Multimedia Applications and Streaming Video

Daniel Zappala

CS 460 Computer Networking
Brigham Young University

Multimedia Applications

Video

- **high bit rate**
  - 100 kbps to 3 Mbps
  - predicted to be 90% of Internet traffic by 2015
- compression
  - 24 to 30 frames per second
  - spatial redundancy
  - temporal redundancy
- often create multiple versions at various bit rates (compression levels)
  - user or software chooses the best version

Audio

- significantly lower bandwidth
  - 14 kbps for speech
  - 128 kbps for music
- analog to digital
  - sample audio at a fixed rate, e.g. 8,000 samples per second
  - round each sample to a finite value, e.g. 8-bit audio yields 256 possible values
  - bit rate = samples per second x bit value (PCM = 8,000 x 256 = 64 kbps)
  - tradeoff between quality and bit rate/storage requirements
- compression: MP3, AAC

Multimedia Applications

- **streaming stored audio and video**
  - content pre-recorded and stored at server
  - user begins playback before entire file received
  - content played continuously, at same rate as original recording
  - user can pause, rewind, fast-forward, index content
- **conversational voice- and video-over-IP**
  - content is sent live, rather than pre-recorded, still continuous
  - delay-sensitive
  - delay $< 150ms$ not perceptible, $150 - 400ms$ acceptable, > 400 ms not tolerable
  - loss-tolerant
- **streaming live audio and video**
  - multiple users receiving simultaneously
  - application-layer multicast or multiple unicast streams from a CDN
  - delay of 10s of seconds from live are acceptable

Streaming Stored Video

Streaming Stored Video

- applications make the best out of best-effort service
- client buffering
  - download video into a buffer
  - video arrives at a variable rate (depending on available bandwidth)
  - play from buffer at a constant rate
  - guess a playback delay that prevents buffer from running out

UDP Streaming

- transmit at a fixed rate
  - no congestion control
  - simple to implement with a small client buffer
- drawbacks
  - dropped frames during congestion, with no retransmission
  - not fair to other Internet traffic
  - server must maintain state for each client to keep track of when video is paused or rewound
  - many firewalls block UDP traffic

HTTP Streaming

- overview
  - store video as a file
  - client fetches file as fast as it can with TCP
  - store in a buffer and then play at a continuous rate
- variable TCP rate
  - if less than video bit rate, then alternating between periods of continuous play and pauses for buffering
  - if greater than video bit rate, then continuous play with no interruptions
- early termination wastes bandwidth, so use small buffers

Adaptive HTTP Streaming

- DASH: Dynamic Adaptive Streaming over HTTP
- store video on server
  - divide into 2 second chunks
  - encode each chunk with multiple bit rates
- client downloads using HTTP GET requests
  - fetch a low quality version
  - if received in plenty of time, fetch higher quality next time
  - if not going to be received in time, abort and fetch lower quality
  - constantly adjust rate as congestion allows
- allows client to easily adapt to varying conditions – mobile to high-speed connectivity
- able to avoid freezing if adaptive algorithm is good
- able to use HTTP caches

Content Distribution Networks

- streaming from a massive data center is infeasible
  - clients may be far away, with a transcontinental bottleneck
  - wasted bandwidth as popular videos sent many times to different clients
  - single point of failure
- CDN
  - many servers, spread across geographically diverse areas
  - private (Google) or third-party (Akamai for Netflix and Hulu)
  - deep – locate servers in many ISPs close to users
  - bring home – locate servers near IXPs and POPs near many ISPs, connect them with high-speed private links

