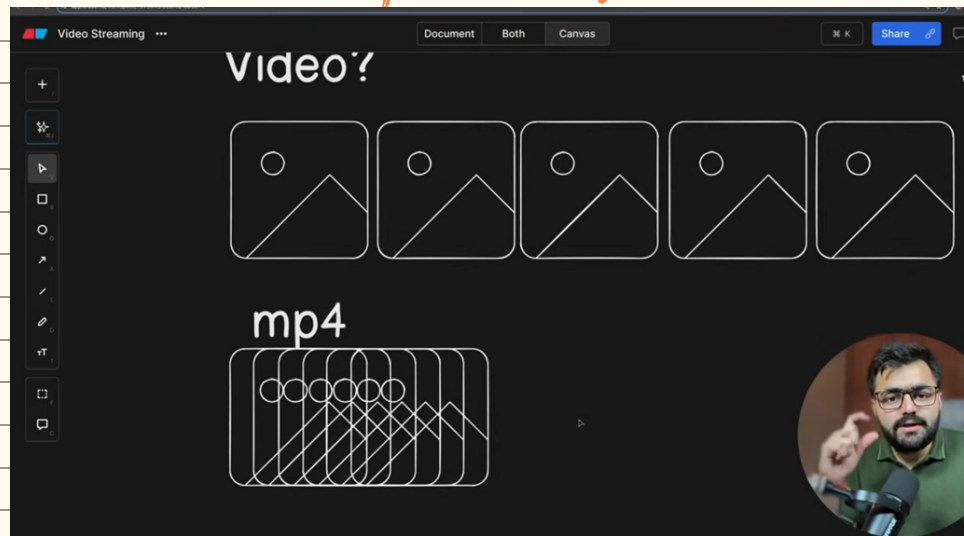


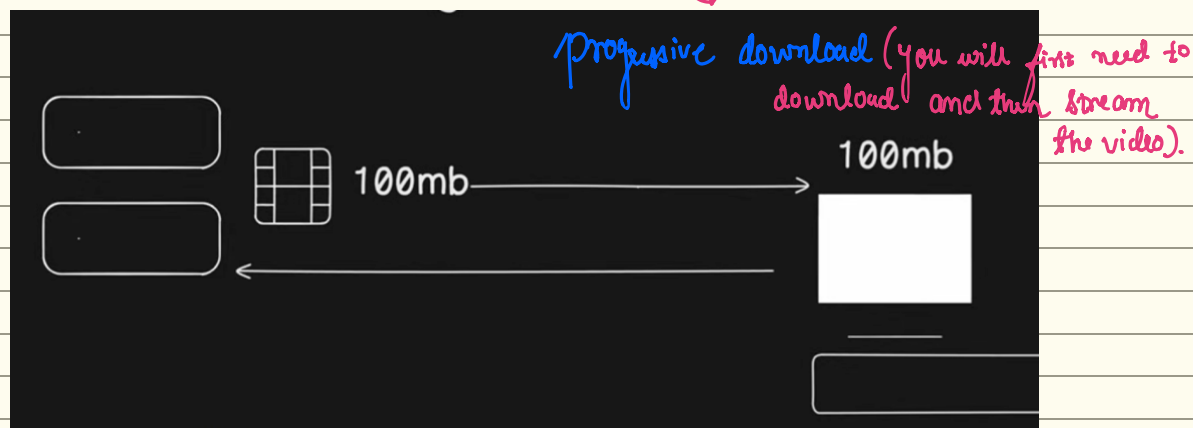
# SYSTEM DESIGN

How video streaming works on scale.

→ Videos:- Videos are sequence of photos and frames in motion



in early 2004:- Streaming platforms



→ Streaming:-

∴ RTMP:- Real time messaging protocol.

∴ RTSP:- Real time streaming protocol.

RTMP  
RTSP

→ Specialized streaming protocol.

Low latency.  
Efficient use of Bandwidth.

Live Streaming support.

(i). RTMP was developed by Adobe

(ii). RTSP was developed by RealNetworks.

## → Adaptive Bitrate Streaming:-

Adaptive Bitrate Streaming (ABR) is a video streaming technique that automatically adjusts the video quality in real-time based on the viewer's current network speed and device performance.

