

Video Streaming over Wireless Networks using Efficient Real Time Adaptive Algorithm

Ms. Bhagyashri A. Banarase¹, Prof. N. R. Borkar²

M.E. Student, Dept of CSE, Dr. Sau K.G.I.E.T., Darapur, Sant Gadge Baba Amravati University, Maharashtra, India¹

Asst Professor, Dept of CSE, Dr. Sau K.G.I.E.T., Darapur, Sant Gadge Baba Amravati University, Maharashtra, India²

Abstract: In this paper a real time adaptive algorithm is used for video streaming. A large number of live segmented adaptive HTTP video streaming services exist in the internet today. Managing the Quality-of-Experience (QoE) of video streaming for wireless clients is becoming increasingly important due to the rapid growth of video traffic on wireless networks. How efficiently and cost-effectively utilize multiple links to improve video streaming quality needs investigation. In order to maintain high video streaming quality while reducing the wireless service cost, in this paper, the optimal video streaming process with multiple links is formulated as a Markov Decision Process (MDP). The reward function is designed to consider the quality of service (QoS) requirements for video traffic, such as the start-up latency, playback fluency, average playback quality, playback smoothness and wireless service cost. To solve the MDP in real time, we propose an adaptive, best-action search algorithm to obtain a sub-optimal solution.

Keywords: DASH, Markov decision process, video streaming, wireless networks, Adaptive Video Streaming, HTTP protocol.

I. INTRODUCTION

The increasing demand for video streaming has meant video constitutes a large portion of the total data traffic on the Internet. Video streaming means that divides the entire videos into number of segment and then transmitted the segment into client. While transmitting the videos in server, client side can automatically create the buffer for storing the divide segment. If one buffer is full, video can start to play and automatically create another buffer for sorting remaining segment.

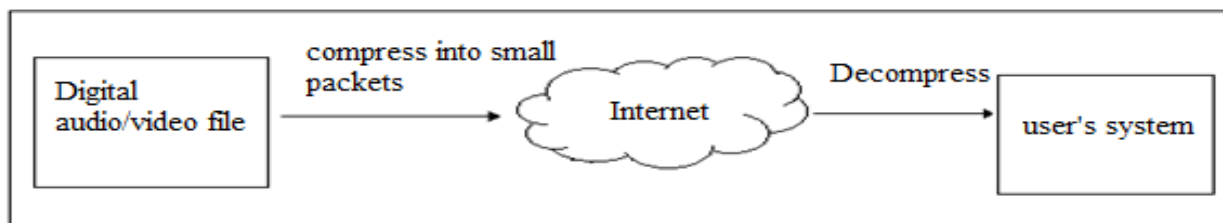


Fig. 1. Video Streaming Works

Fig. 1. show how video streaming works. The digital audio/video file can compress and breaking it into small packets, which are sent, one after another, over the network. When the packets reach their destination, they are decompressed and reassembled into a form that can be played by the user's system. To maintain the illusion of seamless play, the packets are "buffered" so a number of them are downloaded to the user's machine before playback. In a perfect world, streaming video works by downloading the initial portion of the file, which is named the buffer, into the user's player [2]. Video streaming is one way to deliver video over the Internet. Video streaming enables simultaneous delivery and playback of the video, which over comes the problems associated with file download since users do not have to wait for the entire video to be received before viewing it. The basic idea of video streaming is to partition the compressed video source files into parts, transmit them in succession, and decode and playback the video in the receiver.

Hence, users are able to watch the videos just after a small delay at the beginning. Also, the storage requirements of the receiver is comparatively low, because only a small portion of the video is stored in the users' buffer at any point in time. Generally, there are two types of streaming scenarios based on whether the video is pre-encoded and stored for later viewing, or it is captured and encoded for real-time communication. Video conference, video phone and interactive games are examples of real-time video streaming applications, which have stringent delay requirement. On the other hand, currently in many applications video content is pre-encoded and stored in the multimedia server for later request of viewing, which is also called video-on-demand (VoD) [13].



II. LITERATURE SURVEY

Video streaming over wireless networks becomes increasingly important in a variety of applications. To accommodate the dynamic change of wireless networks, Quality of Service (QoS) scalable video streams need to be provided. It is beyond the scope of this work to relate our research to all video streaming research efforts that have been made.

A. Video Delivery via File Download

Probably the most straightforward approach for video delivery of the Internet is by something similar to a file download, but we refer to it as video download to keep in mind that it is a video and not a generic file. Specifically, video download is similar to a file download, but it is a large file. This approach allows the use of established delivery mechanisms, for example TCP as the transport layer or FTP or HTTP at the higher layers. However, it has a number of disadvantages. Since videos generally correspond to very large files, the download approach usually requires long download times and large storage spaces. These are important practical constraints. In addition, the entire video must be downloaded before viewing can begin. This requires patience on the viewers part and also reduces flexibility in certain circumstances, e.g. if the viewer is unsure of whether he/she wants to view the video, he/she must still download the entire video before viewing it and making a decision [5]. In the download mode, the user can play the downloaded file only after the whole file has been downloaded from a server to his/her computer. The full file transfer, in the download mode, can often suffer unacceptably long transfer times, which depend on the size of the media file and the bandwidth of the transport channel. For example, if downloaded from <http://www.mp3.com>, an MP3 audio file encoded at 128 kbit/s and of 5 min duration will occupy 4.8 MB of the user's hard disk. Using a 28.8k modem, it will take about 40 minutes to download the whole file.

