COMPRESSION OF DIGITAL IMAGES OVER LOCAL AREA NETWORKS

by

Bhargavi Gorjala

A THESIS

Presented to the Faculty of

The Graduate College in the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor Khalid Sayood

Lincoln, Nebraska

August, 1991

COMPRESSION OF DIGITAL IMAGES OVER LOCAL AREA NETWORKS

Bhargavi Gorjala, M.S.

University of Nebraska, 1991

Adviser: Khalid Sayood

Differential Pulse Code Modulation (DPCM) has been used with speech for many years. It has not been as successful for images because of poor edge performance. The only corruption in DPCM is quantizer error but this corruption becomes quite large in the region of an edge because of the abrupt changes in the statistics of the signal.

We introduce two improved DPCM schemes; Edge correcting DPCM and Edge Preservation Differential Coding. These two coding schemes will detect the edges and take action to correct them. In an Edge Correcting scheme, the quantizer error for an edge is encoded using a recursive quantizer with entropy coding and sent to the receiver as side information. In an Edge Preserving scheme, when the quantizer input falls in the overload region the quantizer error is encoded and sent to the receiver repeatedly until the quantizer input falls in the inner levels. Therefore these coding schemes increase the bit rate in the region of an edge and require variable rate channels.

In this thesis we implement these two variable rate coding schemes on a token ring network. Timed token protocol supports two classes of messages; asynchronous and synchronous. The synchronous class provides a pre-allocated bandwidth and a guaranteed response time. The remaining bandwidth is dynamically allocated to the asynchronous class. The Edge Correcting DPCM is simulated by considering the edge information under the asynchronous class. For the simulation of the Edge Preserving scheme, the amount of information sent each time is fixed, but the length of the packet or the bit rate for that packet is chosen depending on the available capacity. The performance of the network, and the performance of the image coding algorithms, is studied.

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor, Dr. Khalid Sayood for his constant motivation and guidance throughout my complete Master's program. His patience, understanding and commitment to research has led to the completion of this work.

I am equally grateful to Dr. Gopal Meempat for useful discussions, invaluable suggestions, encouragement and for his help in this thesis.

I also wish to thank Dr. Don J. Nelson for consenting to serve on my reading committee. I am obliged to all those who directly or indirectly helped me.

I would like to thank the NASA Goddard Space Flight Center for supporting me during the course of my graduate studies under the grant NAG 5-916.

Finally, I dedicate this thesis to my parents whom I love with all my heart. Their patience and encouragement were the greatest.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

Digital communication is a rapidly expanding area of the communications field. Emerging communications and information networks require ubiquitous use of digital speech, digital audio and digital video. There are various advantages of digital communication over analog communication. To give a concise list of these advantages, digitized signals are less sensitive to transmission noise, easy to regenerate and store, to error-protect and encrypt, and to multiplex, packetize and mix. An added advantage of digital transmission is its potential for redundancy removal.

With an increase in the need to process, store and transmit huge amounts of data, data compression is becoming more of a necessity. Compression is even more important in the case of image signals, because of the large amount of data involved.

Speech and image signals are characterized as redundant waveforms. In predictive coding systems, waveform redundancy is utilized to realize reductions in bit rate. The Differential Pulse Code Modulation (DPCM) scheme, which is based on the notion of quantizing a prediction error signal, has been successfully used with speech for many years. However the results are quite disappointing when these coders are used with image data. This is because of the nature of the images. The presence of edges in an image causes the statistics of the signal to change rapidly in that region. These predictive coders are based on the statistics of the signal being fixed or slow changing. Therefore, the edges in an image cause the system to operate at low efficiency. The designer must accept either the loss of edge information, or a more complex and expensive system.

In this thesis, we study two improved Differential Pulse Code Modulation schemes, the edge correcting DPCM scheme developed by Sayood and Schekall [1]and the edge preserving coding scheme developed by Rost and Sayood[2]. These two coding schemes provide excellent edge preservation properties and still retain simplicity and efficiency. The only disadvantage is that these coding schemes require variable rate channels. This requirement makes it attractive to use these systems in a packet network environment.

The demand for integrated services local area networks has increased rapidly in recent years with the advent of new applications such as factory and automation. "Local Area" implies that the link is within a limited geographical area, normally within the same building. There is a clear need for local area networks in most organizations. Businesses are discovering that 80 percent of their communications requirements occurs across relatively short distances, such as within a group of offices or in a building.

The progress of LAN technology now makes it economical to introduce computer networks in a variety of different application fields. However, as new applications become interested in LANs, new requirements need to be met. Recently, the use of LAN in manufacturing communication and for integrated transmission of digital voice/video signals and data, has brought attention to the problem of using medium access protocols. These protocols allow a bounded access time for a class of messages, refered to as synchronous, subject to hard delivery deadlines, while retaining good throughput characteristics for the transmission of asynchronous traffic[3]-[4].

In Edge Correcting DPCM and Edge Preservation Coding schemes, edges are detected and a different action in each scheme is taken to correct them. Therefore, these schemes increase the bit rate in the region of an edge which makes them operate at variable rates. In this thesis, we implement these two coding schemes on a token ring network and send edge information only when the network finds bandwidth available for that information. This still retains the good throughput characteristics for the network.

Chapter two contains an overview of the Differential Pulse Code Modulation scheme. Later sections of this chapter discuss adaptive algorithms needed for the efficient prediction of a nonstationary process. Description of the adaptive ARMA predictor and a detailed analysis of the adaptive quantizer is also provided. Finally adaptive DPCM with an ARMA predictor is simulated and the results show that this scheme provides reasonably good performance for images with fewer edges, whereas for images with more edges the system needs to be improved.

In chapter three, the performance of a system is improved with an edge correcting DPCM. In this scheme the edges are detected by keeping track of the quantizer error. The edge performance is improved by sending extra information about the edges. This extra information is a coded version of the quantizer error. The quantizer error is coded by using a recursive quantizer with entropy coding. This scheme is simulated and the results are compared with the results in the second chapter. The performance is vastly improved with only little overhead.

In chapter four, a different approach, known as Edge Preserving Differential Image Coding Scheme is proposed to improve the edge performance. In this scheme reduction of the edge degradation is obtained by reducing or eliminating overload noise. When the quantizer input falls in the overload region, the quantizer error is again encoded and sent to the receiver repeatedly until the quantizer input falls in the inner levels. Finally, the simulation results are presented. The results show that this scheme even at low rates provides excellent edge preservation properties. There are absolutely no edge errors.

Chapter five gives a brief introduction to the token ring network and M/M/1 queuing system. In chapter six, we simulate the edge Correcting DPCM and the Edge Preserving schemes on a token ring network. The extra information in an Edge Preserving scheme is sent through the token ring network as asynchronous information only when the bandwidth becomes available, otherwise the extra information will be discarded. In the Edge Preserving scheme the amount of information sent in each packet is fixed. The length of the packet is varied by changing thє stepsize of the quantizer used to process that packet. The length of the packet is chosen to match the available capacity on the network. Finally, simulation results are presented by operating the network at different loads.

CHAPTER 2 DIFFERENTIAL PULSE CODE MODULATION

In this chapter, a data compression scheme called Differential Pulse Code Modulation is introduced. This scheme is used for compression of digitized speech and images. Later sections of this chapter discuss adaptive algorithms needed for the efficient prediction of a nonstationary process. A more extensive description of the adaptive ARMA predictor and a detailed analysis of the adaptive quantizer will also be provided for future reference. Finally, simulation results using adaptive DPCM are provided.

DPCM has been used successfully with speech for many years. Though image data is highly redundant like speech, the results for image compression are quite disappointing. This is because the statistical properties of an image do not allow for easy removal of redundancy. The presence of edges causes the statistics of the signal to change rapidly. DPCM depends on these statistics being fixed or slowly changing. In this chapter we will discuss the adaptive DPCM with which the edge performance of an image can be improved.

DPCM is a differential approach to Pulse Code Modulation[5]. It uses the correlation between the samples of a waveform to predict the next sample value. The prediction process is implemented at both the transmitter and receiver in the same manner.

The predicted value is subtracted from the input signal at the transmitter. The resulting error signal is encoded using a quantizer and transmitted through the channel. At the receiver the received error signal is added to the predicted value to obtain an estimate of the original signal. If the predictors at the transmitter and receiver are identical then the same predicted value should result at the receiver.

A block diagram of the DPCM scheme is shown in figure 2.1. From figure 2.1 it can be seen that at the encoder, the prediction error e(k) is given by

$$e(k) = s(k) - p(k) \tag{2.1}$$

where the predicted value p(k) is given by

$$p(k) = \sum_{j=1}^{N} a_j \hat{s}(k-j) \tag{2.2}$$

where a_j are the coefficients of the predictor

and

$$\hat{s}(k) = e_q(k)+p(k) \tag{2.3}$$

The quantized prediction error $e_q(k)$ can be expressed as

$$e_q(k) = e(k)+n_q(k) \tag{2.4}$$

where $n_q(k)$ denotes the quantization noise.

If the channel is noisy $e_q(k)$ is received as $\tilde{e}_q(k)$ which is given by

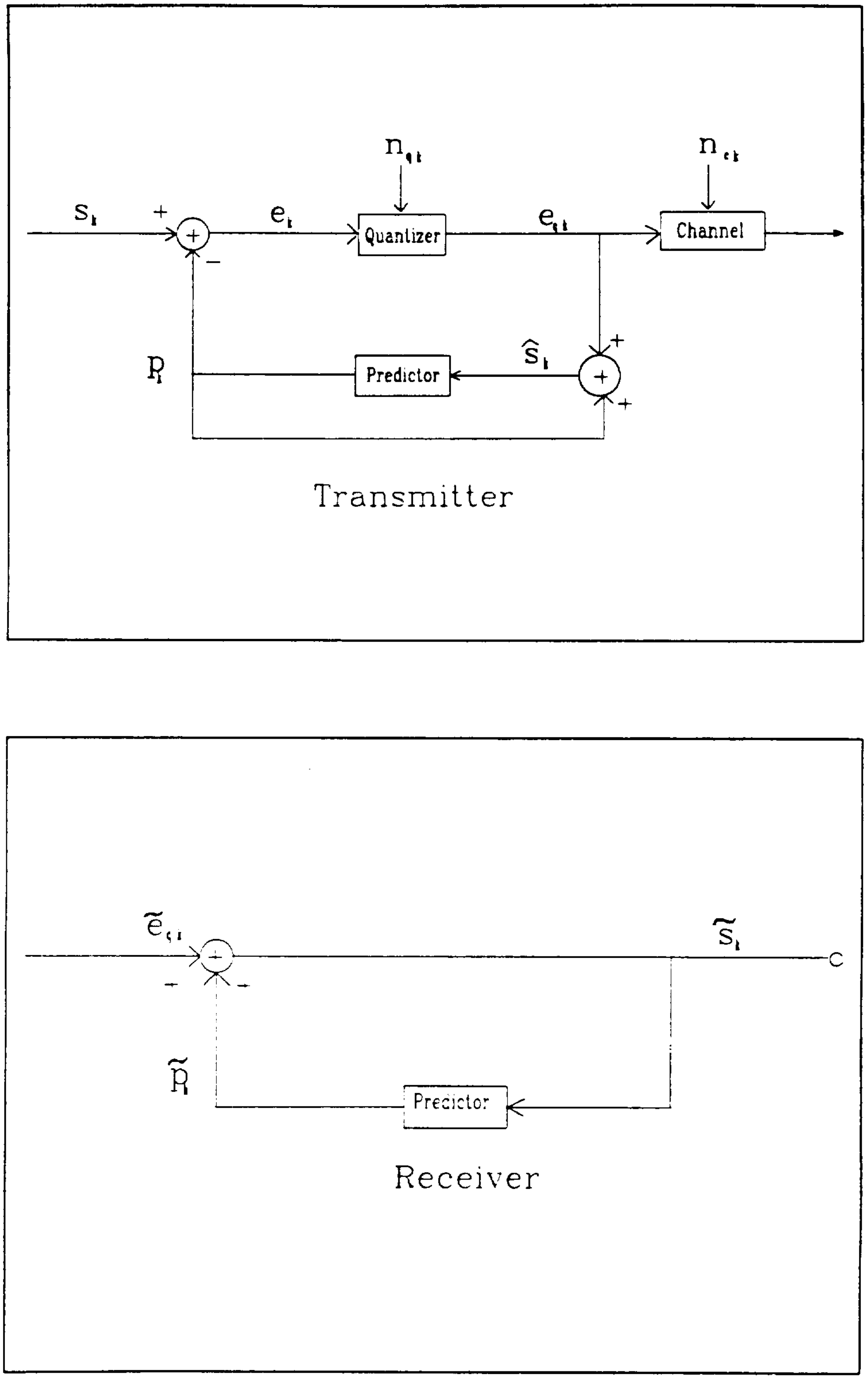


Figure 2.1 Block Diagram of a DPCM

$$\tilde{e}_q(k) = e_q(k)+n_c(k) \tag{2.5}$$

where n_c (k) represents the channel noise. Similarly, at the receiver

$$\tilde{s}(k) = \tilde{p}(k)+\tilde{e}_q(k) \tag{2.6}$$

where

$$\tilde{p}(k) = p(k)+n_p(k) \tag{2.7}$$

the additional term $n_p(k)$ being the result of the introduction of channel noise into the prediction process. Using equations 2.1, 2.4, 2.5 and 2.7 in equation 2.6, we obtain

$$\tilde{s}(k) = s(k)+n_q(k)+n_c(k)+n_p(k) \tag{2.8}$$

If the channel is error free, the last two terms in equation 2.8 will drop out and the difference between the original and reconstructed signal is simply the quantization error. Then equation 2.8 becomes

$$\tilde{s}(k) = s(k)+n_q(k) \tag{2.9}$$

Equation 2.9 indicates that the signal at the receiver is a noise corrupted version of the original source signal. In the region of an edge this corruption becomes quite large and the edge information can be lost. Therefore the goal of the DPCM design is to reduce the quantizer noise $n_q(k)$ to a minimum value.

The optimum predictor coefficients a_i for a particular image can be found by preprocessing the image. The advantage of predicting the signal and sending only the error signal is that it has a lower dynamic range and requires fewer bits to digitize. Another advantage of DPCM is that it is easy to implement which makes it ideal for on-line coding of signals.

This system works best if the input signal is stationary. In the following sections we describe an adaptive DPCM which allows the system to track slow moving changes in the source statistics.

2.1 Adaptive DPCM system:

The general theory of operation of DPCM discussed above still applies here. In ADPCM, both the quantizer and predictor are adaptive. Predictor coefficients and the step size of the quantizer vary to keep track of the changing input. The block diagram for an adaptive DPCM is shown in figure 2.2. Q_a is the quantizer's adaptation function which uses the previous output to determine the step size for the quantizer. P_a is the predictors adaptation function and determines the predictor coefficients a_k with the knowledge of $\hat{s}_k$ and e_{qk}. The quantizer used in this thesis is a 2-bit Jayant quantizer and the predictor is an adaptive ARMA predictor.

2.1.1 Adaptive Quantizer:

The quantizer used in this study is an adaptive quantizer with a one word memory. The idea behind this quantizer is that effective adaptation can be realized with an explicit memory of only one sample under the assumption

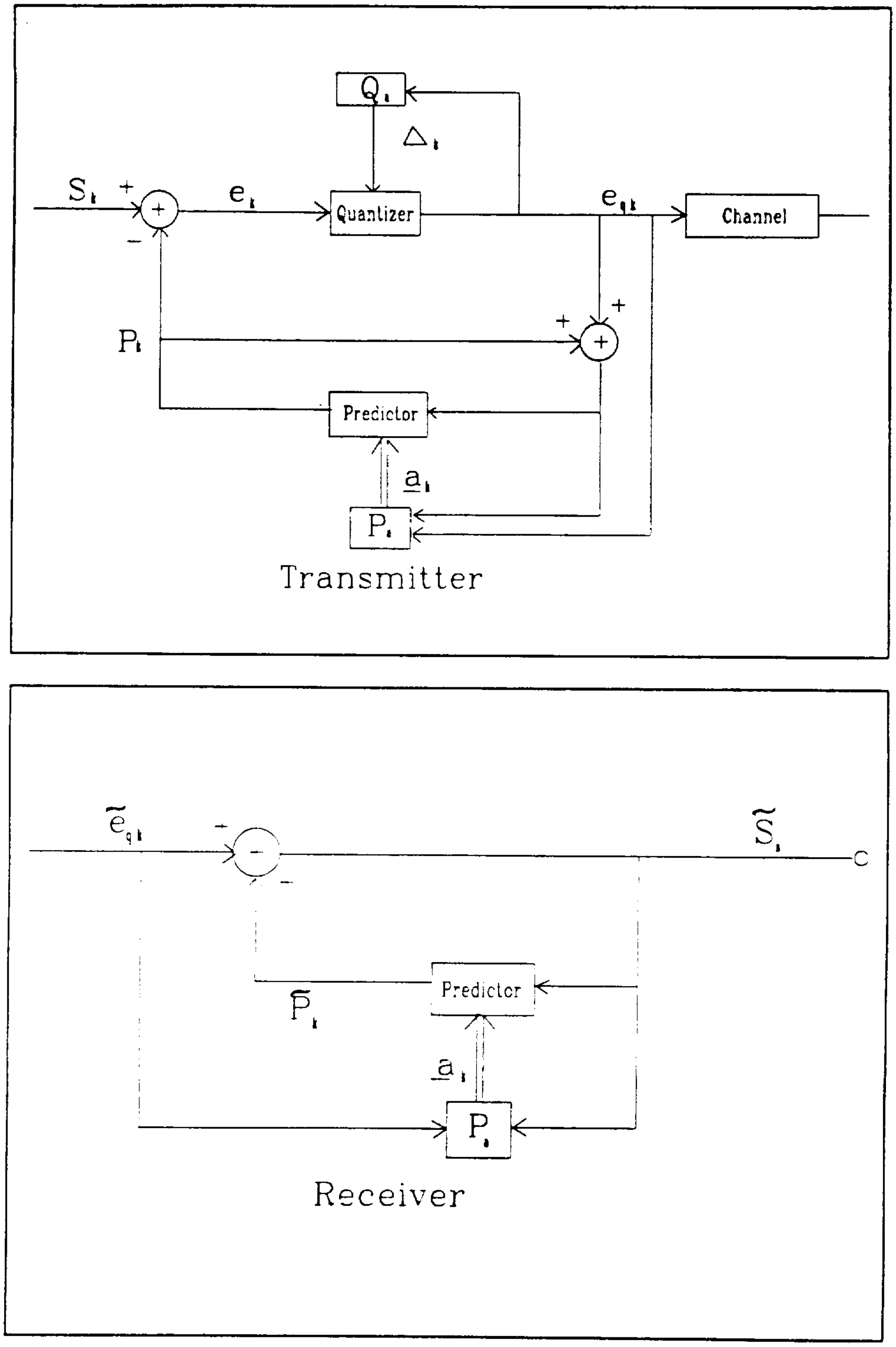


Figure 2.2 Block Diagram of an ADPCM

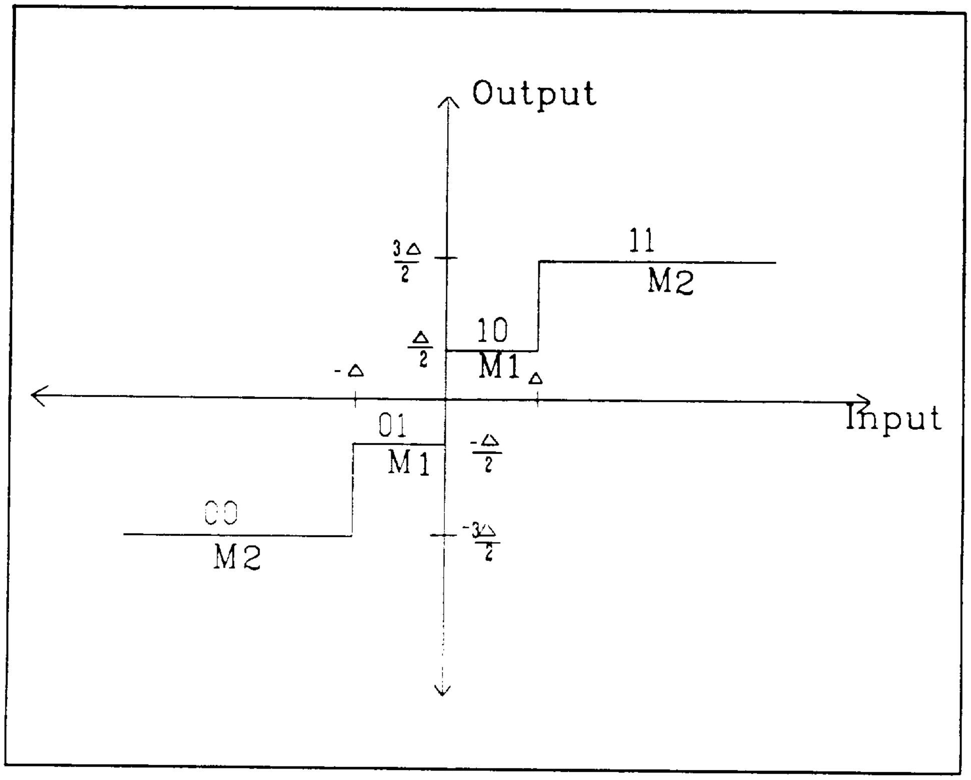


Figure 2.3 I/O Function of a Jayant Quantizer

all this information is available in step size Δ(n-1). It is commonly known as the Jayant quantizer.

The adaptation algorithm used with this quantizer is given by

$$\Delta(k) = M(|H(k-1)|)\ \Delta(k-1) \qquad 2.10$$

where Δ(.) is the step size and M(.) is the multiplier value. The input verses output of a 2-bit uniform Jayant quantizer is shown in Figure 2.3. The two bits given above the levels of the quantizer ranging from '00' to '11' are the output bits of the quantizer. For an input between zero and Δ the quantizer gives an output of '10' and for the input greater than or equal to Δ, the output will be '11' and so on.

The step size of the quantizer varies depending on the previous input value. If the previous input falls between -Δ and Δ, the step size will be multiplied by a value less than one. Otherwise, the step size will be multiplied by a value greater than one. That is, the algorithm assumes that the next sample value will be close to the present value and adjusts the step size according to the input. This is highly desirable for nonstationary image signals because it can adjust its output dynamic range to minimize the quantizer error.

The multiplier values used in this simulation are given below. M(1) corresponds to inner levels and M(2) corresponds to outer levels.

$$M(1) = 0.8$$

$$M(2) = 1.6$$

Tables containing M(.) values for DPCM as well as PCM are given in [6] and [7].

In the following section we describe adaptive ARMA predictors and see how their use affects edge encoding.

2.1.2 Adaptive ARMA Predictor:

The function of the predictor is to predict the next sample value depending on the past input samples. The complexity of a DPCM system is directly rel.ıted to that of the predictor algorithm. The equation for an ARMA predictor is shown below

$$p(k) = \sum_{i=1}^{N} a_i \; s(k-m(i)) + \sum_{j=1}^{M} b_j(k) \; e_q(k-l(j)) \qquad 2.11$$

where

m(i) = 1, 256, 257, ...

and

l(j) = 1, 2, 3, 4, ..., M

a_i are the coefficients of the fixed autoregressive predictor and b_j are the time dependent moving average coefficients.

AR coefficients can be calculated by using the Weiner-Hopf equation given below in matrix form [8]

$$A = R_{xx}^{-1} \; r_{xx} \qquad 2.12$$

where

$$r_{xx} = [R_{xx}(1) \;\; R_{xx}(256) \;\; R_{xx}(255)] \qquad 2.13$$

and

$$R_{xx} = \begin{bmatrix} R_{xx}(0) & R_{xx}(257) & R_{xx}(256) \\ R_{xx}(257) & R_{xx}(0) & R_{xx}(1) \\ R_{xx}(256) & R_{xx}(1) & R_{xx}(0) \end{bmatrix} \qquad 2.14$$

R_{xx} (n) is the discrete autocorrelation function. The statistics necessary to compute the AR coefficients are found by preprocessing the image to be used.

One way to adapt MA coefficients is by a steepest descent method. The idea behind this approach is to change the control parameter, the MA coefficients in this case, in the opposite direction of the cost criterion surface gradient. The cost criterion in this case is the quantized error variance. The gradient of the surface is defined as

$$\tilde{\nabla}(k) = -E[e_q(k)U(k)] \qquad 2.15$$

where E[-] denotes the expected value and

$$U(k) = [e_q(k-l(1))\ e_q(k-l(2)).....\ e_q(k-l(M))]^T \qquad 2.16$$

and M is the number of MA terms. From this, the recursive equation for the MA coefficients can be written as

$$B(k+1) = B(k) + \mu\ E\ [e_q(k)U(k)] \qquad 2.17$$

with the approximation

$$E[e_q(k)U(k)] \approx e_q(k)U(k) \qquad 2.18$$

equation 2.17 becomes

$$B(k+1) = B(k) + \mu e_q(k)U(k) \qquad 2.19$$

where

$$B(k) = [b_1(k)b_2(k).....b_M(k)]^T \qquad 2.20$$

The proper choice of μ will cause the predicted value to tend towards the minimum quantized error variance.

