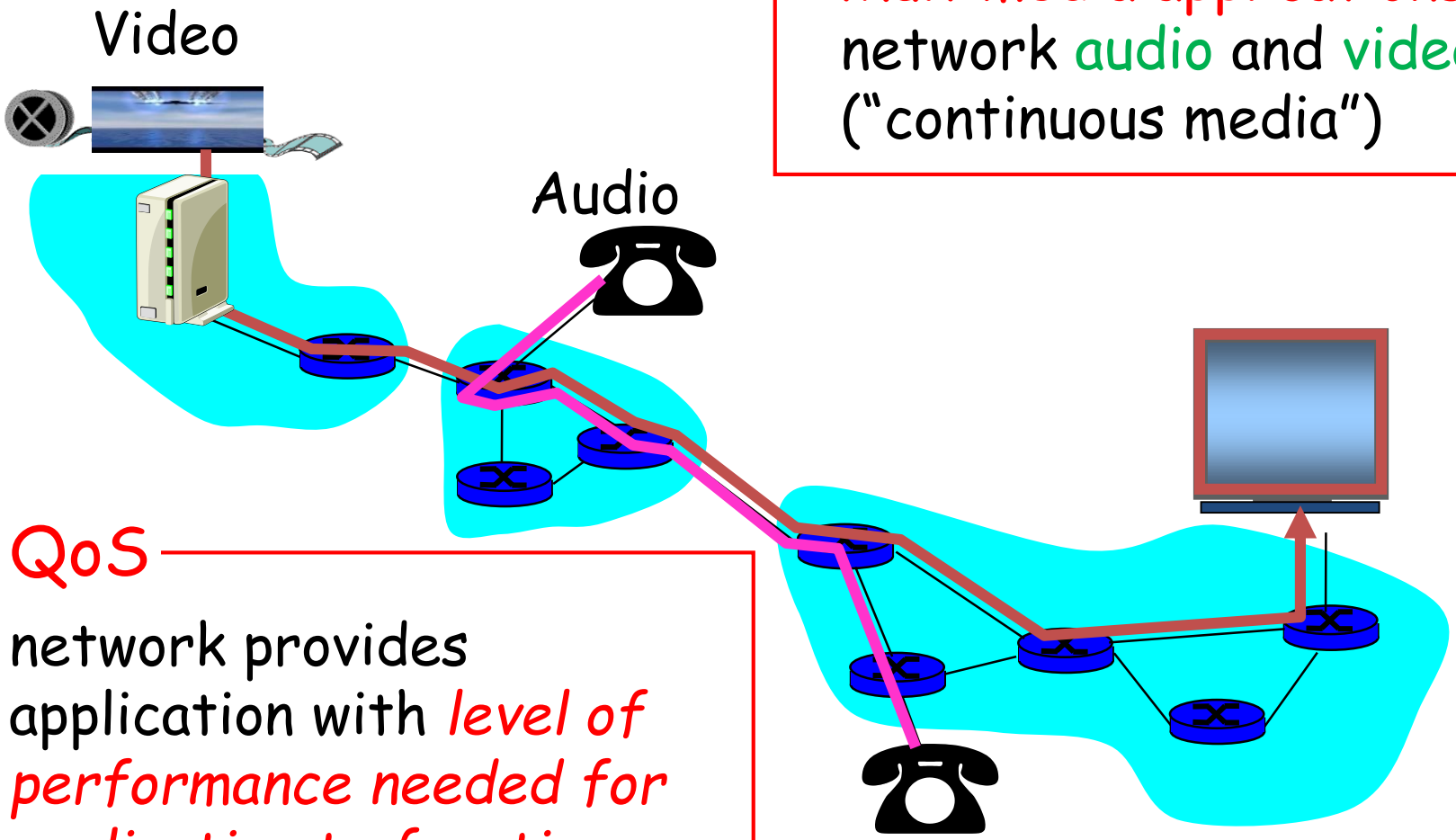
CS348: Computer Networks

Multimedia Networking Applications

Dr. Manas Khatua
Asst. Professor
Dept. of CSE, IIT Guwahati
E-mail: manaskhatua@iitg.ac.in

Multimedia and QoS: What is it?

Multimedia applications
network audio and video
("continuous media")



QoS

network provides application with *level of performance needed for application to function.*

Cont…

- using Internet to watch movies and television shows on demand
- Not only watching!
  - upload and distribute user-generated content,
  - users becoming Internet video producers as well as consumers.
- make telephone calls over the Internet,
- enhance those calls with video and multi-person conferencing
- In the near future,
  - almost all video distribution and voice conversations will take place end-to-end over the Internet,
  - often to wireless devices connected to the Internet via 4G and WiFi

MM Networking Applications

Classes of MM applications:

1) stored streaming
2) live streaming
3) conversational, real-time

Jitter is the variability of packet delays within the same packet stream

Fundamental **characteristics**:

- typically delay sensitive
  - end-to-end delay
  - delay jitter
- loss tolerant: infrequent losses cause minor glitches
- antithesis of data, which are loss *intolerant* but delay *tolerant*.