B. Video Delivery via Streaming

Video delivery by video streaming attempts to overcome the problems associated with file download, and also provides a significant amount of additional capabilities. The basic idea of video streaming is to split the video into parts, transmit these parts in succession, and enable the receiver to decode and playback the video as these parts are received, without having to wait for the entire video to be delivered. Video streaming can conceptually be thought to consist of the follow steps:

- 1) Partition the compressed video into packets
- 2) Start delivery of these packets
- 3) Begin decoding and playback at the receiver while the video is still being delivered

Video streaming enables simultaneous delivery and playback of the video. This is in contrast to file download where the entire video must be delivered before playback can begin. In video streaming there usually is a short delay (usually on the order of 5-15 seconds) between the start of delivery and the beginning of playback at the client. This delay, referred to as the pre-roll delay, provides a number of benefits which are discussed in Section 6. Video streaming provides a number of benefits including low delay before viewing starts and low storage requirements since only a small portion of the video is stored at the client at any point in time. The length of the delay is given by the time duration of the pre-roll buffer, and the required storage is approximately given by the amount of data in the pre-roll buffer [5].

C. Video Streaming - Related Work

The author T. Stock hammer [7], introduced some design principles for DASH. It has been a hot topic in recent years. There are many commercial products which have implemented DASH in different ways, such as Apple HTTP Live Streaming and Microsoft Smooth Streaming. Since the clients may have different available bandwidth and display size, each video will be encoded several times with different quality, bit rate and resolution. All the encoded videos will be chopped into small segments and stored on the server, which can be a typical web server. HTTP-based progressive download does have significant market adoption. Therefore, HTTP-based streaming should be as closely aligned to HTTP-based progressive download as possible. The media preparation process typically generates segments that contain different encoded versions of one or several of the media components of the media content. The segments are then hosted on one or several media origin servers typically, along with the media presentation description (MPD). The media origin server is preferably an HTTP server such that any communication with the server is HTTP-based Based on this MPD metadata information that describes the relation of the segments and how they form a media presentation; clients request the segments using HTTP GET or partial GET methods. The client fully controls the streaming session, i.e., it manages the on-time request and smooth play out of the sequence of segments, potentially adjusting bitrates or other attributes, for example to react to changes of the device state or the user preferences. Massively scalable media distribution requires the availability of server farms to handle the connections to all individual clients. HTTP-based Content Distribution Networks (CDNs) have successfully been used to serve Web pages, offloading origin servers and



reducing download latency. Such systems generally consist of a distributed set of caching Web proxies and a set of request redirectors. Given the scale, coverage, and reliability of HTTP based CDN systems, it is appealing to use them as base to launch streaming services that build on this existing infrastructure. This can reduce capital and operational expenses, and reduces or eliminates decisions about resource provisioning on the nodes. Scalability, reliability, and proximity to the user's location and high-availability are provided by general purpose servers.

The authors R. Mok, X. Luo, E. Chan, and R. Chang [8], proposed an enhancement to Dynamic Adaptation Streaming over HTTP (DASH) by the Quality of Experience (QoE) for users by automatically switching quality levels according to network conditions. Various adaptation schemes have been proposed to select the most suitable quality level during video playback. Adaptation schemes are currently based on the measured TCP throughput received by the video player. Although video buffer can mitigate throughput fluctuations, it does not take into account the effect of the transition of quality levels on the QoE. This paper propose a QoE-aware DASH system (or QDASH) to improve the user-perceived quality of video watching. It integrates available bandwidth measurement into the video data probes with measurement proxy architecture and found that the available bandwidth measurement method facilitates the selection of video quality levels. Moreover, it can assess the QoE of the quality transitions by carrying out subjective experiments. The results show that users prefer a gradual quality change between the best and worst quality levels, instead of an abrupt switching. Hence, the papers propose a QoE-aware quality adaptation algorithm for DASH based findings. Finally, it integrates both network measurement and the QoE-aware quality adaptation into a comprehensive DASH system. QDASH consists of two building blocks 1) QDASH-abw 2) QDASH-qoe. QDASH-abw measures the network available bandwidth, and QDASH-qoe determines the video quality levels. These two modules can be integrated into existing DASH systems, while the modifications to the systems are kept to minimum. QDASH is designed for streaming H.264/AVC video clips, and aims at immediate deployment to current systems. As part of the investigation for optimal streaming strategy for DASH the paper named "Adaptive scalable video streaming in wireless networks,"[3] introduced a rate adaptation algorithm for video streaming in wireless network. Dynamic Adaptive Streaming over HTTP (DASH) which extends the traditional HTTP streaming with an adaptive component addressing the issue of varying bandwidth conditions that users are facing in networks based on the Internet Protocol (IP).