2.2 Results :

Three different images: Girl 256, Couple256 and image04 are simulated. From each of these simulations, Signal to Noise ratio (SNR) and Peak Signal to Noise Ratio (PSNR) values are computed. The resulting images and error images were also collected in order to view and compare edge effects.

The SNR is an indication of how much noise is added during transmission. It is the ratio of signal energy to noise energy and is usually expressed in decibels (db).

$$SNR = 10.0\ LOG_{10}(\sum_{k=o}^{\infty} s(k)^2 / \sum_{k=0}^{\infty} (s(k) - \bar{s}(k))^2) \qquad 2.21$$

The PSNR is a somewhat different measure which is used to compare SNR results from different images. If the PSNR of one image is subtracted from the PSNR of another the result is a number that represents how much more noise is added to one image than is added to the other.

$$PSNR = 10.0 LOG_{10}(255^2 / \sum_{k=0}^{\infty} (s(k) - \bar{s}(k))^2) \qquad 2.22$$

These two indices are computed from statistics gathered over the entire image; therefore only statistically based conclusions can be drawn from them. Nothing very specific can be said about the edge performance using these measures. However, if it is known that there is an overall improvement in edge performance, there will be an increase in the SNR or PSNR. Whereas, if there is an increase in the SNR or PSNR, this does not necessarily mean that there is an overall improvement in edge performance.

In Figure 2.4, the resulting images and error images of the simulations using the Girl256, Couple256 and image04 images are shown. And the results are shown in Table 2.1. The error images show distinctly where the edge errors occur and the full images can be used to see how these errors appear in reality. The error images are enhanced by a factor of 5.

The error image of Girl256 shows some edge errors on the left cheek, up the left shoulder, and in the flowers. Whereas the edges of the background, such as the window, the curtain, and the streak along the side of the picture are not visible. The error image of image04 shows some edge errors on the letters and the edge errors of ribs are almost invisible. These edge errors do not seriously effect the resulting images of Girl256 and 04.

Table 2.1 Results of a DPCM with an ARMA Predictor.

Image	SNR(db)	PSNR(db)
Girl256	24.54	34.08
Couple256	19.89	33.43
Image04	24.62	24.38

The Couple256 image has more edges in it. The error image shows many edge errors along the left edge of the picture and outlining the features of the people. Note also the errors in the edges of the chair, the window, and the vertical lines in the door jamb. These edge errors are quite significant and the resulting image looks blurred.

Therefore DPCM with an ARMA predictor is quite successful for images with few edges and produces a smearing effect in images having many edges. This edge smearing effect can be seen in the case of Couple256 and can not be improved any further with this system. These type of edge errors will be addressed in the following chapters.

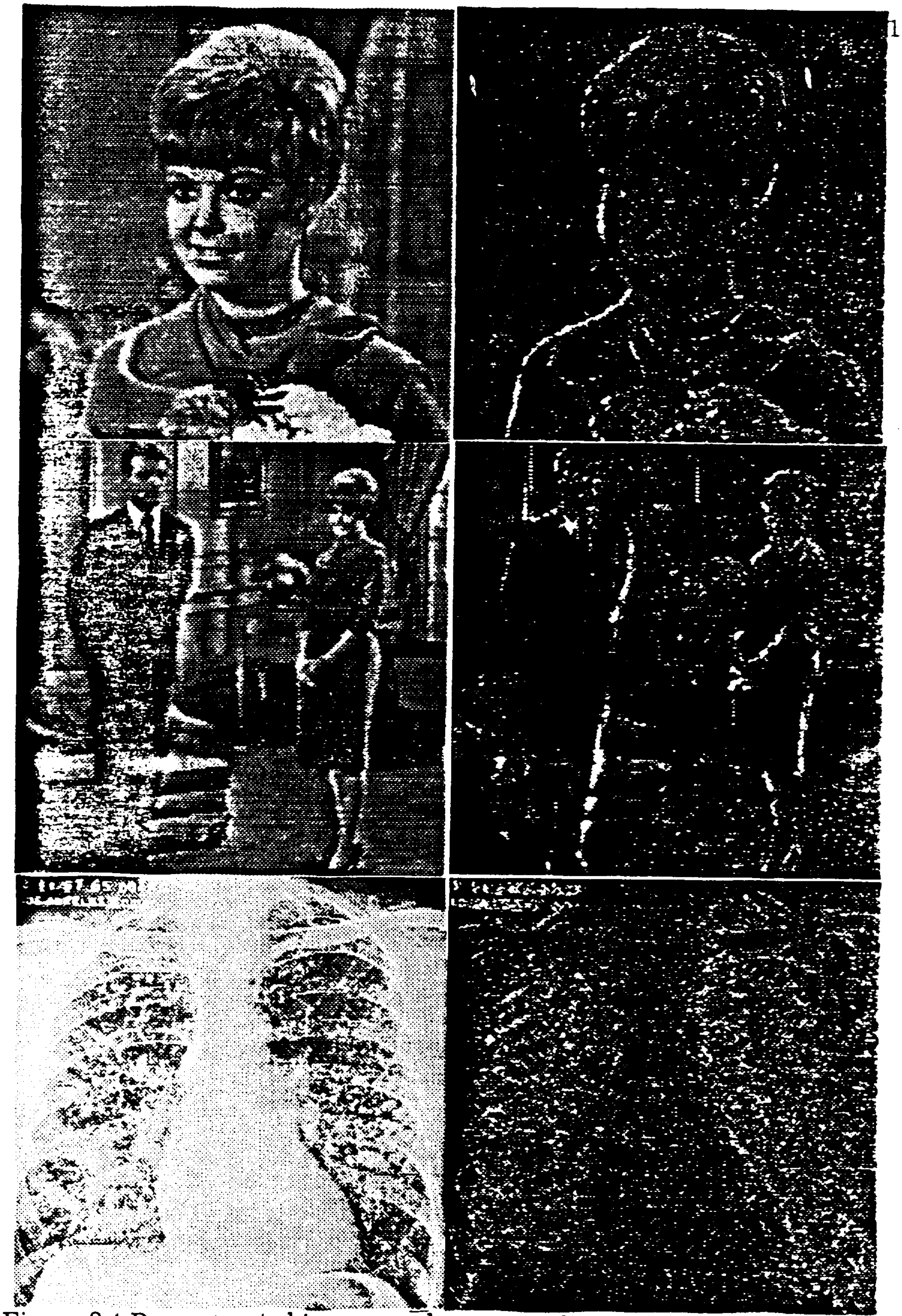


Figure 2.4 Reconstructed images and corresponding error images of a DPCM

CHAPTER 3 AN EDGE CORRECTING DPCM

In the previous chapter, we saw that even improved prediction algorithms like the adaptive ARMA prediction algorithm work well only for slow changing inputs. But in images, when an edge is encountered, the input signal changes rapidly. In these regions, quantizer error becomes quite large and this error propagates because the predictor uses the corrupted signal to make further predictions. This is because the information used to make predictions is strictly limited to past values of the quantized prediction error e_q (k).

In this chapter, we introduce a modified DPCM which uses information other than e_q (k) to detect edges and take action to correct them [9]. This scheme uses the recursive quantizer outside the DPCM loop to detect the quantizer error and sends this as side information. We also provide a brief introduction to entropy coding and finally present simulation results for this scheme.

The basic idea behind this scheme is to find the difference between the Jayant quantizer input and output. If this difference is above some threshold, the difference is quantized using another quantizer and sent as side information to the receiver. Simultaneously, at the receiver, this side information is added to the decoder output to get the original signal. The operation of the edge detection/correction scheme will be discussed in the following sections.

3.1 Edge Detection:

An edge can be characterized as an area where the statistics of the signal changes rapidly. DPCM performs well for slow changing input, but when the input changes rapidly, as when an edge is encountered, quantization noise becomes quite large. Therefore an edge can be defined in terms of DPCM signals as any region in which the quantizer noise is larger than some threshold value.

With the above definition for an edge, the Jayant quantizer itself can be used as an edge detector. As discussed in chapter 2, the step size of Jayant quantizer changes depending on the input signal variations. It has two modes of operation; granular mode corresponding to a steady input and slope overload mode corresponding to fast changing input. In the granular mode the quantizer noise is smaller in magnitude and is bounded by the size of the quantization interval. Whereas in slope overload mode the quantizer noise is unbounded and can be very large depending on the size of the prediction error. The adaptation algorithm used with the Jayant quantizer is

$$\Delta(k) = [M(|H(k-1)|)\Delta(k-1)]^{B} \qquad 3.1$$

The step size Δ(k) expands or contracts depending on whether the input falls in the inner levels or outer levels of the quantizer. For a slow changing or steady input the quantizer operates in a granular mode and the step size reduces to a minimum value and alternately expands or contracts around that minimum. For a fast changing input, the quantizer will switch to a slope overload mode and the step size repeatedly expands to increase the dynamic range of the quantizer. Therefore, it makes sense to conclude that some sort of edge has occurred if the quantizer switches from granular to slope overload mode. In a more general way we can conclude that an edge is encountered when the step size expands for n or more consecutive samples. Where n > 1

The number of stepsize expansions n, is chosen to be greater than one because even in the granular mode the step size alternately expands and contracts. For small values of n, the edges detected by this scheme may not be actual edges that the eye can perceive.

Because we are using the Jayant quantizer itself as an edge detector, the transmitter and receiver stay in synchronization. Whenever the transmitter detects an edge, the receiver will also detect the same edge without any timing information being sent. This makes it easy to implement and saves on overhead.

3.2 Edge Correction:

As discussed in the previous section, when the step size expands for n consecutive samples, the transmitter detects the presence of an edge in the image. Similarly, the receiver also detects the same edge without any additional information being sent. Now we must find a way to improve the performance of the DPCM system with this knowledge of an edge.

In absence of channel noise, the only corruption in DPCM is the quantizer noise. Now it can be detected effectively, whenever the corruption exceeds some threshold value. The performance of a DPCM system can be enhanced by using another quantizer to send this quantizer error as side information. At the receiver the quantizer error can be added it to the output of the receiver. This quantizer is known as a recursive quantizer and will be discussed in the following section.

The block diagram of this edge detection/correction scheme is shown in figure 3.1. The quantization error in the Jayant quantizer is found by subtracting the output of the quantizer from the input to the quantizer. This error is quantized using a recursive quantizer and is given as an input to the multiplier. The other input for the multiplier comes from the edge detector E_D which will be one only when the step size of the Jayant quantizer expands for n or more times. This makes the system send side information only when there is an edge. Similarly, at the receiver the edge detector gives a value of one for the same edge. Therefore, the side information will be added to the receiver output whenever there is an edge.

3.3 Recursive Quantizer:

The purpose of this quantizer is to encode the error information and send it through the channel as side information. This is to be done only

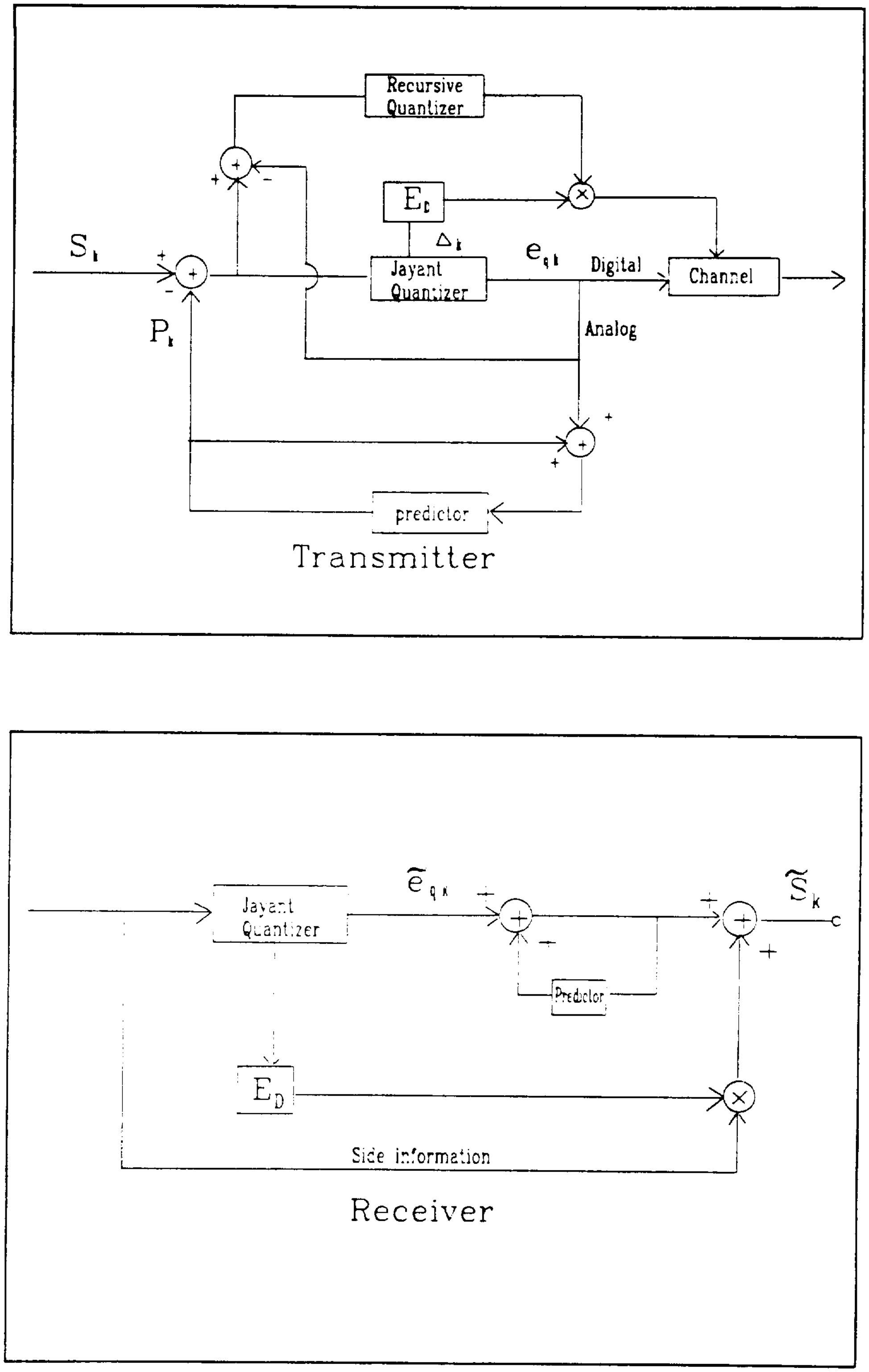


Figure 3.1 Block Diagram of an Edge Correction Scheme

when an edge is encountered. As 8-bit PCM data is used as input for the DPCM system, edges can be reconstructed perfectly with an 8-bit recursive quantizer. The cost, however, is a higher bit rate. A recursive quantizer is designed as a trade-off between bit rate and perfect reconstruction.

The bit rate can be reduced by using entropy coding with the recursive quantizer. Entropy coding is discussed in the following section.

3.3.1 Entropy Coding:

This is a variable length coding procedure which assigns codewords of variable lengths. Highly probable outcomes are assigned shorter codewords and vice versa. The average information content of a source is given by

$$H(s) = -\sum_{k=1}^{L} P_k \log_2 P_k \qquad 3.2$$

P_k is the probability of an occurrence of source output and L is the number of possible output levels.

A method known as Huffman code [10] can be used for designing an entropy code. This code has an important prefix condition property whereby it can be uniquely decoded without any error. The entropy of a code can have values in the range

$$0 \leq H(s) \leq \log_2 L$$

Where L is the total number of possible source symbols or levels of the quantizer. This coding scheme lowers the average word length if the source symbols are not equally distributed.

Like other coding schemes, this code also has drawbacks. First to design a code, the input data must be analyzed prior to transmission. If the source statistics change, the code designed will not be optimum. Secondly, due to variable length codeword, this code requires variable line speed. This is not a problem for this application, because we are implementing entropy coding for the side information, which even otherwise would have required variable line speed for sending the side information.

The following results were obtained by using a quantizer of 16 levels which is optimum for a signal with a Gaussian distribution and entropy coding for the side information. This edge detection correction algorithm was added to the adaptive ARMA simulation developed in chapter two.

3.4 Results:

In Figures 3.2 to 3.5, the error images of image01, image02, image03 and image04 by using the edge detection algorithm are shown. The results are tabulated in Table 3.1. Error images using the ARMA predictor are also shown for comparison. In the case of Girl256, a clear improvement in edge performance is indicated. The edge errors are almost invisible. This same pattern of improvement can be seen in the case of Couple256. This image has more edges in it, so it has more opportunity to be improved. Note that in the Couple256 error image, that the right side of the man as well as the halos around the couple and the entire background are all vastly improved.

In the case of image03, there are not many edges. There is only slight improvement from the ARMA predictor image to the edge detection simulation image. The overall SNR is only improved from 29.25 to 30.5db.

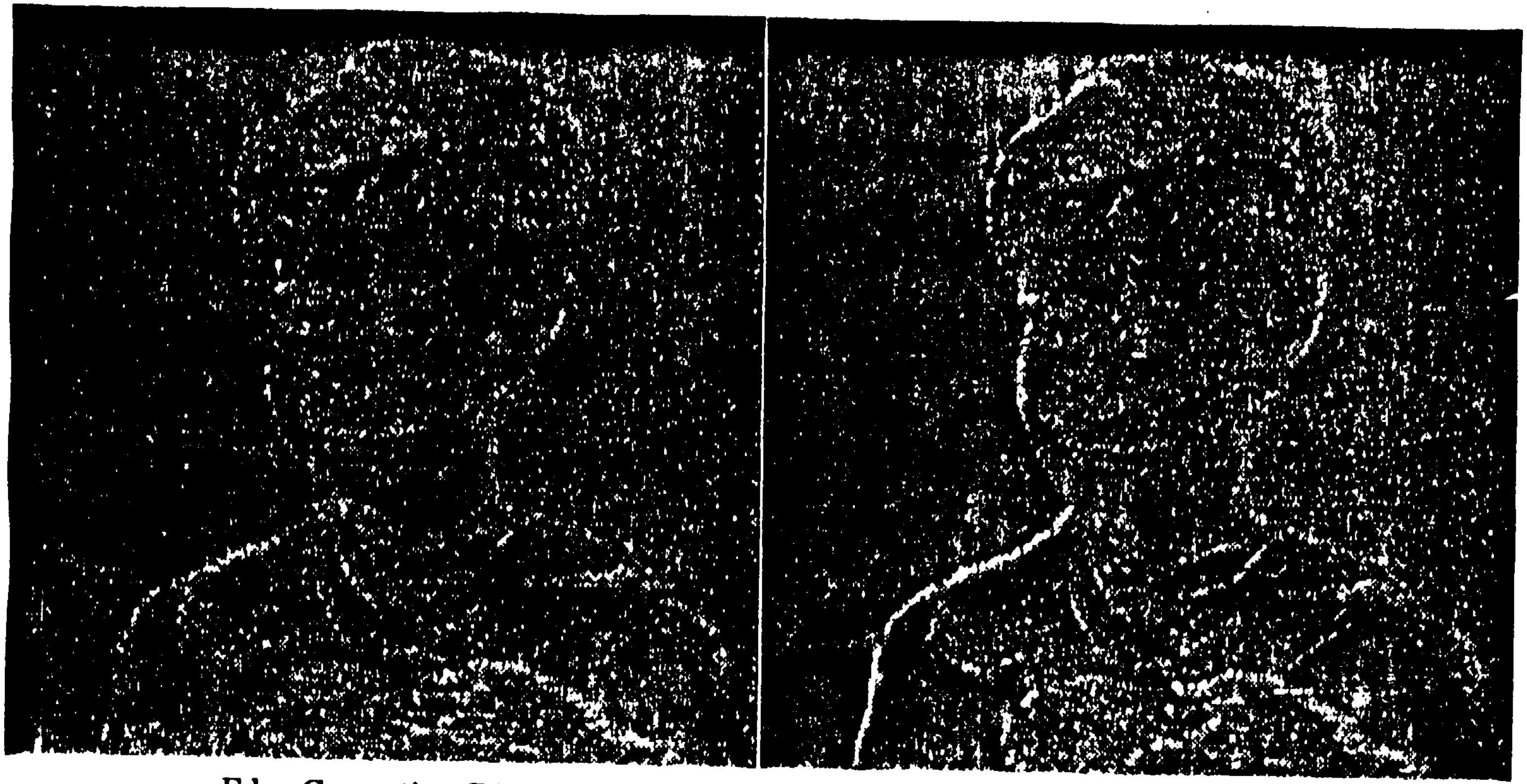
Whereas in the case of image04, the edges are too small to be detected as edges. The ribs and especially the letters in the image are between 1 and 3 pixels wide. So the edge detector can not respond to them, but the system did improve the edges that are wide enough to be detected as edges.

Table 3.1 Results of an Edge Correction Scheme with Entropy Coding.

Image	Rate bit/pix	Side Pixels	SNR (db)	PSNR (db)
Image01	2.179	3765	27.55	37.09
Image02	2.238	4898	24.08	37.62
Image03	2.159	3337	30.50	36.30
Image04	2.143	2939	25.77	28.53

The average side information required in images 01,03 and 04 the is around 0.15 bits/pixel, while with image02 with numerous edges, it required about 0.23 bits/pixel. The cost of the edge enhancement was small indeed.

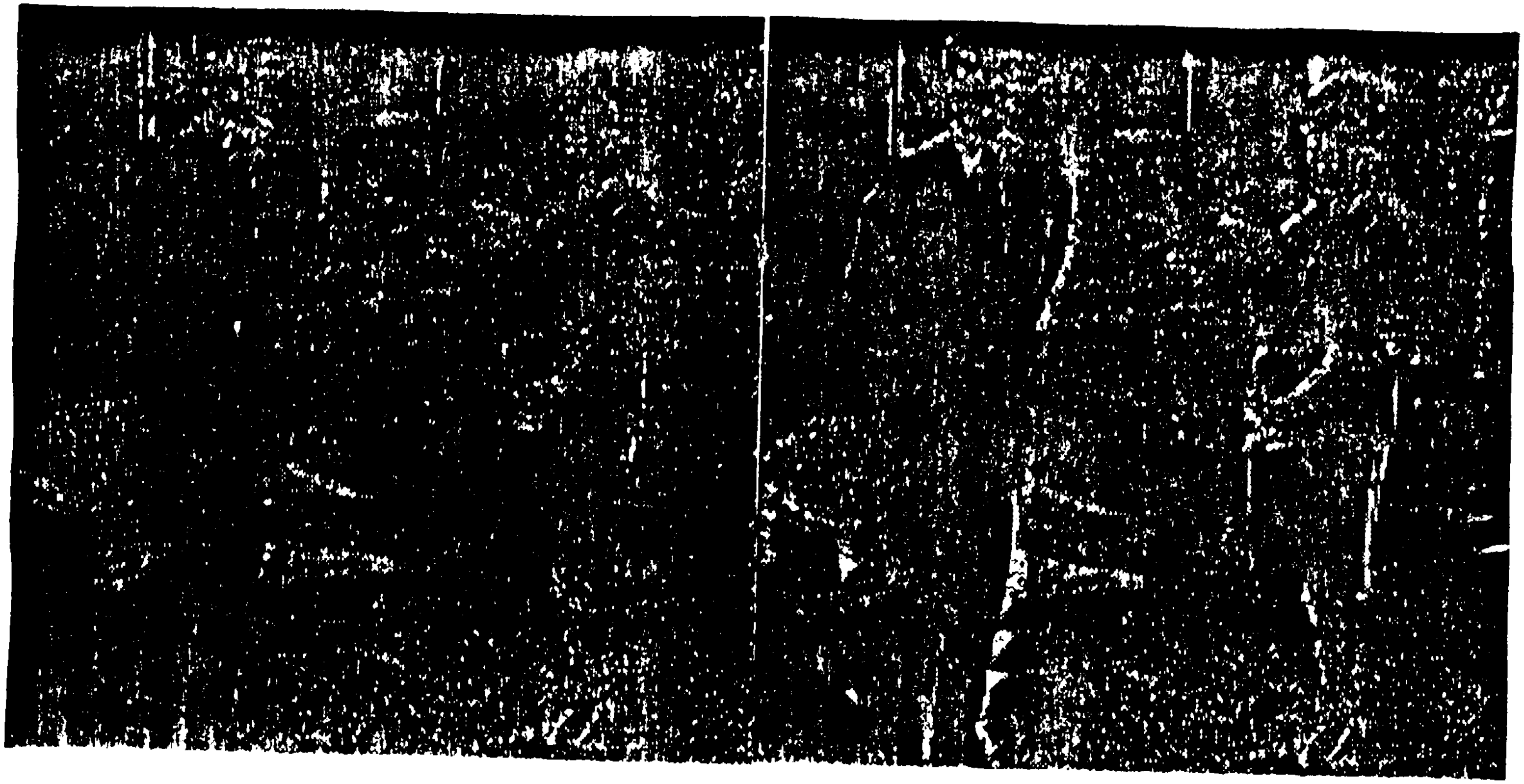
Based on these results we note that the extra information required for this edge correction scheme is justified, except where fine edges are involved. The disadvantage is that the use of side information requires variable line speed or another channel. However, this makes it ideal for use in a packet network environment, which will be discussed in chapter 6.