### How It Works

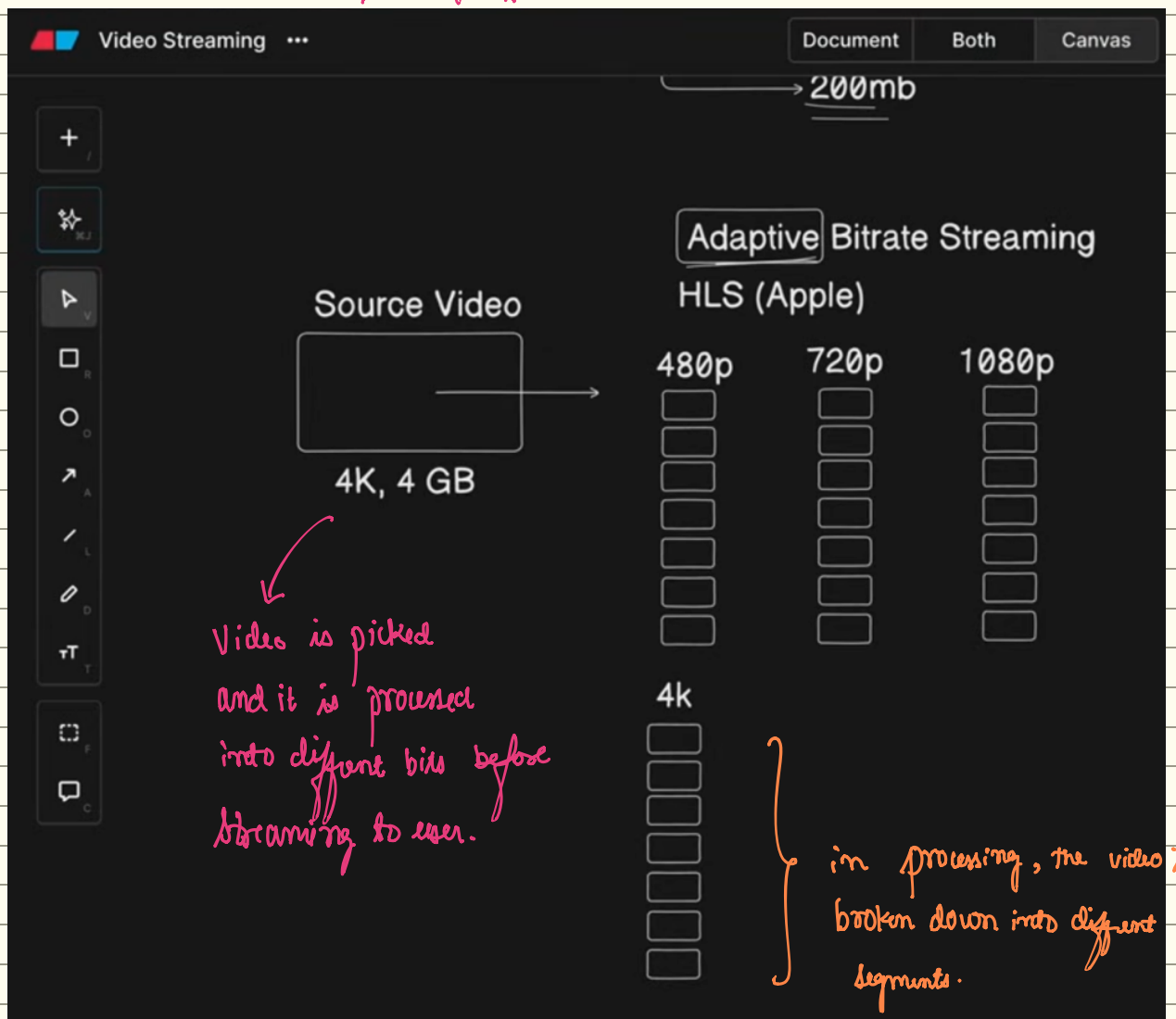
When you stream a video:

1. The video is **divided into small chunks** (segments), each a few seconds long.
2. Each segment is **encoded in multiple quality levels** (e.g., 240p, 480p, 720p, 1080p).
3. The player (on your phone/laptop) continuously checks:
  - Internet speed
  - Buffer health
  - CPU/device capability
4. Based on current conditions, the player **dynamically switches** between quality levels.

