دانشکده آموزش های الکترونیکی دانشگاه شیراز

مالتی مدیا در شبکه

سیستم های چند رسانه ای

وحید اعتمادی

بهار 94

Multimedia Networking Application

- the most salient characteristic of video is its **high bit rate**
- Video distributed over the Internet typically ranges from 100 kbps for low-quality video conferencing to over 3 Mbps for streaming high-definition movies

Multimedia Networking Application

	Bit rate	Bytes transferred in 67 min
Facebook Frank	160 kbps	80 Mbytes
Martha Music	128 kbps	64 Mbytes
Victor Video	2 Mbps	1 Gbyte

Table 7.1 ♦ Comparison of bit-rate requirements of three Internet applications

Multimedia Networking Application

- Therefore, when designing networked video applications, the first thing we must keep in mind is the high bit-rate requirements of video

- Cisco predicts [Cisco 2011] that streaming and stored video will be approximately 90 percent of global consumer Internet traffic by 2015

Multimedia Networking Application

- Another important characteristic of video is that it can be compressed.
- *Spatial redundancy* is the redundancy within a given image.
- *Temporal redundancy* reflects repetition from image to subsequent image.

- نرخ بیت بالاتر، کیفیت بهتر و تجربه بهتر کاربر در استفاده از کاربردهای مالتی مدیا

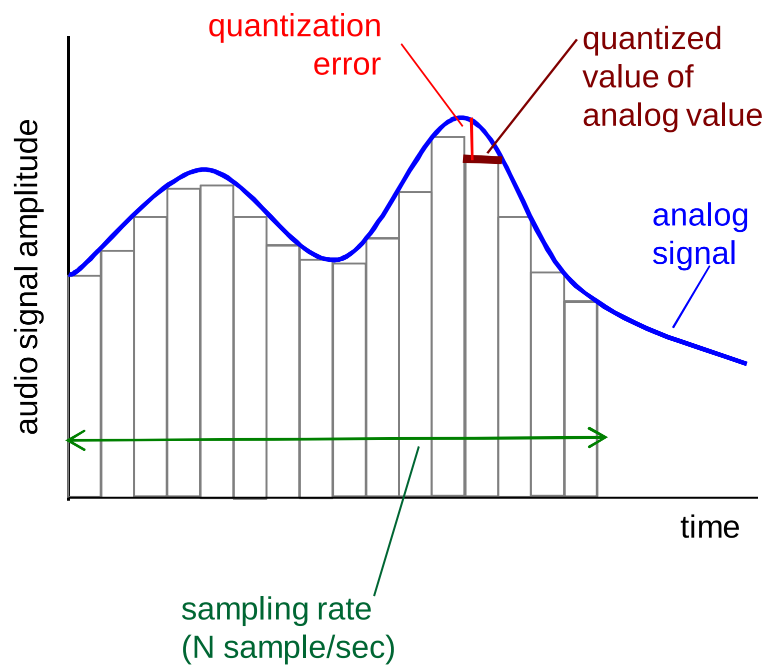
Multimedia Networking Application

- We can also use compression to create **multiple versions** of the same video
  - each at a different quality level

- Users can then decide which version they want to watch as a function of their current available bandwidth

