AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

EC8002 – MULTIMEDIA COMPRESSION & COMMUNICATION

UNIT I – Audio Compression

Part A

1. What are the benefits ofCompression?

- It provides less disk space (i.e) more data in real time.
- Faster file transfer.
- Faster writing and reading.

2. Define frequency masking.

Auditory masking occurs when the perception of one sound is affected by the presence of another sound. Auditory masking in the frequency domain is known as simultaneous masking, frequency masking or spectral masking.

3. What is the principle of adaptive predictive coding?

By varying the number of bits used for difference signals based on its amplitude (i.e) using few bits to encode small difference values can save bandwidth and also improve quality. This is the principle of Adaptive Differential Pulse Code Modulation(ADPCM).

4. What is the basic principle of ADPCM?

By varying the number of bits used for difference signals based on its amplitude (i.e) using few bits to encode small difference values can save bandwidth and also improve quality. This is the principle of Adaptive Differential Pulse Code Modulation (ADPCM).

5. What are the different delays suffered by CELP coders?

There are two delays through which a CELP-Code Excited Linear Prediction coding suffers, they are,

Processing Delay: This delay occurs when each block of digitized samples are analyzed by the encoder & the speech being reconstructed at the decoder.

Algorithmic delay:The time required to accumulate the block of samples is known as Algorithmic Delay.

6. What is the advantage of adaptive predictive coding?

- ❖ APC has a reduced bandwidth up to 8kbps.
- ❖ The quality of the data is maintained, even after compression.

7. What is meant by delta modulation?

Delta modulation is the one-bit version of differential pulse code modulation.

8. Define entropy encoding.

Entropy coding is a type of lossless coding to compress digital data by representing frequently occuring patterns with many bits.

9. Define differential encoding.

Differential encoding is usedinapplicationswheretheamplitudeofavalueor signalcoversalargerangebutthedifference inamplitudebetween successivevaluesor signals is relatively small.Techniquesthattransmit informationbyencodingdifferencesarecalled differential encoding. Differential encoding schemes are very popular for speech coding.

10 Give one application each suitable for lossy& lossless compression.

Compressionofsatelliteimagesisanexampleforlosslesscompression whereas compression of general images(move stills) is good example for lossy compression.

11. Why LPC is not suitable to encode music signal?

Sound obtained by LPC is very synthetic. Hence it is suitable mainly for speech. Since music has high bandwidth, the naturalness of sound is further detoriated. Therefore LPC is not used for music compression.

12. What is perceptual coding?

In perceptual coding only perceptual feature of the sound are stored. This gives high degree of compression. Human ear is not sensitive to all frequencies equally. Similarly masking of weaker signal takes place when louder signal is present nearby. These parameters are used in perceptual coding.

13. What is concept of CELP Principles?

CELP uses more sophisticated model of vocal tract.

Every digitized segment is compared with waveform templates in code book

The matching template is differentially encoded and transmitted.

At the receiver, it is differentially encoded and Codeword selects the matching template from codebook.

Standard audio segments are stored as waveform templates. Encoder and decoder both have same set of templates. It is called codebook.

14. What is Dolby AC – I?

Dolby AC – Iis used for audio recording. It is MPEG audio coding standard. It used psychoacoustic model at the encoder and has fixed bit allocations to each subband.

15. Define algorithm delay.

It is the time required to accumulate each block of samples in the memory.

16. What are voiced and unvoiced sounds?

Voiced Sounds: These sounds are generated through vocal chords. The sounds like 'P', 'A', 'L' etc are voiced sounds.

Unvoiced Sounds: The sounds are generated with open vocal chords are called Unvoiced sounds. Letter like 'S', 'F', 'V' etc are unvoiced sounds.

17. What is pitch?

The pitch of the signal gives an information about fundamental frequency pitch of every person is different. However it is in the similar range for males and some another range for females.

PART B

1. **Discuss the techniques of DPCM with neat diagram.**

Differential pulse code modulation (**DPCM**) is a derivative of standard PCM and exploits the fact that, for most audio signals, the range of the differences in amplitude between successive samples of the audio waveform is less than the range of the actual sample amplitudes. Hence if only the digitized difference signal is used to encode the waveform then fewer bits are required than for a comparable PCM signal with the same sampling rate. A DPCM encoder and decoder are shown in Figure 4.1(a) and a simplified timing diagram for the encoder is shown in Figure 4.1(b).

Essentially, the previous digitized sample of the analog input signal is held in the register (R) – a temporary storage facility – and the difference signal (DPCM) is computed by subtracting the current contents of the register (R_o) from the new digitized sample output by the ADC (PCM). The value in the register is then updated by adding to the current register contents the computed difference signal output by the subtractor prior to its transmission. The decoder operates by simply adding the received difference signal (DPCM) to the previously computed signal held in the register (PCM). Typical savings with DPCM, are limited to just 1 bit which, for a standard PCM voice signal, for example, reduces the bit rate requirement from 64 kbps to 56 kbps.

As we can deduce from the circuit shown in Figure 4.1(a), the output of the ADC is used directly and hence the accuracy of each computed difference signal – also known as the **residual** (signal) – is determined by the accuracy of the previous signal/value held in the register. As we saw in Section 2.2, all ADC operations produce quantization errors and hence a string of, say, positive errors, will have a cumulative effect on the accuracy of

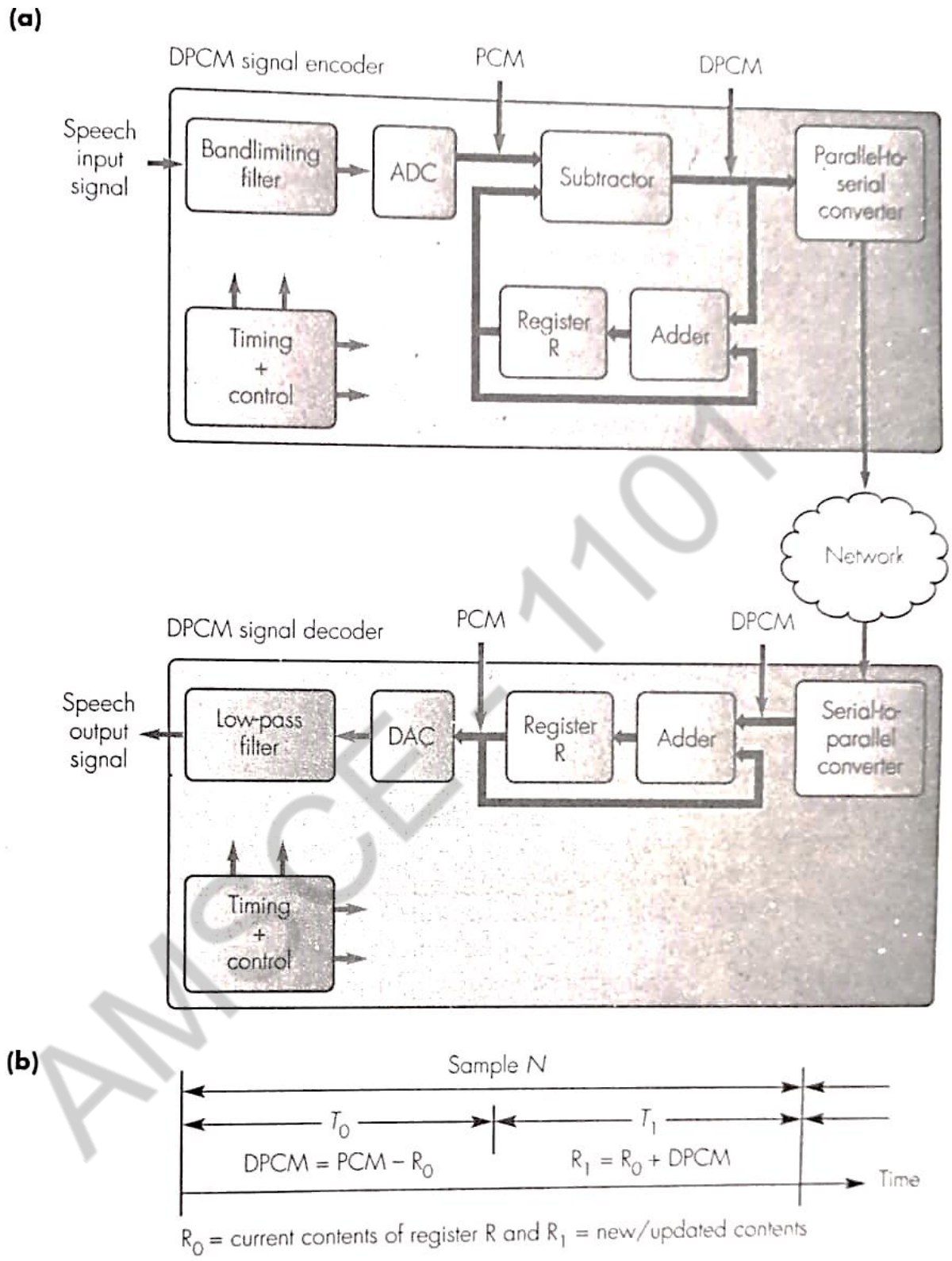


Figure 4.1 DPCM principles: (a) encoder/decoder schematic; (b) encoder timing.

the value that is held in the register. This means, therefore, that with a basic DPCM scheme, the previous value held in the register is only an approximation. Hence more sophisticated techniques have been developed for estimating – also known as predicting – a more accurate version of the previous signal. To achieve this, these predict the previous signal by using not only the estimate of the current signal but also varying proportions of a number of the immediately preceding estimated signals. The proportions used are determined by what are known as **predictor coefficients** and the principle is shown in Figure 4.2.

In this example, the difference signal is computed by subtracting varying proportions of the last three predicted values from the current digitized value output by the ADC. For example, if the three predictor coefficients have the values $C_1 = 0.5$ and $C_2 = C_3 = 0.25$, then the contents of register R1 would be shifted right by 1 bit (thereby multiplying its contents by 0.5) and registers R2 and R3 by 2 bits (multiplying their contents by 0.25). The three shifted values are then added together and the resulting sum subtracted from the current digitized value output by the ADC (PCM). The current contents of register R1 are then transferred to register R2 and that of register R2 to register R3. The new predicted value is then loaded into register R1 ready for the next sample to be processed. The decoder operates in a similar way by adding the same proportions of the last three computed PCM signals to the received DPCM signal. By using this approach, a similar performance level to standard PCM is obtained by using only 6 bits for the difference signal which produces a bit rate of 32 kbps.

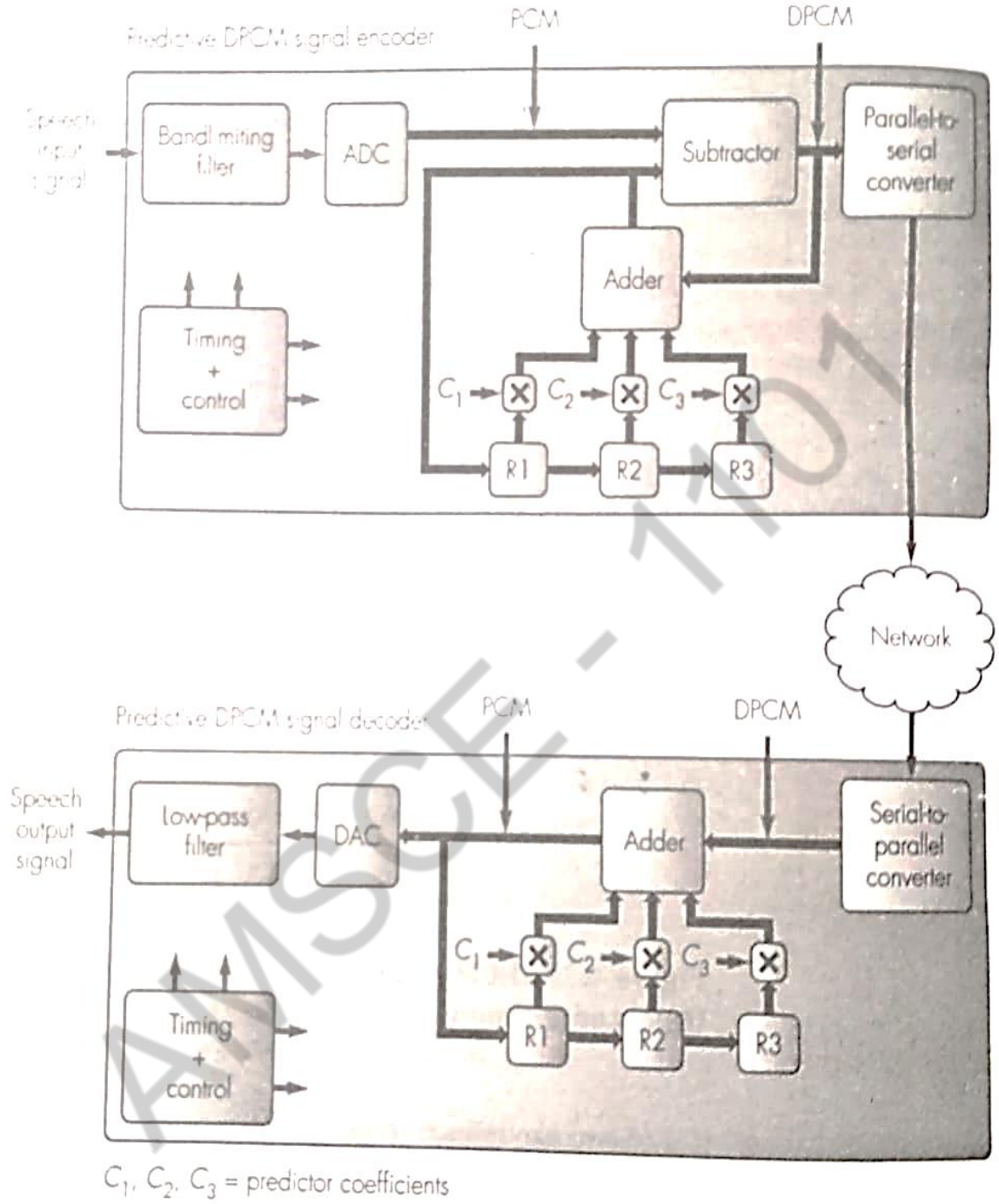


Figure 4.2 Third-order predictive DPCM signal encoder and decoder schematic.

2. Discuss the techniques of ADPCM with neat diagram.

Additional savings in bandwidth – or improved quality – can be obtained by varying the number of bits used for the difference signal depending on its amplitude; that is, using fewer bits to encode (and hence transmit) smaller difference values than for larger values. This is the principle of **adaptive differential PCM** (**ADPCM**) and an international standard for this is defined in **ITU-T Recommendation G.721**. This is based on the same principle as DPCM except an eight-order predictor is used and the number of bits used to quantize each difference value is varied. This can be either 6 bits – producing 32 kbps – to obtain a better quality output than with third-order DPCM, or 5 bits – producing 16 kbps – if lower bandwidth is more important.

A second ADPCM standard, which is a derivative of G.721, is defined in **ITU-T Recommendation G.722**. This provides better sound quality than the G.721 standard at the expense of added complexity. To achieve this, it uses an added technique known as **subband coding**. The input speech bandwidth is extended to be from 50 Hz through to 7 kHz – compared with 3.4 kHz for a standard PCM system – and hence the wider bandwidth produces a higher-fidelity speech signal. This is particularly important in conferencing applications, for example, in order to enable the members of the conference to discriminate between the different voices of the members present. The general principle of the scheme is shown in Figure 4.3.

To allow for the higher signal bandwidth, prior to sampling the audio input signal, it is first passed through two filters: one which passes only signal frequencies in the range 50 Hz through to 3.5 kHz, and the other only frequencies in the range 3.5 kHz through to 7 kHz. By doing this, the input (speech) signal is effectively divided into two separate equal-bandwidth signals, the first known as the **lower subband signal** and the second the **upper subband signal**. Each is then sampled and encoded independently using

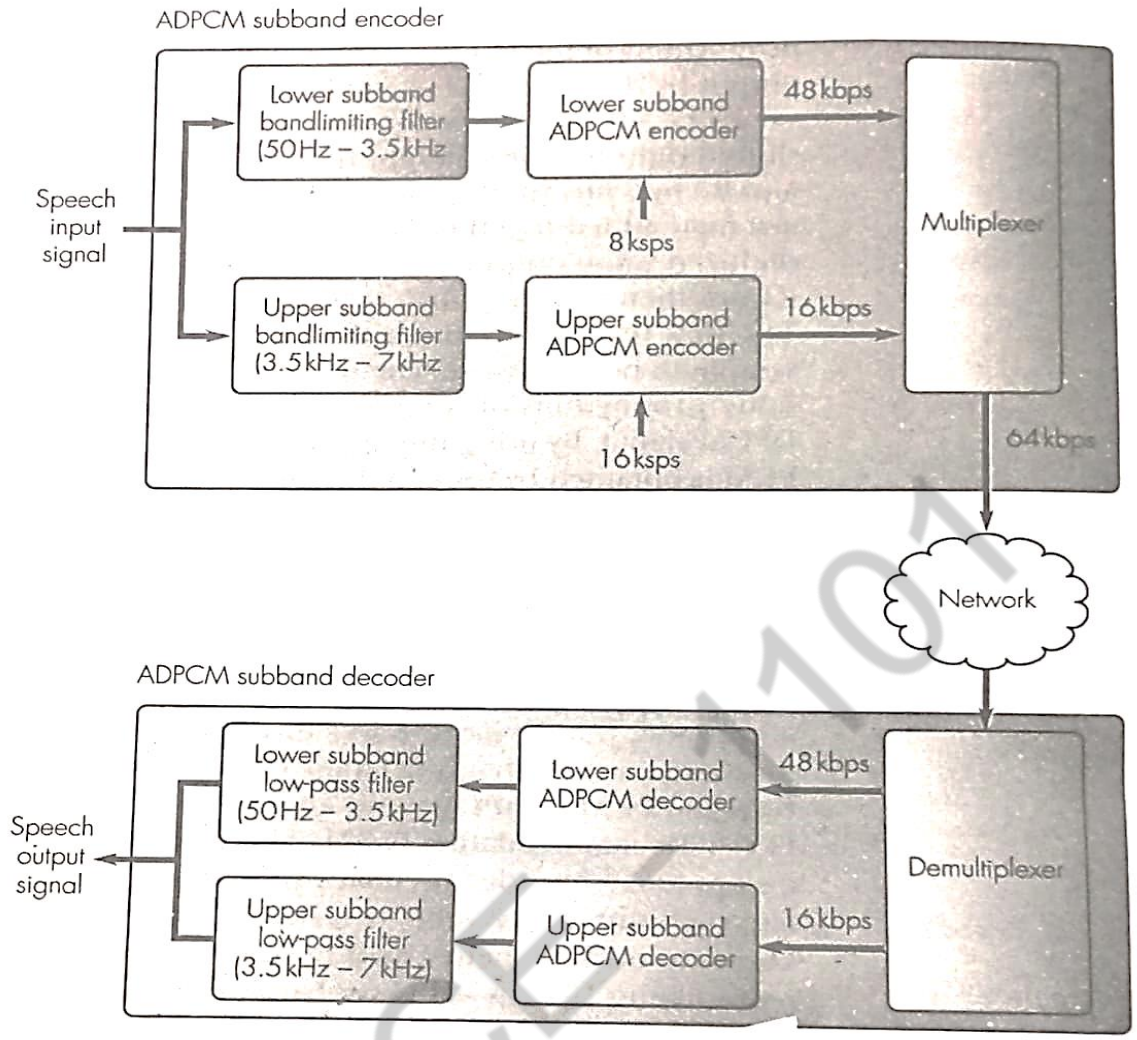


Figure 4.3 ADPCM subband encoder and decoder schematic.

ADPCM, the sampling rate of the upper subband signal being 16 ksps to allow for the presence of the higher frequency components in this subband.

The use of two subbands has the advantage that different bit rates can be used for each. In general, the frequency components that are present in the lower subband signal have a higher perceptual importance than those in the higher subband. The operating bit rate can be 64, 56, or 48 kbps. With a bit rate of 64 kbps, for example, the lower subband is ADPCM encoded at 48 kbps and the upper subband at 16 kbps. The two bitstreams are then multiplexed (merged) together – to produce the transmitted (64 kbps) signal – in such a way that the decoder in the receiver is able to divide them back again into two separate streams for decoding.

A third standard based on ADPCM is also available. This is defined in **ITU-T Recommendation G.726**. This also uses subband coding but with a speech bandwidth of 3.4 kHz. The operating bit rate can be 40, 32, 24, or 16 kbps.

3. Discuss in detail about the Linear Predictive Codes (LPC) with neat diagram.

All the algorithms we have considered so far are based on sampling the time-varying speech waveform and then either sending the quantized samples directly (PCM) or sending the quantized difference signal (DPCM and its derivatives). With the advent of inexpensive digital signal processing circuits, an alternative approach has become possible which involves the source simply analyzing the audio waveform to determine a selection of the perceptual features it contains. These are then quantized and sent and the destination uses them, together with a sound synthesizer, to regenerate a sound that is perceptually comparable with the source audio signal. This is the basis of the **linear predictive coding** (**LPC**) technique and, although with this the generated sound – normally speech – can often sound synthetic, very high levels of compression (and hence low bit rates) can be achieved.

Clearly, the key to this approach is to identify the set of perceptual features to be used and, in terms of speech, the three features which determine the perception of a signal by the ear are its:

- pitch: this is closely related to the frequency of the signal and is important because the ear is more sensitive to frequencies in the range 2–5 kHz than to frequencies that are higher or lower than these;
- period: this is the duration of the signal;
- loudness: this is determined by the amount of energy in the signal.

In addition, the origins of the sound are important. These are known as **vocal tract excitation parameters** and classified as:

- voiced sounds: these are generated through the vocal chords and examples include the sounds relating to the letters m, v, and l;
- unvoiced sounds: with these the vocal chords are open and examples include the sounds relating to the letters f and s.

Once these have been obtained from the source waveform, it is possible to use them, together with a suitable model of the vocal tract, to generate a synthesized version of the original speech signal. The basic features of an LPC encoder/decoder are shown in Figure 4.4.

The input speech waveform is first sampled and quantized at a defined rate. A block of digitized samples – known as a segment – is then analyzed to determine the various perceptual parameters of the speech that it contains. The speech signal generated by the vocal tract model in the decoder is a function of the present output of the speech synthesizer – as determined by the current set of model coefficients – plus a linear combination of the previous set of model coefficients. Hence the vocal tract model used is adaptive

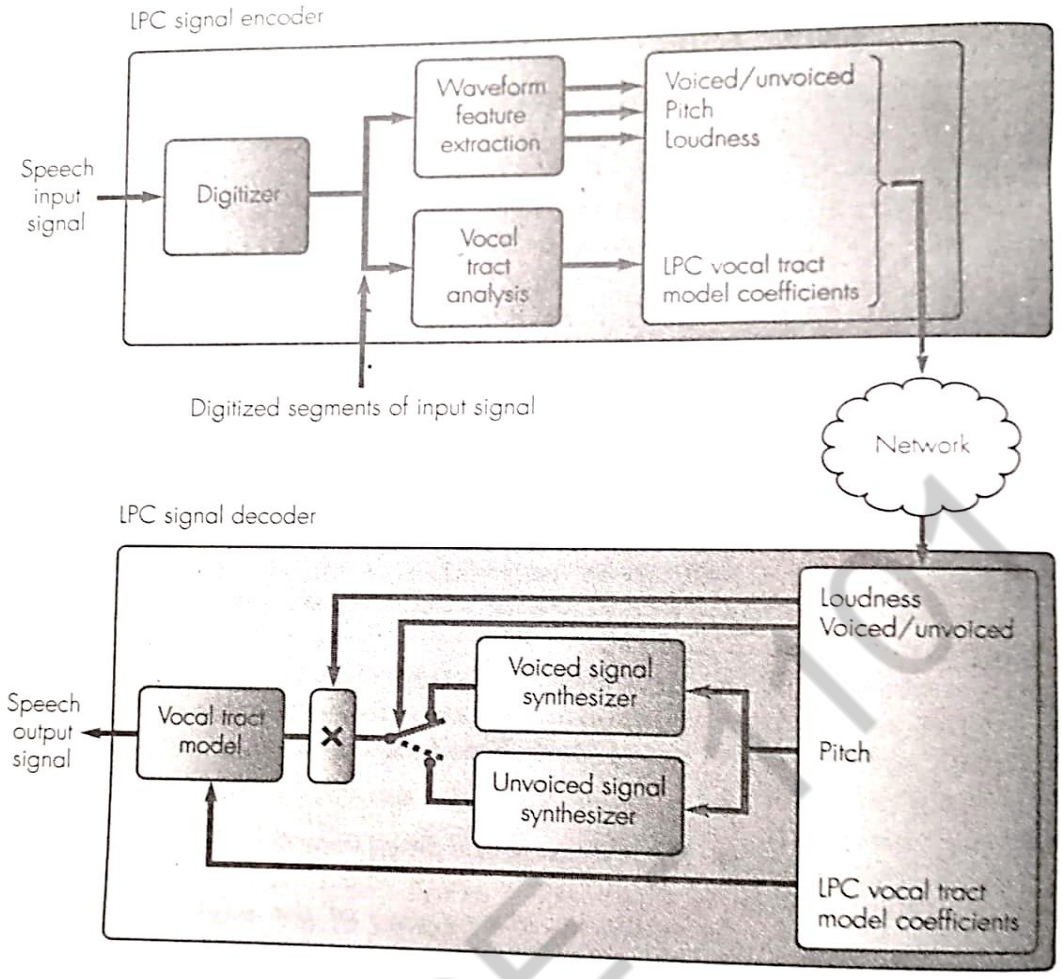


Figure 4.4 Linear predictive coding (LPC) signal encoder and decoder schematic.

and, as can be seen, the encoder determines and sends a new set of coefficients for each quantized segment.

As we can see from the above, the output of the encoder is a string of frames, one for each segment. Each frame contains fields for pitch and loudness – the period is determined by the sampling rate being used – a notification of whether the signal is voiced or unvoiced, and a new set of computed model coefficients. Some LPC encoders use up to ten sets of previous model coefficients to predict the output sound (**LPC-10**) and use bit rates as low as 2.4 kbps or even 1.2 kbps. As indicated, however, the generated sound at these rates is often very synthetic and hence LPC coders are used primarily in military applications in which bandwidth is all-important.

4. Explain the concept of Code – Excited LPC with necessary diagrams.

The synthesizers used in most LPC decoders are based on a very basic model of the vocal tract. A more sophisticated version of this, known as a **code-excited linear prediction (CELP) model**, is also used and, in practice, is just one example of a family of vocal tract models known as **enhanced excitation (LPC) models**. These are also intended primarily for applications in which the amount of bandwidth available is limited but the perceived quality of the speech must be of an acceptable standard for use in various multimedia applications.

In the CELP model, instead of treating each digitized segment independently for encoding purposes, just a limited set of segments is used, each known as a **waveform template**. A precomputed set of templates are held by the encoder and decoder in what is known as a **template codebook**. Each of the individual digitized samples that make up a particular template in the codebook are differentially encoded. Each codeword that is sent selects a particular template from the codebook whose difference values best match those quantized by the encoder. In this way, there is continuity from one set of samples to another and, as a result, an improvement in sound quality is obtained.

There are now four international standards available that are based on this principle. These are **ITU-T Recommendations G.728**, **729**, **729(A)**, and **723.1** all of which give a good perceived quality at low bit rates.

