

미디어 스트리밍을 위한 적응적 동기 지연시간을 이용한 그룹 동기화 기법

권 동 우*, 옥 기 수*, 김 현 우*, 주 흥 택^o

Group Synchronization Method

Using Adaptive Synchronization Delay Time for Media Streaming

Dongwoo Kwon*, Kisu Ok*, Hyeonwoo Kim*, Hongtaek Ju^o

요 약

본 논문에서는 모바일 스마트 기기들 간에 스트리밍 중인 미디어의 재생 위치를 동기화하기 위해서 재생 중인 미디어의 비트전송률에 따른 적응적인 지연시간을 적용하는 동기화 재생 기법을 제안한다. 이 기법은 미디어의 재생 정보와 비트전송률에 따라 계산된 동기 지연시간을 클라이언트 기기들에게 전달하는 서버 측 동기화 알고리즘과 서버 기기로부터 전달받은 제어 정보를 이용하여 정밀한 재생 동기를 유지하는 클라이언트 측 동기화 알고리즘으로 구성된다. 동기화 재생 알고리즘은 네트워크 연결시간, 동기화 제어패킷 전송시간, 스트리밍 데이터 전송시간, 동기화 처리시간의 네 가지 지연시간을 반영한다. 본 논문에서는 제안된 동기화 재생 알고리즘을 적용한 Android 애플리케이션을 구현하고, 동기화 처리시간과 기기별 동기화 성능 및 그룹 동기화 성능을 평가한다. 성능 실험 결과로 서버 기기와 클라이언트 기기들 간의 그룹 동기 재생위치 차이는 평균 39ms 미만으로 높은 정밀도의 성능을 보인다.

Key Words : Group synchronization, Adaptive delay time, Media streaming, Play sharing

ABSTRACT

In this paper, we propose a group playback synchronization method using adaptive synchronization delay time by the bit rate of media to synchronize a play position of streaming media between mobile smart devices. This method consists of streaming server-side and client-side synchronization algorithms based on synchronization delay time which includes connection time, control packet transmission time, streaming data buffering time, and synchronization processing time. We implement the Android media player application with synchronization support using the proposed algorithms and present the result of performance evaluation.

※ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 IT/SW 창의연구과정의 연구결과로 수행되었고(NIPA-2013-H0502-13-1099), 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과이며(2012H1B8A2025942), 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2012R1A1A2006331).

♦ First Author : Keimyung University, Department of Computer Engineering, dwkwon@kmu.ac.kr, 학생회원

o Corresponding Author : Keimyung University, Department of Computer Engineering, juht@kmu.ac.kr, 종신회원

* Keimyung University, Department of Computer Engineering, {ok7323, hwkim84}@kmu.ac.kr, 학생회원

논문번호 : KICS2014-10-385, Received October 6, 2014; Revised December 30, 2014; Accepted March 16, 2015

I. 서 론

스마트폰과 스마트 태블릿으로 대표되는 모바일 스마트 기기들의 급속한 확산으로 인하여 미디어 스트리밍 서비스와 미디어의 재생을 공유하는 동기화 재생 애플리케이션이 다양한 분야에서 활용되고 있다. Android 모바일 운영체제에서 동작하는 삼성전자의 Group Play는 미디어 소스를 가지고 미디어 서버 역할을 하는 단일 스마트 기기에서 다수의 다른 스마트 기기들에게 스트리밍을 제공하고 재생 위치를 실시간으로 동기화하여 재생하는 대표적인 모바일 애플리케이션 중 하나이다.

동기화 재생의 관점에서 유선 네트워크에 비해 네트워크 품질이 일정하지 못한 무선 네트워크의 특성과 미디어별로 서로 다른 인코딩 비트전송률(bit rate)은 스트리밍 중인 미디어의 정밀한 동기 재생을 어렵게 만드는 요인이 된다. 따라서 높은 정밀도를 가지는 동기화 재생을 위해서는 무선 네트워크의 특성과 미디어별로 변화하는 비트전송률을 고려한 재생 동기화 알고리즘이 필요하다.

본 논문에서는 미디어 재생과 함께 스트리밍 서버 역할을 하는 스마트 기기로부터 클라이언트에 해당하는 다수의 스마트 기기들에게 미디어 스트림을 전송하고, 동시에 재생되는 미디어의 재생 위치를 실시간으로 동기화하는 기법을 제안한다. 제안하는 동기화 재생 기법은 서버 측 동기화 알고리즘과 클라이언트 측 동기화 알고리즘으로 구성되며, 네트워크 연결시간, 동기화 제어패킷 전송시간, 스트리밍 데이터 전송시간, 동기화 처리시간을 포함한 동기 지연시간을 고려한다. 특히 미디어의 비트전송률에 따라 적응적으로 동기 지연시간을 조정하여 동기 정확도를 향상시키고 서버와의 미세한 재생 위치 차이를 보정하여 높은 정밀도의 동기화 성능을 제공한다.

