T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

ADAPTIVE VIDEO TRANSFER OVER WI-FI
(PROJECT 1257)

Capstone Project Proposal

Zakaria Alzawawi ,

Ghaith Nakawah

Advisors: Prof. Dr. Mehmet Alper Tunga , Software Engineering

Asst. Prof. Ece Gelal Soyak , Computer Engineering

ISTANBUL, September 2020

STUDENT DECLARATION

By submitting this report, as partial fulfillment of the requirements of the Capstone course, the students promise on penalty of failure of the course that

- they have given credit to and declared (by citation), any work that is not their own (e.g. parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.);
- they have not received unpermitted aid for the project design, construction, report or presentation;
- they have not falsely assigned credit for work to another student in the group, and not take credit for work done by another student in the group.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WiFi	Wireless Fidelity
QoS	Quality of Service
IEEE	The Institute of Electrical and Electronics Engineers
QoE	Quality of Experience
MPEG	Moving Pictures Expert Group
DASH	Dynamic Adaptive Streaming Over HTTP
HTTP	HyperText Transfer Protocol
H265/HEVC	High Efficiency Video Coding
JSON	JavaScript Object Notation
XML	Extensible Markup language
HLS	HTTP Live Stream
MPD	Media Presentation Description
AP	Access Point
STA	Station
PN	Project Network
WBS	Work Breakdown Structure
RM	Responsibility Matrix

1. OVERVIEW

The project is about transferring video over wifi while reducing buffers with bad connectivity users.

Zakaria Alzawawi CMP: will design the algorithm and implement the server & client and he is the project integration responsible.

Ghaith Nakawah SEN: will implement the dashboard application that calculates and displays the video buffer statistics and he is the project planning and documentation responsible.

1.1. Identification of the need

Adaptive Video Transfer protocol (Over Wi-Fi), that solves the need of the viewer with bad connectivity to wait for the video to buffer, and on the other hand supply high video quality for those with good connectivity. Basically a protocol that maximizes the user experience with the given constraints while streaming on-demand videos.

1.2. Definition of the problem

These days on-demand videos exist in high quality for viewers to stream. However, viewers with bad connectivity still exist. In the era of speed, these viewers would prefer streaming with lower qualities instead of having to wait for the video to load. So an adaptive streaming protocol is needed to adjust the quality of the video based on the connectivity.

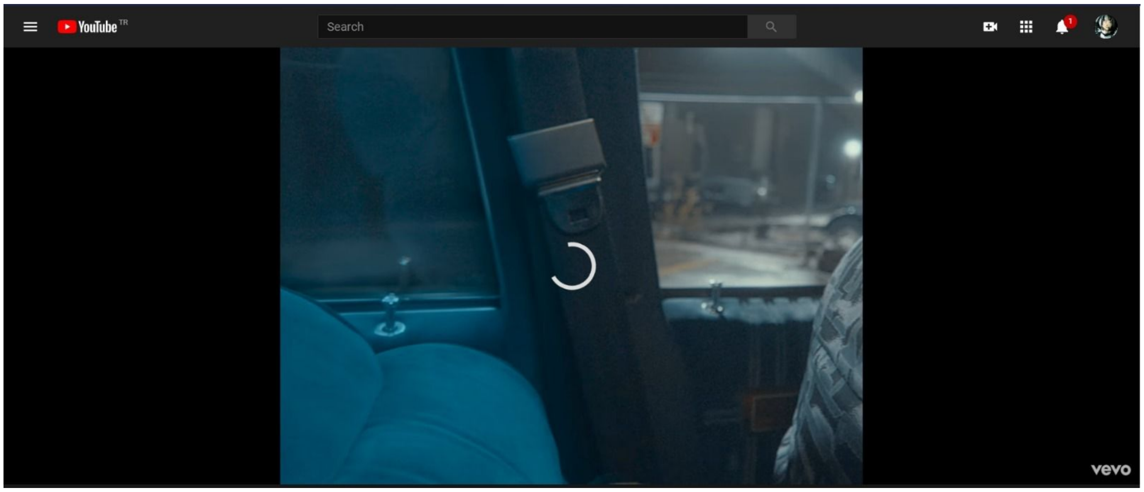

Figure.1 video buffering due to low connectivity

Functional requirements:

1. Video Transfer of Wi-Fi.
2. Display buffer statistics.
3. Automatically adjust streaming quality

Performance requirements:

1. The product should be able to measure the requesting user’s data rate in real time.
2. The product should be able to adjust the streaming quality with respect to the current data rate detected.
3. The product should be able to function within a specific buffer performance and calculate the buffer statistics accordingly.

Metrics:

[1] With respect to adaptive video streaming our performance metrics split into 2 sections, Quality of Service (QoS) and Quality of Experience (QoE).

Since the system depends on the user satisfaction, QoS measurements aren’t enough to measure the system’s performance, and they wouldn’t make complete sense. So in order to be able to measure the user’s QoE a set of QoE metrics should be defined to be able to measure the performance of the system. The QoE we are going to measure our performance based on are:

➠ Initial delay: It is the time the user has to wait between initiating the streaming session and the start of 2playing the video.Usually due to the reason that the buffer needs a minimum amount of content to start playing the video. (Measured in milliseconds).

➠ Buffering ratio: It is the ratio of total time the video is buffering due to the buffer being starved(empty).

➠ Rate of buffering: Total amount of times the video buffered with respect to the sessions length.