Properties of Video & Audio

- High bit rate
  - video over the Internet at Kbps to Mbps
- Compare **3 applications**:
  - Looking at a new photo in Facebook every 10 sec. Avg. photo size 200 KB
  - Listening musing streaming over the Internet. MP3 encoding rate 128 Kbps
  - Watching streaming video encoded at 2 Mbps

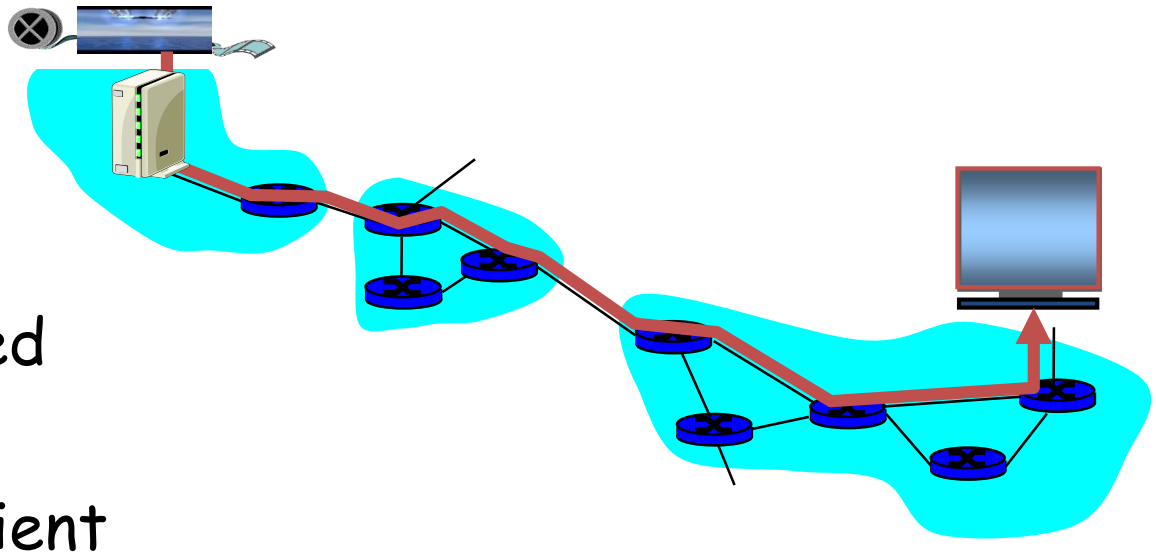
	Bit Rate	Byte transferred in 4K seconds
Photo in Facebook	160 Kbps	80 MB
MP3 Music	128 Kbps	64 MB
Video	2 Mbps	1 GB

- Video Compression
  - Spatial redundancy in image
  - Temporal redundancy in subsequent images
  - Multi version video
    - Tradeoff between bit rate and video quality

- Digital audio has less bandwidth requirement than video
- Audio size depends on modulation rate and compression techniques
  - MP3 commonly use 128 Kbps
- Users are more sensitive to audio glitches than video

Streaming Stored Multimedia

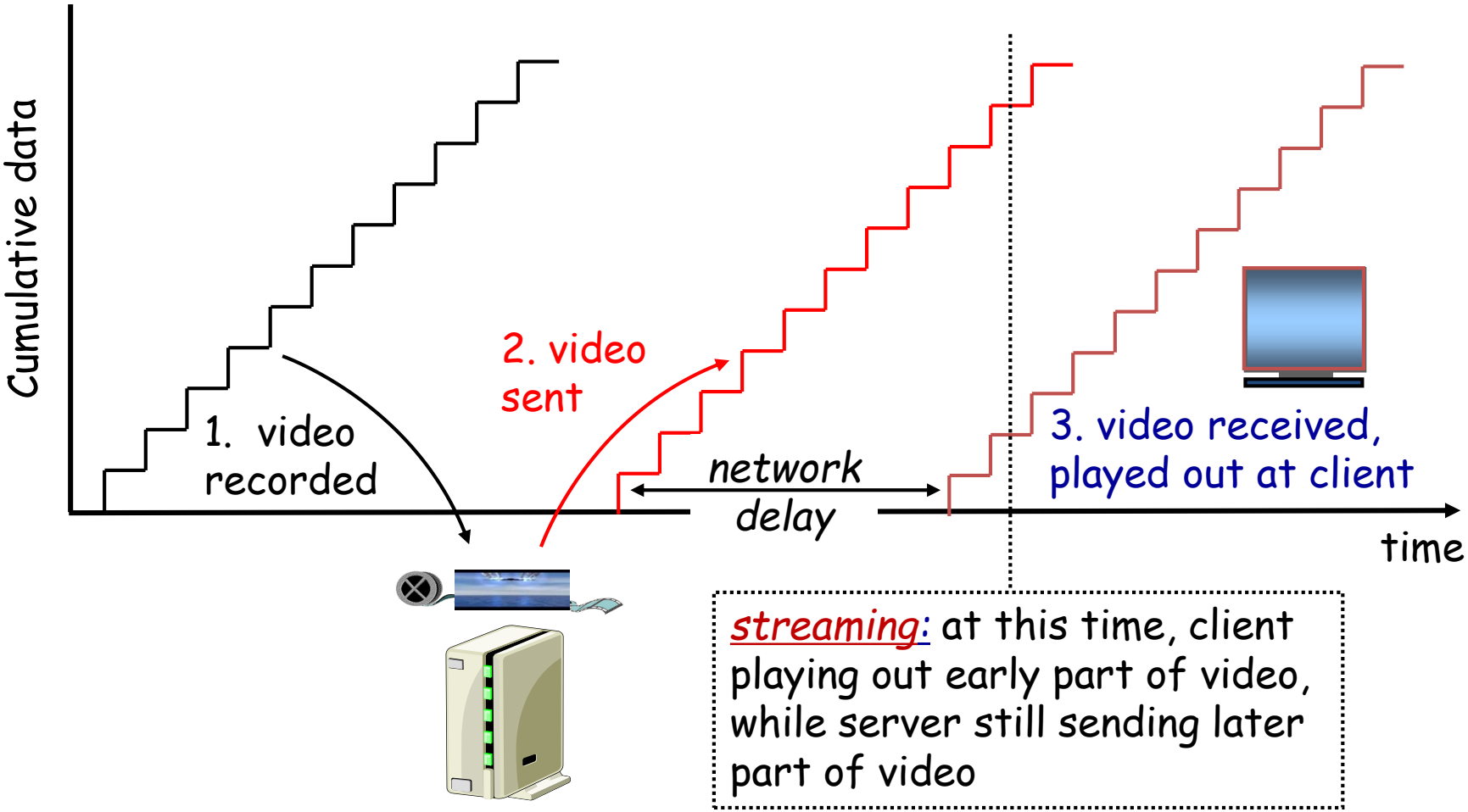
- Streaming stored video has three key distinguishing features
  - Streaming
  - Interactivity
  - Continuous Playout