All coders of this type have a delay associated with them which is incurred while each block of digitized samples is analyzed by the encoder and the speech is reconstructed at the decoder. The combined delay value is known as the coder's **processing delay**. In addition, before the speech samples can be analyzed, it is necessary to buffer – store in memory – the block of samples. The time to accumulate the block of samples is known as the **algorithmic delay** and, in some CELP coders, this is extended to include samples from the next successive block, a technique known as **lookahead**. These delays occur in the coders, of course, and hence are in addition to the end-to-end transmission delay over the network. Nevertheless, the combined delay value of a coder is an important parameter as it often determines the suitability of the coder for a specific application. For example, in a conventional telephony

application, a low-delay coder is required since a large delay can impede the flow of a conversation. In contrast, in an interactive application that involves the output of speech stored in a file, for example, a delay of several seconds before the speech starts to be output is often acceptable and hence the coder's delay is less important. Other parameters of the coder that are considered are the **complexity** of the coding algorithm and the **perceived quality** of the output speech and, in general, a compromise has to be reached between a coder's speech quality and its delay/complexity.

The delay associated with a basic PCM coder is very small as it is equal to the time interval between two successive samples of the input waveform. Hence at the basic PCM sampling rate of 8 ksps the delay is equal to 0.125 ms. This same delay also applies, of course, to ADPCM coders. In contrast, the four CELP-based standards have delay values in excess of these as multiple samples are involved. These are summarized in Table 4.1 which also includes the bit rate(s) associated with each standard and the principal application for which each has been developed. The use of the extension .1 with G.723.1 is used to discriminate this standard from the earlier G.723 standard which has now been integrated with G.721 into the G.726 standard.

5. Explain the concept of Perceptual Coding with necessary diagrams.

Both LPC and CELP are used primarily for telephony applications and hence the compression of a speech signal. Perceptual encoders, however, have been designed for the compression of general audio such as that associated with a digital television broadcast. They also use a model but, in this case, it is known as a **psychoacoustic model** since its role is to exploit a number of the limitations of the human ear.

Using this approach, sampled segments of the source audio waveform are analyzed – as with CELP-based coders – but only those features that are perceptible to the ear are transmitted. For example, although the human ear is sensitive to signals in the range 15 Hz through to 20 kHz, the level of sensitivity to each signal is non-linear; that is, the ear is more sensitive to some signals

than others. Also, when multiple signals are present – as is the case with general audio – a strong signal may reduce the level of sensitivity of the ear to other signals which are near to it in frequency, an effect known as **frequency masking**. In addition, when the ear hears a loud sound, it takes a short but finite time before it can hear a quieter sound, an effect known as **temporal masking**. A psychoacoustic model is used to identify those signals that are influenced by both these effects. These are then eliminated from the transmitted signals and, in so doing, this reduces the amount of information to be transmitted.

Sensitivity of the ear

We defined the dynamic range of a signal in Section 2.2.2. It is the ratio of the maximum amplitude of the signal to the minimum amplitude and is measured in decibels (dB). In the case of the ear, its dynamic range is the ratio of the loudest sound it can hear to the quietest sound and is in the region of 96 dB. As we have just seen, however, the sensitivity of the ear varies with the frequency of the signal and, assuming just a single-frequency signal is present at any one time, the perception threshold of the ear – that is, its minimum level of sensitivity – as a function of frequency is shown in Figure 4.5(a).

As we can see, the ear is most sensitive to signals in the range 2–5 kHz and hence signals within this band are the quietest the ear is sensitive to. The vertical axis, therefore, indicates the amplitude level of all the other signal frequencies relative to this level – measured in dB – that are required for them to be heard. Hence in the figure, although the two signals A and B have the same relative amplitude, signal A would be heard – that is, it is above the hearing threshold – while signal B would not.

Frequency masking

When an audio sound consists of multiple frequency signals is present, the sensitivity of the ear changes and varies with the relative amplitude of the signals. For example, the curve shown in Figure 4.5(b) shows how the sensitivity of the ear changes in the vicinity of a loud signal. In this example, signal B is larger in amplitude than signal A and, as we can see, this causes the basic sensitivity curve of the ear to be distorted in the region of signal B. As a result, signal A will no longer be heard even though on its own, it is above the hearing threshold of the ear for a signal of that frequency. This is the origin of the term frequency masking and, in practice, the masking effect also varies with frequency as we show in Figure 4.6.

The various curves show the masking effect of a selection of different frequency signals – 1, 4, and 8 kHz – and, as we can see, the width of the masking curves – that is, the range of frequencies that are affected – increase with increasing frequency. The width of each curve at a particular signal level is known as the **critical bandwidth** for that frequency and experiments have shown that, for frequencies less than 500 Hz, the critical bandwidth remains constant at about 100 Hz. For frequencies greater than 500 Hz, however, the critical bandwidth increases (approximately) linearly in multiples of 100 Hz. For example, for a signal of 1 kHz (2 × 500 Hz), the critical bandwidth is

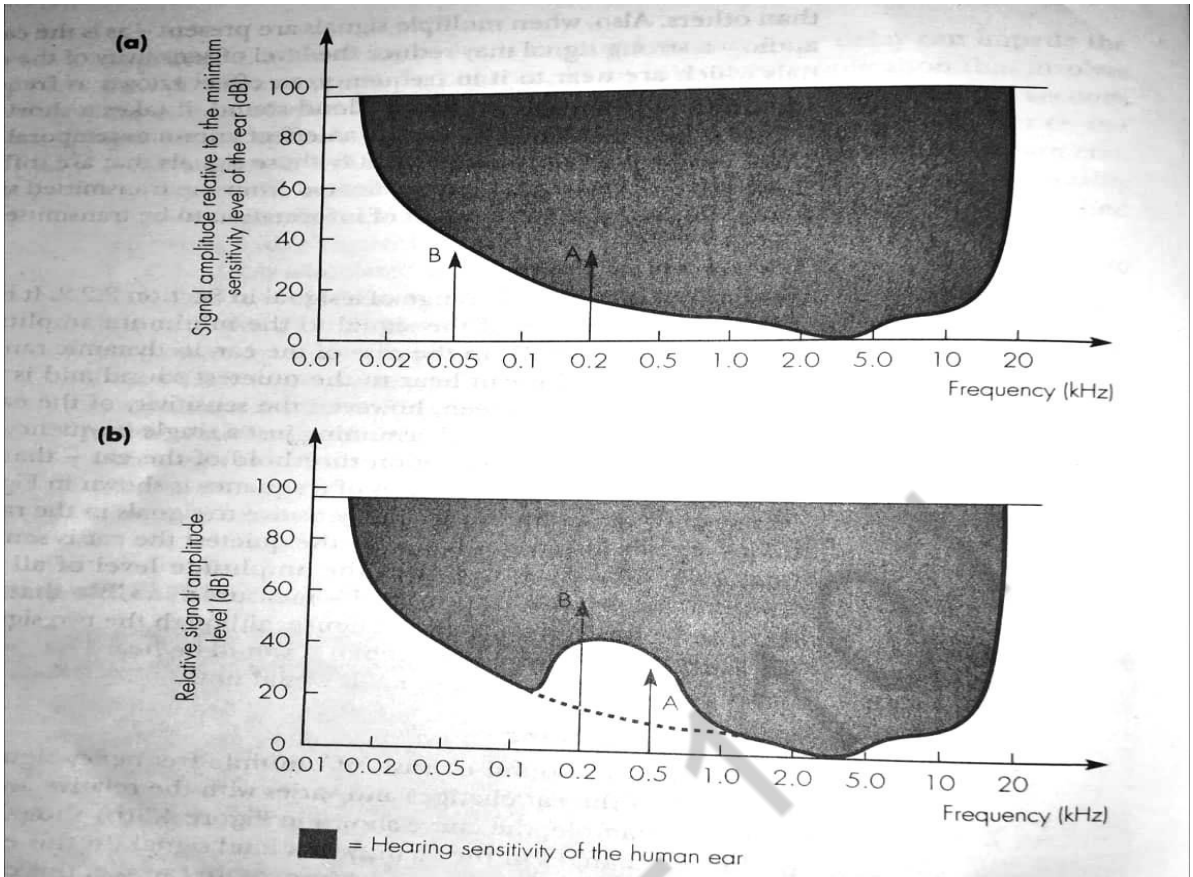


Figure 4.5 Perceptual properties of the human ear: (a) sensitivity as a function of frequency; (b) frequency masking.

about 200 (2 × 100) Hz while at 5 kHz (10 × 500 Hz) it is about 1000 (10 × 100) Hz. Hence if the magnitude of the frequency components that make up an audio sound can be determined, it becomes possible to determine those frequencies that will be masked (and hence inaudible) and do not therefore need to be transmitted.

Temporal masking

As indicated earlier, after the ear hears a loud sound, it takes a further short time before it can hear a quieter sound. This is known as temporal masking and the general effect is shown in Figure 4.7. As we can see, after the loud

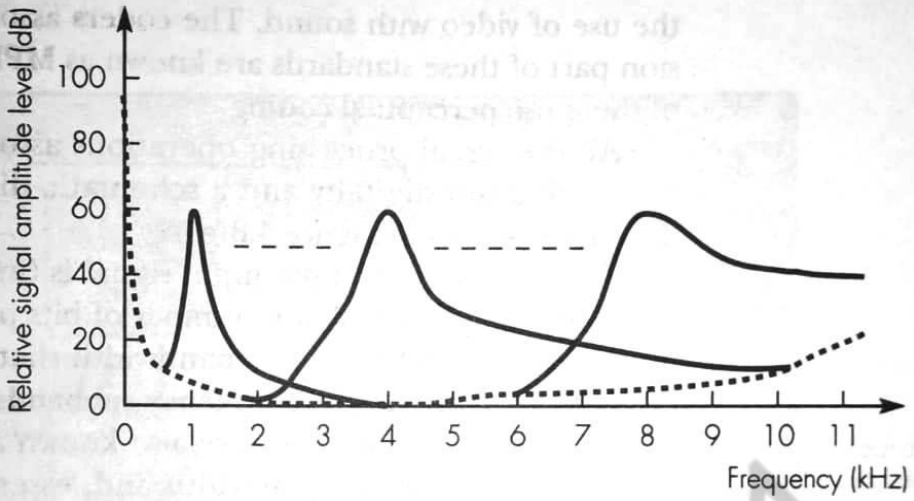


Figure 4.6 Variation with frequency of effect of frequency masking.

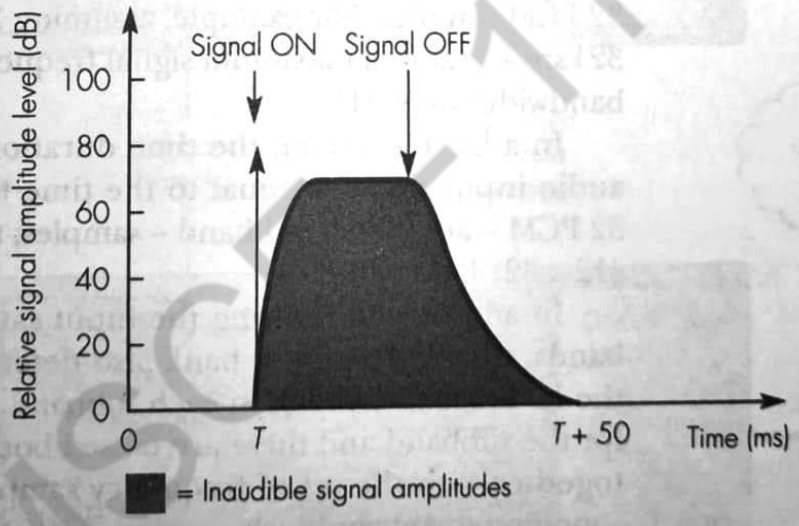


Figure 4.7 Temporal masking caused by a loud signal.

sound ceases it takes a short period of time (in the order of tens of milliseconds) for the signal amplitude to decay. During this time, signals whose amplitudes are less than the decay envelope will not be heard and hence need not be transmitted. Clearly, however, in order to exploit this phenomenon, it is necessary to process the input audio waveform over a time period that is comparable with that associated with temporal masking.

6. Explain in detail about the digitization principles.

Fourier analysis can be used to show that any time- varying analog signal is made up of a possiblyinfinitenumber of single-frequency sinusoidal signals whose amplitude and phase vary continuously with time relative to eachotherSignal bandwidthFig2.1.

The bandwidth of thetransmissionchannelshouldbe equaltoorgreater thanthebandwidthofthesignal -band limitingchannel.

Encoderdesign

A band-limiting filter and an analog-to-digital converter (ADC), the latter comprising asample-and-hold and a quantizer shown in Fig 2.2.

- Remove selected higher-frequency components from the source signal(A)
- (B) is then fed to the sample-and-holdcircuit
- Sampletheamplitudeofthefilteredsignalatregulartime intervals (C) andhold the sampleamplitudeconstantbetween samples (D)
- Quantizer circuit which converts each sample amplitude into a binary value known as a codeword (E)
- The signal to be sampled at a rate which is higher than the maximum rate of change of the signal amplitude
- The number of different quantization levels used to be as large as possible

- Nyquist sampling theorem states that: in order to obtain
- an accurate representation of a time-varing analog signal, its amplitude must be sampled at a minimum rate that is equal to or greater than twice the highest sinusoidal frequency component that is present in the signal
- Nyquist rate: samples per second (sps)
- The distortion caused by sampling a signal at a rate lower than the Nyquist rate
- Quantization intervals
- A finite number of digits is used, each sample can only be represented by a corresponding number of discrete levels
- If Vmax is the maximum positive and negative signal amplitude and n is the number of binary bits used, then the magnitude of each quantization interval, q

Figure 2.1 Signal properties: (a) time-varying analog signal; (b) sinusoidial frequency components; (c) signal bandwidth examples; (d) effect of a limited bandwidth transmission channel.

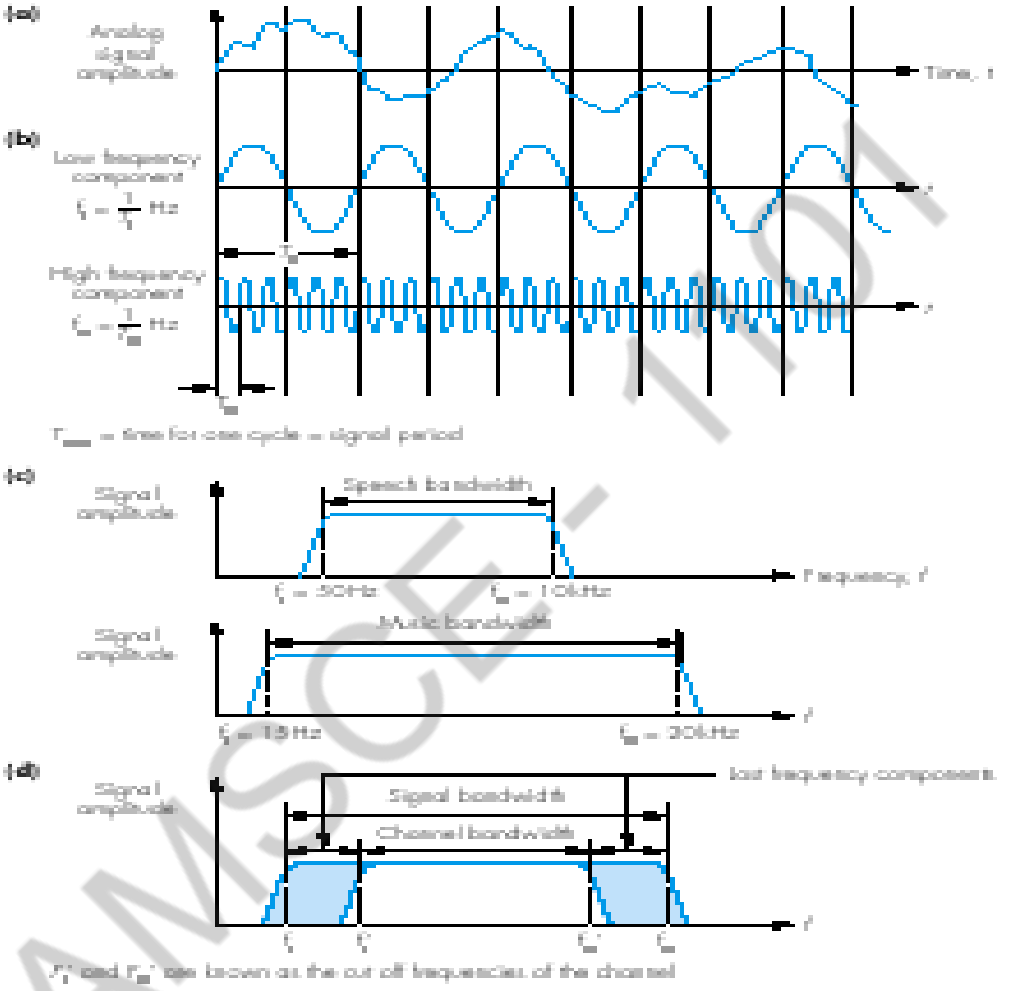
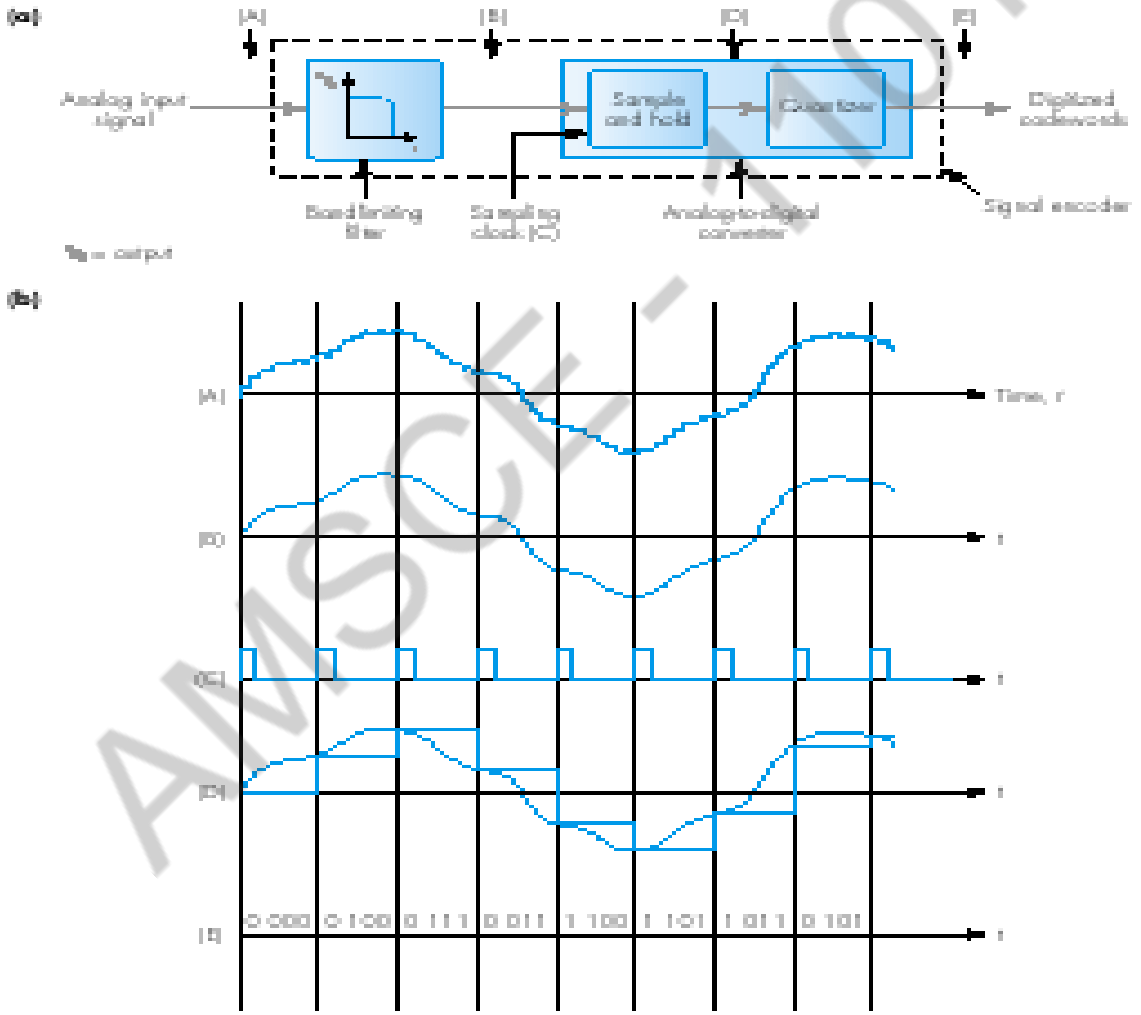
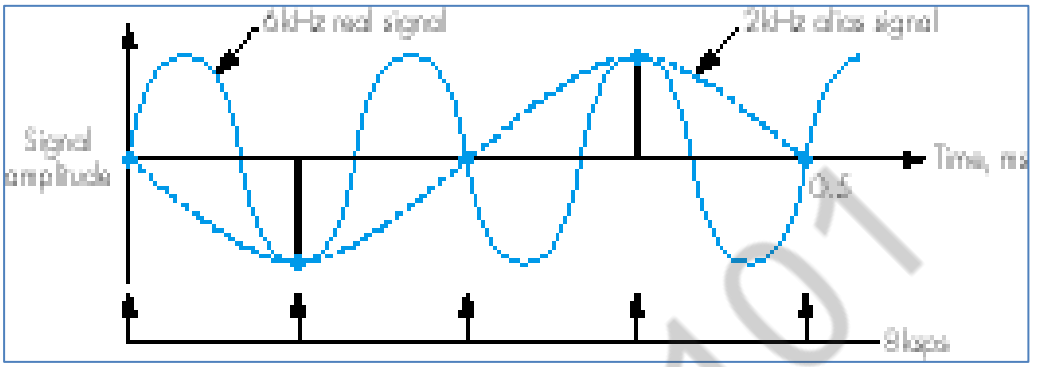


Figure 2.2 Signal encoder design: (a) circuit components; (b) associated waveform set.





- Each codeword corresponds to a nominal amplitude level which is at the center of the corresponding quantization interval
- The difference between the actual signal amplitude and the corresponding nominal amplitude is called the quantization error (Quantization noise)
- The ratio of the peak amplitude of a signal to its minimum amplitude is known as the dynamic range of the signal, D (decibels or dB)

$$D = 20\log_{10}\left(V_{max}/V_{min}\right) dB$$

- It is necessary to ensure that the level of quantization noise relative to the smallest signal amplitude is acceptable

Decoder design

- Reproduce the original signal, the output of the DAC is passed through a low-pass filter which only passes those frequency components that made up the original filtered

– signal (C)

– Audio/video encoder-decoder or audio/video codec

Figure 2.4 Quantization procedure: (a) source of errors; (b) noise polarity.

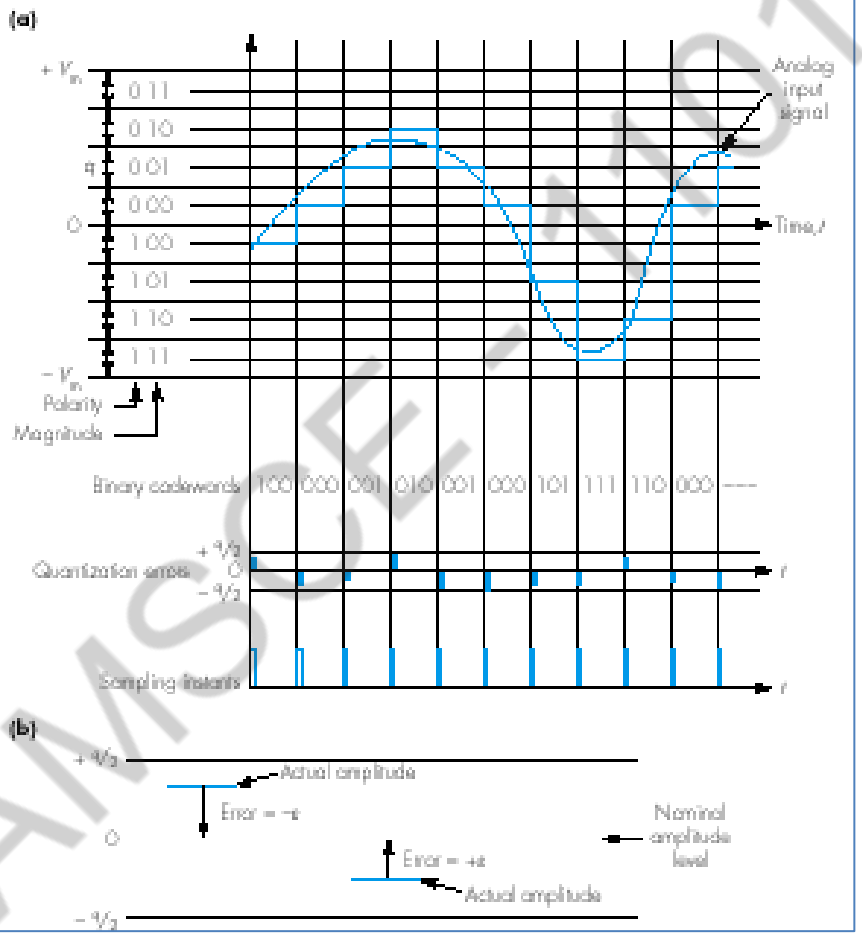
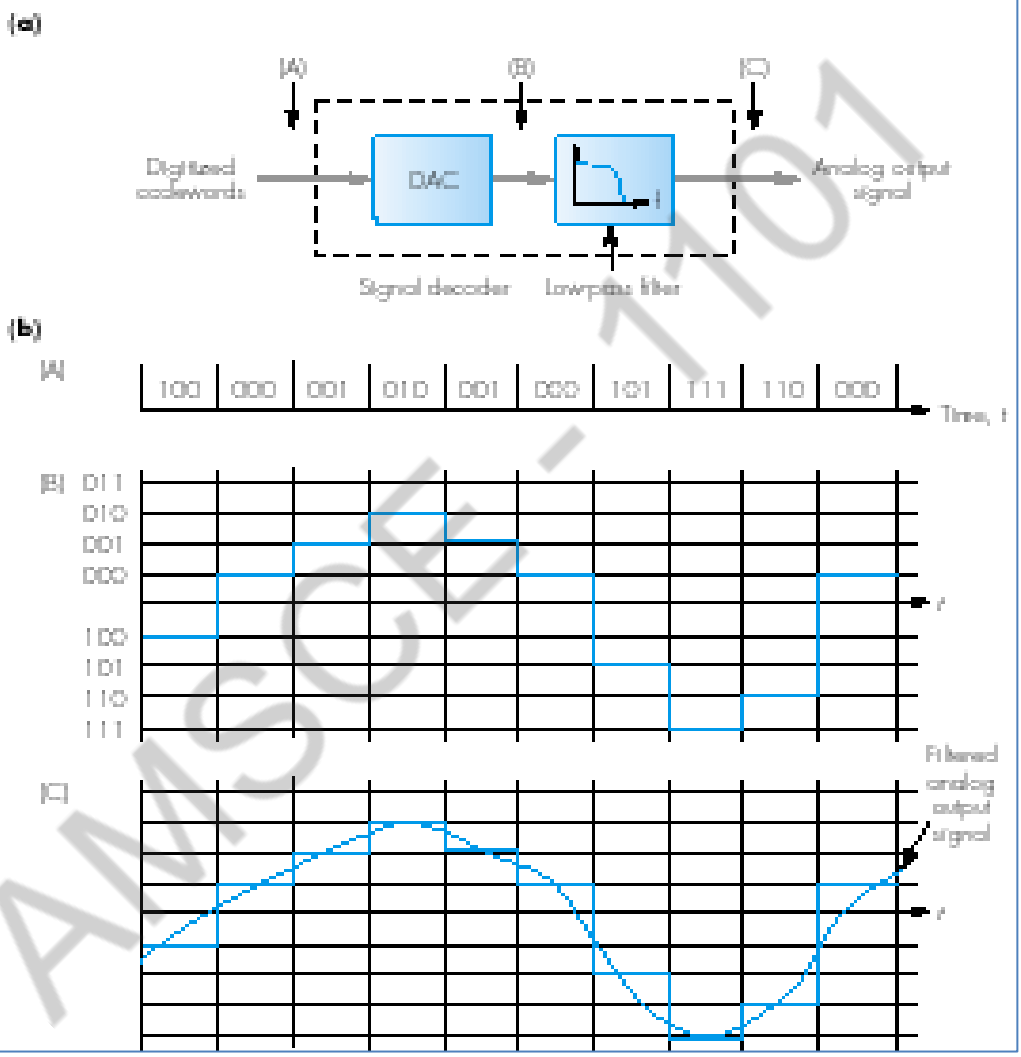


Figure 2.5 Signal decoder design: (a) circuit components; (b) associated waveform set.



