적응형 재생제어를 이용한 동기화된 일대다 미디어 스트리밍

정회원 조 진 용*, 김 종 원**

Synchronized One-to-many Media Streaming employing Server-Client Coordinated Adaptive Playout Control

Jin-Yong Jo* , JongWon Kim** *Regular Members*

요 약

본 논문에서는 1:N 멀티캐스트 미디어 스트리밍을 위한 적응형 재생제어 기법을 제안한다. 제안된 적응형 재생제어에서는 음성의 시간규모 변환(time-scale modification)을 통해 음성과 영상이 함께 있는 미디어의 재생속도를 조절한다. 수신자의 동기화 상태 및 버퍼 점유율에 기초해 재생속도가 조절되며 재생 품질에 영향을 미치지 않는 범위 내에서 속도 변화가 가해지게 된다. 이를 통하여 시스템의 불안정성 및 네트워크 혼잡에 의해 발생할 수 있는 미디어 재생의 끊김 현상을 최소화하고 멀티미디어 품질을 극대화시킬 수 있다. 또한 적응형 재생제어 기법은 재전송에 의한 손실 복구 시 복구를 위한 가용 시간을 보상해 줄 수 있다. 네트워크 시뮬레이터에 기초한 모의실험을 통해 제안된 멀티캐스트 스트리밍 기법이 재생 시 발생하는 끊김 현상을 줄이고 그룹 참가자들 간의 이질성을 완화시킴을 확인한다.

Key Words: Multicast media streaming; inter-client synchronization; source specific multicast; adaptive playout control.

ABSTRACT

A new inter-client synchronization framework for multicast media streaming is proposed employing a server-client coordinated adaptive playout control. The proposed adaptive player controls the playback speed of audio and video by adopting the time-scale modification of audio. Based on the overall synchronization status as well as the buffer occupancy level, the playout speed of each client is manipulated within a perceptually tolerable range. Additionally, the server implicitly helps increasing the time available for retransmission while the clients perform an interactive error recovery mechanism with the assistance of playout control. The network-simulator based simulations show that the proposed framework can reduce the playout discontinuity without degrading the media quality, and thus mitigate the client heterogeneity.

I. Introduction

Realizing an effective one-to-many media streaming over the IP multicast faces lots of challenges. It requires feasible signaling/transport protocols, stable multicast-enabled networks, and network-adaptive applications to distribute the broadband media in an acceptable quality[1]. For the network side, source specific multicast (SSM) model[2] alleviates several complications (e.g., by

* 광주과학기술원 정보통신공학과 Networked Media 연구실, 2003년 1월부터 KT 안양지사 근무 (jiny92@kt.co.kr),
** 광주과학기술원 정보통신공학과 Networked Media 연구실 (jongwon@kjist.ac.kr)
논문번호 : 020305-0718, 접수일자 : 2002년 7월 22일
※본 연구는 정보통신부 정보통신연구진흥원에서 지원하고 있는 정보통신기초연구지원사업의 연구결과입니다.

restricting application scenario) of conventional any source multicast (ASM), especially for one-to-many media streaming. For the protocol side, real-time transport protocol pair (RTP/RTCP[3]) can provide interoperable/monitored real-time transport channel over the IP network. RTP/RTCP in itself does not guarantee quality of service (QoS) to streaming media applications but acts as a helper. Streaming media applications at the server and clients are responsible to adaptively take charge of network congestion/error and system resource limitation.

To cope with the above challenges and provide high-quality media streaming, an adaptive multicast streaming framework has to be established. To overcome the network and client variation, it is important to deploy multicast rate (e.g., congestion and flow) and error control. It also needs to address the playback synchronization issue in the intra-/inter-client aspect while properly controlling the network buffers. Multicast rate and error control issues are addressed quite often. However, multicast playback synchronization has been rarely discussed. Thus, in order to reduce the playback discontinuity and mitigate the heterogeneity, we propose to establish a *synchronized multicast streaming framework*, where the synchronized playout of all clients is adaptively managed. Note that synchronization issue becomes more important as the media streaming goes broadband and high-quality.

In the proposed synchronized streaming framework, each client is required to keep synchronization coping with the exceptional system events as well as the network fluctuations. To assist each client for this challenge, we propose to adaptively control the playback speed of playout. By extending the audio/speech adaptive playout with time-scale modification in [4],[5] to audio/video, the playback speed of client can be varied. That is, based on the combined buffer (i.e., network and application buffers) occupancy level, the player at the client is adaptively *expanding* or *contracting* the playout within the range that it does not hurt the viewers perceptually. Thus, we can proactively conduct the adaptive playback not only to reduce the playback discontinuity but also to guarantee high-quality playback with flexible error controls.

The proposed streaming framework consists of 1) local playback adaptation (guided by a playback factor) based on the combined buffer occupancy with error recovery support, 2) unicast RTCP feedback on the presentation point as well as the channel status, and 3) inter-client synchronization with the synchronization aid from the server. More specifically, each client locally controls the playback speed to prevent buffer overflow/underflow (subsequently to prevent playback discontinuity) and to assist the delay-constrained retransmission attempt if allowed. This local adaptation is then reviewed at the server based on the aggregated client feedback and the server will issue target presentation point to coordinate and synchronize all the clients. Note that RTCP-compatible signaling between the server and group-clients is performed for this server-aided inter-client synchronization.

