Congestion Controlling for Streaming Media Through Buffer Management and Jitter Control

Dr.P.Bala Krishna Prasad†, **Mr.G.Murali**††,**G.Gurukesava Das**†††,**K.Siva Krishna Rao** ††††
Principal, St.Ann's Engineering College, Chirala-523187, A.P,India†
Associate professor, Dept of CSE, St.Ann's Engineering Coolege, Chirala-523187,A.P, India††
Associate professor, Dept of CSE, St.Ann's Engineering College, Chirala-523187,A.P, India†††
Associate professor, Dept of CSE, PNC & VIJAY Instute of Engg & Tech, Repudi-522529, A.P, India

Summary:
This paper deals with the introduction and designing of a buffer manager, which is a part of the quality manager of the client. The aim of designing buffer manager is to prevent the buffer from overflowing, under flowing, control the jitter and send the congestion indication signals to the server. When the play out buffer indicates buffer overflow, it is not possible for the decoder to decode all frames with in stipulated time, then some of the low quality frames will be dropped. To drop frames, an algorithm named as 'Balanced Frame Drop' is proposed in this paper.. The congestion indication mechanism at receiver has been discussed in this paper. If the number of packets loss is greater than the threshold value, then it is treated as another indication for congestion. Jitter can cause jerkiness in playback due to the failure of some samples to meet their presentation deadlines. The use of buffering effectively extends the presentation deadlines for all media samples and in most cases practically eliminates playback jerkiness due to delay jitter. The playback buffer reduces the number of packets that arrived with in their playback deadline. The extended presentation deadlines for the media samples allow retransmission to take place when packets are lost; it is possible through playback buffer and play out scheduler.

Key words:
Frame drop, play out buffer, Jitter, buffer underflow, buffer overflow

1. Introduction

Due to the explosive growth of the Internet and increasing demand for multimedia information on the web, streaming video over the Internet has received tremendous attention from academia and industry. Streaming video requires bounded end-to-end delay, so that packets can arrive at the receiver in time to be decoded and displayed. If a video packet does not arrive in time, the play out process will pause, which is annoying to human eye. A video packet that arrives beyond its play out time is useless and can be regarded as lost. Hence in order to provide continuous play out, a buffer at the receiver/client was introduced before decoding, which is called play out buffer. It is common for streaming media clients to have 5 to 15 seconds of buffering before the playback starts. Buffering provides a number of important advantages in the performance of streaming systems over best effort networks such as the Internet. Some of the advantages are:

- The buffer at client allows compensating for short term variations in packet transmission delay.
- It allows the client to perform error recovery through retransmissions, error resilience through inter leaving.
- It allows the client to continue playing back the content during loses in network bandwidth.
- It allows the content to be coded with variable bit rate, which can improve overall quality.

Jitter causes jerkiness in playback due to the failure of some samples to meet their presentation deadlines, so, the packets must be skipped or delayed. The effective use of buffering extends the presentation deadlines for all media samples and in most cases, practically eliminates playback jerkiness due to delay jitter. The playback buffer reduces the number of packets that arrive after their playback deadline. The extended presentation deadlines for the media samples allow retransmission to take place when the packets are lost; this is possible only through playback buffer. The use of buffers is exemplified in the figure 1.

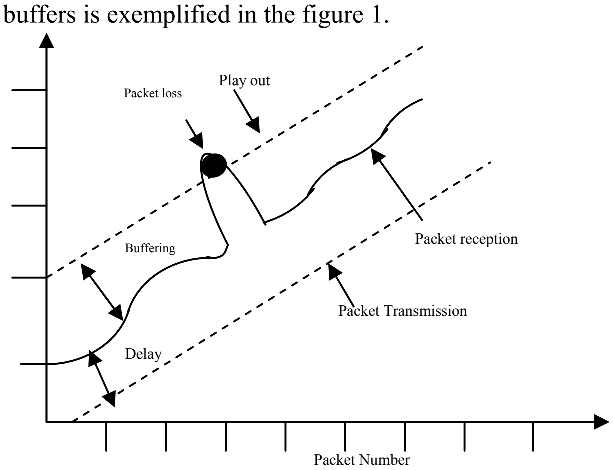


Figure 1: Role of play out Buffer in streaming media

Manuscript received February 5, 2009
Manuscript revised February 20, 2009

In the figure it is shown how the packets are transmitted and played at a constant rate and the playback buffer reduces the number of packets that arrive after their playback deadline. In the buffer management scheme we use two types of buffers, one is adaptive buffer and the next is play out buffer, which are depicted in Figure2.

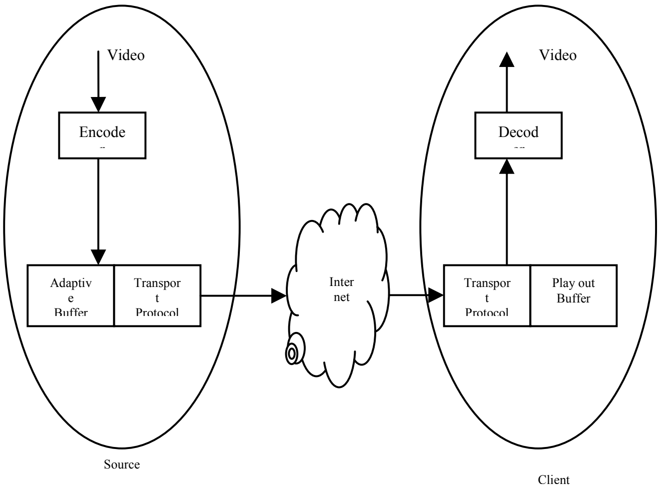


Figure 2: Two types of buffers: Adaptive and play out

The adaptive buffer [1] at sender is required if the encoder cannot adapt as quickly as the network conditions vary. This buffer can be used to differentiate the packets according to their priority and send only the most important packets in the available bandwidth, according to the feedback information about the congestion; the adaptive buffer drops the low priority packets.

