International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 6, Issue 9, September2017, ISSN: 2278 – 1323

K-means based rate adaptation algorithm in DASH

Asim Rafiq Bhat.

***Abstract*— Video is becoming one the biggest traffic generating element and the current progressive download approach has been found inefficient in terms of resource utilization .A machine learning based approach is proposed for the video streaming with DASH (Dynamic adaptive streaming over HTTP) as an underlying architecture to help client adapt to the changing streaming environment. The reason for using machine learning for adaptation is to make client learn about the environment that too in unsupervised manner. The various work related to DASH has also been discussed with a brief introduction to the DASH working.**

***Index Terms*—DASH,), KB (knowledge base), ML (Machine learning, PSNR, RTP.**

I. INTRODUCTION

Internet is seeing a large number of applications being added on the over the already present services. This includes online games; e-classes like Google classes, newly featured video conferencing sessions, HDTV etc. This has led to the increase in the content generation over internet channels.

It is estimated that nearly 65% of the internet traffic will be of video type [1]. This requires a new approach and framework for streaming video over the internet and DASH is being proposed as one of the possible solutions. Rather than streaming statically as was done in progressive downloads DASH allows streaming to be dynamic and its fidelity adaptive to the current network signature. This requires moving on from RTP being used at the application layer to the HTTP as DASH is based on HTTP.

A. HTTP VS RTP [8]

There are various reasons for using HTTP over RTP. The reasons include:

- RTP packets not allowed through firewalls.
- RTP streaming requires maintaining separate session for each client, making large scale deployments resource intensive.
- HTTP is cost effective for no additional resources are required as HTTP is already part of standard web use.
- HTTP is stateless protocol, so no need to maintain session information at server side.
- Firewalls are automatically configured to support outgoing connections.

Moreover HTTP uses TCP at the transport layer, thus increasing reliability.

Before discussing the related work section a brief introduction of DASH.

B. DASH working

DASH (Dynamic adaptive Streaming over HTTP) is emerging as an alternative to the *progressive download* because of the various disadvantages the later has over the former. These include

- Bandwidth wastage during content switch.
- Progressive download is not data fidelity adaptive.
- It does not support live media services [2].

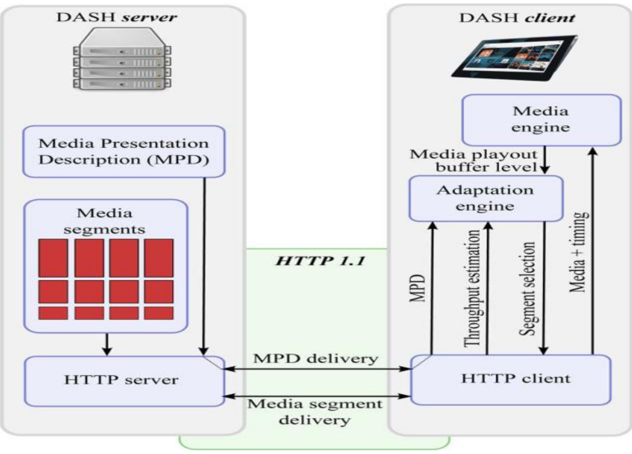

Fig. Dash Working [10]

DASH helps in addressing the concerns of the RTP streaming and progressive downloads. DASH works as follows:

The media file is divided into segments temporally (i.e. time) not in terms of file length(KB/MB). Different encoded versions known as representations are stored at the media server or are calculated on the fly for various HTTP requests. Whether to store all the representation at the first hand or determine on the fly is storage and computing power trade off. When user clicks for a video

Manuscript received sept, 2017

***Asim Rafiq Bhat**, CSE NIT Srinagar, India.*

to be streamed a MPD file is downloaded first. MPD is XML file that needs to be parsed by the video application for the available representations and the adaption part of the video stream application selects the best representation depending on the user preferences or the device state and network properties. Each segment is downloaded using the same procedure.

Thus the control for the type of bitrate requests to be generated by the video application is given to the client. The adaptation can also be performed same way either at the server side or the proxy side.

Let us now discuss about the related work

II. RELATED WORK

The field of video streaming with DASH as an approach is relatively a new research area. There have been some tools by the private MNC's and as well some proposed algorithms by some researchers. K.P.Mok[3] in 2011 evaluated the relationship between three levels of QoS in HTTP Streaming, i.e. network QOS, application QoS, user QoS and found that network throughput is lowered by packet losses and RTT thus increasing re-buffering frequency and Re- buffering frequency was the main factor affecting MoS(mean opinion score) variance. Sameer Akshabi [4] evaluated the three players in the market Microsoft smooth-streaming player, Adobe player OSMF, and Netflix player and found that Adobe player does not converge to best available representation, while smooth streaming reacts very late and for too long to bitrate changes and although Netflix provides better video quality by being aggressive to changes but leads to oscillations. M Gobhadi [5] proposed a server mechanism for rate limiting the amount of the traffic to be generated for the particular client but the only problem is for the server to get the client device parameters like screen size, resolution i.e. spatial properties.