Stored streaming:

- media prerecorded
- stored at source
- transmitted to client
- *streaming:* client playout begins *before* all data has arrived
  - *Continuous playout*: timing constraint for still-to-be transmitted data: in time for playout

Cont…


Cumulative data
1. video recorded
2. video sent
network delay
3. video received, played out at client
time
streaming: at this time, client playing out early part of video, while server still sending later part of video

Cont…

- *Interactivity* : *VCR-like functionality* - client can pause, rewind, fast forward, push slider bar, etc.
  - 10 sec initial delay OK
  - 1-2 sec until command effect OK

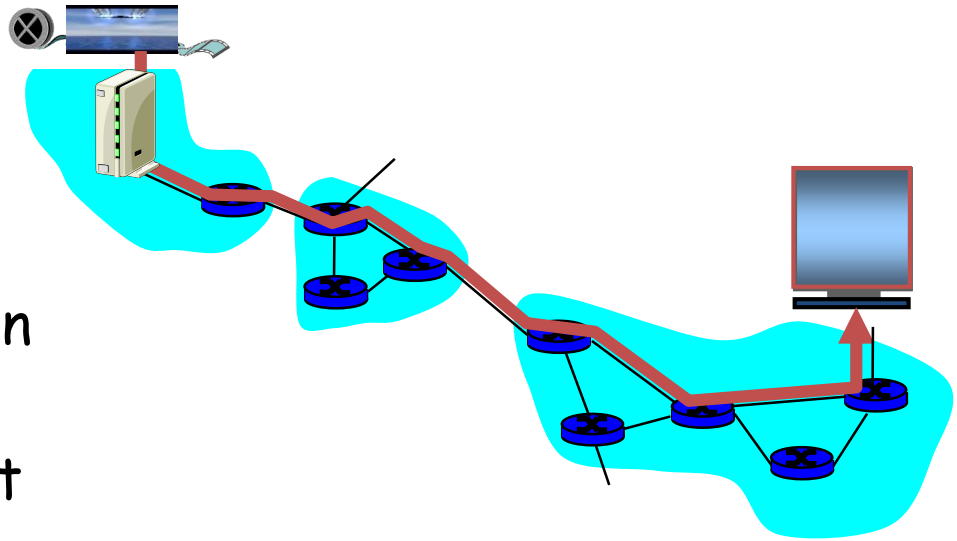
- *timing constraint* for still-to-be transmitted data
  - in time for playout
  - Else, experience video frame freezing or frame skipping

- *Performance Measure:*
  - Average throughput
    - Must be at least >= bit rate of the video
  - Continuous playout is possible in presence of throughput fluctuation
    - by using buffering and prefetching

Streaming Live Multimedia

Streaming (as with streaming stored multimedia)

- playback buffer
- playback can lag tens of seconds after transmission
- still have timing constraint



Interactivity

- fast forward impossible
- rewind, pause possible

Examples:

- Internet radio talk show
- live sporting event

Conversational Multimedia in Real Time

- applications: IP telephony (commonly called VoIP), video conference, distributed interactive worlds

- highly delay-sensitive applications
  - end-end delay requirements:
  - audio: < 150 ms not perceived by human;

