Multipoint Multimedia Synchronization: A Petri Net Based Approach

ANIMESH DUTTA[1]
KONINIKA PAL[2]
RANJAN DASGUPTA[3]
SWAPAN BHATTACHARYA[4]

[12]Department of Information Technology
National Institute of Technology
Durgapur- 713209, India
[3]Department of Computer Science & Engineering
National Institute of Technical Teacher's Training and Research
Kolkata-700106, India
[4]Department of Computer Science & Engineering
Jadavpur University
Kolkata-700032, India
[12](animeshrec,koninikapal)@gmail.com
[3]ranjandasgupta@hotmail.com, [4]bswapan2000@yahoo.co.in

Abstract. Maintaining the satisfactory QoS, synchronization is a big challenge to the researcher in the field of Information and Communication Technology (ICT). In this paper we address a multimedia synchronization issue called multipoint synchronization which is necessary in some distributed collaborative application like teleteaching, teleconferencing etc., involving the play out process of same media stream at different receivers at the same time to achieve fairness among the receivers. We found the incapability of existing petri net to model the above synchronization issue. Some new features are added in existing petri nets to increase its modeling and analyzing power, called self modifying stochastic color petri net (SMSCPN).Then the synchronization issue is modeled with the help of SMSCPN and an generalized algorithm is proposed to achieving multipoint synchronization in IP network. Also some metrics are defined that can measure the performance of the proposed synchronization algorithm.

Keywords: multimedia, multipoint, synchronization, petri nets.

(Received January 4^{th}, 2012 / Accepted September 1^{st}, 2012)

1 Introduction

The real time applications like video conferencing, teleteaching, telemedicine etc. are increasing day by day, results a drastic change in mode of communication in society and also increase the use of IP network. The real-time distributed multimedia systems are characterized by one or several sources transmitting (unicast or multicast) multimedia streams to one or several receivers, playing one or several of the streams at receiver side. Media can be categorized in static media and continuous media. Differ from static media; Continuous media (video and audio) have well-defined temporal relationships between subsequent Media Data Units (MDUs). After capturing media from different sources, it is digitized and packetized and send through the IP network from sender to receivers. Due to nondeterministic delay in IP network, jitter is introduced to the media packet, results change in order of subsequent media packet at receiver side. To maintain the order with temporal relationship between MDUs same as sender side, synchronization among the media packets at receiver side is necessary. Different types of multimedia synchronization are described below.

1.1 Intrastream synchronization

Intra-stream synchronization [10], [1] refers to the temporal relationship between the MDUs of one time-dependent media stream. Different buffer control mechanism [19], [24] are used for achieving such synchronization. Suppose a video sequence was captured at a generation rate of 25frames/s i.e. each frame has to be displayed for 40 ms in the visualization device. After reaching at the receiver, the frame will be stored in a reception buffer to guarantee the intra-stream synchronization.

1.2 Inter stream synchronization

Interstream synchronization [1] refers to the temporal relationship between the MDUs of different (time-dependent or non-time dependent) media streams. Lip synchronization is a common example of interstream synchronization between the audio stream and the associated lip movements in video stream [3]. Interstream synchronization can be classified in three ways- point synchronization, real time continuous synchronization and adaptive synchronization [13].

1.3 Multipoint synchronization

Multipoint synchronization [1],[16] refers the playout of media streams to all the receivers at the same time in multicasting scenario to achieve fairness among receivers. We can cite an example of tele-quiz application where the same video frame is needed to display at same time to all the participating nodes. Other examples of such applications are teleteaching, teleconferencing, e-meeting etc. There exist some protocols [4] that are proposed to support such type of application.

In our previous work [18] we have already proposed an algorithm to achieve multipoint synchronization for single media stream and also extended that work in [17] where we proposed an algorithm to achieve multipoint synchronization for specifically one audio and one video stream. In the proposed study we extend our previous work.

1.4 Petri nets

Multimedia systems are very complex. Modeling is required for effective implementation of it. A common graphical tool used to model concurrent systems is a petri net [26]. To satisfy the requirement to model the system more specifically basic petri net definition is extended in Color petri net(CPN) [9], Dynamic petri net(DPN) [12], Stochastic Petri net [14], Generalized stochastic petri net [15] etc.

1.5 Organization of paper

In this paper, some related research papers is reviewed with their shortcomings in section 2, section 3 presents scope of our work in related research field, the problem statement of multipoint multimedia synchronization is explained formally in section 4. Some metrics are defined in section 5, section 6 represents the inter media specification, self modifying stochastic petri net tool is defined in section 7, the multipoint synchronization issue is modeledin section 8 and generalized algorithm is given in section 9, section 10 represents the simulation results with discussion and at last conclusion is drawn in section 11.

2 Related works

Instead of defining a new protocol, [22] authors have proposed an extension of RTP/RTCP to provide synchronization taking the advantage of feedback capability of RTCP. In this paper group synchronization and inter-stream synchronization was discussed using sender as the Synchronizer Source. Using RTCP RR packet, synchronizer source is able to determine the playout point of the master stream at all the receivers. Authors have proposed swarm synchronization mechanism in [20] using the PTP and RTP and forward error control mechanism is introduced to prevent packet loss. Synchronization of multimedia stream with multiple participants has been addressed in this paper and also in [7]. All the papers discussed above, resolve the synchronization issue by skipping or pausing media stream whenever the asynchrony arises that can lead to important data loss and source control mechanism is used.

A temporal algebra system definition is given in [2] for scheduling a multimedia presentation. Using that definition the synchronized engine generates a scheduling of consistent document to edit and reference temporal constraints and the encoder translate the scheduling to a relative SMIL code. Authors have defined a way to specify the temporal relationship between multimedia but the real time data and multipoint synchronization have not been addressed. In [5] a logical synchronization model is proposed that can specify temporal relationship among the multimedia data and authors have proposed a metric for measuring synchronization error and correction mechanism of synchronization error. But they have not emphasized on the quality of the media.

Confort tool is proposed in paper [16] to achieve multipoint multiple stream synchronization by hybridization of TSPN and PNSVS model and use NTP for clock synchronization. A synchronization agency framework comprising of static and mobile agents and synchronization database is described in paper [13]. Adaptive synchronization mechanism is used in this paper. An algorithm for multipoint multimedia synchronization problems is presented in paper [27]. Effectiveness of algorithm with respect to packet loss is not explicitely measured in all the above works.

Most comprehensive analysis and comparison of the most-known multimedia group and inter-stream synchronization approaches are presented in [1]. Several types of multimedia synchronization are identified and a classification of the main synchronization techniques included in most of the analyzed algorithms complements the paper. Finally, a table is presented summarizing the main characteristics of each analyzed algorithm according to those techniques and other critical issues.

In [4] authors have proposed a protocol that works between session layer and application layer. Clock synchronization algorithm have also proposed using a reference node in distributive manner. But authors have not discussed the synchronization mechanism for multiple streams.

A receiver-based playout scheduling scheme is explained to improve the tradeoff between buffering delay and late loss for real-time voice communication over IP networks in [11]. An important functionality has implemented at receiver is the concealment of lost packet but quality of presentation may compromise.

Virtual-time rendering (VTR) algorithm is introduced in [25] giving priority more on the intra-stream synchronization quality of voice over the interstream synchronization quality between haptic media and voice. Emphasizing on inter and intra stream synchronization Virtual timing model is proposed for synchronize media stream using virtual clock in [12] and adaptive buffering scheme for real time multimedia is proposed in paper [23]. Multipoint synchronization issue has not been discussed in all these paper for multiple receivers.

