Module - 5

Multimedia Networking, Network support for Multimedia

Multimedia Network Application:

Any n/w app' that employs audio or video.

Properties of video

i) Most important characteristic of video is its high bit rate.

100 kbps for - low quality video conferencing.

3 Mbps - high-quality movies.

eg:

User Activity	Bit rate	Bytes transferred in 67 min
Using Facebook	160 kbps	80 Mbytes
Downloading Hearing Audio	128 kbps	64 Mbytes
Waitha Watching Video	2 Mbps	1 Gbyte

Bit rate requirements

ii) Another important characteristic of video is that it can be compressed, thereby trading off video quality with bit rate.

A video is a sequence of images, typically (played) displayed at a constant rate. (eg: 24-30 images / sec).

There are 2 types of redundancy with respect to video compression.

i) Spatial redundancy - redundancy within a given image. An image with more white space can be compressed efficiently.

ii) Temporal redundancy - repitition from image to subsequent image. If an image & its subsequent image is same then there is no need to re-encode the subsequent image.

Compression is used to create multiple versions of the same video, each at a different quality level. Users can select the versions (300 kbps, 1 Mbps & 3 Mbps) based on the speed of the Internet connectivity.

Properties of Audio

Digital audio (speech & music) has significantly lower bandwidth requirements than video.

Converting analog audio (human & musical instruments generate) to digital signal: Encoding:

i) The analog audio signal is sampled to some fixed rate.

ii) This fixed rate of the samples are rounded to a finite no' of values (quantization values) & this process is referred to as quantization. [eg: 256 quantization value - power of 2].

iii) Each of the quantization values is represented by a fixed no' of bits. 256 quantization values - 1 byte. The concatenation of all the bit values results in the digital representation of the signal.

For playback through audio speaker the digital signal is converted into analog signal (decoding).

Encoding technique: i) Pulse Code Modulation (PCM).

ii) Popular compression technique for high quality music is MP3 (MPEG 1 layer 3).

iii) Advanced Audio Coding (AAC) - by Apple.

i) Streaming ...
ii) Conversational voice/video-over-IP.
iii) Streaming live audio/video.

i) Streaming stored Audio & Video.

The stored (pre-recorded) audio/video is saved in the server & the client requests for it.

Features:

i) Streaming: The client starts playing the video, then the video is getting received from the server, while the client is watching the video the other parts of the video is also getting received. This reduces the delay of receiving the entired video at a time.

ii) Interactivity: The user may pause, reposition forward, backward the video as it is pre-recorded.

iii) Continuous layout: The video is played as per the original timing of the recording.

ii) Conversational voice-and Video-over-IP

It is also called as Internet telephony. eg: Skype, G Talk, ..

Timing considerations & tolerance of data loss are important for conversational voice & video app'.

a) Delay-sensitive. delay < 150 ms — not perceived by human listener.
$150 <$ delay < 400ms - acceptable. delay > 400ms - frustrating.

b) loss-tolerant - loss only causes occasional glitches.

Streaming Live Audio & Video.

eg: Live sporting event or news event.

This is accomplished using the IP multicasting technique.

As the event is live, delay can also be an issue. Delay up to 10 sec can be tolerated.

Streaming Stored Video.

Pre-recorded videos are placed in servers & the user can request the servers to view the video on demand. The user may pause, forward, backward,... the video.

