Modelling of VoIP Based Tele-teaching System: A Petri Net Based Approach

Animesh Dutta [1], Ranjan Dasgupta[2] and Swapan Bhattacharya[3]

[1]Department of Information Technology, National Institute of Technology Durgapur, West Bengal, India

[2] Department of Computer Science & Engg, National Institute of Technical Teacher's Training & Research Kolkata, West Bengal, India

[3]Department of Computer Science & Engg, Jadavpur University Kolkata, West Bengal, India

Abstract

In this paper we propose an algorithm called Interactive Group Synchronization to reduce the chance of audio speech overlapping in VoIP based conferencing. We also extend the capability of colored petri nets by introducing time parameter in it. The propose TCPN (Time Colored Petri Nets) tool is applied here to model the interactive group synchronization algorithm in teleteaching system and then verify and check the correctness of the model.

Keywords— *VoIP, Tele teaching, Synchronization, Algorithm, Modelling, Verification, Petri Nets.*

1. INTRODUCTION

The customers are familiar to get the consistent audio quality and high reliability provided by the traditional circuit switch network. The challenge of transporting multimedia data through a single converged IP network and cost saving therein motivates the adaptation of Voice over IP (VoIP) technology [2].

There are several steps involved in sending real time audio/video data over the Internet. First, the analog audio/video input signals of the sender are captured. Then these input signals are encoded at some sampling frequency. The resulting data is partitioned into frames representing signal evolution over a specified time period. Each frame is then encapsulated into a packet and sent using a transport protocol (usually UDP) towards the destination. At the receiver end the received packets are first decoded, then by using some synchronization technique the audio frames and video frames are synchronized and played out.

Teleteaching [3, 4] has become a major tool to impart education in many countries, where the student doesn't have to think about the geographical boundaries while going for studying or learning. Teleteaching is becoming an important tool, in areas where lifelong learning is necessary i.e. Companies, universities etc. Content of e-learning can range from subjects to soft skills, from tutorials to even real time audio/video streaming using VoIP technology. The materials used in this learning technique include text files, multimedia etc. Teleteaching has in it the provisions which cater to the needs of the teachers and students alike, which enhances web learning experience. The same resources can be combined with others to set up different courses. There are some synchronization issues involved in this technology.

Intra-stream synchronization refers to the temporal relationship between the media units (MUs) of one time-dependent media stream. In [5], there is a survey of algorithms providing this type of synchronization. The intra stream synchronization is tackled by buffering method.

Inter-stream Synchronization refers to the synchronization between media streams. In remote user speech, the synchronization between the user's audible words and the associated movement of the lips, referred to as Lip synchronization or lip-sync, is an example of this type of synchronization. This is also achieved by buffering technique.

Multipoint or inter-destinations synchronization [5,6,7] involves the synchronization of the play- out processes of same streams in different receivers, at the same time, to achieve fairness among the receivers. We can cite the example of tele teaching applications in which a teacher could send (multicast) a video sequence (documentary or film—stored content stream) and, during the session, sometimes the teacher could make occasional comments about the video (live content stream).Network quizzes are other examples, in which the same multimedia question must appear at the same time to all the participants to guarantee fair play. In the first example, a simultaneous play out of the streams is important for both stored content and live content streams. Even if we only send the video stream (documentary or film), each video MU (frame) should be played simultaneously in all the receivers (students) and then the students could comment on the video content with other students. The multipoint synchronization is achieved by a group of receiver by skipping and pausing the playout process of each receiver.

Interactive group synchronization can be seen as multiple users' synchronization. This synchronization task aims to avoid semantics incoherence in the dialogue between all users. This situation happens, when a large number of distributed nodes participating in a conference session and more than one audio stream generates from more than one node at the same time and reaches the mix streams to the listeners' node, it becomes a non understandable audio. So audio stream generated at different nodes must be synchronized with each other. We have already proposed an algorithm for achieving interactive group synchronization [1].

Petri net provides useful mathematical formalism for modeling concurrent system and their behaviors. An example of High level Petri Nets is Coloured Petri Nets [8], which is a graph oriented language for design, specification, simulation and verification of systems. It is in particular well-suited for systems that consist of a number of processes which communicate and synchronize. It's a combination of Petri Nets and programming language where Petri Nets control the structures, synchronization, communication and resource sharing and data manipulation are described by functional programming language.

With this background we extend our previous work[1] of interactive group synchronization (IGS Algorithm) by modelling it through time coloured Petri nets and explain with example.

2. RELATED WORK

In the first section we described some of the tools already developed by different organization for teleteaching/video conferencing and in the second section of this paper describe some synchronization algorithms.

Forum [9] was developed by the Texas A&M University of Laredo. FORUM is a web-based application and its services are available through common web browsers. The FORUM server needs a Novell network server and runs over the Microsoft Windows Platform. FORUM offers the following capabilities: Group authoring and documents processing, file server service, online text-based chat, multiple sessions, predefined lesson structure for the students.

Virtual-U [10] offers capabilities for design, creates and transmits educational material over the web. It is based on the results of human and computer interactions during network-based distance learning. It offers capabilities for videoconference, creation of educational material and administration of student's grades.

Symposium [11]is a web-based application, which offers both synchronous and asynchronous distance learning. Someone can access the service of the Symposium through common web browsers. It offers capabilities for delivery of educational material through the web, collaboration between virtual teams, asynchronous distance learning, synchronous communication between the participants and creation of educational material.

Web-CT [12] was developed by the Computer Science department of British Columbia University of Canada. It is a web-based application, which offers enhanced capabilities for creation of educational material and management of the education procedure.

First Class [13] was a client-server application that offers capabilities for e-mail exchange, threaded discussion and remote access. The last version of First Class is a web-based application. It was used by the Open University in United Kingdom for distributed learning. It offers capabilities for asynchronous learning, creation of educational material and administration of educational procedure.

ooVoo [14] is a video-conferencing software available for Windows and Mac. After a quick registration, users are able to communicate with people through text-chat, video-audio conference (up to six people) and also to record video messages. User can also share up to 20 files at once to as many contacts as you want, up to 25 MB per file. Still in beta, it is free to download and use.

Adobe Connect Now [15]is a part of the new Acrobat.com of online collaboration tools. This one lets users create online meetings where people can do videoconferencing, VoIP conversations, whiteboard, share files, chat, and share your screen. They can change a person's role at anytime, and move the activity pods as they wish during the meeting. The service is free to use, as the rest of the suite, after a simple registration.

VSee [16]is a free videoconferencing and application sharing service and it allows users to talk with multiple people on their computer. They can remotely edit and annotate documents, share applications and desktops, transfer files, record and share videos, pan, tilt, and zoom remote cameras. Vsee is free to use for an unlimited number of people.

SightSpeed [17] is a cross-platform videoconferencing system that is light, performing and cheap. Users can use it to have video calls with up to 9 people, text-chat with them, share files, record their sessions and send video messages. Free for two people or $9.95/month for 4 people and unlimited video storage. Also, a new web based version of the program is available.

