

**International Journal of Innovative Research in Science,
Engineering and Technology**

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

Dynamic Allocation of Circular Buffer with Circular Linked List Technique for Real Time Multimedia Application across Slow Speed Links

R.Jayavadivel¹, Dr.J.Sundararajan², G.Arulselvam³

Assistant Professor, Department of IT, Paavai Engineering College, Pachal, Namakkal, Tamilnadu, India.¹

Principal, Paavai College of Technology, Pachal, Namakkal, Tamilnadu, India.²

B.Tech, Department of IT, Paavai Engineering College, Pachal, Namakkal, Tamilnadu, India.³

ABSTRACT: Today Multimedia and its applications play a major role of importance in Internet, at the same time it also have an equivalent drawback parameters in accessing and using them. In this paper, the common problem in Multimedia application has been discussed by providing a common solution for that. This paper is investigating a new concept called "a Circular buffer with circular linked list concept with dynamic allocation method was proposed in adjusting the video frame rate into a TCP-circular buffer .It also developed in aim to eradicate the Buffer and Packet Loss drawback parameters across the slow speed links .The simulation program was written in Matlab v1.4 to monitor the Buffer fullness and circular Liked list links the buffer towards each other virtually where the H.264 encoder is allowed to trace and test the data. In this arrangement, the buffer occupancy was made above 95% usage and maintains 20% of packet loss avoiding techniques.

KEYWORDS: Circular Linked List, Packet Loss, Buffer Occupancy.

I. INTRODUCTION

Over the year, the internet growth and usage became wider in ranges across the globe. However the ranges of applications such as multimedia, documents, etc., get tremendous growth in extend. as it can be observed that traditional applications like bulk file transfer, static websites are no longer sufficient for today's internet application usages[1].hence, we need to overcome the higher bandwidth allocation and wide spectrum are need in using such applications efficiently. Today most of the applications are based on real-time delivery of data such as video conferencing, internet-telephony and video & audio streaming are in usages. Video conferencing enables the continuous delivery and play back of video which overcomes the drawback of downloading and uploading the files in internet. The United States Federation Communication defined the Slow-Speed links as data transmission speed of almost 4Mbps and download stream as 1Mbps.the new application gives rise to the other drawbacks such as packet loss, delivery delay, buffer Management, transmission rates, etc., several problems to be solved before the transmission of video using data streaming activities. Circular buffer linked list increases higher data transmission rate with real-time implementation. It also overcomes the packet loss drawbacks too.

The video get into several processes before it introduced into the network system. Ithas to do a plenty of mechanisms in transferring the data from source to destination along a network path. In using TCP-IP buffer on figuration management, the data to be transmitted get compressed in reducing its size and encoded (encrypted) format for security and memory size capability too. Theactual encoded data the passes into the buffer for temporary management and storage of data where to avoid the network traffic (congestion) level to higher extends. Butmeanwhile there are many drawbacks available in establishing the data to the network leads to video buffering are video streaming mechanism to avoid such drawbacks in an efficient manner the Dynamic Circular buffer get allocated in maintain high speed and minimum buffering problem to their grater extends in this method ,we are using Circular linked list concept and

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

dynamic double buffer mechanism in avoiding packet loss and videostreaming or slow speed links in real-time multimedia applications

II.RELATED WORKS

Real-time Multimedia application is an active research area studied an efficient delay distribution scheme that allocates the end to end delay bound into the end station shaper and the networks. Ioan and Niculescu proposed two mathematical model of an algorithm named leaky bucket mechanism used for the parameter usage of connection control in multimedia networks they reviewed the effect of switch delay and shaping on network performance based on the empirical envelope traffic model, and analyzed the impact of shaping via computer simulation offered a system design and implementation that ameliorates some of the important problems with video streaming over the internet.

The scheme adapts the coding rate of the video streaming to the variations available bandwidth connections, while smoothing the changes in the image quality between consecutive video scenes. Bajic et al., addressed the issue of robust and efficient scalable video communication by integrating and end-to-end buffer management congestion control at the source with the frame rate play out adjustment mechanism at the receiver. Lam et al., addressed the challenge of the lack of end-to-end quality of service supporting the current internet which caused significant difficulties to ensuring playback continuity in video streaming application by investigating the new adaptation algorithm to adjust the bit rate of video data in response with the available to improve playback continuity. ATM networks promises in providing high speed real-time multi-point to central video transmission for telemedicine application in rural hospitals and in emergency medical services. The ramifications of this technology are clear, speed, reliability, and increased capacity in video transmission for telemedicine. It also leads to studied the buffering schemes for the VBR video streaming in heterogeneous networks.

