الجامعة الافتراضيّة السوريّة

SYRIAN VIRTUAL UNIVERSITY

Multimedia Networking

Doctor Raouf Hamdan

ISSN: 2617-989X

Books & Refrences

تشبيك الوسائط المتعددة

الدكتور رؤوف حمدان

من منشورات الجامعة الافتراضية السورية

الجمهورية العربية السورية 2020

Multimedia Networking

Dr. Raouf Hamdan

Publications of the Syrian Virtual University (SVU)

Syrian Arab Republic, 2020

Index

Chapter 1
Trends in communications and networking

Keywords:

Multimedia Technologies, Multimedia Applications, Internet TV and Over-The-Top services. Internet of Things.

Abstract:

This chapter will present the different forces have consistently driven the architecture and evolution of data communications and networking facilities: such as Traffic growth, the emergence of new services, and the Advances in technology. Also, we present Data Transmission & Network Capacity Requirements.

Learning Outcomes:

At the end of this lesson the student will be aware of the following topics:

- The different forces that have consistently driven the architecture and evolution of data communications and networking.
- Evolution of communications and networking.
- The convergence and the merger of distinct telephony and information technologies and markets
- Internet TV and Over-The-Top services

References for this chapter:

1. Data and Computer Communications, Tenth Edition, by William Stallings, Pearson 2014
2. Fundamentals of Multimedia, Ze-Nian Li and Mark S. Drew, 2nd ed.
3. Articles in Multimedia and Internet of Things

1. Introduction:

- With the increasing availability of broadband access to the Internet has come an increased interest in Internet-based multimedia applications.
- definition:
  - Multimedia: Human–computer interaction involving text, graphics, voice, and video

With the increasing availability of broadband access to the Internet has come an increased interest in Internet-based multimedia applications.

The terms multimedia and multimedia applications are used rather loosely in the literature and in commercial publications, and no single definition of the term Multimedia. Let us consider this simple definition:

Multimedia: Human–computer interaction involving text, graphics, voice, and video

2. Multimedia Taxonomy :

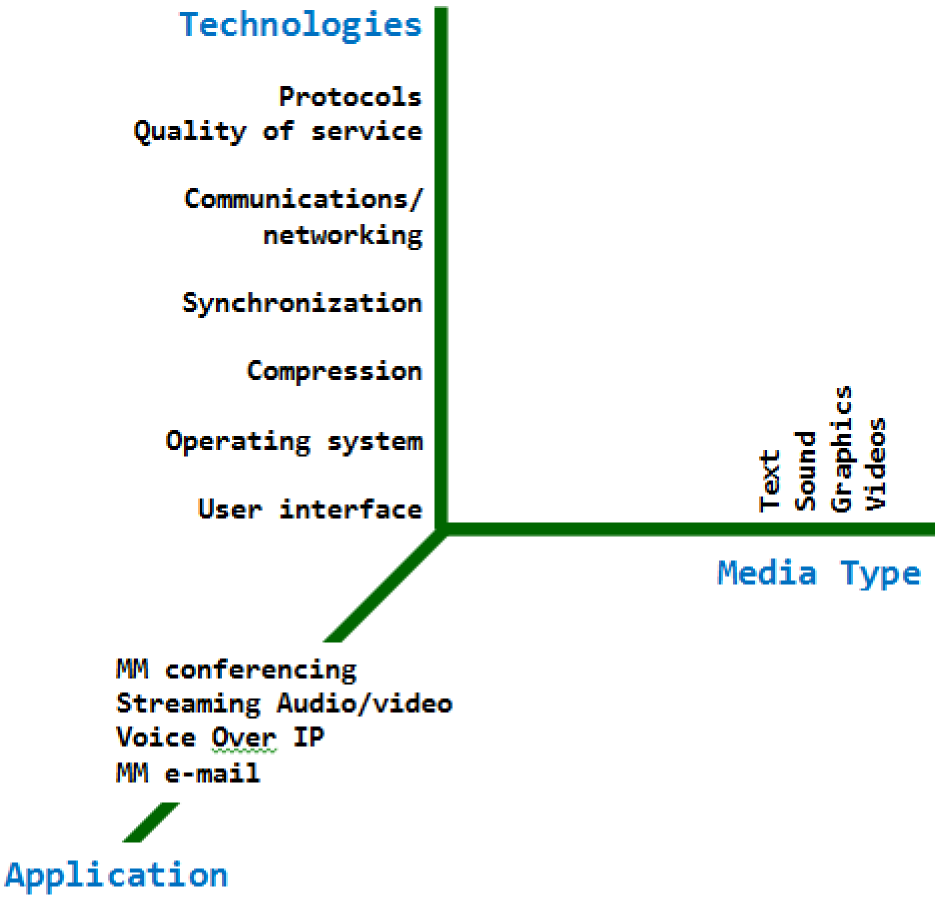


Figure 1.1 Multimedia Taxonomy, after [Stallings[1]]

One way to organize the concepts associated with multimedia is to look at a taxonomy that captures a number of dimensions of this field. Figure 1.1 looks at multimedia from the perspective of three different dimensions: type of media, applications, and the technology required to support the applications.

3. Multimedia Applications:

- The Internet has been dominated by information retrieval applications, e-mail, and file transfer, plus Web interfaces that emphasized text and images.
- Increasingly, the Internet is being used for multimedia applications that involve massive amounts of data for visualization and support of real-time interactivity.
- Streaming audio and video are perhaps the best known of such applications. Examples:
  - virtual training environment involving distributed simulations and real-time user interaction,
  - Computer games, digital video, audio, Videoconferencing, virtual communities
  - Online training, electronic books, streaming media.

The Internet, until recently, has been dominated by information retrieval applications, e-mail, and file transfer, plus Web interfaces that emphasized text and images.
Increasingly, the Internet is being used for multimedia applications that involve massive amounts of data for visualization and support of real-time interactivity.
Streaming audio and video are perhaps the best known of such applications. An example of an interactive application is a virtual training environment involving distributed simulations and real-time user interaction, Computer games, digital video, audio, Videoconferencing, virtual communities, Online training, electronic books, streaming media.

4. Multimedia Technologies:

Figure 1.1 lists some of the technologies that are relevant to the support of multimedia applications.

This course will explore some of these technologies:

- Compression: Digitized video and audio can generate an enormous amount of traffic on a network. A streaming application, which is delivered to many users, magnifies the traffic. Accordingly, standards have been developed for producing significant savings through compression. The most notable standards are JPG for still images and MPG for video.
- Communications/networking: This broad category refers to the transmission and networking technologies that can support high-volume multimedia traffic.
- Protocols: A number of protocols are instrumental in supporting multimedia traffic. One example is the Real-time Transport Protocol (RTP), which is designed to support inelastic traffic

Figure 1.1 lists some of the technologies that are relevant to the support of multimedia applications.

This course will explore some of these technologies, especially those related to communications and networking, Compression, and Protocols.

- **Compression**:

  Digitized video and audio can generate an enormous amount of traffic on a network. A streaming application, which is delivered to many users, magnifies the traffic. Accordingly, standards have been developed for producing significant savings through compression. The most notable standards are JPG for still images and MPG for video.

- **Communications/networking**:

  This broad category refers to the transmission and networking technologies that can support high-volume multimedia traffic.

- **Protocols**:

  A number of protocols are instrumental in supporting multimedia traffic. One example is the Real-time Transport Protocol (RTP), which is designed to support inelastic traffic

5. Driving forces:

Three different forces have consistently driven the architecture and evolution of data communications and networking facilities:

- Traffic growth,
- Development of new services,
- Advances in technology.

5.1. Traffic growth:

- Communication traffic, both local and long distance, has been growing at a high and steady rate for decades. Network traffic is no longer limited to voice and data and increasingly includes image and video.
- Increasing business emphasis on web services, remote access, online transactions, and social networking means that this trend is likely to continue.
- Business managers are constantly pressured to increase communication capacity in cost effective ways.

5.2. Development of new services:

1. Businesses rely more and more on information technology: the range of services that business users desire to consume is expanding.
2. Mobile users are increasingly demanding high-quality services to support their high-resolution camera phones, favorite video streams.
3. In turn, the growth in high-speed network offerings at competitive price encourages the expansion of mobile applications and services.
4. **Thus, growths in services and in traffic capacity go hand in hand**.

For example, mobile broadband traffic growth is exploding as is the amount of data being pushed over mobile networks by business users' smart phones and tablets. Similar demand growth is seen in landline access to the Internet and private networks.

To keep up with traffic generated by both consumers and business users, mobile service providers have to keep investing in high-capacity networking.

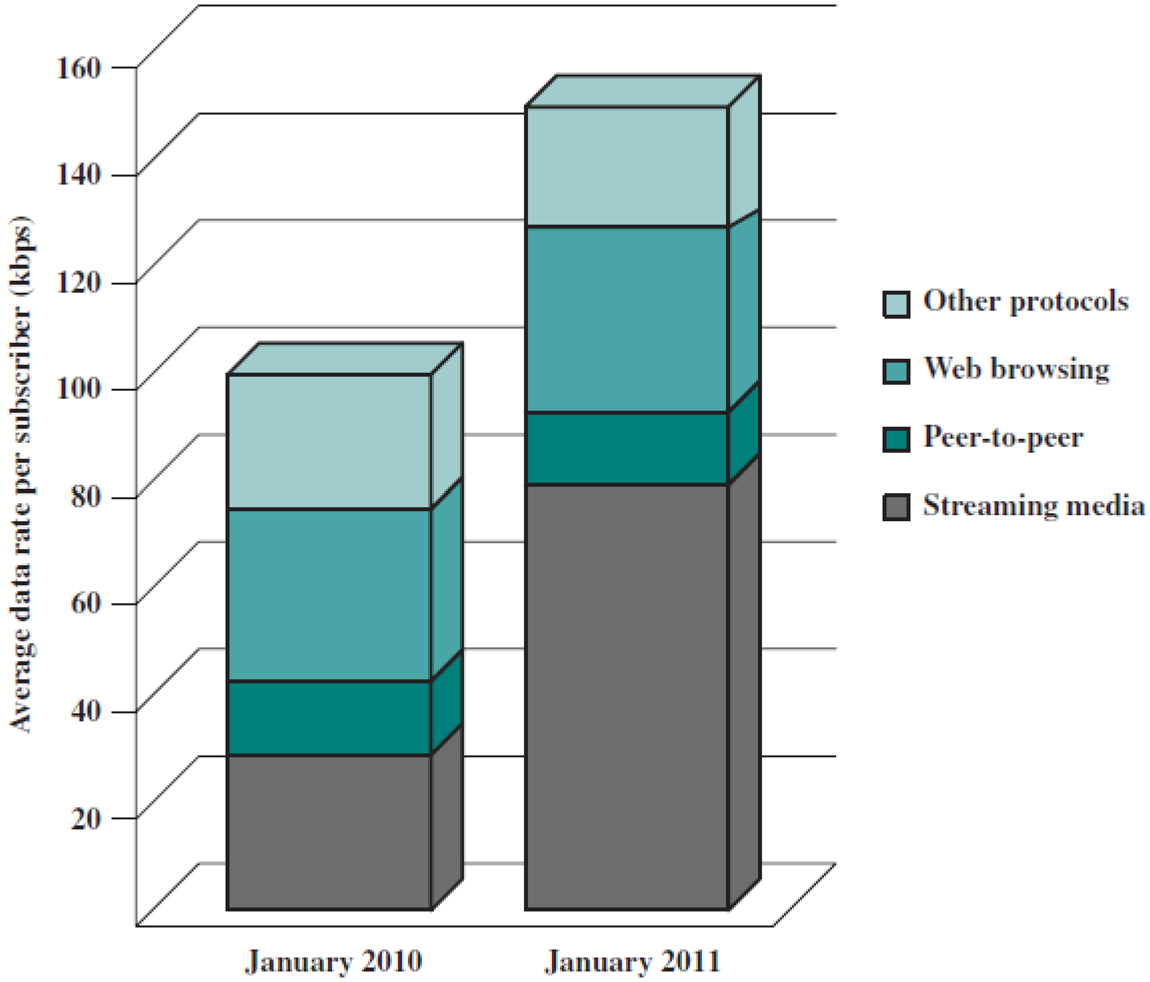


Figure 1: Average Downstream Traffic per Internet Subscriber [[2]]

Figure 1 shows the increasing demand for high data rates, where **streaming media** constitute the service of the large consumer.

5.3. Advances in technology:

Trends in technology enable the provision of increasing traffic capacity and the support of a wide range of services. Four technology trends are particularly notable:

1. The trend toward **faster and cheaper**, in both computing and communications.
2. Networks are more "**intelligent**" than ever.
3. The Internet, the Web, and associated applications have emerged as dominant features for both business and personal network. The migration to "**everything over IP**".

4. Mobility is newest frontier for ICT managers, and popular consumer devices such as the iPhone, and iPad have become drivers of the evolution of business networks.

Networks are more “intelligent” than ever. Today’s networks can offer differing levels of quality of service (QoS), and provide a variety of customizable services in the areas of network management and security.
The migration to “everything over IP” continues and has created many opportunities and challenges for information and communications technology (ICT) managers.

6. Technology:

The trend toward faster and cheaper, in both computing and communications, continues.

- In terms of **computing**, this means more powerful computers and clusters of computers capable of supporting more demanding applications, such as multimedia applications.
- In terms of **communications**, the increasing use of optical fiber and high-speed wireless has brought transmission prices down and greatly increased capacity.

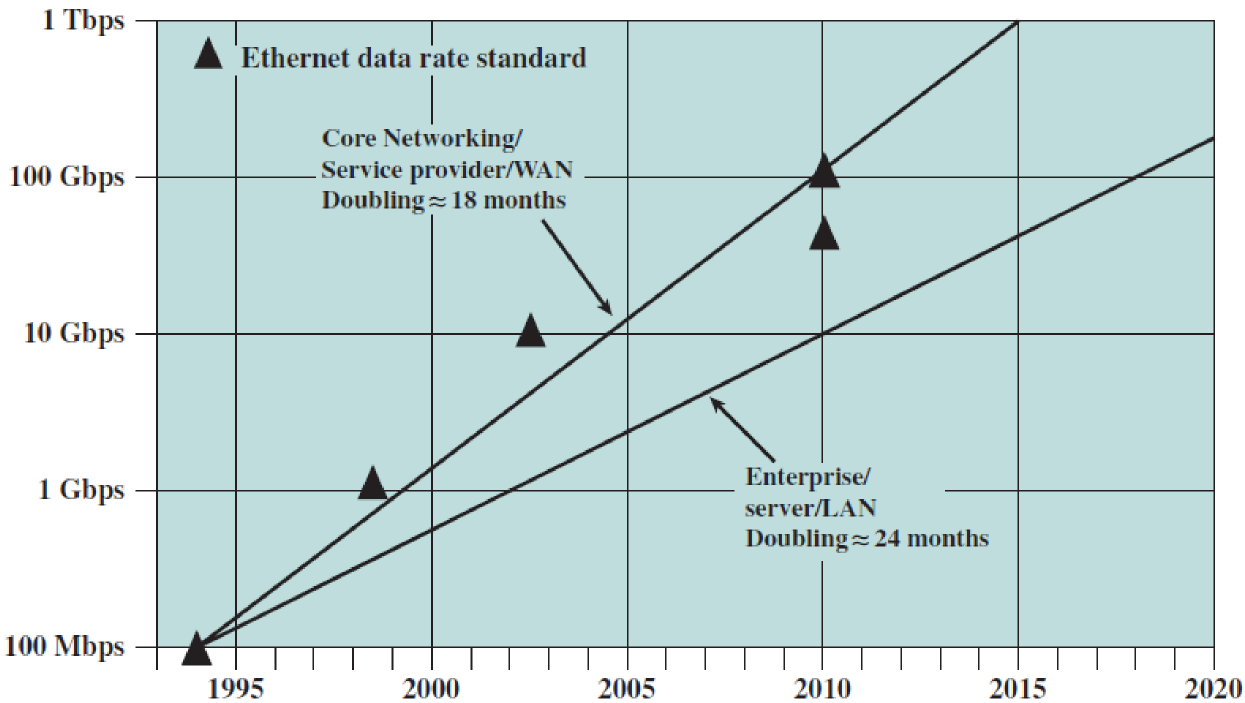


Figure 2: Local area networks (LANs), many enterprises now have 40-Gbps Ethernet or 100-Gbps Ethernet backbone networks.

7. Data Transmission & Network Capacity Requirements:

Changes in the way organizations do business and process information have been driven by changes in networking technology and at the same time have driven those changes. The use of the Internet by both businesses and individuals reflects this cyclic dependency:

- The availability of new image-based services on the Internet has resulted in an increase in the total number of users and the traffic volume generated by each user. This, in turn, has resulted in a need to increase the speed and efficiency of the Internet.
- On the other hand, it is only such increased speed that makes the use of Web-based applications palatable to the end user.

8. Convergence:

Convergence refers to the merger of distinct telephony and information technologies and markets. We can think of this convergence in terms of a three layer model of enterprise communications: **Applications**, **Enterprise services**, **Infrastructure**:

- **Applications**: Convergence integrates communications applications, such as voice calling (telephone), voice mail, e-mail, and instant messaging, with business applications, such as workgroup collaboration, customer relationship management.
- **Enterprise services**: At this level, the manager deals with the information network in terms of the services it provides to support applications. The network manager needs design, maintenance, and support services related to the deployment of convergence-based facilities.
- **Infrastructure**: consists of the communication links, LANs, WANs, and Internet connections available to the enterprise. Increasingly, enterprise network infrastructure also includes private and/or public cloud connections to Data centers which host high-volume data storage and Web services.

A key aspect of convergence is the ability to carry **voice, image**, and **video** over networks that were originally designed to carry data traffic. Infrastructure convergence has also occurred for networks that were designed for voice traffic. For

example, video, image, text, and data are routinely delivered to smart phone users over cell phone networks.
One example is multimedia messaging, which enables a user to employ a single interface to access messages from a variety of sources (e.g., office voice mail, office e-mail, beeper, and fax).

9. Example: Triple Play of Services

"Triple play" of services: voice, video and data. The triple-play vision is that consumers can subscribe to one service that provides voice, data (Internet and other online services), and video (live broadcast and on-demand) – all three brought into the home or office over one line or feed, and by one service provider.

Convergence in simple terms: Convergence involves moving voice into a data infrastructure, integrating all the voice and data networks inside a user organization into a single data network infrastructure, and then extending that into the wireless arena. The foundation of this convergence is packet-based transmission using the **Internet Protocol (IP)**.

10. Internet TV and Over–The–Top OTT services:

- Emerging models for Over–The–Top (OTT) services: Voice over IP, instant messaging, and streaming video and music.
- OTT (Over The Top) is a generic term commonly used to refer to the delivery of audio, video, and other media over the Internet without the involvement of the operator in the control or distribution of the content.

Over–the–top (OTT) service is an online service which could substitute for traditional telecommunications and audiovisual services such as voice telephony, SMS and television.

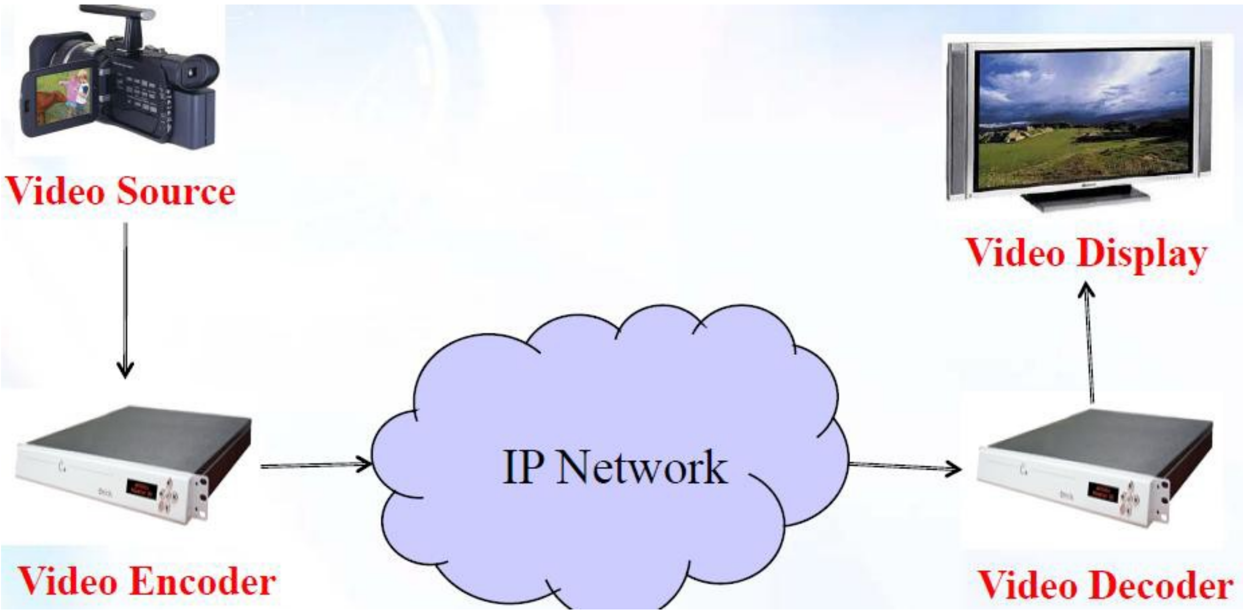
Voice over IP (VoIP) can be viewed as having been the first major OTT service. (Skype)

A range of online services substitute for financial services.

11. What is IPTV?

IPTV: Internet Protocol Television

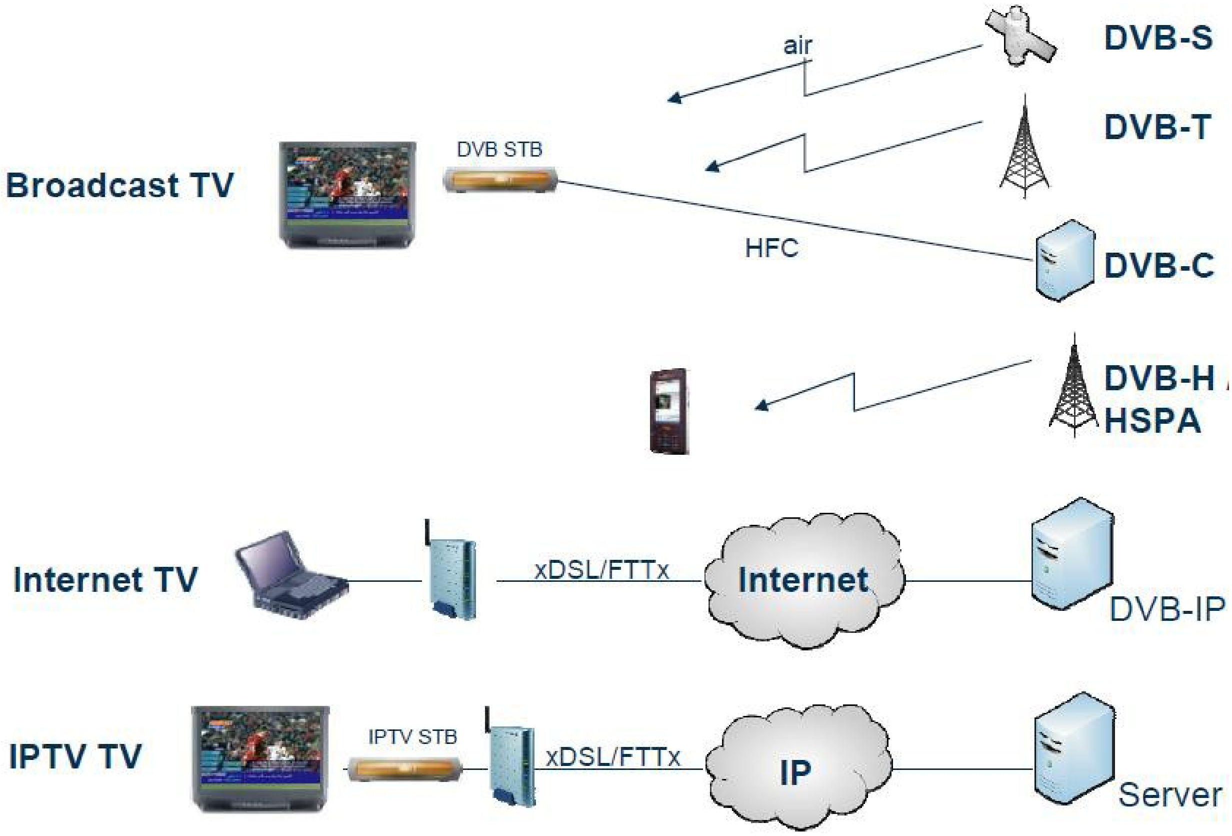
Television through Internet Protocol



The definition formulated by the ITU-T (International Telecommunication Union) focus group on IPTV:

- IPTV is multimedia services of delivering television/audio/text/graphics/data over IP based networks managed to provide the required level of Quality of Service (QoS)/Quality of Experience (QoE), security, interactivity and reliability."

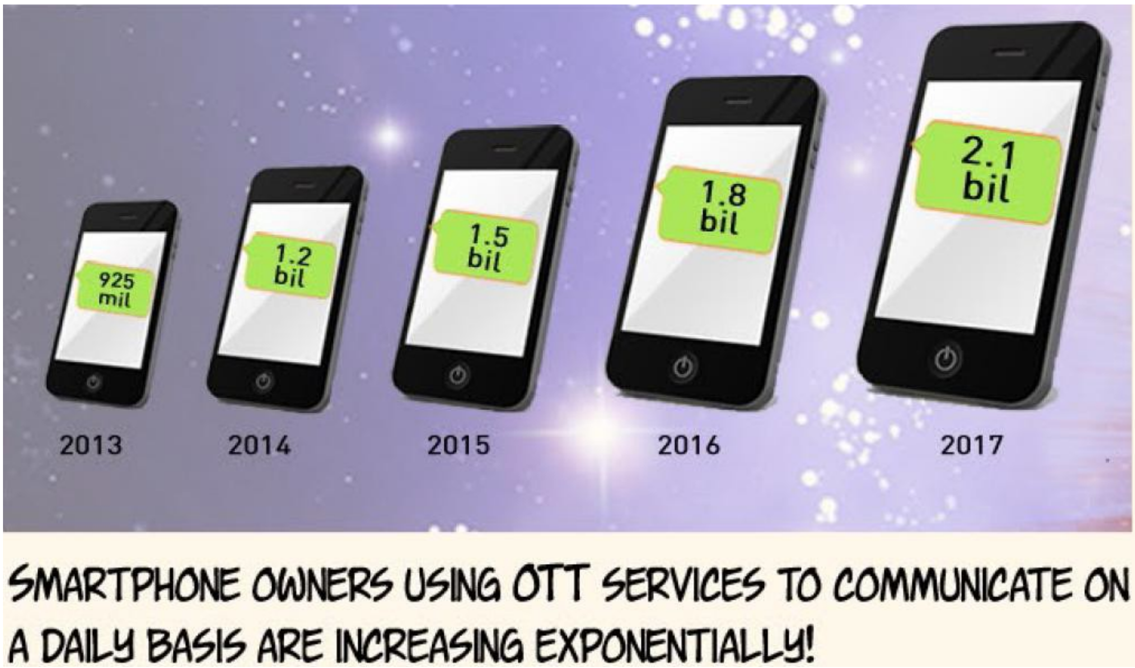
12. Broadcast, Internet TV, IPTV:



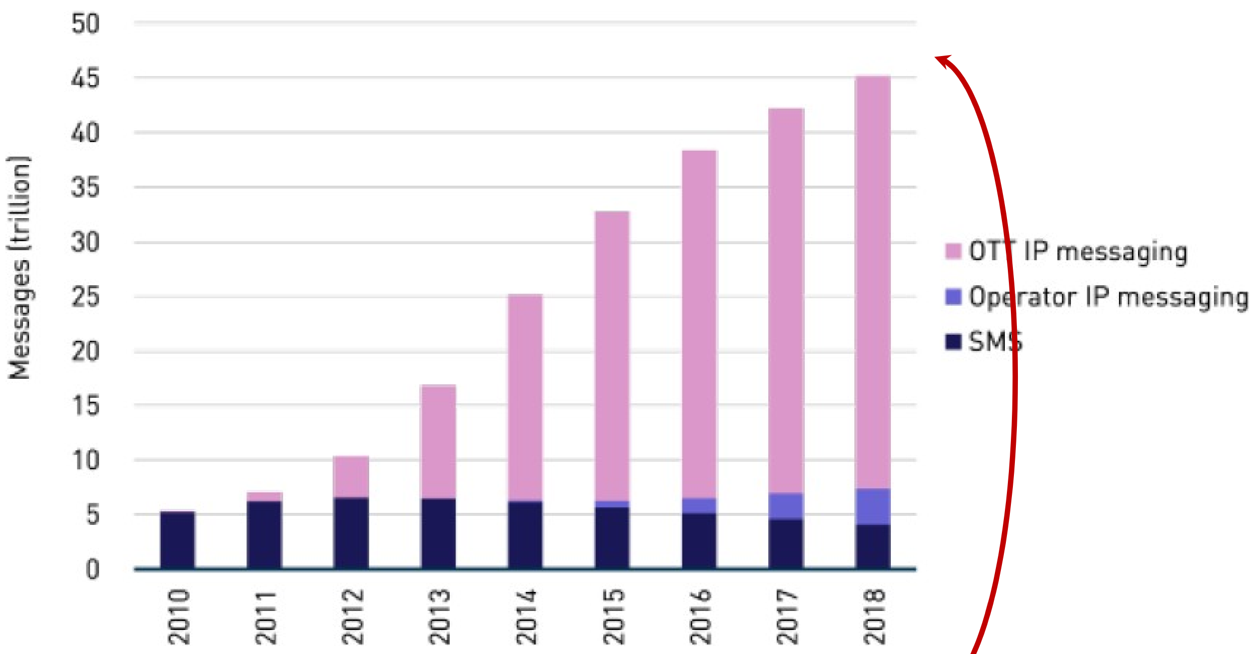
13. Managed vs. Online services:

- OTT services represent a subset of online services, which differ from managed services.
- Managed services are those where the provider offering the service has substantial control over the fixed or mobile access network used for its distribution. The provider may be able to use this control to size its network, or to reserve network capacity to guarantee the Quality of Service.
- Online services depend on the public Internet for their delivery; consequently, no single network operator can guarantee the quality of the service delivered.

14. OTT services:

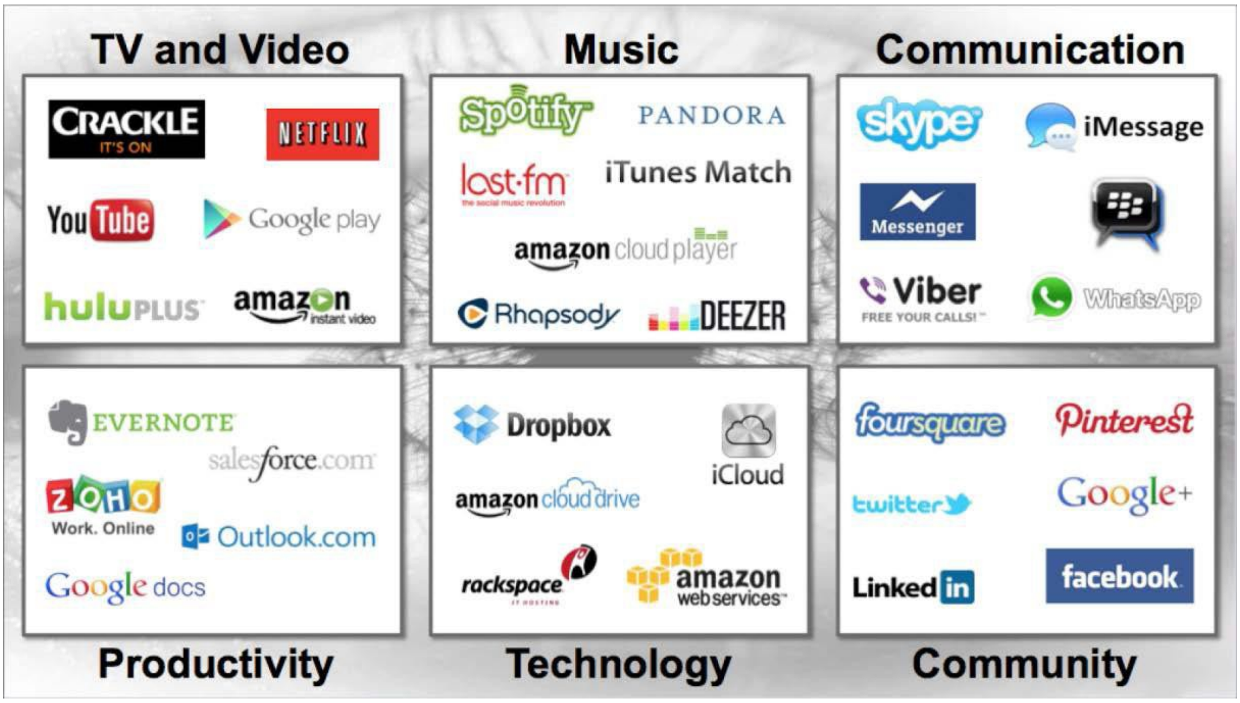


OTT IP services:



Worldwide volume of messages sent using OTT IP services such as WhatsApp exceeded the volume sent using the traditional operator SMS service.

15. Examples of online services:



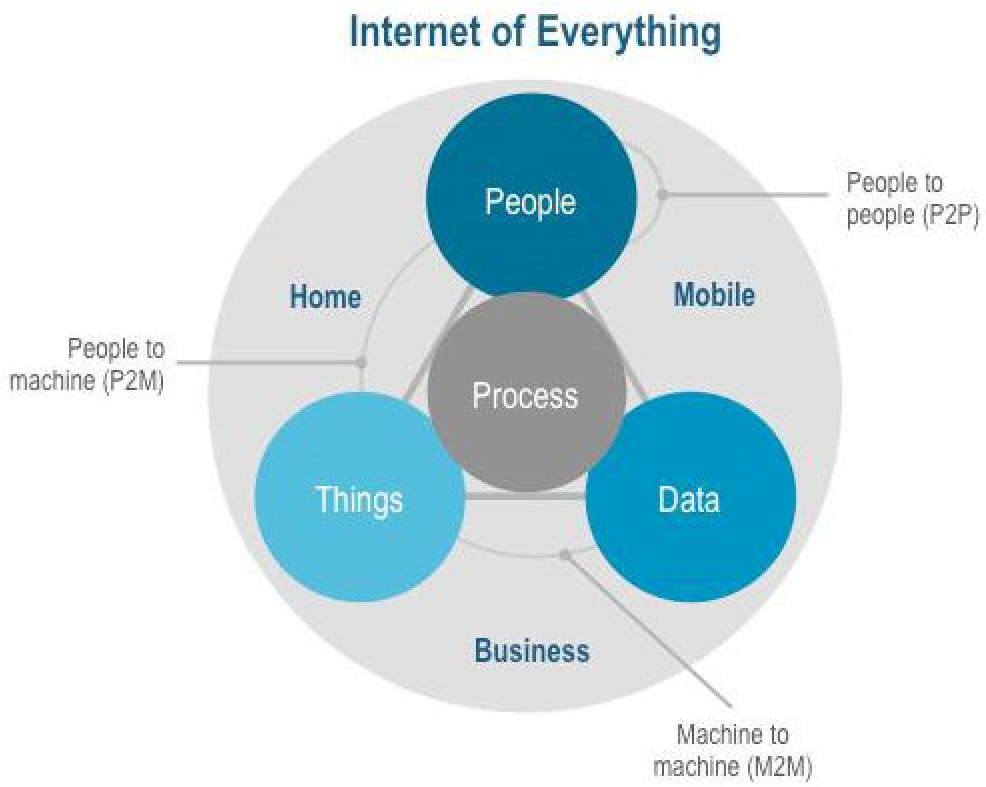
16. Online services and substitution effects:

- The growing popularity of online telephony and messaging services **Skype** and **WhatsApp** appears to be impacting the usage and thus the revenues of traditional voice telephony and SMS providers.
- Traditional television and related audiovisual service providers appear to face similar competition from online service providers (**Netflix** represents a prominent new OTT player in the television market.)

17. Future outlook: 5G and the Internet of Things (IoT):

- Many organizations are currently experiencing the Internet of Things (IoT), the networked connection of physical objects.
- As things add capabilities like context awareness, increased processing power, and energy independence, and as more people and new types of information are connected, IoT becomes an Internet of Everything.

18. Components of IoT: people, process, data, and things:



People themselves will become nodes on the Internet.

As the Internet evolves toward IoE, we will be connected in more relevant and valuable ways. For example, in the future, people will be able to swallow a pill that senses and reports the health of their digestive tract to a doctor over a secure Internet connection.

19. Example: City 24/7 Smart Screen Locations

Goals of the City24/7 Smart Screens

- Inform by instantly connecting people with information that is relevant to their immediate proximity
- Protect by giving local police and fire departments a citywide sensing, communications, and response network that can direct needed personnel and resources exactly where and when they are needed
- Revitalize by increasing levels of commerce, investment, and tourism

20. Outlook on OTT looking forward towards 2020:

- Internet traffic is forecast to continue on its dramatic growth path, reaching 168 Exabytes per month by 2020.
- According to Cisco projections, by 2019 eighty percent of IP traffic will be via mobile and wireless connections.
- Cisco estimates Internet video to represent 64% of total internet traffic by 2019

Demo: Short video 3:16 about IoT (from Intel)

Please refer to related video in the resources folder
(IntelIoT–WhatDoesInternet–of–ThingsMean)

21. Course Overview:

We can safely predict that future video distribution and voice conversations will take place over the Internet, often to wireless devices connected to the Internet. In this context, we present this course to examine video and audio streaming in some detail. We'll explore many of the underlying principles behind multimedia acquisition, authoring, storage, and streaming. We present these techniques from the perspective of well known case studies such as: (Skype, YouTube, Netflix, and Hulu, IPTV, etc…).

21.1. First Part:

The first part of the course presents the basic multimedia components such as Image, Video, Audio, Text, Graphics, etc …, and the compression techniques for the requirements of storage and transmission: Still Image compression, Video compression, and Audio compression standards.

21.2. Second Part:

The second part will concentrate on multimedia streaming over Internet protocol. First we study the Voice over Internet Protocol (VoIP) (Skype case study), then the Video and Audio over Internet protocol (YouTube, Netflix, and Internet Protocol Television IPTV case studies).

Also we present the techniques to control the Quality of Service in multimedia transmission.

And we conclude by the topic of Content-Based Image and Video retrieval as an active axe of research in the emerging large image data-bases in the Internet.

Chapter 2
Image Representations

Keywords:

binary images, gray–level images, color images, image file formats.

Abstract:

In this chapter we look at images, starting with 1–bit images, then 8–bit gray images, then 24–bit color images and 8–bit versions of color images. The we introduce the specifics of file formats for storing such images.

Learning Outcomes:

At the end of this lesson the student will:

- be able to distinguish between the different types of digital images
- Understand the internal structure of binary images, gray–level images, then true color images and 8–bit versions of color images.
- Be aware of some popular image file formats.
- Practice the basic of image manipulation as: Reading Images, Displaying Images, and Modifying Images

- ## References for this chapter:

Fundamentals of Multimedia, Ze–Nian Li and Mark S. Drew, 2^{nd} ed.

1. Binary Images:

- Each pixel is stored as a single bit (0 or 1), so also referred to as binary image.
- So–called 1–bit monochrome image

Fig. 3.1 Monochrome 1–bit image

Images consist of pixels—picture elements in digital images. A 1–bit image consists of on and off bits only and thus is the simplest type of image. Each pixel is stored as a single bit (0 or 1). Hence, such an image is referred to as a binary image. It is also sometimes called a 1–bit monochrome image since it contains no color. Figure 3.1 shows a 1–bit monochrome image (called "Lena" by multimedia scientists—this is a standard image used to illustrate many algorithms).

2. Gray-Level Images:

- Each pixel has a gray-value between 0 and 255. (0 → black, 255 → white)
- Image resolution refers to *the number of pixels* in a digital image
- A 640 x 480 grayscale image requires ??? kB (One byte per pixel)

Fig. 3.2 Grayscale image

Each pixel has a gray-value between 0 and 255. So each pixel is represented by a single byte; e.g., a darkest pixel might have a value of 0, and a bright one might be between 200 and 255.

The entire image can be thought of as a two-dimensional array of pixel values. We refer to such an array as a bitmap—a representation of the graphics/image data that parallels the manner in which it is stored in video memory.

Image resolution refers to the number of pixels in a digital image (higher resolution always yields better quality). Fairly high resolution for such an image might be 1,600 x 1,200, whereas lower resolution might be 640 x 480.

Each pixel is usually stored as a byte, so a 640×480 grayscale image requires 300kB of storage (640×480 = 307,200). Figure 3.2 shows the Lena image again, this time in grayscale.

3. True Color images (24-bit color images):

- Each pixel is represented by three bytes, usually representing RGB.
- This format supports 256x256x256 (16,777,216) possible colors.
- A 640x480 24-bit color image would require 921.6 kB

Fig. 3.2 True color image (24-bit)

- In a color 24-bit image, each pixel is represented by three bytes, usually representing RGB.
- This format supports 256 x 256 x 256 possible combined colors, or a total of 16,777,216 possible colors.
- However such flexibility does result in a storage penalty: A 640 x 480 24-bit color image would require 921.6 kB of storage without any compression.

An important point: many 24–bit color images are actually stored as 32–bit images, with the extra byte of data for each pixel used to store an alpha value representing special effect information (e.g., transparency).

Next Figure shows the image forestfire.bmp, a 24–bit image. Also shown are the grayscale images for just the Red, Green, and Blue channels, for this image.