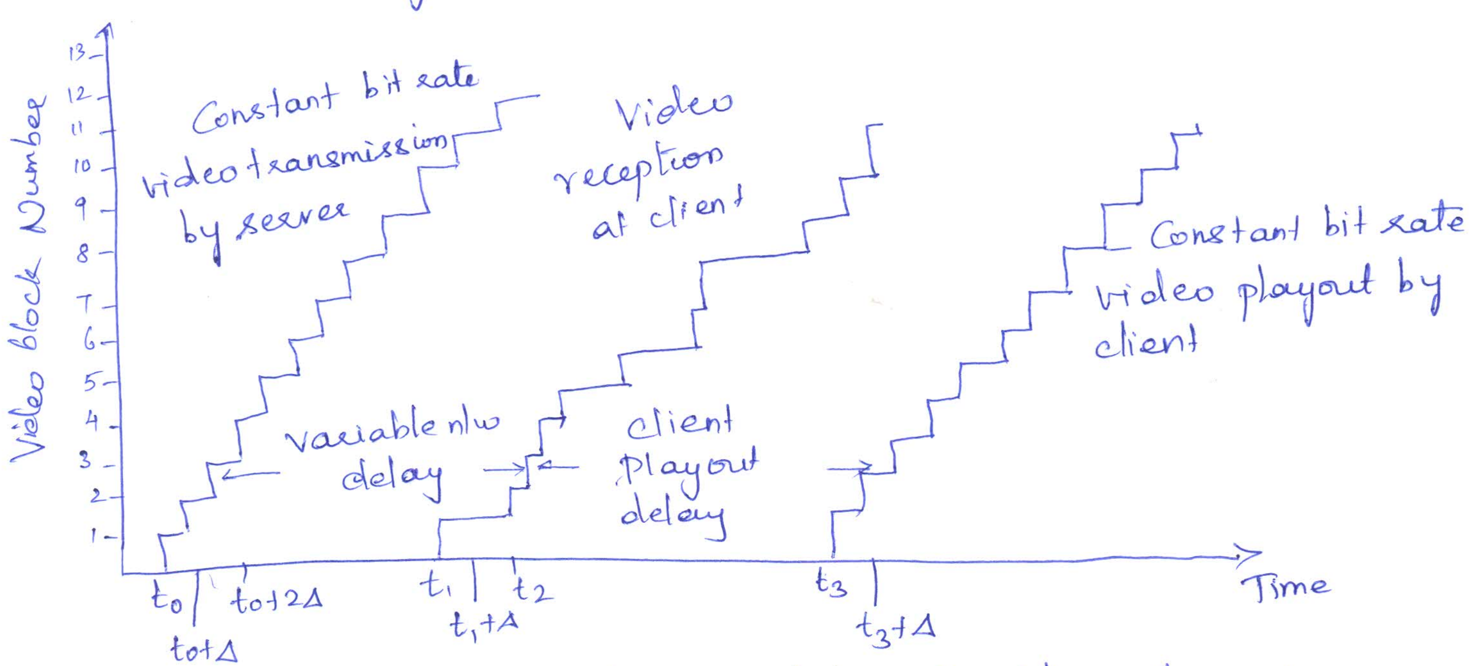
Categories:

a) UDP Streaming b). HTTP Streaming c) Adaptive HTTP streaming

Client-side Buffering :- Common characteristic of all 3 forms.

The video is started playing, the other parts of the video is also loaded & stored in client-side buffer, this reduces unwanted delays.

Δ - fixed amount of time.



Client playout delay in video streaming.

Inorder to avoid delays, the client have to delay the start of playout, such that the other parts of video will be loaded.

UDP Streaming

With UDP streaming, the server transmits video at a rate that matches the client's video consumption rate by clocking out the video chunks over UDP at a steady rate.

Before passing the video chunks to UDP, the server will encapsulate the video chunks within transport packets specially designed for transporting audio & video using the Real-Time Transport Protocol (RTP).

The client & server maintains, a separate control connection over which the client sends commands req' session state changes (pause, resume, ..).

Drawbacks:

i) Fail to provide continuous layout.

ii) It requires a media ctrl server to process client-to-server interactivity requests & to track client state.

iii) Many firewalls are configured to block UDP traffic.

HTTP Streaming

The video is simply stored in a HTTP server as an ordinary file with a specific URL.

When a user wants to see the video, the client establishes a TCP connection with the server & issues an HTTP GET req' for that URL.

The server then sends the video file, within an HTTP response msg. as quickly as TCP congestion & flow control allows, the response msg.