2. Play out Buffer

It has been observed that available bandwidth fluctuates with time, if buffering is not used playback disruption would occur, if the instantaneous available bandwidth is lower than the media rate, in order to reduce the number of playback disruptions, clients would need a buffer [2]. Generally all the clients of streaming media typically employ a buffer of 5 to 15 seconds of buffering capacity, with such buffering, a client can pre fetch the data that is not immediately needed when the available bandwidth is above the media rate and then the client drains the buffer when the available bandwidth is below the media rate. A video packet that arrives beyond its payout time is useless and can be regarded as lost. Hence in order to provide continuous play out, a buffer at receiver/client has been introduced before decoding, it is called play out buffer, and the major aim of introducing the play out buffer is to smoother the bit rate variations [3,4]. The buffer can convert the frames from VBR to CBR (Variable Bit Rate to Constant Bit Rate) which is depicted in figure 2 (a). The bit rate variations not only depend on the image contents but also depend on different frame types i.e. at a similar quantization setting and quality level an I-frame generates more bits than a P or B-frame.

Another aim is to control jitter, Therefore the play out buffer can also be called as jitter buffer. Jitter is the delay variations [5] between packet arrivals for a given stream; which is depicted in the figure 3.

The packets transmitted by sender and play out by client at a constant rate according to the sequential order. But suppose the receiver received the packets with delay variations; this can be controlled by jitter buffer.

The play out buffer introduces additional delay (BD= Buffer Delay) with the aim of producing a play out schedule which meets the synchronization requirements. Here play out buffers which are set as holding area for packets, their scheduled play out time is in the future.

Packets which arrive after their scheduled play out time are considered late and are discarded. Common algorithms for managing a play out buffer take one of the two basic approaches: first is fixed approach in which it is assumed that the range of delay is predicable and hence we use a static buffer size and schedule.

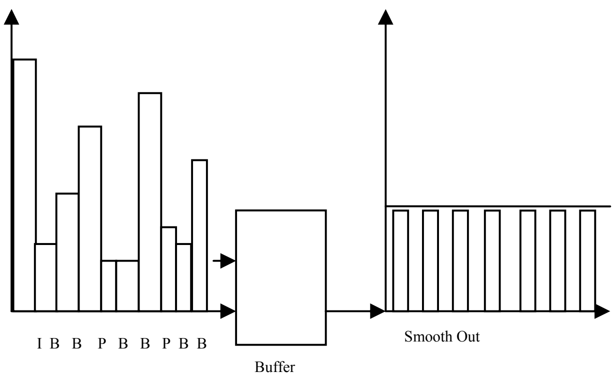


Figure: 2(a) VBR to CBR conversion

Second is Reactive approach which is common in the Internet, can measure immediate jitter and can be used to dynamically adjust the buffer size and schedule to avoid the delay. Fixed approaches are known buffering delay but with potentiality large packet delay, while reactive approaches avoid the delay but at the expense of potentially very high buffering delays. In this work an attempt has been made to use the CONCORD Algorithm.

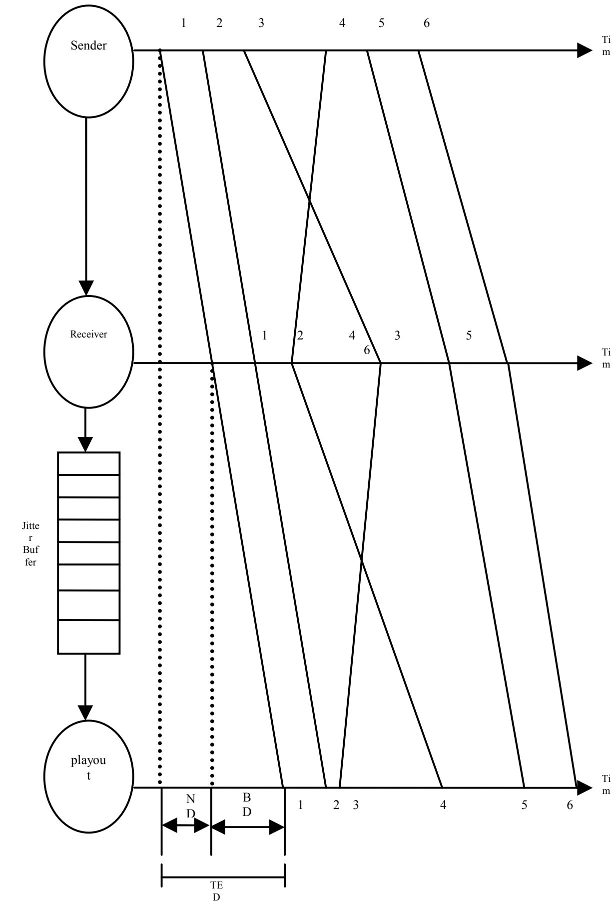


Figure 3 Jitter control through play out / jitter buffer

2.1 Frame Drop

It has been observed that whenever the play out buffer indicates buffer overflow and it is not possible for decoder to decode all frames in time, some of the low priority frames get dropped. Here the question is which the low priority frame is and how to identify this? Before answering these questions it is essential to know about various types of frames.