PalBee.com [18]is a video conferencing system, which has just launched a new version of the product. It lets user set up video meetings with up to 10 people, who can all whiteboard, upload PowerPoint presentations, record for one hour anything that happens in the conference and publish it as a video on YouTube. The service is completely free to use.

The subject of Teleteaching in distributed IP network has engaged researchers all over the world. Many solutions have been proposed for distance learning and collaboration over the Internet. Indicative commercial tools for collaboration over the Internet are Microsoft NetMeeting and WhitePine Enhanced CU-SeeMe (WhitePine, http://www.wpine.com). Various methods have been proposed for synchronous learning [19-21], asynchronous learning [22, 23] or asynchronous learning with an on-line facilitator (Wang and Karmouch) [24,25]. Various models for collaborative Systems that come to cover the communicational needs of collaborative work in a learning System either synchronous [26, 27], or asynchronous [28] have also been proposed.

Numerous algorithms were found for achieving intra and inter stream synchronization in different scenario [29-30]. There exist few papers for achieving group synchronization [1, 35]. The algorithm proposed in [35], synchronizes the initial play out time for all receiver, subsequently a solution for multipoint synchronization control with continuous media has been proposed in [34,36]. The algorithm proposed in [1] describe how audio signal mixing between the speakers is minimized but there is a need of formalization and verification of this algorithm.

There are many papers which model and verify the Distributed Algorithm using Petri Nets [37,38] but few papers exist, which was designed and modelled using Coloured Petri Nets[36,39,40]. In [36], the Dining Philosophers problem has been addressed with the CPN Model which describes how a number of processes (philosophers) share a common resource (chopsticks), this problem is one of the traditional examples used by computer scientist to illustrate new concepts in the area of synchronization and concurrency. Telephone System [39], paper describes how public telephone system as it is conceived by user and Hierarchical Protocol [40], describe how simple protocol turned into a Hierarchical protocol and describe CPN Model of sender and receiver behaviour.

3. SCOPE OF WORK

The market of video conferencing based tools is increasing rapidly. But practical experiments show that no collaborative working environment and in particular, no videoconferencing application fulfill the collaborative working environments like tele teaching.

In this paper we offer a system in a distributed environment, so that the chance of audio signal mixing between the speakers is minimized during a tele lecture session. Also synchronization among the participants is not the only requirement for conferencing in teacher-student environment, all the participants need to speak in the context of the subject under discussion, we propose a new algorithm, where each of the participants (students) can speak out in an appropriate sequence (as decided by the individual participant) and as controlled by the group server.

In our earlier work [1], we have proposed an approach, which covers the Interactive group synchronization algorithm of a Teleteaching System to tackle the problem of audio speech overlapping. Instead of allowing multiple participants (teacher/ student) to speak simultaneously, the algorithm chooses one participant at a time depending of the context of current discussion.

In this work first we try to model this synchronization problem with the existing formal tools of distributed problem modelling like general Petri Nets [38,41], Time Petri Nets[42], Coloured Petri Nets[8]. But we find the inability of these tools for expressing our problem. So, we extend the existing Coloured Petri Nets by introducing a time parameter in it and name it as Time Coloured Petri Nets (TCPN), which we find more expressive in modelling time dependent distributed control algorithm.

4. MODELLING THE PROBLEM

4.1 Time coloured petri nets

Coloured Petri Nets is a graph oriented language for design, specification, simulation and verification of systems. It is in particular well-suited for systems that consist of a number of processes which communicate and synchronize; it's a combination of Petri Nets and programming language. Where Petri Nets control the structures, synchronization, communication and resource sharing, where as data manipulation are described by functional programming language.

It constructs of:

Ellipses and circles: are called places, they describe the states of the system.

Rectangles: are called transitions, describe the action.

Arrows: are called arcs, arc expression described how the state of the CP-net changes when transitions occur.

Tokens: Each place contains a set of tokens, each of token carries a data value with a given type called colour.

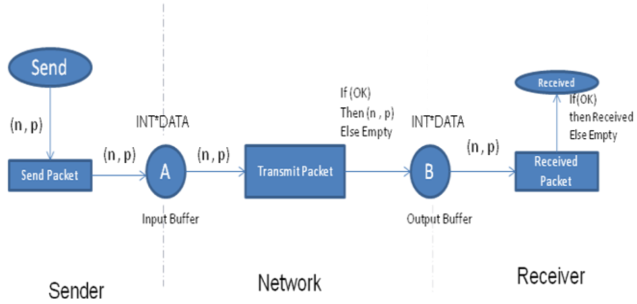

Fig-1: Example of colour Petri Nets

In the above Fig-1 transition Send Packet is enabled when there is a token (n, p) on place Send, where n=packet no. and p=packet content. When transition occurs token is removed from input places to place A(input buffer).Transition Transmit Packet is enabled when <n,p, OK=true>, Then packet is transferred from place A to B. Transition Received Packet is enabled when token is on place B, packet Received at Place Received and packet is OK.

A CPN associated with a time interval called Time Colour Petri Nets (TCPN).It consists of two types of places one Time Place, another Non-time Place and a Transition.

Time Place: Associated with time interval. When a token is created in a time place the token is locked for a duration of fixed time interval. Token becomes unlocked at the output arc when the duration of time interval over.

Non-time Place: A token created in non time place is to be unlocked at all times.

Transition: Transition is enabled if all of its input places contain at least one unlocked token.

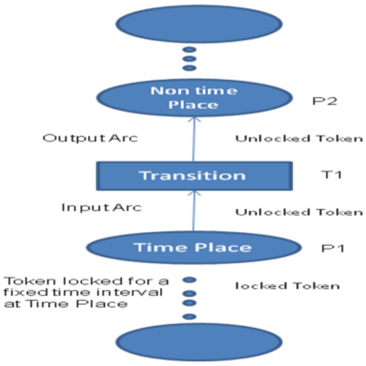

Fig-2: TCPN

4.1.1 *Formal definition of TCPN:*

A TCPN is a tuple TCPN= ($\sum$, P, T, A, N, C, I, O, t)

Where:
(i) $\sum$ is a finite set of non-empty types, called colour sets.
(ii)) P=(p1,p2,p3….pm) ,where m>=0 is a finite set of places.
(iii) T=(T1,T2,T3……Tn) ,where n >= 0 is a finite set of transition, where P∩T=Ø , i.e.: set of places and transition are disjoint.
(iv) A is finite set of arcs such that: $P \cap T = P \cap A = T \cap A = \phi$.
(v) N is a node function .It is defined from A into $P\times T \cup T\times P$.
(vi) C is a colour function .It is defined from P into $\sum$.
(vii) I:T→P, is input arc mapping from transition to place.
(viii) O:T→P, is output arc mapping from transition to place.
(ix) t=(t1,t2,…..tq) ,where q>=0 is a finite set of time interval.

4.1.2 Properties of TCPN Model

4.1.2.1 Safety property:

This property states that the teacher terminate the lecture session when all the listeners have accepted. We represent this property by
$\forall x, x \in N$, Terminated (t) $\longrightarrow$ Accepted(x)

4.1.2.2 Liveness property

This property states that every transition in the TCPN model is firable from every states of the reachablity graph in a finite time.

