Journal of Korea Multimedia Society Vol. 7, No. 2, February 2004(pp. 194-203)

A Smooth Playing Mechanism for Multimedia Synchronization in Mobile Environment

Keun-Wang Lee[†], Gi-Sung Lee[††]

ABSTRACT

In this paper we propose a Mobile Petri-net Model (MPM) as a new specification model including the QoS parameter that minimizes the transmission delay time. In MPM, we propose a playing time algorithm for supporting the seamless presentation and the smooth playing. The proposed model has a higher guarantee of QoS such as playing time than the previous work.

모바일 환경에서 멀티미디어 동기화를 위한 유연한 재생 기법

이근왕[†], 이기성[††]

요 약

본 논문에서는 전송 지연 시간을 최소화하기 위해 서비스 품질 파라미터를 포함한 새로운 규격모델로 모바일 페트리넷 모델을 제안한다. 그리고 미디어의 재생시 끊어짐이 없는 유연한 재생을 처리하기 위한 재생 시간 알고리즘을 제안한다. 제안한 모델은 기존 연구보다 재생시간과 같은 서비스 품질의 보장을 향상시키는 결과를 나타내었다.

Key words: Multimedia Synchronization(멀티미디어 동기화), Quality of Service(서비스 품질)

1. Introduction

The interest in multimedia services coupled with a growing high-bandwidth structure has increased in the area of wired communications. Recently, with the advancement in wireless communication systems and portable computing technologies, demand for Mobile Multimedia Services (MMSs) is increasing in the area of wireless networks, such as data transmission, video transmission, WWW access[1-3]. MMSs are a complex concept of wireless extensions of wired multimedia services and mobile specific multimedia services. That is, MMSs are to add user mobility to multimedia services in wired networks[4].

The synchronization in the networks has been continuously studied, but it is scarce of study in mobile networks. In mobile networks, a Mobile Host (MH) has small memory, and a Base Station (BS) supports more the limited resources as compared with the sites in the networks. Moreover, a mobile communication system has low power, and must provide high-quality access with the MH. Consequently, the synchronization in mobile

※ 교신저자(Corresponding Author) : 이근왕, 주소 : 충남 홍성군 홍성읍 남장리 산29(350-701), 전화 : 041)630-3291, FAX : 041)634-8700, E-mail : kwlee@cwunet.ac.kr
접수일 : 2003년 7월 14일, 완료일 : 2003년 9월 24일
[†] 종신회원, 청운대학교 멀티미디어학과 조교수
[††] 정회원, 호원대학교 컴퓨터학부 조교수
(E-mail : ygslee@sunny.howon.ac.kr)
※ 본 연구는 2003년도 청운대학교 학술연구조성비 지원에 의하여 연구 되었음.

networks is more complex than that in wired networks. In particular, because of the small memory and the limited resources, the MH may suffer from either the buffer underflow or the buffer overflow. Therefore, an adaptive synchronization method is needed for ME.

The previous works have been done in describing the synchronization model for multimedia applications[5-10]. Among those, Petri-net based specification models provide a good method to specify temporal relationships. The integration of various media and the definition of QoS requirements should be supported and easily described. However, previous extended Petri-net based model like the Object Composition Petri-net (OCPN) and the Real Time Synchronization Model (RTSM) have some constraints in modeling the QoS parameters of both intramedia and intermedia.

In this paper we propose a Mobile Petri-net Model (MPM) as a new specification model including the QoS parameters that minimizes the transmission delay time. The proposed mechanism is composed of a two-step synchronization. First, we propose a model which satisfies the demand conditions for the synchronization between media streams during their transmission and presentation. All real-time applications have the limited condition such as the delay time between media data and QoS for synchronization between media data. Second, a playing time algorithm is considered for supporting the seamless presentation, which does not keep closely the periodic playing time and smooths the media streams that is sensitive to time.

2. Related Work

At the time of creation of multimedia informations, a user needs a model to specify temporal constraints among various data which must be observed at the playing time[3]. Such specification models are the OCPN and RTSM which are a timed Petri-net.