MPEG (Motion picture expert group) [6,7] is a standard; which defines 3 types of frames. I, P, B. The I-frames (Intra) are coded as still images, they contain absolute picture data and are self contained and they require no additional information for decoding. I-frames have only spatial redundancy providing that least compression among all frame types. Hence they are not transmitted more frequently than necessary. Consider the figure 4 for frame type

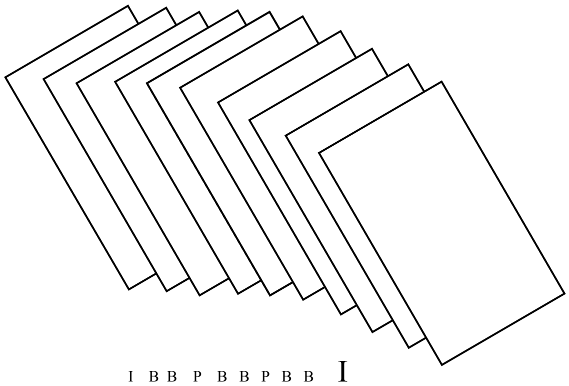


Figure 4: I,P,B Frames

The second type of frames is P-frames (Predicted). They are forward predicted from the most recently reconstructed I or P frames.

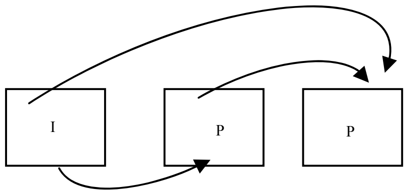


Figure 5: Dependencies of predicted frame (P-Frame)

P frames are not self contained, i.e. if the previous reference frame is lost, decoding becomes impossible. The third type is B (Bi-directional predicted) frames. They use both forward and backward prediction. I.e. a B-frame can be decoded from a previous or later I or P frame. This is depicted in figure 6

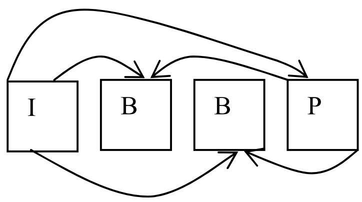


Figure 6 Bidirectional predicted frame

B frame require resource intensive compression techniques, but, they also exhibit the highest compression ratio.

2.1.1 GOP (group of pictures)

Decoding of I frames needs no previous frame, decoding can begin at I coded information. An I frame together with all the frames before the next I frame, form a group of pictures (GOP) [8,9]. GOP (n) and GOP (n+1) are two consecutive GOPS depicted in figure 7

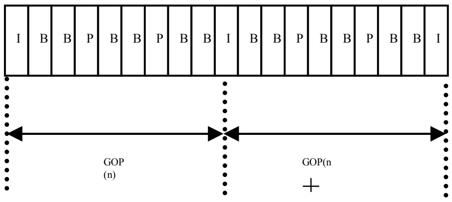


Figure: 7 Group of Pictures (GOP)

The GOP length is flexible, but 12 (or) 15 frames are common value on average, P frames requires roughly half the data of an I-frame and B frame requires roughly ¼ th data of an I- frame (or) half the data of a P-frame. The last B-frame in GOP requires the I-frame in the next GOP for decoding and so the GOPS are not truly inside pendent, it is depicted in the figure 8

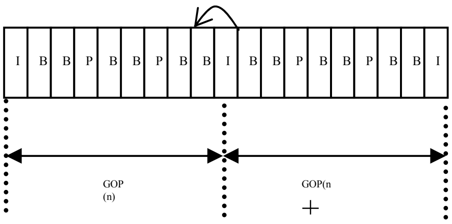


Figure: 8 the last B-frame requires the I-frame in the next GOP

2.1.2 Priority of frames

Skip the frames, when it is not possible to decode all frames in time. Generally this situation occurs when there is a buffer overflow. The I-frame is the most important frame in a GOP because all other frames depend on it [10]. If an I-frame is lost within the GOP, then decoding of all consecutive frames in the GOP will not be possible.

B-frames are the least important ones because they are not reference frames. Skipping one B-frame will not make problem on others, while skipping one P-frame will cause the loss of it, subsequent frames and the two preceding B-frames with in the same GOP. Hence it is better to drop B-frames first, then P-frames and finally the I- frames.

Priority (I) > priority (p) > priority (B)

2.1.3 Skipping Distribution

It is seen that a GOP with evenly skipped B frames will be smoother than a GOP with unevenly skipped B frames, suppose a GOP is I B_1 B_2 P_1 B_3 B_4 P_2 B_5 B_6 P_3 B_7 B_8, then, even skipping of frames, i.e., I-B_2P_1-B_4P_2-B_6P_3-B_8 will give smoother video than uneven skipping of frames, i.e. I - -P_1 $B_3B_4P_2B_5B_6$- - - hence it is better to drop the B-frames evenly.

2.1.4 Buffer Overflow and Underflow

Input to the client is underflow and frame buffer (or) play out buffer overflow when the decoder is too fast, i.e. when the decoding catches are too small and /or the display latency is too large. Input overflow and output underflow occur when the decoder is too slow .i.e. when the decoding latency is too large and/or the display latency is too small. In the case of output underflow, the display does not have a new frame to display, in this situation it is necessary to retain the previous frame for display until a new one arrives. The buffer occupancy ratio can be calculated as follows: Buffer occupancy ratio (BOR) at time interval between K and K+1 is:

$$BOR = \frac{(BUF)_{K+1}}{B_S}$$

Where B_S is the buffer size at client, $(BUF)_{K+1}$ is the number of packets in the client buffer at time k+1, which is calculated as follows:

$$(BUF)_{K+1} = (BUF)_K + R_K - P_K$$

Where R_K is the number of packets transmitted by server, in time interval 'K', P_k is already used packets for playback and $(BUF)_K$ is the number of packets in the client buffer at time 'k'. This section introduces two alert points, first is 'threshold point' second one is 'underflow point'. The calculation of these two is based on BOR. The threshold value is fixed at (0.8 x BOR).