4.1.2.3 Reachablity property

The speech must be reachable to the every participant of the lecture session. After a firing sequence token must be reached from initial marking to final marking means all the transition is successful.

4.2 TCPN Model of the problem

In a tele lecture session, there is a coordinator, who controls the entire session. At first Coordinator behaves as a sender, starts the conference with his speech and the others behave as receiver, and they receive the speech and make request or question against the particular speech.

Now in this modeling we describe two behaviours *Sender behaviour:* who gives the speech and *Receiver Behaviour:* who listens to the speech and make the request or question against the sender speech

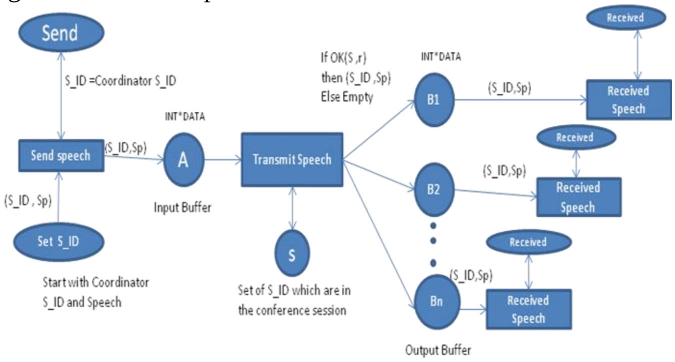

Fig-3: Sender Behaviour

In the Fig-3 initially the Send place has only one token with coordinator S_ID and Set S_ID place binding the Coordinator S_ID and Speech (Sp).Transition send speech is enabled, because there is a S_ID token on place Send and (S_ID,Sp) token on place Set S_ID .When transition occurs it removes the two specified token from the input places ,but the token S_ID immediately put back on place Send ,due to double arc. So coordinator speech sent by adding it to the input buffer(A) of the network. When the (S_ID, Sp) token is put on place A, transition Transmit speech become enabled with two different condition if OK(S,r)=True and OK(S,r)=False where S=Set of Speech-ID of a particular conference session and the function OK(S,r) checks whether r $\in$S for r $\in$ (S_ID-1, S_ID-2 …S_ID-n). If OK(S,r)=True then speech transferred from place A to B otherwise discard. When (S_ID, Sp) token arrives at place B then transition Receive speech becomes enabled.

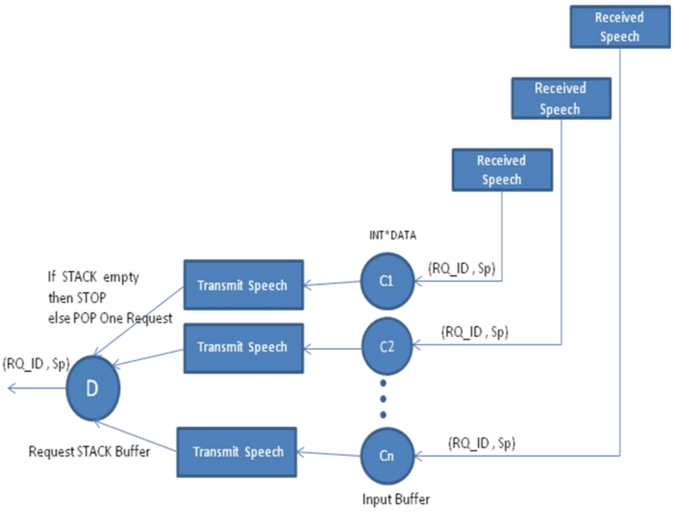

Fig-4 Receiver Behaviour

In the receiver side, only those receivers, who have a request or question against the speech are only considered as a receiver. In the above Fig-4, after transition *Received Speech* is enabled every receiver binding with their RQ_ID and Request Speech and copy to the place C1,C2……Cn respectively where n is the number of receiver who have a request or question .When (RQ_ID,Sp) token is at place C then transition *Transmit speech* woks as a similar way to *Transmit speech* of Sender behavior and request moves from place C to D. Place D is Request STACK buffer where all the requests are stored at this place with their speech. Request STACK buffer operates with condition that if STACK is Empty then STOP otherwise POP one element from STACK and place on the output arc of place D.

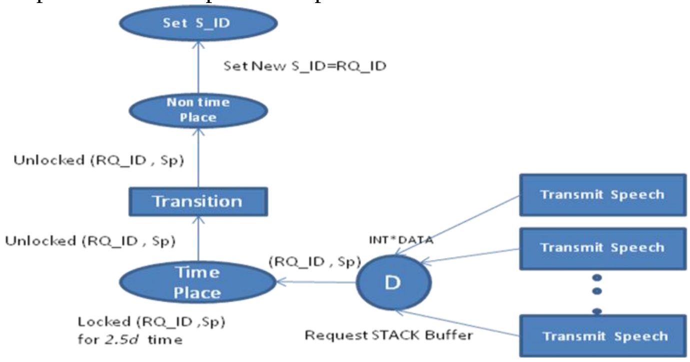

Fig-5 Time Behaviour

In the above Fig-5 when token request-ID and speech(RQ_ID, Sp) reach at *time place* then it locked for 2.5d (here 2d is considered as round trip delay between sender and receiver in the network and 0.5d is considered as request processing time) time duration and then the token is unlocked at input arc of *transition*, now transition is enabled with unlocked token (RQ_ID ,Sp) and it is transfered to the input arc of *Non-time place*.

At *Set ID* place set the new S_ID (S_ID=RQ_ID) for next Speaker .These the entire step continues until the Request STACK buffer is empty.

Now we combined these three behaviors which make a complete TCPN Model for the above problem shown below in Fig-6

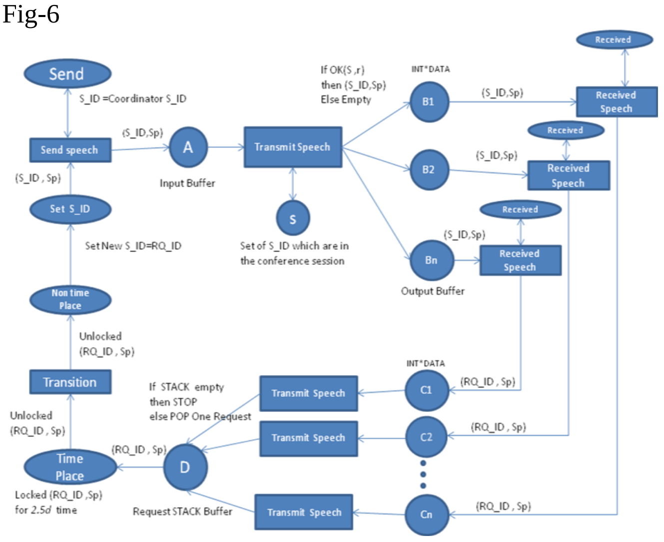

Fig-6 TCPN Model

Declaration:

type INT=integer;
type DATA=Audio packet;
type BOOL=Boolean;
type INT×DATA=product INT*DATA;
var S_ID : INT;
var Sp : DATA;
var ok : BOOL;

5. FORMAL VERIFICATION OF THE TCPN MODEL

We present a state diagram of the TCPN model for its verification. In state diagram at first teacher behave as a speaker to start the lecture session with his speech and others behave as listener who receive the speech and make request against the speaker's speech. Now we take x and y as any participant of lecture session where x,y $\in$N, where N is a set of participant in the Lecture session. In a particular time instance x behave as a speaker and y as a listener or vice versa. In the state diagram when x is the speaker then x delivers the speech to the y, where y is a set of listener and xUy = Participants.