7. Explain about lossless predictive coding.

Lossless Predictive Coding:

The error-free compression approach does not require decomposition of an image

into a collection of bit planes. The approach, commonly referred to as lossless predictive coding, is based on eliminating the interpixel redundancies of closely spaced pixels by extracting and coding only the new information in each pixel. The new information of a pixel is defined as the difference between the actual and predicted value of that pixel.

Figure 8.1 shows the basic components of a lossless predictive coding system. The system consists of an encoder and a decoder, each containing an identical predictor. As each successive pixel of the input image, denoted f_n, is introduced to the encoder, the predictor generates the anticipated value of that pixel based on some number of past inputs. The output of the predictor is then rounded to the nearest integer, denoted $\hat{f}_n$ and used to form the difference or prediction error which is coded using a variable-length code (by the symbol encoder) to generate the next element of the compressed data stream.

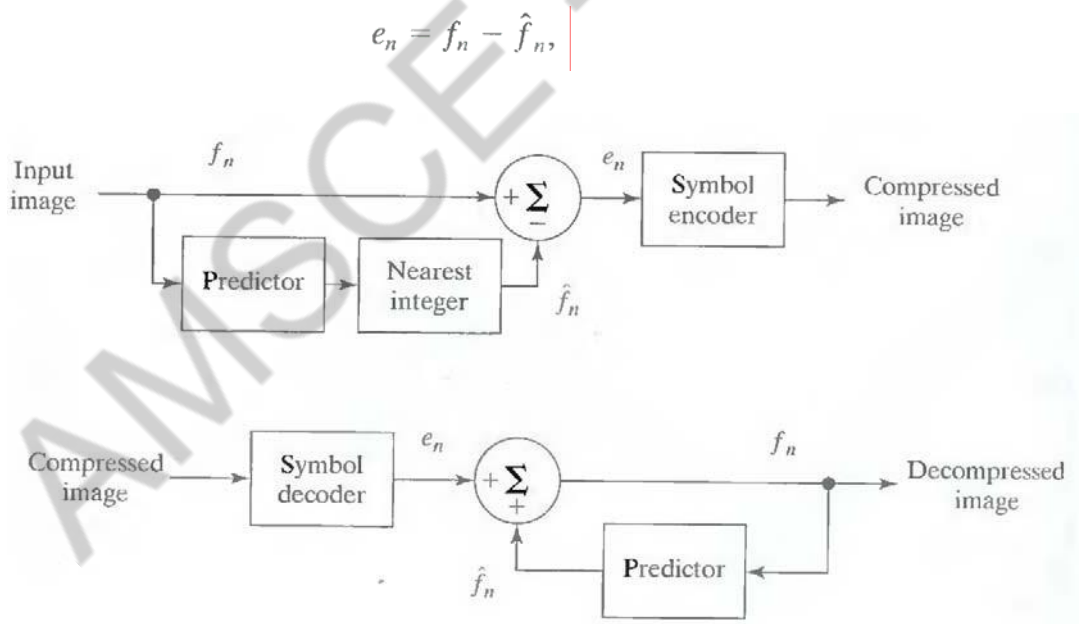


Fig.8.1 A lossless predictive coding model: (a) encoder; (b) decoder

The decoder of Fig. 8.1 (b) reconstructs en from the received variable-length code words and performs the inverse operation

$$f_n = e_n + \hat{f}_n.$$

Various local, global, and adaptive methods can be used to generate f^$_n$. In most cases, however, the prediction is formed by a linear combination of m previous pixels. That is,

$$\hat{f}_n = \text{round}\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

where m is the order of the linear predictor, round is a function used to denote the rounding or nearest integer operation, and the α_i, for i = 1,2,..., m are prediction coefficients. In raster scan applications, the subscript n indexes the predictor outputs in accordance with their time of occurrence. That is, f_n, f^$_n$ and e_n in Eqns. above could be replaced with the more explicit notation f (t), f^(t), and e (t), where t represents time. In other cases, n is used as an index on the spatial coordinates and/or frame number (in a time sequence of images) of an image. In 1-D linear predictive coding, for example, Eq. above can be written as

$$\hat{f}_n(x, y) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(x, y - i)\right]$$

where each subscripted variable is now expressed explicitly as a function of spatial coordinates x and y. The Eq. indicates that the 1-D linear prediction f(x, y) is a function of the previous pixels on the current line alone. In 2-D predictive coding, the prediction is a function of the previous pixels in a left-to-right, top-to-bottom scan of an image. In the 3-D case, it is based on these pixels and the previous pixels of preceding frames. Equation above cannot be evaluated for the first m pixels of each line, so these pixels must be coded by using other means (such as a Huffman code) and considered as an overhead of the predictive coding process. A similar comment applies to the higher-dimensional cases.

8. Explain about lossy predictive coding.

Lossy Predictive Coding:

In this type of coding, we add a quantizer to the lossless predictive model and examine the resulting trade-off between reconstruction accuracy and compression performance. As Fig.9 shows, the quantizer, which absorbs the nearest integer function of the error-free encoder, is inserted between the symbol encoder and the point at which the prediction error is formed. It maps the prediction error into a limited range of outputs, denoted e^ₙ which establish the amount of compression and distortion associated with lossy predictive coding.

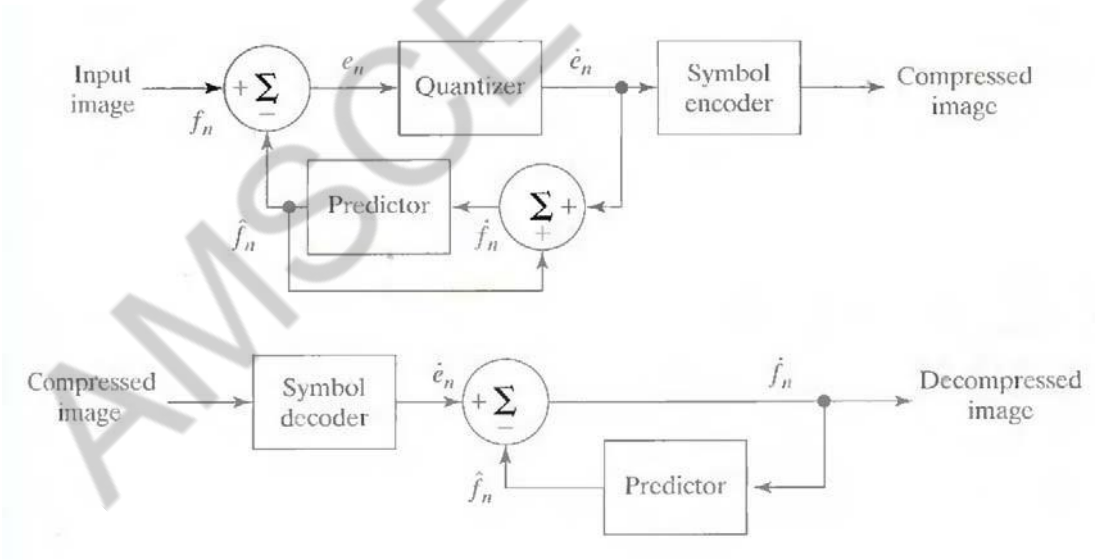


Fig. 9 A lossy predictive coding model: (a) encoder and (b) decoder.

In order to accommodate the insertion of the quantization step, the error-free encoder of figure must be altered so that the predictions generated by the encoder and

decoder are equivalent. As Fig.9 (a) shows, this is accomplished by placing the lossy encoder's predictor within a feedback loop, where its input, denoted $\dot{f}_n$, is generated as a function of past predictions and the corresponding quantized errors. That is,

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$

This closed loop configuration prevents error buildup at the decoder's output. Note from Fig. 9
(b) that the output of the decoder also is given by the above Eqn.

Optimal predictors:

The optimal predictor used in most predictive coding applications minimizes the encoder's mean- square prediction error

$$E\{e_n^2\} = E\{[f_n - \hat{f}_n]^2\}$$

subject to the constraint that

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f_n$$

and

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i f_{n-i}.$$

That is, the optimization criterion is chosen to minimize the mean-square prediction error, the quantization error is assumed to be negligible ($\dot{e}_n \approx e_n$), and the prediction is constrained to a linear combination of m previous pixels.1 These restrictions are not essential, but they simplify the analysis considerably and, at the same time, decrease the computational complexity of the predictor. The resulting predictive coding approach is referred to as differential pulse code modulation (DPCM).

AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
EC8002 – MULTIMEDIA COMPRESSION & COMMUNICATION
UNIT II – Image and Video Compression
PART – A

1. What are the profiles in MPEG-2 video standard?

MPEG-2 defines 7 profiles at different applications

- Simple
- Main
- SNRscalable
- Spatiallyscalable
- High
- 4:2:2
- Multi-view

2. List the major features of H.263standard.

H.263 supports CIF, QCIF, sub QCIF, 4CIF and 16CIF. For the compressed video, the standard defines maximum bit rate per picture measured in units of 1.024 bits.

3. List the different types of compression.

Lossless Compression — after decompression gives an exact copy of the original data. Examples: Entropy Encoding Schemes (Shannon-Fano, Huffman coding), arithmetic coding,LZW algorithm used in GIF image file format.

Lossy Compression — after decompression gives ideally a 'close' approximation of the original data, in many cases perceptually lossless but a byte-by-byte comparision of files shows differences. Examples: Transform Coding — FFT/DCT based quantisation used in JPEG/MPEG differential encoding, vector quantisation.

4. What is the need for Compression?

In terms of storage, the capacity of a storage device can be effectively increased with methods that compress a body of data on its way to a storage device and decompresses it when it is retrieved.

In terms of communications, the bandwidth of a digital communication link can be effectively increased by compressing data at the sending end and decompressing

data at the receiving end.

At any given time, the ability of the Internet to transfer data is fixed. Thus, if data can effectively be compressed wherever possible, significant improvements of data throughput can be achieved. Many files can be combined into one compressed document making sending easier.

5. Define is coding redundancy?

If the gray level of an image is coded in a way that uses more code words than necessary to represent each gray level, then the resulting image is said to contain coding redundancy.

6. Define interpixel redundancy?

The value of any given pixel can be predicted from the values of its neighbors. The information carried by is small. Therefore the visual contribution of a single pixel to an image is redundant. Otherwise called as spatial redundant geometric redundant or interpixel redundant.

Eg: Run length coding

7. What is run length coding?

Run-length Encoding, or RLE is a technique used to reduce the size of a repeating string of characters. This repeating string is called a *run*; typically RLE encodes a run of symbols into two bytes, a count and a symbol. RLE can compress any type of data regardless of its information content, but the content of data to be compressed affects the compression ratio. Compression is normally measured with the compression ratio.

8. **Define psycho visual redundancy?**

In normal visual processing certain information has less importance than other information. So this information is said to be psycho visual redundant.

9. Define encoder

Source encoder is responsible for removing the coding and interpixel redundancy and psycho visual redundancy.

There are two components

A) Source Encoder
B) Channel Encoder

10. Define source encoder

Source encoder performs three operations

1) Mapper -this transforms the input data into non-visual format. It reduces the interpixel redundancy.

2) Quantizer - It reduces the psycho visual redundancy of the input images .This step is omitted if the system is error free.

3) Symbol encoder- This reduces the coding redundancy .This is the final stage of encoding process.

11. Define channel encoder

The channel encoder reduces reduces the impact of the channel noise by inserting redundant bits into the source encoded data.

Eg: Hamming code

12. What is Variable Length Coding?

Variable Length Coding is the simplest approach to error free compression. It reduces only the coding redundancy. It assigns the shortest possible codeword to the most probable gray levels.

13. Define Huffman coding

- Huffman coding is a popular technique for removing coding redundancy.
- When coding the symbols of an information source the Huffman code yields the smallest possible number of code words, code symbols per source symbol.

14. What are the coding systems in JPEG?

1. A lossy baseline coding system, which is based on the DCT and is adequate for most compression application.

2. An extended coding system for greater compression, higher precision or progressive reconstruction applications.

3. Lossless independent coding system for reversible compression.

15. What are the basic steps in JPEG?

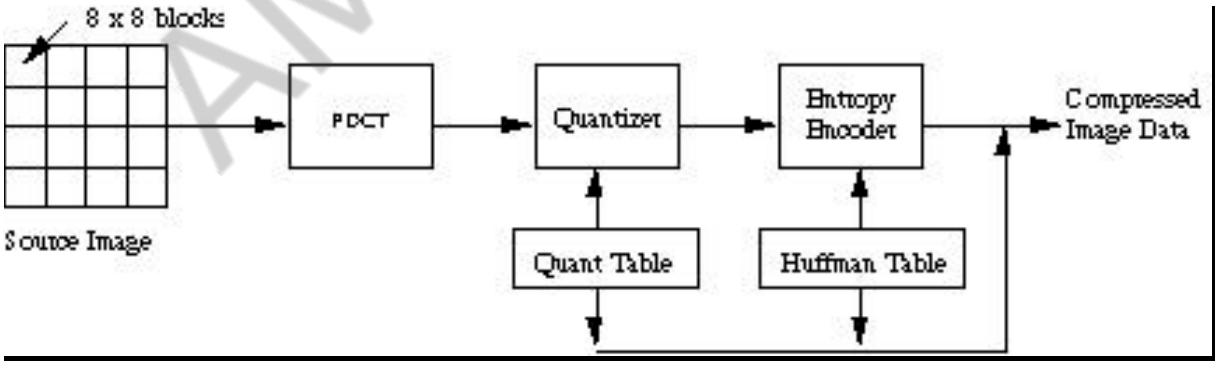
The Major Steps in JPEG Coding involve:

DCT (Discrete Cosine Transformation)
Quantization
Zigzag Scan
DPCM on DC component
RLE on AC Components
Entropy Coding

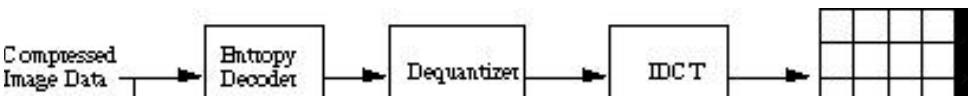
16. What is MPEG?

The acronym is expanded as "Moving Picture Expert Group". It is an international standard in 1992. It perfectly Works with video and also used in teleconferencing

17. Draw the JPEG Encoder.



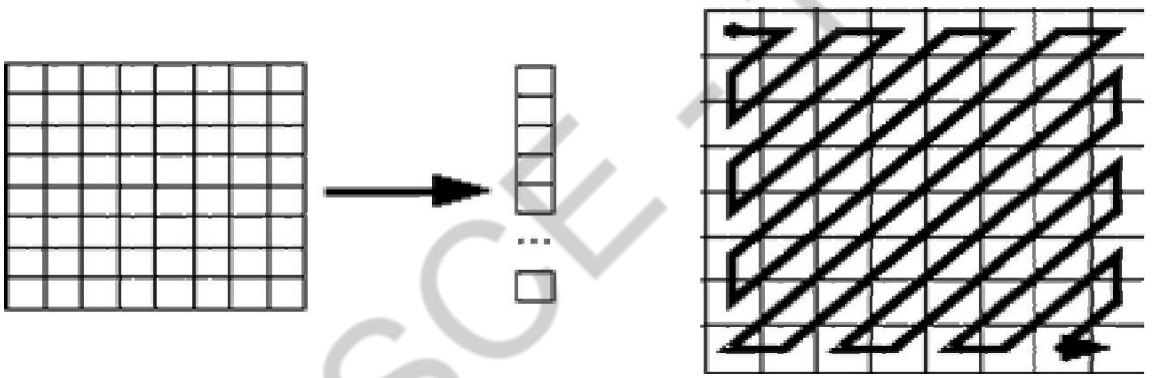
18. Draw the JPEG Decoder.



19. What is zig zag sequence?

The purpose of the Zig-zag Scan:

To group low frequency coefficients in top of vector. Maps 8 x 8 to a 1 x 64 vector



20. Define I-frame

I-frame is Intraframe or Independent frame. An I-frame is compressed independently of all frames. It resembles a JPEG encoded image. It is the reference point for the motion estimation needed to generate subsequent P and P-frame.

21. Define P-frame

P-frame is called predictive frame. A P-frame is the compressed difference between the current frame and a prediction of it based on the previous I or P-frame

22. Define B-frame

B-frame is the bidirectional frame. A B-frame is the compressed difference between the current frame and a prediction of it based on the previous I or P-frame or

next P-frame. Accordingly the decoder must have access to both past and future reference frames.

PART B

1. Write a brief note on MPEG perceptual coders.

Perceptual audio coder (PAC) is an algorithm, like MPEG's MP3 standard, used to compress digital audio by removing extraneous information not perceived by most people. It is used by Sirius Satellite Radio for their DARS service.

"Transmission bandwidth increases continuously, but the demand increases even more

need for compression technology "Applications of audio coding

- audio streaming and transmission over theinternet
- mobile musicplayers
- digitalbroadcasting
- soundtracks of digital video (e.g. digital television andDVD)

Requirements for audio coding systems Requirements for audio codingsystems

Compression efficiency: sound quality vs. bit-rate

Absolute achievable quality

- often required: given sufficiently high bit-rate, no audible difference compared to CD-quality original audio

"Complexity

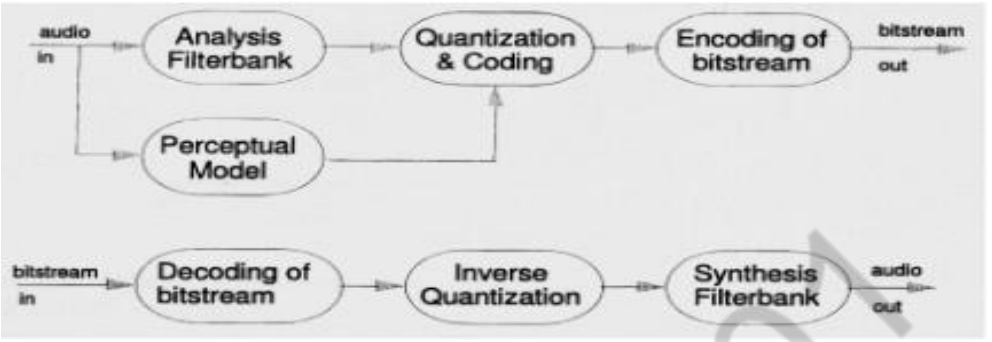
- computational complexity: main factor for general purpose computers
- storage requirements: main factor for dedicated silicon chips
- encoder vs. decoder complexity

•the encoder is usually much more complex than the decoder

•encodingcanbedoneoff-lineinsomeapplicationsRequirements(cont.)

Algorithmicdelay

- depending on the application, the delay is or is not an importantcriterion
- very important in two way communication (~ 20 msOK)
- not important in storageapplications
- somewhat important in digital TV/radio broadcasting (~ 100 ms)



Editability

- a certain point in audio signal can be accessed from the codedbitstream
- requires that the decoding can start at (almost) any point of the bitstream

Errorresilience

- susceptibility to single or burst errors in the transmissionchannel
- usually combined with error correction codes, but that costsbits

Filter bank

- used to decompose an input signal into subbands or spectral components (time-frequencydomain)

Perceptual model (aka psychoacoustic model)

- usually analyzes the input signal instead of the filterbank outputs (time-domain input provides better time and frequencyresolution)
- computes signal-dependent masked threshold based on psychoacoustics

Quantization andcoding

- spectral components are quantized andencoded
- goal is to keep quantization noise below the maskedthreshold

Framepacking

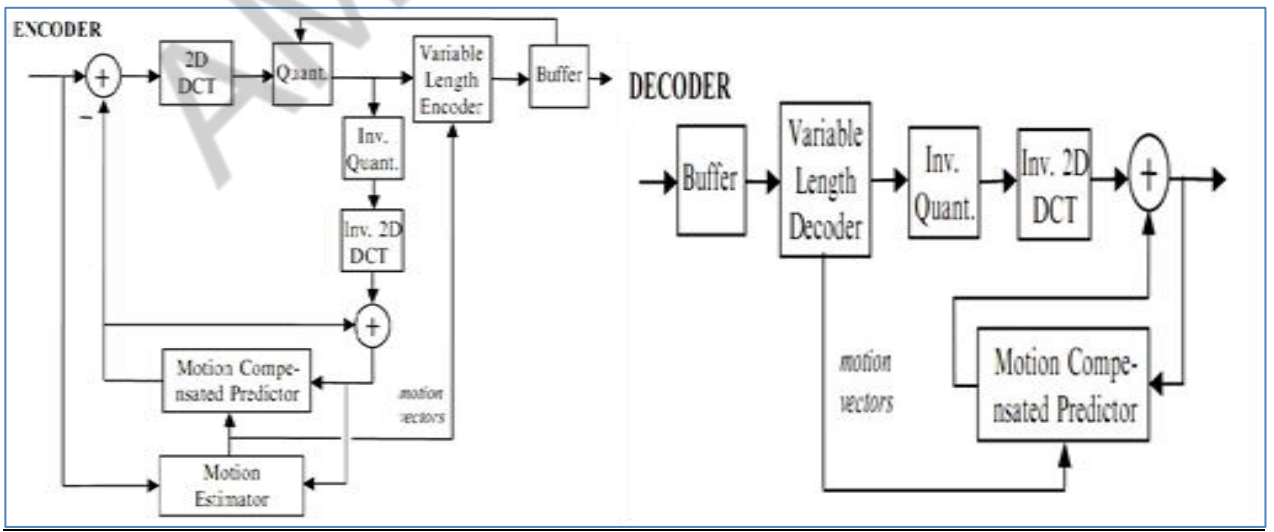
- bitstream formatter assembles the bitstream, which typically consists of the coded data and some sideinformation
- Perceptual models: maskedthreshold
- Perceptual models: tonalityestimation
- Perceptual models: MPEG-1 Layer2

2. Describe the principle of MPEG 4 with diagrams of encoder and decoder

MPEG-4 absorbs many of the features of MPEG-1 and MPEG-2 and other related standards, adding new features such as (extended) VRML support for 3D rendering, object-oriented composite files (including audio, video and VRML objects), support for externally specified Digital Rights Management and various types of interactivity.

MPEG-4 provides a series of technologies for developers, for various service- providers and for end users:

- MPEG-4 enables different software and hardware developers to create multimedia objectspossessingbetterabilitiesofadaptabilityandflexibilitytoimprovetheq uality of such services and technologies as digital television, animation graphics, the World Wide Web and theirextensions.
- Data network providers can use MPEG-4 for data transparency. With the help of standardprocedures,MPEG-4datacanbeinterpretedandtransformedintoother signal types compatible with any availablenetwork.
- TheMPEG-4formatprovidesendusers withawiderangeofinteractionwithvarious animatedobjects.
- Standardized Digital Rights Management signaling, otherwise known in the MPEG community as Intellectual Property Management and Protection(IPMP).



3. Give a brief note on H.263 video compression standard.

H.263 that was adopted by ITU-T, we would like to tell the reader s why we do not talk about any standard called H.262, which should have been logically there in between the H.261 and H.263.

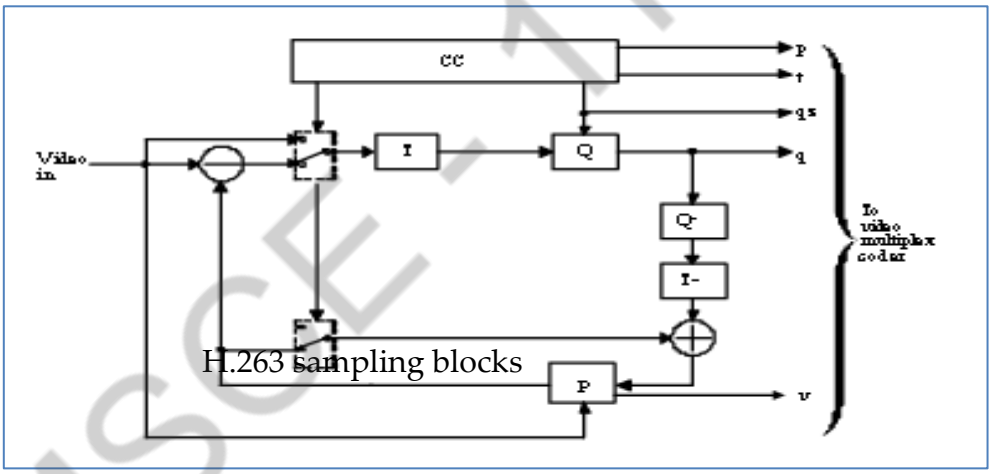
The other requirements of H.263 standardization were:

- Use of available technology
- Interoperability between the other standards, like H.261

Flexibilityfor future extensions

- Quality of service parameters, suchas resolution, delay, frame-rate etc.
- Subjective quality measurements.

H.263 block diagram



H.263 sampling blocks

- 4:2:0 sampling
  - luminance Y to chrominance CB, CR
- Block:
  - –8 x 8 pixels
- Macro block (MB):
  - –4 Y + CB + CR blocks
- Group of blocks (GOB):
  - –One or more rows of MBs
  - –In GOB header: resynchronization

The H.263 standard supports five pictures formats

- Sub-QCIF 128 x 96 pixels (Y), 64 x 48 pixels (U,V)
- QCIF 176 x 144 pixels (Y), 88 x 72 pixels (U,V)

- CIF 352 x 288 pixels (Y), 176 x 144 pixel (U,V)
- 4CIF 704 x 576 pixels (Y), 352 x 288 pixel (U,V)
- 16 CIF 1408 x 1152 pixels (Y), 704 x 576 pixel (U,V)

Two (six) frame types:

- I-frames: intra
- P-frames: predictive (inter)
- B-frames (optional): bidirectionalpredicted
- PB-frames (optional): decoded B and P frame as one unit
- EI-frames (optional): enhanced I-frame
- EP-frames (optional): enhanced P-frame

4. Elaborate on various video compression standards with emphasis on their supportingfeatures(any two standards).give required diagrams.