제안하는 동기화 재생 알고리즘은 높은 정밀도의 동기화를 유지하면서 스트리밍 서버와 클라이언트 간에 전송되는 동기화 제어 패킷의 수를 최소화하여 제어 부하를 줄인다. 그러므로 대용량의 스트리밍 트래픽으로 네트워크가 혼잡한 상황에서도 높은 정확도의 동기화 상태를 유지할 수 있다. 제안하는 동기화 기법은 Android 애플리케이션으로 구현하였으며, IEEE 802.11 infrastructure 네트워크 환경에서 실험을 수행하여 성능을 평가하였다.

이후 본 논문의 구성은 다음과 같다. 2장에서는 미디어 스트리밍과 동기화 재생에 관련된 연구에 대해 기술한다. 3장에서는 본 논문에서 제안하는 적응적인

동기 지연시간을 이용한 동기화 재생 알고리즘에 대해 설명한다. 동기화 재생 알고리즘은 서버 측 동기화 알고리즘과 클라이언트 측 동기화 알고리즘으로 구성되어 있다. 4장에서는 동기화 알고리즘의 성능평가를 위한 실험환경 구성과 성능실험 결과에 대해서 분석한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해 논의한다.

II. 관련 연구

F. Boronat 등^[1]은 멀티미디어의 그룹(group) 및 스트림 간(inter-stream) 동기화에 관한 기존 연구들을 비교 분석하였다. 그룹 동기화 기술을 Master/slave receiver scheme, Synchronization Maestro Scheme (SMS) 및 Distributed Control Scheme (DCS)으로 분류하고 시각 기준, 네트워크 지연 제한, 미디어 유형, master/slave 관계성, 스트리밍 프로토콜 등의 기준으로 나누어 비교한 결과를 제시하였다.

H. Stokking 등^[2]은 목적지 간 미디어 동기화(Inter-Destination Media Synchronization, IDMS) 표준화에 대해 IETF AVTCORE와 ETSI TISPAN의 기술 명세에 중점을 두고 기술하였다. IDMS 기술과 IDMS 표준화에 관해 설명하고, IETF IDMS 기술에서 RTCP를 이용한 상태 정보 및 동기화 설정의 교환에 관한 표준화 내용과 ETSI IDMS 기술과의 차이점에 대해 논하였다.

D. Suh 등^[3]은 네트워크 품질 변동이 심한 네트워크 환경에서 Dynamic Adaptive Streaming over HTTP (DASH)에 기반을 둔 미디어 스트리밍 서비스를 제공하기 위해 사용자가 체감하는 품질(Quality of Experience, QoE) 저하를 감소시키는 적응적인 비트 전송률 알고리즘을 제안하였다. 제안한 알고리즘을 기존 연구들과 비교한 결과, 네트워크 품질이 가변적이거나 네트워크가 혼잡한 상태에 강건한 스트리밍 서비스를 제공할 수 있음을 확인하였다.

S. Son 등^[4]은 연결된 네트워크의 성능이 열악한 환경에서 고품질의 미디어 스트리밍 서비스를 제공하기 위해 요구 대역폭 추정을 통하여 적응적으로 서비스 품질(Quality of Service, QoS)을 제어하는 기법을 설계하고 구현하였다. 단방향 전송지연(Relative One-Way Delay, ROWD) 추정을 통하여 네트워크 성능을 추정하고, Scalable Video Coding (SVC) 중 시간적 부호화 방법을 사용하여 서비스 품질의 제어가 가능하도록 하였다. 제안된 기법을 Fiber To The Home (FTTH) 네트워크 환경에서 실험한 결과, 사용

가능한 네트워크 대역폭에 따라 미디어의 품질이 조정되어 스트리밍 서비스의 전반적인 품질이 향상됨을 확인하였다.