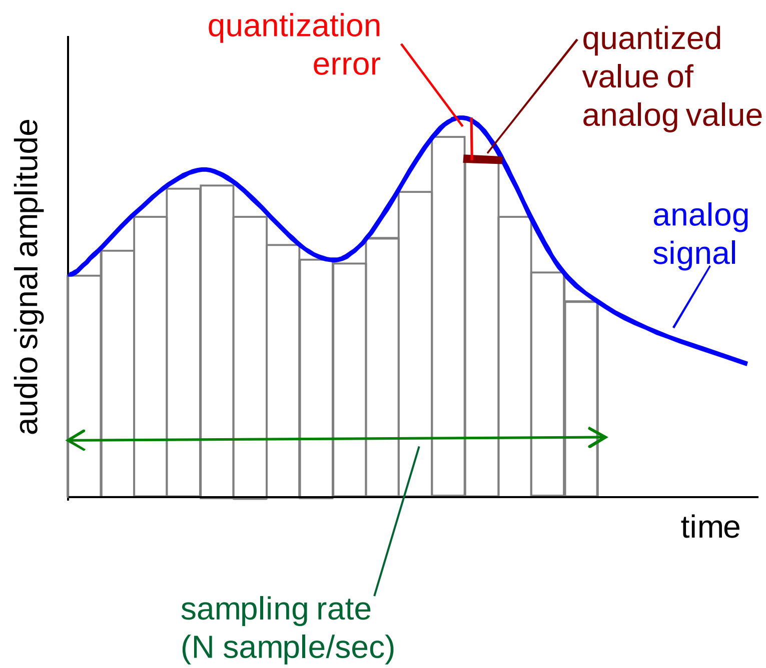
Multimedia: audio

- analog audio signal sampled at constant rate
  - telephone: 8,000 samples/sec
  - CD music: 44,100 samples/sec
- each sample quantized, i.e., rounded
  - e.g., 2^8=256 possible quantized values
  - each quantized value represented by bits, e.g., 8 bits for 256 values



Multimedia: audio

- example: 8,000 samples/sec, 256 quantized values: 64,000 bps
- receiver converts bits back to analog signal:
  - some quality reduction



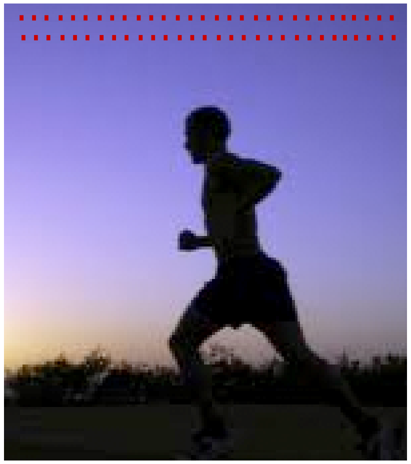
example rates

- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3 kbps and up

Multimedia: video

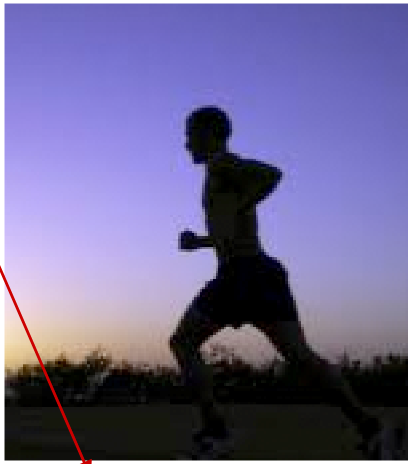
- video: sequence of images displayed at constant rate
  - e.g. 24 images/sec
- digital image: array of pixels
  - each pixel represented by bits
- coding: use redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

spatial coding example: instead of sending N values of same color (all purple), send only two values: color value (purple) and number of repeated values (N)



frame i

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i

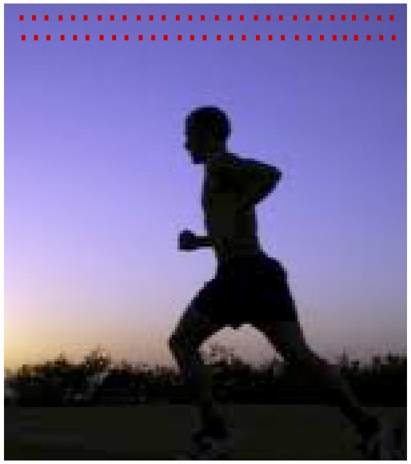


frame i+1

Multimedia: video

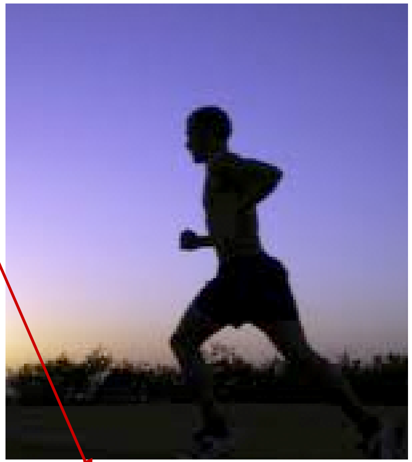
- CBR: (constant bit rate): video encoding rate fixed
- VBR: (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes
- examples:
  - MPEG 1 (CD-ROM) 1.5 Mbps
  - MPEG2 (DVD) 3-6 Mbps
  - MPEG4 (often used in Internet, < 1 Mbps)

spatial coding example: instead of sending N values of same color (all purple), send only two values: color value (purple) and number of repeated values (N)