So if your network slows down → video automatically shifts to lower quality (prevents buffering).

If your network improves → quality jumps back to HD/4K.

## HLS (developed by Apple)

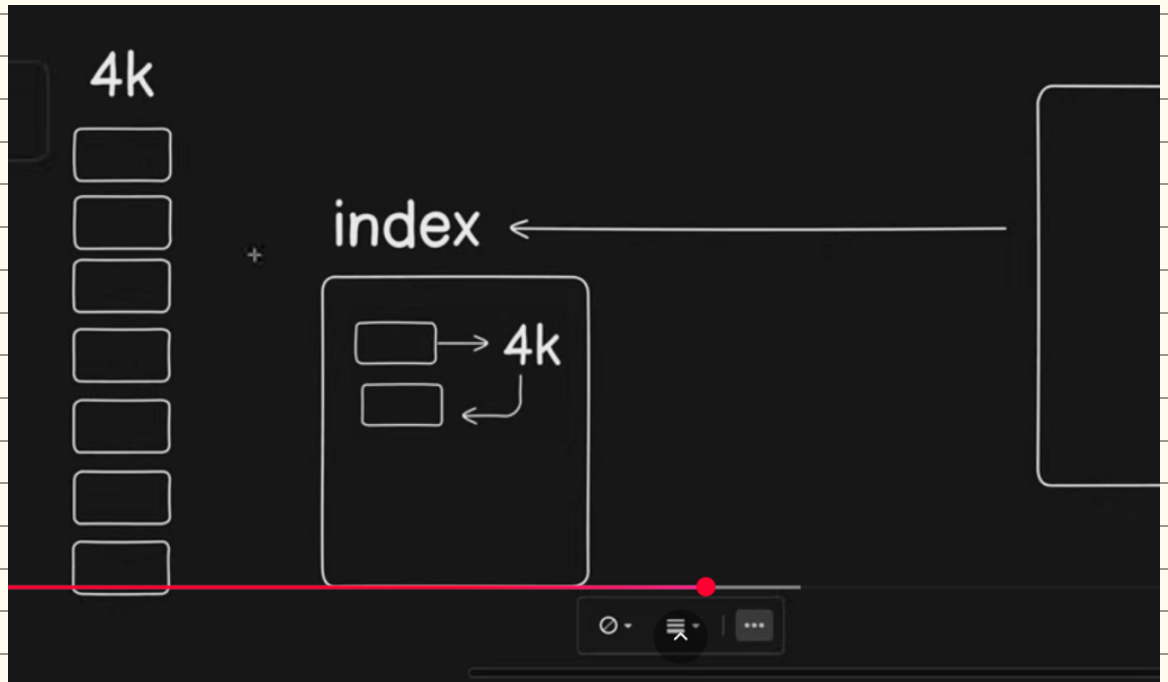


→ in processing, it encodes the video into different segments.

→ **index.m3u8** :-

An index.m3u8 file is a manifest file used in HLS (HTTP Live Streaming).

It does not contain video data itself — instead, it lists and organizes the video segments that a video player should download and play.



The index file tracks which segment is of which format and where is its next segment.



# Adaptive Bitrate Streaming

Learn how to use Adaptive Bitrate Streaming for long videos with ImageKit.

Copy page

Adaptive Bitrate Streaming (ABS) enables the optimum streaming video viewing experience for different types of devices over a broad set of connection speeds. This results in very little buffering, a fast start time and a good experience for both high-end and low-end connections.

The client (e.g. a video player) loads the manifest file and then chooses an appropriate segment, usually starting from the lowest bit rate stream to speed up initial playback. If network throughput is greater than the bit rate of the downloaded segment, it requests a higher bit rate segment. The client continues to adapt to changes in network throughput to ensure a smooth playback experience. The exact algorithm of choosing which segment to load can vary from client to client, but fundamentally it remains the same -- choose the appropriate segment and adapt as network (or other device constraints) changes.

For ABS to work, the client needs a manifest file that contains information about segments of different variants at varying bitrates. ImageKit can generate and deliver all necessary variants and manifest files from a single source video that is accessible through your ImageKit's account. The original video can be hosted in the ImageKit Storage or external storage integrated with ImageKit. Extra storage created because of generated variants and manifest files is counted towards your media library storage.

