive video coding algorithm should have.

3.3 Evaluation of Hypothetical Video Coding Algorithms

This section presents a study of the six hypothetical video coding algorithms introduced in the previous section, using the performance metrics defined in Section 3.1. The purpose of the study is to identify the desirable characteristics of adaptive video coding algorithms.

Several assumptions were made while studying the hypothetical video coding algorithms. These assumption were as follows:

- A video sequence with images of size 640 × 480 pixels and 24 bits of colour information per pixel.
- A 100 Mbps network.
- A 1-1 direct mapping between the generic parameter space of the video coding algorithm and the network load (expressed as a percentage). What this means is that at a network load of 10% the parameter setting for the video coding algorithm is 10, and so on.

Figure 2 shows the transmission time curves for the hypothetical video coding algorithms. The values for the transmission time are calculated using the formula for transmission time given in Equation 2, and the compression profiles shown in Figure 1.

The NALC and NAHC algorithms represent non-adaptive video coding algorithms. From Figure 2, it is seen that under low load conditions, the transmission time for the NALC algorithm is comparable to that of other algorithms, but as the load increases beyond 20%, the transmission time for the NALC algorithm is much higher than the other algorithms. The transmission time increases for NALC since the compressed image size is always the same, regardless of the network load. The NAHC algorithm has the lowest transmission time across the full range of network loads, since it offers the highest compression.

The four adaptive video coding algorithms have similar qualitative behaviour. At first, there is a general decrease in transmission time as network load increases from low to moderate, since the level of compression increases, and the value of S_c in the numerator of Equation 2 decreases. Then there is an increase in transmission time as the load increases from moderate to high, since increases in the network load, which affect the denominator of Equation 2, have greater effect than further reductions in S_c. The ASFC algorithm has the highest transmission times for load levels of 0%-20% because its compression ratio at these loads is lower than the compression ratio of the other algorithms at these loads. For network loads greater than 20%, the ASFC, ALIC and ASIC algorithms have

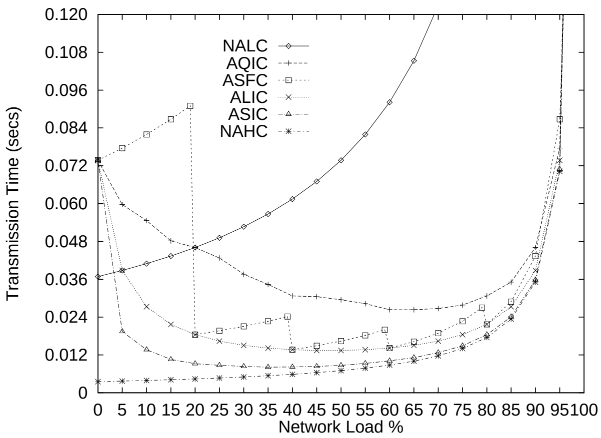

Figure 2: Transmission Time Curves for Hypothetical Video Coding Algorithms

similar values for the transmission times because the compression ratios offered by these algorithms are very similar. The transmission time for AQIC is slightly higher because of its lower compression in this range.

Overall, the adaptive video coding algorithms are more robust than the non-adaptive video coding algorithms. The adaptive algorithms are able to maintain a reasonably bounded image transmission time over a fairly wide range of network loads. The transmission times for all the algorithms begin to increase very rapidly at loads of 80% or greater. Even though the images are highly compressed, it takes a long time for the sender to get the bandwidth required to transmit the compressed images.

3.4 Domain of Adaptive Video Coding Algorithms

Having identified the general characteristics for a video coding algorithm to be adaptive, it is necessary to determine the network speeds on which adaptive video coding algorithms are feasible (i.e., the domain of adaptive video coding algorithms). The following network speeds are considered:

- 100 Kbps: This link speed is representative of some of today's telecommunication links (64 Kbps) and Internet links.
- 1 Mbps: Today's T1 (1.544 Mbps) links, as used in the current CA*net backbone, are examples of networks operating at about 1 Mbps.
- 10 Mbps: An example of a network operating at 10 Mbps is the Ethernet LAN.
- 100 Mbps: These are representative of FDDI networks (100 Mbps) and OC-3 links (155 Mbps) for B-ISDN networks.
- 1 Gbps: These are representative of the gigabit networks of the future.