OCPN model has been proved to have the ability to specify any arbitrary temporal relationship among media and it may be applied to both stored data applications and live applications[3,5]. The OCPN model weakly describes temporal relationships with synchronization between media at an object level. Multimedia application that are comprised of video, audio and text object can be expressed by the OCPN technique that is shown in Fig. 1. Generally, audio data is very jitter sensitive and cannot tolerate random delays between frame segments; hence, delay between audio frames may result in unrecognizable audio quality. In contrast, some extent of random delay between video frames may be acceptable in video data.

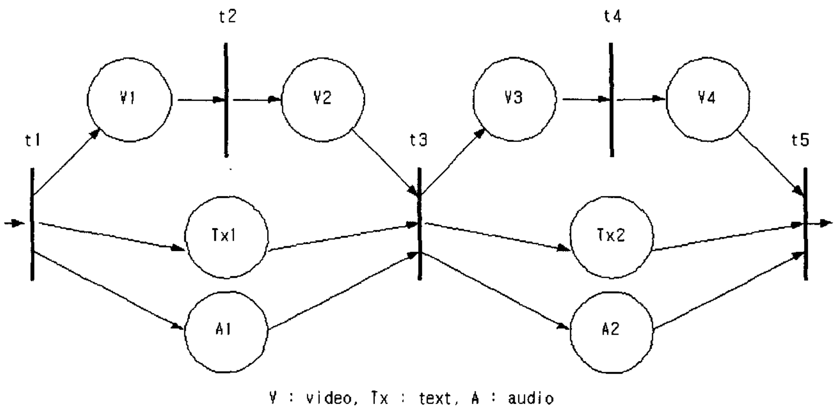


Fig. 1. An example of OCPN model

Therefore a late transmission of video frames does not influence the QoS.

When taking the real-time multimedia and the random delay of packet/cell networks into consideration, the OCPN and other Petri-net based models are not sufficient to deal with the late transmission of packets. Here, we define the late transmission of a packet to be that packet that fails to reach its destination in time and should be dropped. We explain the insufficiency of the OCPN model that is shown in Fig. 1. A delay of video objects in the OCPN model will give an impact on playing out an audio object. Such a delay of audio objects in a real-time application will cause a significant degradation of its QoS.

RTSM is introduced to guarantee the QoS of real-time applications on the delay of intermedia. A jitter-sensitive audio object is defined as a key medium and represented as enforced place[6]. However it is possible that a real-time constraint is exceeded due to the delay of key medium itself. If the text object is one of the key media in Fig. 1, it is assigned an enforced place just as an audio object. Once one of the enforced places is unblocked, each transition of the RTSM is fired. When an audio object is unblocked, the transition is fired, even if the text object is still activated. Therefore, the text object is not guaranteed to display its content. When the transition is fired in the RTSM model, the remaining part of the other medium is dropped. That is, when taking a random delay into account, it is not sufficient to deal with the late transmission of packets, and it is also insufficient in modeling the QoS parameters.

3. Mobile Petri-net Model (MPM)

The explosive growth of Internet access in parallel with technological advances has motivated mobile computing and multimedia applications in mobile networks. We assume that Multimedia Servers (MSs) on the Internet act as repositories for multimedia applications orchestrated according to the synchronization model such as OCPN model. A MS searches the requested data from its databases and transmits it to the BS. Fig. 2 shows the mobile system structure.

Multimedia data consists of many multimedia objects: a unit of synchronization, which are stamped with the current local time to allow a BS to calculate a round trip delay, jitter and inter-arrival time, during which the MS transmits multimedia objects. The multimedia objects need to be buffered at the BS for carrying out synchronization as interface between the Internet and mobile networks. ME consists of three parts: MSs, BSs and MHs. In addition to the conventional functions of the BS such as frequency resource management and error control, the BSs in the mobile multimedia network have the additional function such as synchronization, rate matching, etc. The MH must access MSs over the Internet through the BS.

As shown in Fig. 2, multimedia data is transmitted from the MS to the BS. After buffering the multimedia data in order to smooth inter-packet jitter delays that occurs on Internet, the BS then transmits the synchronized data to the MH.