➠ Average bitrate: average amount of bitrate per millisecond.

➠ Rate of Quality switching: Total number of times the video quality changes with respect to the sessions length.

However, while QoE will focus on the aforementioned, QoS is needed to be able to measure such metrics. The QoS of metrics needed would be bandwidth, latency and packet loss. Mapping the QoS to QoE can be a little bit challenging, but for the sake of simplicity in this project, we will map a direct relation between both metrics. [1]

In conclusion, the client aims to obtain the best QoE by providing a fast start-up, fast video seek and reducing skips, buffering and jitters.

1.3. Standards and constraints

Standards and protocols :

1. MPEG Dash https://mpeg.chiariglione.org/standards/mpeg-dash
2. 802.11 wifi standard https://tools.ietf.org/html/rfc7494
3. HTTP (RFC 7540 (HTTP/2)) https://tools.ietf.org/html/rfc7540
4. H265/HEVC codec https://www.itu.int/rec/T-REC-H.265-201911-I/en
5. JSON and XML

Constrains

1. Buffer size: Buffer size is limited due to a limited amount of memory a client can have, this presents a challenge to the amount of segments the client can keep at once.
2. Packet-Loss: Wireless networks aren't reliable. Packets may be dropped for various reasons such as congestion or packet dropping by routers or switches. This means that some packets need to be retransmitted and that causes a delay.
3. Bandwidth: Even though 5G and 4G supply high bandwidth, bandwidth is still limited and is lower than theory values.
4. More constraints may be added as we progress further in the project.

1.4. Conceptual solutions

In recent years the delivery of audio-visual content over the Hypertext Transfer Protocol (HTTP) got a lot of attention and with adaptive streaming over HTTP. Many approaches have been implemented including DASH, iDash, and HLS. Also, companies such as adobe and apple have implemented their own solutions.(Adobe HTTP dynamic streaming, Apple HTTP live streaming) [2]

Our approach in this project would build on the MPEG-DASH Standard and complement it.

In adaptive video transfer the video is partitioned into short segments (couple of seconds long) and each segment is encoded at the requested quality (rate and format). Each segment isn't dependent on any other segment and decoded independently. Then the client sends a HTTP Get request that requests a specific segment with specific quality and the server then responds and sends that segment. The quality of the segment requested by the Get request would vary based upon the output of the adaptive video algorithm at the request time.

Many papers showed their research results, but unfortunately all of them use their own private dataset which is not publicly available. Therefore, it is almost impossible to compare, e.g., adaptation algorithms in an objective way due to the lack of a common dataset which can be used as the basis for such experiments. This means that a standard is needed hence we use MPEG-Dash Standard. [3]

The MPEG-DASH specification provides us with a standardized description of the content stored on a HTTP server.

The content which is stored at the server can be divided into two parts:

- Media Presentation Description (MPD) : a XML-based manifest file providing information about the content and its properties.

- Segments, the video is partitioned into segments and encoded at the wanted bitrate.

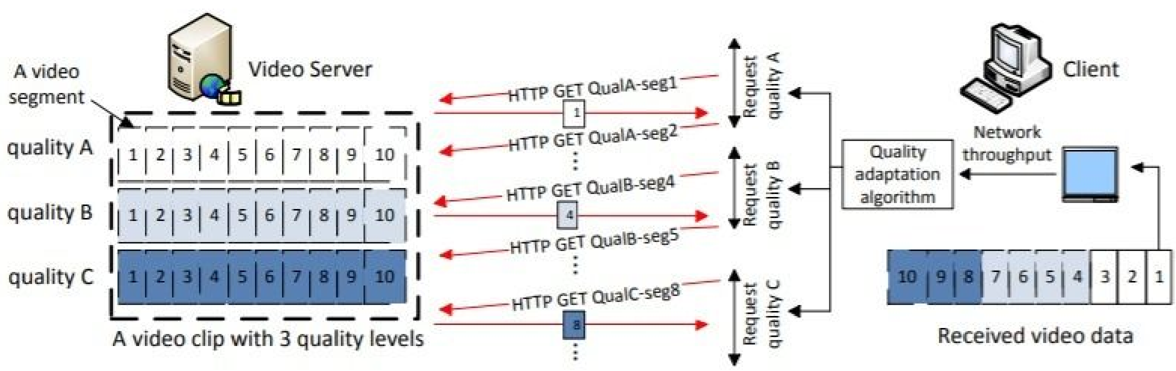

Figure.2 : A typical DASH system

1.5. Physical architecture

The network we are gonna operate on must include a wireless access point to transmit the video over Wi-Fi, offline storage that stores the videos that are going to be streamed, and at least 2 stations (STA) for different connectivity tests at the same time. The connection between the AP and the STA is over Wi-Fi while the storage unit and the AP is it connected using an ethernet cable

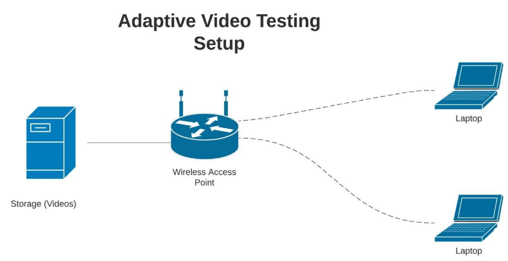

Figure.3 Adaptive Video Testing Setup

2. WORK PLAN

2.1. Work Breakdown Structure (WBS)

1 . Adaptive Video Transfer Protocol

- 1.1 Back-End
  - 1.1.1 Server-client connection
    - 1.1.1.1 Client sending the appropriate GET request
    - 1.1.1.2 Server executing the proper service response
  - 1.1.2 Transfer algorithm
    - 1.1.2.1 Video encoding
    - 1.1.2.2 Video bitrate encoding switch
    - 1.1.2.3 Buffer rate
- 1.2 Front-End
  - 1.2.1 Application dashboard
    - 1.2.1.1 User interface design
    - 1.2.1.2 User interface implementation and coding
    - 1.2.1.3 Back-end connection

