Computer Networks and Applications

COMP 3331/COMP 9331

Week-11 (Multimedia Networking)
Chap 9, Sections 9.1 and 9.2

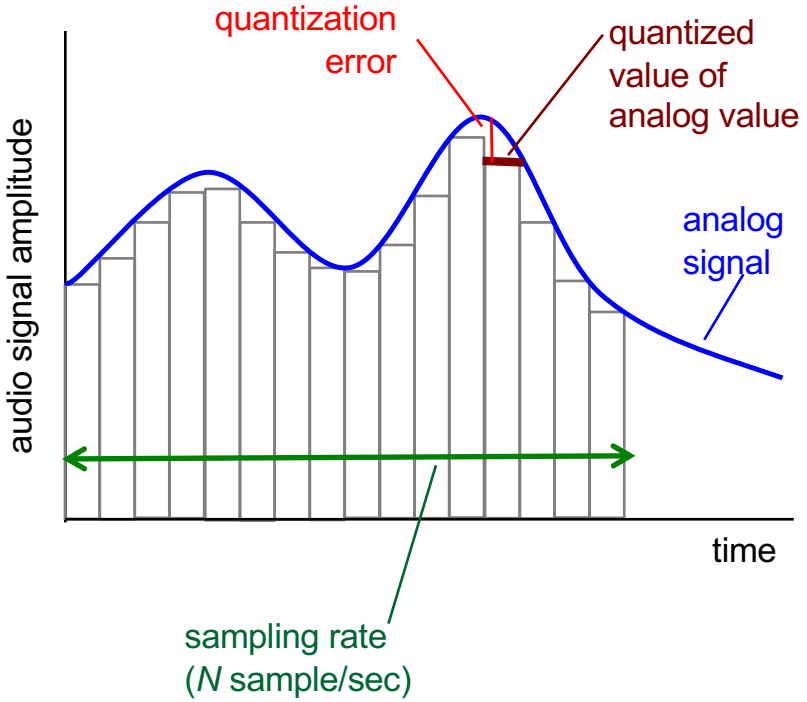
Multimedia networking: outline

9.1 multimedia networking applications

9.2 streaming *stored* video

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

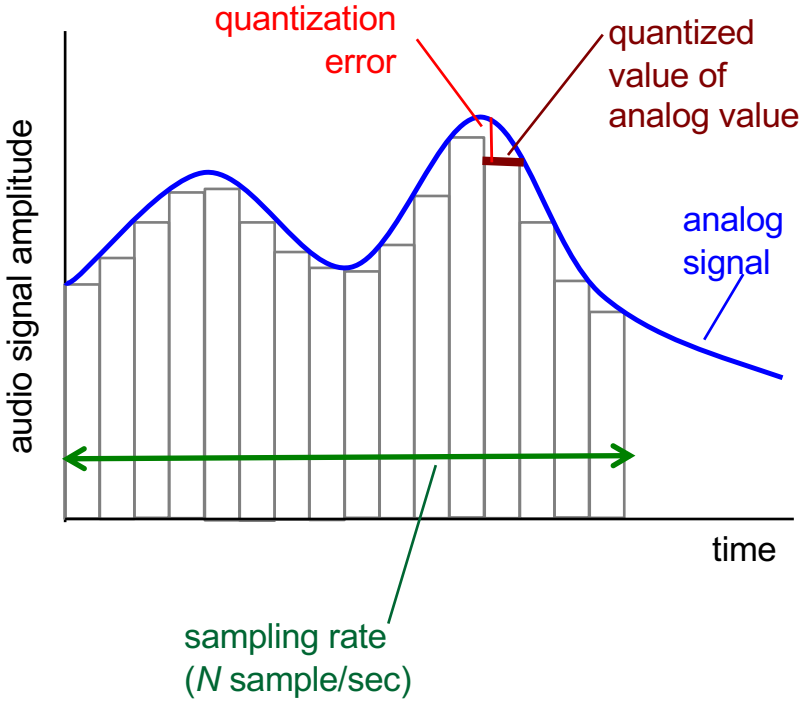


Multimedia: audio

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

example rates

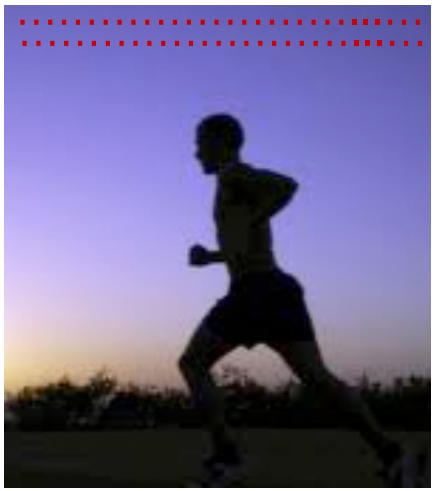
- CD: 1.411 Mbps
- MP3: 96, 128, 160 kbps
- Internet telephony: 5.3 kbps and up



Multimedia: video

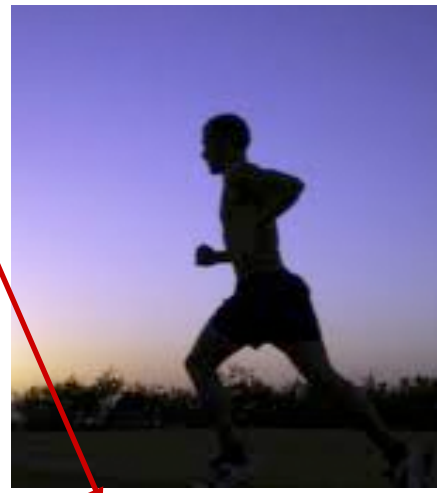
- video: sequence of images displayed at constant rate
