A Petri Net Based Model for Multipoint Multistream Synchronization in Multimedia Conferencing

Koninika Pal, Prajna Devi Upadhyay, and Animesh Datta

Dept. of Information Technology, National Institute of Technology, Durgapur, India
{koninikapal,kirtu26,animeshrec}@gmail.com

Abstract. Distributed multimedia application over IP network is an evolving paradigm for the researcher in the field of Information and Communication Technology (ICT). There are several applications of this technology —video conferencing, teleteaching and any collaborative application. Maintaining the satisfactory Quality of Service [1] and synchronization is a big challenge to the researchers in this field. In this paper, we extend our previous work [2] by ensur-ing multistream multipoint synchronization as an issue, involving the play out process of two streams (audio and video that were generated at the same time and bear temporal relationship) at different receivers at the same time and maintaining the temporal relationship between them to achieve fairness among the receivers. In this paper, we propose a distributed algorithm for achieving multi-stream multipoint synchronization and we also model the algorithm with Stochastic Petri Nets and verify its correctness.

Keywords: Multimedia, multipoint, petri net, synchronization.

1 Introduction

1.1 Media Synchronization

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. Continuous media, e.g. video and audio, have well defined relationships between subsequent Media Data Units (MDUs). In temporal synchronization we can distinguish between intra-stream synchronization, inter-stream synchronization and multipoint (or inter-destination) synchronization.

Intra Stream Synchronization. Intra-stream synchronization [3] refers to the temporal relationship between the MDUs of one time-dependent media stream i.e. between the MDUs of the same stream during their play out. Moreover, the play out process should be able to consume the MDUs with the same appropriate rate and sequence.

Inter Stream Synchronization. Inter-stream synchronization[3] refers to the synchronization, during the play out processes of different media streams (time-dependent or not) involved in the application.

R.-S. Chang, T.-h. Kim, and S.-L. Peng (Eds.): SUComS 2011, CCIS 223, pp. 64–73, 2011.

Multipoint Synchronization. In multicast communications, another type of synchronization, called group or multipoint synchronization[3,4], involving the synchronization of the play out processes of different streams in different receivers, at the same time, to achieve fairness among the receivers. We can cite the example of teleteaching applications, network quizzes. We should guarantee that the initial play out instant should be the same for all the receivers. Once the play out processes have started simultaneously in each receiver, not only the temporal relationships between MDUs of the same stream should be maintained by the intra stream synchronization for that media stream but also temporal relationship between multiple stream should be maintained by inter-stream synchronization. Nevertheless, due to the difference between end-to-end delays, resynchronization processes will be needed to maintain the receivers synchronized (multipoint synchronization). The maintenance of temporal relationships within a stream or among the multimedia streams usually depends on network delay, network jitter etc.

In this paper, we present a novel approach for multipoint multimedia synchronization in a distributed environment. There must be some application layer mechanism that relies on the QoS guaranteed by the underlying network layer to synchronize multiple receivers. This approach specifically deals with audio-video conferencing applications. The algorithm used to provide multipoint multimedia synchronization is described in section 4. Our approach uses global time provided by clock synchronization protocol NTP and delay calculation provided by RTCP feedback messages.

1.2 Petri Net

Multimedia distribution systems are very complex, so it is required to model them for effective implementation. A common tool used to model concurrent sys-tems is a petri net [5], characterized by concurrency, synchronization, and mutual exclusion, which are typical features of distributed environment.

To satisfy the requirement to model the system more specifically new features are added to basic petri net. It is difficult to capture the synchronization behavior and precedence constraints through any model meant to depict random behavior. A stochastic Petri net [6] is a useful tool for modeling and performance analysis of stochastic systems. Generalized Stochastic Petri Nets (GSPN) is allow to classify the transition of SPN in two way, timed and immediate and also introduced inhibitor input arc [7]. CSPN uses the set of color token instead of normal token in GSPN to obtain more precise specification.

A CSPN is a 7 tuple $(P, T, W^-, W^+, W^h, \Lambda, C)$ where

- $P = \{P_i \mid 1 \leq i \leq |P| = n\}$; finite set of places.
- $T = \{ T_j \mid 1 \leq j \leq |T| = n\}$; finite set of Transitions, where $P \cap T = \emptyset$ i.e., the set of places and transitions are disjoint. $T_I \cap T_T = \emptyset$; T_I is a set of immediate transition and T_T is a set of timed transition.
- W^-: $P \times T \rightarrow N$; Input connection function.
- W^+: $P \times T \rightarrow N$; Output connection function.
- W^h: $P \times T \rightarrow N$; Inhibitor arc.
- Λ: $\{\lambda_k \mid 1 \leq k \leq |T_T|\}$; firing rates of timed transitions.
- C: finite set of colors.