Exmaple of R, G, B color channel images:

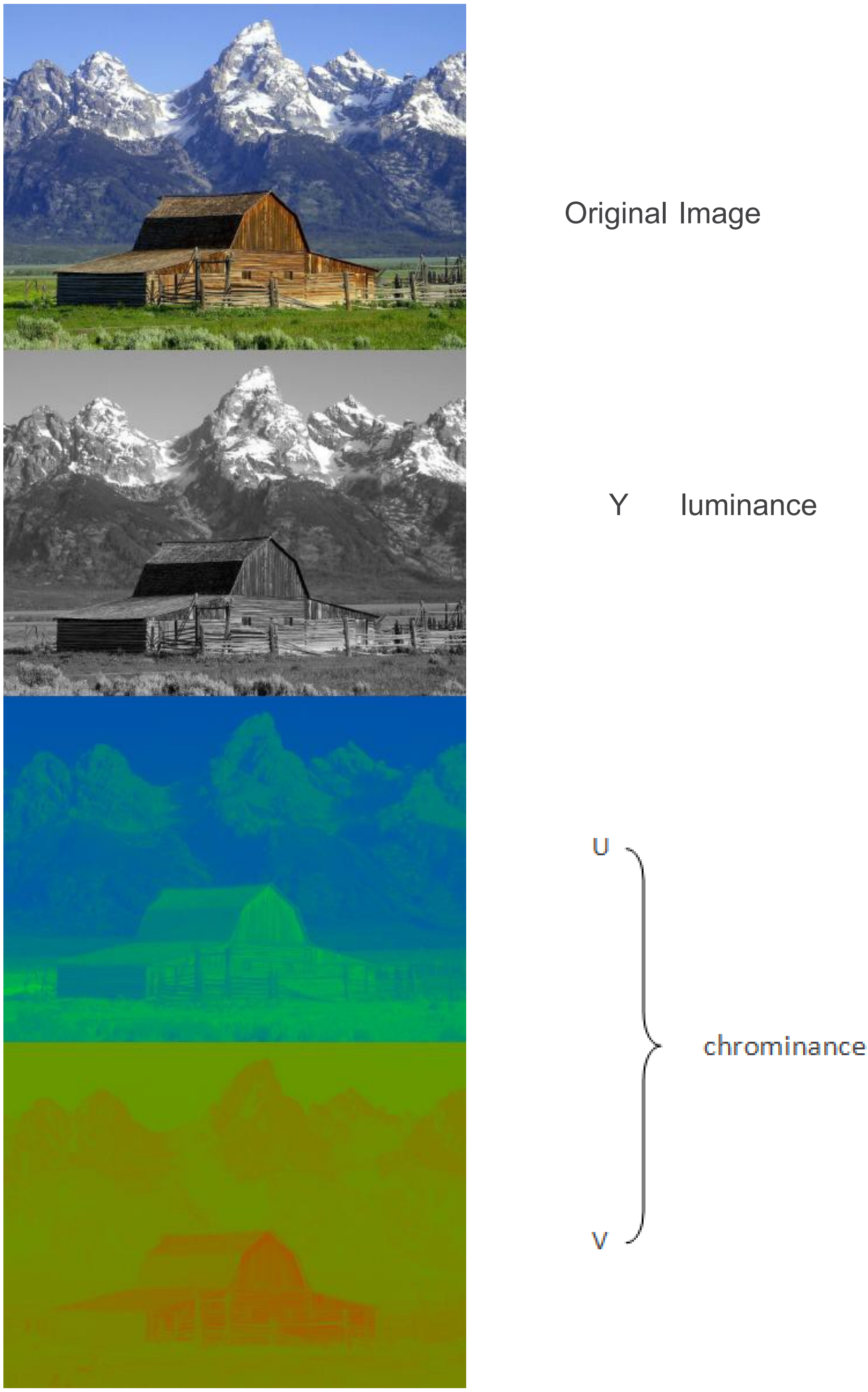
(a)

(b)

(c)

(d)

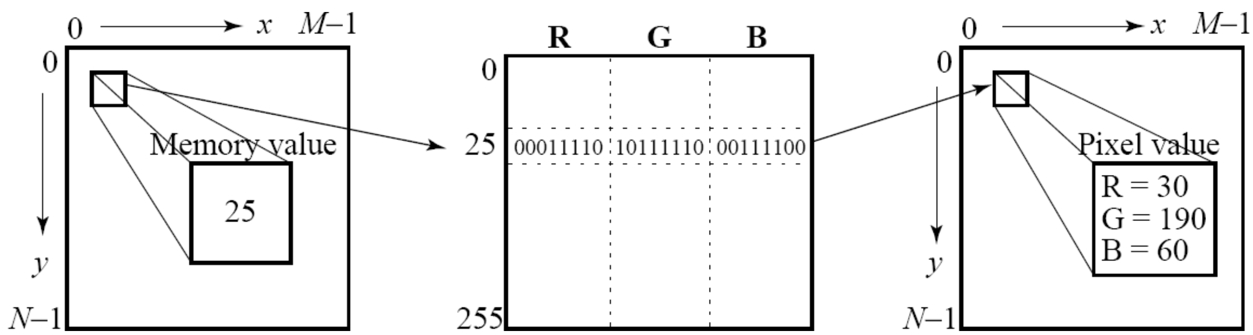
Fig. 3.3: High–resolution color and separate R, G, B color channel images. (a): Example of 24–bit color image "forestfire.bmp". (b, c, d): R, G, and B color channels for this image.



There are many color spaces for color representation. As an example the figure shows and image in the Y U V color space, where the Y component represents the luminance.

4. bit Color Images:

- Many systems can make use of 8 bits of color information (the so–called "256 colors") in producing a screen image. Such image files use the concept of a lookup table to store color information.

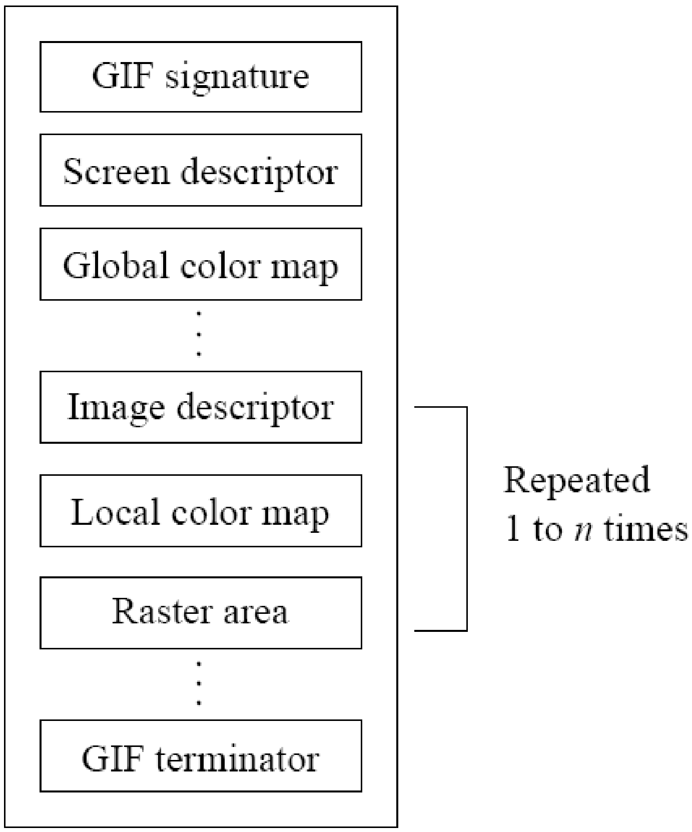


Color LUT for 8–bit color images.

- Basically, the image stores not color, but instead just a set of bytes, each of which is actually an index into a table with 3–byte values that specify the color for a pixel with that lookup table index.
- The idea used in 8–bit color images is to store only the index, or code value, for each pixel. Then, e.g., if a pixel stores the value 25, the meaning is to go to row 25 in a color look–up table (LUT).

5. Popular File Formats:

1. **8-bit GIF**: one of the most important formats because of its historical connection to the WWW and HTML markup language as the first image type recognized by net browsers.



For the standard specification, the general file format of a GIF87

- GIF standard: Limited to 8-bit (256) color images only, which, while producing acceptable color images, is best suited for images with few distinctive colors (e.g., graphics or drawing).
    - GIF standard supports interlacing — successive display of pixels in widely- spaced rows by a 4-pass display process.
- GIF actually comes in two flavors:
    - 1. GIF87a: The original specification.

- 2. GIF89a: The later version. Supports simple animation via a Graphics Control Extension block in the data, provides simple control over delay time, a transparency index, etc.

2. **JPEG**: currently the most important common file format and the current standard for image compression.
   - The human vision system has some specific limitations and JPEG takes advantage of these to achieve high rates of compression.
   - JPEG allows the user to set a desired level of quality, or compression ratio (input divided by output).
3. **PNG format**: standing for Portable Network Graphics — meant to supersede the GIF standard, and extends it in important ways. Special features of PNG files include:
   - Support for up to 48 bits of color information — a large increase.
   - Files may contain gamma-correction information for correct display of color images, as well as alpha-channel information for such uses as control of transparency.
   - The display progressively displays pixels in a 2-dimensional fashion by showing a few pixels at a time over seven passes through each 8 x 8 block of an image.

6. Working with Images in MATLAB:

- Reading Images
- Displaying Images
- Modifying Images

From the Image Processing Toolbox:

- imfinfo– Returns info about graphics file.
- imread – Read image from graphics file.
- imwrite– Write image to graphics file.
- imshow – Display image.
- image – Create and display image object.
- imagesc – Scale data and display as image.

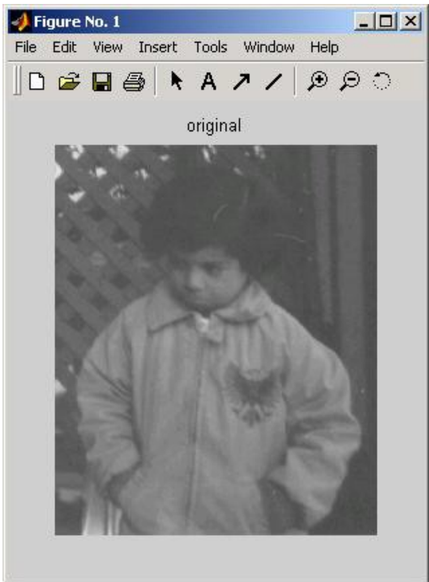
Converting Image Formats:

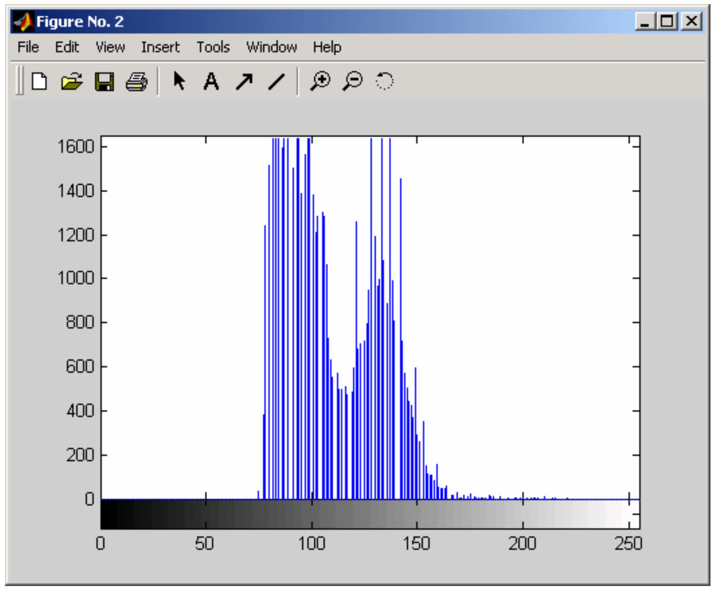
- ind2gray – indexed image to intensity image.
- ind2rgb – indexed image to RGB image (MATLAB).
- gray2ind – intensity image to indexed image.
- rgb2gray – RGB image or colormap to grayscale.
- rgb2ind – RGB image to indexed image.
- mat2gray – matrix to intensity image.
- im2bw – image to binary image by thresholding.

Example:

The histogram of an image shows the distribution of gray levels.

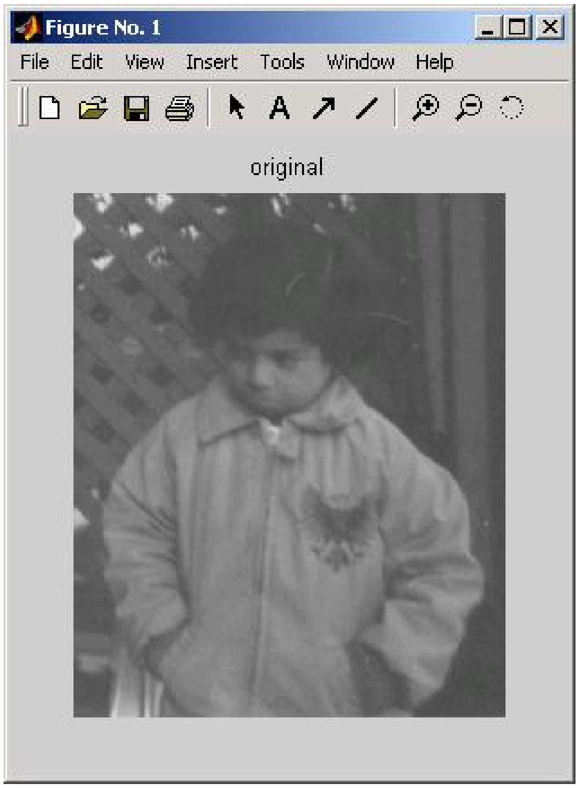
```
>> I = imread('pout.tif');
>> imshow(I)
>> figure, imhist(I)
```





The histeq function can be used to equally distribute the histogram and enhance the contrast.

```
J = histeq(I);
```



Exercise 1: Loading and Viewing an Image

Load in the lena.bmp file into MATLAB.

1. What type of image is it?
2. Display the loaded image.
3. Convert it to a binary (black and white) image.

Chapter 3
Image Compression

Keywords:

Run Length Encoding, Lossless and Lossy compression, JPEG compression.

Abstract:

In this chapter we look at image compression techniques such as JPEG …

Learning Outcomes:

At the end of this lesson the student will:

- Understand the concepts of image compression.
- Understand the JPEG compression standard.
- Be aware of some image compression formats.

Outline:

1. The general idea behind compression.
2. Lossless and Lossy compression.
3. Transform Based Compression

1. Why to compress :

- Although compression is not directly related to the subject of multimedia, multimedia transmission is not possible without first compressing the data. (huge files)
- Compression plays a crucial role in multimedia communication due to the large volume of data exchanged.
- In compression, we reduce the volume of data to be exchanged.
- Simply we compress to satisfy the requirements of storage and transmission of the large image files.

2. General Concepts:

- How Can an Image be Compressed, AT ALL!?
  - If images are random matrices, better not to try any compression
  - Image pixels are highly correlated → redundant information
- Information, Uncertainty and Redundancy
  - Information is uncertainty (to be) resolved
  - Redundancy is repetition of the information we have
  - Compression is about removing redundancy because
- Entropy and Entropy Coding
  - Entropy is a statistical measure of uncertainty, or a measure of the amount of information to be resolved
  - Entropy coding: approaching the entropy (no-redundancy) limit

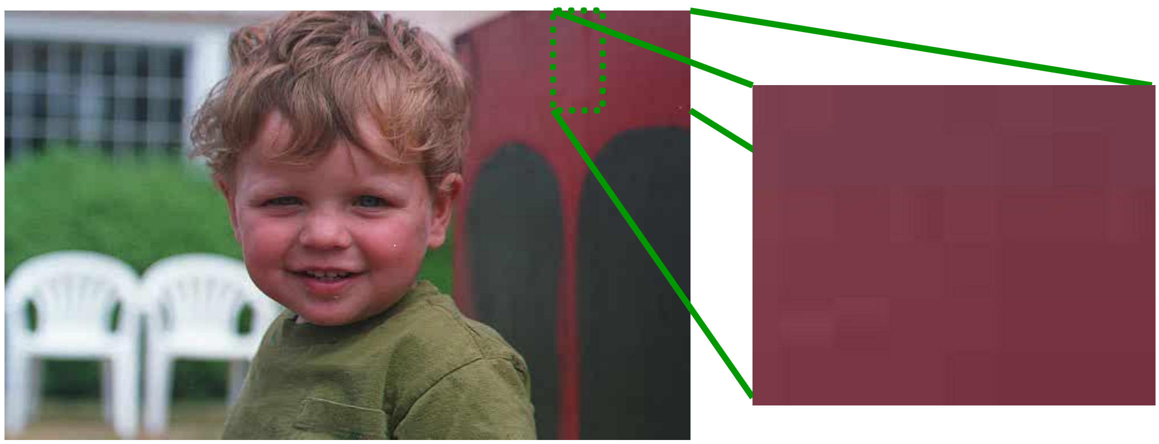
3. Compression Techniques :

Compression techniques take advantage of redundancy in digital images

Types of redundancies

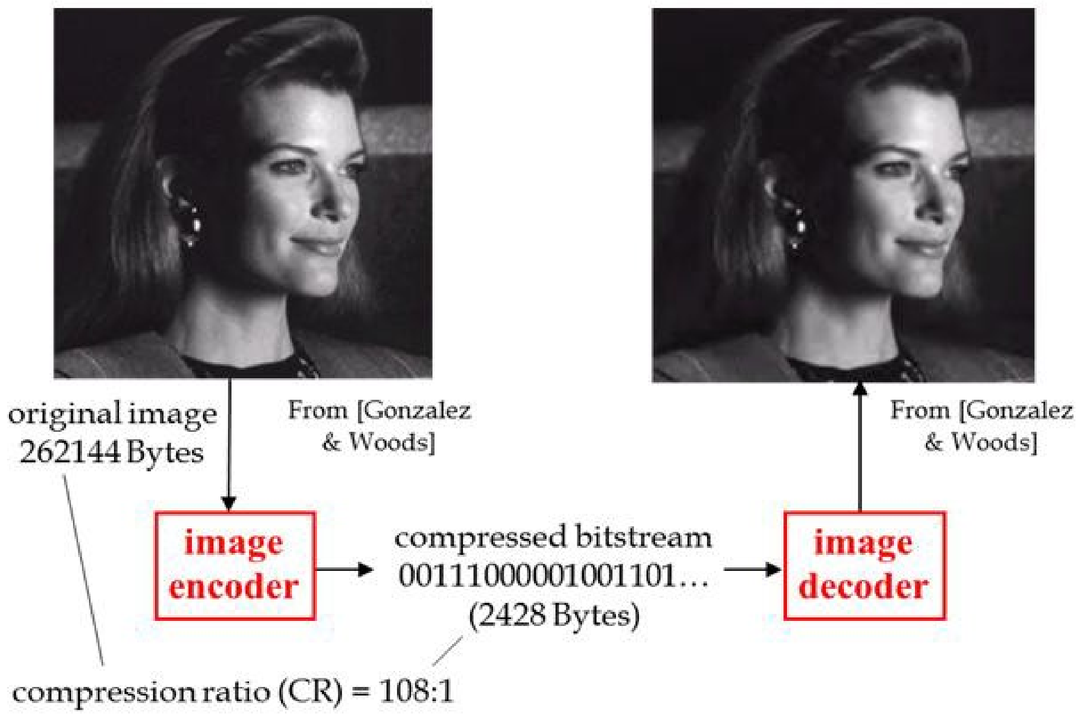
- **Spatial redundancy:** due to the correlation between neighboring pixel values
- **Spectral redundancy:** due to the correlation between different color planes or spectral bands

Spatial Redundancy:



Take advantage of similarity among most neighboring pixels

4. Image Compression: Coding and Decoding:



More Concepts

- Bit Rate and Compression Ratio
  - Bit rate: bits/pixel, sometimes written as bpp
  - Compression ratio (CR)
  - CR= (number of bits to represent the original image / number of bits in compressed stream)
- Binary, Gray-Scale, Color Image Compression
  - Original binary image: 1 bit/pixel
  - Original gray-scale image: typically 8bits/pixel
  - Original Color image: typically 24bits/pixel
  - **Lossless, Nearly lossless and Lossy Compression**
  - Lossless: original image can be exactly reconstructed
  - Nearly lossless: reconstructed image nearly (visually) lossless
  - Lossy: reconstructed image with loss of quality (but higher CR)

5. Lossless Compression:

- In lossless compression, the integrity of the data is preserved because the compression and decompression algorithms are exact inverses of each other: no part of the data is lost in the process.
- Lossless compression methods are normally used when we cannot afford to lose any data. For example, we must not lose data when we compress a text file or an application program.
- Some Lossless compression methods:
  - Run–Length Encoding
  - Dictionary coding
  - Huffman coding

5.1. Run-Length Encoding (RLE):

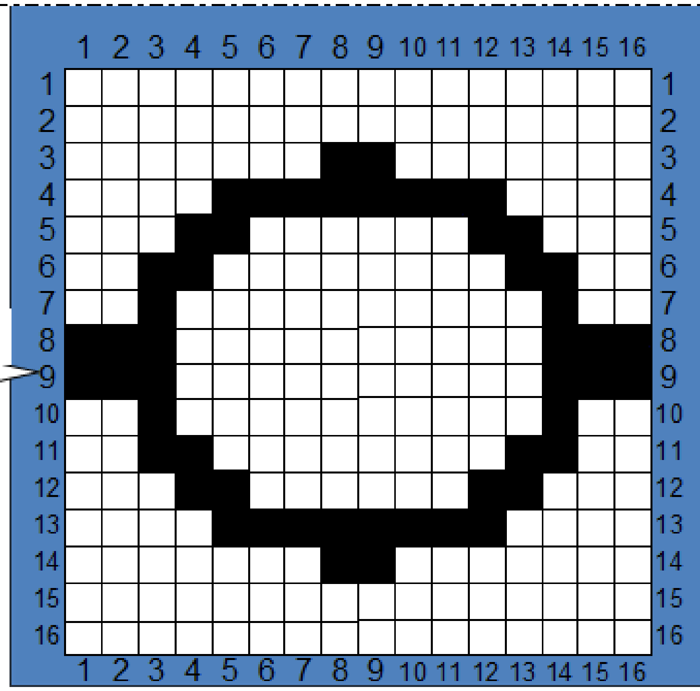
- Run–length coding, sometimes referred to as run–length encoding (RLE), is the simplest method of removing redundancy. It can be used to compress data made of any combination of symbols.
- The method replaces a repeated sequence, run, of the same symbol with two entities: a count and the symbol itself.

5.2. Run Length Coding: Decoding Example

A binary image is encoded using run length code row by row, with “0” represents white, and “1” represents black. The code is given by

Row 1: "0", 16
Row 2: "0", 16
Row 3: "0", 7, 2, 7
Row 4: "0", 4, 8, 4
Row 5: "0", 3, 2, 6, 3, 2
Row 6: "0", 2, 2, 8, 2, 2
Row 7: "0", 2, 1, 10, 1, 2
Row 8: "1", 3, 10, 3
Row 9: "1", 3, 10, 3
Row 10: "0", 2, 1, 10, 1, 2
Row 11: "0", 2, 2, 8, 2, 2
Row 12: "0", 3, 2, 6, 3, 2
Row 13: "0", 4, 8, 4
Row 14: "0", 7, 2, 7
Row 15: "0", 16
Row 16: "0", 16

decode

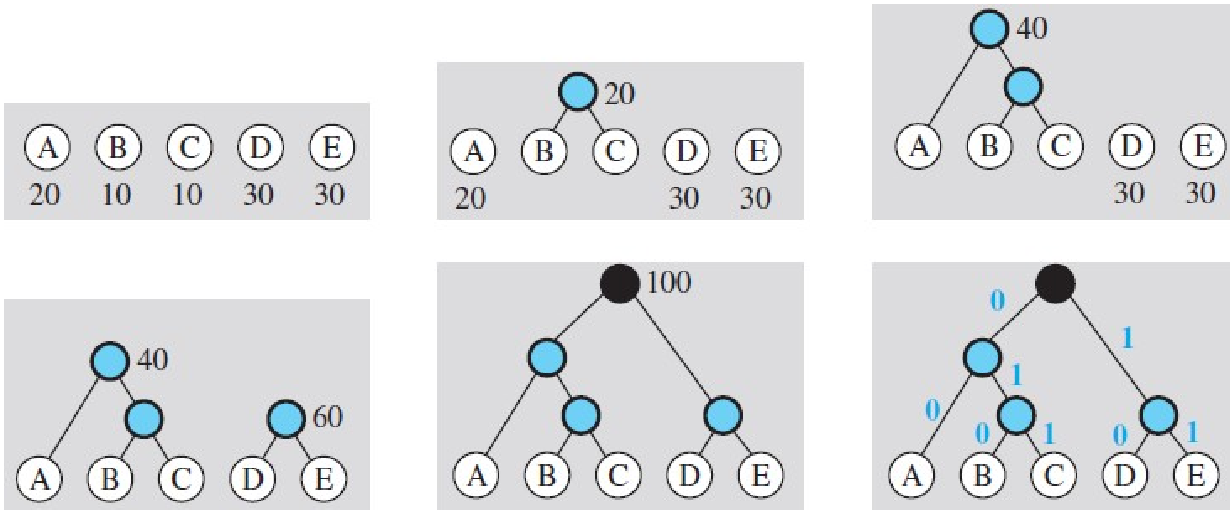


5.3. Huffman Coding:

- When we encode data as binary patterns, we normally use a fixed number of bits for each symbol. To compress data, we can consider the frequency of symbols and the probability of their occurrence in the message. Huffman coding assigns shorter codes to symbols that occur more frequently and longer codes to those that occur less frequently.
- For example, imagine we have a text file that uses only five characters (A, B, C, D, E) with the frequency of occurrence of (20, 10, 10, 30, 30).

5.4. Huffman Tree:

The Huffman tree is a tree in which the leaves of the tree are the symbols. It is made so that the most frequent symbol is the closest to the root of the tree (with the minimum number of nodes to the root) and the least frequent symbol is the farthest from the root. Figure shows the process.



6. Coding Table:

Symbol	*Code*	*Symbol*	*Code*	*Symbol*	*Code*
A	00	**C**	011	**E**	11
B	010	**D**	10		

After the tree has been made, we can create a table that shows how each character can be encoded and decoded. The code for each character can be found by starting at the root and following the branches that lead to that character.

7. Variable Word Length Coding: Example

- Variable–length code that assigns the shortest possible code words to the most probable gray levels.
- A 4x4 4bits/pixel original image is given by

Default Code Book

0: 0000
1: 0001
2: 0010
3: 0011
4: 0100
5: 0101
6: 0110
7: 0111
8: 1000
9: 1001
10: 1010
11: 1011
12: 1100
13: 1101
14: 1110
15: 1111

2	8	6	6
6	8	8	8
8	8	10	10
9	10	10	14

encode

0010	1000	0110	0110
0110	1000	1000	1000
1000	1000	1010	1010
1001	1010	1010	1110

Bit rate = 4bits/pixel

Total # of bits used to represent the image:

4x16 = 64 bits

- Encode the original image with a CODE BOOK given left

Huffman Code Book

0: 0000000
1: 0000001
2: 0001
3: 0000010
4: 0000011
5: 0000100
6: 01
7: 0000101
8: 10
9: 00100
10: 11
11: 0000110
12: 0000111
13: 001010
14: 0011
15: 001011

2	8	6	6
6	8	8	8
8	8	10	10
9	10	10	14

encode

0001	10	01	01
01	10	10	10
10	10	11	11
00100	11	11	0011

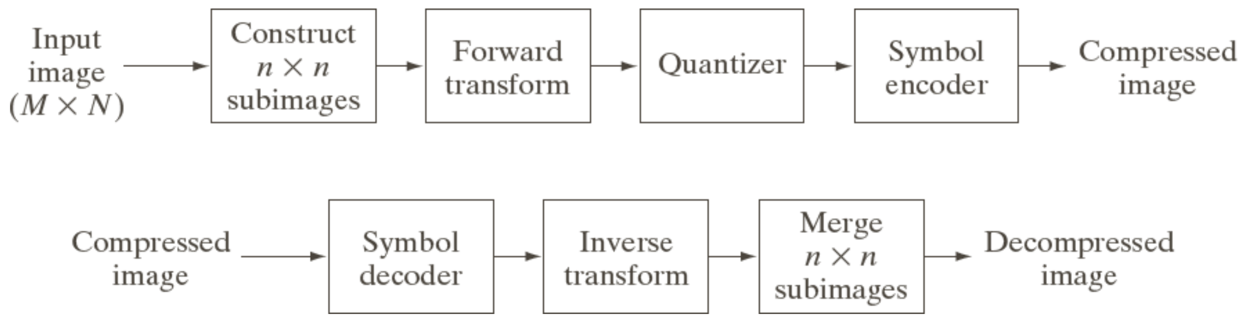
Total # of bits used to represent the image:

4+2+2+2+2+2+2+2+2+
2+2+2+5+2+2+4
= 39 bits

Bit rate = 39/16
= 2.4375 bits/pixel

CR = 64/39 = 1.6410

8. Transform Based Compression:

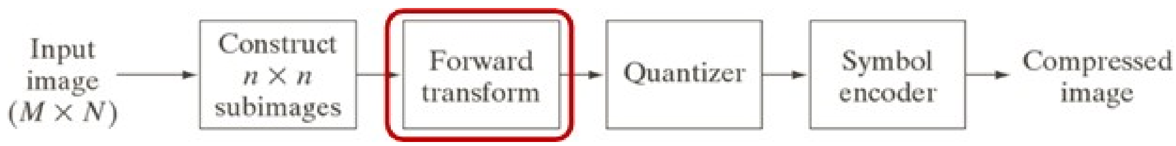


General Compression Model:

- Encoder vs. Decoder

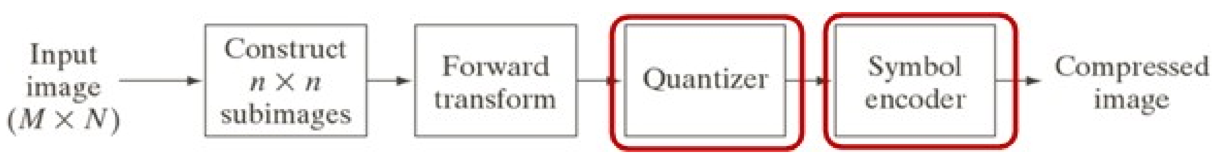
9. Image Compression Model:

9.1. Encoder → Forward transform:



- The Forward transform (mapper) transforms the input data into a format designed to reduce interpixel redundancies in the input image. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image.
- Run–length coding is an example of a mapping that directly results in data compression in this initial stage.
- In the opposite case; the mapper transforms the image into an array of coefficients, making its inter pixel redundancies more accessible for compression in later stages of the encoding process. (eg. DCT)

9.2. Quantizer & Symbol Encoder:



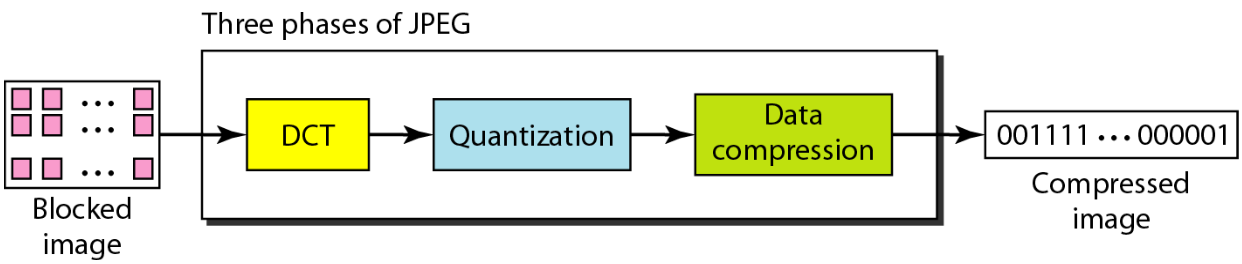
1. The Quantizer block reduces the accuracy of the mapper's output in accordance with some pre–established fidelity criterion. This stage reduces the psychovisual redundancies of the input image. (This operation is irreversible)
2. The symbol encoder creates a fixed– or variable–length code to represent the quantizer output and maps the output in accordance with the code.

9.3. Decoder:

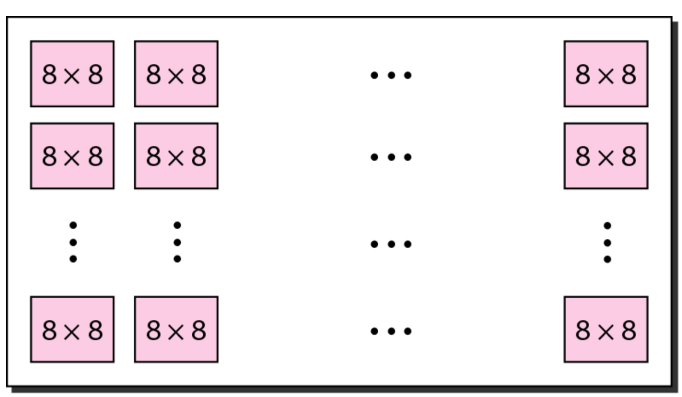
- The decoder contains the components:
    1. a Symbol decoder,
    2. an inverse transform (inverse mapping)
    3. Merging
- These blocks perform, in reverse order, the inverse operations of the source encoder's symbol encoder and mapper blocks.
- Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general source decoder model.

10. Example:

JPEG compression:

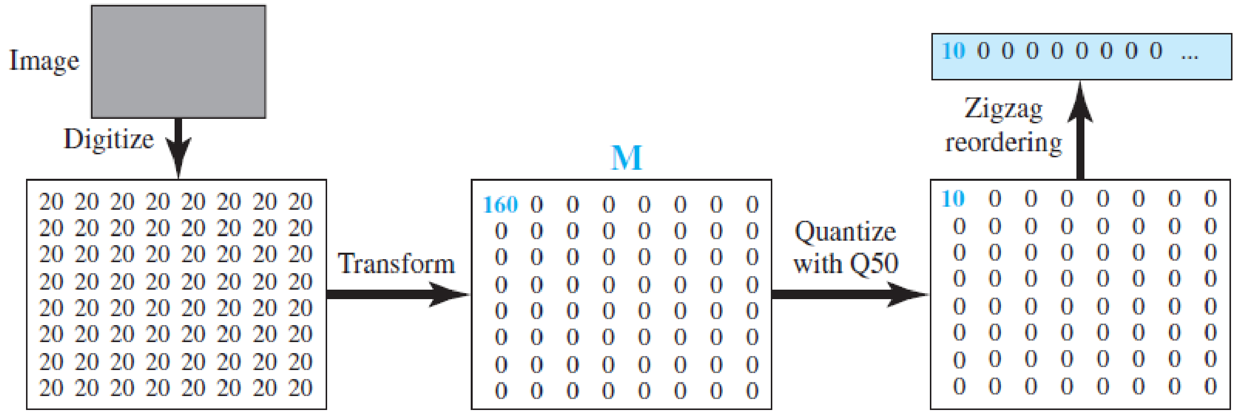


JPEG gray scale:

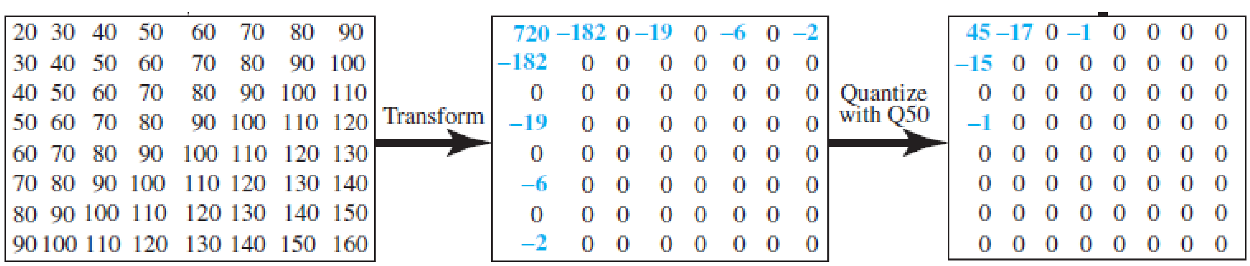


Blocked Image

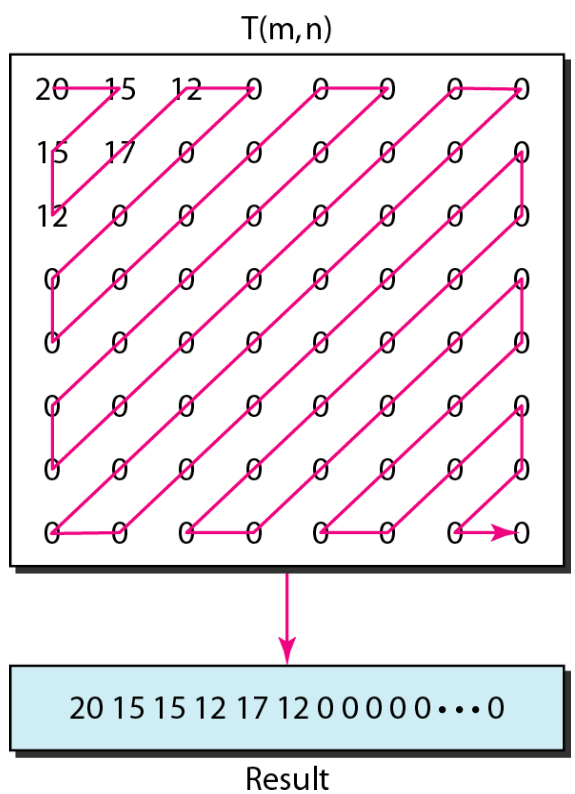
11. Case 1 : DCT on uniform gray scale :



12. Case 2: DCT on Gradient gray scale:



13. Reading the table:



14. Discrete Cosine Transform (DCT):

- **2D DCT:**

$$F(u,v) = \frac{2C(u)C(v)}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} 1(x,y) \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

- **Inverse 2D DCT:**

$$I(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

The constants C(u) and C(v) are determined by:

$$\boxed{C(u) = \begin{cases} \frac{\sqrt{2}}{2} & if\ u = 1 \\ 1 & otherwise \end{cases}}$$

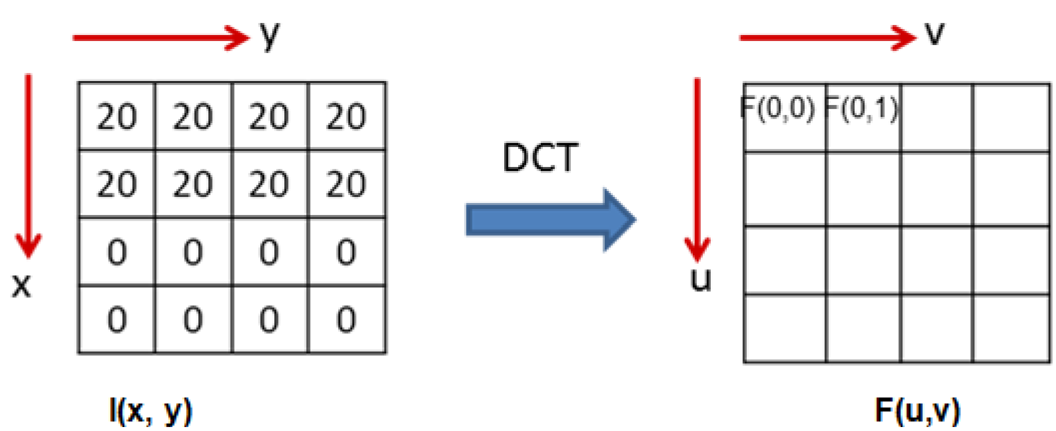
15. DCT for JPEG case:

In JPEG compression M = N = 8

$$F(u,v)=\frac{C(u)C(v)}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} I(x,y)cos\left(\frac{(2x+1)u\pi}{16}\right)cos\left(\frac{(2y+1)v\pi}{16}\right)$$

16. Example 1 (DCT):

Consider the sub–image below, find the coefficient of the DCT at F(0, 0) and F(0, 1).

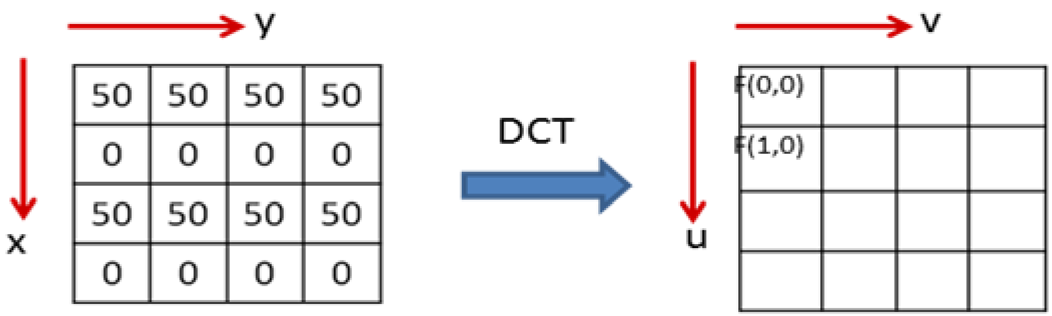


$$F(0,0)=\frac{1}{4}\sum_{x=0}^{3}\sum_{y=0}^{3} I(x,y)\cos(0)\cos(0)=40$$

$$F(0,1)=\frac{\sqrt{2}}{4}\sum_{x=0}^{3}\sum_{y=0}^{3} I(x,y)\cos(0)\,cos\left[\frac{(2y+1)\pi}{8}\right]=10\sqrt{2}\left[cos\left(\frac{\pi}{8}\right)+\left(\frac{3\pi}{8}\right)+cos\left(\frac{5\pi}{8}\right)+cos\left(\frac{7\pi}{8}\right)\right]$$

17. Example 2 (DCT):

Consider the sub-image below, find the coefficient of the DCT at F(0, 0) and F(1, 0).



