NEW APPROACH FOR APPLYING LIVE VIDEO STREAMING OVER INTERNET

Mostafa Ashour Mohamed Helala*

ABSTRACT

Video streaming technology along with internet broadcasting system is great advantage in corporate telecast, internet video broadcast, distant learning programs and other applications.
E-based learning faces many issues that retard the speed of progress such as bandwidth limitations, content challenges and many other issues. The most important technical issue of video streaming is delay in live video streaming application. This paper discusses applications of live video streaming over internet and the delay in this application along with the new approaches for improving internet delay in live video streaming applications.

INTRODUCTION

Network multimedia applications such as distant learning, corporate live meetings etc. includes the transmission and streaming of live videos or stored videos. Sometimes, a live media is temporarily transfused with stored media stream to transmit it for other various destinations. As for example, in e-based learning, a lecturer transmits the stored video stream to students while explaining the matter of the video with live audio and video media streams. In live video streaming, the media units -- MU are first captured by a camera and/or microphone that is then transmitted, while in stored media transmission, each media unit to be transmitted is read out from the source of stored media.

The delay problem of live multimedia streaming refers to the problems caused by delay between the transmission of media unit and its output at destination. The main reason for delay problem is the bandwidth limitations for transmission of media units

LITERATURE REVIEW

Video streaming is a term used to refer to transmission of "live" audio-visual media units through internet to various destinations. One does not need to download and play the media file; rather they can watch the "live" action as streaming media file starts playing immediately within few seconds while data is being transmitted. There is a few second of delay between the starting of video play at end destination and the transmission of video at source. The delay time may vary with bandwidth available at the destination point, during this delay time; the transmitted media is streamed into a buffer after which the video starts playing while the buffer remains just ahead of the playback.

Streaming media can be of two types, real time, live video streaming and stored video streaming on demand. In real-time video streaming, the live action happening at a place is transmitted via camera and microphone to many other places simultaneously. The importance of live video is that one can watch the live action within a few seconds delay. The less the delay time, the more successful is live video streaming. Various researches are going on in the direction to make the video streaming more efficient and successful technology (Horn, 2001) (2).

The key issues of live or stored video streaming is to make maximum possible usage of bandwidth and to reduce the start-up and buffering delay periods.

- Stored versus live video streaming differences: (Shuji, Tasaka: 2000) (3).

***Project Engineer, Egyptian Radio and Television Union "ERTU" Engineering Broadcast Sector**

1 - The main differences is the timing.
2 - The major timing is the processing time.
3 - Keeping the delay jitter to acceptable level which is the time the eye can detect which is 40ms according to the fact that the eye can detect 25 frames \ second

*** Finally:**

The bit rate of the live video streaming is not known but this can be over came by smoothing the coming signal for what so ever the rate is being and what so ever the rate of capturing.

* And smoothing the o \ p rate.

METHODOLOGY

In our experiment and also in the experiment of Sen, Zhing and Dey (4), a server and a receiver were built and two types of video samples were used to transmit from server to receiver. One was Fast Sapce Jam Video and other was slow Blue Brothers. The transmission was processed by using a smoothing algorithm with an optimal selection of its variations across the transmission path between server and client (Sen, Zhang, Dey, 2002) (4).

ALGORITHM

Transmitting a live video on internet not only requires the high variable bandwidth, but it also faces the problem of simultaneously streaming a heavy multimedia unit of high bandwidth to a huge number of diverse recipients with different internet capacities like buffering and bandwidth. To solve this problem of sending a live video streaming from single source to several diverse recipient sources, an algorithm to integrate work-ahead smoothing with multicast can be applied across the tree of transmission from single source to several heterogeneous recipients.

In order to facilitate temporal caching of Media Units at intermediate nodes of transmission and distribution tree, we can use technique of differential caching so that work-ahead smoothing can be achieved.

To integrate smoothing with differential caching technique, a set of algorithms can be used so that a set of optimally smooth transmission schedule can be computed to minimize peak rates and variability of MUs transmission through different links or branches in the transmission tree in presence of *interferences* of either buffer or bandwidth constraint. In presence of such *interferences* in the transmission, we can employ another algorithm to calculate the minimal buffer allocation for all nodes in the tree.

We can achieve a smooth feasible transmission schedule for the transmission of a particular video to all the recipient sources across the tree by pro-viding the calculated minimum buffer allocation for all individual nodes. In their experiment to emphasize the importance of Optimal Multicast Smoothing of Streaming Video, Sen, Zhang and Dey (4) presented MPEG-2 trace driven calculateions to demonstrate the integrated multicast smoothing and differential caching techniques and the corresponding results in substantial savings of buffer and bandwidth (Sen, Zhang, Dey,2002) (4).

The influence of buffering value on the performance of video streaming transmission from a single source to several heterogeneous recipients, Ishibashi, Tasaka and Inoue illustrated by experiment that as the buffering value increases the quality and performance of the transmission becomes higher and better. On the other hand:

The recipients with smaller buffers and congestd bandwidths will also be able to receive whole of the media units at fitting rates to such paths Quick responses are necessity of such operations; hence, a large period of buffering is often undesirable (Ishibashi, Tasaka, Inoue, 2000) (3).

It is clearly visible that least delay time are desired in both cases of live video streaming and stored video streaming. As the proposed algorithm achieves more work-ahead smoothing of transmission video across the tree at all nodes and hence reduces the delay function, it improves the quality of transmission.

