

Buffering of Several Frame with additional Buffer

By

Chirag Joshi¹, Aditya Kishore Saxena², Vishal Bharti³

Assistant Professor, DIT UNIVERSITY, DEHRADUN, INDIA¹

Assistant Professor, DIT UNIVERSITY, DEHRADUN, INDIA²

Professor, DIT UNIVERSITY, DEHRADUN, INDIA³

chirag.joshi@dituniversity.edu.in¹, aditya.kishore@dituniversity.edu.in²,
vishal.bharti@dituniversity.edu.in³

ABSTRACT

Video Streaming Quality of Service (QoS) has been an area of research recently. Video Buffering is an important part of wireless media. Internet speed has increased in last few years but still the buffering of video is not appropriate and it is a challenge to improve the buffering speed. In this paper, we focus on an algorithm which will increase the buffering speed and Quality of Service (QoS). Also it helps users to store the video easily just after watching the video. The algorithm proposed in this paper will use a packet scheduling algorithm and also work on buffering algorithm.

Keywords

Wireless transmission, Video Streaming, Quality of Service (QoS), Packet scheduling algorithm, Buffering.

1. INTRODUCTION

Wireless media services have increased in recent years, error prone nature and because of which everyone is focused on Quality of Service (QoS). Wireless Networks has different methods for transferring frames and one of the best ways to provide it using centralized control, Access Point (AP). Most of the wireless MAC protocol works on the centralized control technique. They are IEEE 802.16 MAC [2] which works for wireless MANs, IEEE 802.11[3] and IEEE 802.15 MACs [4] both of them work for wireless LANs and PANs. Video Streaming is a part of multimedia service. It transmitted large amount of data volumes which requires high rate and less delay. Video Streaming suffers lots from noises which in turn degraded the video quality at receiver's end. Currently, researchers are aiming to reduce this noise and packet losses using different algorithms like ARQ and FEC technique [5], special scheduling algorithm [6, 9]. The frames of video are divided into different Video Packet (VP) and then those video packets are sending using different packet scheduling algorithm. In VP it's easy to detect the error as compared to frame and also it's easy to resend them. Frames are of

different types like I,P and B frames There are different packet scheduling algorithm described in[10].As we are using packet scheduling algorithm for sending the frame from sender to receiver, we are focusing on an algorithm for sending packets from the sender to the receiver. The algorithms in[10] focused on sending the VP with or without data partitioning. Both of the algorithm in[10] have their advantages and disadvantages.

At the receiver end the Quality of media should be intact so that the user can watch the video without much interruption and error. For this several algorithms was proposed in this work and most of the algorithm were able to improve the Quality of Service (QoS) of video streaming. One of those methods was Buffering of Several Frames with an Initial Delay. In this method both sender (Access Point) and receiver were agreed on an Initial delay before sending the frame. In initial stage sender sends the frame, but the receiver does not fetch them from its buffer. Concept of Initial delay for wireless video streaming was introduced in[9]. There is one more algorithm which is Buffering of Several Frames is done with an Initial Delay[1].In this algorithm frames are being sent from sender to receiver with an Initial Delay to improve the packet loss ratio and quality of the frame.

2. RELATED WORK

Video Packets are sending from the sender to receiver using different methods which contain scheduling of VP on the basis of earliest deadline first. One of those scheduling method is described in [10]. In this method VP are sent using with or without data partitioning with re-synch marker. In without data partitioning Frame is sent from the sender to the receiver, each and every frame has some deadline and if a frame is reached after the deadline that frame would be rejected and error is generated. In the second method motion and texture information is inserted between VP. The advantage

of this algorithm is that if at any point of time information of texture is lost, it uses motion information for finding the error. There are two methods describe in [1], one of those method is Buffering of one frame method. In this method two different kind of algorithm was used, first is fixed number of polls per packet and second one is common reserve of polls. Basic difference between both of these is that in Fixed number of polls per packet there are M polls and equal retry to send a packet successfully, if a packet is not transmitted in M polls, it is dropped and new packet is sent. In Common reserve of polls every packet is transmitted till transmission is completed successfully or another frame arrives. This method has better throughput. There are different Packet scheduling algorithm which uses I, B and P frame in Frame based Scheduling [10]. In frame based scheduling frames are divided into Video Packets (VP) which may be separated into motion and texture fields by the motion marker. Algorithm proposed in [10] used video packet with and without data partitioning mode.