I(x, y) **F(u,v)**

$$F(0{,}0) = \frac{1}{4}\sum_{x=0}^{3}\sum_{y=0}^{3} I(x,y)\cos(0)\cos(0) = 100$$

$$F(1{,}0) = \frac{\sqrt{2}}{4}\sum_{x=0}^{3}\sum_{y=0}^{3} I(x,y)cos\left[\frac{(2x+1)\pi}{8}\right]\cos\,(0)$$

$$= \frac{\sqrt{2}}{4}\left[50cos\left(\frac{\pi}{8}\right) + 50cos\left(\frac{\pi}{8}\right) + 50cos\left(\frac{\pi}{8}\right) + 50cos\left(\frac{\pi}{8}\right)\right]$$

$$= 50\sqrt{2}\left[cos\left(\frac{\pi}{8}\right)\right] + 50\sqrt{2}\left[cos\left(\frac{5\pi}{8}\right)\right.$$

18. 2D – DCT:

image block

$$\begin{bmatrix} 55 & 36 & -34 & 25 & 66 & 35 & 4 & 37 \\ 55 & 25 & -12 & 48 & 59 & 38 & 2 & 41 \\ 51 & 40 & 43 & 54 & 51 & 42 & 3 & 39 \\ 49 & 49 & 51 & 49 & 51 & 37 & 3 & 39 \\ 50 & 50 & 51 & 48 & 54 & 36 & 2 & 43 \\ 51 & 52 & 52 & 51 & 55 & 41 & 4 & 41 \\ 51 & 51 & 52 & 54 & 55 & 42 & 1 & 45 \\ 52 & 51 & 53 & 51 & 53 & 42 & 2 & 41 \end{bmatrix}$$

2D-DCT

DC component

low frequency

high frequency

low frequency

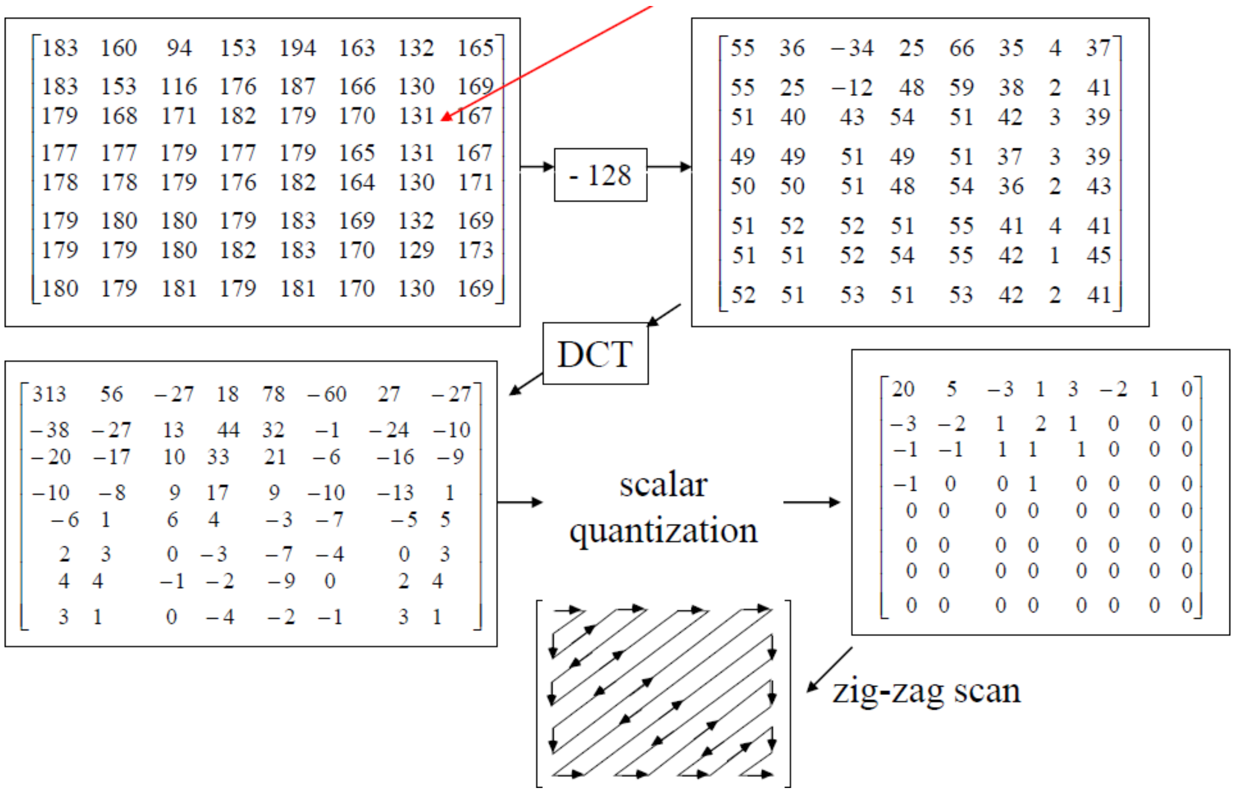
high frequency

$$\begin{bmatrix} 313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\ -38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\ -20 & -17 & 10 & 33 & 21 & -6 & -16 & -9 \\ -10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\ -6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\ 2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\ 4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\ 3 & 1 & 0 & -4 & -2 & -1 & 3 & 1 \end{bmatrix}$$

DCT block

19. JPEG Compression:

- Partition the image into 8x8 blocks, for each block



- Adjust Quantization Step to Achieve Tradeoff between CR and distortion

Original: 100KB

JPEG: 9KB

JPEG: 5KB

– Artifacts:

Inside blocks: blurring (why?); Across blocks: blocking (why?)

Chapter 4
Audio

Keywords:

Digital Audio, Sampling, Nyquist Theorem, Quantization, Pulse Code Modulation.

Abstract:

In this chapter we look at Analog to Digital Sound Conversion, Sampling, sampling rate, Nyquist Theorem, Quantization, the basic Pulse Code Modulation, differentiated PCM, Adaptive PCM, Signal–to–Noise Ratio, and the various Audio file formats.

Learning Outcomes:

At the end of this lesson the student will:

- Understand the process of Analog to Digital Sound Conversion, Sampling, Nyquist Theorem, and Quantization,
- Learn the different stages of Pulse Code Modulation.

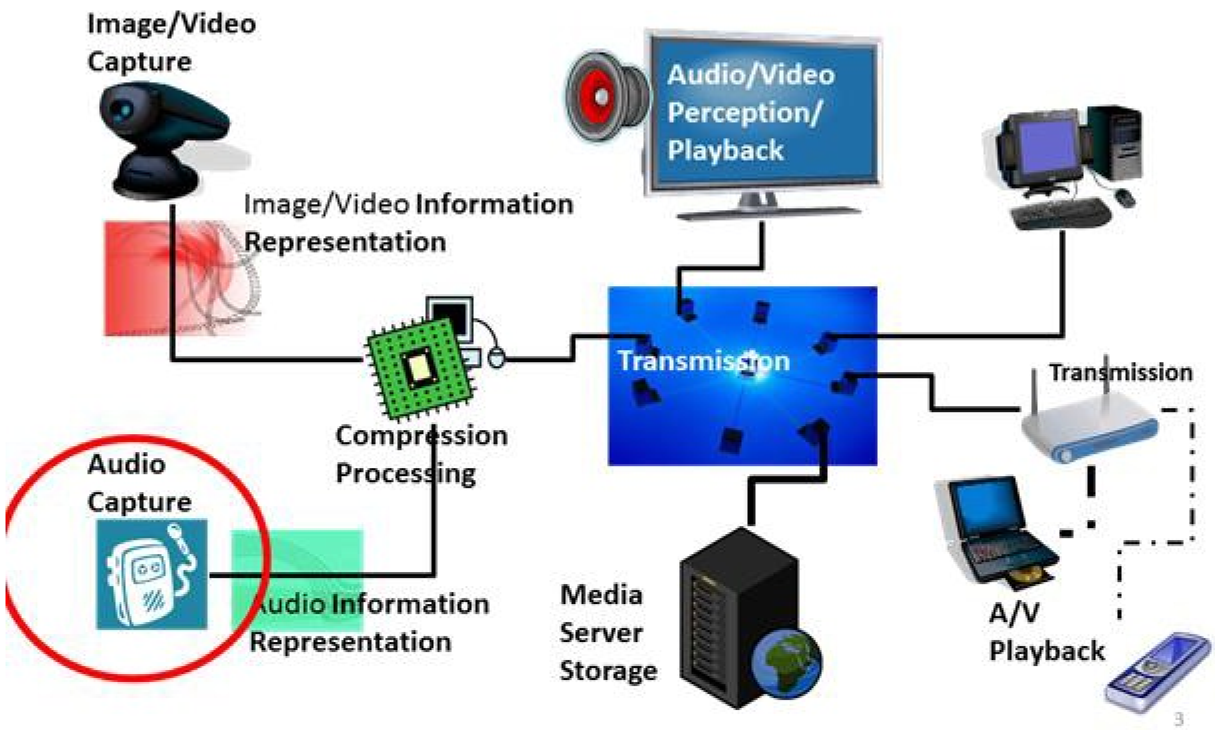
Outline:

- Integrating Aspects of Multimedia
- Characteristics of Sound
- Digital Representation of Audio
  - Sampling
  - Quantization
- Audio Files
- Pulse Code Modulation (PCM)

References for this chapter:

Fundamentals of Multimedia, Ze–Nian Li and Mark S. Drew, 2^{nd} ed.

1. Integrating Aspects of Multimedia:



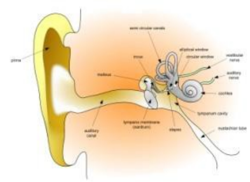
- How can a continuous wave form be converted into discrete samples?
- How can discrete samples be converted back into a continuous form?

2. What is SOUND?

Sound comprises the spoken word, voices, music and even noise. It is a complex relationship involving:

- a vibrating object (sound source)

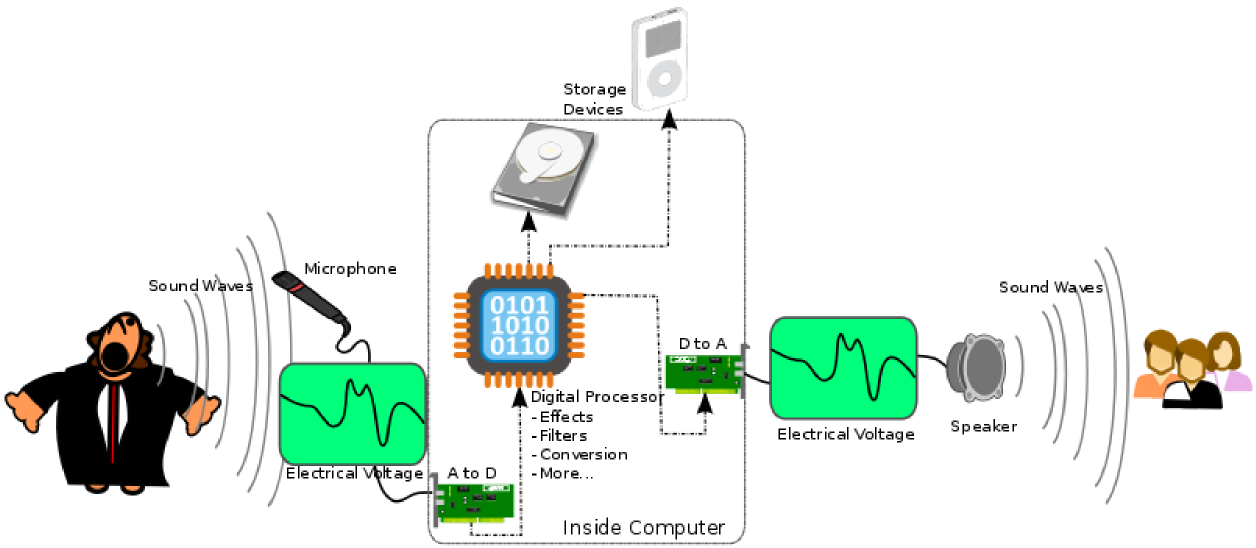
- a transmission medium (usually air)



- a receiver (ear) and;

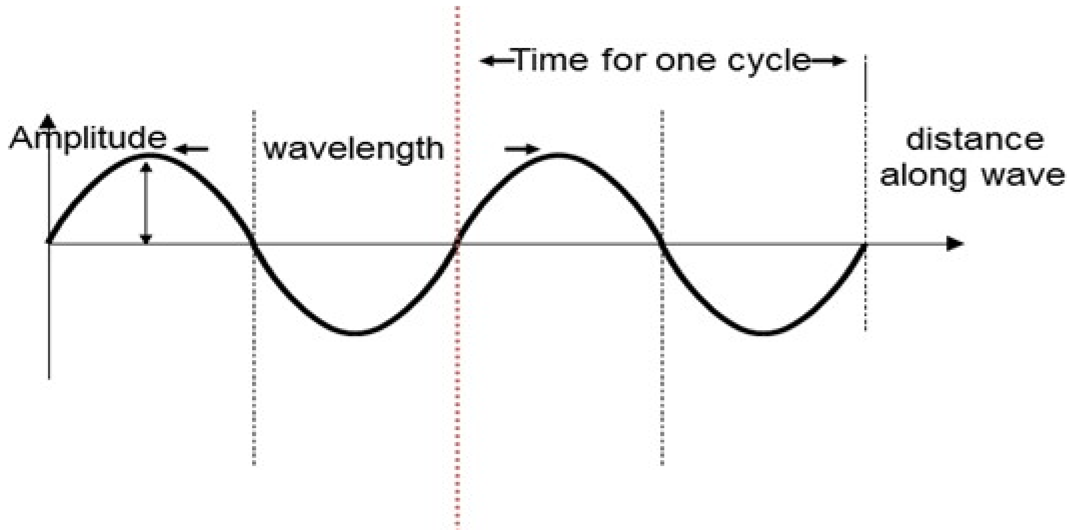
- a preceptor (brain).

3. Lifecycle from Sound to Digital to Sound:



Microphone converts mechanical sound waves into electrical ones. The output of the Analog to Digital Conversion stage is fed to the processing or storage devices. The output stage consist of the Digital to Analog Converter which convert the filtered digital signal or processed digital signal an into analog again to be fed to speakers.

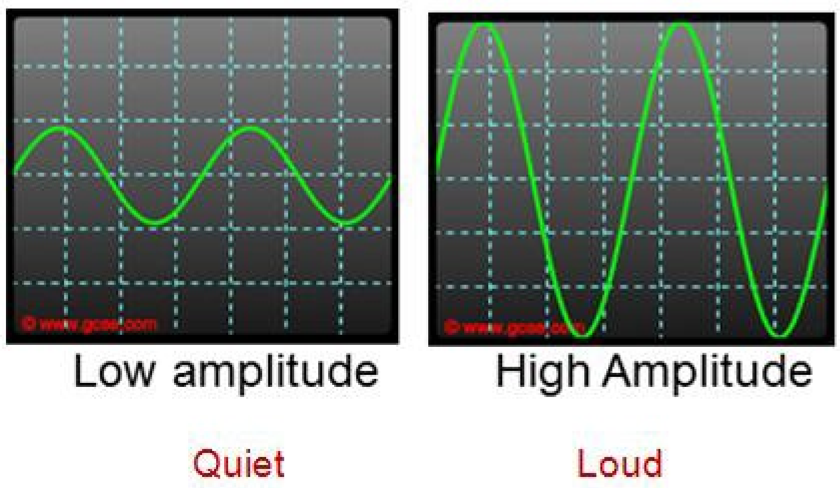
4. Characteristic of Sound Waves:



- Amplitude A
- Wavelength
- Frequency (F)
- Hearing: [20Hz – 20KHz]
- Speech: [200Hz – 8KHz]

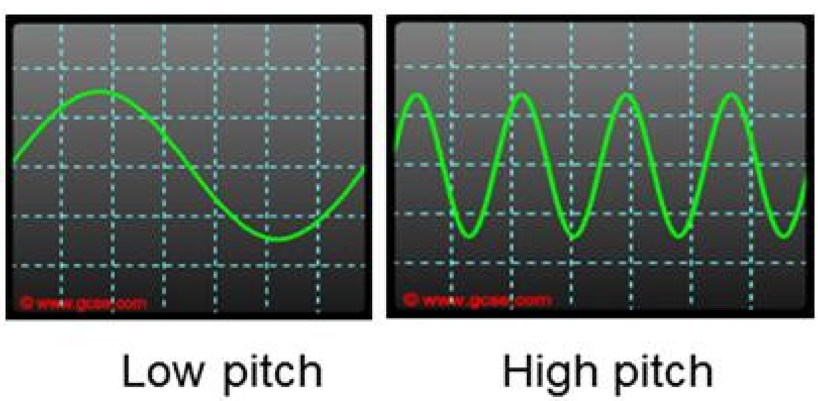
4.1 Amplitude:

The louder a sound, the more energy it has. This means loud sounds have a large amplitude.

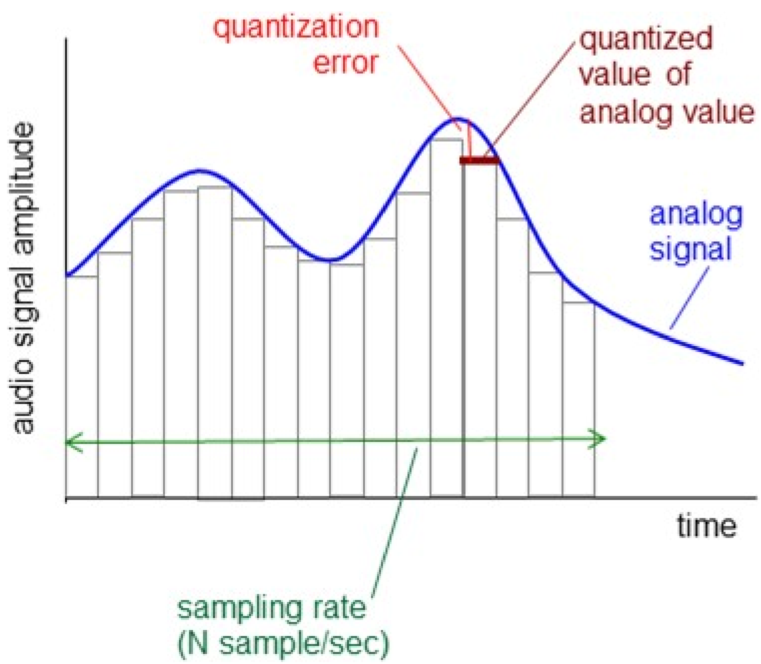


4.2 Frequency:

Frequency is a measure of how many vibrations occur in one second. This is measured in Hertz (abbreviation Hz) and directly corresponds to the pitch of a sound. The more frequent vibration occurs the higher the pitch of the sound.



5. Digital Representation of Audio:

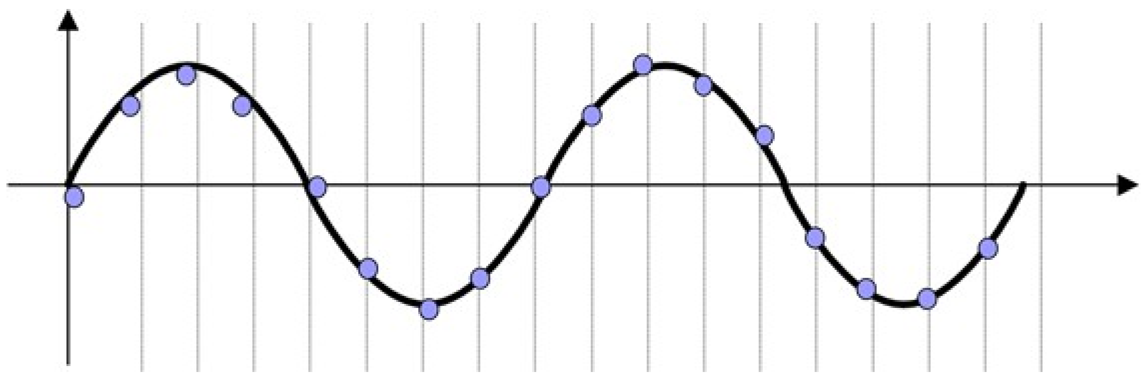


Must convert wave form to digital

- sample
- quantize
- compress

In order to process the analog audio signals by computers it must be converted to digital which involve: sampling, quantization and coding. We look at these stages in some details in the following slides.

5.1 Sampling (in time):



- Measure amplitude at regular intervals, but how many times should we sample?
- Sampling theorem: "If a signal is sampled at regular intervals at a rate higher than twice the highest signal frequency, the samples contain all information in original signal"
- eg. 4000Hz voice data, requires 8000 sample per sec

Nyquist Theorem

- For lossless digitization, the sampling rate should be at least twice the maximum frequency.
- In mathematical terms: $f_s > 2*f_m$
- where f_s is sampling frequency and f_m is the maximum frequency in the signal

Nyquist Limit

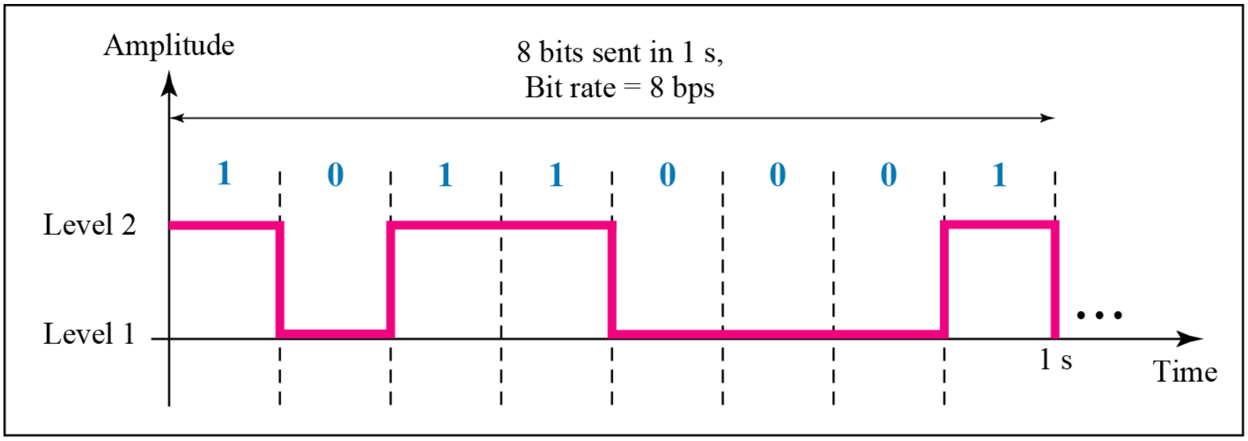
- Max data rate = 2 H $\log_2 V$ *bits/second,* where
  - *H* = bandwidth (in Hz)
  - *V* = discrete levels (bits per signal change)
- Shows the maximum number of bits that can be sent per second on a *noiseless* channel with a bandwidth of H, if V bits are sent per signal

Example: what is the maximum data rate for a 3kHz channel that transmits data using 2 levels (binary) ?

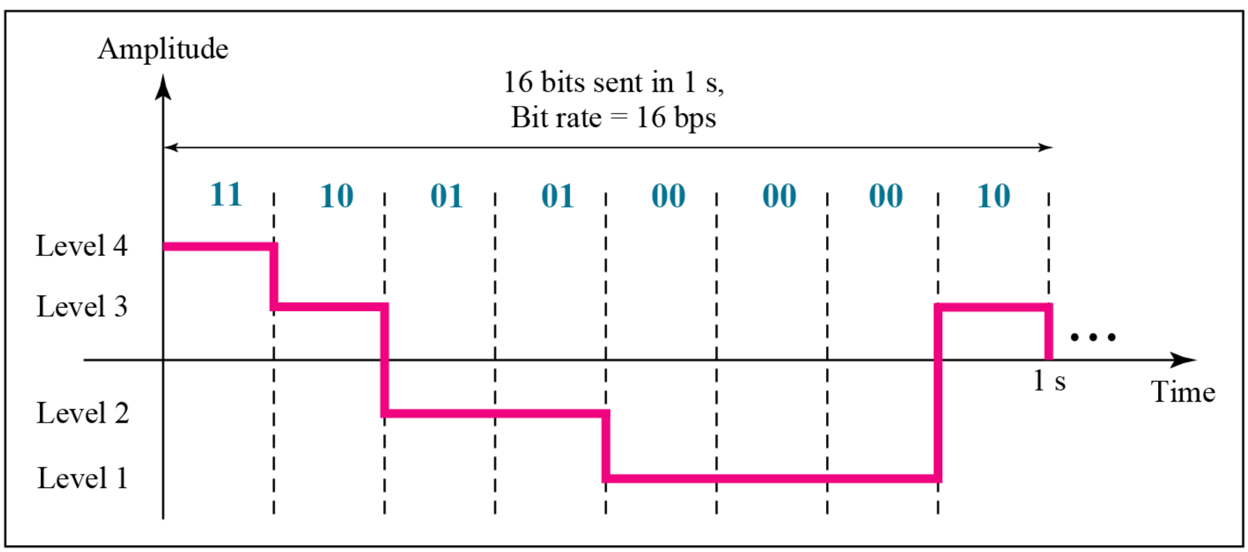
Solution: H = 3kHz; V = 2;

max data rate = 2 x 3,000 x $\log_2 2$ = 6,000bits/second

Illustration of signal levels:



a. A digital signal with two levels



b. A digital signal with four levels

With two signal levels we can only send one bit per signal element.

With four signal levels each signal element can represent more than one bit (two bits)

Limited Sampling

- But what if one cannot sample fast enough?
- →Reduce signal frequency to half of maximum sampling frequency
  - low–pass filter removes higher–frequencies
  - e.g., if max sampling frequency is 22kHz, must low–pass filter a signal down to 11kHz

Common Sampling rates:

Sampling Rates	Used As....
8000	Telephony Standard, Popular in UNIK Workstations
11000	Quarter of CD rate, Popular on Macintosh
16000	G.722 Standars (Federal Standard)
18900	CD–ROM XA Rate
22000	Half CD rate, Macintosh rate
32000	Japanese HDTV, British TV audio, Long play DAT
37800	CD XA Standard
44056	Professional audio industry
44100	CD Rate
48000	DAT Rate

5.2 Quantization:

- Typically use
  - 8 bits = 256 levels
  - 16 bits = 65,536 levels
- How should the levels be distributed?
  - Linearly? (PCM)
  - Perceptually? (u–Law)
  - Differential? (DPCM)
  - Adaptively? (ADPCM)

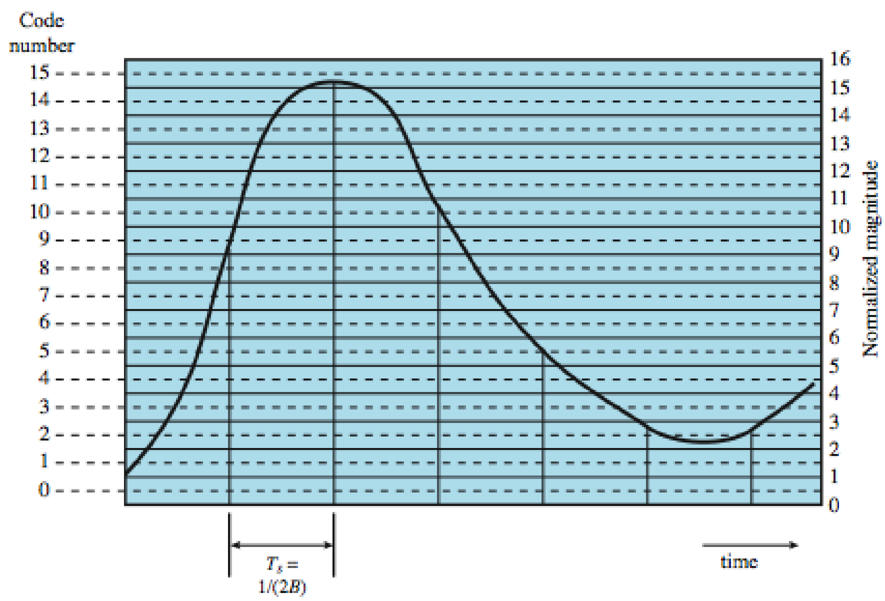
6. Audio Files:

- A sound file's format is simply a recognized methodology for organizing and (usually) compressing the digitized sound's data bits and bytes into a data file.
- There are many ways to store the bits and bytes that describe a sampled waveform sound.
- The main method used is: **Linear Pulse Code Modulation (LPCM)**, often shortened to **PCM**

7. Pulse Code Modulation (PCM):

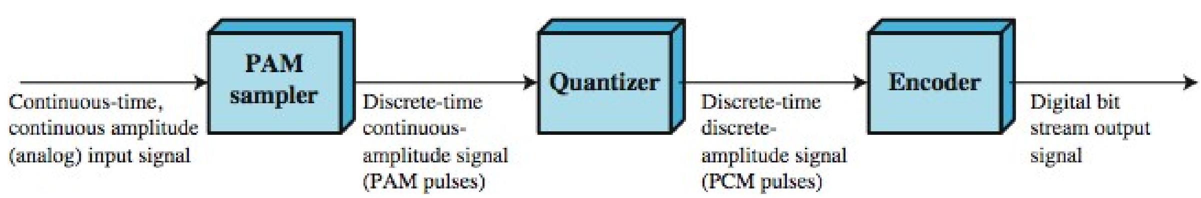
- Pulse modulation
  - Use discrete time samples of analog signals
  - Transmission is composed of analog information sent at different times
  - Variation of pulse amplitude or pulse timing allowed to vary continuously over all values
  - Pulse Amplitude Modulation (PAM)
- PCM
  - Analog signal is quantized into a number of discrete levels

7.1 PCM Example:



PAM value	1.1	9.2	15.2	10.8	5.6	2.8	2.7
quantized code number	1	9	15	10	5	2	2
PCM code	0001	1001	1111	1010	0101	0010	0010

7.2 PCM Block Diagram:



8. Audio Formats:

MP3 is the newest format for compressed recorded music. The term MP3 has become synonymous with digital music.

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Commonly used in music players. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5.
Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. MP4 video is the upcoming video format on the internet. This leads to automatic support for MP4 audio by all browsers.

9. Practice Part:

Record audio with Sound Recorder (Windows)

All Programs → Accessories → Sound Recorder

Sound Recorder lets you record a sound and save it as an audio file on your computer. You can record sound from different audio devices, such as a microphone that's plugged into the sound card on your computer. The types of audio input sources you can record from depend on the audio devices you have and the input sources on your sound card.

Sound Recorder

1. Make sure you have an audio input device, such as a microphone, connected to your computer.
2. Click to open Sound Recorder.
3. Click Start Recording.
4. To stop recording audio, click Stop Recording.
5. (Optional) If you want to continue recording audio, click Cancel in the Save As dialog box, and then click Resume Recording. Continue to record sound, and then click Stop Recording.
6. Click the File name box, type a file name for the recorded sound, and then click Save to save the recorded sound as an audio file.

Note:

To use Sound Recorder, you must have a sound card and speakers installed on your computer. If you want to record sound, you also need a microphone (or other audio input device).

Chapter 5
Audio Compression

Keywords:

Psychoacoustic model, Frequency Masking, Temporal Masking, Perceptual Coding.

Abstract:

In this chapter we look at the famous audio compression technique: MPEG Audio that uses the Psychoacoustics model of hearing. First we introduce the Psychoacoustics model, and then we present the MPEG audio compression that takes advantage of psychoacoustic models. A main technique used in evaluating audio content for possible compression makes use of a psychoacoustic model of hearing and generally referred to as perceptual coding.

Learning Outcomes:

At the end of this lesson the student have a strong foundation about:

- The Psychoacoustics model of hearing.
- MPEG audio compression technique and know how it takes advantage of psychoacoustic model.

- ## Reference:

Fundamentals of Multimedia, Ze-Nian Li and Mark S. Drew, 2nd ed.

1. Introduction:

The most common compression technique used to create CD–quality audio is perceptual coding, which is based on the science of psychoacoustics. Algorithms used in perceptual coding first transform the data from time domain to frequency domain; the operations are then performed on the data in the frequency domain.

2. Psychoacoustics:

Psychoacoustics is the study of subjective human perception of sound. Perceptual coding takes advantage of flaws in the human auditory system. The lower limit of human audibility is 0 dB. This is only true for sounds with frequencies of about 2.5 and 5 kHz. The lower limit is less for frequencies between these two frequencies and rises for frequencies outside these ranges, as shown in Figure 5–0 (a) below. We cannot hear any frequency whose power is below this curve; thus, it is not necessary to code such a frequency.

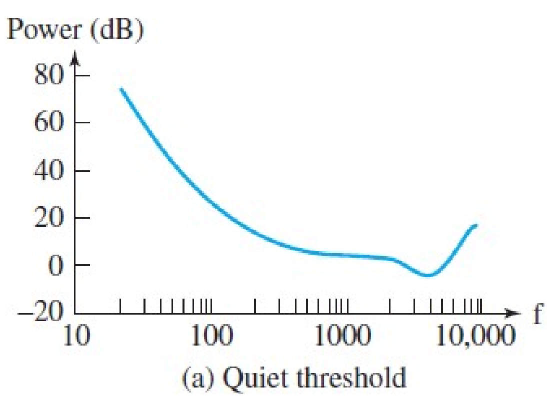


Fig 5–0

We can save even more using the concepts of frequency masking and temporal masking. Frequency masking occurs when a loud sound partially or totally masks a softer sound if the frequencies of the two are close to each other. In Figure 5–0 (b) above, a loud masking tone, around 700 Hz, raises the threshold of the audibility curve between frequencies of about 250 to 1500 Hz.

- The range of human hearing is about 20 Hz to about 20 kHz (Sounds at higher frequencies are ultrasonic)

- The frequency range of the voice is typically only from about 500 Hz to 4 kHz
- The dynamic range, the ratio of the maximum sound amplitude to the quietest sound that humans can hear, is on the order of about 120 dB

2.1. Fletcher-Munson Curves:

The Fletcher–Munson equal–loudness curves display the relationship between perceived loudness (in phons) for a given stimulus sound volume (Sound Pressure Level, dB), as a function of frequency. Fig.5.1 shows the ear's perception of equal loudness.

- The bottom curve shows what level of pure tone stimulus is required to produce the perception of a 10 dB sound
- All the curves are arranged so that the perceived loudness level gives the same loudness as for that loudness level of a pure tone at 1 kHz
- The abscissa—the x axis is frequency, in kHz.
- The ordinate axis is sound pressure level—the actual intensity (i.e., loudness) of the tone generated in an experiment.

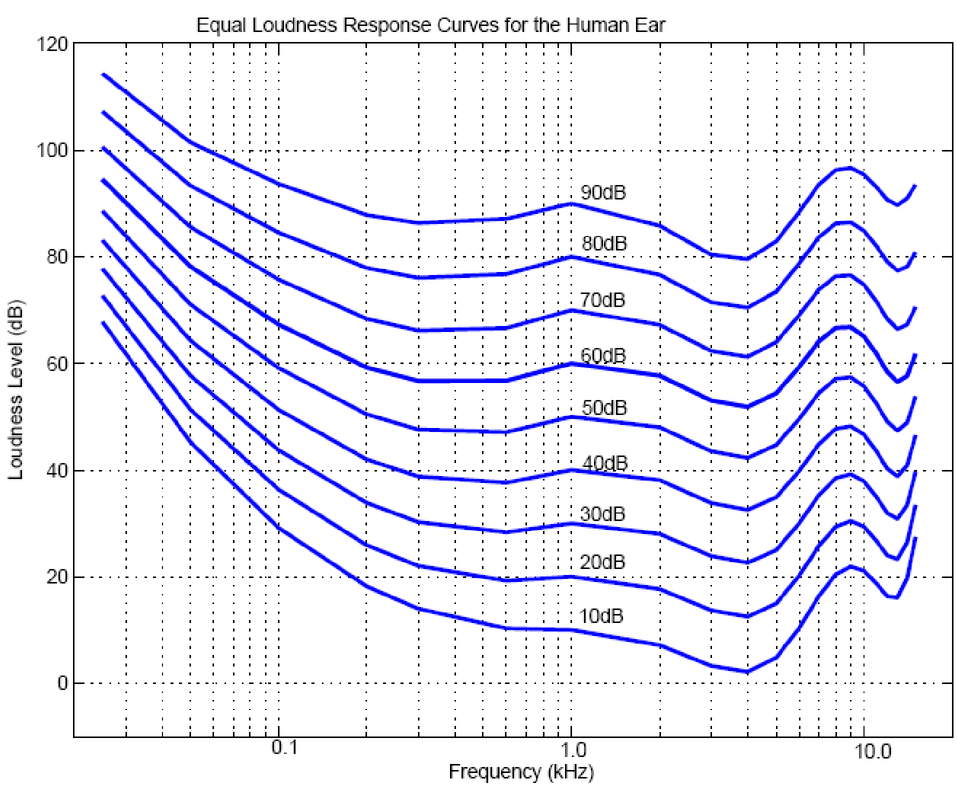


Fig. 5.1: Flaetcher–Munson Curves

2.2. Frequency Masking:

Lossy audio data compression methods, such as MPEG/Audio encoding, remove some sounds which are masked anyway

The general situation in regard to masking is as follows:

1. A lower tone can effectively mask (make us unable to hear) a higher tone
2. The reverse is not true – a higher tone does not mask a lower tone well
3. The greater the power in the masking tone, the wider is its influence – the broader the range of frequencies it can mask.
4. As a consequence, if two tones are widely separated in frequency then little masking occurs

How does one tone interfere with another? This question is answered by masking curves. Also, masking answers the question of how much noise we can tolerate before we cannot hear the actual music. Lossy audio data compression methods, such as MPEG Audio encoding, which is popular in movies, remove some sounds that are masked anyway, thus reducing the total amount of information.

Threshold of Hearing

A plot of the threshold of human hearing for a pure tone

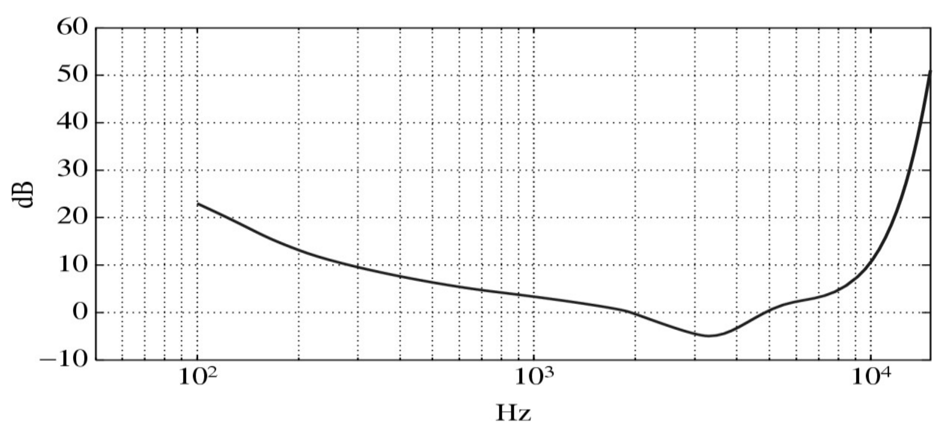


Fig 5-2: The threshold of hearing curve

The threshold of hearing curve:

if a sound is above the dB level shown then the sound is audible (Fig 5-2).

Turning up a tone so that it equals or surpasses the curve means that we can then distinguish the sound

An approximate formula exists for this curve:

The threshold units are dB; the frequency for the origin (0,0) in the above formula is 2,000 Hz: Threshold(f) = 0 at f =2 kHz

2.3. Frequency Masking Curves:

- Frequency masking is studied by playing a particular pure tone, say 1 kHz again, at a loud volume, and determining how this tone affects our ability to hear tones nearby in frequency
  - one would generate a 1 kHz *masking* tone, at a fixed sound level of 60 dB, and then raise the level of a nearby tone, e.g., 1.1 kHz, until it is just audible
- The threshold in Fig.5.3 plots the audible level for a single masking tone (1 kHz)
- Fig. 5.4 shows how the plot changes if other masking tones are used

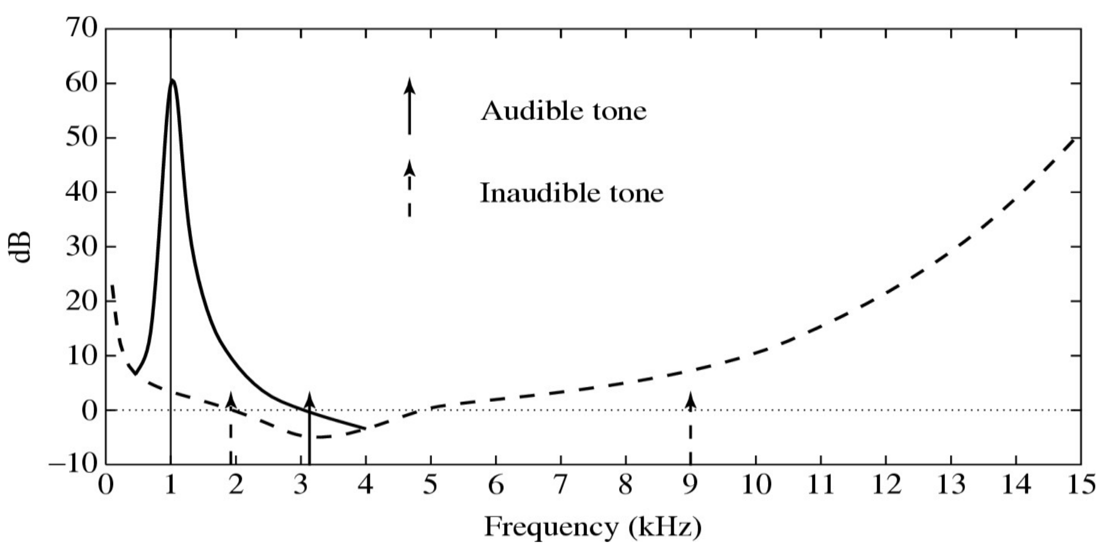


Fig. 5.3 Effect on threshold for 1 kHz masking tone

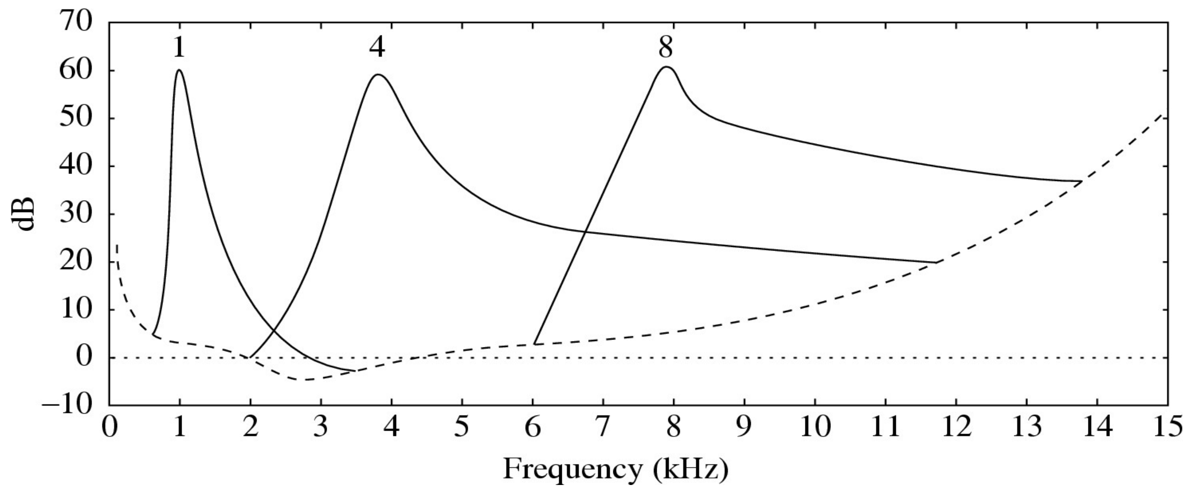


Fig. 5.4: Effect of masking tone at three different frequencies

As a demo listen to the associated video about Psychoacoustics and Masking (Taken from the course: **Internet of Things: Multimedia Technologies**. (coursera)

2.4. Critical Bands:

- **Critical bandwidth** represents the ear's resolving power for simultaneous tones or partials
  - At the low-frequency end, a critical band is less than 100 Hz wide, while for high frequencies the width can be greater than 4 kHz
- Experiments indicate that the critical bandwidth:
- for masking frequencies < 500 Hz: remains approximately constant in width (about 100 Hz)
- for masking frequencies > 500 Hz: increases approximately linearly with frequency
- See the next tables about Critical Bands and Bandwidth

The human hearing range naturally divides into critical bands, with the property that the human auditory system cannot resolve sounds better than within about one critical band when other sounds are present. Hearing has a limited, frequency-dependent resolution.