Among many aspects of the proposed framework, in this paper, we are interested in verifying and evaluating the proposed framework. So, we choose to evaluate the proposed framework by conducting simple yet extensive network simulations. Results show that the proposed framework can reduce the playback discontinuity without degrading the media quality while enhancing the inter-client synchronization by mitigating the client heterogeneity.

The rest of this paper is organized as follows. The proposed one-to-many media streaming framework is introduced in Section 2. The core components of the framework are explained subsequently in the order of the adaptive playout with audio time-scale modification, the local adaptation with error recovery support, and the server-aided inter-client synchronization with feedback. Section 3 shows the simulation setup and the verification results of the proposed framework. After reviewing related works in Section 4, we conclude the paper in Section 5.

II. Synchronized Multicast Media Streaming Framework

1. Proposed Framework Overview

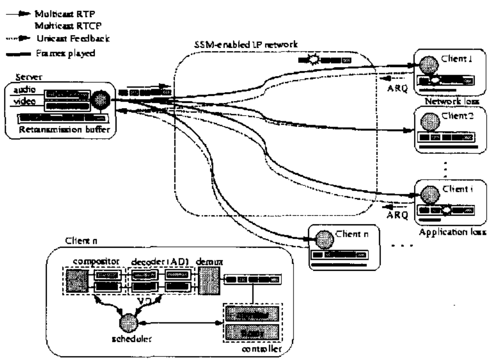


Fig. 1. The proposed one-to-many media streaming framework.

Fig. 1 illustrates the overall framework of the proposed synchronized multicast media streaming, where the adaptive playout control for audio/video streams (e.g., MPEG-2 and MPEG-4) is deployed to perform the media delivery through the SSM-enabled IP network. Multiplexed audio/video stream (e.g., MPEG-2 program stream (PS)[6]) is transmitted to all the clients joined through a PIM-SM (protocol independent multicast - sparse mode) routing. It is assumed that a single-layer-encoded stream is delivered to moderate number of clients (e.g., around 100) with receiver buffers. Under this kind of situation, one can expect clients to lose synchronization easily and play slightly different portion of stream compared to the other clients. Note also that we are assuming the lightly coupled synchronization, where the server and clients need to be synchronized within an allowed range. Note that this is in comparison to the tightly synchronized playout employing time synchronization such as network time protocol (NTP)[7].

The desired inter-client synchronization is accomplished in this work by performing the playout adaptation with audio time-scale modification[4],[5]. The synchronized playout is assisted with the feedback-based streaming control, which also can encompass the rate and error controls in the future. The simple retransmission-based error recovery can be enhanced with the explicit help of playout control, which tries to expand the time available for retransmission by slowing down the playback temporarily. Server can also help the error recovery implicitly by relaxing the target presentation point to that of slowest client to secure the time for retransmission. Also, the client can not do multicast feedback, since the neighboring router for each client is configured by IGMPv3[8] protocol to filter a source in the SSM. Instead, each client uses a customized unicast RTCP receiver report (RR) while the server multicasts a RTCP sender report (SR) to deliver control message[9].

2. Adaptive Playout with Audio Time-scale Modification

To provide high-quality video streaming and ease the client adaptation to network fluctuations and system dynamics, the client-based adaptive playout is adopted in this work. As shown in the magnified part of Fig. 1, the multiplexed stream is stacked at the receiver buffer of each client to wait the decoding and playback. The stream is de-multiplexed into the respective audio and video decoding buffers, where the presentation timestamp (PTS) located at the header of MPEG-2 PS is recorded into the scheduler. It also gets the current buffer occupancy level from the buffer monitor. Based on the timing and buffering status, the scheduler controls the adaptive playout by *expanding/contracting* the playout at both compositor and decoder. The buffer monitor of the controller checks the sequence number of incoming packets and determines the loss from network - it performs gap-based loss detection. The associated timer of controller provides the required timing information and controls the retransmission timing.

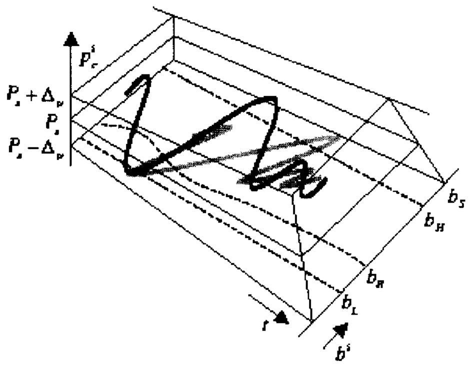


Fig. 2. Desired operation of the proposed adaptive playout control.

The main role of adaptive playout control is to reduce the discontinuity incurred by packet over-/underflow and momentary CPU overload. An effective playout adaptation allows us to avoid excessive packet dropping at the application, conceal the network fluctuation and thus minimize the degradation of perceptual media quality. With audio time-scale modification[4],[5], we can significantly enhance the adaptation capability of client at the cost of increased computation. To perform the adaptive playout with time-scale modification, we need to know the allowed range of ratio within which a player can manipulate the playback speed without being detected by the user's perception. Even though the allowed playout variation differs based on the type of audio (including the silence), playout variation up to 25% is usually unnoticeable. In this paper, we define the playback factor (Δ_p) to specify this variation ratio.

