STREAMING TECHNOLOGIES

CSEN 233 - COMPUTER NETWORKS

AUTHOR:
SIMRAN DHAWAN
7700011429

UNDER THE GUIDANCE OF
Dr. Keyvan Moataghed

Audience

The audience to this report are people having acquaintance with communication technology and want to know about streaming technologies. This report is specifically targeted to computer engineers, IT developers, system architects, system engineers and any technology enthusiast who wants to understand streaming technologies and its protocols.

The reader is expected to have basic knowledge about computer networks to move forward with this report.

The report covers the overview about streaming technologies protocol. It talks about the most common protocols used for streaming video. It also explains the difference between video on demand and live streaming. The report also covers the example of YouTube streaming service to explain further and make users clear about the applications of streaming protocols discussed.

However the report gives the basic idea about different streaming protocols but it does not explain the exact procedure to code those. It does not provide the commands or how to code the streaming protocols.

TABLE OF FIGURES

TITLE TABLE

1. INTRODUCTION

The Streaming Technologies report discussed here is a research work for understanding basics about streaming technologies. The guide covers basic concepts of Streaming Technologies while developing the understanding about different streaming protocols used in sharing the videos to a person or a large number of users. Along with that the report discusses how video on demand is different from live streaming. It gives an insight about what kind of streaming protocols, different companies like: YouTube, facebook, tiktok, instagram, netflix etc uses.
The report is organized as follows:
Chapter 1 Introduction covers the idea about the topics that are discussed in this research.
Chapter 2 Introduction to Streaming Technologies covers the basics about the streaming technologies and why they are required? How have streaming technologies evolved and what is their history? It discusses why streaming protocols were required to share video from one computer to another computer?

Chapter 3 Common Protocols for Streaming Services covers the different types of protocols used for streaming, their basic architecture and their advantages. Here, HLS (HTTP Live Streaming), RTMP (Real- Time Messaging Protocol), SRT (Secure Reliable Protocol), MPEG-DASH (Dynamic Adaptive Streaming over HTTP), MSS (Microsoft Smooth Streaming) and WebRTC (Web Real - Time Communication) protocols are discussed.

Chapter 4 Live Streaming v/s Video on Demand Streaming dives into discussion about what is the difference between video on demand and live video streaming. What are the different types of challenges in two types of streaming and how have they evolved over the years?

Chapter 5 YouTube: An Example It discusses about YouTube how it uses different protocols. What is the role of encoders? How big is this streaming service platform? How does a client request for streaming video?

Chapter 6 Conclusion and Future Aspects provides the conclusion of understanding developed while creating the report and discusses what work is in progress in developing the streaming technologies in today's world and what more improvements we can expect in this field?

2. INTRODUCTION TO STREAMING TECHNOLOGIES

2.1 WHAT ARE STREAMING TECHNOLOGIES

In today's world everyone is using some kind of streaming service. From students in school to state officers, from doctor to engineer, everyone is using video conferencing which is an important application of streaming. We use Netflix, Facebook, YouTube etc for entertainment. Even the doctors provide consultation via Zoom streaming. So streaming of content is everywhere and that is why to share media or videos over the internet, streaming technologies have been developed over the years.

The streaming multimedia is different from downloading because the complete multimedia file is not present but only a portion of media is present which is being played or streamed while the rest of video is not transmitted. Earlier we used to have CD/DVDs for videos but this is video over the internet. Since CD/DVDs data is available on the device so there is no issue of latency. Even when we want to watch after downloading a video then also latency does not matter but when we want to stream over the internet, i.e. watch without having full video on computer then latency also comes into picture along with security and protocols for video transmission and reception. When you are live streaming a match on ESPN or video calling your near and dear visiting another country, you would want to have zero latency. We will discuss further how streaming services can also be of two types. One kind of streaming service requires interaction as in zoom appointment with doctor or online class and another kind of streaming service is non-interactive as watching video on YouTube.

The video we want to transmit should be compressed or encoded so that it can be quickly transmitted. The video to be sent is divided into chunks from transmitter to receiver, so at the receiver side some kind of decoding algorithm should be present so that the receiver can reassemble those chunks of video and play them. So some standardized protocols are developed for streaming video which are known as streaming protocols.

2.2 HISTORY AND EVOLUTION

Streaming is the buzzword of the era. However it was not this common in the beginning of the 21st century. However, since the mid 20th century, there have been several experiments to play media over the computers but hardware and resources were limited because of very high costs. Streaming media was via communication over satellites which was displayed on our T.V. Basically, streaming protocols and streaming technologies have been present and were used in T.V., radio, transmitter, C.D., playback hard disks etc. But streaming over the internet has evolved mostly in the past decade with the advancement of CPUs and bus bandwidths with the evolution of 3G/4G/5G technologies, which provide bandwidth for real time computing for streaming. With the availability of Ethernet switches strong computer networks were evolved which led to availability of streaming in schools as well as corporations. But the sending of large amounts of data at real time is impossible so data compression algorithms are

developed which compresses video data at the transmitter and decompresses it at the receiver to be available to the end user in understandable format.
Today without the streaming services we can not imagine life but the services like Netflix, Amazon Prime etc have evolved over the last decade only.
In 2007, Apple Iphone and Netflix launched streaming services. In around 2010, instead of cable TV, smartTV started attracting attention. In 2011, Amazon Prime Video was launched and in 2012, around 10M authenticated devices watched the Summer Olympics via streaming platform which was revolutionary. Then in 2013, Google launched Chromecast and by that time online video was over 50% of total internet traffic. It was one big milestone in the streaming industry.
In around 2015, almost 80% of people got the broadband connection at home and 40% of smartphones were used for streaming videos by the consumers. Amazon launched the Fire TV with its voice search. Now having TV at home means smart TV or TV using Firestick. By 2018, around 76% of households in the USA have internet connected TV and average time spent on TV media increased by 15%. OTT (Over-the-top) media subscribers have almost 182M which are greater than 50% of total population. Now, even OTT companies have started spending billions on their own content and this business is blooming among people of all ages.[2]

3. COMMON PROTOCOLS FOR STREAMING SERVICES

3.1 HTTP LIVE STREAMING (HLS)

3.1.1 Introduction

HLS is a communication protocol. It is an HTTP based protocol which was developed by Apple. It is a widespread protocol used in media players, mobile devices, web browsers and streaming media servers.
It supports both types of streaming: on-demand streaming and live streaming.