Critical Bands and Bandwidth:

Band #	Lower Bound (HZ)	Center (HZ)	Upper Bound (HZ)	Bandwidth (HZ)
1	–	50	100	–
2	100	150	200	100
3	200	250	300	100
4	300	350	400	100
5	400	450	510	110
6	510	570	630	120
7	630	700	770	140
8	770	840	920	150
9	920	1000	1080	160
10	1080	1170	1270	190
11	1270	1370	1480	210
12	1480	1600	1720	240
13	1720	1850	2000	280
14	2000	2150	2320	320
15	2320	2500	2700	380
16	2700	2900	3150	450
17	3150	3400	3700	550
18	3700	4000	4400	700
19	4400	4800	5300	900
20	5300	5800	6400	1100
21	6400	7000	7700	1300
22	7700	8500	9500	1800
23	9500	10500	12000	2500
24	12000	13500	15500	3500
25	15500	18775	22050	6550

Bark unit:

Bark unit is defined as the width of one critical band, for any masking frequency

- The idea of the Bark unit: every critical band width is roughly equal in terms of Barks (refer to Fig. 5.5)

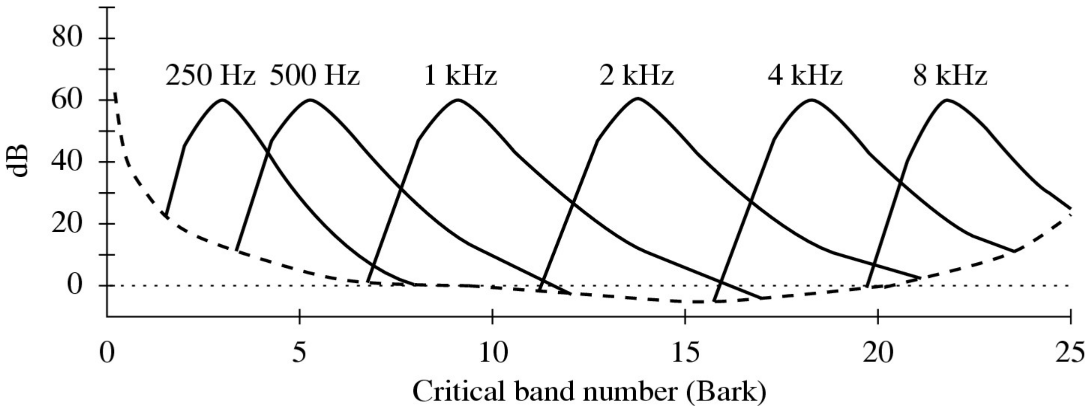


Fig. 5.5: Effect of masking tones, expressed in Bark units

The new unit defined is called the Bark, named after Heinrich Barkhausen (1881–1956), an early sound scientist. One Bark unit corresponds to the width of one critical band, for any masking frequency.

2.5. Temporal Masking:

- The **Phenomenon**: any loud tone will cause the hearing receptors in the inner ear to become *saturated* and require time to recover
- The following figure shows the results of Masking experiments:

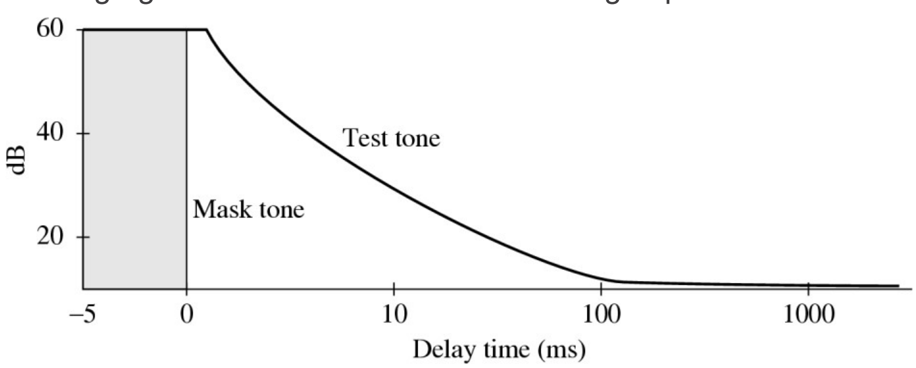


Fig 5-6: results of Masking experiments

The louder is the test tone, the shorter it takes for our hearing to get over hearing the masking.

Recall that after the dance it takes quite a while for our hearing to return to normal. Generally, any loud tone causes the hearing receptors in the inner ear to become saturated, and they require time to recover. (Many other perceptual systems behave in this temporally slow fashion—for example, the receptors in the eye have this same kind of “capacitance” effect.)

3. MPEG audio:

1. **MPEG audio compression** takes advantage of psychoacoustic models, constructing a large multi–dimensional lookup table to transmit masked frequency components using fewer bits
2. MPEG Audio Overview
    - 2.1. Applies a filter bank to the input to break it into its frequency components
    - 2.2. In parallel, a psychoacoustic model is applied to the data for bit allocation block
    - 2.3. The number of bits allocated are used to quantize the info from the filter bank – providing the compression

MPEG audio compression takes advantage of psychoacoustic models, constructing a large multi–dimensional lookup table to transmit masked frequency components using fewer bits

3.1. MPEG Layers:

MPEG audio sets out three layers of audio compression, each able to understand the lower layers. Each offers more complexity in the psychoacoustic model applied and correspondingly better compression for a given level of audio quality. However, an increase in complexity, and concomitantly in compression effectiveness, is accompanied by extra delay.

Layer 1 quality can be quite good, provided a comparatively high bitrate is available

(Digital Audio Tape typically uses Layer 1 at around 192 kbps). While outdated by Layers 2 and 3, Layer 1 formed the basis for MPEG Audio. It is still largely supported, e.g., audio in packages in Ubuntu linux. Layer 2 has more complexity and was proposed for use in digital audio broadcasting. Layer 3 (MP3) is most complex and was originally aimed at audio transmission over ISDN lines. Each of the layers also uses a different frequency transform.

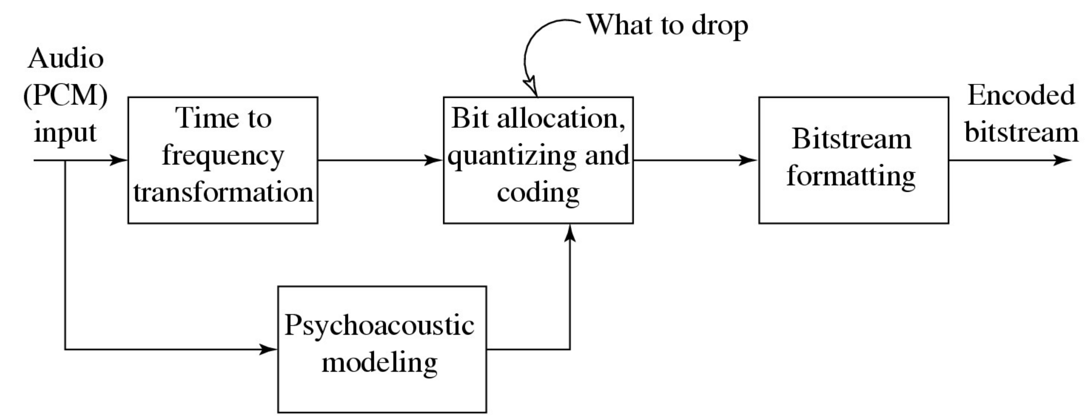
Most of the complexity increase is at the encoder rather than at the decoder side, and this accounts for the popularity of MP3 players. Layer 1 incorporates the simplest psychoacoustic model, and Layer 3 uses the most complex. The objective is a good tradeoff between quality and bitrate.

3.2. MPEG Audio Strategy:

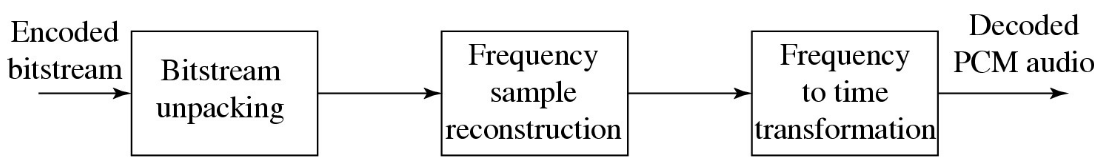
The MPEG approach to compression relies on quantization, of course, but also recognizes that the human auditory system is not accurate within the width of a critical band, both in terms of perceived loudness and audibility of a test frequency. The encoder employs a bank of filters that act to first analyze the frequency components of the audio signal by calculating a frequency transform of a window of signal values. The bank of filters decomposes the signal into subbands. Layer 1 and Layer 2 codecs use a quadrature-mirror filter bank, while the Layer 3 codec adds a DCT. For the psychoacoustic model, a Fourier transform is used.

- **Frequency masking:** by using a psychoacoustic model to estimate the just noticeable noise level:
  - Encoder balances the masking behavior and the available number of bits by discarding inaudible frequencies
  - Scaling quantization according to the sound level that is left over, above masking levels
  - Scaling quantization according to the sound level that is left over, above masking levels

3.3. MPEG Audio Compression Algorithm:



(a) MPEG Audio Encoder



(b) MPEG Audio Decoder

Fig 5-7 Basic MPEG Audio encoder and decoder.

The Figure 5-7 shows the basic MPEG audio compression algorithm. It proceeds by dividing the input into 32 frequency subbands, via a filter bank. This is a linear operation that takes as its input a set of 32 PCM samples, sampled in time, and produces as its output 32 frequency coefficients. If the sampling rate is f_s = 48 Ksps (kilosamples per second), then by the Nyquist theorem, the maximum frequency mapped will be $f_s/2$. Thus, the mapped bandwidth is divided into 32 equal-width segments, each of width $f_s/64$ (these segments overlap somewhat).

- The algorithm proceeds by dividing the input into 32 frequency subbands, via a filter bank

- A linear operation taking 32 PCM samples, sampled in time; output is 32 frequency coefficients
- In the Layer 1 encoder, the sets of 32 PCM values are first assembled into a set of 12 groups of 32s
  - an inherent time lag in the coder, equal to the time to accumulate 384 (i.e., 12×32) samples
- Fig. 5-8 shows how samples are organized
  - A Layer 2 or Layer 3, frame actually accumulates more than 12 samples for each subband: a frame includes 1,152 samples

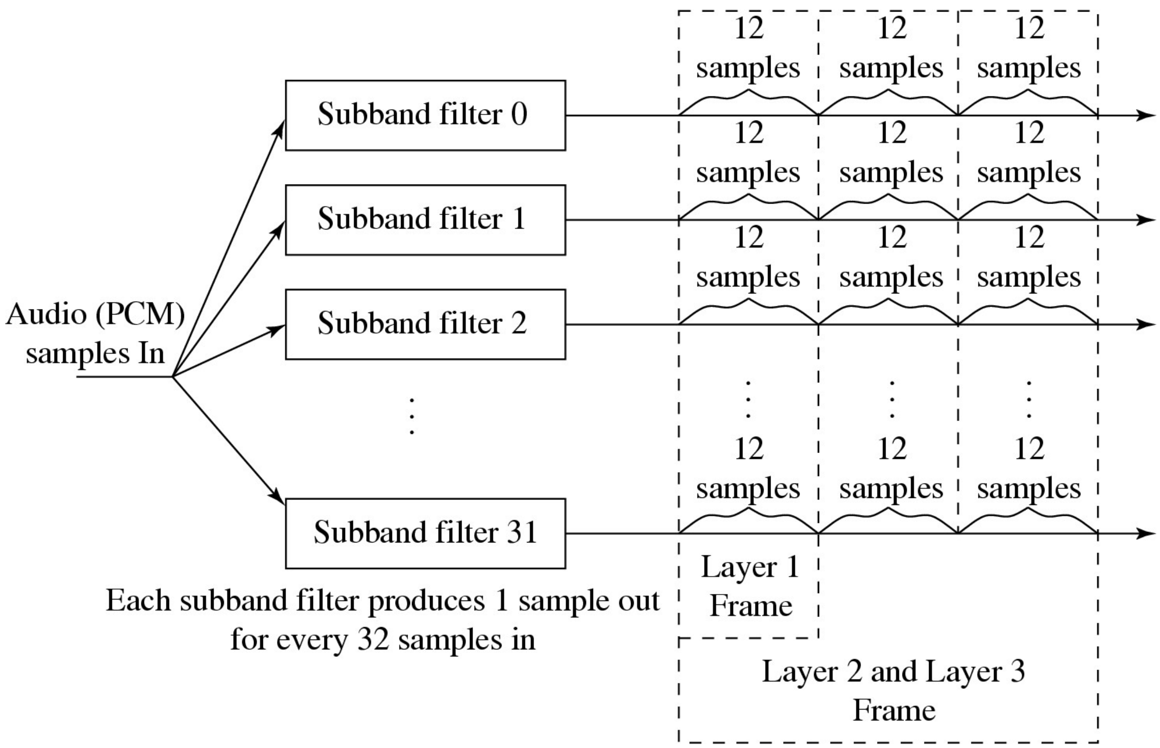


Fig 5-8 MPEG Audio Frame Sizes

3.4. Bit Allocation Algorithm:

- **Aim**: ensure that all of the quantization noise is below the masking thresholds
- **One common scheme**:

- For each subband, the psychoacoustic model calculates the *Signal–to–Mask Ratio* (SMR)in dB
- Then the "Mask–to–Noise Ratio" (MNR) is defined as the difference (as shown in Fig. 5.9):
- The lowest MNR is determined, and the number of code–bits allocated to this subband is incremented
- Then a new estimate of the SNR is made, and the process iterates until there are no more bits to allocate

MNR and SMR:

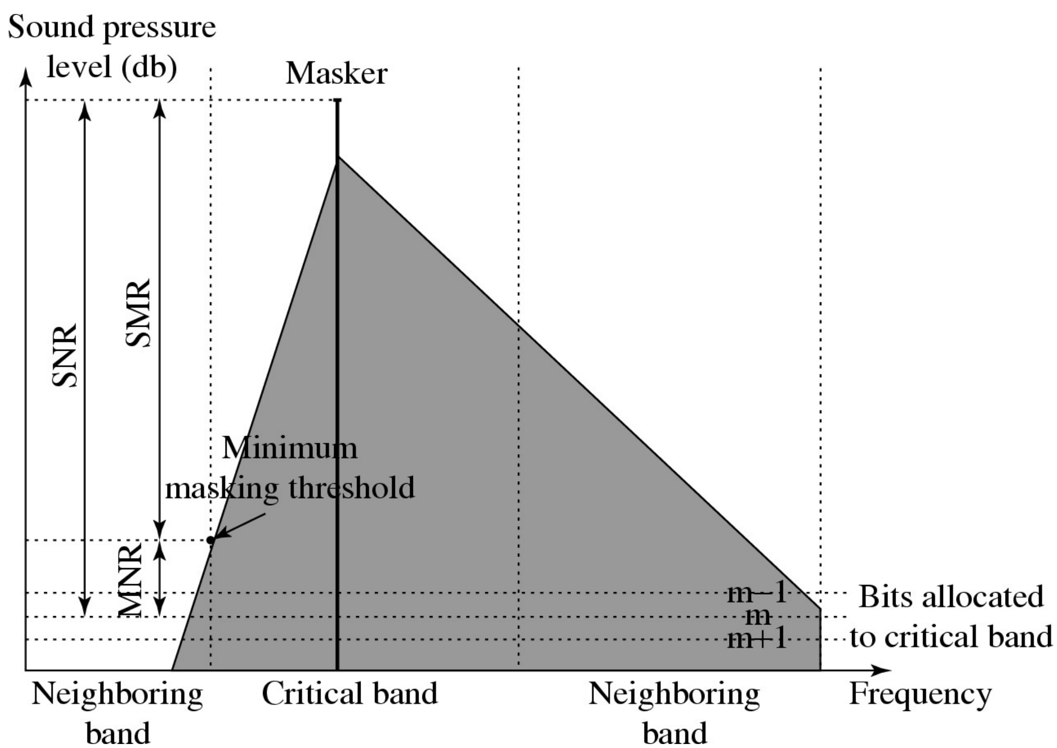


Fig. 5.9: MNR and SMR. A qualitative view of SNR, SMR and MNR are shown, with one dominate masker and m bits allocated to a particular critical band.

Mask calculations are performed in parallel with subband filtering.

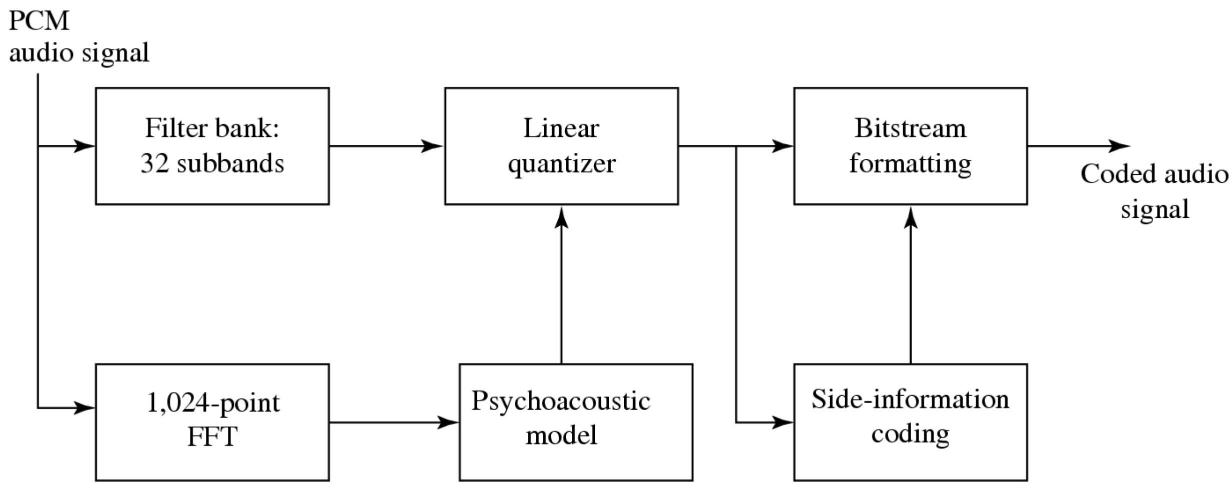


Fig 5–10 MPEG–1 Audio Layers 1 and 2

In Mask calculations are performed in parallel with subband filtering, as in Figure 5–10. The masking curve calculation requires an accurate frequency decomposition of the input signal, using a Discrete Fourier Transform (DFT). The frequency spectrum is usually calculated with a 1,024–point Fast Fourier Transform (FFT).

Chapter 6
Fundamental Concepts in Video

Keywords:

Digital video, sequence of digital images, HDTV.

Abstract:

we consider the following aspects of video and how they impact multimedia applications: Digital video, Video display interfaces and High Definition TV.

Learning Outcomes:

At the end of this lesson the student will:

- Be familiar with the basic concepts of digital video
- Understand the process of Chroma Subsampling
- Explore the Standards for Digital Video

Outline:

- Digital Video
- Chroma Subsampling
- Standards for Digital Video
- HDTV
- Video Display Interfaces

Reference:

Fundamentals of Multimedia, Ze-Nian Li and Mark S. Drew, 2^{nd} ed.

1. Digital Video:

The advantages of digital representation for video are many. For example:

- Video can be stored on digital devices or in memory, ready to be processed (noise removal, cut and paste, etc.), and integrated to various multimedia applications;
- Direct access is possible, which makes nonlinear video editing achievable as a simple, rather than a complex, task;
- Repeated recording does not degrade image quality;
- Ease of encryption and better tolerance to channel noise.

In earlier Sony or Panasonic recorders, digital video was in the form of composite video. Modern digital video generally uses component video, although RGB signals are first converted into a certain type of color opponent space. The usual color space is YCbCr.

2. Chroma Subsampling:

- The reduction of colour resolution in digital component video signals in order to save storage and bandwidth. The colour components are compressed by sampling them at a lower rate than the brightness (luma).
- Since humans see colour with much less spatial resolution than they see black and white, it makes sense to “decimate” the chrominance signal.
- Interesting names have arisen to label the different schemes used.
- To begin with, numbers are given stating how many pixel values, per four original pixels, are actually sent:

Since humans see colour with much less spatial resolution than black and white, it makes sense to decimate the chrominance signal. Interesting but not necessarily informative names have arisen to label the different schemes used.

- The chroma subsampling scheme “4:4:4” indicates that no chroma subsampling is used: each pixel's Y, Cb and Cr values are transmitted, 4 for each of Y, Cb, Cr.

- The scheme "4:2:2" indicates horizontal subsampling of the Cb, Cr signals by a factor of 2. That is, of four pixels horizontally labelled as 0 to 3, all four Ys are sent, and every two Cb's and two Cr's are sent, as (Cb0, Y0)(Cr0, Y1)(Cb2, Y2)(Cr2, Y3)(Cb4, Y4), and so on.
- The scheme "4:1:1" subsamples horizontally by a factor of 4.
- The scheme "4:2:0" subsamples in both the horizontal and vertical dimensions by a factor of 2. Theoretically, an average chroma pixel is positioned between the rows and columns as shown Fig.6.1.

Scheme 4:2:0 along with other schemes is commonly used in JPEG and MPEG

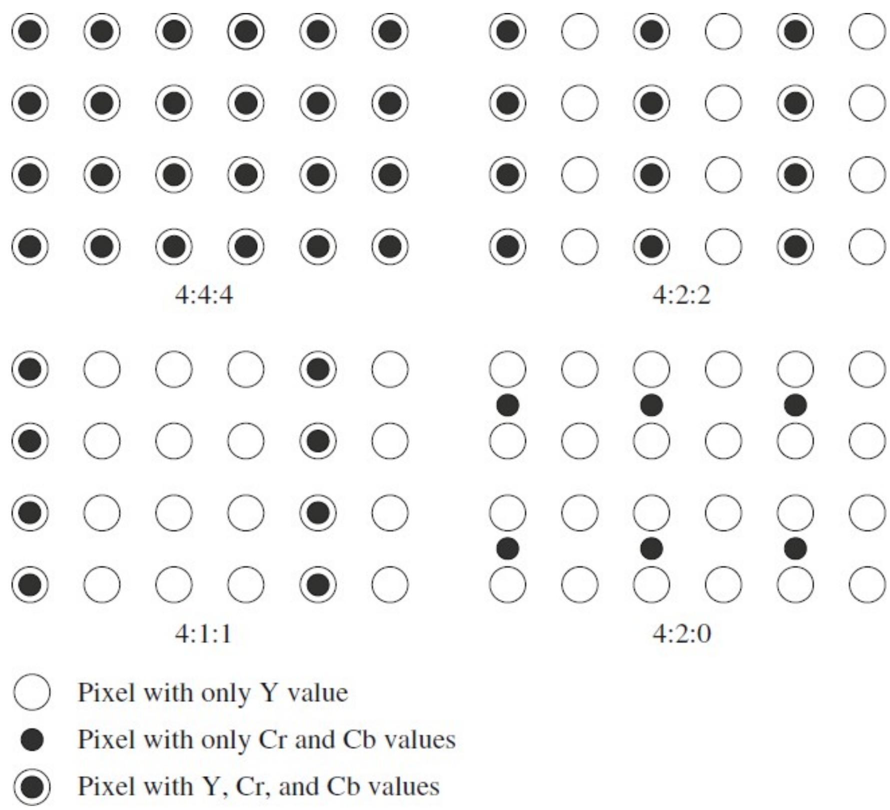


Fig 6-1 Chroma subsampling

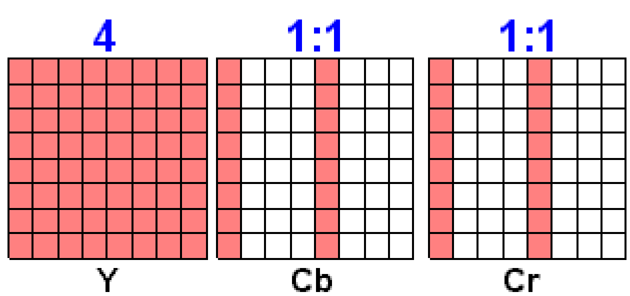


Fig 6-2

3. CCIR Standards for Digital Video:

CCIR is the **C**onsultative **C**ommittee for **I**nternational **R**adio, and one of the most important standards it has produced is CCIR-601, for component digital video.

- This standard has since become standard ITU-R-601, an international standard for professional video applications, adopted by certain digital video formats including the popular DV video.
- Table 6.1 shows some of the digital video specifications, all with an aspect ratio of 4:3. The CCIR 601 standard uses an interlaced scan, so each field has only half as much vertical resolution.

Table 6-1: Digital video specifications

	Rec.601 525/60 NTSC	**Rec.601 625/50 PAL/SECAM**	**CIF**	**QCIF**
Luminance resolution	720×480	720×576	352×288	176×144
Chrominance resolution	360×480	360×576	176×144	88×72
Color subsampling	4:2:2	4:2:2	4:2:0	4:2:0
Aspect ration	4:3	4:3	4:3	4:3
Fields/sec	60	50	30	30
Interlaced	Yes	Yes	Yes	No

4. HDTV (High Definition TV):

The main thrust of HDTV (High Definition TV) is not to increase the "definition" in each unit area, but rather to increase the visual field especially in its width.

- The first generation of HDTV was based on an analog technology developed by Sony and NHK in Japan in the late 1970s.
- MUSE (MUltiple sub-Nyquist Sampling Encoding) was an improved NHK HDTV with hybrid analog/digital technologies that was put in use in the 1990s. It has 1,125 scan lines, interlaced (60 fields per second), and 16:9 aspect ratio.
- Since uncompressed HDTV will easily demand more than 20 MHz bandwidth, which will not fit in the current 6 MHz or 8 MHz channels, various compression techniques are being investigated.
- It is also anticipated that high quality HDTV signals will be transmitted using more than one channel even after compression.

The introduction of wide-screen movies brought the discovery that viewers seated near the screen enjoyed a level of participation (sensation of immersion) not experienced with conventional movies. Apparently the exposure to a greater field of view, especially the involvement of peripheral vision, contributes to the sense of "being there." The main thrust of High-Definition TV (HDTV) is not to increase the "definition" in each unit area, but rather to increase the visual field, especially its width.

First-generation HDTV was based on an analog technology developed by Sony and NHK in Japan in the late 1970s.

5. ATSC (Advanced Television Systems Committee):

The ATSC – is the responsible for the standard for TV broadcasting of HDTV.

The standard supports video scanning formats shown in Table 6.2. In the table, "I" mean interlaced scan and "P" means progressive (non–interlaced) scan.

Table 6–2: Advanced Digital TV formats supported by ATSC

# of Active Pixel per line	# of Active Lines	Aspect Ratio	Picture Rate
1,920	1,080	16:9	60I 30P 24P
1,280	720	16:9	60P 30P 24P
704	480	16: & 4:3	60I 60P 30P 24P
640	480	4:3	60I 60P 30P 24P

- For video, MPEG–2 is chosen as the compression standard. For audio, AC–3 is the standard. It supports the so–called 5.1 channel Dolby surround sound, i.e., five surround channels plus a subwoofer channel.
- The salient difference between conventional TV and HDTV:
    - HDTV has a much wider aspect ratio of 16:9 instead of 4:3.
    - HDTV moves toward progressive (non–interlaced) scan. The rationale is that interlacing introduces serrated edges to moving objects and flickers along horizontal edges.

Consumers with analog TV sets will still be able to receive signals via an 8–VSB (8–level vestigial sideband) demodulation box. The FCC has planned to replace all analog broadcast services with digital TV broadcasting. The services provided will include:

- SDTV (Standard Definition TV): the current NTSC TV or higher.
- EDTV (Enhanced Definition TV): 480 active lines or higher, i.e., the third and fourth rows in Table 3.2.
- HDTV (High Definition TV): 720 active lines or higher.

6. Video Display Interfaces:

Interfaces for video signal transmission from some output devices to a video display (TV, monitor, projector, etc.).

There have been a wide range of video display interfaces, supporting video signals of different formats (analog or digital, interlaced or progressive), different frame rates, and different resolutions.

We start our discussion with some analog interfaces, including Component Video, Composite Video, and S–Video, and then digital interfaces, including DVI, HDMI, and DisplayPort.

We now discuss the interfaces for video signal transmission from some output devices (set–top box, video player, video card, and etc.) to a video display (TV, monitor, projector, etc.). There have been a wide range of video display interfaces, supporting video signals of different formats (analog or digital, interlaced or progressive), different frame rates, and different resolutions. We start our discussion with some analog interfaces, including Component Video, Composite Video, and S–Video, and then digital interfaces, including DVI, HDMI, and DisplayPort.

Component Video:

make use of three separate video signals for the red, green, and blue image planes. This kind of system has three wires (and connectors) connecting the camera or other devices to a TV or monitor.

Composite video:

color ("chrominance") and intensity ("luminance") signals are mixed into a single carrier wave. Chrominance is a composite of two color components (I and Q, or U and V). This is the type of signal used by broadcast color TV;it is downward compatible with black–and–white TV.

Fig 6-3 Connectors for typical analog display interfaces. From left to right: Component video,Composite video, S-video, and VGA

S-Video:

As a compromise, S-video (separated video, or super-video) uses two wires: one for luminance and another for a composite chrominance signal. As a result, there is less crosstalk between the color information and the crucial grayscale information.

Video Graphics Array (VGA):

The Video Graphics Array (VGA) is a video display interface that was first introduced by IBM in 1987, along with its PS/2 personal computers. It has since been widely used in the computer industry with many variations, which are collectively referred to as VGA.

7. Digital Display Interfaces:

Given the rise of digital video processing and the monitors that directly accept digital video signals, there is a great demand toward video display interfaces that transmit digital video signals. Today, the most widely used digital video interfaces include Digital Visual Interface (DVI), High–Definition Multimedia Interface (HDMI), and DisplayPort, as shown in Fig.4

7.1. Digital Visual Interface (DVI):

was developed by the Digital Display Working Group (DDWG) for transferring digital video signals, particularly from a computer's video card to a monitor. It carries uncompressed digital video and can be configured to support multiple modes, including DVI–D (digital only), DVI–A (analog only), or DVI–I (digital and analog).

Fig 6–4 Connectors of different digital display interfaces. From left to right: DVI, HDMI, DisplayPort

7.2. High-Definition Multimedia Interface (HDMI):

HDMI is a newer digital audio/video interface developed to be backward–compatible with DVI. It has been widely used in the consumer market since 2002. The HDMI specification defines the protocols, signals, electrical interfaces, and mechanical requirements.

HDMI differs from DVI in the following aspects:

1. HDMI does not carry analog signal and hence is not compatible with VGA.
2. DVI is limited to the RGB color range (0–255). HDMI supports both RGB and YCbCr 4:4:4 or 4:2:2. The latter are more common in application fields other than computer graphics.
3. HDMI supports digital audio, in addition to digital video.

The maximum pixel clock rate for HDMI 1.0 is 165MHz, which is sufficient to support 1080P and WUXGA (1,920 × 1,200) at 60 Hz. HDMI 1.3 increases that to 340MHz, which allows for higher resolution.

8. Common Video Formats:

Format	File	Description
MPEG	.mpg .mpeg	MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Used to be supported by all browsers, but it is not supported in HTML5 (See MP4).
AVI	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
WMV	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
QuickTime	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers.
RealVideo	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. It is still used for online video and Internet TV, but does not play in web browsers.
Flash	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
Ogg	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
WebM	.webm	WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
MPEG-4 or MP4	.mp4	MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.

Chapter 7
Video Compression

Keywords:

MPEG2, MPEG4, Motion Vector, Temporal Compression, MPEG Transport Stream (TS).

Abstract:

First we present the main services of MPEG compression then we introduce the MPEG2 Transport Stream Construction which is the container for many video transmission techniques.

Learning Outcomes:

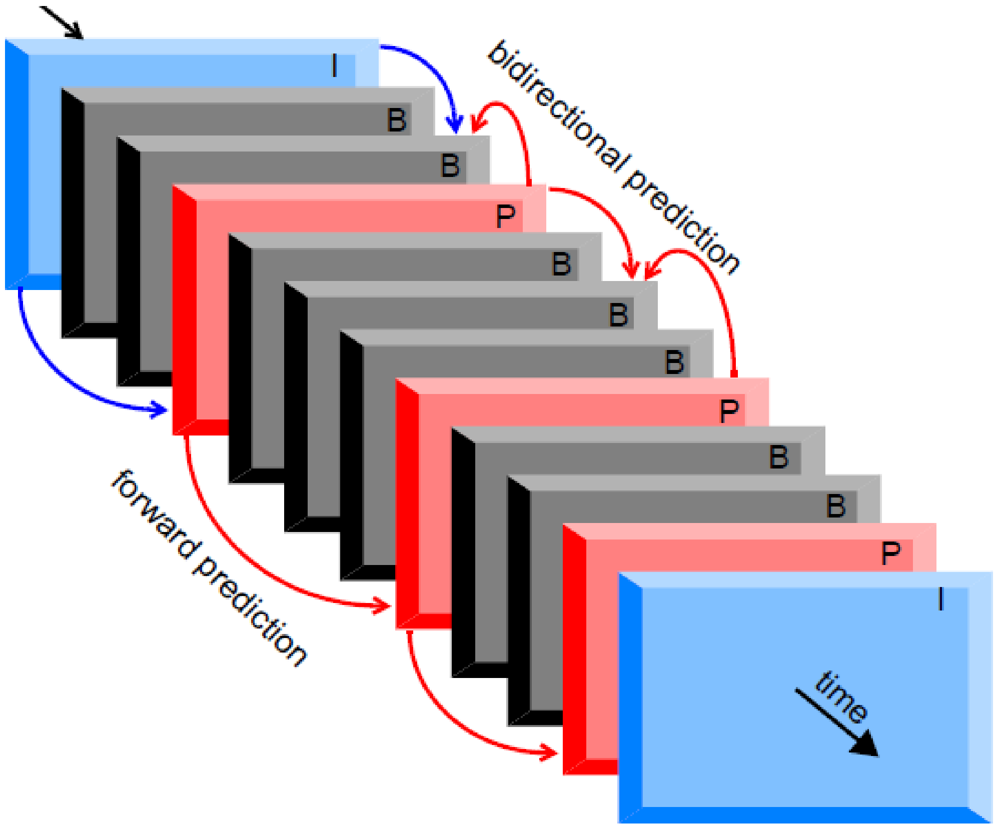
This Lecture will introduce:

- The main characteristics of the Moving Picture Experts Group (MPEG).
- MPEG Transport Stream (TS)

Outline:

- Video Services and MPEG
- MPEG construction
- MPEG Transport Stream (TS)

1. Video structure:



- Video is composed of multiple frames; each frame is one image.
- A video consists of a sequence of frames. If the frames are displayed on the screen fast enough, we get an impression of motion. The reason is that our eyes cannot distinguish the rapidly flashing frames as individual ones.
- Video file requires a high transmission rate.

2. Video Delivery:

- Video must be displayed at 24–30 frames per second. Multimedia video data must be delivered at a rate which guarantees 24–30 frames/second
- In general, a rate of 24 to 30 frames per second is necessary for video to appear smooth to human eyes. (The eye retains the image of each frame for a short time after it has been presented, a characteristic known as persistence of vision. A rate of 24 to 30 frames per second is fast enough to appear continuous.)
- The video file must be accessed from the file system at a rate consistent with the rate at which the video is being displayed.
- Continuous–media data is data with specific rate requirements

Example:

Let us show the transmission rate for some video standards:

- Color broadcast television takes 720 × 480 pixels per frame, 30 frames per second, and 24 bits per color. The transmission rate without compression is as shown below.

  720 x 480 x 30 x 24 = 249 Mbps

- High definition color broadcast television takes 1920 × 1080 pixels per frame, 30 frames per second, and 24 bits per color: The transmission rate without compression is as shown below.

  1920 080 x 30 x 24 = 1.5 Gbps

3. Compression:

Compression plays a crucial role in multimedia communication due to the large volume of data exchanged. In compression, we reduce the volume of data to be exchanged.

Because of the size and rate requirements of multimedia systems, multimedia files are often compressed into a smaller form before transmission.

3.1. MPEG Compression:

- MPEG-2 – Used for compressing DVD and high-definition television (HDTV)
- MPEG-4 – Used to transmit audio, video, and graphics. Can be delivered over very slow connections (56 Kbps)

3.2. Video Compression: MPEG

- Motion Picture Experts Group (MPEG) is a popular method to compress video.

1. In principle, a motion picture is a rapid flow of a set of frames, where each frame is an image. In other words, a frame is a spatial combination of pixels, and a video is a temporal combination of frames that are sent one after another.

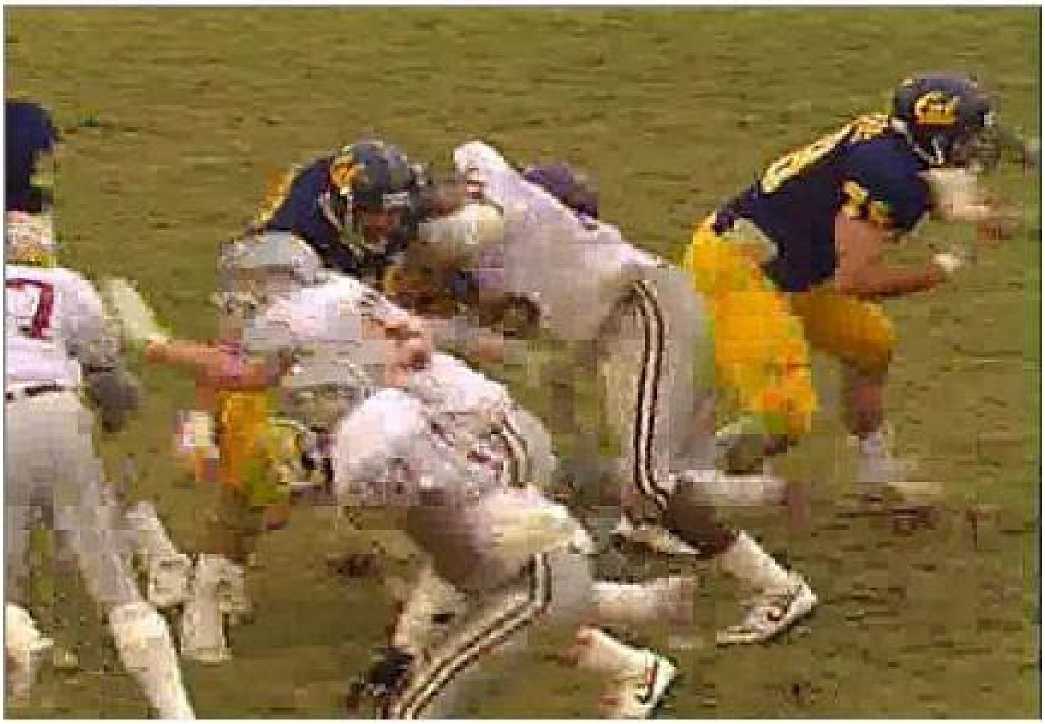
3.3. Video Compression Technologies:

- A standard video signal using MPEG–2 encoding uses about 3.75 Mbps of bandwidth over an IP network.
- A high definition signal may require 12–15 Mbps.
- So in order to deliver 2 channels of SD encoded TV to a home, almost 8 Mbps bandwidth is required. If xDSL is being used to access the home, it is easy to see why bandwidth is an issue.
- One way to alleviate bandwidth restrictions is to use new video compression technologies such as H.264/AVC which can offer up to a 50% reduction in bandwidth utilization for the same picture quality compared to existing MPEG– 2 compression.

Examples to illustrate the impact of the coding bitrate on video quality (1):

Still frame from MPEG–2 video sequence encoded at 5000Kbps (610KB/s)

Examples to illustrate the impact of the coding bitrate on video quality (2):



Still frame from MPEG-2 video sequence encoded at 1127 Kbps (138KB/s)

4. Spatial & Temporal Compression:

- Compression is needed to send video over the Internet. (due to the file size)
- Redundancy
    - Spatial Compression (within image): compression of each frame is done with JPEG (or a modification of it). Each frame is a picture that can be independently compressed.
    - Temporal Compression (from one image to next): redundant frames are removed. When we watch television, we receive 50 frames per second. However, most of the consecutive frames are almost the same. For example, when someone is talking, most of the frame is the same as the previous one except for the segment of the frame around the lips, which changes from one frame to another.

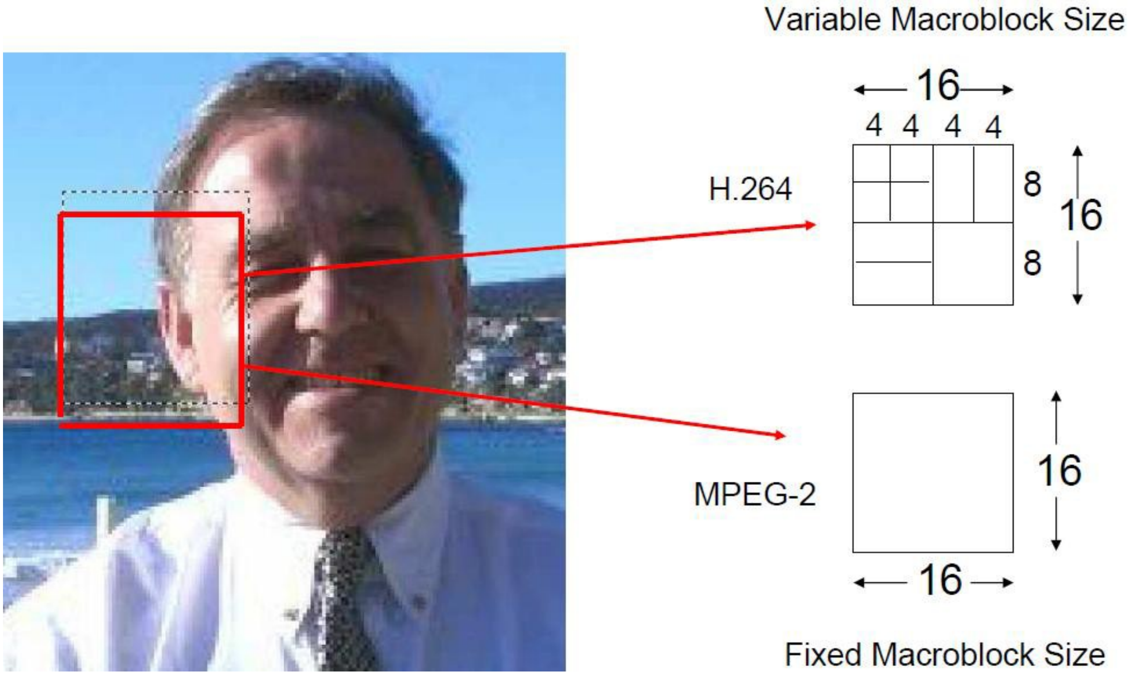
5. Intra–coding vs. Inter–coding:

- **Intra–coding:** Compression that works entirely within one picture; also known as spatial coding.
- **Inter–coding:** Compression that uses redundancy between successive pictures; also known as temporal coding.