If BOR > (0.8 X Bs), it is signaled as dangerous situation as BOR reached threshold value. Similarly, if BOR < (0.2 X Bs), it is treated as buffer underflow caused by the congestion. In our design the client should send the feedback information to sender about the congestion information, BOR value. Server can adapt to the conditions immediately and act accordingly.

In this dissertation introduced a frame drop algorithm named as "Balanced frame drop". This is explained in the next section.

2.1.5 Balanced Frame Drop Algorithm

One method of speeding up decoding upon buffer overflow situation is to drop some frames, but, dropping the wrong frame, at the wrong time can result in a noticeable disturbance in the played video stream. The proposed algorithm selects the frames evenly and correctly, to drop. The algorithm is as follows.

Algorithm:

1. Count the number of frames in each GOP and assigned this value into variable 'S' where S:= number of frames
2. The priority values for the frames are 1,2,3,4,5,- - - -S

3. Assign the highest priority i.e. least value (1) to the frame I

I	B	B	P	B	B	P	B	B	P	B	B

4. Identify all the P-frames within the GOP and given numbers from 1 to k.
   {P1, P2, P3, P4, P5 - - - - - - - - -P_K}
5. Give the priority to the P-frames as follows.
   (i) A P-frame which is closest to the I-frame is given the highest priority among all P-frames and named as P1.
   (ii) A P-frame which is the next closest frame to the I-frame is given the next priority in the group of P-frames {P1, P2, P3, P4, P5 - - - - - - -P_K} and it is named as P_2 .
   (iii) P_K is the least priority one of the P-frames

I	-	-	P1	-	-	P2	-	-	P3	-	-

P1 is the closest one to the I-frame among all P-frames hence P_1 will be given next high priority to the I-frame

	B1	B2	P1	B3	B4	P2	B5	B6	P3	B7	B8

6. Assign the priority to the B-frames according to their bit size. Suppose B_3 is the largest B-frame then assign the priority value 5 to B_3, 6 to next largest B-frame.

1 2 5 3 4 : priorities

I	B1	B2	P1	B3	B4	P2	B5	B6	P3	B7	B8

7. If there is a need to drop less than 3 frames and then drops the three B-frames, which are the least sized ones.
8. Other wise (> 3), the B-frames to be dropped should be even, because a GOP with evenly skipped 'B' frames will be smoother than a 'GOP' with unevenly skipped B-frames. Hence drop the least sized B-frames in either even positions (or) odd positions.
   B-even:B2→B4→B6→B8.
   B-odd: B_1→ B_3→ B_5→ B7.
   If size (B2 + B_4 + B_6 + B_8) > size (B_1+ B_3+ B_5+ $B_{7)}$
   Then drop B-odd frame set
   Else
   Drop B- even positioned set.

2.2 Congestion Indication Mechanism

Before knowing congestion indication technique, it is important to know video communication system. Video is encoded and packetized into individually decodable packets to prevent the propagation of errors caused by the packet loss on the large time scale at the group of pictures level, encoder adapts to the network rate, it is based on estimated available bandwidth on the smaller time scales, (when the GOP is already encoded, but not get transmitted) the actual transmission rate is regulated by the transmission buffer. The buffer manager gets feed back from the congestion control scheme about the current network conditions. Consider the figure 9, which depicts the congestion control mechanism.

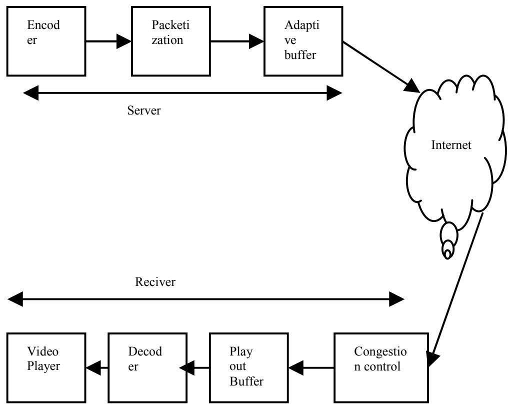


Figure 9: Block diagram of video communication system.

A congestion control scheme [11-13] serves to minimize burst losses in the network. It ensures network scalability and is fair to other flows. None of the packets transmitted by a sender are discarded due to lack of buffers, the receiver is required to periodically inform to the sender about the congestion, Buffer occupancy ratio and space availability in buffer. Based on this information the sender either reduce the flow or the receiver drop the frames. The congestion indication mechanism at receiver is as follows.