Standard	Application	Bit Rate
JPEG	Still image compression	Variable
H.261	Video conferencing over ISDN	P x 64 kb/s
MPEG-1	Video on digital storage media (CD-ROM)	1.5Mb/s
MPEG-2	Digital Television	2-20 Mb/s
H.263	Video telephony over PSTN	33.6-? kb/s
MPEG-4	Object-based coding, synthetic content, interactivity	Variable
JPEG-2000	Improved still image compression	Variable
H.264/ MPEG-4 AVC	Improved video compression	10's to 100's kb/s

H.264, also known as MPEG-4 AVC (Advanced VideoCoding) or MPEG4part 10, improves video compression when compared to MPEG-4 and MPEG-2by using advanced algorithms, simplified integertransforms, and in-loop deblocking filter.

MPEG-4

MPEG-4 is one of the most widely used codecs for video security. It offers improved quality relative to MPEG-2. This codec is designed to operate within awide range of bit rates and resolutions, so it is well suitedfor the video surveillance industry.

MPEG-2

MPEG-2was approved as a standard in 1994 and was designed for high frame and bit rates. MPEG-2 extends the earlier MPEG-1 compression standard to produce high quality video at the expense of a lower compression ratio and at a higher bit-rate. The frame rate islocked at 25 (PAL)/30 (NTSC) fps, as is the case for MPEG-1.

JPEG

JPEG is a format specified in the JPEG still picture coding standard in which each video frame is separately compressed as a JPEG image. JPEG is a very well-known standard and is widely used in video surveillance applications and still image cameras. The first generation of DVRs all used JPEG, but this is no longer the case.

JPEG 2000

JPEG 2000 is a wavelet-based image compression standard created by the JointPhotographic Experts Groupcommittee that provides better compression for still image coding by filtering, sub-sampling, and "smoothing" video data to removeunnecessary details. JPEG 2000 is very scalable and brings many new tools thatimprove compression, but requires significantly more processing power then JPEGto encode an image.

5. Explain the different types of frames in videocompression principles.

Three frame types:

- I-Picture (Intra-frame picture)
- P-Picture (Inter-frame predicted picture)
- B-Picture(Bi-directionalpredicted-interpolated pictures)

I-frames

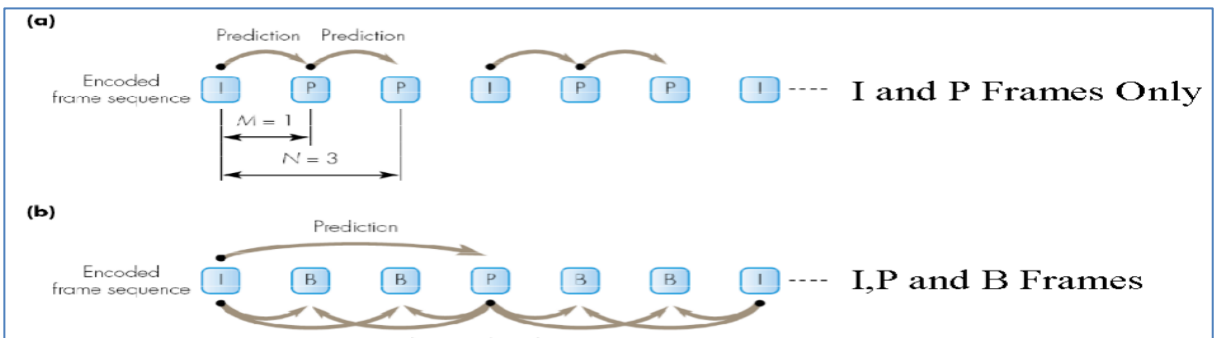
Are encoded without reference to any other frames. Each frame is treated as a separate (digitized) picture and the Y, Cb and Cr matrices areencoded independently using the JPEGencoded independently using the JPEG algorithm

P-frames

The encoding of a P-frame is relative to thecontents of either a preceding I-frame or a contents of either a preceding I frame or a preceding P-frame.P-frames are encoded using a combination Pframes are encoded using a combination of motion estimation and motion compensation

B-frames

Their contents are predicted using search regions in both pastand future frames.Allowing for occasional moving objects, this also provides better motionestimation this also provides better motion estimation.



6. Describe the operation of JPEG encoder and Decoder with neat diagrams.

JPEG is an image compression standard that was developed by the "JointPhotographic Experts Group". JPEG was formally accepted as an international standard in1992.JPEG is a lossy image compression method. It employs a transform coding method using the DCT (*Discrete CosineTransform*).

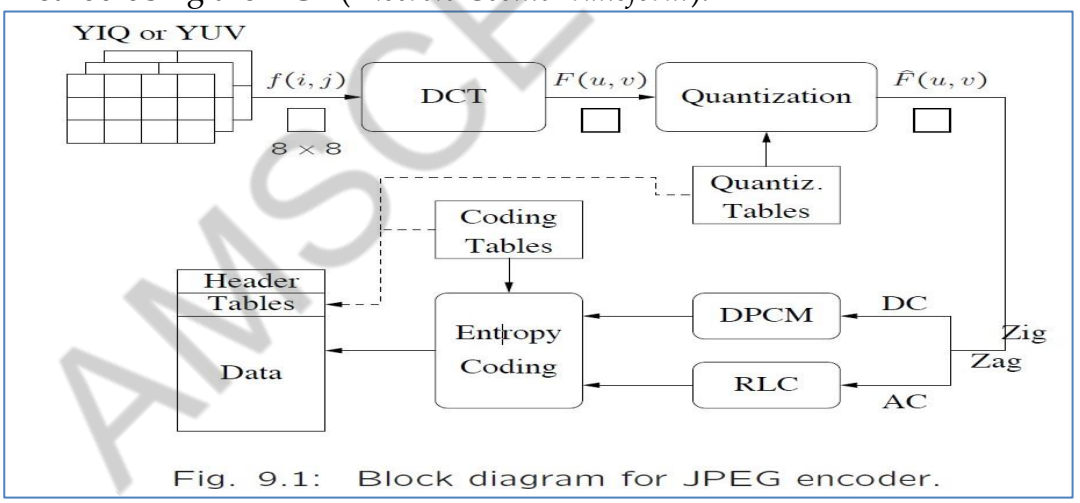


Fig. 9.1: Block diagram for JPEG encoder.

Main Steps in JPEG Image Compression

- Transform RGB to YIQ or YUV and subsample color.
- DCT on image blocks.
- Quantization.
- Zig-zag ordering and run-length encoding.
- Entropy coding.

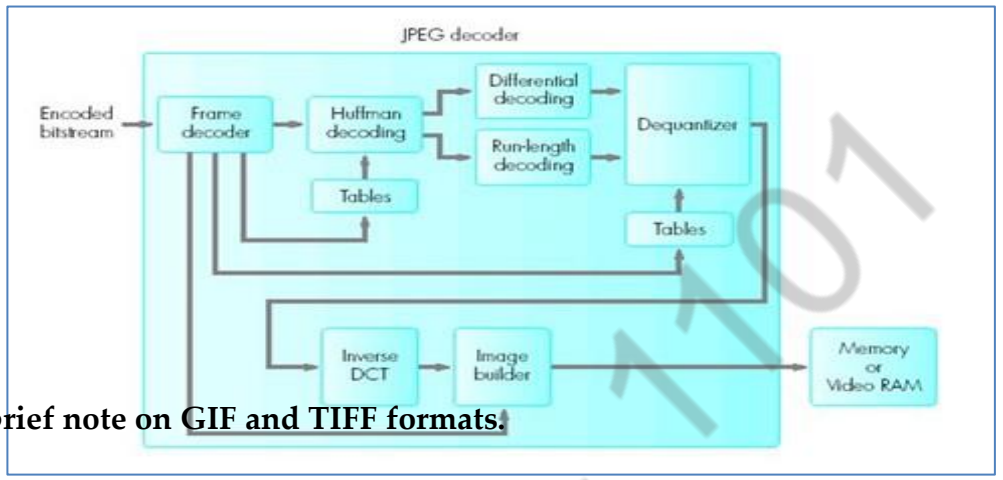
Run-length Coding (RLC) on AC coefficients

Entropy Coding

- The DC and AC coefficients finally undergo an entropy coding step to

gain a possible further compression.

– Use DC as an example: each DPCM coded DC coefficient is represented by (SIZE, AMPLITUDE), where SIZE indicates how many bits are needed for representing thecoefficient, and AMPLITUDE contains the actual bits.



7. Give a brief note on GIF and TIFF formats.

GIF(F) : stands for "Graphics Interchange Format."

A small, limited-color raster file. It is used for on-screen viewing only, when a very small file with just a few solid colors is needed. A bit-mapped file format used for graphicsasopposedtophotographicimages.GIFsupports8-bitcolor (maximumof256colors, compared to Jpegs 16 million colors.) It's widely used on the Web because the files compress well. GIFs include a color table that includes the most representative 256 colors used. Not recommended for files with a lot of color shading!

Limited color palette

LZW compression

Transparency

Interlacing

Animation

Specsheet

Resolution

- Name:GIF
- Developer:CompuServe
- Release date:1987
- Type of data:bitmap
- Number of colors:2, 4, 8, 16, 32, 64, 128 or 256
- Color spaces:RGB Compression algorithms:LZW
- Ideal use:internet publishing
- Extension on PC-platform: .gif
- Macintosh file type:?

- Special features:support for transparency, interlacing, and animation

TIFF: stands for "Tagged Image File Format" and is one of the most widely supported file formats for storing bit-mapped images on personal computers (both PCs and Macintosh computers).

How to edit TIFF files

All professional image editing applications on the market are capable of opening TIFFfiles. My favorite is Adobe Photoshop.

How to convert TIFF files

There are tons of converters that can convert a TIFF file to a JPEG, PNG, EPS, PDF or other file format.

8. Define image compression. Explain about the redundancies in a digital image.

The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information. A clear distinction must be made between data and information. They are not synonymous. In fact, data are the means by which information is conveyed. Various amounts of data may be used to represent the same amount of information. Such might be the case, for example, if a long-winded individual and someone who is short and to the point were to relate the same story. Here, the information of interest is the story; words are the data used to relate the information. If the two individuals use a different number of words to tell the same basic story, two different versions of the story are created, and at least one includes nonessential data. That is, it contains data (or words) that either provide no relevant information or simply restate that which is already known. It is thus said to contain data redundancy.

Data redundancy is a central issue in digital image compression. It is not an abstract concept but a mathematically quantifiable entity. If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy R_D of the first data set (the one characterized by n_1) can be defined as

$$R_D = 1 - \frac{1}{C_R}$$

where C_R , commonly called the compression ratio, is

$$C_R = \frac{n_1}{n_2}.$$

For the case $n_2 = n_1$, $C_R = 1$ and $R_D = 0$, indicating that (relative to the second data set) the first representation of the information contains no redundant data. When $n_2 << n_1$, C_R € ∞ and R_D€1, implying significant compression and highly redundant data. Finally, when $n_2 >> n_1$, C_R€ 0 and R_D €∞, indicating that the second data set contains much more data than the original representation. This, of course, is the normally undesirable case of data expansion. In general, C_R and R_D lie in the open intervals (0, ∞) and (- ∞, 1), respectively. A practical compression ratio, such as 10 (or 10:1), means that the first data set has 10 information carrying units (say, bits) for every 1 unit in the second or compressed data set. The corresponding redundancy of 0.9 implies that 90% of the data in the first data set is redundant.

In digital image compression, three basic data redundancies can be identified and exploited: coding redundancy, interpixel redundancy, and psychovisual redundancy. Data compression is achieved when one or more of these redundancies are reduced or eliminated.

Coding Redundancy:

In this, we utilize formulation to show how the gray-level histogram of an image also can provide a great deal of insight into the construction of codes to reduce the amount of data used to represent it.

Let us assume, once again, that a discrete random variable r_k in the interval [0, 1] represents the gray levels of an image and that each r_k occurs with probability p_r (r_k).

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots, L-1$$

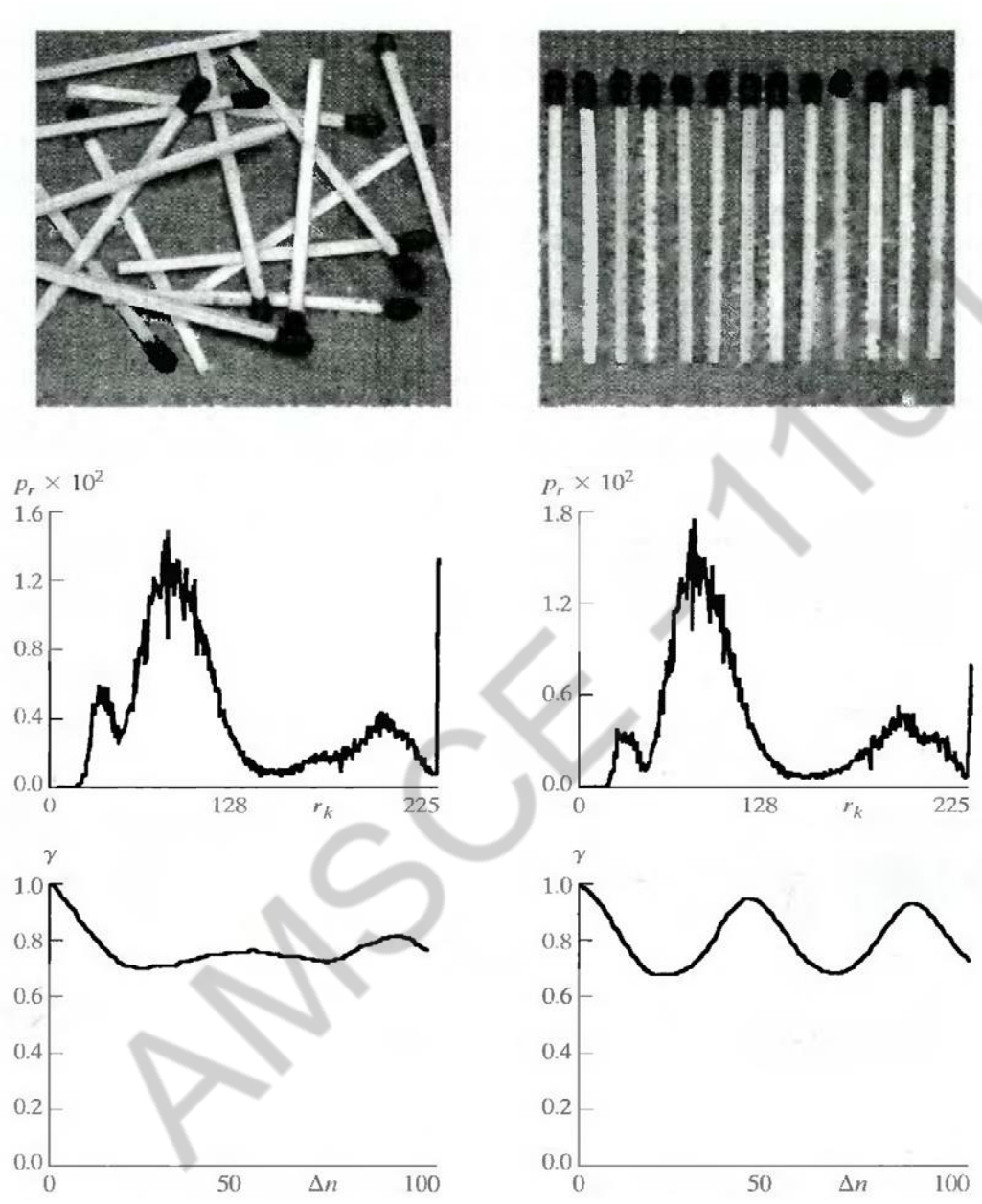
where L is the number of gray levels, n_k is the number of times that the kth gray level appears in the image, and n is the total number of pixels in the image. If the number of bits used to represent each value of r_k is l (r_k), then the average number of bits required to represent each pixel is

$$L_{\text{avg}} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k).$$

That is, the average length of the code words assigned to the various gray-level values is found by summing the product of the number of bits used to represent each gray level and the probability that the gray level occurs. Thus the total number of bits required to code an M X N image is MNL_{avg}.

Interpixel Redundancy:

Consider the images shown in Figs. 1.1(a) and (b). As Figs. 1.1(c) and (d) show, these images have virtually identical histograms. Note also that both histograms are trimodal, indicating the presence of three dominant ranges of gray-level values. Because the gray levels in these images are not equally probable, variable-length coding can be used to reduce the coding redundancy that would result from a straight or natural binary encoding of their pixels. The coding process, however, would not alter the level of correlation between the pixels within the images. In other words, the codes used to represent the gray levels of each image have nothing to do with the correlation between pixels. These correlations result from the structural or geometric relationships between the objects in the image.



a b
c d
e f

Fig.1.1 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

Figures 1.1(e) and (f) show the respective autocorrelation coefficients computed along one line of each image.

$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)}$$

where

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) f(x, y + \Delta n).$$

The scaling factor in Eq. above accounts for the varying number of sum terms that arise for each integer value of Δn. Of course, Δn must be strictly less than N, the number of pixels on a line. The variable x is the coordinate of the line used in the computation. Note the dramatic difference between the shape of the functions shown in Figs. 1.1(e) and (f). Their shapes can be qualitatively related to the structure in the images in Figs. 1.1(a) and (b).This relationship is particularly noticeable in Fig. 1.1 (f), where the high correlation between pixels separated by 45 and 90 samples can be directly related to the spacing between the vertically oriented matches of Fig. 1.1(b). In addition, the adjacent pixels of both images are highly correlated. When Δn is 1, γ is 0.9922 and 0.9928 for the images of Figs. 1.1 (a) and (b), respectively. These values are typical of most properly sampled television images.

These illustrations reflect another important form of data redundancy—one directly related to the interpixel correlations within an image. Because the value of any given pixel can be reasonably predicted from the value of its neighbors, the information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the basis of the values of its neighbors. A variety of names, including spatial redundancy, geometric redundancy, and interframe redundancy, have been coined to refer to these interpixel dependencies. We use the term interpixel redundancy to encompass them all.

In order to reduce the interpixel redundancies in an image, the

2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient (but usually "nonvisual") format. For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type (that is, those that remove interpixel redundancy) are referred to as mappings. They are called reversible mappings if the original image elements can be reconstructed from the transformed data set.

Psychovisual Redundancy:

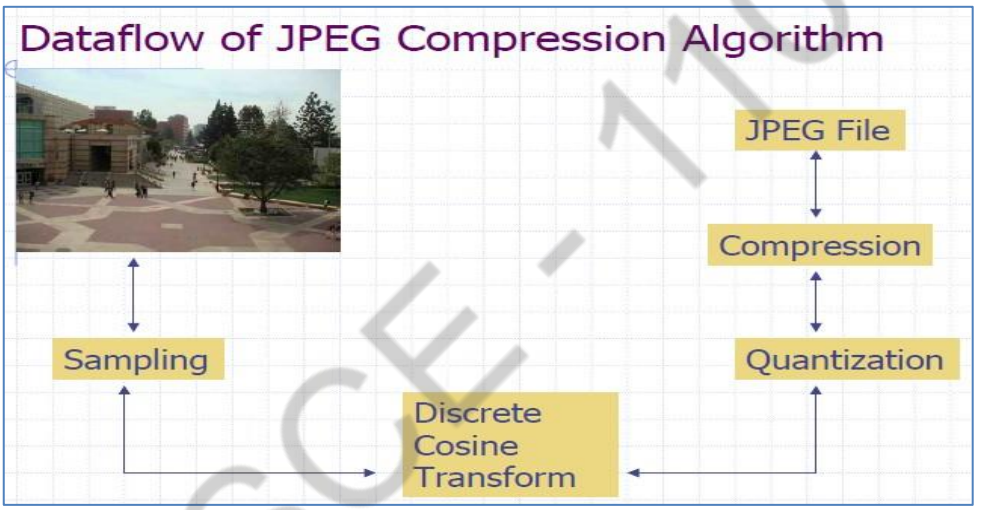
The brightness of a region, as perceived by the eye, depends on factors other than simply the light reflected by the region. For example, intensity variations (Mach bands) can be perceived in an area of constant intensity. Such phenomena result from the fact that the eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processing. This information is said to be psychovisually redundant. It can be eliminated without significantly impairing the quality of image perception.

That psychovisual redundancies exist should not come as a surprise, because human perception of the information in an image normally does not involve quantitative analysis of every pixel value in the image. In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings. The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process. Psychovisual redundancy is fundamentally different from the redundancies discussed earlier. Unlike coding and interpixel redundancy, psychovisual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psychovisually redundant data results in a loss of quantitative information, it is commonly referred to as quantization.

This terminology is consistent with normal usage of the word, which generally means the mapping of a broad range of input values to a limited number of output values. As it is an irreversible operation (visual information is lost), quantization results in lossy data compression.

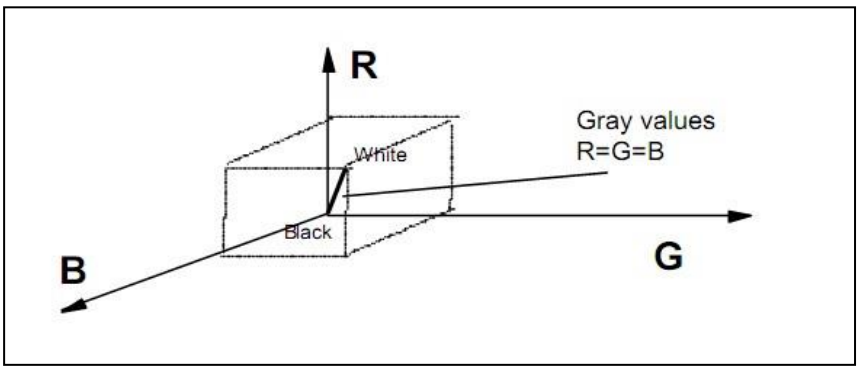
10. With the aid of the diagram explain how individual 8x8 blocks of pixel values are derived by the image and block preparation stage for a monochrome and RGB image

Why 8x8 Blocks



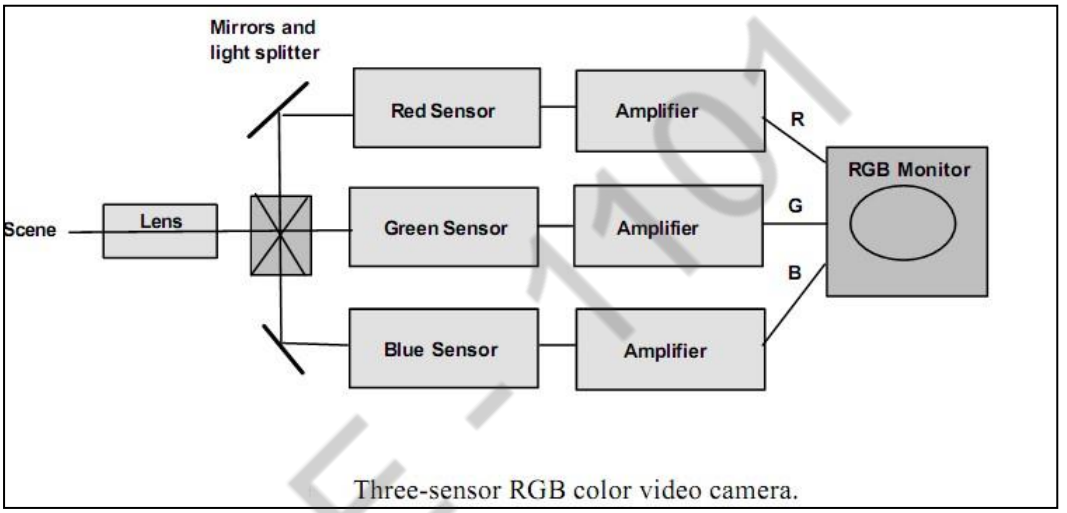
RGB color system

- Three component representation of the color of a pixel
- Represents the intensities of the red, green, and blue components
- 24 bit "True Color"
- Each component represented with 8 bits of precision
- The components each contain roughly the same amount of information

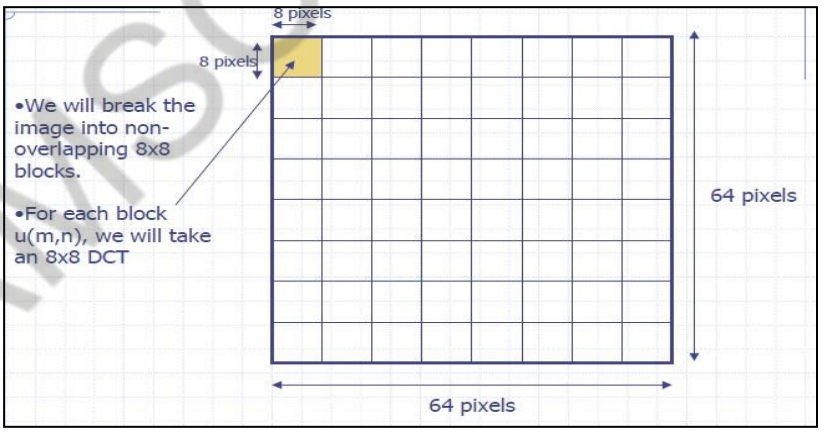


YUV Color Space

- An ideal format for JPEG compression
- The brightness and color information in an image are separated
- Concentrates the most important info into one component, allowing for greater compression
- Y component represents the color intensity of the image (equivalent to a black and white television signal)
- U and V represent the relative redness and blueness of the image



Three-sensor RGB color video camera.



oY = 0.299R + 0.587G + 0.114B
oU = -0.1687R -0.3313G + 0.5B + 128
oV = 0.5R –0.4187G – 0.0813B + 128

11. Design a Huffman code and find average length for a source that puts letters from an alphabet A={a1,a2,a3,a4,a5} with P(a1)=P(a3)=P(a4)=0.1,P(a2)=0.3 and P(a5)=0.4

a1=0.1 a2=0.3 a3=0.1 a4=0.1 a5=0.4
Path length
a1=0.1=4 a2=0.3=2 a3=0.1=4 a4=0.1=3 a5=0.4=1
Optimal path will be
= (4*0.1) + (2*0.3) + (4*0.1) + (3*0.1) + (1*0.4)
=0.4+0.6+0.4+0.3+0.4=2.1

12. Describe dynamic Huffman code for the same output source with the above probabilities.

This coding scheme presupposes a previous determination of the symbol distribution. The actual algorithm starts with this distribution which is regarded as constant about the entire data. If the symbol distribution changes, then either losses in compression or a completely new construction of the code tree must be accepted (incl. header data required).In the following an example for a simple code tree is presented together with some principal considerations.
Example: "abracadabra" Symbol Frequency

a 5
b 2
r 2
c 1
d 1

According to the outlined coding scheme the symbols "d" and "c" will be coupled together ina first step. The new interior node will get the frequency 2.
1. Step

Symbol Frequency Symbol Frequency

a 5 a 5
b 2 b 2
r 2 r 2
c 1 ----------> 1 2
d 1

2. Step

Symbol Frequency Symbol Frequency a 5 a 5
b 2 b 2
r 2 ----------> 2 4
1 2

3. Step

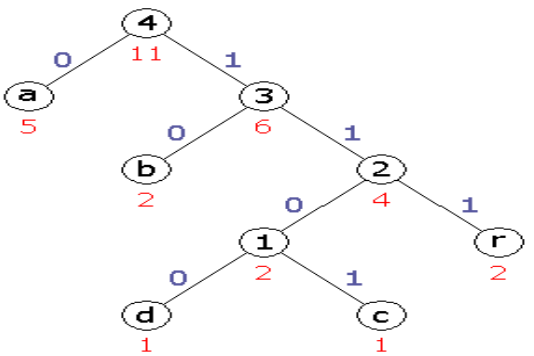
Symbol Frequency Symbol Frequency
a 5 a 5
2 4 ----------> 3 6
b 2

4. Step
Symbol Frequency Symbol Frequency
3 6 ----------> 4 11 a 5

Code Table
If only one single node is remaining within the table, it forms the root ofthe Huffman tree. The paths from the root node to the leaf nodes define the code word used for the corresponding symbol:

Symbol	Frequency	Code Word
a	5	0
b	2	10
r	2	111
c	1	1101
d	1	1100

Complete Huffman Tree:



DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

EC8002 – MULTIMEDIA COMPRESSION & COMMUNICATION

UNIT I – Audio Compression

Part A

9. Define frequency masking.

Auditory masking occurs when the perception of one sound is affected by the presence of another sound. Auditory masking in the frequency domain is known as simultaneous masking, frequency masking or spectral masking.