Edge Correcting DPCM

DPCM

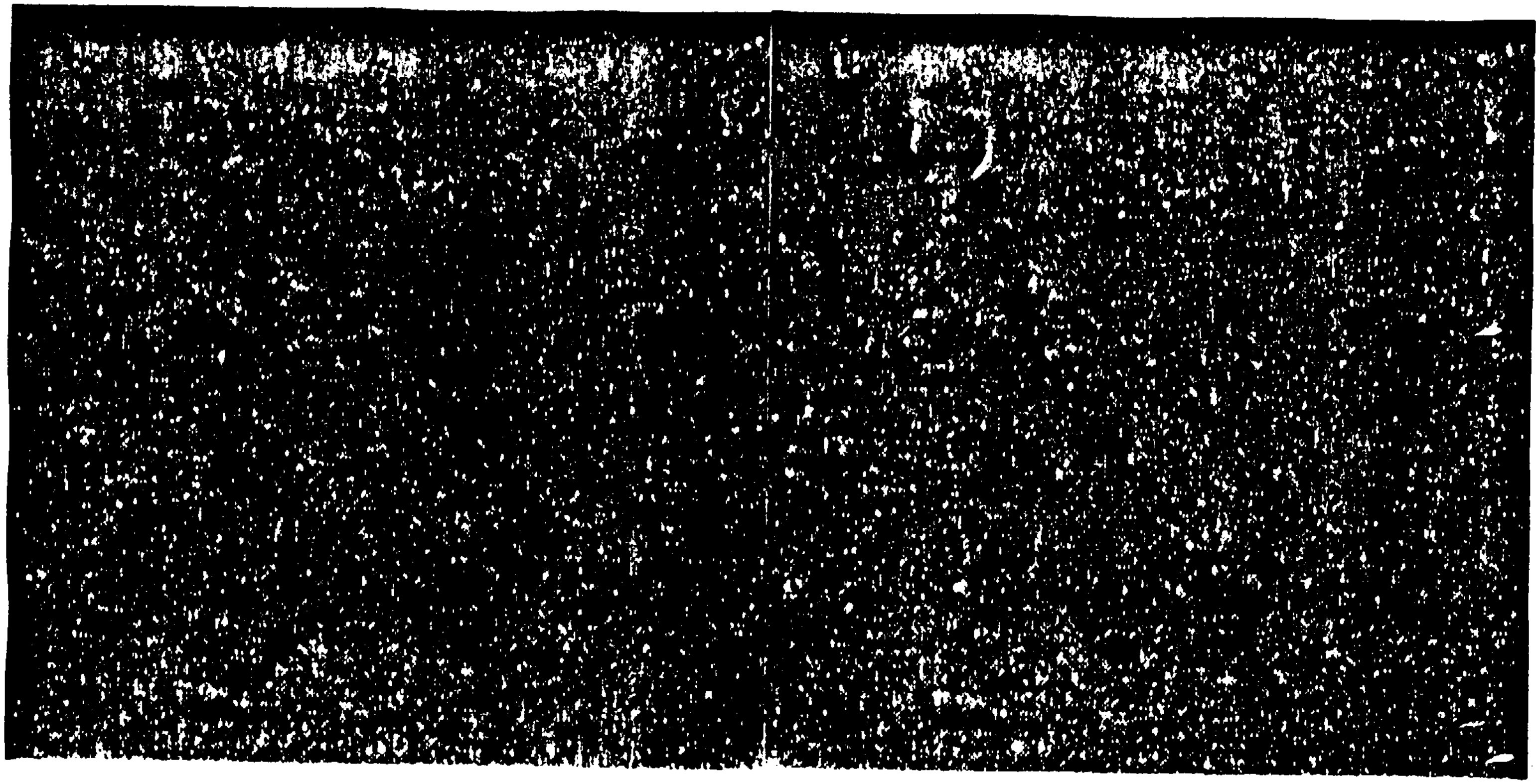
Figure 3.2 Error images of Image01 with an Edge Correcting DPCM and a DPCM



Edge Correcting DPCM

DPCM

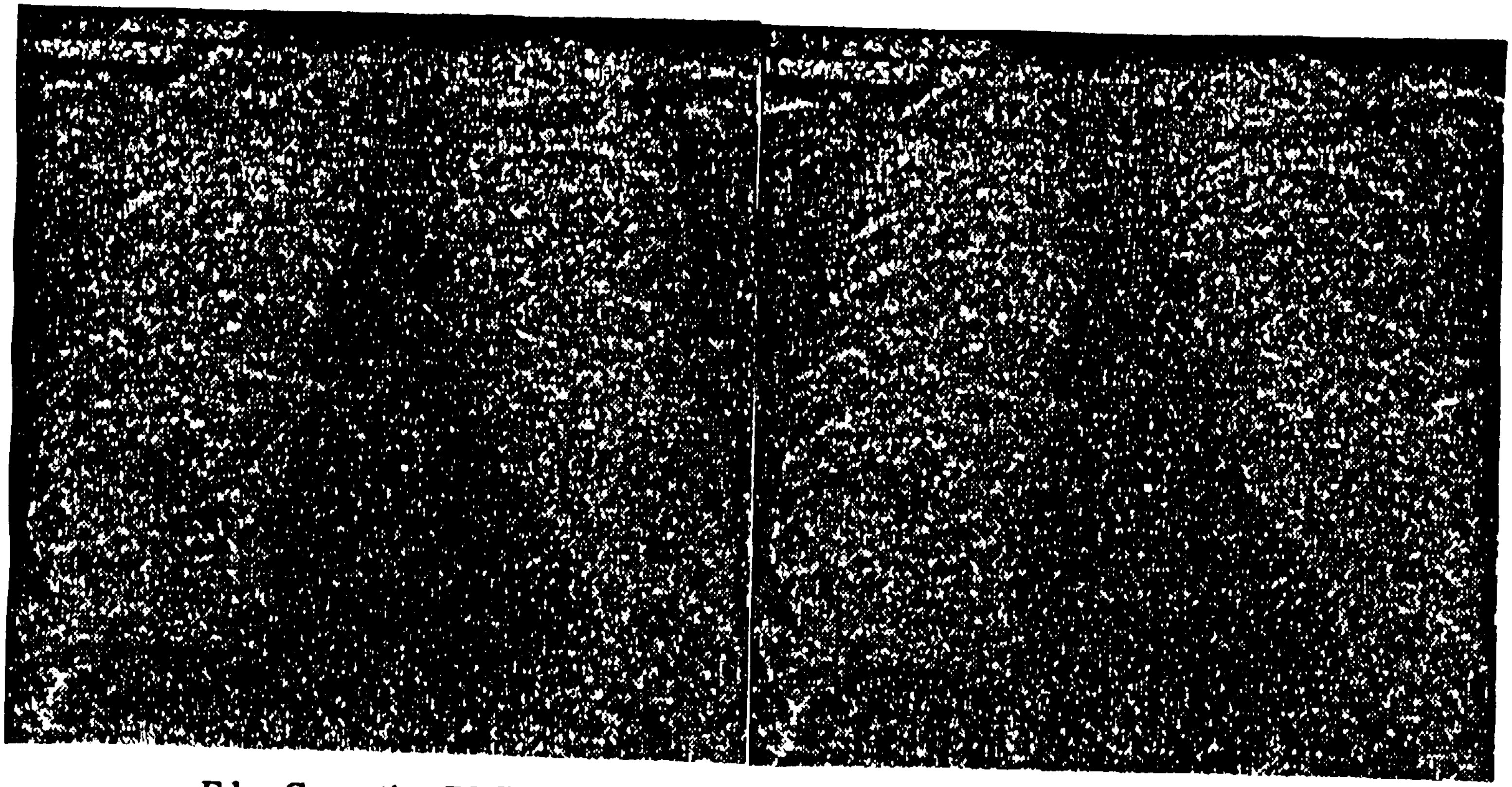
Figure 3.3 Error images of Image02 with an Edge Correcting DPCM and a DPCM



Edge Correcting DPCM

DPCM

Figure 3.4 Error images of Image03 with an Edge Correcting DPCM and a DPCM



Edge Correcting DPCM

DPCM

Figure 3.5 Error images of Image04 with an Edge Correcting DPCM and a DPCM

CHAPTER 4 AN EDGE PRESERVING IMAGE CODING SCHEME

In this chapter, we present a simple, easy to implement differential image coding system with excellent edge preservation properties. This coding system is slightly different from the edge correction scheme discussed in the previous chapter.

In an edge correction scheme, a Jayant quantizer is used as an edge detector and the quantizer noise is sent through a side channel whenever it exceeds some threshold value, i.e. when it encounters an edge. Then at the receiver this side information is added to the output to correct the edge. The advantage of this scheme is that it can be added to existing ADPCM systems. But the disadvantage is that it requires a side channel to send the quantizer noise.

In this chapter, we propose a different approach for edge preservation which does not require a side channel. In this approach, if the prediction error falls in the outer levels of the quantizer, the quantizer noise is re-encoded until it falls in one of the inner levels of the quantizer. The basic idea is to increase the rate whenever the quantizer input falls in the outer levels.

4.1 Problem Notation:

As described in chapter two, in a DPCM system, the next sample value is predicted and removed from the waveform at the transmitter and is added at the receiver. The difference between the input signal and the predicted value is called the prediction error. The prediction error is quantized and sent to the receiver. As we have seen from equation 2.9, when the channel is error free, the reconstruction error at the receiver is simply the quantization error.

When the quantizer input falls in the inner levels, the quantizer noise is known as granular noise. This noise is small and is bounded by the size of the quantization interval. If the input falls in the outer levels, the quantizer noise is known as overload noise. This noise is unbounded and can be very large depending on the input.

As we know, the input signal changes rapidly when an edge is encountered. These edge pixels are difficult to predict and the corresponding prediction errors are generally large. These large prediction errors fall in the outer levels of the quantizer and generate overload noise. Furthermore, these errors affect not only the current pixel but also future predictions. Therefore, the prediction error for the next few pixels tends to be large, leading to an edge "smearing" effect.

The performance of a DPCM system can therefore be enhanced by reducing the overload noise. One simple way to reduce the overload noise is to increase the number of levels for the quantizer i.e., increase the rate. But the rate increase is not required when the input falls in the inner levels. Therefore, instead of increasing the rate for each and every pixel, increase the rate only when an input falls in the overload region. The proposed scheme is discussed in the following section.

4.2 An Edge Preserving Scheme:

The basic idea behind the proposed scheme is that slope overload noise can be reduced by increasing the rate. This technique was developed by Rost and Sayood [2]. The advantage of this scheme is that the rate can be chosen depending upon the available channel capacity.

8-bit PCM data is used as an input to the DPCM system. If we could use a 256 level quantizer the input signal could be reconstructed perfectly at the receiver. But this increases the rate which is contrary to our purpose of data compression. Therefore we design the quantizer in such a way that it increases the number of levels only when input falls in the overload region.

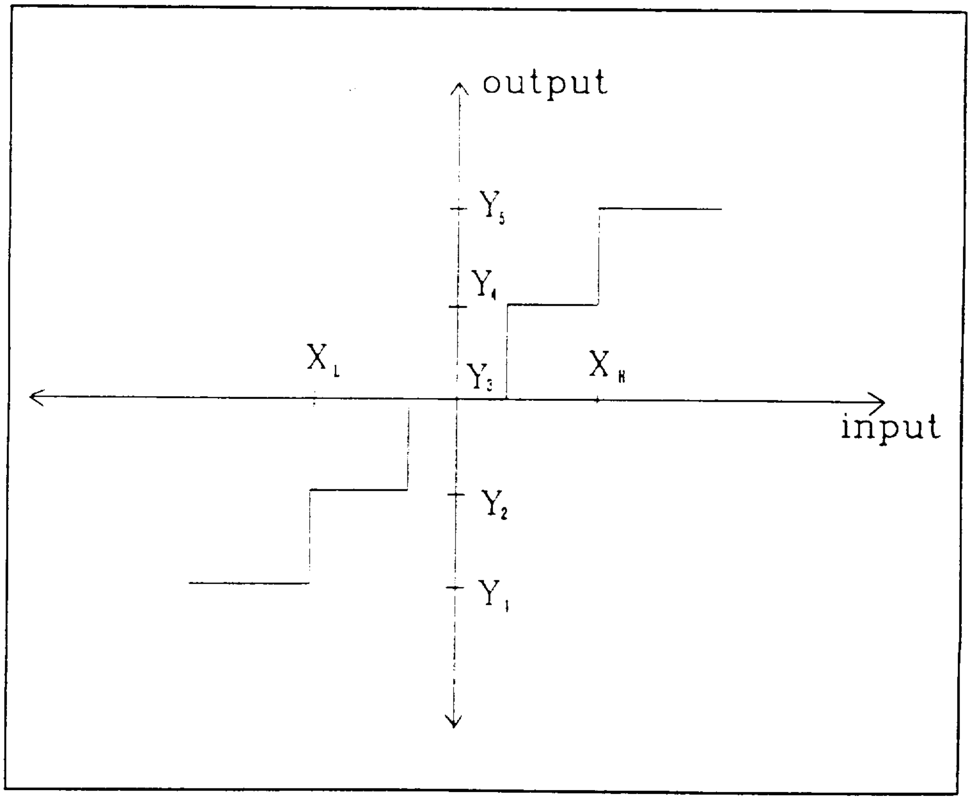


Figure 4.1 Input versus Output of a quantizer

When the input falls between X_L and X_H, the quantizer noise will be less than or equal to half of the quantization interval and the quantizer puts out the corresponding channel symbol. If the input falls in the range of less than X_L or greater than X_H, the encoder puts out the corresponding channel symbol, i.e. Y_1 or Y_5, depending upon whether the input is less than X_L or greater than X_H. When the quantizer error is obtained, this error signal is encoded and the process is repeated until the remaining error falls into one of the inner levels of the quantizer. Therefore, for an input in the range of less than X_L or greater than X_H, the output is more than one symbol depending upon the input.

Similarly, the receiver stops decoding when Y_1 or Y_5 is received and waits for the next symbols until a symbol other than Y_1 or Y_5 is received. Then the values corresponding to all these symbols are added up to reconstruct the corresponding pixel.

Let us say, for example, $X_L = -3$, $X_H = +3$, $Y_1 = -4$, $Y_5 = +4$ and the quantizer is a uniform quantizer. For a prediction error, say $e(k) = 1.5$, which falls in the inner levels of the quantizer, the encoder output will be Y_3, i.e. 2.0. For a prediction error of $e(k) = 7.0$, which is larger than X_H, the encoder puts out the corresponding codeword $Y_5=4$. Then it finds the quantizer error, which is $7-4 = 3$. This error signal is again encoded and the corresponding channel symbol, which is $Y_5 = 4$ is transmitted. Now the error signal is $3-4 = -1$, this is again encoded and the output symbol is $Y_3 = 0$. Note that for an input of 7, the output sequence is 4, 4, 0.

image or can be varied within the image, if the available channel capacity changes. Available channel capacity is usually a fixed quantity except in the case of a packet network environment.

Four images are run through the simulator; image01 and image02 of size 256X256 and Tiffany and Lena of size 512X512. Each image is run for different multiplier values, which gives different rates. But the rate or multiplier value of the step size is fixed within the image.

Image01, image02 and their resultant images are shown in figure 4.2 at a rate of 2.0 bpp. And the results at different rates for these two images are tabulated in table 4.1.

Table 4.1a Results at different bit rates for image01.

S.No.	Multiplier Value	Rate bpp	SNR(db)	PSNR (db)
1	1.0	1.087	18.41	27.95
2	0.8	1.121	20.19	29.73
3	0.5	1.254	24.11	33.65
4	0.4	1.354	25.85	35.38
5	0.3	1.529	27.99	37.53
6	0.2	1.896	30.93	40.47
7	0.175	2.057	31.97	41.51
8	0.15	2.264	33.15	42.69
9	0.1	2.946	35.84	45.38

Table 4.1b Results at different bit rates for image02.

S.No.	Multiplier Value	Rate bpp	SNR(db)	PSNR (db)
1	1.0	1.113	15.53	29.07
2	0.8	1.160	17.50	31.05
3	0.5	1.304	21.54	35.08
4	0.4	1.403	23.36	36.90
5	0.3	1.572	25.65	39.19
6	0.2	1.928	28.69	42.24
7	0.175	2.066	29.78	43.32
8	0.15	2.243	30.82	44.36
9	0.1	2.832	33.61	47.15

At the same bit rate, image01 shows better performance than image02. This is because image02 contains more details and the prediction error falls quite often into the outer levels of the quantizer generating overload noise. Whenever the quantizer input falls into the overload region, the error is encoded repeatedly until it falls into the granular region, which increases the bit rate.

The resultant and error images of Tiffany and Lena are shown in figure 4.3 at a rate of 2.0 bpp. The results with different multiplier values are tabulated in table 4.2.

Table 4.2a Results at different bit rates for the Tiffany iinage.

S.No.	Multiplier Value	Rate bpp	SNR(db)	PSNR (db)
1	1.0	1.180	27.85	35.51
2	0.8	1.256	29.71	37.38
3	0.5	1.502	33.49	41.16
4	0.4	1.670	35.29	42.96
5	0.3	1.956	37.51	45.18
6	0.2	2.533	40.51	48.18
7	0.175	2.786	41.60	49.27
8	0.15	3.125	42.50	50.17
9	0.1	4.283	45.34	53.00

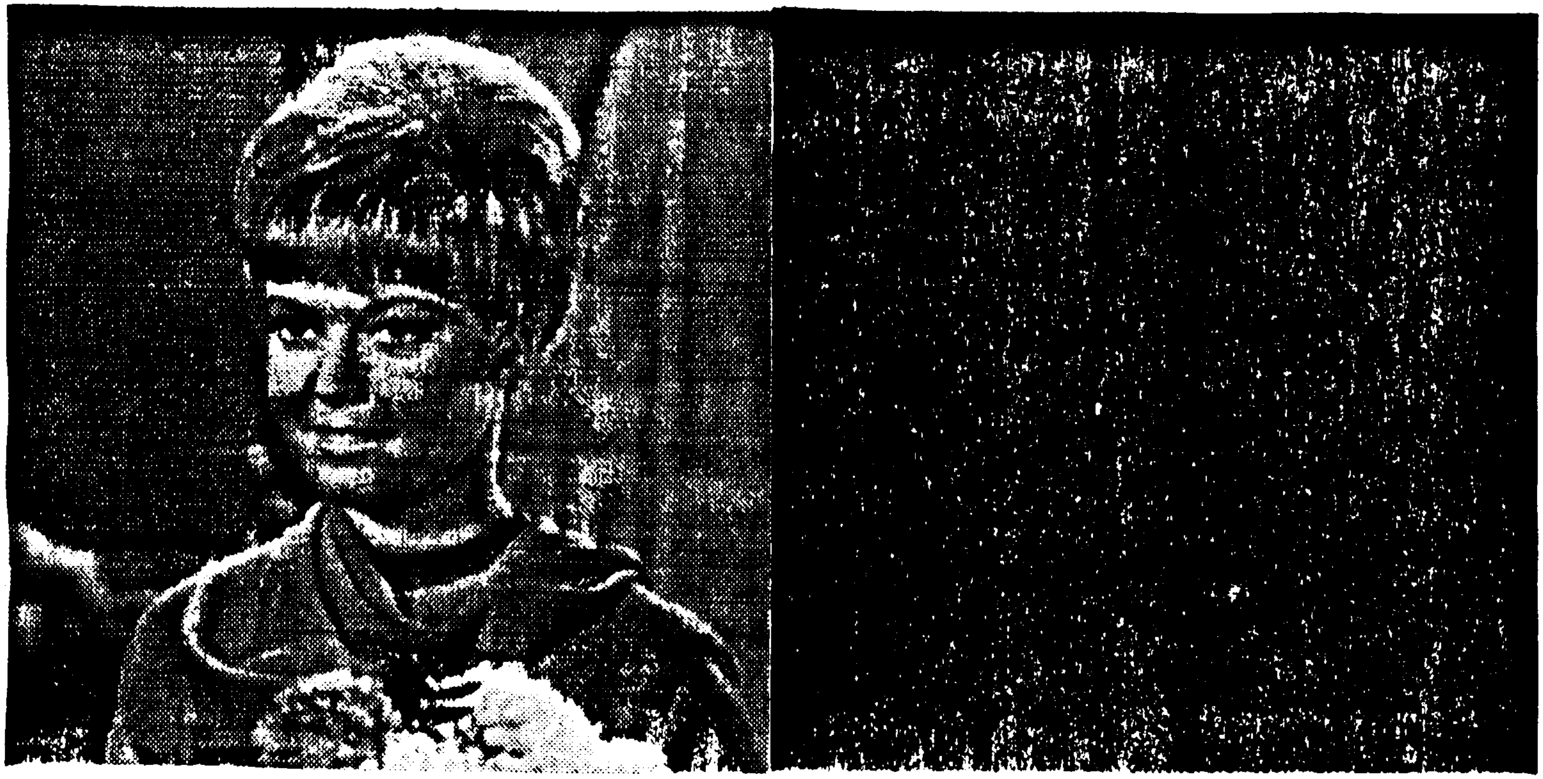
Table 4.2b Results at different bit rates for the Lena image.

S.No.	Multiplier Value	Rate bpp	SNR(db)	PSNR (db)
1	1.0	1.068	18.74	31.90
2	0.8	1.099	20.70	33.86
3	0.5	1.203	24.65	37.81
4	0.4	1.287	26.39	39.55
5	0.3	1.437	28.53	41.69
6	0.2	1.762	31.62	44.78
7	0.175	1.902	32.72	45.88
8	0.15	2.088	33.86	47.03
9	0.1	2.748	36.76	49.92

The image Lena which is also known as Hat girl contains more edges, and the Tiffany image contains face and a background with little detail. As the details or edges in an image increase, the quantizer falls more often into the overload region and increases the bit rate for the image. But in all the images shown at a rate of 2.0 bpp there are absolutely no edge errors.

This edge preserving scheme is easy to implement in real time and has excellent edge preservation properties over a wide range of bit rates. The edge preserving property is especially useful in encoding scientific and medical images.