Paper [16] Shin-Hung Chang, Jan-Ming Ho, and Yen-Jen Oyang , concentrates on the negative effects introduced when multiple clients are competing for a bottleneck and how proxies are influencing this bandwidth competition. The clients request individual portions of the content based on the available bandwidth which is calculated using throughput estimations. A consequence of this requesting scheme is that only some parts of the content are stored on proxy servers, which are intercepting the connection between the client and the content server. This uncontrolled distribution of the content influences the adaptation process that assumes that the measured throughput is the throughput to the content server. The impact of this falsified throughput estimation could be tremendous and leads to a wrong adaptation decision which may impact the Quality of Experience (QoE) at the client. Our first and probably simplest approach to decrease the frequent switching and as a consequence the negative effects, that could be caused due to that switching, is an adaptation logic with an exponential back off. This approach decreases the number of switch up points if a switch down occurs. But this technique does not consider whether a bandwidth fluctuation is self-caused or network caused.

III. PROBLEM DEFINITION

Presently, progressive download was the most widely deployed streaming method which buffers large amount of video data. The computer systems have a limited buffer size and if the watcher decided to stop watching the video, the buffered videos will be wasted and also it can't support multiple links. To overcome these disadvantages of progressive download the existing system called DASH (Dynamic Adaptive Streaming over HTTP) with multiple links has been proposed. In a DASH system, multiple copies of pre-compressed videos with different resolution and quality are stored in segments. The rate adaptation decision is made at the client side. For each segment, the client can request the appropriate quality version based on its screen resolution, current available bandwidth, and buffer occupancy status. This pull-based DASH scheme can be extended to support multiple links, i.e., we can let the client request different parts of one segment over different links. Traditional streaming generally uses a stateful protocol, e.g., the Real-Time Streaming Protocol (RTSP): Once a client connects to the streaming server the server keeps track of the client's state until the client disconnects again. Typically, frequent communication between the client and the server happens. Once a session between the client and the server has been established, the server sends the media as a continuous stream of packets over either UDP or TCP transport. In contrast, HTTP is stateless. If an HTTP client requests some data, the server responds by sending the data and the transaction is terminated. Each HTTP request is handled as a completely standalone one-time transaction. According to the DASH system it will take an amount of time for waiting each request-replay communication.

The main aim of the project is to avoid delay between the playback and the start up one. Using UDP as protocol for streaming will no need excess amount of time for communication as in HTTP. Currently all hand held devices are



equipped with multiple wireless links like 3G, WiFi, Bluetooth etc but all will use a single link for video streaming. High resilience to bandwidth variation and easy deployment are both important requirements for video streaming applications. While there occurs any variation in the bandwidth, the downloading will be interrupted and will affect the playback smoothness. Using multiple links can improve video streaming in several aspects: the aggregated higher bandwidth can support video of higher bit rate and when one wireless link suffers poor link quality or congestion, the others can compensate for it [6].

In recent years much work has been done in the area of adaptive video streaming. The key to adaptive video streaming is the design of its adaptation algorithm, i.e., the algorithm to determine when and what video bit-rate to use for requesting new video segments. Each video will be encoded several times with different quality, bit rate and resolution. All the encoded videos will be chopped into small segments and stored on the server, which can be a typical web server [1]. These small segments will be downloaded to the browsers' cache and played by the client (browser). The video rate adaptation is performed at the client side, which is also called the pull-based approach. The client will determine the quality version of the requested video segment according to its current available bandwidth, resolution and the number of buffered unwatched segments. After the current segment is completely downloaded, the rate adaptation algorithm will be invoked again for the next segment.

IV.METHODLOGY

We consider how to utilize wireless access networks for video streaming, e.g., using a combination of WiFi, and request over 3G simultaneously. Here, as an example, 3G network and WiFi access networks are considered as we do not have end-to-end control over cellular links and our work can be extended when other types of wireless access networks or more than two wireless access networks are used. Since a wireless channel may suffer from time-varying fading, shadowing, interference and congestion, the available bandwidth of a wireless link may vary all the time. In addition, different systems may have different resolution & size. Using these aspects into consideration, the server should store several copies of video with different quality. The videos are encoded with SVC into a base layer and several enhancement layers and chopped into segments and each segment can be played with a fixed duration [20].

A. System architecture

We consider how to utilize wireless access networks for video streaming, e.g., using a combination of WiFi, and request over 3G simultaneously. Here, as an example, 3G network and WiFi access networks are considered as we do not have end-to-end control over cellular links and our work can be extended when other types of wireless access networks or more than two wireless access networks are used. Since a wireless channel may suffer from time-varying fading, shadowing, interference and congestion, the available bandwidth of a wireless link may vary all the time. In addition, different systems may have different resolution & size. Using these aspects into consideration, the server should store several copies of video with different quality. The videos are encoded with SVC into a base layer and several enhancement layers and chopped into segments and each segment can be played with a fixed duration [20].

