

A New Technique for Removing Jitter in Network Multimedia Communication to Achieve Guaranteed QoS over Packet Network

Shyamalendu Kandar
Assistant Professor,
Computer Sc. & Engineering
Haldia Institute of Technology
Haldia, West Bengal, India

C.T.Bhunia
Director, Bengal Institute of
Technology & Management,
Bolpur, West Bengal India

A.Chaudhuri
Professor
Computer Sc. & Engineering
Jadavpur University, West Bengal,
India

ABSTRACT:

Multimedia data are sensed by human. These types of data are delay intolerable but error tolerable to some extent. Two important parameters that degrade the quality of service (QoS) of multimedia services are Skew and Jitter. Achieving guaranteed Quality of Service (QoS) of multimedia service is a great research challenge. Different researchers proposed different techniques for removing jittering and skew to achieve guaranteed QoS. Accelerating and De accelerating technique, Buffer size estimation technique and Clock time synchronization between sender and receiver are some suitable techniques. But all of the techniques have some advantages and disadvantages. In this current work of the drawbacks of the available techniques for removing jitter is described and new techniques for removing jitter is proposed. A comparison of the available techniques with the proposed technique is made to show the superiority of the proposed technique than the already available techniques.

KEYWORDS

Jitter, QoS, Protocol, Clock Synchronization

1. INTRODUCTION

Multimedia data are integration of one or multiple media components like Text, Graphics, Audio, Video and Animation.[1][2]. Multimedia services are made of two distinct services: Time dependent services and Time independent services, respectively often known as constant bit rate (CBR) services and variable bit rate (VBR) services.[3] In CBR services, in order to achieve some guaranteed QoS, two issues that are paramount importance are: jitter and skew[4][5]. Jitter refers to variable delays caused during transportation through network between packets of a particular service and skew refers to the variable delay between the two (or more) corresponding packets of two (or more) services during transportation in the network. Jitter occurs due to the variable phase delay among the packets of a particular service (May be Audio or Video) from source to destination. The phase lag between packets differs from the source end to the destination end because the total transfer delay varies from packet to packet. This phase delay occurs due to Propagation Delay, Transmission Delay, Queuing Delay or Node processing delay at the time of transmission of packets in the network. [6][7] Among these first one is constant and the next three are variable. Due to jittering problem, a

sending voice “ I shall go home” may be received as “ I shallgo home”. to the transmitter, the phase delay between “i” and “shall” has increased and that between “shall” and “go” has reduced to zero at the receiver.

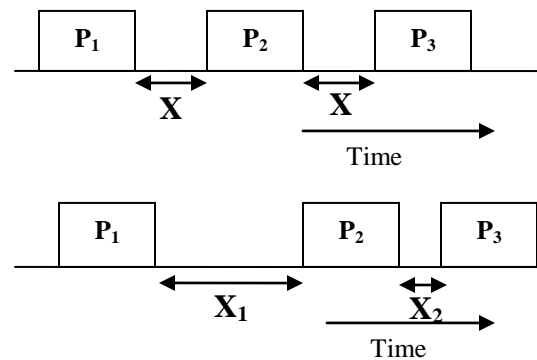


Fig 1: Occurrence of Jitter

In the above given figure the phase lag between P₁ and P₂ and P₂ and P₃ are X at the transmitter side. At the receiver side, due to the delay in network, the phase lag becomes X₁ and X₂ respectively. If X₁ ≠ X and /or X₂ ≠ X, then jitter occurs. Jitter can be of positive or negative. If Jitter is zero then there is no jitter. The jitter is calculated at the receiver side. It is not a suitable technique to calculate the jitter between each two packets, but at the receiver side mean jitter is calculated after a certain interval. The mean jitter calculated, could be limited by increasing the bit rate capacities of the link and by adopting efficient routing technique among others. Yet it is seen that jittering effect can not be solved so simply. Jittering can be minimized in the receiver side but can not be totally removed.

Section 2 describes and finds merits and demerits of the existing techniques for minimizing jitter. Section 3 describes the basic idea of the new protocol. Section 4 elaborately discusses the new protocol for minimizing jitter. Section 5 describes the experiment process for the implementation of the new protocol. Section 6 gives the results and Section 7 draws the conclusion and Section 8 points out on the future scope.

