VIDEO STREAMING WITH MOBILITY

Tejaswini Patil[1], Dr.Amarsinh Vidhate[2]

[1]Student, Department of Computer Engineering, Ramrao Adik Institute of Technology, Nerul Navi Mumbai ,Maharashtra, India.

[2] Professor, Department of Computer Engineering, Ramrao Adik Institute of Technology, Nerul Navi Mumbai ,Maharashtra, India.

---***---

Abstract - *As the use of mobile phones for the internet is increasing day by day, applications like audio/video streaming are increasing by to fold but to support such applications, mobility is a great challenge. To support such applications seamlessly/continuously, efficient & sufficient buffer management is expected. In this paper, we study various proposals for video streaming with mobility. Also, we study various proposals in this regard and the state of the art analytical analysis is presented followed by our notations.*

***Key Words*: Video streaming, buffering, delay, mobility, handoff.**

1. INTRODUCTION

Nowadays Wireless technologies have wide use of multimedia. In streaming applications, media streams have to be transmitted continuously. It has to overcome the network problems such as delay, jitter, handoff, packet loss, QoS, congestion. In wireless video streaming, buffering part plays an important role. To provide better performance for streaming multimedia over best effort networks, such as the Internet and wireless networks, buffer techniques are often used on the server side, network and on the client side.

In this paper, we are going to study the Server side and client side buffers. first we will study the server side buffer which is act as multibuffer. In multibuffer we are studying two schemes scheduling and rate control. Now we will move to client-side buffer. In client side buffer they are having different names for this such as pre-buffer, playback buffer, playout buffer. In prebuffer we will study an architecture that enables the streaming client to predicatively pre-buffer multimedia data based on the input from a Network Coverage map Service (NCMS). To provide the streaming client with coverage notifications, the users share their current network characteristics and geo-location with the NCMS [6].

Playback buffering is a typical way to reduce the delay jitter of media packets before the playback, it will incur longer end-to-end delay jitter.In this buffer, we improve the balance between the emulation of delay jitter and the decrease of end-to-end delay[7].

A new play-out buffer-aware hand-off control. It aims to prevent a freeze video for as long as possible, maximizing the expected time until freezing[8].

2. RELATED WORK

Reza Rejaie et al. [1] propose an RAP (Rate-based Congestion Control) technique for unicast playback of real-time streams and other semi-reliable rate-based applications. Video streaming is much more depends on the bandwidth, also, it have to suffer from the handoff.

Lawrence Chow et al. [8] presents Wireless hand-off control typically considers only connectivity strength from the mobile terminal to alternative access points. .here video freezing must be avoided at the mobile terminal, the play-out buffer level should also be considered by hand-off control.

Jaspher W. Kathrine et al. [2] suggests packet scheduling algorithm improve the performance of the wireless networks. Packet scheduler decides which packet to be serviced or which to be dropped. Dropping of packets will be based on network parameters such as bandwidth, packet arrival rate, the deadline of packet and packet size.

Saamer Akhshabi et al. [11] search for the rate-adaptation mechanisms. How streaming affect by network available bandwidth, bitrates, delay and bottleneck problem.

Seungwan Ryuet al. [3] propose an urgency- and efficiency-based wireless packet scheduling algorithm that is able to schedule real-time and non-real-time traffics at the same time while supporting multiple users simultaneously at any given scheduling time instant.

Y. Falik et al. [4] describe an adaptive streaming algorithm that improves the QoS. In this available buffers are utilized for getting high QoS. Hongali Luo et al. [5]propose a multi-buffer scheduling scheme and control algorithm. In this server side maintains multiple buffers for packets of different importance levels. To reduce end-to-end distortion it schedules the transmission of packets based on the source buffer size and playback deadline.

Wanqing Tu et al. [7] used probing scheme for adjusting playback buffer by calculating step length with the help of delay margin and delay jitter.

3. BUFFERS IN VIDEO STREAMING

Buffering is done at each level as shown in Fig.1. Buffers are at server side, on the network, and at the client side. If the buffering is done in a good manner then it can overcome the problems like delay, handoff, jitter, QoS and congestion.

3.1. Server side buffer

On the server side Multiple buffers are present. The server side maintains multiple buffers for packets of different important levels. It schedules the transmission of each packet based on the source buffer size and playback deadline time to reduce the end-to-end distortion.