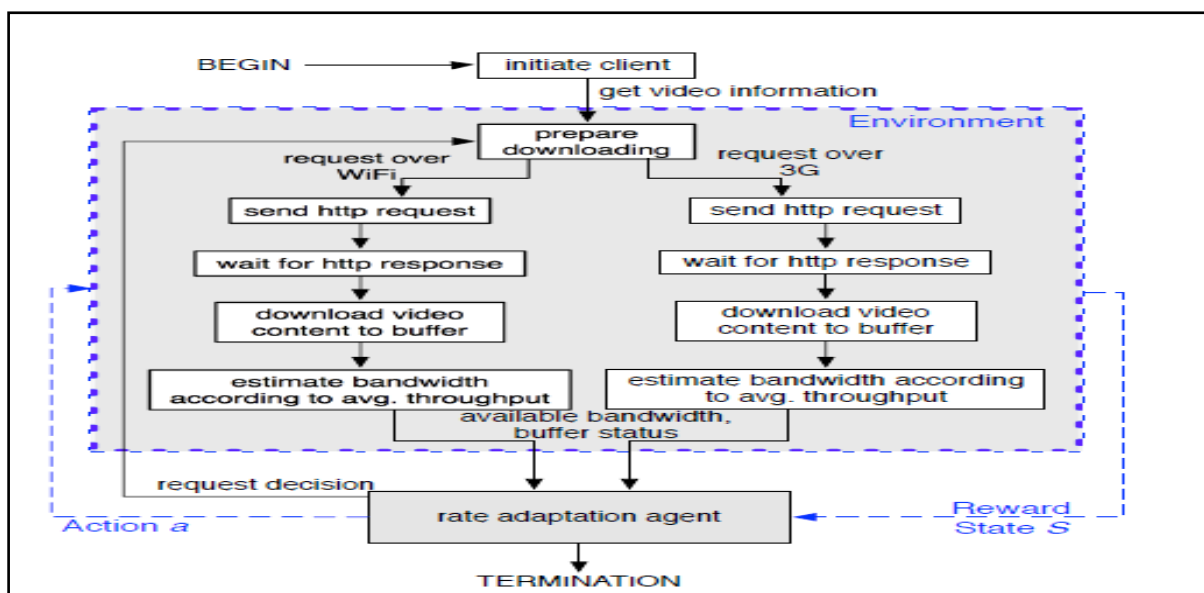


Fig. 2. System Architecture



We design a pull-based algorithm for video streaming, as shown in Figure.2. After initialization, the client will request the video information which includes video resolutions, bitrates and qualities from the server through both the WiFi and Bluetooth links. The rate adaptation agent will request a video segment of appropriate quality version based on the current queue length and estimated available bandwidth. Once the request decision is made, HTTP requests over both WiFi and Bluetooth will be issued to download the video segment. This process will continue until the completion of downloading the last segment or the termination of the video streaming by the user [1].

B. Real time search algorithm

This project proposed a Real time Adaptive algorithm to adapt the bit rate based on the band width condition of network for high quality with less buffering video streaming delivery [2]. A real-time recursive best-action search algorithm, is use to obtain a sub-optimal solution for the future steps consideration to avoid long computation time which is shown in Algorithm 1[1]. To meet the requirement of the real-time search, an important thing is to reduce the search duration for each state to an acceptable value. So for this we used small search depth D to invoke the search algorithm. For the current state s , all the possible actions $A(s)$ will be enumerated. The recursive reward search algorithm is invoked to obtain the reward of state s with action a by enumerating all the possible future states S' and their associated actions $A'(s)$ [1]. Real time adaptive algorithm is more stable and adaptable in dynamic situations, emphasizing the benefit of resource aggregation in multipath network scenarios [12].

The project is basically work on video streaming process for downloading the video from the server this videos are stored on server with several format with different resolutions and this video is in the form of segments or called as frames. In wireless streaming applications, the characteristics of the source, the quality of the channel, and the occupancy of the playback buffer play major roles in the target video quality. Automatic repeat request (ARQ) and forward error correction (FEC) are commonly used to improve the reliability of the wireless link [15]. To reduce the 3G traffic cost while maintain high video streaming quality, we have carefully devised the reward functions. By adjusting the parameters of the reward functions, we can make a trade-off between QoE metrics and the total cost. Through dynamic programming, we have obtained the optimal policy under the reward function settings [17].

Algorithm 1 Real-Time Best-Action Search Algorithm

```

1: Start procedure GETBESTACTION(s)
2: Initialize action ← -1, Qmax ← -∞
3: Generate all possible actions A(s) for state s
4: for all Action a ∈ A(s) do
    q ← REWARDSEARCH(s, a, 0)
    Check if q > Qmax then
        Qmax ← q, action ← a
    end if and for loop.
5: return action
6: end procedure
7: Start procedure REWARDSEARCH(s, a, d)
8: q ← reward of (s, a)
9: if d ≥ D then
    Return q and end if statement.
10: Generate all possible next states S' of (s, a)
11: for all s' from S' do
12: Qmax ← -∞
13: Generate all possible actions A'(s) for state s'
14: for all Action a' ∈ A'(s) do
    Qt ← REWARDSEARCH(s', a', d+1)
    Check if Qt > Qmax then Qmax ← Qt
    end if
    end for
15: q ← q + γ Pss' Qmax
16: end for
17: return q
18: end procedure

```

C. Adaptive Search Depth

The search depth can determine how good the search result is, and a larger value of depth will achieve a better result. Meanwhile, with the increment of the search depth, the search time to obtain the action for a segment will be increased

exponentially. Therefore, the search depth can be viewed as a trade-off between the video quality and the search time [1].