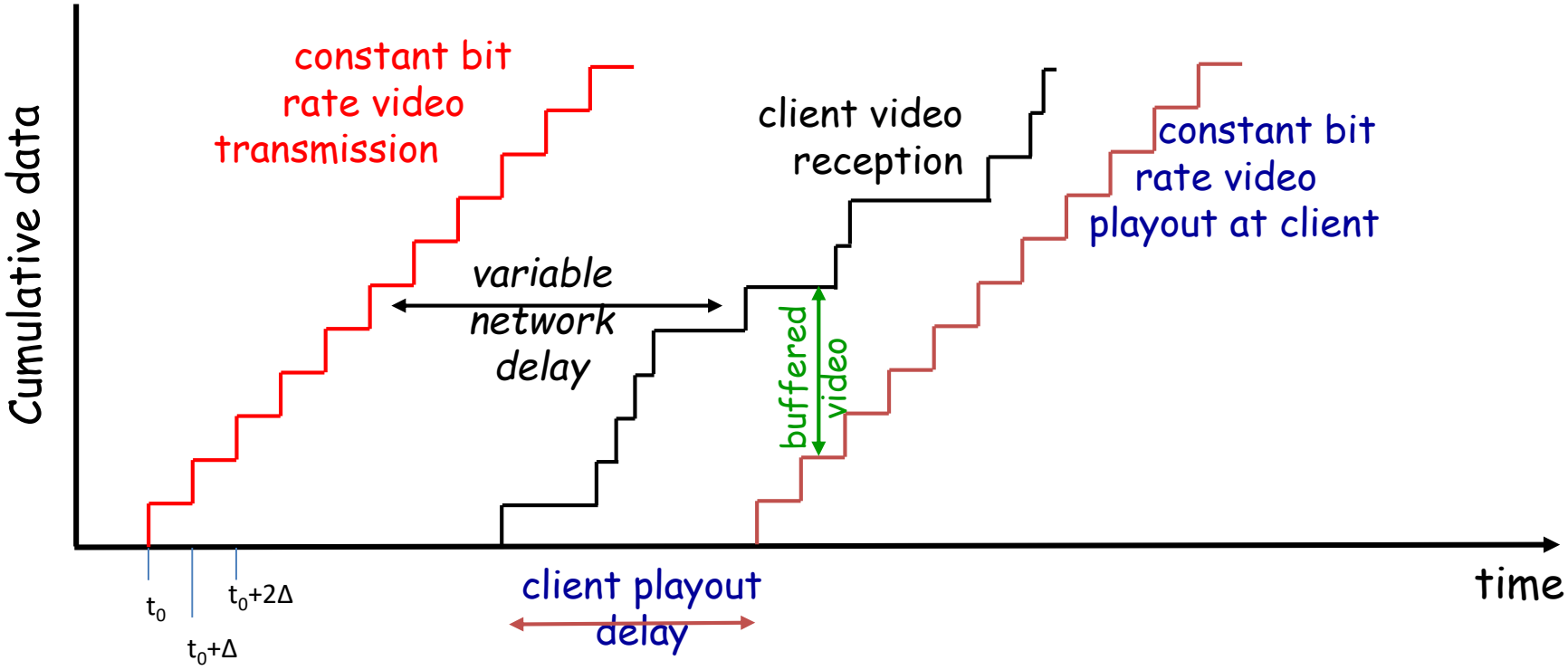
    >150ms & < 400ms ==> Acceptable;

    >400 ms result in frustrating
    - includes application-level (packetization) and network delays
    - higher delays noticeable ==> impair interactivity
- Up to a level of loss-tolerant
  - occasional loss only causes occasional glitches in audio/video playback
  - losses can often be partially or fully concealed

Streaming Multimedia: Client Buffering

- Types of streaming video systems
  - UDP streaming,
  - HTTP streaming,
  - adaptive HTTP streaming

- Common characteristics: **client buffering**



- Advantage: playout delay compensate for network-added delay, delay jitter; i.e., mitigate the effect of varying end-to-end delay.
- tolerate client side bandwidth drops for few moments, i.e. tolerate varying available bandwidth

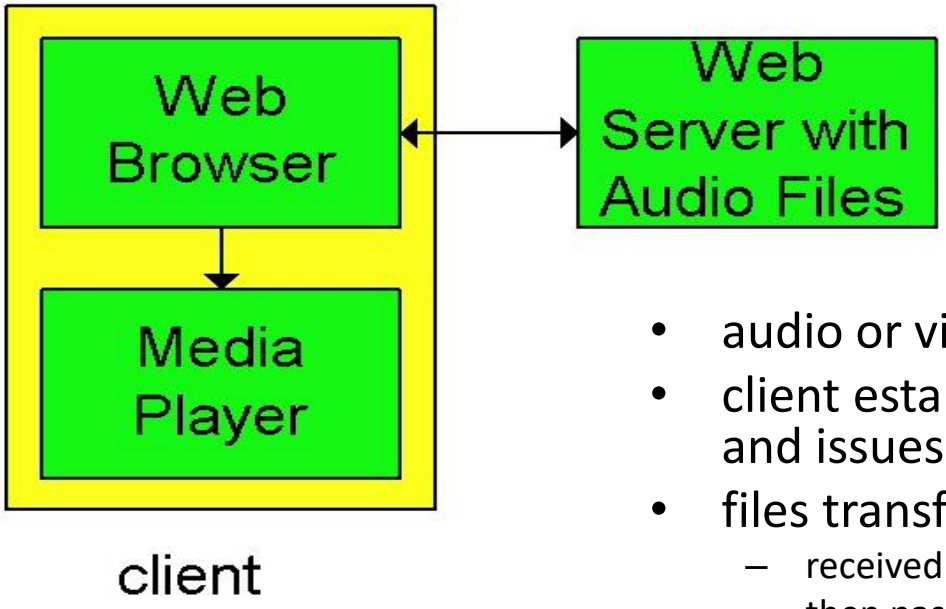
UDP Streaming

- employed in many open-source systems and proprietary products

- Advantages:
  - server sends at rate appropriate for client
    - by clocking out the video chunks at a steady rate.
    - no flow control, no congestion control. So, steady rate works
  - client and server maintain a separate control connection
    - client sends commands regarding session state changes (such as pause, resume, reposition, and so on).