3.1.2 What are Benefits?

a) **HTTP based Protocol**
HTTP (Hypertext Transfer Protocol) is an application layer protocol where messages are exchanged in the format of request and response. It is used for transferring the data between different devices connected to the network. However, while we are streaming using HTTP protocol, initial channel or connection setup is done using request-response format and after that channel remains open and the client need not send a request message. The server will push the data segments to the client in the opened channel.

The implementation of HTTP based Protocol for streaming is very easy at both client side as well as server side because all the internet connected devices are having support of HTTP. This is the reason it is found to be the most popular streaming technology protocol in the Annual video industry survey in 2022.

b) Adaptive Bitrate Streaming

Adaptation of Bitrate while streaming, based on bandwidth as shown in below fig. 1. is called the adaptive bitrate streaming. Here, at server side the content to be transmitted is encoded at different bit rates by encoder. The client can request for high bit rate or low bit rate depending on bandwidth of the channel. However, initially the client requests for lower bitrate only and if it finds out that the throughput by network is larger that the downloaded segment it will request for higher bitrate. And if the client finds that the network deteriorates, it will again request a low bitrate segment. There are specialized Adaptive bitrate algorithms which are implemented by clients before requesting a segment.

Since HLS is adaptive bitrate streaming, the clients can enjoy the high quality video and also watch the video without noticing when bandwidth available is smaller, i.e. resources are less.

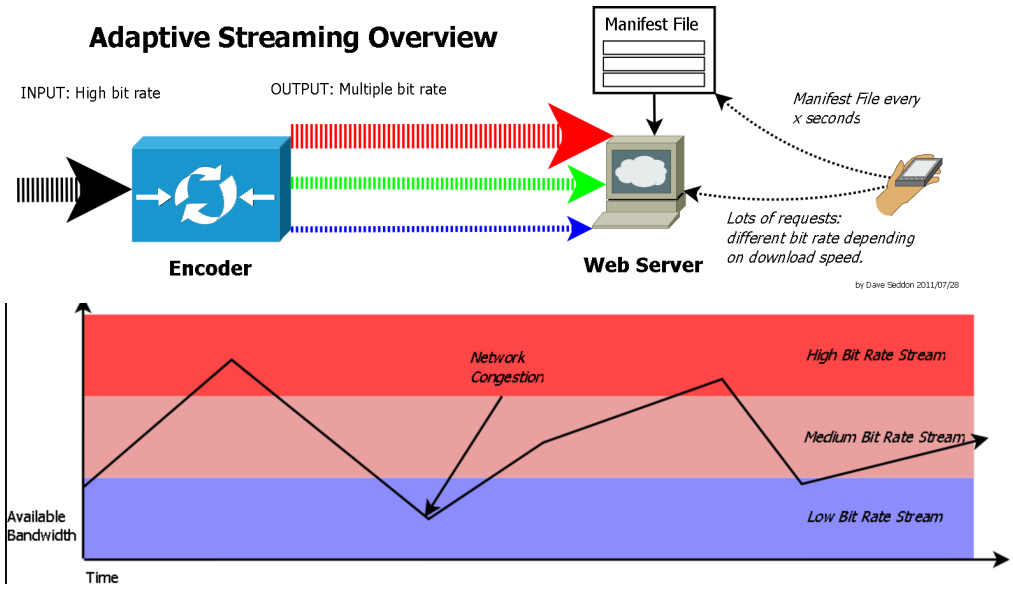

Fig. 1 : Adaptive Bitrate Streaming[4]

c) Digital rights management (DRM)

As we know that standard HTTP transactions support many features including traversal of any firewall or proxy server, so the HLS also supports the same. So, contents are shared between server and client via

already existing HTTP servers and using the HTTP- based content delivery networks. Encryption mechanism for content security is also present and source key is distributed using HTTPS (Hypertext Transfer Protocol Secure) which helps in managing the copyrighted content or providing access to digital content.

3.1.3 How does it work?

It works by dividing the stream of video and audio into a sequence of small HTTP-based file downloads. These sequences of HTTP files are encoded and transported at different bit rates and are sent to various clients using extended M3U playlists technology.[6]

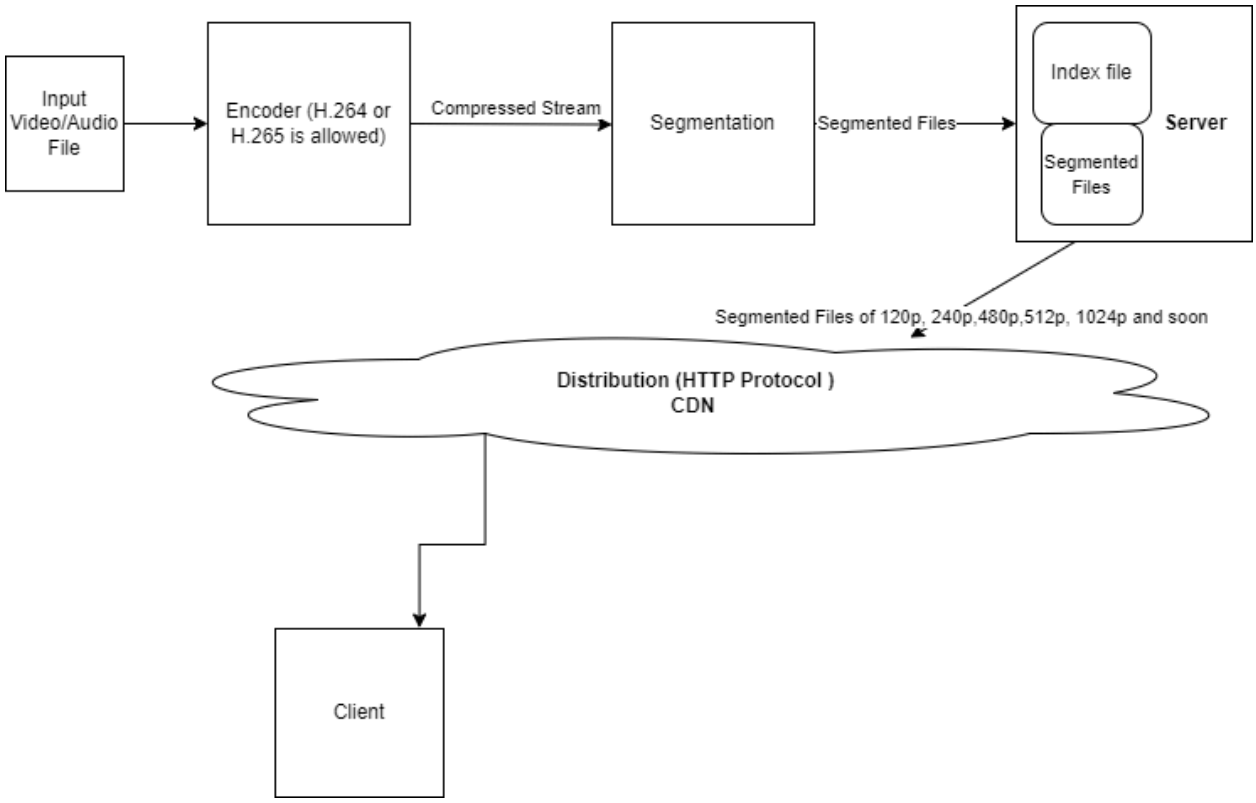

Fig. 2 : HLS Streaming Architecture

Fig.2 explains the architecture of HLS streaming. It has following components of steps in process:

1. **Server:** The server is the platform/device from where HLS streaming originates. It can be of both types: On-Demand streaming(if file is already stored in server, e.g. You tube videos) and Live streaming (e.g. Youtube Live). These are the steps which are performed at HLS server before transmitting the video at client:

    a. **Encoder:** The encoder will encode the video using H.264 or H.265 encoding which any device can use to interpret and recognize the video data.

b. **Segmentation:** The encoded videos are segmented into multiple segments of different lengths (different seconds). You might have noticed that while using YouTube if the internet connection somehow fails we are able to see a portion of video up to a specific length because that video segment has been received by the client.
c. **Index File:** This is a very important feature which tells in what order reassembly of videos at client side should be done. This file is generated at the server side and used by the client to stream the video.
d. **Redundancy:** Since HLS services use adaptive bitrate streaming, so the video supports different quality of videos: 240p, 480p, 720p, 1080p etc.

2. **Communication Protocol:** Content Delivery Network (CDN) protocol is used to distribute the encoded segments of videos and the caches are also present in the network to serve the client quickly.

3. **Client:** Referring index file the client assembles the video and it can be streamed over any device with HTTP based protocol. The video quality will depend on the service you have subscribed to and the network.

3.2 REAL-TIME MESSAGING PROTOCOL (RTMP)

3.2.1 Introduction

Real-Time Messaging Protocol (RTMP) is a streaming protocol for transferring audio, video or data over the Internet. It is a TCP based protocol. It is streamed via multiple variations:[7]

1. RTMP proper, which is “plain” protocol
2. RTMPS, have some kind of security(TLS/SSL)
3. RTMPE has propriety implementation, for Adobe’s usage
4. RTMPT Encapsulated version for carrying all above types of stream packets.
5. RTMFP which transmits RTMP packets but via UDP protocol . It is used for secure and real time media flow protocol developed by Adobe Systems.

3.2.2 3 step Process

We can learn about functionality of RTMP via three steps of process defined below:

1. **Handshake:**

The TCP connection has to be established, client and server do handshake in RTMP by exchanging three packets which is shown in fig. 3. Here we call packets as chunks.

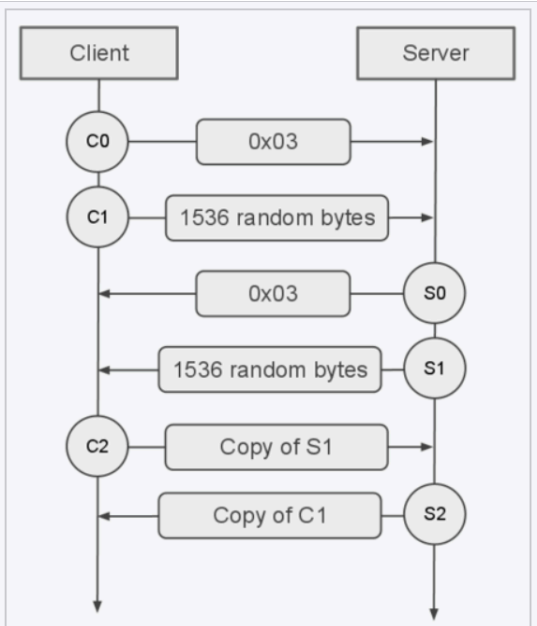

Fig. 3: 3 way Handshake of RTMP message[7]

a) In the first chunk, the client will let the server know about which protocol version is being used for this communication.
b) On another chunk, without waiting for any acknowledgement it will send the random number which is basically the timestamp at which request is sent.
c) The server acknowledges by sending the timestamp at which it receives.
d) The client sends a third packet containing a timestamp(received by the server and one with a client timestamp).
e) After this packet, the handshake is complete.

2. Connection

The TCP connection is created and messages are encoded in AMF (Action Message Format) messages where header and body are of format shown in fig. 4.

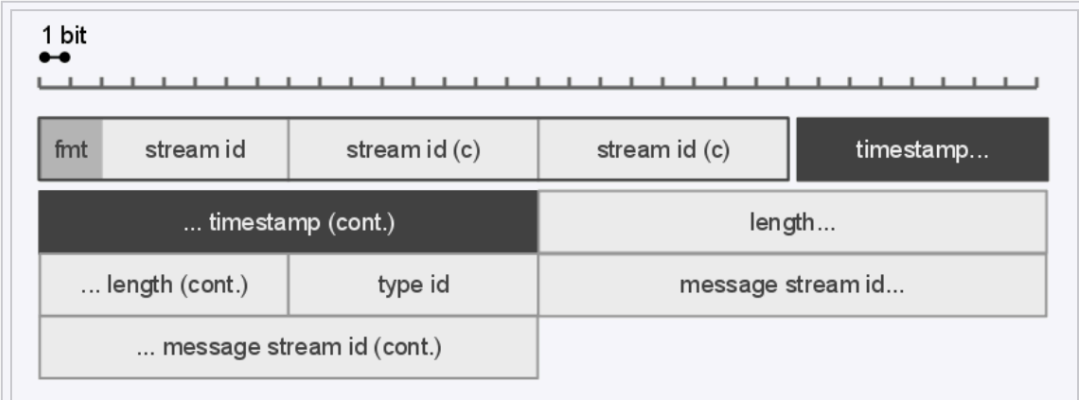

Fig. 4: Packet Structure of RTMP[7]

Here headers are of two types: Basic header which is constant and chunk header of message which is identified by message stream id which gives information about specific features of the packet whether it is acknowledged or controlled or video or audio packet etc.

3. Stream

CreateStream, Ping and Play are three types of messages that can be sent by client to server for streaming.