S. Hong 등^[5]은 실시간 미디어 스트리밍 서비스의 품질을 향상시키기 위해서 네트워크 계층 간 통합 및 최적화를 이용한 패킷 스케줄링 기법을 제안하였다. 프레임의 우선순위 판단을 통해 프레임 손실 시의 패킷 중요도를 구분하는 Significance Aware Packet Scheduling (SAPS) 알고리즘을 제시하였다. SAPS 알고리즘은 이용 가능한 대역폭 및 큐 크기에 적합한 패킷을 선택하고 전송을 수행하기 때문에 한정된 자원을 더욱 효율적으로 사용하여 미디어 스트리밍 서비스의 품질을 향상시킨다. 제안된 알고리즘의 성능을 측정하기 위해서 크기 기반 패킷 스케줄링 기법 및 비트전송률 기반 최선(best effort) 스케줄링 기법과 비교한 결과, 패킷 차별 효과를 통한 스트리밍 서비스의 품질이 개선됨을 확인하였다.

S. H. Kim 등^[6]은 이 기종 네트워크에서 다수의 멀티미디어 스트리밍 데이터를 동기화하고 통합하는 방법을 제안하였다. 클라이언트는 서버에 의해 전달되는 타이밍 정보를 이용하여 두 개 이상의 서버로부터 전송되는 스트리밍 데이터의 동기화를 제공한다. 타이밍 정보는 서버와 클라이언트 사이의 전파지연을 고려한 지연시간을 이용하여 계산된다. 이 방법은 클라이언트 측에서 미디어 스트림을 조절할 수 있도록 유연성을 제공하여 동기화가 이루어지도록 하였다.

M. A. Mughal 등^[7]은 라이브 모바일 협력 비디오 제작 시스템에서 새로운 유형의 동기화 문제를 논의하고, 비디오 제작 시스템에서 동기를 맞추기 위한 알고리즘을 제안하였다. 제안된 솔루션은 변동적인 대역폭 상황에서 비디오 프레임 속도를 조절하여 동기화가 이루어지도록 하는 기법을 적용하였다. 이 방법은 버퍼링(buffering) 현상을 방지하여 동기화 지연시간을 감소시켰다.

D. Lee 등^[8]은 스테레오 비디오를 동기화시켜 3차원 효과를 최대화하는 비디오 전송 방법에 대해서 제안하였다. 이 방법은 사람이 인식할 수 있는 최대한의 3차원 효과 범위를 고려하여 다중 비디오를 동기화한다. 3차원 효과는 스트리밍 받은 미디어의 취득시간 차이를 구하고 일정 허용범위 내에서 차이가 발생할 경우에 다중 미디어를 합성하여 구현된다. 제안된 기법을 사용하여 Mean Opinion Score (MOS)를 측정할 결과, Differential Mean Opinion Score (DMOS) 스케일의 좋음 범위 이상의 성능임을 확인하였다.

S. Moriyasu 등^[9]은 Video on Demand (VoD) 서

비스를 제공함에 있어서 동일한 공간에서 동일한 화면을 시청하는 효과를 줄 수 있는 동기화 재생 방법을 제안하였다. 온라인상에서 동기화 시청 그룹을 생성하고 하나의 호스트가 다른 클라이언트들의 재생을 제어하는 방법으로 재생/정지/탐색 각 명령에 대해서 동기화를 수행하며, Simple Network Time Protocol (SNTP)로 클라이언트들 간의 시각을 동기화하였다. 동기화 정확도는 고성능 시스템을 사용한 실험에서 120ms 이하의 정확도를 보였다.

F. Boronat 등^[10]은 응용 애플리케이션에서 사용 가능한 멀티미디어 그룹 동기화를 위한 솔루션을 제안하였다. Real-time Transport Protocol (RTP)/Real-time Transport Control Protocol (RTCP)을 이용하여 클러스터 사이에서 미디어를 스트리밍하고 RTCP Receiver Report Extension (RR EXT)과 RTCP Application-defined Action (APP ACT) 패킷을 이용하여 동기화를 제어하였다. 각 클러스터는 Network Time Protocol (NTP)을 이용하여 시각을 동기화하고, 스트림 간 동기화 오차는 $\pm 80ms$ 를 재생 위치 임계값으로 설정하여 평균 40ms 이하임을 보였다.

기존의 연구들은 대부분 유선 네트워크 환경에서 미디어 스트리밍 및 동기화 재생 기술들을 적용하여 실험을 수행하였다. 본 논문에서는 유선 네트워크에 비해 상대적으로 네트워크 품질이 고르지 못한 무선 네트워크 환경에서 다수의 모바일 스마트 기기들 간의 그룹 동기화 재생 방법을 제안하는데 초점을 두고 있다.