The time required by the encoder to encode a packet is ΔT_e, which is named as encoder delay. The time the packet stayed in the buffer is named as encoder buffer delay (ΔT_{eb}). The channel transmission delay is named as ΔTc. The decoder buffer delay is named as Δ Tdb and Δ Td is the decoder delay, refer figure 3.10 . The end to end delay is the sum of all the five delays and it is named as ΔT where ΔT is

$\Delta T = \Delta T_e + \Delta T_{eb} + \Delta T_c + \Delta T_{db} + \Delta T_d$

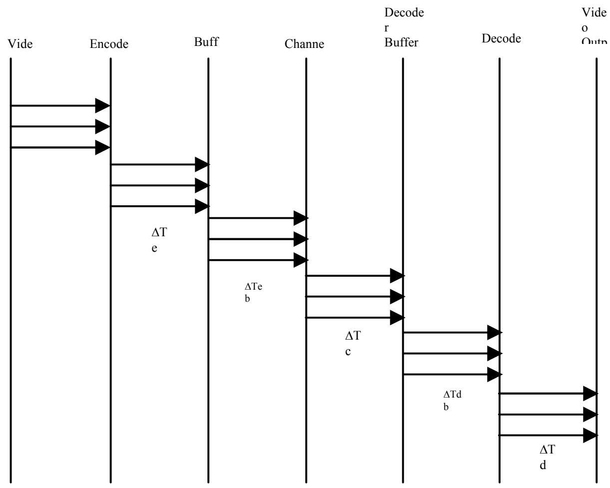
If the delay of a frame is greater that ΔT, it is one of the indications for congestion. If the number of frames in the decoder buffer is less than 20 % of buffer size (i.e. buffer underflow), then the buffer delay may increase. The number of video frames stored in Decoder buffer is:

$$\Delta N = \frac{\Delta T_{db}}{T_f}$$

Where T_f is the time interval for one video frame, ΔN is the buffer delay in terms of number of frames. If the lower value of ΔN indicates the frames temporally stored in the buffer is with in small period of time, because no packets are kept stock in the buffer; this is also indication for congestion. The congestion controller observes the packet losses [14], which is another indication for congestion.

3. Jitter Control

Variations in the network delay is called jitter, it destroys the temporal relationships between periodically transmitted media units that constitute a real time media stream [15,16]. In the absence of jitter and packet loss, video frames can be played as they are received, resulting in a smooth play out as depicted in the figure 11



ΔT_e = encoder delay
ΔT_{eb} = encoder buffer delay
ΔT_c = channel transmission delay
ΔT_{db} = decoder buffer delay
ΔT_d = decoder delay

Figure10: Different types of delays in video communication

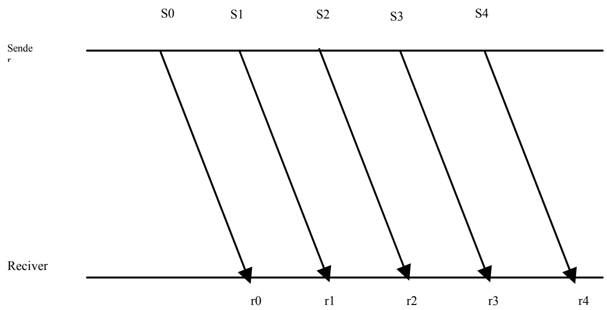


Figure 11(a) A Jitter free and Loss free stream

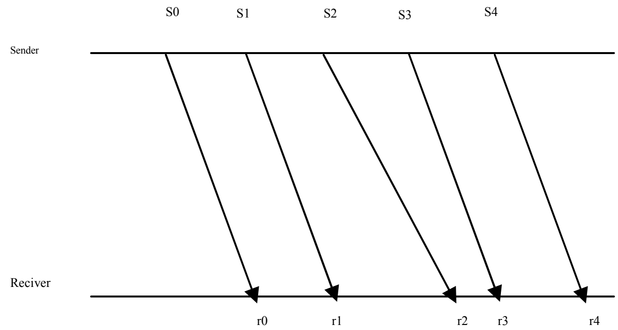


Figure 11(b) A Stream with Jitter, but loss free of frames

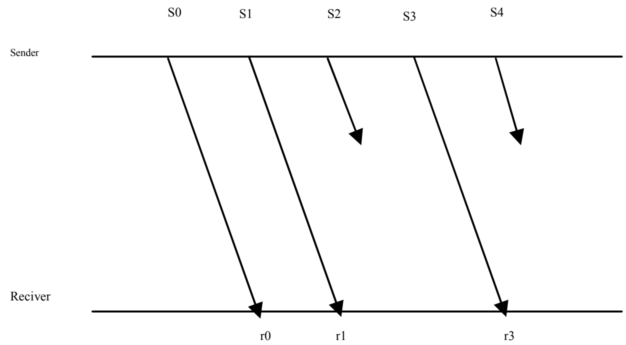
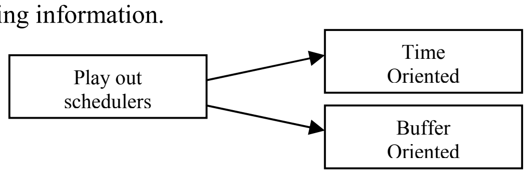


Figure 11(c) a Stream with packet loss and jitter

However in the presence of jitter, inter arrival times vary. In the figure 11(b), the third frame arrives at r2, which is late, in this scenario, the user would see the frozen image of the most recently delivered frame (frame two, i.e. S1) until the tardy frame (frame three i.e. S2) arrives, it would then be played in order to preserve the timing for the subsequent frame (frame 4). In the presence of the packet loss, some frames will not even arrive at the receiver as depicted in the figure 11(c). The third and fifth frame does not arrive at the receiver. In the case of loss of frame 3, the viewer would see a frozen image of the most recently delivered frame (frame two i.e. S1) and the video stream would then jump to the next frame that arrived (frame 4, i.e. S3).

Delay buffering can compensate for jitter at the expense of latency; transmitted frames are buffered in memory by the receiver, allowing each frame to be played out with a constant latency, achieving a steadier stream. Jitter delays the temporal relationships between periodically transmitted video frames. Temporal relationship refers to the spacing between subsequent frames, which is dictated by the frame production rate; typically it is 25 to 30 frames per second.

