CS 352
Multimedia Data Representations

Lecture 6.1

http://www.cs.rutgers.edu/~sn624/352

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Multimedia networking

- Many applications on the Internet use audio or video
- IP video traffic will be 82 percent of all IP traffic […] by 2022, up from 75 percent in 2017
- CCTV traffic over the Internet will increase sevenfold between 2017 to 2022
- Internet video to TV will increase threefold between 2017 to 2022.
- Consumer Video-on-Demand (VoD) traffic will nearly double by 2022

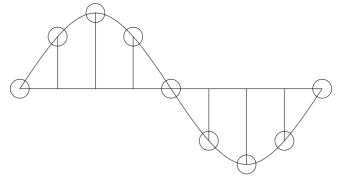
Source: Cisco visual networking index 2017--22

What's different about these applications?

- Traditional applications (HTTP(S), SMTP)
  - Delay tolerant but not loss tolerant
  - Data used *after* transfer complete
- Multimedia applications are often real time
  - Data delivery time *during transfer* matters for user experience
- Video/audio streaming
  - Delay-sensitive
- Real-time audio and video
  - Delays > 400 ms for audio is a bad user experience
  - Somewhat loss tolerant

Digital representation of audio and video

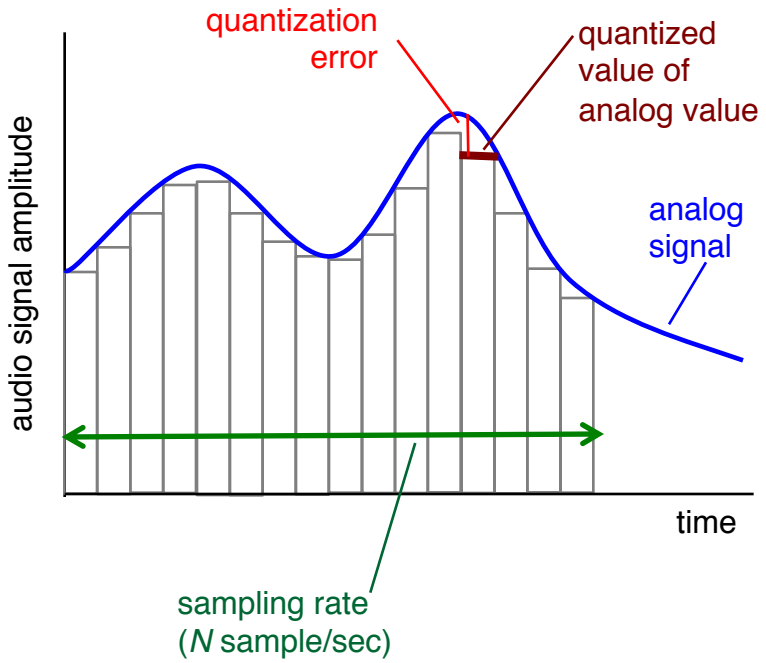
Digital representation of audio



- Must convert analog signal to digital representation
- Sample
  - How many times (twice the max frequency in the signal)
- Quantize
  - How many levels or bits to represent each sample
  - More levels ➔ more accuracy
  - More levels ➔ more bits to store & more bandwidth to transmit
- Compress
  - Compact representation of quantized values

Audio representation

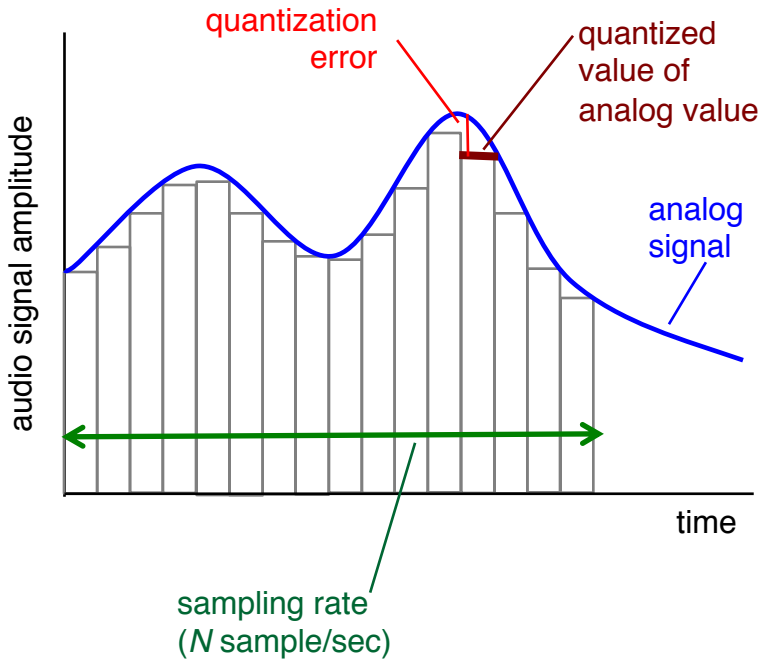
- analog audio signal sampled at constant rate
  - telephone: 8,000 samples/sec
  - CD music: 44,100 samples/sec
- each sample quantized, i.e., rounded
  - e.g., 2^8=256 possible quantized values
  - each quantized value represented by bits, e.g., 8 bits for 256 values