3.2.3 RTMP advantages:

1. **Low latency:** Even with less stable internet connection experience is good as it is providing low latency so streamed video is quickly resumed even in bad network conditions.
2. **Adaptable and Flexible:** It is flexible because it supports different video formats and audio,video and text are blended seamlessly. It is adapted as the user is not locked to watch the full video. We can start a video from any portion, skip or forward it. Live streaming events can also be seen after it has started.

3.2.4 RTMP disadvantages:

1. Incompatible to HTTP: RTMP stream can not be transported directly over HTTP. To support this, web browsers should have a special server, like browsers should have Flash media player.

2. Bandwidth Issues: Users experience is ruined because at low bandwidths it causes interruptions, unlike previous protocols which demand for a basic level for bandwidth before setting the connection.

3. HTML5 incompatible: HTML5 players are becoming the standard in today's world whereas flash players are obsolete. RTMP will need a converter to make it HTML5 compatible.

3.3 SECURE RELIABLE PROTOCOL(SRT)

3.3.1 Introduction

It is an open source UDP based protocol. However it uses UDP based protocol it is same as reliable as any TCP based protocol. It provides control and reliable transmission at the application layer not at transport layer as it is a UDP based protocol. Its packets are encrypted via AES (Advanced Encryption Standard).
However, it was designed for fast transmission of files reliably, using the method for sequence numbers, acknowledgements, retransmission, selective or NAK- based retransmissions.
SRT can provide source controlled latency or speed control with help of timestamp based datagram delivery and usage of buffers at both server and client side which reports if a packet is dropped during the communication.[10]

3.3.2 SRT Features and Functionality

3.3.2.1 UDP Protocol

User datagram protocol is transport layer protocol which is connectionless. Here the messages are transmitted as datagrams and no prior connection or path is set up between source and destination. There is no handshaking mechanism as we have seen in RTMP which uses TCP protocol. It provides data integrity by usage of checksums and other encoding mechanisms.[14]

3.3.2.2. ARQ Request

Automatic Repeat Request protocols are used for providing reliability in this UDP type streaming implementation protocol so that speed is achieved but reliability is also ensured. The client will request for retransmission of the packet if it finds a packet missing after a specific threshold time.

3.3.2.3. AES (Advanced Encryption Standard)

AES is a symmetric-key block cipher and very easy and fast software and hardware based implementation which helps in streaming videos securely preventing unauthorized access to users.

3.3.2.4. Adaptive bit rate

This protocol can support variable bitrate depending on network conditions in order to provide best quality while reducing the latency.

3.4 DYNAMIC ADAPTIVE STREAMING OVER HTTP(MPEG - DASH)

3.4.1 Introduction

DASH, Dynamic Adaptive Streaming over HTTP is a streaming technology in which data is delivered before it is fully loaded at client device. It is very similar to HLS service in terms of functionality. Here, also streams of data are broken down before getting transmitted over the internet via HTTP.

SProxy is one such streaming system for streaming videos via HTTP. It showed the beneficial effects of breaking in video in segments before streaming. This reduces the time and experience at the user end if the user wants to watch only a few portions of videos. Also segments should have some playback time so that if the user wants to rewatch, they should be available on the device for various types of live events the playback time is for hours, e.g. sports event, news etc. The content is available in a range of bitrates. Whenever, MPEG client requests for streaming video, the client also has a specific kind of mechanism, bit rate adaptation by which they choose which bit rate to choose so that the clients do not face unnecessary stalls due to the network. MPEG DASH client dash.js which use following types of algorithms for adjusting to network and providing best quality of videos:

1. Buffer based (BOLA) bit rate adaptation which is one of the optimal bitrate adaptation protocols. It uses a buffer to check if the buffer is full to adjust the bit rate. Buffer level is an indicator of bit rate.
2. Dynamic bitrate adaptation is a technique which is used in HLS as discussed above where depending on network bandwidth bit rate is adjusted.

3.4.2 What are Benefits?

a) **Open Source:** Libdash is a tool that helps developers build applications using DASH. Libdash is a software library which is a reference for DASH technology and it provides an object oriented interface for MPEG-DASH streaming.
b) **HTTP based Protocol:** HTTP protocol comes with its own advantage as already servers are available for handling it and it is secure and not blocked by firewalls. Since, the caching mechanism is already present for HTTP/CDN, it is fast to access.

c) **Dynamic Bit rate streaming:** Depending on network conditions the DASH technology can switch to different bitrates providing users with a good and seamless streaming experience.

3.4.3 How does it work?

Fig. 5 shows the architecture of MPEG-DASH. It has same basic functionality as HLS which is explained in following steps:

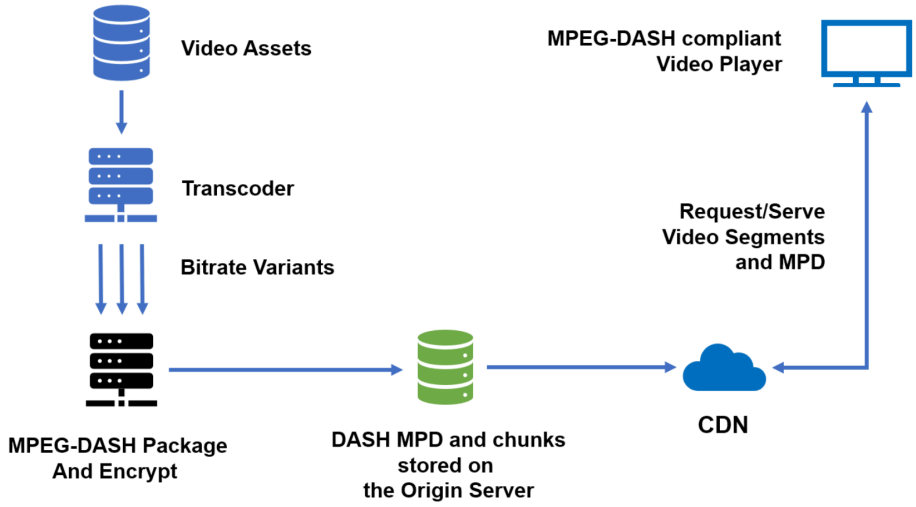

Fig. 5: MPEG-DASH Architecture[16]

a) Video assets are fed into the transcoder or encoder.
b) The transcoder encodes the video with any standard and the complete video is segmented into different pieces of segments of size 2-4 seconds.
c) Here MPD (Media Presentation Description) document is created which records how video is being segmented so that during decoding it can be used. MPD or Manifest DASH is one important part of MPEG DASH as it creates the appropriate HTTP requests to be sent over the network.
    i) MPD Format: XML schemas are used for MPD which are very complex, so at server side it is developer responsibility to create an accurate MPD corresponding to the video.
    ii) MPD Encoding: UTF-8 encoding is used for MPD.
    iii) MPD MIME Type: Standard content-type header of HTTP request that is used for streaming is MIME type/subtype. The MIME type is defined so that the browser knows how to process this request the same as we provide file extensions for the file so that the app can understand the same. DASH MPD servers should be configured with application/dash+xml type.