The proposed adaptive rate control algorithm provide significantly improved streaming quality compared to non-control policy. Xiong et al., again pointed out the requirement for transmitting the real time video with acceptable paying performance via TCP and presented the stochastic prediction model which can predict the sending delays of video frames.

However these available frame rate adaptation techniques are not suitable for slow speed network environment. It is therefore imperative to develop a new technology that will meet this challenge. Dynamic Circular buffer with linked list concept is thus presented.

III. THE PROPOSED SCHEME (DCBLL) DESCRIPTION

A. Video Transmission over the Internet

As described in the introduction part, the video transmitted over the internet contains several mechanisms to follow. First, the video to be transmitted get compressed as raw video using encoding (CODER-DECODER) such as H.263, H.264, etc in reducing its size the compressed and encoded (encrypted) format for security and memory size capability too at the sender side. The actual encoded, compressed video data then passes into the sender buffer within application layer and then send to TCP sender buffer for temporary management and storage of data where to avoid the network traffic (congestion) level to higher extend. At that level they also get fragmented into TCP data packets. From the TCP sender buffer the fragmented data packets moved across the network. the data packets are later received by the TCP receiver buffer wherein they get defragmented and decompressed. The decompressed video frames are then sent into the receiver play out-buffer. figure.1 shows the actual step by step procedure in transmission of video over the network from source to destination places.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

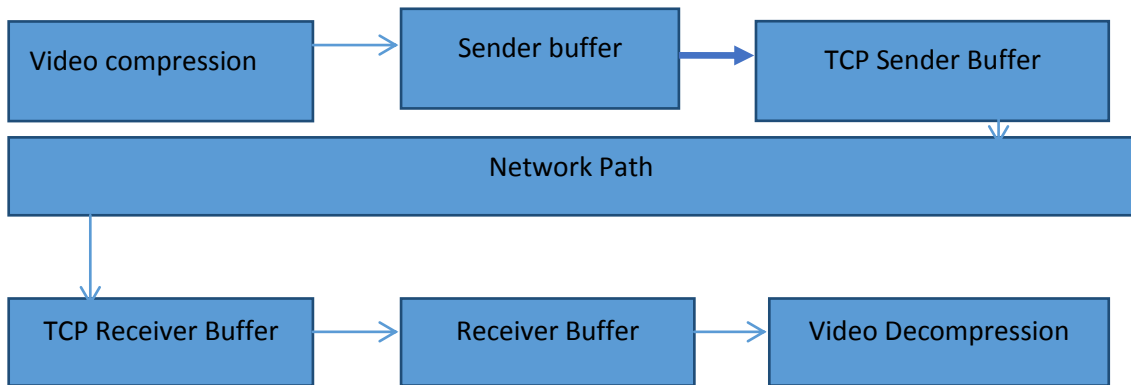


Figure.1 video transmission over the internet.

B. The Dynamic Circular Buffer with Linked List (DCBLL)

Because of the total release of the data packets from the sender buffer leads to traffic congestion in the network. Hence to avoid such problem dynamic circular buffer with linked list concept is being proposed. A network of buffers is a collection of service centers which represents the server of the buffer and customer which represents video frames being transmitted.[2].As shown in the Figure.2 the compressed video get transmitted to segmentation block where sub-segmentation block presides into it.

At this stage the actual raw video get segmented before introducing into the TCP-sender buffer. it minimizes the work done by the TCP sender buffer in converting frames to data packets easier from segmented frames.Next,the segmented frames gradually moves to sender buffer block and sequentially to TCP sender buffer.

Now the DCBLL get raised in collecting the data packets from TCP sender buffer to First queue buffer and allocates dynamic storage to store these data packets within them. It also report to the Circular Buffer and Second queue buffer about its dynamic allocated value for the packets and insist them to allocate the same value for storage.at the same time the circular buffer starts receiving the data packets from first queue buffer to it and get circularly linked towards each data packets with one another. It avoids the packet loss as it get connected towards themselves by Circular Linked List concept implementation. Finally the Data from circular linked list transfer its data Second queue buffer for gradual transmission of data packets over the network path. Here, the Second Queue buffer acts as a carrier sensing component in sensing the network traffic and transmitting its data packets frequently. At the Receiver point it follows a Reversible process of de-segmentation, and decompression techniques in getting their original data frames.

