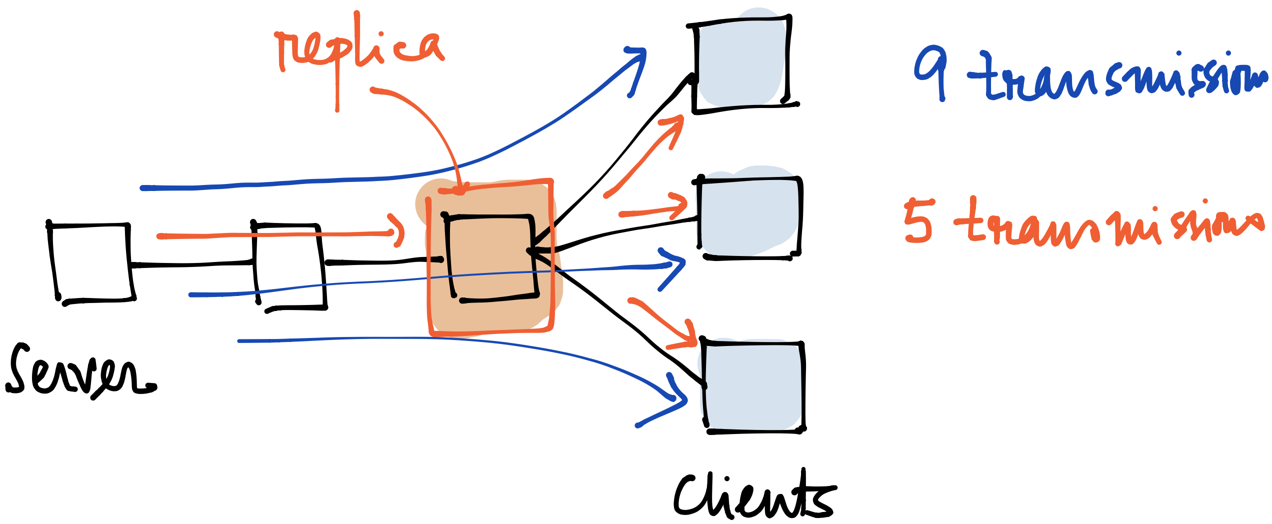
# Content Distribution Network

purpose: faster delivery of packets to clients.

motivation
- Increase of load to popular sites




- Congestion is reduced
- difference with cache -
  this is maintained by the server side
- the responsibility of keeping the info updated is with the server
- doesn't need the validation (cache)

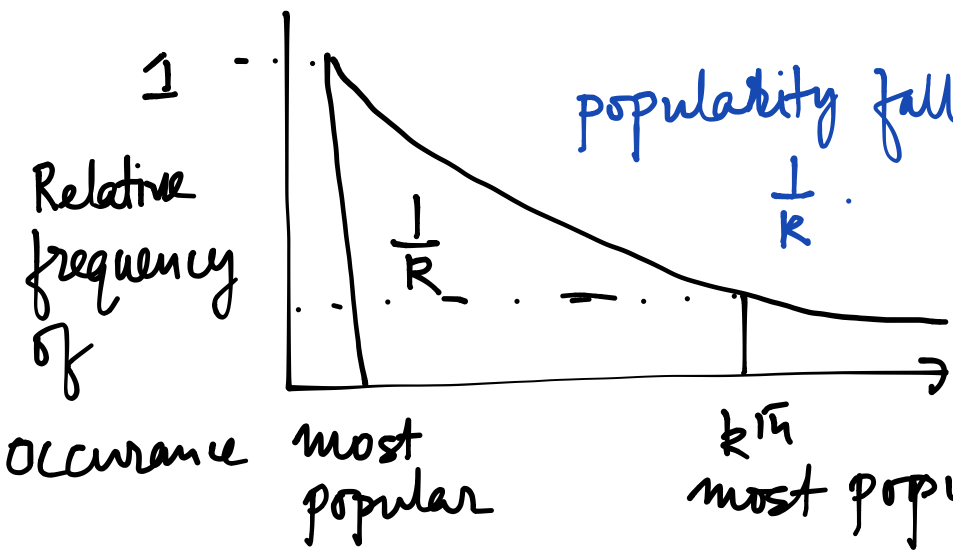
Benefits :

① Reduces server/network load

② Lower PLT.