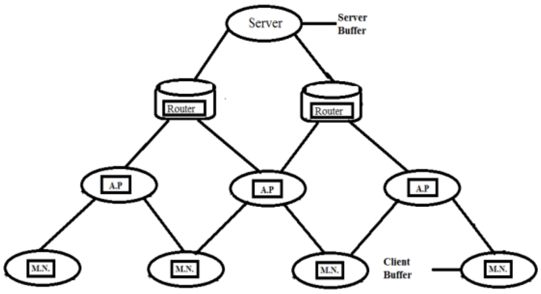

Fig.1: Buffer Locality

There are some issues with related to multibuffer they are as follows: Delay, Jitter, Handoff, Congestion, Packet loss, Bandwidth, QoS. So, to reduce the related issues multi-buffer scheme plays an important role. There are two schemes, first one is Rate Control algorithm and second one is Packet Scheduling scheme. Rate Control algorithm decides suitable transmission rate with the help of receivers buffer , playback requirement, and congestion on the network. Which packet should send first and which is latter it depends on rate constraint and also on the playback requirement.

Packet scheduling scheme decides the time and rate at which packets to be sent. Packets will be discarded if they came after the deadline because they will lead to degradation of performance. Packet scheduler takes the decision after analyzing the bandwidth, client buffer size, server buffer size, and playback requirements.

3.1.1 Structure of the Packet Scheduler

The packet scheduler operating at server side for delivering QoS to users. The packet scheduling system in a base station consists of three blocks: a packet classifier, a buffer management block (BMB), and a packet scheduler as shown in Fig.2. The packet classifier classifies incoming packets according to their types and QoS profiles and sends them to buffers in the BMB. The BMB maintains QoS statistics such as the arrival time and delay deadline of each packet, the number of packets. Finally, the packet scheduler transmits packets to users according to the scheduling priority obtained using the channel status reported by the user equipment and QoS statistics maintained in the BMB[2].

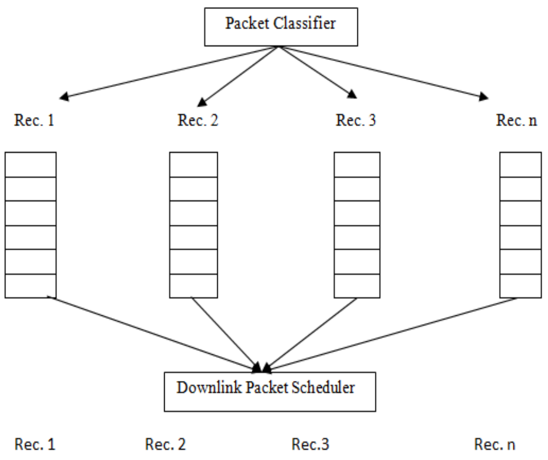

Fig.2: Packet scheduling scheme for multibuffer [3]

3.2. Client side Buffer

The client uses a buffer, which holds the data that is being sent from the network before playing it. Its input is the data received from the network and its output is the media played at the rate it was encoded in. Its size can be measured either in bits or in playing time. There are some issues with related to client buffer they are as follows: Delay, Jitter, Handoff, Congestion, Packet loss, Bandwidth, QoS.

3.2.1 Pre-buffering

Pre-buffering is used in video streaming at the client side. It is the most important in video streaming with mobility. It is used to overcome the delay, jitter, congestion. The streaming client needs to pre-buffer the data in the average media rate and available throughput to continuously play back media during the congestion.

For Pre-buffering an architecture which enables the streaming client to predicatively pre-buffer multimedia data based on the input from a Network Coverage Map Service (NCMS). To provide the streaming client with coverage notifications, the users share their current network characteristics and geo-location with the NCMS [6]. Then streaming client will look for bad coverage. When client discover the bad coverage then it will calculate the time to buffer the media. Fig. 3 shows the GLASS architecture give all information like coverage updates, congestion control and rate control[6].

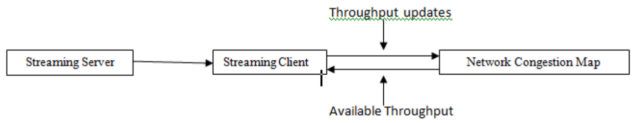

Fig. 3: GLASS Architecture [6]

3.2.2 Adaptive Playback buffer

Adaptive playback buffer (APB) minimizes the delay and jitter by adjusting the playback buffer. APB provides accuracy and efficiency. We are having architecture adaptive playback buffer at the client side.