It can be described by

$$\forall_y \exists_x Sp(y,x) \rightarrow TRUE$$

Where |x| = 1 and x,y $\in$ Participants
Sp(y,x) : x delivering speech to y

After delivering the speech, x(speaker) waits for request from y(listener)

After listening the speech when listener makes request to the speaker that can be described by

$$\exists x, \exists y RSp(x, y) \rightarrow TRUE$$

$$\text{where } |x| = |y| = 1$$

RSp(x,y): y makes request to x.

All request against the particular speech that can be described by

$$\forall y, \exists x \overline{Sp}(x, y) \rightarrow TRUE$$

$$where\ |y| \geq 0, |x| = 1$$

$\overline{Sp}(x, y)$: where y is the listener who sends the request to the speaker x.

After receiving all the request, a participant corresponding to the request ID on the top of the RSB is selected as speaker, that can be described by $\exists x_1 Sp_s(x_1) \rightarrow TRUE$,where x_1 is Listener and Rqid(x_1)=RSB[TOP], Rqid(x) $\rightarrow$Request ID of x, is a new speaker.

When all the listeners get chance to give the speech then Request STACK buffer (RSB) generate the final speech sequence at input arc of *Accepted* place that can be describe by $\exists x Sp_s(x) \rightarrow TRUE$,where 1≤ x≤ n ,n is number of participants in the lecture session.

When $\overline{Sp_s}(x)$ in the input arc of *Accepted* place then pending of listener request is end and $\overline{Sp_s}(x)$ forwarded to the output arc of *Accepted* place then the lecture session terminated.

5.1 Speaker state diagram of TCPN model

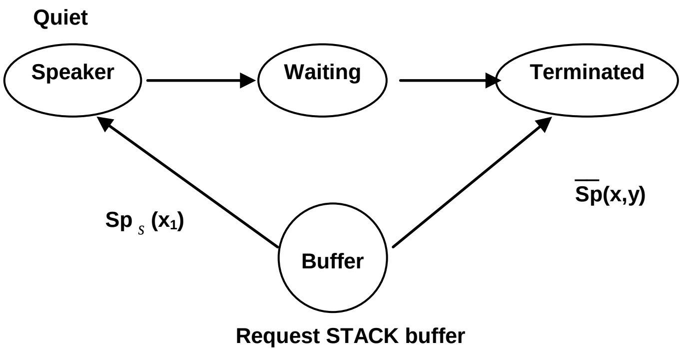

Fig-7 Speaker State diagram of TCPN Model

A teacher (t) is a initiator of a lecture session so at first t is the speaker in the speaker state diagram in fig-7.
(i)When speech is not yet delivered, the state **Quiet** holds.
(ii)When speaker is waiting for listener's request the state **waiting** holds.
(iii)When speaker receive the entire request against his speech then state **terminated** holds.
Speaker t always remain one of the three states, denoted by $(\forall t, t \in N)$, Quiet (t) + waiting (t) + terminated (t) =1 (1)
Where N is set of participants and the above expression is a Boolean expression.

5.2 Listener state diagram of TCPN model

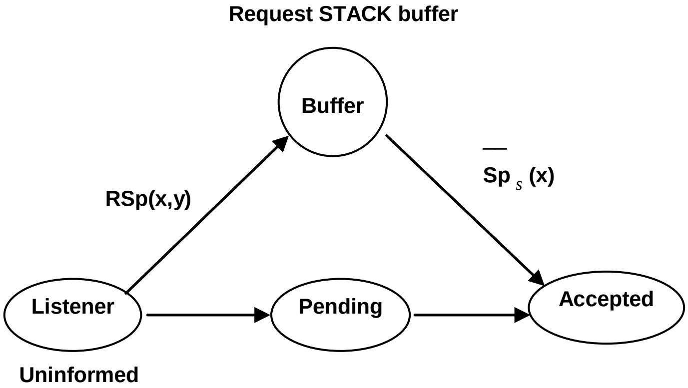

Fig-8 Listener State diagram of TCPN Model

In the listener state diagram in fig-8,
(i)When listener is not getting any speech from the speaker the state **uninformed** holds.
(ii)When the listener is waiting to give the speech, the state **pending** holds.
(iii)When all the listener gets chance to deliver speech then the state **Accepted** holds.
Listener x remains in one of the three state denoted by $(\forall x, x \in N)$
Uninformed(x) +Pending(x)+Accepted(x)=1 (2)
The above expression is a Boolean expression.
Now we combine the speaker and listener state diagram of TCPN model.

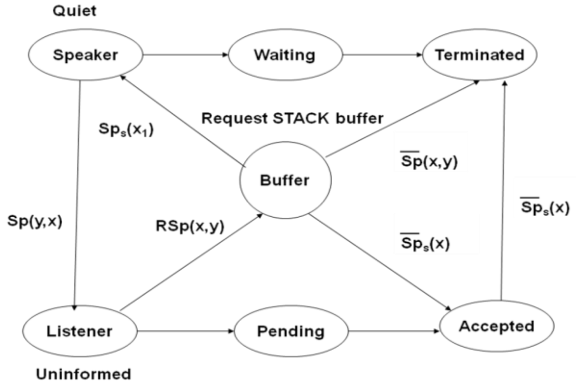

Fig-9 State Diagram of TCPN Model

Now considering speaker and listener behaviour we get in fig-9.
$\forall x, \forall y; x, y \in N$ Quiet(y) + Uninformed(y) + Pending(x, y)+ Pending(y, x)+terminated(x)+ Accepted(x)= 1 (3)

Where
(i)Final speech sequence is in Request STACK buffer then **RSB** holds.
(ii)When y make a request to x but not getting chance to speak then **Pending(x,y)** holds. Similarly for a request from x to y then **Pending(y,x)** holds.

To show that the teacher terminates the session when all other participant's request are accepted, it sufficient to show that a teacher only terminates when no participant's request is **uninformed** or **pending**.
So we have to prove that
Terminated(t)→ Accpeted(x).

5.3 Lemma: TCPN model of the problem satisfies the Safety property

Proof: From the equation (3)
$\forall x, \forall y : x, y \in N :$
Pending (x,y) $\rightarrow$($\neg$ Uninformed (y)) $\wedge$ $\neg$ accepted(y)) (4)

Equation (2) implies

$\forall$ y, y$\in$N($\neg$ Uninformed(y) $\wedge$ $\neg$ accepted(y)) $\rightarrow$pending(y) (5)

Now we combining (4) and (5) together, we get

$\forall x, \forall y : (x, y) \in N :$ Pending(x,y) $\rightarrow$ (pending(y) $\vee$ y=t) (6)

Where y= t means teacher is the initiator of lecture session and pending(y) $\rightarrow$ Listener y is in pending state.