Second method which we are also considered in this paper is buffering of several frames with initial delay. Sender and receiver agreed on an initial delay till which receiver does not fetch any frame from its buffer. This algorithm also used Delay Bound parameter which is defined in [3]. After the initial delay receiver starts fetching the frame at a fix rate. If the rate of transmitting is slow, then some of the frames may be lost and if the rate is high receiver buffer was not able to store them properly in its buffer. A signal was sent from receiver to sender to stop sending the frames. In this technique packet loss is less and it assumes fixed number of video frame

in the system. The size of buffer in both sender and receiver is enough to store the maximal number of video frames. This method shows the distribution of frames through discrete Markov Process. It made a probability matrix of size $(L+1)*(L+1)$ which is used for calculating the how receivers buffer was filled during the interval. The conclusion drawn by this algorithm is that during initial delay Access Point (AP) sends the frame, but the receiver does not fetch it from its buffer.

3. PROPOSED ALGORITHM

We have proposed an algorithm which used a packet algorithm [10] and buffering algorithm [1]. We have made some changes in both the algorithm to propose a new algorithm which will transfer the Video from sender to receiver with less error and improved Quality of Service(QoS). For sending the Frame from the sender to receiver our algorithm divides the frames contained in a Video into different VP. Let a video have different frames like F1, F2, F3,.....Fn. Each of these frames are divided into different VP like frame F1 is divided into V11, V12, V13,.....V1m Video Packets[10]. Here V11 denotes the first video packet of the first frame and V1m denotes the mth video packet of first frame. Architecture of this transmission is shown below in Fig.1.