In general, the buffer size b_s of each client is typically constrained by the hardware complexity and the application requirement on the latency and error. When dealing with number of clients, it will be simple and much easier to coordinate the playbacks of whole client with larger size buffer. However, when it comes to the hardware cost, smaller size buffers with elegant buffer control is preferred. In this paper, we fix the buffer size of all clients as small as possible but large enough to service the worst case client with the largest round trip time (RTT) and loss. Then, we denote the current buffer occupancy level of client i at time t as $b^i(t)$ and specify two fixed thresholds for the receiver buffer as shown in Fig. 2. They are *low limit* (b_L) and *high limit* (b_H) thresholds. When $b^i(t)$ falls under b_L or exceeds b_H, there is high risk of buffer underflow or overflow, respectively. Normally, the control needs to keep $b^i(t)$ floating between b_L and b_H. By keeping the speed of client i at t, $p_c^i(t)$ within acceptable range $P_s\cdot(1-\Delta_p)$ ~ $P_s\cdot(1+\Delta_p)$ (i.e., $1\pm\Delta_p$ of its normal playback speed P_s), we can prevent exceptional events that may result in unnecessary packet discard. We introduce another implicit threshold, namely *retransmission limit* (b_R), to guide the retransmission-based error recovery. With this guideline, the controller determines whether it request the retransmission of lost packet or not.

3. Local Playout Adaptation with Error Recovery Support

Under the proposed framework, each client is performing local playout adaptation. The major role of local playout is heavily tied with the buffer control. It adapts to control the combined (in this work, decoder only) buffer occupancy guided by a playback factor. As discussed above, the local playout adaptation attempts to manage $b^i(t)$ between b_L and b_H. By helping the buffer-level control within the range as shown in Fig. 3, the adaptive playout reduces the risk of buffer underflow/overflow.

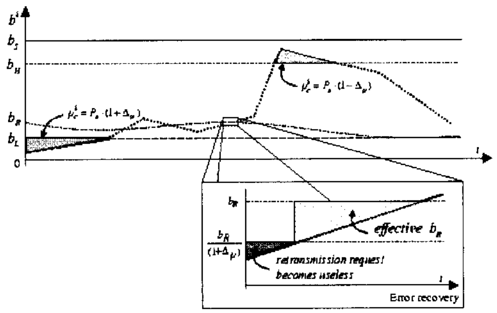


Fig. 3. Key roles of local playout adaptation.

With the adaptive playout based on time-scale modification, we can manage the buffer level more flexibly. It helps us to overcome the exceptional system events and the network fluctuations. For unicast case, even delay-stringent media streaming (towards interactive and conversational) can be better accommodated with the playout adaptation[5].

We can extend the role of local playout adaptation in order to secure the time required for retransmission request/reply. It is well understood that, the error recovery with retransmission is applicable only when the RTT is short enough. Normally, when a packet of client i is lost, the buffer occupancy level $b^i(t)$ should be higher than b_R to secure time for retransmission. Thus, actually we need to maintain the buffer level between b_R and b_H. Note that b_R is replacing b_L in this case. However, with adaptive playout control, we can secure additional time by slowing down the playback speed. The threshold buffer level for successful retransmission-based error recovery - the *effective* b_R - can be lowered to $\frac{b_R}{(1+\Delta_p)}$. It thus enlarges the possible range for retransmission to $\frac{b_R}{(1+\Delta_p)} \leq b^i(t) \leq b_H$ as shown in the magnified portion of Fig. 3.

Another key role of local playout adaptation at each client is towards the synchronized playout. If there exists noticeable discrepancy in the audio and its corresponding video or temporary disruption of playout, it disturbs the audience a lot. With synchronization, we typically refer to the reconstruction of temporal relation between different media objects. There are several types of synchronization such as *intra-media*, inter-media, and *inter-client* synchronization[10]. For local playout adaptation, the relevant synchronization issue is so called intra-client (both intra-media and inter-media) synchronization. That is, each client separately manages the synchronization based on its own time reference. Intra-media synchronization defines the temporal constraints of consecutive media contents (e.g., packets of a single stream) and handles the timing dependency not to degenerate the perceptual media quality. Dynamic mechanisms like discarding and skipping, shortening and extension of output duration, and virtual time contraction and expansion have been developed for intra-media synchronization to compensate the delay jitter[11]. Inter-media synchronization (e.g., lip synchronization for video and audio) defines the temporal relationships between different types of media objects. Controlling the skew, i.e., the time difference between time-dependent presentation units, is a key issue to be solved here. The permissable skew level between different types of media objects is tightly coupled with human perception.

In this paper, we assume that globally synchronized time reference is not available (i.e., lightly coupled synchronization). Thus, each client is responsible to manage the synchronization based on its own time reference, namely local virtual time. The time reference is guided by PTS timestamps delivered with media packets. Also, we focus on the intra-media aspect of intra-client synchronization. Since the intra-client synchronization issue is closely tied with inter-client synchronization, we will examine the detailed playout adaptation for both types of synchronization in the following section.