How popular are the contents ?

popularity of items follow Zipf's law — relative popularity of English words




popularity falls as $\frac{1}{R}$.

true even for content on the web.

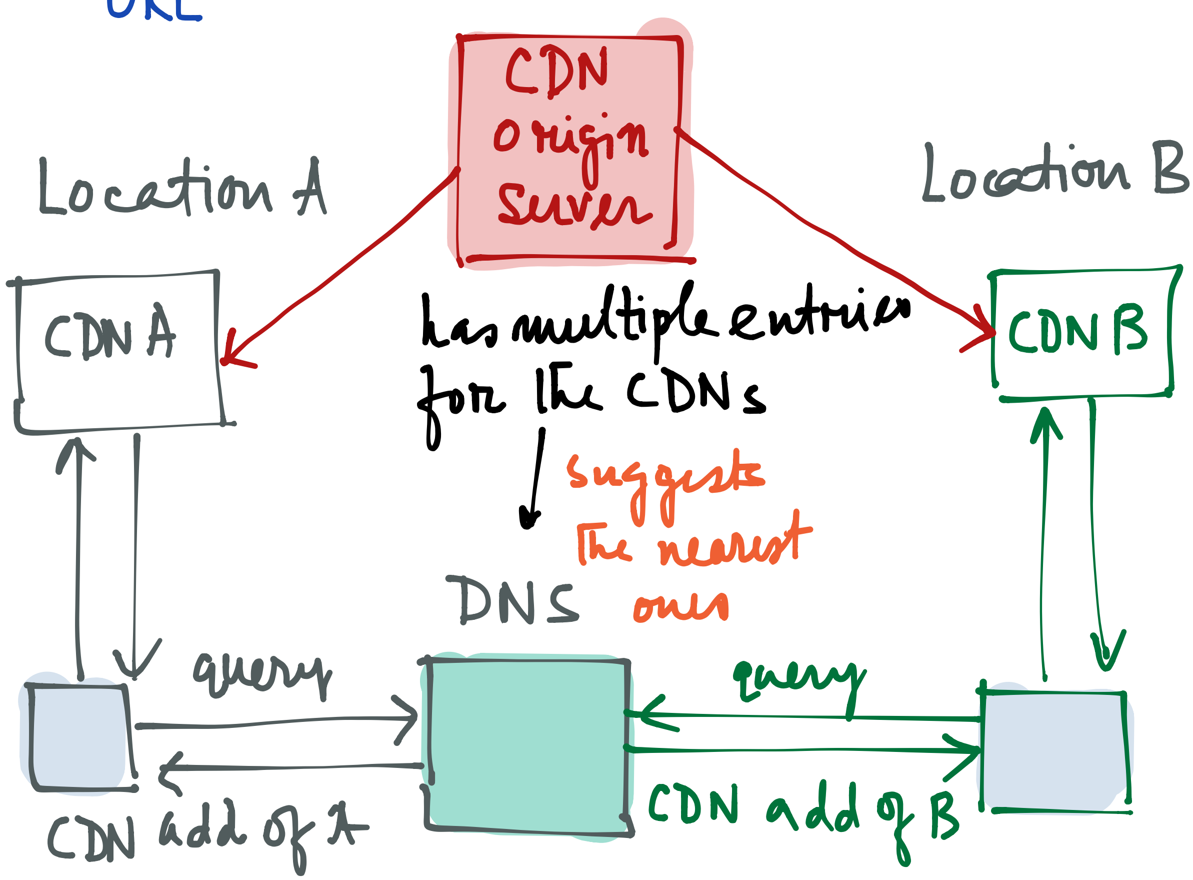
Observations

① Very few contents are very popular — CDN

② Not so popular things are also significant — caching

How to place replicas near the clients?