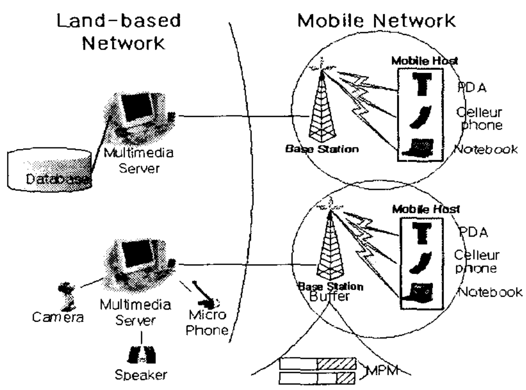


Fig. 2. Mobile Multimedia System Structure

3.1 Definition of MPM

In the MPM model, the definition to specify Petri-net model in any BS is the following thing.

The MPM model specified by the tuple [*P, T, K, A, DIFF, R, D, J, Re, M*]
where

$P = p_1, p_2, p_n$; *Regular places (single circles),*
$T = t_1, t_2, , t_m$; *Transitions,*
$K = k_1, k_2 ,, k_i$; *Key place,*
$A : (X \times T) \cup (T \times X) \rightarrow I, I=1, 2, 3, \ldots$; *Directed arcs.*
$X = P \cup K$; *All places.*
$DIFF : X \rightarrow Realnumber$; *Difference between maximum duration time and actual duration time*
$R : X \rightarrow Realnumber$; *Relative duration time*
$D : X \rightarrow Realnumber$; *Maximum duration time of places.*
$J : X \rightarrow Realnumber$; *Maximum delay jitter*
$Re : X \rightarrow r_1, r_2, \ldots, r_k$; *Type of media.*
$M : X \rightarrow I, I = 0, 1, 2$; *State of places.*
Each place may be in one of the following states:
0: no token
1: token is blocked Cross in the place.
2: token is unblocked Dot in the place

Places are used to represent a medium unit and their corresponding actions. The place can have tokens. A place without token means that it is currently inactive. A place with a token is active and could be in one of the two states; blocked or unblocked. When a transition is fired and then a token is added into a place, the action of the place is executed, and the token is blocked before finishing the action. The token is unblocked after finishing. Each place has some parameter that determines its relative importance compared with other medium. Two factors, the importance of the medium and the jitter-sensitivity of the medium, should be considered in deciding the time medium with the absolute time and the multiple key media parameter.

3.2 Firing Rule of MPM

Firing rules of the MPM model are described as follows: 1) Firing occurs immediately if the end of a time medium is done. 2) A transition t_i fires immediately when the arrived key medium places contain an unblocked token. 3) Upon firing, a set of backtracking rules is exercised to remove tokens from their input places. 4) the transition t_i removes a token from each of its input places and adds a token to each of its output places. 5) After receiving a token, the place p_j remains in the active state for the interval specified by the duration j. During this interval, the token is blocked.

The MPM model can easily represent the synchronization procedure between streams and events, and determine the playing conditions by its precedence relations by correcting the problems of OCPN model and RTSM model. The simulation procedure for a MPM playing algorithm is summarized as follows. It starts from the early marking ($M(p_j) = 1, \forall j : A(p_j, t_i) > 0$). That is, all active transitions are fired and create a new marking.

Algorithm Playing
Playing( ) {
$M(p_j) = 1, \forall j : A(p_j, t_i) > 0$
key medium decision {
if (T end) or (∀key medium playing) then
{
if $M(p_j) < 1, \forall j : A(p_j, t_i) > 0$ then
t_j fire, $t_j : A(t_j, p_j) > 0$ }
$M(p_j) = M(p_j) - 1, \forall j : A(p_j, t_i) > 0$
$M(p_k) = M(p_k) + 1, \forall k : A(t_i, p_k) > 0$ }
}

As described in firing rule (3), if the transition t_i is fired before removing a token from its input place, the backtracking is performed for removing forcibly the remained tokens. That is, the backtracking is executed for the places without token in the input place of the corresponding transition t_i.

3.3 Concept of Multiple Key Media and Time Medium

In MPM, there are two kinds of places as regular places and key places. To differentiate a key place from regular places, a double circle is drawn for key place as shown in Fig. 3 where the transition t_4 is fed by two regular places, V_3 and Tx_1 and one key place, Au_1.