4. Server-aided Inter-client Synchronization with Feedback

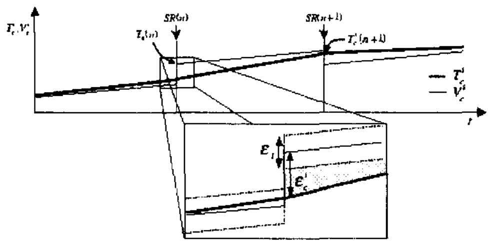


Fig. 4. Server-aided client playout adaptation.

Fig. 4 depicts the proposed playout operation to maintain the playback close to a target presentation time. As mentioned above, each client manages the synchronization based on its own time reference, namely local virtual time. Initially, the virtual time $v^i_c(t)$ starts concurrently with the start of each playout at t_{start} and the timer records the current presentation time $T^i_c(t_{start})$. Note that media stream is assumed to contain necessary presentation timing information (e.g., PTS). Using $T^i_c(t_{start})$ as the reference, we can to estimate the virtual presentation time at $t_{start}+\Delta_t$. Each client then compares its current presentation time $T^i_{c,t}$ to the virtual one $v^i_c(t)$. The target range is controlled by ε_t, which is centered around the $v^i_c(t)$ at t. Here, we denote by ε^i_c the amount of playout time discrepancy of $T^i_c(t)$ with respect to the $v^i_c(t)$. That is, ε^i_c means time difference between the virtual ($v^i_c(t)$) and actual ($T^i_c(t)$) presentation time. When the ε^i_c falls in the target synchronization range, i.e., $|\varepsilon^i_c| < 0.5 \cdot \varepsilon_t$, the playout adaptation is not necessary. When ε^i_c becomes greater than $\varepsilon_t/2$ or less than $-\varepsilon_t/2$, each client attempts to adjust the current presentation time and consequently move the buffer level. Each player either fasten or loosen its playback speed to compensate the time discrepancy ($|\varepsilon^i_c - 0.5 \cdot \varepsilon_t|$) until the $T^i_c(t)$ goes into the target range.

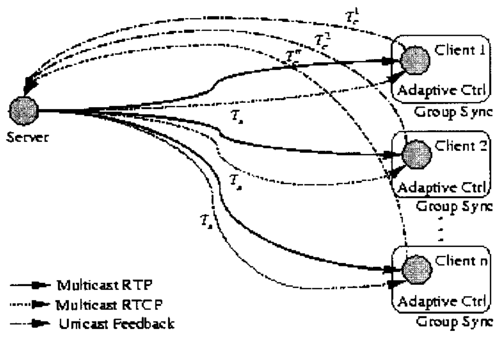


Fig. 5. Server-client signalling for inter-client synchronization.

Fig. 5 depicts the proposed operation of inter-client synchronization to maintain the playback close to a target presentation time. Remember that, in this paper, we are talking about lightly coupled synchronization, where the server and clients need to be synchronized within an allowed range. Thus, feedback-based synchronization is adopted to accomplish the required server-client synchronization. RTCP-compatible signalling between the server and group-clients is performed to exchange controlling messages including the presentation time and other status. Note that the exchange of controlling message needs to be restricted based on the RTCP bandwidth rule and, due to the employed SSM, each feedback is unicasted. The RTCP RR interval is determined based on the bandwidth of standard RTP and RTCP application-defined (APP) packet is customized to deliver the necessary timing information for the synchronization. When the server receives the compound RTCP RR packets, it gets the current presentation time of the client ($T^i_c(t)$) and it estimates the RTT^i - RTT^i of each client is approximated at the sender utilizing the unicasted RTCP RR reports. $RTT^i = t^i_{A_{RR}} - t_{DLSR} - t_{LSR}$, where $t^i_{A_{RR}}$ is the arrival time of RR packet at the server, t_{DLSR} is the delay elapsed from receiving last SR at the client and t_{LSR} is the time that has been copied from the server SR packet at the receiver. They are aggregated to predict the target presentation time, T_s. Each client is then notified about this target presentation time when it receives the RTCP SR from the server.

To determine the target T_s from the aggregated feedbacks from all the clients, the server needs to set an arbitration policy. It is important to adjust the fairness or performance of all the clients in terms of synchronization. For example, we can choose a simple averaging for the arbitration and thus T_s is calculated by averaging all values of T^{i}_{c}'s. Or we can use other policies such as median, minimum or maximum. The detailed calculation for T_s is as follows. Even though we are talking about gathering feedback from all the clients as a response to the server SR, the feedback from each client is returning at arbitrary time point (or maybe lost during feedback). To compensate this, we maintain the delay d^{i}_{s} for each client at the server, i.e., the arbitrary delay after feedback reception till the server SR. Also the feedback from each client takes different RTT^{i} (to be precise, half of RTT^{i}) to reach the server. Finally, the target presentation time T_s is calculated as

$$T_s = F_{arbitration}\{T^{i}_{c}, d^{i}_{s}, RTT^{i} \mid i = 1.....N_c\},$$

where $F_{arbitration}$ implies the selected arbitration policy, i stands for client i, N_c is an active group size, d_{s^i} is the processing delay from last RR to current SR, respectively. With the calculated T_s, the server multicasts the compound RTCP packets containing T_s to all clients and each client adapts its playback speed.