- By using the DNS
- DNS is already a distributed network
- configured to provide the IP address of the nearest CDN server for that URL



## CDN Endnotes

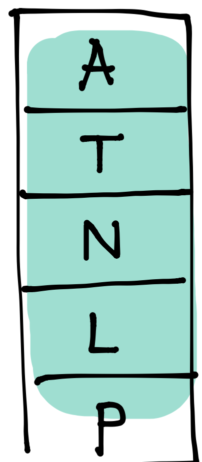
– Both ISP and the Server gains from CDN

less network cost

better PLT.

---

## Quality of Service




The QoS is the measure of the goodness of a service

Contexts :

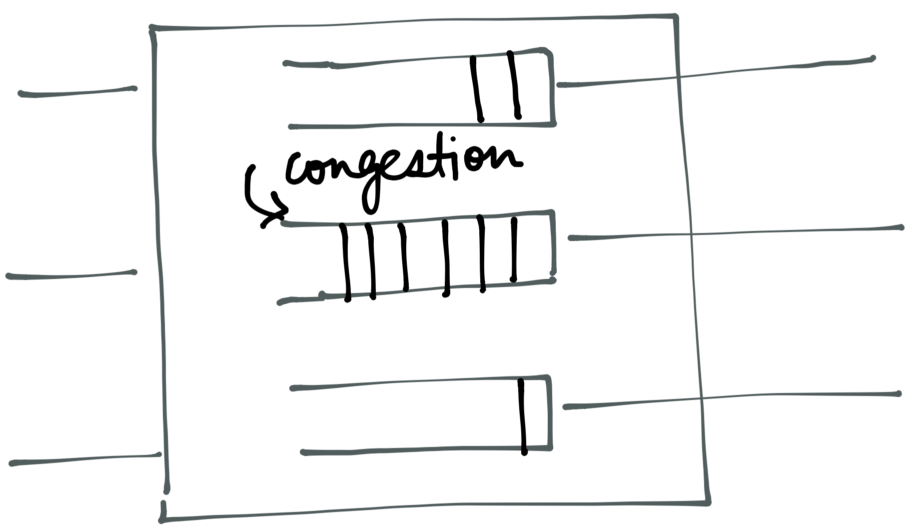
1. high / low bandwidth
2. delay
3. loss
4. signal to noise ratio (physical layer)

   data rate increases when SNR is high.

QoS is a cross-layered approach.

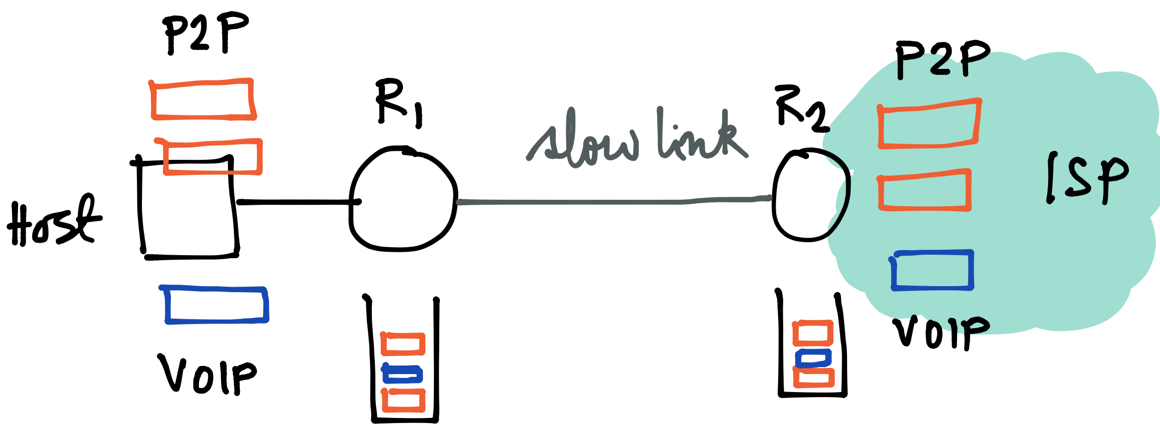
Internet, in its current form, is a best-effort service, doesn't guarantee a QoS.