*** New Modified Algorithm:**

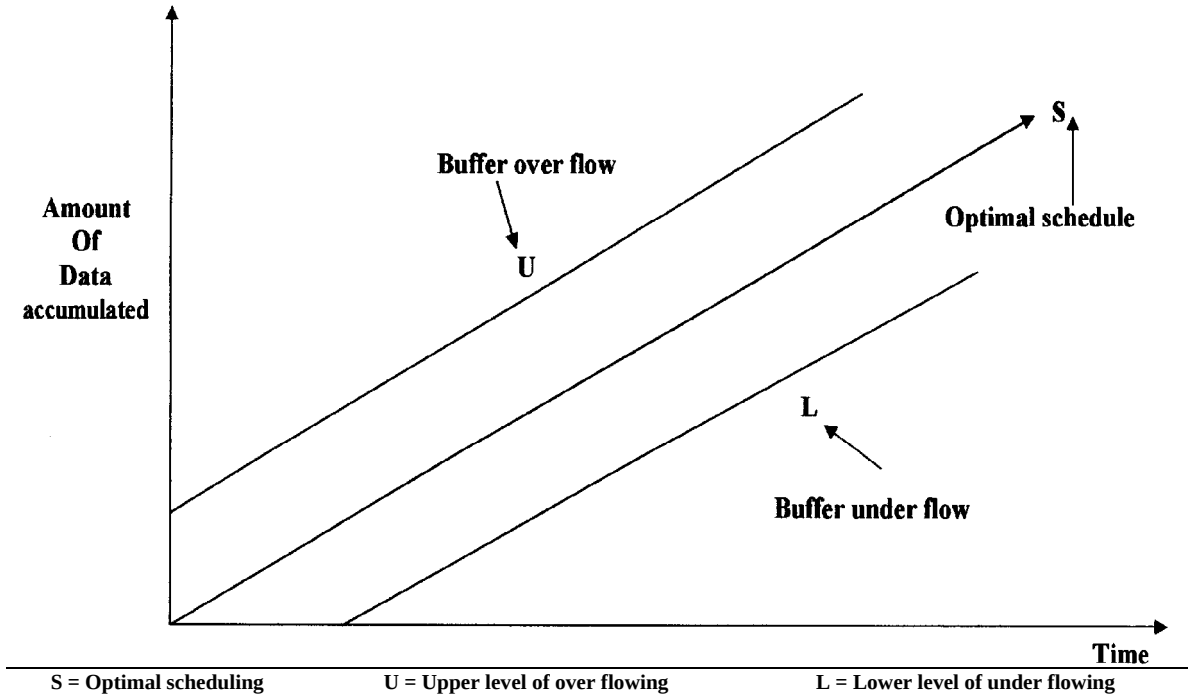
I - We used the same algorithm with modification to get optimal performance by settling the schedule in middle between overflowing and under flowing of buffering.

II - Why the selected schedule is the optimal schedule between the set of schedules:

1 - It is the middle between the two curves of under flow and over flow of buffers at the receiver side.

2 - It is of minimum risk in getting trapped in under flow or over flow status.

III - In applying this algorithm to live video streaming we insert delay function as a necessary factor for studying the algorithm in case of live video streaming.



IV - Also:

An internet delay model was inserted which is a shifted gamma distribution (A Corlett, D.I. Pullin and S. Sargood, 2002) (6).

V - We shall use heuristic approach to investigate:

1 - The effect of smoothing on delay.
2 - The effect of refresh rate on buffer usage.
3 - The effect of refresh rate on delay.

The above measurements are to verify the capability of the technique to support application of live media streaming over internet.

DISCUSSION OF RESULTS

The experiments were performed for extracting variable results for improving delay with smoothing and comparing those results with the results in case of without smoothing transmission in variable circumstances such as transmission of different types of video samples -- Space Jam and Blue Brothers, transmission with variable proto-cols, transmission with variable values of buffer at each node and transmission for different values of refresh rate of smoothing calculations (Sen, Zhang, Dey, 2002) (4).

The smoothing techniques used in the experiment are appropriate and applicable for all multicast and broadcast connections to reduce server and network load.

As the work done in (Sen, Zhang, Dey, 2002) (4).

The baseline dissemination algorithm transmitted unsmoothed video to all the clients at different nodes with identical buffer space. The baseline transmission required bandwidth of 1.74 MB/s for Blue Brothers and 682 MB/s for Space Jam.

By using differential caching smoothing techniques algorithm which is applicable for all nodes. The bandwidth delay jitter was reduced to between 30 - 50% by smoothing over unsmoothed transmission.

In case of fast Space Jam video transmission that required higher bandwidth, the bandwidth requirement was reduced significantly when only a few megabytes of buffer were allotted.

- An internet delay model-with both high mean low variance and low mean high variance- was applied.

The qualitative results can be summarized as: 30% to 50% reduction in delay jitter was noted in case of transmission of two different samples of

videos after differential caching smoothing algorithm when compared to unsmoothed algorithm transmission of same videos. That is, smoothing supports the stored video and live video streaming to great extent by reducing bandwidth requirement and improving delay. In case of applying smoothing algorithm, the requirement of maximum buffer usage decreases with decreasing refresh rate. Thus, smaller refresh rate supports live video streaming to greater extent. For refresh rates of 500 ms to 100 ms, maximum buffer usage decreases by 50% or more. Furthermore, with decrease in buffer usage, the delay jitter in transmission at receiving end also decreases. For a decrease of 4MB to 2MB in buffer value, delay jitter decreases with 50% or more. In case of TCP: with High value of mean low variance from 1000ms to 100 ms, the delay jitter drops to almost 40%.