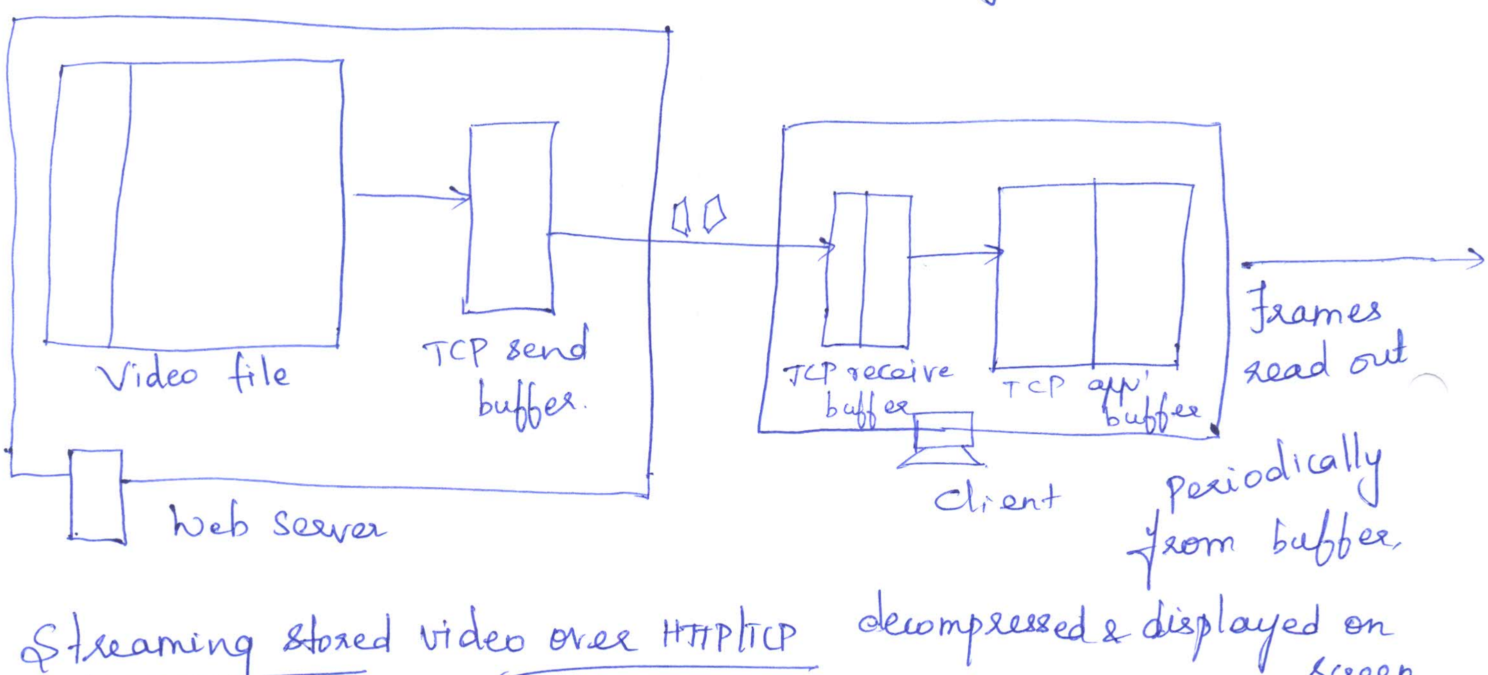
The client app' grabs video frames from the client app' buffer, decompresses the frames, & displays them on the user's screen.

The transmission rate can be varied & in TCP connection HTTP over TCP can traverse firewalls. It doesn't need any media ctrl server (RTSP).

Thus app' like Youtube & Netflix are using HTTP streaming.

Prefetching video.

Video is prefetched that are to be consumed in the future. ref This is done for streaming stored video.