Benjamin Rainer [6] proposed javascript based solution using Google chrome API Media source extension for adaptation at the browser level, but the solution is not browser dependent.

Y Lui [7] proposed a machine learning based approach for selecting the best connection server for the single client among the various available using State vector Regression technique, but this resulted in high error rate. DK Krishnappa [8] used DASH for YouTube as case study and found 95% bandwidth requirement reduction for low quality videos and 83% for HD videos. K.M Chan [9] proposed a quality driven approach to rate control to reduce a request-response overhead. Ricky et.al proposed a QDASH algorithm for users for whom video degradation is more annoying than subtle improvements. It was based on RTT variations for measuring bandwidth variations.

III. PROPOSED WORK

In our work we proposed an approach based on machine learning which uses two parameters for adaptation namely buffer size and network throughput. The adaptation is proposed to be performed at client side. We are proposing the unsupervised machine learning approach in the algorithm is supposed to learn while working in real-time. To give user best video quality experience we have proposing a variable length buffer to react to changing environment. If the buffer is full with particular representation and network is allowing us to switch to the higher representation we increase the next bitrate request and if determined by algorithm decrease the size of available buffer to accommodate higher quality segments. Similar is the case when the network conditions are degrading for the higher representations and the next bitrate request representation is decreased to accommodate lower representation to prevent re-buffering freeze.

Decision parameters

1. Re-buffer frequency: It is the time spent as freeze time while video content is getting buffered.
2. Accumulation ratio: it is the ratio of download rate to required bitrate.

The proposed DASH algorithm has been simulated in java. The main focus of the algorithm is improving the QoE for the users.

It has been assumed that lesser the Re-Buffering time and lesser inverse accumulation ratio better the quality of experience and vice versa.

N is the number of Segments to be requested. E.g., if video length is 100 sec and segment length= 4sec then N=25.

The objective parameters have been estimated over three download rates given as: 1Mbps. 2Mbps. 3Mbps, and compared with progressive download static data representation given by:

Data formats	Representation symbols	BitRate [13]
144p	R_0	80 kbps
240p	R_1	350 kbps
360p	R_2	520 kbps
480p	R_3	830 kbps
720p	R_4	1.6 mbps
1080p	R_5	3 mbps

The reason for estimating the objective parameters over standard accumulation ratio of 1.25 is that it has been found that for smooth streaming in YouTube the ratio maintained is nearly 1.25 [14].

In adaptation, the algorithm used is K-means, a clustering based unsupervised machine learning algorithm [15]. The initial step of the algorithm to initialize the clusters to random centroids has been modified to be initialized as standard download rate for each representation given by, determined with a possibility of a cluster being empty also for lower available download rates at the end of learn phase.

The algorithm makes transition decisions over a range of IAR value changes to make transitions smooth and incorporate error in case of bandwidth spikes.

IV. Results and Conclusion

The objective variables over which the QoE have been evaluated are Re-buffer time as already discussed. The dynamic algorithm is compared with static HD video requests over the three different download rates and has been found that the re-buffer time is far less than static requests. Moreover the re-buffer time remains same for all the download rates meaning that the dynamic algorithm converges well to the maximum possible bitrate without degrading the QoE. Fig 2 and Fig. 3 shows the results of 1000 sec video being streamed over the four different variations, one dynamic and 3 static .In dynamic approach the Re-buffer time is approximately 5 secs and for static variations it varies from 10 sec to minutes. The re-buffer time is more for static highest quality request over smallest possible download rate over which the simulation has been performed.