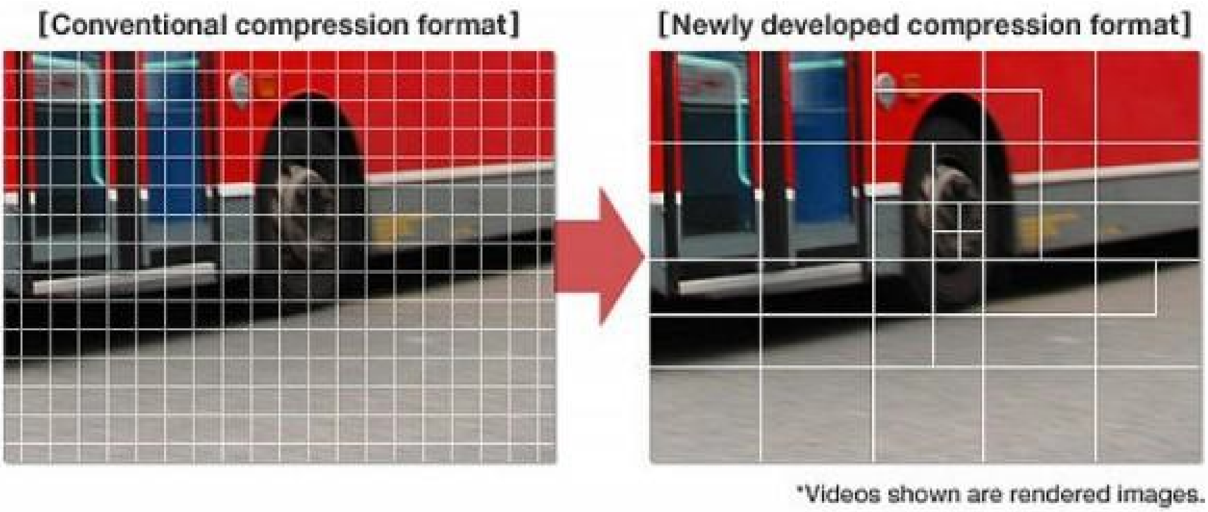
Examples:

Example1. Intra–frame Compression MPEG–2 vs. MPEG–4 (H.264):

JPEG (or a modification of it)



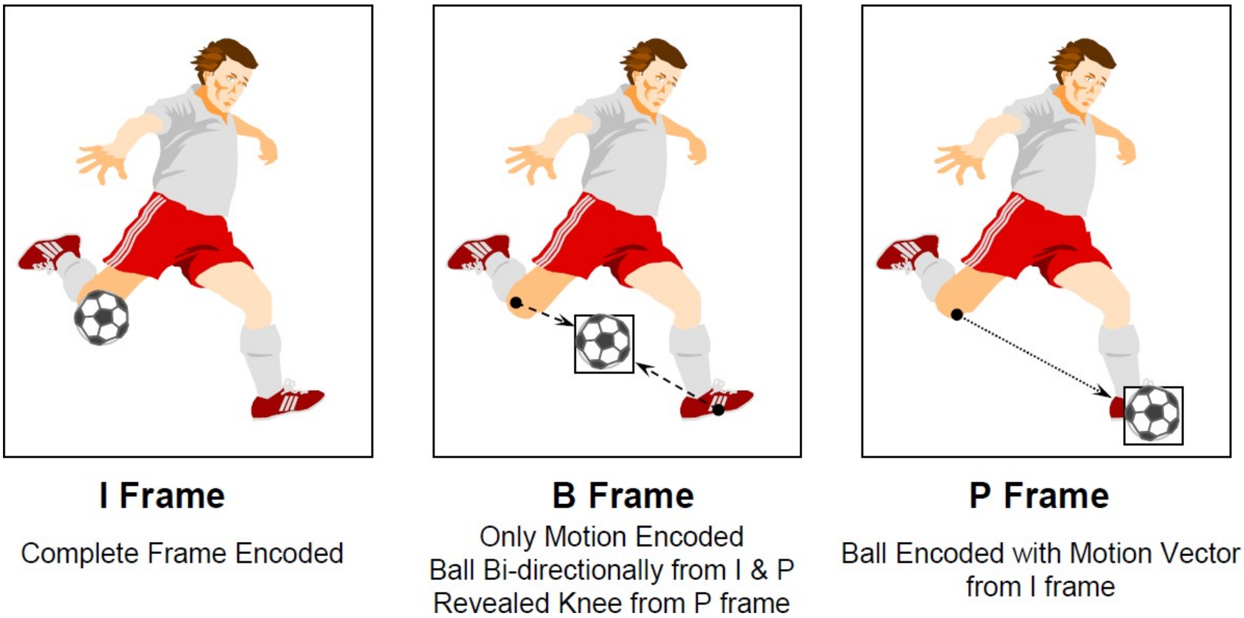
Example2. MPEG–2 vs. MPEG–4:



6. Temporal Compression:

To temporally compress data, the MPEG method first divides a set of frames into three categories: I–frames, P–frames, and B–frames.

Temporal Redundancy



frames:

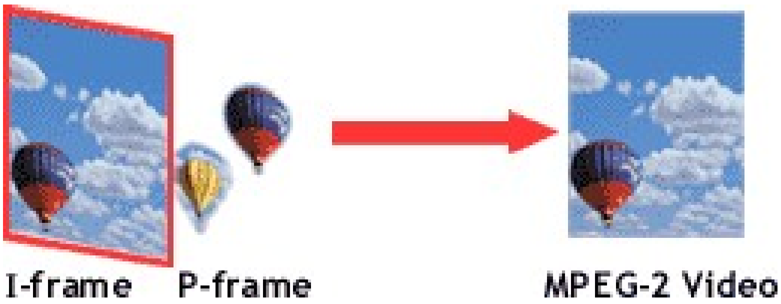
An intra–coded frame (I–frame) is an independent frame that is not related to any other frame. They are present at regular intervals.

- An I–frame must appear periodically to handle some sudden change in the frame that the previous and following frames cannot show.
- Also, when a video is broadcast, a viewer may tune in at any time. If there is only one I–frame at the beginning of the broadcast, the viewer who tunes in late will not receive a complete picture. I–frames are independent of other frames and cannot be constructed from other frames.

P–frames:

A predicted frame (P–frame) is related to the preceding I–frame or P–frame. In other words, each P–frame contains only the changes from the preceding frame. P–frames can be constructed only from previous I– or P–frames. P–frames carry much less information than other frame types and carry even fewer bits after compression.

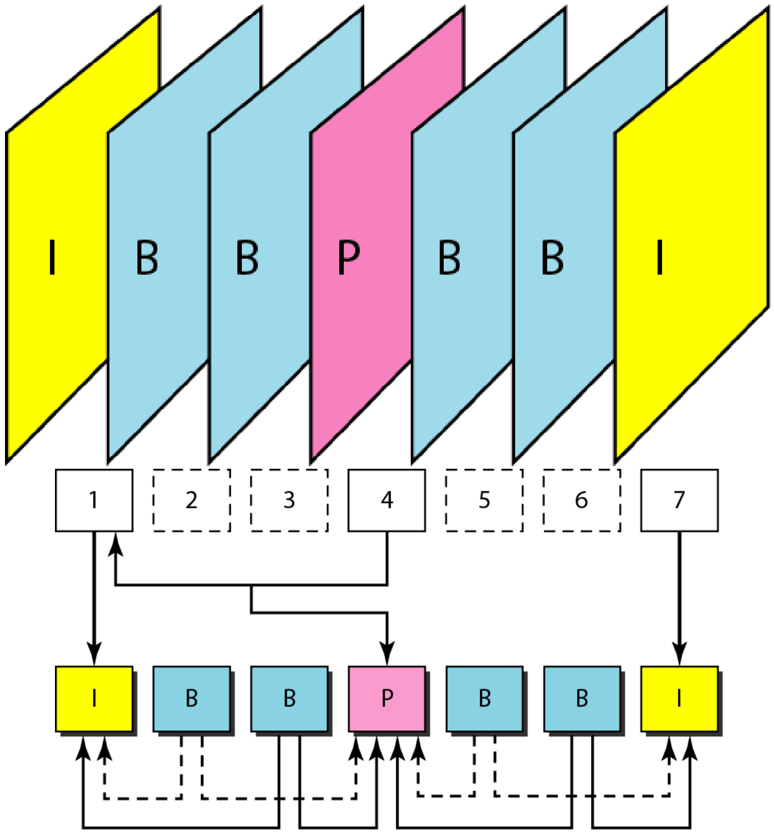
P–Frame only contains the differences compared to the original I–Frame. The source and the receiver run the same algorithm, the source receives the error and creates a P frame to fix it.



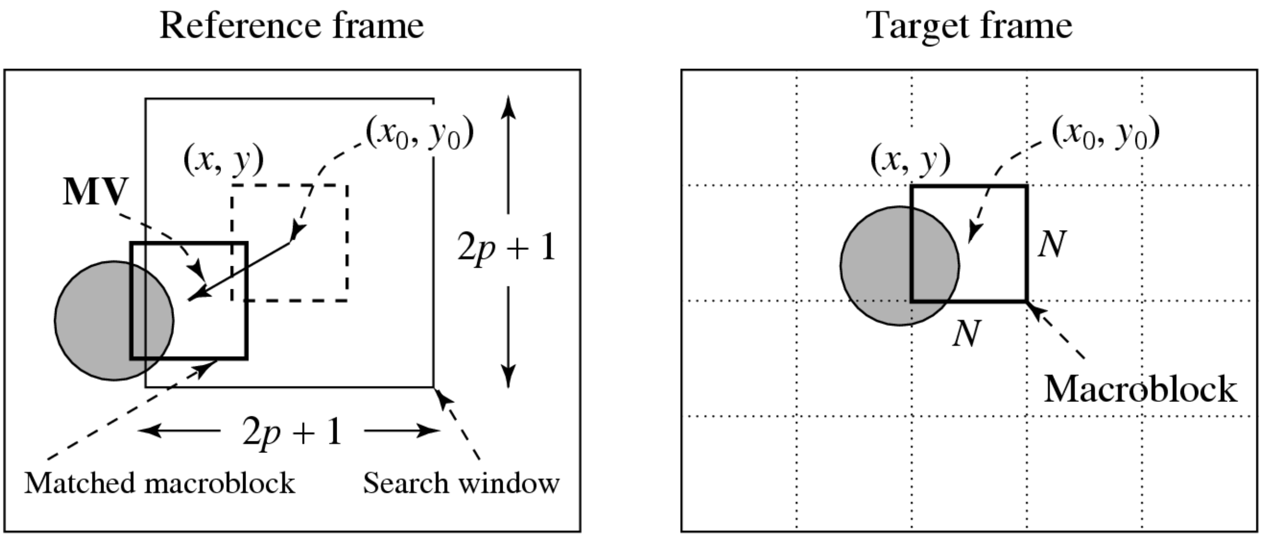
B–frames: A bidirectional frame (B–frame) is related to the preceding and following I–frame or P–frame. In other words, each B–frame is relative to the past and the future (forward and backward prediction).. Note that a B–frame is never related to another B–frame.

In theory, the number of B–frames that may occur between any two I– and P–frames is unlimited. In practice, there are typically up to twelve P– and B–frames occurring between each I–frame. One I–frame will occur approximately every 0.4 seconds during video showtime.

7. MPEG frames construction:



8. Search for Motion Vector:



Macroblocks and Motion Vector in Video Compression

MV search is usually limited to a small immediate neighborhood — both horizontal and vertical displacements in the range $[-p, p]$.

This makes a search window of size $(2p + 1) \times (2p + 1)$.

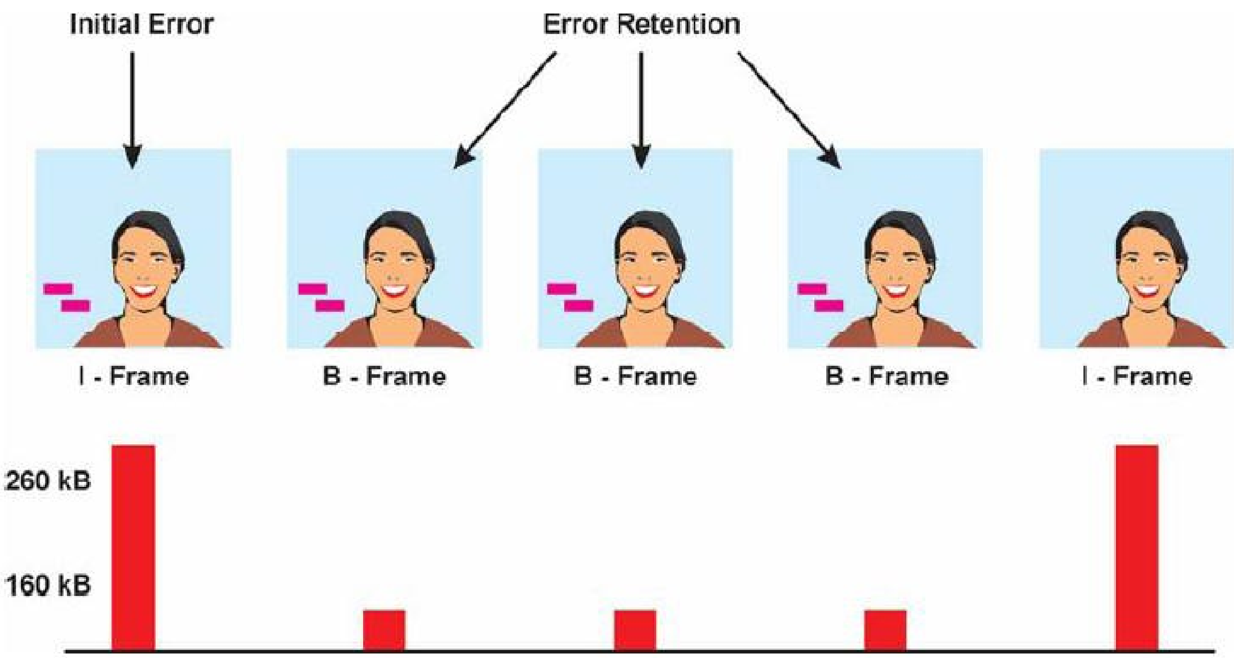
9. Packet Loss:

- Single B–frame IP packet loss (1 frame affected)
- Single I–frame IP packet loss(14 frames affected)

(a)

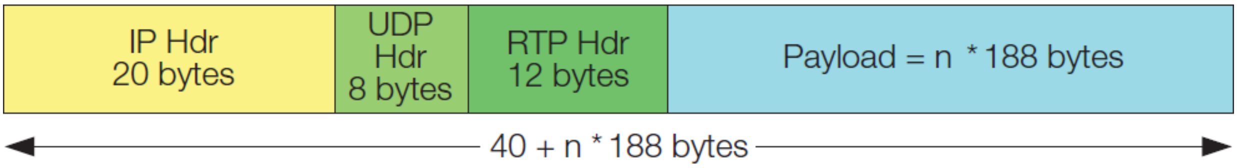
(b)

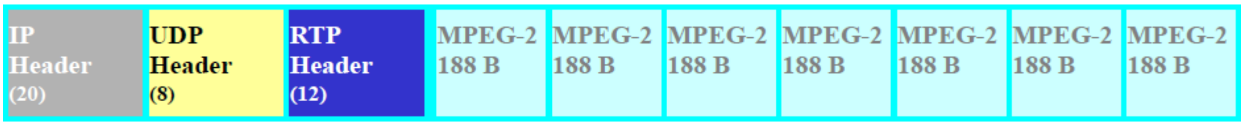
10. MPEG Error Retention:



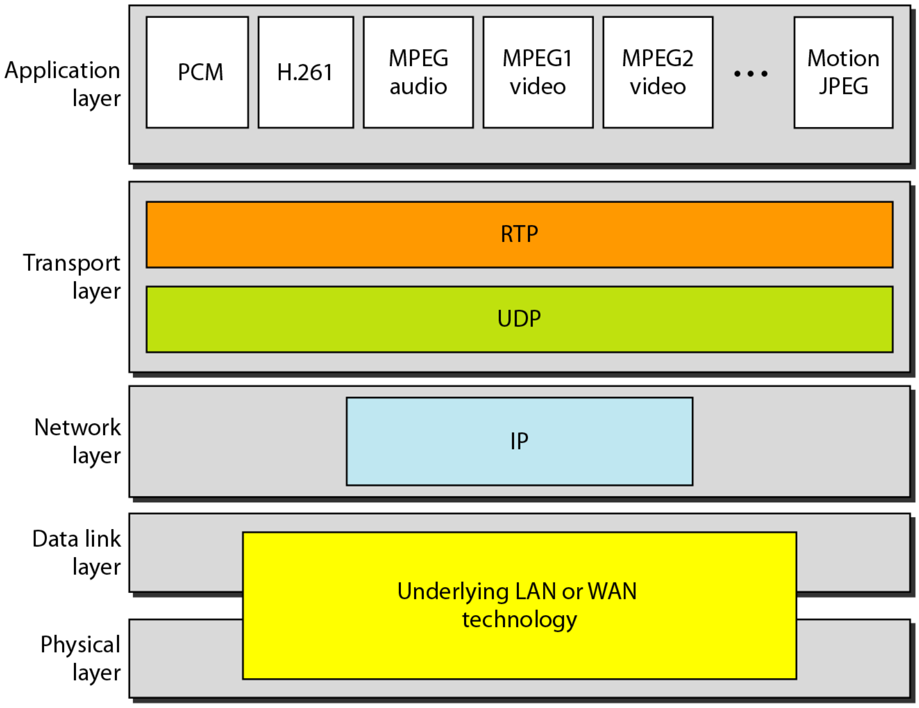
11. MPEG Transport Stream (TS) over UDP:

- I, B and P–frames are carried across the network in 188 byte MPEG Transport Stream (TS) packets which are encapsulated in IP packets.
- Typically 7 MPEG–TS packets in one UDP packet 7x188B = 1316B payloads

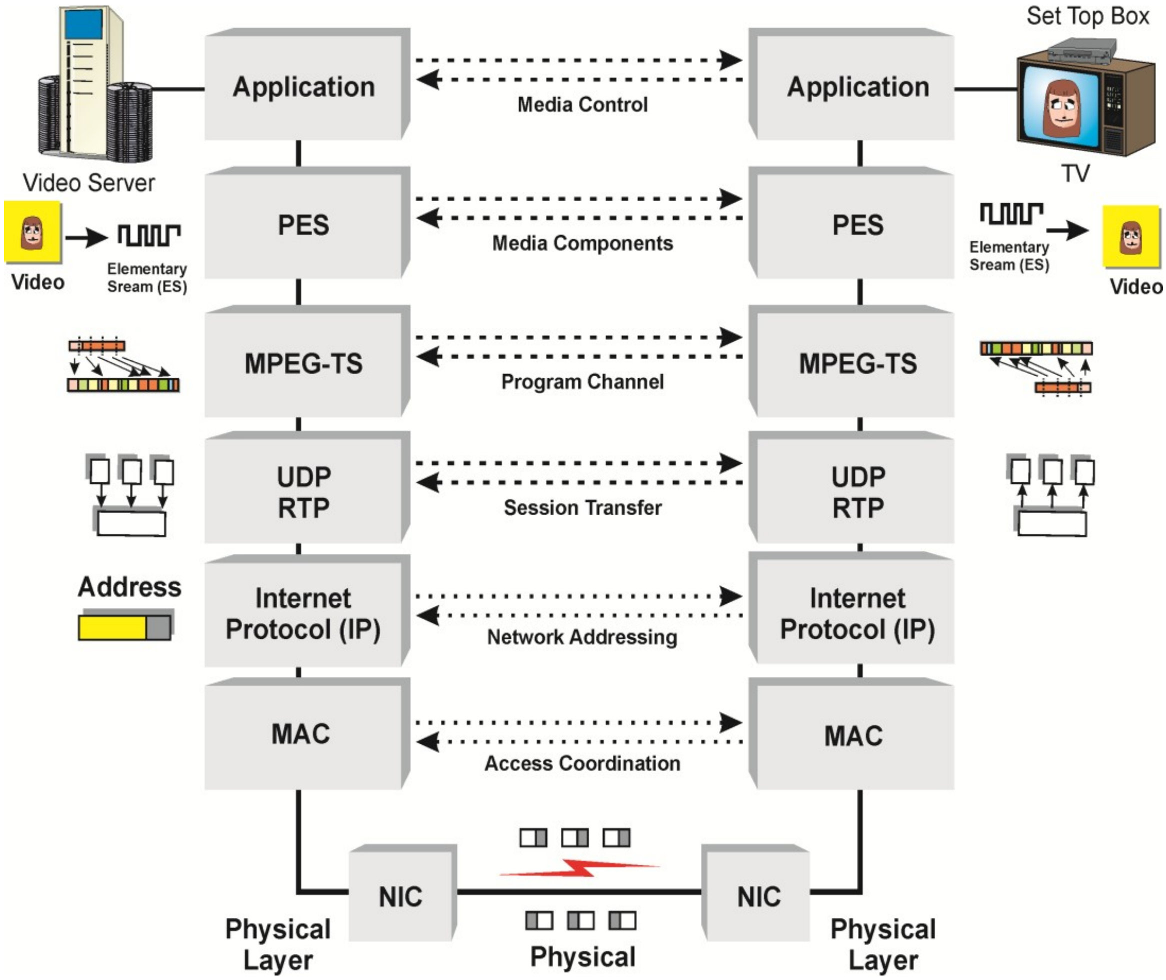




12. RTP over UDP:



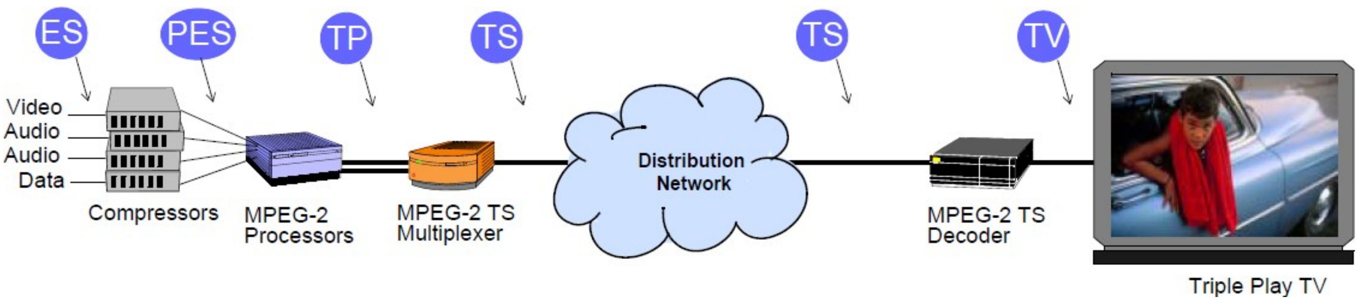
13. IPTV Protocol Stack Example:



IPTV Protocols:

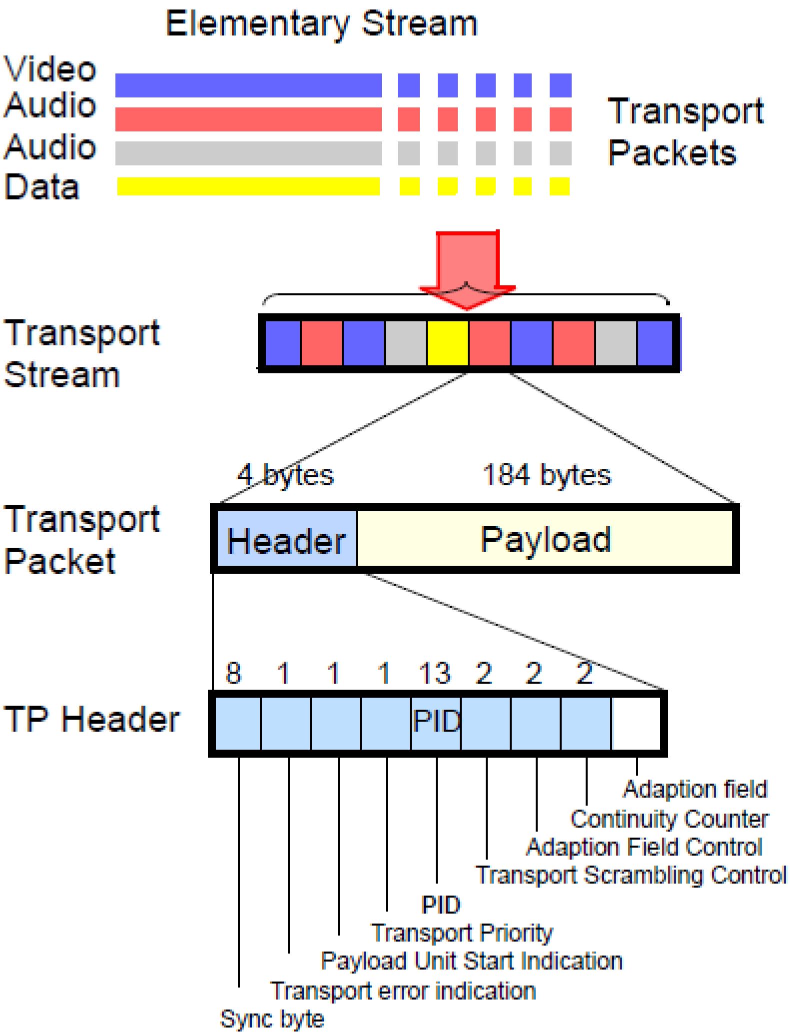
- The first 3 layers are typical for IPTV transmission.
- The UDP/RTP session layer is responsible for transferring packets between the sender and the receiver.
- The MPEG-TS transport stream layer combines multiple media streams (video, audio, data) into a single program transport stream.
- The PES layer assigns video and audio to specific packet streams.
- The application layer performs encoding and decoding of the video and audio using MPEG-2, MPEG-4, VC-1 or other formats.

14. MPEG2 Transport Stream Construction:

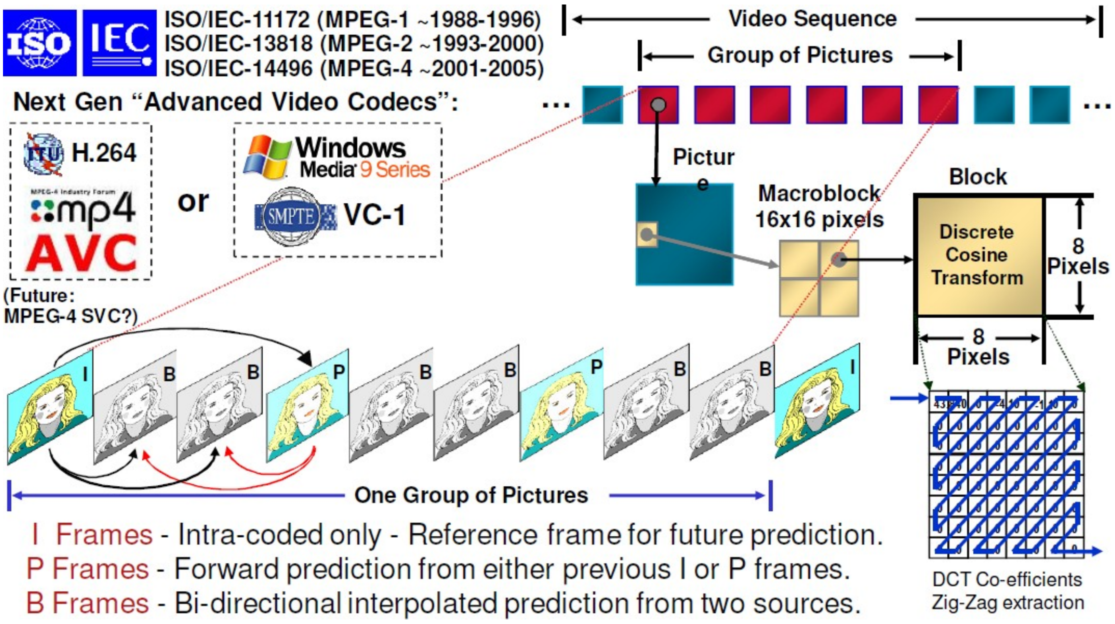


- The most basic component known as an **Elementary Stream (ES)**. A program contains a combination of elementary streams (typically video, one or more for audio, control data, subtitles, etc.)
- Each ES that is, output by MPEG audio, video and data encoders contains a single type of signal. There are various forms of ES, including: Digital Control Data, Digital Audio, Digital Video and Digital Data.
- The ES is sent to a processor that creates a stream called a Packetized Elementary Stream (PES). This is then broken into fixed–size Transport Packets (TP) and combined with the **Transport Stream (TS)**.
- The MPEG–2 TS is the transport format, but the contents are not necessarily MPEG–2.

15. MPEG2 Transport Stream:



16. MPEG Video Summary:



Chapter 8
Multimedia Streaming

Keywords:

Multimedia Streaming, Streaming stored audio/video, conversational voice/video-over-IP, streaming live audio/video

Abstract:

This chapter divides the multimedia in the Internet into three categories: streaming stored audio/video, streaming live audio/video, and real-time interactive audio/video. Then it describes the features and characteristics of different streaming modalities.

Learning Outcomes:

At the end of this lesson the student will:

- Understand the concept of video streaming in some details.
- Explore many of the underlying principles behind video streaming, including client buffering, and adapting video quality to available bandwidth.
- Be able to differentiate between different streaming approaches

Outline:

- Streaming Audio and Video
- Streaming stored audio/video
- Conversational voice/video-over-IP
- Streaming Approaches
- DASH: Dynamic, Adaptive Streaming over HTTP

Reference for this chapter:

Computer Networking: A Top-Down Approach / James F. Kurose, Keith W. Ross.— 6th ed. Pearson 2013

1. Streaming Audio and Video:

- New age of Internet TV
- Netflix and Hulu in North America and Youku and Kankan in China have practically become household names.
- People are becoming Internet video producers as well as consumers (youtube).
- Network applications such as Skype, Google Talk allow people to not only make “telephone calls” over the Internet, but to also enhance those calls with video and multi-person conferencing.

People in all corners of the world are currently using the Internet to watch movies and television shows on demand. Internet movie and television distribution companies such as Netflix and Hulu in North America and Youku and Kankan in China have practically become household names. But people are not only watching Internet videos, they are using sites like YouTube to upload and distribute their own user-generated content, becoming Internet video producers as well as consumers. Moreover, network applications such as Skype, Google Talk allow people to not only make “telephone calls” over the Internet, but to also enhance those calls with video and multi-person conferencing. In fact, we can safely predict that by the end of the current decade almost all video distribution and voice conversations will take place end-to-end over the Internet, often to wireless devices connected to the Internet.

2. Types of Multimedia Network Applications:

Internet supports a large variety of useful and entertaining multimedia applications. We classify multimedia applications into three broad categories:

- streaming stored audio/video
- conversational voice/video–over–IP
- streaming live audio/video

Streaming Stored Audio and Video

Streaming stored audio is very similar to streaming stored video, although the bit rates are typically much lower.

In this type of streaming, the underlying medium is prerecorded video, such as a movie, a television show, a prerecorded sporting event, or a prerecorded user generated video (such as on YouTube). These prerecorded videos are placed on servers, and users send requests to the servers to view the videos on demand. Many Internet companies today provide streaming video, including YouTube (Google), Netflix, and Hulu. By some estimates, streaming stored video makes up over 50 percent of the downstream traffic in the Internet access networks today.

3. Multimedia key characteristics:

Multimedia has two key characteristics that must be well understood to deal with it successfully:

- Multimedia uses extremely high data rates.
- Multimedia requires real–time playback.

4. Real–Time Traffic Requirements:

- Low jitter
- Low latency
- Integrate non–real–time and real–time services
- Adapts to changing network / traffic conditions
- Good performance for large nets / connections
- Modest buffer requirements within the network
- High effective capacity utilization
- low overhead in header bits per packet

5. Fundamental characteristics:

- Delay sensitive
  - end–to–end delay
  - delay jitter
- **Loss tolerant:** infrequent losses cause minor glitches (cause of redundancy)
- antithesis of data, which are loss intolerant but delay tolerant.

	Multimedia	Data
Delay	sensitive	tolerant
Loss	tolerant	intolerant

6. Transport service requirements of common apps:

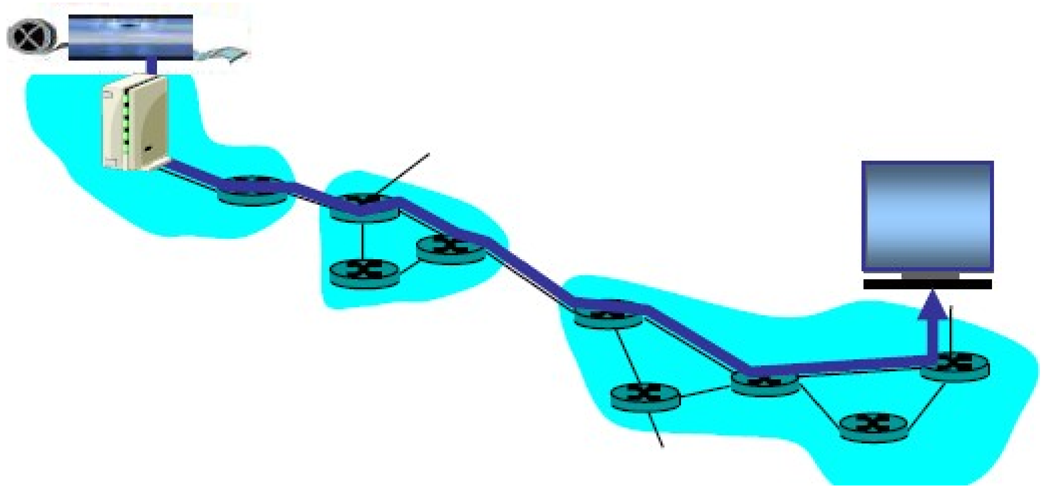
(Note: The text of this image is available in → /resource/RealtimeProtocols.ppt > slide31)

application	Data loss	Throughput	Time Sensitive
filter transfer	no loss	elastic	No
e-mail	no loss	elastic	No
web documents	no loss	elastic	No
real-time audio/video	loss-tolerant	audio: 5Kbps-1Mbps video: 10Kbps-5Mbps	Yes, 100's msec
stored audio/video	loss-tolerant	same as above	Yes, few secs
interactive games	loss-toleant	few kbps up	Yes, 100's msec
instant messaging	no loss	elastic	Yes and no

7. Streaming stored video: challenges

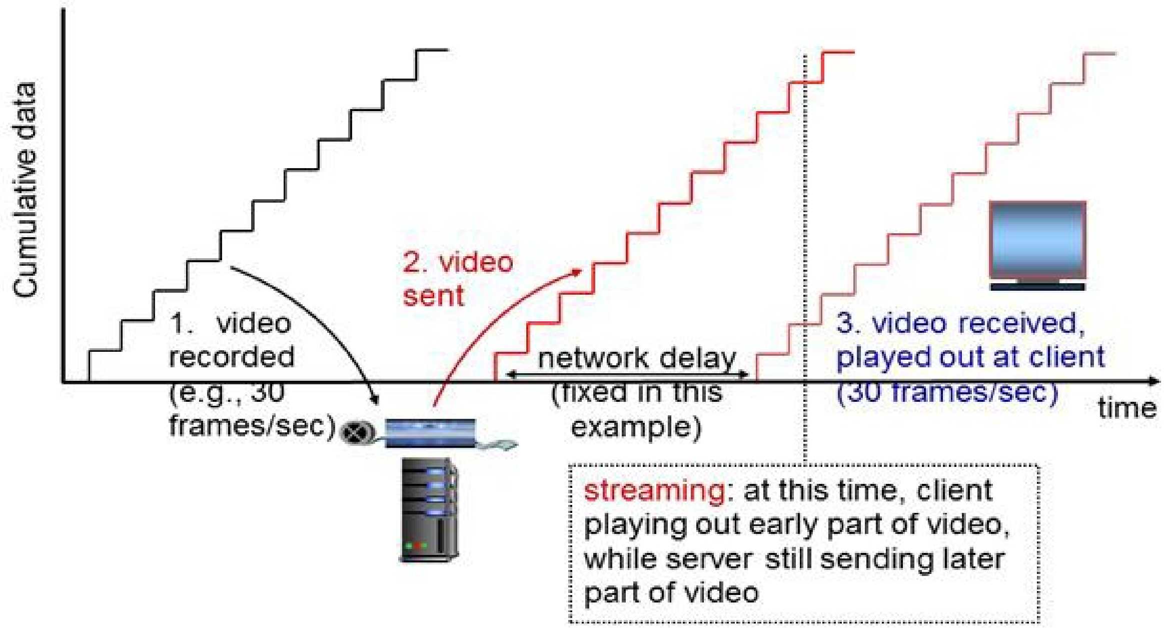
- continuous playout constraint: once client playout begins, playback must match original timing
  - but network delays are variable (jitter), so will need client-side buffer to match playout requirements
- other challenges:
  - client interactivity: pause, fast-forward, rewind, jump through video
  - video packets may be lost, retransmitted

8. Streaming Stored Multimedia:



1. media stored at source
2. transmitted to client
3. **streaming: client playout begins *before* all data has arrived Streaming Stored Multimedia:**

What is it?



This animation figure illustrate the concept of streaming

9. Streaming *Live* Multimedia:

Examples:

- Internet radio talk show
- live sporting event

Streaming (as with streaming *stored* multimedia)

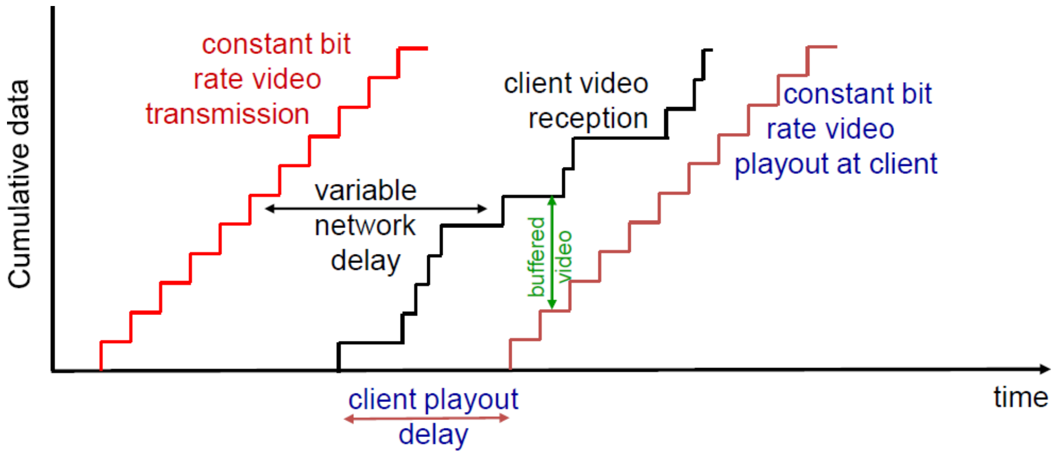
- playback buffer
- playback can lag tens of seconds after transmission
- still have timing constraint

Interactivity

- fast forward impossible!
- rewind, pause possible

10. Real–Time Interactive Multimedia:

- applications: IP telephony, video conference, distributed interactive worlds
- end–end delay requirements: audio: < 150 msec good, < 400 msec OK
  - includes application–level (packetization) and network delays
  - higher delays noticeable, impair interactivity
- session initialization
  - how does callee advertise its IP address, port number, encoding algorithms?

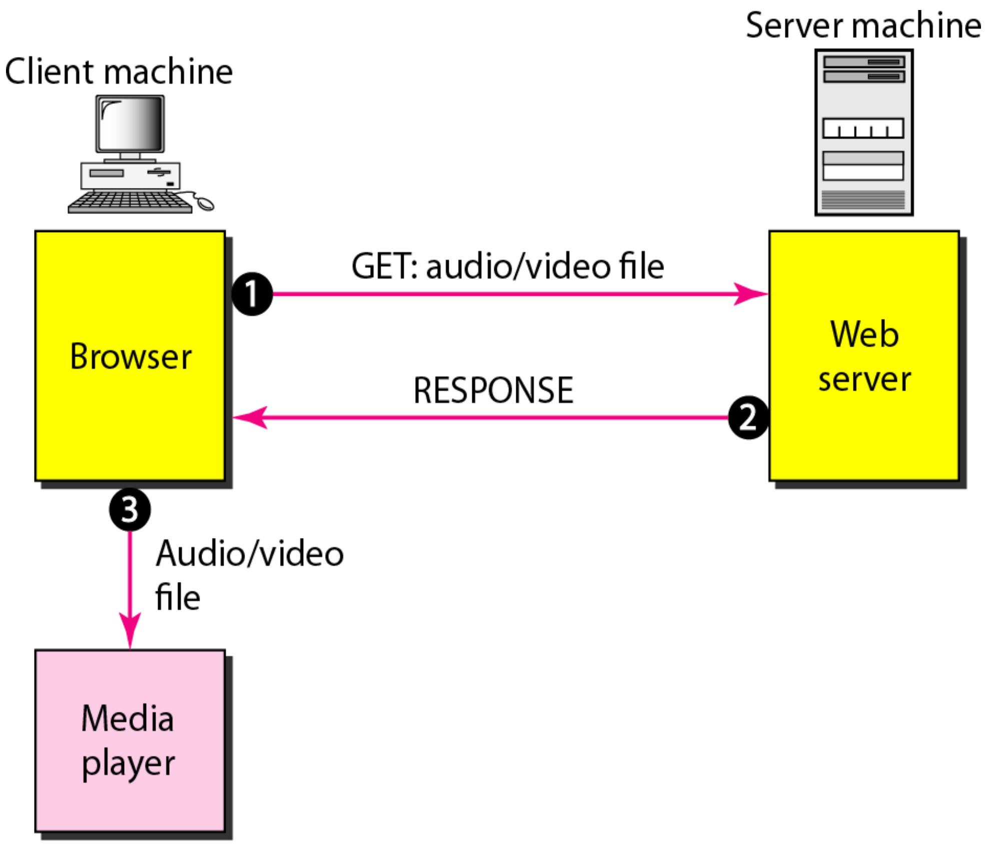


Client–side buffering and playout delay to compensate for network–added delay, delay jitter

11. Streaming Stored Multimedia Approaches:

- **First Approach:** Using a Web Server
- **Second Approach:** Using a Web Server with a Metafile
- **Third Approach:** Using a Media Server
- **Fourth Approach:** Using a Media Server and RTSP

12. First Approach: Using a Web Server:

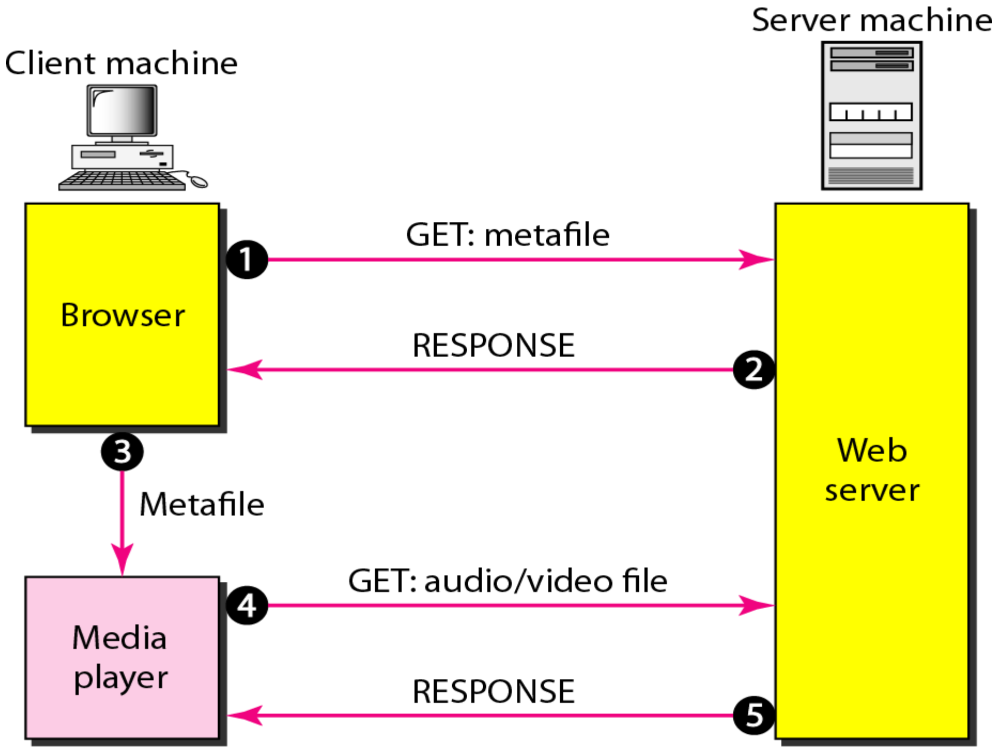


- Audio or video stored in file
- Files transferred as HTTP object
  - received in entirety at client then passed to player
- Audio, video not streamed: no, “pipelining,” long delays until playout!

