ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

SMART STREAMING FOR ONLINE VIDEO SERVICES USING REAL TIME ADAPTIVE ALGORITHM

Suriyakumar.E.R [1], Matheswaran.V [2]

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India
2. Asst.Professor, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

***Abstract*—Bandwidth cost is a significant concern for online video service providers. Today's video streaming systems mostlyuse *HTTP streaming*, with users accessing video segments as HTTP requests. A frequently used strategy is to serve all user requests as fast as possible, as if the user is downloading a file. The downloading rate can often far exceed the playback rate, when the system is below the peak load. This is known as *progressive downloading*. Since users may quit before viewing the complete video, however, much of the downloaded video can be "wasted." The authors developed a *smart streaming* strategy that can significantly improve overall streaming service quality under given server bandwidth. The improvement is achieved by avoiding the waste in buffer. The proposed smart streaming technique is evaluated by modeling, analysis, and simulation, as well as experimentation using a prototype implementation.**

***Index Terms*— Quality of experience (QoE), streaming strategy, user behavior, video-on-demand (VoD) service.**

I.INTRODUCTION

Internet technology is changing at a rapid pace and the faster the technology changes, the more people expect from the Internet. Users were once satisfied with text and still images on their web pages. Now they want to see video and want it fast. Users want the quality to be as good as what they see on their television. Video streaming is one way to deliver video over the Internet. Though far from a perfect solution, streaming video technology is becoming more powerful all the time. With video streaming, designers can broadcast lectures, make announcements, deliver seminars, or show exactly how something is supposed to work. Users can see it anytime, quenching some of their thirst for fast, high-quality video. Video streaming provides flexibility as well. Users can view what they want and when they want. Streaming media is a sequence of "moving images" that are sent incompressed form over the Internet and displayed by the viewer as they arrive. It is streaming video with sound. With streaming video or streaming media, a Web user does not have to wait to download a large file before seeing the video or hearing the sound. Instead, the media is sent in a continuous stream and is played as it arrives. The user needs a player, which is a special program that uncompressed and sends video data to the display and audio data to speakers. A player can be either an integral part of a browser or downloaded from the software maker's Web site. Streaming video is usually sent from prerecorded video files, but can be distributed as part of a live broadcast "feed." In a live broadcast, the video signal is converted into a compressed digital signal and transmitted from a special Web server.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

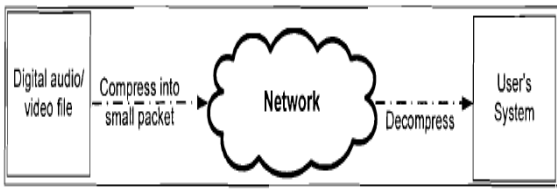

Fig.1 How Media Streaming Works

QoS Metrics

Several objective tests were conducted to evaluate the performance of the proposed approach using several QoE

related QoS metrics. For instance, according to , the layer variations will decrease the users' watching experience, and thus the number of layer variations was used to reflect the smoothness in our experiments. The performance metrics used in our experiments are listed as follows:

•**Startup latency (SL):** It is defined as the duration from sending out the first segment request to the beginning of playing back the first segment, which can be also regarded as buffering time.

Playback fluency: The number of segments that miss the playback deadline, and the playback freeze ratio, which calculates the percentage of the duration of playback freezes over the total video streaming time are evaluated.

Average playback quality (APQ): The average PSNR of all received frames.

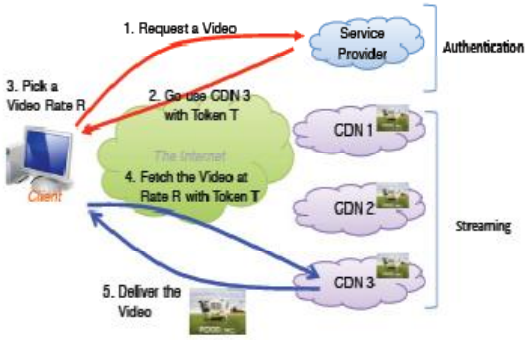

Fig.2 Common Architecture of Video Streaming Services

II. LITERATURE SURVEY

The paper named "**Dynamic adaptive streaming over HTTP –: standards and design principles" [1],** the author T. Stock hammer 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 playout of the sequence of segments,

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

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 research paper named "**Qdash: A Qoe-Aware Dash System**" [2], authors R. Mok, X. Luo, E. Chan, and R. Chang 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 integrate 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 paper propose a QoE-aware quality adaptation algorithm for DASH based findings. Finally, it integrate 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 [5] **"A Proxy Effect Analysis and Fair Adaptation Algorithm for Multiple Competing Dynamic Adaptive Streaming over Http Clients**" concentrates on the negative effects introduced when multiple clients are competing for a bottleneck and how proxies are influencing this

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

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. fair adaptation scheme (FAS) aims to address the problem identified in Section 3. 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 backoff. 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 networkcaused.