## Best Effort Service




Physical links / devices and their capacities are fixed.

QoS approach requires intelligent planning.




VOIP gets slower because the queues are FIFO and P2P packets are generated more rapidly.

- P2P doesn't need a delay guarantee
- VOIP needs delay guarantee.

## Possible Solutions

1. Split the link into two and divide traffic explicitly – P2P loses
2. Create two queues and prioritize the VOIP queue. – Weighted Fair Queue – better.

– QoS requirement depends on the application

# Application Requirements

H, M, L – denotes high, medium, low stringent requirements of those factors

variation in delay

| App | BW | Delay | Jitter | Loss |
|---|---|---|---|---|
| Email | L | L | L | M |
| P2P | H | L | L | M |
| Web access | M | M | L | M |
| Audio on demand | L | L | H | L |
| Video on demand | H | L | H | L |
| Telephony | L | H | H | L |
| Video conferencing | H | H | H | L |

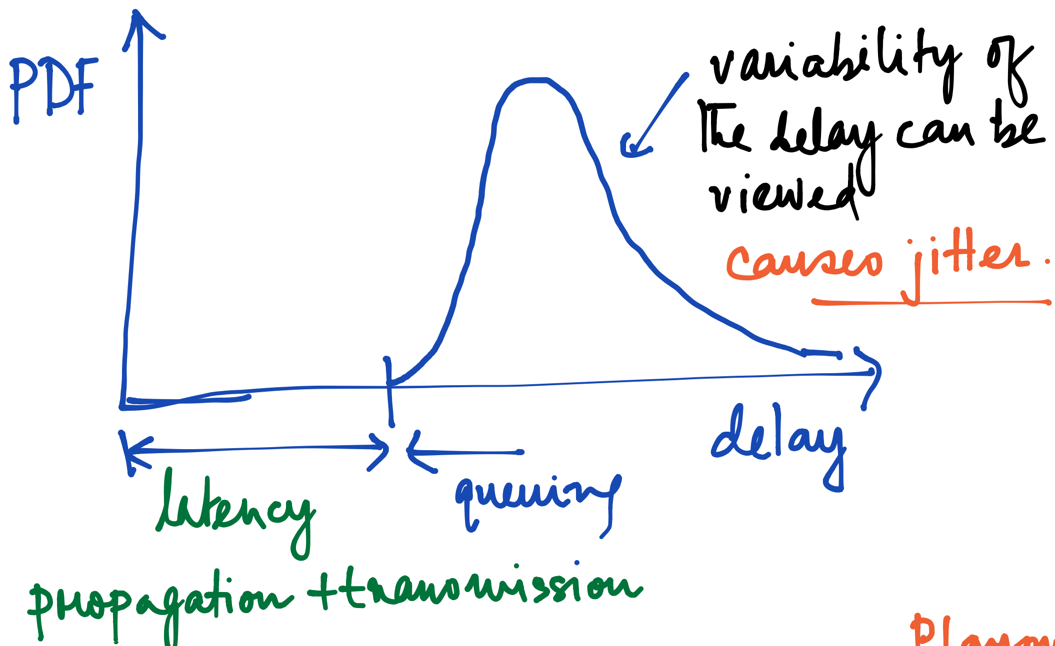
Topics:

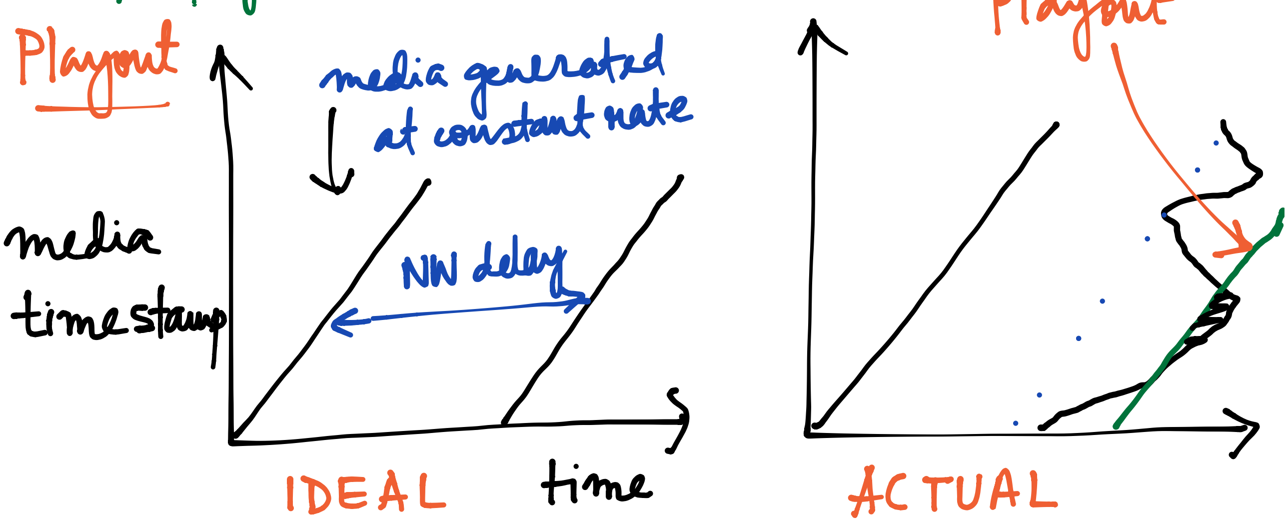
1. Real time transport
2. Streaming Video
3. Fair Queuing
4. Traffic Shaping
5. differentiated service
6. Rate/delay guarantees.

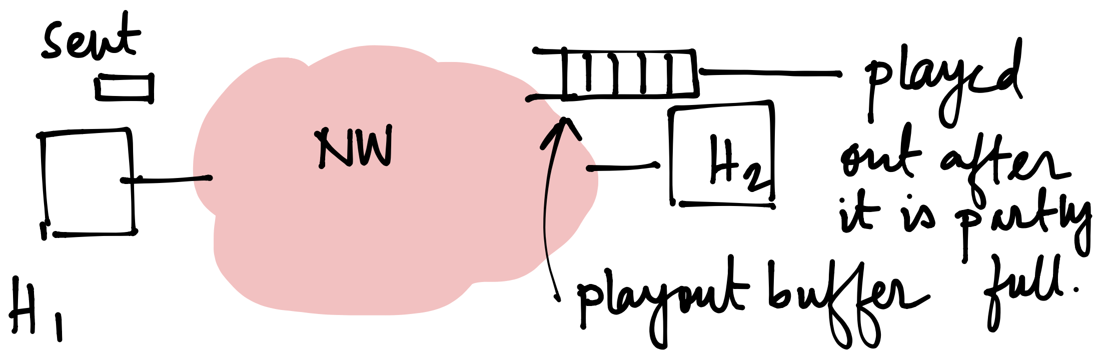
# Real time transport

Skype, Zoom, Google meet – both parties wait for the message.

Network delay is variable (Unidirectional)