본 논문에서는 기기의 시각을 기준으로 하여 비트 전송률에 따른 동기 지연시간을 적응적으로 조정함으로써 적은 수의 동기화 제어 패킷만으로도 높은 정확도의 동기를 유지하는 재생 기법을 제안한다. 제안하는 기법을 기반으로 Android 미디어 동기 재생 애플리케이션을 구현하고, 실제 스마트 기기들을 사용하여 동기화 성능을 측정하는 방법을 제시하며 본 연구와 유사한 실험환경에서 동기화 성능을 측정할 기존 연구들^[9,10]과 성능을 비교하여 평가한다.

III. 그룹 동기화 재생 알고리즘

3.1 동기화 제어 패킷과 동기 지연시간

본 논문에서 제안하는 동기화 재생 알고리즘을 기술하기에 앞서 스트리밍 서버와 클라이언트들 간의 동기화를 위해 전송하는 제어 패킷의 형식에 대해 설명한다. 동기화 제어 패킷은 미디어 클라이언트 기기에서 스트리밍 서버 기기로 현재 재생 정보를 요청할

때 발생하며 서버 기기와 클라이언트 기기 간의 재생 위치를 비교하고 동기화 작업을 처리하는데 사용된다. 그러므로 RTP/RTSP/RTCP와 같은 특정 스트리밍 프로토콜에 의존하지 않고 스트리밍 프로토콜 종류에 상관없이 동기화 재생이 이루어지도록 한다. 동기화 제어 패킷의 형식은 표 1과 같다.

동기화 제어 패킷은 Play Status, Device Time, Media Play Position, Sync Delay Time 네 개의 필드로 구성되어 있다. Play Status 필드는 현재 서버의 재생 상태를 나타내는 필드로서 서버 측에서 재생 위치가 변경되었을 때 1, 재생 위치가 변경되지 않고 계속 재생 중일 때는 2, 재생이 중지되었을 때는 0으로 설정되며 초기 설정 값은 2이다. Device Time 필드는 동기화 기준점이 되는 시각으로 서버 기기의 시각으로 설정된다. Media Play Position 필드는 현재 서버가 재생 중인 미디어의 현재 재생 위치(재생시간)를 의미하며, 밀리 초(milliseconds)의 정밀도를 가진다. 마지막으로 Sync Delay Time 필드는 동기화 재생을 위해 사용되는 지연시간이다.

동기 지연시간(sync delay time)은 서버 기기가 현재 재생 위치를 전송하고 클라이언트 기기가 송신된 재생 위치를 전송 받기까지의 소요되는 시간과 클라이언트 기기에서 수신 받은 재생 위치를 재생하기 위해 필요한 스트리밍 데이터 전송시간을 고려하여 클라이언트 기기가 동기화 재생 전에 대기해야 할 지연시간을 의미한다. 즉, 네트워크 연결시간, 동기화 제어 패킷 전송시간, 스트리밍 데이터 수신시간 및 동기화 처리시간을 종합한 시간이다. 동기 지연시간은 스트리밍 데이터 수신시간에 가장 큰 영향을 받으며, 이 시간은 스트리밍 중인 미디어의 비트전송률에 비례하여 증가한다. 본 논문에서 제안하는 동기화 재생 알고리즘은 미디어의 비트전송률에 적응적으로 동기 지연시간을 재계산하여 동기화 정밀도를 향상시킨다. 계산된 동기 지연시간은 모든 클라이언트 기기들에서 동일한 값으로 적용이 된다.

스트리밍에 참여하는 모든 기기(서버 및 클라이언트)들이 서버 기기의 시각으로 동기화를 한 상태에서

서버 기기가 클라이언트 기기에게 제어 패킷에 서버의 현재 시각을 저장하여 송신하면 이 패킷을 수신한 클라이언트는 서버와 클라이언트 기기 간의 정확한 제어패킷 전송시간을 계산할 수 있다. 동기화 제어패킷의 전송시간을 알면 Media Play Position 필드의 값을 이용하여 현재 시각에 스트리밍 서버 기기가 재생 중인 위치의 계산이 가능하다. 또한 클라이언트 기기는 계산된 재생 위치와 자신의 현재 재생 위치를 비교하여 동기화 재생이 얼마나 정확하게 이루어지고 있는지 계산할 수 있다.

동기화 제어 패킷은 서버에서 각 클라이언트로 500ms마다 전송되고 패킷의 크기는 15바이트 이하이다. 본 논문에서 제시하는 실험에서는 제어패킷 전송 간격을 최대 1,000ms로 설정할 때까지 동기화 정확도에 영향을 미치지 않았다. 하지만 재생위치 이동과 정지 상태의 동기화 반응속도를 높이기 위해서 500ms 간격으로 제어 패킷을 전달하도록 설정하였다.