V. SYSTEM IMPLEMENTATION

In this section, we will see how we implement our system how it works on system .Microsoft visual studio 2010 is used for implementation of system. It integrates computation, visualization and programming in easy to use environment. In our implementation we have use real time adaptive search algorithm for video streaming.

A. Login Section

To upload or download the video from server user has to entities in our system .firstly user have to login with their credentials at the time he have the input login mane that is to be create on the time or registration of user . This module includes the user name, address of user, gender, email id of user, the profile photo of user, the user has to enter the user id which user want for the login of system and has to set the password for the login id creation. This phase is called as login id creation for every new user this registration is compulsory and user which already registered he /she has to only give to input of their id and password for login. The link forget password is provided for the user if user forge the password then he /she is able to change the password as display in fig. 3.

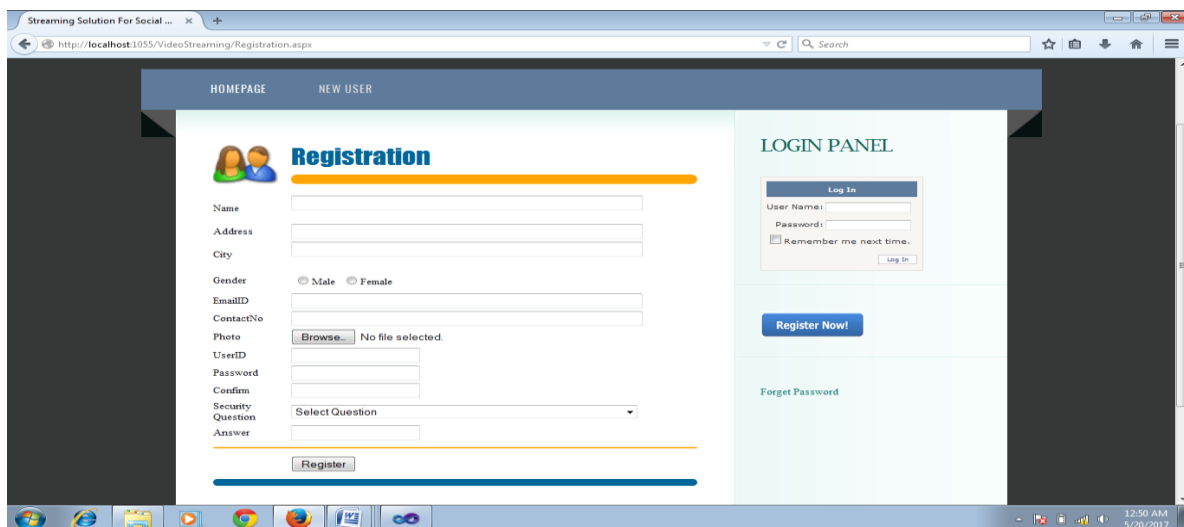


Fig. 3. Login Section

B. Video Downloading

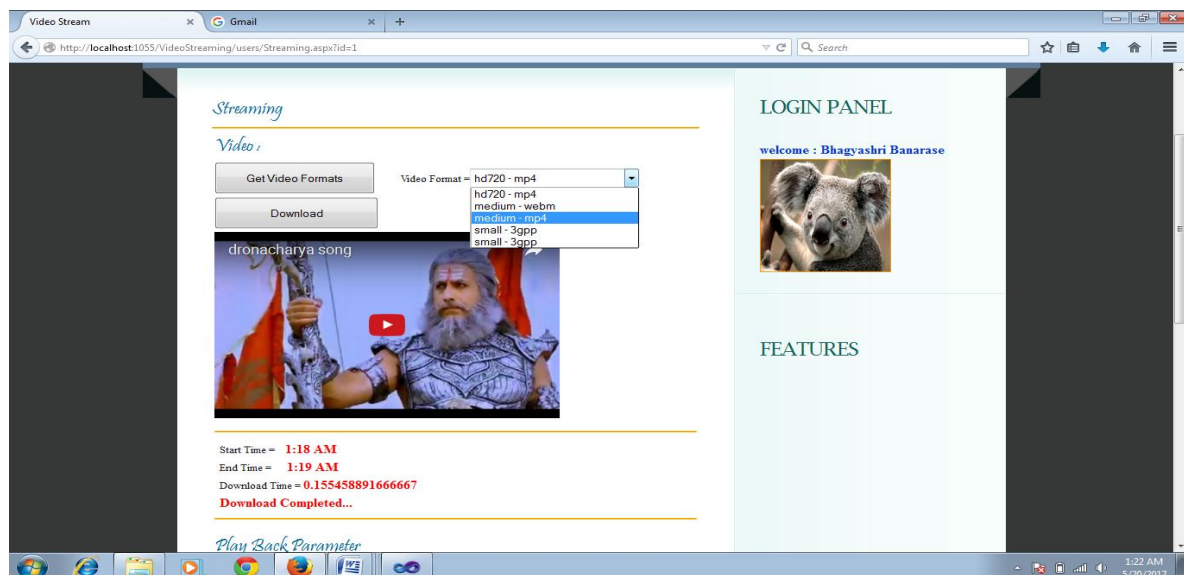


Fig. 4. Video Downloading

From this link the user can download the video by selecting the video, after selecting video the user has to select the video format which they want and then starts downloading the video, when complete video is download the complete information about selected video is generate by the system like the start time, end time, complete download time and for the same video information about play back data is also display like aspect ratio, frame rate for streaming, size of video, duration, height, width this parameter give the complete information about downloaded video as display in fig.4. Simultaneously the video streaming bit rate is also display with graph format. As the X-axis is for time in seconds and Y-axis is for Bits of video (bits per second) see fig.5