For each participant x there is another participant y, who is the parent of x. If x is pending then we obtain a sequence of pending state.

Therefore we write

$\forall x, x \in N :$ Pending(x) $\rightarrow \exists z :$ Pending (z, t) (7)

Where z is all pending student and x $\subset$ z

Equation (3) implies

Pending (z, t) $\rightarrow \neg$ terminated (t) (8)

Combining equation (7) and (8) together, we get

(i)Pending(x) $\rightarrow \neg$ Terminated (t) (9)

From equation (3) we get

Uninformed(x) $\rightarrow \neg$ Accepted(x) (10)

When listener send request to the teacher and they are in pending state then Pending(t, x) holds where x is the listener in pending state and in the end the teacher must terminate the session.

So we get

$\forall x, x \in N :$ Pending(x) $\rightarrow \exists t :$ Pending (t,x) (11)

Which means x is waiting for t to give him chance to speak.

From equation (3)

Pending(x)$\rightarrow \neg$ Uninformed(x) $\wedge \neg$ Accopeted(x) (12)

$\neg$ Uninformed(x) $\wedge \neg$ Accepted(x)$\rightarrow$pending(x) (13)

So, from equation (9) and (13) we can write

$\neg$ Uninformed(x) $\wedge \neg$ Accepted(x)$\rightarrow$terminated(t).

So, Terminated (t) $\rightarrow$ Accepted(x).

5.4 Lemma: TCPN model of the problem satisfies the Reachablity and liveness property.

The components of TCPN Model shown in fig-5 are designated as follows for proving the reach ability of the model.

In transition state diagram of fig-10, we describe Place **Set S_ID** as p1, **send** place as p2, **S** place as p3,**A** place as p4,**B** place (B1,B2…Bn) as p5,**C** place (C1,C2…..Cn) as p6, **D** place as p7 and **received** place as p8 and transition from send place to A place as t1, set s_id place to A as t1, S place to B place as t2, A place to B place as t2, B place to C place as t3, C place to D place as t4. D place to set s_id place as t5, D place to receive placed as t5 and a transition t6 from received place that consumes the received token.

i) Transition state diagram of TCPN Model:

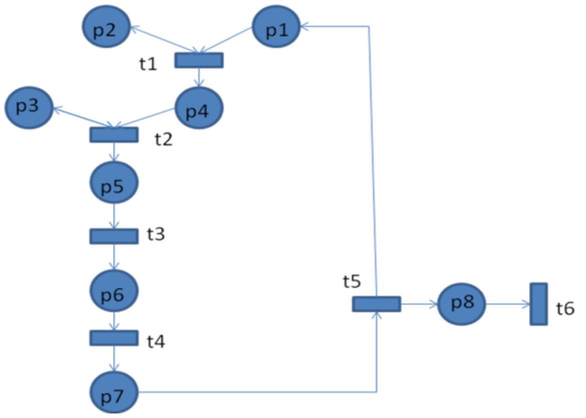

Fig-10 Transition State Diagram of TCPN Model

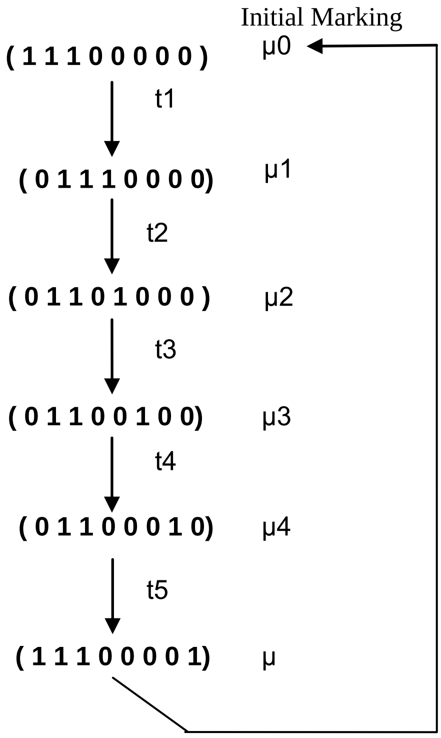

Fig- 11 Reachablity Graph

In the reachability graph shown in fig-11, a sequence of transition σ =t1,t2,t3,t4,t5 is a firing sequence from µ0

Iff $\mu 0 \xrightarrow{t1} \mu 1 \ldots\ldots\ldots\ldots \xrightarrow{tn} \mu n$

we also write $\mu 0 \xrightarrow{\sigma} \mu n$.

Now P and T represente place and transition respectively where P=(p1,p2…pk) and T=(t1,t2……tm).

We define k×m incidence matrix [T]

where [T](i.j)= ϕ (tj,pi) - ϕ (pi,tj)

ϕ (tj,pi) =no. of token added.

ϕ (pi,tj)=no. of token remove.

[T](i.j)=Changed in place I when transition j fires once.

Now if marking is reachable then equation

$\mu0 + [T].\ \#\sigma = \mu$ holds. Where μ0=Initial marking,μ=Final marking, $\#\sigma$=m dimensional vector with its j^{th} entry denoting the no. of time transition t_j occurs in σ and [T]=Incidence matrix.

$$\mu 0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},\quad \mu = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\#_\sigma = \begin{bmatrix} t1 \\ t2 \\ t3 \\ t4 \\ t5 \\ t6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \text{ and}$$

$$[T] = \begin{vmatrix} -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{vmatrix}$$

L.H.S= $\mu0 + [T].\ \#\sigma$ =

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + [T]. \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \mu = \text{R.H.S}$$

So, $\mu0 + [T].\ \#\sigma = \mu$ holds for firing sequence σ =t1,t2,t3,t4,t5, So the tokens are reached to the final marking by sequence of successful transition.

In the reachablity graph, we see that from every state, all the transitions (t1, t2, t3, t4, t5, t6) are fireable in finite time. So, we can say that the petri net model satisfies liveness property.

6. PROPOSED IGS ALGORITHM

6.1. Terminology Used

6.1.1 Conference Session

A conference session is defined as the duration for which a conference proceeds. The coordinator (defined later) starts and ends the session. A session consists of a coordinator and a number of participants (defined later) who exchange real time video communication and are connected through network in a distributed environment.

6.1.2 Participant

A participant is defined as a node which is connected in the conference session. A participant normally watches the conference (both audio & video) and also may participate through a speech or question. For this purpose, it has to inform the coordinator through a request. A participant can deliver his speech only on getting the consent from the coordinator.

6.1.3 Coordinator

Coordinator is one of the participants who controls the entire session. In a teacher-student environment, the teacher normally plays the role of the coordinator. All the requests generated by the participants in time scale are stored in first come first serve basis but processed by the coordinator (also called Group Server) so that all the participants get the opportunity to deliver the speech or ask the question in an appropriate sequence. The term appropriate sequence means that all the speech related to a specific context should be discussed in sequence (see algorithm). The coordinator can stop any participant at any point of time and may choose any other instead.

6.1.4 Request Tree

It is a tree of participant, whose root node is T, which symbolizes, that the coordinator has initiated the session. In any level of the tree the children represent that they have something to speak with the context of the speech of their parent node.