Parts of MPD File

MPD file structure is shown in fig. 6, below. It can be explained from steps below:

1. MPD file is actually a sequence of Periods which is basically indexing for the parts of video measured in seconds
2. One period can contain different adaptation rate sets.
3. Adaptations sets have different representations which is basically the quality of video that is visible to consumers.
4. Representation sets can contain the segments
5. Segments of MPD contain the actual data and corresponding metadata.

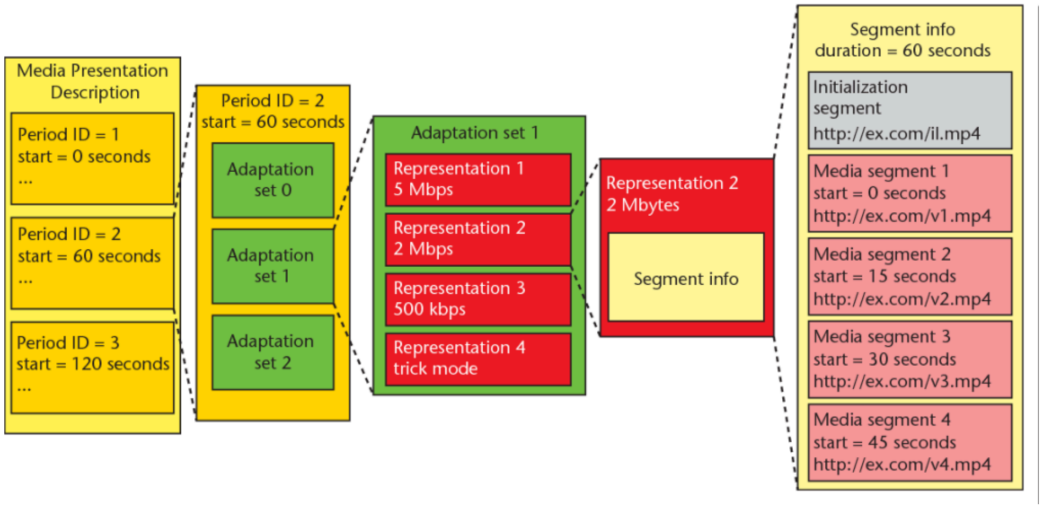

Fig. 6. MPD File structure[17]

d) The video segmented in packets and packaged along with Manifest DASH is stored at webserver and ready to be streamed over CDN(Content Delivery Network).
e) At the client side, the client should have an MPEG-DASH complaint player which can apply adaptive bitrate algorithms and select the appropriate bit rates.
f) The MPD file is parsed by MPEG-DASH player and then video is streamed.
g) The MPEG-DASH player will monitor the bandwidth at regular intervals and will select the appropriate bitrate for streaming in the player.

3.4.4 MPEG-DASH v/s HLS

MPEG-DASH and HLS are widely and most commonly used streaming technology. They have similar functionality. Table 1 shows the differences between these two technologies.

Parameter	MPEG - DASH	HLS
Encoding Format	MPEG -DASH does not have any specific encoding format. It supports all formats.	HLS only supports H.264 or H.265 encoding format.
Device Support	MPEG-DASH is not supported by many devices, especially apple devices as the client needs to read the MPEG manifest file.	HLS is widely supported all over the world.
Segment Length	MPEG-DASH segment length is of the range 2-10 seconds.	HLS segment length range is of 6-10 seconds(generally larger than MPEG-DASH)
Standardization	It is International Standard Protocol	Even widely supported it is not international standardized as developed by Apple

Table 1: Difference between MPEG-DASH and HLS

3.5 MICROSOFT SMOOTH STREAMING (MSS)

Microsoft Smooth Streaming service is used to stream content onXBox 360, Windows Phone 7, Microsoft Silverlight, TV platforms etc. Here adaptive bitrate streaming is supported.
Microsoft Zencoder is used to encode the files which can be audio or video and they are segmented but here the number of segments created are less in number. Also if we have segments, client and server manifests or index files should also be required. The request is HTTP based request as shown in fig. 7 below, so the same HTTP servers can be used for caching audio or video which are used for other HTTP traffic.

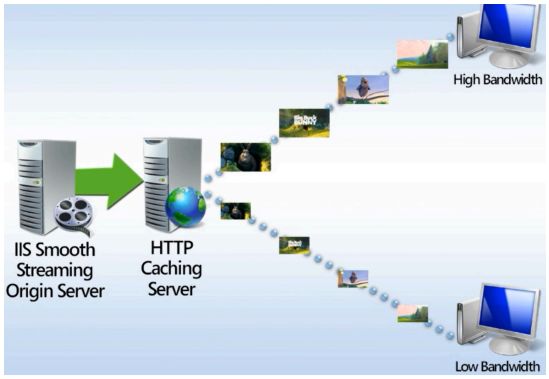

Fig. 7: ISS Streaming in Microsoft Streaming[25]

It is an ISS Media Extension and format specification of Microsoft Streaming Service is based on ISO base media file format and Microsoft has standardized it as Interoperable file format in

protected mode. For smooth streaming Microsoft generally provides an additional smooth streaming client software development kits for its own system as Windows Phone 7 and Silverlight. However for other clients operating systems as Apple iOS or Linux or Android etc smooth streaming kit is provided. Here, with the help of this smooth streaming kit, the encoded audio or video files are reencoded to Apple HTTP Adaptive Streaming format and then delivered to iphone or ipad or other streaming devices.

3.6 WEB REAL-TIME COMMUNICATION(WebRTC)

It is an open source protocol which supports real time communication to the application which uses this technology. It is prominently used for audio, video streaming or any generic data between different users. It uses web based APIs
mostly JavaScript API available on major browsers.