Authors have explained the Colored Petri Nets in [8] and designed a simple protocol consisting of a sender transferring a number of data packets to a receiver. They also have presented CPN Tool as an industrial-strength computer tool for constructing and analyzing CPN models. But CPN cannot model the reactive scenario in a system and also time parameter is not embedded in CPN. Dynamic Petri Net structure is another extension of basic petri net explained in [21] with control function, control output arc and dynamic place to model iteration in system and event driven characteristic of system. DPN is used to design a multimedia orchestration tool with user interaction but it can not model real time scenario. Another extension of CPN is proposed, called SMCPN in [6] for modeling the system that handle user manipulations and network events such as network congestion. Authors have modeled self-modifying protocol that can change by the systems while communicating using SMCPN. SMCPN can model the non-determinism without human intervention but cannot measure the non deterministic system performance.

3 Scope of works

In the literature review we found a number of authors addressed multipoint synchronization issue for real time multimedia communication. Most of them resolved this synchronization problem by arbitrary skipping pausing in playout of different media streams. As a result the valuable media data (video frame, audio frame etc) may be lost, which leads to degradation of media presentation quality at receiver end.

There are also some algorithms in literature for achieving multipoint synchronization which are centralized in approach that can create a bottleneck for the system in long run.

In this paper we extend our work [18],[17] and try to model the synchronization issue using petri net and find some incapability of existing petri net while analyzing multimedia system model quantitively. So there is a need to extend the existing petri net by introducing some new elements within it. Then we model the synchronization issue using newly defined petri net model and analyze it.

We also propose an algorithm that satisfies the distributed approach for multipoint synchronization in multiple streams and also define two metrics that can measure the performance of the synchronization algorithm.

4 Formal definition of problem statement

Let there are n number of nodes communicating in multicasting scenario. A node can send up to m number of streams.

N_i represent the i^{th} participating node where $1 \leq i \leq n$. $S(N_i)$ represents node N_i is sender where

$1 \leq i \leq n$. R(N_i) represents node N_i is receiver where $1 \leq i \leq n$. We take the multicasting scenario, $\exists$ N_i $\forall$ N_j S(N_i)$\rightarrow$R(N_j) where $1 \leq i,j \leq n$ and $i \neq j$. S_{ix} represents x^{th} media stream sent by node N_i where $1 \leq i \leq m$ and S(N_i) = true. t_{ix}(p) represents time at which p^{th} frame of S_{ix} starts transmitting. d_{ijx}(p) represents delay introduced in p^{th} frame of S_{ix} at N_j where $1 \leq i \leq m$, S(N_i)= true, R(N_j)= true, $1 \leq i, j \leq n$ and $i \neq j$. at_{ijx}(p) represents arrival time of p^{th} frame of S_{ix} at N_j where $1 \leq x \leq m$, S(N_i)= true, R(N_j)= true, $1 \leq i, j \leq n$ and $i \neq j$.

Due to non deterministic delay in IP network, it may happens that $\forall$x$\exists$p d_{ijx}(p)$\neq$$d_{ijx}$(p) that implies $\forall$x$\exists$p at_{ijx}(p)$\neq$$at_{ijx}$(p) [arrival time = transmitting time + dealy in network] where S(N_i)= true, R(N_j)= true, R(N_k)= true and $1 \leq i, j, k \leq n$, $i \neq j$, $i \neq k$. pt_{ijx}(p) denotes presentation time of p^{th} frame of S_{ix} at N_j where $1 \leq x \leq m$, S(N_i)= true, R(N_j)= true and $1 \leq i, j \leq n, i \neq j$.

So we need to find adj_{ijx}(p)= the adjusting time(for skipping or pausing) of p^{th} frame of S_{ix} at N_j where $1 \leq i \leq m$, S(N_i)= true, R(N_j)= true and $1 \leq i, j \leq n$, $i \neq j$; such that $\forall$ p, x, j, k pt_{ijx}(p) = pt_{ikx}(p) where $1 \leq i, j, k \leq n$, $i \neq j$,$i \neq k$ and S(N_i)= true, R(N_i)= true, R(N_k)= true.[pt_{ijx}(p)=t_{ix}(p)+d_{ijx}(p)+adj_{ijx}(p), pt_{ikx}(p)=t_{ix}(p)+d_{ikx}(p)+adj_{ikx}(p)]. R is a set that hold type of relation between MDUs of two different streams. A function rel(S_{ix}(p), S_{iy}(p)) maps relation of p^{th} frame of different stream S_{ix} and S_{iy} to set R. If N_i where S(N_i)= true , send more than one stream then we need to find adjustment time such that $\forall$ p, x, j, k pt_{ijx}(p) = pt_{ikx}(p) where $1 \leq i, j, k \leq n$, $i \neq j$,$i \neq k$, $1 \leq x \leq m$ and S(N_i)=true, R(N_j)= true, R(N_k)= true and also $\forall$p, x, y r(S_{ix}(p), S_{iy}(p))$\rightarrow$R holds, where $1 \leq x, y \leq m$ and $x \neq y$.

5 Metric definition

Two metrics are defined here to measure the performance of the synchronization algorithm with respect to loss of data and asynchrony among receivers.

5.1 Loss metric

Loss metric(Ml) can measure the percentage of loss at receiver end with respect to the MDU received at receiver side.

Loss can be occurred due to network as well as the synchronization process. Let R_i, N_i represents the number of packet received and played for ith stream during time t respectively. So,loss metric

$$Ml = \frac{\sum_{i=1}^{m}(R_i - N_i)}{\sum_{i=1}^{m}(R_i)} \times 100\%$$

5.2 Asynchrony metric

Two types of asynchrony metrics are defined here, relative asynchrony and overall asynchrony.

5.2.1 Relative asynchrony

The playout time difference of each packet for all media streams at one receiver with respect to playout time of the packet to another receiver is defined as the relative asynchrony between two receivers.

Let P_{ij} represents number of packet played out at receiver side in i^{th} receiver for j^{th} stream during time t. Pt_{ij}(p) represents playout time of p^{th} packet at receiver side in i^{th} receiver for j^{th} stream. Relative asynchrony between two receivers i,k is measured by the equation -

$$\left|\frac{\sum_{i=1}^{m}\sum_{p=1}^{p^{ij}}(P_{ij}(p) - P_{kj}(p))}{\sum_{i=1}^{p^{ij}}(P_{ij})}\right| \forall \text{ i,j where } 1 \leq i, k \leq n \& i \neq k$$

5.2.2 Overall asynchrony

The playout time difference of each packet for all media streams at all receivers with respect to standard playout time (expected playout time of the packet calculated by some algorithm) of the packet in the system is defined as the overall asynchrony of the system. Standard playout time of packet p in j^{th} stream for i^{th} receiver is $Pstd_{ij}$(p). So, overall asynchrony of system is -

$$\sqrt{\frac{\sum_{i=1}^{n}\sum_{i=1}^{m}\sum_{p=1}^{p^{ij}}(Pstd_{ij}(p) - P_i j(p))}{\sum_{i=1}^{n}\sum_{i=1}^{p^{ij}}(P_{ij})}}$$

6 Intermedia specification

There is well defined temporal relationship between MDUs of continuous media as well as static media of different media streams. The well defined logical representation within MDUs of inter media is established according to their temporal relationship. The possible relationship of two MDUs may be within two continuous media or one continuous and one static media or both can be static. We define a specification that can represent all possible relationship holds between two MDUs of different media streams. Size of MDU may or may not be same for two different media streams. Three types of relations can hold between MDUs - precedes, succeeds and overlaps. Relations are defined below using two MDUs of different media streams denoted as A and B. ts(A) and d(A) represent the starting time and duration of presentation of frame A.