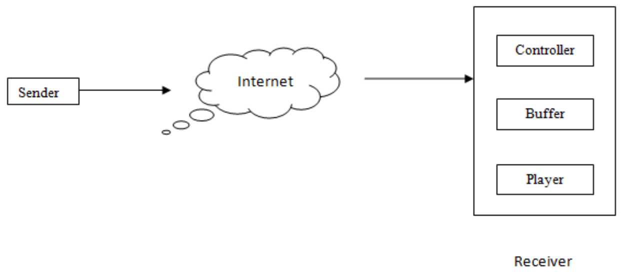

Fig.4: Adapive Playback Buffer Architecture in Clients[7]

Fig. 4 shows the architecture of the adaptive playback buffer in client. APB Controller is the component to adaptively adjust the playback buffer. The achieved instantaneous network situations are sent to the APB Controller. It then calculates the current playback buffer delay. At last, APB Controller adjusts the playback buffer. To adjust the playback buffer, it is important to know that the acceptable performances should not exceed the delay jitter bound and the end-to end delay bound . We can adjust APB with delay jitter bound *and* end-to end delay bound to achieve the real-time and continuous streaming media at the receiver. Define the delay jitter margin as the difference between the current delay jitter and jitter bound. Define the delay margin as the difference between the current end-to-end delay and end-to end delay bound. Our basic idea of adjusting APB is, using with the current playback buffer delay, to decrease the end-to-end delay by utilizing the delay jitter margin and to eliminate the frequency of the delay jitter.[7].

3.3.3 Playout Buffer

In wireless video streaming, the mobile terminal is typically equipped with a playout buffer, where downloaded video content is stored and then display to the user. During weak connectivity periods the playout buffer may run out of packets, triggering a video freeze and disrupting the user experience. Therefore, hand-off control during video streaming should not only take into account the connectivity strength to various Access Points (AP) but also the content level in the playout buffer so as to avoid video freezes. A playout buffer sensitive hand-off control for streaming video to a mobile terminal that can dynamically connect to any of a number of available APs to download the requested video content.

An example of just two APs is shown in Fig. 5 The mobile can connect to the APs via wireless channels to download content. Each AP's channel to the mobile fluctuates between various quality states. "Good" channel states (low interference, etc.) allow for successfully transmitting video packets to the mobile with high probability and achieving a high effective download rate. "Bad" channel states (high interference, etc.) induce low effective download rates

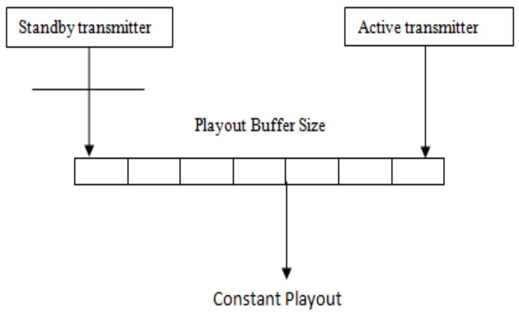

Fig. 5: Content downloading from active access point(AP) [8]

The mobile is equipped with a playout buffer, where downloaded content is stored and played out to the user at constant rate. The mobile can find the requested content at each and every AP (which could communicate themselves via a high-speed wired infrastructure network). Therefore, when it is currently connected to an AP with a bad channel, while there is another AP with a good channel, the mobile has the incentive to hand-off to the second AP. The catch is that during the handoff process no content can be downloaded. Hence, the mobile runs the risk of having a playout freeze due to buffer under run. The objective of efficient hand-off control is to avoid or delay as much as possible a video playout freeze by deciding which AP to hand-off to, based on the channel state fluctuations and playout buffer level parsimonious model is developed to capture the relevant tradeoffs and allow for computation of the optimal hand-off control. The system probe the optimal hand-off control and investigate its performance in easy way, yet insightful, case of two APs with two states per channel. Emerging key parameters are: 1) the probability that each channel persists in its current state in a time slot, and 2) the probability that an attempted hand-off completes in a time slot. Moreover, it is demonstrated that there is a certain playout buffer level

– a *tipping point* –above which a hand-off should be attempted under optimal control.