  - e.g. 24 images/sec
- digital image: array of pixels
  - each pixel represented by bits
- coding: use redundancy *within* and *between* images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

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

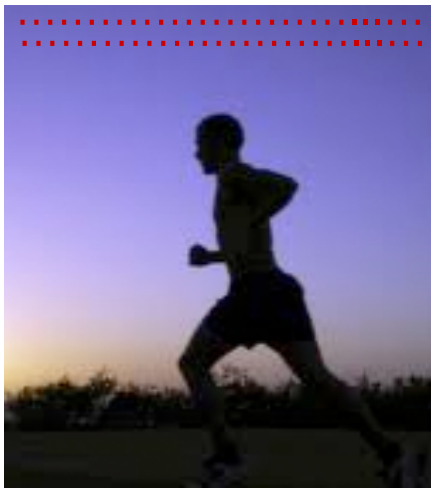


frame *i+1*

Multimedia: video

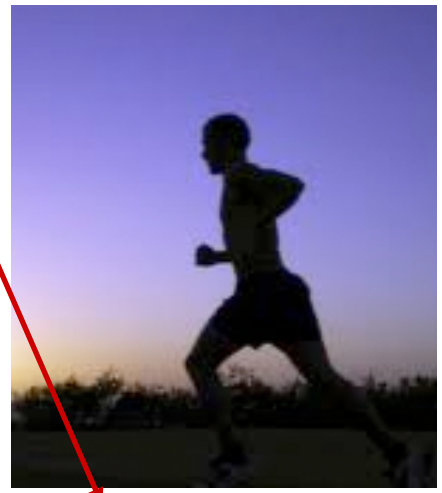
- *CBR: (constant bit rate):* video encoding rate fixed
- *VBR: (variable bit rate):* video encoding rate changes as amount of spatial, temporal coding changes
- *examples:*
  - MPEG 1 (CD-ROM) 1.5 Mbps
  - MPEG2 (DVD) 3-6 Mbps
  - MPEG4 (often used in Internet, < 1 Mbps)