6.1 A $\rightarrow_P$ B

This relationship holds when A finishes its playout before starting the playout of B. Logical representation of the condition is ts(A) + d(A) $\leq$ ts(B).

6.2 A $\rightarrow_O$ B

This relationship holds when A and B satisfy one of the following three conditions.
First condition:

A and B both start and finishes play out at the same time or A starts play out after the starting of B but finishes before the end of play out of B or A starts play out after starting of B but both finish together.
Second condition:

Frame A starts its play out after the start of B but ends after the end of B.
Third condition:

B starts its play out after the start of A but ends after the end of B.

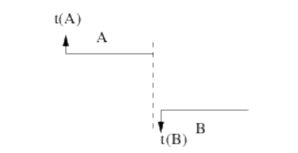
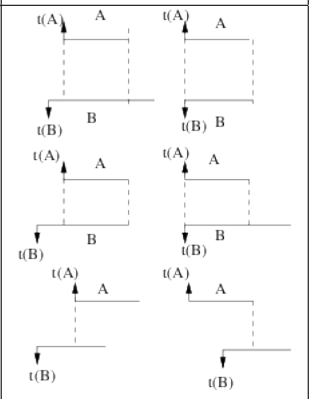
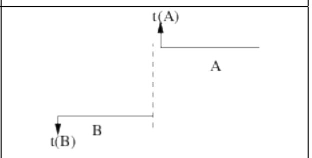
Logical representation of the condition is (ts(A)= ts(b) && ts(A)+d(A)=ts(B)+d(B)) || (ts(A)$\geq$ts(b) && ts(B)+d(B)$\geq$ts(A)) || (ts(A)$\leq$ts(b) && ts(A)+d(A)$\geq$ts(B)).

6.3 A$\rightarrow_S$B

This relationship holds if A starts its playout after the end of play out of B. Logical representation of the condition is ts(A) $\geq$ ts(B) + d(B).

Table 1 represents all posible pictorial representations of intermedia relationship and corresponding temporal relation.

Table 1: Intermedia specification.

Relation	*pictorial representation*
A$\rightarrow_P$B	
A$\rightarrow_O$B	
A$\rightarrow_S$B	

7 Self modifying stochastic color petri net

Self Modifying Stochastic Color Petri Net is proposed as follows:

SMSCPN has 9 tuple {P, T, A, λ, C, I, Tn, F,CL}.
P: {p_1, p_2,.., p_x} where $x \geq 0$, is a finite set of places. P = P_N U P_F where P_N is the set of places where no function is executed in arrival of resource token and P_F is the set of places where some function is executed in arrival of resource token.

T: {t_1, t_2,..,t_m} where $m \geq 0$, is a finite set of transitions. T = $T_I \cup T_T$ and $T_I \cap T_T = \varnothing$ where, T_I is the set of immediate transitions that fire the token immediately when token is available at input place. T_T is the set of timed transitions that take some time to fire tokens from input place to output place.

A: $(P \times T) \cup (T \times P)$ is the finite set of arcs. A = $I^- \cup I^+ \cup I^h$ and $(I^- \cup I^+) \cap I^h = \varnothing$ where, I^- refers finite set of input arcs and $I^- \subseteq$ (P $\times$ T). I^+ refers finite set of output arcs and $I^- = T \times P$. I^h refers finite set of inhibitor arcs i.e. if input place does not hold any token then transition is enabled to fire and a token is produced in output place and $I^h \subseteq$ P $\times$ T.

λ: $\lambda_1, \lambda_2, .., \lambda_j$, where $j \geq 0$ and j= |T_T| is a finite set of transition rate assigned to timed transition.

C: {c_1, c_2,.., c_n} where n $\geq$ 0, is a finite sets of commands.

I: {i_1, i_2,.., i_k} where k $\geq$ 0, is a finite set of informations flow through net.

T^n : $\{(C^* - \varepsilon) \cup P(I)\}$, is a finite set of token. T^n = CT $\cup$ RT and CT $\cap$ RT = $\varnothing$ where, CT = {($C^* - \varepsilon$) } is a set of color token. RT = P(I) is a set of resource token.

F: {f_1, f_2,.., f_l} where $l \geq$ 0is a set of functions that execute in P_F when a resource token is arrived at the place. Function can generate color token, modify the information in resource token, add new information to resource token i.e. change resource token.

CL: is the finite set of clock.

In the propose model places are represented by the circle, timed and immediate transitions are represented by the rectangle and bar, input and output arcs are represented by arrow, inhibitor arcs are represented by circle headed arrow, place, arc and transition created by color token is represented by dashed line.

Set of commands with particular sequence represent by the color token. It can change the net structure temporarily to accommodate with new environment and control the flow of resource token. Table 2 represents list of commands execute on different component of petri net in the model.

Table 2: Command executed in SMSCPN.

$Component\ of\ petri\ net$	$Command$
Place	Create a place delete a place
Transition	Enable transition Disable transition Create transition Delete transition
Arc	Create a arc delete a arc
Clock	Set the clock value count down the clock

Transition rules: Let S= {s_1, s_2,.., s_n}where $n \geq 0$, is the set of types of information(I). M: I$\rightarrow$S is a function that maps information to a type. Now α: RT$\rightarrow$X where X = {$S^* - \varepsilon$}, is a function where $\forall rt \in RT, \alpha(rt) = x$ where x $\in \{S^* - \varepsilon\}$ and if rt= {i_1, i_2,.. $i_{|x|}$} then x= {M(i_1), M(i_2),.. M($i_{|x|}$)}. $\beta : A \rightarrow$ {$S^* - \epsilon$} is a function that binds arc to a type of resource token. A resource token can transit through a arc if β(x) = α(rt) where $a \in A$ and rt$\in$ RT.

δ: CT $\times$ RT$\rightarrow$ CL is a function that maps color token combined with resource token to a clock for timed transition. When clock value goes down to zero, the corresponding resource token transits to output place. When resource token transits from input place, color token that is created at that place is deleted itself and also changes done by color token is removed from net.

8 Multipoint mulimedia synchronization model

In this section we model the scenario where three types of media streams are received at receiver node using the proposed SMSCPN tool. Here we take all possible case of arrival of media streams and show how to process these streams such that multipoint synchronization is achieved among receivers with interstream synchronization at each node. There are some constrains to maintain real time interactive multimedia synchronization scenario. The maximum tolerable delay for interactive communications is 250ms [16], refer as primary delay constrain and acceptable asynchrony within audio video stream is 10ms [16]. In our model control message is available at receiver side if the maximum delay between sender to all receivers is within primary delay constrain otherwise more QoS support is demanded for that receiver. Reference delay that is used to synchronize all receiver, calculated from that maximum delay value in control message. In our model I={audio, video, static media, network delay, time of arrival, duration of presentation, reference delay, generation time, waiting time} and S={a, v, s, d, arr, du, dref, ger, wt}. Different resource token used in this model are of the following types.

x1=(a,d,arr,du); x2=(dref); x3=(x1,x2); x4=(x3,wt); x5=(v, d, arr, du); x6=(x5,x2); x7=(x5,wt); x8=(x1,x5); x9=(s, d, arr, du); x10=(x9,x2); x11= (x9,wt); x12 = (x1,x9).