Audio representation

- example: 8,000 samples/sec, 256 quantized values
- Bandwidth needed: 64,000 bps

- receiver converts bits back to analog signal:
  - some quality reduction

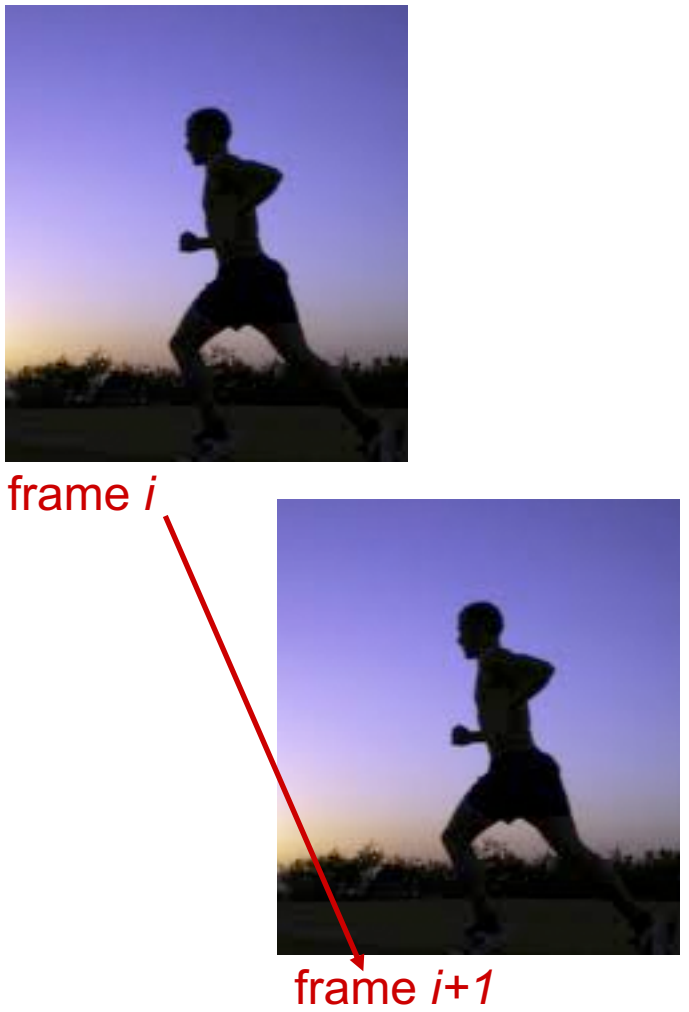


Example rates

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

Video representation

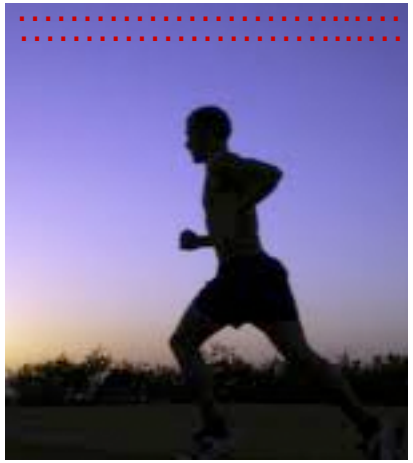
- Video: sequence of images displayed at constant rate
  - e.g., 30 images/sec
  - Appear continuous due to “persistence of vision”/stroboscopic effect.