2.2 Responsibility Matrix (RM)

Task	Zakaria	Ghaith
Statistics		R
Performance Metrics	S	R
Implementing the Network	R	S
Implementing the algorithm	R	
Planning	S	R
Report Writing	S	R
Integration	R	
	R= Responsible	S= Support

Table 1. Responsibility Matrix for the team.

2.3. Project Network (PN)

-

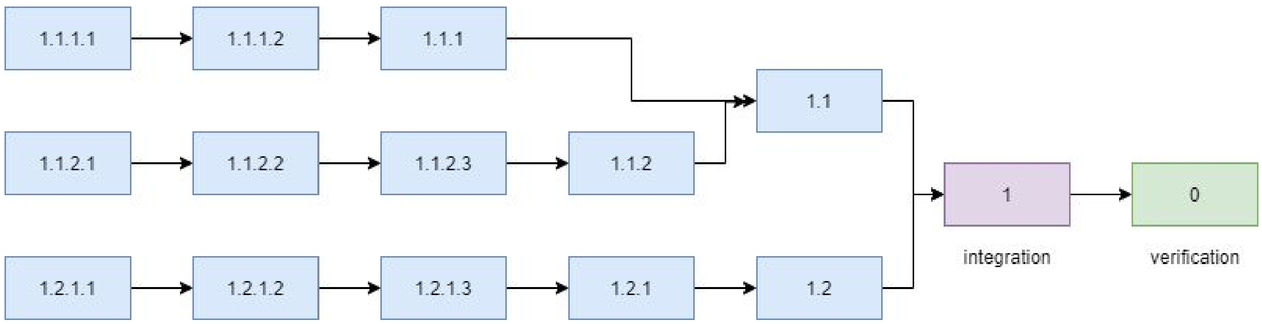

Figure 4. Project network

2.4. Gantt chart

week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Zakaria	1.1.1.1		1.1.1.2	1.1.2.1		1.1.2.2					1.2.1.3				
Ghaith	1.1.1.1			1.1.2.1				1.1.2.3		1.2.1.1	1.2.1.2				
integration												1			
verification													0		
documentation										draft			report		
presentation													draft	report	

Figure 5. Timeline for the project

3. DESIGN PROCESS

3.1. Computer Engineering

3.1.1. Definition of the problem

Designing and Implementing the adaptive video streaming protocol to allow maximise the user experience when streaming on-demand videos.

3.1.2. Review of technologies and methods

-Hyper Text Transfer Protocol (HTTP):

HTTP is an application layer protocol that is used for transferring data over the Internet. It uses a request-response concept as a standard of communication in a client-server connection. In our case the client is the media player and the server is the server holding the content to be streamed. The scenario plays when the client sends an HTTP request to the server (i.e. GET request). The server then has a service which is then executed that performs a function (i.e. prepares the segment to be sent) and sends a response to the client (i.e video segment to be streamed).

-Dynamic Adaptive Streaming Over HTTP (DASH):

DASH or MPEG-DASH is an adaptive streaming protocol that is based on HTTP that is used to stream high quality media over the internet. It divides the video to be streamed into segments which are then sent to the client via HTTP. The information of the segment sent is sent alongside it as a media presentation description (MPD). DASH's architecture is independent since different video formats and bitrates (quality) can be sent, and segments are independent to each other (a segment doesn't rely on the previously sent or to be sent segment). Thus it gives us the ability to select segments based on our own metrics (or logic/algorithm)(such as connectivity and bandwidth).

-Media presentation description (MPD):

MPD is a XML formatted document in the server that contains information about the video segments available. This information includes media data and the relationship between video segments . MPD consists of periods,adaptation sets and representations where:

Periods: are a slot of the media content.

Adaptation Sets: contain different media streams.

Representation: Each representation is a different encoded version of the adaptation set (different bitrate or quality as an example).

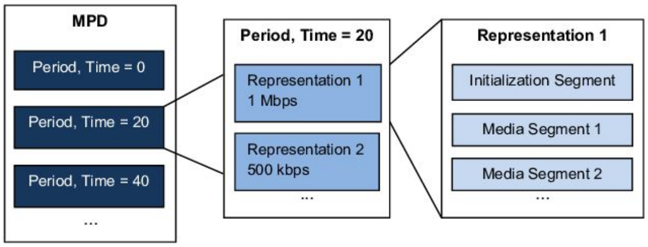

Figure 6. MPD

Each representation contains segments that are sent over to the client upon their request. Upon initiating the connection the server sends the client the MPD, and then the client uses it to request the suitable segment.

3.1.3. Standards and constraints

-MPEG Dash https://mpeg.chiariglione.org/standards/mpeg-dash

-802.11 wifi standard https://tools.ietf.org/html/rfc7494

-HTTP (RFC 7540 (HTTP/2)) https://tools.ietf.org/html/rfc7540

-H265/HEVC codec https://www.itu.int/rec/T-REC-H.265-201911-I/en

-JSON and XML

3.1.4. Conceptualization

There are several ways of measuring the quality of a network. Throughput, bandwidth and packet loss are all QoS metrics as previously mentioned. But the challenge is mapping those QoS metrics to be able to choose the video quality accordingly.