8.1 Multipoint synchronization model for multiple media streams

In Figure 1 the model for multipoint synchronization is shown for one audio,video and static media. It consists of 25 places and 36 transitions. Here P_F = {p_2, p_3, p_4, p_5, p_6, p_{10}, p_{11}, p_{18}, p_{17}, p_{16}, p_{15}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}} and P_N = {p_1, p_8, p_7, p_9,p_{12}, p_{19}, p_{20}, p_{25}}. Transition t_1 transits at arrival rate of audio frame receive from internet. Place p_1 acts as the receiving buffer. When token is available at p_1 and control message is available at p_7 then resource token is taken one by one for processing in place p_2. Hence inhibitor arc is used at transition t_2. Function f_1 executed at place p_2 is-

while *resource token arrives* **do**
  if *delay of MDU* $\leq$ *reference delay* **then**
    create color token ct_1;
  else
    create color token ct_2;
  end
end

ct_1 executes the following commands
1. Disable transition t_4
2. Enable transition t_3

ct_2 executes the following commands
1. Disable transition t_3
2. Enable transition t_4

If transition t_3 fires then resource token transits to p_3. Function f_2 executed at p_3 is-

If transition t_4 fires then resource token transits to p_4. Function f_3 executed at p_4 is-

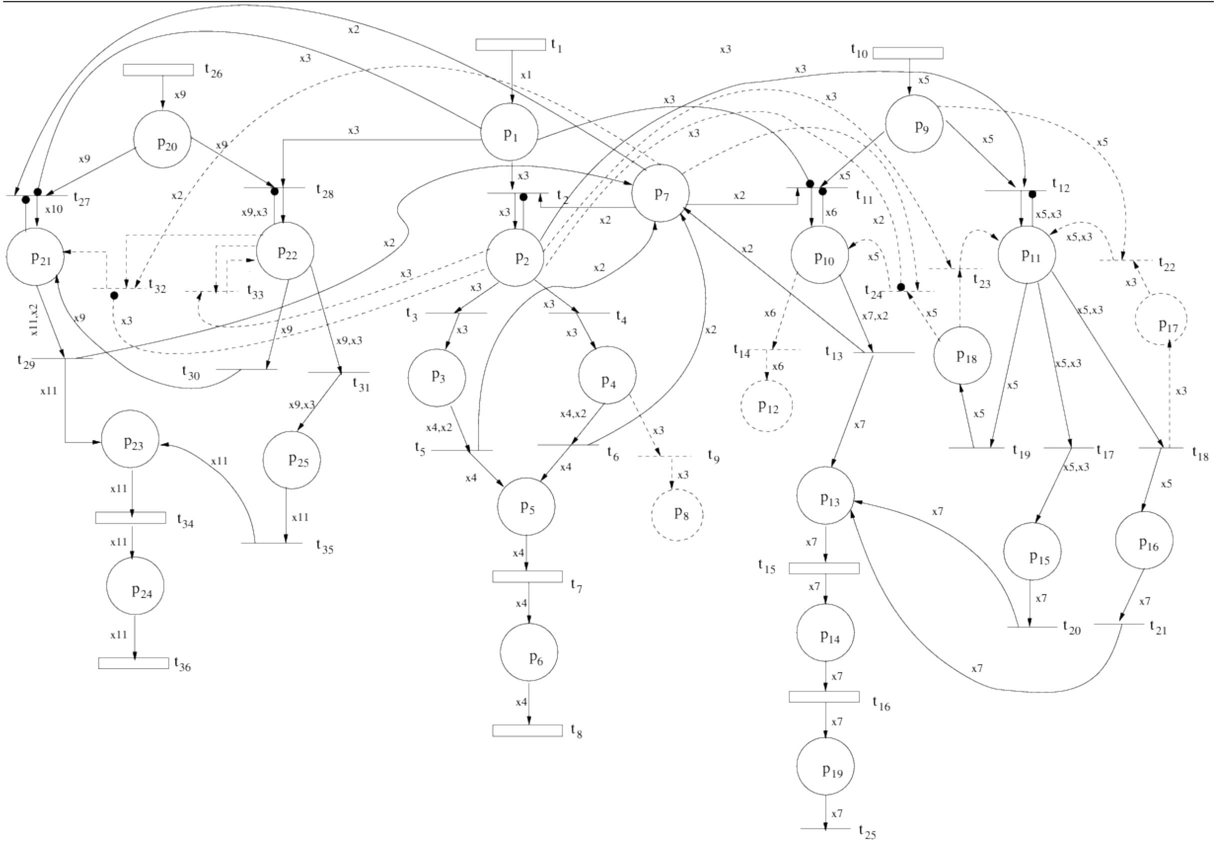


Figure 1: Multipoint synchronization model.

while *resource token arrives* **do**
  calculate waiting time of MDU;
  Create resource token of type x4;
end

while *resource token arrives* **do**
  if *delay of MDU≤reference delay+10ms≤250ms* **then**
    calculate waiting time of MDU;
    Create resource token of type x4;
  else
    create color token ct_3;
  end
end

Color token ct_3 executes the commands-
1. Disable transition t_6
2. Create place p_8
3. Create transition t_9
4. Create arc p_4t_9, t_9p_8

When t_5 or t_6 is fired resource token transits to p_5 and p_7 according to type bound to the arc. Function f_4 executed at place p_5 is-

while *resource token arrives* **do**
  create color token ct_4;
end

Color token ct_4 executes following commands-
1. Set a clock to the transition with information of waiting time associate with resource token.
2. Count down the clock.

After firing of t_7 token transits to place p_6 and function f_5 executed at p_6 is-

while *resource token arrives* **do**
  create color token ct_5;
end

Color token ct_5 executes the following commands-
1. Set the clock to transition t_8 with fixed value of frame duration.

2. Count down the clock.

When transition t_8 is fired then token leaves the net that means the media frame is played out synchronously at receiver side.

Transition t_{10} transits at arrival rate of packet received from internet. Place p_9 stores the resource token coming from network. Now video stream is synchronized at multipoint in absence of audio stream otherwise it is synchronize with the audio stream. According to normal transition rule either transition t_{11} or t_{12} fires. Transition t_{11} is fired if audio is not available at p_1 and resource token transit to p_{10}. Function f_6 executed at p_{10} is-

while *resource token arrives* **do**
  if *delay of MDU* $\leq$ *reference delay + 30ms* **then**
    calculate waiting time of MDU;
    Create resource token of type x7;
  else
    create color token ct_6;
  end
end

Color token ct_6 executes some commands-

1. Disable t_{13}
2. Create place p_{12}
3. Create transition t_{14}
4.Create arc $t_{10}p_{14}$ and $t_{14}p_{12}$

When transition t_{14} is fired, the token transits to new created place p_{12}, that means system demands new control message for continuing the communication. When t_{13} is fired resource token transits to p_{13} and p_7 according to type bound to the arc. If t_{12} is fired in presence of audio frame at p_1 video resource token transit to place p_{11} when audio resource token come to place p_2 for processing. Function f_7 executed at place p_{11} is-

while *resource token arrives* **do**
  if *token holds succedes relation with audio token* **then**
    create color token ct_{10};
  else if *token holds precedes relation with audio token* **then**
    create color token ct_8;
  else
    create color token ct_{12};
  end
end

Color token ct_8 executes some commands-

1. Disable t_{17} and t_{18}
2. Enable t_{19}

Color token ct_{10} execute commands-

1. Disable t_{19} and t_{17}
2. Create place p_{17}
3. Crate arc $t_{18}p_{17}$
4. Enable t_{18}

Color token ct_{12} executes commands-

1. Disable t_{19} and t_{18}
2. Enable t_{17}

According to the color token created at place p_{11} transition t_{19} or t_{17} or t_{18} is fired and token transit to p_{18} or p_{15} or p_{16} and p_{17} respectively. At p_{18} function f_8 is executed.

while *resource token arrives* **do**
  create color token ct_9;
end

Color token ct_9 executes following commands-

1. Create transition t_{23} and t_{24}.
2. Create an inhibitor arc p_2t_{24} and normal arc p_7t_{24}, p_2t_{23}, $t_{24}p_{10}$, $t_{23}p_{11}$, $p_{18}t_{24}$, $p_{18}t_{23}$.