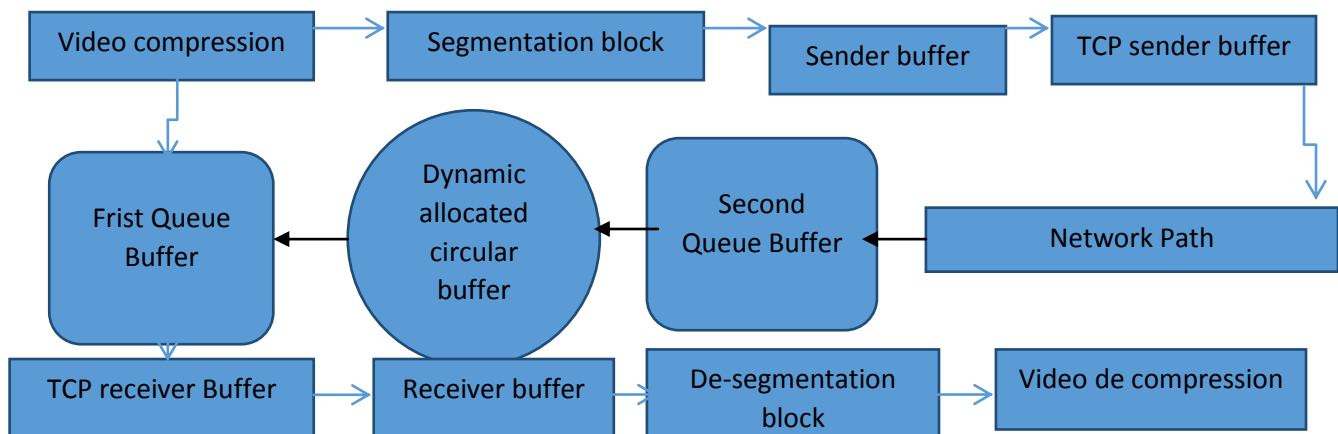


Figure.2 the Dynamic Circular Buffer with Linked List (DCBLL)

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

C. Algorithm

The Following Algorithm helps in implementing the DBCLL concept along the Video Streaming Slow Speed drawback rectifications. It follows the gradual sequential steps in its working process with some calculation part included additionally.

1. Compress the video by (CODEC) encoding schemes using H.263,H.264,etc.

2. Segment the raw video into segmented video frames (f) by following the condition as

```
If(f>=(realInteger>certaintlimit))  
{  
Sub-segment (f);  
}
```

The above statement shows that, the segmentation block checks the video frames is not equal to null and below the certain limit conditions. If condition get satisfied, then sub-segment the raw video frames from the compressed zone.

3. Transmit the segmented frames into the Sender buffer and the to TCP sender buffer, the TCP sender buffer converts the actual segmented frames into data packets of various sizes and gradually moves to First Queue Buffer Block.

4. Allocate the Queue buffer in dynamic pattern where the actual size of the data packets from the TCP sender buffer cannot be predicted are determined already .hence the dynamic allocation of First Queue Buffer maintains as in the sequence as follows:

Segment (f) = Buffer Space Allocated.

It indicates that the actual data packet fragmented from segmentation block and TCP sender buffer is equal to the buffer size going to be allocated, hence there should be a message passing to be done before sending data frames and packets from Segmentation block and TCP sender buffer block to the First queue buffer. The data is passed in First in First Out (FIFO) concepts.

5. Similarly, next it transmit to Circular buffer where the data packets to be filled dynamically, it also follows a general rule that,

$N(\text{space}) = \text{segment}(f) = \text{queue buffer } 1;$

Then,

$N(\text{space}) = N(\text{space}) + 1;$

It allocate dynamically the data packets into it, and then the circular linked list get connected towards each other from one data towards another sequentially. it is equationally defined as

$[N[\text{space}] \rightarrow N[\text{space}] + 1] + [N[\text{space}] + 1 \rightarrow N[\text{space}] + 2] + \dots + [N[\text{space}] + n - 1 \rightarrow N[\text{space}] + n]$.

Where $N[\text{space}]$ is the actual single space in the Circular buffer. This equation forms a dynamic allocation of space and circularly linked data packets within the circular buffer.