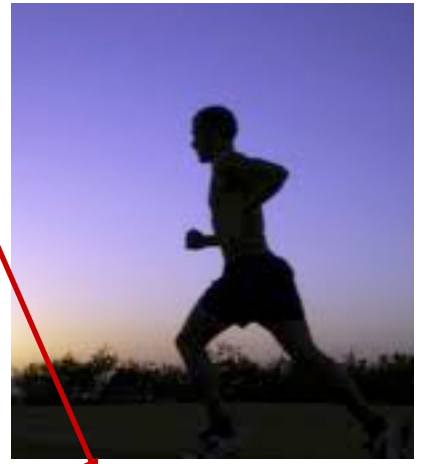
Video representation

- 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)
- Coding/decoding algorithm often called a codec

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*



frame *i+1*

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

Video representation

- Video *bit rate*: effective number of bits per second of the video after encoding
  - Depends on quality of each image. More pixels, more detail per pixel = more bits
  - Depends on effectiveness of compression in the codec
- Video bit rate is typically correlated with quality.
- CBR: (constant bit rate): fixed bit-rate video
- VBR: (variable bit rate): different parts of the video have different bit rates, e.g., changes in color, motion, etc.
- Examples of average video bit-rates:
  - MPEG 1 (CD-ROM) 1.5 Mbps. MPEG2 (DVD) 3-6 Mbps
  - MPEG4 (often used in Internet, < 1 Mbps)
  - In general, one Internet video stream takes up a few Mbit/s (that's it.)

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., Spotify, YouTube, Netflix
- *conversational* voice/video over IP
  - interactive nature of human-to-human conversation limits delay tolerance
  - e.g., Zoom
- *streaming live* audio, video
  - e.g., live sporting event (e.g., superbowl)