In this paper we propose a CSPN to model our algorithm described in section 5, verification of the model in section 6 and lastly cite an example using our model in section 7.

2 Related Works

Data transmission in IP is connectionless. A connectionless data transfer mode is lightweight as no connection establishment is necessary. IP has a powerful mechanism, called IP Multicast, for conducting multipoint-to-multipoint communications. The Multicast Backbone (MBONE) has offered an IP Multicast service on the Internet for low-bandwidth video and audio conferences since 1992 [8].

There are some serious synchronization issues attached with this IP network based audio video conferencing [9], they are inter stream synchronization, intra stream synchronization and group synchronization. Numerous algorithms are developed for achieving intra and inter stream synchronization in different scenarios [10], [11]. An algorithm synchronizes the initial play out time for all receivers, subsequently a solution for group synchronization control with continuous media has been proposed in [12]. [13] proposes a media synchronization algorithm by enhancing virtual time rendering (VTR). Using virtual time expansion and contraction of the target output time[14], the algorithm skips the Media Units. [4] presents the comparison of the most known multimedia group and inter-stream synchronization approaches. A table is presented summarizing the main characteristics of each analyzed algorithm according to those techniques and other critical issues.

There are many papers which model and verify the Distributed Algorithm using Petri Nets [15]. Hierarchical Protocol [16] describe how simple protocol is turned into a Hierarchical protocol and describe CPN Model of sender and receiver behavior. [6] introduces stochastic petri net extension. GSPN is discussed in [7] with performance analysis and an example.

3 Scope of Works

The algorithm for achieving multipoint synchronization in the earlier works synchronize the media streams at the receiver's end by skipping or pausing the play out data arbitrarily without considering where some relevant data is being lost. In our previous work [2], we look for the silent zones and make necessary adjustments only on the silent zone and prevent the possible loss of significant information that may have been loss because of arbitrary skipping of data stream during play out at the receiver's end. The limitation of this work [2] is that it deals with only audio signal and we could analyze the signal in a deterministic way.

Here in this work, we extended the previous work [2] by incorporating the video signal with the audio signal. We analyzed both the signals, stochastically, which is more realistic and propose an algorithm for achieving multi-stream (audio and video) multipoint synchronization, with the minimum data loss to retain the quality of service. Also model the algorithm using CSPN and prove its correctness.

4 Proposed Algorithm

4.1 Delay and Expected Play Out Time Calculation and Initial and Periodic Synchronization

We consider a multicast scenario. Sender chooses the maximum of all delays (within threshold value) received from the receivers to calculate expected play out time is referred in [2]. Multimedia conferencing takes place in a distributed environment so clock synchronization which is done initially and at a regular interval by using NTP service is provided by the network of server located at the Internet referred in [2].

4.2 Multimedia Specification

The following specification represents the relationship that may hold between two frames of different streams. Here MDU is represented as a frame whose size may or may not be same for two different media stream.

A precedes B ($A \rightarrow_P B$). This relationship holds when frame A finishes its playout before the start of the play out of frame B.
A overlaps B ($A \rightarrow_O B$). This relationship holds when A and B have the following temporal relationship.

A During B. This relationship holds when frame A and B both start play out at the same time or frame A starts play out after the starting of frame B but finishes before the end of play out of frame B or frame A starts play out after starting of frame B but both finish together.
A Meets B. Frame A starts its play out after the start of B but ends after the end of B.
B Meets A. Frame B starts its play out after the start of A but ends after the end of B.

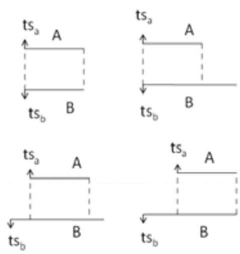
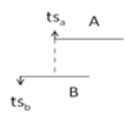
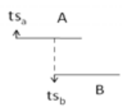
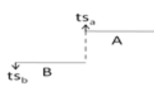
A succeeds B ($A \rightarrow_S B$). This relationship holds if frame A starts its play out after the end of play out of frame B.

Table 1 represents all possible temporal relationships between two frames with logical specification. In diagram line represents the time duration of frame denoted by d^{frame} and ts_{frame} represents the starting time of generation of frame.