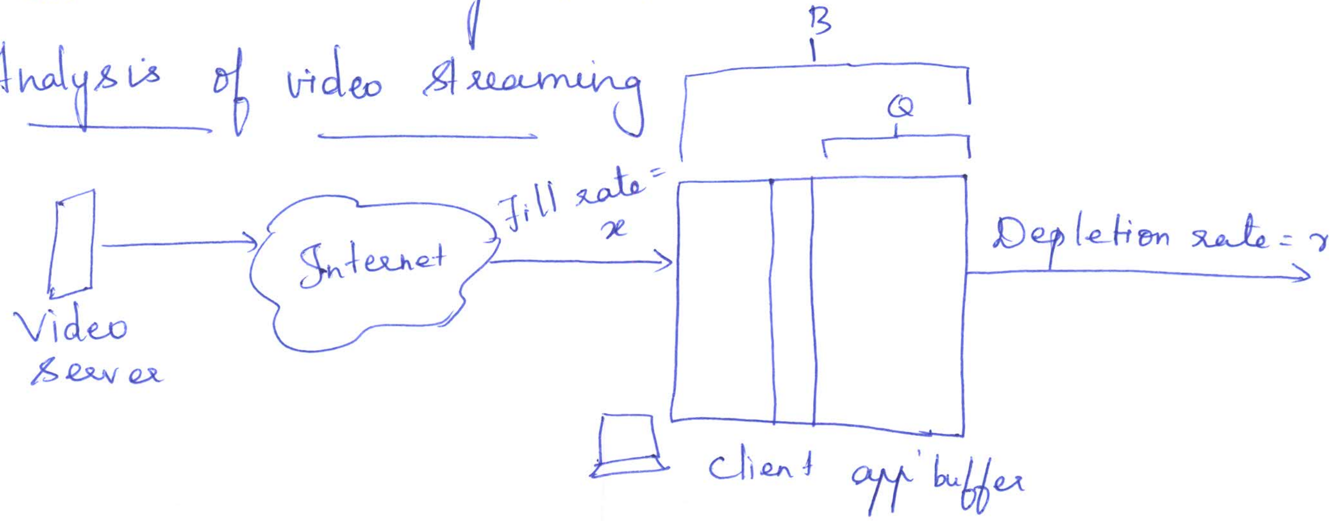
Streaming stored video over HTTP/TCP

Client App' Buffer & TCP Buffers

If the client side buffer is full, then the server buffer will also full & can't send anything. once the client buffer become free. Thus the server send rate can be no higher than the video consumption rate at the client.

Therefore, a full client app' buffer indirectly imposes a limit on the rate that video can be sent from server to client when streaming over HTTP.

Analysis of video streaming



Analysis of client-side buffering for video streaming

B - size of client's app' buffer.

Q - no' of bits that must be buffered before the client app' begins playout.

r - video consumption rate.

x - server send rate.

If $x < r$ → client buffer will never become full.

If client buffer becomes empty, it has to wait for t_p sec.

If $x > r$ → the user will enjoy continuous playout until the video ends.

Early Termination & Repositioning the video.

HTTP streaming s/m's often make use of the HTTP byte-range header in the HTTP GET request msg. which specifies the specific range of bytes the client currently wants to retrieve from the desired video.

When the user wants to reposition (jump) to a future point in time in the video, the client sends a new HTTP req. indicating with the byte-range header from which byte in the file should the server send data.

Early termination - will waste the bandwidth, because of unwanted buffering.

Adaptive Streaming and DASH (Dynamic Adaptive Streaming over HTTP).

The video is encoded in different bit rate & correspondingly different quality levels.

If bandwidth is high, then client selects high-rate version
If " low, " low-rate ".

It allows clients to adapt with bandwidth changes.

Each video version is stored in the HTTP server, each with a different URL. HTTP server also maintains a manifest file - which provides a URL for each version along with its bit rate.

The client first requests the manifest file & learns about the various versions. The client then selects one chunk & gives HTTP GET req msg.

After downloading the client also measures the received bandwidth & runs a rate determination alg.

If the measured bandwidth is high it chooses high-rate version else low-rate version.

If lot of video is buffered - high rate
less " - low rate

DASH allows the client to freely switch b/w different quality levels.

The client can also use the HTTP-byte-range req'.

Content Distribution N/w's (CDN)

Providing continuous playout & high interactivity video to all over the world is a challenging task.

Straightforward approach is to use data center.

Problems:

i) The data center may be far away creating delays.
ii) Popular videos can be sent many times in same link.
iii) Single point of failure.

Inorder to overcome this CDN has been used.

A CDN manages servers in multiple geographically distributed locations, stores copies of the videos in its servers, & attempts to direct each user req' to a CDN location that will provide the best experience.

The CDN may be a private CDN or a 3rd-party CDN.