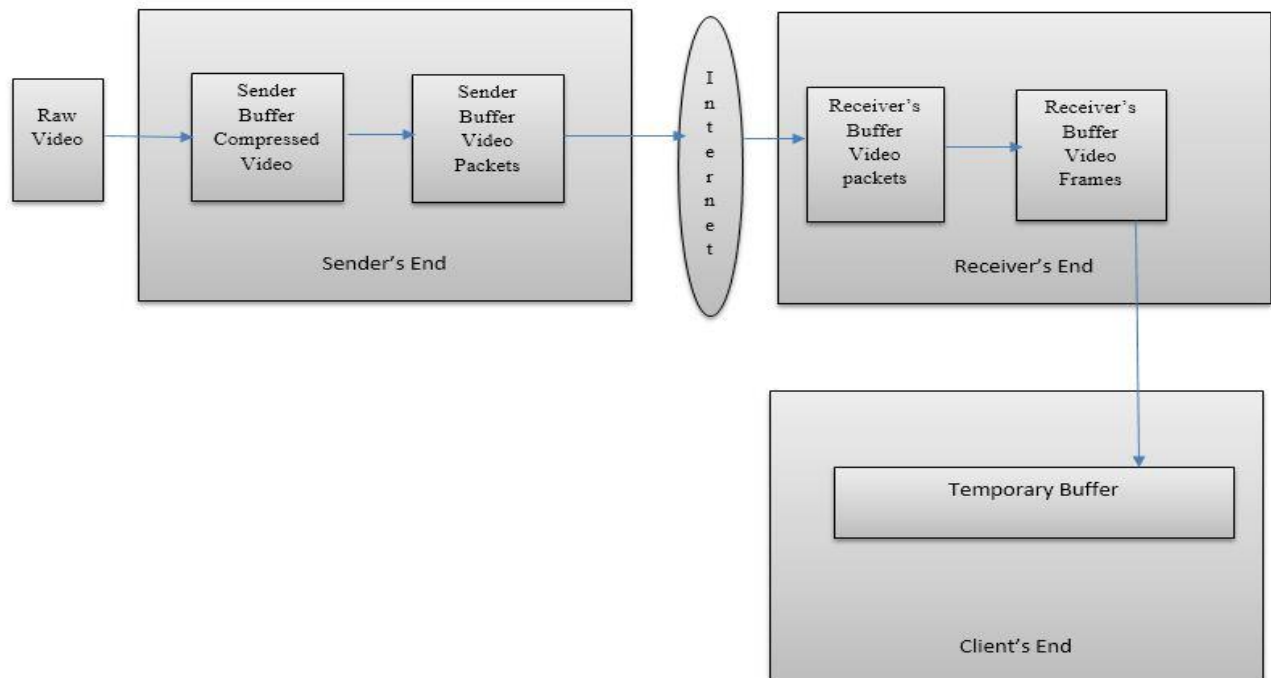


Fig.1 Architecture of Transmission

First a raw video is compressed into video frames and stored in sender's buffer. This video frames are further converted into video packets which will be sent to the receiver. At receiver end first all the video packets are collected and arranged in an order according to their sequence number, when receiver gets all the video packets all the video packets are transformed into frame. Receiver sends the video frames to the temporary buffer and from where client can watch the video.

A video is consist of number of frames and these frames are composed of I,B and P frames in which most of the time I frame is the starting frame. At sender end a video packet is divided into frames and is stored in a buffer. We does not send these frames to the receiver instead the frames are divided into different video packets. We are assuming that the number of packets in each frame is same. After a frame is divided into VP a hello packet is sent to the receiver containing the information about the video. Fig.2 shows a hello packet which contains size of video in MB and time (Minutes) , with that it also contains number of video frames and number of video packet in each of the frame.

Video Size (MB)	Video Size (MINT)	NVF	NVF (in each packet)
-----------------	-------------------	-----	----------------------

Fig.2 Hello Packets

Temporary Buffer: This temporary buffer will be situated in the permanent storage. It is called temporary because it will store the video for some time and if the user does not save the video at any other location it will be removed from that storage. The size of this buffer is double of receiver's buffer size. If the size of the receiver's buffer is 10MB, size of temporary buffer is 20MB. In first 10 MB it will store the parts of the frame it receives from the receiver and in the rest 10MB it will store the part of the video which was viewed by the user. This next 10 MB is used for storing the video.

Each of the video packet has a number which will identify that which frame's video packet is lost. The receiver will rearrange the video packets in their order and when it receives the entire video packet it compress them and stored as a frame in its buffer. When receiver receives 20% of total number of frames then only it transmit the frame to the temporary buffer which will be responsible for showing the video at the user

end. If a video packet is lost in between receiver sends a negative acknowledgment of that video packet after receiving the next two video packets. For example if a V17(17th video packet of first frame) is lost the sender waits for V18 and V19 or another two packets, if V17 received by receiver before coming of next two video packets receiver does not send the negative acknowledgment.

In our algorithm we have made some changes in the proposed algorithm which is discussed in [1]. Our algorithm uses a temporary buffer (extra buffer) at the receiver's end and it will help the end user to watch the video without waiting for much time (which the user waited earlier in buffering). Also, it helps the user to wait only for some initial moment and video is also downloaded in that temporary buffer, which will give an option of storing that video in permanent storage at the end of transmission from temporary buffer. Fig.3 showing the scenario at sender's end.

It is assumed that the transmission between sender and receiver will be done as explained in [1]. In this algorithm we have used a sequence number which will be used by sender when it sends a frame to the receiver. The receiver when fetch the frames, it will rearrange them with the help of sequence number. The frames from sender will transmitted in an order according to a sequence number.