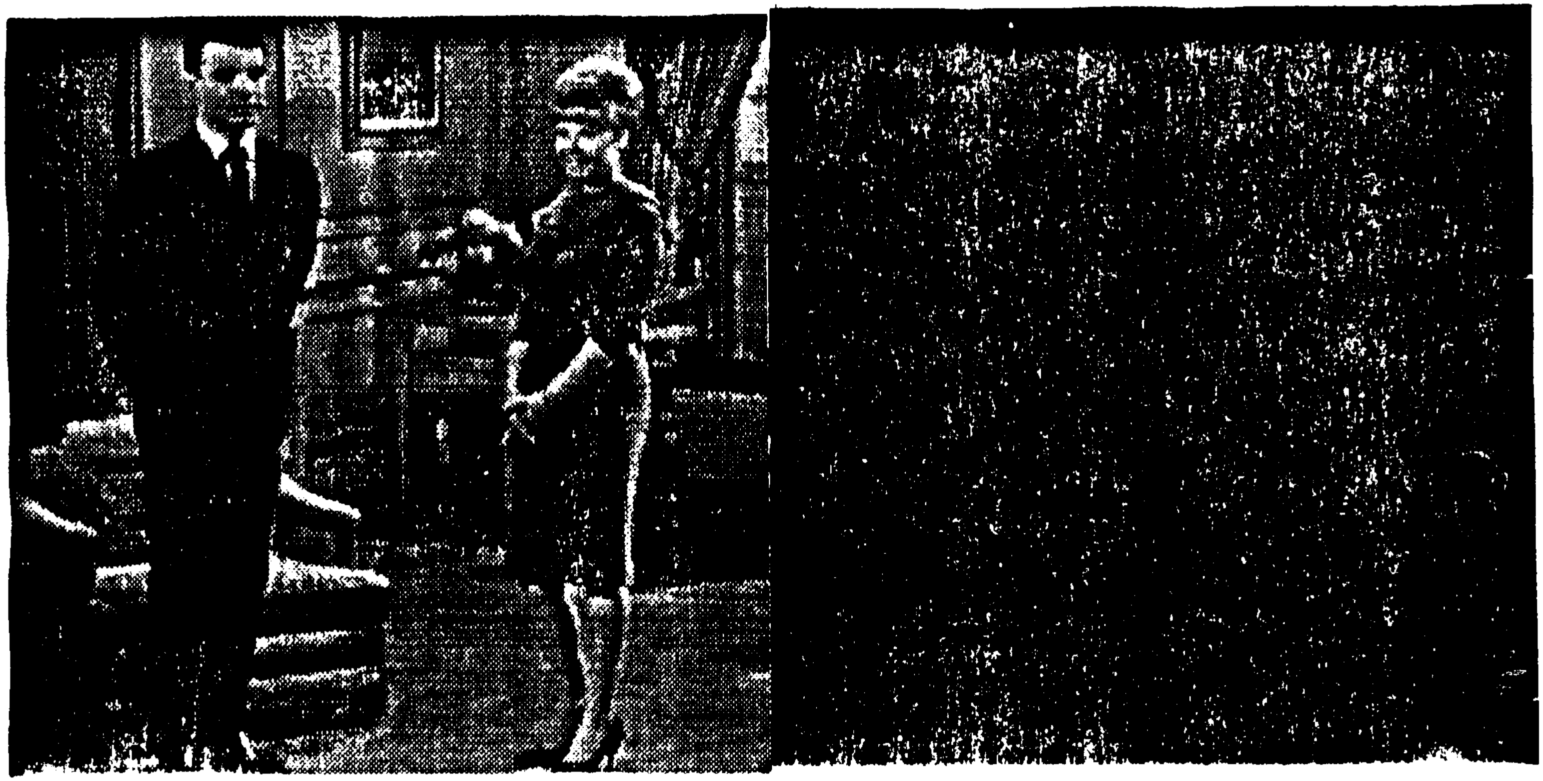
This scheme therefore requires operating at different rates. In a packet switched network, the available channel capacity is not a fixed quantity, but fluctuates as a function of the load on the network. If we implement this scheme on a network, it can take advantage of increased capacity when it becomes available while providing graceful degradation when the rate decreases to match decreased available capacity. This will be addressed in chapter 6.



Resultant image

Error image

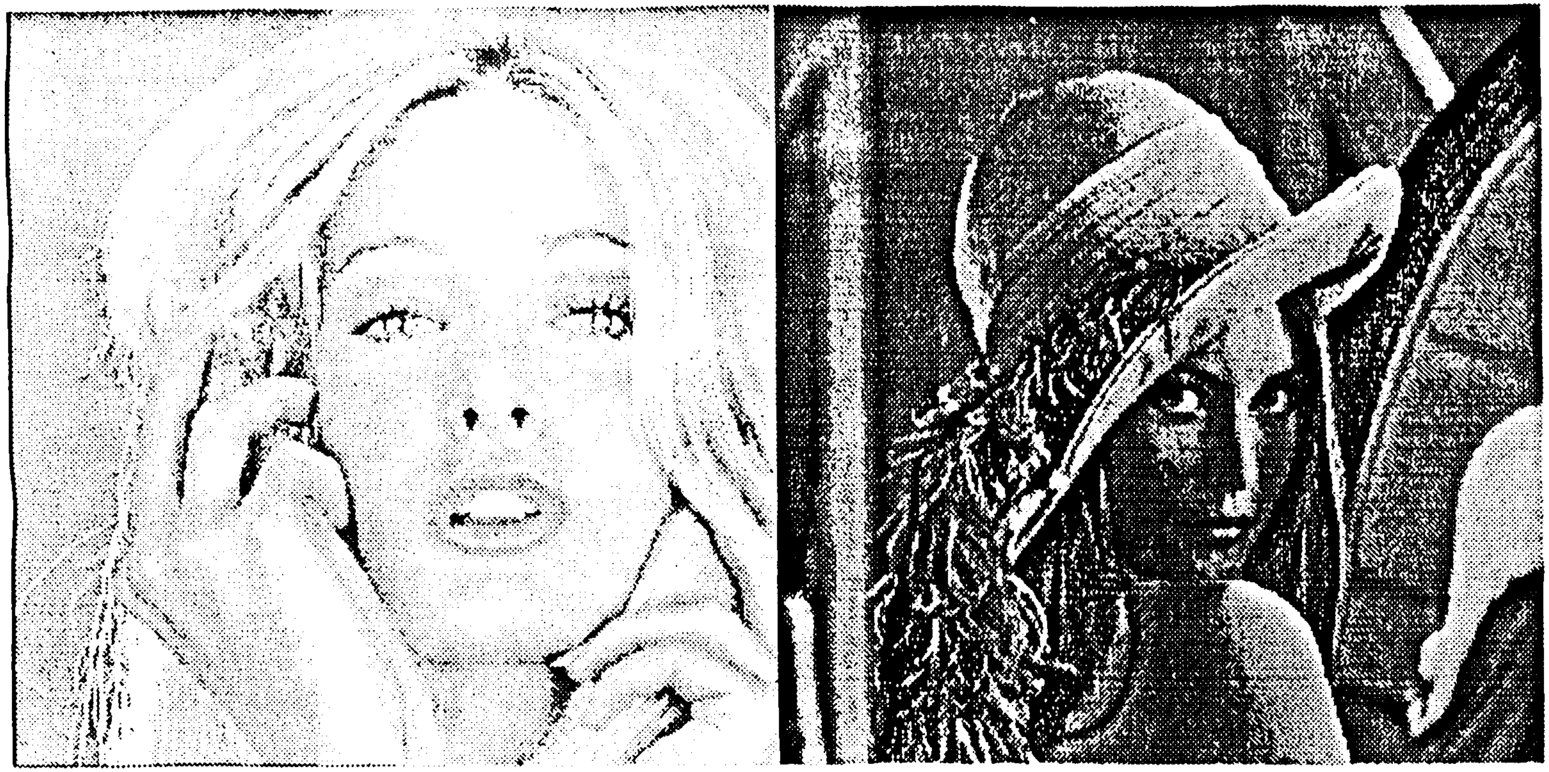
Figure 4.2 Resultant image and error image of Image01 with an Edge Preserving Scheme



Resultant image

Error image

Figure 4.3 Resultant image and error image of Image02 with an Edge Preserving Scheme



Tiffany image

Lena image

Figure 4.4 Resultant images of Tiffany and Lena with an Edge Preserving Scheme

CHAPTER 5 TOKEN RING NETWORK

In this chapter we provide an introduction to the token ring network and define the terminology associated with it. In the next chapter, the results of simulating the use of the image coding schemes discussed in the previous chapter on this type of network will be presented.

During the recent years, local area networks have been used for high-speed data exchanges and resource sharing at a single site such as an office building, factory, campus, etc. The token ring network constitutes one of the popular approaches to implementing local area networks.

5.1 Introduction:

A token ring network is shown in figure 2.1 [11]. In this network, nodes are arranged logically in a ring with each node transmitting to the next node around the ring. Each node simply relays the received bit stream from the previous node to the next with at least one bit delay. The token is defined as a special bit pattern which circulates on the ring whenever all the stations are idle. Whatever node has the token is allowed to transmit a packet. When the packet has been transmitted, the token is passed on to the next node. That is, whenever the node that is currently transmitting a packet finishes the transmission, it places the token, for example 01111110, at the end of the packet. When the next node reads this token, it simply passes the token if it

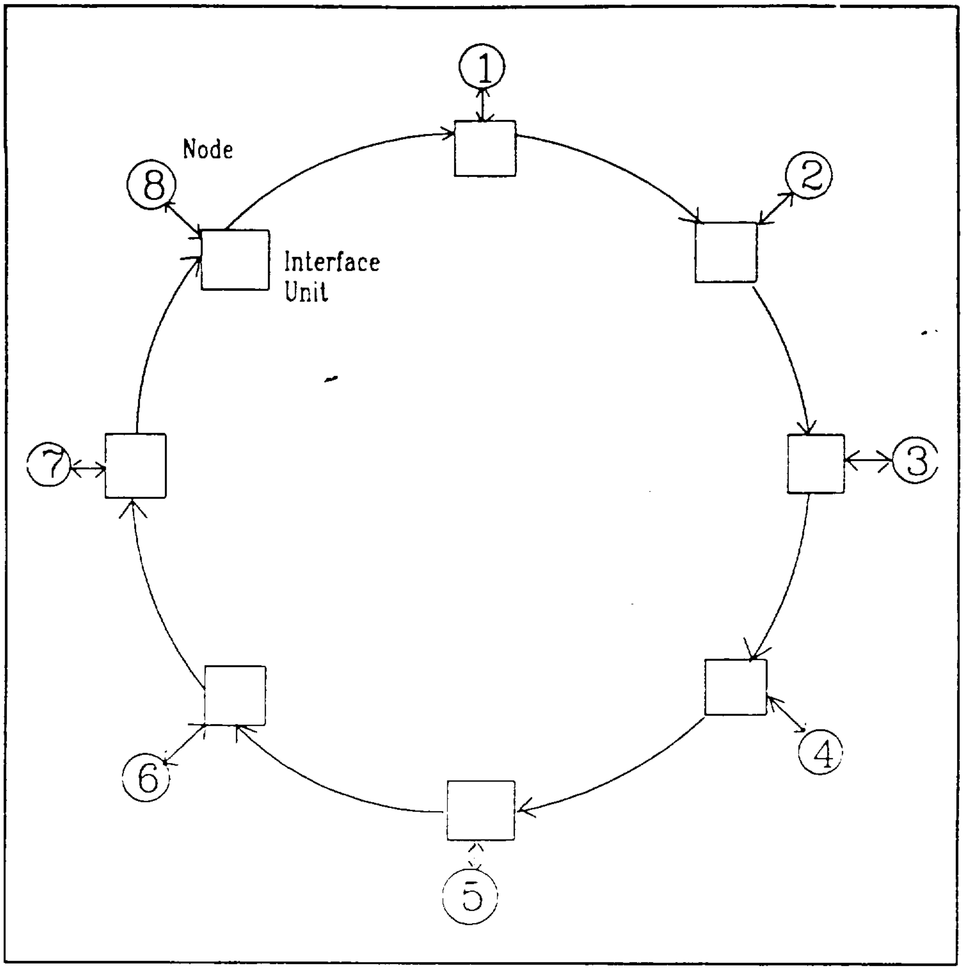


Figure 5.1 Token ring network

has no packet to send. If it does have a packet to send, it inverts the last token bit, turning the token into 01111111. The station or node then breaks the interface connection and enters its own data onto the ring.

Suppose for example, that at time zero, node 1 receives a free token and it has a packet to transmit. It inverts the last bit to form a busy token, and then starts to transmit a packet. Each subsequent node around the ring simply delays this bit stream by one bit per node and relays it on to the next node. The intended recipient of the packet both reads the packet into the node and also relays it around the ring. After the round trip delay, the bit stream or packet gets back to the originator, node 1 for our example, and is removed by it. The Originator verifies whether or not the packet was correctly received. If correct reception occurs, then a free token is transmitted; otherwise, a busy token is sent followed by a retransmission.

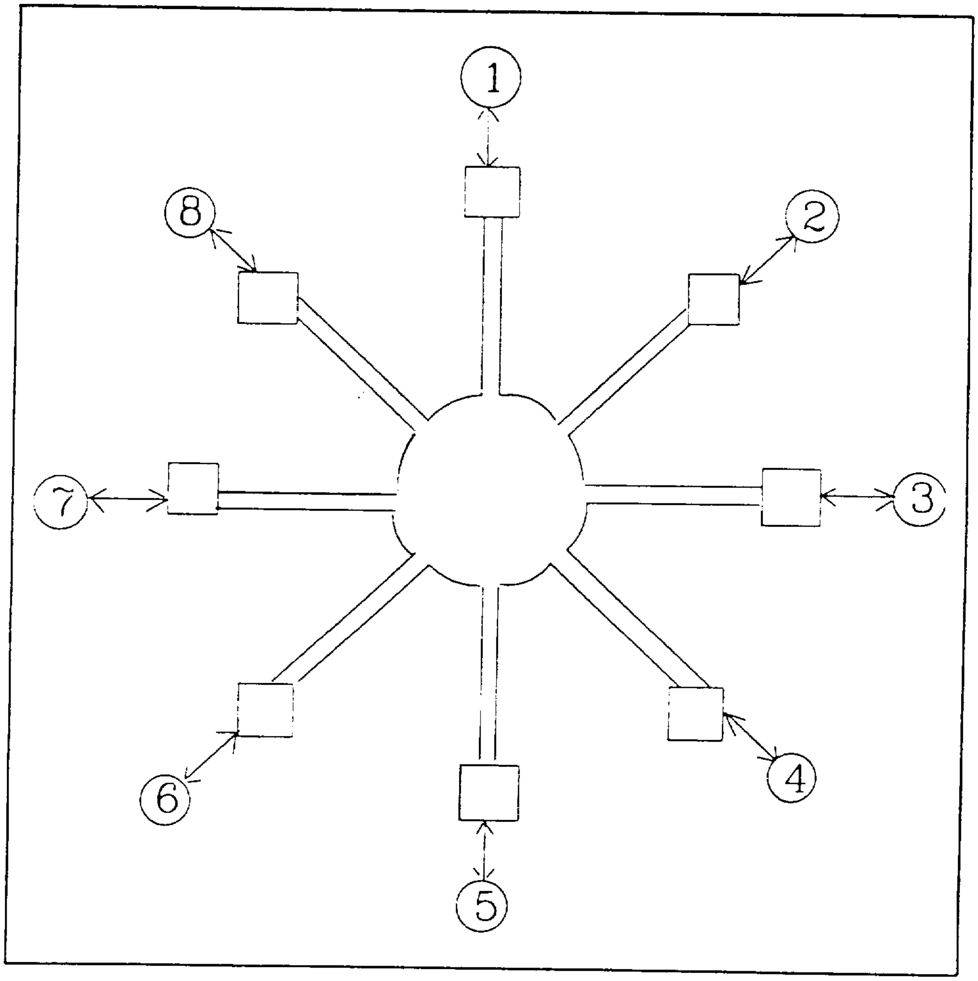


Figure 5.2 A ringnet in a star configuration.

One disadvantage of a ringnet is that if an interface at one node malfunctions then the entire ring fails. By physically connecting each interface at a common location, it is easier to find a failed interface and bypass it at the central site, such a topology is called a star topology. The token ring network in a star configuration is shown in figure 5.2. If the interface at the common location goes down, the entire network is down. Usually, token rings exists as star-ring topologies, which means that the physical topology is a star topology with individual cables running from a centralization point to each user location. The logical topology is the same as a ring network where there is no central node and all control is at the individual stations. It is wired as a star, but acts like a ring.

5.2 Packet:

A packet is a message with a physical header and trailer. A packet or token ring frame consists of 9 fields as shown below.

Frame Status	Ending Delimeter	Frame Check Sequence	Inform-ation	Source Address	Destin-ation Address	Frame Control	Access Control	Starting Delimeter

Figure 5.3 Packet format.

The starting delimiter is 8-bits wide and employs a unique Manchester code violation which indicates the start of the frame. The physical control fields consists of two frames, access control and frame control. Both fields are 8-bits wide in length and are used to manage the physical layer protocol. The first three bits in the access control field indicate the priority of the token frame in the ring. Bit four is set by the active monitor when it repeats a frame or token with a priority greater than zero. Communication between network controllers is through a special class of frames called Medium Access Control (MAC). The scheduling of packet transmission is part of the MAC layer of a local area network. The frame control field defines the MAC control field type.

The next two fields contain node addresses. The destination address, contains the address of the node for which the data packet is intended. Each node examines a frame as it passes through the ring. If the destination address matches the address of the local node, it is captured. The source address containing the address of the node which originates the data packet. The information field contains the actual data message. It must be at least one byte in length. The frame check sequence is an appended CRC calculation of all the fields before it, with the exception of the starting delimiter. The CRC algorithm is used to check for errors in the frame.

The ending delimiter defines the end of the frame. The frame status field indicates whether the frame's destination address was recognized, and whether the frame was copied by the intended node.

5.2 Bit Stuffing:

The message is a combination of 0's and 1's and it may contain a bit pattern which is similar to the token. When a packet is passed from one node to the other, a node reads the packet. If the message contains a bit pattern similar to the token, then the receiving node may consider it to be a token and change it to a busy "token" and start transmitting its own packet. Bit stuffing is used to prevent a bit pattern similar to a token from occurring in the data. For example, consider an 8-bit token, a bit stuffing algorithm would insert a zero into the data stream after each sequence of seven consecutive ones. The data receiver would use a similar algorithm to dispose of the inserted bit following any sequence of seven consecutive ones before decoding.

5.4 Priorities:

In practical system implementations, messages are often divided into different priority levels according to their delay time constraints. For example, a network control message, or a packetized voice message would require strict delay time limits while an ordinary data message may afford to experience higher delay fluctuations.

Priorities can be introduced fairly easily in a ringnet by having a field for priorities in a fixed position. A high priority station can reserve the next transmission right in the reservation field of the message transmitted by the low priority station when the reservation field passes through the high priority station. Then the free token at the high priority level is passed to the high priority station. After all messages from the high priority stations are transmitted, the free token is returned to the station in the down stream of the previous low-priority station [12].

5.5 Service Discipline:

Most work [13]-[14], carrying out delay analysis for token schemes assume that messages arriving at a station belong to a single class and employ a single service discipline, such as exhaustive, partially gated and gated or limited.

In an exhaustive system, when a node receives a free token, it empties all the waiting packets in the queue at that node before sending the free token to the next node. Therefore, each node sends a free token to the next node only when it finds its queue is empty.

In a partially gated service, a node sends only those packets which were present at the time of free token arrival, before sending the free token to the next node. Those packets which come after free token arrival will wait for the next free token.

In a gated or limited service network, each node can send only one packet with each free token capture. After sending one packet, a node will pass the free token to the next node, even if there are packets waiting in the queue.

The Delay time is the sum of delays on each subnet link traversed by the packet. One of the most important performance measures of a data network is the average delay required to deliver a packet from origin to destination. Queuing theory is the primary methodological frame work for analyzing network delay. In this thesis, an M/M/1 queuing model is used to analyze the network performance.

5.6 The M/M/1 Queuing System:

The M/M/1 queuing system consists of a single queuing station with a single server like a single transmission line in a communication context. Customers arrive according to a poisson process with rate λ, and the probability distribution of the service time is exponential. The meaning of these terms is explained below.

The first letter indicates the nature of the arrival process, M stands for memoryless, which here means a poisson process i.e., exponentially distributed inter-arrival times.

The second letter indicates the nature of the probability distribution of the service times. M stands for exponential distribution. Successive inter-arrival times and service times are assumed to be statistically independent of each other.

The last number indicates the number of servers. In this case we take only one server i.e., one node at each queue.

According to M/M/1 queuing theory, packet arrival process at each node is a poisson process and the inter-arrival time is an exponential distribution. Let us use F(x) as the cumulative distribution function and f(x) as probability density function. F(x) and f(x) are related by

$$F(x) = \int_{-\infty}^{x} f(x)\ dx \qquad 5.1$$

Let x have an exponential distribution, then the probability distribution function is given by

$$f(x) = 1 - e^{-x/\lambda} \qquad if\ x \geq 0 \qquad 5.2$$

$$= 0 \qquad \text{otherwise}$$

where λ is the mean inter-arrival time

Substituting 5.2 into 5.1 gives

$$F(x) = \int_{0}^{x} (1 - e^{-x/\lambda})\ dx \qquad 5.3$$

To generate an exponential random number, first generate a random number R in the range (0,1), then equate the number to the cumulative distribution function and solve for x. Cumulative distribution function F(x) is shown in the figure 5.4.

$$\rightarrow \quad x = -\lambda \ln(1-R) \qquad 5.4$$

The inter-arrival times of packets which have an exponential distribution are found by using the above formula.

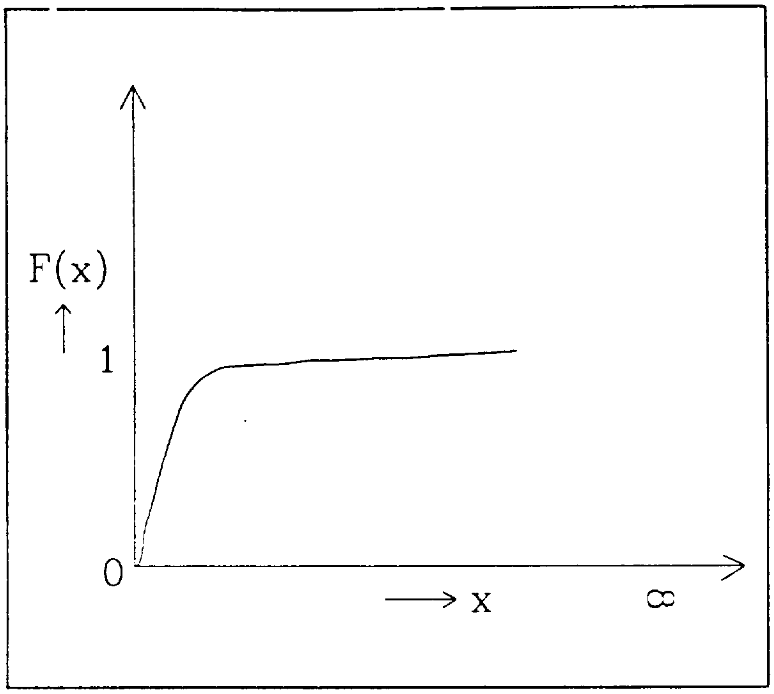


Figure 5.4 Cumulative distribution function.

5.7 Throughput:

Another parameter used to analyze the performance of a network is throughput ρ, which is given by

$$\rho = \frac{b\lambda_1}{r}$$

b = bits/packet

λ_1 = number of packets/second

r = channel capacity

Throughput tells about how efficiently the channel is utilized by the network.

The institute of Electrical and Electronics Engineers has developed a set of standards, divided into five parts, 802.1 to 802.5. The 802.1 standard deals with interfacing the LAN protocols to higher layers and 802.2 is a data link control standard. 802.3 to 802.5 are MAC standards referring to CSMA/CD, token bus and token ring systems, respectively.

The physical ring provides for IEEE 802.5 unidirectional point-to-point transmission of signals to/from up to 250 workstations attached to one ring. The rings can be connected through several bridges which can also radiate through a building or office complex. This standard has been implemented in VLSI chips to implement a token ring running at 4 megabits per second.

CHAPTER 6 SIMULATION

In this chapter, Edge Correcting DPCM and the Edge Preserving scheme will be simulated on a token ring network and finally the results of the simulation will be presented.

In chapter three, we discussed an edge correcting scheme, in which the edges were corrected by sending a quantized representation of the noise through a side channel. The advantages of this scheme are that the performance of a DPCM system can be improved with only little overhead and also it can be added to existing ADPCM systems.

The disadvantage is that the use of a side channel introduces synchronization problems. This scheme requires a variable rate channel, because the side information needs to be sent only when there is an edge.

These synchronization problems can be overcome by implementing this scheme in a packet network environment. Token ring networks support two major classes of traffic: synchronous and asynchronous. The synchronous class provides a pre-allocated bandwidth and a guaranteed response time, and may be used for time critical messages such as voice and video signals. The remaining bandwidth is dynamically allocated to asynchronous class traffic with non-time critical messages such as data. This network can also support multiple classes of asynchronous traffic by means of a timer-based priority mechanism. Analytical and simulation models have been used to evaluate the performance of the 802.4 priority scheme for homogenous token-passing bus networks supporting multiple classes of traffic [15]-[16].

To implement the edge correcting scheme on a token ring network, side information can be considered as asynchronous traffic and regular information as synchronous traffic. Apart from delay limits, side information is not very important and can be sent only when the bandwidth becomes available for the asynchronous traffic.

In the edge preserving scheme, the edges are preserved by increasing the rate whenever the prediction error falls in the overload region of a quantizer. This rate is increased by repeatedly encoding the quantizer error signal until it falls in one of the inner levels of the quantizer.

The advantages of this scheme are that it is easy to implement in real time and has excellent edge preservation properties. This edge preserving quality is very much required in the encoding of scientific and medical images.

This coding scheme can be operated at variable rates which makes it especially attractive for use in the packet network environment. In the following sections,the simulation of this scheme on a token ring network will be discussed.

In both of the simulations, we assume that there is no station fault, no transmission error and no token loss.

6.1 Timed Token Protocol Description:

In a token ring network, any station may capture the token by removing it from the ring. After the removal of the token, the station may begin to transmit information frame(s). When the transmission of information is completed, the station immediately issues a new token. It supports two classes of traffic;

(1) Synchronous traffic: A class of data transmission service whereby each requester is pre-allocated a maximum bandwidth and guaranteed a response time not to exceed a specific delay.