- Disadvantages:
  - fail to provide continuous playout
    - due to unpredictable and varying amount of available bandwidth
  - require a media control server (e.g. RTSP server)
    - to process client-to-server interactivity requests and to track client state
    - It increases the overall cost and complexity
  - many firewalls are configured to block UDP traffic
    - preventing the users to receive UDP video

Internet multimedia: Simplest approach



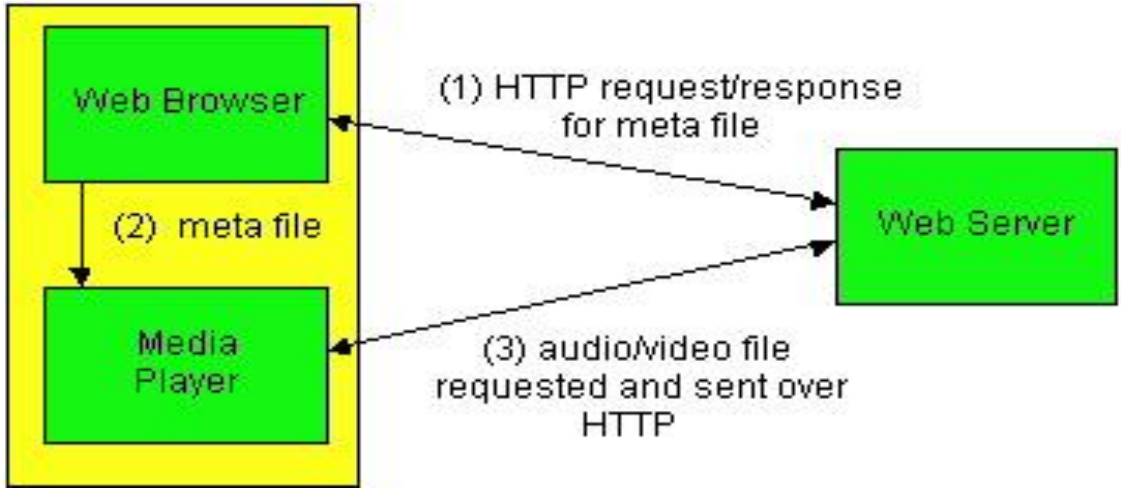
- audio or video stored as a file
- client establishes a TCP connection with the server and issues an HTTP request for the multimedia file
- files transferred as HTTP object
  - received in entirety at client
  - then passed to player

- the conventional wisdom in the 1990s was that video streaming would never work well over TCP
  - Because of TCP congestion control flow control
  - Solution: client buffering and prefetching

audio, video are not streamed:

- no pipelining, long delays until playout!

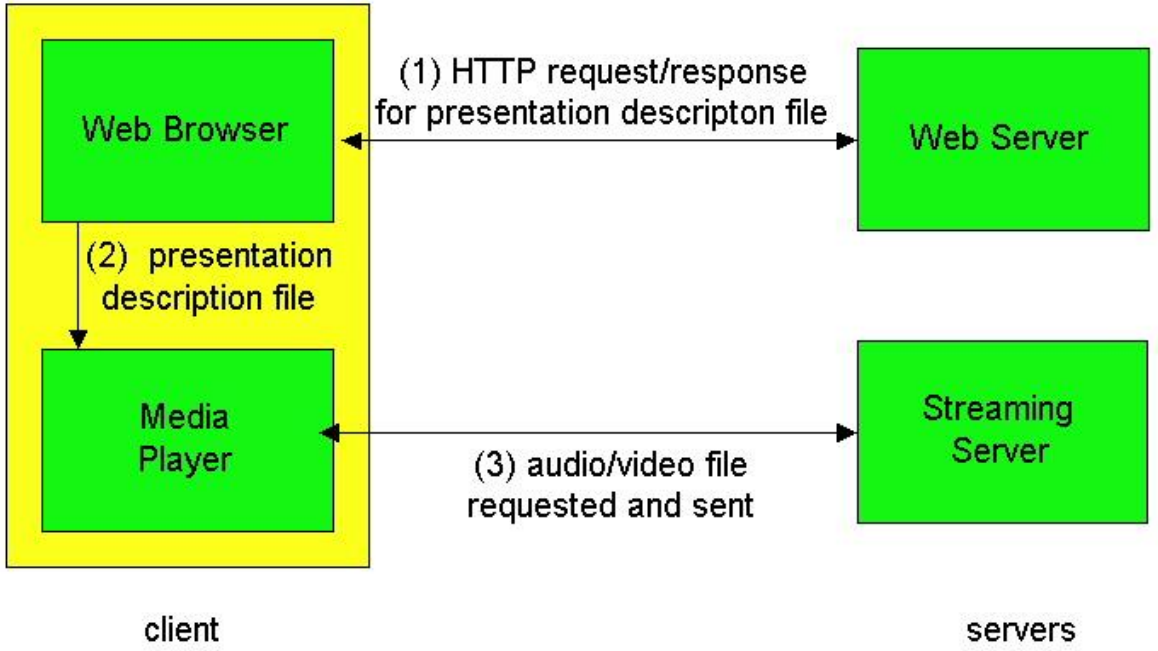
Internet multimedia: Streaming approach



- browser GETs **metafile**
- browser launches player, passing metafile
- player contacts server
- server streams audio/video to player

- HTTP over TCP also allows the video to traverse firewalls and NATs more easily
- Streaming over HTTP also obviates the need for a media control server, such as an RTSP server

Streaming from streaming server



- allows for non-HTTP protocol between server, media player
- UDP or TCP for step (3), more shortly

Prefetching Video

- client-side buffering can mitigate the effects of varying end-to-end delays and varying available bandwidth
- Basic requirement: the server transmits video at the rate at which the video is to be played out

- Prefetching: for streaming stored video, the client can attempt to download the video at a rate higher than the consumption rate,
  - This prefetched video is stored in the client application buffer.
  - If data rate drops, the client will be able to continue to provide continuous playback due to the reserve video

  - Example:
  - Let video consumption rate is 1 Mbps but the network is capable of delivering the video from server to client at a constant rate of 1.5 Mbps.
  - Using prefetching, client can be able to increase the amount of buffered video data by 500 Kbits every second, that will be useful if datarate drops.