3.2 스트리밍 서버 기기의 동기화 재생 알고리즘

스트리밍 서버 기기와 클라이언트 기기들은 동기화 제어 패킷을 송수신하여 현재 재생 위치를 동기화한다. 그림 1은 서버 기기 측의 동기화 재생 알고리즘을 나타낸 것이다.

본 논문에서 제안하는 동기화 재생 알고리즘은 스마트 기기의 시각을 기반으로 동기화 기준점이 설정되기 때문에 동기화 과정에 앞서 서버 기기의 시각을 NTP 서버를 이용하여 표준 시각과 동기화한다. 동기화가 완료되면 선택된 미디어의 스트리밍을 시작하고 스트리밍이 시작되면 서버 기기는 동기화 제어 패킷에 넣을 현재 시각과 현재 재생 위치 및 동기 지연시간을 계산한다.

동기 지연시간은 재생 중인 미디어의 비트전송률에 대해 적응적으로 계산되어 적용된다. 미디어의 비트전송률에 따른 스트리밍 데이터 수신 지연시간은 수식 (1)에 의해 계산된다. 재생시간(total play time)은 미디어의 전체 재생시간을 의미하고 미디어 크기(media size)는 미디어의 전체 데이터 용량을 의미한다. 비트전송률 근사 값은 미디어 크기에 미디어 재생시간을 나누어 계산한다. 수식 (1)에서 지연 조정 계수(delay adjustment factor)는 비트전송률을 적합한 지연시간 단위로 변환하고 그 크기를 조정하기 위한 계수로서 본 논문에서는 반복된 실험을 통하여 320을 동기화에 적합한 수로 산출하여 적용하였다.

표 1. 동기화 제어 패킷 형식
Table 1. Synchronization control packet format

Field	Length
Play Status	1 byte
Device Time	9 byte
Media Play Position	variable
Sync Delay Time	variable

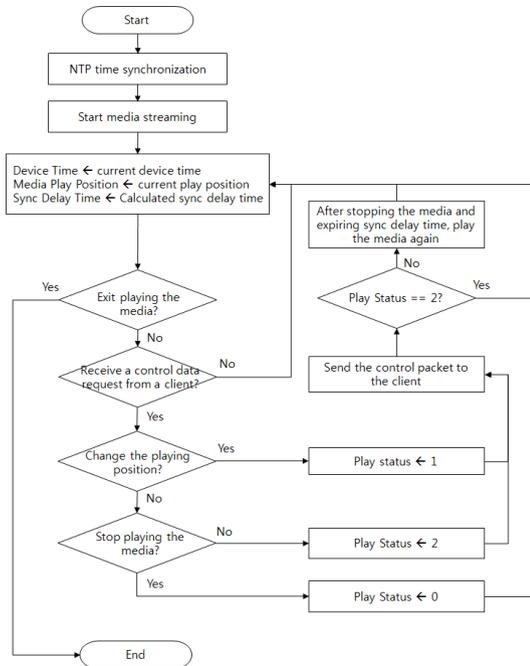


그림 1. 스트리밍 서버 측 그룹 동기화 재생 알고리즘
Fig. 1. Streaming server-side group playback synchronization algorithm

$$\text{Streaming Delay Time} = \frac{(\text{Media size})}{(\text{Total playtime}) \times (\text{Delay adjustment factor})} \quad (1)$$

이후 클라이언트 기기로부터 동기화 재생 정보에 대한 요청이 있으면 다음과 같은 세 가지 상태를 고려하여 표 1의 형식을 가진 동기화 제어 패킷을 전달한다. 첫 번째 상태는 서버 기기에서 재생 위치를 변경한 경우로 Play Status 필드에 재생 위치가 변경되었다는 의미로 데이터 1을 기록하고 현재 시각과 재생 위치, 그리고 동기 지연시간을 제어 패킷에 담아 클라이언트에게 전송한다. 그리고 서버 기기는 재생을 일시 중지하고 동기 지연시간만큼 대기한 후 다시 재생을 시작한다. 두 번째 상태는 서버 기기에서 재생 위치를 이동하지 않고 재생이 계속되고 있는 경우로 Play Status 필드에 데이터 2를 기록하고 제어 패킷의 나머지 필드에 데이터를 채운 후 클라이언트에게 전송한다. 마지막 상태는 서버 기기에서 재생을 중단한 상태로 Play Status 필드에 데이터 0을 기록하고 나머지 필드에 데이터를 기록한 후 제어 패킷을 클라이언트 기기에게 전달한다. 서버 기기 측의 동기화 재생 알고리즘은 서버 기기의 재생 상태에 따라 재생 중인 미디어의 재생 정보를 클라이언트 기기에게 전달하는

역할을 한다.