4.3 Synchronous Play Out Algorithm

For multipoint synchronization we take the audio stream as master stream and synchronize the audio stream at multipoint. The video stream is synchronized with audio stream using the relationship that they hold during the generation time at sender side. For multipoint synchronization all receivers play signal at expected play out time mentioned by sender. If audio reaches earlier then it block until the expected play out time otherwise it starts playing. Now expected play out time for video is calculated according to the presentation time of corresponding audio. Before processing a video frame we find out the temporal relationship of corresponding audio frame using the logical specification defined earlier. In following algorithm we describe how synchronization is done.

Table 1. Formal Specification

Relation	Diagram	Logical Representation
$A \rightarrow_P B$ A precedes B		$ts_a < ts_b$ $\&\& \ ts_a \geq ts_a + d^a$
$A \rightarrow_O B$ A overlaps B		
Case I A during B		$ts_a \geq ts_b$ && $ts_a + d^a \leq ts_b + d^b$
Case II A meets B		$ts_a > ts_b$ && $ts_a + d^a > ts_b + d^b$
Case III B meets A		$ts_a < ts_b$ && $ts_a + d^a > ts_b$
$A \rightarrow_S B$ A succeeds B		$ts_a > ts_b \ \&\& \ ts_a \geq ts_b + d^b$

```
If (audio frame && video frame in the buffer) then
        If (audio precedes video) then
                process_audio_frame()
        elseif (audio succeeds   video)
                process_only_video_frame()
        else
                process_audio_frame()
                process_video_frame()
        end if
elseif (only audio frame in the buffer)
        process_audio_frame
else
        process_only_video_frame()
end if
```

Table 2. List of Variable

d^a, d^v	Duration of play out for audio frame and video frame respectively
ts_a,ts_v	Timestamp of audio frame and video frame respectively
R_a, R_v	Sequence no. of RTP packet for audio and video signal respectively
R_{ap}	Sequence no. of RTP packet which is most recently sent to play out buffer for audio
R_{ac}	Sequence no. of RTP packet that is currently processed for audio
S_{ap},R_{vp}	Sequence no. of frame which is most recently sent to play out buffer for audio and video respectively.
S_{ac},R_{vc}	Sequence no. of frame currently processed for audio and video
d_a, d_v	Delay for audio frame and video frame respectively
t_e	Expected playout time given in RTP audio packet
at_a, at_v	Actual arrival time in the receiving buffer for audio and video
t_p	Playout time of last processed frame for video
t_{pl}	Time at which frame is polled from buffer

Table 3 shows the functions that take the frames and calculate the delay that the frame should wait before presentation. Table 2 shows all the information used in three functions.

Table 3. Functions used to caculate presentation time

Process_audio_frame()	process_video_frame()	process_video_only_frame()

If ($R_{ac} = R_{ap}$) Then
If ($S_{ac} = S_{ap} + 1$) Then
$d_a \leftarrow d_a + d^a$
else
$d_a \leftarrow d_a+(S_{ac} - S_{ap})*d^a$
endif
else
$d_a \leftarrow t_e - t_{pl} + (S_{ac} - 1)*d^a$
endif
	$d_v \leftarrow d_a + (ts_v - ts_a) +$
	$(at_a - t_{pl})$
	If($d_v < 0$) Then
	$d^v \leftarrow d^v + d_v$,
	$d_v \leftarrow 0$,
	$t_p \leftarrow at_v + d^v$,
	endif
		If ($R_{vc} = R_{vp} + 1$) Then
		If (tp + dv > atv) Then
		dv ← tp + dv - tpl
		else
		$d^v \leftarrow d^v + (t_p + d^v - t_{pl})$,
		$d^v \leftarrow 0$
		endif
		else
		$dv \leftarrow t_p + (R_{vp} - R_{vc})* d^v - t_{pl}$
		endif

5 Petri Net Model for Synchronous Play Out

The CSPN for synchronous play out consists of 11 places and 15 transitions. There are two types of transitions, timed (t_1, t_2, t_{12}, t_{13}, t_{14}, t_{15}) and immediate transition (t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}). Immediate transitions fire immediately whenever a token is available in input place represented by bars, while timed transitions take some time to fire represented by rectangles. Each place contains a set of markers called color tokens. Each of these tokens carries a data value which belongs to a given type to be distinguished from each other. To be able to occur, a transition must have sufficient tokens on its input place and these places must have tokens that match the arc expressions. The places P_1 and P_2 represent the receiving buffer for audio and video and carry tokens of type l and m respectively. From the buffer, the tokens are moved to places P_3 and P_4. The transition T_3 is enabled when we have a token at place P_1 and no token at place P_2. At place P_6 audio is processed. The transition T_5 is enabled when we have a token at place P_2 and no token at place P_1 and token moved to place P_{11} for processing. If P_1 and P_2 both hold token then T_4 transits. T_6, T_7, T_8 occur according to the temporal relationship hold between audio and video frame. Place P_9 and P_{10} hold the audio and video token respectively for the time it has to wait in the buffer before play out. Video frame is processed at Place P_7 if it has a temporal relationship with audio. Audio and video frames are played out after the time associated with transitions t_{12} and t_{13} elapses.