6. Repeat the step 4 once again in filling the Second queue buffer.

It also performs some special operation than the other buffers, that, itsense the network path for traffic congestion state to transmit the data in un-collision environment. Hence it follows the rules such as ;

```
If (network traffic=="true")  
{  
Wait (t seconds);  
}
```

Else

Transmit (D);

Where D is the data Packets.

7. Finally data packets send via the network path to receive at the receiver side to watch the video frames sequentially.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

D. Special Cases

The data packets transmitted over a network from TCP sender buffer to First queue buffer, Circular queue buffer, Second queue buffer will follow a certain threshold value limits across the network path in avoiding collision with other data packets being transmitted in such considerations, there are three major cases to be analyzed for the probability of error or collision occurrences within the network path during the transactions. Let's assume the dynamic allocation of data packets have threshold limitations of 10 frames per second will be allocated as a maximum limited value in the buffer for its fullness status depends upon the status of the buffer and threshold value which we assume, three different cases can be derived from the above methods as follows:

Threshold value=10 frames/sec, then

Case:1

Whenever the frame limit (ie, threshold value exceeds) $>10/\text{sec}$

In such situations there is no proper way to maintain and manage the data which will burst of the entire buffer occupancy ranges. Hence control the buffer allocation mechanism and discards the frames which are overcomes the buffer space in FIFO queue basis order.

The Discarded data packets or frames will be waiting for the prior packets to complete their execution status gradually.

Case:2

Whenever the frame limit (ie, threshold value exceeds) $<10/\text{sec}$

In this situations, the data lacking will be happens and the time of buffering get increased in its ranges. to avoid such circumstances two way of solution will be provided

1. Push the data packets into the circular buffer until the buffer get full or it meets the threshold value in normal stage.
2. Wait until the data get filled within the buffer space to give proper sequential output display without buffering the data packets.
3. The rotational time latency will be introduced in this occasion to speed up the data packets as compared to linear dynamic buffer allocation arrangements.

Case:3

Whenever the frame limit (ie, threshold value exceeds) $=10/\text{sec}$ It is the normal measurement where the proper process get proceeded within the network transactions.

Case Algorithm

The above cases have its own algorithm in execution as follows:

Case:1

```
if ( threshold value > 10 frames/sec )
```

```
{  
Initialize the buffer value =10;  
Scan whether buffer get filled  
If(again threshold value>10)  
{  
Discards (overflow frames);  
Reundo(discards)- -;  
}}  
}
```

Case:2

```
if ( threshold value < 10 frames/sec )
```

```
{  
Initialize the buffer value =10;  
Scan whether buffer get filled  
If(again threshold value<10)  
{  
push (underflow frames);  
frames++;  
}  
}
```

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

```

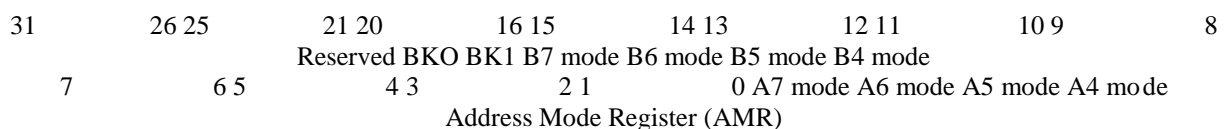
}
Case:3
if ( threshold value == 10 frames/sec )
{
Initialize the buffer value =10;
Scan whether buffer get filled
If(again threshold value= =10)
{
Send frames();
Get new frames();
}}

```

E. Calculations

Whenever the data or an object which across the circular path increases its speed in higher speed, since the Circular buffer is a concept proposed in speed up the data transmission which minimizes the packet delay drawbacks. The average speed of the data is considered to be 10nsps, then when they cross over the First queue buffers it doesn't have any change in its speed, because it is in linear stretch. But, when it raised into the circular buffer, it slows down the speed initially to complete its rotation and linked list operation in the first semi-circle, the another semicircle to complete the rotation is responsible for achieving the greater speed as compared to the linear buffers. Because the rotational latency of an circular object get its speed increase in ranges. since the same speed maintenance ,it travels through the second queue buffer and along the network path to achieve the packet delay problem rectifications. In the calculation part, the average speed of the linear buffer and the rotational speed of the circular buffer are to be calculated to find the original speed of the data passes through the internet path. Since the actual data/packet speed in the linear buffer is calculated as follows: Circular addressing uses pointer manipulation to add the new samples to the buffer by overwriting the oldest available samples hence reusing the memory buffer. When the pointer reaches the last location of the delay line it needs to wrap back to the beginning of the line. This would normally involve some amount of software overhead. When circular addressing is used, the pointer automatically wraps back to the top whenever the bottom of the buffer is reached. As a consequence, the memory locations appear to be tied together in a circular manner hence the name 'circular buffer'. Most digital signal processors implement circular buffering in hardware in order to conserve memory and minimize software overhead.