3.3 클라이언트 기기의 동기화 재생 알고리즘

클라이언트 기기 측의 동기화 재생 알고리즘은 그림 2와 같이 구성된다. 서버 기기로부터 동기화 제어 패킷을 수신 받은 클라이언트 기기들은 패킷에 담긴 미디어 재생 정보를 바탕으로 동기화 과정을 수행한다. 우선 클라이언트 기기는 서버 측 기기와 동일한 NTP 서버를 사용하여 표준 시각과 동기화한다. 그리고 클라이언트 기기는 서버 기기에게 미디어 스트리밍을 요청하고 스트리밍 재생을 시작한다. 재생이 시작되면 클라이언트 기기는 서버 기기에게 동기화를 위한 미디어 재생 정보를 요청하고 응답받은 재생 정보를 저장한다.

이후 클라이언트 기기는 서버 기기로부터 응답받은 제어 패킷의 Play Status 필드 데이터에 따라 다음과 같이 네 가지의 형태의 동기화 과정을 거친다. 먼저 Play Status 필드 데이터가 2이고 서버 기기로부터 제어 패킷을 처음 수신한 경우에는 저장된 스트리밍 서버의 재생 위치에 동기 지연시간을 더한 위치를 동기 지연시간만큼 정지한 후에 재생을 시작한다. 이 경우에 서버 측의 스트리밍 시작 위치에서 동기 지연시간의 길이만큼의 부분이 재생되지 않게 되는데 이는 스트리밍 시의 데이터 버퍼링과 동기화 재생을 위한 위치 조정으로 인해 발생하는 문제이다. Play Status 필

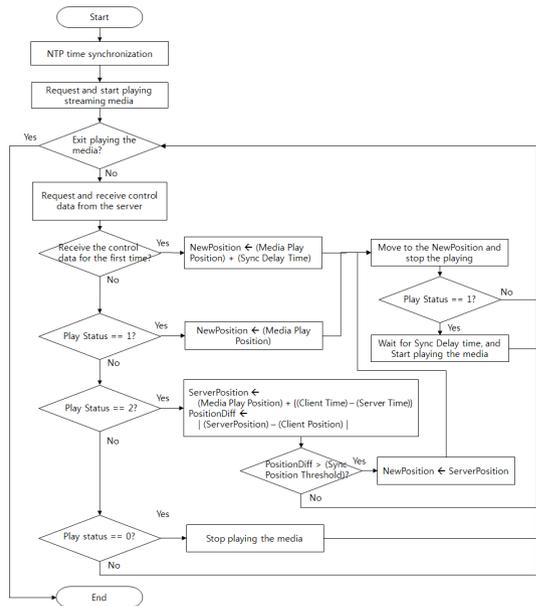


그림 2. 클라이언트 측 동기화 재생 알고리즘
Fig. 2. Client-side group playback synchronization algorithm

드 데이터가 0이면 서버 기기에서 스트리밍을 중단한 경우를 나타내며, 클라이언트 기기 측에서는 재생을 중단한다.

다음으로는 Play Status 필드 데이터가 1인 경우로서 서버의 미디어 재생 위치가 변경된 상태를 의미한다. 클라이언트 기기는 수신 받은 서버의 재생 위치로 이동하고 동기 지연시간만큼 대기한 후에 재생을 시작한다. 동기 지연시간에는 서버로부터 제어 패킷을 전송 받는 시간과 스트리밍 데이터 수신시간, 동기화 처리 시간이 반영되어 있기 때문에 동기 지연시간 후에 클라이언트가 재생을 재개할 시점에는 서버 기기와 동일한 재생 위치에서 재생이 시작된다.

그 외에 Play Status 필드 데이터가 2인 경우는 서버 기기에서 재생 위치의 변경 없이 계속적으로 재생이 이루어지고 있는 상태를 나타낸다. 클라이언트 기기는 수신 받은 제어 패킷의 재생 정보를 이용하여 서버 기기와의 재생 위치 차이를 계산하여 비교한다. 서버의 현재 재생 위치는 수식 (2)와 같이 서버로부터 전송받은 동기화 제어 패킷의 Media Play Position 필드의 재생 위치와 클라이언트 및 서버 시각(Device Time 필드 값) 차이의 합으로 계산된다.

$$Server\ Position = (Media\ Play\ Position) + \{(Client\ Time) - (Server\ Time)\} \quad (2)$$