(2) Asynchronous traffic: A class of data transmission service whereby all requests for service contend for a pool of dynamically allocated ring bandwidth and response time.

A set of timers and several parameters are used to limit the length of time a station may transmit messages before passing the token to the next station and the duration of information transmission of each class within a station [17]. Each station maintains two timers, the Token_Rotation_Timer (TRT) and the Token_Holding_Timer (THT). TRT at node j is used to time the interval taken by the token to circulate around the ring starting from node j. When node j recaptures the token, TRT is reset and restarted immediately. Before resetting TRT, its current value is assigned to THT. TRT and THT both become active during message transmission at node j. THT is reset when the token is passed to the next station, and it becomes inactive while TRT continues to run until the token arrives at node j again.

When the network is initialized, the stations decide the value of a target token rotation time (TTRT), so that the requirements for maximum access time are met. The upper bounds on the maximum and average token rotation time have been studied in [18]; the results show that the token rotation time can not exceed twice the value of TTRT, while the average rotation time is not greater than TTRT. The extension to several priority classes is obtained by introducing a target rotation time for each additional class, and by using that value to check whether or not the station is allowed to transmit frames of that class.

If a station captures the token before its TRT reaches the value of TTRT, it is called as an 'early' token. If it captures the token after the TRT has exceeded the value of TTRT, then it is called a 'late' token. An 'early' token may be used to transmit both synchronous and asynchronous traffic while a 'late' token may only be used for synchronous traffic. The difference between the TTRT and TRT will be the available bandwidth for the asynchronous information. The amount of time a station can transmit is limited by the THT. In the following section, the simulation of the edge correcting scheme on this protocol will be discussed.

6.2 Simulation of Edge Correcting Scheme:

In this scheme, we have regular DPCM output as the synchronous message and the quantizer error as side information. This system can not afford to lose any of the regular DPCM output and also this information has timing constraints. The side information is not very important because the image can be reconstructed at the receiver with some degradation even without this side information.

In the analysis of a token protocol, it is generally assumed that the queues of asynchronous messages ready to be sent are heavily loaded, so that messages are always available for transmission. In our case, the asynchronous information queue will not be loaded heavily because the side information needs to be sent only when there is an edge.

The size of the packet for the synchronous traffic is fixed. Whenever the node captures an 'early' token, the size of the packet will be increased to match the available capacity and the regular information followed by the side information if present will be sent.

In the description of the protocol, we saw that the transmission ends when THT is reached or when there is no more asynchronous information to transmit. In this simulation, the transmission ends when the available bandwidth is utilized or when there is no more side information to send, because the asynchronous traffic is not heavily loaded. The most recent side information will be transmitted in the bandwidth available for asynchronous traffic. If there is any side information left after the transmission, it will be discarded.

It is assumed that TTRT is the time taken by a token to get back to the same node when all the nodes have a packet to send. If some node/nodes does/do not have a packet to send, the token comes earlier than the expected time. If node 1 captures an 'early' token i.e. TRT is less than TTRT, then (TTRT-TRT) becomes the available for the side information. If node 1 captures a 'late' token i.e. TRT is greater than TTRT, then only the regular information will be sent and the side information will be held in the queue.

At the receiver, whenever it receives an increased size packet it takes the bits received after the regular size of the packet as side information. This side information will be added to the corresponding most recent pixels received to correct the edge errors.

The inter-arrival time of packets are found by using equation 5.4 the parameter, λ in that equation indicates the average inter-arrival time. The inverse of λ gives how many packets arrive at each node in one unit of time, which is defined as the load in this thesis.

The following parameters were assumed in this simulation :

Number of nodes	= 50
Bit travelling speed	= 200,000 met/msec
The distance between nodes	= 100 meters
Data generation rate	= 11,000 bits/msec
Size of the packet for synchronous information	= 1540 bits
The time taken by a node to read the data	= 10μsec
Channel capacity of the coaxial cable	= 12,000 bits/msec

The system is assumed to work according to the following general conditions :

- The packet arrival process at each node follows a Poisson distribution. The actual image information is taken at node 1 with regular DPCM output information arriving into one buffer and side information into the other buffer.
- The messages transmitted by each station belong to two classes, i.e. asynchronous and synchronous messages.
- The access mechanism is based on the timed token approach but different classes of asynchronous messages are not considered.
- The queues of asynchronous messages are not heavily loaded.
- When the network is started, the token rotation will allow only the transmission of synchronous messages; the second token rotation will allow both synchronous and asynchronous messages.

6.3 Results for an Edge Correcting Scheme:

The token ring is simulated by taking image information obtained using the edge correcting scheme at node 1 and random information at all other nodes for the two cases given below:

(i) Messages transmitted by all the nodes belong to two classes i.e. synchronous and asynchronous messages.

(ii) Messages transmitted by all the nodes belong to only one class i.e. synchronous.

Load versus delay and throughput versus delay characteristics are plotted for both the cases and are shown in figures 6.1 and 6.2 respectively. The graph in figure 6.1, shows that at a particular value of the load, the average delay of a packet in a network with both the classes of traffic is more compared to the case where only synchronous messages are transmitted. This is true especially at low loads; as the traffic increases there is not much difference in the delay for the two cases, because the network will not have badwidth available for asynchronous traffic when the traffic is busy. It is quite obvious that the transmission of asynchronous messages increases the delay for the synchronous traffic.

The token ring network transmitting both synchronous and asynchronous messages provides better delay versus throughput characteristics. From the graph, it can be seen that at a particular value of throughput, the delay is less or at a particular value of delay, the throughput is more for the first case. Here again, at large values of throughput there is not much difference between the curves. The reason for the better throughput versus delay characteristics is that even at low loads the network can utilize the channel efficiently by transmitting asynchronous messages whenever the bandwidth becomes available.

Image02 and image05 are simulated and the results obtained at different network loads are shown in tables 6.1 and 6.2 respectively.

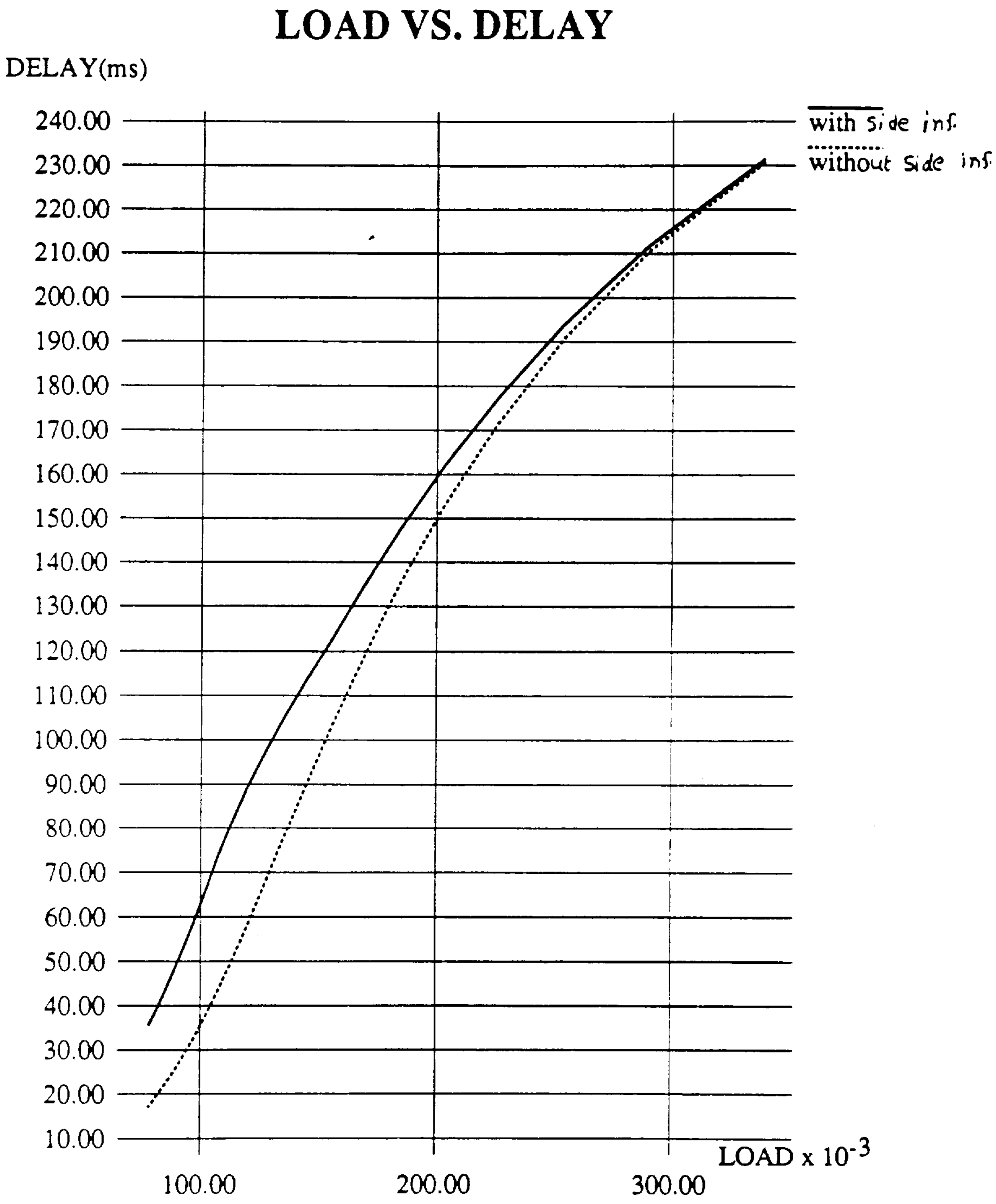


Figure 6.1 Load versus Delay for the Token Ring

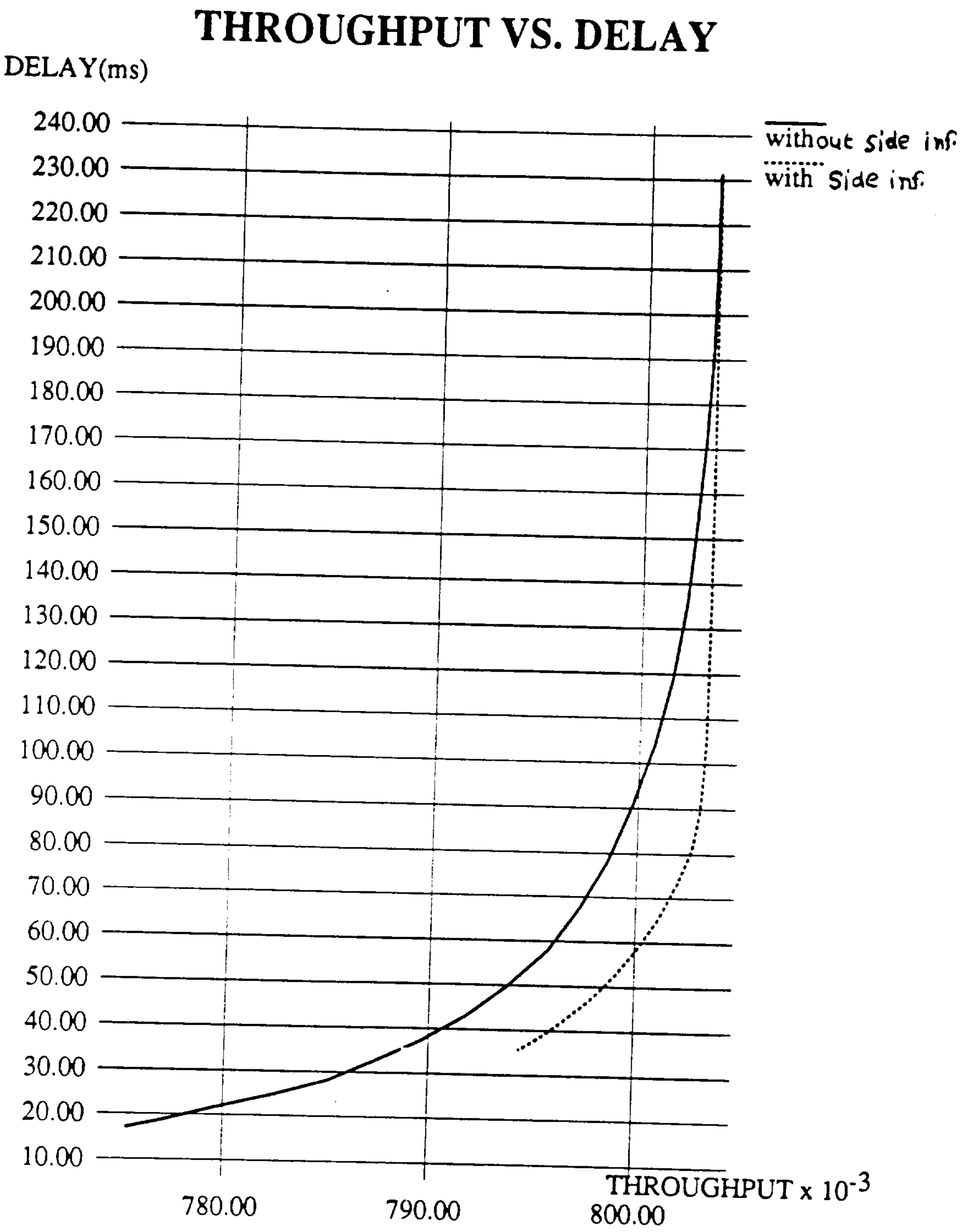


Figure 6.2 Throughput versus Delay for the Token Ring

The first two readings in these tables are taken by operating the network at high loads which is in the unstable region. At these high loads almost every node will have a packet to send and will not have bandwidth available for side information. As the load decreases, more and more side information will be transmitted, which provides a better reconstructed image at the receiver. In this simulation, at a load of around 0.09, node 1 finds enough bandwidth to transmit all the side information and after that, decreasing the load will not improve the image.

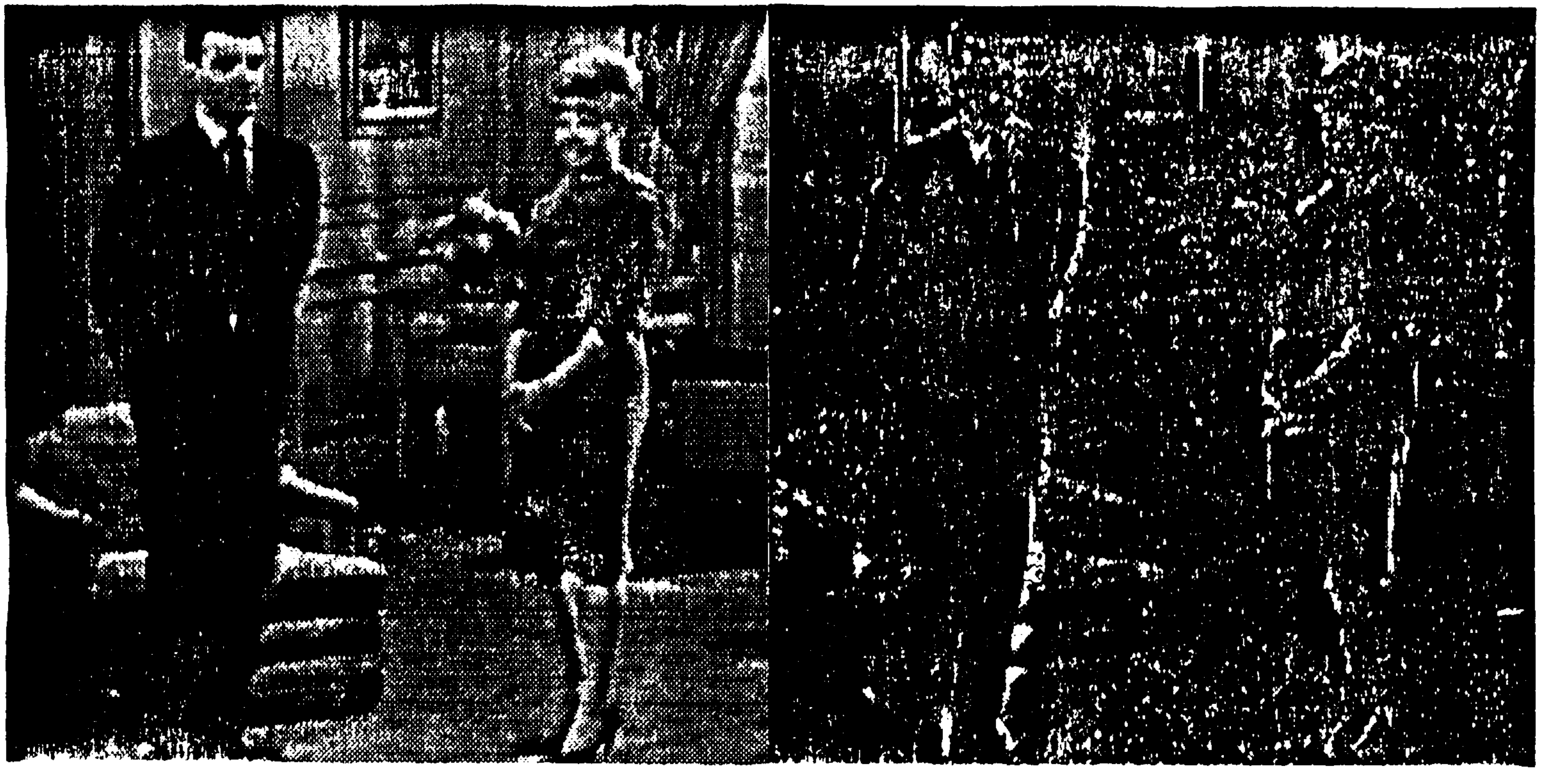
Resultant images and error images of image02 obtained at four different network loads are shown in figure 6.3. The image and the error image obtained without any side information is shown in figure 6.3a for comparison. The image shown in figure 6.3b, is obtained by sending side information in the areas of woman's hands, woman's left knee and in some portion of their heads. In figure 6.3 c, the edge errors are corrected in the region of woman's hands, man's shoulder, photo frame and their heads. Some of the edge errors at the man's leg are also corrected. But in this case, edge errors are present at the woman's left knee. In the image shown in figure 6.3d, all the edge errors are corrected except few errors at the intersection of man's leg and chair. Whereas for the image shown in figure 6.3e, all the side information is sent and there are absolutely no errors.

Table 6.1 Results obtained at different network loads for image02.

S. No.	Load	Delay msec	Throughput	Rate bpp	SNR db	PSNR db
1	.226	177.3	0.8033735	2.011	19.79	33.33
2	.185	147.8	0.8033297	2.014	19.82	33.36
3	.156	123.3	0.8032793	2.057	21.22	34.76
4	.136	105.3	0.8032306	2.126	22.15	35.69
5	.119	89.19	0.8030134	2.221	22.51	36.05
6	.107	73.17	0.8019965	2.223	23.68	37.22
7	.097	58.35	0.8000950	2.237	24.05	37.59
8	.085	42.82	0.7967703	2.238	24.08	37.62
9	.081	38.87	0.7956204	2.238	24.08	37.62

Table 6.2 Results obtained at different network loads for image05.

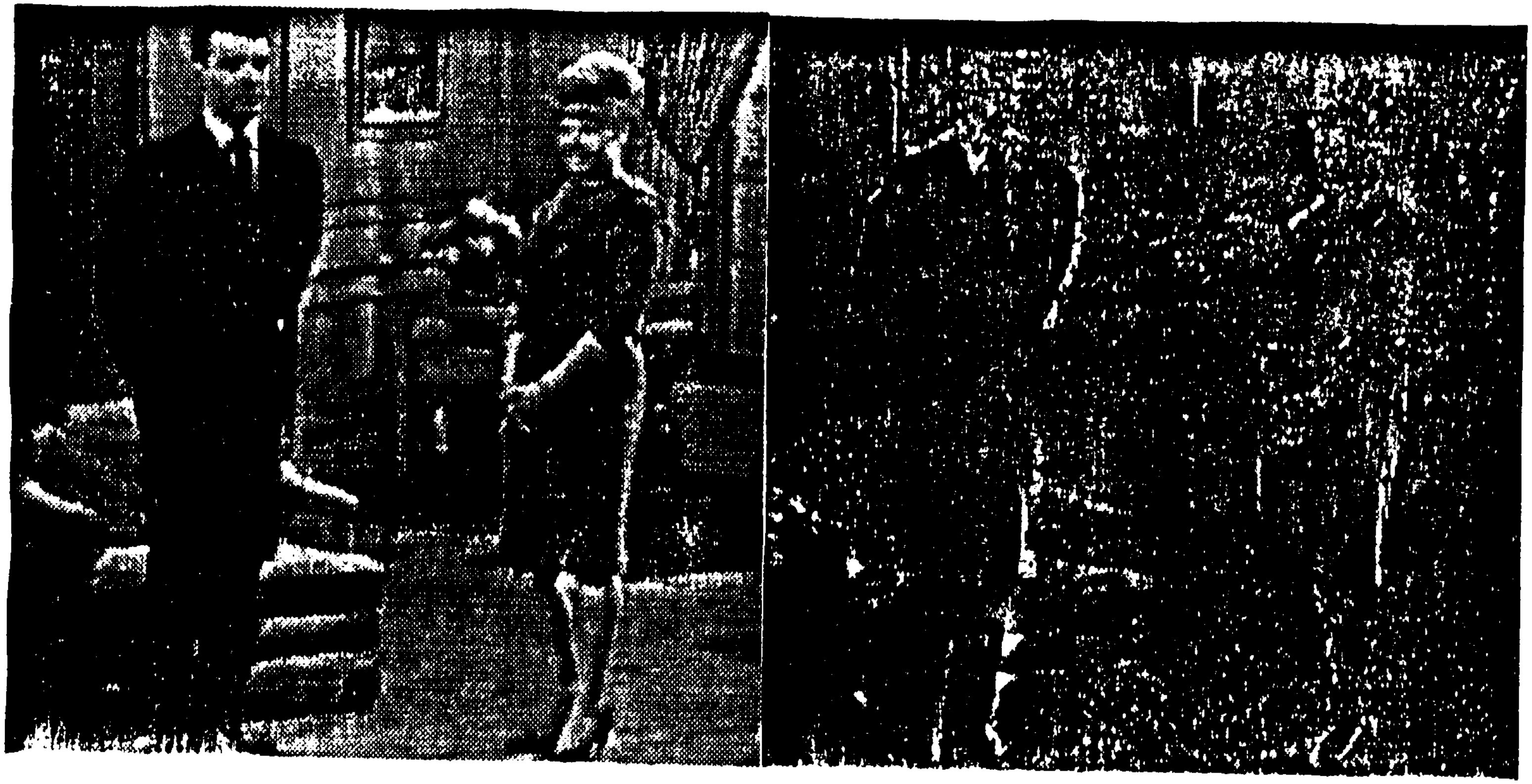
S. No.	Load	Delay msec	Throughput	Rate bpp	SNR db	PSNR db
1	.254	194.1	0.8033794	2.002	24.39	29.13
2	.169	135.3	0.8033345	2.016	24.47	29.22
3	.156	123.6	0.8033151	2.039	24.58	29.32
4	.145	114.36	0.8033025	2.075	25.03	29.78
5	.127	97.52	0.8032005	2.183	25.93	30.68
6	.107	70.32	0.8019068	2.227	26.16	30.90
7	.092	51.79	0.7989234	2.237	26.28	31.02
8	.081	38.42	0.7954938	2.237	26.28	31.02