This paper named **"Using HTTP Pipelining to Improve Progressive Download over Multiple Heterogeneous Interfaces"[5]**authorD. Kaspar, present an improved version that utilizes HTTP's capability of request pipelining in combination with range retrieval requests. The use of very small segments no longer impairs the efficiency of throughput aggregation, which additionally makes the solution robust against link variances and agnostic to network heterogeneity. Major hurdle in the deployment of a multilink solution is the lack of server-side support. Although there exist suggested modifications to TCP standard transport protocols are unable to provide host-base aggregation of individual flows. Thus, a common approach is to provide specialized libraries for transparent partition of application-layer data into multiple independent transport streams. However, the implementation of such middleware requires software modifications to all involved clients and servers. In order to provide easy deployment and interoperability with existing server infrastructure, paper proposed a purely client-based solution for progressivelydownloading a single large file over multiple interfaces. HTTP pipelining is a method that "allows a client to make multiple requests without waiting for each response, allowing a single TCP connection to be used much more efficiently, with much lower elapsed time, in the absence of pipelining each range retrieval request must be sequentially handled by the server before the client can send the next request. Thus, for each request, an average time overhead of one round-trip time is incurred. For a large number of small file segments, this overhead significantly impairs the throughput of high-latency connections Another paper named "**Quality-Adaptive Scheduling for Live Streaming over Multiple Access Networks"**[6] focus on achieving smooth and quality-adaptive streaming of live video. Paper present a client-side scheduler that retrieves segments of several video encodings over heterogeneous network interfaces simultaneously. By extending the DAVVI streaming platform with support for multi homing, the proposed scheduler's performance is experimentally evaluated. The results show that the scheduler reduces the video interruptions and achieves a higher and more stable average quality over multiple, truly heterogeneous wireless interfaces. This paper introduces an adaptive, pull-based scheduler that achieves smooth

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

playback by scheduling requests for video segments of different quality levels over multiple interfaces simultaneously. Results show that the combined operation of multiple interfaces significantly enhances the quality of live video streaming. Even in a truly heterogeneous environment with WLAN and HSDPA links, the presented scheduler achieves an increased video quality and reduces playback interrupts.

"**Using Bandwidth Aggregation To Improve The Performance Of Quality-Adaptive Streaming**",Paper [7]focused on bandwidth aggregation on host multi homed devices. Even though bandwidth aggregation has been a research field for several years, the related works have failed to consider the challenges present in real world networks properly, or does not apply to scenarios where a device is connected to different heterogeneous networks. In order to solve the deployment challenges and enable the use of multiple links in a way that works in a real-world network environment, have created a platform-independent framework, called MULTI. MULTI was used as the foundation for designing transparent (to the applications) and application-specific bandwidth aggregation techniques. MULTI works in the presence of Network Address Translation (NAT), automatically detects and configures the device based on changes in link state, and notifies the application(s) of any changes. The application-specific bandwidth aggregation technique presented in this paper was optimized for and evaluated with quality-adaptive video streaming. The technique was evaluated with different types of streaming in both a controlled network environment and real-world networks. Adding a second link gave a significant increase in both video and playback quality. However, the technique is not limited to video streaming and can be used to improve the performance of several, common application types.

III. EXISTING SYSTEM

Video streaming is gaining popularity among mobile users recently. Considering that the mobile devices have limited computational capacity and energy supply, and the wireless channels are highly dynamic, it is very challenging to provide high quality video streaming services for mobile users consistently. It is a promising trend to use multiple wireless network interfaces with different wireless communication techniques for mobile devices. Meanwhile, as video data are transmitted over HTTP protocols, the video streaming service can be deployed on any web server. However, the video quality version can only be manually selected by users and such decision can be error-prone.