A Packet Video Receiver (PVR) consists of a play out buffer and a play out scheduler [17]. Buffer is for the temporary storage of incoming frames and scheduler is for determining the presentation initiation time and the presentation duration of each frame. The play out scheduler [18] is able to regulate the presentation duration of a video frame. The presentation duration is equal to the inverse of the frame production rate. The general principles that drive the operation of the scheduler in those large discontinuities between congestive frames are undesirable as they are easily detected by users and therefore it is desirable to break them into discontinuities of smaller duration that may be unnoticed due to human perceptual limitations in the detection of motion. Un presented frames that wait in the play out buffer increase the end-to-end delay of each newly arriving frame. The end-to-end delay measures the time between the encoding of a frame at the service and its presentation at the receiver. The play out scheduler can control the jitter effectively, basically play out schedules are of two types, based on whether the system uses or does not use the timing information.



Time oriented schedules put time stamps on MU'S (Media units) and use clocks at the sender and the receiver in order to measure the network delay (jitter). Buffer oriented play out schedulers involve schedulers that delay with the fundamental synchronization tradeoff but do not require the time stamping of MU'S or the use of clocks. Buffer oriented schedulers shape some resemblance with time-oriented schedulers that use differential delay methods to adjust the play out part. Buffer oriented schedulers implicitly assess jitter by observing the occupancy of the play out buffer. The adjustment of the play out point is based on the occupancy of the buffer; which is depicted in the figure 12.

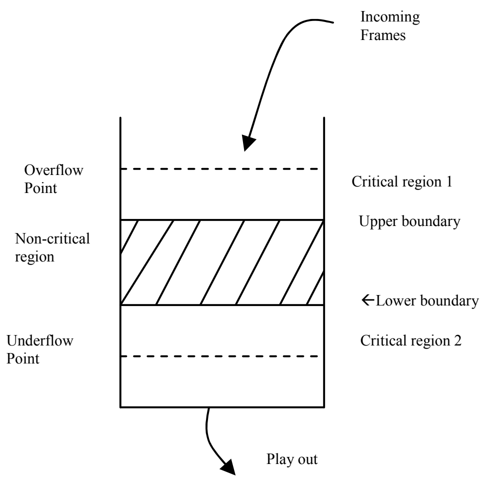


Figure 12 Buffer occupancy and threshold values

The scheduler enters an adoption phase with the aim of returning the occupancy inside the non-critical region. This is accomplished by modifying the receiver consumption rate until the occupancy returns in the non-critical region.

If UB<BO<CR1

Then play out scheduler increases the presentation rate, since the buffer

Occupancy (BO) is above the upper boundary.

Else If BO=CR1

Then presentation rate will be increased up to maximum play out rate,

Else excess packets will be lost.

If (BO<LB) & (BO>UFP)

Then play out rate will be decreased until the

Occupancy resides within the non-critical region.

Else (BO<UFP)

Where *BO* = buffer occupancy

LB = lower boundary of non-critical region

UFP = underflow point.

CR1 = critical region 1.

This situation is signaled as dangerous and minimize the play out rate as minimum as possible, If the condition is still worst, then this information has to be passed to the sender regarding the buffer occupancy ratio. Generally this situation is faced at the time of congestion. The sender treats this as a signal of congestion and activates the bit rate controller (which will be introduced in the 4th chapter) to take appropriate action.

The scheduler considers the current occupancy of the play out buffer as an implicit indication of jitter and takes all actions based on this information.

λ_f is the transmitted rate, generally it is 25 to 30 frames/sec. Space or duration between each frame (T) is $1/\lambda_f$

$$\therefore \qquad T = \frac{1}{\lambda_f}$$

The i^{th} frame interval x_i, is given by

$x_i = T + (D_{n,i}) - (D_{n,i-1})$ where $(D_{n,i})$ is the network delay of i^{th} frame.

If $(D_{n,i} = D_{n,i-1})$

Then inter arrival spacing is equal to the inter departure spacing.

If $D_{n,i} > D_{n,i-1}$

Then

The two frames drift apart (Xi >t).

Else

They approach each other. (Xi <T).

If $(D_{n,i-1} = D_{n,i} + T)$

Then

The two frames arrive concurrently at the PVR(packet video receiver) this is called clustering of frames.

The expected duration of intervals is E{X} = T, the variance of intervals is var{X} = 2 var {D_n} the distribution is symmetrical around its mean value.

The PVR utilizes a maximum play out rate (μ), when the number of buffered frames 'i' is greater than or equal to a given upper boundary (UB), $i \geq UB$ where i is number of frames in the buffer. The play out rate will be reduced if i <UB, it is

$$\mu(i) = \frac{\mu.i}{UB}$$

When the system operates above UB, it employs a minimum play out rate 'μ' which is greater than 'λ'.

$$\mu > \lambda \begin{cases} if \quad i > UB \quad then \\ \\ \quad \mu > \lambda \end{cases}$$

Where λ is actual video frame rate and μ is maximum video play out rate. If μ > λ It means the system displays frames faster than the normal video rate.

The frames are presented at a linearly decreasing rate when the play out buffer occupancy drops below UB. The value of UB shows its impact on the system performance. More specifically the value of 'UB' is examined in relation to the probability of an empty buffer (Π_0), the frame loss probability due to buffer overflow P_L and the mean presentation rate is $\bar{B}$. It is observed that both Π_0 and $\bar{B}$ decrease with UB/Threshold, while P_L increases with UB. The selection of UB can be seen as a trade off between the preservation of play out continuity, which is captured by Π_0 and the reduction of mean play out rate with respect to actual video rate, which is captured by $\bar{B}$. The play out controller of the system displays frames at a rate μ equal to the actual video rate **λ** when 'i' is greater than or equal to UB.