When the average TCP throughput is roughly twice the media bit rate, streaming over TCP results in minimal starvation and low buffering delays.

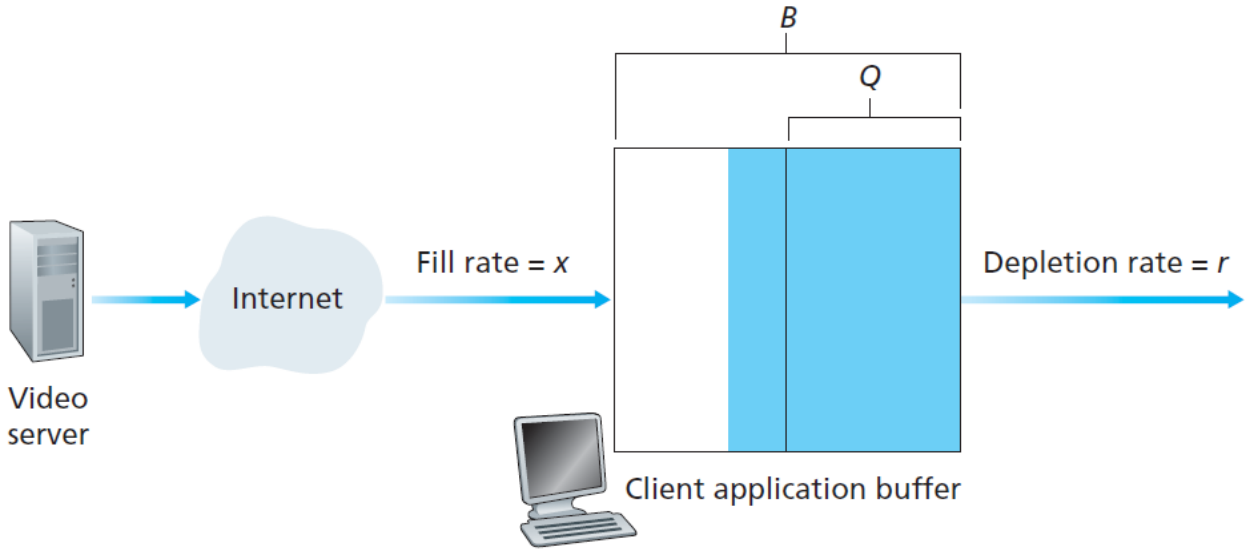
Analysis of Video Streaming

Example:

- Let B denote the size (in bits) of the client's application buffer,
- Q denote the number of bits that must be buffered before the client application begins playout.

- *r* denote the video consumption rate

- If the video's frame rate is 30 frames/sec,
- each (compressed) frame is 10^5 bits,
- then r = 3 Mbps.



- Suppose at time *t* = 0, the application buffer is empty and video begins arriving to the client application buffer.
- At what time (t=?) does playout begin?

- Ans: if server sends bits at a constant rate *x, then* initial buffering delay *t= Q/x*

Cont…

- At what time (t=?) does the client application buffer become full?

- **Solutions**:
- **Case1**: If x < r
  - that is, if the server send rate is less than the video consumption rate, then the client buffer will never become full!

- **Case2**: If always x < r
  - eventually the client buffer will empty out entirely, at which time the video will freeze on the screen while the client buffer waits another Q/x seconds to build up Q bits of video.

- **Case3**: if x > r
  - starting at time Q/x, the buffer increases from Q to B at rate (x-r) since bits are being depleted at rate r but are arriving at rate x
  - Buffer will become full at (Q/x + B/(x-r))

Adaptive Streaming and DASH

- Shortcomings of HTTP Streaming:
  - 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
- Solution: Dynamic Adaptive Streaming over HTTP (DASH)
  - video is encoded into several different versions,
  - each version having a different bit rate and a different quality level

- In DASH, client can select different chunks based on its interest
- DASH allows clients with different Internet access rates to stream in video at different encoding rates
- DASH allows a client to adapt to the available bandwidth if the end-to-end bandwidth changes during the session
- DASH allows the client to switch among different quality levels.
- By dynamically monitoring the available bandwidth and client buffer level, and adjusting the transmission rate with version switching,
- DASH can often achieve continuous playout at the best possible quality level without frame freezing or skipping

Content Distribution Networks

- Goal:
  - need to distribute on-demand multi-Mbps video streams to millions of users on a daily basis
- Example:
  - YouTube
- Challenges:
  - Streaming all this traffic to locations all over the world
  - providing continuous playout
  - maintaining high interactivity
- Naïve Solution:
  - build a single massive data center,
  - store all of its videos in the data center,
  - stream the videos directly from the data center to clients worldwide

Better solution:
Content Distribution Network (CDN)