6.1.5 Request-Id

Whenever a participant sends a request to speak out with the context of someone's speech, a unique request-id will be generated for that particular request. From the format of the request Id, it can be identified, the position of the request, i.e. on whose context the participant would like to speak.

6.1.6 Speech-Id

When a participant is allowed to speak depending on his request, a unique speech-id will be generated for that participant's speech.

6.2 Basic idea of Algorithm

This algorithm works in the group server. It chooses one participant at a time in a particular pattern of session topic to act as a speaker and other as listener to avoid the semantics incoherence in the dialogue between more than one participant. With this idea every participant in the group gets the turn to act as speaker. This continues for the entire life time of the conference session. For a participant when it gets its turn for speaking, either he can speak out or he can discard the chance and pass it to the next participant for acting as speaker.
At the beginning of the session, each node sends a dummy packet with timestamp to all other nodes for finding the round trip delay (Drt). With this it creates a matrix Drt[n][n] where Drt[i][j], denotes the round trip delay between the i^{th} node and j^{th} node and the maximum network delay d is defined as:
d = (max (Drt[i][j]))/2 , for i=1 to n and j=1 to n. It may also be said that after sending packet by any participant, it will take maximum d time to reach all other participants.

6.3 Algorithmic steps

We will discuss the distributed algorithm for serving the request of the participants. The objective of the algorithm is to organize the conference discussion, by ordering all the participants from the request tree in the appropriate sequence.

Algorithm: IGS Algoritm.
Input:
   i. Initiation of the conference session.
   ii. Request by the participant to speak out.
Output:
   i. Set of speech id.
   ii. Set of request id.
   iii. Request tree.

C1. Coordinator:
1. Initiates the conference session.
2. Generate the speech-id (1 in this case).
3. Transmit the speech-id to all connected in the session.
4. Deliver the speech (assume coordinator is the first speaker).
5. On completion of the speech, declare speech end.

P1. Participant:
1. Listen for the speech from the coordinator.
2. If interested, note the speech-id and accordingly raise request, wait for instruction from coordinator.
3. Else do nothing.

C2. Coordinator:
1. Wait for 2.5 d for receiving requests from participant(s).
2. If no request received, declare end of session, stop.
3. Else generate request-id and make request tree by inserting request-id as the child node of the previous speaker.
4. Explore the request tree in DFS order and transmit the request-id of first un-explored node to its corresponding participant (P_i).
5. Generate the speech-id for P_i and transmit it to all.
6. Wait for more requests

P2. Participant:
1. P_i: Receive the request–id and speak out and declare end of speech.
2. Other: Listen to the speech from P_i, if interested, note the speech-id and accordingly raise request, wait for instruction from coordinator.

C3. Coordinator:
1. If more requests reach, extend the tree and explore the tree else explore the tree for unexplored node.
2. If unexplored node found transmits the request-id of it to the corresponding participant else declare end of session and stop.
3. Go to step C2.5.

Request STACK Buffer Operation:
Illustration with Example:

In TCPN Model request STACK buffer (RSB) works as follow:

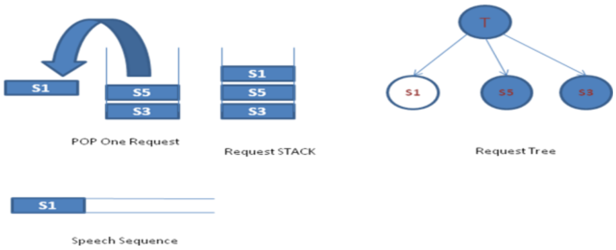

Fig-12: RSB & Request tree operation

In fig-12, first the co-ordimator (T) initiate the conference session by giving his first speech. On his speech, there are some queries form the participants s3,s5 and s1. So, the request from s3, s5, and s1 has arrived and are pushed in the stack of RSB. To form the request tree, s3, s5, and s1 are put as the child node of T. With the pop operation on RSB, a request (s1) from the stack is popped out and s1 is allowed to give speech.

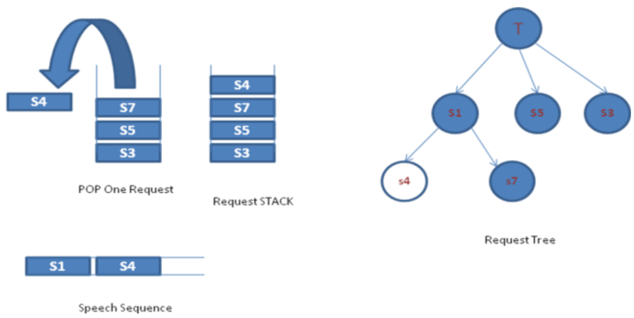

Fig-13: RSB & Request tree operation

After the speech of s1, there are queries from s7 and s4 in fig-13 on the context of s1. So, request from s7 and s4 are put as the child node of s1 in the request tree. Also, s7 and s4 are pushed in the stack of RSB. With the pop operation on RSB, a

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012
ISSN (Online): 1694-0814
www.IJCSI.org

request (s4) from stack is popped out and s4 is allowed to give his speech.

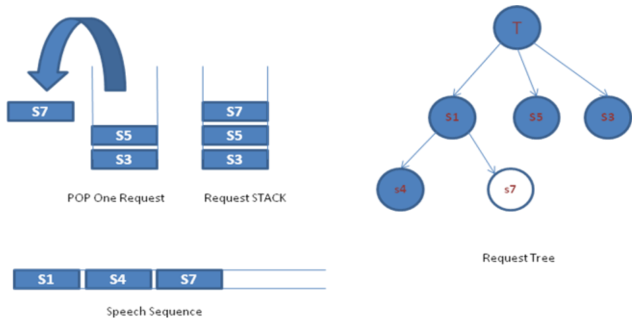

Fig-14: RSB & Request tree operation

After the speech of s4, in fig-14, there are no queries against the speech of s4. So, the request (s7) is popped out with the pop operation on RSB and s7 is allowed to give his speech.

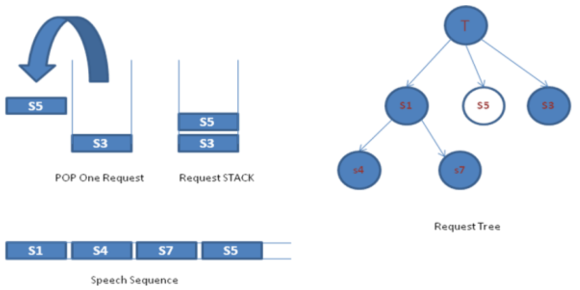

Fig-15: RSB & Request tree operation

After the speech of s7, in fig-15, there are no queries against the speech of s7. So, the request (s5) is popped out with the pop operation on RSB and s5 is allowed to give his speech.