2. Define is coding redundancy?

If the gray level of an image is coded in a way that uses more code words than necessary to represent each gray level, then the resulting image is said to contain coding redundancy.

3. Define interpixel redundancy?

The value of any given pixel can be predicted from the values of its neighbors. The information carried by is small. Therefore the visual contribution of a single pixel to an image is redundant. Otherwise called as spatial redundant geometric redundant or interpixel redundant.

Eg: Run length coding

4. Define the term 'run length coding'

RunLengthEncoding(RLE)isverysimpleformofdatacompression inwhich consecutive sequences of the same datavalue are stored or transmitted as asingle data value and count**.**

5. **Define the term compression ratio.**

If the total number of bits required to represent the data before compression is B0 and the total number of bits required to represent the data after compression B1, then compression ratio.

6. Bring out the difference between loseless and lossy compression.

S.No	Lossless compression	Lossy compression
1.	In lossless compression,original data is exactly restored after decompression.	Inlossycompression,originaldataisnot exactly restored after decompression.
2.	Mainly used for text data compression & decompression	Mainlyusedforimagedatacompression& decompression
3.	Compression ratio is less.	Compression ratio is high.
4.	Ex: Run length coding, Huffman coding, Arithmetic coding.	Ex: Wavelet transform, Discrete cosine transform.

7. Differentiate static and dynamic coding with respect to text compression.

S.No	Static coding	Dynamic coding
1.	In static coding, the shortest	In dynamic coding, codewords used can
	codewords are used for	change as when transfer takes place.
2.	Staticcoding isappliedfor	Indynamiccoding,thereceiverisableto

8. When does a codeword said to have prefix property?

A code is said to have the prefix property if no codeword is a prefix of any other codeword.

9. Define the procedure for Huffman shift.

List all the source symbols along with its probabilities in descending order.Divide the total number of symbols into block of equal size. Sum the probabilities of all the source symbols outside the reference block. Now apply the procedure for reference block, including the prefix source symbol. The code words for the remaining symbols can be constructed by means of one or more prefix code followed by the reference block as in the case of binary shift code.

10. Define arithmetic coding

In arithmetic coding one to one corresponds between source symbols and code word doesn't exist where as the single arithmetic code word assigned for a sequence of source symbols. A code word defines an interval of number between 0 and 1.

11. State the principles of dynamic Huffman coding.

In this method, the transmitter and receiver develop the Huffman coding tree dynamically depending upon the characters in the data stream. If the character is not present in the data stream, then it is transmitted in its uncompressed form. The receiver identifies this character and adds to the coding tree. If the character is present in the coding tree, then its codeword is transmitted. Depending upon this codewords, the receiver identifies the character. The transmitter and receiver then both increment the frequency of occurrence of such character. The position of the character in Huffman tree is adjusted depending upon its frequency of occurrence. This is also called adaptive Huffman coding.

12. Compare Static and Dynamic Coding.

S.No	Static Coding	Dynamic coding
1	Code words are fixed throughout compression	Codewords change dynamically during compression
2	Statical characteristics of the data are known	Statical characteristics of the data are not known
3	Receiver knows the set of codewords	Receiver dynamically calculates the codewords
4	Example: Static Huffman Coding	Example: Dynamic Huffman Coding

13. Compare Huffman and Arithmetic Coding.

S.No.	Huffman Coding	Arithmetic Coding
1	Codes for the characters are derived.	Coding is done for messages of short lengths.
2	Shannan's rate is achieved only if character probabilities are all integer powers of ½.	Shannon's rate is always achieved irrespective of probabilities of characters.
3	Precision of the computer does not affect coding.	Precision of the computer determine length of the character string that can be encoded.
4	Huffman coding is the simple technique.	Arithmetic coding is complicated.

14. Compare Huffman and LZ Coding.

S.No.	Huffman	Lempel-Ziv
1	A Huffman encoder takes a block of input characters with fixed length and produces a block of output bits of variable length.	Lempel-Ziv is a variable to fixed length code.
2	Huffman coding is an entropy encoding algorithm used for lossless data compression.	It is the most widely used technique for lossless file compression.

15. What is Lempel-Ziv Welch Code (LZW)?

LZW is a dictionary based compression method. It maps a variable number of symbols to a fixed length code. LZW is a good example of compression or communication schemes that 'transmit the model', rather than 'transmits the data'.

16. Write the drawbacks of LZW.

- Too many bits per word
- Everyone needs a dictionary
- Only works for English text

PART – B

1. **Explain Lempel Ziv Welsh Compression APR/MAY2017 (16) , APR/MAY 2015(6)**

The LZW algorithmis a very common compression technique. This algorithm is typically used in GIF and optionally in PDF and TIFF. Unix's 'compress' command, among other uses. It is lossless, meaning no data is lost when compressing.

LZW compression works by reading a sequence of symbols, grouping the symbols into strings, and converting the strings into codes. Because the codes take up less space than the strings they replace, we get compression. Characteristic features of LZW includes,

LZW ENCODING
PSEUDOCODE

```
1    Initialize table with single character strings
2    P = first input character
3    WHILE not end of input stream
4        C = next input character
5        IF P + C is in the string table
6          P = P + C
7        ELSE
8        output the code for P
9                        add P + C to the string table
10         P = C
11       END WHILE
12   output code for P
```

Compression using LZW

Example 1: Use the LZW algorithm to compress the string: BABAABAAA The steps involved are systematically shown in the diagram below.

Example 1: LZW Compression Step 1

BABAABAAA
P=A
C=empty

ENCODER output code	OUTPUT representing	STRING codeword	TABLE string
66	B	256	BA

Example 1: LZW Compression Step 2

BABAABAAA
P=B
C=empty

ENCODER output code	OUTPUT representing	STRING codeword	TABLE string
66	B	256	BA
65	A	257	AB

Example 1: LZW Compression Step 3

BABAABAAA
P=A
C=empty

ENCODER output code	OUTPUT representing	STRING codeword	TABLE string
66	B	256	BA
65	A	257	AB
256	BA	258	BAA

Example 1: LZW Compression Step 4

BABAABAAA
P=A
C=empty

ENCODER output code	OUTPUT representing	STRING codeword	TABLE string
66	B	256	BA
65	A	257	AB
256	BA	258	BAA
257	AB	259	ABA

Example 1: LZW Compression Step 5

BABAABAAA
P=A
C=A

ENCODER output code	OUTPUT representing	STRING codeword	TABLE string
66	B	256	BA
65	A	257	AB
256	BA	258	BAA
257	AB	259	ABA
65	A	260	AA

Example 1: LZW Compression Step 6

BABAABAAA
P=AA
C=empty

ENCODER output code	OUTPUT representing	STRING codeword	TABLE string
66	B	256	BA
65	A	257	AB
256	BA	258	BAA
257	AB	259	ABA
65	A	260	AA
260	AA		

LZW Decompression Algorithm
PSEUDOCODE

```
1  Initialize table with single character strings
2  OLD = first input code
3  output translation of OLD
4  WHILE not end of input stream
5      NEW = next input code
6    IF NEW is not in the string ta
7     S = translation of OLD
8         S = S + C
9    ELSE
10        S = translation of NEW
11    output S
12   C = first character of S
13   OLD + C to the string table
14   OLD = NEW
15    END WHILE
```


2. A series of message is to be transferred between two computers. The message comprises of the characters A,B,C,D and E. the probabilities of occurrence of the above characters are0.4,0.19,0.16,0.15,and0.1 respectively. Use Huffman coding to obtain a codeword for the above characters. Determine the average number of bits per codeword.

Symbol	0ccurrence	Probability	Codeword
A	1	0.4/5	0
B	1	0.19/5	10
C	1	0.16/5	110
D	1	0.15/5	1110
E	1	0.1/5	1111
SUM	5		

AVERAGE SEARCH LENGTH

Avg L = Σ Li * P (i)

=1/5*1+1/5*2+1/5*3+1/5*4+1/5*4=1+2+3+4+4/5=14/5=2.8

Space required to store the original message=5*8=40 bits

Space required to store the decoded message=14 bits

3. **Explain the operation of LZ compression algorithm. Assume a dictionary of 16,000 words and an average word length of 5 bits; derive the average compression ratio that is achieved relative to using 7 bit ASCII code word.**

Solution:

In general, a dictionary with an index of n bits can contain up to 2^n entries.

Now, assume a dictionary of 16,000 words

2^{14}=16384and hence an index of 14 bits is required

Using 7 bit ASCII codeword and an average of 5 characters per word requires 35bits

Hence compression ratio 35/14= 2.5:1

(ii)With help of diagram identify the five main stages associated with the baseline mode of operation JPEG and give a brief description of the role of each stages

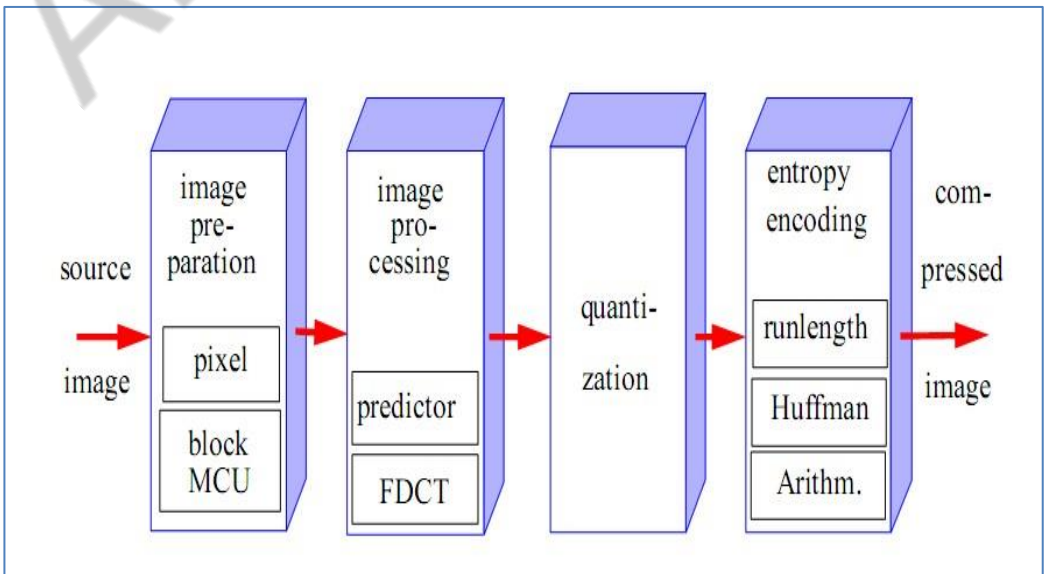
JPEG: Joint Photographic Expert Group

Intermediate Standard

For digital compression and coding of continuous-tone still images

Gray-scale color

- Preparation: analog to digital conversion
- Processing: Transform Data into a domain easier to compress
- Quantization: Reduce precision at which the output is stored
- Entropy encoding: Remove redundant information in the resulting data stream



4. **Explain a method of generating variable length codes with an example.**

Variable-Length Coding:

The simplest approach to error-free image compression is to reduce only coding redundancy. Coding redundancy normally is present in any natural binary encoding of the gray levels in an image. It can be eliminated by coding the gray levels. To do so requires construction of a variable-length code that assigns the shortest possible code words to the most probable gray levels. Here, we examine several optimal and near optimal techniques for constructing such a code. These techniques are formulated in the language of information theory. In practice, the source symbols may be either the gray levels of an image or the output of a gray-level mapping operation (pixel differences, run lengths, and so on).

Huffman coding:

The most popular technique for removing coding redundancy is due to Huffman (Huffman [1952]). When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol. In terms of the noiseless coding theorem, the resulting code is optimal for a fixed value of n, subject to the constraint that the source symbols be coded one at a time.

The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction. Figure 4.1 illustrates this process for binary coding (K-ary Huffman codes can also be constructed). At the far left, a hypothetical set of source symbols and their probabilities are ordered from top to bottom in terms of decreasing probability

values. To form the first source reduction, the bottom two probabilities, 0.06 and 0.04, are combined to form a "compound symbol" with probability 0.1. This compound symbol and its associated probability are placed in the first source reduction column so that the probabilities of the reduced source are also ordered from the most to the least probable. This process is then repeated until a reduced source with two symbols (at the far right) is reached.

The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. The minimal length binary code for a two-symbol source, of course, is the symbols 0 and 1. As Fig. 4.2 shows, these symbols are assigned to the two symbols on the right (the assignment is arbitrary; reversing the order of the 0 and 1 would work just as well). As the reduced source symbol with probability 0.6 was generated by combining two symbols in the reduced source to its left, the 0 used to code it is now assigned to both of these symbols, and a 0 and 1 are arbitrarily

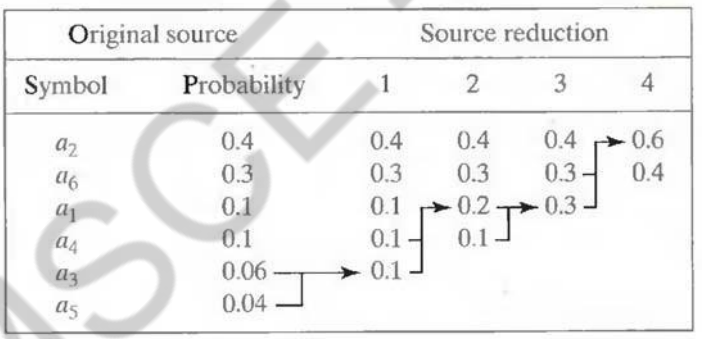


Fig.4.1 Huffman source reductions.

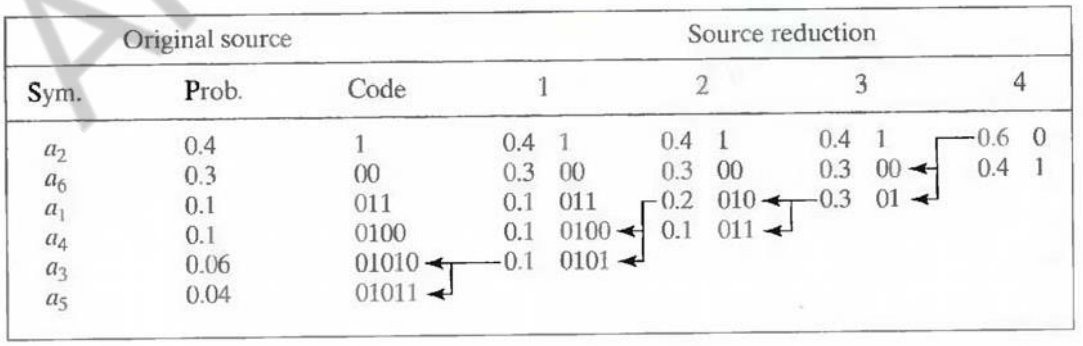


Fig.4.2 Huffman code assignment procedure.

appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original source is reached. The final code appears at

the far left in Fig.4.2. The average length of this code is

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5)$$
$$= 2.2 \text{ bits/symbol}$$

and the entropy of the source is 2.14 bits/symbol. The resulting Huffman code efficiency is 0.973.

Huffman's procedure creates the optimal code for a set of symbols and probabilities subject to the constraint that the symbols be coded one at a time. After the code has been created, coding and/or decoding is accomplished in a simple lookup table manner. The code itself is an instantaneous uniquely decodable block code. It is called a block code because each source symbol is mapped into a fixed sequence of code symbols. It is instantaneous, because each code word in a string of code symbols can be decoded without referencing succeeding symbols. It is uniquely decodable, because any string of code symbols can be decoded in only one way. Thus, any string of Huffman encoded symbols can be decoded by examining the individual symbols of the string in a left to right manner. For the binary code of Fig. 4.2, a left-to-right scan of the encoded string 010100111100 reveals that the first valid code word is 01010, which is the code for symbol a_3 .The next valid code is 011, which corresponds to symbol a_1. Continuing in this manner reveals the completely decoded message to be $a_3a_1a_2a_2a_6$.

5. Explain arithmetic encoding process with an example.

Arithmetic coding:

Unlike the variable-length codes described previously, arithmetic coding generates nonblock codes. In arithmetic coding, which can be traced to the work of Elias, a one-to-one correspondence between source symbols and code words does not exist. Instead, an entire sequence of source symbols (or message) is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1. As the number of symbols in the message increases, the interval used to represent it becomes smaller and the number of information units (say, bits)

required to represent the interval becomes larger. Each symbol of the message reduces the size of the interval in accordance with its probability of occurrence. Because the technique does not require, as does Huffman's approach, that each source symbol translate into an integral number of code symbols (that is, that the symbols be coded one at a time), it achieves (but only in theory) the bound established by the noiseless coding theorem.

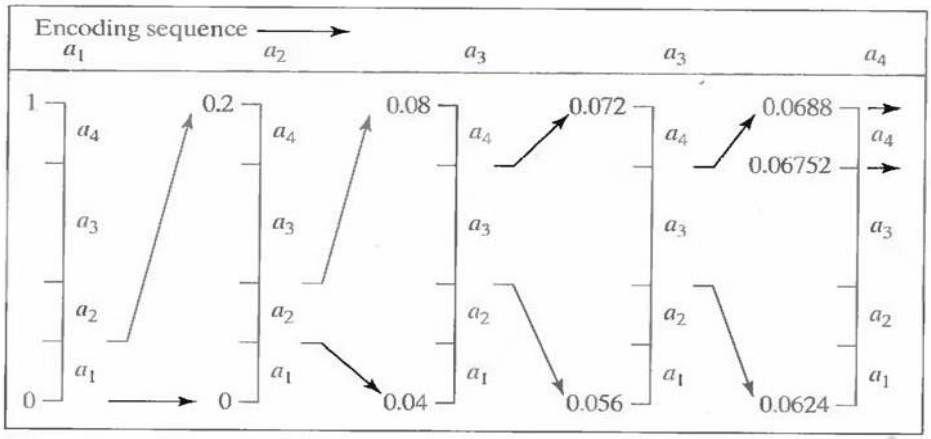


Fig.5.1 Arithmetic coding procedure

Figure 5.1 illustrates the basic arithmetic coding process. Here, a five-symbol sequence or message, $a_1a_2a_3a_3a_4$, from a four-symbol source is coded. At the start of the coding process, the message is assumed to occupy the entire half-open interval [0, 1). As Table 5.2 shows, this interval is initially subdivided into four regions based on the probabilities of each source symbol. Symbol ax, for example, is associated with subinterval [0, 0.2). Because it is the first symbol of the message being coded, the message interval is initially narrowed to [0, 0.2). Thus in Fig. 5.1 [0, 0.2) is expanded to the full height of the figure and its end points labeled by the values of the narrowed range. The narrowed range is then subdivided in accordance with the original source symbol probabilities and the process continues with the next message symbol.

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

Table 5.1 Arithmetic coding example

In this manner, symbol a_2 narrows the subinterval to [0.04, 0.08), a_3 further narrows it to [0.056, 0.072), and so on. The final message symbol, which must be reserved as a special end-of-message indicator, narrows the range to [0.06752, 0.0688). Of course, any number within this subinterval—for example, 0.068—can be used to represent the message.

In the arithmetically coded message of Fig. 5.1, three decimal digits are used to represent the five-symbol message. This translates into 3/5 or 0.6 decimal digits per source symbol and compares favorably with the entropy of the source, which is 0.58 decimal digits or 10-ary

units/symbol. As the length of the sequence being coded increases, the resulting arithmetic code approaches the bound established by the noiseless coding theorem.

In practice, two factors cause coding performance to fall short of the bound: (1) the addition of the end-of-message indicator that is needed to separate one message from an- other; and (2) the use of finite precision arithmetic. Practical implementations of arithmetic coding address the latter problem by introducing a scaling strategy and a rounding strategy (Langdon and Rissanen [1981]). The scaling strategy renormalizes each subinterval to the [0, 1) range before subdividing it in accordance with the symbol probabilities. The rounding strategy guarantees that the truncations associated with finite precision arithmetic do not prevent the coding subintervals from being represented accurately.

6. Explain LZW coding with an example.

LZW Coding:

The technique, called Lempel-Ziv-Welch (LZW) coding, assigns fixed-length code words to variable length sequences of source symbols but requires no a priori knowledge of the probability of occurrence of the symbols to be encoded. LZW compression has been integrated into a variety of mainstream imaging file formats, including the graphic interchange format (GIF), tagged image file format (TIFF), and the portable document format (PDF).

LZW coding is conceptually very simple (Welch [1984]). At the onset of the coding process, a codebook or "dictionary" containing the source symbols to be coded is constructed. For 8-bit monochrome images, the first 256 words of the dictionary are assigned to the gray values 0, 1, 2..., and 255. As the encoder sequentially examines the image's pixels, gray- level sequences that are not in the dictionary are placed in algorithmically determined (e.g., the next unused) locations. If the first two pixels of the image are white, for instance, sequence "255- 255" might be assigned to location 256, the address following the locations reserved for gray levels 0 through 255. The next time that two consecutive white pixels are encountered, code word 256, the address of the location containing sequence 255-255, is used to represent them. If a 9-bit, 512-word dictionary is employed in the coding process, the original (8 + 8) bits that were used to represent the two pixels are replaced by a single 9-bit code word. Cleary, the size of the dictionary is an important system parameter. If it is too small, the detection of matching gray- level sequences will be less likely; if it is too large, the size of the code words will adversely affect compression performance.

Consider the following 4 x 4, 8-bit image of a vertical edge:

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

Table 6.1 details the steps involved in coding its 16 pixels. A 512-word dictionary with the following starting content is assumed:

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

Locations 256 through 511 are initially unused. The image is encoded by processing its pixels in a left-to-right, top-to-bottom manner. Each successive gray-level value is concatenated with a variable—column 1 of Table 6.1 —called the "currently recognized sequence." As can be seen, this variable is initially null or empty. The dictionary is searched for each concatenated sequence and if found, as was the case in the first row of the table, is replaced by the newly concatenated and recognized (i.e., located in the dictionary) sequence. This was done in column 1 of row 2.

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

Table 6.1 LZW coding example

No output codes are generated, nor is the dictionary altered. If the concatenated sequence is not found, however, the address of the currently recognized sequence is output as the next encoded value, the concatenated but unrecognized sequence is added to the dictionary, and the currently recognized sequence is initialized to the current pixel value. This occurred in row 2 of the table. The last two columns detail the gray-level sequences that are added to the dictionary when scanning the entire 4 x 4 image. Nine additional code words are defined. At the conclusion of coding, the dictionary contains 265 code words and the LZW algorithm has successfully identified several repeating gray-level sequences—leveraging them to reduce the original 128-bit image lo 90 bits (i.e., 10 9-bit codes). The encoded output is obtained by reading the third column from top to bottom. The resulting compression ratio is 1.42:1.

A unique feature of the LZW coding just demonstrated is that the coding dictionary or code book is created while the data are being encoded. Remarkably, an LZW decoder builds an identical decompression dictionary as it decodes simultaneously the encoded data stream. . Although not needed in this example, most practical applications require a strategy for handling dictionary overflow. A simple solution is to flush or reinitialize the dictionary when it becomes full and continue coding with a new initialized dictionary. A more complex option is to monitor compression performance and flush the dictionary when it becomes poor or unacceptable. Alternately, the least used dictionary entries can be tracked and replaced when necessary.

7. Explain the concept of bit plane coding method.

Bit-Plane Coding:

An effective technique for reducing an image's interpixel redundancies is to process the image's bit planes individually. The technique, called bit-plane coding, is based on the concept of decomposing a multilevel (monochrome or color) image into a series of binary images and compressing each binary image via one of several well-known binary compression methods.

Bit-plane decomposition:

The gray levels of an m-bit gray-scale image can be represented in the form of the base 2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_1 2^1 + a_0 2^0.$$

Based on this property, a simple method of decomposing the image into a collection of binary images is to separate the m coefficients of the polynomial into m 1-bit bit planes. The zeroth- order bit plane is generated by collecting the a_0 bits of each pixel, while the (m - 1) st-order bit plane contains the a_{m-1}, bits or coefficients. In general, each bit plane is numbered from 0 to m-1 and is constructed by setting its pixels equal to the values of the appropriate bits or polynomial coefficients from each pixel in the original image. The inherent disadvantage of this approach is that small changes in gray level can

have a significant impact on the complexity of the bit planes. If a pixel of intensity 127 (01111111) is adjacent to a pixel of intensity 128 (10000000), for instance, every bit plane will contain a corresponding 0 to 1 (or 1 to 0) transition. For example, as the most significant bits of the two binary codes for 127 and 128 are different, bit plane 7 will contain a zero-valued pixel next to a pixel of value 1, creating a 0 to 1 (or 1 to 0) transition at that point.

An alternative decomposition approach (which reduces the effect of small gray-level variations) is to first represent the image by an m-bit Gray code. The m-bit Gray code $g_{m-1}...\ g_2g_1g_0$ that corresponds to the polynomial in Eq. above can be computed from

$$g_i = a_i \oplus a_{i+1} \qquad 0 \le i \le m-2$$
$$g_{m-1} = a_{m-1}.$$

Here, ⓧ denotes the exclusive OR operation. This code has the unique property that successive code words differ in only one bit position. Thus, small changes in gray level are less likely to affect all m bit planes. For instance, when gray levels 127 and 128 are adjacent, only the 7th bit plane will contain a 0 to 1 transition, because the Gray codes that correspond to 127 and 128 are 11000000 and 01000000, respectively.

AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

EC8002 – MULTIMEDIA COMPRESSION & COMMUNICATION

UNIT IV – Guaranteed Service Model

Part A

1. **What are the responsibilities of interface and information designers in the development of a multimedia project?**
   - ✓ An interface designer is responsible for: i) creating software device that organizes content, allows users to access or modify content and present that content on the screen, ii) building a user friendlyinterface.
   - ✓ Information designers, who structure content, determine user pathways and feedback and select presentationmedia.
2. **List the features of multimedia.**
   - ✓ A Multimedia system has four basiccharacteristics:

- Multimedia systems must be computercontrolled.
- Multimedia systems areintegrated.
- The information they handle must be representeddigitally.
- The interface to the final presentation of media is usuallyinteractive.

3. **What are the multimedia components?**

   ✓ Text, Audio, Images, Animations, Video and interactive content are the multimediacomponents.

   ✓ The first multimedia element is text. Text is the most common multimediaelement.

4. **Definemultimedia.**

   ✓ 'Multi' means 'many' and 'media' means 'material through which something can be transmitted or send'.

   ✓ Information being transferred by more than one medium is called asmultimedia.