If t_{23} is enable then token again compared with next audio token otherwise t_4 is enable and token transit to place p_{10} and process accordingly. Function f_9 executed at place p_{17} is-

while *resource token arrives* **do**
  create color token ct_{11};
end

Color token ct_{11} performs following commands

1. Create transition t_{22}
2. Create arc p_9t_{22}, $t_{22}p_{11}$, $p_{17}t_{22}$.

After firing of t_{22} token transits to p_{11}. Function f_{10} executed at p_{15} and p_{16} is-

while *resource token arrives* **do**
  calculate waiting time of MDU;
  Create resource token of type x7;
end

When t_{13} or t_{20} or t_{21} is fired token moves to place p_{13}. Function f_4 is executed at that place. After firing of t_{15}, the token moves to place p_{14} and f_{11} is executed given below.

while *resource token arrives* **do**
  create color token ct_7;
end

Color token ct_7 executes following commands-

1.Enable transition t_{25}.

2.Set the clock to transition t_{16} with value of frame duration.

3.Count down the clock.

When t_{16} is fired resource token transit to place p_{19} and block until t_{25} is enable.

Transition t_{26} is fired if static media data is arrived from Internet. Place p_{20} is stored the resource token coming from network. Now this media data can be synchronized at multipoint in absence of audio stream otherwise it is synchronized with the audio stream. Transition t_{27} is fired if token is available at p_1. After occurrence of t_{27}, resource token transits to p_{21}. f_{12} is executed at p_{21}.

while *resource token arrives* **do**
  calculate waiting time of MDU;
  Create resource token of type x11;
end

When t_{29} occurs, token moves to p_7 and p_{23} according to type bound with arc. If t_{28} is fired, resource token transit to place p_{22} when audio resource token come to place p_2 for processing. At p_{22} function f_{13} is executed.

while *resource token arrives* **do**
  if *token holds succedes relation with audio token* **then**
    create color token ct_{13};
  else if *token holds precedes relation with audio token* **then**
    create color token ct_{15};
  else
    create color token ct_{14};
  end
end

Color token ct_{13} executes following commands-

1.Create transition t_{32} and t_{33}

2.Create a inhibitor arc p_2t_{32} and normal arc p_2t_{33}, p_7t_{32}, $t_{31}p_{21}$, $t_{32}p_{22}$, $p_{22}t_{32}$, $p_{22}t_{33}$.

Color token ct_{14} executes few commands-

1.Disable transition t_{30}

2.Enable transition t_{31}

Color token ct_{15} executes few commands-

1.Disable transition t_{31}

2.Enable transition t_{30}

After execution of commands by color token if transition t_{13} is fired the token transit to place p_{25}. At that place f_{12} is executed. When t_{29} or t_{35} is fired resource token transits to p_{23} and f_4 is executed. When t_{34} is fired, resource token transits to p_{24} and f_5 is executed at that place. When t_{36} is fired token leaves the net.

8.2 Probabilistic analysis of the model

Using self modifying stochastic color petri net we can model the stochastic nature of the system. We can resolve the conflict between two transitions by imposing probability to the transitions. In our model for audio, let token is arrived maintaining the Poisson process at an average rate α then in time interval t probability that there is a token in place p_1 is $P_{P1}=\sum_{k=1}^{\infty} \frac{\alpha e^{\alpha t}}{k!}$. As delay in network layer follows the normal distribution with mean and variance μ and σ respectively, control message available at p_7 i.e. there is a token at p_7 is $P_{P7}= \int_{dmin}^{250} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\, dx$. Rate of transition t_2 is 1 but it is marking dependent. So the probability that there is a token at p_2 is same as probability of (P_{P1}. P_{P7}). Then there is a conflict between t_3 and t_4. Now t_3 is enabled when network delay incurred to the token is less than the reference delay. So probability of enabling transition t_3 is x= $\int_{dmin}^{dref} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\, dx$. Now transition t_4 occurs when delay is larger than ref delay i.e. the probability of enabling transition t_4 is y = $\int_{dref}^{\infty} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\, dx$. So there is a token in p_3 with probability P_{P3} = (P_{P2}.x). Rate of transition t_5 is 1, only depend on presence of token in p_3. Similarly the Probability that there is a token at p_4 is P_{P4} = P_{P2}.y . Here probability of enabling the transition t_6 is z= $\int_{dref+10}^{dref} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\, dx$ and probability of enabling transition t9 is $\int_{dref+10}^{\infty} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\, dx$.So probability that there is a token at place p_5 is P_{P5}=(P_{P3} + P_{P4}.z). Network delay distribution for particular source and destination in IP network follows the normal distribution. In our scenario we shape the play out time according to a fixed reference delay for a session (until the reference delay information is changed). So the time for waiting before presentation is also maintained normal distribution with mean and variable $\mu 1$ and $\sigma 1$ respectively. So the mean time of transition t_6 is tt1= $\int_{0}^{dref+10-dmin} \frac{e^{\frac{(x-\mu 1)^2}{2\sigma 1^2}}}{\sqrt{2\pi\sigma 1^2}}\, xdx$. So rate of transition at t_6 is R1 = P_{P5}/tt1. So probability of presence of a token at place p_6 is P_{P6}= e^{-R1t}. At place p_6 token is blocked for fixed time that equals to the time is taken by an audio MDU to generate at sender side. If generation rate at sender side is R2 then the final rate at which the token is departed i.e. the rate of transition t_8 is P_{P6}.R2. Also taking measure of arrival rate at t_1 and departure rate at t_8 we can calculate buffer size such that no loss occurs due to shortage of buffer.