On the C6000 processor data values in memory are accessed by setting up a register as a pointer to memory and then loading values using indirect or pointer addressing. The pointers to memory locations can be manipulated in linear or circular mode. Of the 32 registers available (A0–A15 and B0–B15), circular addressing can be implemented using any of eight registers (A4–A7, B4–B7) as a pointer. A 32 bit address mode register (AMR), is used to indicate whether registers A4–A7 and B4–B7 are enabled in linear or circular mode.



The block size fields (BK0 and BK1) in AMR can be used in conjunction with any of the registers to specify the length of the buffer as shown in Table 1. A value N in the register BK0/BK1 corresponds to a block size of 2(N+1) bytes. With the 5 bit field in BK0 and BK1 it is possible to specify 32 different block lengths for circular addressing from 2 bytes to 4G bytes (which incidentally is also the maximum address reach using a 32 bit pointer register). This makes it convenient to implement filters with very large number of taps. The circular buffer is always aligned on the block size byte boundary. This is necessary for the hardware to correctly locate the start and end of the circular buffer. For instance, if the register A4 contains an address 0x80000005 and the block size is 64 bytes then the start address of the defined circular buffer is strapped to 0x80000000 and the end address is 0x8000003F.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

**Table 1. AMR Mode Field Encoding
Mode Addressing Option**

00	Linear Mode
01	Circular Mode using BK0 size
10	Circular Mode using BK1 size
11	Reserved

To understand the circular addressing, let us consider a case where we require a pointer A4 to be set up in circular mode to point to a buffer of length 16 bytes. Firstly, the pointer A4 needs to be setup in circular mode by initializing bit field [1:0] of register AMR with '01' (if using BK0) or '10' (if using BK1). The field BK0 set to 00011 resulting in a circular buffer size of 23+1 or 16 bytes. So finally, AMR register will have bit field [1:0] set to 01 and field [25:21] set to 00011. Let us assume that pointer A4 contains the address 0x8000000E. Now if the pointer A4 is post incremented by using an instruction such as LDH ; *A4++, A8

This instruction loads the contents of 0x8000000E into A8 and then increments the pointer A4 by 2 bytes, then the pointer will end up at location 0x80000000 due to circular addressing. Circular addressing hardware automatically defines address 0x80000000 as the top of the buffer and 0x8000000F as the end of the 16 byte long buffer as shown in Figure 4. In fact, if the pointer A4 were to point to any shaded location in the buffer shown in Figure 4, the start and end addresses of the circular buffer would be the same for the given definition of circular buffer.

The buffer is defined to be greater than the block size (number of filter outputs) and to be a power-of-2 (2N). To understand this we consider a simple case with a block of 6 input samples (16bits each) and a 4 tap (16bits) FIR filter. The data buffer needs to be large enough to contain the samples and needs to be a power-of-2. Let us assume the buffer to be 16 half words (25 or 32bytes long). Figure 5 illustrates the state of the data samples in the buffer on 4 consecutive calls to the filtering algorithm. The first time the function is called after 6 new samples are added to the buffer. To calculate the first set of outputs y[0]-y[5],

$$y_0 = a_0x_0 + a_1x_{-1} + a_2x_{-2} + a_3x_{-3}$$

$$y_1 = a_0x_1 + a_1x_0 + a_2x_{-1} + a_3x_{-2}$$

....