- The Resulting Value of Delay Jitter

N.B.:

The acceptable value of delay = 1000 /25 = 40 ms, where eye detects 25 Fr\s

Where the O / p values in results of delay is lower than this value

EXPLANATION OF RESULTS

For The Effect of Smoothing With Variable Refresh Rate on Buffer Usage and Delay

By applying smoothing of videos in live video streaming, the delay jitter can be improved signifycantly because smoothing function superimposes itself with delay function. With a decrease in refresh rate, the buffer usage value also decreases due to small amount of units to be timely transmitted with the applied smoothing schedule. As the buffer usage value, decreases due to transmission of smaller amounts successively, the recipient becomes able to play the video with least delay jitter.

And for The Effect of Refresh Rate on Delay

In case of UDP, absence of ack. confirms that timing of refresh rate has no significant effect on delay. Yet, in case of TCP, smaller refresh rate ensures smaller delay in transmission.

CONCLUSION AND FUTURE WORK

The paper considers the problem of delivering streaming of various types of multimedia to variable multiple heterogeneous recipients on a diverse network of internet. The paper proposes an application level network for caching the video that helps to reduce bandwidth requirements and improves delay in transmission. The paper discusses a developed work-ahead smoothing algorithm with application level multicasting and temporal caching of VBR videos efficiently from server to several recipients with variable bandwidth and buffer capacities. The work provided a way to develop necessary and sufficient conditions to check the possibility of transmitting the video to all recipients without any overflow of available buffer space and to provide an algorithm to compute the set of optimal feasible transmission schedules for the transmission.

The paper presented an analytical, theoretical and algorithmic treatment for the problem of transmitting live video streaming on a heterogeneous platform like internet from a server to a client. The paper suggested that by smoothing the video, quality (delay) of transmission can be improved and bandwidth requirement can be decreased. With application of smaller buffer usage the delay also improves.

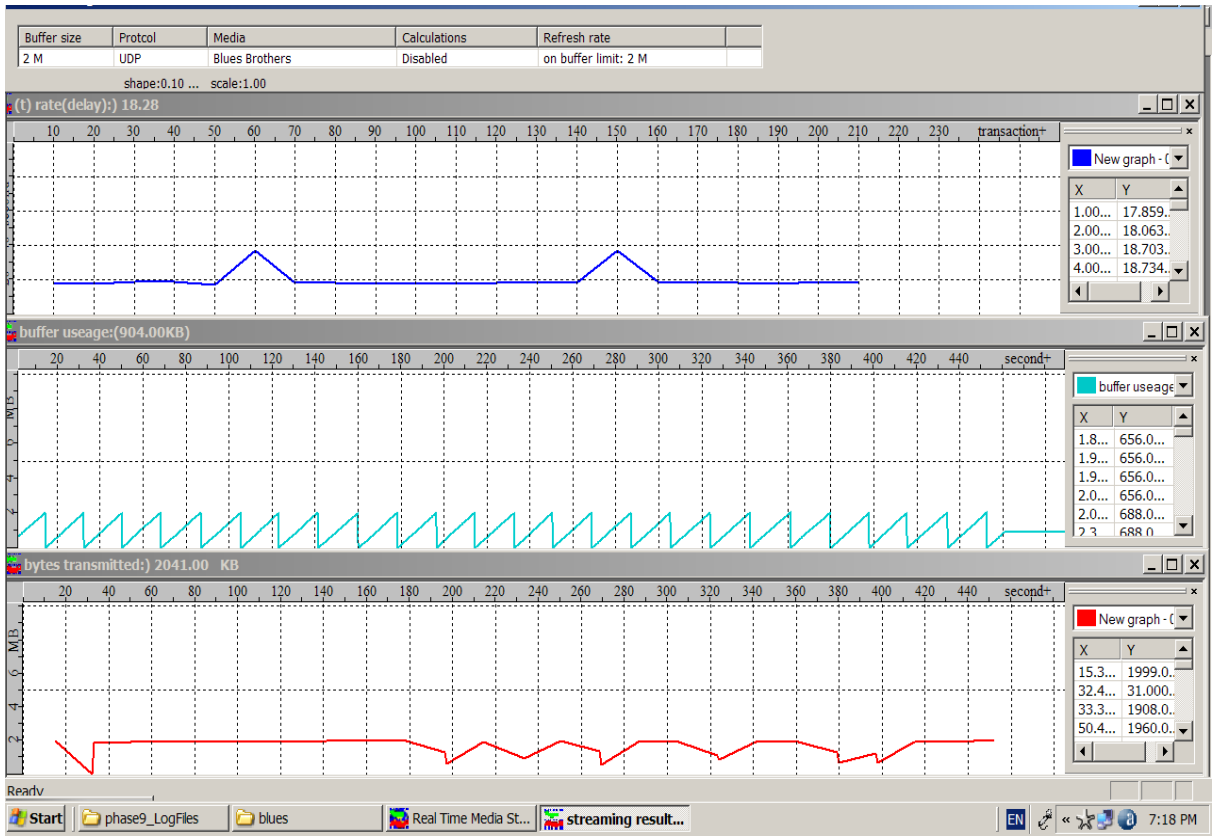
The paper proposed a solution for one to one server-client path which can be generalized to a multi cast case as stated in the literature (Sen, Zhang, Dey, 2002) (4).