CS 352
Video Streaming

Lecture 6.2

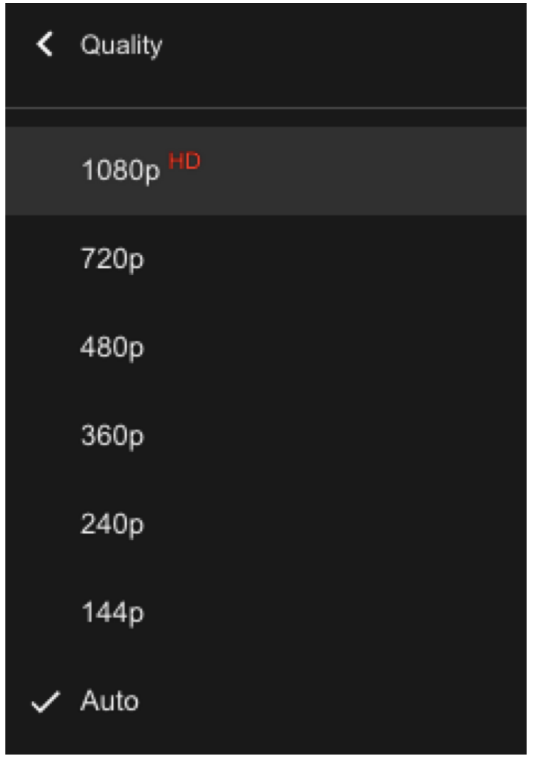
http://www.cs.rutgers.edu/~sn624/352

Srinivas Narayana

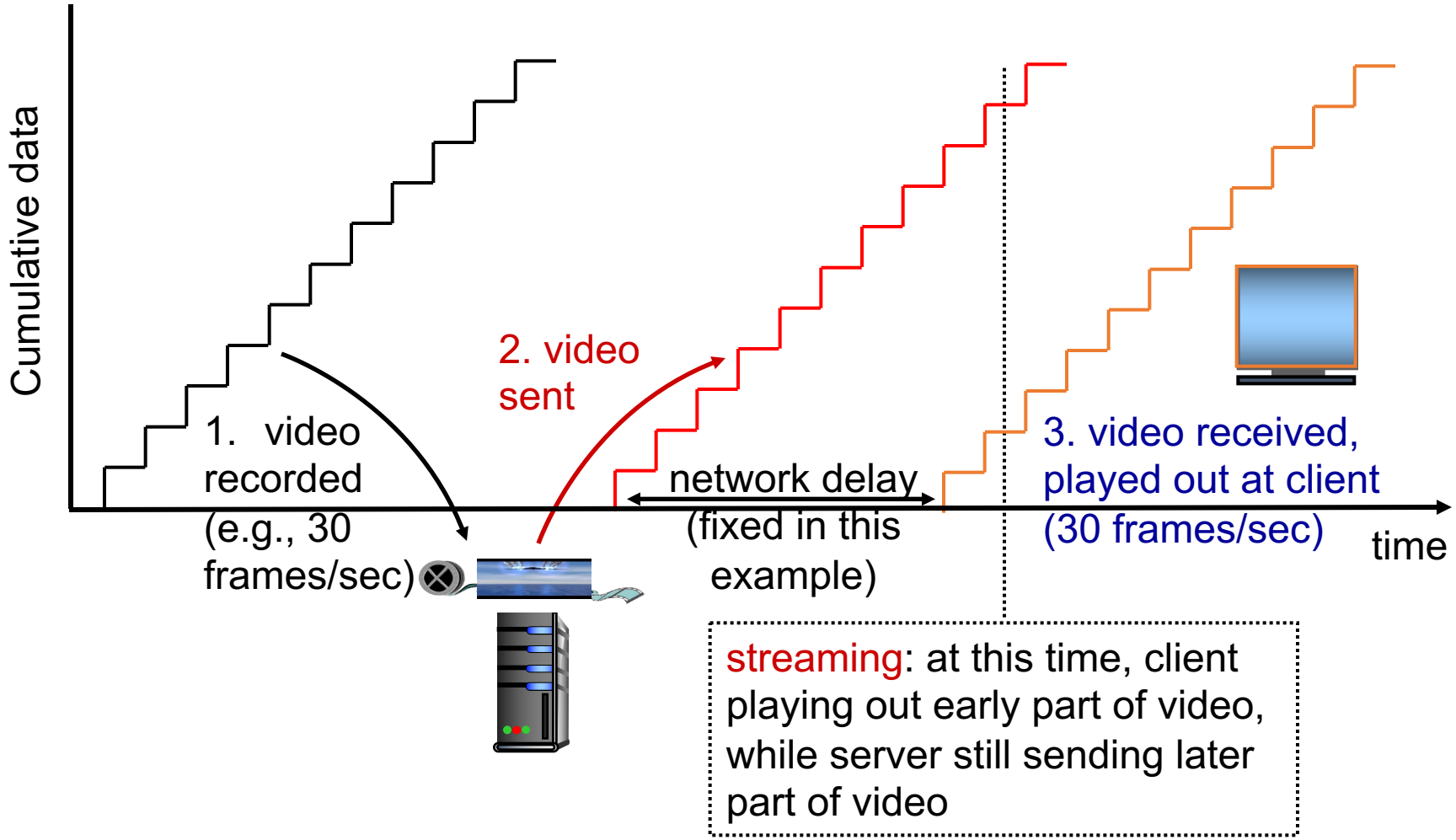
RUTGERS
UNIVERSITY | NEW BRUNSWICK

Streaming stored video

- Media is prerecorded at different qualities
- Client downloads an initial portion and starts viewing
- Rest downloaded as time progresses
- No need to wait for entire content to be downloaded
- Can change content sites mid-stream based on network conditions