The firing rule about key places is that once multiple key places get unblocked, the transition following it will be immediately fired regardless of the states of other places feeding this transition. Therefore, if Au_1 becomes unblocked in Fig. 3, the transition t_4 will be immediately fired regardless of the states of V_3 and Tx_1. At the same time, tokens in the places before the transition t_4, such as V_1, V_2, V_3 or Tx_1, must be cleared since these places are obsolete due to the firing of t_4. We call this action of clearing all obsolete tokens backtracking, since the action is backtracked from the fired transition in MPM. In this way, the synchronization anomaly due to late transmission is solved.

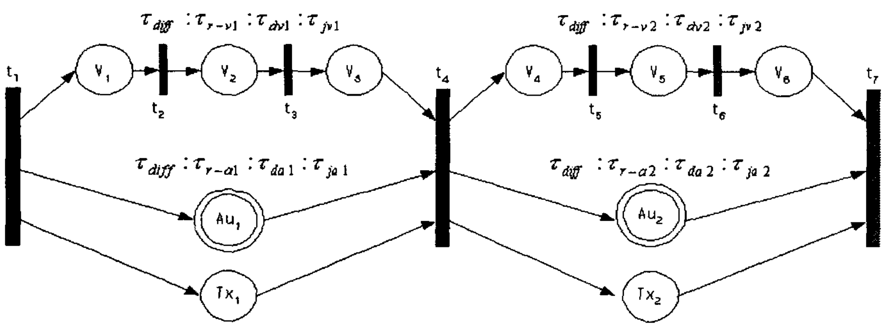
Time medium is a virtual medium and normally contains a deterministic duration of time which specifies the real-time constraint between two transitions. Since all media are transmitted across the network, it is also possible for the multiple key media to suffer a long delay such that its real-time constraint is exceeded. Fig. 3 is still waiting for its packets due to late transmission while its real-time constraint has expired. In such a case, it is reasonable to fire the transition t_4 to activate Au_2, V_4 and Tx_2 instead of waiting for Au_1 in order to maintain the quality of audio.

4. MPM Including the QoS Parameter

Each place p_i has time-related four parameters τ_{diff}, τ_r, τ_d, and τ_j as well as key medium in order to handle QoS parameters for intermedia synchronization. τ_{ma} is arrival time of media. τ_b is maximum delay time. τ_r represents a relative duration time to playout an arrived media. P_{ma} *is* maximum size for media to playout during a duration time. P_{ms} is arrived size during a smoothing buffer delay. P_r represents a playout rate of P_{ma}/P_{ms}. τ_d is the duration time for which a playout or a display of a media. τ_{r-a} is relative duration time of audio. τ_{diff} is the time difference between a duration time and a relative duration time. τ_{r-m} is relative duration time of other media except audio. τ_j is the acceptable maximum delay jitter time between each medium. τ_{r-mp} is playout time of media.

Algorithm that decides the relative duration time is the followings.

RalativeTime( ) {
    if τ_{ma} <= τ_b then
        τ_r := τ_b;
    else {
            $P_{ma} = P_{ms} \times \tau_b \,/\, \tau_{ma}$;
            $P_r = P_{ma} \,/\, P_{ms}$;



V : Video object   Au : Audio object   Tx : Text object

Fig. 3. MPM with the QoS Parameter

```
            τ_r = P_r × τ_d ;
        }
}
```

Jitter-compensatory time algorithm is the following.

```
JitterTime( ) {
  τ_{r-a} := relative_duration_time(Au);
  τ_diff :=  τ_d -  τ_{r-a};
  while media then {
   τ_{r-m} := relative_duration_time(m);
  if  τ_diff <=  τ_j then {
         wait ( τ_diff ) ;
          τ_{r-mp} := τ_{r-a} +  τ_diff; }
  else {
         wait (  τ_j ) ;
          τ_{r-mp} := τ_{r-a} +  τ_j; }
  }
}
```

Fig. 3 presents the MPM with QoS parameter. Inter-media skew value 0 means that the two media streaming is completely synchronized. For instance, it can be considered as the synchronized state if it is within 80ms between audio and video streams. If skew value 80ms is applied to key medium of audio, the maximum delay jitter τ_j will be destroyed, which will result in degrading the audio quality. Because the maximum delay jitter of the audio within media is 10ms, waiting longer than 10ms could be a factor that destroys the audio quality. Therefore, τ_j is compensated by means of applying the maximum delay jitter τ_j.