- ✓ It is the combination of text, image, audio, video, animation, graphic & hardware, that can be delivered electronically / digitally which can be accessedinteractively.
- ✓ It is of two types: Linear & Non –Linear.

5. Describe the applications ofmultimedia.

- ✓ Multimedia in Education: It is commonly used to prepare study material for the students and also provide them proper understanding of differentsubjects.
- ✓ Multimedia inEntertainment:

a) Movies: Multimedia used in movies gives a special audio and videoeffect.

b) Games: Multimedia used in games by using computer graphics, animation, videos has changed the gamingexperience.

- ✓ Multimedia inBusiness:

a) Videoconferencing: This system enables to communicate using audio and video between two different locations through theircomputers.

b) Marketing and advertisement: Different advertisement and marketing ideas about any product on television and internet is possible withmultimedia.

6. Write the difference between multimedia andhypermedia.

S.No	Multimedia	Hypermedia
1.	Multimedia is the presentation of media as text, images, graphics, video & audio by the use of computers or the information contentprocessing devices.	Hypermedia is the use of advanced form of hypertext like interconnected systems to store and present text, graphics & other media types where the content is linked to each other by hyperlinks.

2.	Multimedia can be in linear or non-linear content format, but the hypermedia is only in non-linear content format.	Hypermedia is an application of multimedia, hence a subset ofmultimedia.

7. Why multimedia networking is needed

The importance of communications or networking for multimedia lies in the new applications that will be generated by adding networking capabilities to multimedia computers, and hopefully gains in efficiency and cost of ownership and use when multimedia resources are part of distributed computing systems.

8. Mention the applications of multimedianetworking.

- StreamingVideo
- IPtelephony
- Internet Radio
- Teleconferencing
- Interactive games
- Virtualworlds
- Multimediaweb

9. What are the scheduling mechanisms used in multimedia networking?

1. First In First Out (FIFO)
2. Priority Queuing
3. Round Robin and Weigthed Fair Queuing

10. What are the policing criteria used in multimedia networking?

Three policing criteria are used and each differing from the other, according to the time scale over which the packet flows is policed.

1. Average Rate
2. Peak Rate
3. Burst Size

11. Name the two key features of integrated services architecture and define them.

Two key features of integrated service architecture is

1. Reserved Resources
2. Call setup

Reserved Resources:

A router is required to know what amounts of its Resources are already Reserved for ongoing sessions.

Call setup:

Call setup also known as call admission it requires the participation of each router on the path.

12. Define multicast overlay networks.

Multicast overlay network consists of servers scatted throughout the ISP network. These servers and the logical links between them collectively form an overlay network which multicasts traffic from the source to the millions of users.

13. Write the principal characteristics of RSVP.

1. It provides reservations for bandwidth in multicast trees.
2. It is receiver oriented that is, the receiver of a data flow initiates and maintains the resource reservation used for that flow.

14. What are the drawbacks of RTSP?

RTSP does not define compression schemes for audio and video.

RTSP does not define how audio and video are encapsulated in packets for transmission over a network.

RTSP does not restrict how the media player buffers the audio/video.

RTSP does not restrict how streamed media is transported.

15. Difference between RTSP and RSVP.

Sl. No.	RTSP	RSVP
1	It is an out of band protocol	It is a signaling protocol
2	It is used for media player	It is used for the internet
3	It sends a message over TCP or UDP	It sends a message using router buffers

PART B

1. **Discuss about the Best effort service model and its drawback in detail.**

Real-time conversational voice over the Internet is often referred to as **Internet telephony**, since, from the user's perspective, it is similar to the traditional circuit-switched telephone service. It is also commonly called **Voice-over-IP (VoIP)**. In this section we describe the principles and protocols underlying VoIP. Conversational video is similar in many respects to VoIP, except that it includes the video of the participants as well as their voices. To keep the discussion focused and concrete, we focus here only on voice in this section rather than combined voice and video.

The Internet's network-layer protocol, IP, provides best-effort service. That is to say the service makes its best effort to move each datagram from source to destination as quickly as possible but makes no promises whatsoever about getting the packet to the destination within some delay bound or about a limit on the percentage of packets lost. The lack of such guarantees poses significant challenges to the design of real-time conversational applications, which are acutely sensitive to packet delay, jitter, and loss.

In this section, we'll cover several ways in which the performance of VoIP over a best-effort network can be enhanced. Our focus will be on application-layer techniques, that is, approaches that do not require any changes in the network core or even in the transport layer at the end hosts. To keep the discussion concrete, we'll discuss the limitations of best-effort IP service in the context of a specific VoIP example. The sender generates bytes at a rate of 8,000 bytes per second; every 20 msecs the sender gathers these bytes into a chunk. A chunk and a special header (discussed below) are encapsulated in a UDP segment, via a call to the socket interface. Thus, the number of bytes in a chunk is (20 msecs)·(8,000 bytes/sec) = 160 bytes, and a UDP segment is sent every 20 msecs.

If each packet makes it to the receiver with a constant end-to-end delay, then packets arrive at the receiver periodically every 20 msecs. In these ideal conditions, the receiver can simply play back each chunk as soon as it arrives. But unfortunately, some packets can be lost and most packets will not have the same end-to-end delay, even in a lightly congested Internet. For this reason, the receiver must take more care in determining (1) when to play back a chunk, and (2) what to do with a missing chunk.

Packet Loss

Consider one of the UDP segments generated by our VoIP application. The UDP segment is encapsulated in an IP datagram. As the datagram wanders through the network, it passes through router buffers (that is, queues) while waiting for transmission on outbound links. It is possible that one or more of the buffers in the path from sender to receiver is full, in which case the arriving IP datagram may be discarded, never to arrive at the receiving application.

Loss could be eliminated by sending the packets over TCP (which provides for reliable data transfer) rather than over UDP. However, retransmission mechanisms are often considered unacceptable for conversational real-time audio applications such as VoIP, because they increase end-to-end delay [Bolot 1996]. Furthermore, due to TCP congestion control, packet loss may result in a reduction of the TCP sender's transmission rate to a rate that is lower than the receiver's drain rate, possibly leading to buffer starvation. This can have a severe impact on voice intelligibility at the receiver. For these reasons, most existing VoIP applications run over UDP by default. [Baset 2006] reports that UDP is used by Skype unless a user is behind a NAT or firewall that blocks UDP segments (in which case TCP is used).

But losing packets is not necessarily as disastrous as one might think. Indeed, packet loss rates between 1 and 20 percent can be tolerated, depending on how voice is encoded and transmitted, and on how the loss is concealed at the receiver. For example, forward error correction (FEC) can help conceal packet loss. We'll see below that with FEC, redundant information is transmitted along with the original information so that some of the lost original data can be recovered from the redundant information. Nevertheless, if one or more of the links between sender and receiver is severely congested, and packet loss exceeds 10 to 20 percent (for example, on a wireless link), then there is really nothing that can be done to achieve acceptable audio quality. Clearly, best-effort service has its limitations.

End-to-End Delay

End-to-end delay is the accumulation of transmission, processing, and queuing delays in routers; propagation delays in links; and end-system processing delays. For real-time conversational applications, such as VoIP, end-to-end delays smaller than 150 msecs are not perceived by a human listener; delays between 150 and 400 msecs can be acceptable but are not ideal; and delays exceeding 400 msecs can seriously hinder the interactivity in voice conversations. The receiving side of a VoIP application will typically disregard any packets that are delayed more than a certain threshold, for example, more than 400 msecs. Thus, packets that are delayed by more than the threshold are effectively lost.

Packet Jitter

A crucial component of end-to-end delay is the varying queuing delays that a packet experiences in the network's routers. Because of these varying delays, the time from when a packet is generated at the source until it is received at the receiver can fluctuate from packet to packet, as shown in Figure 7.1. This phenomenon is called **jitter**. As an example, consider two consecutive packets in our VoIP application. The sender sends the second packet 20 msecs after sending the first packet. But at the receiver, the spacing between these packets can become greater than 20 msecs. To see this, suppose the first packet arrives at a nearly empty queue at a router, but just before the second packet arrives at the queue a large number of packets from other sources

arrive at the same queue. Because the first packet experiences a small queuing delay and the second packet suffers a large queuing delay at this router, the first and second packets become spaced by more than 20 msecs. The spacing between consecutive packets can also become less than 20 msecs. To see this, again consider two consecutive packets. Suppose the first packet joins the end of a queue with a large number of packets, and the second packet arrives at the queue before this first packet is transmitted and before any packets from other sources arrive at the queue. In this case, our two packets find themselves one right after the other in the queue. If the time it takes to transmit a packet on the router's outbound link is less than 20 msecs, then the spacing between first and second packets becomes less than 20 msecs.

The situation is analogous to driving cars on roads. Suppose you and your friend are each driving in your own cars from San Diego to Phoenix. Suppose you and your friend have similar driving styles, and that you both drive at 100 km/hour, traffic permitting. If your friend starts out one hour before you, depending on intervening traffic, you may arrive at Phoenix more or less than one hour after your friend.

If the receiver ignores the presence of jitter and plays out chunks as soon as they arrive, then the resulting audio quality can easily become unintelligible at the receiver. Fortunately, jitter can often be removed by using **sequence numbers**, **timestamps**, and a **playout delay**, as discussed below.

2. Explain in detail about the Scheduling policies.

First-In-First-Out (FIFO)

Figure 7.17 shows the queuing model abstractions for the FIFO link-scheduling discipline. Packets arriving at the link output queue wait for transmission if the link is currently busy transmitting another packet. If there is not sufficient buffering space to hold the arriving packet, the queue's **packet-discarding policy** then determines whether the packet will be dropped (lost) or whether other packets will be removed from the queue to make space for the arriving packet. In our discussion below, we will ignore packet discard. When a packet is completely transmitted over the outgoing link (that is, receives service) it is removed from the queue.

The FIFO (also known as first-come-first-served, or FCFS) scheduling discipline selects packets for link transmission in the same order in which they arrived at the output link queue. We're all familiar with FIFO queuing from bus stops (particularly in England, where queuing seems to have been perfected) or other service centers, where arriving customers join the back of the single waiting line, remain in order, and are then served when they reach the front of the line.

Figure 7.18 shows the FIFO queue in operation. Packet arrivals are indicated by numbered arrows above the upper timeline, with the number indicating the order

in which the packet arrived. Individual packet departures are shown below the lower timeline. The time that a packet spends in service (being transmitted) is indicated by the shaded rectangle between the two timelines. Because of the FIFO discipline, packets leave in the same order in which they arrived. Note that after the departure of packet 4, the link remains idle (since packets 1 through 4 have been transmitted and removed from the queue) until the arrival of packet 5.

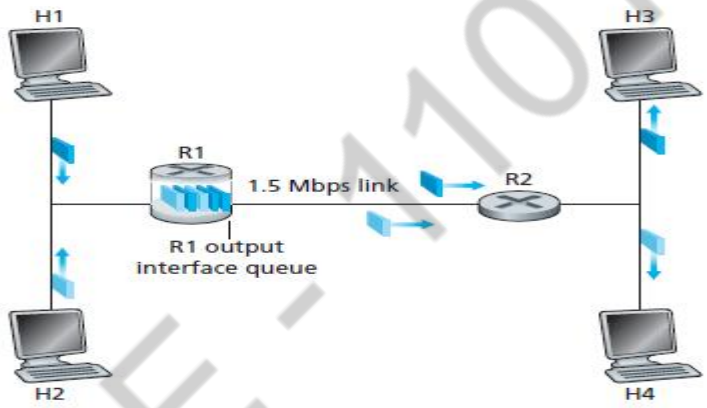


Figure 7.17 ◆ FIFO queuing abstraction

Priority Queuing

Under **priority queuing**, packets arriving at the output link are classified into priority classes at the output queue, as shown in Figure 7.19. As discussed in the previous section, a packet's priority class may depend on an explicit marking that it carries in its packet header (for example, the value of the ToS bits in an IPv4 packet), its source or destination IP address, its destination port number, or other criteria. Each priority class typically has its own queue. When choosing a packet to transmit, the priority queuing discipline will transmit a packet from the highest priority class that has a nonempty queue (that is, has packets waiting for transmission). The choice among packets *in the same priority class* is typically done in a FIFO manner.

Figure 7.20 illustrates the operation of a priority queue with two priority classes. Packets 1, 3, and 4 belong to the high-priority class, and packets 2 and 5 belong to the low-priority class. Packet 1 arrives and, finding the link idle, begins transmission. During the transmission of packet 1, packets 2 and 3 arrive and are queued in the low- and high-priority queues, respectively. After the transmission of packet 1, packet 3 (a high-priority packet) is selected for transmission over packet 2 (which, even though it arrived earlier, is a low-priority packet). At the end of the transmission of packet 3, packet 2 then begins transmission. Packet 4 (a high-priority packet) arrives during the transmission of packet 2 (a low-priority packet). Under a nonpreemptive priority queuing discipline, the transmission of

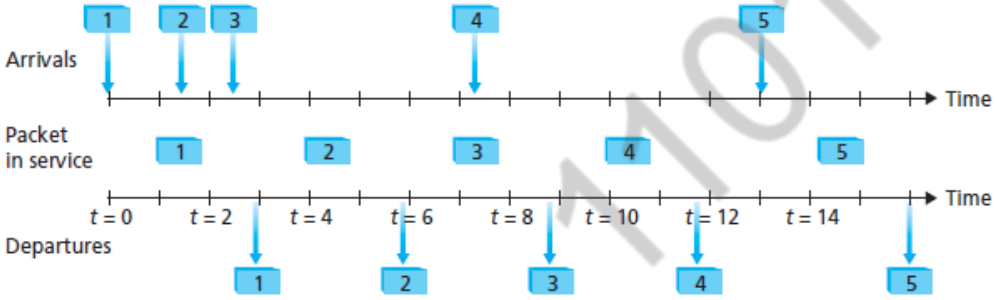


Figure 7.18 ♦ The FIFO queue in operation

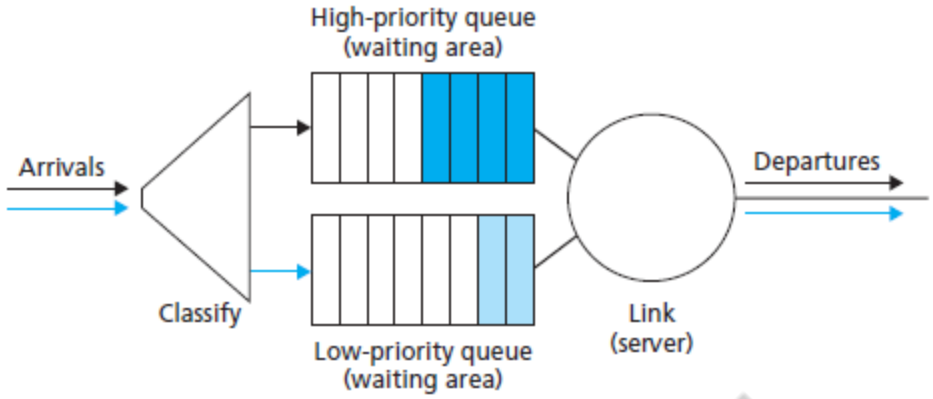


Figure 7.19 ♦ Priority queuing model

a packet is not interrupted once it has begun. In this case, packet 4 queues for transmission and begins being transmitted after the transmission of packet 2 is completed.

Round Robin and Weighted Fair Queuing (WFQ)

Under the **round robin queuing discipline**, packets are sorted into classes as with priority queuing. However, rather than there being a strict priority of service among classes, a round robin scheduler alternates service among the classes. In the simplest form of round robin scheduling, a class 1 packet is transmitted, followed by a class 2 packet, followed by a class 1 packet, followed by a class 2 packet, and so on. A so-called work-conserving queuing discipline will never allow the link to remain idle whenever there are packets (of any class) queued for

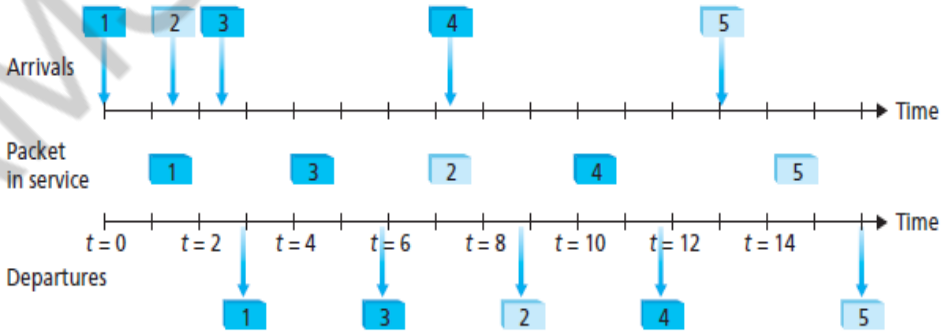


Figure 7.20 ♦ Operation of the priority queue

transmission. A **work-conserving round robin discipline** that looks for a packet of a given class but finds none will immediately check the next class in the round robin sequence.

Figure 7.21 illustrates the operation of a two-class round robin queue. In this example, packets 1, 2, and 4 belong to class 1, and packets 3 and 5 belong to the second class. Packet 1 begins transmission immediately upon arrival at the output queue. Packets 2 and 3 arrive during the transmission of packet 1 and thus queue for transmission. After the transmission of packet 1, the link scheduler looks for a class 2 packet and thus transmits packet 3. After the transmission of packet 3, the scheduler looks for a class 1 packet and thus transmits packet 2. After the transmission of packet 2, packet 4 is the only queued packet; it is thus transmitted immediately after packet 2.

A generalized abstraction of round robin queuing that has found considerable use in QoS architectures is the so-called **weighted fair queuing** (WFQ) discipline [Demers 1990; Parekh 1993]. WFQ is illustrated in Figure 7.22. Arriving packets are classified and queued in the appropriate per-class waiting area. As in round robin scheduling, a WFQ scheduler will serve classes in a circular manner—first serving class 1, then serving class 2, then serving class 3, and then (assuming there are three classes) repeating the service pattern. WFQ is also a work-conserving queuing discipline and thus will immediately move on to the next class in the service sequence when it finds an empty class queue.

WFQ differs from round robin in that each class may receive a *differential* amount of service in any interval of time. Specifically, each class, i, is assigned a weight, w_i. Under WFQ, during any interval of time during which there are class i packets to send, class i will then be guaranteed to receive a fraction of service equal to $w_i/(\sum w_j)$, where the sum in the denominator is taken over all classes that also have packets queued for transmission. In the worst case, even if all classes have queued packets, class i will still be guaranteed to receive a fraction $w_i/(\sum w_j)$ of the

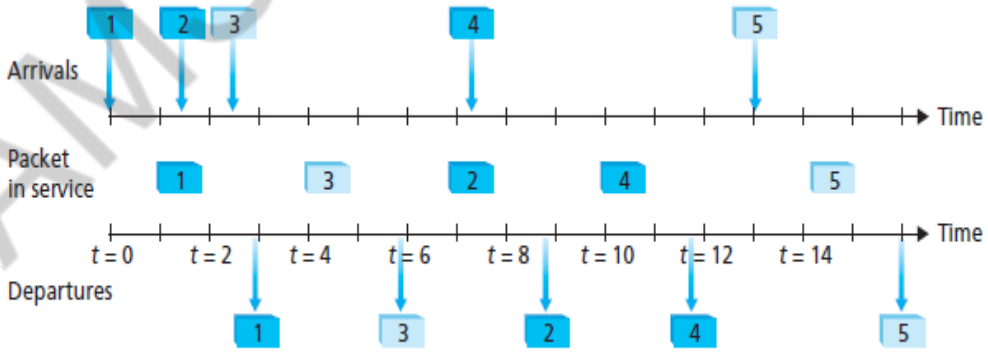


Figure 7.21 ♦ Operation of the two-class round robin queue

bandwidth. Thus, for a link with transmission rate R, class i will always achieve a throughput of at least $R \cdot w_i/(\sum w_j)$. Our description of WFQ has been an idealized one, as we have not considered the fact that packets are discrete units of data and a packet's transmission will not be interrupted to begin transmission of another packet; [Demers 1990] and [Parekh 1993] discuss this packetization issue. As we will see in the following sections, WFQ plays a central role in QoS architectures. It is also available in today's router products [Cisco QoS 2012].

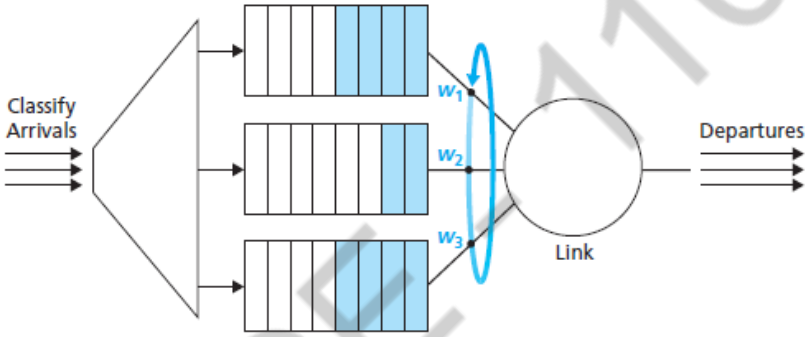


Figure 7.22 ♦ Weighted fair queuing (WFQ)

3. Explain in detail about the Leaky Bucket policies

Policing: The Leaky Bucket

One of our earlier insights was that policing, the regulation of the rate at which a class or flow (we will assume the unit of policing is a flow in our discussion below) is allowed to inject packets into the network, is an important QoS mechanism. But what aspects of a flow's packet rate should be policed? We can identify three important policing criteria, each differing from the other according to the time scale over which the packet flow is policed:

- *Average rate.* The network may wish to limit the long-term average rate (packets per time interval) at which a flow's packets can be sent into the network. A crucial issue here is the interval of time over which the average rate will be policed. A flow whose average rate is limited to 100 packets per second is more constrained than a source that is limited to 6,000 packets per minute, even though both have the same average rate over a long enough interval of time. For example, the latter constraint would allow a flow to send 1,000 packets in a given second-long interval of time, while the former constraint would disallow this sending behavior.

- *Peak rate.* While the average-rate constraint limits the amount of traffic that can be sent into the network over a relatively long period of time, a peak-rate constraint limits the maximum number of packets that can be sent over a shorter period of time. Using our example above, the network may police a flow at an average rate of 6,000 packets per minute, while limiting the flow's peak rate to 1,500 packets per second.
- *Burst size.* The network may also wish to limit the maximum number of packets (the "burst" of packets) that can be sent into the network over an extremely short interval of time. In the limit, as the interval length approaches zero, the burst size limits the number of packets that can be instantaneously sent into the network. Even though it is physically impossible to instantaneously send multiple packets into the network (after all, every link has a physical transmission rate that cannot be exceeded!), the abstraction of a maximum burst size is a useful one.

The leaky bucket mechanism is an abstraction that can be used to characterize these policing limits. As shown in Figure 7.23, a leaky bucket consists of a bucket that can hold up to b tokens. Tokens are added to this bucket as follows. New tokens, which may potentially be added to the bucket, are always being generated at a rate of r tokens per second. (We assume here for simplicity that the unit of time is a second.) If the bucket is filled with less than b tokens when a token is generated, the newly generated token is added to the bucket; otherwise the newly generated token is ignored, and the token bucket remains full with b tokens.

Let us now consider how the leaky bucket can be used to police a packet flow. Suppose that before a packet is transmitted into the network, it must first remove a

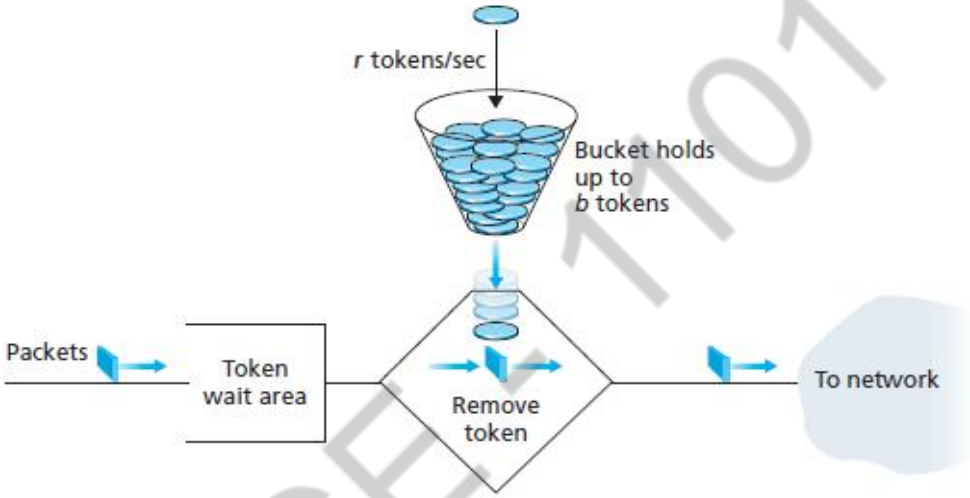


Figure 7.23 ♦ The leaky bucket policer

token from the token bucket. If the token bucket is empty, the packet must wait for a token. (An alternative is for the packet to be dropped, although we will not consider that option here.) Let us now consider how this behavior polices a traffic flow. Because there can be at most b tokens in the bucket, the maximum burst size for a leaky-bucket-policed flow is b packets. Furthermore, because the token generation rate is r, the maximum number of packets that can enter the network of *any* interval of time of length t is $rt + b$. Thus, the token-generation rate, r, serves to limit the long-term average rate at which packets can enter the network. It is also possible to use leaky buckets (specifically, two leaky buckets in series) to police a flow's peak rate in addition to the long-term average rate; see the homework problems at the end of this chapter.

Leaky Bucket + Weighted Fair Queuing = Provable Maximum Delay in a Queue

Let's close our discussion of scheduling and policing by showing how the two can be combined to provide a bound on the delay through a router's queue. Let's consider a router's output link that multiplexes n flows, each policed by a leaky bucket with parameters b_i and r_i, $i = 1, \ldots, n$, using WFQ scheduling. We use the term *flow* here loosely to refer to the set of packets that are not distinguished from each other by the scheduler. In practice, a flow might be comprised of traffic from a single end-to-end connection or a collection of many such connections, see Figure 7.24.