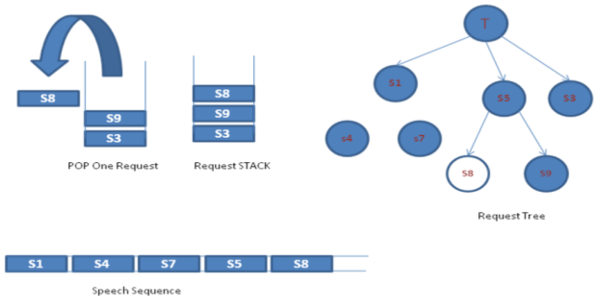

Fig-16: RSB & Request tree operation

After the speech of s5, there are queries from s9 and s8 in fig-16 on the context of s5. So, request from s9 and s8 are put as the child node of s5 in the request tree. Also, s9 and s8 are pushed in the stack of RSB. With the pop operation on RSB, a request (s8) from stack is popped out and s8 is allowed to give his speech.

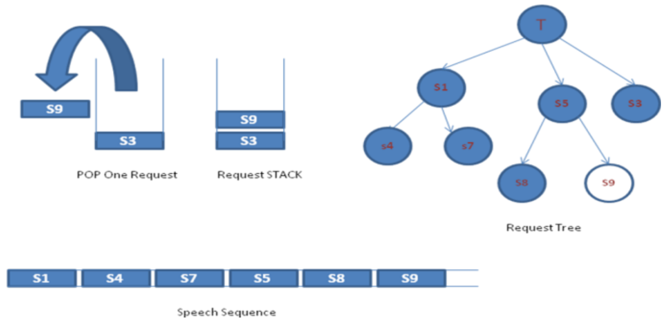

Fig-17: RSB & Request tree operation

After the speech of s8, in fig-17, there is no query against the speech of s8. So, the request (s9) is popped out with the pop operation on RSB and s9 is allowed to give his speech.

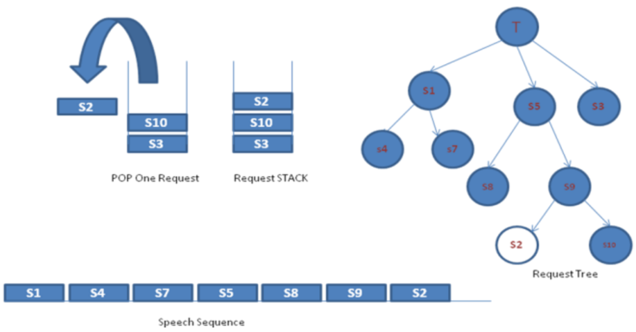

Fig-18: RSB & Request tree operation

After the speech of s9, there are queries from s10 and s2 in fig-18 on the context of s9. So, request from s10 and s2 are put as the child node of s9 in the request tree. Also, s10 and s2 are pushed in the stack of RSB. With the pop operation on RSB, a request (s2) from stack is popped out and s2 is allowed to give his speech.

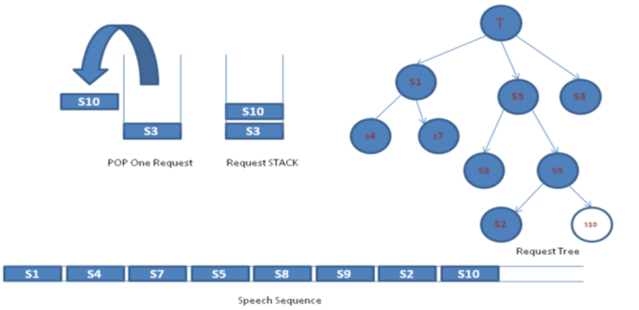

Fig-19: RSB & Request tree operation

After the speech of s2, in fig-19, there is no query against the speech of s2. So, the request (s10) is popped out with the pop operation on RSB and s10 is allowed to give his speech.

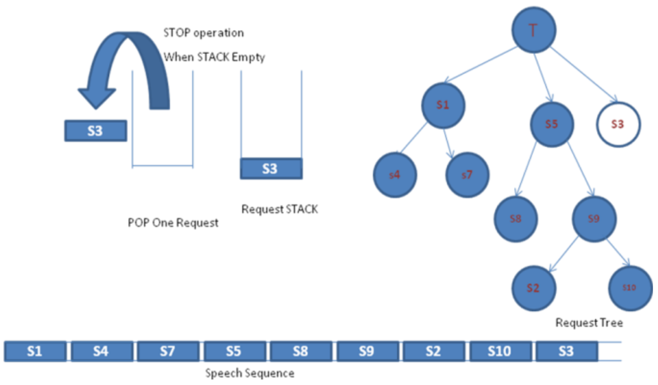

Fig-20: RSB & Request tree operation

After the speech of s10, in fig-20, there is no query against the speech of s10. So, the request (s3) is popped out with the pop operation on RSB and s3 is allowed to give his speech.

After the speech of s3, there is no query against speech of s3. So, the stack is empty, which denotes the end of session.

Final Speech Sequence:

Final Sequence with Coordinator Speech is as follows in fig-21, which proves that our proposed synchronization algorithm can reduce the audio speech overlapping by avoiding the multiple speakers to speak at the same time. The algorithm chooses one participant at a time as a speaker and the other as listener so as to complete the session in a finite time.

Fig-21: RSB & Request tree operation

7. CONCLUSION

A new type of synchronization called Interactive Group Synchronization has been addressed and a solution to achieve appropriate ordering has been presented. Such synchronization is very important in distributed video conferencing based teleteaching system in order to guarantee that the all sender sends and receive their audio stream synchronously. Here we defined an extension of petri nets called TCPN to model the problem and formally verified the correctness, reach ability and liveness property of the model. Future scopes of this work can be to analyse the fault tolerance and scalability issues of the proposed algorithm.

References

[1] D.Animesh, D.Ranjan, B.Swapan, “A Group Synchronization Algorithm for VoIP Conferencing”, In Proceedings 8th WSEAS International Conference on Software Engineering, Parallel and Distributed Sytems (SEPADS '09) , at University of Cambridge, Cambridge, UK, 21-23 February 2009.

[2] Bur Goode, “Voice Over Internet Protocol (VoIP).” Proceedings of the IEEE, VOL. 90, NO. 9, pp:1495--1517 September 2002.

[3] B.Pantelis, S.Ioannis, M.Lazaros, “Potential and limitations of a teleteaching environment based on H.323 audio-visual communication systems”, Elsevier, Computer Networks and ISDN Systems, 6(34, (2000), pp: 945-958.

[4] Ch. Bouras, A. Gkamas, V. Kapoulas, Th. Tsiatsos, “Evaluation of teleteaching services over ATM and IP networks”, Elsevier, Telematics and Informatics 20 (2003) ,pp: 1–18.

[5] B.Fernando, L.Jaime, G.Miguel, “Multimedia group and inter-stream synchronization techniques: A comparative study”, Journal Information Systems Volume 34 Issue 1, March, 2009.

[6] O.Philippe, “Enforcing Multipoint Multimedia Synchronisation in Videoconferencing Applications”, IDMS 2000, LNCS 1905, pp: 65-76.

[7] B.Fernando, J.Segui , L.M.Jaime , “Multimedia Group Synchronization Algorithm Based on RTP/RTCP”, In proc. Eighth IEEE International Symposium on Multimedia, 2006. ISM'06, San Diego, CA Dec. 2006 pp: 754 - 757 .