3.6.1 Architecture

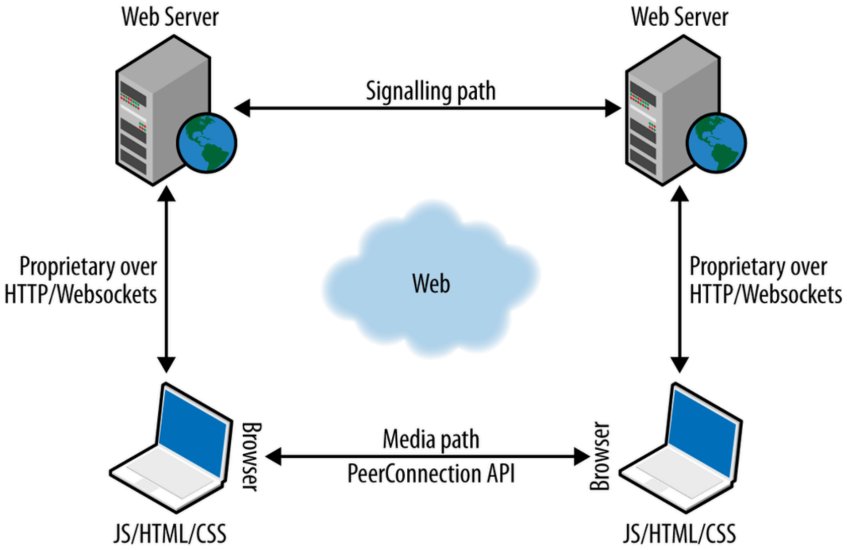

Fig. 8: WebRTC : An architecture[24]

The most fascinating thing about its architecture as shown in fig. 8, is its ability to provide peer to peer connection.
Two browsers on two different client devices are running the two different applications downloaded from different web servers. Two types of connects can be seen:

a) Between the servers we can see that there is signaling based communication. Signaling mechanisms like: SIP or Jingle are established and web servers initiate or terminate the connection and communicate with each other using these signaling protocols.

b) Then there is a data path, PeerConnection which is directly between client devices which allows streaming or media or file content directly between the users.

3.6.2 WebRTC Designing

WebRTC (Web Real-Time Communication) can even be used for peer to peer file sharing along with voice or video calling. Here are three parts of interfaces by which it requests for :

a) **getUserMedia():**
   It allocates resources to device, eg. granting access to camera or microphone and requesting RTC signal connection.
b) **RTCPeerConnection():**
   It is an API using which video or voice calls are configured.
c) **RTCDataChannel():**
   It provides an actual data channel between peers or we can say it is the video or audio connection and channel between users of the network.

3.6.3 Applications

WebRTC was earlier developed for web browsers. However, it is now used for non web browser based services, eg. It is a common protocol in IOT(Internet Of Things) based devices, Web torrents for sharing files peer to peer. In a few CDNs(e.g. Microsoft's Peer5), the client acts as an edge server and files are transferred from one client to another connected client using this streaming protocol.

4. LIVE STREAMING V/S VIDEO ON DEMAND STREAMING

Live streaming, as the name suggests that the video is transmitted in real time without saving and recording first. Here, video is saved upfront while transmitted or broadcasted to other users using TCP/IP protocol. TCP/IP protocol is a packet switch reliable service. On the other hand, video on demand is video recorded beforehand and already present in the server and transmitted or streamed on demand basis. Fig. 9 shows the block diagram functionality of video on demand and live streaming technologies.

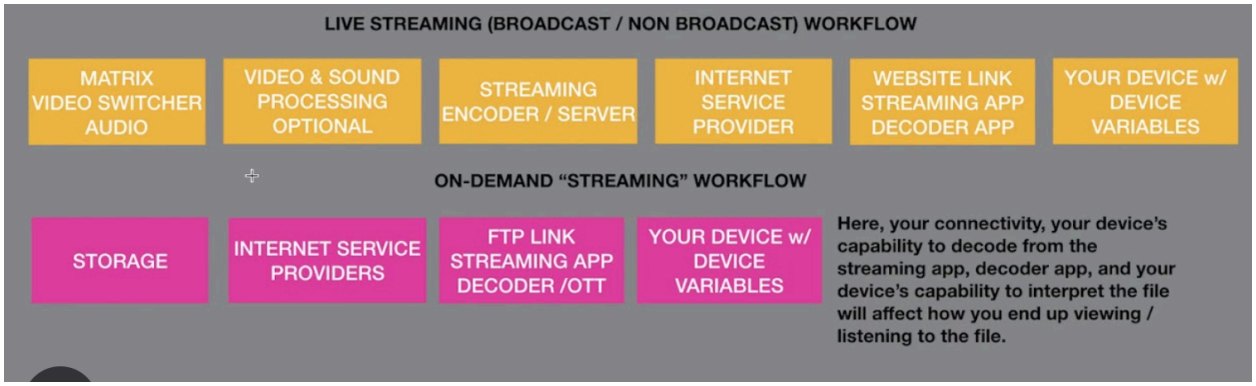

Fig. 9: Live streaming v/s Video on demand[18]

In Live streaming, the video and sound processing is optional before streaming the encoder/server. Here, ISP(Internet Service Provider) will transmit the encoded and segmented videos. The website link or app has a decoder for decoding the encoded video. All of it is done in real time.
However, on demand streaming platforms have stored contents and ISPs via FTP(File transfer Protocol Link) requests for video and our client device should have the proper decoders to stream the downloaded files.

Feature	**Video on Demand (VOD)**	**Live Streaming**
Protocols commonly used	HLS, MPEG-DASH	HLS, MPEG-DASH, RTMP
Latency	Low to medium	Low (ideal), medium acceptable
Scalability	High	High, medium acceptable
Reliability	High	High, medium acceptable
Security	High	High
Typical use cases	Movies, TV shows, pre-recorded content	Live events, webinars, broadcasts

Table 2: Difference between Video on Demand and Live Streaming

Table 2 shows the difference between two technologies in terms of protocol used, security, reliability, scalability, latency requirements and its use cases. We can conclude that HLS and MPEG-DASH, both are very good protocols for both live streaming as well as video on demand. Both of them offer low-latency, high scalability and high reliability. RTMP generally sends the video data in even smaller and manageable segments so it offers even less latency, so it is good for live streaming. The next section covers the different range of video latencies and how they are categorized.

4.1 Standard Categories of Video Latency

The delay is one big factor which we are talking about on streaming platforms. These are categories in which delay is divided according to the industry standards in streaming technologies:

1. **Standard Broadcast Latency:** Here expected delay is between 5 to 18 seconds. It is standard latency for TV and few Video-On-Demand platforms.
2. **Low Latency:** Here, the delay is between 1 to 5 seconds. It can be seen in cable TV or some video on demand platforms.
3. **Ultra-Low Latency:** It is used in real time streaming typically where interaction between users is required for example a zoom call or where two way chats are needed. Its typical delay is less than 1 second.
4. **Real-time Latency:** It is a type of latency where we won't feel delay as delay is of less than 1 second. It has applications such as when two musicians are collaborating online and playing music as a team for the event.