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

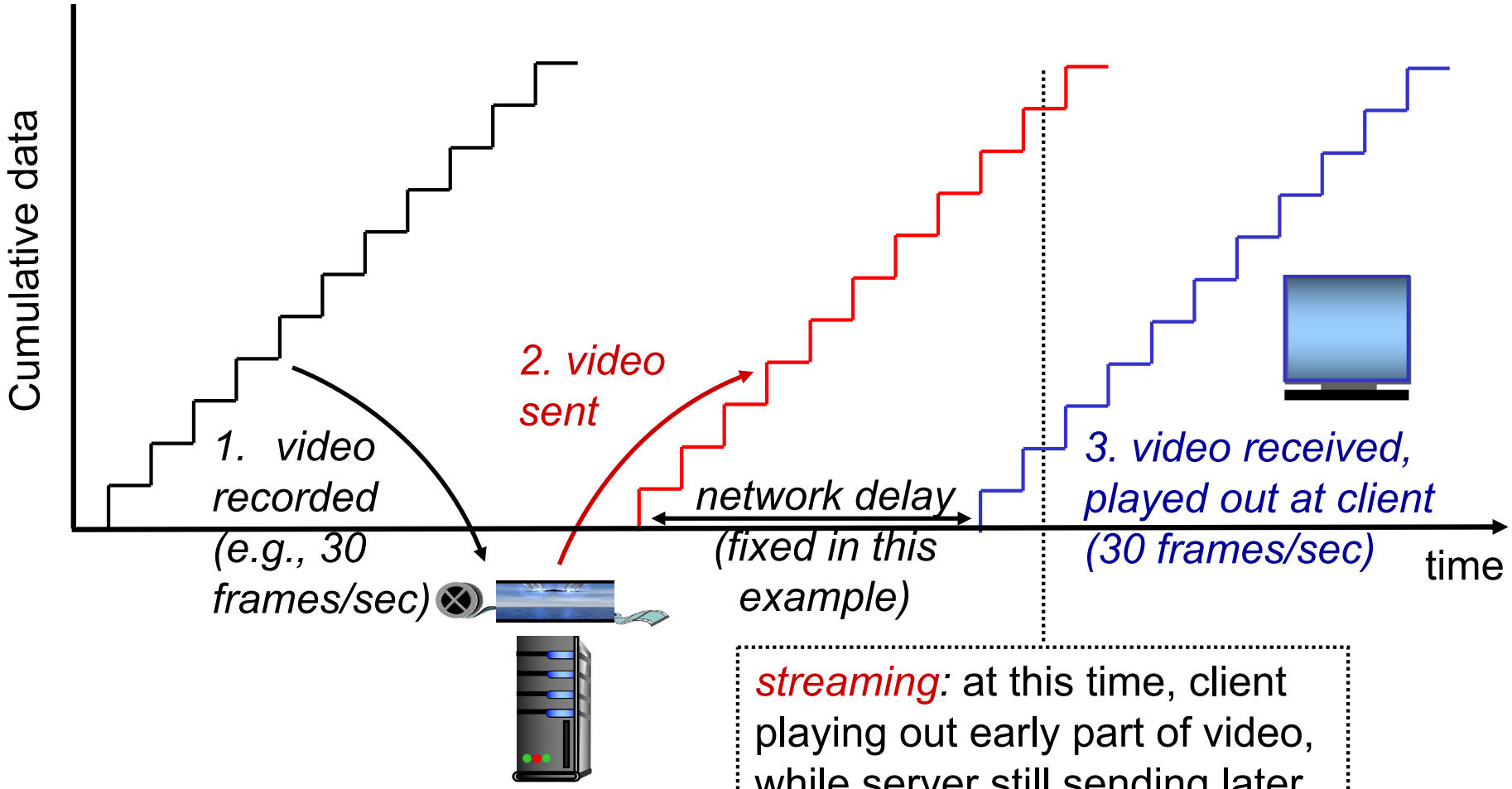
- *streaming, stored* audio, video
  - *streaming:* can begin playout before downloading entire file
  - *stored (at server):* can transmit faster than audio/video will be rendered (implies storing/buffering at client)
  - e.g., YouTube, Netflix, Hulu
- *conversational* voice/video over IP [not covered]
  - interactive nature of human-to-human conversation limits delay tolerance
  - e.g., Skype
- *streaming live* audio, video [not covered]
  - e.g., live sporting event (futbol)