2. ALREADY EXISTING TECHNIQUES

2.1. Accelerating -De accelerating Technique

There are several techniques to reduce the affect of the problem of jitter. One such technique is known as accelerating and de

accelerating.[8] In fact the problem of jitter is due $(D_{i+1} - D_i)$ which is finite and a variable. Here, D_{i+1} and D_i are both variable quantities and represent respectively the total transfer delay of $(i+1)$ th packet and i th packet. To avoid the jittering effect, it is required that $D_{i+1} - D_i = 0$. In the accelerating and de accelerating technique, at the receiver end a variable delay (say W_i for i th packet) is caused to each packet such that $D_i + W_i = K$, a constant for all packets (i.e for $i=0,1,2,3, \dots$) before delivery of the packets to the terminal equipment for play back. By the process, the variable delay caused by the network between two successive packets is made zero as $(D_{i+1} + W_{i+1}) - (D_i + W_i) = 0$. This ensures that the phase delay between packets at the transmitter remains same at the receiver. The scheme is illustrated in the Table 1. As illustrated in the table, the success of the technique depends on the choice of K .

Table1: Results Of Accelerating, De-Accelerating Technique

Packet No	Instant of releasing a packet at transmitter (x_i) in ms	Variable delay with which the packet reaches the receiving node in ms (D_i)	Variable delay (W_i) caused at the receiving buffer (100- D_i) in ms (K has been chosen as 100 ms)	Delay with which the packet is delivered to the terminal device (x_i+100 ms)
1	0	80	20	100
2	10	70	30	110
3	15	85	15	115
4	25	100	0	125
5	30	110	-10	130

Packet-4 is the marginal case. Packet-5 is the failed case. This is described by the following graph in figure 2.

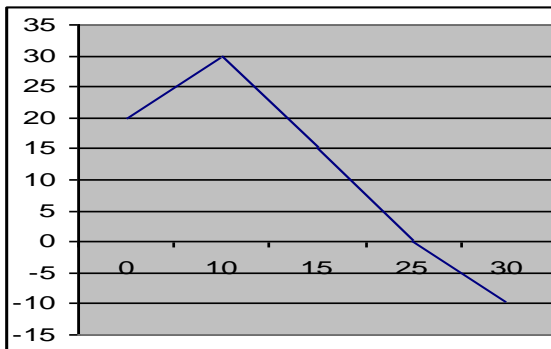


Fig 2: Graph of Variable delay Vs Transmitting Time

Both could have been avoided had the constant K been chosen more than 110 ms in this case. So the success of the technique depends on the choice of fixing K .

2.2. Buffer Size estimation Technique

The investigation for selecting the value of K amounts to investigating the buffer size required at the receiver for some guaranteed QoS. In a recent research work [6], a research in the

direction of finding buffer size was reported. The work was based on the observation [8] that the multimedia streams may be seen in synchronization if the jitter and the skew are kept within a limit. The work also assumed a model of master and slave for multimedia. One stream is taken as master and other streams as slaves. The work then defined that to meet the users' requirement of QoS, the following specifications be used:

$$P(|J_{ri}| > J_{max}) \leq \epsilon_m$$

$$P(|S_{ri}| > S_{max}) \leq \epsilon_s$$

Where $P(\cdot)$ refers to the probability that the residual jitter and skew ($|J_{ri}|$ and $|S_{ri}|$) of the i th media unit increase the maximum admissible values (maximum admissible residual jitter = J_{max} and maximum admissible residual skew = S_{max}). The residual refers to the amount that occurs after applying synchronization at the receiver.) is limited by $\leq \epsilon_m$ and $\leq \epsilon_s$.

For a multimedia where audio is master and video is slave, the research calculated the buffer for master stream, b_m and that for slave stream b_s as follows:

$$b_m = 2((\sigma_m^2/\epsilon_m)^{0.5} - J_{max}) \dots\dots(1)$$

$$b_s = 2((\sigma_m^2 + \sigma_s^2)/\epsilon_s)^{0.5} - S_{max} \dots\dots(2)$$

where jitter expectation of audio stream = 0, jitter variance of audio stream = σ_m^2 , jitter expectation of video stream = 0 and jitter variance of video stream = σ_s^2 . The work did not elaborate to the critical nature of the results obtained.