We can analysis the video stream in same manner. For same receiver let token arrives from network at rate β i.e. transition rate of t_{10} isβ. As arrival process of data stream is followed Poisson process so p_9 hold at least one token within t time with probability P_{P9}=$\sum_{k=1}^{\infty} \frac{\beta e^{\beta t}}{k!}$. Now transition t_{11} or t_{12} is fired depending on the resource token available in place p_1. So probability that a token is in place p_{10} is P_{P10} = P_{P9}.(1- P_{P1}).P_{P7}. Now transition t_{13} occurs if delay information carried by token is within reference delay addition with 30ms. So probability within this range is a= $\int_{dmin}^{dref+30} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\, dx$ and and t_{14} is fired with probability b= $\int_{dref+30}^{\infty} \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\, dx$.The system is failed with probability P_{P10}.b. Now t_{12} is fired when audio master stream available. So probability of enabling the transition t_{12} is (P_{P1}.P_{P7}.P_{P9}). For inter-stream synchronization we do not need any delay information for finding the presentation time of MDU. We only consider the relationship information and delay can be in any range. As no token is going to place $p_{12}$from place p11 probability that a token in place p_{13} is P_{P13}=(P_{P10}.a + P_{P11}). likewise audio the time for waiting before presentation for video is also maintained normal distribution with mean and variable $\mu 2$ and $\sigma 2$ respectively. The token can block for mean time tt2=$\int_{0}^{dref+30-dmin} \frac{e^{-\frac{(x-\mu 2)^2}{2\sigma 2^2}}}{\sqrt{2\pi\sigma 2^2}}\, x dx$. So rate of transition t_{15} is R3=(P_{P13}/tt2). So probability of a token in place P_{P14} is e^{-R3t}. If mean and variance of presentation time is $\mu 3$ and $\sigma 3$ then the mean delay of transition t_{16} is tt3 = $\int_{0}^{30} \frac{e^{-\frac{(x-\mu 3)^2}{2\sigma 3^2}}}{\sqrt{2\pi\sigma 3^2}}\, x dx$. So rate of transition t_{16} is R4 = P_{P14}/tt3. Probability of presence of a token at place P_{P14} is e^{-R4t}. Now transition t_{25} is enabled with probability of (1- e^{-R3t}) and it is an immediate transition. So final rate of transition t_{25} is P_{P14}(1- e^{-R3t}). We can measure buffer size from this analysis similarly as audio stream.

We can analyze the static media in same way as we are analyzed the video frame.

9 Algorithm for multipoint synchronization

According to the proposed model for multipoint Synchronization two things are considered, maintaining the quality of presentation of media stream with minimum significant data loss in multipoint synchronization process and using dynamic delay information to synchronize the multipoint as mean delay between sender and receivers vary with time.

9.1 Initial and periodic synchronization

For clock synchronization which is done initially and at a regular interval we can use NTP algorithm at the server end, NTP date at receiver end. The NTP service is provided by the network server located at the Internet. For resolving the asynchrony between different receivers during the session we use skipping or pausing but not in periodical manner. We only skip the data in the silent part so that we can protect the loss of significant data.

9.2 Delay and expected playout time calculation

In our algorithm to calculate the expected playout time at receiver end we need two parameters, one is the maximum delay at that moment among all delay between sender and receiver and another is extra minimum buffering time use for de-jittering. A delay matrix is given in Table 3 store the most updated information about network delay between all nodes in a scenario. The information of delay matrix is used to decide whether a node can able to continue in conferencing scenario or not. If the delay between sender and receiver cross the delay constrain then the receiver cannot continue with conferencing scenario and should take necessary step for increasing the quality of service in network layer.

Table 3: Delay Matrix.

X	N_1	..	N_j	..	N_n
N_1	X	..	D_{1j}	..	D_{1n}
:	:	:	:	:	:
N_i	D_{i1}	:	x	:	D_{in}
:	:	:	:	:	:
N_n	D_{n1}	:	D_{nj}	:	x

Consider the scenario of an audio conferencing session consisting of n nodes, represented by N_1, N_2, ..., N_n where any of the n nodes may act as the sender. For any sender, all the receivers have realized separate amount of network delay that can vary with time and store in delay matrix. Each row of the matrix represents the amount of delay between a sender to all receivers. D_{ij} represents the delay between the sender N_i to receiver N_j. Whenever the sender sends a data packet, the entries of the corresponding row is continuously updated.

D_{imin} = min(D_{ij}) for all j=1,2,..n from N_i and D_{imax} = max(D_{ij}) for all j=1,2,..n from N_i. We assume the source starts the transmission at time t_0. Network delay can cause the receiver to start its play out process as early as t_0+D_{imin} or as late as t_0+D_{imax} that causes an initial asynchrony of at most D_{imax}-D_{imin} between each pair of receivers. Let ρ be the drift in the playout rate. Let $\triangle$ be the additional time added to allow the initial MDUs of each stream to arrive and be buffered at all receivers. $\triangle$=(D_{imax}- D_{imin})/(1- ρ) Then if t_0 is the time at which the packet is sent, then the expected playout instant of each of the packet is calculated as $T_{expected}$=t_0+D_{imin}+$\triangle$, where $T_{expected}$ is the time represents the time at which the packet has to be played and reference delay is equals to (D_{imin}+$\triangle$).

9.3 Synchronous playout algorithm

Synchronous play out algorithm calculates the time that a frame must wait before presentation and the playout time of the frame using the expected playout time calculation. Stream that is synchronized at multipoint, known as master stream and streams that is synchronized at multipoint according to master stream, known as slave stream and this mechanism of synchronization is called master-slave mechanism. This mechanism is used in our algorithm to achieve multipoint synchronization for multimedia in a multicasting scenario. For different streams there is different receiving buffer for receiving the data from Internet. Each receiver follows the Algorithm 1.

A thread is started for each slave stream during execution of Algorithm 1 that process the MDU of slave stream using intermedia specification between them given in Algorithm 2.

All the functions call from Algorithm 1 and Algorithm 2 use following notation:

Delay information of audio frame (delay),
Time of arrival of audio frame (arr),
Reference delay of the system (dref),
Time of arrival of video frame (arr),
Waiting time of the frame of master stream (wt_m),
Arrival time difference between video frame and corresponding fame of master stream (atd),
Generation time difference of video frame and corresponding fame of master stream (std),
Duration of video frame (du).

The informations produce from the function is
Presentation time(pt) and Playout duration(du).

while *conference is goning on* **do**
  if *audio stream is available* **then**
    Select any one of audio stream as master stream.;
    Take audio MDU of master stream from buffer ;
    process-audio-frame ;
    for *each different stream available* **do**
      Start a thread to process MDU of slave stream;
    end
  else if *video stream is available* **then**
    Select video stream as master stream.;
    Take video MDU of master stream from buffer;
    Call process-video-frame ;
    for *each different media available* **do**
      Start threads to process MDU of slave streams;
    end
  else
    Select any static media as master stream.;
    Take MDU of master stream from buffer ;
    Call process-static-frame ;
    for *each different media available* **do**
      Start a thread to process MDU of slave stream;
    end
  end
end

Algorithm 1: Synchronous playout algorithm.

10 Results and discussion

We took a scenario where three receivers were receiving audio data and mean delay was different for different receivers. Delay was generated randomly for each packet. We simulated the scenario using our algorithm and also without our algorithm (only each receiver use the buffer for de-jitter mechanism) and plotted the playout time against frame number. When presentation time of a frame becomes zero it specifies packet loss. It is clear from Figure 2 that there are seven packet loss in simulation without using our algorithm and presentation time distribution for audio stream is different in different receivers. According to loss metric, for this scenario Ml= $\frac{7}{60} \times 100\%$ =11%. It is noticed from Figure 3 that the presentation time distribution for audio stream in different receivers is merged into one line and there is no loss due to synchronization process while simulating the same scenario with our algorithm.

In scenario1 we took the 1st and 2nd receiver and calculated the relative asynchrony using equation given in section 5. Taking the playout time of frame given by simulation (without using our algorithm) relative asynchrony between receiver1 and receiver2 is equals to 52.87ms which is not less than 10 ms whereas using our algorithm it becomes 0.345ms which is almost zero.