Declaration of data type:

$l : (s, d_a , ts_a ,R_a ,S_a ,t_e , at_a, d^a)$; $m : (s, d_v , ts_v ,R_v ,S_v ,at_a, d^v)$
type float d_v , d^v; type int R_a ,S_a , R_v ,S_v; type time ts_a ,t_e ,at_a , ts_v, at_v ; type signal s

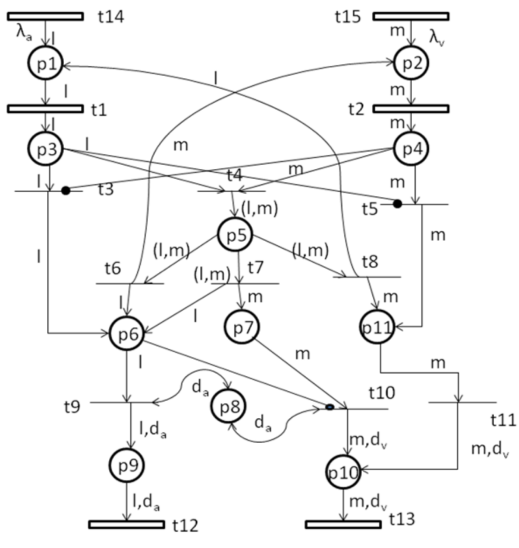


Fig. 1. CSPN Model

6 Verification of Model

Modeling power of a petri net has been examined by reachibility graph. It represents a part of transition system reachable from initial state. A state of a system represented by marking is a vector consists of positive number which represents the no. of token in all places in a net at that state. Reachibility diagram for our CSPN model is given below.

Now P and T represent place and transition respectively where P = (P_1, P_2… P_K) and T = (T_1, T_2…… T_m).

We define k × m incidence matrix [T] , $[T]_{(i, j)}$=Ø(T_j, P_i) - Ø (P_i, T_j) where,

Ø (T_j P_i) =no. of token added; Ø (P_i, T_j)=no. of token remove; $[T]_{(i, j)}$=Changed in place P_i when transition T_j fires once. Now if marking is reachable then the following equation is hold.

$$\mu_0 + [\ T\ ] . \#\sigma = \mu_F \tag{1}$$

Where μ_0 =Initial marking, μ_F =Final marking, #σ =m-dimensional vector with its jth entry denoting the no. of time transition T_j occurs in σ.

If initial marking μ0 is $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$; Transition T14, T1, T3, T9, T12 occur then #σ = $[\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0]^T$. Putting the value of [T], μ_0, #σ in equation (1) we get marking $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$ which is also reachable through the reachibility graph. So we can establish the correctness of reachibility graph for our petri net.

We represent verification of only one sequence of transition here. Other sequences of transition can be verified in similar way.

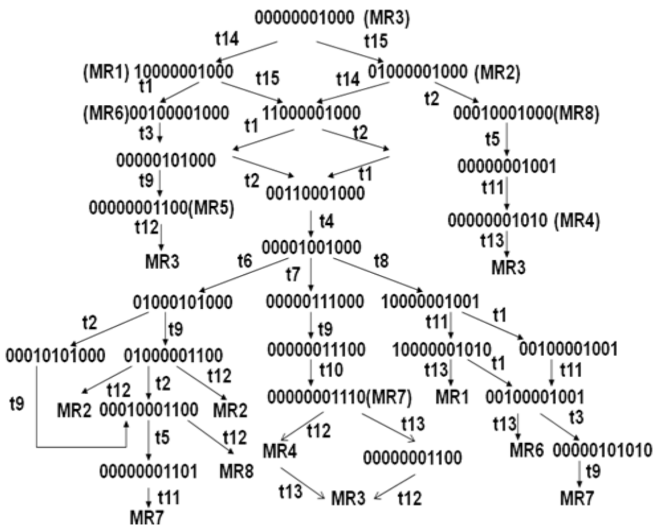


Fig. 2. Reachibility Graph

7 Discussion with an Example

The example in fig 3 shows how our algorithm works for different cases of asynchrony. It consists of audio and video stream at two receivers. Each stream is built with fixed length unit called frame. Here two audio frames of 2 time unit duration come together in one RTP packet and one video frame with 3 time unit duration come in one RTP packet. We recognize an audio frame with (sequence no. of RTP packet, sequence no. of frame within RTP packet), e.g., (1, 2), and video frame with sequence no. of RTP packet. We poll both buffers in receiver at unit time interval. Fig. 3 shows the original signal before sending, after receiving. After receiving the signal at rcv1 and rcv2 we apply our algorithm to synchronize the signal before playing out at each receiver. The playing out signal is also shown in Fig. 3.