For example the user select the video name as Dhronacharya song from available vide list the size of video is (5,852,025 bytes) means 5.58 MB video, for downloading this video from server for complete streaming the algorithm takes the time of 90 seconds and the other parameters related to video is also display like video height, width, duration of video, bit rate, aspect ratio and frame rate. Simultaneously the graph of streaming video is also display with bits per seconds.

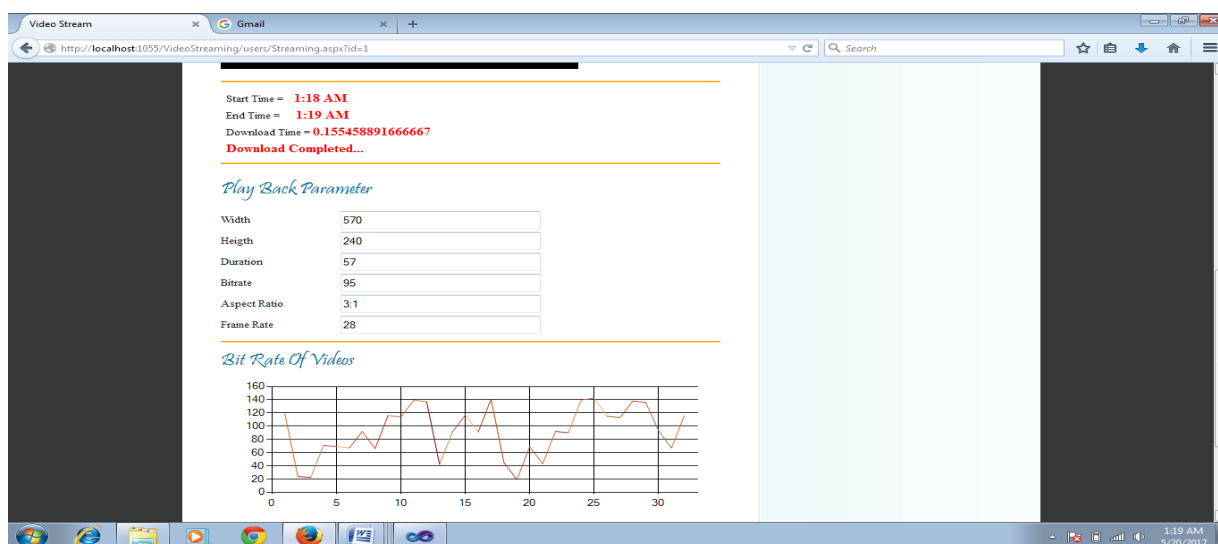


Fig. 5. Video Streaming Graph

VI. EXPERIMENTAL RESULT AND ANALYSIS

In this chapter we have seen how our proposed system will work with the help graph in execution details and also find out the time required for video streaming.

Using the application we have important module in our system i.e. download video which is stored on server .while downloading the video the streaming process is start and streaming graph will be generated and also show the time taken by video for downloading. Download scenario are totally related with video downloading .when user select the video for downloading the streaming process is getting started as the video is divided in to the various segments and this segments are transfer to the user side from server. The speed of download is depending on the network quality and size of video. For video streaming the algorithm used is real time best action search algorithm. For video streaming the user select the video from available video list and select the video format for download the video, after completing the download process the total time taken for the video downloading is display.

Table. 1. Time calculation for downloading video

Sr. No	File name	Size (Bits)	Time (second)	Video format	Aspect Ratio	Frame Rate
1	Dhronacharya Songs	5.58MB	80	medium-MP4	3:2	31
2	Benefits of Mutual Fund Hindi	2.25 MB	59	medium-webm	3:1	28
3	Baby video	1.05 MB	25	small-3gpp	3:2	31
4	Funny girl	1.84 MB	32	medium-MP4	3:2	29

Table.1.shows the total time required for video downloading for the video on the basis of timing constraints the following graph will be generated. In this graph X- axis is for time and Y - axis is for size of video the ratio is defined



in fig.6. The graph shows the streaming time of defined size videos which can clearly mention the total time required to download the video from server using Real time adaptive algorithm.