## Playout Buffer

Tradeoff

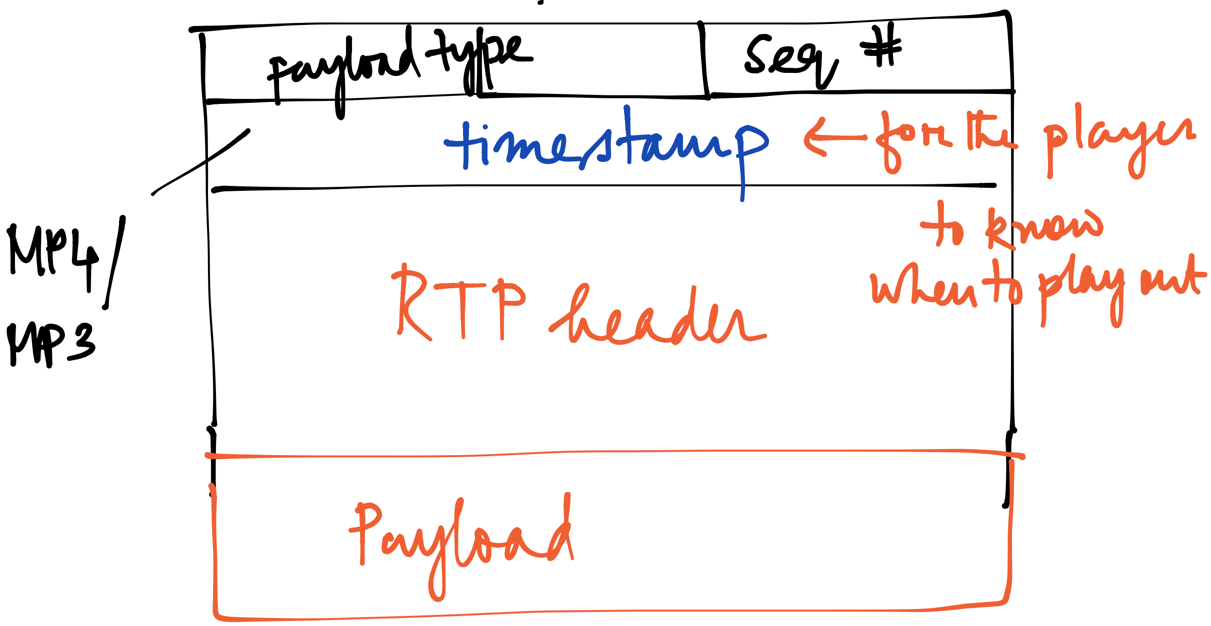
- longer acceptable network delay
  - larger buffer & delay, less loss
- smaller acceptable delay
  - smaller buffer & delay, more loss

A real-time call consists of several components.

1. Call setup, SIP →
2. Session description, SDP
3. Media transport, RTP →
4. Media playout, buffer.

## Media Transport, RTP

RTP is used to carry media over UDP
(Real-time Transport Protocol)

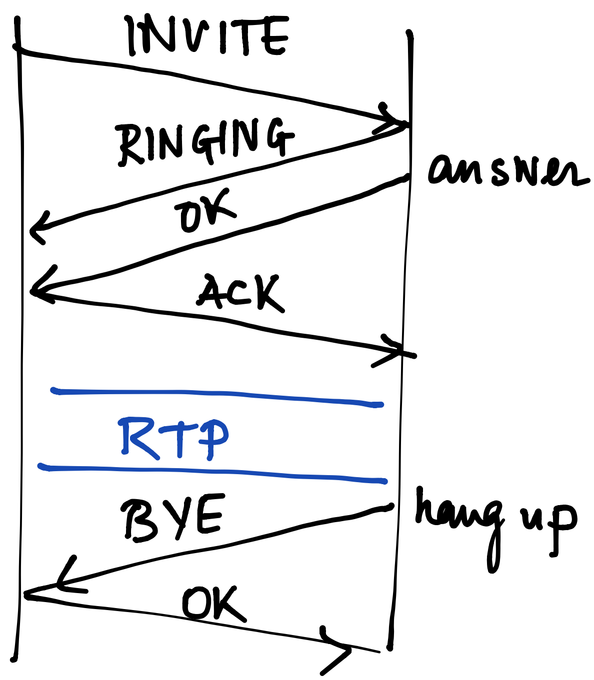



## Establishing the call, SIP
Session Initiation Protocol

- Open protocol for establishing voice and video calls over IP

# SIP signaling

- simple request/response codes
- Runs on UDP or TCP (agnostic of the TRANSPORT layer)