The client (browser) can use the services of HTTP and send a GET message to download the file. The Web server can send the compressed file to the browser. The browser can then use a help application, normally called a media player, to play the file. The Figure in slide shows this approach.

This approach is very simple and does not involve streaming. However, it has a drawback. An audio/video file is usually large even after compression. An audio file may contain tens of megabits, and a video file may contain hundreds of megabits. In this approach, the file needs to download completely before it can be played.

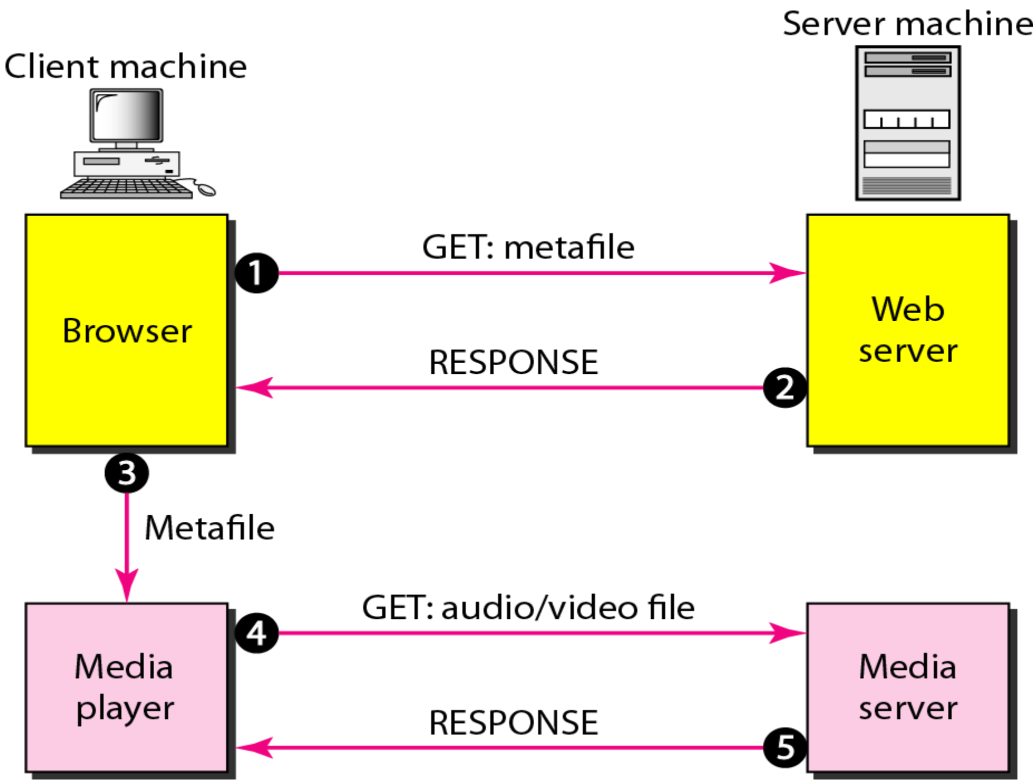
13. Second Approach: Using a Web Server with a Metafile:



1. The HTTP client accesses the Web server using the GET message.
2. The information about the metafile comes in the response.
3. The metafile is passed to the media player. (browser launches player, passing metafile)
4. The media player uses the URL in the metafile to access the audio/video file.
5. The Web server responds. (server streams audio/video to player)

In this approach, the media player is directly connected to the Web server for downloading the audio/video file. The Web server stores two files: the actual audio/video file and a metafile that holds information about the audio/video file. Figure shows the steps in this approach.

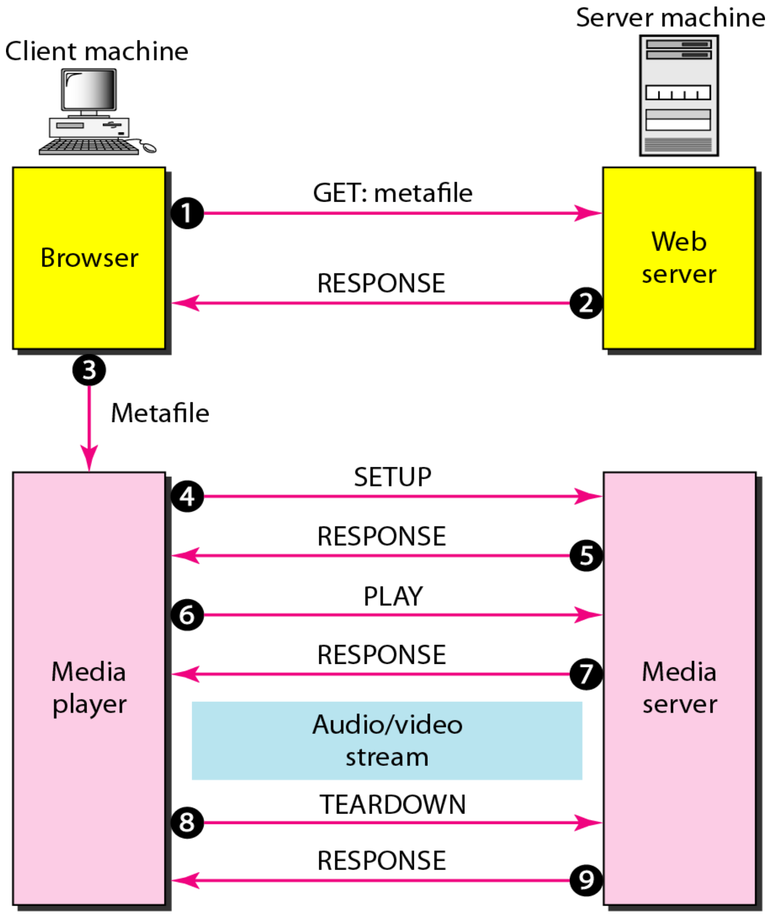
14. Third Approach: Streaming from a streaming server:



1. The HTTP client accesses the Web server using a GET message.
2. The information about the metafile comes in the response.
3. The metafile is passed to the media player.
4. The media player uses the URL in the metafile to access the media server to download the file. Downloading can take place by any protocol that uses UDP.
5. The media server responds. (UDP or TCP for this step)

The problem with the second approach is that the browser and the media player both use the services of HTTP. HTTP is designed to run over TCP. This is appropriate for retrieving the metafile, but not for retrieving the audio/video file. The reason is that TCP retransmits a lost or damaged segment, which is counter to the philosophy of streaming. We need to dismiss TCP and its error control; we need to use UDP. However, HTTP, which accesses the Web server, and the Web server itself are designed for TCP; we need another server, a media server. Figure and steps shows the concept.

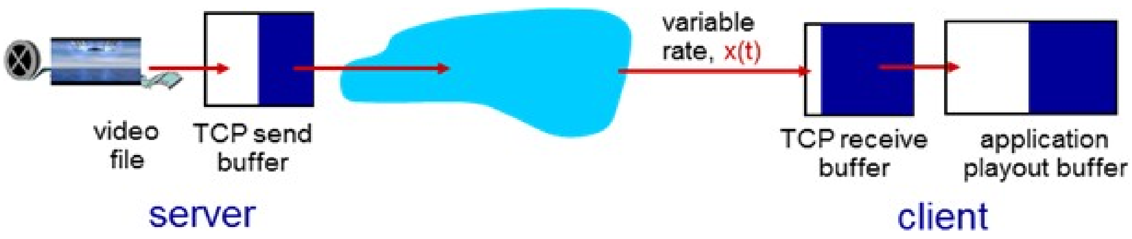
15. Using a media server and Real–Time Streaming Protocol (RTSP):



The Real–Time Streaming Protocol (RTSP) is a control protocol designed to add more functionalities to the streaming process. Using RTSP, we can control the playing of audio/video. RTSP is an out–of–band control protocol that is similar to the second connection in FTP. Figure shows a media server and RTSP.

User Control of Streaming Media: RTSP (explained in next chapter)

16. Streaming multimedia: HTTP



- multimedia file retrieved via HTTP GET
- send at maximum possible rate under TCP
- fill rate fluctuates due to TCP congestion control, retransmissions (in–order delivery)
- larger playout delay: smooth TCP delivery rate
- HTTP/TCP passes more easily through firewalls

17. DASH: Dynamic, Adaptive Streaming over HTTP:

1. server:
   - divides video file into multiple chunks
   - each chunk stored, encoded at different rates
   - manifest file: provides URLs for different chunks
2. client:
   - periodically measures server-to-client bandwidth
   - consulting manifest, requests one chunk at a time
     - chooses maximum coding rate sustainable given current bandwidth
     - can choose different coding rates at different points in time (depending on available bandwidth at time)
3. "intelligence" at client: client determines
   - **when** to request chunk (so that buffer starvation, or overflow does not occur)
   - **what** encoding rate to request (higher quality when more bandwidth available)
   - **where** to request chunk (can request from URL server that is "close" to client or has high available bandwidth)

Chapter 9
Streaming Protocols

Keywords:

RTSP, RTP, SIP

Abstract:

we consider three well known and used protocols for Real-time Transmission and control of multimedia streaming which designed to handle real-time traffic on the Internet. We give a simple demo to illustrate the basic concepts.

Learning Outcomes:

At the end of this lesson the student will have a strong foundation about Multimedia Streaming Protocols such as RTP, RTSP, SIP.

Outline:

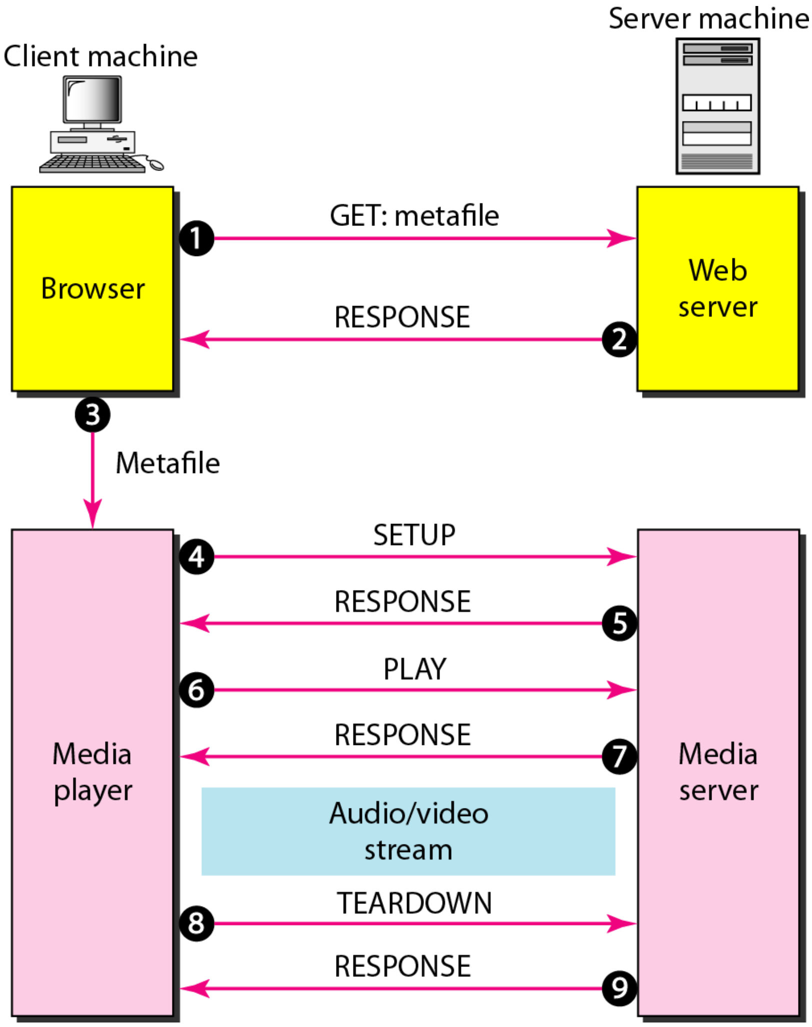
- RTSP
- RTP
- SIP

Reference:

Computer Networking: A Top-Down Approach / James F. Kurose, Keith W. Ross.— 6th ed. Pearson 2013

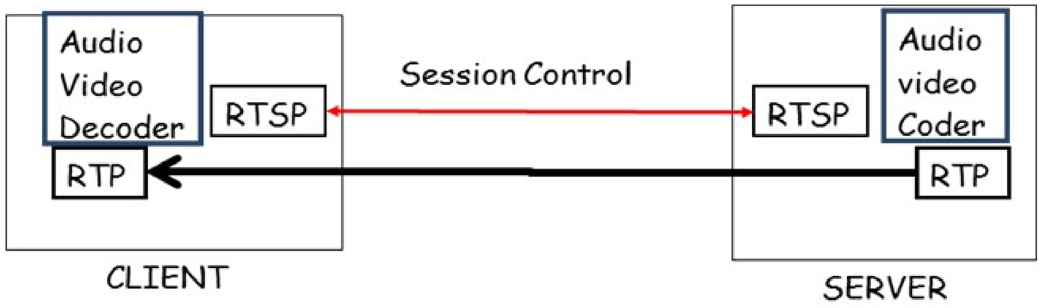
1. Using a media server and Real-Time Streaming Protocol (RTSP):

(Reminder from previous chapter):



1. The HTTP client accesses the Web server using a GET message.
2. The information about the metafile comes in the response.
3. The metafile is passed to the media player.
4. The media player sends a SETUP message to create a connection with the media server.
5. The media server responds.
6. The media player sends a PLAY message to start playing (downloading).
7. The audio/video file is downloaded using another protocol that runs over UDP.
8. The connection is broken using the TEARDOWN message.
9. The media server responds.

2. Real–Time Streaming Protocol (RTSP):



User Control of Streaming Media

2.1. RTSP: RFC 2326

- Application Protocol for Control of multimedia streams
- It is not a data transmission protocol, just remote control protocol between client and server
- User control: setup, play, pause, resume, repositioning, etc…

What it doesn't do:

- doesn't define how audio/video is encapsulated for streaming over network
- doesn't restrict how streamed media is transported (UDP or TCP possible)
- doesn't specify how media player buffers audio/video

Enables controlled, on–demand delivery of real–time data such as audio and video

2.2. RTSP: out of band control

RTSP messages sent out–of–band:

- RTSP control messages use different port numbers than media stream: out–of–band.
    - port 554
- media stream is considered "in–band".

Similar to FTP where the control info (directory changes, file deletion, rename) sent over separate TCP connection

2.3. RTSP Methods:

Request	Direction	Description
OPTIONS	S <-> C	Determine capabilities of server (S) or client (C)
DESCRIBE	C -> S	Get description of media stream
ANNOUNCE	S <-> C	Announce new session description
SETUP	C -> S	Create media session
RECORD	C -> S	Start media recording
PLAY	C -> S	Start media delivery
PAUSE	C -> S	Pause media delivery
REDIRECT	S -> C	Use other server
TEARDOWN	C -> S	Destroy media session

2.4. RTSP Example:

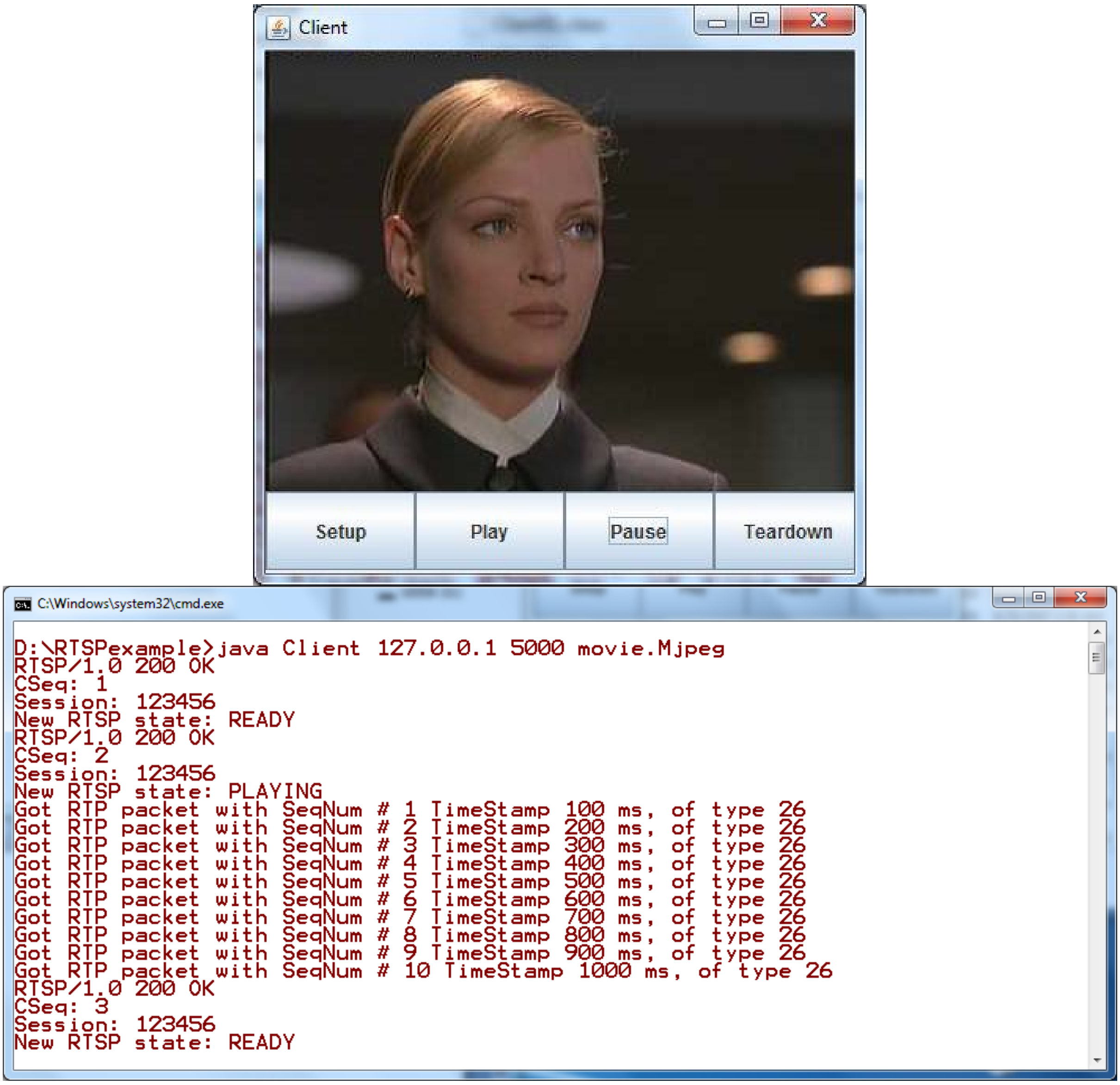
Scenario:

- metafile communicated to web browser
- browser launches player
- player sets up an RTSP control connection, data connection to streaming server

Requirements: Java Virtual Machine installed (normally Bundled with java JDK)

- Run the Example in folder /resources/RTSPexample:
  - start the server locally by double clicking on the sever1.bat
  - start a client by clicking the client1.bat
  - If you run the server on remote machine, then you have to change the script in file client.bat to point to the IP address of the server.
- Read the description of the assignment in the same folder file (StreamingVideoRTSP.htm) and complete the required tasks.

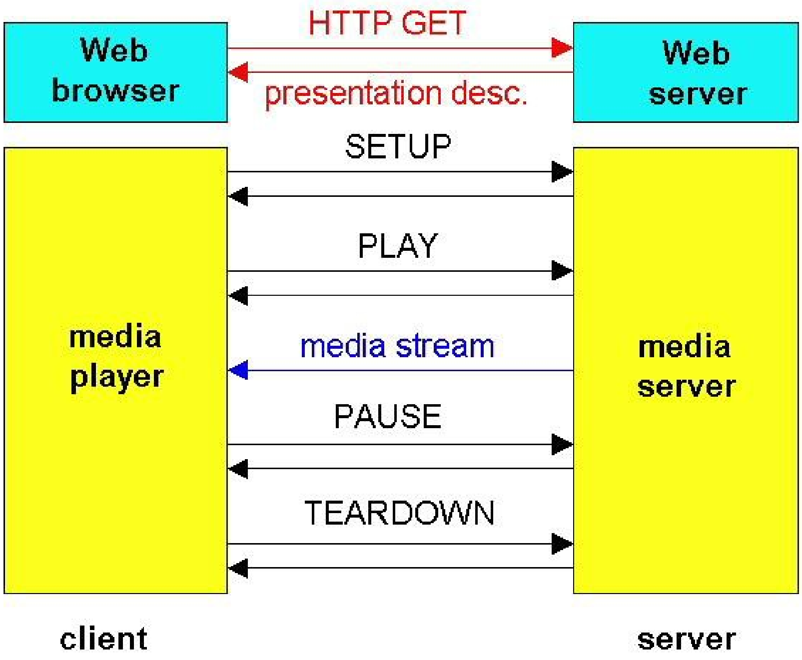
This assignment is taken from chapter 7 of the book: Computer Networking: A Top-Down Approach / James F. Kurose, Keith W. Ross.


Client
Setup
Play
Pause
Teardown
C:\Windows\system32\cmd.exe
D:\RTSPexample>java Client 127.0.0.1 5000 movie.Mjpeg
RTSP/1.0 200 OK
CSeq: 1
Session: 123456
New RTSP state: READY
RTSP/1.0 200 OK
CSeq: 2
Session: 123456
New RTSP state: PLAYING
Got RTP packet with SeqNum # 1 TimeStamp 100 ms, of type 26
Got RTP packet with SeqNum # 2 TimeStamp 200 ms, of type 26
Got RTP packet with SeqNum # 3 TimeStamp 300 ms, of type 26
Got RTP packet with SeqNum # 4 TimeStamp 400 ms, of type 26
Got RTP packet with SeqNum # 5 TimeStamp 500 ms, of type 26
Got RTP packet with SeqNum # 6 TimeStamp 600 ms, of type 26
Got RTP packet with SeqNum # 7 TimeStamp 700 ms, of type 26
Got RTP packet with SeqNum # 8 TimeStamp 800 ms, of type 26
Got RTP packet with SeqNum # 9 TimeStamp 900 ms, of type 26
Got RTP packet with SeqNum # 10 TimeStamp 1000 ms, of type 26
RTSP/1.0 200 OK
CSeq: 3
Session: 123456
New RTSP state: READY

3. Metafile Example:

<title>Twister</title>
<session>
<group language=en lipsync>
<switch>
<track type=audio
e="PCMU/8000/1"
src = "rtsp://audio.example.com/twister/audio.en/lofi">
<track type=audio
e="DVI4/16000/2" pt="90 DVI4/8000/1"
src="rtsp://audio.example.com/twister/audio.en/hifi">
</switch>
<track type="video/jpeg"
src="rtsp://video.example.com/twister/video">
</group>
</session>

4. RTSP Operation:



5. RTSP Exchange Example:

C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0

Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK

Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0

Cseq: 2

Session: 4231

S: RTSP/1.0 200 1 OK

Cseq: 2

Session 4231

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0

Cseq: 3

Session: 4231

S: RTSP/1.0 200 1 OK Session 4231

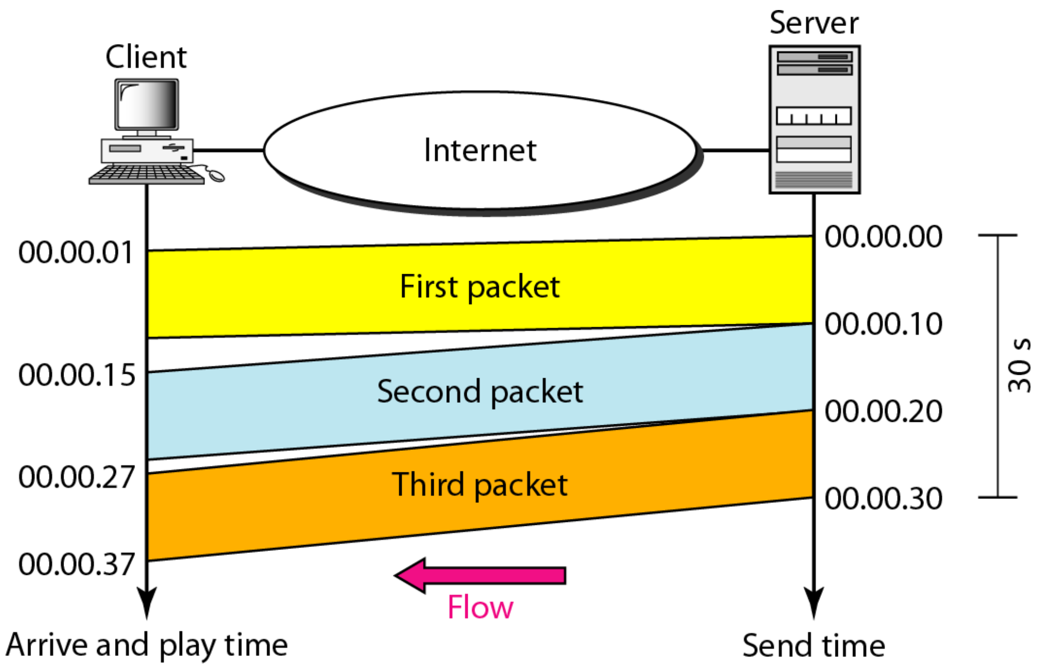
C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0

Session: 4231

6. Requirements for Real–Time Multimedia Traffic:

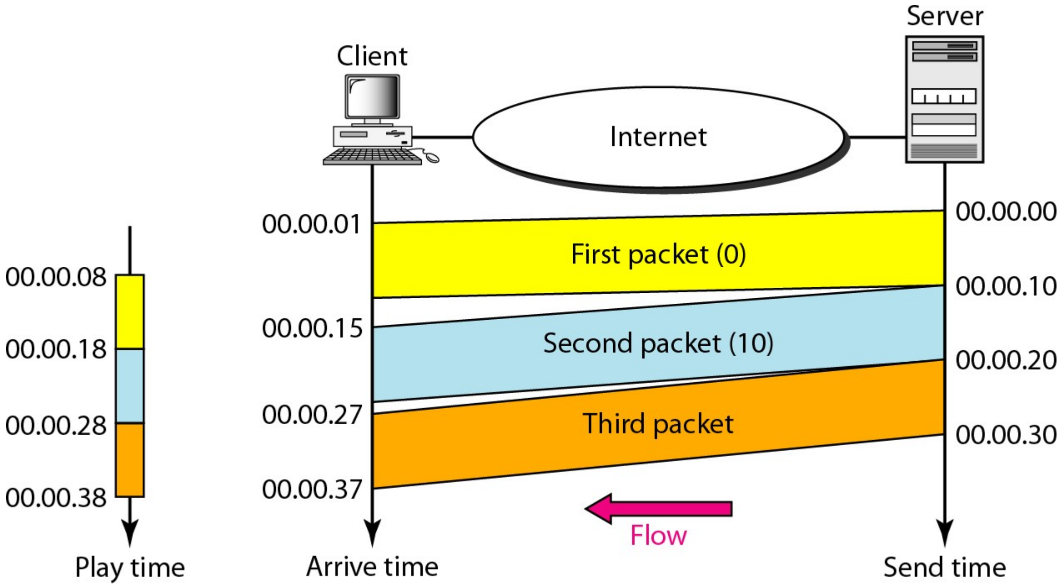
- In order to ensure playback timing and jitter removal timestamps are required.
- In order to ensure the presence and order of data a sequence number is required.
- Most of the real–time multimedia applications are video conferencing where several clients receive data. Therefore the multicast mode is preferred.
- In order to deal with congestion a mechanism for sender notification and change of encoding parameters must be provided.
- In order to display audio and video streams (and user data like titles) within a single A/V session mixers are required.
- In order to use high bit rate streams over a low–bandwidth network the translators are required (multimedia payload encoders and decoders

7. Jitter:



Jitter is introduced in real–time data by the delay between packets

8. Timestamp:



To prevent jitter, we can time–stamp the packets and separate the arrival time from the playback time

9. Translation and Mixing:

Translation means changing the encoding of a payload to a lower quality to match the bandwidth of the receiving network.

Mixing means combining several streams of traffic into one stream

RTP with UDP ?

10. Real–Time Transport Protocol (RTP):

- RTP specifies packet structure for packets carrying audio, video data
- RTP runs in end systems
- RTP packets encapsulated in UDP segments
- interoperability: if two VoIP applications run RTP, then they may be able to work together

10.1. Real-time Transport Protocol (RTP) Services:

- Real–time Transport Protocol (RTP) is the protocol designed to handle real–time traffic on the Internet.
- RTP does not have a delivery mechanism; it must be used with UDP. RTP stands between UDP and the application program.
- The main contributions of RTP are:
  - Payload Type Identification
  - Sequencing
  - Time–stamping
  - Mixing facilities

10.2. RTP Services:

The main contributions of RTP are:

- Payload Type Identification
  - Determination of media coding
  - Source identification
  - RTP works with Profiles: Profile defines a set of payload type codes and their mappings to payload formats
- Sequence numbering (packet sequence numbering)
  - Error detection
- Time–stamping
  - Time monitoring, synchronization, jitter calculation

10.3. Other Services: Support of Heterogeneity

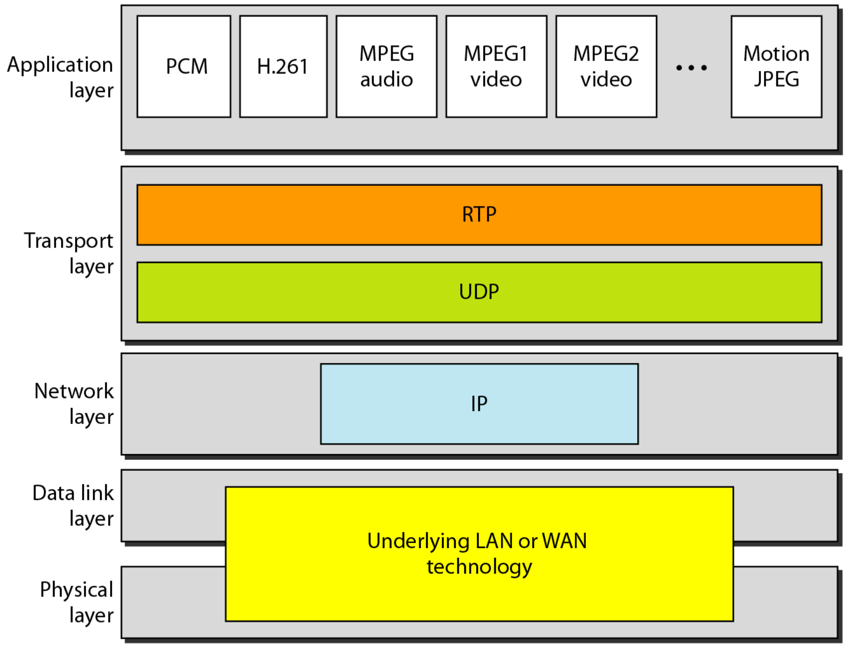
Mixer service:

- Allows for resynchronization of incoming audio packets
- Reconstructs constant 20 ms spacing generated by sender
- Mixes reconstructed audio streams into single stream
- Translates audio encoding to lower bandwidth
- Forwards lower bandwidth packet streams

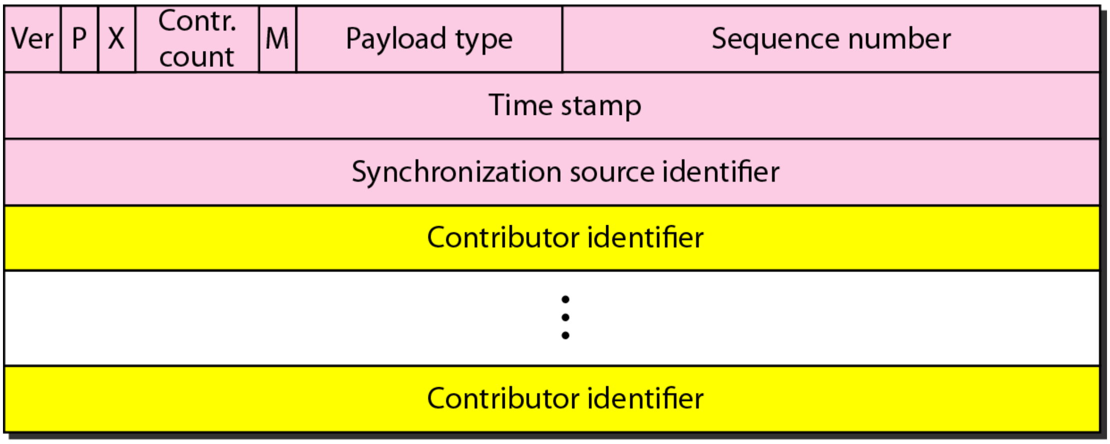
Translator service:

- Allows for translation between IP and other high speed protocols
- translators may change the encoding of data

11. RTP runs on top of UDP:



12. RTP packet header format:



Payload Type (7 bits): Indicates type of encoding currently being used. If sender changes encoding in middle of conference, sender informs receiver via payload type field.

- Payload type 0: PCM mu–law, 64 kbps
- Payload type 26, Motion JPEG
- Payload type 33, MPEG2 video

Type	*Application*	*Type*	*Application*	*Type*	*Application*
0	PCMμ Audio	7	LPC audio	15	G728 audio
1	1016	8	PCMA audio	26	Motion JPEG
2	G721 audio	9	G722 audio	31	H.261
3	GSM audio	10-11	L16 audio	32	MPEG1 video
5-6	DV14 audio	14	MPEG audio	33	MPEG2 video

Sequence Number (16 bits): Increments by one for each RTP packet sent, and may be used to detect packet loss and to restore packet sequence.

Timestamp field (32 bytes long): sampling instant of first byte in this RTP data packet

- for audio, timestamp clock typically increments by one for each sampling period (for example, each 125 usecs for 8 KHz sampling clock)

- if application generates chunks of 160 encoded samples, then timestamp increases by 160 for each RTP packet when source is active. Timestamp clock continues to increase at constant rate when source is inactive.

SSRC field (32 bits long): identifies source of t RTP stream. Each stream in RTP session should have distinct SSRC.

13. RTSP/RTP streaming example:

build a server that encapsulates stored video frames into RTP packets

- grab video frame, add RTP headers, create UDP segments, send segments to UDP socket
- include sequence numbers and timestamps in the RTP headers

Use RTSP in the client side to issue:

- setup / play / pause / … commands
- add new functionalities to the application

Rerun the example we used to illustrate the operation of RTSP protocol. Notice the streaming of the video file movie.MJPG which contains 500 frames, each frame is encapsulated in an RTP packet with the sequence number and a timestamp.

14. SIP: Session Initiation Protocol:

long-term vision:

- all telephone calls, video conference calls take place over Internet
- people identified by names or e-mail addresses, rather than by phone numbers
- can reach callee (if callee so desires), no matter where callee roams, no matter what IP device callee is currently using

14.1. SIP services:

SIP provides mechanisms for call setup:

- for caller to let callee know she wants to establish a call
- so caller, callee can agree on media type, encoding
- to end call

Determine current IP address of callee:

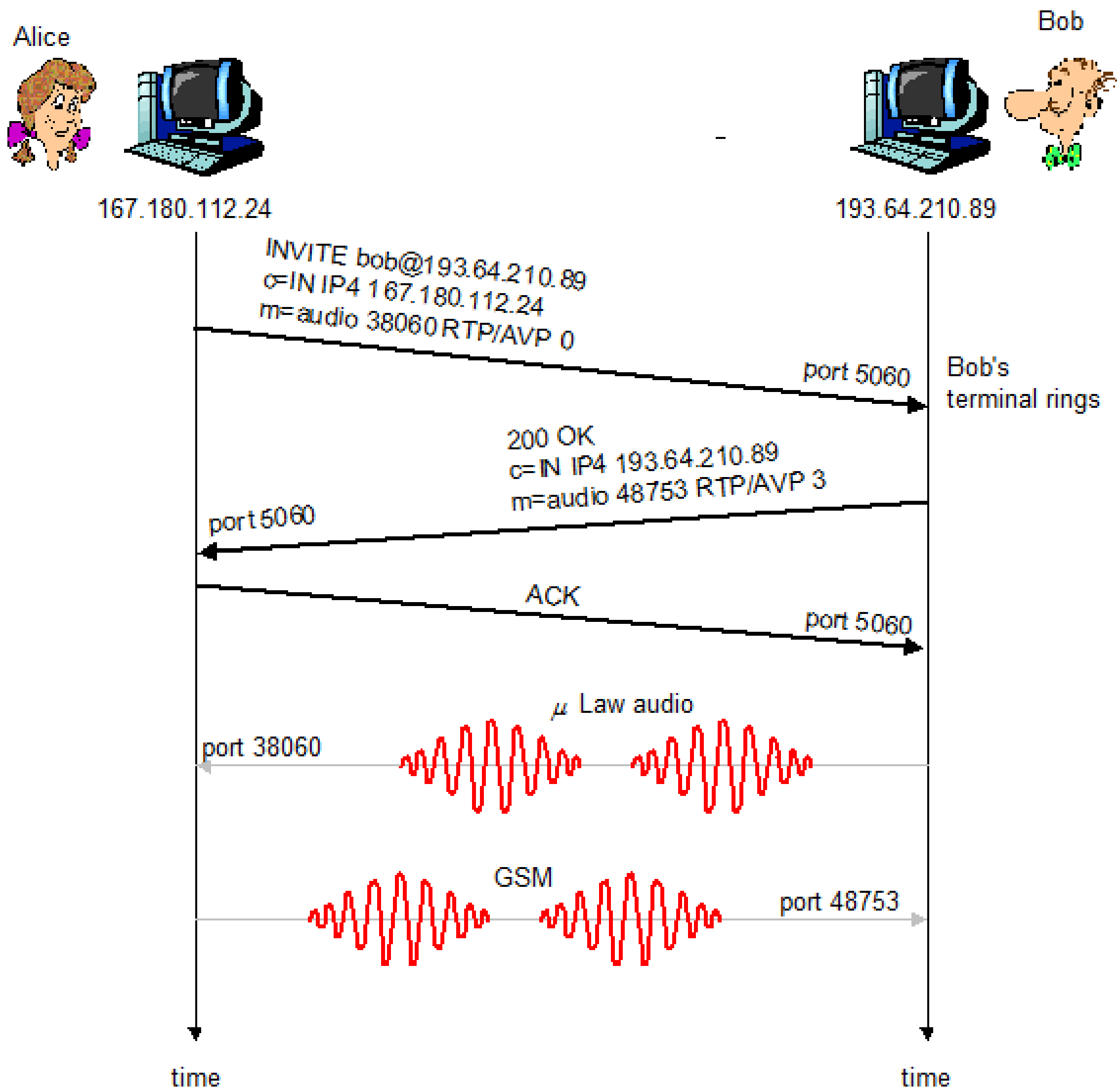
- maps mnemonic identifier to current IP address

Call management:

- add new media streams during call
- change encoding during call
- invite others
- transfer, hold calls

14.2. Ex. setting up call to known IP address:

- Alice's SIP invite message indicates her port number, IP address, encoding she prefers to receive (PCM μlaw)
- Bob's 200 OK message indicates his port number, IP address, preferred encoding (GSM)
- SIP messages can be sent over TCP or UDP; here sent over RTP/UDP
- default SIP port number is 5060



14.3. Setting up a call (more):

- codec negotiation:
  - suppose Bob doesn't have PCM µlaw encoder
  - Bob will instead reply with 606 Not Acceptable Reply, listing his encoders. Alice can then send new INVITE message, advertising different encoder
- rejecting a call
  - Bob can reject with replies "busy," "gone," "payment required," "forbidden"
- media can be sent over RTP or some other protocol

14.4. Example of SIP message:

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885
c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notes:

- HTTP message syntax
- sdp = session description protocol
- Call–ID is unique for every call

14.5. Name translation, user location:

- caller wants to call callee, but only has callee's name or e–mail address.
- need to get IP address of callee's current host:
  - user moves around
  - DHCP protocol
  - user has different IP devices (PC, smartphone, car device)
- result can be based on:
  - time of day (work, home)
  - caller (don't want boss to call you at home)
  - status of callee (calls sent to voicemail when callee is already talking to someone)

Chapter 10
Voice-over-IP

Keywords:

Voice–over–IP, Delay jitter, Adaptive Playout Delay, Recovery from packet, Skype.

Abstract:

Real–time conversational voice over the Internet is often referred to as Internet telephony. It is also commonly called Voice–over–IP (VoIP). In this section we describe the principles and protocols underlying VoIP.