Play out distortion

An attempt is made to construct a mathematical model to derive the optimal frame play out policy. All the frames are presented with an equal duration (f_d) under normal conditions f_a is the frame arrival rate (or) frame production rate. Then

$$f_d = \frac{1}{f_a}$$

The presentation duration is shorter than (f_d), which means there is a transient increase of play out rate and need to expand the presentation duration of a frame (f_d) to reduce the play out rate. The most general method to regulate B, the duration of a frame, is to allow it to take all non-negative values.

$$D f_k \overset{\triangleleft}{=} \text{K.f}$$

Where 'f' is a fraction of the normal frame duration (f_d) such that

$$f_d = \mathbf{C_f \ x \ f}$$

The value of C_f is the cutting factor of (f_d). Using the last relationship DF_k becomes:

$$DF_k = \frac{k}{F_a.C_f}$$

$$F_a = \frac{1}{F_d}$$ Is the normal frame rate, typically 25 or 30 frames/sec. 'f' will be the basic unit for shortening and expanding the duration of a frame. The reduction of freedom in the search for optimal policies is negligible for a small value of 'f' (fraction).

In the following, the choice of an appropriate value of k will be referred to as an 'action' or a 'decision'. Expanding or shortening the duration of a frame presentation introduces a discontinuity. Let dc_{ik} denote the discontinuity that is incurred when the next frame is presented with duration DF_k and the current buffer occupancy is 'i' , then

$$dc_{ik} \overset{\Delta}{=} \left|DF_k - F_d\right| + R_v.I \qquad 0 \leq i \leq N$$

Here R_v is a random variable that adds to dc_{ik} with the effect of buffer underflows. R_v represents the time interval between a buffer underflow instant and the next arrival instant, 'I' is the indicator function, the distortion of play out is defined as

$$DP_{ik} \overset{\triangleleft}{=} dc_{ik} + \bar{l}_{ik}.T$$

Where $\bar{l}_{ik}$ is the expected number of lost frames due to buffer overflow over the next presentation interval.

4. Adaptive Buffer Management

A buffer has been introduced at the source that queues packets from the encoder and de queued packets for transmission, this buffer is named as adaptive buffer. It is used if the encoder can't adopt as quickly as the network conditions change and differentiate the packets according to their priority, send only the most important packets in the available bandwidth. The video encoder produces packets with different priorities like I,P,B. video codes would like the network to treat each packet differently to obtain optimal performance. The output of a video encoder is at a variable bit rate, when transmitted over a traditional constant bit rate network. The video rate is adaptively controlled at the source by encoder so as to maintain a constant transmission rate. It is depicted in the figure 13.

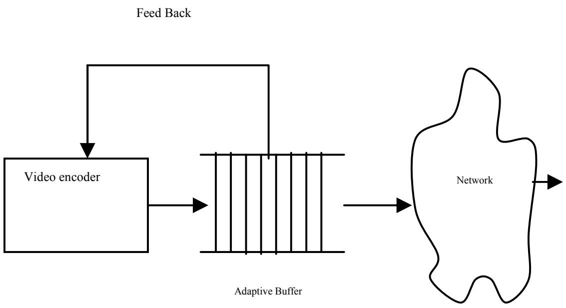


Figure 13: Adaptive Buffer for constant bit rate network

The raw video stream is fed into an encoder and then the output is sent into the buffer that is drained at a constant rate. In order to keep the constant drain rate at the buffer without overflowing or under flow the buffer, the encoder's output should be adaptively controlled.

The buffer data level is used as a feed back signal to control the video output rate from the encoder and the rate adaptation is achieved by adjusting certain encoding parameters, like quantization level, the frame rate and the pixel resolution.

In the second case, the variable bit rate network, in which the feed back is derived from the network. The video rate then adopts according to the changing conditions in the network. In this dissertation the focus is on second case, I.e. on calculating the available bandwidth, adjusts the bit rate according to estimated available bandwidth and feedback information of buffer occupancy. It is depicted in figure14.

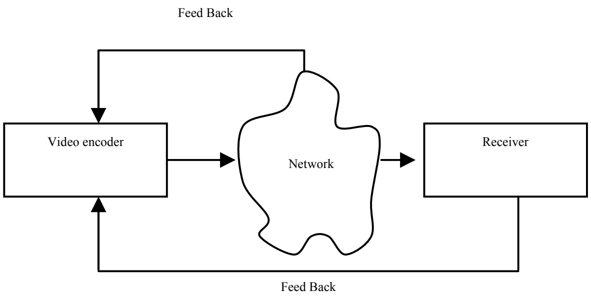


Figure: 14: Rate control according to feed back information

A threshold point within the adaptive buffer is introduced. The low priority packets will be dropped whenever the buffer occupancy reaches to a threshold point.

4.1 Algorithm for source buffer management:

Step: 1 for each packet arrival do

Calculate the queue size (q_{size})

Step: 2 if $q_{size} > Max_{queue}$ then drop the packet.

Step: 3 otherwise if ($q_{size} > q_{thresh}$) **and**

$$\left(\frac{r_{en} - r_{net}}{r_{en}} > 0 \right)$$

Then drop low priority packets,

Otherwise add the packets to the queue,

Here r_{en} is the encoding rate or rate of the application layer, r_{net} is the rate of transport layer or transmission capacity of network. The flow will loose the packets at the rate ($r_{en} - r_{net}$) packets per second, if the current network rate r_{net} is less than the application layer rate (r_{en}) and the buffer occupancy reaches to a threshold point. Choosing the buffer threshold is very important for performance. If the threshold value is small, it will lead to unnecessary packet drops, while having a threshold greater than the receiver buffer; it will lead to large delay and jitter. Threshold value depends on the decoding rate, network capacity and the buffer at the receiver.