Philosophies

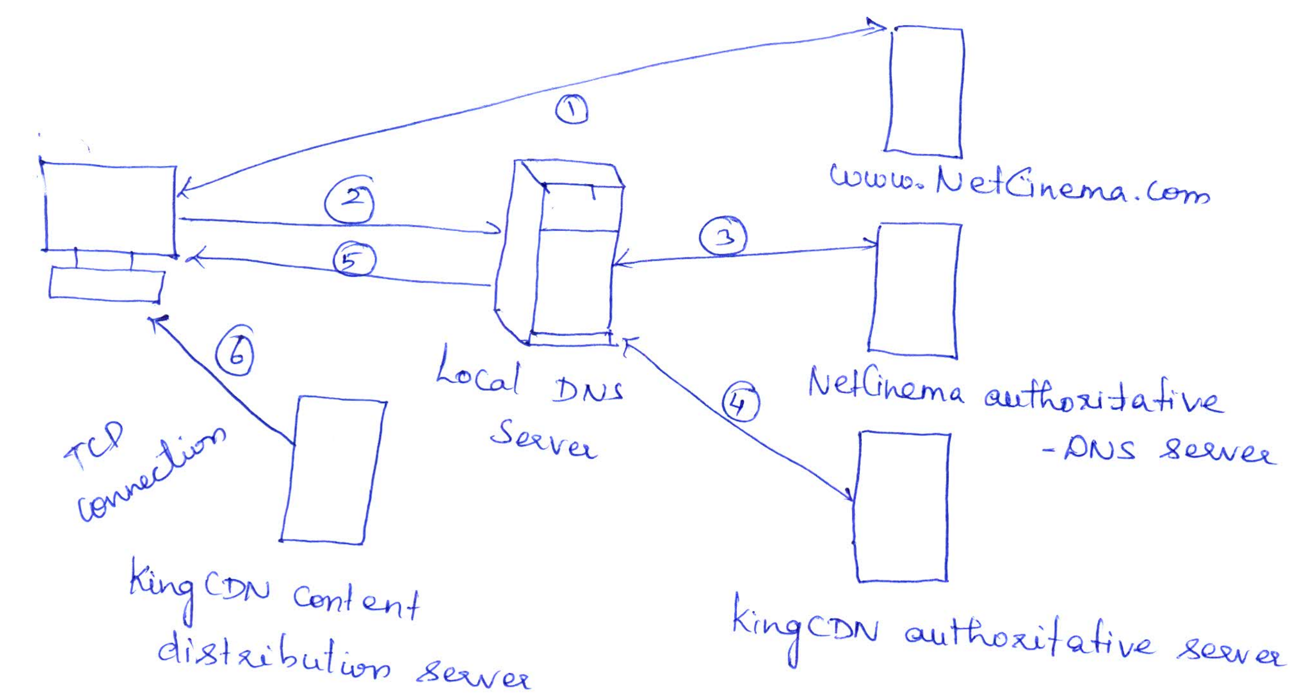
Enter Deep → can be done by deploying server clusters in access ISPs all over the world. This can be done by clusters.

Bring Home → done by bringing the ISPs home by building large clusters at a smaller no' of key locations & connecting these clusters using a private-high-speed n/w.

Once the clusters are in place, the CDN replicates content across its clusters. If the video doesn't exist then it ~~d~~ is cached, when a cluster's storage becomes full, it removes videos that are not frequently requested.

CDN Operation

When a browser in a user's host is instructed to retrieve a specific video, the CDN must intercept the req' so that it can
i) determine a suitable CDN server cluster for that client at that time.
ii) redirect the client's req' to a server in that cluster.



DNS redirects a user's req' to a CDN server.