[8] K.Jensen, “Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use”, Springer. 2nd eds. 1996.

[9] http://www.foruminc.com

[10] http://www.vitual-u.cs.sfu.ca

[11] http://www.centra.com

[12] http://homebrew1.cs.ubc.ca/webCT

[13] http://www.softrc.com

[14] http://www.oovoo.com/

[15] http://www.acrobat.com/

[16] http://vsee.com/

[17] http://www.sightspeed.com/

[18] http://www.palbee.com/

[19] G.Eschelbeck, “An architecture for multimedia communication in a distributed education environment”, In: Proceedings of ED-MEDIA 95, Graz, Austria, June 17-21 June 1995, pp: 217-222.

[20] G.Paquette , C.Ricciardi-Rigualt, J.Bourdeau, C..Paquin, S.Liegeois, “Modeling a Virtual campus environment for interactive distance learning”, In: Proceedings of ED-MEDIA 95, Graz, Austria, 17-21 June 1995, pp: 523-528.

[21] J.Boegh, A.Krebs, L.Petersen , M.Wagner, “A Web based Virtual College”, WebNet 98 ± World Conference of the WWW, Internet & Intranet Orlando, Florida, 7-12 November 1998.

[22] A.F.Smeaton, F.Crimmins, “Virtual lectures for undergraduate teaching: delivery using Real audio and the WWW”, In: Proceedings of ED-MEDIA/ED-TELECOM 97, Calgary, Canada,14-19 June 1997, pp: 990-995.

[23] A.D.Wolf, “The implementation of an open learning environment under World Wide Web”, In: Proceedings of ED-MEDIA 95, Graz, Austria, 17-21 June 1995, pp: 689-

694.
[24] A.Eckert, W.Geyer, W.E€elsberg, "A distance learning system for higher education based on telecommunications and multimedia a compound organisational, pedagogical, and technical approach", In: ED-MEDIA 97 and ED-TELECOM 970 World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications, Calgary, Alberta, Canada, 14-19 June 1997.
[25] E. Mc Daid, W.O'Brien, G.Row , A.Hashim, "The Talent network tutoring system. Telecommunication for Education and Training", Gjovik, Norway 8-11 June 1999.
[26] A.Faro, D.Giordano, G.Gurrieri, "An internet based collaborative environment to learn information systems design", In: Proceedings of ED-MEDIA/ED-TELECOM97, Calgary, Canada, 14-19 June 1997, pp: 346-351.
[27] D.Ponta, A.M. Scapolla , M.Taini, "Telematics for education: the design of a distributed computer-based collaborative learning system", In: Proceedings of ED-TELECOM 96, Boston, MA, USA, 17-22 June 1996, pp: 252-257.
[28] K.N.Tsoi, S.M. Rahman, "Media-on-demand multimedia electronic mail: a tool for collaboration on the web", In: Proceedings of the Fifth IEEE International Symposium on High Performance Distributed Computing, On Center, Syracuse, New York, 6-9 August 1996, pp: 121-126.
[29] A.Boukerche, H.Owens, "Media synchronization and qos packet scheduling algorithms for Wireless systems", Mobile Networks and Applications 10, 2005, pp: 233–249.
[30] JC.Guerri (1997) "Specification and performance evaluation of an adaptive multimedia stream Synchronization protocol based on global time and feedback techniques", with stream control Thesis, U.P.V. June.
[31] JC.Guerri, M.Esteve, DE.Palau, V.Casares (2001) "Feedback flow control with hysteresial techniques for multimedia retrievals. Multimedia Tools and Applications" 13(3), pp:307–332.
[32] K.Isidor, H.Vicky, and W.Anna, "Lip synchronisation for use over the internet: Analysis and implementation", In IEEE Globecom '96, London, UK, November 1996.
[33] N.Laoutaris, I.Stavrakakis. "Intra stream synchronization for continuous media streams: a survey of playout schedulers".IEEE Netw Mag 16(3), pp: 30–40 .
[34] PV.Rangan, S.Ramanathan (1995) "Performance of inter-media synchronization in distributed and heterogeneous multimedia systems", 1993.Comput Netw ISDN Syst 27(4), pp: 549–565.
[35] Akyildiz IF, Yen W (1996) "Multimedia group synchronization protocols for integrated services networks". IEEE J Select Areas Commun 14, pp: 162–173.
[36] J.Kurt,"Dining Philosophers problem",Aarhus University,Denmark(kjensen@daimi.au.dk).
[37] B.Manfred, "On the design and verification of a simple distributed spanning tree algorithm", SFB-Bericht 342/24/90 A, Technische Universtit~it Miinchen, December,1990.
[38] Kindler, E., Reisig, W., V¨olzer, H., Walter, R., "Petri Netbased verification of distributed algorithms", an example. Formal Aspects Comput 9: 409–424, 1997
[39] J.Kurt ,"Telephones System", Aarhus University, Denmark (kjensen@daimi.au.dk).
[40] J.Kurt ,"Hierarchical protocol", Aarhus University, Denmark (kjensen@daimi.au.dk).
[41] M. Tadao, "Petri nets: properties, analysis and applications", *Proc. of the IEEE*, 77(4) 1989, pp:541–580.
[42] P.Louchka, "On Time Petri Nets", Journal of Information Process Cybernetics, EIK 27 (1991) 4, pp: 227-244.

Animesh Dutta received the B.E. in Computer Science and Engineering from Regional Engineering College, Durgapur,India and M.Tech from National Institute of Technology, Durgapur,India in the year 2002 and 2006 respectively. He is currently with the department of Information Technology of National Institute of Technology, Durgapur, India. He is pursuing his Phd work in the area of Real Time Communication in IP network. His other research interests are Distributed Computing, Software Engineering.

Ranjan Dasgupta received the B.Tech in Radio Physics & Electronics, M.Tech and Phd in Computer Science and Engineering from University of Calcutta. He is currently head of the department of Computer Science and Engineering, National Institute of Technical Teacher's Training & Research, Kolkata. His other research interests are Distributed Computing, Software Engineering, and Database etc. He has published numerous research articles in various international journal and conferences. He is a member of IEEE.

Swapan Bhattachraya received the B.Tech in Radio Physics & Electronics, M.Tech and Phd in Computer Science and Engineering from University of Calcutta in the year 1979, 1981 and 1991 respectively. He is awarded Sr. Research Associateship (1999-2001) from National Research Council, USA and Young Scientist Award from UNESCO in 1989. He is currently with the department of Computer Science and Engineering, National Institute of Technology, Suratkal,India and Professor in lien of the department of Computer Science and Engineering, Jadavpur Univeristy, Kolkata,India. He has worked in different prestigious academic and R&D organization in national as well as international level like IIM Kolkata, IIT Mumbai, NITTTR Kolkata, Asian Institute of Technology, Bangkok , Michigan State University, USA , Atilim Univ at Ankara, Turkey , Naval Postgraduate School, Monterey, CA , Univ of New Castle, UK. Some of his research interests are Distributed Computing, Software Engineering, and Interconnection Networks etc. He is a member of IEEE, Operation Research Society of India etc.