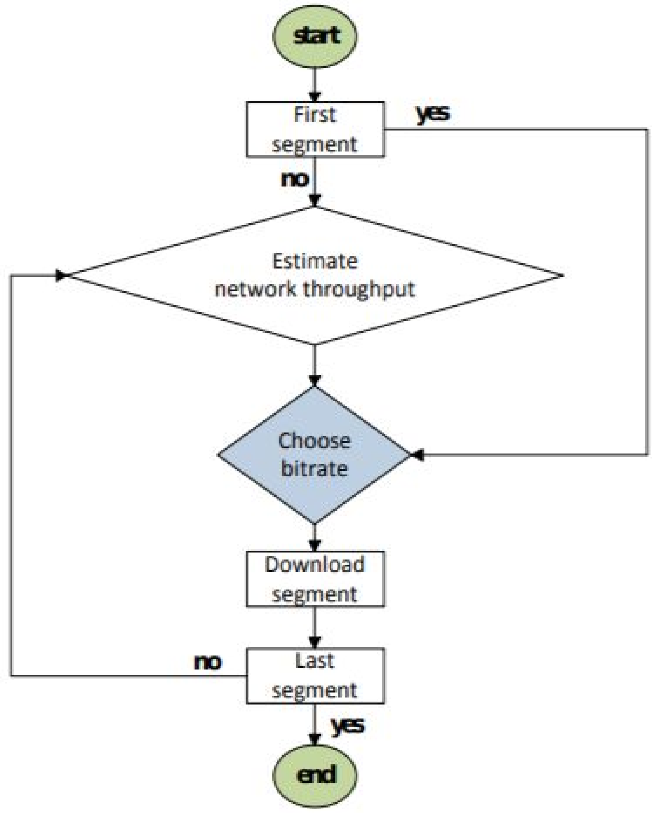

Figure.7 Concept flow chart

-Throughput-based Algorithms:

In our project we are going to use throughput-based adaptive algorithms to determine the proper bitrate to request. Although there are several ways to measure the throughput:

Instant throughput: based on the latest received segment.

Smooth throughput: based on taking into consideration all the downloaded segments instant throughput.

Quantization: based on taking smoothed estimated throughput and map it to the discrete set of the video representations.

Other methods vary upon different types of means and calculations and different timings to calculate the throughput.

Now the algorithm has to obtain 2 main objectives, first being reaching the highest possible bitrate and then maintaining it. We also need to take into consideration that the buffer has to keep a minimum amount of segments cached in it to avoid buffer starvation.

However, a throughput based algorithm that stands out was Probe and adapt (PANDA) algorithm. It guarantees accurate bandwidth measurements, efficiently utilized thanks to probing via incrementing the rate and finally asymmetric shifting (goes up fast but falls down slowly) hence avoids sudden bandwidth drops. [1] [4]

3.1.6. Risk assessment

RISK LEVEL		**Severity** of the event on the project success		
		Minor	Moderate	Major
Probability of the event occuring	Unlikely	VERY LOW	LOW	MEDIUM
	Possible	LOW	MEDIUM	HIGH
	Likely	MEDIUM	HIGH	VERY HIGH

VERY LOW	This event is very low risk and so does not require any plan for mitigation. In the unlikely event that it does occur there will be only a minor effect on the project.
LOW	This event is low-risk; a preliminary study on a plan of action to recover from the event can be performed and noted.
MEDIUM	This event presents a signficant risk; a plan of action to recover from it should be made and resources sourced in advance.
HIGH	This event presents a very signficant risk. Consider changing the product design/project plan to reduce the risk; else a plan of action for recovery should be made and resources sourced in advance.
VERY HIGH	**This is an unacceptable risk. The product design/project plan must be changed to reduce the risk to an acceptable level.**

Table 2. Risk Matrix.

Failure Event	Probability	Severity	Risk level	Plan of Action
Microprocessor faiure	Unlikely This component is known to be reliable.	Moderate Would require replacing or using a simulator.	Low	Have a simulator set up ready to be used instead.
Covid-Lockdown	Likely Limited access to school/shops and some members may not be able to travel.	Minor Most items cam be bought online and the nature of the project allows it to be performed remotely.	Medium	Design the project in a way that allows the workflow to be integrated online and prepare such tools.
Failure of a computer	Unlikely	Major	Medium	Keep back up of the process somewhere on the cloud.

Table 3. Risk assessment table

3.1.7. Materialization

3.2. Software Engineering

3.2.1 Interface Requirements

3.2.1.1 User Interfaces

- The server dashboard app shows the buffer rate while the video is being streamed.
- The client is a basic video player application with a play/ pause button and progress bar.

3.2.1.2 Software Interfaces

The system will use a video player library to play the videos on the client device but the specific library will be chosen in the materialization part.

3.2.2 Functional Requirements

3.2.2.1 Behaviors of the Software Application

Actor Name	Name of Behavior	Description of Behavior
User	*Request video to play*	*Video starts playing dynamically according to the connection status*
User	*Pause/play button press*	*Video stops playing or resumes*
Operator	*Display stat*	*Dashboard application displays the statistics for the video buffer rate*

3.2.2.2 Attributes of the Software Application

Actor Name	Name of Attribute	Description of Attribute
Video	*BufferRate*	*Equals the rate the video is being transferred*

3.2.2.3 Design and Implementation

The project doesn't use any Database system.