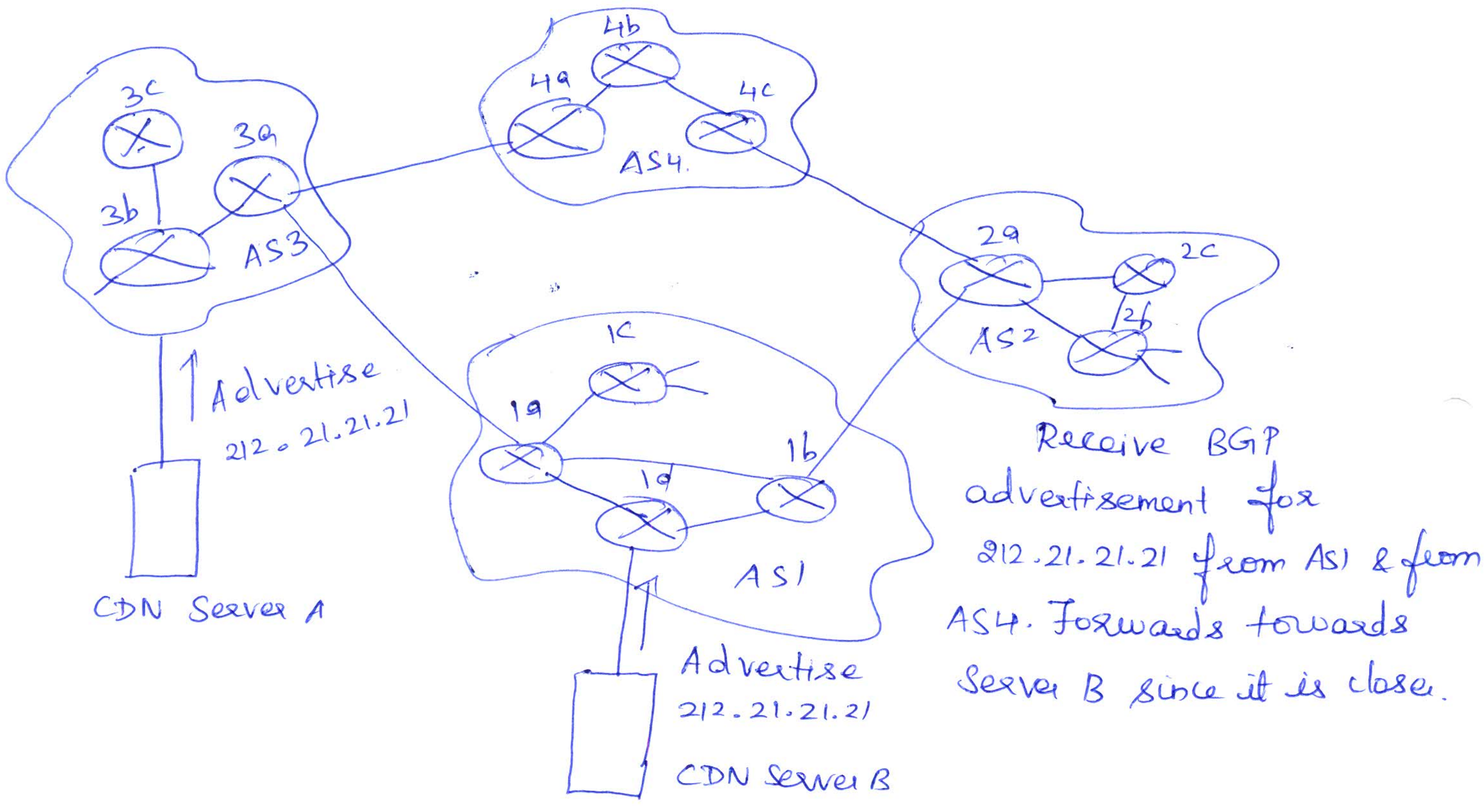
Cluster Selection Strategies

→ A mechanism for dynamically directing clients to a server cluster or a data center within the CDN.

One simple strategy is to assign the client to the cluster that is geographically closest. In order to determine the best cluster for a client based on the current traffic conditions, CDNs can instead perform periodic real-time measurements of delay & loss performance b/w their clusters & clients.

The delay b/w a client & a cluster can be estimated by examining the gap b/w server-to-client SYNACK & client-to-server ACK during the TCP 3-way handshake.

Another approach is to use IP anycast. Idea: is to have the routers in the Internet route the client's packets to the closest cluster as determined by BGP.