$$y_5 = a_0x_5 + a_1x_4 + a_2x_3 + a_3x_2$$

We require 9 data samples x[-3]-x[5] from the buffer as seen in equation (2). Thus, to calculate each output we require the current sample as well (N-1) previous samples where N is the number of taps in the filter. For the first output y₀, the current sample is x[0] while the old samples are x[-3],x[-2] and x[-1]. The calling routine uses index to indicate the first location that needs to be accessed for the current iteration through the filtering loop. Every time the function is called the calling algorithm passes not only the pointers to the data samples but also the index into the buffer as well. Hence, the first time the algorithm is called the index into the data sample buffer is 0 or x[-3]. After the outputs are computed the routine returns to the main calling program. Before the filtering routine is invoked next, a new set (block size=6) of samples is added to the buffer. Hence, the second block of data x[6]-x[11] is added to the delay line and then the function is called a second time. To calculate the outputs y[6]-y[11] we require the data samples x[3]-x[11] which would imply that the index into the delay line. The index can be easily calculated as Index=[Previous_IndexBlock_Size]% Buffer_Size. In our example during the second call to the algorithm, the previous index was 0, block size is 6 and buffer size is 16 hence the index is 6 (pointing to location x[3]).

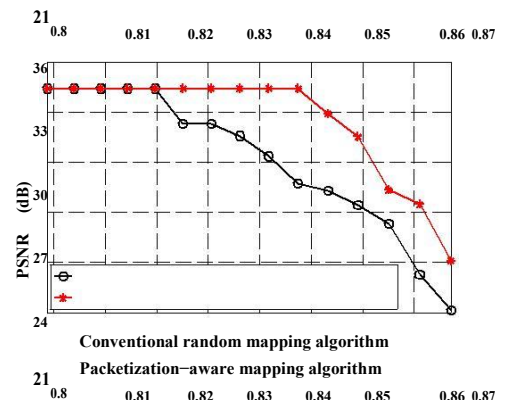
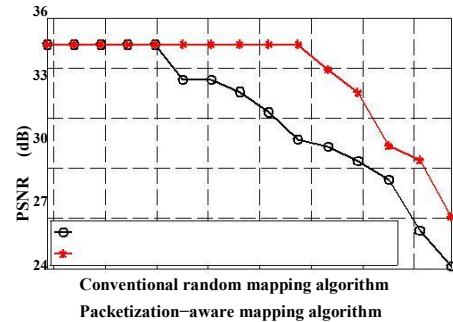
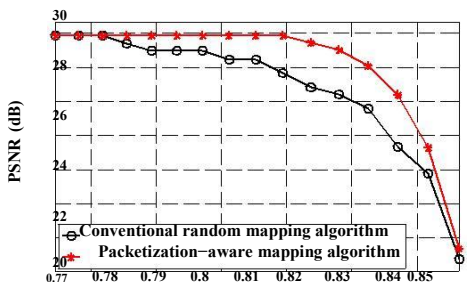
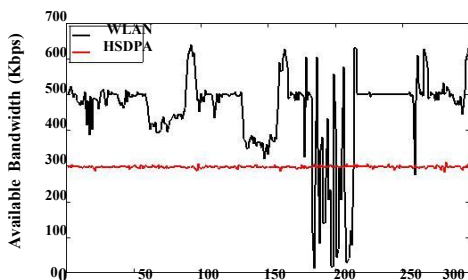
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

VI. EXPERIMENTAL RESULTS

	Throttle	Delay	Drop ($P_{(p+q)}$)
Path 1	300 Kbps	20 msec	0.01
Path 2	150 Kbps	50 msec	0.05
Path 3	100 Kbps	200 msec	0.75



V. CONCLUSION

In this paper, we have presented a video streaming slow speed link problem solution which aimed to manage uncover basic video streaming drawback parameters such as buffering, packet loss and segmentation issues. Our solution enables the use of deduction of buffering the video at high ranges and packet loss avoidance in larger extend. The dynamically allocated circular buffer provides better buffer management services and reduces the network layers working by segmentation and packet loss avoidance which ensures the video streaming reduction in higher extends. Beyondthat, the average speed of the data or packet transaction and video buffering reduction get reduced by achieving 100 frames per second. The video type of any format can be encoded frequently by H.264 encoder techniques is also a greater advantage in this solution.