The above Fig. 4 also illustrates the detailed behavior of server-aided client adaptation with respect to the presentation time. Note that the server-aided client adaptation is tightly coupled with the local playout adaptation. At this stage, we are using simple approach to coordinate local and server-aided adaptations. Simply speaking, we are just replacing local virtual time with the guide from the server. Note however that it is very important to make the switch among local and server-aided adaptations smooth so that the playout at each client is consistent. The detailed procedure is as follows. While each client tries to adapt the local presentation time $T^{i}_{c}(t)$ to its virtual presentation time $v^{i}_{c}(t)$, it receives the target presentation time $T_{s(n)}$ from the RTCP SR. Upon the reception of target presentation time, each client goes through transient adaptive playout period by simply resetting its reference presentation time of virtual timer to server-reported $T_{s(n)}$. That is, the virtual time is restarting with the $T_{s(n)}$ as its reference presentation time when it receives $SR(n)$. After reset, the player goes into the local adaptation state to re-adjust the presentation time within allowed range.

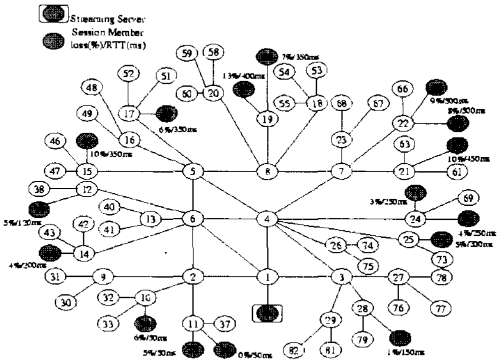


Fig. 6. Simulation topology.

Ⅲ. Simulation Results

1. Simulation Setup

Fig. 6 provides the detailed simulation topology by the network simulator (NS2). The SSM is provided by the PIM-SM multicast routing and each client feeds back its status through an unicast RTCP RR. There exists one server and 16 clients will join to the group from arbitrary locations.

We simulate the MPEG-2 audio/video streaming scenario at average 5 Mbps with 30 f/s frame rate. This frame rate leads us to the temporal granularity of 33 ms. The adopted group of picture (GOP) structure is for 15 frames consisting of one I-frame and 4 P-frames (i.e., IBBPBBPBBPBBPBB) and the bit rates of each frame is tabulated in Table 1. At this stage, the decoding and composition delay is ignored for the sake of simplicity. The timing accuracy of playing a frame is less than 10 ms in simulation.

Table 1. Video Traffic Model

	I(kbyes)	B(kbytes)	P(kbytes)
Avg. Frame Size	50	21.6	10
Std. deviation	8	2	1

All clients joining the multicast group have the same receiver buffer size (default size: 1 sec, unless specified else) and perform the same playout adaptation. Following thresholds are utilized for the receiver buffers. Lower limit b_L is set to $\frac{b_R}{(1+\Delta_p)}$ from the bottom, where the retransmission limit b_R is based on the estimated RTT. Higher limit b_H has margin equal to 100 ms from the top. We set the default *playback factor* (Δ_p) to 0.25 (25%), which is somewhat aggressive choice for time-scale modification. For the T_s arbitration policy, selection of minimum value is employed as default policy. Target synchronization discrepancy ε_t with respect to the virtual presentation time is set to 100 ms. Note also that the playout adaptation needs to be done in terms of video frame. The presentation time discrepancy is thus converted to the number of video frames N^i_f (based on the video frame rate R_f). Thus, we are getting the number of frames to be adaptively played out with time-scale modification, N^i_p, i.e., $N^i_p = N^i_f / \Delta_p$. For example, if we want to compensate 2 frames, then we needs to do the playout for 20 frames with $\Delta_p = 0.1$. In summary, the player of each client has to perform the playout adaptation up to N^i_p / R_f sec as long as the buffer level is properly controlled.

The simulation initiates just after 0 sec by starting the multicast streaming from the server. Each client joins to the group with a gap of 11 sec (fixed). After joining and starting the playback, all the clients are subject to a randomized, momentary CPU overload. Currently the CPU overload is happening for less than 50 ms at full strength (i.e., no other computation is possible) and it is randomly occurring with 10 sec exponential distribution. For the network, various combination of TCP and CBR traffics are traversing the whole network topology and causing network fluctuations between the server and clients. We select independent loss only on the receiver links as a loss scenario. The individual losses are occurred with uniform distribution and several clients experience burst losses with 50 to 100 ms exponentially distributed burst length. The server has one- second retransmission buffer. The average loss fraction is 6%, some clients suffer from maximum loss at 13% and a client experiences no loss (0%). With all these setups, we are using three measures for the synchronized streaming performance.
They are 1) accumulated playout discontinuity per each client during streaming, 2) the maximum difference of playout among clients, and 3) the playout speed variation of a client.

2. Verification Results

To verify the performance of the proposed approach, three different playout cases are compared. 'No playout adaptation' means the case where the playout control for both local adaptation and error recovery is not performed and adaptive playout cases are divided into 'Local adaptation' and 'Server-aided adaptation'. In both local and server-aided adaptation cases, the player request retransmission with the help of playout control.