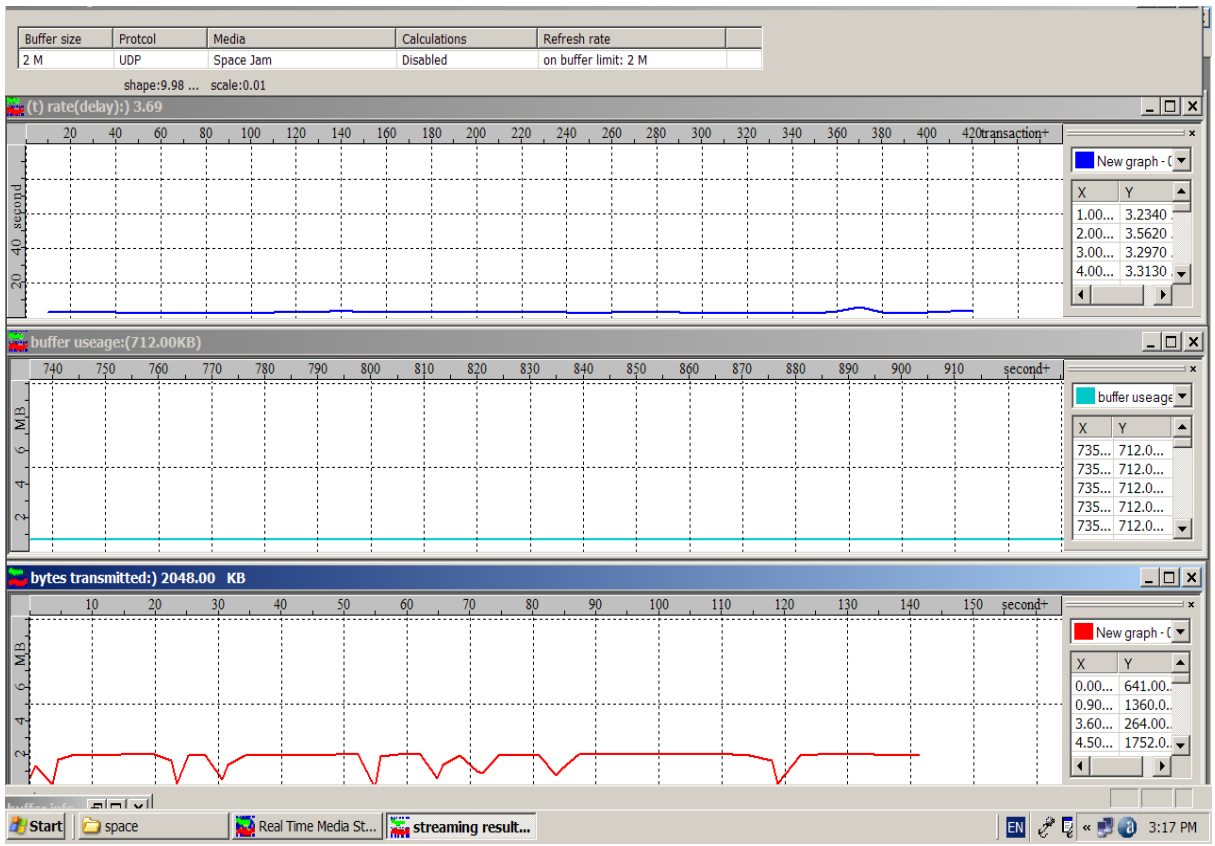
Further work is necessary for the actual implementation issues of the smoothing algorithm that includes the designing of efficient protocols to implement multicast smoothing process. The same set of smoothing process need to check for the case of multicasting of live videos where the smoothing nodes do not have any a priori knowledge of frame sizes and required bandwidth (Sen, Zhang, Dey, 2002) (4). Further research work is going on at various issues.