frame i

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i



frame i+1

Multimedia networking: 3 application types

- *streaming, stored* audio, video
  - *streaming:* can begin playout before downloading entire file
  - *stored (at server):* can transmit faster than audio/video will be rendered (implies storing/buffering at client)
  - e.g., YouTube, Netflix, Hulu
- *conversational* voice/video over IP
  - interactive nature of human-to-human conversation limits delay tolerance
  - e.g., Skype
- *streaming live* audio, video
  - e.g., live sporting event (futbol)

streaming, stored video

- Streaming stored video has three key distinguishing features
  - *Streaming.* In a streaming stored video application, the client typically begins video playout within a few seconds after it begins receiving the video from the server
  - *Interactivity.* Because the media is prerecorded, the user may pause, reposition forward, reposition backward, fast-forward, and so on through the video content
  - *Continuous playout.* Once playout of the video begins, it should proceed according to the original timing of the recording.

- the most important performance measure for streaming video is **average throughput**
- using buffering and prefetching, it is possible to provide continuous playout
- For many streaming video applications, prerecorded video is stored on, and streamed from, a CDN rather than from a single data center

conversational voice over IP

- Real-time conversational voice over the Internet is often referred to as **Internet telephony**
- It is also commonly called **Voice-over-IP (VoIP)**
- Most of today's voice and video conversational systems allow users to create conferences with three or more participants

Challenge:

- Timing considerations are important because audio and video conversational applications are highly **delay-sensitive**
- For voice, delays smaller than 150 milliseconds are not perceived by a human listener, delays between 150 and 400 milliseconds can be acceptable, and delays exceeding 400 milliseconds can result in frustrating

Loss-tolerant

- conversational multimedia applications are **loss-tolerant**
- These losses can often be partially or fully concealed
- These delay-sensitive but loss-tolerant characteristics are clearly different from those of elastic data applications such as Web browsing, e-mail, social networks, and remote login

Streaming Live Audio and Video

- These applications allow a user to receive a *live* radio or television transmission—such as a live sporting event or an ongoing news event—transmitted from any corner of the world
- Because the event is **live**, **delay can also be an issue,** although the timing constraints are much less stringent than those for conversational voice

Streaming Stored Video

- prerecorded videos are placed on servers, and users send requests to these servers to view the videos on demand
- interact with the video by pausing or repositioning to a future or past scene.

Streaming video systems

- Streaming video systems can be classified into three categories:
- **UDP streaming**
- **HTTP streaming**
- **adaptive HTTP streaming**

Streaming video systems

- A common characteristic of all three forms of video:
  - Extensive use of **client-side application buffering** to mitigate(کاهش دادن) the effects of varying end-to-end delays and varying amounts of available bandwidth between server and client

client buffering: advantages

1. client side buffering can absorb variations in server-to-client delay
2. if the server-to-client bandwidth briefly drops below the video consumption rate, a user can continue to enjoy continuous playback

client-side buffering:

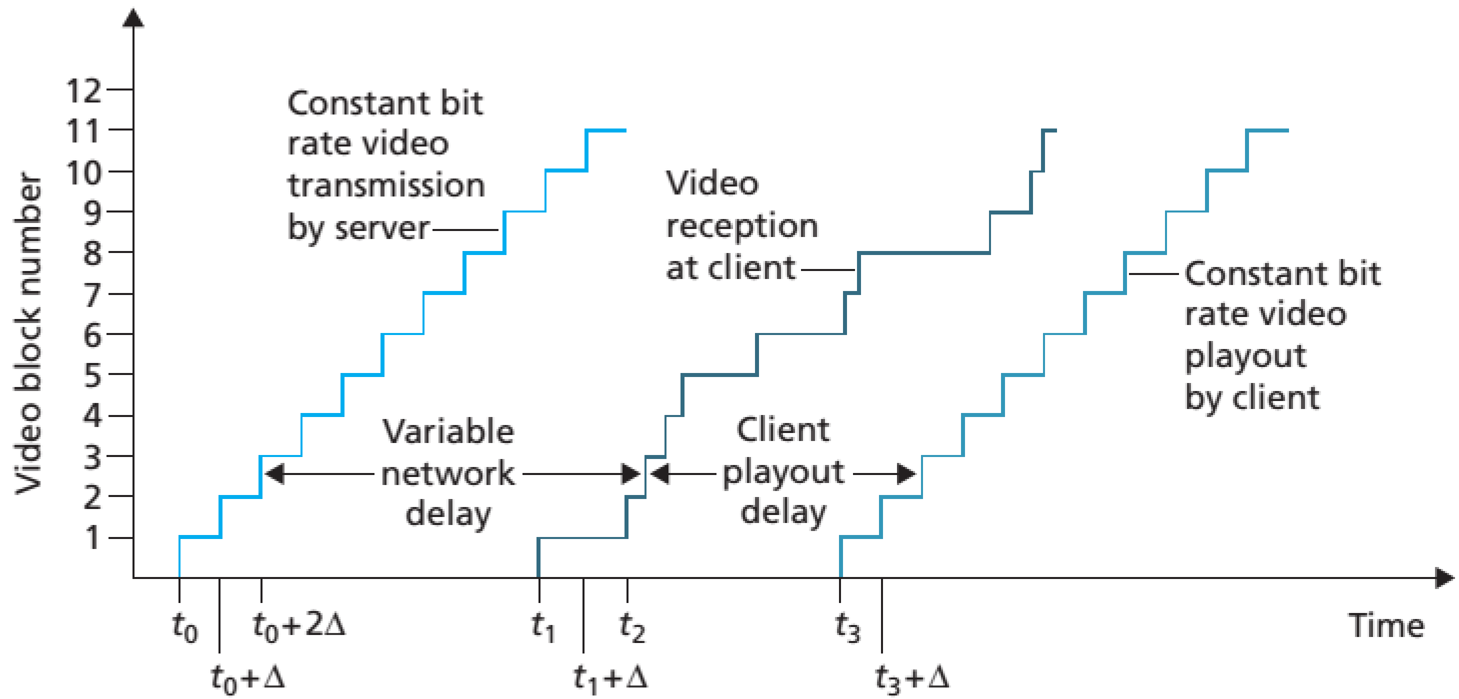


Figure 7.1 ♦ Client playout delay in video streaming

UDP Streaming:

- With UDP streaming, the server transmits video at a rate that matches the client's video consumption rate by clocking out the video chunks over UDP at a steady rate
- UDP streaming typically uses a small client-side buffer

Example

- Video consumption rate is 2 Mbps
- each UDP packet carries 8,000 bits of video

So,

- the server would transmit one UDP packet into its socket every (8000 bits)/(2 Mbps) = 4 msec

RTP

- It’s Completed Before passing the video chunks to UDP
- the server will encapsulate the video chunks within transport packets specially designed for transporting audio and video, using the Real-Time Transport Protocol (RTP)

three significant drawbacks of UDP:

1. due to the unpredictable and varying amount of available bandwidth between server and client, constant-rate UDP streaming can fail to provide continuous playout
2. it requires a media control server, such as an RTSP server, to process client-to-server interactivity requests and to track client state
3. many firewalls are configured to block UDP traffic, preventing the users behind these firewalls from receiving UDP video

HTTP Streaming:

- the video is simply stored in an HTTP server as an ordinary file with a specific URL.
- the client establishes a TCP connection with the server and issues an HTTP GET request for that URL
- The use of HTTP over TCP also allows the video to traverse firewalls and NATs more easily

Prefetching Video:

- client-side buffering can be used to mitigate the effects of varying end-to-end delays and varying available bandwidth

- for streaming *stored* video, the client can attempt to download the video at a rate *higher* than the consumption rate, thereby **prefetching** video frames that are to be consumed in the future