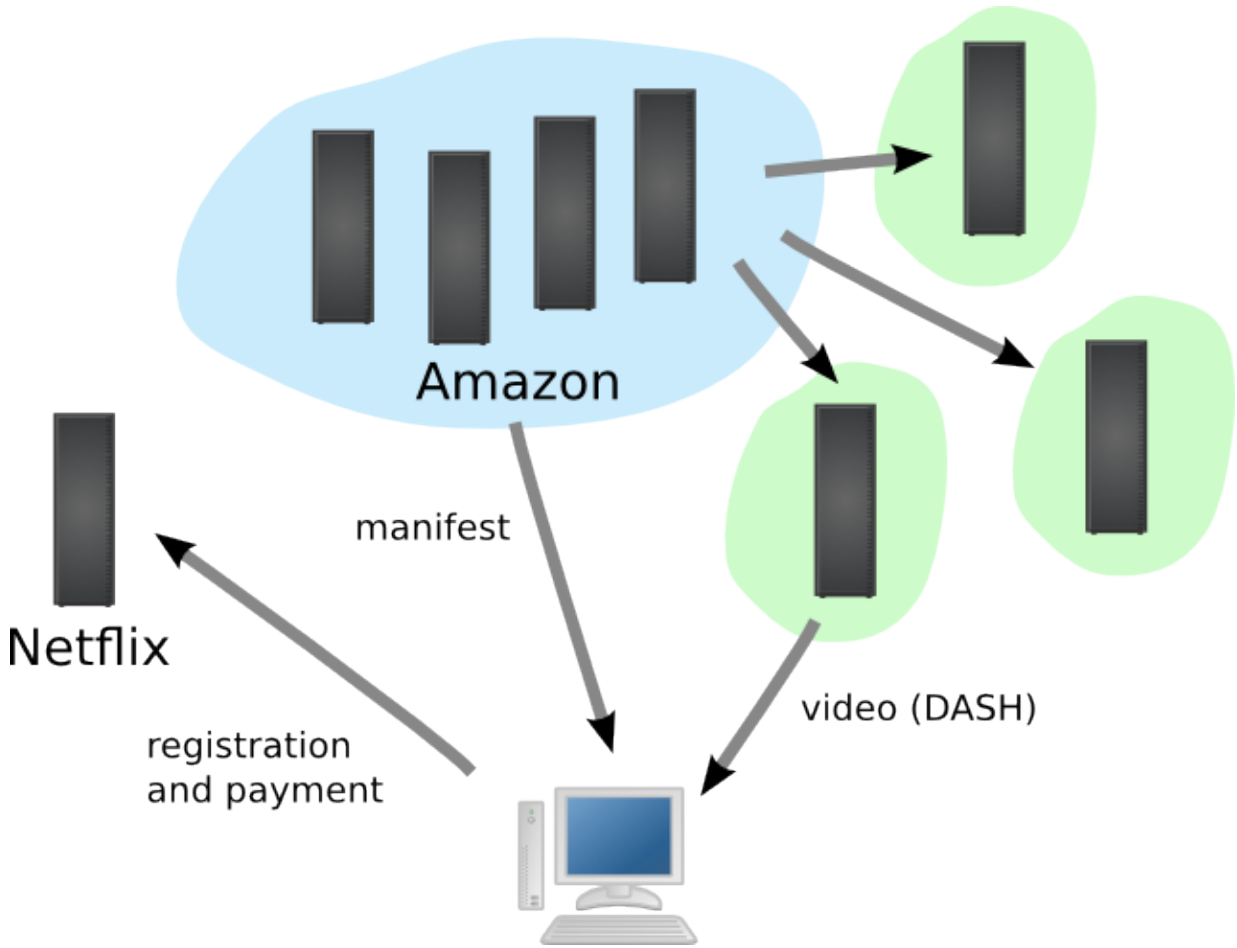
CDN Operation

- DNS
  - company places content on CDN servers
  - forwards DNS queries, e.g. video.company.com to server1.cdn.com
- cluster selection
  - CDN wants to forward client to nearest cluster
  - *geographical*: route to cluster geographically closest to the client IP – may not be lowest delay
  - *real-time measurements*: measure delay and loss between clusters and clients, collect and use for selection
  - *IP anycast*: give all CDN servers the same IP address and use BGP to route to closest one

Case Studies

Netflix

- rents CDN servers from a third party rather than building its own infrastructure



YouTube

- private CDN
- use DNS to redirect client to a cluster
  - usually smallest RTT between client and cluster
  - may be directed to more distant cluster for load balancing
  - may also be redirected if cluster doesn't have the file
- does not use adaptive streaming

Kankan

- P2P video distribution, in China
- similar to BitTorrent
  - contact tracker
  - download chunks of video from peers in parallel
  - focus on downloading chunks needed soon
- swarms of 10,000+ peers for popular videos
- UDP