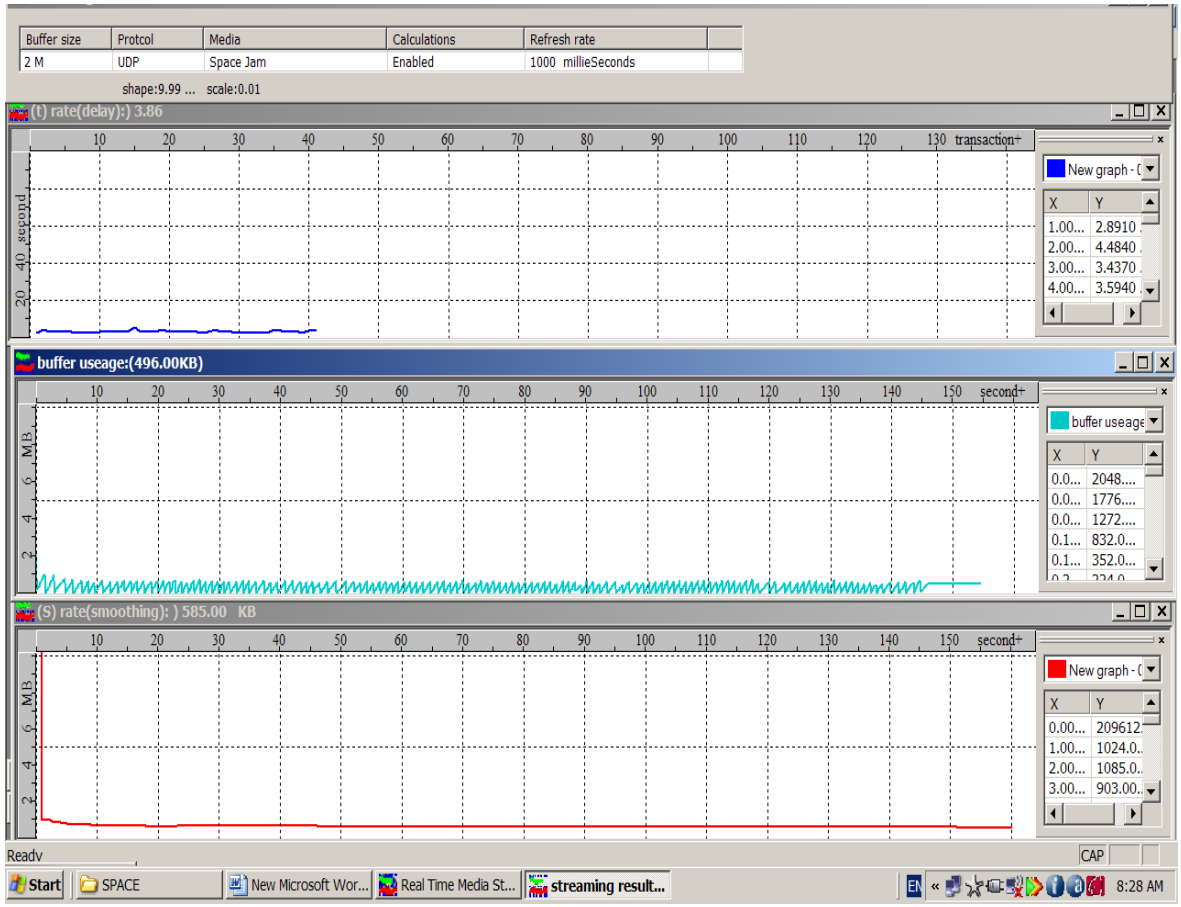
*** Comparison with Previous Work**

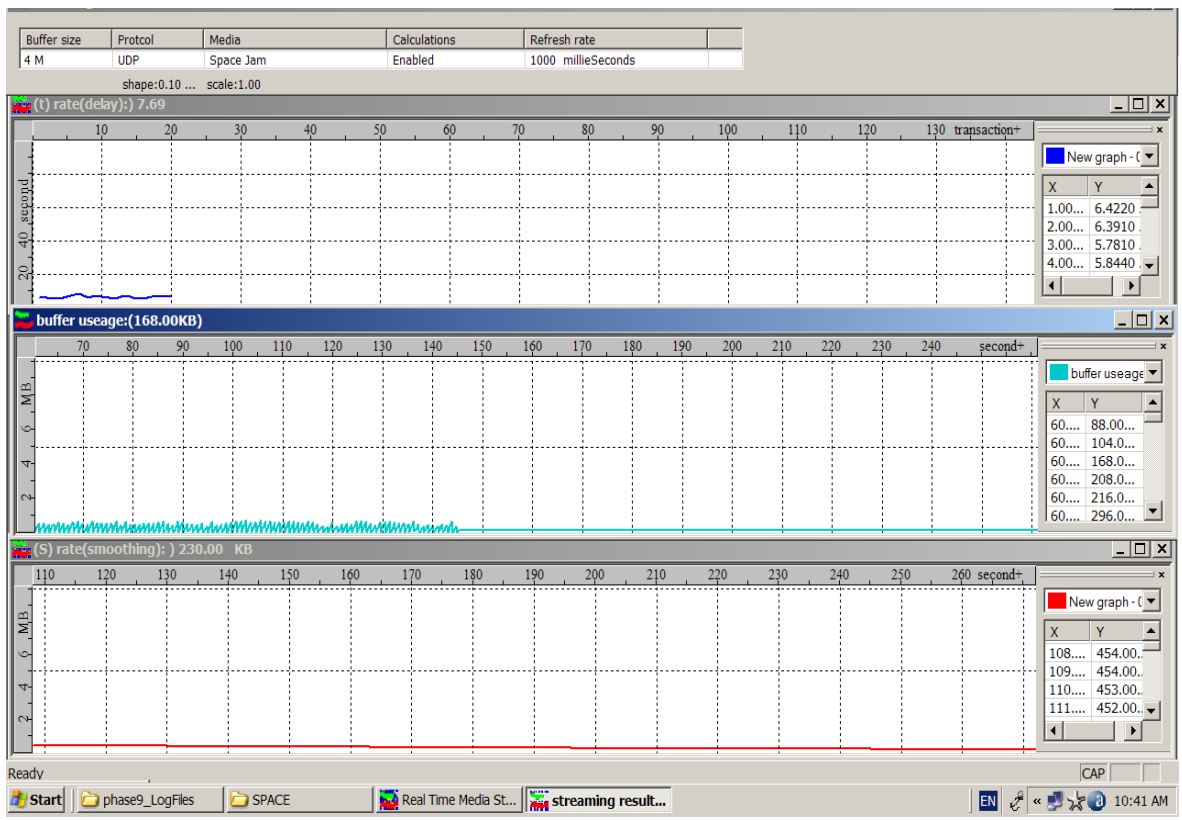
- In our work we control the bit rate which is better than controlling the frame rate as in the previous work (like: Sen, Dey, Towsley) (7).

