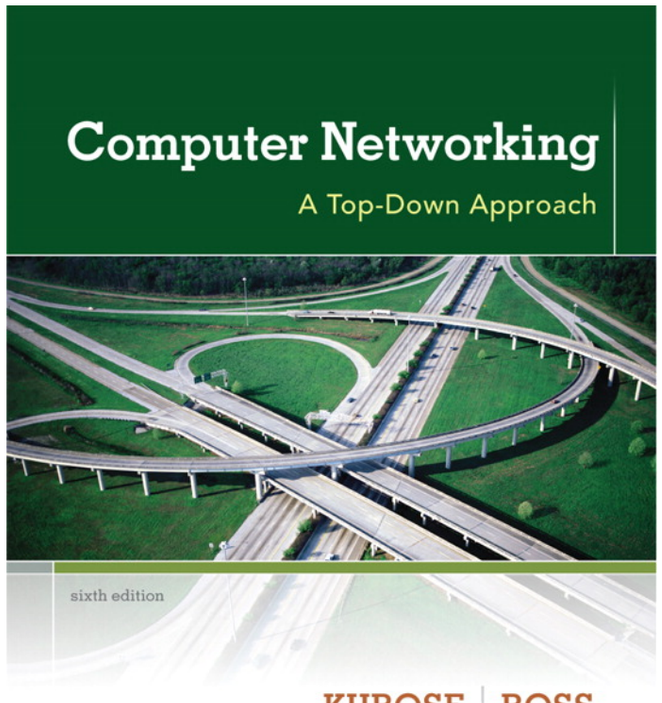
CN-Advanced
L37

MultiMedia Streaming

Dr. Ram P Rustagi
rprustagi@ksit.edu.in
http://www.rprustagi.com
https://www.youtube.com/rprustagi

Acknowledgements

Chapter 7
Multimedia Networking



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a lot of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley
March 2012

Multimedia Streaming

- Streaming stored audio/video
- Conversational audio/video
- Streaming live audio/video

- Streaming stored audio/video
  - UDP Streaming
  - HTTP Streaming
  - Adaptive HTTP Streaming

Streaming Multimedia: UDP

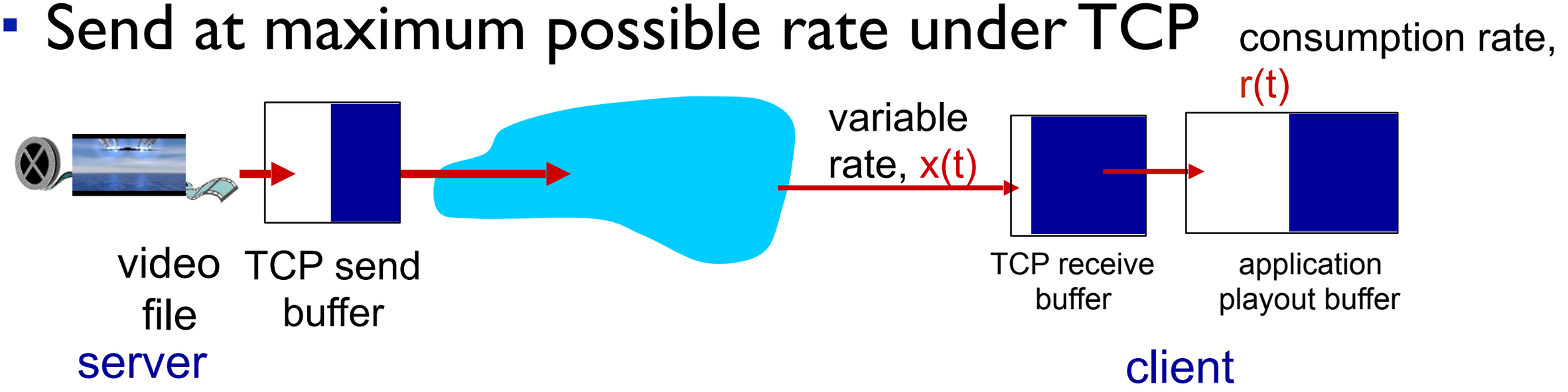
- Server sends at rate appropriate for client
  - Often: send rate = encoding rate = constant rate
  - Transmission rate is oblivious to congestion levels
- Short playout delay
  - (2-5 seconds) to remove network jitter
  - RTP [RFC 2326] protocol: multimedia payload types
- How to repond to user actions
  - Pause, Resume, Backward, Foward etc.
  - RTSP protocol to deal with client commands
  - Similar to FTP control channel
- Error recovery: application-level