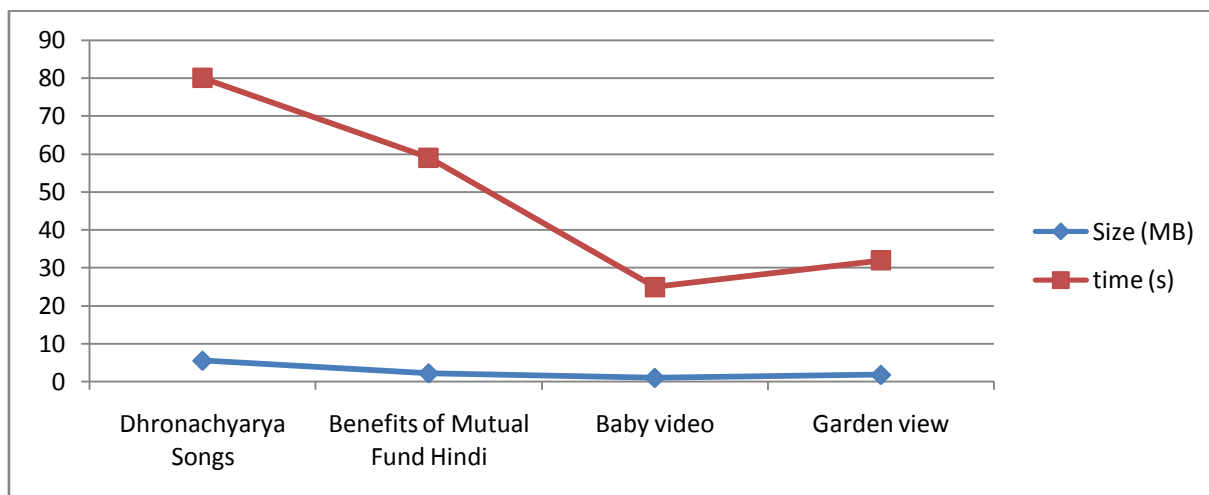


Fig. 6 Video Downloading

VII. APPLICATIONS

Video streaming over wireless networks becomes increasingly important in a variety of applications. For example, commercial applications such as e-learning, video conferencing, stored-video streaming; and military applications such as video surveillance of targeted field or specific objects.

A. Video Conferencing

Live Video Streaming is coming as a great future for corporate world as it provides facility for two-way video web conferencing facility. One application of this real time interactive audio/video is video conferencing. Using a network camera and a head set people can interact as if they are talking face to face in a room. Nowadays if you want to hold a meeting you have to go somewhere to attend the meeting and possibly people from different parts of the country will come to attend the meeting.

But nowadays busy executives don't have time to travel so in such a case one can do videoconferencing, it can be done for holding meetings, it can be done for holding interviews etc. Two types of video conferencing can be done. One is point to point conferencing which is basically a communication link between any two locations; another is multipoint conferencing which is the link between varieties of locations. So you will require not only point-to-point which requires unicast communication but also multicast where multiple users can talk to each other. Similarly video conferencing can be done simultaneously at different places

B. E- Learning

The field of e-learning has been a precursor in using the video streaming over the Internet. Both the synchronous and the asynchronous options have been explored over the last decade, with the asynchronous one becoming the dominant paradigm in recent years.

C. Online Business

Adding video streams to your online business is one of the best things an entrepreneur can do to recover and develop their website. For instance, an electronic gadget shopping website with video clips of how to use electronic items is much more interesting and engaging than a simple website that only have relevant information. The videos add great value to the shopping website and assist viewers get a better understanding and gratefulness of the item they want to buy. Any business niche can apply this method to boost their website visibility. And the buyer gains a superior knowledge of the marketed product or service. Adding streaming video provides your online contact a direct benefit over your strongest competitors, and this is the key reason why you should consider video streaming services for your business websites.

D. Live Video Streaming

The concept of live video streaming is used in News updates for news websites, it can deliver live content such as a football match, a concert or a political speech

VIII. CONCLUSION

In this paper, a real-time adaptive best-action search algorithm for video streaming is used over wireless access networks for video streaming (downloading). To meet the requirement of the real-time search, an important issue is to reduce the search duration for each state to an acceptable value. First, we formulated the video streaming process as an MDP. To achieve smooth video streaming with high quality, the reward function is designed. Second, with the proposed rate adaptation algorithm, we can solve the MDP to obtain a sub-optimal solution in real time. Last, we explained the proposed algorithm. There are still some issues to investigate in the future. Like how to better allocate the loads between several links with finer granularity should be investigated.