References

[1] MAY T.W. IP,WILFRED W.K. LIN,ALLAN K.Y. WONG, THARAM S. DILLON, DIANHUI WANG ,*"An Adaptive Buffer Management Algorithm for Enhancing Dependability and Performance in Mobile-Object-Based Real-Time Computing"* Springer Berlin, Volume 3280/2004.

[2] KOHEI FUJIMOTO, SHINGO ATA, MASAYUKI MURATA, *"Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications"*. Globe com 02, IEEE global tele communications , 2002.

[3] JUAN MIGUEL DEL ROSARIO AND GEOFFREY FOX ,*"Constant bit rate network transmission of variable bit rate continuous media in Video-On-Demand servers"* ,Citeseer 2003.

[4] FENG, W.-C. AND REXFORD, J., *"A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video"*. In *IEEE INFOCOM* (Kobe, Japan,) Apr. 2005.

[5] JOHN G. APOSTOLOPOULOS, WAI- TIAN TAN, SUSIE J. WEE, *"Video Streaming: Concepts, Algorithms,and Systems "*, CiteSeer 2002.

[6] BOUFFIOUX - PHILIPS LEUVEN PHILIPS , *"Audio & Video Compression and its Application"* , Alain Digital Systems Laboratories CiteSeer 2002.

[7] JEAN-CLAUDE DUFOURD, OLIVIER AVARO (STREAMEZZO) CYRIL CONCOLATO , *"The MPEG*

Standard for Rich Media Services", published in *IEEE* Multimedia, vol .,1 ,2004.

[8] "*MPEG-4 The new standard for multimedia on the Internet*", powered by Apple QuickTime, Apple Computer Inc,. 408-996-1010, 2002..

[9] *"MPEG-4 Natural Video Coding – Part II* ", (2004) EPFL Compaq Hitachi Ltd. CH-1015 Lausanne Cambridge, MA 02139 Tokyo

[10] T. EBRAHIMIA, F. DUFAUXB and Y. NAKAYAΓ BARON, S. N. and WILSON, W. R., *"MPEG Overview,"* SMPTE J., 103:(391-394pp), June 2004.

[11] HUAHUIWU, MARK CLAYPOOL AND ROBERT KINICKI , *"Guidelines For Selecting Practical Mpeg Group Of Pictures",* EuroIMSA 2006 , Feb 13-15 Innsbruck, Computer Science Department Worcester Polytechnic Institute Worcester, MA 01609.

[12] D. WU., *"Streaming Video over the Internet: Approaches and Directions",* 2001.

[13] N. FEAMSTER, D. BANSAL AND H. BALAKRISHNAN, *"On the Interactions Between Layered Quality Adaptation and Congestion Control for Streaming Video"*. In 11th International Packet Video Workshop (PV2001), Kyongiu, Korea, April 2001.

[14] S. FLOYD AND K. FALL., *"Promoting the Use of End-to-End Congestion Control in the Internet*." *IEEE/ACM* Transactions on Networking, August 1999.

[15] S. M. W. GROUP, *"Synchronized Multimedia Integration Language (SMIL)* "1.0 Speci_cati Portal , The ACM digital Library, (121-123pp), 2003.

[16] NICK FEAMSTER AND HARI BALAKRISHNAN, *"Packet Loss Recovery for Streaming Video",* M.I.T. Laboratory for Computer Science Cambridge, MA, published in SiteSeer www://nms.lcs.mit.edu/projects/videocm/, 2003.

[17] DINESH C. VERMA HUI ZHANG DOMENICO, *"Delay Jitter Control for Real-Time Communication in a Packet Switching Network"* , Ferrari Computer Science Division, papers of Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 2004.

[18] DERP.M RESEARCH PAPERS, *"Delay-jitter control in multimedia applications",* Department of Computer Science and Engineering, Arizona, 2004.

Authors:

Dr. P. Bala Krishna Prasad received M.Tech Degree from Andhra University in 2000. He received Ph.D in Computer Science & Engineering from ANU in 2007. He is having 14 years of Teaching Experience in varies Engineering Colleges. At present he is working as Principal for St. Ann's Engineering College. He is renowned author in the field of Computer Science. He has Authored 6 Textbooks and Published number of papers. He taught more than 20 subjects for engineering students. His research interest includes Computer Networks, Streaming Media. He is a Member of ISTE,IETE.

Mr. G. Murali, received his Master's degree in Computer science in the year 2001. At present he is working as an Associate Professor in the Department of computer science, St.Ann's college of engineering and technology. He is having 7 years of teaching experience and 1 year of software industry experience. He is a trained administrator of Oracle 10g Database.

Mr. G.Guru kesava dasu received M. Tech (I.T) Degree in 2003 and received M.E (C.S.E) Degree from ANNA University in 2008. He is having 9 years of Teaching Experience in varies Engineering Colleges. At present he is working as Associate Professor in St. Ann's Engineering College, Chiral. He is a certified professional of IBM Rational Rose from Rational University.

Mr. k. Siva Krishna Rao received M.E (C.S.E) degree from ANNA University in 2005. He is doing Ph.D in S.V.University, Tirupati. He is having 7 years of Teaching Experience in varies Engineering Colleges. At present he is working as Associate Professor in P.N.V. Vijay Institute of Engineering & Technology, Repudi, Guntur (DT).