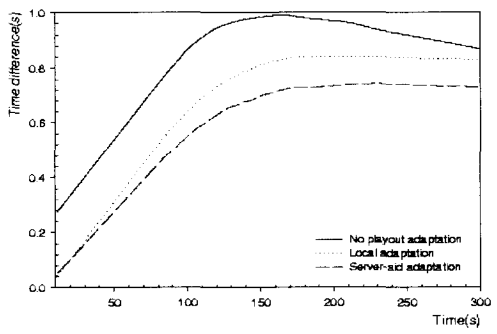


Fig. 7. Maximum playout time difference.

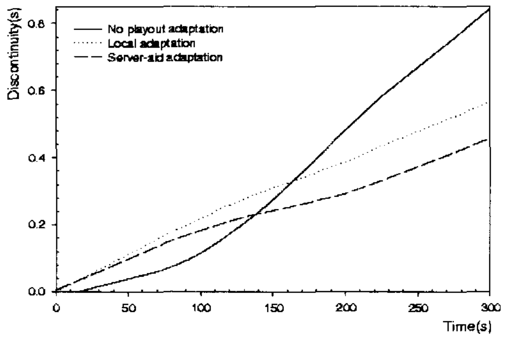


Fig. 8. Accumulated playout discontinuity averaged per each client.

For these cases, the maximum playout time difference between leading and trailing clients at each time t is shown in Fig. 7. Without the adaptive playout, the time difference increases up to maximum buffer occupancy level, i.e., b_S. Packet overflows or intentional flushing is useful in reducing the gap, but it pays the playout discontinuity. Check the accumulated discontinuity per each case in Fig. 8. It illustrates 0.8 sec discontinuity is occurring to each client on the average without the playout adaptation. In comparison, the proposed playout adaptation can bound the gap more effectively. If the player adopts local adaptation only, there is no way to restore inter-client synchronization since there is no guidance. Eventually it is subject to flushing. But with the help of server coordination, we can maintain and the gap throughout the whole playout. The accumulated playout discontinuity in Fig. 8 shows that the proposed playout adaptation with server-aided synchronization efficiently reduces the playout discontinuity to the half of no playout adaptation.

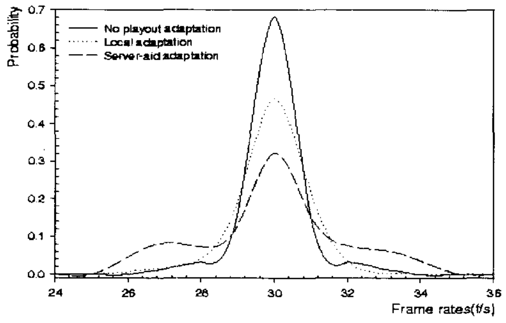


Fig. 9. Playout speed variation (a long-run client).

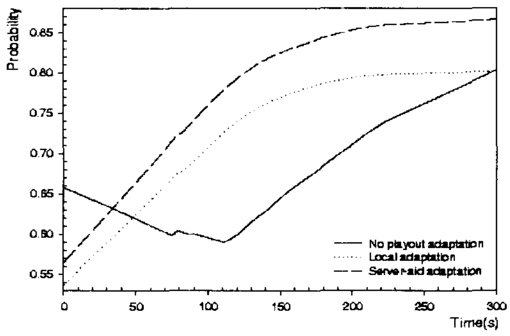


Fig. 10. Average repair probability at time t in a group.

Fig. 9 depicts the playout speed variation of a long-run client in the group. It is no doubt that the server-aided adaptation experiences more playout speed variation than the other cases at the expense of reduced discontinuity. Fig. 10 illustrates the averaged repair probability of lost packets in each client. It is confirmed that the coordination of the server and clients, i.e., server-aided adaptation, increases repair probability. Without the adaptive playout, the repair probability is highly fluctuating over the time.

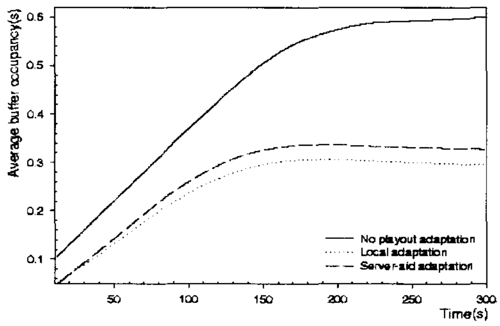


Fig. 11. Average buffer occupancy at time t in a group.

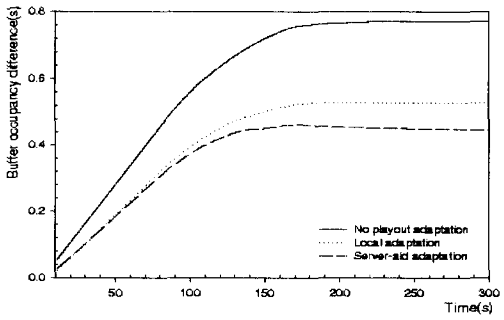


Fig. 12. Maximum buffer occupancy difference.