3.2.3 Nonfunctional Requirements

3.2.3.1 Performance Requirements

- Response time: the program should be fast and responsive so the user does not get bored using it (0.1 sec - 1.0 sec).

3.2.3.4 Software Quality Attributes

- Learnability : the program controls must be clear so the user feels comfortable with the program from the first use.
- Usability : the program should be as simple as it can so the user does not get overwhelmed with lots and lots of features.
- Availability : the program will always be available as far as the server is running.

3.2.3.5 Business Rules

Dash board operator : can only see the buffer statistics but users can't.

3.2.4 Use-case Modeling

3.2.4.1 Actor Glossary

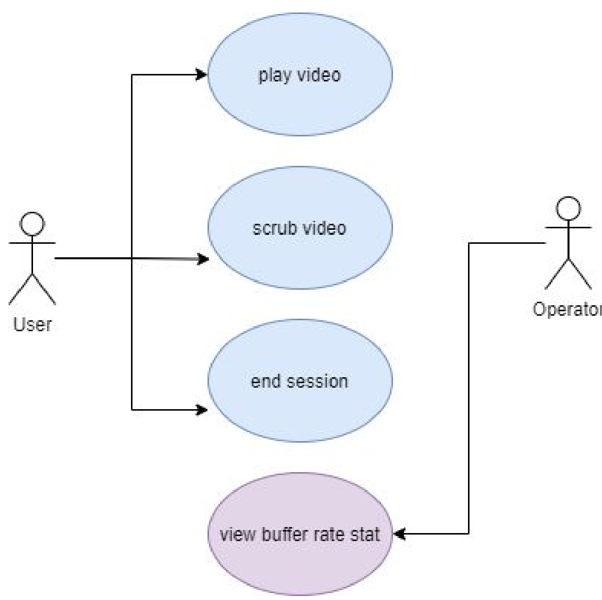

Figure.8 Actor Glossary

3.2.4.2 Use-case Glossary

Use-case Name	Description	Participating Actors
Play video	*User plays the video*	*User*
Video scrub	*User scrubs throw the video*	*User*
End session	*User ends the viewing session*	*User*
Check rate	*Operator can check the video buffer rate statistics*	*Operator*

3.2.4.3 Use-case Scenarios

Use-case Name	*Play video*
Use-case Description	*User plays the video*
Actors	*User*
Pre-Condition	*System must be connected to the server*
Post-Condition	*Buffer rate is displayed on the dashboard app*
Normal Flow	*Step1 : User clicks on desired video to play* *Step2 : client app sends request for the video and it starts playing in the specific bitrate according to the network status*
Alternate Flow	*Alt-step: try again*
Business Rules	*If the connection is lost stop playing the video*

Use-case Name	*Video scrub*
Use-case Description	*User scrubs throw the video*
Actors	*User*

Pre-Condition	*Video must be playing*
Post-Condition	*Video moves in time*
Normal Flow	*Step1 : User clicks on play head* *Step2 : user moves the layer head to the desired time in the video* *Step3 : client send request for the new time in the video*
Alternate Flow	-
Business Rules	-

Use-case Name	*End session*
Use-case Description	*User ends the viewing session*
Actors	*User*
Pre-Condition	*Video viewing must be started already*
Post-Condition	*Video will be closed*
Normal Flow	*Step1 : User clicks closing button* *Step2 : video player exits the video*
Alternate Flow	-
Business Rules	*Video session must be started before ending session*

Use-case Name	*Check rate*
Use-case Description	*Operator can check the video buffer rate statistics*
Actors	*Operator*

Pre-Condition	*System must be connected to the server*
Post-Condition	*Buffer rate is displayed on the dashboard app*
Normal Flow	*Step1 : operator opens dashboard application* *Step2 : dashboard application displays transfer statistics*
Alternate Flow	-
Business Rules	*If the connection is lost stop playing the video*