Maximum delay jitter τ_j represents the maximum delay jitter of the place p_i. The delay is an interval between medium i and medium $i+1$, and it is used to obtain the better quality of service by additionally playing out as much as the time of the maximum delay jitter. Audio medium τ_{diff} represents a time difference between maximum duration time and relative duration time. Video medium τ_{diff} is a result of a jitter-compensatory time algorithm and it is obtained from comparing τ_{diff} with τ_{ja1} of audio medium. Applying maximum jitter delay not only improves QoS but also has an effect on compensating a time of playing out a place except key medium.

Transition t_4 is fired only by A_1 with key medium among input places in Fig. 3. Let's assume that it takes 135ms for audio object and video object to reach the destination transit. A relative duration time of audio object will be 115ms according to a relative duration time algorithm and τ_{diff} will be 10ms by a jitter-compensatory time algorithm, which means that audio object will wait as much as 10ms because it does not exceed the maximum delay jitter of 10ms. Video object τ_{diff} will take a smaller one between τ_{diff} and τ_{ja1} of audio object. The relative duration time of video object will be 115ms by the relative duration time algorithm.

10ms from an audio object is added to 115ms of relative duration time of video object. Video object can be compensated in cases that the total time added with τ_{diff} of video object does not exceed 125ms. That is, video object can be compensated as much as 10ms because it does not exceed 125ms. If jitter is not considered, the delay of audio and video object gives a bad impact on playing a next frame, while a proposed jitter-considerable scheme results in playing a next frame efficiently. Symbols used in a playing policy is the following. P_n is the number of media frames to be played out during the maximum duration time. τ_{m-f} is the frame time to be played out from each media. M_c is the number of frame arrived in the current media. τ_p is playing time of a video frame. τ_{p-t} is smoothing playing time of a media. A smoothing buffer 125ms is used at the BS to avoid jitter delay in applications. The delay of a smoothing buffer can be acceptable to video conference, tele-teaching and real-time applications. The playing time algorithm is the following.

Algorithm Playing Time
PlayingTime( ) {
While *media* then
$\tau_{m-f} := \tau_d \ / \ p_n$;
$M_c := \tau_{r-m} \ / \ \tau_{m-f}$;
$M_c := INT \ (M_c)$;
$\tau_p := \tau_{r-mp} \ / \ M_c$;
$\tau_{p-t} := \tau_p \ X \ M_c$;
}

The display of the frames for duration time is defined as playing control. Note that a frame is not always playing out within the exact time interval whenever a frame buffer is available. The previous playing control skips simply the frames that do not arrive within the maximum delay time of a smoothing buffer. However, if the frames are skipped frequently, some scene is unnatural, so user will prefer to reduce the number of the omitted frames for a smooth movie rather than to increase the number of the omitted frame for the stick playing.

In case that the playing time loss is incurred because the network states and the alternation of average delay for duration time, we try to compensate for the video loss by changing the playing time of video. In case that audio and video contains the playing data of 90ms and 125ms, respectively, only video frame100ms is presented because audio is defined as key media. This means that the loss of the playing time of 25ms video frames occurs. Playing time 90ms of audio divided by the number of frames gives 30ms. However, if it is compensated for the maximum delay jitter, and then divided into the number of frames, the playing time becomes 33ms.

5. Performance Evaluation

Media format used in the simulation is in the following. The 1Kbytes audio data is encoded by the PCM encoding scheme, and resolution of video frames is 120×120. Note that the number of video places corresponding to one audio place is random due to the following reason. At the BS, the application gets one audio packet from the audio device every 125ms, and it gets one, two or third video frames from the video frame grabber, which is determined by the run-time processing overhead of the operating system.

In the paper, the simulation environment is set to the MH in ME. The sample data to be used in the simulation are generated by the Poisson process and the network jitter delay is equally applied to all media. The number of media is limited to two types for comparing the performance. Notice that the delay value generated by the normal distribution is forced to be within the range (that is, 0.5*mean to 4*mean) in the program to make the delay values more realistic. The number of transitions is set to 200 and the maximum delay jitter is 10ms. On the purpose to verify the algorithm, the OCPN and RTSM are also evaluated in the same simulation environment. The playing rate of the proposed delay jitter scheme based on the maximum delay jitter time is compared to the existing methods. The efficiency of the proposed scheme is observed by being comparing with the playing rate of the different models, in case of the normal and abnormal arrival of audio streams to the BS, respectively. It is assumed that the average delay is 100ms and the variance is 20ms in case of audio streams with delay. Fig. 4 shows the result in case of the normal arrival of audio object and the late arrival of video object. We can see that the proposed MPM model presents smoothly video object better than the RTSM and OCPN.