Fig. 11 shows the averaged buffer occupancy level of the session members at time t in the group. Note that, without adaptive playout, the average buffer occupancy becomes two times higher than those of adaptive playout. It is also observed that the buffer occupancy of server-aided adaptation is maintained slightly higher than that of local adaptation case. The maximum buffer occupancy difference between leading and trailing clients in the group is illustrated in Fig. 12. With the adaptive playout, the buffer level difference between group members is significantly reduced and server-aided adaptation maintains slightly smaller difference over the local adaptation. Since the server selects the minimum value as T_s under the default arbitration policy, it uses the presentation time of the slowest client in the group as the reference presentation time under the server-aided case. The adaptation to T_s at the clients maintains the buffer occupancy for all clients towards the same level (making the distribution of levels more uniform).

Table 2. Impact of T_s arbitration policy on the performance.

Ts selection	Discontinuity(s)	Playout Speed Variation(f/s)
Minimum	0.4812	2.8345
Median	0.5562	2.5396
Maximum	0.5812	2.3805
Average	0.4870	2.8291

The appropriate selection of T_s impacts on the overall performance as shown in Table 2. The result shows that T_s derived from minimum arbitration policy (i.e., $\min(T_c^1 + d_s^1 + RTT^1, \cdots, \cdots, T_c^{N_c} + d_s^{N_c} + RTT^{N_c})$ is preferable to the others with respect to playback discontinuity. That is, if we choose T_s according to the slowest client in the group, it increases the time available for retransmission for all clients. This in turn contributes to the enhanced error recovery and hence results in reduced discontinuity compared to other arbitration policies. The averaged T_s selection also reduces the playout discontinuity and speed variation compared to the other policies.

Ⅳ. Related Works

Inter-client synchronization and error recovery with playout adaptation in combination with TCP-friendly rate control[12] are necessary to provide high-quality video streaming service. Synchronized playout of participating clients forms the support foundation on which a server easily deploy its network adaptation mechanisms (e.g., including flow and congestion controls). That is, the necessary server-client interaction for the rate, error, and synchronization can be better accomplished with the adaptive playout of each client.

The *playout adaptation* is required to maintain the playout quality considering the network fluctuations and system limitations[13],[14],[15]. Previous works on the adaptive playout control mostly focus only on the intra-client synchronization issues. Multimedia player with an application-level CPU scheduler adapts the playout speed based on the buffer occupancy level, which is linked to the system loads[14]. It is however focused on the system capability variation than the network fluctuation. An adaptive stream synchronization protocol (ASP)[15] is used to control the synchronized playout based on the buffer fullness. It supports the notion of master and slave streams to coordinate inter-stream synchronization. However, it covers the signaling for synchronization, and the specific mechanisms to make streams synchronized are kept open. As a promising solution candidate, the adaptive playout with audio time-scale modification[5],[16] is proposed to synchronize the clients without hurting the playout quality (i.e., without any disruption). They allow us to perform the playout adaptation to gracefully conceal the network delay jitters and packet losses in the Internet. Especially, [16] addresses the adaptive media playout to relax the delay constraint of retransmission and minimize the buffer underflow. Based on both channel condition and buffer fullness, [16] considers delay and loss resiliency with the help of adaptive playout control.

The *inter-client synchronization* is rarely discussed yet. It is important to achieve harmonized playout among clients while maintaining fairness among multicast group members. It should be done by effectively mitigating the network and system heterogeneity of clients. [15] deals with the inter-client synchronization as well as the intra-client one by propagating the adapt messages. [17] and [18] introduce inter-client synchronization for live and stored media in multicast environments, respectively. In [17], a synchronization agent is used to exchange control packets. To adjust the presentation time, the delay (expressed in levels) at each client is reported. However, it lacks in considering the propagation delay and the feedback implosion due to short-term fluctuations in the network. For the stored media, [18] utilizes a multicast group disseminating control packets from a master to slave destinations. It requires message exchanges among session members, which is impossible in the SSM environment. Also, other works [19],[20],[21] consider the fairness among group members. However, these differ from our work since we exploit the gain of adaptive playout for the inter-client synchronization and the error recovery under the multicast streaming environment.

V. Conclusions

This paper proposed a new inter-client synchronization framework with adaptive playout control suitable for one-to-many multicast media streaming for low latency applications. By employing a server-client coordinated adaptive playout control with feedback for presentation time synchronization and error recovery, it varies the playback speed within the perceptual limit and adapts the presentation time of each client to help the inter-client synchronization. For the error recovery, each client performs retransmission request with the assistance of playout control to conceal long repair latency and the server also implicitly helps securing the time available for retransmission by coordinating the clients. The proposed adaptive playout controls the playback speed of audio and video by adopting the time-scale modification of audio. RTCP-compatible signalling between the server and group-clients is also proposed, where the exchange of controlling message is restricted. The network-simulator based simulations show that the proposed framework can reduce the playout discontinuity without degrading the media quality, and thus mitigate the client heterogeneity.

REFERENCES

[1] C. Diot and B.N. Levine and B. Lyles and H. Kassem and D. Balensiefe, "Deployment issues for the IP Multicast Service and Architecture", *IEEE Network*, vol. 14, pp. 88-98, Jan. 2000.

[2] H. Holbrook and B. Cain, "Source-specific multicast for IP", draft-ietf-ssm-arch-00.txt, IETF, Nov. 2001.