Resultant image

Error image

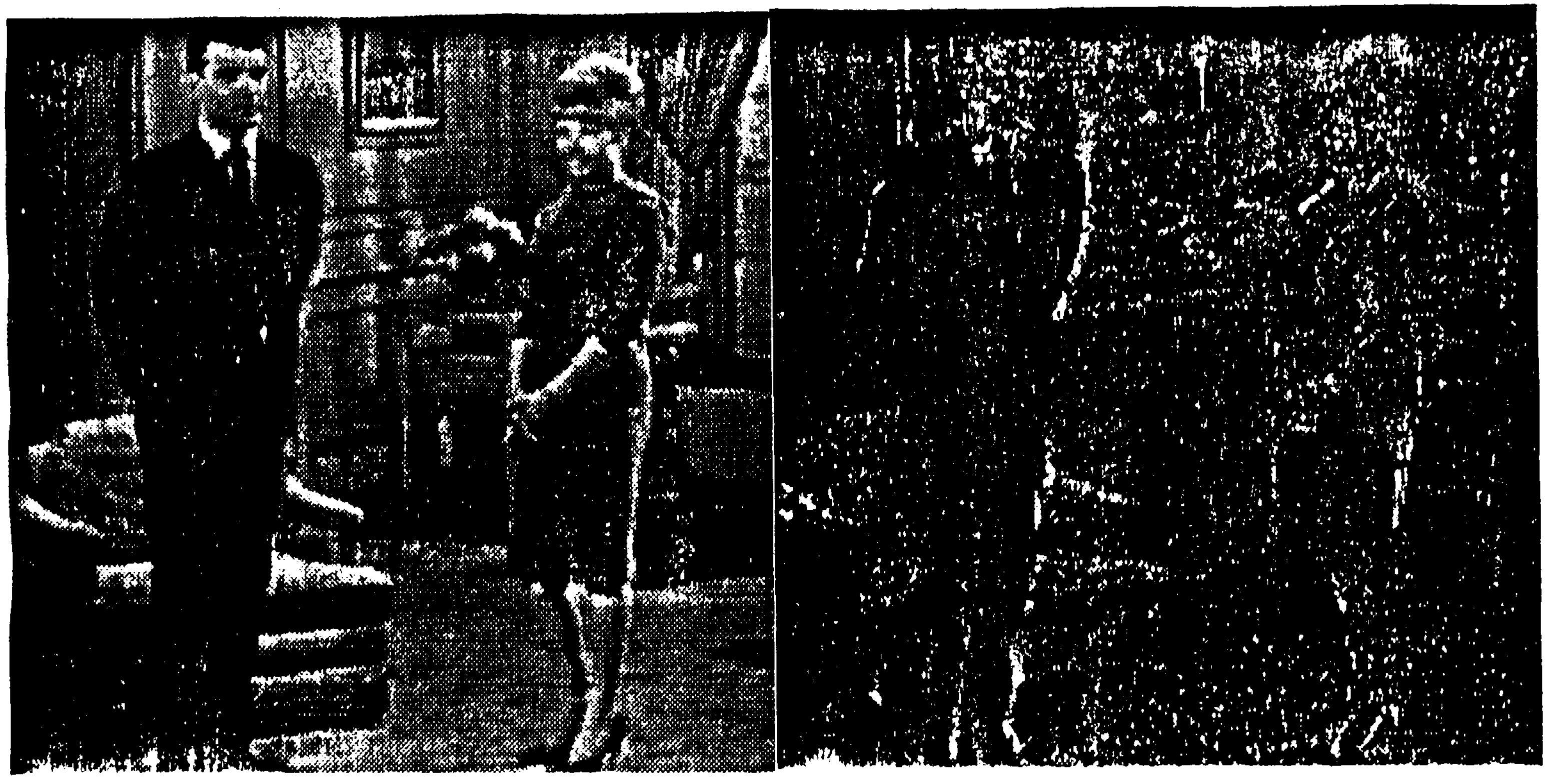
Figure 6.3a Resultant image and error image of Image02 at rate=2.0, SNR=19.72dB



Resultant image

Error image

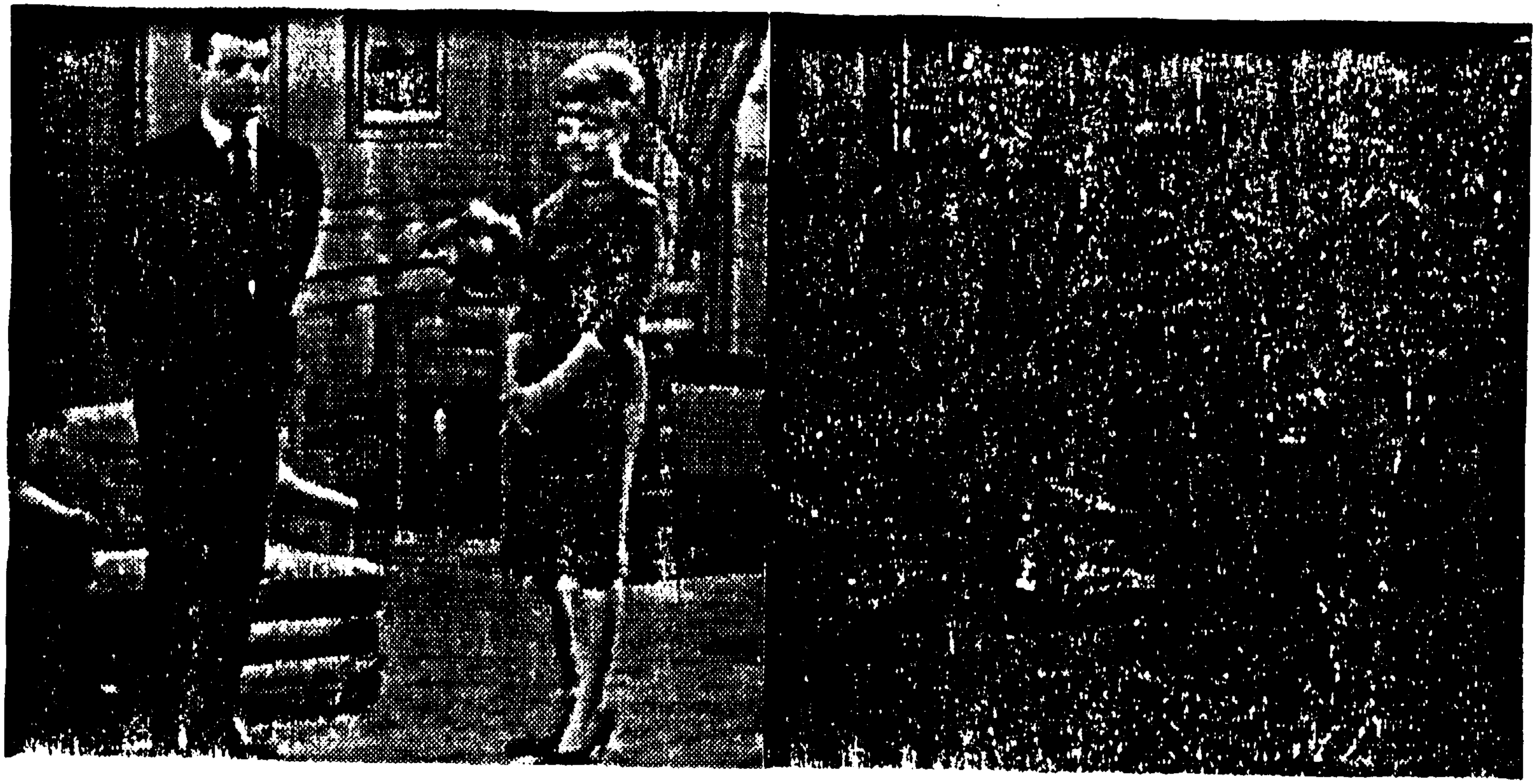
Figure 6.3b Resultant image and error image of Image02 at rate=2.08, SNR=20.92 dB



Resultant image

Error image

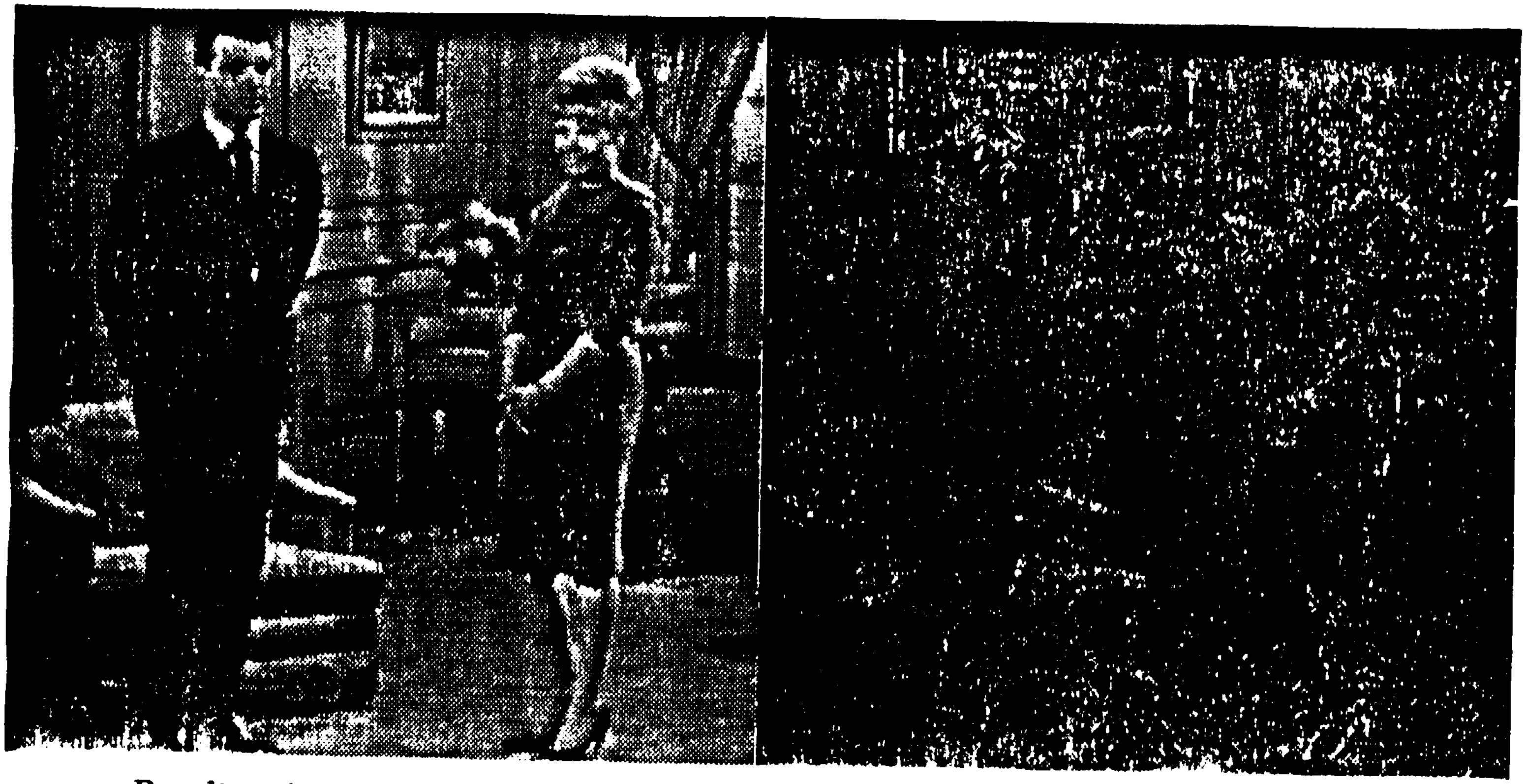
Figure 6.3c Resultant image and error image of Image02 at rate=2.13, SNR=22.15 dB



Resultant image

Error image

Figure 6.3d Resultant image and error image of Image02 at rate=2.23, SNR=23.44 dB



Resultant image

Error image

Figure 6.3e Resultant image and error image of Image02 at rate=2.24, SNR=24.08 dB

6.4 Introduction to Protocol with reference to buffer occupancy:

A free token will be rotating around the token ring whenever all the stations are idle. A node wanting to send a message captures this free token, changes it to the busy token and starts transmitting the packet. After completion of the packet transmission, the node sends a free token to the next node. So a node can send only one packet with each token captured and the other packets will be waiting in the queue. The size of the buffer at each node is finite. It can accommodate only a certain number of packets and packets coming after that will be lost. To avoid that kind of situation, the network must operate at loads where the maximum number of packets waiting in the queue matchs the buffer capacity, i.e. in the stable region. To find this region, load versus delay characteristics are drawn in figure 6.4. It can be seen from the graph that at high loads or when the traffic is busy, the delay increases enormously which can not be tolerated in real time.

In timed token protocol, we had two classes of traffic; asynchronous and synchronous. Asynchronous messages do not have timing constraints and can be transmitted only when the bandwidth becomes available. Now consider a case where all the traffic belorgs to only one class, but the size of the packet can be changed depending upon the available capacity. The amount of information sent in each time is fixed, but only the length of the packet is variable. It should be able to take advantage of increased capacity when it becomes available and to decrease the size of the packet when the available

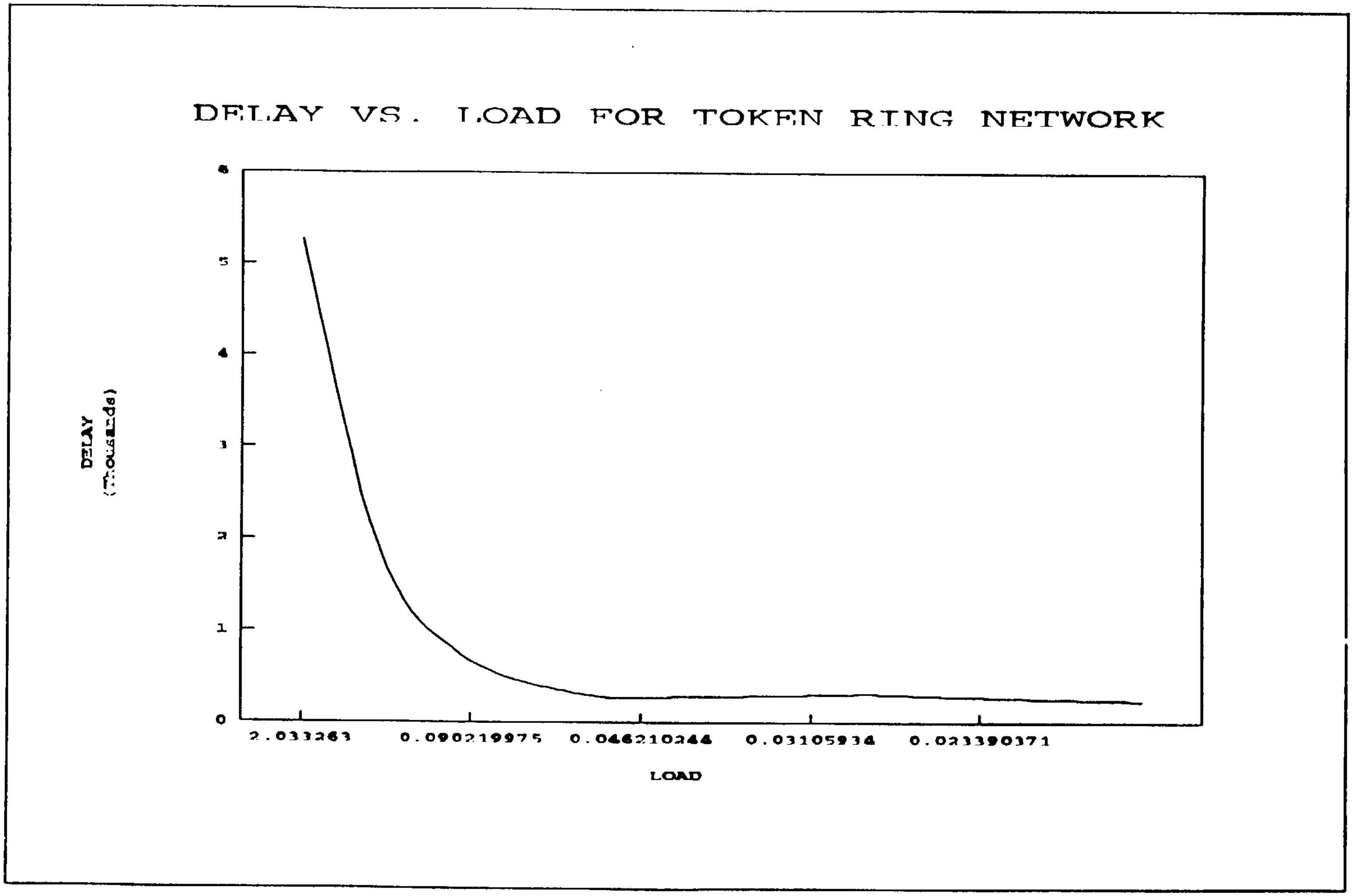


Figure 6.4 Delay versus Load Characteristics for the Token Ring Network

capacity decreases But the data generator while generating the packet itself needs to know about the bandwidth available for that packet. Therefore, it should predict the available capacity for the future coming packets depending on the present traffic conditions. Because once the packet is processed, the system can not afford to lose any information.

One simple way to predict available bandwidth for the future is to compare the buffer occupancy. The number of packets waiting in the queue is directly related to the traffic because a node can send only one packet with each token capture independent of how many packets are waiting in the queue. Therefore, depending upon the buffer occupancy, the size for the next coming packet can be decided. We can fix thresholds for the buffer, with which at different levels of occupancy the data generator will generate packets with different lengths. The edge preserving scheme based on this principle will be simulated in the next section.

6.5 Simulation of an Edge Preserving Scheme:

This scheme is simple, easy to implement in real time and has excellent edge preservation properties. This can be operated either at a fixed rate or at variable rates.

The stepsize of the quantizer can be varied to get the desired bit rate. When the quantizer input falls in the inner levels, the rate obtained is fixed. For a fast changing input signal, the prediction error falls in the overload region, which usually happens when it encounters an edge in the image. When the quantizer input falls in the outer levels which are in the overload region, the error will be encoded repeatedly until it falls in one of the inner levels of the quantizer. The bit rate corresponding to this rapidly changing input is variable and is higher than the usual rate.

This scheme would be especially useful in transmitting images over channels where the available bandwidth is variable. This makes it attractive for use in the packet network environment.

Like the previous simulation, this scheme is implemented on a token ring network by taking the image information at node 1 and random information at all other nodes. In this simulation, the number of pixels per packet at node 1 is fixed, whereas the length of the packet is varied. It is assumed that the length of the packet is fixed at all the nodes except at node 1. In this thesis seven different thresholds are fixed for the buffer occupancy. The data generator, before processing the packet, will observe the buffer occupancy at that node and then read the length of the packet corresponding to that level of occupancy. Then select the bit rate to match the packet length. This is the length of the packet which can be transmitted in the available bandwidth.

Once the packet arrives, the length of the packet can not be changed and also the system can not afford to lose any information. It is assumed that no information is lost, so all the packets coming after the seventh threshold of buffer occupancy are processed at a rate cf 1.0 bpp and stored in the buffer.

With each packet, the stepsize of the quantizer chosen to process that packet is also sent which gives an overhead of three bits. The first three bits in the information of a frame represents the stepsize. At the receiver, the image will be reconstructed with the corresponding stepsize. The network parameters used in this simulation were the same as the parameters used in the edge correcting scheme simulation.

6.6 Results for an Edge Preserving scheme:

Transmission of Image29, which is the Tiffany image and image35, which is the Lena image, was simulated. Results were obtained by considering the buffer thresholds and the corresponding rates given in table 6.3. The Tiffany image is a face with little detail while the Lena image has extensive edge information.

The graph of signal to noise ratio versus network load is drawn in figure 6.5 for the Tiffany and Lena images. The shape of the curve is the same for both the images. But the Tiffany image which contains less detail provides better performance than the image Lena which contains more edges. At low loads, high SNR values can be obtained for either image, but at these loads the throughput of the network would be low. At moderate loads, the SNR value varies depending upon the load. It is quite obvious that at high loads, i.e. in the unstable region, we can only transmit the image with low rates, because

Table 6.3 Buffer thresholds and corresponding packet lengths.

Buffer Occupancy bits	Packet Length bits
0 - 2999	1632
3000 - 5999	1509
6000 - 8999	1078
9000 - 11999	978
12000 - 14999	878
15000 - 17999	809
≥ 18000	770

there will always be numerous packets waiting in the queue. Since the shapeof this curve is independent of an image for the given network, the operating load for the network can be chosen depending upon the requirement.

The same image is transmitted through the network for about ten times, and its performance is studied. When the network is operated in the unstable region, the bit rate achieved for an image is usually low. Even if initially the network allows node 1 to transmit the image at reasonably good rates, after two or three runs the rate will fall down to the minimum value which is 1.0 bpp in this thesis. But when the network is operated in the stable region, the SNR value fluctuates around the average value.

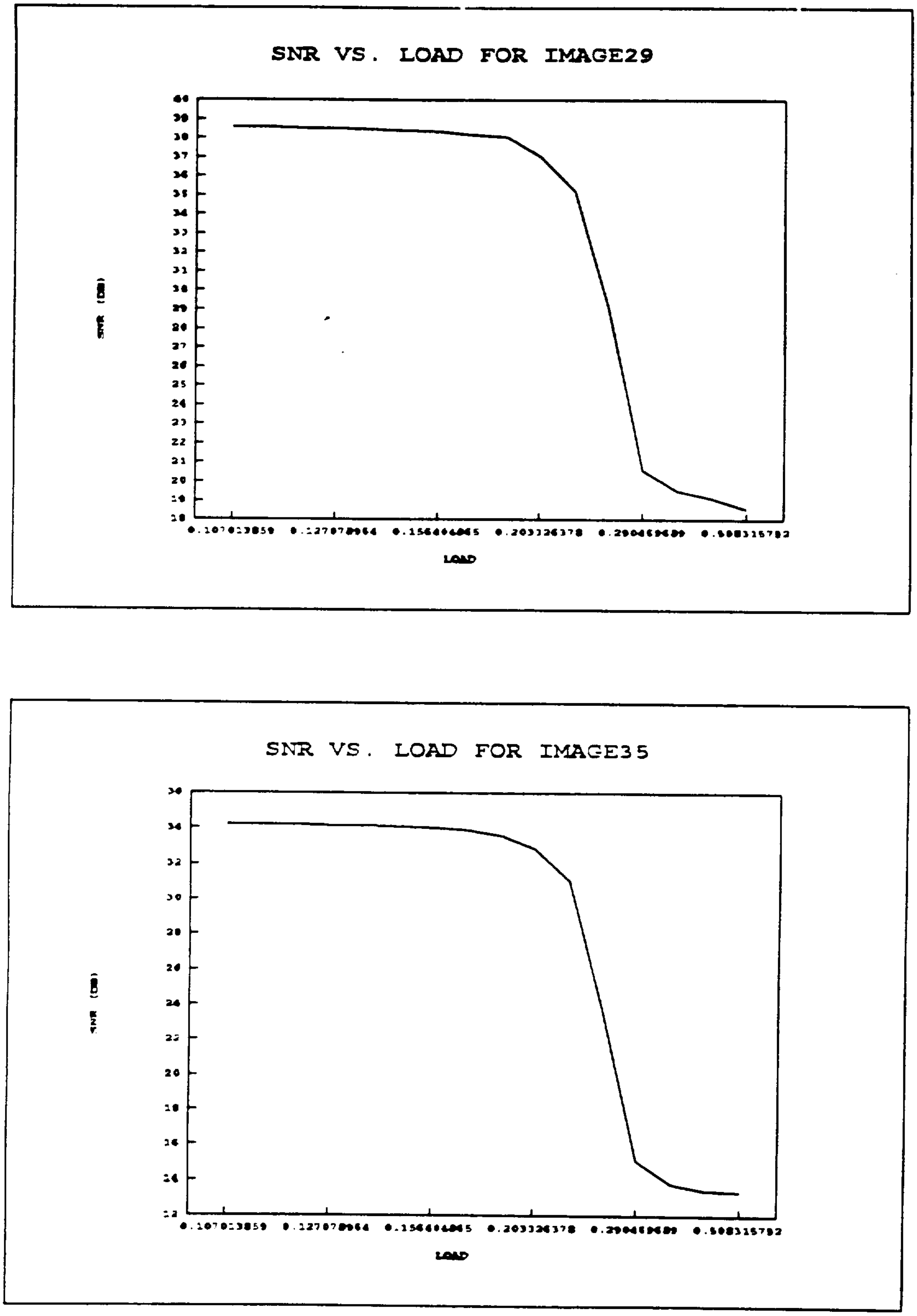


Figure 6.5 SNR versus Network Load for the reconstructed image

For the Tiffany image, the graphs of SNR versus packet number and delay versus packet number for three runs are drawn in figure 6.6 and the corresponding resultant images are shown in figure 6.7. The results are tabulated in table 6.4 for both the Tiffany and Lena images. It clearly shows that there is peak in delay characteristics corresponding to the crest in the SNR characteristics and vice versa. From this, we can say that the assumption we made in the description of the protocol, that the packet length or the rate is chosen depending on the buffer occupancy, is satisfied.

If we compare the delay characteristics of Tiffany image with the delay characteristics of other images for the same network conditions they are exactly the same. SNR characteristics are a little different, because they are not only a function of network parameters but also of the packet in the image.

The Edge Correcting scheme and the edge preserving scheme have been successfully implemented on a token ring network. In the simulation of an edge correcting scheme, it is assumed that TTRT is same as the time taken by a token to return to the same node when every node has a packet to send. Depending upon the application and network traffic TTRT can be changed.

In the simulation of an edge preserving scheme, thresholds for the buffer occupancy and corresponding packet lengths are chosen based on the required performance of an image. In both the cases, though different regions in an image are transmitted at different rates, the reconstructed image looks almost uniform.

Table 6.4 Results at a network load of 0.24 while sending the same image.