In Fig. 5, the comparison of the playing time of the three models is plotted against the translation number in case of the late arrival of an audio object and the normal arrival of video object. That is, when an audio is delayed, it shows the improved result than other models by adopting the proposed relative duration time algorithm.

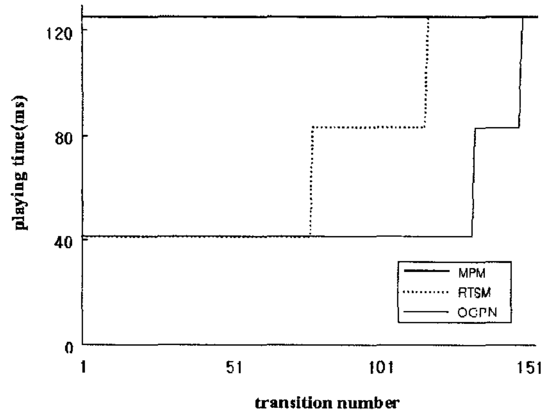


Fig. 4. Simulation result in case of abnormal video

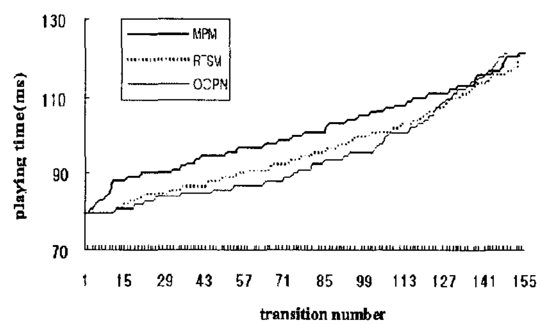


Fig. 5. Simulation result of abnormal audio

In Fig. 6, the comparison of the playing time of the three models is plotted against the translation number in case of the late arrival of two media. That is, when an audio and a video object are delayed, it represents the duration time of the OCPN and RTSM model, and shows the duration

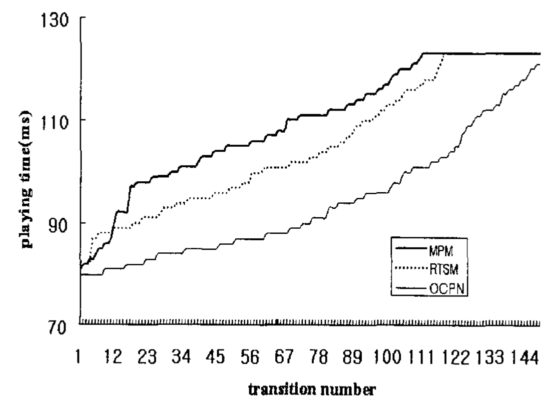


Fig. 6. Simulation result of two abnormal media

time of MPM model is improved than that of other models by adopting the relative duration time for two media.

Fig. 7 shows the comparison result of the playing time of the RTSM and MPM model; the RTSM model without applying jitter, the proposed models with applying delay jitter and smooth the duration time. As a simulation result, we can see the playing time of the proposed model is shorter than the RTSM model.

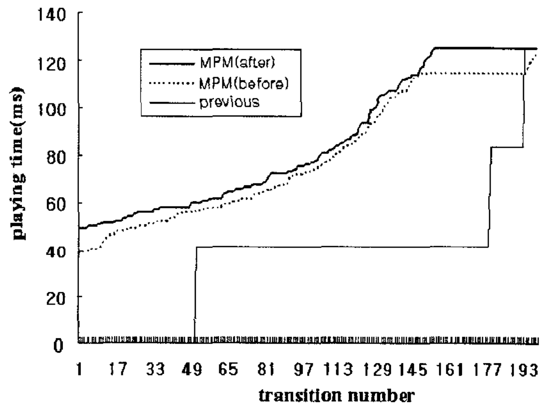


Fig. 7. Comparison result of playing time

6. Conclusions

In this paper we have proposed a new specification model to describe the QoS parameter and the delay time between media data in ME. The proposed MPM has a distinct property that allows QoS parameter. This is very important since most of applications represent without QoS in ME. In summary, the contributions of this paper are listed in the following.