We took another scenario where two receivers were

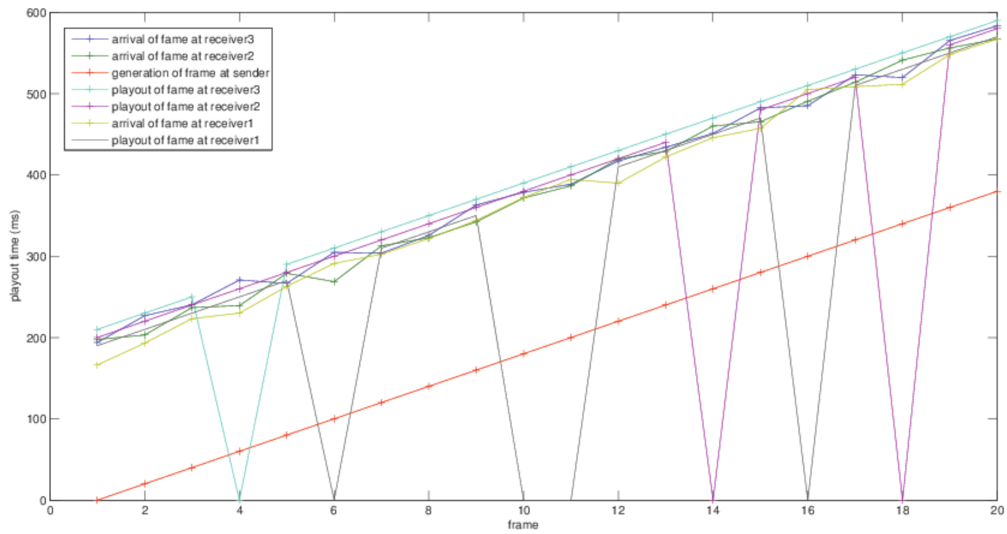


Figure 2: Graph for scenario1 without using algorithm.

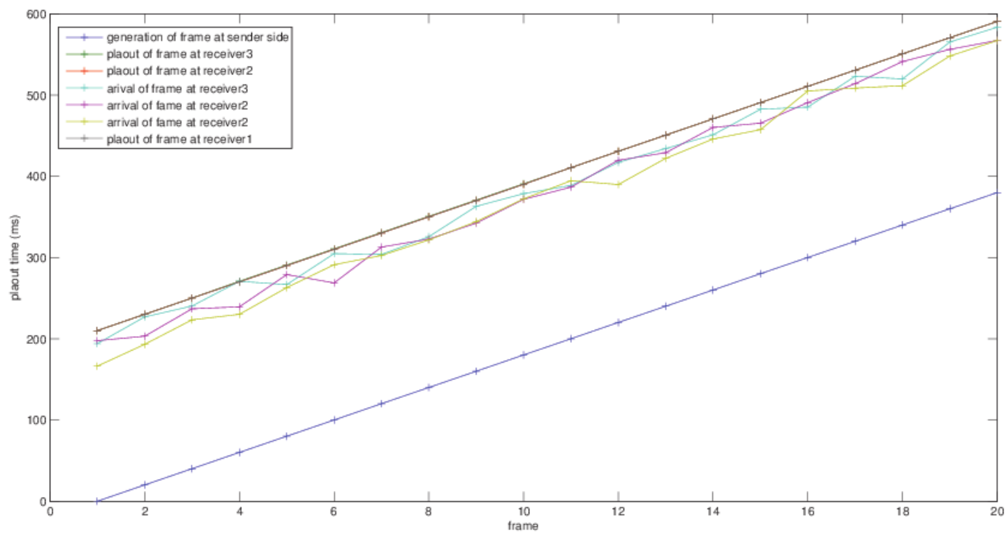


Figure 3: Graph for scenario1 using algorithm.

receiving two continuous media (one audio and another video with different frame size) with different mean delay for different receivers. Delay is generated randomly for each packet. In Figure 4 the presentation time is plotted against frame number for both receivers. The generation time is also plotted to compare the inter relationship between two media streams. From the Figure 4 we can see the presentation time of two different media streams at different receiver is same and also the vertical difference between two different media streams is same as the generation time which proofs that the synchronization process at multipoint does not changes the inter media relationship. In the second scenario, taking the calculated play out time given by simulation we calculate the overall asynchrony using the metric given in section 5. Overall asynchrony value is come in fol-

if *MDU precedes MDU of master stream* **then**
  while *MDU precedes MDU of master stream = true* **do**
    if *stream is audio* **then**
      Call process-audio-frame ;
    else if *stream is video* **then**
      Call process-video-frame;
    else
      Call process-static-frame ;
    end
    Take next MDU;
  end
else if *MDU overlaps MDU of master stream = true* **then**
  if *stream is audio* **then**
    Call process-audio-slave;
  else if *stream is video* **then**
    Call process-video-slave
  else
    Call process-static-slave
  end
else
  return to the buffer
end

Algorithm 2: Thread to process MDU of slave stream.

Require: $flag \leftarrow 0$ and $temp \leftarrow 0$;
diff←(dref-delay);
if *diff≥0 and flag=0* **then**
  $wt \leftarrow diff$;
else if *diff≤-10 and flag=0* **then**
  $wt \leftarrow 0$,$flag \leftarrow 1$,$temp \leftarrow -diff$
else if *diff≥0 and flag=1* **then**
  $wt \leftarrow (temp + diff)$;
else
  if *(temp+diff)≥0* **then**
    $wt \leftarrow (temp + diff)$;
  else
    if *diff≤-10* **then**
      wt ←0, temp←-diff;
    else
      system failed;
    end
  end
end
pt←(arr + wt);

Algorithm 3: Process-audio-frame.

diff←(dref-delay);
if *diff≥0* **then**
  $wt \leftarrow diff$;
else if *diff≤-30* **then**
  $wt \leftarrow 0$, $du \leftarrow (du - diff)$;
else
  system failed;
end
pt←(arr + wt);

Algorithm 4: Process-video-frame.

diff←(dref-delay);
if *diff≥0* **then**
  $wt \leftarrow diff$;
else
  $wt \leftarrow 0$, $du \leftarrow (du - diff)$;
end
pt←(arr + wt);

Algorithm 5: Process-static-frame.

Require: $flag \leftarrow 0$ and $temp \leftarrow 0$;
diff←(wt$_m$-std+atd);
if *diff≥0 and flag=0* **then**
  $wt \leftarrow diff$;
else if *diff≤-10 and flag=0* **then**
  $wt \leftarrow 0$,$flag \leftarrow 1$,$temp \leftarrow -diff$;
else if *diff≥0 and flag=1* **then**
  $wt \leftarrow (temp + diff)$;
else
  if *(temp+diff)≥0* **then**
    $wt \leftarrow (temp + diff)$;
  else
    if *diff≤-10* **then**
      wt ←0, temp←-diff;
    else
      $wt \leftarrow 0$,$du \leftarrow 0$;
    end
  end
end
pt←(arr + wt);

Algorithm 6: Process-audio-slave.

lowing manner- 11ms, 2.14 ms, 3.152 ms, 0ms, 6.43ms in 5 consecutive simulation of second scenario. All the value is very close to zero. So we can conclude that our algorithm is succeeded to solve the problem we addressed.

11 Conclusion

In this work we developed an algorithm that works in application layer to resolve the multipoint synchronization issues in real time multimedia communication. We also define some metrics that verify the algorithm and got satisfactory result regarding of multipoint synchronization and packet loss due to synchronization process.