수식 (3)과 같이, 서버 기기의 현재 재생 위치 계산이 완료되면 클라이언트 재생 위치와의 차의 절대 값을 통하여 재생 위치 차이를 구할 수 있다. 만약 주어진 임계값(sync position threshold) 이상의 재생 위치 차이가 발생하면 재생 위치를 즉시 이동시켜 정밀한 동기화를 가능하게 한다. 이러한 재생 위치의 미세한 차이는 NTP 시각 동기화 시에 발생하는 오차와 모바일 스마트 기기의 성능과 상태에 따른 재생 지연으로 인하여 발생한다. 재생 위치의 차이가 매우 작기 때문에 추가적인 데이터 버퍼링 시간이 필요하지 않다. 따라서 재생 위치를 즉시 이동시킬 수 있다.

$$Position\ Difference = |(Server\ Position) - (Client\ Position)| \quad (3)$$

선행음 효과(Haas effect)^[11]에 따르면 인간은 동일한 두 개의 소리가 1~30ms의 간격으로 발생할 때 두 소리를 구분하지 못하고, 두 소리의 간격이 50ms 이상인 경우에는 두 소리가 분리된 것과 같이 인식된다고 정의하였다. 본 논문에서 재생 위치 임계값은 인간이 동일한 두 소리를 확연히 구분할 수 있는 50ms의

시간과 스마트 기기 자체의 시각 오차 근사 값인 25ms의 합으로 설정(75ms)하여 실험을 수행하였고 이와 같은 클라이언트 동기화 재생 알고리즘을 적용하여 정밀한 수준의 동기화가 가능하다.

IV. 성능 평가

본 논문에서 제안한 동기화 재생 기법의 성능실험을 수행하기 위하여 제안된 동기화 알고리즘을 적용한 Android 미디어 재생 애플리케이션을 구현하고, 그림 3과 같은 실험환경을 구성하였다. 실험을 위한 모바일 스마트 기기는 삼성 갤럭시 플레이어 5.8(Android API 15, Ice Cream Sandwich) 세 대를 이용하였다. 이 기기들은 무선접속점(wireless access point) 장치에 연결되어 IEEE 802.11 infrastructure 네트워크를 구성하며 무선 네트워크 규격은 IEEE 802.11n 규격을 사용하였다.

그리고 비트전송률에 따른 적응적 지연시간을 이용한 동기화 성능을 측정하기 위해서 서로 다른 비트전송률을 가진 미디어들을 이용하여 실험을 수행하였다. 실험에 사용된 미디어의 비트전송률은 2000kbps, 1500kbps, 1000kbps, 500kbps, 100kbps의 다섯 종류로서 인코딩 포맷은 MPEG-4로 모두 동일하다. 미디어 스트리밍은 HTTP 프로토콜 기반 기술인 의사 스트리밍(pseudo streaming) 방식을 사용하였다. 의사 스트리밍은 점진적 다운로드(progressive download) 방식의 단점을 보완한 기술로써 메타 정보(meta data)를 사용하여 스트리밍 중에 다운로드 되지 않은 부분으로 재생위치 이동이 가능한 스트리밍 기술이다.

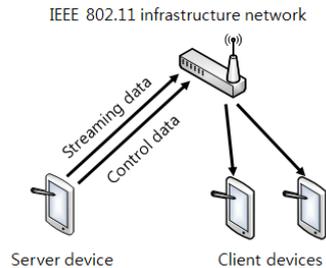


그림 3. 실험 네트워크 환경
Fig. 3. Experiment Environment

4.1 동기화 처리시간 측정 및 결과분석

첫 번째 실험은 모든 참여 기기들이 동기화가 완전히 이루어진 상태에서 재생 위치를 변경하고 다시 동기화가 이루어질 때까지의 소요시간을 측정하였다. 그

림 4는 각 비트전송률별로 미디어 당 10회씩의 동기화 완료시간을 측정한 값들의 평균과 표준편차를 나타내고 있다.

본 논문에서 제안하는 동기화 재생 기법은 동기화 정확도를 높이기 위해서 비트전송률에 따라 동기 지연시간을 적응적으로 조정한다. 따라서 그림 4의 실험 결과와 같이 비트전송률이 높은 미디어를 동기 재생 할 경우에는 비트전송률이 낮은 미디어보다 상대적으로 긴 동기화 완료시간을 가진다. 이것은 비트전송률이 높아짐에 따라 동기 지연시간이 길게 설정되기 때문이다. 만약 비트전송률이 높음에도 불구하고 충분한 동기 지연시간을 보장하지 않으면 스트리밍 데이터의 수신 지연시간으로 인하여 동기화의 정확도가 급격히 낮아지게 된다. 비트전송률에 따라 동기 지연시간을 적응적으로 조정함으로써 높은 비트전송률을 가진 미디어를 동기화 재생할 경우에도 원활한 재생이 가능하다.