1) The MPM to describe the QoS parameter for inter-media synchronization in ME is proposed. 2) The concept of multiple key media in the MPM for synchronization control is suggested. 3) The concept of time medium associated with multiple key media to deal with late transmissions of multiple key media is suggested. 4) The slave media stream is presented smoothly by using a playing time

algorithm for supporting the seamless presentation and the smooth playing.

The proposed model supplies flexible description of the temporal relationships among various media. This model can provide modeling specifications from the live applications to the pre-orchestrated applications in ME. The MPM model adaptively manages the waiting time of smoothing buffer, which leads to minimize the gap from the variation of delay time and it results in naturally playing to MH. Also it is suitable to the system which requires the high guarantee of QoS, and improves QoS such as decrease of loss rate and increase of playing rate.

In future work, Many issues need to be a specified model which includes time constrained by handoff procedures to satisfy real-time requirement of multimedia data and the modeling of multimedia, which are expected to provide new impact on the MH.

References

[1] D. H. Nam and S. K. Park, Adaptive Multimedia Stream Presentation in Mobile Computing Environment, Proc. of IEEE TENCON, PP. 966-969, 1999.

[2] A. Boukerche, S. Hong and T. Jacob, MoSync: A Synchronization Scheme for Cellular Wireless and Mobile Multimedia Systems, Proc. of the Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems IEEE, PP. 89-96, 2001.

[3] M. Woo, N. Prabhu and A. Grafoor, Dynamic Resource Allocation for Multimedia Services in Mobile Communication Environments, IEEE J. Selected Areas in Comm., Vol.13, No.5, PP. 913-922, 1995.

[4] D. H. Nam, S. K. Park, A Smooth Playback Mechanism of Media Streams in Mobile Computing Environment, Proc. of ITC-CSCC'98, 1998.

[5] P. W. Jardetzky, and C. J. Sreenan, and R. M. Needham, Storage and Synchronization for Distributed Continuous Media, Multimedia Data Base Systems, Springer-Verlag, 1995.

[6] C. C. Yang and J. H. Huang, A Multimedia Synchronization Model and Its Implementation in Transport Protocols, IEEE J. selected Areas in Communications, Vol.14, No.1, Jan., 1996.

[7] R. Steinmetz, Human Perception of Jitter and Media Synchronization, IEEE J. Selected Areas in Communications, Vol.14, No.1, Jan., 1996.

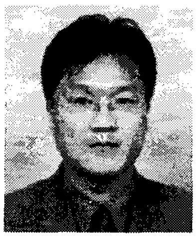
[8] D. L. Stone and K. Jeffay, An Empitical Study of Delay Jitter Management Policies, Multimedia Systems/Springer-Verlag, 1995.

[9] G. Blakowski and R. Steinmetz, A Media Synchronization Survey: Reference Model, Specification, and Case Studies, IEEE J. Selected Areas in Communications, Vol.14, No.1, Jan., 1996.

[10] M. J. Perez-Luque and T. D. C. Little, A Temporal Reference Framework for Multimedia Synchronization, IEEE J. Selected Areas in Communications, Vol.14, No.1, Jan., 1996.

이 근 왕

1993년 한밭대학교 전자계산학과 공학사
1996년 숭실대학교 컴퓨터학과 공학석사
2000년 숭실대학교 컴퓨터학과 공학박사
2000~2001. 숭실대학교 BK21 Post-Doc
2001년~현재 청운대학교 멀티미디어학과 조교수
관심분야 : 멀티미디어 통신, 멀티미디어 응용, 모바일 통신



이 기 성

1993년 숭실대학교 컴퓨터학과 공학사
1996년 숭실대학교 컴퓨터학과 공학석사
2001년 숭실대학교 컴퓨터학과 공학박사
2001년~현재 호원대학교 컴퓨터학부 조교수
관심분야 : 멀티미디어 통신, 멀티미디어 보안, 모바일 통신