We need the QoS support in network layer to pro-

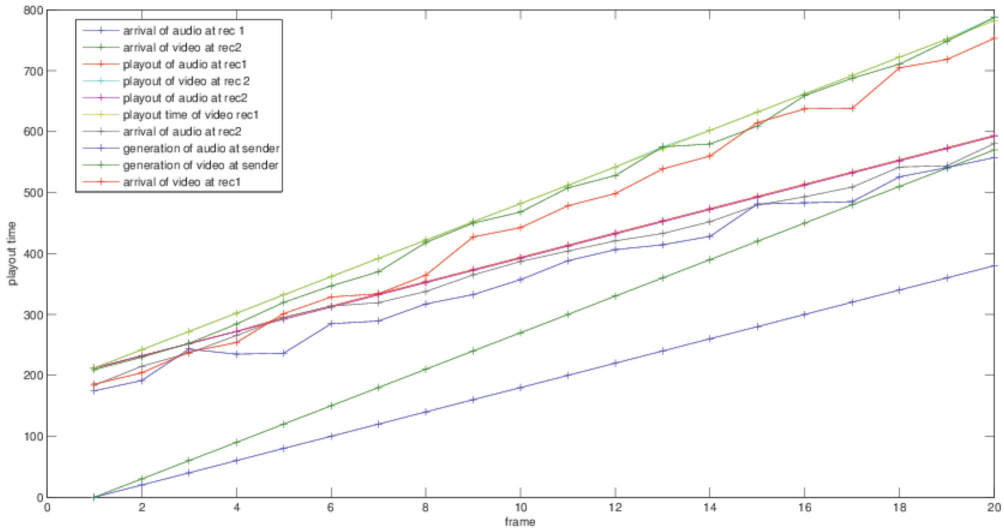


Figure 4: Graph for scenario2.

diff←(wt$_m$-std+atd);
if *diff≥0* **then**
    $wt \leftarrow diff$;
else if *diff≤-30* **then**
    $wt \leftarrow 0$, $du \leftarrow (du - diff)$;
else
    $wt \leftarrow 0$,$du \leftarrow 0$;
end
pt←(arr + wt);
Algorithm 7: Process-video-slave.

diff←(wt$_m$-std+atd);
if *diff≥0* **then**
    $wt \leftarrow diff$;
else
    $wt \leftarrow 0$, $du \leftarrow (du - diff)$;
end
pt←(arr + wt);
Algorithm 8: Process-static-slave.

vide the primary delay constrain. So future works can be extend by finding the proper mechanism to provide QoS for live media streaming and incorporating bandwidth adaptation to fulfill the QoS requirement of user.

References

[1] Boronat, F., Llorat, J., and Garcia, M. Multimedia group and interstream synchronization techniques: a comparative syudy. *Elsevier B.V., Information system*, 34:108–131, 2009.

[2] Chang, A. Y. An intelligent semantic for design of consistent online smil presentation. In *AINA '05 Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 151–156, 2005.

[3] Chang, M. Low-latency lip-synchronized videoconferencing system. In *Conference on Human Factorsin Computing Systems (CHI2003), Fort-Lauderdale, FL,USA5(1)*, pages 465–471, 2003.

[4] Dommel, H. and Verma, S. Multipoint synchronization protocol. In *Systems, Man and Cybernetics, 2004 IEEE International Conference*, volume 5, pages 4631–4635, 2004.

[5] e. Ramirez, S. E. P. H. L. A. M. R. J. and Gomez, G. R. Logical mapping: an intermedia synchronization model for multimedia distributed system. *IEEE Transaction on Multimedia*, 3(5):33–41, Dec 2008.

[6] Guan, S. and Liu, W. Self-modifiable color petri nets for modeling user manipulation and network event handling. *IEEE Transaction on Computers*, 52(7):920–932, July 2003.

[7] Ishibashi, Y. and Tasaka, S. A group synchronization mechanism for live media in multicasting

communication. In *IEEE GLOBECOM'97*, pages 746–752, Nov. 1997.

[8] K. Jensen, L. M. K. and Wells, L. Colored petri nets and cpn tool for modeling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(3):213–254, May 2007.

[9] Kristensen, L. M., Jorgensen, J. B., and Jensen, K. Application of color petri nets in system development. In *Verlag B. H. ACPN 2003 and LNCS 3098*, pages 626–685, 2004.

[10] Laoutaris, N. Intrastream synchronization for continuous media stream: a survey of playout schedulers. *IEEE Network*, 16(3):30–40, May/June 2002.

[11] Liang, Y. J., Farber, N., and Girod, B. Adaptive playout scheduling loss concealment for voice communication over ip network. *IEEE Transaction on Multimedia*, 5(4):532–543, Dec. 2003.

[12] Liu, H. and Zerki, M. E. Towards the delay and synchronization control for networked real-time multiobjct multimedia applications. In *Proceedings of Ninth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems WORDS 2003*, pages 79–86, Oct. 2003.

[13] Manvi, S. and Venkataram, P. An agent based synchronization scheme for multimedia application. *Elesvier, The journal of systems and software*, 79:701–713, 2006.

[14] Marco Ajmone Marsan, S. D., Andrea Bobbio. Petri nets in performance analysis: An introduction. In *Petri Nets*, pages 211–256, 1996.

[15] Marsan, M. A. Stochastic petri net: an elementary introduction. *Lecture Notes in Computer Science*, 424:1–29, 1990.

[16] Owezarski, P. Enforcing multipoint multimedia synchronization in videoconferencing application. *Springer-Verlag B. H., IDMS 2000, LNCS 1905*, pages 65–76, 2000.

[17] Pal, K., Upadhyay, P. D., and Dutta, A. A petri nets based model for multipoint multistream synchronization in multimedia conferencing. In *The second international conference on security enriched urban computing and smart grid*, volume 223, pages 64–73, 2011.

[18] Pal, K., Upadhyay, P. D., and Dutta, A. A petri nets based model for multipoint synchronization in audio conferencing. In *Third International Conference on Communication Systems and Networks*, pages 65–76, Jan. 2011.

[19] Qiao, Z., Venkatasubramanian, R. K., Sun, L., and Ifeachor, E. C. A new buffer algorithm for speech quality improvement in voip system. *Springer science business media, LLC*, pages 189–206, 2007.

[20] Rautiainen, M., Aska, H., T.Ojala, Hosio, M., Makivirta, A., and Haatainan, N. Swarm synchronization for multi-recipent multimedia streaming. In *Multimedia and Expo, ICME , IEEE International Conference*, pages 786–789, June/July 2009.

[21] R.Tan and Guan, S. A dynamic petri net model for iterative and interactive distributed multimedia presentation. *IEEE Transaction on Multimedia*, 7(5), Oct. 2005.

[22] Segui, F. B., Cebollada, J. C. G., and Mauri, J. L. An rtp/ rtcp based approach for multimedia group and inter-stream synchronization. *Multimedia Tools and Applications*, 40(2):285–319, 2008.

[23] Xie, Y., Liu, C., Lee, M. J., and Saadawi, T. Adaptive multimedia synchronization in teleconferencing system. *Springer- Verlag, multimedia system*, 7:326–337, 1999.

[24] Xu, Y., Chang, Y., and Liu, Z. Calculation and analysis of compensation buffer size in multimedia system. *IEEE communications letters*, 5(8):355–357, Aug. 2001.

[25] Y. Ishibashi, T. K. and Tasaka, S. Inter-stream synchronization between haptic media and voice in collaborative virtual environ- ments. In *12th annual ACM international conference on Multimedia, New York, USA*, pages 604–611, Oct. 2004.

[26] Yen, H. Introduction to the petri net theory. In *Recent Advances in Formal Languages and Applications*, volume 25, pages 343–373. Springer, 2006.

[27] Zarros, P. N., Lee, J., and Saadswi, T. N. Interparticipent synchronization in real-time multimedia conferencing using feedback. *IEEE/ACM transaction on Networks*, 4(2):173–180, April 1996.