4. ANALYSIS

Table I: Notations

Notation	Description
B	Total source Buffer size
D_1	End-to-end delay of server side buffer
D_2	Total end-to-end delay
d_c	Codec delay
d_{pp}	Propagation delay
d_{pr}	Processing delay
d_t	Transmission delay
F	Playback duration for one GOP
g(n)	No. of group of pictures
N	No. of links

As a part of analysis, the end-to-end delay of each buffer is calculated for both the server side as well as client side.

$$D_{end\text{-}end}=\text{No.of links}\ [d_{transmission}+d_{propagation}+d_{processing}]$$

End-to-end delay contains Number of links, transmission delay, propagation delay, processing delay.

Transmission delay: -It is the time required to push all packets into the wireless network.

Propagation delay: -It is the amount of time it takes for the head of the signal to travel from sender to receiver.

Processing delay: - It is the time takes router to process the packet header

4.1 Multi-buffer Scheduling

For a packet, the delay is the time it may experience before it is decoded and played at the client is denoted as the server side delay.

$$D_1= N[B/d_t + d_{pp} +d_{pr} +g(n)*f]$$

4.2 Pre-buffering

For a packet, the delay it may experience after it is decoded and played at the client is denoted as client side delay.

$$D_2= D_1+N[d_t + d_{pp} + d_{pr}]$$

4.3 Adaptive Playback buffer

For a packet, the delay it may experience after it is decoded and played at the client is denoted as client side delay.

$$D_2= D_1+N[d_t + d_{pp} + d_{pr}+d_{pc}]$$

Table II: PERFORMANCE EVALUTION

Parameters **Buffer**	**Delay**
Multibuffer	$D_1= N[B/d_t + d_{pp} +d_{pr}]$
Pre-buffer	$D_2= D_1+N[d_t + d_{pp} + d_{pr}]$
Playback	$D_2=D_1+N[d_t+d_{pp}+d_{pr}+d_{pc}]$
Playout	$D_2= D_1+N[d_t + d_{pp} +d_{pr}]$

5. CONCLUSION

In this paper, we have considered various issues of video streaming along with various proposals. A state of the art analysis is presented based on the issues. We have studied server side multibuffering, as well as client side, prebuffering and play out pre-buffering playout buffering. The analysis shows that by adding various buffering stages between the server and client , impact of mobility reduces drastically.

REFERENCES

[1] Reza Rejaie, Mark Handley, Deborah Estrin "*RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet*" by DARPA as part of SPT and VINT projects.

[2] Jaspher W. Kathrine Arun Raj "Packet Scheduling Algorithms in Different Wireless Networks A Survey" International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 8, October – 2012.

[3] Seungwan Ryu, Byung-Han Ryu, Hyunhwa Seo, Muyong Shin, and SeiKwon Park "Wireless Packet Scheduling Algorithm for OFDMA System Based on Time-Utility and Channel State" ETRI Journal, Volume 27, Number 6, December 2005.

[4] Y. Falik Æ A. Averbuch Æ U. Yechiali "Transmission algorithm for video streaming over cellular networks" Springer Science Business Media, LLC 2009.

[5]Hongli Luo1, Mei-Ling Shyu1, Shu-Ching Chen2 "A MULTI-BUFFER SCHEDULING SCHEME FOR VIDEO STREAMING" 2005 IEEE.

[6] Varun Singh†, J¨org Ott†, Igor D. D. Curcio‡ †Comnet, Aalto University, Espoo, Finland ‡Nokia Research Center, Tampere, Finland "Predictive Buffering for Streaming Video in 3G Networks" 2012 IEEE.

[7] Wanqing Tu**,** Weijia Jia "Adaptive Playback Buffer for Wireless Streaming Media" IEEE 2004.

[8] Lawrence Chow, Bradley Collins, Nicholas Bambos, Christoph Peylo, Hans Einsiedler, Nico Bayer, Peter Duly, Andreas Kassler "Playout-Buffer Aware Hand-Off Control for Wireless Video Streaming " IEEE 2012.

[9] Mahbub Hassan and Salil Kanhere "Content Streaming for Mobile Environments" June 2012.

[10]Sahel Mastoureshgh "Measurement and Method for Receiver Buffer Sizing in Video Streaming" Professor Mark Claypool, Thesis Advisor Professor Emanuel Agu, Reader April 2012.

[11] Saamer Akhshabi, Constantine Dovrolis, Ali C. Begen "An Experimental Evaluation of Rate Adaptation Algorithms in Adaptive Streaming over HTTP" ACM Multimedia Systems Conference 2011 Cisco Systems, San Jose, United States February 24, 2011.