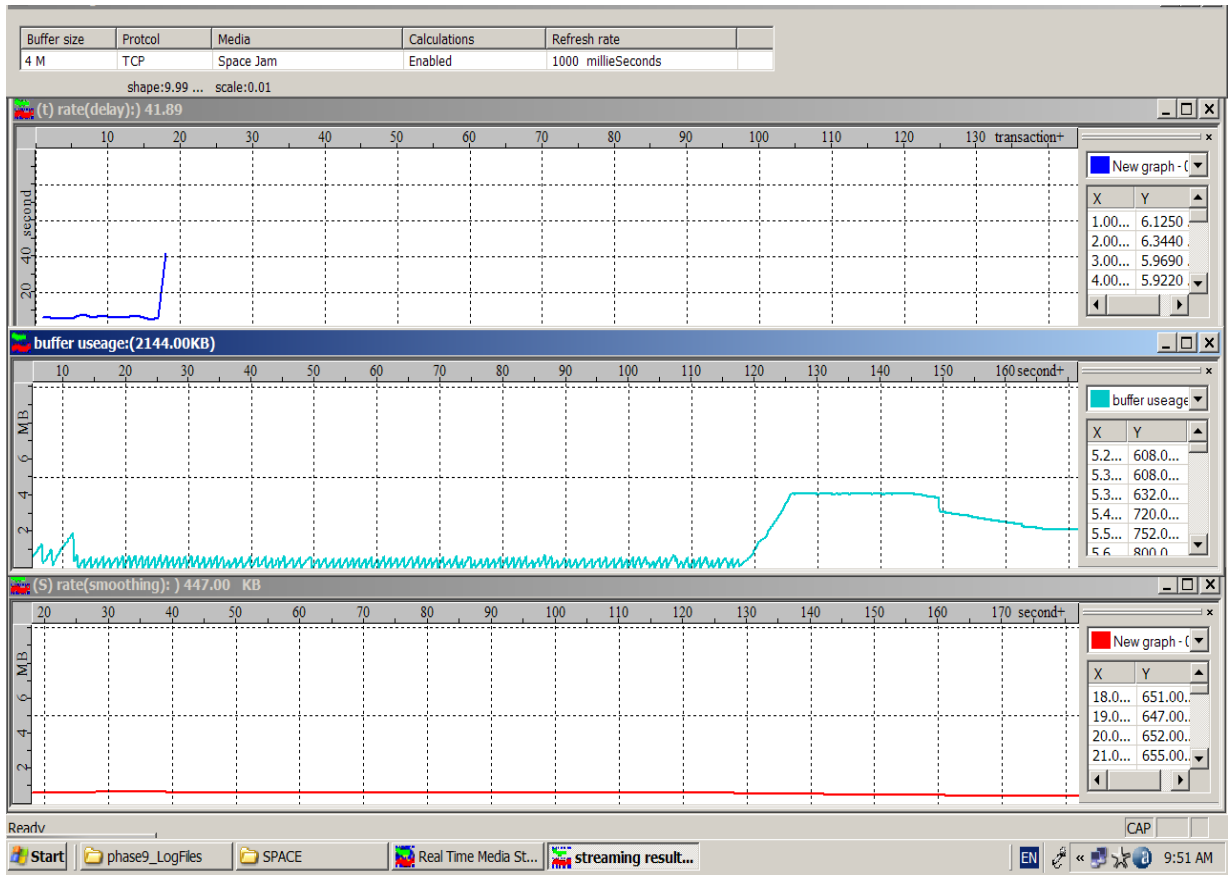
- No previous mention for combining delay with smoothing as in our work.