- Three major problems of the Naïve approach
- First:
  - If the client is far from the data center, server-to-client packets will cross many communication links and
  - likely pass through many ISPs
  - one of these links may provide a throughput that is less than the video consumption rate
  - the end-to-end throughput will also be below the consumption rate, resulting in annoying freezing delays for the user
- Second:
  - Popular video will likely be sent many times over the same communication links
  - waste network bandwidth
  - Pay cost to ISP multiple times for the same video
- Third:
  - Single data center represents a single point of failure

CDN

- A CDN
  - manages servers in multiple geographically distributed locations
  - stores copies of the videos and other types of Web content
- Goal:
  - direct each user request to a CDN location that will provide the best user experience
- **Private CDN**:
  - owned by the content provider itself
  - e.g. Goolge CDN provides Youtube video
- **Third-party CDN**:
  - distributes content on behalf of multiple content providers
  - e.g. Akamai's CDN distributes Netflix content

- CDNs typically adopt one of two different server placement philosophies
- Enter Deep:
  - enter deep into the access networks and deploy server clusters in access ISPs all over the world
  - e.g. Akamai places clusters in approx. 1,700 locations
- Bring Home:
  - bring the ISPs home by building large clusters at a smaller number of key locations and connecting these clusters using a private high-speed network
  - e.g. Limelight CDN
- Generally CDNs do not push videos to their clusters, but instead use a pull strategy
- If a client requests a video from a cluster that is not storing the video
  - then the cluster retrieves the video (from a central repository or from another cluster)
  - stores a copy locally while streaming the video to the client
  - when a cluster's storage becomes full, it removes videos that are not frequently requested.

Case study: Google's Network Infrastructure

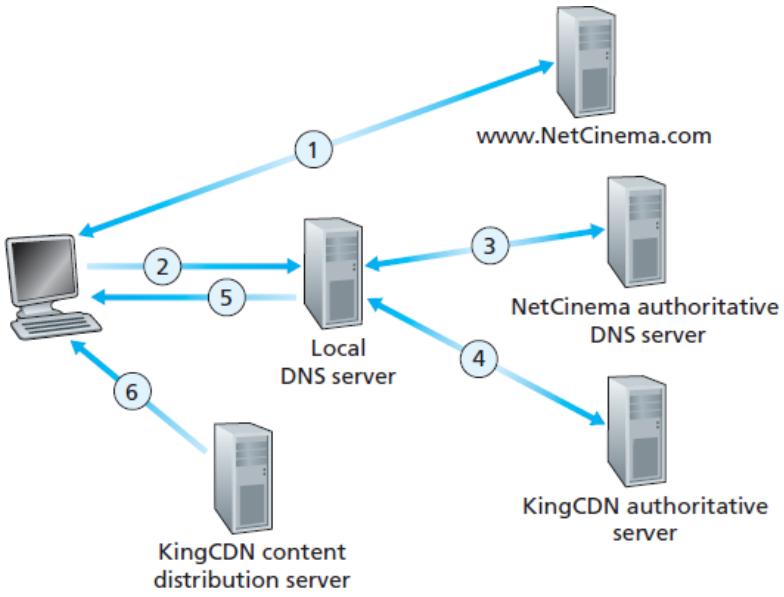
- Google has vast array of cloud services
  - search,
  - gmail,
  - calendar,
  - YouTube video,
  - maps,
  - documents,
  - social networks
- For this, Google has deployed an extensive private network and CDN infrastructure
- Google's CDN infrastructure has three tiers of server clusters
- Eight "mega data centers" till 2012
  - 6 in USA, 2 in Europe
  - each data center having on the order of 100,000 servers
  - responsible for serving dynamic contents (e.g. search result, email)
- About 30 "bring-home" clusters
  - each cluster consisting on the order of 100-500 servers
  - responsible for serving static content (e.g. video)
- Many hundreds of "enter-deep" clusters
  - a cluster typically consists of tens of servers
  - serve static content including static Web pages
- All of these data centers and cluster locations are networked together with Google's own private network, as part of one enormous autonomous system (AS)

How a CDN works?

- NetCinema: is a content provider; KingCDN: is a third-party CDN company
- NetCinema asks the KingCDN to distribute its videos to its customers.

Goal: a browser in a user's host is interested to retrieve a specific video (identified by a URL)

On the NetCinema Web pages, each of its videos is assigned a URL that includes the string "video" and a unique identifier for the video itself say http://video.netcinema.com/6Y7B23V for 'Transformers 7' video



Six steps occur:

1. user visits the Web page at NetCinema
2. user's host sends a DNS query for http://video.netcinema.com
3. user's Local DNS Server (LDNS) relays the DNS query to an authoritative DNS server for NetCinema. Then authoritative DNS server returns to the LDNS a hostname in the KingCDN's domain, for example, http://a1105.kingcdn.com
4. The user's LDNS then sends a second query, now for http://a1105.kingcdn.com , and KingCDN's DNS system eventually returns the IP addresses of a KingCDN content server to the LDNS.
5. The LDNS forwards the IP address of the content-serving CDN node to the user's host.
6. Client user establishes a direct TCP connection with the server at that IP address and issues an HTTP GET request for the video.