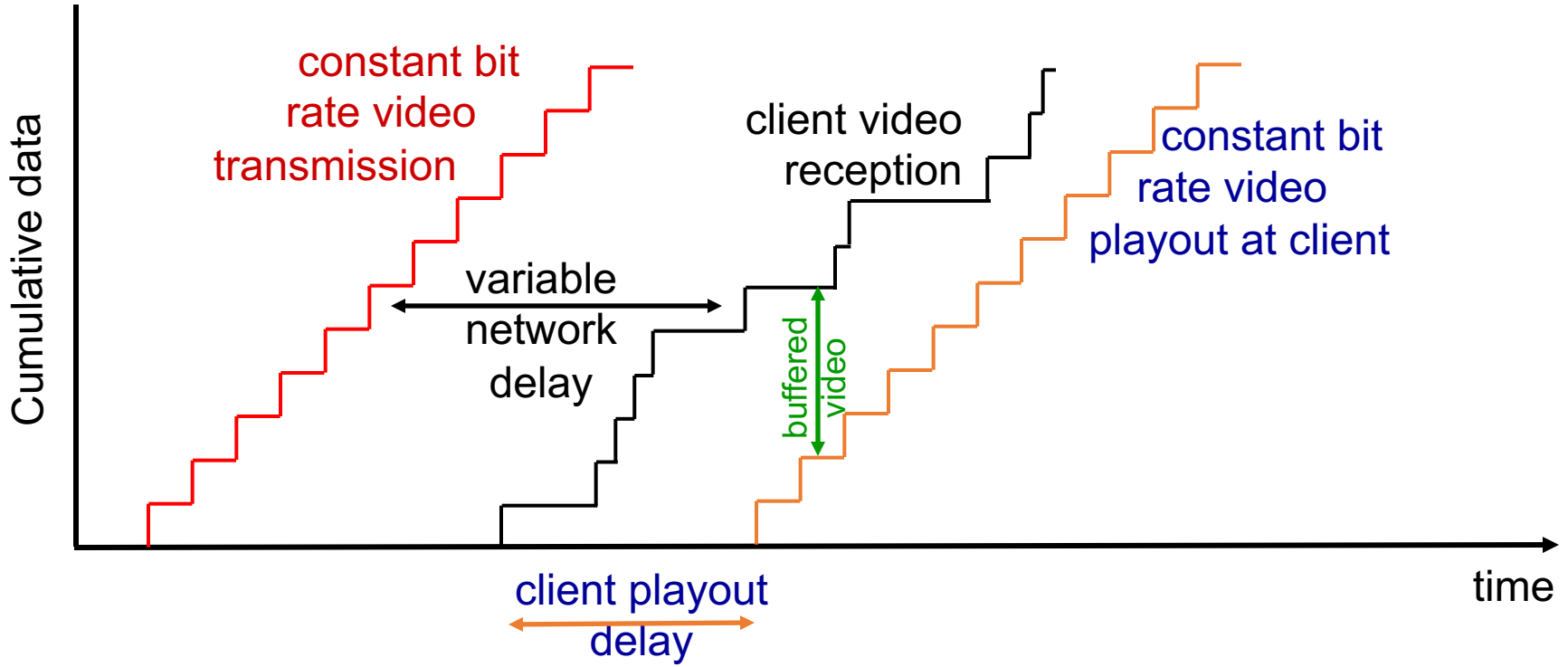
Streaming stored video



Streaming stored video: challenges

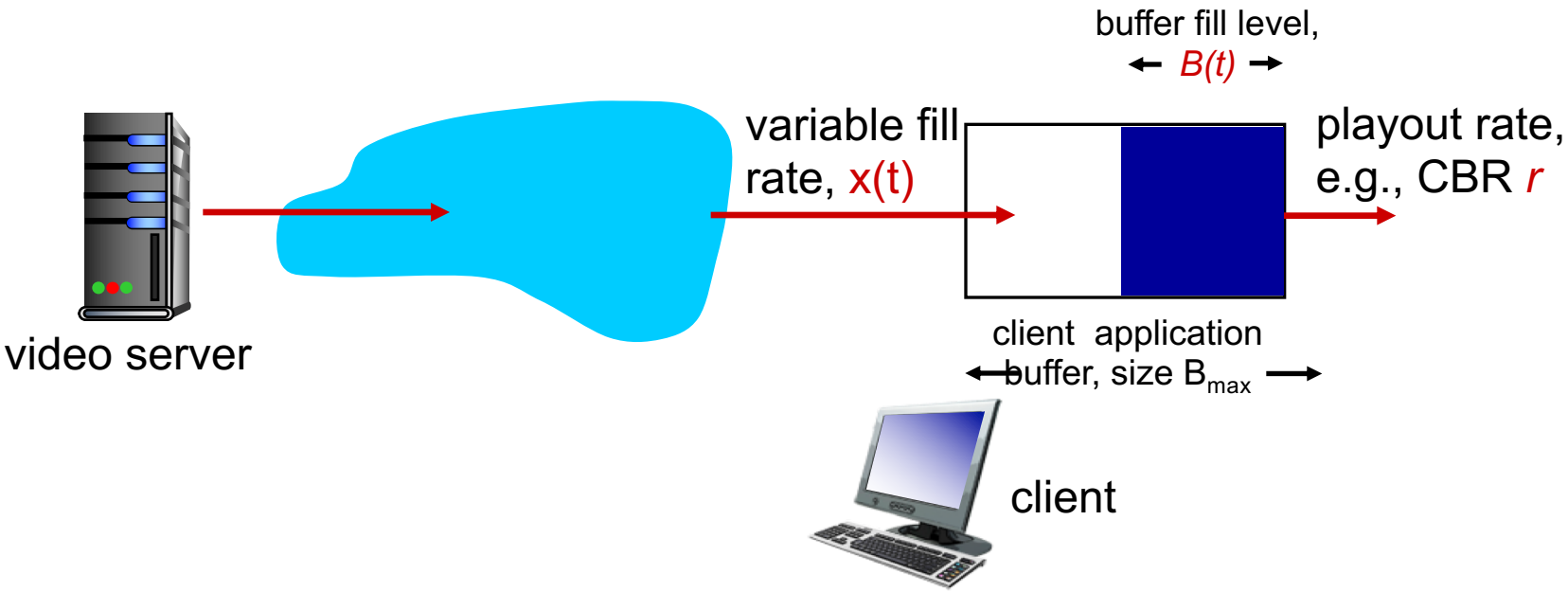
- continuous playout constraint: once client playout begins, playback must match original timing
  - … but network delays are variable (jitter), so will need client-side buffer to match playout requirements
- other challenges:
  - client interactivity: pause, fast-forward, rewind, jump through video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