Learning Outcomes:

At the end of this lesson the student will:

- Understand the VoIP characteristics: packet loss and delay, Delay jitter, etc …
- Know the different techniques used to handle jitter and the recovery from packet loss
- Be familiar with the architecture and the operation of Skype

Outline:

- VoIP characteristics
- VoIP: packet loss and delay
- Delay jitter
- Fixed playout delay & Adaptive Playout Delay
- VoIP: recovery from packet loss
- Voice–over–IP: Skype
- Example: Adaptive playout delay

Reference:

Computer Networking: A Top–Down Approach / James F. Kurose, Keith W. Ross.—6th ed. Pearson 2013

1. Voice-over-IP:

- VoIP end-end-delay requirement: needed to maintain "conversational" aspect
  - higher delays noticeable, impair interactivity
  - < 150 msec: good
  - > 400 msec bad
  - includes application-level (packetization,playout), network delays
- session initialization: how does callee advertise IP address, port number, encoding algorithms?
- value-added services: call forwarding, screening, recording

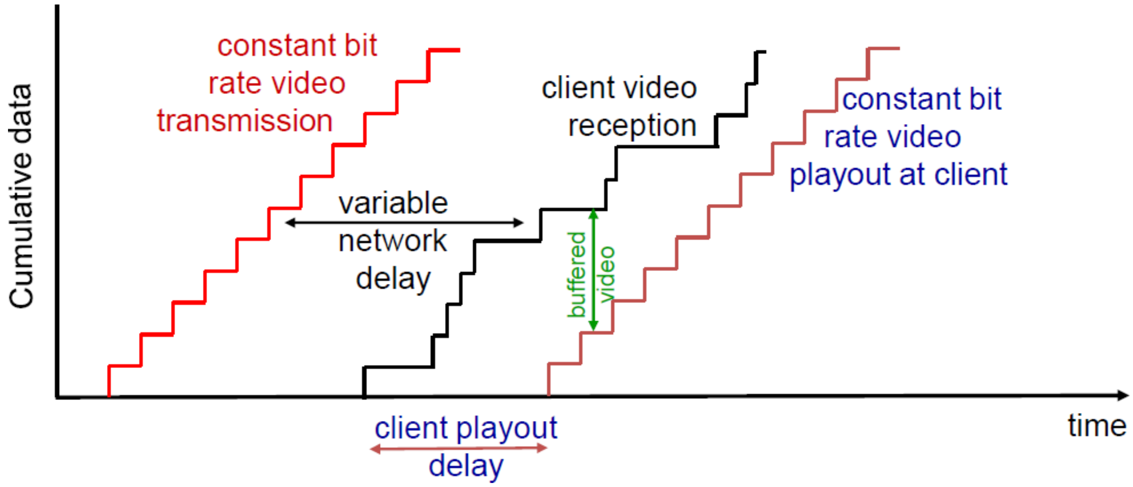
2. VoIP characteristics:

- speaker's audio: alternating talk spurts, silent periods.
  - 64 kbps during talk spurt
  - packets generated only during talk spurts
  - 20 msec chunks at 8 Kbytes/sec: 160 bytes of data
- application-layer header added to each chunk
- chunk+header encapsulated into UDP or TCP segment
- application sends segment into socket every 20 msec during talk spurt

3. VoIP: packet loss and delay:

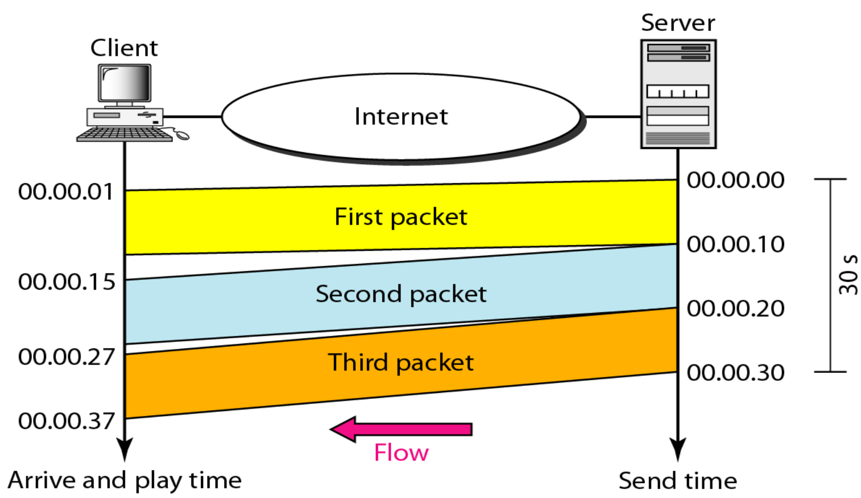
- network loss: IP datagram lost due to network congestion (router buffer overflow)
- delay loss: IP datagram arrives too late for playout at receiver
  - delays: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms
- loss tolerance: depending on voice encoding, loss concealment, packet loss rates between 1% and 10% can be tolerated

4. Delay 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 the Figure. This phenomenon is called jitter.

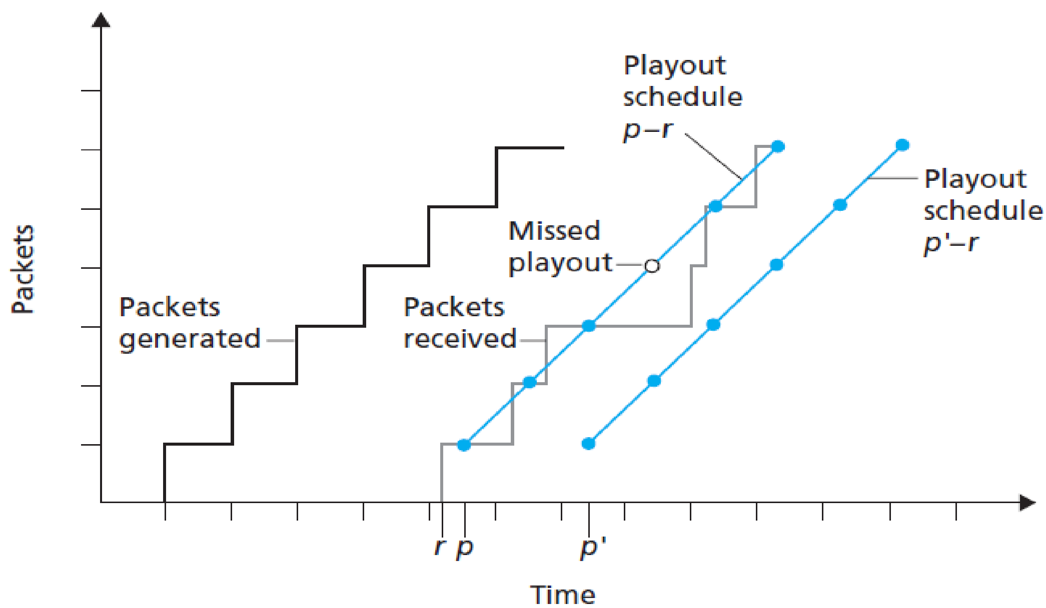
Client-side buffering and playout delay to compensate for network-added delay, delay jitter



end-to-end delays of two consecutive packets: difference can be more or less than 10 sec (transmission time difference)

5. VoIP: fixed playout delay:

- receiver attempts to playout each chunk exactly *q* msecs after chunk was generated.
  - chunk has time stamp *t:* play out chunk at *t+q*
  - chunk arrives after *t+q*: data arrives too late for playout: data "lost"
- tradeoff in choosing *q*:
  - ***large q:*** less packet loss
  - ***small q:*** better interactive experience



Sender generates packets every 20 msec during talk spurt. The first packet in this talk spurt is received at time r. As shown in the figure, the arrivals of subsequent packets are not evenly spaced due to the network jitter.

- first playout schedule: begins at p
- second playout schedule: begins at p'

6. Fixed playout delay → Adaptive Playout Delay:

- Previous example demonstrates an important delay–loss trade–off that arises when designing a playout strategy with fixed playout delays.
- By making the initial playout delay large, most packets will make their deadlines and there will therefore be negligible loss; however, for conversational services such as VoIP, long delays can become intolerable.

- To deal with this trade–off is to estimate the network delay and the variance of the network delay, and to adjust the playout delay accordingly.

The previous example demonstrates an important delay–loss trade–off that arises when designing a playout strategy with fixed playout delays.
By making the initial playout delay large, most packets will make their deadlines and there will therefore be negligible loss; however, for conversational services such as VoIP, long delays can become bothersome if not intolerable.
The natural way to deal with this trade–off is to estimate the network delay and the variance of the network delay, and to adjust the playout delay accordingly at the beginning of each talk spurt.

7. Adaptive playout delay:

- t_i = the timestamp of the *i* th packet = the time the packet was generated by the sender
- r_i = the time packet *i* is received by receiver
- p_i = the time packet *i* is played at receiver
- The end–to–end network delay of the *i*th packet is $r_i - t_i$. Due to network jitter, this delay will vary from packet to packet.
- Let d_i denote an estimate of the average network delay upon reception of the *i*th packet.

- goal: low playout delay, low late loss rate
- approach: adaptive playout delay adjustment:
  - estimate network delay, adjust playout delay at beginning of each talk spurt
  - silent periods compressed and elongated
  - chunks still played out every 20 msec during talk spurt
- adaptively estimate packet delay: (EWMA – Exponentially Weighted Moving Average):

$$\underbrace{d_i}_{\text{delay estimate after ith packet}} = (1-\underbrace{\alpha}_{\text{small constant, e.g. 0.1}})d_{i-1} + \alpha\,\underbrace{(\underbrace{r_i}_{\text{time received}} - \underbrace{t_i}_{\text{time sent (timestamp)}})}_{\text{measured delay of ith packet}}$$

- also useful to estimate average deviation of delay, vi :

- estimates d_i, v_i calculated for every received packet, but used only at start of talk spurt
- for first packet in talk spurt, playout time is:

remaining packets in talkspurt are played out periodically

8. VoIP: recovery from packet loss (1):

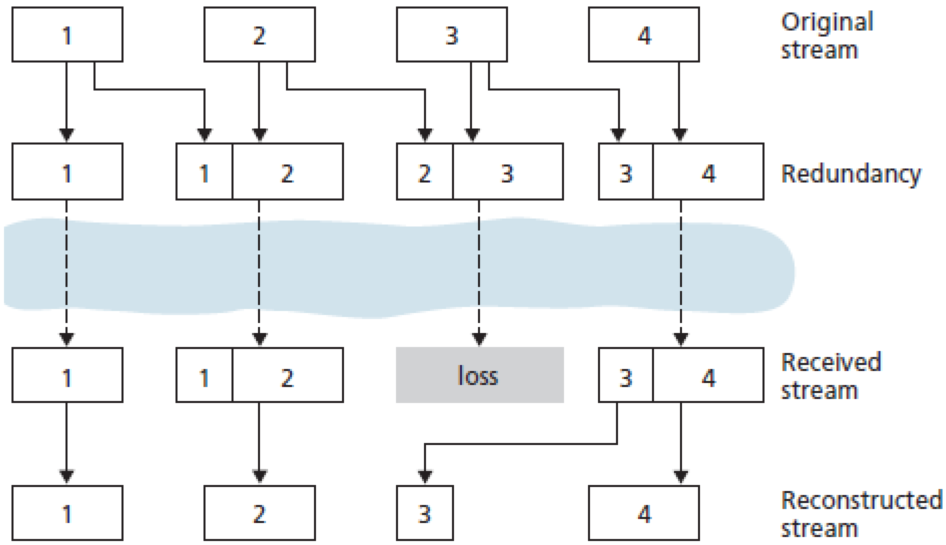
Challenge: recover from packet loss given small tolerable delay between original transmission and playout

- each ACK/NAK takes ~ one RTT
- alternative: Forward Error Correction (FEC)
  - send enough bits to allow recovery without retransmission (recall two-dimensional parity)

simple FEC

- for every group of n chunks, create redundant chunk by exclusive OR-ing n original chunks
- send n+1 chunks, increasing bandwidth by factor 1/n
- can reconstruct original n chunks if at most one lost chunk from n+1 chunks, with playout delay

9. VoIP: recovery from packet loss (2):

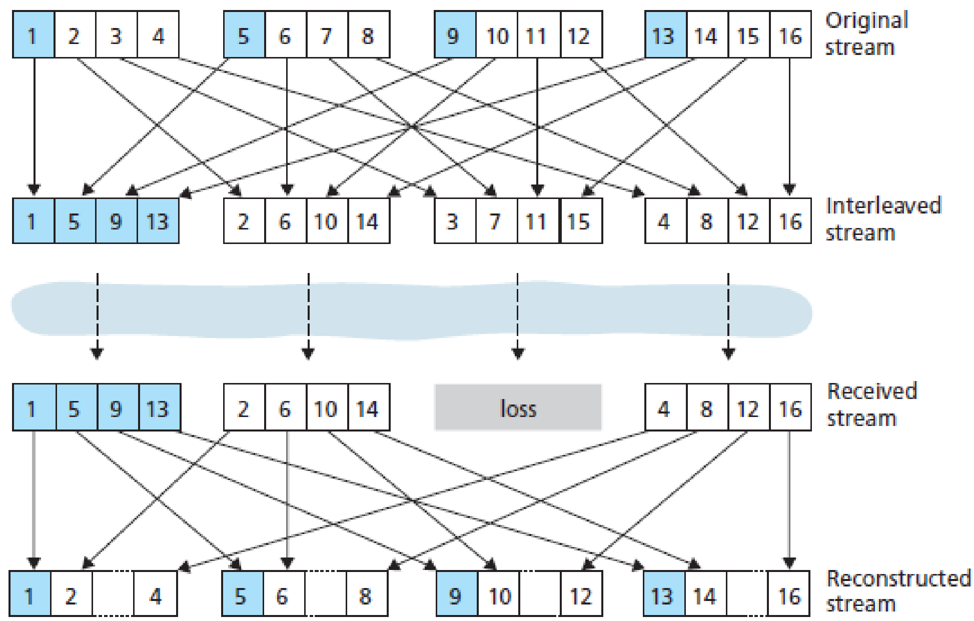


another FEC scheme:

- “piggyback lower quality stream”
- send lower resolution audio stream as redundant information

e.g., nominal stream PCM at 64 kbps and redundant stream GSM at 13 kbps

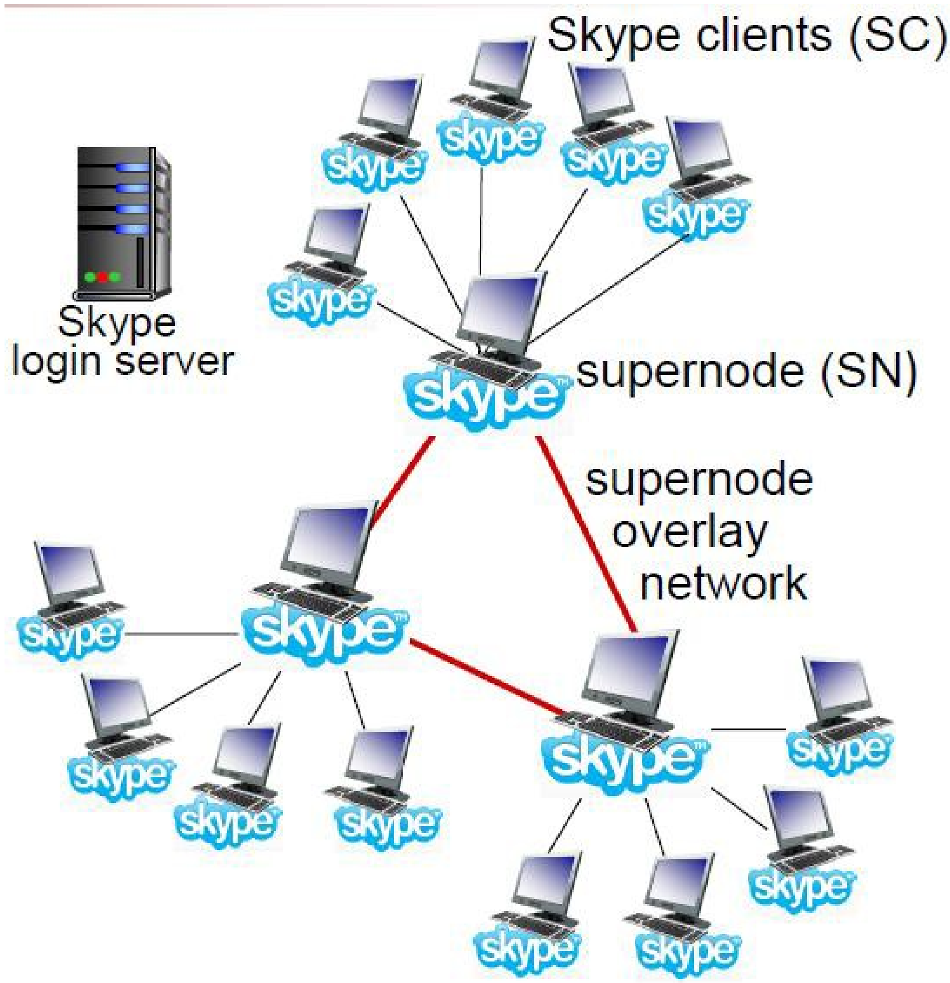
10. VoIP: recovery from packet loss (3):



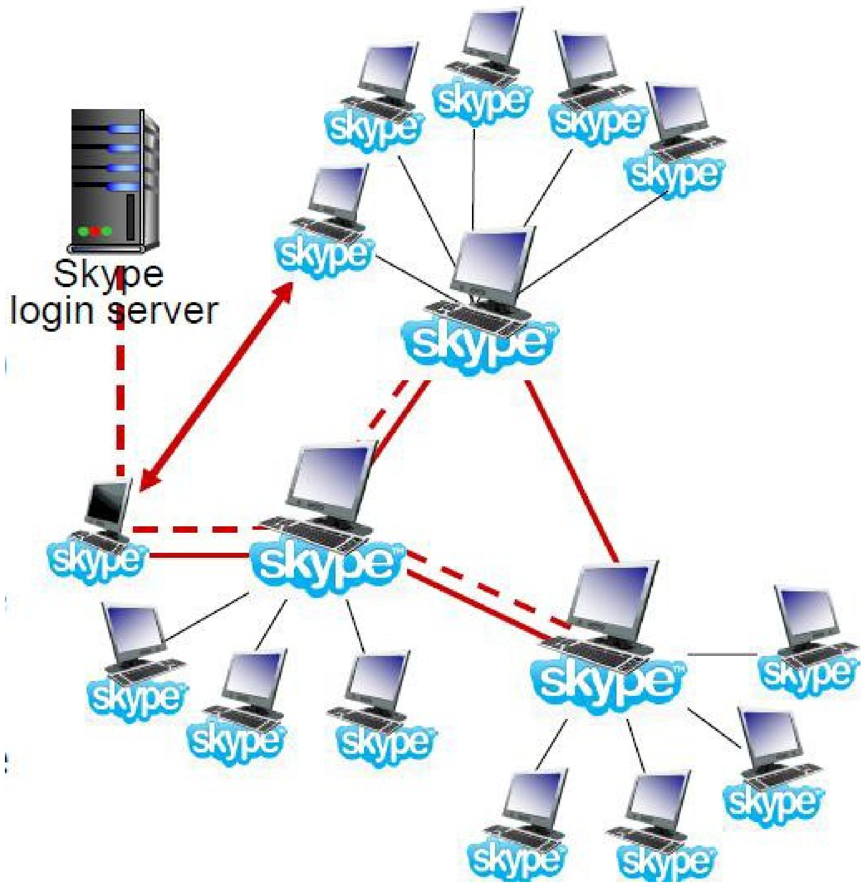
interleaving to conceal loss:

- audio chunks divided into smaller units, e.g. four 5 msec units per 20 msec audio chunk
- packet contains small units from different chunks
- if packet lost, still have most of every original chunk
- no redundancy overhead, but increases playout delay

11. Voice-over-IP: Skype



- proprietary application-layer protocol
  - encrypted msgs
- P2P components:
  - **clients**: skype peers connect directly to each other for VoIP call
  - **super nodes (SN)**: skype peers with special functions
  - **overlay network**: among SNs to locate SCs
  - **Login server**

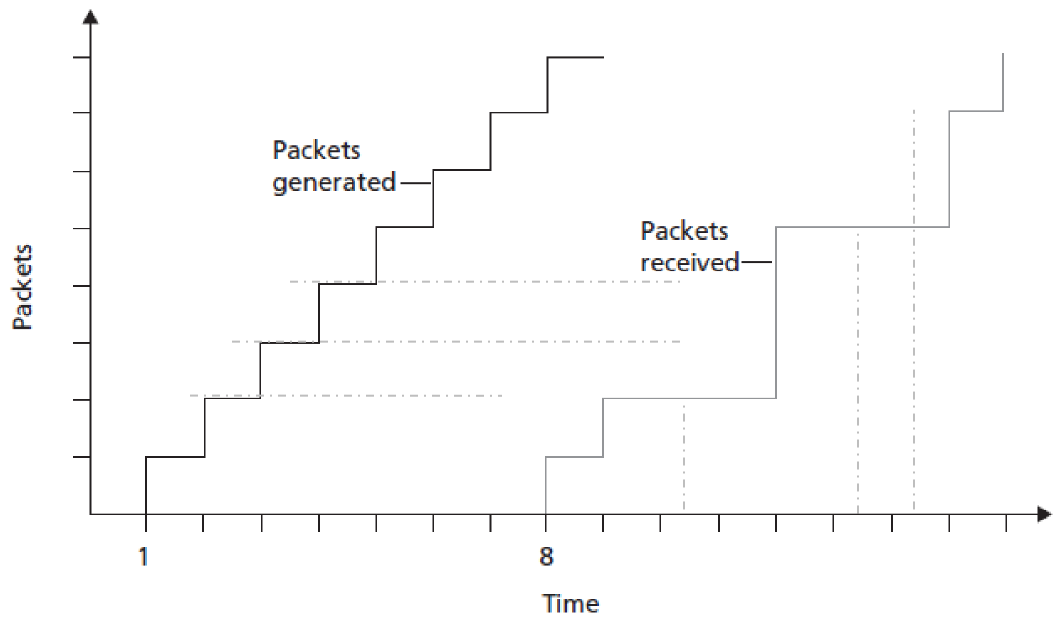


skype client operation:

- joins skype network by contacting SN (IP address cached) using TCP
- logs–in (usename, password) to centralized skype login server
- obtains IP address for callee from SN, SN overlay
- Initiate call directly to callee

12. Example: Adaptive playout delay

Consider the figure below. A sender begins sending packetized audio periodically at t = 1. The first packet arrives at the receiver at t = 8.



- **a.** What are the delays (from sender to receiver) of packets 2 through 8? Note that the horizontal line is divided in time slots.
- **b.** If audio playout begins at t = 9, which of the first eight packets sent will not arrive in time for playout?
- **c.** Compute the estimated delay for packets 2 through 8, using the formula for Adaptive Playout Delay estimation: $d_i = (1 - u)\, d_{i-1} + u\,(r_i - t_i)$. Use a value of u = 0.1 (so the receiver can adaptively adjust its playout delays.)

Solution

- **a.** The delay of packet 2 is 7 slots. The delay of packet 3 is 9 slots. The delay of packet 4 is 8 slots. The delay of packet 5 is 7 slots. The delay of packet 6 is 9 slots. The delay of packet 7 is 8 slots. The delay of packet 8 is > 8 slots.
- **b.** Packets 3 and 6 will not be received in time for their playout if playout begins at t=9.

t_i = the timestamp of the ith packet = the time the packet was generated by the sender

r_i = the time packet i is received by receiver

Packet Number	$r_i - t_i$	delay
1	7	$d_1 = 8 - 1 = 7$
2	7	$d_2 = 0.9\ d_1 + 0.1\ (r_2 - t_2) = 0.9 \times 7 + 0.7 = 7$
3	9	$d_3 = 0.9\ d_2 + 0.1\ (r_3 - t_3) = 0.9 \times 7 + 0.9 = 7.2$
4	8	$d_4 = 0.9\ d_3 + 0.1\ (r_4 - t_4) = 0.9 \times 7.2 + 0.8 = 7.28$
5	7	$d_5 = 0.9\ d_4 + 0.1\ (r_5 - t_5) = 0.9 \times 7.28 + 0.7 = 7.25$

Practice Simulation: Configuring VoIP in PacketTracer

(Packet Tracer is a network simulator from CISCO)

Video: ConfiguringVoIPinPacketTracer_cbtvid.mp4

Please refer to related video in the resources folder

(ConfiguringVoIPinPacketTracer_cbtvid)

Chapter 11

Quality of Service

Keywords:

Quality of Service, Bandwidth, Packet Loss, Delay, Jitter, marking, policing, scheduling.

Abstract:

First we introduce the parameters that affect directly QoS such as Bandwidth, Packet Loss, Delay, Jitter, then we present the principles of Quality of Service (QoS) management at the network level.

Learning Outcomes:

Upon completion of this lecture, you will be able to:

- Understand QoS Constraints
- Understand the nature of multimedia data and the scheduling and resource issues associated with it.
- Understand the nature of Quality of Service and the system support that it requires.

Outline:

- Characteristics of multimedia applications
- The constraints of QoS
- Quality of Service (QoS) management

Reference:

Computer Networking: A Top-Down Approach / James F. Kurose, Keith W. Ross.— 6th ed. Pearson 2013

1. Characteristics of multimedia applications:

- Large quantities of continuous data
- Timely and smooth delivery is critical
  - Deadlines
  - throughput and response time guarantees
- Interactive MM applications require low round-trip delays
  - End-to-end delay (transmission, processing, queuing delays in routers; propagation delays in the links; end-system processing delays) for interactive applications < 400 ms

2. Resources required:

Processor cycles in workstations and servers

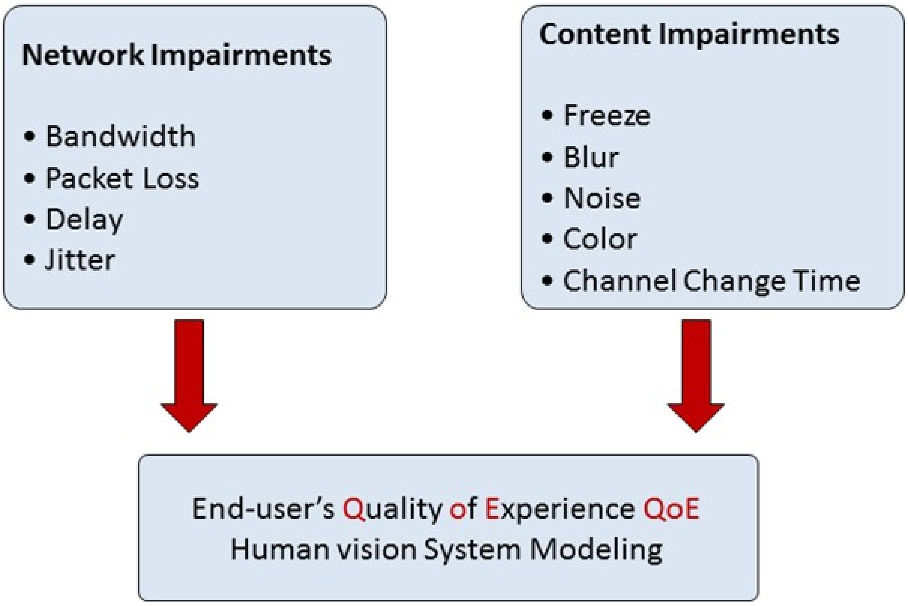
Network bandwidth (+ latency)

Dedicated memory

Disk bandwidth (for stored media)

At the right time
and in the right quantities

3. QoS and QoE:



4. Parameters that affect directly QoS:

The most important parameters:

- Bandwidth
- Packet Loss
- End to end Delay
- Jitter (enter packets delay variation)

4.1. Bandwidth requirements:

- Bandwidth is the most important parameter for offering services with high QoS.
- Thus, in order to manage bandwidth usage we should use some compression techniques.
- There are different MPEG compression methods. The most common methods used are MPEG2 and MPEG4.

Bandwidth requirements for offering one TV channel with MPEG-2 compression are:

- MPEG-2: SDTV ~ 4 Mbps → HDTV ~ 16 Mbps

Bandwidth requirements for offering one TV channel with MPEG-4 compression are:

- MPEG-4: SDTV ~ 2 Mbps → HDTV ~ 8 Mbps

These requirements for broadband dictate the needs for using broadband access technology. The best case is using FTTH. In rural areas this wireline technology is very expensive, so in these areas it is more appropriate to use wireless technology WiMAX.

4.2. Packet Loss:

- Network loss: IP datagram lost due to network congestion (router buffer overflow)
- Delay loss: IP datagram arrives too late for playout at receiver
  - delays: processing, queueing in network; end-system (sender, receiver) delays
  - typical maximum tolerable delay: 400 ms

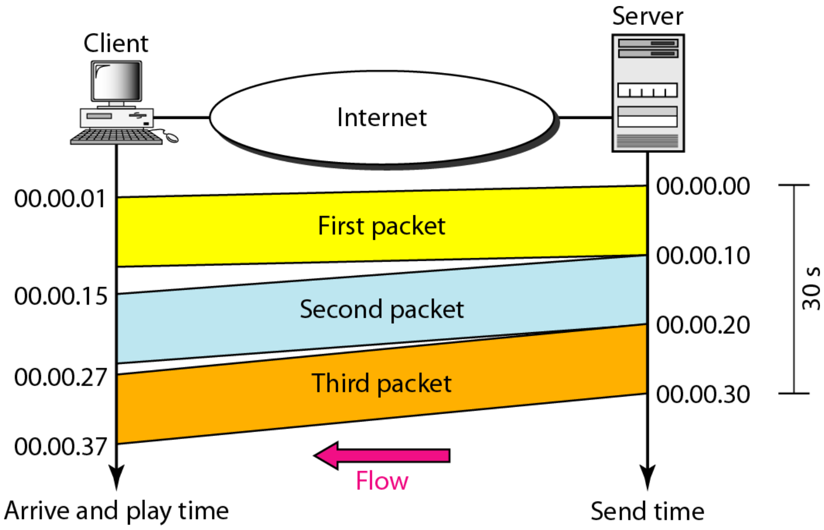
Packet Loss example:


Single B-frame IP packet loss (1 frame affected)


Single I-frame IP packet loss(14 frames affected)

4.3. Jitter:

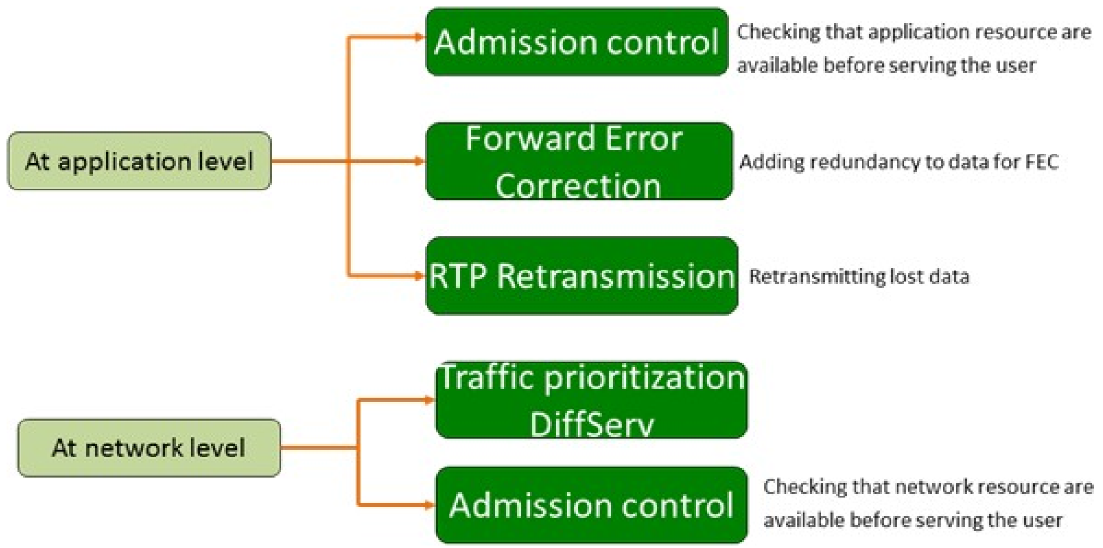
Jitter is introduced in real–time data by the delay between packets



5. QoS mechanisms can be implemented at different levels:

- At the **network level** it includes traffic management mechanisms such as **buffering** and **scheduling** employed to differentiate between traffics belong to different applications.
- At levels other levels than the transport include loss concealment, Forward Error Correction (FEC)

QoS Mechanisms & handling level:



6. Dimensioning best effort networks:

- ***approach:*** deploy enough link capacity so that congestion doesn't occur, multimedia traffic flows without delay or loss
  - low complexity of network mechanisms (use current "best effort" network)
  - high bandwidth costs
- challenges:
  - **network dimensioning:** how much bandwidth is "enough?"
  - **estimating network traffic demand:** needed to determine how much bandwidth is "enough" (for that much traffic)

7. Providing multiple classes of service:

- making the best of best effort service
  - one–size fits all service model
- alternative: multiple classes of service
  - partition traffic into classes
  - network treats different classes of traffic differently (analogy: VIP service versus regular service)

8. Multiple classes of service: scenario mixed HTTP and VoIP:

example: 1Mbps VoIP, HTTP share 1.5 Mbps link.

- HTTP bursts can congest router, cause audio loss
- want to give priority to audio over HTTP

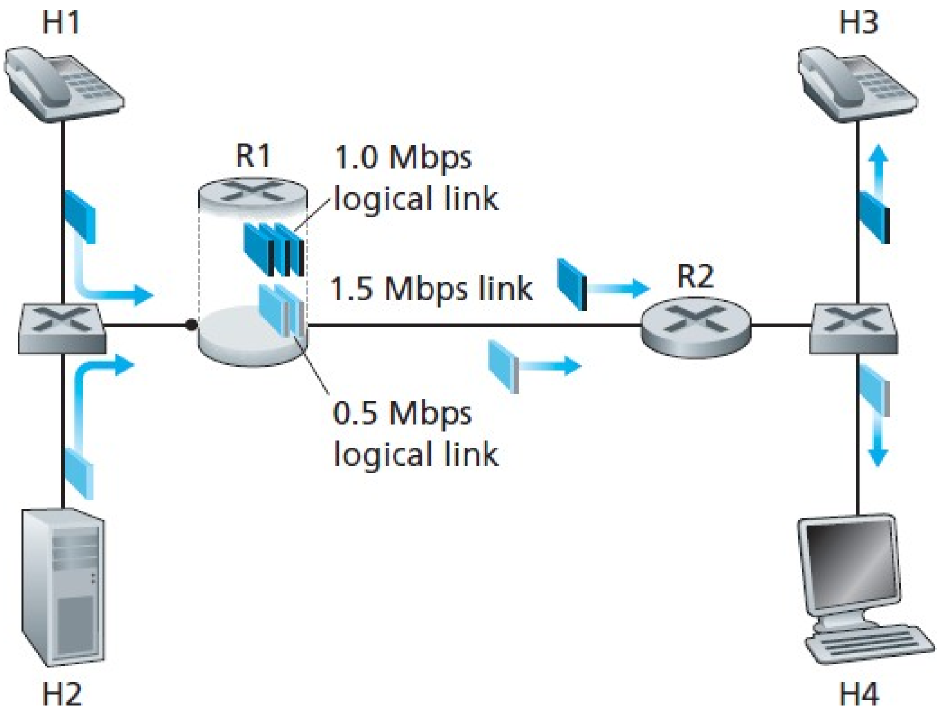
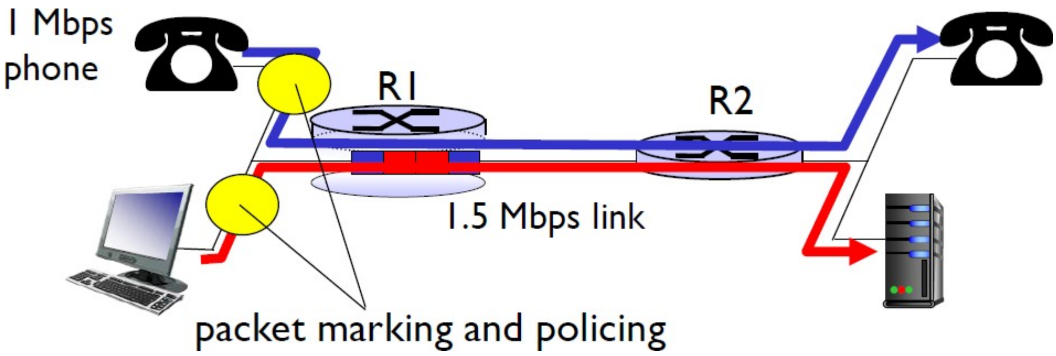


Figure shows a simple network scenario in which two application packet flows originate on Hosts H1 and H2 on one LAN and are destined for Hosts H3 and H4. The routers on the two LANs are connected by a 1.5 Mbps link. Let's assume the LAN speeds are significantly higher than 1.5 Mbps, and focus on the output queue of router R1; it is here that packet delay and packet loss will occur if the aggregate sending rate of H1 and H2 exceeds 1.5 Mbps. Let's further suppose that a 1 Mbps audio application shares the 1.5 Mbps link between R1 and R2 with an HTTP Web–browsing application that is downloading a Web page from H2 to H4.

9. Principles for QOS guarantees:

- what if applications misbehave (VoIP sends higher than declared rate)
  - policing: force source adherence to bandwidth allocations
- **marking, policing** at network edge



Principle 1

packet marking needed for router to distinguish between different classes; and new router policy to treat packets accordingly

In this scenario, a burst of packets from the Web server could potentially fill up the queue, causing IP audio packets to be excessively delayed or lost due to buffer overflow at R1. How should we solve this potential problem? Given that the Web-browsing application does not have time constraints, our intuition might be to give strict priority to audio packets at R1. Under a strict priority scheduling discipline, an audio packet in the R1 output buffer would always be transmitted before any HTTP packet in the R1 output buffer. In order for
R1 to distinguish between the audio and HTTP packets in its queue, each packet must be marked as belonging to one of these two classes of traffic.

10. Mechanisms for providing isolation among traffic classes :

We note two broad approaches can be taken.

- First, it is possible to perform traffic policing
- A complementary approach for providing isolation among traffic classes is for the link-level packet-scheduling mechanism to explicitly allocate a fixed amount of link bandwidth to each class.

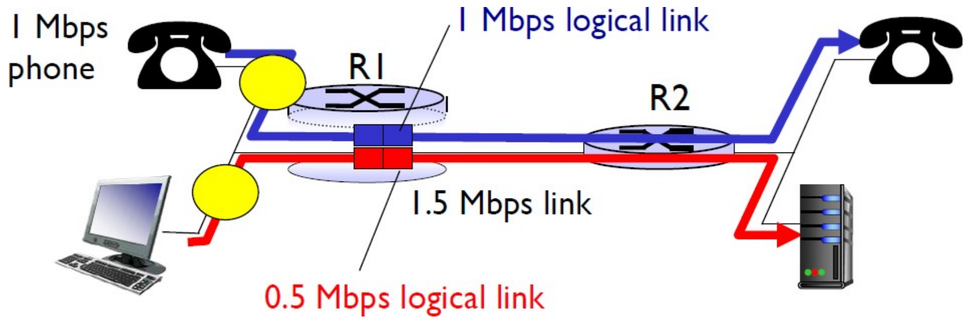
Principle 2:

provide protection (isolation) for one class from others

But what happens if the audio application starts sending packets at a rate of 1.5 Mbps or higher (either maliciously or due to an error in the application)? In this case, the HTTP packets will starve, that is, they will not receive any service on the R1-to-R2 link. Similar problems would occur if multiple applications (for example, multiple audio calls), all with the same class of service as the audio application, were sharing the link's bandwidth; they too could collectively starve the FTP session. Ideally, one wants a degree of isolation among classes of traffic so that one class of traffic can be protected from the other.

11. Principles for QOS guarantees (more):

allocating fixed (non-sharable) bandwidth to flow: inefficient use of bandwidth if flows doesn't use its allocation

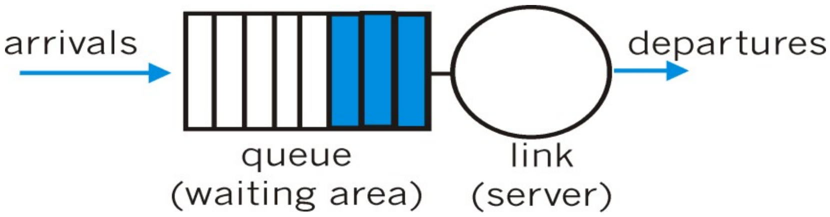


Principle 3:

While providing isolation among classes or flows, it is desirable to use resources (link bandwidth and buffers) as efficiently as possible.

For example, the audio class could be allocated 1 Mbps at R1, and the HTTP class could be allocated 0.5 Mbps. In this case, the audio and HTTP flows see a logical link with capacity 1.0 and 0.5 Mbps, respectively, as shown in the Figure. With strict enforcement of the link level allocation of bandwidth, a class can use only the amount of bandwidth that has been allocated; (it cannot utilize bandwidth that is not currently being used by others). For example, if the audio flow goes silent (for example, if the speaker pauses and generates no audio packets), the HTTP flow would still not be able to transmit more than 0.5 Mbps over the R1-to-R2 link, even though the audio flow's 1 Mbps bandwidth allocation is not being used at that moment.

12. Scheduling Mechanisms:

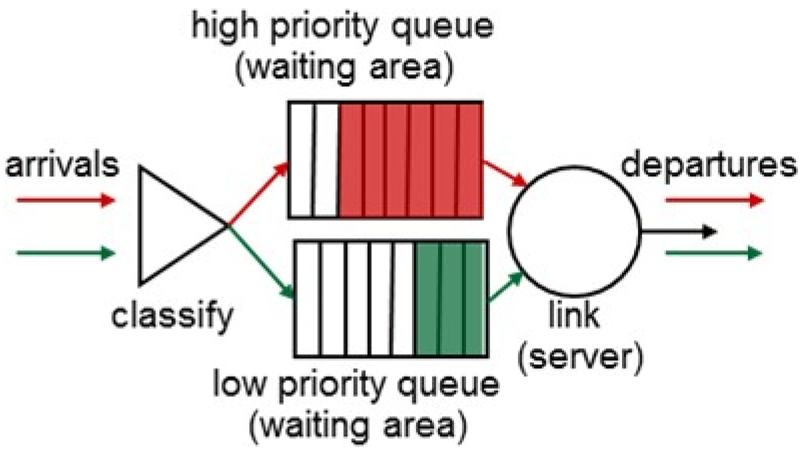


- **scheduling:** choose next packet to send on link
- **FIFO (first in first out)** scheduling: send in order of arrival to queue
- **discard policy:** if packet arrives to full queue: who to discard?
  - Tail drop: drop arriving packet
  - priority: drop/remove on priority basis
  - random: drop/remove randomly

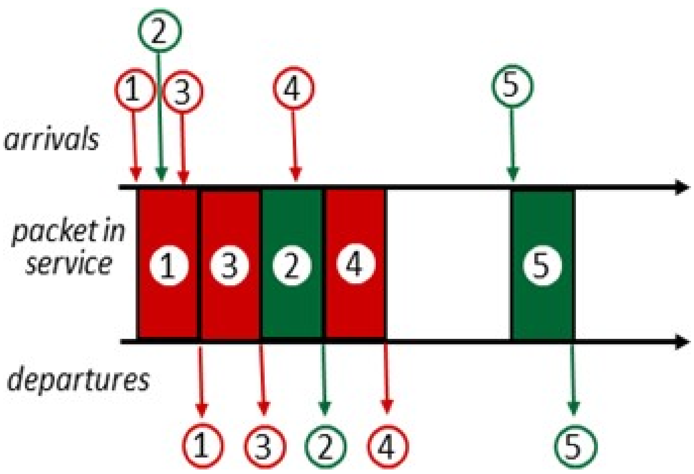
13. Scheduling policies: priority

priority scheduling: send highest priority queued packet

- multiple *classes*, with different priorities
- class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc.