With an NTSC standard image of 640 × 480 pixels with 24 bits of colour information per pixel, the

Table 1: Transmission Time in Seconds for Non-Adaptive Video Coding (NALC)

Utilization	100 Kbps	1 Mbps	10 Mbps	100 Mbps	1 Gbps
0.0	72.000	7.200	0.720	0.072	0.007
0.1	80.000	8.000	0.800	0.080	0.008
0.2	90.000	9.000	0.900	0.090	0.009
0.3	102.857	10.285	1.028	0.102	0.010
0.4	120.000	12.000	1.200	0.120	0.012
0.5	144.000	14.400	1.440	0.144	0.014
0.6	180.000	18.000	1.800	0.180	0.018
0.7	240.000	24.000	2.400	0.240	0.024
0.8	360.000	36.000	3.600	0.360	0.036
0.9	720.000	72.000	7.200	0.720	0.072

Table 2: Transmission Time in Seconds for Adaptive Video Coding (ALIC)

Utilization	100 Kbps	1 Mbps	10 Mbps	100 Mbps	1 Gbps
0.0	72.000	7.200	0.720	0.072	0.007
0.1	20.000	2.000	0.200	0.020	0.002
0.2	15.000	1.500	0.150	0.015	0.001
0.3	12.857	1.285	0.128	0.012	0.001
0.4	12.000	1.200	0.120	0.012	0.001
0.5	12.000	1.200	0.120	0.012	0.001
0.6	12.857	1.285	0.128	0.012	0.001
0.7	15.000	1.500	0.150	0.015	0.001
0.8	20.000	2.000	0.200	0.020	0.002
0.9	36.000	3.600	0.360	0.036	0.003

five networks were evaluated mathematically for the NALC and ALIC compression profiles from Figure 1. The transmission times for the NALC and ALIC compression profiles, obtained using Equation 2, are shown in Tables 1 and 2, respectively. The first column in the table denotes the utilization of the network at the time the image is compressed.

Tables 1 and 2 help to illustrate which network speeds can support video applications. Low speed networks such as the 100 Kbps and 1 Mbps networks do not possess the raw bandwidth needed to support video applications. For example, even at a network utilization of 0.5, an image (as described previously) compressed using the ALIC compression profile requires 1200 msecs to be transmitted over a 1 Mbps network. On the other hand, gigabit networks can provide the bandwidth requirements of video traffic even without compression (see, for example, the rightmost column of Table 1).

With the compression scenario under consideration, 10 Mbps networks cannot support real-time video. However, 10 Mbps networks can support still image transfer applications, or slow scan video with a frame rate of 5-10 frames per second.

4 Experimental Model and Results

This section presents further experiments that were conducted as part of this research. The goals of the experiments were to validate the mathematical model discussed in Section 3, and to evaluate several existing video coding algorithms with respect to their adaptivity. Only one algorithm (JPEG) is discussed in this paper. The full results for four different video coding algorithms (JPEG, EPIC, Fractal, and MPEG) are available elsewhere [5].

4.1 Experimental Setup

A simulation model was built to test the adaptivity of several video coding algorithms. The simulator, shown in Figure 3, is very simple with a sender, which compresses the image and transmits it over the network, and a receiver, which decompresses and displays the compressed image received. The network is loaded with cross-traffic from a sender-receiver pair.

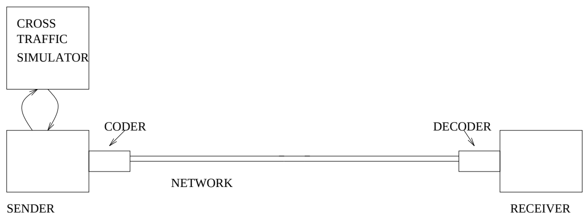

Figure 3: The Simulation Model

Simple assumptions were made in constructing the simulator. First, the cross traffic is modeled using a Poisson process. Second, the sender can monitor the cross traffic model at all times, thus knowing of any changes in the network load as soon as the changes occur. Third, once the image is compressed and the transmission started, even if the load changes, the transmission of the current image is completed.

Figure 4: The Test Image

Network load characteristics are generated using the simulator, but compression profiles, compression times, decompression times, and image quality measures were determined from software implementations of the algorithms. All algorithms were implemented in C, and run on a DEC Alpha in an Ethernet local area network environment.