Scenario 1: Constant bit-rate video



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

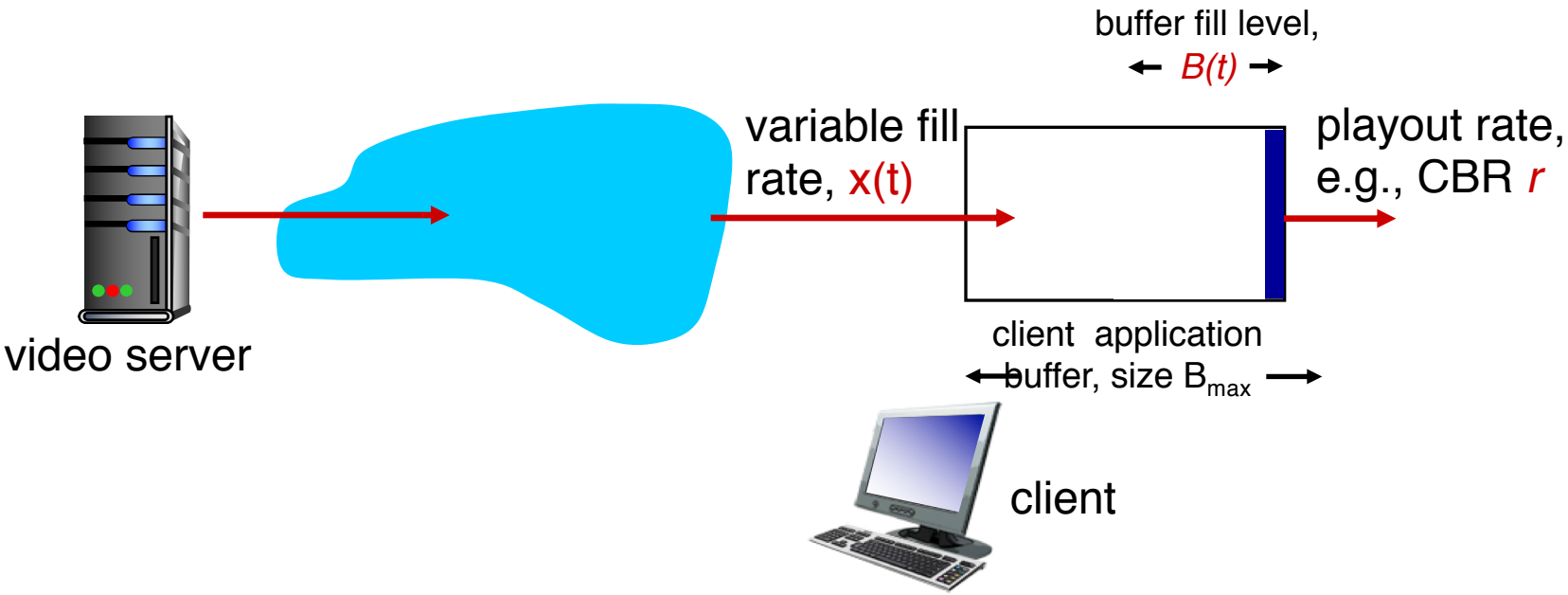
Client-side buffering, playout



Video is downloaded chunk by chunk (typically using the HTTP protocol)