[3] A. Schulzrinne and S. Casner and Frederderick and V. Jacobson, "RTP: A transport protocol for real-time applications", IETF RFC 1889, Jan 1996.

[4] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech", in *Proc. IEEE ICASSP*, Apr. 1993, pp. 554-57.

[5] F. Liu, J. Kim, and C.-C.J.Kuo, "Adaptive delay concealment for internet voice applications with packet-based time-scale modification", in *Proc. IEEE ICASSP*, May 2001.

[6] D. Hoffman and et. al., "RTP payload format

for MPEG1/MPEG2 video", IETF RFC 2250, Jan 1998.

[7] D. L. Mills, "Internet time synchronization: The network protocol", *IEEE Trans. on Communications*, vol. 39, no. 10, pp. 1482-93, Oct. 1991.

[8] B. Cain, S. Deering, and A. Thyagarajan, "Internet group management protocol, version3", Internet Draft draft-ietf-idmr-igmp-v3-01.txt, IETF, Feb. 1999.

[9] J. Chesterfield and et. al., "RTCP extensions for single-source multicast sessions with unicast feedback", Internet Draft draft-ietf-avt-rtcpssm-00.txt, IETF, Feb. 2002.

[10] Y. Ishibashi and S. Tasaka, "A comparative survery of synchronization algorithms for continuous media in network environments", in *Proc. IEEE LCN 2000*, Nov. 2000, pp. 337-48.

[11] Y. Ishibashi, S. Tasaka, and H. Ogawa, "A comparison of media synchronization quality among reactive control schemes", in *Proc. IEEE INFOCOM*, Apr. 2001, vol. 1, pp. 77-84.

[12] W. Tan and A. Zakhor, "Real-time internet video using error resillient scalable compression and TCP-friendly transport protocol", *IEEE Trans. on Multimedia*, vol. 1, no. 2, pp. 172-86, 1999.

[13] B. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment: performance bounds and algorithms", *ACM/Springer Multimedia Systems*, vol. 5, no. 1, pp. 17-28, 1998.

[14] J. Walpole and et. al., "A player for adaptive MPEG video streaming over the Internet", in *Proc. of the SPIE 26th Applied Imagery Pattern Recognition Workshop (AIPR)*, Oct. 1997.

[15] K. Rothermal and T. Helbig, "An adaptive stream synchronization protocol", in *Proc. of the Fifth international Workshop on Network and Operating System Support for Digital Audio and Video (NoSSDAV)*, Oct. 1997, vol. LNCS 1018, pp. 189-202.

[16] E. Steinbach, N. Farber, and B. Girod, "Adaptive playout for low-latency video streaming", in *Proc. International Conference on Image Processing (ICIP)*, Oct. 2001, pp. 962-65.

[17] Y. Ishibashi and S. Tasaka, "A group synchronization mechanism for live media in multicast communications", in *Proc. IEEE GLOBECOM*, Nov. 1997, vol. 2, pp. 746-52.

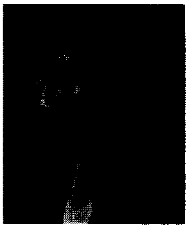
[18] Y. Ishibashi, A. Tsuji, and S. Tasaka, "A group synchronization mechanism for stored media in multicast communications", in *Proc. IEEE INFOCOM*, Apr. 1997, vol. 2, pp. 692-700.

[19] J. Escobar, C. Partridge, and D. Deutsch, "Flow synchronization protocol", *IEEE/ACM Trans. on Networking*, vol. 2, no. 2, pp. 111-21, Apr. 1994.

[20] Y. Ishibashi, S. Tasaka, and Y. Tachibana, "Adaptive casuality and media synchronization control for networked multimedia applications", in *Proc. IEEE ICC*, 2001, vol. 3, pp. 952-58.

[21] J. Jo and J. Kim, "Synchronization one-to-many media streaming with adaptive playout control", in *Proc. SPIE ITCOM*, July 2002.

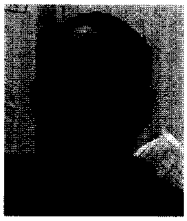
조 진 용(Jin-Yong Jo) 정회원



1999년 2월 : 전남대학교 컴퓨터 공학과 졸업
2002년 8월 : 광주과학기술원 정보통신공학과 석사
2003년 1월 ~ 현재 : KT 안양지사

<주관심분야> Internet QoS, Media Streaming, Network Management.

김 종 원(JongWon Kim) 정회원



1987년 2월 : 서울대학교 제어계측공학과 학사
1989년 2월 : 서울대학교 제어계측공학과 석사
1994년 2월: 서울대학교 제어계측공학과 박사
2001년 9월~현재: 광주과학기술원 정보통신공학과 부교수
2000년 7월~2001년 6월: 미국 InterVideo Inc., Fremont, CA, 개발자문
1998년 12월~2001년 7월: 미국 Univ. of Southern California, Los Angeles, CA, EE-Systems Department 연구조교수
1994년 3월~1999년 7월: 공주대학교 전자공학과 조교수

<주관심분야> Networked Media Systems and Protocols focusing "Reliable and Flexible Delivery for Integrated Media over Wired/Wireless Networks (네트워크미디어: http://netmedia.kjist.ac.kr)