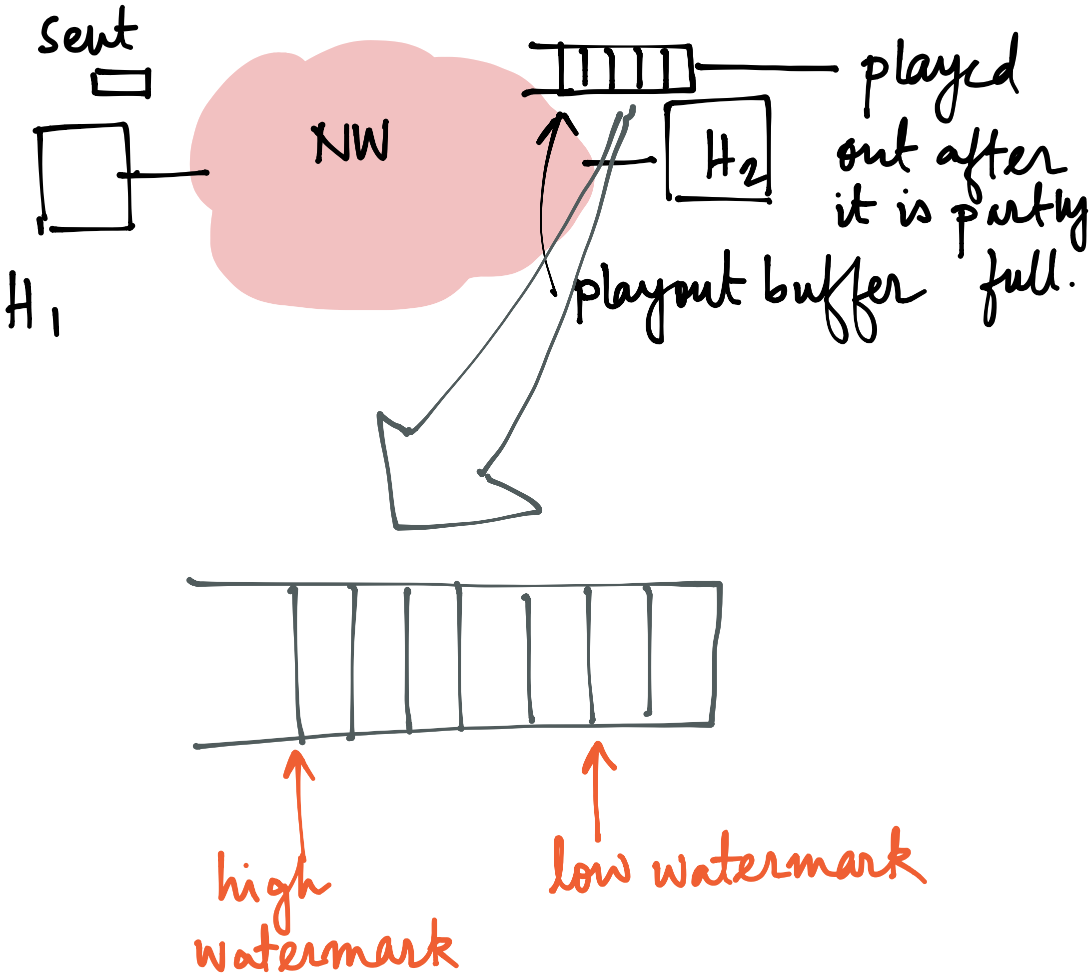

# Streaming Media

- Best effort service of internet
- Popular applications - Youtube, Videolectures

## Comparison with real-time media

- less demanding
- unidirectional, hence delay is not an issue - rather jitter is the issue

# Handling Jitter




Stop pulling data from server at HIGH

Start pulling data from server at LOW

→ Start playout (video buffering)

## Handling Bandwidth

Send file in one of the encodings
- higher quality encoding → more bandwidth
- select best encoding given bandwidth

Example: video qualities are made worse when the bandwidth is low.