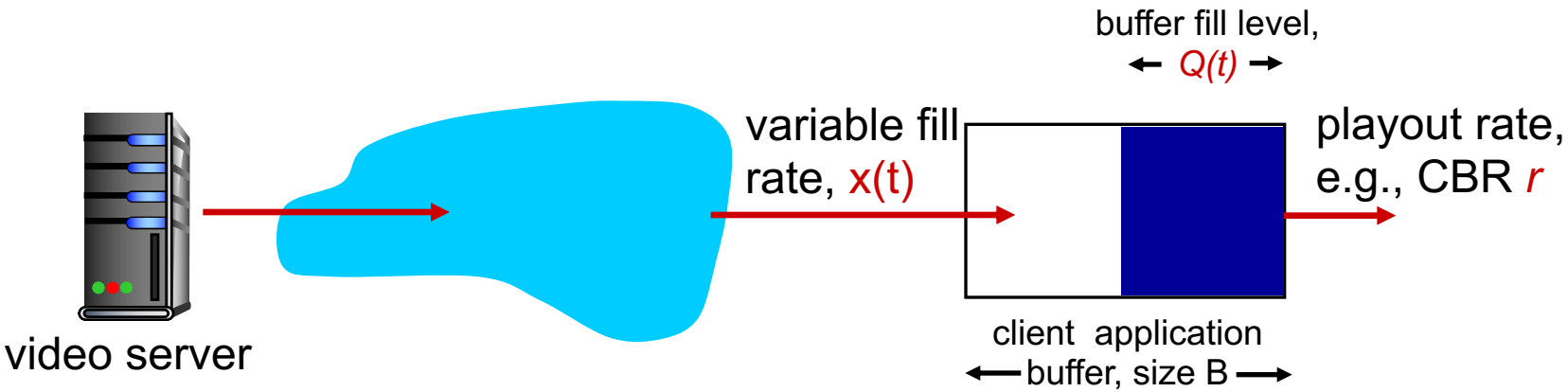
For example: a chunk might be 4 seconds worth of video.

Client-side buffering, playout

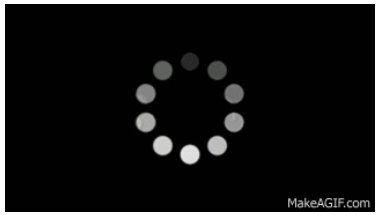


1. Initial fill of buffer until playout begins at t_p
2. playout begins at t_p,
3. buffer fill level varies over time as fill rate x(t) varies and playout rate r is constant

Client-side buffering, playout

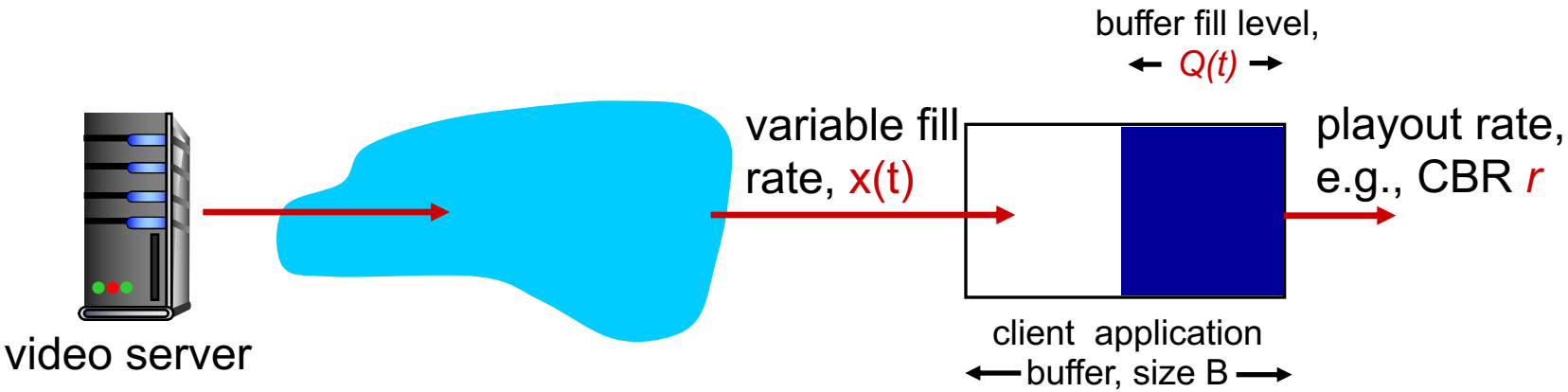


playout buffering: average fill rate ($\bar{x}$), playout rate (r):



- $\bar{x} < r$: buffer eventually empties (causing freezing of video playout until next chunk downloaded). Rebuffering event
- $\bar{x} > r$: buffer will not empty, provided initial playout delay is large enough to absorb variability in x(t)
  - *initial playout delay tradeoff:* buffer starvation less likely with larger delay, but larger delay until user begins watching

Client-side buffering, playout



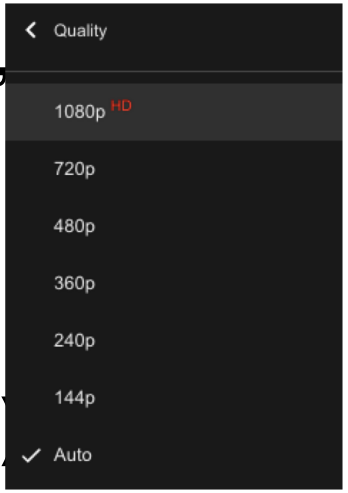
playout buffering: average fill rate ($\bar{x}$), playout rate (r):

- is $\bar{x} < r$ or $\bar{x} > r$ for a given network connection?