Recall from our discussion of WFQ that each flow, i, is guaranteed to receive a share of the link bandwidth equal to at least $R \cdot w_i/(\sum w_j)$, where R is the transmission

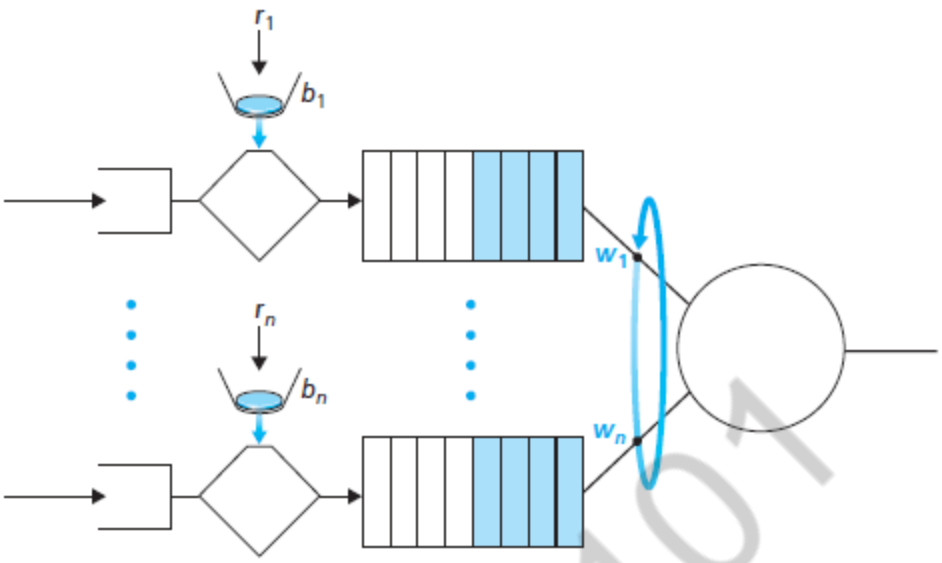


Figure 7.24 ♦ *n* multiplexed leaky bucket flows with WFQ scheduling

rate of the link in packets/sec. What then is the maximum delay that a packet will experience while waiting for service in the WFQ (that is, after passing through the leaky bucket)? Let us focus on flow 1. Suppose that flow 1's token bucket is initially full. A burst of b_1 packets then arrives to the leaky bucket policer for flow 1. These packets remove all of the tokens (without wait) from the leaky bucket and then join the WFQ waiting area for flow 1. Since these b_1 packets are served at a rate of at least $R \cdot w_i/(\sum w_j)$ packet/sec, the last of these packets will then have a maximum delay, $d_{\max,}$ until its transmission is completed, where

$$d_{\max} = \frac{b_1}{R \cdot w_1 / \sum w_j}$$

The rationale behind this formula is that if there are b_1 packets in the queue and packets are being serviced (removed) from the queue at a rate of at least $R \cdot w_1/(\sum w_j)$ packets per second, then the amount of time until the last bit of the last packet is transmitted cannot be more than $b_1/(R \cdot w_1/(\sum w_j))$. A homework problem asks you to prove that as long as $r_1 < R \cdot w_1/(\sum w_j)$, then $d_{\max}$ is indeed the maximum delay that any packet in flow 1 will ever experience in the WFQ queue.

4. Explain in detail about the Resource Reservation and Call admission.

we have seen that packet marking and policing, traffic isolation, and link-level scheduling can provide one class of service with better performance than another. Under certain scheduling disciplines, such as priority scheduling, the lower classes of traffic are essentially "invisible" to the highest-priority class of traffic. With proper network dimensioning, the highest class of service can indeed

achieve extremely low packet loss and delay—essentially circuit-like performance. But can the network *guarantee* that an ongoing flow in a high-priority traffic class will continue to receive such service throughout the flow's duration using only the mechanisms that we have described so far? It cannot. In this section, we'll see why yet additional network mechanisms and protocols are required when a hard service guarantee is provided to individual connections.

Let's return to our scenario from Section 7.5.2 and consider two 1 Mbps audio applications transmitting their packets over the 1.5 Mbps link, as shown in Figure 7.27. The combined data rate of the two flows (2 Mbps) exceeds the link

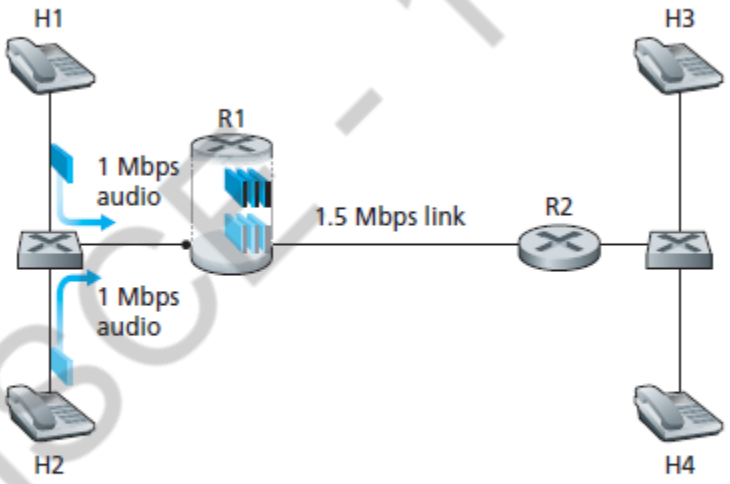


Figure 7.27 ♦ Two competing audio applications overloading the R1-to-R2 link

capacity. Even with classification and marking, isolation of flows, and sharing of unused bandwidth (of which there is none), this is clearly a losing proposition. There is simply not enough bandwidth to accommodate the needs of both applications at the same time. If the two applications equally share the bandwidth, each application would lose 25 percent of its transmitted packets. This is such an unacceptably low QoS that both audio applications are completely unusable; there's no need even to transmit any audio packets in the first place.

Given that the two applications in Figure 7.27 cannot both be satisfied simultaneously, what should the network do? Allowing both to proceed with an unusable QoS wastes network resources on application flows that ultimately provide no utility to the end user. The answer is hopefully clear—one of the application flows should be blocked (that is, denied access to the network), while the other should be allowed to proceed on, using the full 1 Mbps needed by the application. The telephone network is an example of a network that performs such call blocking—if the required resources (an end-to-end circuit in the case of the telephone network) cannot be allocated to the call, the call is blocked (prevented from entering the network) and a busy signal is returned to the user. In our example, there is no gain in allowing a flow into the network if it will not receive a sufficient QoS to be considered usable. Indeed, there is a cost to admitting a flow that does not receive its needed QoS, as network resources are being used to support a flow that provides no utility to the end user.

By explicitly admitting or blocking flows based on their resource requirements, and the source requirements of already-admitted flows, the network can guarantee that admitted flows will be able to receive their requested QoS. Implicit in the need to provide a guaranteed QoS to a flow is the need for the flow to declare its QoS requirements. This process of having a flow declare its QoS requirement, and then having the network either accept the flow (at the required QoS) or block the flow is referred to as the **call admission** process. This then is our fourth insight (in addition to the three earlier insights from Section 7.5.2) into the mechanisms needed to provide QoS.

- *Resource reservation.* The only way to *guarantee* that a call will have the resources (link bandwidth, buffers) needed to meet its desired QoS is to explicitly

allocate those resources to the call—a process known in networking parlance as **resource reservation**. Once resources are reserved, the call has on-demand access to these resources throughout its duration, regardless of the demands of all other calls. If a call reserves and receives a guarantee of x Mbps of link bandwidth, and never transmits at a rate greater than x, the call will see loss- and delay-free performance.

Call admission. If resources are to be reserved, then the network must have a mechanism for calls to request and reserve resources. Since resources are not infinite, a call making a call admission request will be denied admission, that is, be blocked, if the requested resources are not available. Such a call admission is performed by the telephone network—we request resources when we dial a number. If the circuits (TDMA slots) needed to complete the call are available, the circuits are allocated and the call is completed. If the circuits are not available, then the call is blocked, and we receive a busy signal. A blocked call can try again to gain admission to the network, but it is not allowed to send traffic into the network until it has successfully completed the call admission process. Of course, a router that allocates link bandwidth should not allocate more than is available at that link. Typically, a call may reserve only a fraction of the link's bandwidth, and so a router may allocate link bandwidth to more than one call. However, the sum of the allocated bandwidth to all calls should be less than the link capacity if hard quality of service guarantees are to be provided.

- *Call setup signaling.* The call admission process described above requires that a call be able to reserve sufficient resources at each and every network router on its source-to-destination path to ensure that its end-to-end QoS requirement is met. Each router must determine the local resources required by the session, consider the amounts of its resources that are already committed to other ongoing sessions, and determine whether it has sufficient resources to satisfy the per-hop QoS requirement of the session at this router without violating local QoS guarantees made to an already-admitted session. A signaling protocol is needed to coordinate these various activities—the per-hop allocation of local resources, as well as the overall end-to-end decision of whether or not the call has been able to reserve sufficient resources at each and every router on the end-to-end path. This is the job of the **call setup protocol**, as shown in Figure 7.28. The **RSVP protocol** [Zhang 1993, RFC 2210] was proposed for this purpose within an Internet architecture for providing quality-of-service guarantees. In ATM networks, the Q2931b protocol [Black 1995] carries this information among the ATM network's switches and end point.

Despite a tremendous amount of research and development, and even products that provide for per-connection quality of service guarantees, there has been almost no extended deployment of such services. There are many possible reasons. First and foremost, it may well be the case that the simple application-level mechanisms that we studied in Sections 7.2 through 7.4, combined with proper

network dimensioning (Section 7.5.1) provide "good enough" best-effort network service for multimedia applications. In addition, the added complexity and cost of deploying and managing a network that provides per-connection quality of service guarantees may be judged by ISPs to be simply too high given predicted customer revenues for that service.

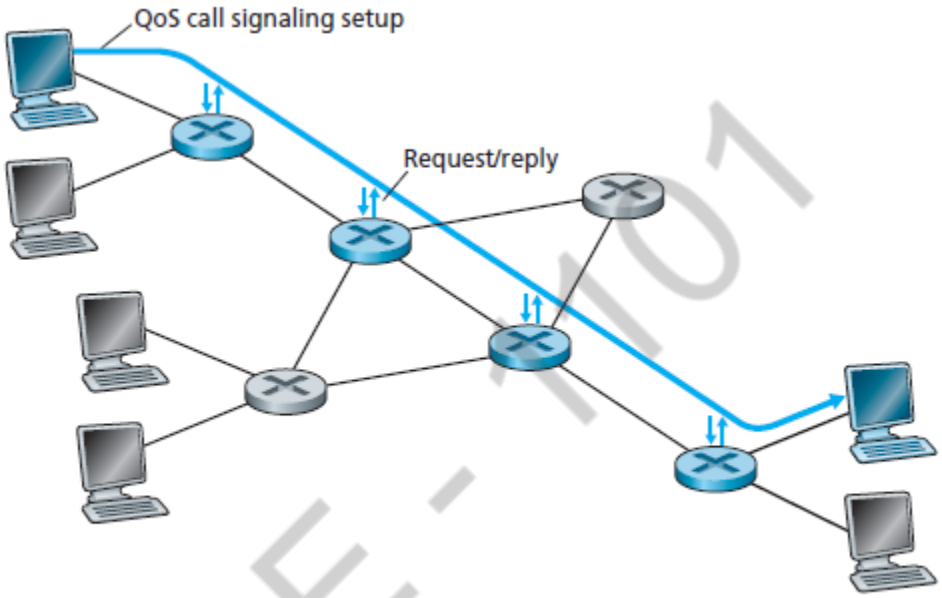


Figure 7.28 ♦ The call setup process

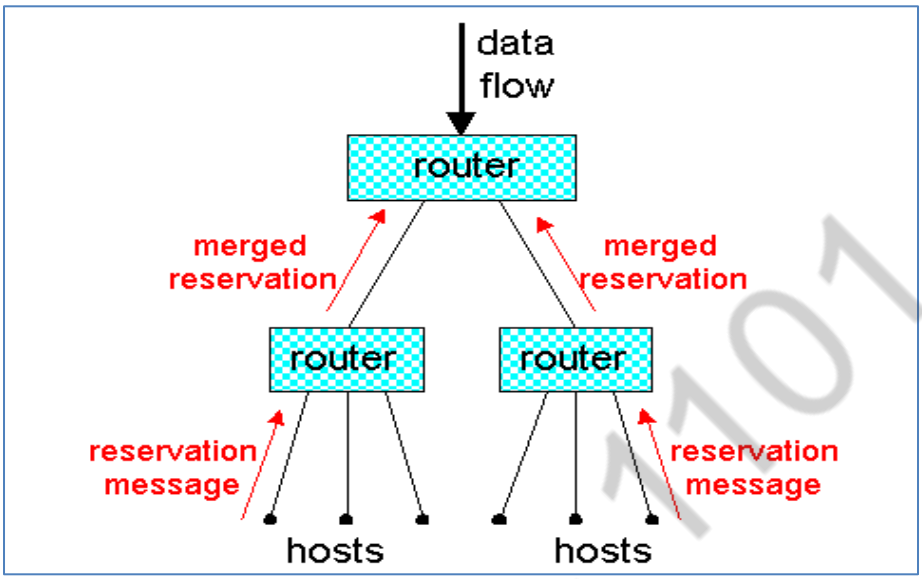
5. Explain in detail about the RSVP Protocol with suitable diagram.

The RSVP protocol allows applications to reserve bandwidth for their data flows. It is used by a host, on the behalf of an application data flow, to request a specific amount of bandwidth from the network. RSVP is also used by the routers to forward bandwidth reservation requests. To implement RSVP, RSVP software must be present in the receivers, senders, and routers. The two principle characteristics of RSVP are:

1. It provides reservations for bandwidth in multicast trees (unicast is handled as a special case).

2. It is receiver-oriented, i.e., the receiver of a data flow initiates and maintains the resource reservation used for that flow.

These two characteristics are illustrated in Figure:.



The above diagram shows a multicast tree with data flowing from the top of the tree to six hosts. Although data originates from the sender, the reservation messages originate from the receivers. When a router forwards a reservation message upstream towards the sender, the router may merge thereservation message with other reservation messages arriving from downstream.Before discussing RSVP in greater detail, we need to recall the notion of a **session**. As with RTP, a session can consist of multiple multicast data flows. Each sender in a session is the source of one or more data flows; for example, a sender might be the source of a video data flow and an audio data flow. Each data flow in a session has the same multicast address. To keep the discussion concrete, we assume that routers and hosts identify the session to which a packet belongs by the packet's multicast address. This assumption is somewhat restrictive; the actual RSVP specification allows for more general methods to identify a session. Within a session, the data flow to which a packet belongs also needs to beidentified. This could be done, for example, with the flow identifier field in IPv6.

WhatRSVPisNot

We emphasize that the RSVP standard does not specify how the network provides the reserved bandwidth to the data flows. It is merely a protocol that allows the applications to reserve the necessary link bandwidth. Once the reservations are in

place, it is up to the routers in the Internet to actually provide the reserved bandwidth to the data flows. This provisioning is done with the scheduling mechanisms. It is also important to understand that RSVP is not a routing protocol -- it does not determine the links in which the reservations are to be made. Instead it depends on an underlying routing protocol (unicast or multicast) to determine the routes for the flows. Once the routes are in place, RSVP can reserve bandwidth in the links along these routes. (We shall see shortly that when a route changes, RSVP re- reserves resources.) And once the reservations are in place, the routers' packet schedulers can actually provide the reserved bandwidth to the data flows. Thus, RSVP is only one piece - albeit an important piece - in the QoS guaranteee puzzle.RSVP is sometimes referred to as a *signaling protocol*. By this it is meant that RSVP is a protocol that allows hosts to establish and tear-down reservations for data flows. The term "signaling protocol" comes from the jargon of the circuit-switched telephony community.

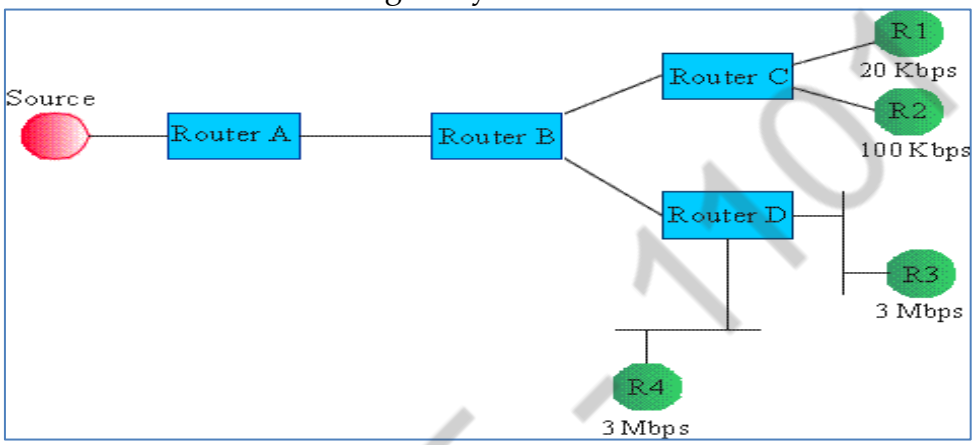
HeterogeneousReceivers

Some receivers can receive a flow at 28.8 Kbps, others at 128 Kbps, and yet others at 10 Mbps or higher. This heterogeneity of the reservers poses an interesting question. If a sender is multicasting a video to a group of heterogeneous receivers, should the sender encode the video for low quality at 28.8Kbps, for medium quality at 128 Kbps, or for high quality at 10 Mbps? If the video is encoded at 10Mbps, then only the users with 10 Mbps access will be able to watch the video. On the other hand, if the video is encoded at 28.8 kbps, then the 10 Mbps users will have to see a low-quality image when they know they can something much better.To resolve this dilemma it is often suggested that video and audio be encoded in layers. For example, a video might be encoded into two layers: a base layer and an enhancement layer. The base layer could have a rate of 20 Kbps whereas the enhancement layer could have a rate of 100 Kbps; in this manner receivers with 28.8 access could receive the low-quality base-layer image, and receivers with 128 Kbps could receive both layers to construct a high-quality image.

We note that the sender does not have to know the receiving rates of all the receivers. It only needs to know the maximum rate of the all its receivers. The sender encodes the video or audio into multiple layers and sends all the layers up to the maximum rate into multicast tree. The receivers pick out the layers that are appropriate for their receiving rates. In order to not excessively waste bandwidth in the network's links, the heterogeneous receivers must communicate to the network the rates they can handle. We shall see that RSVP gives foremost attention to the issue of reserving resources for heterogeneous receivers.

Example:

Let us first describe RSVP in the context of a concrete one-to-many multicast example. Suppose there is a source that is transmitting into the Internet the video of a major sporting event. This session has been assigned a multicast address, and the source stamps all of its outgoing packets with this multicast address. Also suppose that an underlying multicast routing protocol has established a multicast tree from the sender to four receivers as shown below; the numbers next to the receivers are the rates at which the receivers want to receive data. Let us also assume that the video is layered encoded to accommodate this heterogeneity of receiver rates.



Crudely speaking, RSVP operates as follows for this example. Each receiver sends a reservation message upstream into the multicast tree. This reservation message specifies the rate at which the receiver would like to receive the data from the source. When the reservation message reaches a router, the router adjusts its packet scheduler to accommodate the reservation. It then sends a reservation upstream. The amount of bandwidth reserved upstream from the router depends on the bandwidths reserved downstream. In the example in Figure 6.9-2, receivers R1, R2, R3 and R4 reserve 20 kbps, 120 kbps, 3 Mbps and 3 Mbps, respectively. Thus router D's downstream receivers request a maximum of 3Mbps. For this one-to-many transmission, Router D sends a reservation message to Router B requesting that Router B reserve 3 Mbps on the link between the two routers. Note that only 3 Mbps is reserved and not 3+3=6 Mbps; this is because receivers R3 and R4 are watching the same sporting event, so there reservations may be merged. Similarly, Router C requests that Router B reserve 100 Kbps on the link between routers B and C; the layered encoding ensures that receiver R1's 20 Kbps stream is included in the 100 Mbps stream. Once Router B receives the reservation message from its downstream routers and passes the reservations to its schedulers, it sends a new reservation message to its upstream router, Router A. This message reserves 3 Mbps of

bandwidth on the link from Router A to Router B, which is again the maximum of the downstream reservations.

We see from this first example that RSVP is **receiver-oriented**, i.e., the receiver of a data flow initiates and maintains the resource reservation used for that flow. Note that each router receives a reservation message from each of its downstream links in the multicast tree and sends only one reservation message into its upstream link.

AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

EC8002 – MULTIMEDIA COMPRESSION & COMMUNICATION

UNIT V – Multimedia Communication

Part A

1. **Define packet jitter.**

   JitterinIPnetworksisthevariationinthelatencyonapacketflowbetweentwosystems, when some packets take longer to travel from one system totheother. A jitterbuffer (or de-jitter buffer) can mitigate the effects of jitter, either in the network on a router or switch, or on a computer.

2. **What is meant by RSVP.**

   RSVP (Resource Reservation Protocol) is a set of communication rules that allows channels or paths on the Internet to be reserved for the multicast(one source to many receivers) transmission of video and other high-bandwidthmessages. RSVP is part of the Internet Integrated Service (IIS) model, which ensures best-effort service, real-time service, and controlled link-sharing.

3. **Define any 4 quality of service parameters related to multimedia data transmission.**

   Decompressin , Jitter removal

   Error correction , GUI

4. **What are the limitations of best effort service**

   The limitations of best-effort service are packet loss, excessive end-to-end delay and packet jitter.

5. **What is meant by streaming**

   Streaming media is video or audio content sent in compressed form over the internet & played immediately. It avoids the process of saving the data to the hard. By streaming, a user need not wait to download a file to play it.

6. Give the applications of real time streaming protocol

Real Time Streaming Protocol(RTSP) is used by the client application to communicate to the server information such as the requesting of media file, type of clients applications, mechanism of delivery of file & other actions like resume, pause, fast- forward & rewind. It is mostly used in entertainment & communication system to control streaming mediaservers.

7. Write the shortcomings of integrated services

The shortcomings of integrated service(intserv) is that, the per-flow resource reservation may give significant workload to routers & also it does not allow more qualitative definitions of service distinctions.

PART - B

1. **Explain in detail the network architecture and protocol design of SIP.**

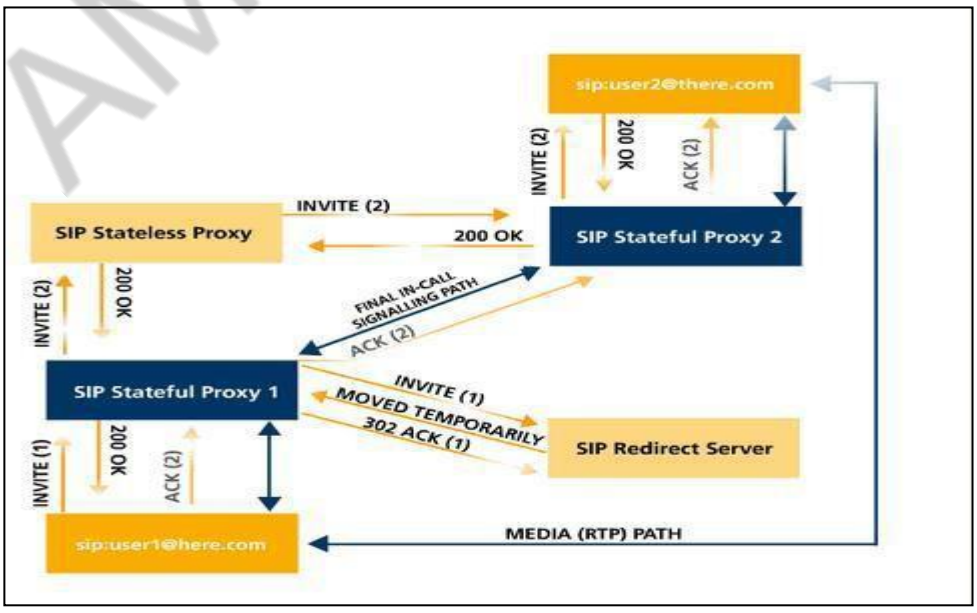
SIP

SIP (Session Initiation Protocol) is a protocol to initiate sessions

- It is an application layer protocol usedto
  - establish
  - modify
  - terminate
- It supports name mapping and redirection servicestransparently

Used by a UA to indicate its current IP address and the URLs for which it would like to receive calls.

- INVITE
- initiatesessions(sessiondescriptionincludedinthemessagebody encoded usingSDP)
- ACK
- confirms sessionestablishment
- BYE
- terminatessessions
- CANCEL
- cancels a pendingINVITE
- REGISTER
- binds a permanent address to a currentlocation
- OPTIONS
- capabilityinquiry
- Other extensions have beenstandardized
  - e.g. INFO, UPDATE, MESSAGE, PRACK, REFER,etc.

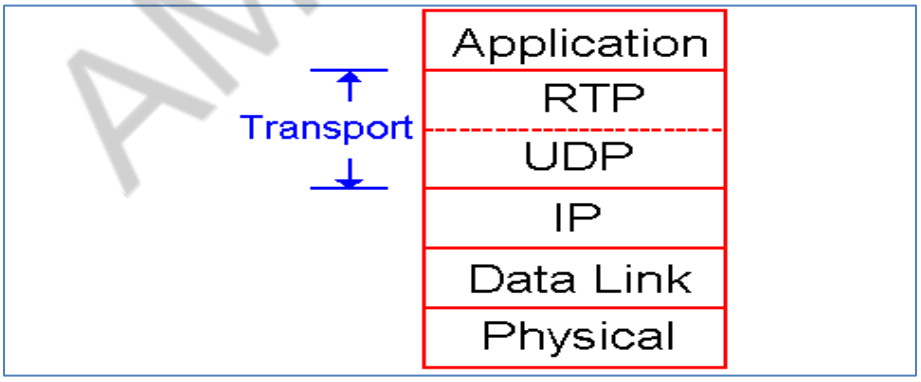


2. Explain in detail about the RTP with necessary diagram.

Welearnedthatthesendersideofamultimediaapplicationappendsheaderfieldstotheaudio/videochunksbeforepassingthechunkstothetransportlayer.Theseheaderfieldsincludesequencenumbersandtimestamps.Sincemostallmultimedianetworkingapplicationscanmakeuseofsequencenumbersandtimestamps,itisconvenienttohaveastandardizedpackettstructurethatincludesfieldsforaudio/videodata,sequencenumberandtimestamp,aswellasotherpotentiallyusefulfields.RTPcanbeusedfortransportingcommonformatssuchasWAVorGSMforsoundandMPEG1andMPEG2forvideo.Itcanalsobeusedfortransportingproprietary soundandvideoformats.ToprovideareadableintroductiontoRTPandtoitscompanionprotocol,RTCP.Wealsodiscusstheroleof RTPinthe H.323standardforreal-timeinteractiveaudioandvideoconferencing.

RTPBasics

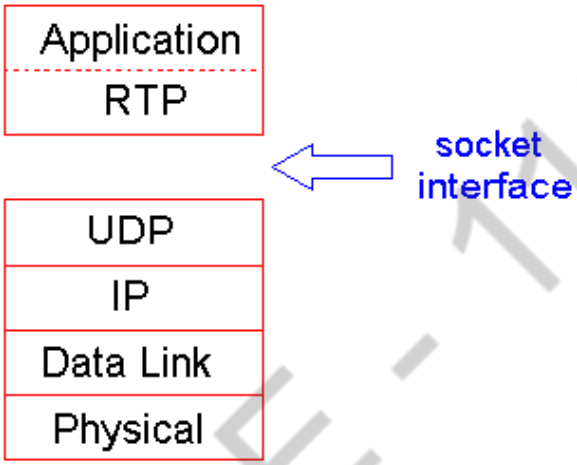
RTPtypicallyrunsontopofUDP.Specifically,audioorvideochunksofdata,generatedbythesendingsideofamultimediaapplication,areencapsulatedinRTPpackets,andeachRTPpacketisinturnencapsulatedinaUDPsegment.BecauseRTPprovidesservices(timestamps,sequence numbers,etc.)tothemultimediaapplication,RTPcanbeviewedasasublayerofthetransportlayer,asshowninFigure.



FigureRTPcanbeviewedasasublayerofthetransportlayer.