It is found based on the results (equ 1 & 2)[9], three situations of different QoS obtained may be defined as:

a) worst case analysis when $\epsilon_m = \epsilon_s = 1$; and then

$$b_m = 2((\sigma_m^2)^{0.5} - J_{max}) \text{ and}$$

$$b_s = 2((\sigma_m^2 + \sigma_s^2)^{0.5} - S_{max})$$

b) average case analysis when $\epsilon_m = \epsilon_s = 0.5$; and then

$$b_m = 2((2\sigma_m^2)^{0.5} - J_{max}) \text{ and}$$

$$b_s = 2(2(\sigma_m^2 + \sigma_s^2)^{0.5} - S_{max})$$

c) best case analysis when $\epsilon_s = \epsilon_m = 0$;

$$\text{and then } b_m = b_s = \infty$$

Further, in first two cases the required buffer would be finite to meet with the QoS only when $(\sigma_m^2)^{0.5} > J_{max}$ and $(2\sigma_m^2)^{0.5} > J_{max}$ respectively.

In both the existing techniques, the estimate of buffer size is the key challenge. Table 2 further illustrates the issues. Two important observations are then obtained: (a) equation(1) is invalid when $\sigma_m < 250$ ms and (b) even when $\sigma_m < 250$, the requirement of the buffer for slave will be finite. Thus, for maintaining required QoS, the buffering of slave packet appears more important than that of the master.

Table2: Result Analysis of Buffer Size Estimation Technique

$J_{max} = 250$ ms $S_{max} = 80$ ms		Worst Case	Average Case	Best case
$\sigma_m = 250$ ms	b_m	0	207 ms	∞
$\sigma_s = 80$ ms	b_s	365 ms	582 ms	∞
$\sigma_m = 500$ ms	b_m	500 ms	914 ms	∞
$\sigma_s = 160$ ms	b_s	890 ms	1325 ms	∞

It is established that calculating buffer space for jittering and skew to provide a guaranteed QoS is the function of the very characteristics and the spread of the jittering and that of the skew. The concept of master and slaves services has determined that the buffer space of slave is more than that of the master. This has been established in three possible cases of worst, average and best. However none of the above techniques offers substantial reduction of jitter. We propose a technique of max packet/cell for the purpose of reducing skew.

2.3. Clock time synchronization technique

Jitter occurs due to the variable phase delay of the packets of a particular service between sender and receiver. In receiver side there is no way to measure the jitter as the receiver does not know the sending time difference of two packets from sender. In a current research work a focus was put on the clock time synchronization between sender and receiver. From sender side a packet with the reference clock time of the sender let A_{ref} is sent to the receiver. Upon receiving this packet, the receiver synchronizes its clock with the sender. Sender periodically sends these packets in every T_{ref} second. Within a fixed interval let T , the packets of the multimedia data is sent from sender. In the receiver side the arrival times of all packets (N) received between two successive packets carrying the reference clock value are registered. Let for i -th packet the registered time of receive at the destination is A_{ai} . If there is no jitter the arrival time of i th packet at the receiver is

$$A_{ti} = A_{ref, j} + iT \quad \text{-----}(3)$$

$A_{ref, j}$ is the reference time of the sender carried by j -th packet.

Jitter for i -th packet is

$$J_i = A_{ti} - A_{ai} \quad \text{-----}(4)$$

From the equation (3) and (4) calculated mean jitter (μ_k) is

$$\mu_k = \frac{1}{N} \sum_{i=1}^n J_i \quad \text{-----}(5)$$

The value of μ_k can be positive, negative or zero.

Jitter may occur among the packets carrying reference clock value of sender. Theoretical arrival time ($A_{th, ref, j}$) of such j -th packet is

$$A_{th, ref, j} = A_{a, ref, j-1} + (N+1) T \quad \text{-----}(6)$$

The corrected theoretical arrival time of such j -th packet is

$$A_{c, th, ref, j} = A_{th, ref, j} - \mu_k \quad \text{-----}(7)$$

By this process the jitter between two packets carrying reference clock value of sender is reduced. The jitter of the packets carrying reference clock value is

$$J_{ref, j} = A_{c, th, ref, j} - A_{a, ref, j} \quad \text{-----}(8)$$

This jitter can be minimized at the time of receiving the next packet containing reference clock value of sender.

This process is described in the Figure 3.