- It is hard to predict this in general. How to set r?
- Too low a bit-rate r: video has poorer quality than needed
- Too high a bit-rate r: buffer might empty out. Rebuffering!

Scenario 2: Adaptive bit–rate video

- Motivation: Want to provide high quality video experience, without rebuffering
- Adapt bit rate collaboratively between the video client (e.g., YouTube player on your browser) and the server
- Adaptive bit-rate (ABR) video: change the bit-rate (quality) of next chunk, based on network and client conditions.
- A typical strategy: Buffer-based rate adaptation

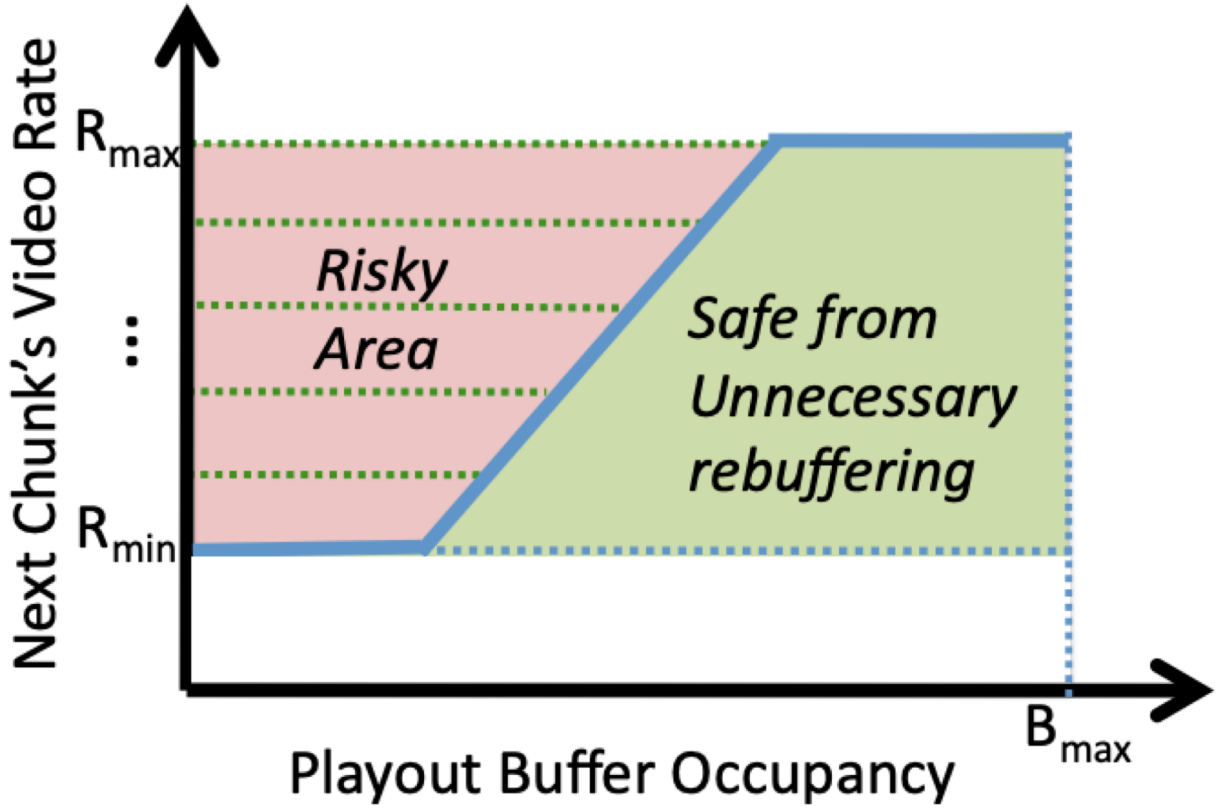


Buffer-based Rate Adaptation

- Key idea: If there is a large stored buffer of video, optimize aggressively for video quality, i.e., high bit rates

- Conversely, if there is a small stored buffer of video, be conservative and ask for a lower quality
  - The hope is that the lower bandwidth requirement can be satisfied by the connection more easily.

Buffer-based bit-rate adaptation



A highly effective method to provide high video quality despite variable and intermittently poor network conditions.

Used by Netflix.

http://yuba.stanford.edu/~nickm/papers/sigcomm2014-video.pdf

A Buffer-Based Approach to Rate Adaptation