The experiments are carried out on a non-changing image sequence with each image corresponding to the type of image seen in video teleconferencing applications. The image is a grayscale image with 128×128 pixels and 8 bits of information per pixel. The sample image is shown in Figure 4.

4.2 Experimental Factors

The factors considered in the simulation experiment are the following:

- Video coding algorithms: Four algorithms were chosen for the experimental study: Joint Photographic Experts Group algorithm (JPEG) [10], Efficient Pyramid Image Coding (EPIC) [1], fractal [3] and Motion Picture Experts Group (MPEG) [8]. Only the JPEG results are presented in this paper. Complete results are available in [5].

- Network: Two network speeds have been identified for the experiments: 10 Mbps and 100 Mbps. Simulations with these network speeds will indicate the feasibility of adaptive video coding algorithms on these networks and validate the simple mathematical evaluation in Section 3.

- Cross traffic scenarios: Four traffic scenarios are considered:

  - Scenario GI: there is a Gradual Increase in the network load. The network load increases in small steps, in a linear fashion.

  - Scenario GD: there is a Gradual Decrease in the network load. The network load decreases linearly in small steps.

  - Scenario RI: there is a Rapid Increase in the network load from a state of low load to being almost fully loaded.

  - Scenario RD: there is a Rapid Decrease in the network load. The network goes from a state of being almost fully loaded to a state of being very lightly loaded.

  While not particularly realistic, these four cross traffic scenarios provide useful benchmarks (and stress tests) for evaluating the adaptivity of video coding algorithms.

4.3 Performance Metrics

The performance metrics used in the experiments are the following:

- Compression Ratio: Compression ratio is defined as the ratio of the size of the original image (in bytes) to the size of the compressed image. The compression ratio indicates the level of compression that the video coding algorithm offers. The higher the compression ratio, the better.

- Transmission Time: This is defined as the difference between the time the sender starts transmitting the first bit of the compressed image to the time the last bit is received by the receiver. The transmission time depends on the size of the compressed image, the network speed, the network load level and the cross traffic scenario.

- Image Quality: There are two means of evaluating the quality of an image: subjective and objective evaluation. Subjective evaluation requires a human judgement of the perceptual quality of the image. Objective measures give a quantitative evaluation of the quality of the reconstructed image by comparing it to the original image. In this research the *average error per pixel* is used as the measure of the image quality. The average error per pixel ($\overline{E}$) is computed by summing up the absolute differences in the intensities of all corresponding pixels in the original and reconstructed image and dividing the sum by the total number of pixels in the image. Therefore the higher the average error per pixel, the lower the image quality, and vice versa. Mathematically,

  $$\overline{E} = \frac{\sum_{i=0}^{N-1} \mid P_i - \hat{P}_i \mid}{N}$$

  where P_i and $\hat{P}_i$ are the grayscale values of pixel number i in the original and reconstructed image, respectively, and N is the total number of pixels in the image.

4.4 Experimental Results

Figure 5 shows the compression profile of the JPEG algorithm. The adaptive parameter in the JPEG algorithm is the quantization level Q, which controls the granularity of the regions produced by the algorithm. The quantization level of the JPEG algorithm was increased as the network load increased resulting in higher compression ratios. The compression ratios range from 1:1 to 28:1 and increase almost quadratically as the quantization parameter is changed. The compression ratio offered by the adaptive JPEG algorithm depends solely on the network load at the time of image compression. Therefore the four cross traffic scenarios all have the same compression ratio at any particular network load and the compression curves for the four cross traffic scenarios coincide.

Figure 6 shows the average error per pixel curve for the JPEG algorithm. The average error per pixel increases as the network load level increases because the image is being compressed more, and more information is lost. The average error per pixel curves for the four cross traffic scenarios coincide because the four cross traffic scenarios provide the same level of compression as each other at any network load level and images that are compressed the same have the same average error per pixel.

Figure 7 shows the transmission times for the adaptive JPEG algorithm on a 100 Mbps network. The transmission time is found to remain below the 33 milliseconds per frame delivery time required for real-time video over a wide range of network loads for three of the cross traffic scenarios (RD, GD, and GI). The variation in the transmission time is also reduced: the transmission time remains almost constant even though the network load increases because of the adaptivity of the algorithm (i.e., JPEG compresses the image more at higher load levels).