REFERENCES

- [1] C. Eklund, R. Marks, K. Stanwood, and S. Wang, "IEEE standard 802.16: a technical overview of the WirelessMAN air interface for broadband wireless access," IEEE Commun. Mag., vol. 40, no. 6, pp. 98–107, 2002.
- [2] A. Bria, F. Gessler, O. Queseth, R. Stridh, M. Unbehaun, J. Wu, J. Zander, and M. Flament, "4th-generation wireless infrastructures: scenarios and research challenges," IEEE Personal Commun., vol. 8, no. 6, pp. 25–31, 2001.
- [3] D. MacKay, "Fountain codes," IEE Communications, vol. 152, no. 6, pp. 1062–1068, 2005.
- [4] M. Luby, "LT codes," IEEE Symposium on Foundations of Computer Science, pp. 271–280, 2002.
- [5] A. Shokrollahi, "Raptor codes," IEEE Trans. Inf. Theory, vol. 52, no. 6, pp. 2551–2567, 2006.
- [6] P. Maymounkov, "Online codes," Research Report TR2002-833, New York University, 2002
- [7] Q. Zhang, C. Guo, Z. Guo, and W. Zhu, "Efficient mobility management for vertical handoff between WWAN and WLAN," IEEE Commun.Mag., vol. 41, no. 11, pp. 102–108, 2003.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Special Issue 6, May 2015

- [8] J. Mitola III, "Software radios: Survey, critical evaluation and future directions," IEEE Aerosp. Electron.Syst. Mag., vol. 8, no. 4, pp. 25–36, 2002.
- [9] D. Jurca and P. Frossard, "Media-specific rate allocation in heteroge- nous wireless networks," J. Zhejiang University-Science A, vol. 7, no. 5, pp. 713–726, 2006.
- [10] Y. Pan, M. Lee, J. Kim, and T. Suda, "An end-to-end multipath smooth handoff scheme for stream media," IEEE J. Sel. Areas Commun., vol. 22, no. 4, pp. 653–663, 2004.
- [11] E. Altman and A. Jean-Marie, "The distribution of delays of dispersed messages in an M/M/1 queue," in Proc. IEEE Infocom, Boston, MA, USA, 1995.
- [12] P. Dubea, O. Ait-Hellal, and E. Altman, "On loss probabilities in pres- ence of redundant packets with random drop," Elsevier Perf. Eval., vol. 53, pp. 147–167, 2003.
- [13] P. Humblet, A. Bhargava, and M. G. Hluchyj, "Ballot theorems applied to the transient analysis of D/D/1 queues," IEEE Trans. Netw., vol. 1, no. 1, pp. 81–95, 1993.
- [14] O. Gurewitz, M. Sidi, and I. Cidon, "The ballot theorem strikes again: Packet loss process distribution," IEEE Trans. Inf. Theory, vol. 46, no. 7, pp. 2588–2595, 2000.
- [15] T. Stockhammer, H. Jenkac, and G. Kuhn, "Streaming video over vari- able bit-rate wireless channels," IEEE Trans. Multimedia, vol. 6, no. 2, pp. 268–277, 2002.
- [16] J. F. He and K. Sohraby, "New analysis framework for discrete time queueing systems with general stochastic sources," in Proc. IEEE Infocom 2001, Anchorage, AK, USA, 2001, pp. 1075–1084. [17] A. Y. Privalova and K. Sohraby, "Playout in slotted CBR networks: Single and multiple nodes," Probl. Inf. Transm., vol. 43, no. 2, pp. 143–166, 2007.
- [18] [Online]. Available: <http://techcrunch.com/2010/11/19/web-video-37-percent-internet-traffic/> [19] X. Cheng, C. Dale, and J. C. Liu, "Statistics and social network of YouTube videos," in Proc. IEEE IWQoS, Enschede, The Netherlands, 2008, pp. 229–238.
- [20] S. Alcock and R. Nelson, "Application flow control in YouTube video streams," ACM Comput. Commun. Rev., vol. 41, no. 2, pp. 25–30, 2011.
- [21] Y. D. Xu, X. X. Wu, and J. C. S. Lui, "Cross-Layer QoS scheduling for layered multicast streaming in OFDMA wireless networks," Wireless Pers. Commun., vol. 51, no. 3, pp. 565–591, 2009.
- [22] Y. D. Xu and E. Altman et al., "Probabilistic analysis of buffer star- vation in Markovian queues," in Proc. IEEE Infocom 2012, tech. rep. [Online]. Available: <http://arxiv.org/abs/1108.0187>
- [23] Y. D. Xu, E. Altman, R. Elazouzi, S. E. Elayoubi, and M. Haddad, "QoE analysis of media streaming in wireless data networks," in Proc. IFIP Networking 2012.
- [24] Y. D. Xu, S. E. Elayoubi, E. Altman, and R. Elazouzi, "Impact of flow- level dynamics on QoE of video streaming in wireless networks," in Proc. IEEE Infocom 2013