## Streaming over TCP or UDP?

- UDP minimizes message delay for real-time interactive sessions
- TCP is preferred for streaming
  - low delay is not essential for streaming
  - flow control ensures smooth buffer fill-up.
  - loss recovery improves presentation

HTTP/TCP

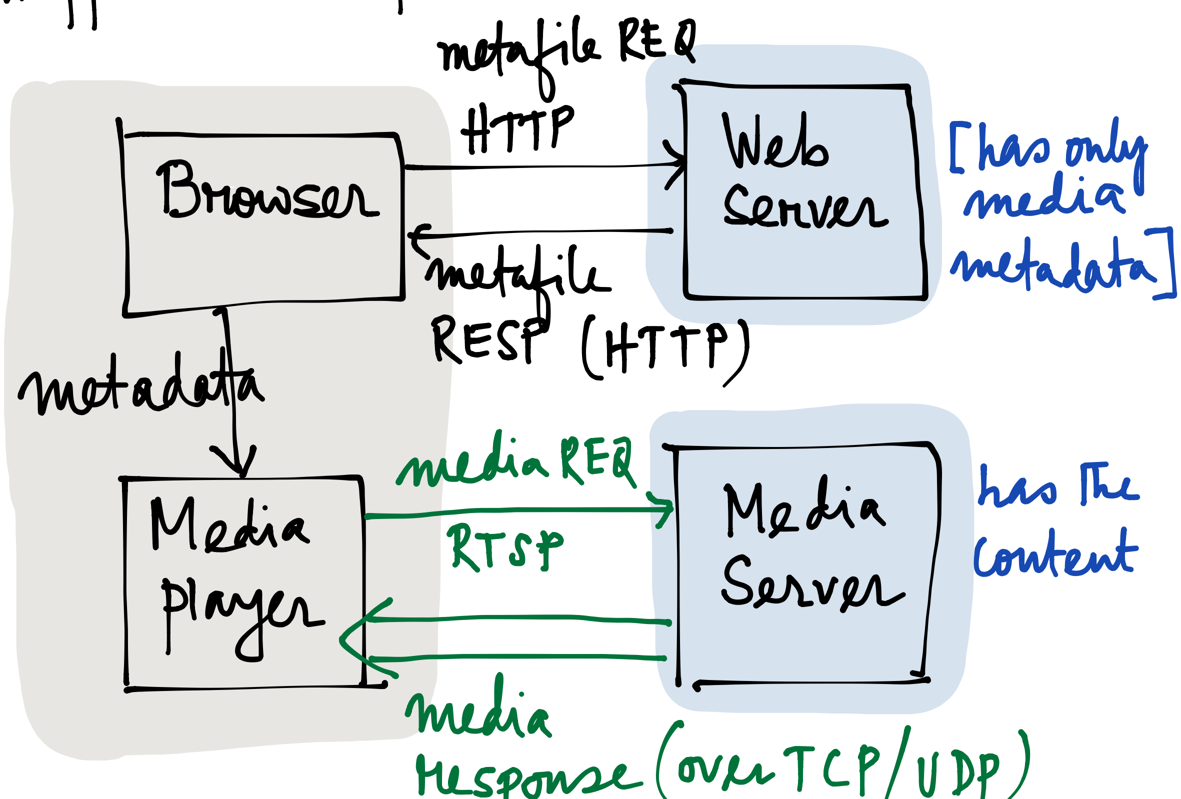
## Components of Streaming Media

- Signaling, e.g., with RTSP
- Media transport, e.g., with HTTP
- Media playout, using buffer

---

## Streaming with RTSP (real time streaming protocol)

Happens in two phases




buffering done for a smooth playout.

# Streaming with HTTP

replacing the second part altogether

Steps:

1. Fetch media description data
   - indices of clips, rates etc.
2. Fetch small segments
   - put in playout buffer, play according to HIGH/LOW watermark
3. Adapt to the selection of encoding
   - based on buffer occupancy