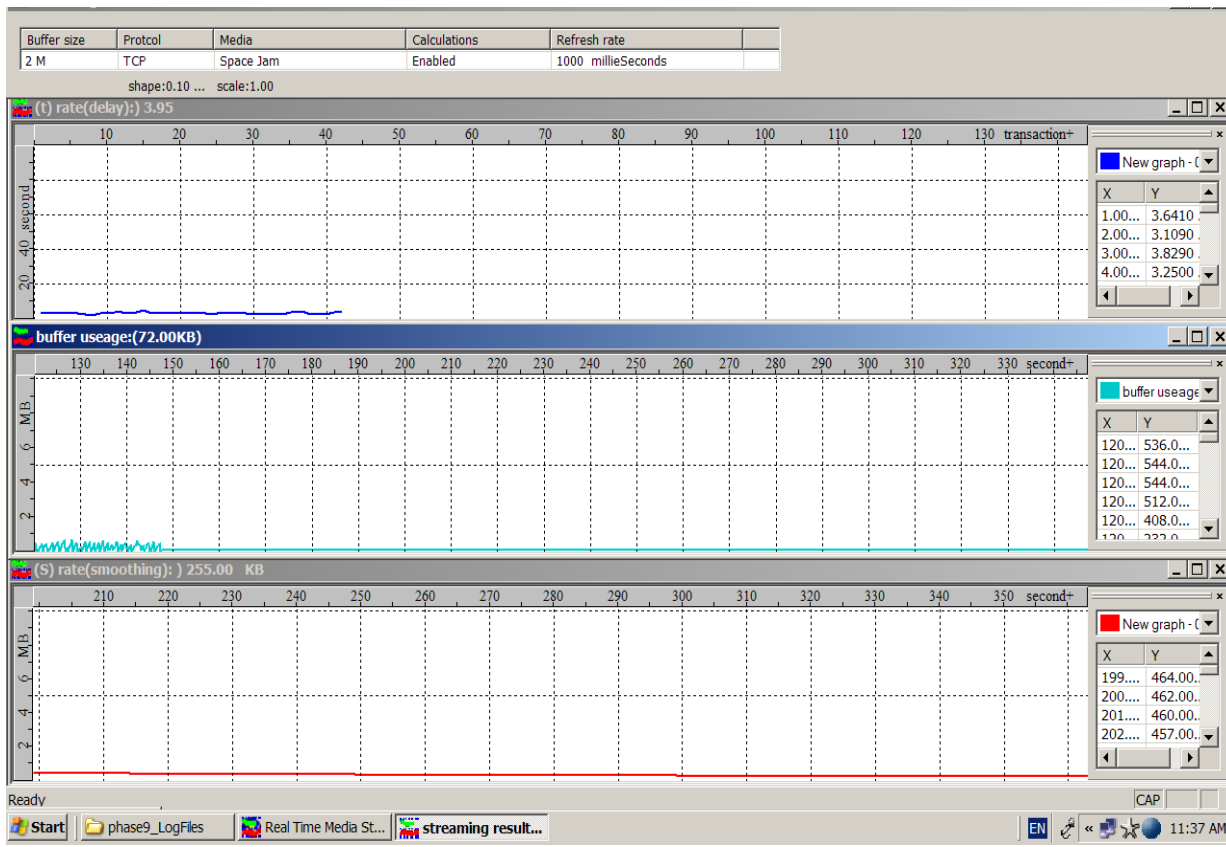

Buffer size | Protcol | Media | Calculations | Refresh rate
2 M | UDP | Blues Brothers | Disabled | on buffer limit: 2 M
shape:0.10 ... scale:1.00
(t) rate(delay):) 18.28
transaction+
buffer useage:(904.00KB)
second+
bytes transmitted:) 2041.00 KB
Ready
Start
phase9_LogFiles
blues
Real Time Media St...
streaming result...
7:18 PM


Buffer size | Protcol | Media | Calculations | Refresh rate
2 M | UDP | Space Jam | Disabled | on buffer limit: 2 M
shape:9.98 ... scale:0.01
(t) rate(delay):) 3.69
transaction+
buffer useage:(712.00KB)
second+
bytes transmitted:) 2048.00 KB
Start
space
Real Time Media St...
streaming result...
3:17 PM


Buffer size
Protcol
Media
Calculations
Refresh rate
2 M
UDP
Space Jam
Enabled
1000 millieSeconds
shape:9.99 ... scale:0.01
(t) rate(delay):) 3.86
transaction+
second
New graph - (
X
Y
1.00... 2.8910
2.00... 4.4840
3.00... 3.4370
4.00... 3.5940
buffer useage:(496.00KB)
second+
MB
buffer useage
0.0... 2048....
0.0... 1776....
0.0... 1272....
0.1... 832.0...
0.1... 352.0...
(S) rate(smoothing):) 585.00 KB
0.00... 209612
1.00... 1024.0..
2.00... 1085.0..
3.00... 903.00..
Ready
CAP
Start
SPACE
New Microsoft Wor...
Real Time Media St...
streaming result...
8:28 AM


Buffer size
Protcol
Media
Calculations
Refresh rate
4 M
UDP
Space Jam
Enabled
1000 millieSeconds
shape:0.10 ... scale:1.00
(t) rate(delay):) 7.69
transaction+
second
New graph - (
X
Y
1.00... 6.4220
2.00... 6.3910
3.00... 5.7810
4.00... 5.8440
buffer useage:(168.00KB)
second+
MB
buffer useage
60.... 88.00...
60.... 104.0...
60.... 168.0...
60.... 208.0...
60.... 216.0...
60.... 296.0...
(S) rate(smoothing):) 230.00 KB
108.... 454.00..
109.... 454.00..
110.... 453.00..
111.... 452.00..
Ready
CAP
Start
phase9_LogFiles
SPACE
Real Time Media St...
streaming result...
10:41 AM


Buffer size
Protcol
Media
Calculations
Refresh rate
4 M
TCP
Space Jam
Enabled
1000 millieSeconds
shape:9.99 ... scale:0.01
(t) rate(delay):) 41.89
transaction+
second
New graph - (
X
Y
1.00... 6.1250
2.00... 6.3440
3.00... 5.9690
4.00... 5.9220
buffer useage:(2144.00KB)
second+
MB
buffer useage
5.2... 608.0...
5.3... 608.0...
5.3... 632.0...
5.4... 720.0...
5.5... 752.0...
(S) rate(smoothing):) 447.00 KB
second+
MB
New graph - (
18.0... 651.00..
19.0... 647.00..
20.0... 652.00..
21.0... 655.00..
Ready
CAP
Start
phase9_LogFiles
SPACE
Real Time Media St...
streaming result...
EN
9:51 AM


Buffer size
Protcol
Media
Calculations
Refresh rate
2 M
TCP
Space Jam
Enabled
1000 millieSeconds
shape:0.10 ... scale:1.00
(t) rate(delay):) 3.95
transaction+
second
New graph - (
X
Y
1.00... 3.6410
2.00... 3.1090
3.00... 3.8290
4.00... 3.2500
buffer useage:(72.00KB)
second+
MB
buffer useage
120... 536.0...
120... 544.0...
120... 544.0...
120... 512.0...
120... 408.0...
(S) rate(smoothing):) 255.00 KB
second+
MB
New graph - (
199.... 464.00..
200.... 462.00..
201.... 460.00..
202.... 457.00..
Ready
CAP
Start
phase9_LogFiles
Real Time Media St...
streaming result...
EN
11:37 AM