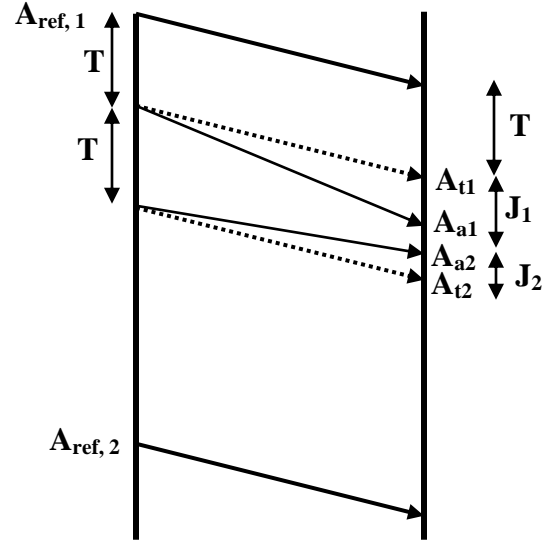


Fig-3: Clock Time Synchronization Technique

Advantages:

Let, total number of packets sent is n . Let, in between two packets carrying reference clock value N number of packets are sent. So, total number of packets carrying reference clock value is n/N . If no such packets were sent, the jitter would be

$$\mu_{gen} = 1/n \sum J_i \quad \text{-----}(9)$$

Now the jitter is theoretically zero for sending the total data in n packets.

Disadvantages:

- i) Jitter of the N packets, received between two successive packets carrying reference clock value is measured and an attempt is proposed to minimize the jitter. But jitter in between two packets of these N packets, till exist.
- ii) Every time the reference clock value of the receiver is modified upon receiving a packet with reference clock value from sender.
- iii) If video conferencing is done then both of them will work as sender and receiver. Then the reference clock value will create a problem in every step.
- iv) Number of overhead increases due to the packets with reference clock value.

3. PROTOCOL TO REALIZE THE BASIC IDEA

To realize the basic idea proposed we propose a protocol as shown in Fig 4. This is based on Automatic Repeat Request (ARQ) protocol used in error control in networks.[6] The protocol is based on the feed back from the receiver. Initially transmitter transmits a packet with reference clock value of the sender. Upon getting the packet, the receiver synchronizes its clock with the clock value of sender attached with the packet. After that the sender sends the packets of the multimedia data of size N in each T time unit interval. [10][11]

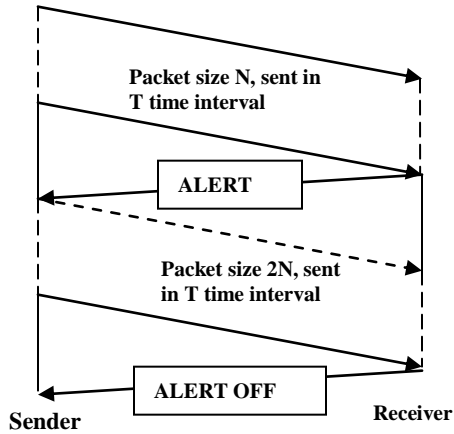


Fig. 4: Description of the protocol

In the receiver side the arrival times of all packets (n) received between two successive packets carrying the reference clock value are registered and the mean jitter is calculated according to equation-5. If the mean jitter increases the permissible limit, receiver sends an ALERT signal to the sender. On receiving the ALERT signal the transmitter increases the size of packets and/or increases accordingly the time span of sending the packets containing multimedia data till it gets ALERT OFF signal from the receiver. The receiver sends an ALERT OFF signal only when the mean jitter decreases the permissible limit because of the increased packet size and increase of time span of sending packets with multimedia data. On receiving ALERT OFF signal, transmitter switches back to original packet size and original time span of sending packets with containing multimedia data. The process repeats accordingly. In the new protocol technique three different techniques are proposed

- a. Packet size 2N and n/2 packets are sent, each in T time interval.
- b. Packet size 2N and n/2 packets are sent, each in 2T time interval.
- c. Packet size N and n packets are sent, each in 2T time interval.

In the following section only the first technique is discussed and obtained results are shown.

4. ELABORATION OF THE NEW PROTOCOL TO MINIMIZE JITTER

Packet size 2N, n/2 packets are sent each in T time interval

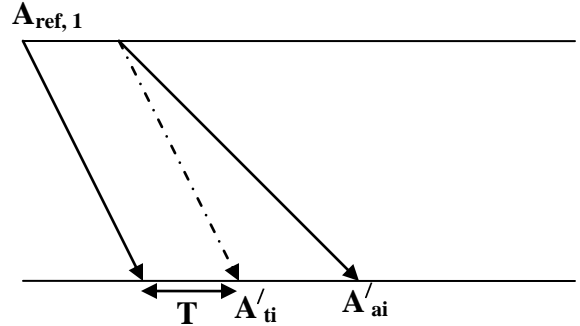


Fig.5. Time Diagram of Packet size 2N, n/2 packets are sent each in T time interval