5. YouTube : An Example

YouTube is one of the largest online video sharing platforms which is owned by google. It is the second most visited website in the world after Google Search. YouTube has initially started as just a method to share the videos. Later on, after being acquired by google it started the "Broadcast Yourself" Era. Around 2019, 400 hours worth of videos start getting uploaded in a minute. It became available as a social video sharing platform with ads and a premium service which was without ads. In covid time its usage further exceeded.

5.1. Components

The fig. 10, below gives the high level components of YouTube.

There is a user which requests for video from the web server. If some video is already cached in CDN network then the web server can deliver the video from there otherwise the web server will need to upload the video from the video storage location.

The video before uploading should be processed in the form of segments and Video transcoding is done to make available different quality of videos and depending on network traffic the video quality can be streamed.

At the client-side, there is a recommendation system which through training of users location, history and other information provides recommendations. Machine Learning modules are implemented to analyze metadata of video and user interactions with the video so that relevant information or content is coming in feed for a particular user account.

Load balancer deployed, will help in dividing the traffic among several servers present in the form of clusters, so that resources can be utilized in an optimized way and reducing the probability of the system getting overloaded.

Data Storage systems store metadata and history and analytic data (required for monitoring the health, performance etc. of the system) of user accounts along with video

content. For security of data sent encryption mechanisms are also employed while transmitting videos over the internet. API server basically provides an interface to third party services, like advertising, social networking platforms like facebook, instagram etc and content management system.

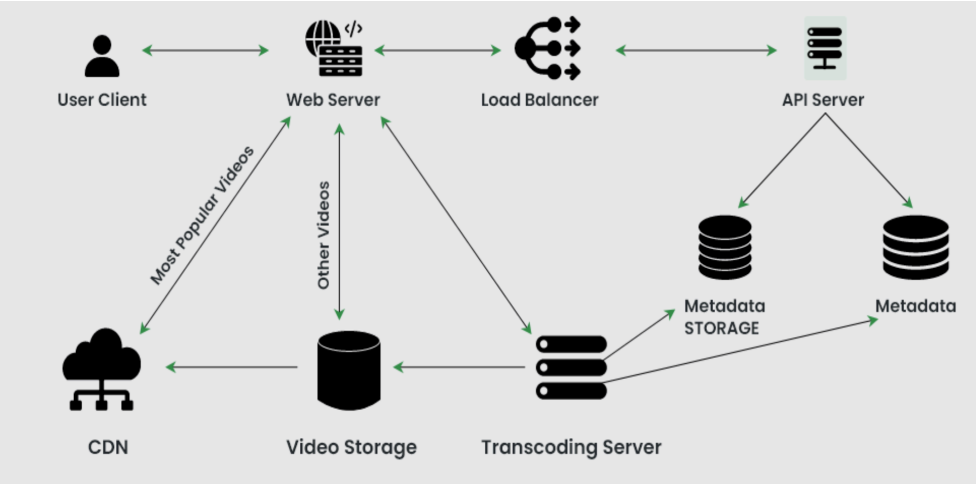

Fig.10: YouTube System[21]

5.1.1 Encoders

The type of protocol that YouTube uses depends highly on quality and latency requirements. Also the capability of streaming encoder and other tools like network quality, security needed etc. decides the protocol to be used.

YouTube uses HTML5 protocol which is a variant of HLS or DASH(Dynamic Adaptive Streaming over HTTP) protocol. However, HLS, MPEG-DASH, RTMP, RTMPS ingestion protocols are also used. Ingestion protocols are the protocols which are used to transport video from the encoder to the streaming server.

The VP9 and H.264/MPEG AVC video codecs are used. For increasing the quality of video content while keeping the bandwidth less, AV1 hardwares and encoders were also started getting used in recent years.

In earlier days for YouTube video to be played on the system we would need Adobe Flash Player installed in our web browser. However around 2010, using HTML5 and via MPEG-DASH protocols the browser lost the dependency on Adobe Flash player and they were equipped to stream video without separate extensions. We know that MPEG-DASH has adaptive bitrate streaming so high quality video and less latency of videos were ensured depending on network quality. Fig. 11 shows that YouTube can support different ranges of video quality. Here, video quality can go as low as 144p for limited cellular data plans and as high as 1080p.

YouTube also has started supporting 360 degree live streaming videos with 4K users. For its premium subscribers it provides premium 1080p Premium option to its iOS users.

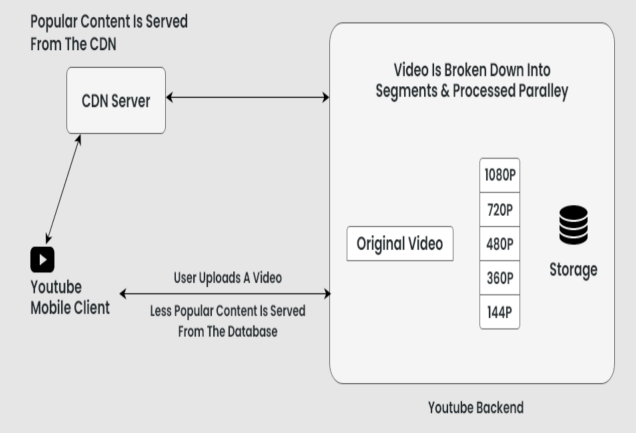

Fig. 11: YouTube Backend different rates and CDN cache support[21]

5.2 How does it work?

The videos can be uploaded by clients and made available publicly.
Whenever someone wants to watch a video it can send HTTP GET requests to the server. Fig. 12 shows the sending and processing of video requests on browser.
HTTP request which is sent:
GET /get_video?My_demo_video.
A particular video is identified by some unique video identifier. Here it is: My_demo_video (I have kept a simple , understandable name for reference here as compared to fig. 12 for better understanding).
First the video will be searched in cache or CDN server, if server did not find the video it will give HTTP 303 which is a redirection request.
The redirection request will also have the location to which the request to be redirected. Here the load balancing mechanisms also come in picture. E.g. If I am searching for a video whose location is Europe, we will be routed to a server in Europe and then in Europe location in London , then again redirection to London will be received.
The redirection request can be in format:
HTTP/1.1 303
Location: http://london.youtube.com/get_video?video_id=My_demo_video
The data is then streamed to the client.