IX. FUTURE SCOPE

As with every research work, there is room for improvement in this thesis. Many topics within the scope of the video streaming research area still remain open for further discussion and research. With respect to the next generation video streaming system, we envision that there will be a close integration with emerging search, recommender and social networking systems. This will raise new functional requirements to the streaming services, e.g., allowing users to select arbitrary clips from different video sequences, combining these with user generated media and playing the whole clip as one personalized video. Users must therefore have an upload interface, the video content must be carefully coded and segmented, and efficient mechanisms to analyze, annotate and index the material are necessary. Additionally, these systems must deal with a lot of heterogeneity, because the users will access video content over various networks, and the streamed content may be viewed on different displays ranging from small, mobile hand-held's to high-definition home theaters. In this part of the talk, we also give a presentation of DAVVI. This is a prototype of our next generation multimedia entertainment platform providing a personalized, topic-based user experience blurring the distinction between content producers and consumers. It delivers multi-quality video content in a torrent-similar way over the Hypertext Transfer Protocol. Thus, small segments of each video are retrieved from possibly different locations where the quality of the video is adapted to the available resources. Furthermore, through applied search, personalization and recommendation technologies, end-users can efficiently search and retrieve highlights and combine arbitrary events in a customized manner using drag and drop creating playlists of video segments, i.e., providing a brand new, personalized user experience.

REFERENCES

- [1] Min Xing, Siyuan Xiang and Lin Cai, "A Real-Time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks," IEEE Journal On Selected Areas In Communications, Vol. 32, No. 4, April 2014.
- [2] Dhanya, Anantharaman.G., "Literature Review on Video Streaming in Social Networks," IJSRD - International Journal for Scientific Research & Development| Vol. 2, Issue 12, 2015 | ISSN (online): 2321-0613.
- [3] Altman.E, "Constrained Markov decision processes," Chapman & Hall/CRC, 1999.
- [4] T. Stockhammer, "Dynamic adaptive streaming over HTTP : standards and design principles," in ACM MMSys'11, 2011, pp. 133–144.
- [5] John G. Apostolopoulos, Wai-tian Tan, Susie J. Wee, "Video Streaming: Concepts, Algorithms, and Systems," Mobile and Media Systems Laboratory HP Laboratories Palo Alto HPL-2002-260 September 18th, 2002*
- [6] Prathibha T P, "DASU: High Quality Video Streaming Over Multiple Wireless Access Network Using Real Time Adaptive Algorithm," International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438
- [7] T. Stock hammer, "Dynamic adaptive streaming over HTTP –: standards and design principles," in ACM MMSys'11, 2011.
- [8] R. Mok, X. Luo, E. Chan, and R. Chang, "QDASH: a QoE-aware DASH system," in ACM MMSys'12, 2012
- [9] D. Kaspar, K. Evensen, P. Engelstad, and A. Hansen, "Using HTTP pipelining to improve progressive download over multiple heterogeneous interfaces," in IEEE ICC'10, 2010, pp. 1–5.
- [10] K. Evensen, T. Kupka, D. Kaspar, P. Halvorsen, and C. Griwodz, "Quality-adaptive scheduling for live streaming over multiple access networks," in ACM NOSSDAV'10, 2010, pp. 21–26.
- [11] K. Evensen, D. Kaspar, C. Griwodz, P. "Using bandwidth aggregation to improve the performance of quality-adaptive streaming," Signal Processing: Image Communication, vol. 27, no. 4, pp. 312–328, 2012.
- [12] Haakon Riiser, "Adaptive Bitrate Video Streaming over HTTP in Mobile Wireless Networks," ISSN 1501-7710, June 16, 2013
- [13] Ashwini N. Nikumbhe, Chhaya Nayak, "A DASH Based Scheme for Video Streaming over Multiple Wireless Access Networks: A Review," International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 6, June 2015
- [14] H. Wang and N. Moayeri, "Finite-state Markov channel-a useful model for radio communication channels," IEEE Trans. Veh. Technol., vol. 44, no. 1, pp. 163–171, 1995.
- [15] S. Xiang, L. Cai and J. Pan, "Adaptive scalable video streaming in wireless networks," in ACM MMSys'12, 2012, pp. 167–172.
- [16] Shin-Hung Chang, Ray-I Chang, Jan-Ming Ho, and Yen-Jen Oyang, "A Proxy Effect Analysis and Fair Adaptation Algorithm for Multiple Competing Dynamic Adaptive Streaming over Http Clients," IEEE TRANSACTIONS ON BROADCASTING, VOL. 53, NO. 1, MARCH 2007
- [17] Min Xing, Siyuan Xiang and Lin Cai, "Rate Adaptation Strategy for Video Streaming over Multiple Wireless Access Networks," Wireless Networking Symposium 978-1-4673-0921-9/12/\$31.00 ©2012 IEEE
- [18] Dhanya, Anantharaman.G., "Literature Review on Video Streaming in Social Networks," International Journal for Scientific Research & Development, Vol. 2, Issue 12, 2015 | ISSN (online): 2321-0613
- [19] Hong Shen Wang and Nader Moayeri, "Finite-State Markov Channel-A Useful Model for Radio Communication Channels," IEEE TRANSACTIONS ON Vehicular Technology. Vol. 44, No. 1, February 1995
- [20] X. Hou, P. Deshpande, and S.R. Das, "Rate Adaptation Strategy for Video Streaming over Multiple Wireless Access Networks," In ACM NOSSDAV'10, pages 21–26, 2010.