Client Application Buffer and TCP Buffers

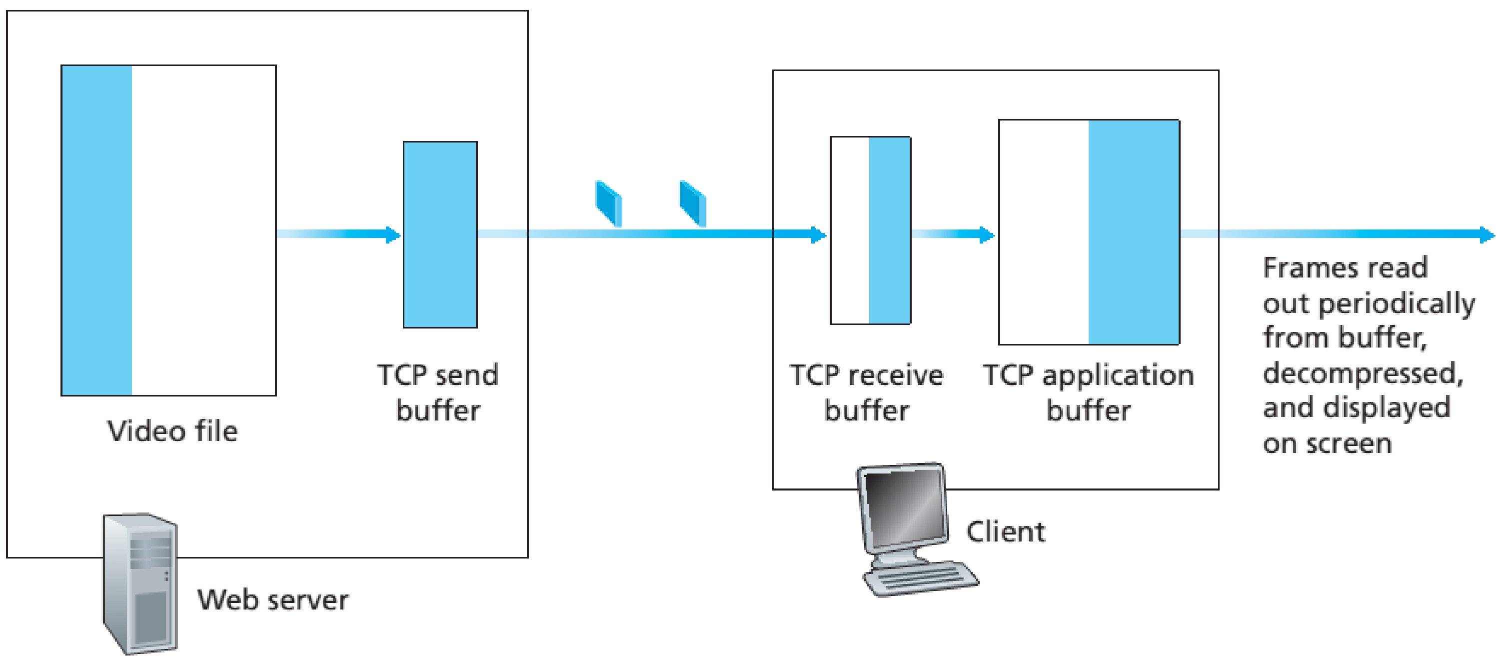


Figure 7.2 ♦ Streaming stored video over HTTP/TCP

Client Application Buffer

- Note that if the client application buffer is larger than the video file, then the whole process of moving bytes from the server's storage to the client's application buffer is equivalent to an ordinary file download over HTTP

Client Application Buffer

- During the pause period, bits are not removed from the client application buffer, even though bits continue to enter the buffer from the server
- once the client application buffer becomes full, bytes can no longer be removed from the client TCP receive buffer, so it too becomes full
- **if the user pauses the video, the server may be forced to stop transmitting**

Adaptive Streaming and DASH

- major shortcoming: All clients receive the **same encoding** of the video, despite the large variations in the amount of bandwidth available to a client, both across **different clients** and also **over time for the same client**
- Led to development of **Dynamic Adaptive Streaming over HTTP (DASH)**

Adaptive Streaming and DASH

- In DASH, the video is encoded into several different versions, with each version having a different bit rate and, correspondingly, a different quality level
- When the amount of available bandwidth is high, the client naturally selects chunks from a high-rate version; and when the available bandwidth is low, it naturally selects from a low-rate version

Adaptive Streaming and DASH

- DASH allows a client to adapt to the available bandwidth if the end-to-end bandwidth changes during the session
  - This feature is particularly important for mobile users
  - Comcast Case Study(8 to 10 different MPEG-4)

Adaptive Streaming and DASH

- With DASH, each video version is stored in the HTTP server, each with a different URL
- The HTTP server also has a **manifest file**, which provides a URL for each version along with its bit rate
  - requests the manifest file and learns about the various versions
  - selects one chunk at a time by specifying a URL

Adaptive Streaming and DASH

- While downloading chunks, the client also measures the received bandwidth and runs a *rate determination algorithm* to select the chunk to request next
  - Rate determination algorithm???!!
  - **a lot of video buffered** and **receive bandwidth is high** then choosing a chunk from a high-rate version and vise versa

Adaptive Streaming and DASH

- Server-Side Scalability: dynamically monitoring the available bandwidth and client buffer level, and adjusting the transmission rate with version switching
  - Achieve continuous playout at the best possible quality level