Here

$$A'_{ti} = A_{ref,j} + iT \quad \dots\dots\dots(10)$$
 Where $i = 1 \dots\dots n/2$

Jitter for ith packet is

$$J'_i = A'_{ai} - A'_{ti} \quad \dots\dots\dots(11)$$

Where $i = 1 \dots\dots n/2$

$$\mu'_k = 2/n \sum_{i=1}^{n/2} J'_i \quad \dots\dots\dots(12)$$

Jitter occurs due to Propagation delay, Transmission delay, Queuing delay and / or node processing delay. Among these Propagation delay (P_d) is constant. Transmission delay (T_d) increases due to larger packet size [$A'_{ai} > A_{ai}$ in general]. Queuing delay (Q_d) decreases due to less number of packets. Node processing delay (N_d) decreases due to less number of packets.

For n packets each with size N

$$\text{Delay } (D_n) = P_d + T_d + Q_d + N_d \quad \dots\dots\dots(13)$$

For n/2 packets each with size 2N

$$\text{Delay } (D_{n/2}) = P_d + 2T_d + Q_d/2 + Q_p/2 \text{ in general } \dots\dots\dots(14)$$

In this technique $J'_i > J_i$ but $J'_i \neq 2J_i$

For n/2 number of packets $\mu'_k < \mu_k$

5. DISCUSSION OF THE EXPERIMENT

The experiment is done by socket programming of Java. One machine is set as sender and another machine is set as receiver. Two socket programs named server socket and client socket are created. An allowable jitter value is set. Server socket is run in the machine marked as sender, and client socket is run in the machine marked as receiver. At the time of initiating

communication the clock time at that instant of the sender is sent to the receiver. Upon receiving the time, the receiver clock time is set with the sender's clock. Data packets of the multimedia data are sent from sender to receiver in a fixed time interval. Upon receiving the packets, the receiver calculates the time difference say J_i between two consecutive packets in respect to receiver's clock value. After a particular number of packets say n acceptance the mean jitter is calculated according to equation 5. If the mean jitter is greater than allowable jitter, an ALERT message is sent by the receiver to the sender.

On receiving the ALERT message from the receiver the packet size is doubled and number of packets within a particular time is minimized to half by the sender. If after receiving $n/2$ number of packets the mean jitter is less than the allowable jitter, for a specified time span; an ALERT OFF message is sent by the receiver to the sender. On receiving the ALERT OFF message the sender switches back to its original packet size and original time span of sending packets with containing multimedia data. The process repeats accordingly.

6. RESULT AND ANALYSIS

Packet Size: 1000 byte

Number of Packets: 6

Allowable Jitter: 1650 ms

Sending Time difference between two consecutive packets : 1000ms

Sender Side

File Name	abc.wma
Time	1261481209250
Sent Packet 1	Size 1000
Sent Packet 2	Size 1000
Sent Packet 3	Size 1000
Sent Packet 4	Size 1000
Sent Packet 5	Size 1000
Sent Packet 6	Size 1000
Message From receiver	JITTER ON

Receiver Side

FTP Server Started on Port Number 5217

Waiting for Connection

FTP client Connected

SEND command received.....

Received Packet 1	Size 1000
Received Packet 2	Size 1000
Received Packet 3	Size 1000
Received Packet 4	Size 1000
Received Packet 5	Size 1000
Received Packet 6	Size 1000
Mean Jitter	1658
Message Sent to Sender	JITTER ON

Sender Side

Time	1261481215312
Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Message From receiver	ACK

Receiver Side

Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Mean Jitter	1343
Message Sent to Sender	ACK

Sender Side

Time	1261481218390
Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Message From receiver	JITTER OFF

Receiver Side

Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Mean Jitter	1343
Message Sent to Sender	JITTER OFF

Sender Side

Time	1261481221474
Sent Packet 1	Size 1000
Sent Packet 2	Size 1000
Sent Packet 3	Size 1000
Sent Packet 4	Size 1000
Sent Packet 5	Size 1000
Sent Packet 6	Size 1000
Message From receiver	ACK

Receiver Side

Received Packet 1	Size 1000
Received Packet 2	Size 1000
Received Packet 3	Size 1000
Received Packet 4	Size 1000
Received Packet 5	Size 1000
Received Packet 6	Size 1000
Mean Jitter	1648
Message Sent to Sender	ACK