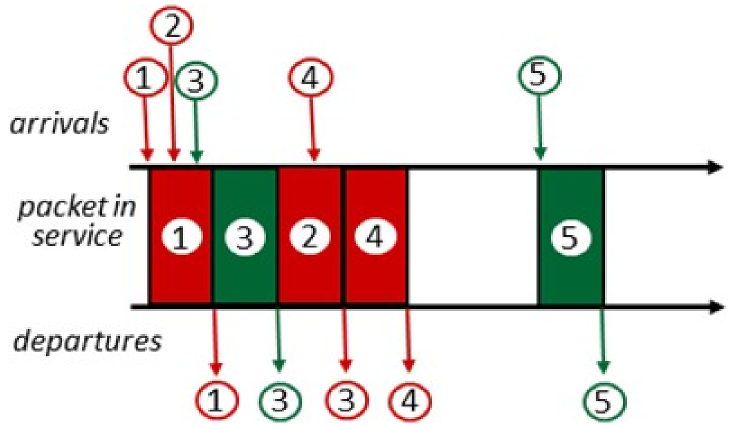


Note for developer: figure with the animation available in file QoS_IP.ppt in the resource folder



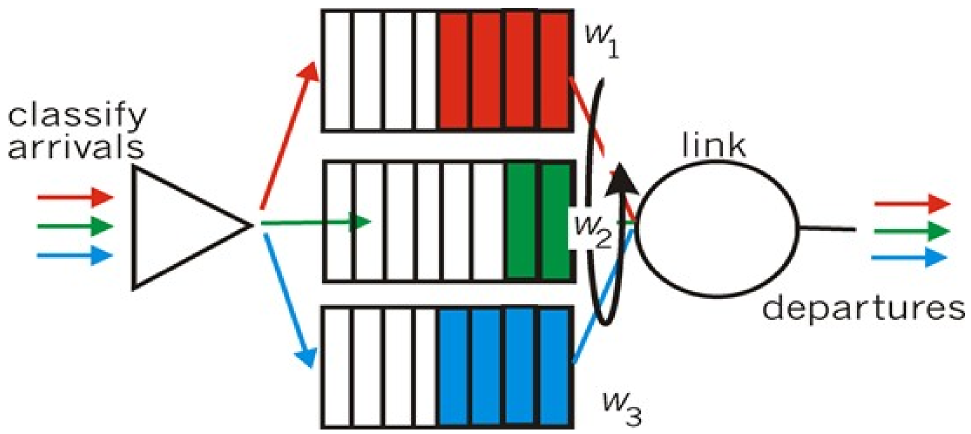
14. Round Robin (RR) scheduling:

- multiple classes
- cyclically scan class queues, sending one complete packet from each class (if available)



15. Weighted Fair Queuing (WFQ):

- generalized Round Robin
- each class gets weighted amount of service in each cycle



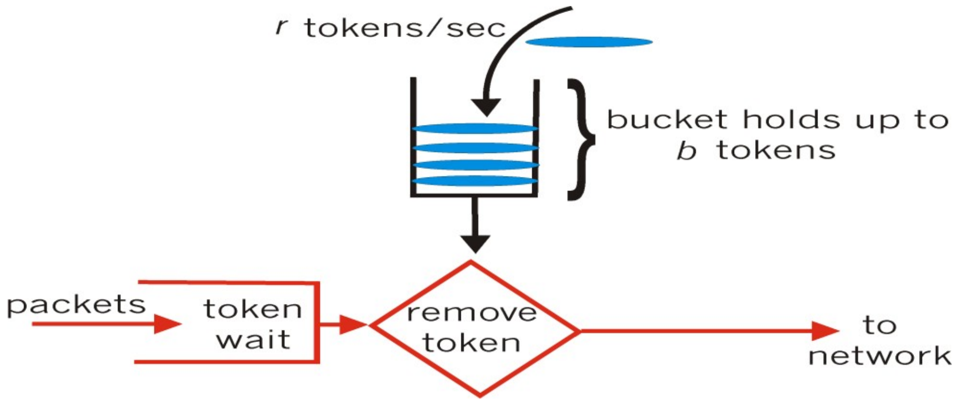
16. Policing mechanisms:

goal: limit traffic to not exceed declared parameters

Three common–used criteria:

- **(long term) average rate:** how many pkts can be sent per unit time (in the long run)
  - crucial question: what is the interval length: 100 packets per sec or 6000 packets per min have same average!
- **peak rate:** e.g., 6000 pkts per min (ppm) avg.; 1500 ppm peak rate
- **(max.) burst size:** max number of pkts sent consecutively (with no intervening idle)

17. Policing mechanisms: implementation:



token bucket: limit input to specified *burst size* and *average rate*

- bucket can hold b tokens
- tokens generated at rate *r token/sec* unless bucket full
- over interval of length t: number of packets admitted less than or equal to (r t + b)

Chapter 12

Cloud Computing for Multimedia Services

Keywords:

Cloud Computing, Computation Offloading, Content Delivery Network (CDN), Netflix, Multimedia Services, Interactive Cloud Gaming.

Abstract:

First we introduce the cloud computing that changed the service models for modern computer applications. Utilizing elastic resources in powerful data centers, it enables end users to conveniently access computing infrastructure, platforms, and software provided by remote cloud providers (e.g., Amazon, Google, and Microsoft).
This new generation of computing paradigm, offering reliable, elastic, and cost-effective resource provisioning, can significantly mitigate the overhead for enterprises to construct and maintain their own computing, storage, and network infrastructures.
In this context we introduce Computation Offloading for Multimedia Services and Interactive Cloud Gaming.

Learning Outcomes:

Upon completion of this lecture, you will be able to:

- Understanding how Cloud Computing could provide various computing and storage services for Multimedia over the Internet.
- Be familiar with Computation Offloading for Multimedia Services

Outline:

- Cloud Computing Overview
- Cloud-Assisted Media Sharing
- Computation Offloading for Multimedia Services
- Interactive Cloud Gaming

References:

1. Fundamentals of Multimedia, Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, Second Edition 2014
2. Computer Networking: A Top-Down Approach / James F. Kurose, Keith W. Ross.—6th ed. Pearson 2013

1. Introduction to Cloud Computing:

Cloud computing is a fast growing technology with bright career prospects for students and IT engineers. With big & small companies shifting their focus towards this technology, to move their online applications to cloud, the demand for trained and skilled professionals in cloud computing is ever increasing.

Cloud computing is a state-of-the-art, internet-based technology that provides access to services, storage, and resources on demand without the worry of downloading or installing anything on your computer.

A recent survey by Microsoft & the International Data Corporation predicts that there will be a need for about 14 million cloud computing professionals by 2015.

2. What is Cloud Computing?

- "Cloud" refers to large Internet services running on 10,000s of machines (Google, Facebook, etc)
- "Cloud computing" refers to services by these companies that let external customers rent cycles
  - Amazon EC2: virtual machines, billed hourly
  - Amazon S3: storage
  - Windows Azure: applications using Azure API
- Attractive features:
  - Scale: 100s of nodes available in minutes
  - Fine-grained billing: pay only for what you use
  - Ease of use: sign up with credit card, get root access
- Old ideas:
  - **Software as a Service (SaaS)**: delivering applications over the Internet
- Recently: "[Hardware, Infrastructure, Platform] as a service"
  - Poorly defined so we avoid all "X as a service"
- **Utility Computing**: pay-as-you-go computing
  - Illusion of infinite resources
  - Fine-grained billing (e.g. hourly)

3. Cloud Computing Overview:

- A new paradigm, allowing for computing resources to be purchased or shared using a pay for usage model.
- Much like traditional utility resources, such as electrical power, users are now free to purchase or lease resources as needed.
- Many existing applications, from content sharing to media streaming have been migrated to this model with great success in terms of costs and system efficiency.
- Examples include the file synchronization system DropBox and the media streaming service Netflix

The emergence of cloud computing has dramatically changed the service models for modern computer applications. This new generation of computing paradigm, offering reliable, elastic, and cost-effective resource provisioning, can significantly mitigate the overhead for enterprises to construct and maintain their own computing, storage, and network infrastructures.

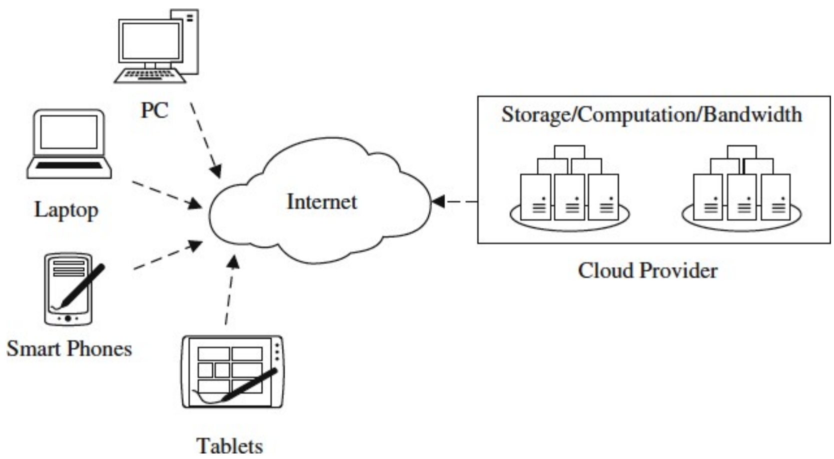
Start-up companies can easily implement their novel ideas into real products with minimum investment in the initial stage and expand the system scale without much effort later on. A representative is Dropbox, a typical cloud storage and file synchronization service provider, which, from the very beginning, has relied on Amazon's S3 cloud servers for file storage and uses Amazon's EC2 cloud instances to provide such key functions as synchronization and collaboration among different users. We have also seen such new generation of multimedia services as cloud-based VoD and gaming that have emerged in the market and may change the whole business model in the coming years. A prominent example is Netflix, a major Internet streaming video provider, which migrated its infrastructure to Amazon's cloud platform in 2010 and has since become one of the most important cloud users.

4. Cloud Computing: Private vs. Public:

Clouds a generally split into two categories private and public:

- **Private clouds** are generally maintained on private networks and usually only utilized by a single datacenter tenant to provide services to their customers.
- **Public Clouds** are available as a public utility where multiple tenants can purchase resources to provide services to their customers. For example, Amazon Elastic Compute Cloud (EC2).

5. A conceptual overview of cloud computing:



A conceptual overview of cloud computing [1]

- Cloud users use an internet enabled device to connect to a cloud computing provider.
- The cloud provider can provide superior computing resources such as storage, computation and bandwidth.
- End users can store data in the cloud, making ubiquitous data access possible.

6. Why the Cloud?

- in cloud computing, the user's computer may contain almost no software or data (perhaps a minimal operating system and web browser only), serving as little more than a display terminal for processes occurring on a network of computers far away
- No installing programs, instead all work is done through web apps
- Minimal PC storage the majority of user data is stored in the cloud
- Web applications are run on servers so much of the heavy processing can be run off system

7. Cloud Computing: Characteristics:

- **On-Demand Self Service:** A user can unilaterally provision computing resources.
- **Resource Pooling and Rapid Elasticity:** Cloud resources are available for provisioning by multiple users and resources can be dynamically assigned
- **Measured Service:** Computing resource usage (eg. computation, storage, bandwidth) can be monitored, controlled and reported to both the cloud provider and user.
- **Broad Network Access:** Persistent and high quality network access is available.

8. Cloud Service Models (Everything as a Service):

- Software as a Service (SaaS)
  - Allows applications to run on the infrastructure and platforms offered by the cloud. Often storing both the users data and application in the cloud. Examples include Google Drive (Formely Google Docs).
- Platform as a Service (PaaS)
  - Delivers a development environment, typically include preselected operating systems, programming environments, databases management systems, and webservers. Google's App Engine is a typical example.
- Infrastructure as a Service (IaaS)
  - Provides a pool of resources, which the user accesses and configures for their needs. Users are usually free to install their own operating systems and applications. Examples include the virtual machine based Amazon EC2 and the storage service Amazon S3.

Software as a Service (SaaS):

SaaS, probably the most widely used cloud service model to date, allows an application to run on the infrastructure and platforms offered by the cloud rather than on local hardware/software. As such, the user of an application does not have to heavily invest on it own servers, software, license, etc. SaaS is usually priced on a usage basis, or with a monthly or yearly flat fee per user.

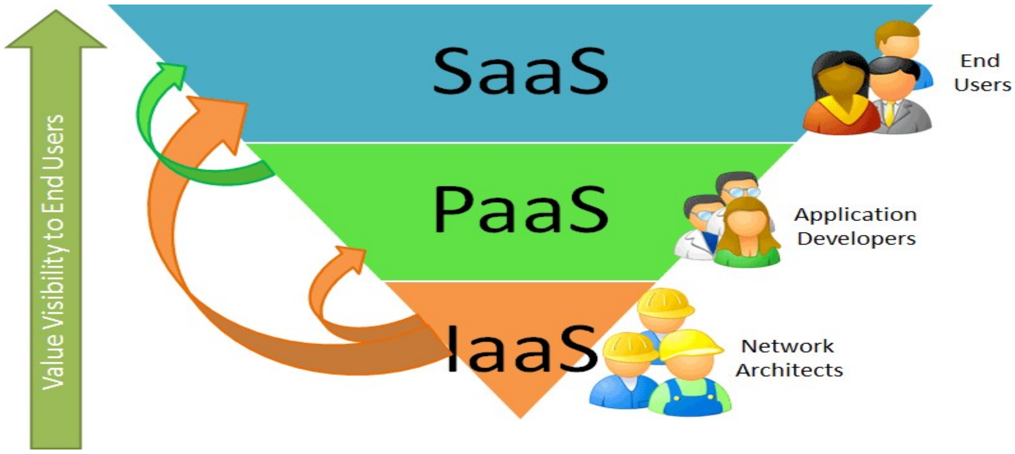
Platform as a Service (PaaS):

PaaS delivers development environments as a service, which typically includes the operating system, programming language execution environment, database, web server, etc. Applications can be built and run on the PaaS provider's infrastructure and then delivered to end users via the Internet. As such, the cost and complexity of purchasing and managing the underlying hardware and software layers can be greatly reduced. Moreover, the underlying computation and storage resources can scale automatically to match the applications' demands. Google's App Engine is a typical example of PaaS.

Infrastructure as a Service (IaaS):

IaaS is the very basic and essential cloud service. Well-known examples of IaaS include such infrastructure as the Amazon's Elastic Compute Cloud (EC2), which allow users to rent virtual machines on which to run applications, and such cloud storage services as Amazon's Simple Storage Service (S3), which allow users to store and retrieve data, at any time, from anywhere on the Web. In general, an IaaS provider offers a pool of computation resources, in the form of physical machines, or more often, *virtual machines*, as well as other resources, e.g., virtual-machine disk image library, data block or file-based storage, firewalls, load balancers, IP addresses, virtual local area networks, and software bundles.

9. An illustration of cloud service models:



10. Cloud Computation Example: Amazon EC2:

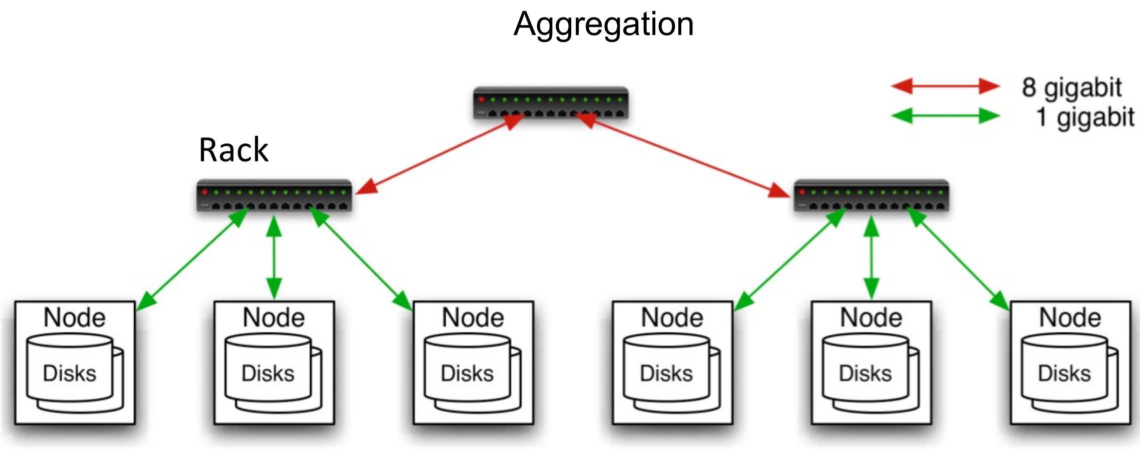
- Amazon's Elastic Compute Cloud (EC2) is a public computational cloud that provides resizable compute instances.
  - Implemented using Virtual Machines (VM) provided by the Xen virtualization system.
  - VMs can be resized dynamically by the users, ex. To increase CPU resources, available ram, and disk space.
- Different instance provisioning strategies exist, namely "on-demand" and "spot" instances.
  - On-Demand: Allows users to purchase computation capacity by the hour at a set price.
  - Spot: Users bid on unused EC2 capacity and their instances run while their bid price exceeds the current spot price.

11. Enabling technologies for cloud computing:

Roots of clouds computing:

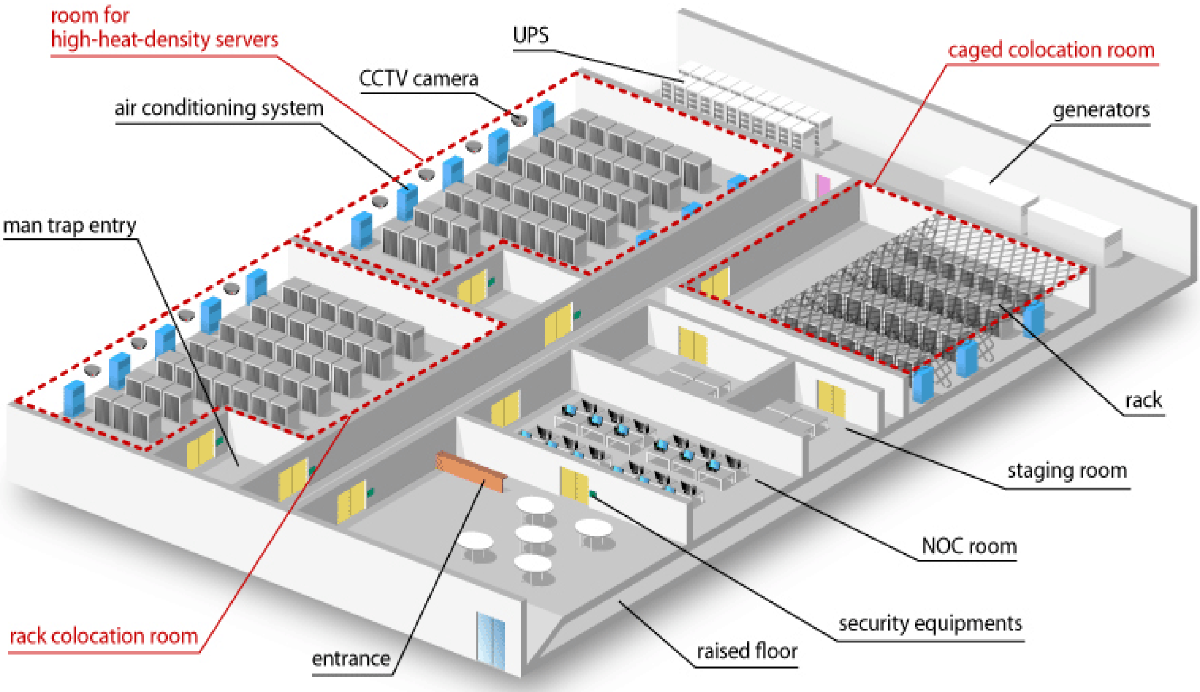
- **Virtualization,** multi–core chips (advancement in hardware technology)
- Internet technologies (**Web Services,** Service–Oriented Architectures, Web 2.0)
- **Distributed computing** (clusters, grids)
- **Data center automation**

11.1. Typical Cluster:



- 40 nodes/rack, 1000−4000 nodes in cluster
- 1 Gbps bandwidth in rack, 8 Gbps out of rack
- Node specs (Facebook): 8−16 cores, 32 GB RAM, 8×1.5 TB disks

11.2. Data center:


room for
high-heat-density servers
UPS
caged colocation room
CCTV camera
air conditioning system
generators
man trap entry
rack
staging room
NOC room
security equipments
rack colocation room
raised floor
entrance

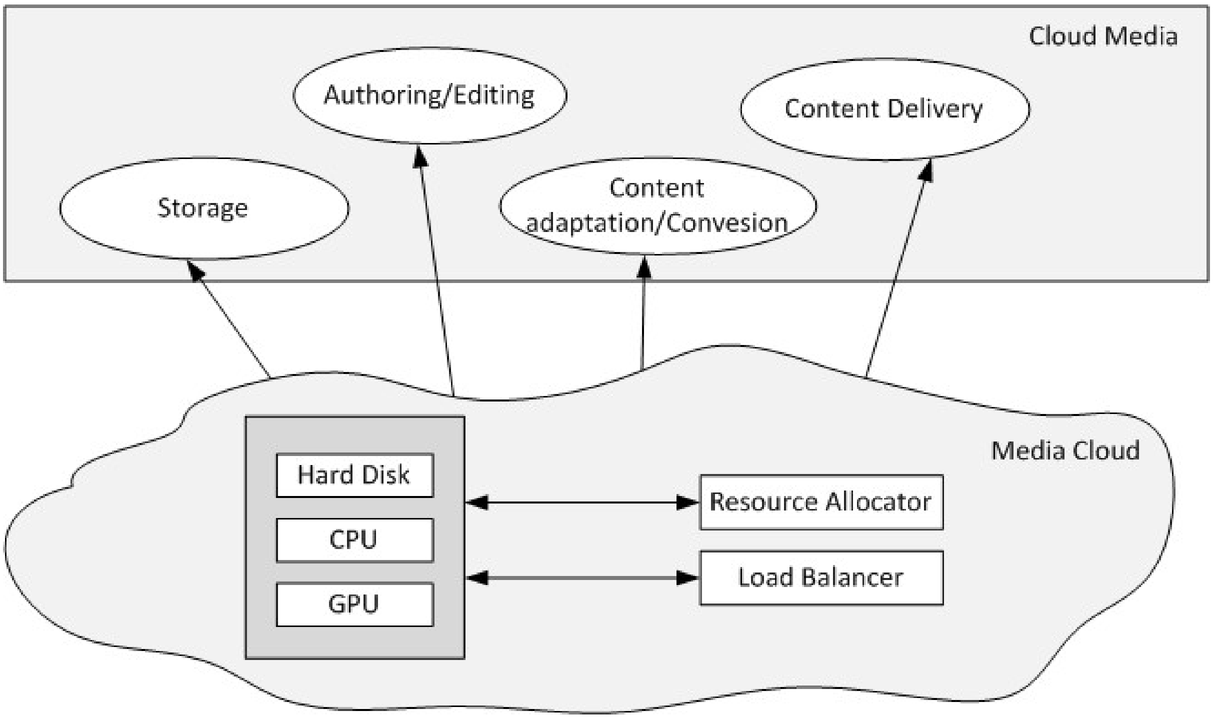
12. Multimedia Cloud Computing:

- Distribution of multimedia content has always been a fundamental issue for media service and content providers.
- The large computational, storage and bandwidth capacity of the Cloud make it a very attractive platform for multimedia applications. Further, large public cloud often have many strategically geographically located data-centers allowing for more efficient distribution of content.
- Multimedia Cloud Computing and general purpose Cloud Computing share many commonalties. However, multimedia services are highly diverse; examples include voice over IP, video conferencing, and video streaming.

For multimedia applications and services over the Internet, there are strong demands for cloud computing, due to the massive storage and computation required for serving millions of wired Internet or mobile network users. In this new multimedia cloud computing paradigm, users can store and process their multimedia data in the cloud in a distributed manner, eliminating full installation of the media application software on local computers or devices.

Multimedia cloud computing shares many common characteristics with general purpose cloud computing. Yet, multimedia services are highly heterogeneous. There exist a wide array of types of media and associated services, such as voice over IP, video conferencing, photo sharing and editing, image search, and image-based rendering. Besides service heterogeneity, different types of devices, such as smart TVs, PCs, and Smartphones, have different capabilities for multimedia processing

13. Modules and their relation in Multimedia Cloud Computing:



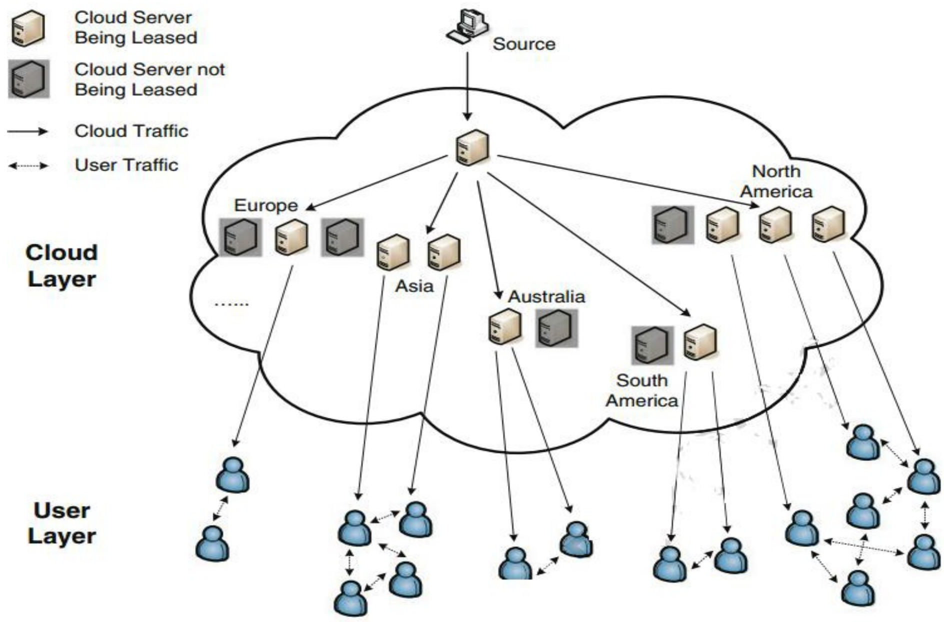
The heavy demands of multimedia data access, processing, and transmission would create bottlenecks in a general-purpose cloud. Today's cloud design has largely focused on allocating computing and storage resources through utility like mechanisms, while QoS requirement in terms of bandwidth, delay, and jitter have yet to be addressed.

To realize multimedia cloud computing, a synergy between the cloud and multimedia services becomes necessary; that is, multimedia-aware cloud computing with enhanced QoS support for multimedia applications and services, and cloud-aware multimedia that facilitates content storage, processing, adaptation, rendering, in the cloud with optimized resource utilization, as the above Figure illustrates.

14. Cloud-Assisted Media Sharing:

- Media sharing and distribution services are evolving extremely quickly. New services and applications are being developed constantly, some become hugely successful like YouTube, many others fade into obscurity.
- Developers face a trade off in the early stage of any multimedia service.
  - How to provision enough resources to provide a high quality experience to their current users, as well as allowing room for the user base to grow, while not overextending themselves on costly infrastructure projects and overhead.
- The Cloud's "pay-as-you-go" service allows a service to start small and scale large without huge initial capital investments.

15. Generic Framework Media Streaming:

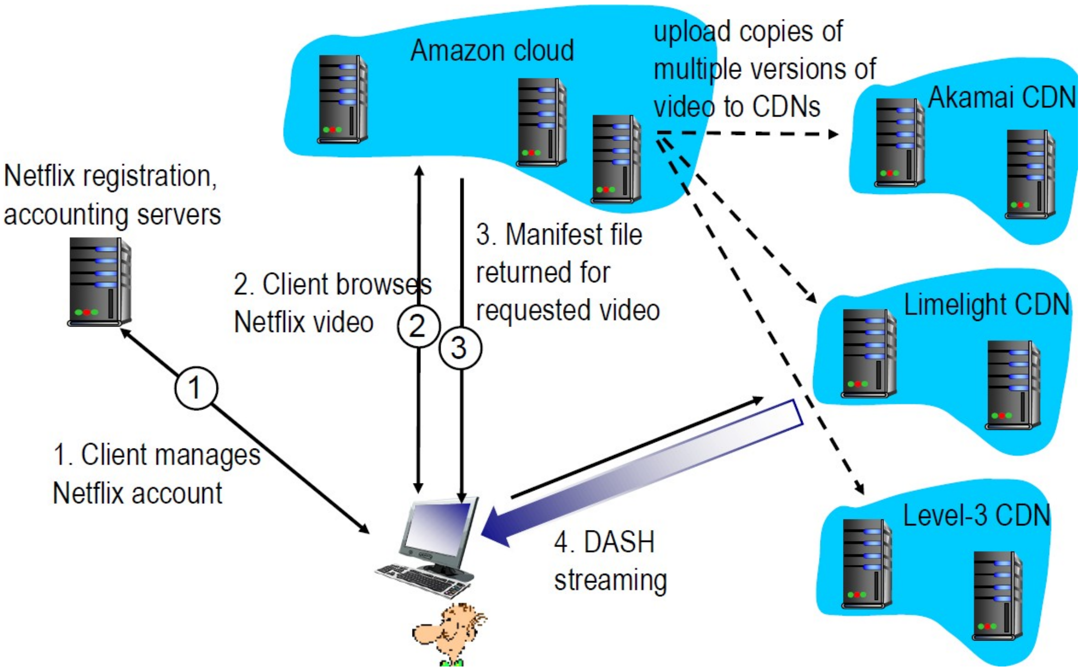


A generic framework for migrating live media streaming service to the cloud [1]

Figure shows a generic framework that facilitates the migration of existing live media streaming services to a cloud-assisted solution. It is divided into two layers, namely, Cloud Layer and User Layer. The Cloud Layer consists of the live media source and dynamically leased cloud servers. Upon receiving a user's subscription request, the Cloud Layer will redirect this user to a properly selected cloud server. Such a redirection is transparent to the user, i.e., the whole Cloud Layer is deemed to be a single source server from a user's perspective. Since the user demands change over time, which are also location-dependent, the Cloud Layer will accordingly adjust the amount and location distribution of the leased servers. Intuitively, it will lease more server resources upon demand increase during peak times, and terminate leases upon decrease.

16. Case Study: NetFlix:

- Combing huge network traffic requirements and dynamic unpredictable demand bursts Netflix has become a prominent example of a business greatly enhanced by multimedia Cloud Computing.
- As of 2012 nearly all of Netflix's media infrastructure has been moved to Amazon public cloud.
- Netflix leverages Amazon's cloud services to provide the following features:
  - **Content Conversion:** Using EC2 to convert master copies to over 50 different versions, with varying levels of video resolution, video quality and audio quality.
  - **Content Storage:** Utilizing S3 to store the many converted copies.
  - **Content Distribution:** Data is distributed from S3 to various CDNs, where the media is ultimately streamed to the end user.
- 30% downstream US traffic in 2011
- owns very little infrastructure, uses 3rd party services:
  - own registration, payment servers
  - Amazon (3rd party) cloud services:
    a. Netflix uploads studio master to Amazon cloud
    b. create multiple version of movie (different endodings) in cloud
    c. upload versions from cloud to CDNs
    d. Cloud hosts Netflix web pages for user browsing
- three 3rd party CDNs host/stream Netflix content: Akamai, Limelight, Level-3



The cloud-based Netflix architecture (from Kurose & Ross [2])

17. Computation Offloading for Multimedia Services:

Many computation-intensive applications are now being migrated to the cloud.

- Users can access high-performance servers and clusters as needed meaning they do not have to maintain expensive infrastructure.
- Further, users on relatively weak computational platforms such as cell phones and tablets can offload some of their computational tasks to save battery and improve performance.

Besides storage, computation is another rich resource offered by the cloud. Many computation-intensive tasks can now be migrated to the cloud, and the users do not have to maintain the ultra expensive high-performance servers or server clusters but just pay for the cost in an on-demand fashion.

Such computation offloading effectively expands the usability of local devices beyond their physical limits, which is particularly attractive for mobile devices. Today's smartphones and tablets are increasingly penetrating into people's everyday life as efficient and convenient tools for communication and entertainment. The touch screen and all kinds of sensors provide even richer user experiences than desktop PCs do.

18. Service Partitioning for Video Coding:

- Video encoding/compression, an essential task in a broad spectrum of mobile multimedia applications.
- Uploading the raw video without efficient encoding inevitably leads to high bandwidth cost and large transmission energy consumption.
- Video encoding often incurs heavy computation, which results in high energy consumption as well.

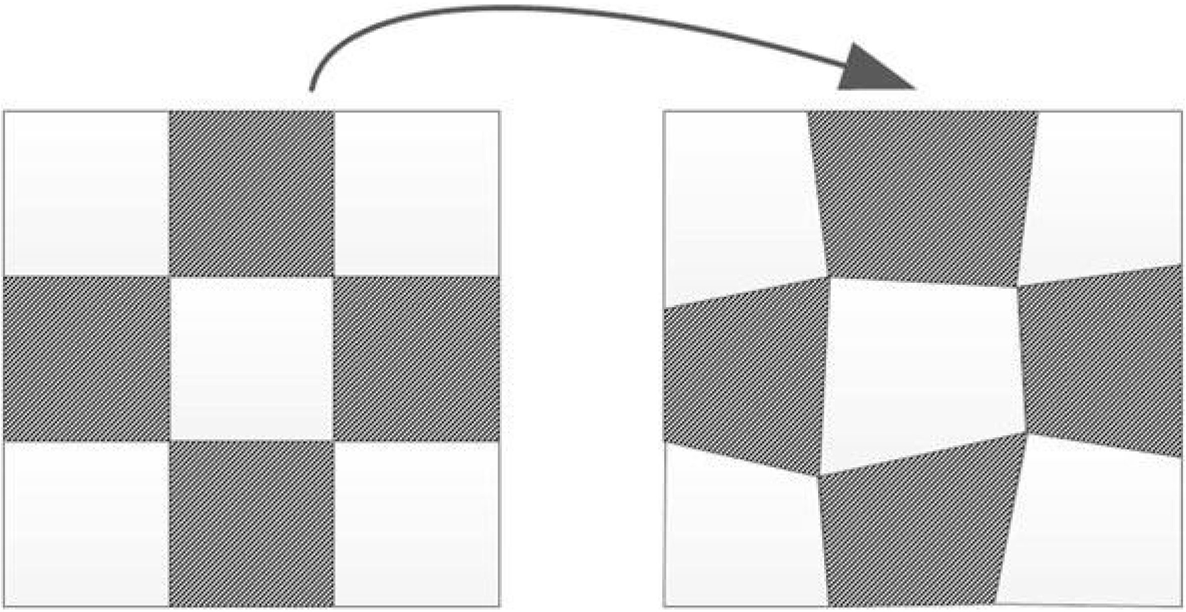
Consider video encoding/compression, an essential task in a broad spectrum of mobile multimedia applications. A local user uses his/her mobile terminal to capture video in real-time, expecting to encode the video and then stream it to others in real-time as well. Directly uploading the raw video without efficient encoding inevitably leads to high bandwidth cost and large transmission energy consumption. On the other hand, video encoding often incurs heavy computation, which results in high energy consumption as well.

Offloading the whole video compression task to the cloud, however, is not practical because it is identical to directly uploading the raw video data. It is known that motion-estimation is the most computation-intensive module, accounting for almost 90% of the computation. This module obviously should be the focus of offloading.

19. Case Study: Cloud–Assisted Motion Estimation:

It is necessary to ensure that a minimum amount of data is to be uploaded to the cloud and yet allow estimation to be done accurately.

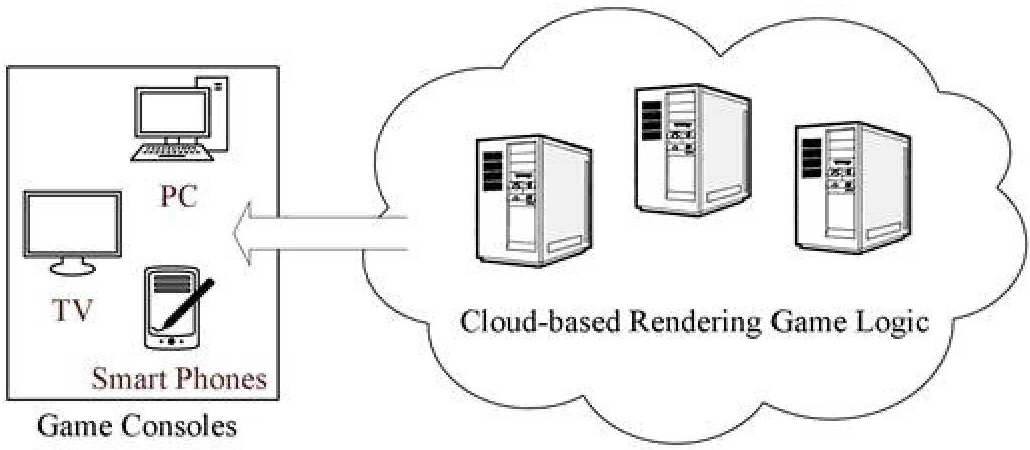
Cloud–Assisted Motion Estimation (CAME) addresses this issue by using mesh–based motion estimation, which partitions a frame into a coarse–grained mesh sketch and estimates one Motion Vector (MV) for each mesh node.



Mesh–based motion estimation between two frames, using a regular mesh for the predictive frame

As illustrated in the Figure, a set of regular mesh nodes are sampled on the P–frames of the macro block (MB). Only the sampled mesh nodes and the reference frame are uploaded to the cloud. The MVs are then calculated on the cloud, and the result, to be sent back to the mobile terminal, is a set of MVs that describe the movements of the mesh nodes. Using this design, the most computation–intensive part of mesh–based motion estimation is offloaded to the cloud, while the local motion estimation for individual macro–blocks within the mesh becomes much simpler.

20. Interactive Cloud Gaming:

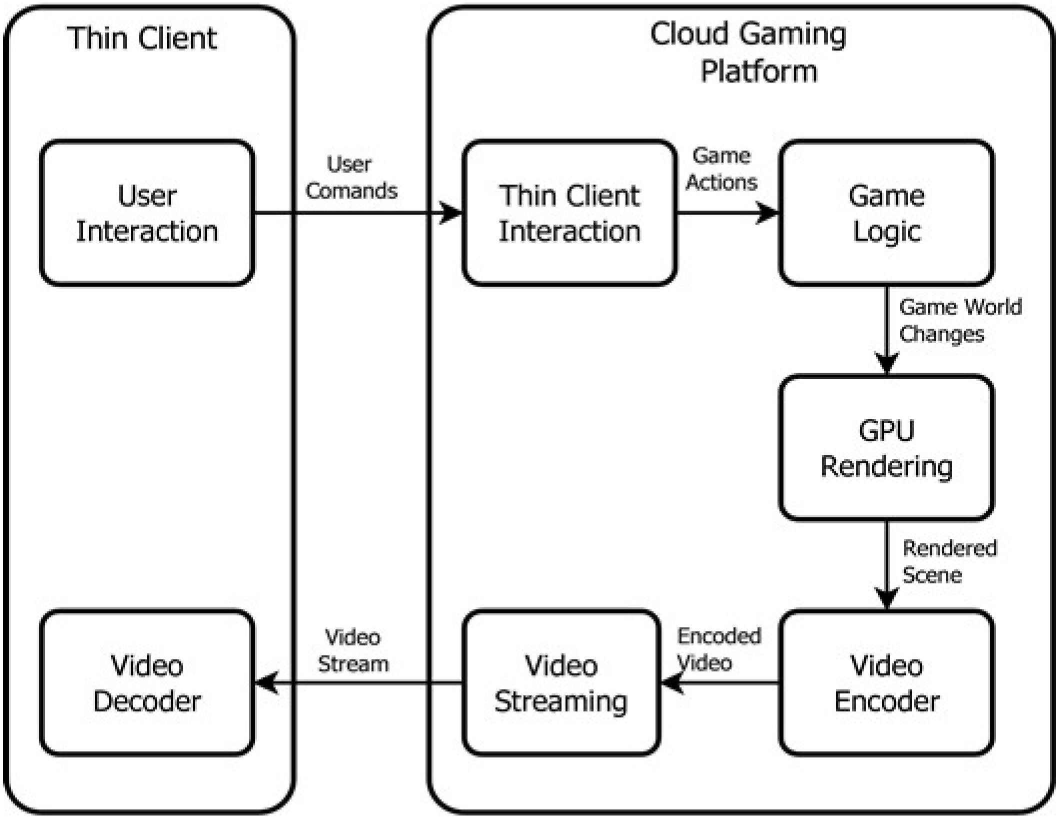


- Cloud Gaming at its core uses thin–clients to interact with powerful cloud–based servers to render interactive games remotely and stream the images back.
- Many of todays top interactive games require powerful desktop class components to provide an enjoyable gaming experience. Many relatively weaker devices such as tablets and smart phones do not meet the minimum requirements.

21. Cloud Gaming Challenges:

- From ultra low latency live video streaming to high performance 3D processing, cloud gaming has overcome considerable challenges.
- In the time frame of only 100-200 milliseconds a cloud gaming system must collect a player's actions, transmit them to the cloud server, process the action, render the results, encode/compress the resulting changes to the game-world, and stream the video (game scenes) back to the player.
  - Very little time to employ video streaming buffers often found in other media streaming applications.
- Many optimizations and technologies must be used to lower the players interaction delay (delay between a player's action and resultant game scene).
  - GPU manufactures such as NVidia are developing cloud enable GPUs specifically targeted at cloud gaming.

22. Cloud Gaming Architecture Overview:



A generic framework of cloud gaming. From reference[1]

Recently, advances in cloud technology have expanded to allow offloading not only of traditional computation but also of such more complex tasks as high definition 3D rendering, which turns the idea of Cloud Gaming into a reality. Cloud gaming, in its simplest form, renders an interactive gaming application remotely in the cloud and streams the scenes as a video sequence back over the Internet to the player. A cloud gaming player interacts with the application through a thin client, which is responsible for displaying the video from the cloud rendering server as well as collecting the player's commands and sending the interactions back to the cloud.

Figure above shows a high-level architectural view of such a cloud gaming system with thin clients and cloud-based rendering.

Cloud gaming can bring great benefits by expanding the user base to the vast number of less-powerful devices that support thin clients only, particularly smartphones and tablets. As such, mobile users can enjoy high-quality video games without performing the computation-intensive image rendering locally.