Disadvantages of Existing System:

- The smart phones only have limited storage space; it is impractical to maintain a very large buffer size.
- The buffered unwatched video may be wasted if the user turns off the video player or switches to other videos.
- Instead, ideally, when one bit is in error, the effect on the reconstructed video should be unperceivable, with minimal overhead. In addition, the perceived video quality should gracefully and proportionally degrade with decreasing channel quality.

IV. PROPOSED SYSTEM

In this paper we proposed dynamic adaptive streaming over HTTP has been proposed. In a DASH system, multiple copies of pre-compressed videos with different resolution and quality are stored in segments. We formulate the multi-link

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

video streaming process as a reinforcement learning task. For each streaming step, we define a state to describe the current situation, including the index of the requested segment, the current available bandwidth and other system parameters. A finitestate Markov Decision Process (MDP) can be modeled for this reinforcement learning task. The reward function is carefully designed to consider the video QoS requirements, such as the interruption rate, average playback quality, and playback smoothness, as well as the service costs Maintain the playback rate for the purpose of benchmarking. It does not incur any waste even if the user departs early. The existing system requires more time to execute. In existing System progressive download, one of the most popular and widely deployed streaming techniques, buffers a large amount of video data to absorb the variations of bandwidth. Video data are transmitted over HTTP protocols; the video streaming service can be deployed on any web server. In which smart phones only have limited storage space, it is impractical to maintain a very large buffer size. The buffered unwatched video may be wasted if the user turns off the video player or switches to other videos. In the proposed system Adaptive bit rate streaming method is used. To improve client side streaming adaptive bit streaming method is used. Multiple video formats can be displayed. Christo Ananth et al. [8] discussed about a system, In this proposal, a neural network approach is proposed for energy conservation routing in a wireless sensor network. Our designed neural network system has been successfully applied to our scheme of energy conservation. Neural network is applied to predict Most Significant Node and selecting the Group Head amongst the association of sensor nodes in the network. After having a precise prediction about Most Significant Node, we would like to expand our approach in future to different WSN power management techniques and observe the results. In this proposal, we used arbitrary data for our experiment purpose; it is also expected to generate a real time data for the experiment in future and also by using adhoc networks the energy level of the node can be maximized. The selection of Group Head is proposed using neural network with feed forward learning method. And the neural network found able to select a node amongst competing nodes as Group Head. When the connection is good, the viewer gets a high-quality, high-data-rate stream, but if connection speed drops, the server will send a lower-data-rate file to ensure a continuous connection, although at lower quality. Adaptive streaming provides the best of all possible worlds: great quality-video for those with the connection speed to retrieve it.

Advantages of Proposed System:

- ✓ Smooth and high quality video streaming.
- ✓ Avoid playback interruption and achieve better smoothness and quality.

Proposed Algorithm for Video Streaming

The proposed algorithm works with the following steps;

Step 1: Initiate Client Request by connecting to the Server with information about screen resolution and bit rate.

Step 2: Server Receives Request and identifies the occurrences of connectiion

Step 3: Server response with Stream prepared message and start sending video packets Bit rate adjusted accoring to clients' link.

Step 3.1: After making connection the data send over the direct connection defined protocol

Step 4: Server initiate Band width measurement for the link

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

Step 6: Based on the Band width Server split the video content to separate stream and send through the established link

Step 7: Client receives data coming through link to a unique buffer. Displaying video while downloading.

Step 9: Stop

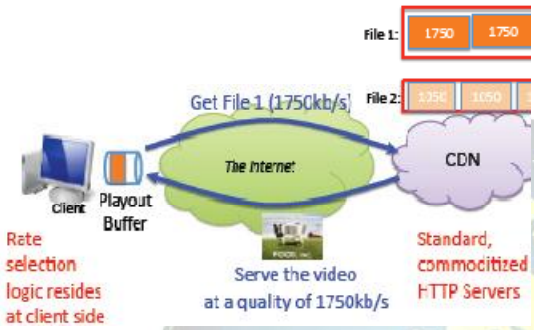

Fig. 3 Architecture of Video Rate Selection Over HTTP

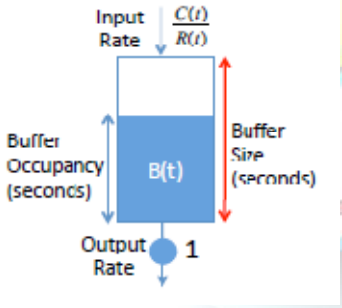

Fig 4: The video playback buffer, tracked in seconds. Every second, the buffer outputs one second of video to play to the user. The buffer also receives video at the rate that can be represented as a ratio between the system capacity and the selected video rate, C (t)/R (t).