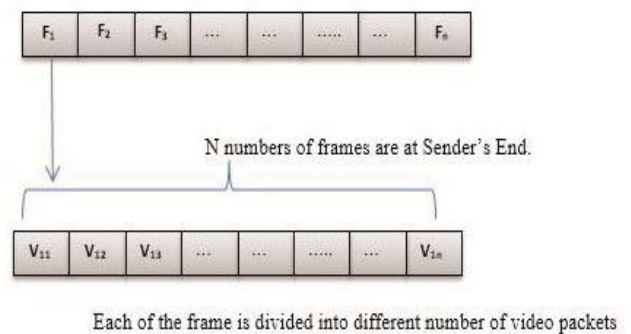
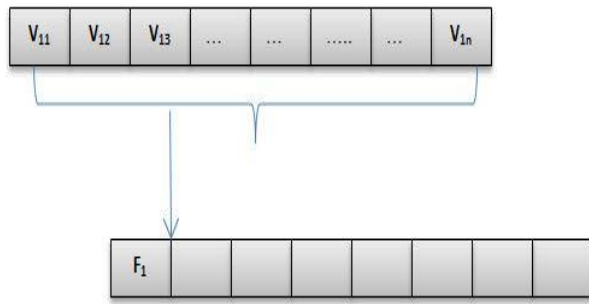


Fig.3 at Sender's End

The number of frames that can be transmitted at once will be decided by the sender and the receiver before the start of the transmission. If any frame is lost, the receiver will send the negative acknowledgment of that frame .Algorithm in [1] uses a Delay bound parameter which will also be used here if any of the frames was delayed. Figure 4 shows scenario at

receiver's end.



Video Packets are compressed into frame.

Fig.4 at Receiver's End

First of all sender and receiver will agree on an Initial delay as in [1] and with that they will also decide the size of buffer. Sender sends a hello packet to the receiver which will contain the details of video frame. After receiver got the packet, it will read the values from hello packet. Receiver sends a packet to the temporary buffer which shows him the size of video in MB. Size of receiver's buffer is initially more than the size of video frame and can be extended on requirement during transmission. Temporary buffer after reading that packet and sets its size according to the size of video frame (which is double of size of video frame). Sender stores the frames in its buffers and after storing the frames the sender divides the frames into different number of video packets.

When receiver received a video packet it checks its sequence number and put at its proper place (except the first video packet), receiver can use any of the sorting algorithm to arrange the entire video packet in an order. If any of the video packets missed its deadline receiver sends a negative acknowledgment after receiving next two video packets. At receiver when all the video packets of a frame are received, it compressed them into a frame. Receiver waits before sending the frame to temporary buffer, till it receives 20 % of frame of the total size of frame (in minutes). When it receives the 20 % of frame, It will transfer that 20 % of frame to the temporary buffer. Receiver uses a signal which will be sent to temporary buffer when it starts receiving the frame from sender. If there is a delay in receiving the frame, receiver sends only 10% of frame to the temporary buffer excepts for the first time it sends frame to the temporary buffer. At temporary buffer, a user who is watching that video has to wait only for first 20 % of time else there will not be any interruption in watching the video. After temporary buffer receives all the video frames, user gets an option of saving the video to its permanent storage. If user denies saving the video, temporary buffers removes that video from its buffer. Figure 5 shows how receiver sends a frame to temporary buffer.

Let's assume if channel bandwidth is β VP/sec and distance between sender and receiver is d . Therefore time required for sending 1 video packet is $1/\beta$. If Δ is the total time for initialization of video packet at sender end and arranging the video packet at receiver end, total time for sending 1 video packet is $((1/\beta) + \Delta)$. Now if total number of video packets are m then total time for sending all the video packet is $m((1/\beta) + \Delta)$. If a video packet missed its deadline then it will be resend so the total time will be increased by α where α will be equal to $((1/\beta * 2) + 1/\beta)$ that will be $3/\beta$. If a frame consist of m video packets total time for sending a frame is equal to $m(1/\beta) + \Delta + \alpha$.