From Figure 7, it is observed that the transmission time for the RI traffic scenario is much higher than the other three cross traffic scenarios for almost all network loads. The reason for the transmission time being so high is that the image is compressed at a low load (low compression) and the very next instant the network becomes almost fully loaded. Thus, the sender is trying to transmit a large image in a heavily loaded network resulting in a large transmission time. For the GD and RD scenarios, the transmission time is very low and remains relatively constant for load levels of 0%-95% because the image is compressed at a high load (and therefore higher compression) and the transmission takes place at a lower load. For the GI scenario the transmission time is the same as the GD and RD scenarios and remains constant for load levels of 0%-75%. The transmission time for the GI scenario is higher than in the GD and RD scenarios for load levels greater than 75% because the JPEG algorithm provides the same level of compression in the GI, GD and RD scenarios , but the load increases during transmission in the GI scenario (as opposed to decreasing in the GD and RD scenarios).

5 Summary of Results

Six hypothetical video coding algorithms with compression profiles similar to existing algorithms were studied in order to determine the load conditions and network speeds on which adaptive video coding algorithms are feasible. Experiments were conducted

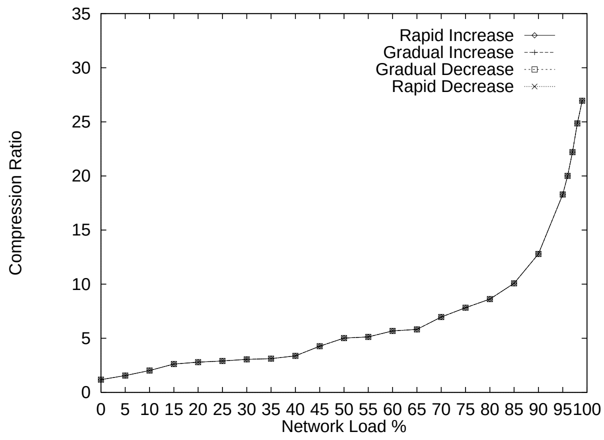

Figure 5: Compression Profile for Adaptive JPEG Video Coding Algorithm

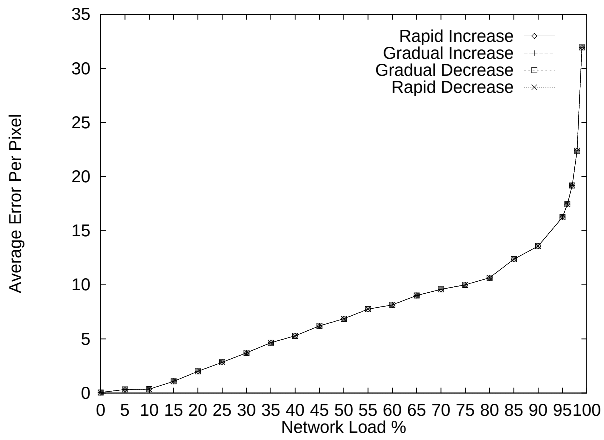

Figure 6: Error Curve for Adaptive JPEG Video Coding Algorithm

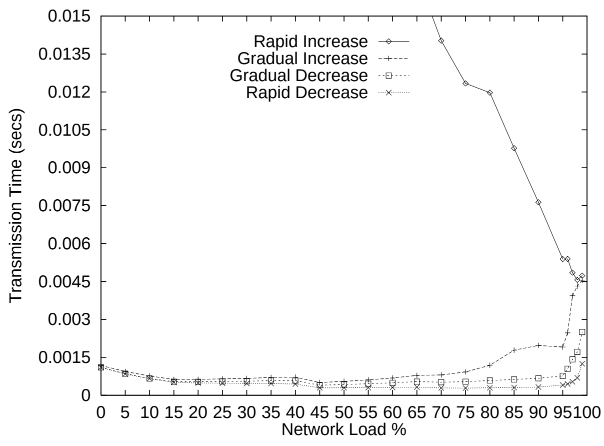

Figure 7: Transmission Time Curves for Adaptive JPEG Video Coding Algorithm on a 100 Mbps Network

with the JPEG algorithm to validate the mathematical model.