UDP Streaming: Drawbacks

- Unpredictable and varying amount of available bandwidth
  - Fails to provide continuous playout
  - Example: video consumption rate is 2Mbps
    - Bandwdith is more than 2Mbps, but
    - Occasionally falls below 2 Mbps every few minutes
      - Leads to poor user experience
- Require a separate media (RTSP) server
  - To track client interactivity (video pause/play)
- UDP may *not* go through firewalls
  - Most firewalls block incoming UDP traffic

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
- Q: what happens when
  - x < r
  - x > r

Streaming Multimedia: HTTP

- Advantages (comparison) with UDP streaming
  - Repositioning supported by HTTP byte range
  - No need for separate media (RTSP) server
  - Firewalls in general (always) allow HTTP traffic
  - TCP Congestion control (retransmit)
    - perceived to be a blocker to performance
      - Obviated by client side buffering and prefetching
  - Experiments shows that when TCP throughput is approx twice the media consumption (play) rate
    - There is minimal starvation and buffering delays.
- Youtube initially started with HTTP Streaming

Streaming Multimedia: HTTP

- Drawbacks
  - When client jumps to a position ahead (future point)
  - Client sends HTTP byte-range header request
  - Server discards current data stream being pushed and starts sending data from indicated byte
  - Client application buffer B is full.
  - Future time $t > t_0 + B/r$
  - Entire buffer gets wasted
  - non-technical example: cooking extra food and wasting it

Streaming multimedia: DASH

- Major shortcoming of HTTP Streaming
  - All clients receive same encoding of video
  - Clients have different Internet access bandwidth
  - Client internet bandwidth varies over time.
  - HTTP Streaming has no provision for adjusting the quality
    - Poor viewing experience for a client
- Solution:
  - DASH: **D**ynamic, **A**daptive **S**treaming over **H**TTP

Streaming multimedia: DASH

- *Server:*
  - Divides video file into multiple chunks
  - Each chunk stored, encoded at different rates
  - Manifest file: provides URLs for different chunks
- *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)

Streaming multimedia: DASH

- *DASH: Dynamic, Adaptive Streaming over HTTP*
- *"intelligence"* at client: client determines
  - *when* to request chunk (so that buffer starvation, or overflow does not occur)
    - Smoothen the viewing by progressively requesting low quality encoding.
    - No sudden quality change
  - *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)
- Application: Netflix, Youtube, …

Content distribution networks

- *challenge:* how to stream a chosen content
  - src: from millions of videos
  - dst: millions of simultaneous users?

- *option 1:* single, large "mega-server"
  - single point of failure
  - point of network congestion
  - long path to distant clients
  - multiple copies of video sent over outgoing link
- ….quite simply: this solution *doesn't scale*