3.2.4.4 Use-case Diagram

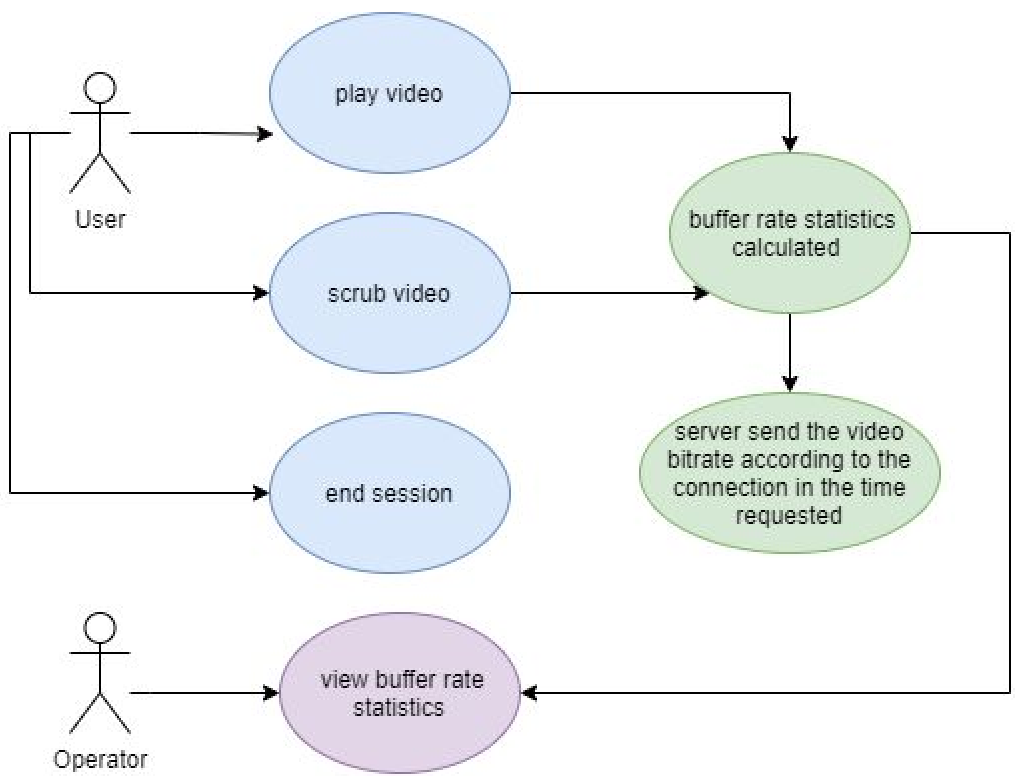

Figure.9 use-case diagram

3.2.5 Data Modeling

3.2.5.1 Activity Diagram

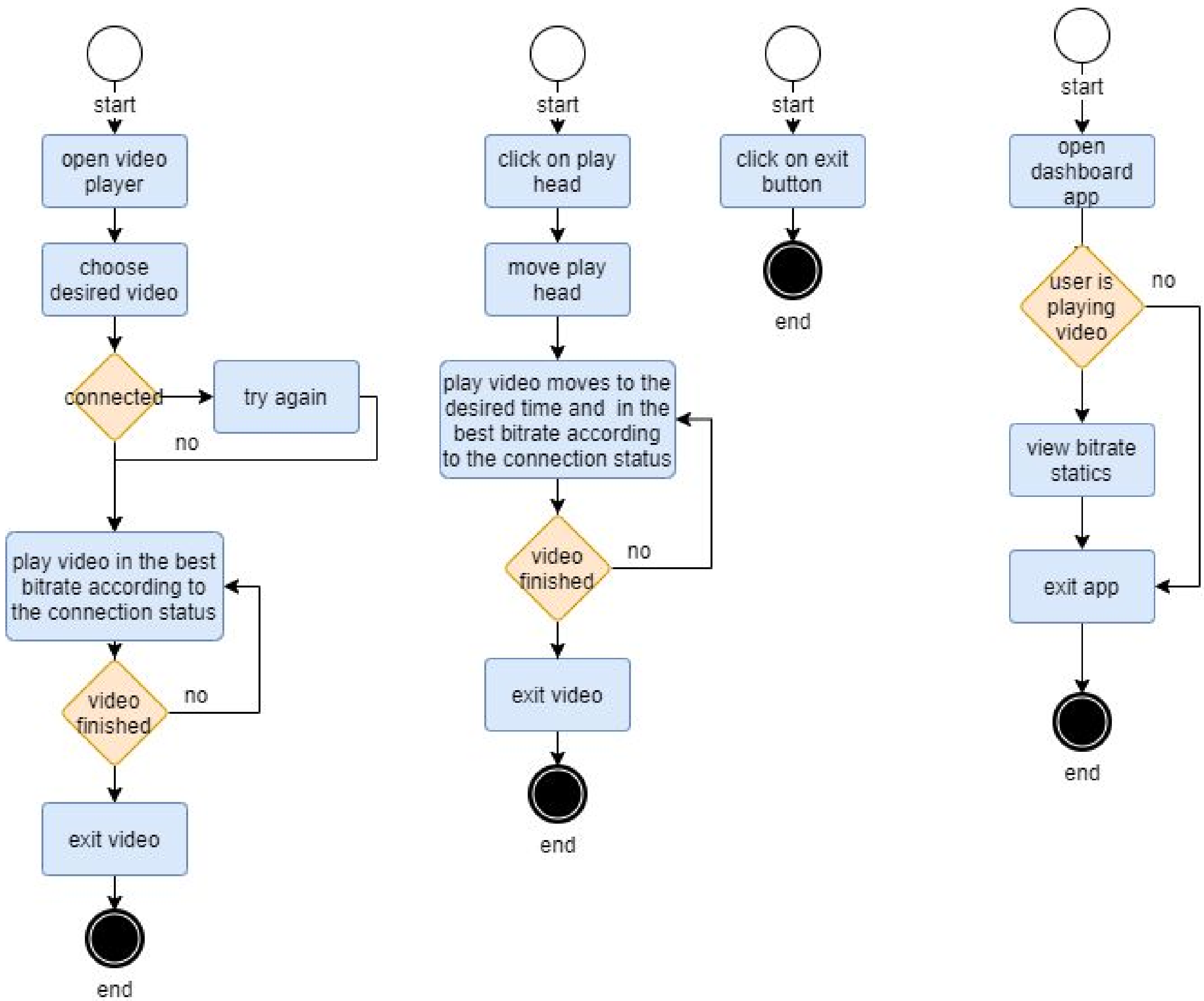

Figure.10 activity diagram

3.2.5.2 Sequence Diagrams

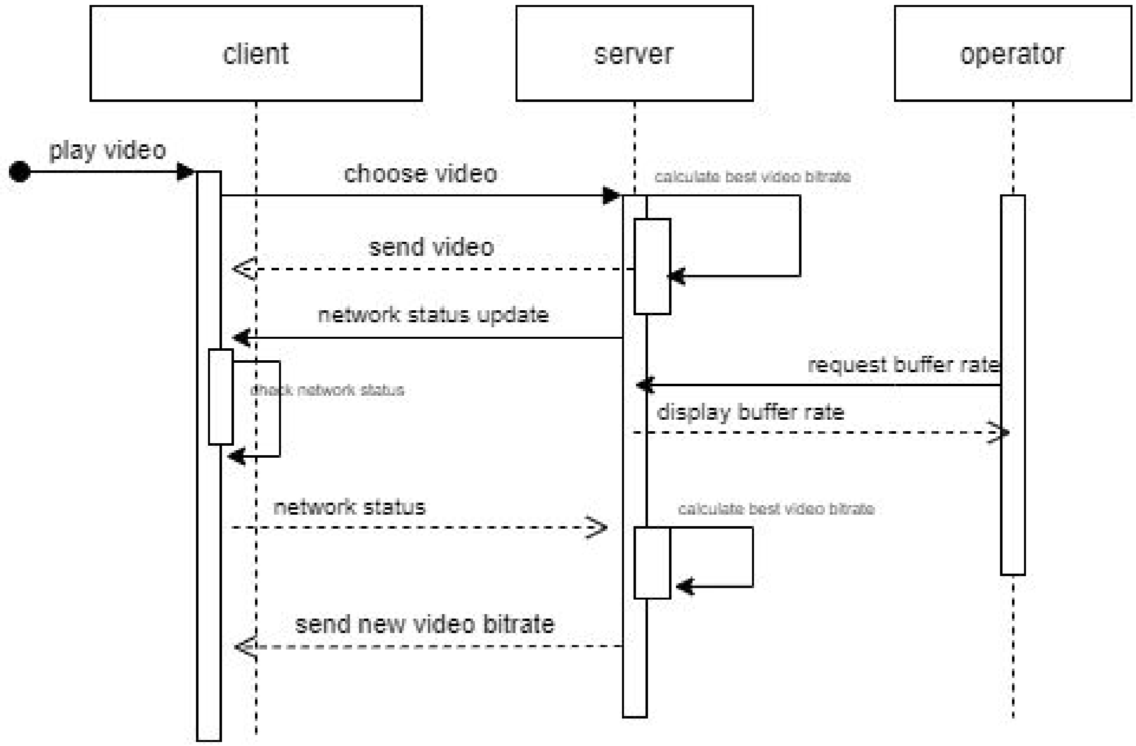

Figure.11 Sequence Diagram

3.2.5.3 Process Diagram

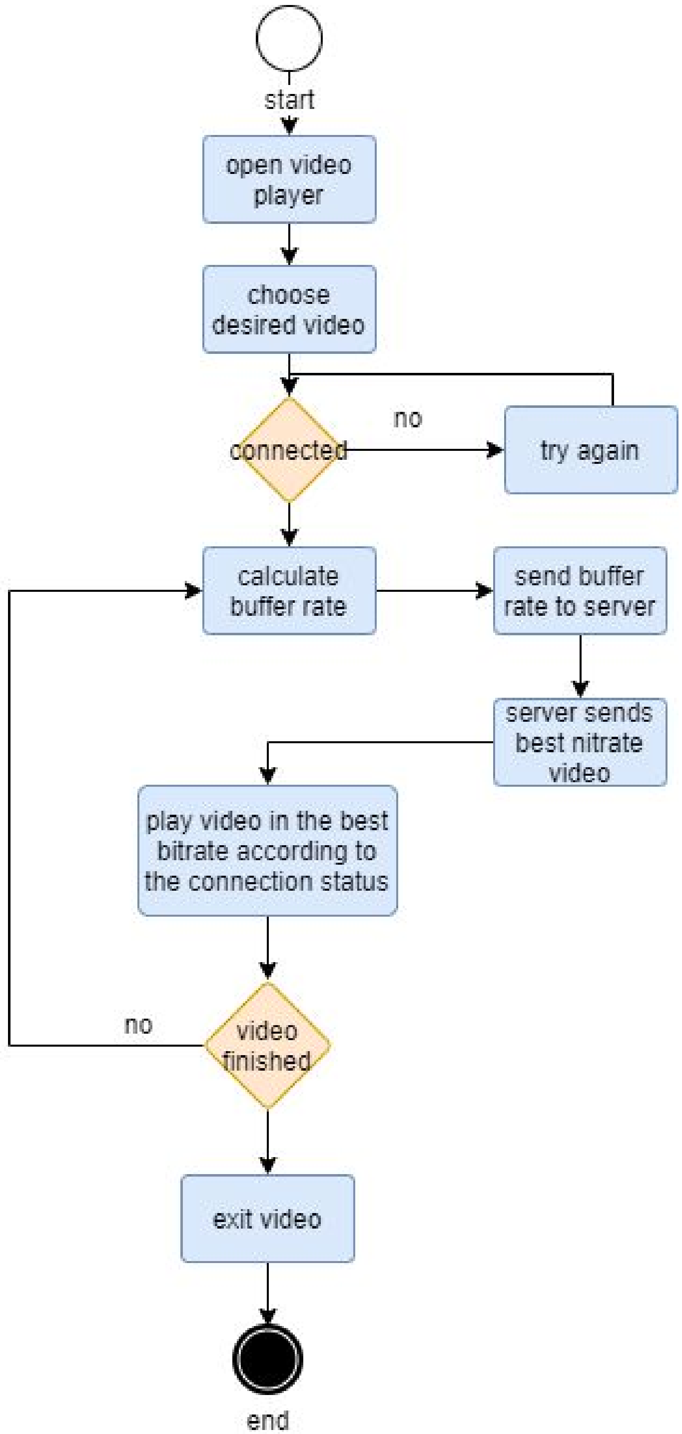

Figure.12 Process Diagram

3.2.5.4 Data Flow Diagram

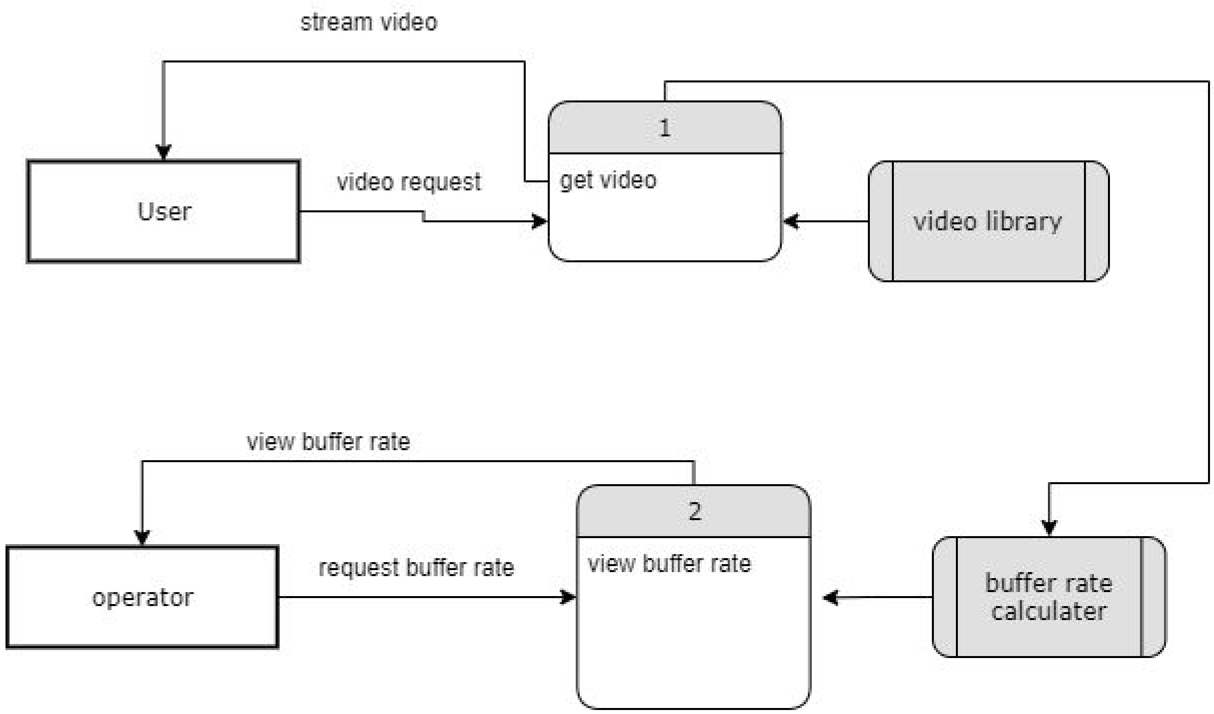

Figure.13 Data Flow Diagram

3.2.5.5 Design/Block Diagram

3.2.5.6 Object Diagram

In the materialization process this part will be more clear .

3.2.5.7 Class Diagram

In the materialization process this part will be more clear .

4. COSTS

Component	Cost (TL)
Raspberry pi 4 GB RAM + power adapter	665