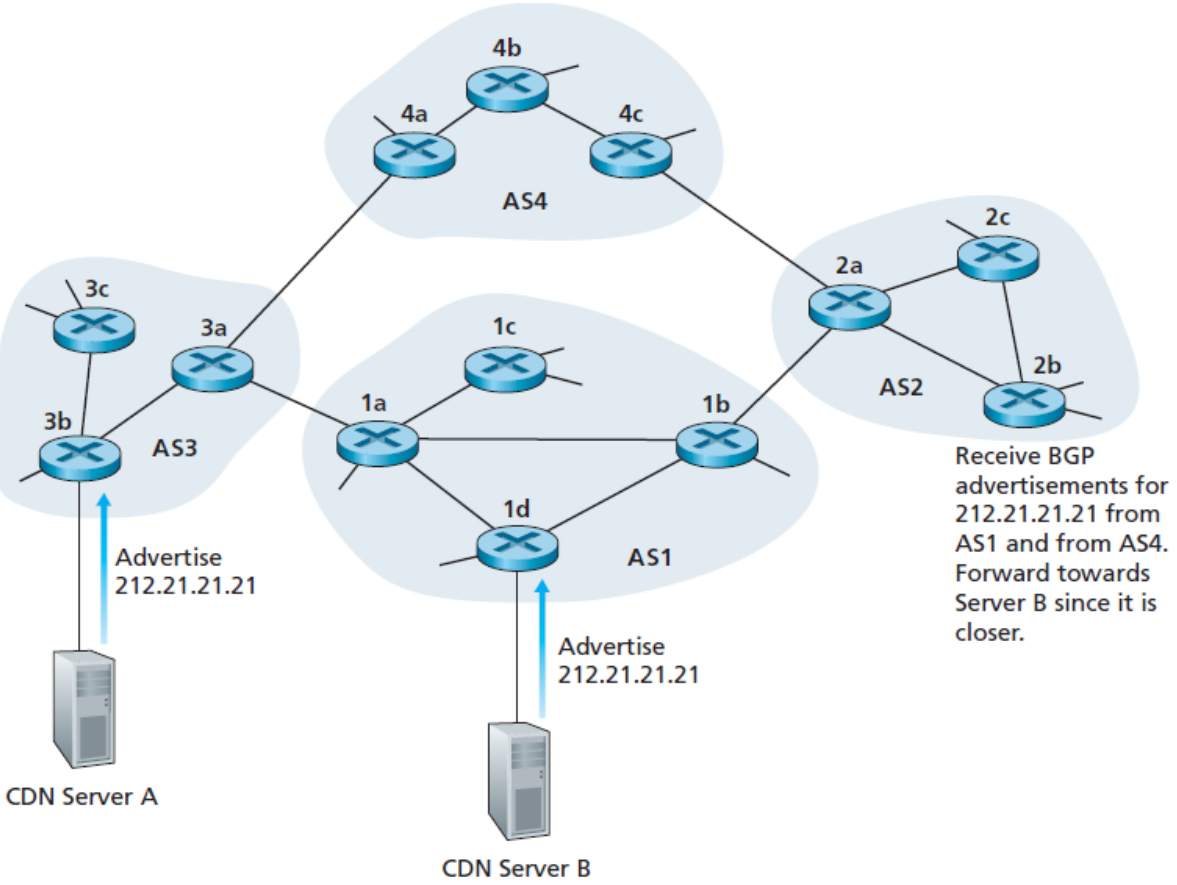
Cluster Selection Strategy

- The core of any CDN deployment is a cluster selection strategy
  - Mechanism for dynamically directing clients to a server cluster or a data center within the CDN
- The CDN learns the IP address of the client's LDNS server via the client's DNS lookup.
- Then, the CDN needs to select an appropriate cluster based on this IP address.
- All cluster selection strategies are normally proprietary protocols.
- None fits for all.

- Simple strategy: assign the client to the cluster that is geographically closest
- Shortcomings:
  a) the geographically closest cluster may not be the closest cluster along the network path
  b) some end-users are configured to use remotely located LDNSs. So, the LDNS location may be far from the client's location
  c) It ignores the variation in delay and available bandwidth over time of Internet paths
- To determine the best cluster for a client based on the *current* traffic conditions, CDNs wishes to perform real-time measurements of delay and loss between their clusters and clients
- How to do that?
  I. Cluster to client path probing
  II. use the characteristics of recent and ongoing traffic between the clients and CDN servers

Cont…

- One good strategy: IP anycast [RFC 1546]
- The idea behind IP anycast is to have the routers in the Internet route the client's packets to the "closest" cluster, as determined by BGP
- IP-anycast configuration stage
  - assigns the *same* IP address to each of its clusters
  - uses standard BGP to advertise this IP address
- When a BGP router receives multiple route advertisements for this same IP address
  - it treats these advertisements as providing different paths to the same physical location (but actually not!)
  - the BGP router will then pick the "best" route to the IP address according to its local route selection mechanism



- When any client wants to see any video, the CDN's DNS returns the anycast address, no matter where the client is located.
- When the client sends a packet to that IP address, the packet is routed to the "closest" cluster as determined by the preconfigured forwarding tables

Thanks!

Content of this PPT are taken from:

1) **Computer Networks: A Top Down Approach**, by J.F. Kuros and K.W. Ross, 6th Eds, 2013, Pearson Education.

2) **Data Communications and Networking**, by B. A. Forouzan , 5th Eds, 2012, McGraw-Hill.

3) **Chapter 3 : Transport Layer,** PowerPoint slides of "Computer Networking: A Top Down Approach", 6th Eds, J.F. Kurose, K.W. Ross