Sender Side

Time	1261481227532
Sent Packet 1	Size 1000
Sent Packet 2	Size 1000
Sent Packet 3	Size 1000
Sent Packet 4	Size 1000
Sent Packet 5	Size 1000
Sent Packet 6	Size 1000
Message From receiver	JITTER ON

Receiver Side

Received Packet 1	Size 1000
Received Packet 2	Size 1000
Received Packet 3	Size 1000
Received Packet 4	Size 1000
Received Packet 5	Size 1000
Received Packet 6	Size 1000
Mean Jitter	1659
Message Sent to Sender	JITTER ON

Sender Side

Time	1261481233587
Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Message From receiver	ACK

Receiver Side

Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Mean Jitter	1344
Message Sent to Sender	ACK

Sender Side

Time	1261481224515
Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Message From receiver	JITTER OFF

Receiver Side

Sent Packet 1	Size 2000
Sent Packet 2	Size 2000
Sent Packet 3	Size 2000
Mean Jitter	1345
Message Sent to Sender	JITTER OFF

7. FUTURE SCOPE

To minimize jitter only one technique is discussed in this section. But jitter can be minimized by increasing the time span of the modified packets. It is a future direction of this work to find that suitable time span where the jitter is mostly minimized. For video conferencing it's a challenge to synchronize the clock time between sender and receiver, it can be done by providing a global clock. The implementation of the technique is a research challenge.

8. CONCLUSION

The new technique for minimizing jitter is better than the existing techniques. For accelerating and de-accelerating technique a variable delay is added at the time before delivery of the packets to the terminal equipment for play back. Here the delay is variable. And the receiver needs to continuously monitor for the marginal cases where delay is zero. Most of the cases fail case occurs. The success of the technique depends on the selection of the variable delay at continuous basic. If the delay is large enough, then the success rate is maximum, but it can not be called life broadcasting as delay is more.

For adding variable delay, the system needs a buffer to store. If the delay is more then required buffer size is also large. Again in the case of video data communication, where there are two multimedia components audio and video, the system needs to

store both of the components to reduce jitter and reducing the possibility of occurring skew. From the results obtained in Table 2 it is clear that this technique can not reduce jitter to a substantial limit.

Clock time synchronization technique also fails to remove jitter between two consecutive packets. But in the new technique the jitter is reduced to a substantial limit by making packet size double and number of packets half. JITTER ON and JITTER OFF message from receiver helps to switch on switch off the conversion process. Upon getting JITTER OFF message the sender switches back to original packet size and original packet number to reduce the load from the network nodes.

REFERENCES

- [1] Ralf Steinmetz and Klara Nahrstedt, *Multimedia: Computing, Communications and Applications*, Pearson Education Asia, 1995
- [2] Ranjan Parekh, "Principles of Multimedia", Tata McGraw Hill, 2006
- [3] C T Bhunia, *Multimedia Services with Guaranteed QoS over Packet Networks/ Internet*, J IE(I), CP, Vol 87, Nov'06, pp 38
- [4] Jerry D Gibson, *Multimedia Communications*, Academic Press, 2001
- [5] John F Koegel Buford, *Multimedia Systems*, Addison Wesley, 2000
- [6] C T Bhunia, *ARQ Techniques- Review and Modifications*, J IETE Technical Review, New Delhi, Vol 18, No 5, pp 381-401
- [7] C T Bhunia, S.Kandar, M.Barman , *Technique in removing skew and analysis*, Proc National Conf. RAFTIT, 2008, Tech Pub, pp 184-189,
- [8] R Steinmetz, *Human Perception of jitter and media synchronization*, IEEE J Select Areas Commn, Vol 14, 1996, pp 61-72
- [9] Yan Xu et al, *Calculation and Analysis of Compensation Buffer Size in Multimedia Systems*, IEEE Communication Letters, Vol 5, No 8, 2001, pp. 355 –357
- [10] S.Kandar, C.T.Bhunias, "A New Protocol for Minimizing Jitter for Guaranteed QoS in Network Multimedia Communication" ACE 2010, June 21 -22, 2010, Bangalore, Published in IEEE explore.
- [11] C.T.Bhunias, S.Kandar, *Review of the proposed Techniques for removing jitter*, International Journal BITM transaction on EECC,pp- 41 - 45