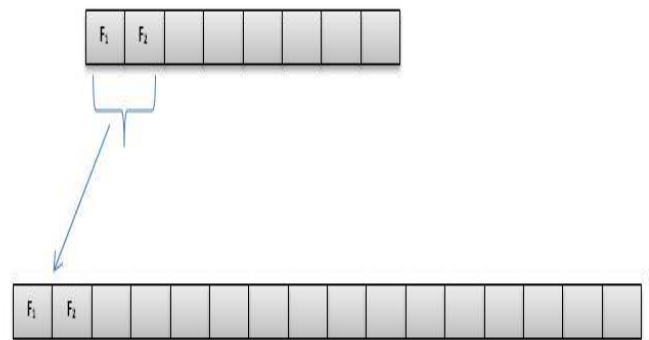


Fig.5 Receiver Sends a Frame to Temporary Buffer

Assume if there are total 10 number of video packets that needs to be sent on a channel whose bandwidth is 6V P/sec and $\Delta = 0.6$ Then total time for sending a complete frame $= 10((1/6) + 0.6) = 7.67$ sec. Now the maximum value of α will be when all the video packets of a frame missed their deadline then in above example $\alpha = 5$ sec. so total time in which a frame can be sent is $= 12.67$ sec. If total number of frame in a video is F then total time for sending a complete video will be equal to $F(m((1/\beta) + \Delta))$. Now if there are 10 frames in a video of 10 minutes it will take 126.67sec. If number of video frames in this video is increased to 20 from 10 (video packets are 10) then time taken for sending this video will be 253.33sec. Now if we increase number of video packets in each frame from 10 to 20 and video frames will be 10 in numbers .Total time for sending a video is 153.33sec. When number of frames is increased keeping the number of video packets same time required for sending the video is increases by double but when number of frames is same and number of video packets in each frame increases time required for sending a video is increased by 21%. If number of video packets are increased instead of number of video frames Quality of Service (QoS) will improved.

4. CONCLUSION

Video Buffering is an important wireless service and the challenge in this field is to improve the Quality of Service (QoS). In our algorithm we have used a previous algorithm of [1] which uses initial delay when buffering several frame. We have added an extra buffer at the receiver end so it will help the user to watch the video without waiting for buffering, also it will give an option to store the video which was earlier has to be done separately. If a video frame is divided into different video packets its always easy to find the error. Also, every video packet is arranged and compressed into frame at the receiver end which will help in smooth streaming of video. This algorithm will improve the Quality of Service(QoS) of a video.

5. REFERENCES

[1] Andrey Lyakhov and Oleg Sidorkin, Study of buffering algorithm for video streaming over wireless network, **2008**.

[2] IEEE Std. 802.16-**2004**, IEEE standard for local and metropolitan area networks, Air Interface for Fixed Broadband Wireless Access Systems.

[3] IEEE Std. 802.11-**2007**, Revision of IEEE Std. 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer(PHY) specification, June2007.

[4] T.P.-C. Chen and T. Chen, Fine-Grained Rate Shaping for

Video Streaming over Wireless Networks,IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP **2003**, Hong Kong,China.

[5] A. Majumdar, D.G. Sachs, I.V.Kozintsev, K.Ramchandaran, and M.M Yeung, Multicast and unicast real-time video streaming over wireless LANs,IEEETrans. Circuits Syst.Video Technol., vol.12, pp.524-534, June **2002**.

[6] K. Sripanidkulchai and T. Chen, Network-adaptive video coding and transmission, Visual Communications and Image Processing, San Jose,January **1999**.

[7] Z. Miao and A. Ortega, Expected run-time distortion based scheduling for delivery of scalable media, Packet Video Workshop, April **2002**.

[8] S. H. Kang and A. Zakhor, Packet scheduling algorithm for wireless video streaming, Packet Video Workshop, April **2002**.

[9] M.-H. Lu, P. Steenkiste, T. Chen, Video Streaming Over 802.11 WLAN with Content-Aware Adaptive Retry, ICME **2005**, pp.723-726.

[10] Sang H.Kang and Avidesh Zakhor,Packet Scheduling Algorithm for Wireless Video Streaming.