Content distribution networks

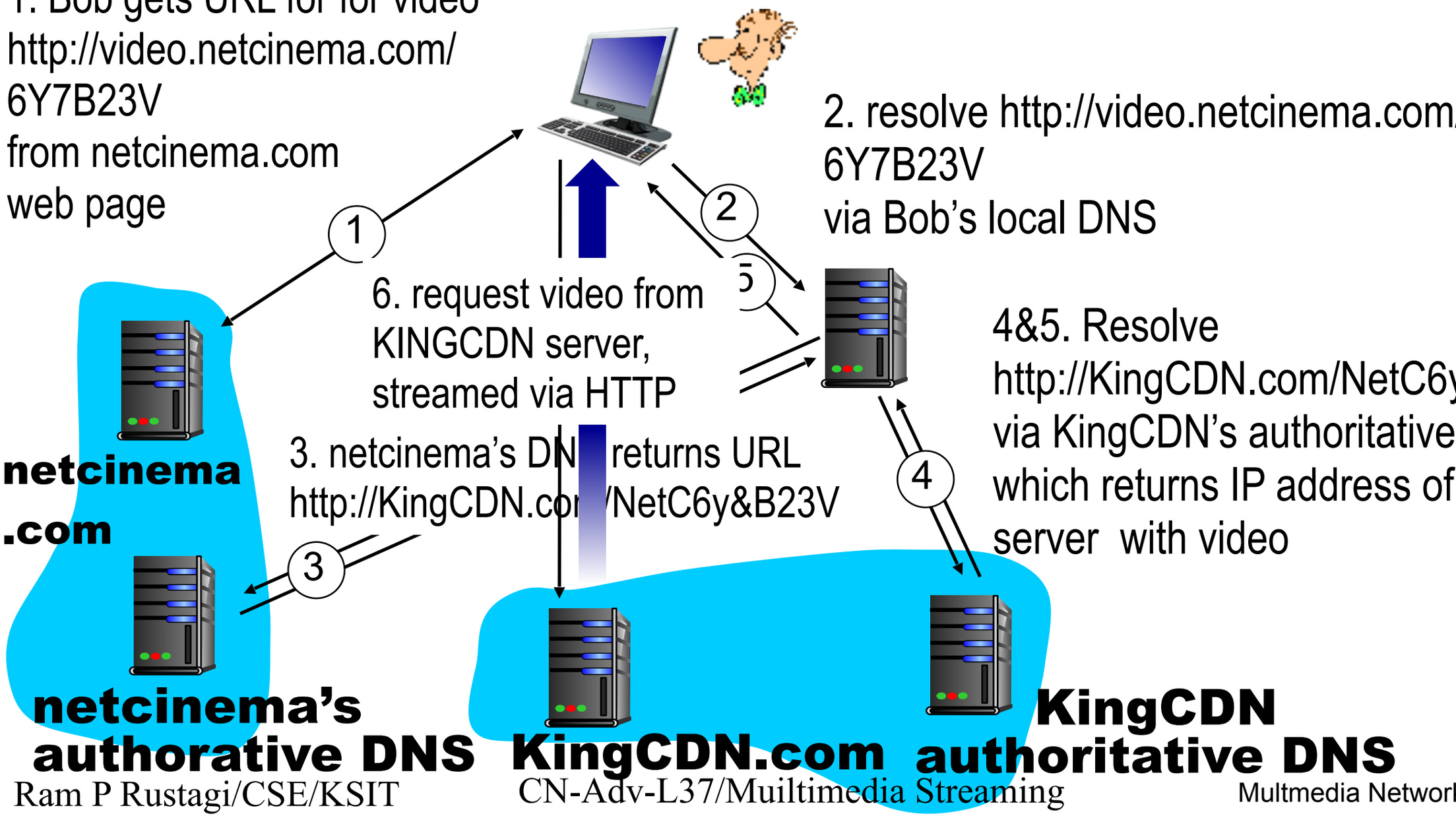
- *option 2:* store/serve multiple copies of videos at multiple geographically distributed sites *(CDN)*
  - ***enter deep***:
    - push CDN servers deep into many access networks
    - Close to users
      - Used by Akamai, 130 countries, 240K servers
      - https://www.akamai.com/us/en/what-we-do/world-class-digital-experiences.jsp?
- **Bring home:**
  - smaller number (10's) of larger clusters in POPs near (but not within) access networks
  - used by Limelight

CDN: "simple" content access scenario

Bob (client) requests video http://netcinema.com/6Y7B23V

- video stored in CDN at http://KingCDN.com/NetC6y&B23V

1. Bob gets URL for for video http://video.netcinema.com/6Y7B23V from netcinema.com web page



2. resolve http://video.netcinema.com/6Y7B23V via Bob's local DNS

4&5. Resolve http://KingCDN.com/NetC6y&B23 via KingCDN's authoritative DNS, which returns IP address of KlingCDN server with video

CDN cluster selection strategy

- *challenge:* how does CDN DNS select "good" CDN node to stream to client
  - pick CDN node geographically closest to client
  - pick CDN node with shortest delay (or min # hops) to client (CDN nodes periodically ping access ISPs, reporting results to CDN DNS)
    - IP anycast
- *alternative:* let *client* decide - give client a list of several CDN servers
  - client pings servers, picks "best"
  - Netflix approach

Case Stuty: Youtube

- YouTube stats:
  - src: https://merchdope.com/youtube-stats/
    - Number of people using youtube: 1.3B
    - Number of video hours uploaded: 300hrs/min
    - Number of videos watched per day: 5B
    - Number of visitors per day: 30 Million
- Started the service in Apr 2005
  - Acquired by google in Nov 2006
- Uses CDN technology extensively
  - Employes private CDN
  - Clusters in almost every ISP
  - Uses DNS Redirect to connect to specific cluster
    - based on lowest RTT between client and server

Case Stuty: Youtube

- When chosen cluster doesn't have the video
  - Either fetches the video from elsewhere
  - or, uses HTTP redirect
- Stream technology
  - Started with HTTP Streaming (as per book)
    - Requires user to select the version
    - Each version has a different bit rate and quality
  - Currently, uses DASH (for mobile devices)
  - Uses HTTP Byte-range header to limit BW waste
    - Limits the amount of video fetched
    - Next video is fetched next

YouTube IN

Search

Application Layer Overview

Dr. Ram P Rustagi
rprustagi@ksit.edu.in
http://www.rprustagi.com
https://www.youtube.com/rprustagi

Quality
720p HD
480p
360p
240p
144p
Auto

0:00 / 19:03

Summary

- Streaming multimedia
  - UDP
  - HTTP
  - DASH
- CDN
- Case study: YouTube