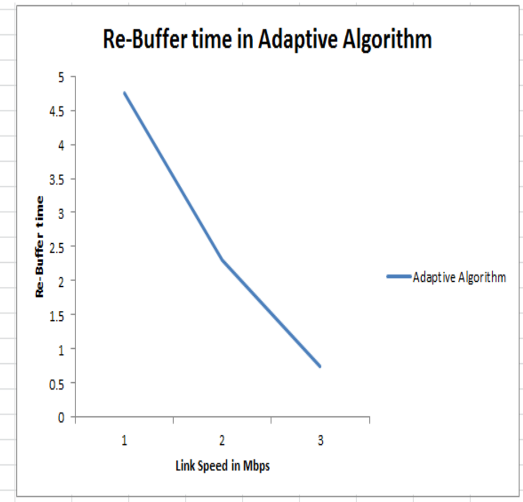

Figure. 2

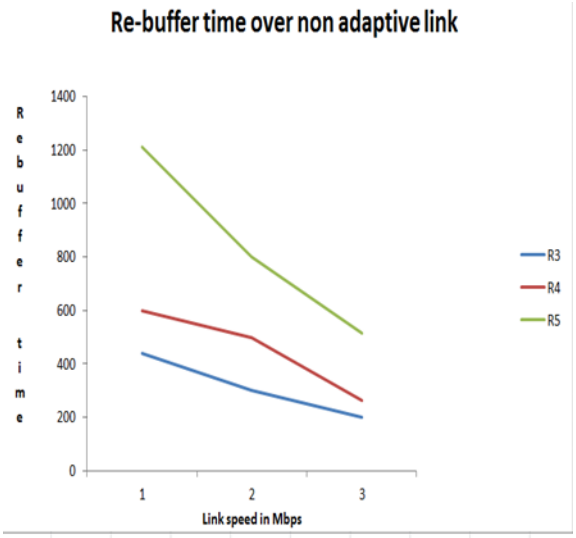

Figure. 3

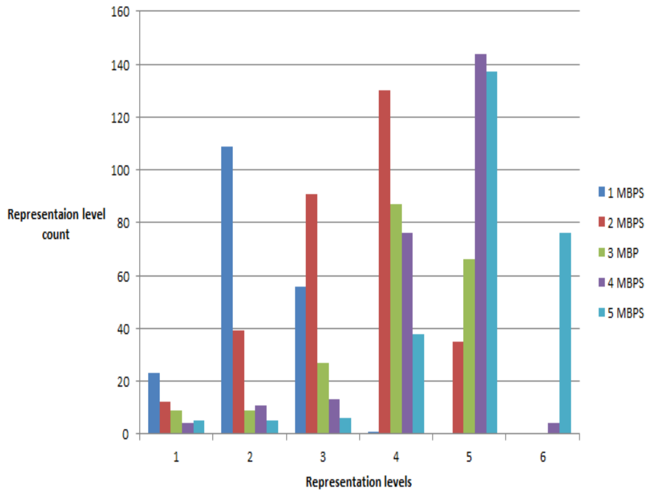

Figure. 4

Figure 4 shows the algorithm transitions to different levels in proposed adaptive streaming approach. The algorithm converges well and tries to stay in particular level for a particular representation level thus avoiding spikes and oscillations.

Video streaming is getting a great attention from the researchers nowadays. There is great scope for research to explore the new approach towards video streaming with DASH as a framework. There have been various proposals with adaptation being the central focus of attention. The adaptation can be performed either at the server side or the client side or even in between at proxy.in recent years several researches has been conducted to identify the period of .In our research we proposed an algorithm for adaptation at the client side based on machine learning to improve under QoE by studying objective attributes for QoE measurement.

REFERENCES

[1] Stockhammer, Thomas. "Dynamic adaptive streaming over HTTP--: standards and design principles." Proceedings of the second annual ACM conference on Multimedia systems. ACM, 2011.

[2] Sod agar, Iraj. "The mpeg-dash standard for multimedia streaming over the internet." *IEEE Multimedia* 4 (2011): 62-67

[3] Mok, Ricky KP, Edmond WW Chan, and Rocky KC Chang. "Measuring the quality of experience of HTTP video streaming." *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. IEEE, 2011.

[4] Akhshabi, Saamer, et al. "An experimental evaluation of rate-adaptive video players over HTTP." *Signal Processing: Image Communication* 27.4 (2012): 271-287.

[5] Ghobadi, Monia, et al. "Trickle: Rate Limiting YouTube Video Streaming."*Usenix Annual Technical Conference*. 2012.

[6] Rainer, Benjamin, et al. "A seamless Web integration of adaptive HTTP streaming." *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012.

[7] Tian, Guibin, and Yong Liu. "Towards agile and smooth video adaptation in dynamic HTTP streaming." *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012.

[8] Krishnappa, Dilip Kumar, Divyashri Bhat, and Michael Zink. "DASHing YouTube: An analysis of using DASH in YouTube video service." *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*. IEEE, 2013.

[9] Chan, K. M., and Jack YB Lee. "Improving adaptive HTTP streaming performance with predictive transmission and cross-layer client buffer estimation." *Multimedia Tools and Applications* (2015): 1-21.

[10] Seufert, Michael, et al. "A survey on quality of experience of HTTP adaptive streaming." *Communications Surveys & Tutorials, IEEE* 17.1 (2014): 469-492.

[11] Sapankevych, Nicholas, and Ravi Sankar. "Time series prediction using support vector machines: a survey." *Computational Intelligence Magazine, IEEE* 4.2 (2009): 24-38.

[12] Mok, Ricky KP, et al. "QDASH: a QoE-aware DASH system." *Proceedings of the 3rd Multimedia Systems Conference*. ACM, 2012.

[13] http://www.sciencemedianetwork.org/Blog/20130624_YouTube_bitrates

[14] Casas, Pedro, et al. "When YouTube Does not Work—Analysis of QoE-Relevant Degradation in Google CDN Traffic." *Network and Service Management, IEEE Transactions on* 11.4 (2014): 441-457.

[15] http://www.cse.hcmut.edu.vn/~dtanh/download/Data Mining Concepts and Techniques 2ed - 1558609016.pdf