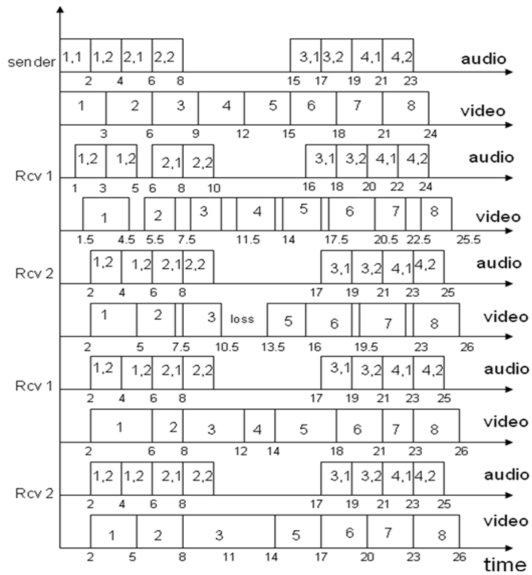


Fig. 3. Original, Received and Adjusted Signal

8 Conclusion

In this work we addressed a media synchronization issue called multimedia multipoint or inter destination synchronization which is required for designing any collaborative environment on distributed IP network. We suggested an algorithm for achieving this synchronization issue and modeled it using CSPN model and verified its correctness. The analysis of the CSPN model is not included in the paper. The future prospect of the work would be a detailed analysis of the CSPN model and implementation of the algorithm.

References

1. Quality of Service for Conferencing Reality or Hype? A guide to understanding and successfully deploying the appropriate level of QoS for conferencing. WainHouse Research (2003)
2. Pal, K., Upadhyay, P.D., Dutta, A.: A Petri Nets Based Model for Multipoint Synchronization in Audio Conferencing. In: Third International Conference on Communication Systems and Networks (COMSNETS), January 4-8 (2011)

3. Boronat, F., Lloret, J., Garcıa, M.: Multimedia group and inter-stream synchronization techniques: A comparative study. Information System 34, 108–131 (2009)
4. Cebollada, J.C.G., Mauri, L., Seguí, F.B.: Multimedia Group Synchronization Algorithm Based on RTP/RTCP
5. Yen, H.-C.: Introduction to Petri Net Theory
6. Bobbio, A.: System Modeling with Petri Nets
7. Marsan, M.A., Balbo, G., Conte, G.: A Class of Generalized Stochastic Petri Nets for the Performance Analysis of Multiprocessor Systems. ACM Transactions on Computer Systems 2(1), 93–122 (1984)
8. Casner, S.: Are you on the MBONE? IEEE Multimedia 1(2) (1994)
9. Weinstein, I.M.: The Universal IP Network for Video Conferencing A Manager‘s Guide to the Deployment of Inter-Enterprise Videoconferencing
10. Guerri, J.C.: Specification and performance evaluation of an adaptive multimedia stream Synchronization protocol based on global time and feedback techniques, with stream control Thesis. In: U.P.V. (June 1997)
11. Guerri, J.C., Esteve, M., Palau, C.E., Casares, V.: Feedback flow control with hysteresial techniques for multimedia retrievals. Multimedia Tools and Applications 13(3), 307–332 (2001)
12. Rangan, P.V., Ramanathan, S.: Performance of inter-media synchronization in distributed and heterogeneous multimedia systems. Computer Network ISDN System 27(4), 549–565 (1995)
13. Ishibashi, Y., Tasaka, S.: A synchronization mechanism for continuous media in multimedia communications. In: Proc. IEEE INFOCOM, pp. 1010–1019 (1995)
14. Tasaka, S., Ishibashi, Y.: Media synchronization in heterogeneous networks: stored media case. IEICE Trans. Commun. E81-B(8), 1624–1636 (1998)
15. Kindler, E., Reisig, W., ViSlzer, H., Humboldt, R.W.: Petri Net Based Verification of Distributed Algorithms: An Example. In: Universit zu Berlin, Institut ffdr Informatik, Berlin, Germany Fraunh
16. Jensen, K.: Hierarchical protocol. Aarhus University, Denmark, kjensen@daimi.au.dk