ImageKit supports the following streaming protocols. Both leverage existing HTTP infrastructure including CDN caching.


- HTTP Live Streaming (HLS)
- Dynamic Adaptive Streaming over HTTP (MPEG-DASH) protocol

*imagekit.io → now simplified Adaptive Streaming.*

Get ready for streaming in seconds, not months

URL-based parameters to generate DASH or HLS streaming manifests in near real-time that work with any video player. No extensive pipelines or cloud configurations needed to get started.

[Learn more](#)



```

HLS  DASH
<html>
<head>
<link href="https://unpkg.com/video.js/dist" />
</head>
<!--
Original video URL is https://ik.imagekit.io/ikmedia/sample-video.mp4/240p
We need to append /ik-master.m3u8?tr=sr-240p to the URL
Read more about it from the docs https://docs.imagekit.io/adaptive-streaming
-->
<body>
<video-js id="example-video" class="video-js">
<source
src="https://ik.imagekit.io/ikmedia/sample-video.mp4/240p-ik-master.m3u8?tr=sr-240p" type="application/x-mpegURL" />
</video-js>
</body>
</html>

```

Elements Console Sources Network Performance Memory Application

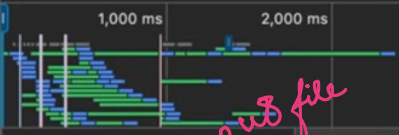
Preserve log Disable cache No throttling

m3

All Fetch/XHR Doc CSS JS Font Img Media Manifest Socket Wasm Other

Big request rows Group by frame

Overview Screenshots



*m3u8 file*

Name	Headers	Payload	Preview	Response	Initiator
ik-master.m3u8?tr=sr-240...	1. #EXTM3U				
480p-pl.m3u8?tr=sr-240...	2. #EXT-X-VERSION:6				
	3. #EXT-X-STREAM-INF:BANDWIDTH=208495,RESOLUTION=426x240				
	4. https://ik.imagekit.io/ikmedia/sample-video.mp4/240p-ik-master.m3u8?tr=sr-240p				
	5.				
	6. #EXT-X-STREAM-INF:BANDWIDTH=383350,RESOLUTION=640x360				
	7. https://ik.imagekit.io/ikmedia/sample-video.mp4/360p-ik-master.m3u8?tr=sr-360p				
	8.				
	9. #EXT-X-STREAM-INF:BANDWIDTH=579810,RESOLUTION=854x480				
	10. https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-ik-master.m3u8?tr=sr-480p				
	11.				
	12. #EXT-X-STREAM-INF:BANDWIDTH=1030932,RESOLUTION=1280x720				
	13. https://ik.imagekit.io/ikmedia/sample-video.mp4/720p-ik-master.m3u8?tr=sr-720p				
	14.				
	15.				

2 / 129 requests 0.0 kB / 1,68

More videos (v) | Elements | Console | Sources | **Network** | Performance | Memory | Application

🔍 | 🔍 | ☒ Preserve log | ☐ Disable cache | No throttling | 📶 | ⬆️ | ⬇️

📁 m3

All | **Fetch/XHR** | Doc | CSS | JS | Font | Img | Media | Manifest | Socket | Wasm | Other

☐ Big request rows | ☐ Group by frame

☒ Overview | ☐ Screenshots

1,000 ms | 2,000 ms | 3,000 ms | 4,000 ms | 5,000 ms

Name	X	Headers	Payload	Preview	Response	Initiator	Timing
ik-master....	14				https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-segs_00003.		
480p-pl.m...	15				#EXTINF:3.040000,		
	16				https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-segs_00004.		
	17				#EXTINF:3.040000,		
	18				https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-segs_00005.		
	19				#EXTINF:6.080000,		
	20				https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-segs_00006.		
	21				#EXTINF:3.040000,		
	22				https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-segs_00007.		
	23				#EXTINF:3.040000,		
	24				https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-segs_00008.		
	25				#EXTINF:2.520000,		
	26				https://ik.imagekit.io/ikmedia/sample-video.mp4/480p-segs_00009.		
	27				#EXT-X-ENDLIST		
	28						

More videos V

2 / 129 requests | 9 characters selected

Segments of  
a 480p  
file.