Proposed Algorithm for Compression

J-bit encoding (JBE) works by manipulate bits of data to reduce the size and optimize input for other algorithm. The main idea of this algorithm is to split the input data into two data where the first data will contain original nonzero byte and the second data will contain bit value explaining position of nonzero and zero bytes. Both data then can be compress separately with other data compression algorithm to achieve maximum compression ratio. Step-by-step of the compression process can be describe as below:

1. Read input per byte, can be all types of file.
2. Determine read byte as nonzero or zero byte.
3. Write nonzero byte into data I and write bit '1' into temporary byte data, or only write bit '0' into temporary byte data for zero input byte.
4. Repeat step 1-3 until temporary byte data filled with 8 bits of data.
5. If temporary byte data filled with 8 bit then write the byte value of temporary byte data into data II.
6. Clear temporary byte data.
7. Repeat step 1-6 until end of file is reach.
8. Write combined output data

a) Write original input length.

b) Write data I.

c) Write data II.

9. If followed by another compression algorithm, data I and data II can be compress separately before combined (optional).

As for step-by-step of the decompression process can be describe below:

1. Read original input length.
2. If was compressed separately, decompress data I and data II (optional).
3. Read data II per bit.
4. Determine whether read bit is '0' or '1'.
5. Write to output, if read bit is '1' then read and write data I to output, if read bit is '0' then write zero byte to output.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

6. Repeat step 2-5 until original input length is reach.

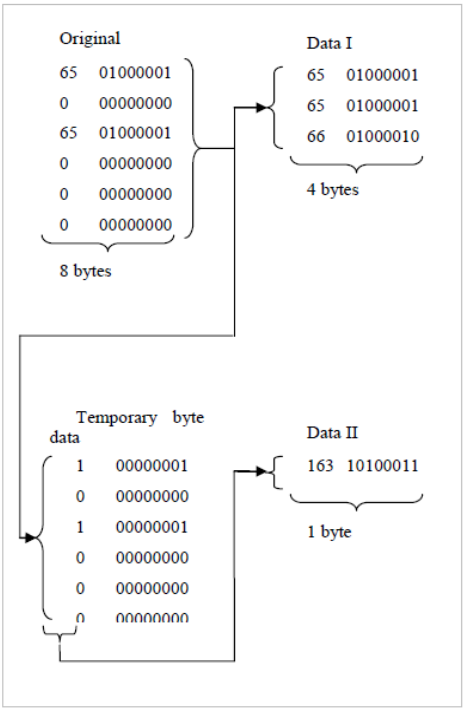

Fig 5 J-bit Encoding process

Figure 2 shows visual step-by-step compression process for J-bit encoding. Inserted original input length into the beginning of the output will be used as information for data I and data II size.

V. MODULES

Users:

It defines how the users have to be interacting with the site. Each user has own authentication based on the registration done. All the users' information is maintained corresponding to their video files downloads.

A. **Client Registration**: Whenever a new client enter means first register to the server side. It contains user name ,password ,email id etc. After registration process complete then only it will communicate to the server. Client will enter the condition and network bandwidth to the server side. It will be very useful for the client side streaming.

B. **Login:** In the client side user first login to the server. User will send request to the server. User give their valid user name and password. Server verify the details and immediately send the response to the client. In the login phase only valid user can enter their details.

Streaming Module:

Defines the procedure of transferring the video files in the forms of streams. Each video file is converted into bit and bytes while playing the file. Each part of downloaded bit and bytes stream is played along with the corresponding another part is downloaded in parallel.

Rate Adaptation Module

The video streaming process can be formulated as a reinforcement learning task. The environment sends a state signal for each video segment to the agent, and the agent will determine the best action correspondingly. For each action, the environment replies a reward to the agent. Considering the Markov property of the system states, a Markov Decision Process (MDP) can be formulated for the streaming process, and the state transition model of the Markov process needs to be devised. 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.