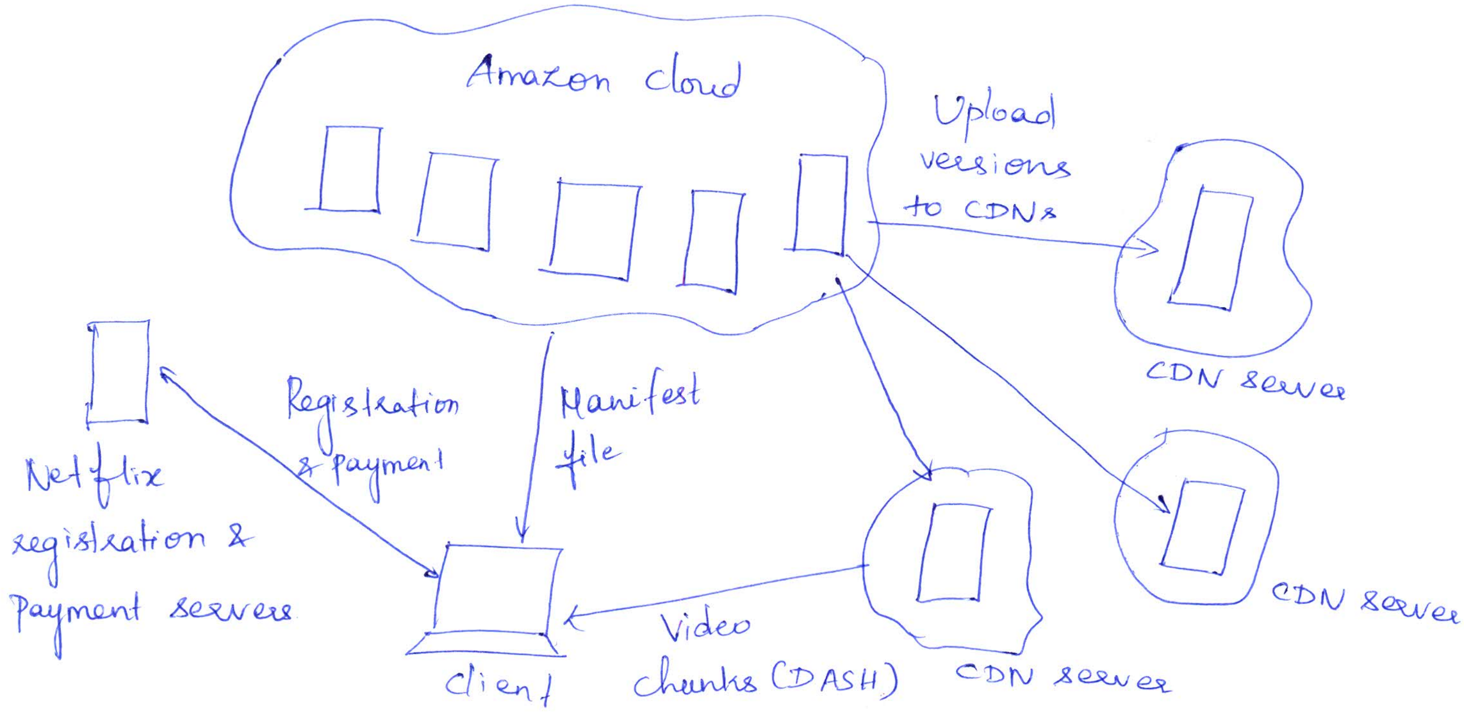


Using IP anycast to route clients to closest CDN cluster.

Case Studies: Netflix, Youtube, Kankan.

Netflix → leading service provider for online movies & TV shows in U.S. It is using extensive 3rd party cloud services & CDNs.

It uses video distribution using a CDN & adaptive streaming over HTTP.



Netflix video streaming platform

Netflix runs its online service by employing machines in the Amazon cloud. Functions in Amazon cloud include:

i) Content ingestion: Before distributing a movie, it must be first ingested & processed.

ii) Content processing: Allows adaptive streaming over HTTP using DASH.

iii) Uploading versions to the CDNs.

When the user selects a movie to "play now," the user obtains a manifest file from servers in the Amazon cloud. Manifest file includes a ranked list of CDNs & URL for different video versions.

After selecting the CDN, DNS is used to contact the CDN server. The client uses the byte-range header in the HTTP GET req msg.

Youtube

World's largest video-sharing site, acquired by Google.

It makes extensive use of CDN technology to distribute it videos. Unlike Netflix, Google doesn't use 3rd party CDN but instead uses its own private CDN to distribute Youtube videos.

The client is connected to the cluster with lowest RTT, but inorder to balance the overload sometimes it may be connected to distant clusters also. (via DNS). If a cluster doesn't have the video then again redirection takes place.

It employes HTTP streaming, it doesn't employ adaptive streaming, it (allows) requires the user to manually select a version. And it uses HTTP byte range req'.

Youtube uploaders also upload their videos from client to server over HTTP. This video is converted in Youtube video format by the Google data centers.

Kankan

Netflix & Youtube not only have to pay for the server h/w, but also for the bandwidth the servers use to distribute the videos. [client-server]

Inorder to overcome this issues P2P approach is used in china. Kankan-leading P2P-based video-on-demand provider in china.

It is similar to BitTorrent, when a peer wants to see a video it contacts the peers (may be centralized or DHT).

It uses UDP.