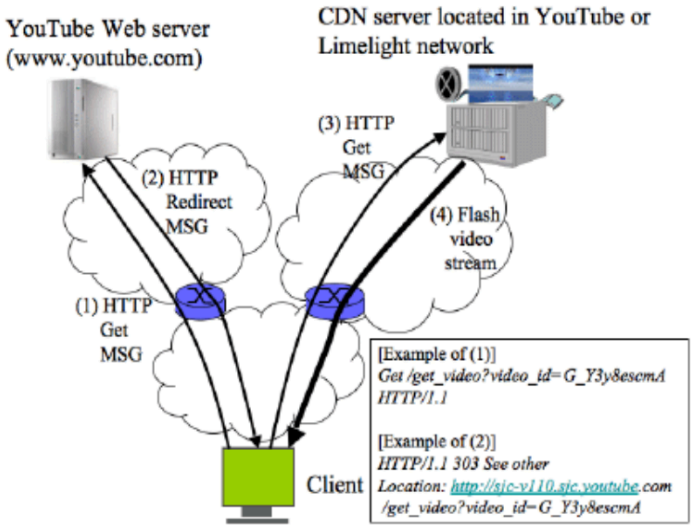

Fig.12: How YouTube Works[22]

6. CONCLUSION AND FUTURE ASPECTS

6.1 CONCLUSION

Streaming technologies are rapidly evolving since last decade and they have revolutionized the entertainment industry. From live events to TV shows, streaming technologies are used in many fields. We have discussed six main types of streaming protocols all of which are useful and have varied applications. We say that encoding and segmentation of videos is an important aspect which is common to all the protocols. We see that HLS, MPEG-DASH provide streaming with HTTP on the application layer and TCP protocol on the transport layer. They have proven to be excellent in adaptive bitrate streaming and are widely used. RTMP is Real Time Message Protocol which provides a low latency transport layer solution but uses flash players and is incompatible with HTML5. We have discussed SRT protocol which is a secure and reliable protocol. It is highly scalable and used widely for broadcasting with safety. WebRTC is a protocol which is as niche as SRT but it is providing low latency solutions used for real time applications and IOT devices. We also discussed the video processing is different and requirements are different for Video on Demand and Live streaming technologies. Then we discussed YouTube which is a big player in the streaming industry and we learnt that it generally uses HTML5 protocol but ingest protocol for it can be MPEG-DASH, HLS, RTMP etc. The protocol used there depends on the encoder and other broadcasting tools used for sending the segmented container files.

6.2 FUTURE ASPECTS

In the coming years, with the advancement and evolution of the communication networks, eg. with the spread of 5G networks and different technologies like AR/VR we can see an evolution in the streaming industry. Artificial intelligence and Machine learning which provide personalized recommendations to users are supposed to enhance the experience on streaming services like Netflix, YouTube, Amazon Prime, Hulu etc. In this era of fashion and modernization, live streaming and interactive content can enhance the experience for users and they will feel that they are experiencing the music or event in person. With Virtual and Augmented Reality, the virtual world will become very close to the real world and many people will be spending considerable amounts of time in the AR and VR world. Streaming technologies will play a key role in the success of the fields mentioned above. Along with that I believe that the advancements in Artificial Intelligence and Machine Learning will revolutionize the video encoding and decoding methods which would lead to tremendous improvements in streaming technologies.

7. ACRONYMS

1. API - Application Programming Interface
2. ABR - Adaptive Bitrate
3. AI - Artificial Intelligence
4. CD - Compact Disk
5. CDN - Content Delivery Network
6. DASH - Dynamic Adaptive Streaming Over HTTP
7. DVD - Digital Versatile Device
8. HLS - HTTP Live Streaming
9. HTML- Hypertext Markup Language
10. HTTP - Hypertext Transfer Protocol
11. IP - Internet Protocol
12. IIS - Internet Information Service
13. ISP - Internet Service Provider
14. LTE - Long Term Evolution
15. MPEG-DASH - Moving Picture Experts Group - Dynamic Adaptive Streaming over Internet
16. ML - Machine Learning
17. MPD - Media Presentation Description
18. MSS- Microsoft Streaming Service
19. OTT - Over the Top
20. RTMP- Real Time Message Protocol
21. RTSP - Real Time Streaming Protocol
22. SRT - Secure Reliable Transport
23. TCP - Transmission Control Protocol
24. UDP - User Datagram Protocol
25. VoD - Video On Demand
26. WebRTC - Web Real time Communication

8. REFERENCES

1. https://www.dacast.com/blog/streaming-protocols/
2. https://en.wikipedia.org/wiki/Streaming_media
3. https://www.dacast.com/blog/video-streaming-technology/
4. https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming
5. https://www.cloudflare.com/learning/video/what-is-http-live-streaming/
6. https://en.wikipedia.org/wiki/HTTP_Live_Streaming
7. https://en.wikipedia.org/wiki/Real-Time_Messaging_Protocol
8. https://larryjordan.com/articles/basics-of-http-live-streaming/
9. https://blog-cloudflare-com.webpkgcache.com/doc/-/s/blog.cloudflare.com/stream-now-supports-srt-as-a-drop-in-replacement-for-rtmp
10. https://www.phoronix.com/news/SRT-Open-Sourced
11. https://www.magewell.com/blog/77/detail
12. https://bitmovin.com/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds
13. https://www.stackpath.com/edge-academy/what-is-real-time-messaging-protocol-rtmp/#:~:text=Real%2DTime%20Messaging%20Protocol%20(RTMP)%20is%20an%20open%20source,to%20stream%20video%20and%20audio.
14. https://en.wikipedia.org/wiki/Secure_Reliable_Transport
15. https://www.coremicro.com/blog/what-is-the-srt-protocol
16. https://teyuto.com/blog/what-is-mpeg-dash-history-pros-and-cons-of-dash
17. https://ottverse.com/structure-of-an-mpeg-dash-mpd/
18. https://www.youtube.com/watch?app=desktop&v=-aY-g_C-UZY
19. https://stackoverflow.com/questions/21179183/youtube-video-streaming-protocol
20. https://en.wikipedia.org/wiki/YouTube
21. https://www.geeksforgeeks.org/system-design-of-youtube-a-complete-architecture/
22. https://www.researchgate.net/publication/228634313_Watch_global_cache_local_YouTube_network_traffic_at_a_campus_network_-_Measurements_and_implications
23. https://www.yellowbrick.co/blog/entertainment/the-future-of-streaming-technology-key-trends-and-innovations
24. https://www.researchgate.net/publication/367554946_DESIGN_AND_IMPLEMENT_VIDEO_AND_CHAT_APPLICATION_USING_MESH_NETWORK
25. https://learn.microsoft.com/en-us/shows/iis-net-site-videos/smooth-streaming