Fromtheapplicationdeveloper'sperspective,however,RTPisnotpartofthetransportlayerbutinsteadpartoftheapplicationlayer.Thisisbecause

thedevelopermustintegrateRTPintotheapplication.Specifically,forthesendersideoftheapplication,thedevelopermustwritecodeintothe applicationwhichcreatestheRTPencapsulatingpackets;theapplicationthensendstheRTPpacketsintoaUDPsocketinterface.Similarly,atthe receiversideoftheapplication,theRTPpacketsentertheapplicationthroughaUDPsocketinterface;thedeveloperthereforemustwritecodeinto theapplicationthatextractsthemediachunksfromtheRTPpackets.ThisisillustratedinFigure .



Fromadeveloper'sperspective,RTPispartoftheapplicationlayer.

AsanexampleconsiderusingRTPtotransportvoice.SupposethevoicesourceisPCM encoded(i.e.,sampled,quantized,anddigitized)at64kbps. Furthersupposethattheapplicationcollectstheencodeddatain20msecchunks,i.e,160bytesinachunk.Theapplicationprecedeseachchunkof theaudiodatawithan**RTPheader,**whichincludesthetypeofaudioencoding,asequencenumberandatimestamp.Theaudiochunkalongwith the RTPheaderformthe**RTPpacket**.TheRTPpacketisthensentintotheUDPsocketinterface,whereitisencapsulatedinaUDPpacket.Atthe receiverside,theapplicationreceivestheRTPpacketfromitssocketinterface.Theapplicationextractstheaudiochunkfromthe RTPpacket,and usestheheaderfieldsoftheRTPpackettoproperlydecodeandplaybacktheaudiochunk.

IfanapplicationincorporatesRTP--insteadofaproprietaryschemetoprovidepayloadtype,sequencenumbersortimestamps--thenthe applicationwillmoreeasilyinteroperatewithothernetworkingapplications.Forexample,

iftwodifferentcompaniesdevelopInternetphone softwareandtheybothincorporateRTPintotheirproduct,theremaybesomehopethatauserusingoneoftheInternetphoneproductswillbeable tocommunicatewithauserusingtheotherInternetphoneproduct.Attheendofthissection weshallseethatRTPhasbeenincorporatedintoan importantpartofanInternettelephonystandard.

ItshouldbeemphasizedthatRTPinitselfdoesnotprovideanymechanismtoensureti melydeliveryofdataorprovideotherqualityofservice guarantees;itdoesnotevenguaranteedeliveryofpacketsorpreventout-of-orderdeliveryofpackets.Indeed,RTPencapsulationisonlyseenatthe endsystems--itisnotseenbyintermediaterouters.Routersdonotdistinguishbetween IPdatagramsthatc arryRTPpacketsandIPdatagramsthat don't.RTPallowseachsource(forexample,acameraoramicrophone)tobeassigneditsowni ndependentRTPstreamofpackets.Forexample,fora videoconferencebetweentwoparticipants,fourRTPstreamscouldbeopened:twostreams fortransmittingtheaudio(oneineachdirection)andtwo streamsforthevideo(again,oneineachdirection).However,manypopularencodingtechn iques--includingMPEG1andMPEG2--bundlethe audioandvideointoasinglestreamduringtheencodingprocess.Whentheaudioandvideo arebundledbytheencoder,thenonlyoneRTPstreamisgeneratedineachdirection.

RTPpacketsarenotlimitedtounicastapplications.Theycanalsobesentoverone-to-manyandmany-to-manymulticasttrees.Foramany-to-many multicastsession,allofthesendersandsourcesinthesessiontypicallysendtheirRTPstream sintothesamemulticasttreewiththesamemulticast address.RTPmulticaststreamsbelongingtogether,suchasaudioandvideostreamsemanat ingfrommultiplesendersinavideoconference application,belongtoanRTPsession.

RTPPacketHeaderFields

AsshownintheFigure,thefourprinciplepacketheaderfieldsarethepayloadtype,sequencen umber,timestampandthesourceidentifier.

RTP Header

ThepayloadtypefieldintheRTPpacketisseven-bitslong.Thus2^7or128differentpayloadtypescanbesupportedbyRTP.Foranaudiostream, thepayloadtypefieldisusedtoindicatethetypeofaudioencoding(e.g.,PCM,adaptivedeltamodulation,linearpredictiveencoding)thatisbeing used.Ifasenderdecidestochangetheencodinginthemiddleofasession,thesendercaninformthereceiverofthechangethroughthispayload typefield.ThesendermaywanttochangetheencodinginordertoincreasetheaudioqualityortodecreasetheRTPstreambitrate.Figure listssomeoftheaudiopayloadtypescurrentlysupportedbyRTP

PayloadTypeNumber	AudioFormat	SamplingRate	Through
0	PCMmu-law	8KH	64Kbps
1	101	8KH	4.8Kbps
3	GS	8KH	13Kbps
7	LP	8KH	2.4Kbps
9	G.72	8KH	48-
14	MPEGAudio	90KH	-
15	G.72	8KH	16Kbps

FigureSomeaudiopayloadtypessupportedby RTP.

Foravideostreamthepayloadtypecanbeusedtoindicatethetypeofvideoencoding(e.g., motionJPEG,MPEG1,MPEG2,H.231).Again,the sendercanchangevideoencodingon-the-flyduringasession.Figure6.4-5listssomeofthevideopayloadtypescurrentlysupportedbyRTP.

PayloadTypeNumber	VideoFormat
26	MotionJPEG
31	H.261

32	MPEG1video
33	MPEG2video

FigureSomevideopayloadtypessupportedby RTP.

SequenceNumberField

Thesequencenumberfieldis16-bitslong.Thesequencenumberincrementsbyoneforeach RTPpacketsent,andmaybeusedbythereceiverto detectpacketlossandtorestorepacketsequence.Forexampleifthereceiversideoftheapplicationreceivesastreamof RTPpacketswithagap betweensequencenumbers86and89,thenthereceiverknowsthatpackets87and88werelost.Thereceivercanthenattempttoconcealthelost data.

TimestampField

Thetimestampfieldis32byteslong.Itreflectsthesamplinginstantofthefirstbyteinthe RTPdatapacket.Aswesawintheprevioussection,the receivercanusethetimestampsinordertoremovepacketjitterintroducedinthenetworkandtoprovidesynchronousplayoutatthereceiver.The timestampisderivedfromasamplingclockatthesender.Asanexample,foraudiothetimestampclockincrementsbyoneforeachsamplingperiod(forexample,each125usecsfora8KHzsamplingclock);iftheaudioapplicationgenerateschunksconsistingof160encodedsamples,then thetimestampincreasesby160foreach RTPpacketwhenthesourceisactive.Thetimestamp clockcontinuestoincreaseataconstantrateevenif thesourceisinactive.

SynchronizationSourceIdentifier(SSRC)

TheSSRCfieldis32bitslong.Itidentifiesthesourceof theRTPstream.Typically,eachstreaminaRTPsessionhasadistinctSSRC.TheSSRCis nottheIPaddressofthesender,butinsteadanumberthatthesourceassignsrandomlywhenthenewstreamisstarted.Theprobabilitythattwo streamsgetassignedthesameSSRCisverysmall.

3. Discuss about the RTCP with necessary diagrams.

AprotocolwhichamultimedianetworkingapplicationcanuseinconjunctionwithRTP. TheuseofRTCPis particularlyattractivewhenthenetworkingapplicationmulticastsaudioorvideotomultipl ereceiversfromoneormoresenders.AsshowninFigure,RTCPpacketsaretransmittedbyeac hparticipantinanRTPsessiontoallotherparticipantsinthesession.TheRTCPpacketsaredis tributedtoalltheparticipantsusingIPmulticast.ForanRTPsession,typicallythereisasingle multicastaddress,andallRTPandRTCPpacketsbelonging tothesessionusethemulticastaddress.RTPandRTCPpacketsaredistinguishedfromeachot herthroughtheuseofdistinctportnumbers.

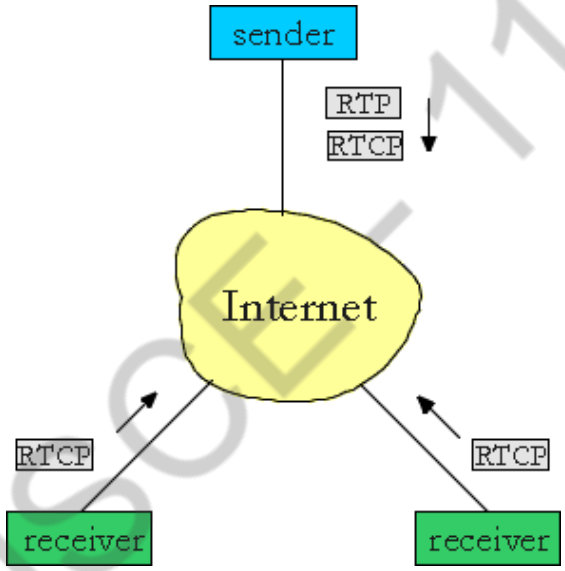


Figure. BothsendersandreceiverssendRTCPmessages.

RTCPpacketsdonotencapsulatechunksofaudioorvideo.Instead,RTCPpacketsare sentperiodicallyandcontainsenderand/orreceiverreports thatannouncestatisticsthatcanbeusefultotheapplication.Thesestatisticsincludenumber ofpacketssent,numberofpacketslostandinterarrival jitter.TheRTPspecification[RFC1889]doesnotdictatewhattheapplicationshoulddowitht hisfeedbackinformation.Itisuptotheapplication developertodecidewhatitwantstodowiththefeedbackinformation.Senderscanusethefe edbackinformation,forexample,tomodifytheir

transmissionrates.Thefeedbackinformationcanalsobeusedfordiagnosticpurposes;fore xample,receiverscandeterminewhetherproblemsare local,regionalorglobal.

RTCPPacket Types

Receiverreceptionpackets

ForeachRTPstreamthatareceiverreceivesaspartofasession,thereceivergeneratesar eceptionreport.Thereceiveraggregatesitsreception reportsintoasingleRTCPpacket.Thepacketisthensentintomulticasttreethatconnectstoge therallttheparticipantsinthesession.Thereception reportincludesseveralfields,themostimportantofwhicharelistedbelow.

- ❍TheSSRCoftheRTPstreamforwhichthereceptionreportisbeinggenerated.
- ❍Thefractionofpacketslostwithin theRTPstream.Eachreceivercalculatesthenu mberofRTPpacketslostdividedbythenumberof RTPpacketssentaspartofthestream.Ifasenderreceivesreceptionreportsindica tingthatthereceiversarereceivingonlyasmall fractionofthesender'stransmittedpackets,thesendercanswitchtoalowerenco dingrate,therebydecreasingthecongestioninthe network,whichmayimprovethereceptionrate.
- ❍Thelastsequencenumberreceivedinthestream ofRTPpackets.
- ❍Theinterarrivaljitter,whichiscalculatedastheaverageinterarrivaltimebetweens uccessivepacketsintheRTPstream.

Senderreportpackets

ForeachRTPstreamthatasenderistransmitting,thesendercreatesandtransmitsRTCPsen der-reportpackets.Thesepacketsincludeinformation abouttheRTPstream,including:

- •TheSSRCoftheRTPstream.
- •Thetimestampandwall-clocktimeofthemostrecentlygeneratedRTPpacketinthestream

- Thenumberofpacketssentinthestream.
- Thenumberofbytessentinthestream.

Thesenderreportscanbeusedtosynchronizedifferentmediastreamswithinan RTPsession.Forexample,consideravideoconferencingapplication forwhicheachsendergeneratestwoindependentRTPstreams,oneforvideoandoneforaudio.Thetimestampsintheses RTPpacketsaretiedtothe videoandaudiosamplingclocks,andarenottiedtothe*wall-clocktime*(i.e.,torealtime).EachRTCPsender-reportcontains,forthemostrecently generatedpacketintheassociatedRTPstream,thetimestampoftheRTPpacketandthewall-clocktimeforwhenthepacketwascreated.Thusthe RTCPsender-reportpacketsassociatethesamplingclocktothereal-timeclock.Receiverscanusethisassociationin theRTCPsenderreportsto synchronizetheplayoutofaudioandvideo.

Sourcedescriptionpackets

ForeachRTPstreamthatasenderistransmitting,thesenderalsocreatesandtransmits source-descriptionpackets.Thesepacketscontain informationaboutthesource,suchase-mailaddressofthesender,thesender'snameandtheapplicationthatgeneratestheRTPstream.Italso includestheSSRCoftheassociatedRTPstream.Thesepacketsprovideamappingbetweenthesourceidentifier(i.e.,theSSRC)andtheuser/host name.RTCPpacketsarestackable,i.e.,receiverreceptionreports,senderreports,andsourcedescriptorscanbeconcatenatedintoasinglepacket.The resultingpacketisthenencapsulatedintoaUDPsegmentandforwardedintothemulticasttree.

RTCP Bandwidth Scaling

Theastutereaderwillhaveobservedthat RTCPhasapotentialscalingproblem.Considerforexamplean RTPsessionthatconsistsofonesender andalargenumberofreceivers.Ifeachofthereceiversperiodicallygenerate RTCPpackets,thentheaggregatetransmissionrateof RTCPpackets cangreatlyexceedtherateof RTPpacketssentbythesender.Observethattheamountoftrafficsentintothemulticasttreedoesnotchangeasthe numberofreceiversincreases,whereastheamountof RTCPtrafficgrowslinearlywiththen

umberofreceivers.Tosolvethisscalingproblem, RTCPmodifiestherateatwhichaparticipantsendsRTCPpacketsintothemulticasttreeasafunctionofthenumberofparticipantsinthesession. Observethat,becauseeachparticipantsendscontrolpacketstoeveryoneelse,eachparticipantcankeeptrackofthetotalnumberofparticipantsin thesession.

RTCPattemptstolimititstrafficto5%ofthesessionbandwidth.Forexample,supposethereisonesender,whichissendingvideoatarateof2

Mbps.ThenRTCPattemptstolimititstrafficto5%of2Mbps,or100Kbps,asfollows.Theprotocolgives75%ofthisrate,or75Kbps,tothe receivers;itgivestheremaining25%oftherate,or25Kbps,tothesender.The75Kbpsdevotedtothereceiversisequallysharedamongthe receivers.Thus,iftherearеRreceivers,theneachreceivergetstosendRTCPtrafficatarateof75/RKbpsandthesendergetstosendRTCPtraffic at arateof25Kbps.Aparticipant(asenderorreceiver)determinestheRTCPpackettransmissionperiodbydynamicallycalculatingthetheaverage RTCPpacketsize(acrosstheentiresession)anddividingtheaverageRTCPpacketsizebyitsallocatedrate.Insummary,theperiodfortransmitting RTCPpacketsforasenderis

$$T = \frac{\text{number of senders}}{.25*.05*\text{session bandwidth}} (\text{avg. RTCP packet size})$$

AndtheperiodfortransmittingRTCPpacketsforareceiveris

$$T = \frac{\text{number of receivers}}{.75*.05*\text{session bandwidth}} (\text{avg. RTCP packet size})$$

4. Explain in detail about the H.323.

H.323isastandardforreal-timeaudioandvideoconferencingamongendsystemsontheInternet.AsshowninFigure6.4-7,italsocovershowend systemsattachedtotheInternetcommunicatewithtelephonesattachedtoordinarycircuit-switchedtelephonenetworks.Inprinciple,if manufacturers ofInternettelephonyandvideoconferencingallconformtoH.323,thenalltheirproductsshouldbeabletointeroperate,andshould be

abletocommunicatewithordinarytelephones.WediscussH.323inthissection,asitprovide sanapplicationcontextforRTP.Indeed,weshall seebelowthatRTPisanintegralpartoftheH.323standard.

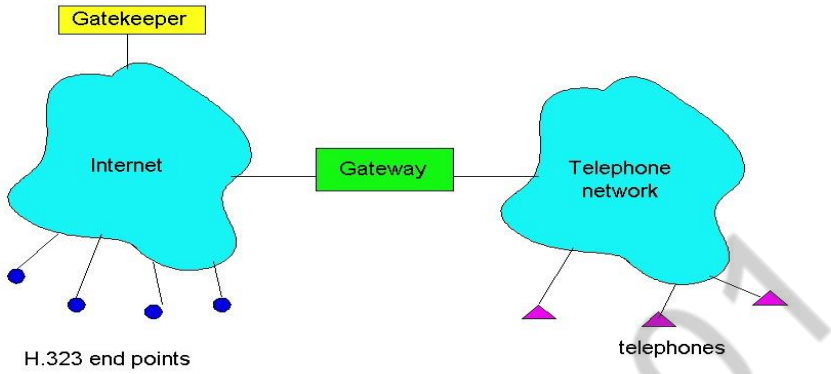


Figure. H.323endsystemsattachedtotheInternetcancommunicatewithtelephonesattachedtoa circuit-switchedtelephonenetwork.

H.323**endpoints**(a.k.a.terminals)canbestand-alonedevices(e.g.,WebphonesandWebTVs)orapplicationsinaPC(e.g.,Internetphoneor video conferencingsoftware).H.323equipmentalsoincludes**gateways**and**gatekeepers**.Gatew ayspermitcommunicationamongH.323endpoints andordinarytelephonesinacircuit-switchedtelephonenetwork.Gatekeepers,whichareoptional,provideaddresstranslation ,authorization, bandwidthmanagement,accountingandbilling.Wewilldiscussgatekeepersinmoredetail attheendofthissection.

TheH.323isanumbrellaspecificationthatincludes:

- Aspecificationforhowendpointsnegotiatecommonaudio/videoencodings.Becaus eH.323supportsavarietyofaudioandvideoencoding standards,aprotocolisneededtoallowthecommunicatingendpointstoagreeonaco mmonencoding.
- Aspecificationforhowaudioandvideochunksareencapsulatedandsentovernetwor k.Asyoumayhaveguessed,thisiswhereRTPcomes intothepicture.

- AspecificationforhowendpointscommunicatewiththeirrespectivegatekeepersAspecificationforhowendpointscommunicatewiththeirrespectivegatekeepers.

- AspecificationforhowInternetphonescommunicatethroughagatewaywithordinaryphonesinthepubliccircuit-switchedtelephone network.

Minimally,eachH.323endpoint*must*supporttheG.711speechcompressionstandard.G.711usesPCMtogeneratedigitizedspeechateither56 kbpsor64kbps.AlthoughH.323requireseveryendpointtobevoicecapable(throughG.711),videocapabilitiesareoptional.Becausevideo supportisoptional,manufacturersofterminalscansellsimplerspeechterminalsaswellasmorecomplexterminalsthatsupportbothaudioand video.

AsshowninFigure6.4-8,H.323alsorequiresthatallH.323endpointsusethefollowingprotocols:

- RTP-thesendingsideofanendpointencapsulatesallmediachunkswithinRTPpackets.SendingsidethenpassestheRTPpacketstoUDP.

- H.245-an"out-of-band"controlprotocolforcontrollingmediabetweenH.323endpoints.Thisprotocolisusedtonegotiateacommon audioorvideocompressionstandardthatwillbeemployedbyalltheparticipatingendpointsinasession.

- Q.931-asignalingprotocolforestablishingandterminatingcalls.Thisprotocolprovidestraditionaltelephonefunctionality(e.g.,dialtones andringing)toH.323endpointsandequipment.

- RAS(Registration/Admission/Status) channelprotocol-aprotocolthatallowsendpointstocommunicatewithagatekeeper(ifgatekeeperispresent).

Figure6.4-8showstheH.323protocolarchitecture

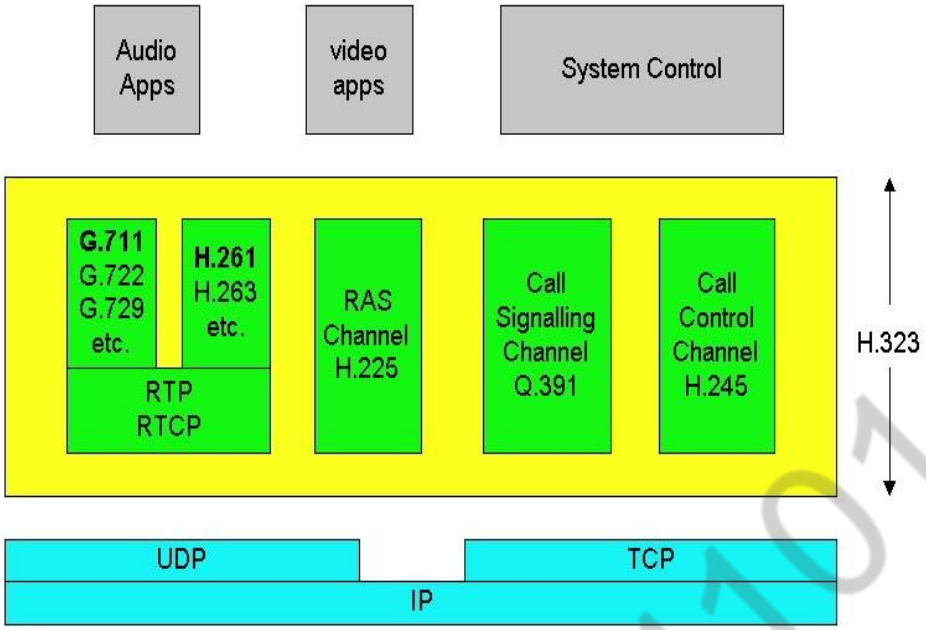


Figure6.4-8H.323protocolarchitecture.

Audio and Video Compression

TheH.323standardsupportsaspecificsetofaudioandvideocompressiontechniques.Let'sfirstconsideraudio.Aswejustmentioned,allH.323 endpointsmustsupporttheG.711speechencodingstandard.BecauseofthisrequirementtwoH.323endpointswillalwaysbeabletodefaultto G.711andcommunicate.ButH.323allowsterminalstosupportavarietyofotherspeechcompressionstandards,includingG.723.1,G.722,G.728 andG.729.Manyofthesestandardscompressspeechtoratesthatwillpassthrough28.8Kbpsdial-upmodems.Forexample,G.723.1compresses speechtoeither5.3kbpsor6.3kbps,withsoundqualitythatiscomparabletoG.711.

AswementionedearliervideocapabilitiesforanH.323endpointareoptional.However,ifanendpointdoessupportsvideo,thenitmust(atthe veryleast)supporttheQCIFH.261(176x144pixels)videostandard.Avideocapableendpointmyoptionallysupportotherh.261schemes, includingCIF,4CIFand16CIF.,andtheH.263standard.AstheH.323standardevolves,it willlikelysupportalongerlistofaudioandvideo compressionschemes.

H.323 Channels

WhenaendpointparticipatesinanH.323session,itmaintainsseveralchannels,asshowninFig ure

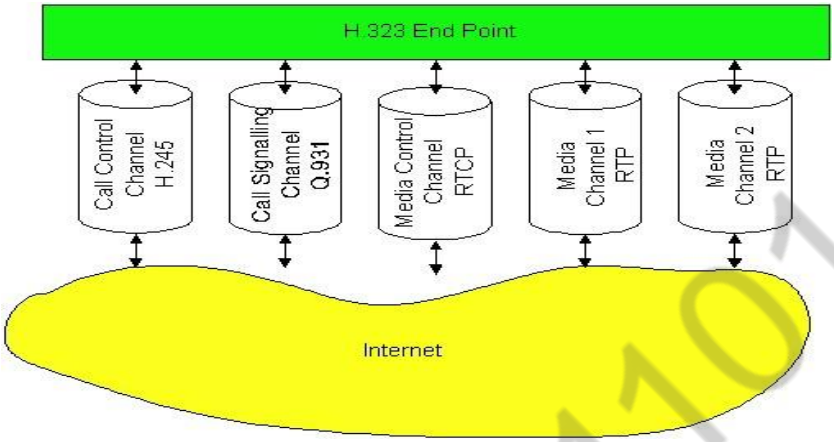
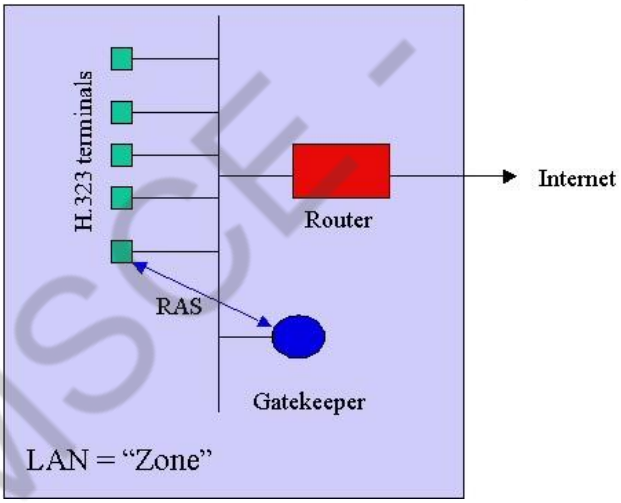


Figure6.4-9H.323channels

ExaminingFigure6.4-9,weseethatanendpointcansupportmanysimultaneousRTPmediachannels.Foreachmediatype,therewilltypicallybeonesendmediachannelandonereceivemediachannel;thus,ifaudioandvideoaresentinseparateRTPstreams,therewilltypicallybefourmediachannels.AccompanyingtheRTPmediachannels,thereisoneRTCPmediacontrolchannel,asdiscussedinSection6.4.3.AlloftheRTPand RTCP channelsrunoverUDP.InadditiontotheRTP/RTCPchannels,twootherchannelsarerequired,thecallcontrolchannelandthecall signaling channel.TheH.245callcontrolchannelisaTCPconnectionthatcarriesH.245controlmessages.Itsprincipletasksare(i)openingandclosingmediachannels;and(ii)capabilityexchange,i.e.,beforesendingmedia,endpointsagreeonandencodingalgorithm.H.245,beingacontrol protocolforreal-timeinteractiveapplications,isanalogoustoRTSP,whichisacontrolprotocolforstreamingofstoredmultimedia.Finally,the Q.931callsignalingchannelprovidesclassicaltelephonefunctionality,suchasdialtoneandringing.

Gatekeepers

ThegatekeeperisanoptionalH.323device.EachgatekeeperisresponsibleforanH.323zone.AtypicaldeploymentscenarioisshowninFigure6.4. Inthisdeploymentscenario,theH.323terminalsandthegatekeeperareallattachedtothesameLAN,andtheH.323zoneistheLANitself.Ifa zonehasagatekeeper,thenallH.323terminalsinthezonearerequiredtocommunicatewithitusingtheRASprotocol,whichrunsoverTCP. Addresstranslationisoneofthemoreimportantgatekeeperservices.Eachterminalcanhaveanaliasaddress,suchasthenameofthepersonatthe terminal,theemailaddressofthepersonattheterminal,etc.Thegatewaytranslatesthesealiasaddressestoaddresses.Thisaddresstranslation serviceissimilartotheDNSservice,coveredinSection2.5.Anothergatekeeperserviceisbandwidthmanagement:thegatekeepercanlimitthe numberofsimultaneousrealtimeconferencesinordertosavesomebandwidthforotherapplicationsrunningovertheLAN.Optionally,H.323calls canberoutedthroughgatekeeper,whichisusefulforbilling



H.323terminalmustregisteritselfwiththegatekeeperinitszone.WhentheH.323application isinvokedattheterminal,theterminalusesRASto senditsIPaddressandalias(providedbyuser)tothegatekeeper.Ifgatekeeperispresentinazone,eachterminalinthezonemustcontact gatekeepertoaskpermissiontomakeacall.Onceithaspermission,theterminalcansendthegatekeeperane-mailaddress,aliasstringorphone extensionfortheterminalitwantstocall,whichmaybeinanotherzone.Ifnecessary,agatekeeperwillpollothergatekeepersinotherzonesto resolveanIPaddress.