Download Module

After initialization, the client will request the video information which includes video resolutions, bitrates and qualities from the server. The rate adaptation agent will request a video segment of

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

appropriate quality version based on the current queue length and estimated available bandwidth. Once the request decision is made, HTTP requests over the network 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. The difference between downloaded and wasted is the amount of bandwidth used for playback. The EB (equal buffer) strategy is used in this module which requires the server to know the buffer state of all users. So buffer wastage can be minimized as possible.

VI.CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we formulate and study a ractical problem for large VoD streaming service providers: how to smartly utilize bandwidth resource to improve streaming QoE and peak load bandwidth requirements. We show that these two goals are highly coupled, and if you can cut down the bandwidth waste you can use the saved bandwidth to improve QoE as well as save peak load bandwidth costs. The key is to understand user early departure behavior. In our study, we were able to collaborate with a large-scale VoD service provider and collected real world user behavior logs in their system. The measurement and subsequent modeling and analysis led us to propose an effective rate allocation strategy for the problem at hand. From both simulation and by experimenting with a prototype implementation, we demonstrate how our ideas can be implemented in practice and the advantage they can bring to VoD services. For future work, we see many interesting directions. On the analytical side, we think it is possible to further refine the abstract model for analyzing smart streaming at an abstract level. This would extend our insights into the problem. We also want to study how to integrate smart streaming with DASH. On the experimental side, we want to extend our evaluation from single video to multiple videos, with different lengths and multiple resolutions. Finally, we will also work closely with the industry to get our ideas adopted in practice.

References

[1] T. Stock hammer, “Dynamic adaptive streaming over HTTP –: standards and design principles,” in ACM MMSys’11, 2011.

[2] R. Mok, X. Luo, E. Chan, and R. Chang, “QDASH: a QoE-aware DASH system,” in ACM MMSys’12, 2012

[3] S. Xiang, L. Cai, and J. Pan, “Adaptive scalable video streaming in wireless networks,” in ACM MMSys’12, 2012

[4] K. Tappayuthpijarn, T. Stockhammer, and E. Steinbach, “HTTP-based scalable video streaming over mobile networks,” in IEEE ICIP’11

[5] 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.

[6] 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.

[7] 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.

[8] Christo Ananth, A.Nasrin Banu, M.Manju, S.Nilofer, S.Mageshwari, A.Peratchi Selvi,

"Efficient Energy Management Routing in WSN", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 1, Issue 1, August 2015,pp:16-19

[9] L. Cai, S. Xiang, Y. Luo, and J. Pan, "Scalable modulation for video transmission in wireless networks," IEEE Trans. Veh. Technol, 2011.

[10] T.Kupka, P. Halvorsen, and C. Griwodz, "Performance of on-off traffic stemming from live adaptive segmented HTTP video streaming," in IEEE LCN'12, 2012, pp.401–409.

[11] D.Kaspar, K. Evensen, P. Engelstad, and A. Hansen, Using HTTP pipelining to improve progressive download over multiple eterogeneous interfaces," in IEEE ICC'10, 2010, pp. 1–5.

12] Kaspar, D., Evensen, K., Engelstad, P., Hansen, A., Halvorsen, P., and Griwodz, C. Enhancing video-on-demand play out over multiple heterogeneous access networks. In Proc. IEEE CCNC (2012).

[13] Kaspar, D., Evensen, K., Engelstad, P., and Hansen, A. F. Using HTTP pipelining to improve progressive download over multiple heterogeneous interfaces. In Proc. IEEE ICC (2013).

[14] C. Liu, I. Bouazizi, and M. Gabbouj, "Parallel adaptive HTTP media streaming," in Proc. ICCCN, 2011, pp. 1–6.

[15] I.Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," IEEE Multimedia, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[16] L. Cai, Y. Luo, S. Xiang, and J. Pan, "Scalable modulation for scalable wireless videocast," in Proc. IEEE Infocom, San Diego, CA, Mar. 2010.

[17] R. Zhang, R. Ruby, J. Pan, L. Cai, and X. Shen, "A hybrid reservation/ contention-based MAC for video streaming over wireless networks," IEEE J. Sel. Areas Commun., vol. 28, no. 3, pp. 389–398, Apr. 2010.

[18] T. Kupka, P. Halvorsen, and C. Griwodz, "An evaluation of live adaptive HTTP segment streaming request strategies," in Proc. of IEEE LCN, 2011, pp. 604–612.