The study of adaptive video coding algorithms leads to the following observations:

- Adaptive video coding algorithms are robust across a wide range of network loads (20% to 80%) and can provide more consistent image transmission time than non-adaptive video coding algorithms by bounding the image transfer time. A non-adaptive video coding algorithm may provide lower transmission times than an adaptive version of the same algorithm (by compressing the image at the highest possible compression), but the image is often of lower quality.

- Under the compression profiles considered in this paper, adaptive video coding algorithms are advantageous at moderate to heavy levels of network load. At low network load (less than 20%) the adaptive video coding algorithms do not outperform non-adaptive video coding algorithms by enough to merit the complexity of adaptivity. For very high network loads (greater than 80%) the performance of both adaptive and non-adaptive video coding algorithms is very poor.

- With the compression scenarios under consideration, adaptive video coding algorithms are best suited for 10 Mbps and 100 Mbps networks. Networks of speed lower than 10 Mbps do not possess the raw bandwidth required to support video applications. Gigabit networks possess more than enough bandwidth to support video applications even without significant compression.

6 Conclusions

This paper studies adaptive video coding: the ability to adjust the encoding rate of a video stream (and therefore the bandwidth required) dynamically based on the amount of bandwidth available in the network. There are two main conclusions from this research.

First, adaptive video coding algorithms are advantageous whenever the network is a shared resource, with other users competing for the bandwidth. In this situation, adaptive video coding algorithms provide robustness by keeping the transmission times below certain limits while providing better picture quality than non-adaptive algorithms.

Second, with the compression profiles considered, the useful domain of adaptive video coding algorithms is fairly narrow. Networks of speeds 10 Mbps and 100 Mbps are targeted for adaptive video coding algorithms. Networks of speeds lower than 10 Mbps do not possess the raw bandwidth required for video applications, and gigabit networks can support video applications even without much compression. Also adaptive video coding algorithms are advantageous only at moderate to heavy levels of network load (20% to 80%). At very heavy and low network loads, the performance of adaptive video coding algorithms does not merit the additional complexity of adaptivity.

References

[1] E. Adelson and E. Simoncelli, "Subband image coding with three-tap pyramids", In *Picture Coding Symposium*, Cambridge, Massachusetts, pp. 1-3, September 1990.

[2] H. Armbruster and K. Wimmer, "Broadband multimedia applications using ATM networks: High performance computing, high-capacity storage, and high-speed communication", *IEEE Journal on Selected Areas in Communication*, Vol. 10, No. 9, pp. 1382-1396, December 1992.

[3] M. Barnsley and A. Sloan, "A better way to compress images", *Byte*, Vol. 13, No. 1, pp. 215-223, January 1988.

[4] M. Garrett and M. Vetterli, "Joint source/channel coding of statistical multiplexed real-time services on packet switched networks", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 1, pp. 71-80, February 1993.

[5] S. Gupta, "A performance study of adaptive video coding algorithms for high speed networks", M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, March 1995.

[6] T. Hoshi, K. Mori, Y. Takahashi, Y. Nakayama, and T. Ishizaki, "B-ISDN multimedia communication and collaboration platform using advanced video workstations to support cooperative work", *IEEE Journal on Selected Areas in Communication*, Vol. 10, No. 9, pp. 1403-1412, December 1992.

[7] H. Kanakia, P. Mishra and A. Reibman, "An adaptive congestion control scheme for real-time packet video transport", *Proceedings of the 1993 ACM SIGCOMM Conference on Communications Architectures, Protocols, and Applications*, pp. 20-31, September 1993.

[8] D. LeGall, "MPEG: a video compression standard for multimedia applications", *Communications of the ACM*, Vol. 34, No. 4, pp. 46-58, April 1991.

[9] T. Tanaka, H. Haseyama, K. Manabe, F. Kishino, "Bit-rate control algorithm on video packet coding and its characteristics", *Signal Processing: Image Communication*, Vol. 3, No. 4, pp. 321-331, September 1991.

[10] G. Wallace, "The JPEG still image picture compression standard", *Communications of the ACM*, Vol. 34, No. 4, pp. 31-44, April 1991.

[11] D. Wright, "Strategic impact of broadband telecommunications in insurance, publishing and health care", *IEEE Journal on Selected Areas in Communication*, Vol. 10, No. 9, pp. 1369-1381, December 1992.