Multimedia networking: outline

9.1 multimedia networking applications

9.2 streaming *stored* video

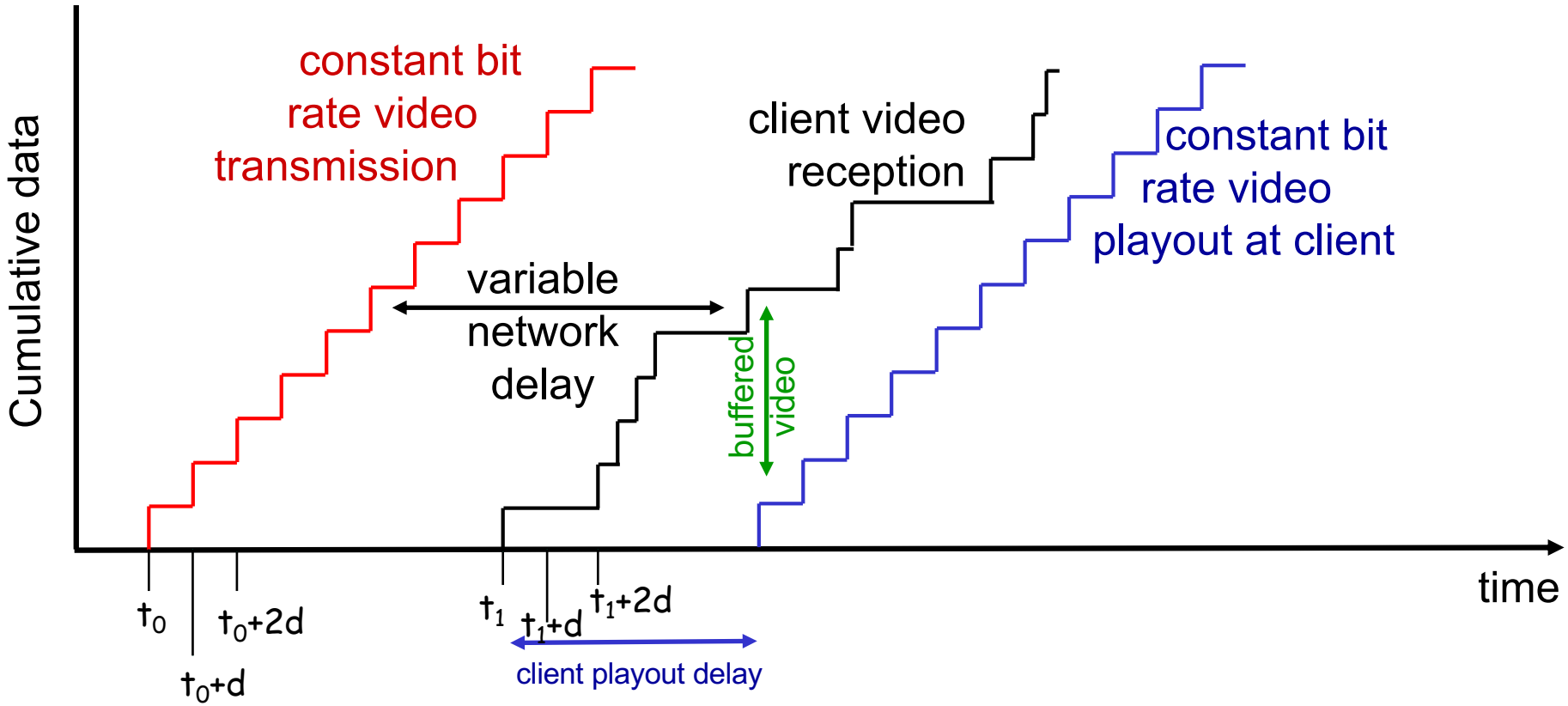
Streaming stored video:



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

Streaming stored video: revisted



- *client-side buffering and playout delay:* compensate for network-added delay, delay jitter

Exercise (based on previous slide)

- Q1: how many video blocks had arrived (buffered) when the client started playout?
- A1: blocks 1-6 had arrived and buffered

- Q2: If the client started playout at t_1 (as soon as the first block arrived), could it play block 2 in time?
- A2: No. Block 2 had to be played out at t_1+d, but it arrived at t_1+2d (video would freeze)

Summary

- Video has strict timing requirements for playout
- Videos are often segmented into chunks or blocks:
  - Each block is about 2-10 seconds long (many frames in each block)
  - Each block has to start playing at strict timing intervals for smooth video
- Streaming video over networks would have no problem if network had constant delay
- Network delay is variable: challenge for video streaming
  - Video frames or blocks may be delayed for playing (*video freezing* effect)
- Stored video streaming solution:
  - Playout delay at client to absorb delay variations
  - Some blocks/frames are initially buffered before the playout of the video starts