Run #	Image29			Image35		
	Rate bpp	SNR (db)	PSNR (db)	Rate bpp	SNR (db)	PSNR (db)
1	1.759	33.11	40.78	1.733	28.31	41.47
2	1.587	32.01	39.68	1.599	26.31	39.47
3	1.926	35.52	43.18	1.933	30.90	44.06
4	1.819	35.32	42.99	1.837	31.10	44.26
5	1.965	37.16	44.82	1.965	32.93	46.09
6	1.858	35.95	43.62	1.856	31.59	44.75
7	1.599	32.75	40.42	1.629	28.59	41.75
8	1.795	35.07	42.74	1.795	30.95	44.11
9	1.660	29.83	37.49	1.598	23.86	37.02
10	1.987	37.47	45.14	1.977	33.10	46.27

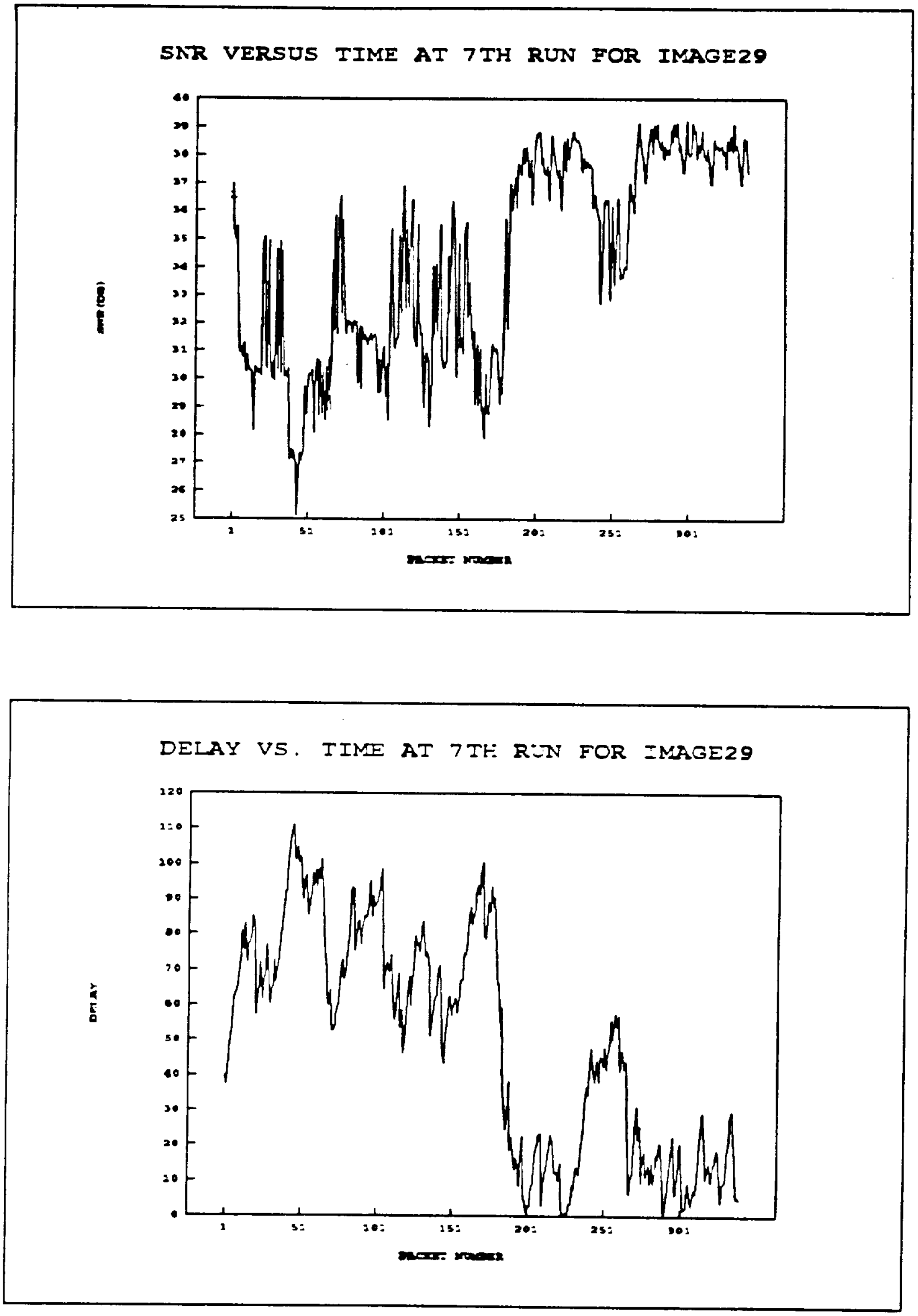


Figure 6.6a SNR vs. Time and Packet Delay vs. Time for image29 (7th run)

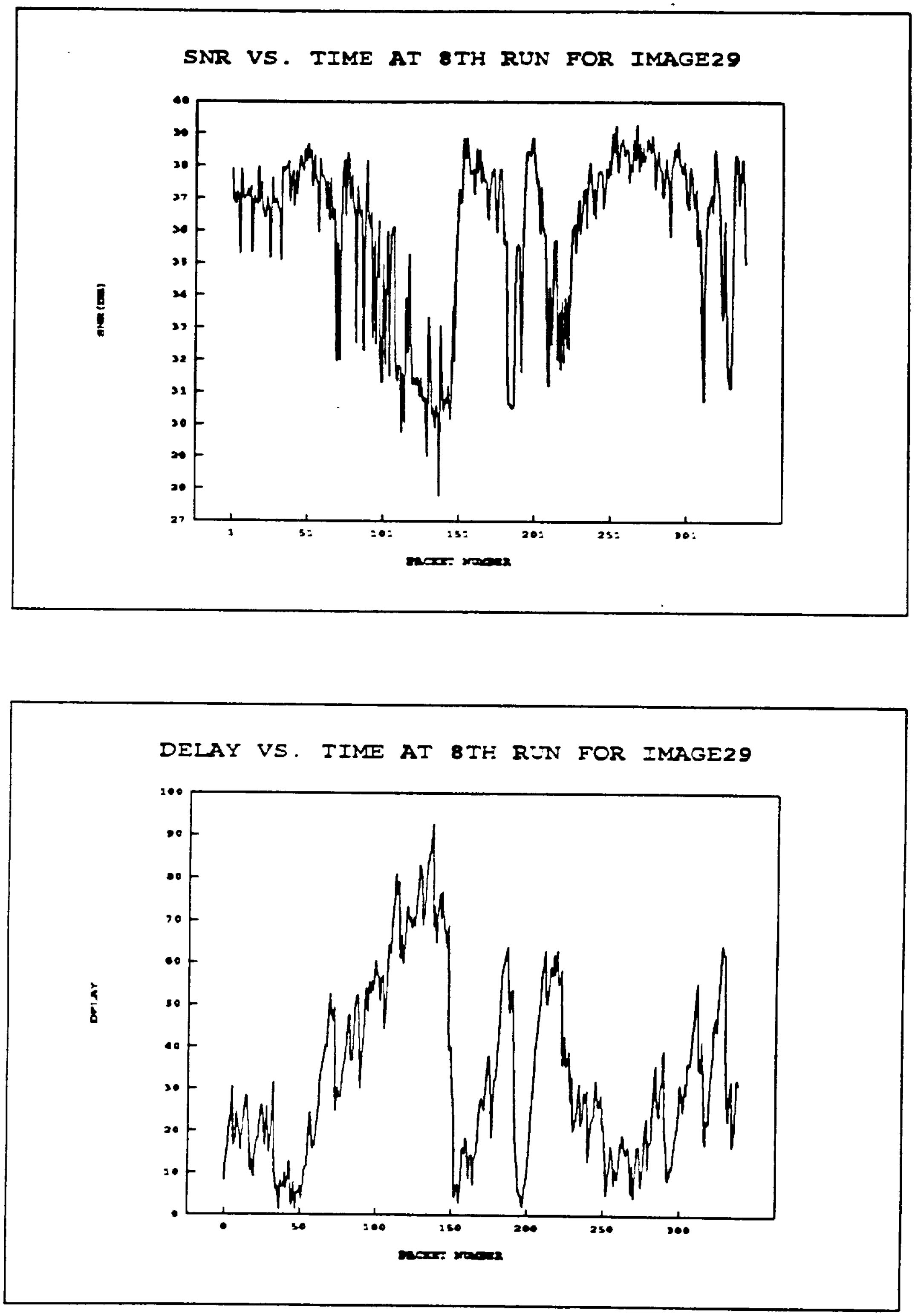


Figure 6.6b SNR vs. Time and Packet Delay vs. Time for image29 (8th run)

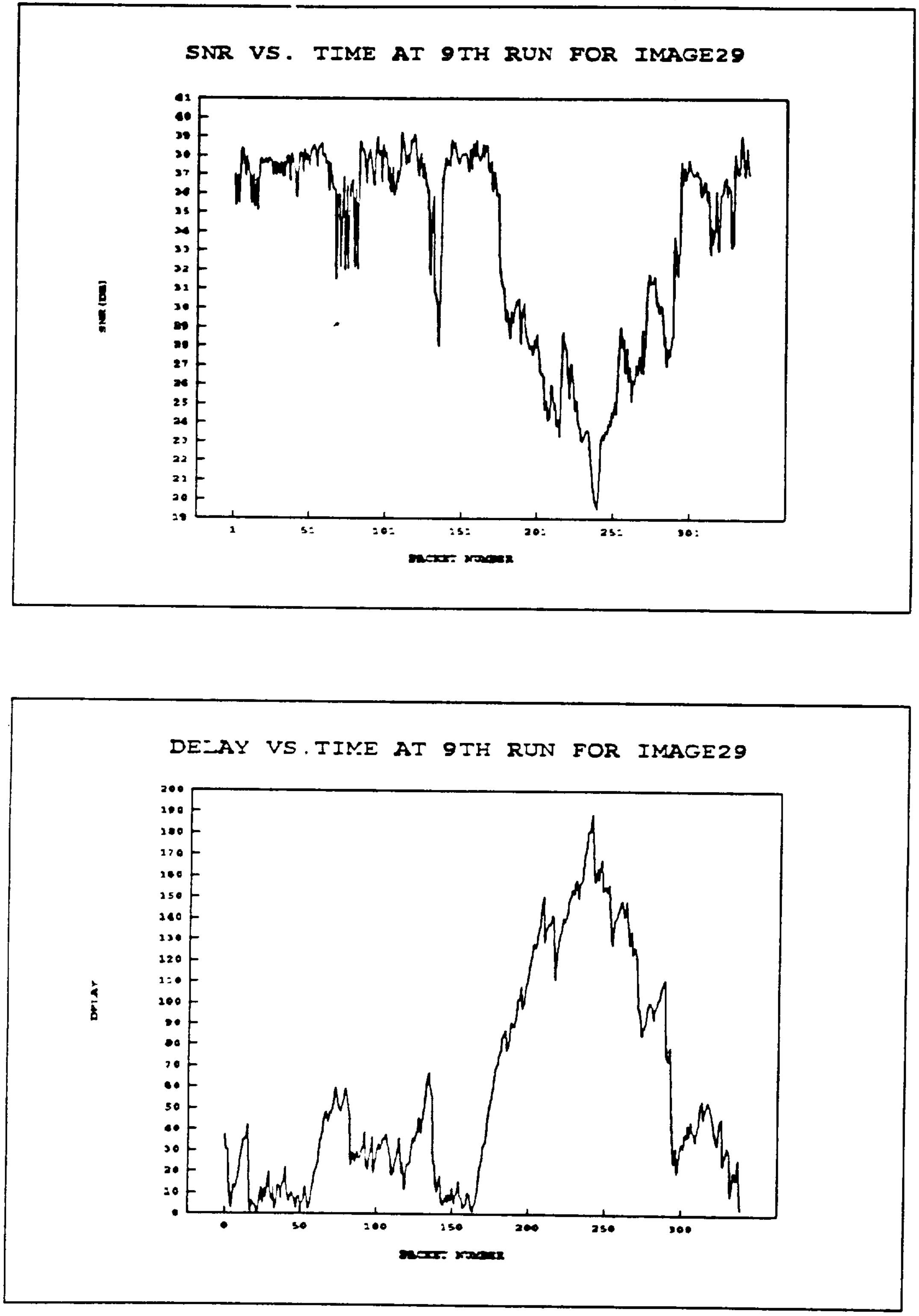


Figure 6.6c SNR vs. Time and Packet Delay vs. Time for image29 (9th run)

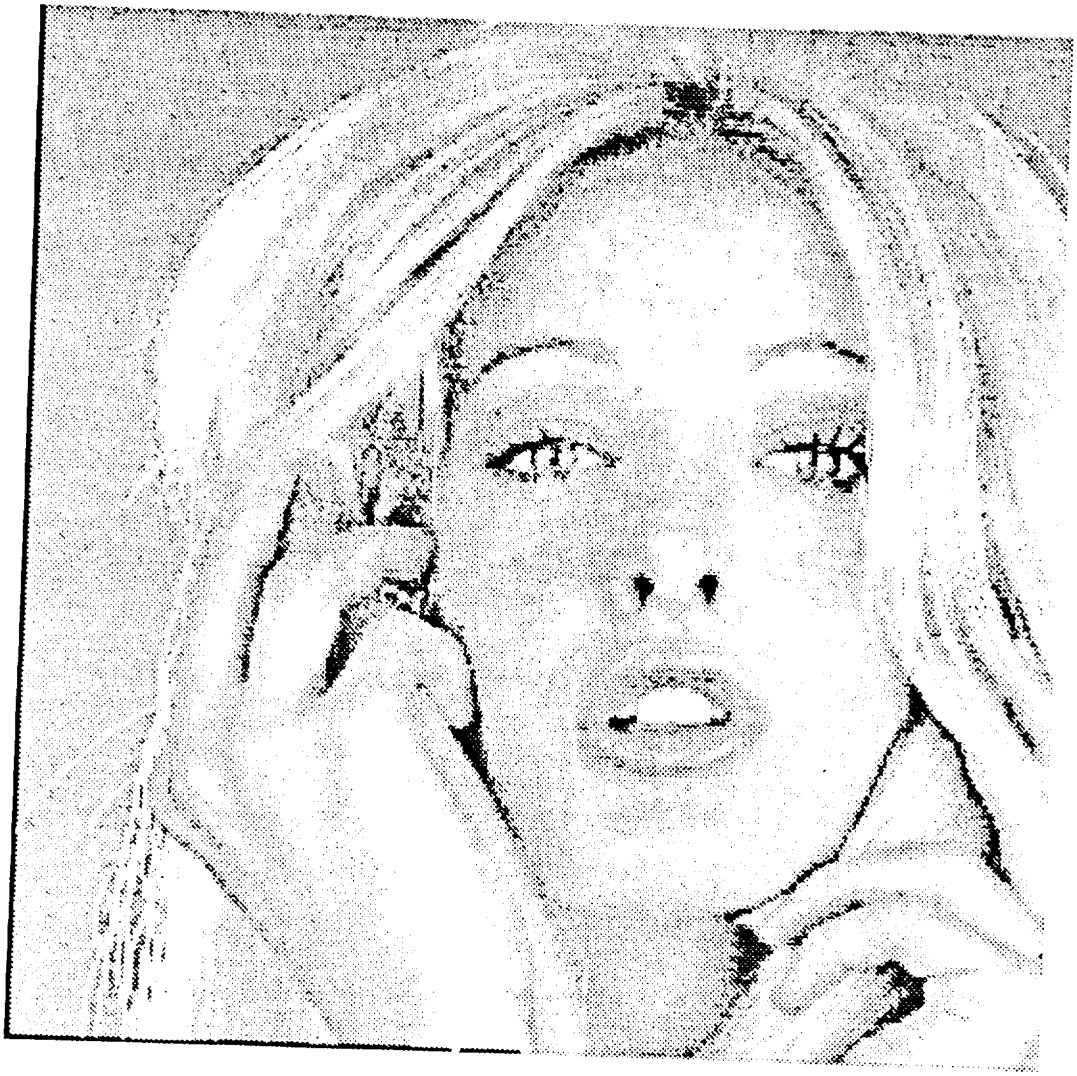


Figure 6.7a Resultant image of Tiffany, rate=1.59, SNR=32.75dB (7th run)

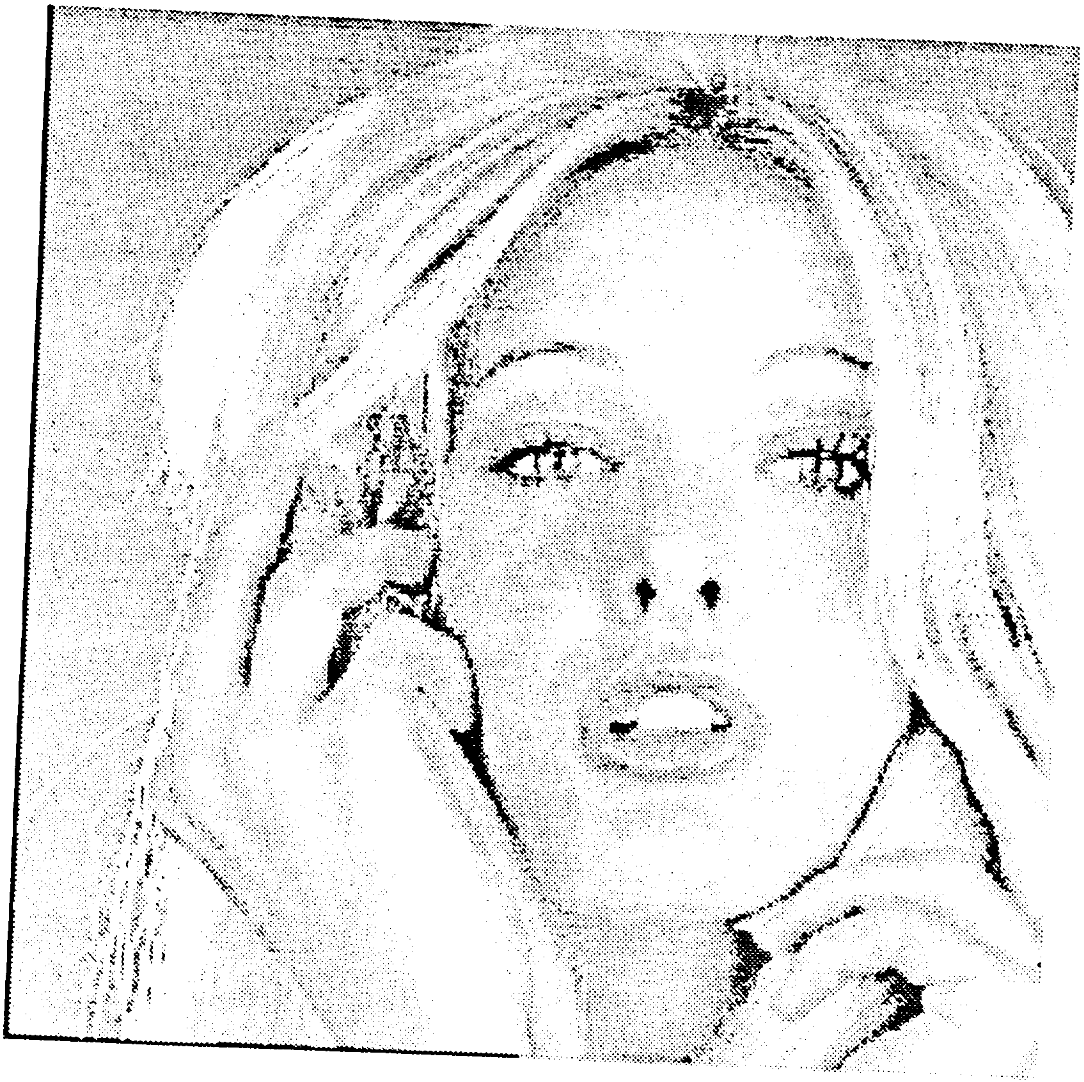


Figure 6.7b Resultant image of Tiffany, rate=1.79, SNR=35.07dB (8th run)

ORIGINAL PAGE IS
OF POOR QUALITY

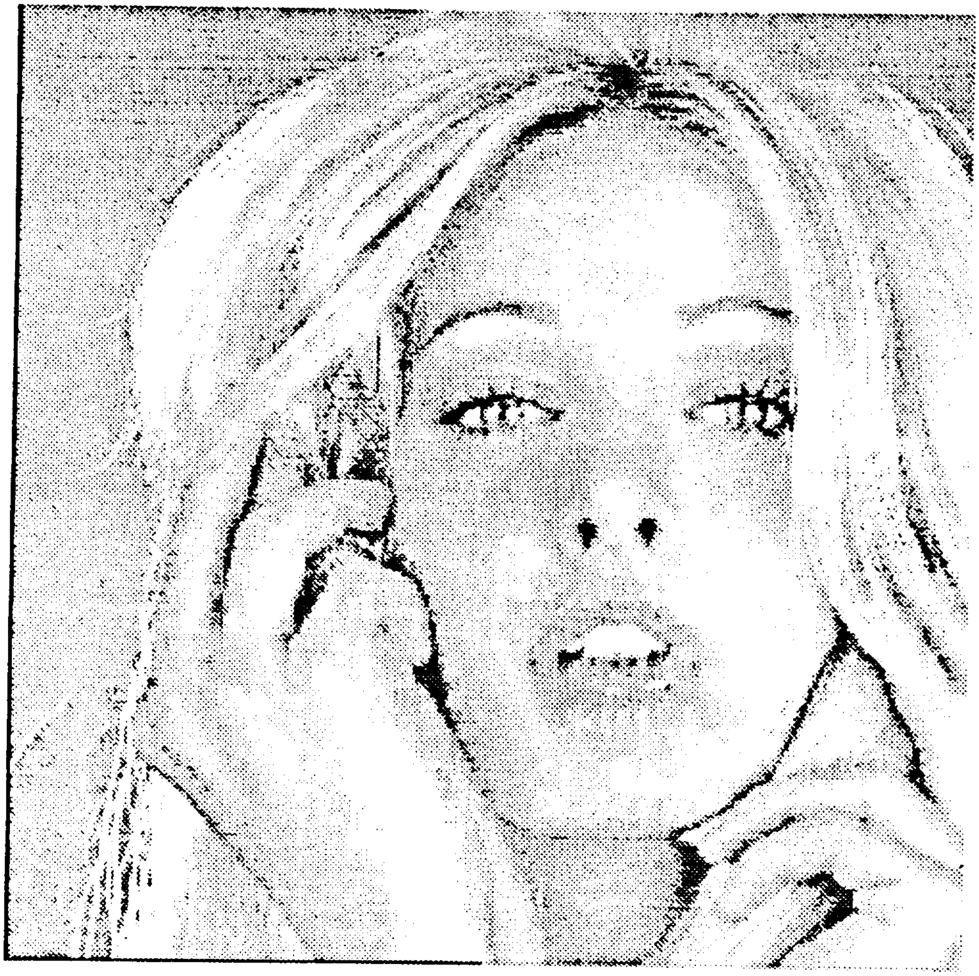


Figure 6.7c Resultant image of Tiffany, rate=1.66, SNR=29.83dB (9th run)

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS

In chapter three, we saw that in an Edge preserving scheme the edge performance is improved by sending the quantizer error to the receiver as extra information. The quantizer error is encoded by using a recursive quantizer with entropy coding. The results show that the system is successful except where fine edges are involved. This is because the edge detector responds for edges which are of at least three pixels wide. This scheme can be added to the existing DPCM systems.

In chapter four, the performance of a DPCM system is enhanced by reducing the overload noise through a scheme known as the edge preservation coding scheme. The sampling rate is increased when the quantizer input falls in the overload region. The results show that this scheme has excellent edge preservation properties over a wide range of bit rates. Even at low rates there are absolutely no edge errors. The bit rate required by this scheme is quite low when compared with an edge correcting scheme to get the same performance. The reason for the difference in the bit rate is that this scheme uses a recursive quantizer unlike the Edge Correcting DPCM which uses a Jayant quantizer. The Edge Preservation coding system is more complex than the other scheme.

The Edge Correcting DPCM and the Edge Preservation schemes requires variable rate channel. In chapter six these two schemes are implemented on a token ring network. The edge information is sent in such a way that it will serve both purposes; (i) to utilize the channel efficiently by sending extra information only when there is bandwidth available (ii) to improve the performance of an image. In the Edge Correcting scheme, whenever the bandwidth becomes available, the recent side information is sent and after that if any side information is left, it is discarded. So, depending on traffic conditions the station or node may discard side information in some regions. Similarly, in the other scheme, the bit rate or the quantizer stepsize for each packet is variable. In both the schemes though the bit rate is variable in different regions of an image the image looks uniform. Implementation of these schemes on a token ring degrades the delay versus load characteristics but improves the delay versus throughput characteristics. These two coding schemes are successfully implemented on a token ring network.

If the network is busy and there are many nodes, the token ring method is better, as it assures each node access within a predetermined time. A token will be rotating on the network from node to node independent of the traffic. If traffic is light at some nodes, the token rotation unnecessarily increases the packet delay at those nodes. The CSMA/CD method is best if the network traffic is low and there are only a few nodes. The users are not waiting for the token to come back around the ring. For further research these schemes can be simulated on a network which uses CSMA/CD method and the performance can be studied.