APPENDICES

Project: Real-Time media streaming engine

Architecture: Client/Server software solution

Operation System: Windows XP

Platform: Microsoft Direct X

Media Format: MPEG 2 (Transport stream)

Communication:

Purpose	Area	Communication Type
Media transmitting	Client/Server	Selectable Sockets: a)TCP b) UDP
Monitoring and Measurements	Client/Server	Sockets: TCP
	Inter-process	Shared memory

Implementation

Implementation is divided into two categories of components:

I- Media streaming engine and players:

a) Microsoft Direct Show Media Player (On the server machine)

b) Microsoft Direct Show Filter (Stream sender filter within the sever player)

c) Microsoft Direct Show Filter (Stream receiver filter on the client machine)

II- Monitoring, Measurement, and Control:

d) Streaming controller and initiator (On the client machine)

e) Streaming monitoring for networking and memory management (Server machine)

f) Streaming manager using special calculations advisory (Server machine)

Implementation Details (Component Architecture):

a) Microsoft Direct Show Media Player (On the Server Machine)

Type: Process

Machine: Server

Architecture: Multithreaded

Communicate with: The streaming controller and initiator on the client machine, component (d).

Sockets: TCP Socket.

b) Microsoft Direct Show Filter (Stream Sender Filter within the Sever player)

Type: COM

Machine: Server

Architecture: Multithreaded

Communicate with:

1) The stream receiver filter on the client machine, component (c)

2) The streaming monitoring for networking and memory management on server machine, component (e)

3) The streaming manager using special calculations advisory on the Server machines, component (f)

Sockets: TCP and/or UDP

Inter-Process Communication: Shared Memory

Data structures: Queue Media Buffer

c) Microsoft Direct Show Filter (stream receiver filter on the client machine)

Type: COM

Machine: Client

Architecture: Multithreaded

Communicate with:

1) The stream sender filter on the server machine, component (b)

2) The streaming manager using special calculations advisory on the Client and the Server machines, component (f)

Sockets: TCP and/or UDP

Inter-Process Communication: Shared Memory

Data structures: Queue Media Buffer

d) Streaming controller and initiator (on the client machine)

Type: Process

Machine: Client

Architecture: Single Threaded

Communicate with:

1) The Media Player on the server machine, component (a)

2) The stream receiver filter on the client machine, component (c)

Sockets: TCP

e) Streaming Monitoring for Networking and Memory Management (Server Machine)

Type: Process

Machine: Server

Architecture: Multithreaded

Communicate with:

1) The stream sender filter on the server machine, component (b)

2) The stream receiver filter on the client machine, component (c)

Sockets: TCP

Inter-Process Communication: Shared memory

f) Streaming Manager Using Special Calculations Advisory (Server Machine)

Type: Process

Machine: Server

Architecture: Multithreaded

Communicate with:

1) The stream sender filter on the server machine, component (b)

2) The stream receiver filter on the client machine, component (c)

Sockets: TCP

Inter-Process Communication: Shared memory

MULTITASKING BASICS

There are two ways to implement multitasking: as a single process with multiple threads or as multiple processes, each with one or more threads. An application can put each thread that requires a private address space and private resources into its own process, to protect it from the activities of other process threads.

A multithreaded process can manage mutually exclusive tasks with threads, such as providing a user interface and performing background calculations.

Creating a multithreaded process can also be a convenient way to structure a program that performs several similar or identical tasks concurrently.

A thread is basically a path of execution through a program. It is also the smallest unit of execution that Win32 schedules. A thread consists of a stack, the state of the CPU registers, and an entry in the execution list of the system scheduler.

Each thread shares all of the process's resources.

A process consists of one or more threads and the code, data, and other resources of a program in memory. Typical program resources are open files, semaphores, and dynamically allocated memory. A program executes when the system scheduler gives one of its threads execution control. The scheduler determines which threads should run and when they should run. Threads of lower priority may have to wait while higher priority threads complete their tasks. On multiprocessor machines, the scheduler can move individual threads to different processors to "balance" the CPU load.

Each thread in a process operates independently.

Unless you make them visible to each other, the threads execute individually and are unaware of the other threads in a process. Threads sharing common resources, however, must coordinate their work by using a Mutex or another method of inter-process communication.

The Multitasking was fulfilled via:

- Microsoft Developer Network (MSDN)Documentation
- Platform SDK documentation
- Visual C++ documentation

REFERENCES

1- Boster, F., Meyer, G; Roberto, A; Inge, C. 2002. A Report on the Effect of the United streaming Application on Educational Performance. United Learning.

2- Horn, Royal Van. 2001. Technology - Streaming Video and Rich Media. Phi Delta Kappa, Gale Group.

3- Ishibashi, Yutaka; Tasaka, Shuji; Inoue, Takeshi. 2000. Joint Synchronization between Live and Stored Media in Network Environment, Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Nagoya.

4- Sen, Subhabrata; Towsley, Don; Zhang, Zhi-Li; Dey, Jayanta K. 2002. "Optimal Multicast Smoothing of Streaming Video on the Internet", IEEE, Vol. 20, pp: 17-21.

5- Walter, F. 2003. "The Technology Teacher". International Technology Education Association, Vol. 62, No. 8, pp: 7-8.

6- A Corlett, D.I. Pull in and S. Sargood, "Statistics of One Way Internet Packet Delays, "53rd IETF, Minneapolis, March 2002.

7- Sen, Dey, Towsley."Online Smoothing of Live, Variable -Bit - Rate Video."