Mini WiFi USB Adapter	30
Ethernet cable	25
TOTAL	**720**

Table 4. Table of costs.

5. CONCLUSION

In this preparation of the project the small team of two students had to work hard to accomplish what a full team can do, but thanks to the perfect chemistry between the team members it become easier, the team now is ready for materialization next semester, the solutions the team choose are the best optimal solutions for the problem in our situation.

ACKNOWLEDGEMENT

We wish to thank our advisers Assist Prof. Ece Gelal SOYAK and Prof. Mehmet Alper TUNGA for their continuous feedback and supervision of our process from the start all the way to the end. And for all the guidance and support they have provided throughout the semester. .

REFERENCES

1. Zakaria Ye. Performance Analysis of HTTP Adaptive Video Streaming Services in Mobile Networks. Networking and Internet Architecture [cs.NI]. Université d'Avignon, 2017. English. ffNNT:2017AVIG0219ff. fftel-01715308f

2. https://ieeexplore.ieee.org/abstract/document/990868?casa_token=u78u2_RmTcYAAAAA:9y4WtfwpE7IhXBRAN28t3qRtPSytdDAzuo45Ha6uURZFbjLOZPpE7zpMhwTHTJcR5I4onNtrvcmmEw

3. Evaluating QoS and QoE Dimensions in Adaptive Video Streaming Stensen, Julianne M. G. Link : https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/262683

4. Z. Li et al., "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," in IEEE Journal on Selected Areas in Communications, vol. 32, no. 4, pp. 719-733, April 2014, doi: 10.1109/JSAC.2014.140405.