REFERENCES

[1] S.M. Schekall and K. Sayood, "An Edge Preserving DPCM Scheme for Image Coding," Proc. 31st Midwest Symposium on Circuits and Systems, St. Louis, pp.904-907,Aug. 1988.

[2] Martin C. Rost and Khalid Sayood, "An Edge Preserving Differential Image coding Scheme," Proceedings of the International Conference on Computers and Communications, Scottsdale, AZ, March 21-23, 1990.

[3] ANSI/IEEE std. 802.4-1985 "Token-passing Bus Access Method and Physical Layer Specifications," The Institute of Electrical electronic Engineers, Inc., New York, 1985.

[4] "FDDI Token Ring Media Access Control," draft proposed-ANSI Standard X3T9.5/83-16, rev. 10, Feb. 28, 1986.

[5] N.S. Jayant and P. Noll, "Digital Coding of Waveforms,". Englewood Cliffs, NJ: Prentice-Hall, 1984.

[6] J. Max, "Quantizing for Minimum Distortion," IRE Trans. on Information Theory, pp. 7-12, March 1960.

[7] D. J. Goodman and R. M. Wilkinson, "A Robust Adaptive Quantizer," IEEE Trans. on Communications, pp. 1362-1365, November 1975.

[8] Boneung Koo and J. D. Gibson, "Experimental Comparison of All-Pole, All-Zero, and Pole-Zero predictors for ADPCM Speech Coding," IEEE Trans. on Acoust. Speech and Signal Processing, vol. COM-34, no. 3, March 1986.

[9] Stanley M. Schekall, "Improved Data Compression of Digital Images using Differential Pulse Code Modulation," Master's thesis, Lincoln, NE, May 1987.

[10] Richard W. Hamming, "Coding and Information Theory," Prentice-Hall, NewJersey, 1980.

[11] Dimitri Bertsekas and Robert Gallagu, "Data Networks," Englewood Cliffs, NJ: Prentice-Hall, 1987.

[12] Zsehong Tsai and Izhak Rubin, "Analysis for Token Ring Networks Operating Under Message Priorities and Delay Limits," IEEE Trans. on Communications, September 1989.

[13] H. Takagi, "Mean Message Waiting Times in Symmetric Multi-Queue Systems with Cyclic Service," Performance Evaluation, no. 5, pp. 271-277, 1985.

[14] S.W. Fuhrmann, "Symmetric Queues Served in Cyclic Order," Operations Research Letters, vol. 4, no. 3, pp.139-144, Oct. 1985.

[15] A.P. Jayasumana, "Performance Analysis of a Token Bus Priority Scheme," Proc. IEEE INFOCOM, pp 46-54, March 1987.

[16] A.P. Jayasumana and G.G. Jayasumana, "Simulation and Performance Evaluation of 802.4 Priority Scheme," Proc. IEEE-ACM Symposium on the Simulation of Computer Networks, pp 232-238, August 1987.

[17] A. Valenzano, Montuschi and L. ciminiera " On the Behavior of Control Token Protocols with Asynchronous and Synchronous Traffic," Proc. IEEE INFOCOM, September 1989.

[18] K.C. Sevcik and M.J. Johnson, "Cycle Time Properties of the FDDI Token Ring Protocol," IEEE Trans. Software Eng., vol. SE-13,No.3, pp. 376-385, March 1987.

APPENDIX A

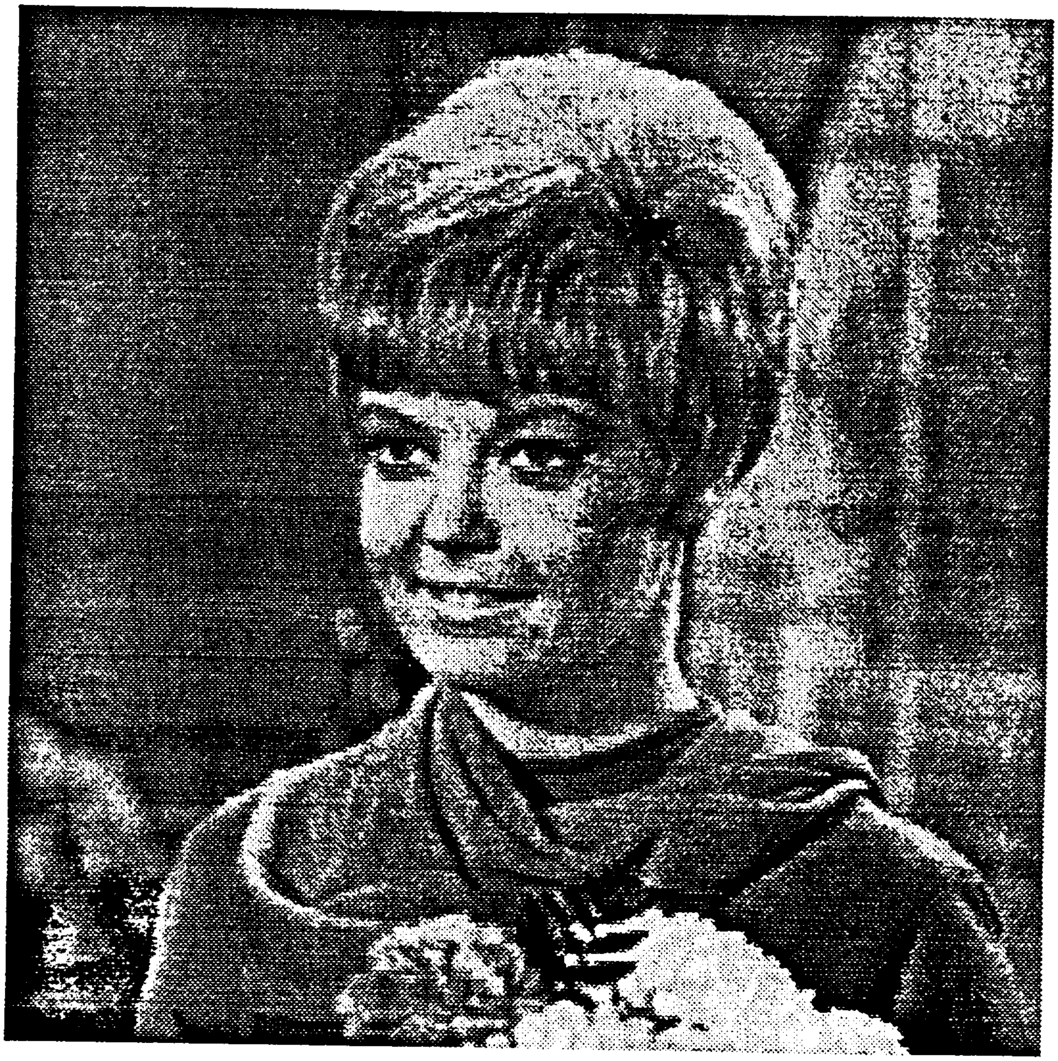


Figure A.1 Original image of Image01 (Girl256)

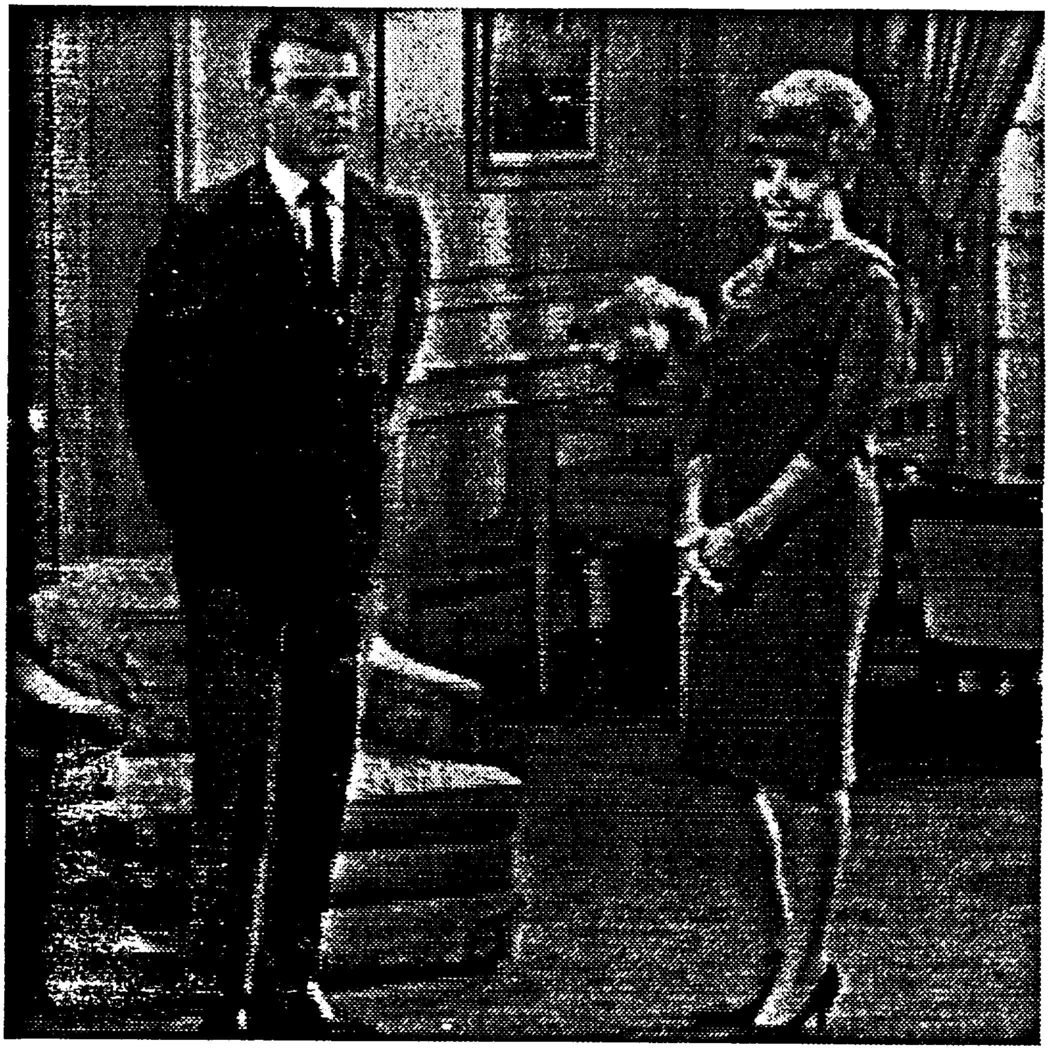


Figure A.2 Original image of Image02 (Couple256)

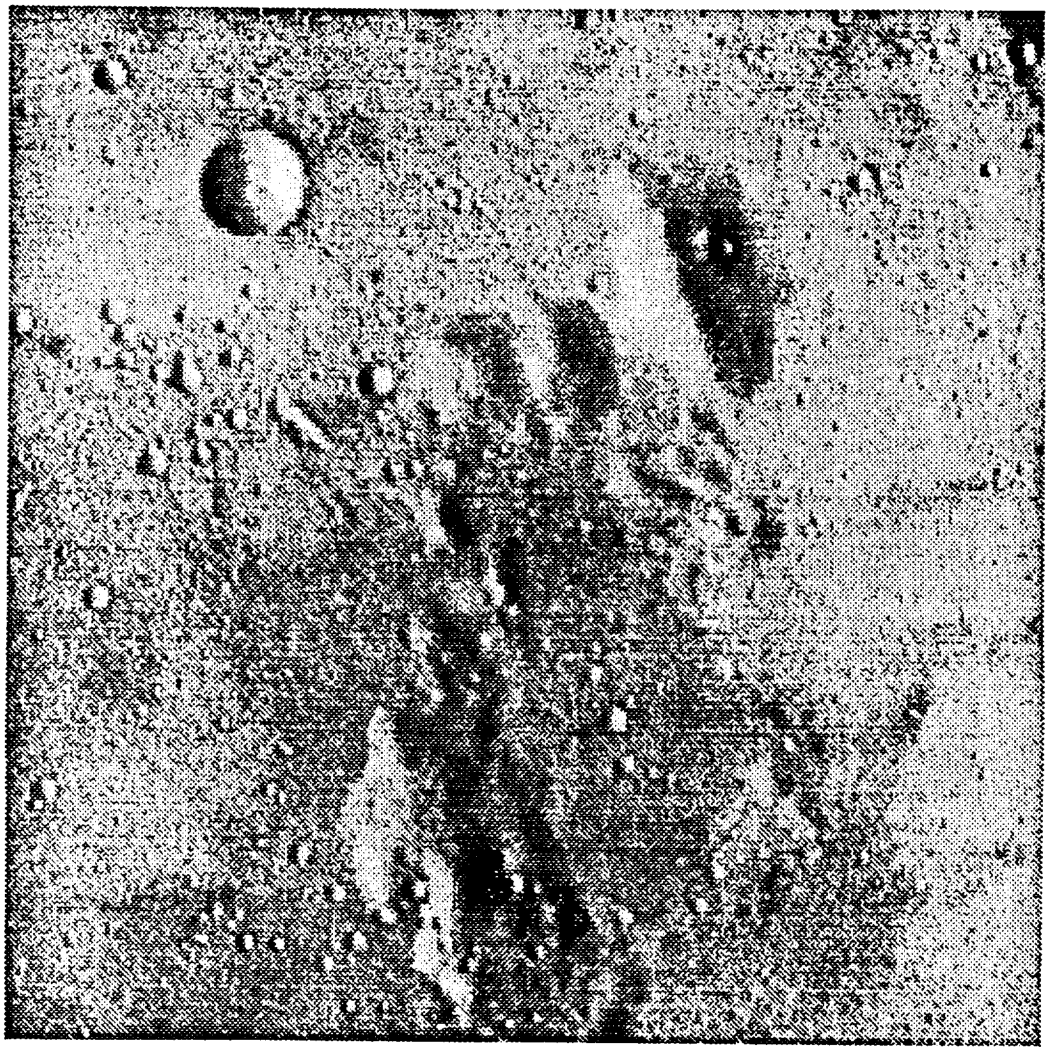


Figure A.3 Original image of Image03

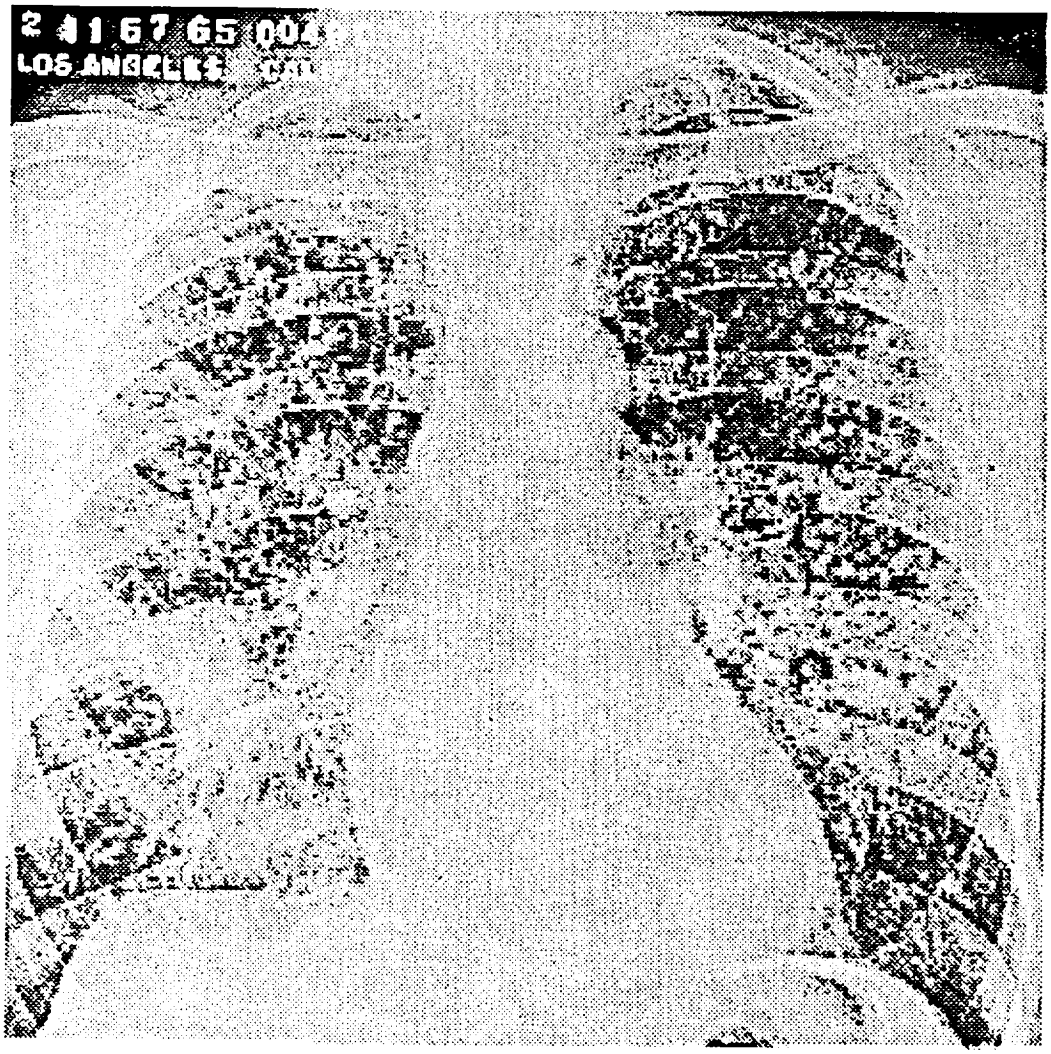


Figure A.4 Original image of Image04

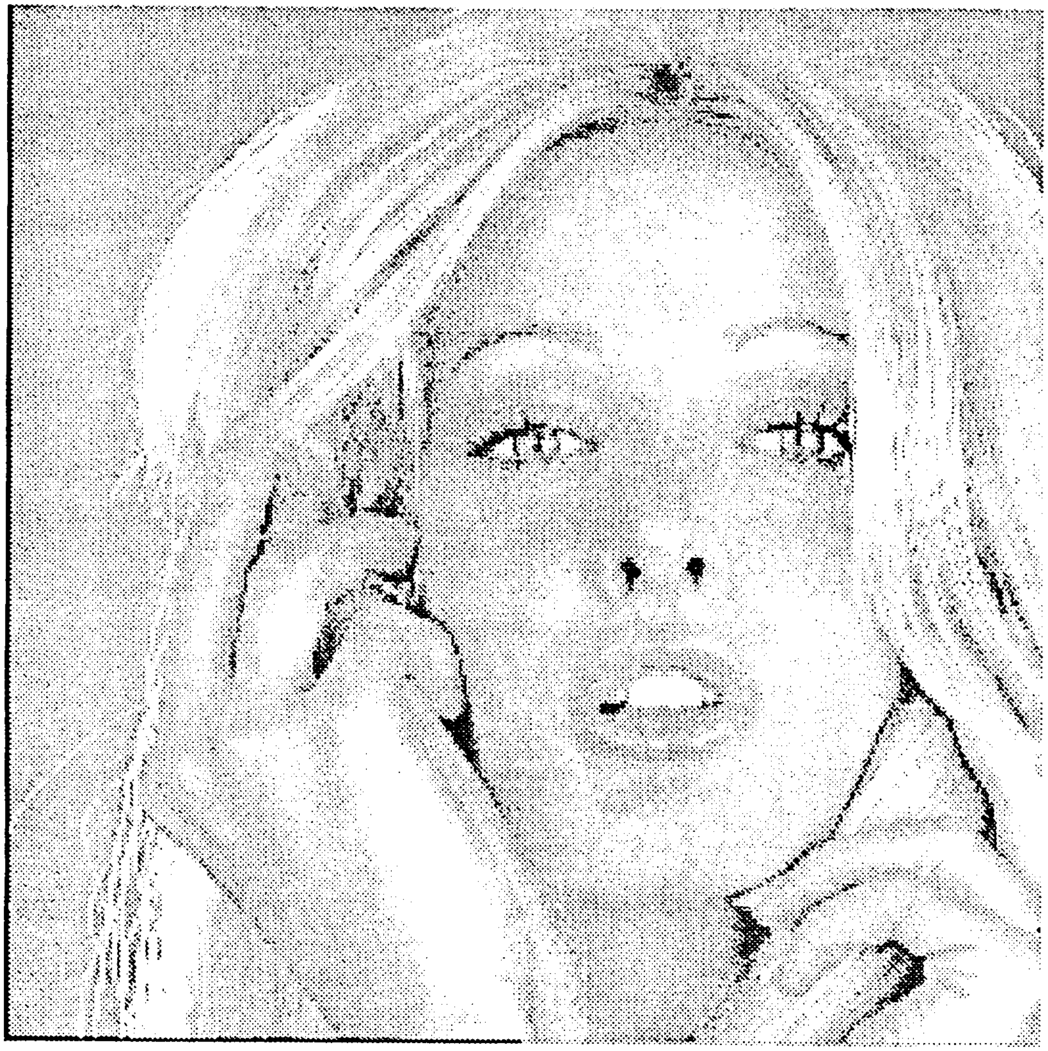


Figure A.5 Original image of Image29 (Tiffany)

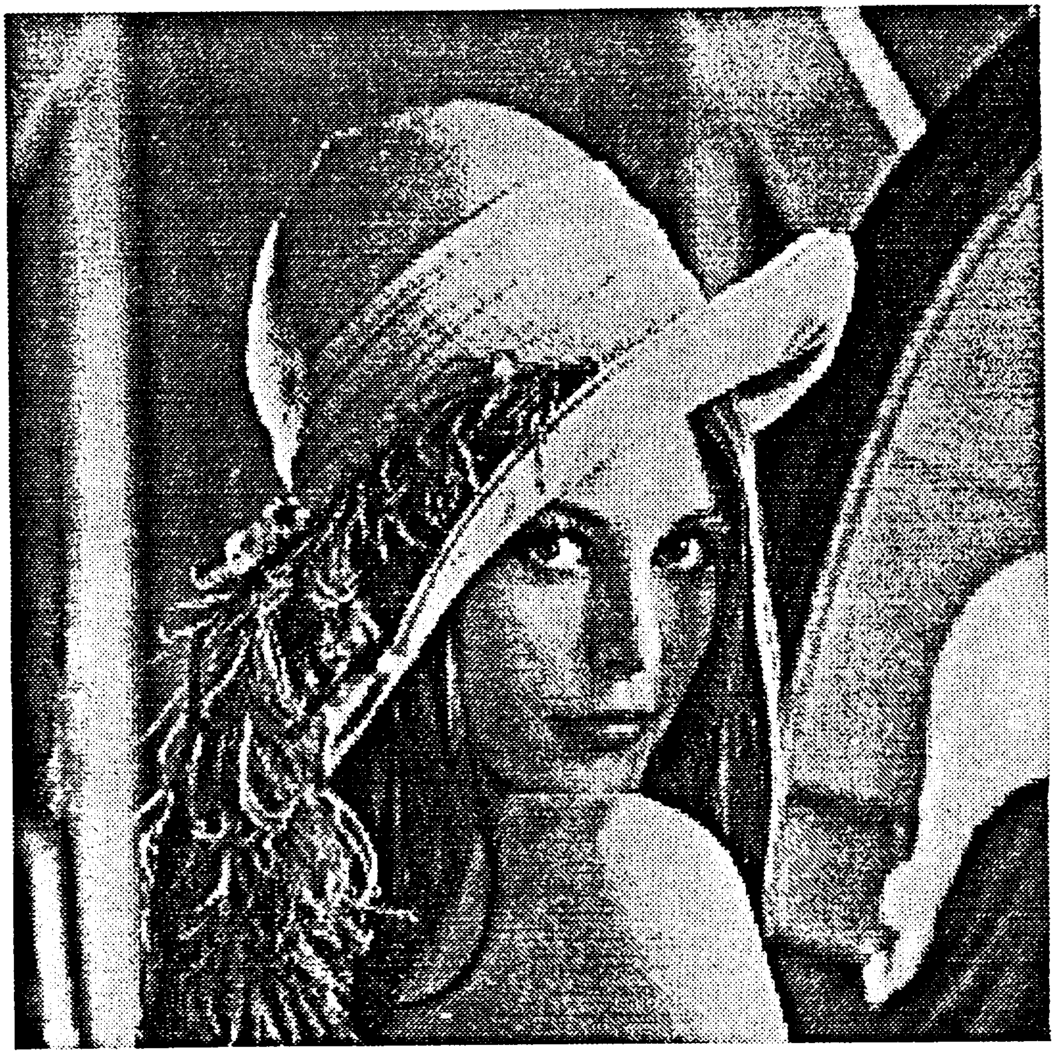


Figure A.6 Original image of Image35 (Lena)