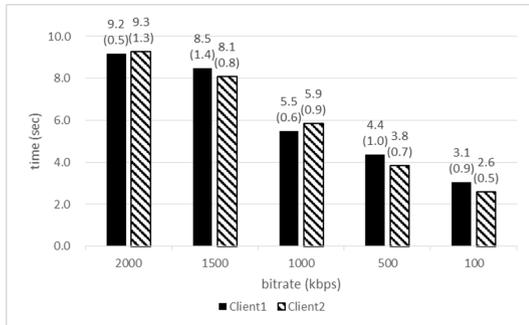


그림 4. 비트전송률에 따른 그룹 동기화 완료시간 측정결과
Fig. 4. Measured time for group playback synchronization at each bit rate

4.2 클라이언트 기기별 동기화 정확도 측정 및 결과분석

동기화 정확도 측정 실험은 두 가지 관점에서 수행되었다. 첫째는 서버 기기와 클라이언트 기기 간의 동기화 정확도를 1:1로 측정하여 정확도를 평가하였다. 둘째는 서버 기기와 동기화 그룹 내의 모든 클라이언트 기기들 간의 동기화 정확도를 측정하고 정확도를 평가하였다. 이 절에서는 클라이언트 기기별 동기화 정확도를 측정하고 그 결과를 분석한다.

수식 (4)는 재생 위치의 차이를 계산하는 수식이다. 이 수식에서 S는 서버 기기를 나타내고 C_i는 i번째 클라이언트 기기를 나타낸다. P_j는 j번째 측정 시의 재생 위치를 나타낸다. 서버 기기의 j번째 재생 위치(SP_j)와 i번째 클라이언트의 j번째 재생 위치(C_iP_j) 간의 차는

서버 기기와 i번째 클라이언트 간의 j번째 측정 시의 재생 위치 차이(C_iG_j)를 나타낸다. 수식 (5)는 i번째 클라이언트의 동기화 정확도를 계산하는 수식으로 n회의 재생 위치 차이들의 평균을 계산한다.

$$C_i G_j = SP_j - C_i P_j \quad (4)$$

$$Client Sync Accuracy_i = \frac{\sum_{j=1}^n |C_i G_j|}{n} \quad (5)$$

그림 5는 수식 (4)와 수식 (5)를 사용하여 1500kbps의 비트전송률을 가지는 미디어를 스트리밍 할 때 각 클라이언트별 동기화 정확도를 측정하여 나타낸 그림이다. 각 클라이언트 기기별로 서버와의 동기화된 재생 위치를 10회 측정하였고, 측정 회별로 20초간 재생 위치를 측정 하되 1초마다 한 번씩 측정한 재생 위치의 차이들을 평균하여 계산하였다.

측정결과, 첫 번째 클라이언트는 서버와 평균 25.38ms의 재생 위치 차이를 보였고, 두 번째 클라이언트는 서버와 평균 24.1ms의 재생 위치 차이를 보였다. 이 평균값들은 모두 50ms 미만의 값으로 선행음 효과에 따르면 인간이 인식하기 힘든 정도의 차이이며 정밀한 동기화가 이루어지고 있음을 확인할 수 있다.

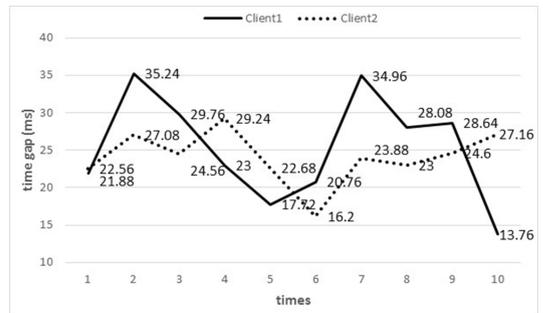


그림 5. 클라이언트 기기별 그룹 동기화 정확도 측정결과
Fig. 5. Play time gaps between the server and each client

4.3 그룹 동기화 정확도 측정 및 결과분석

이 절에서는 서버와 클라이언트 그룹 간의 동기화 정확도를 측정하여 동기화 그룹 내에서 전체적으로 동기화가 얼마나 잘 이루어지고 있는지 분석한다. 수식 (6)은 그룹 동기화 정확도를 계산하는 수식으로, 서버와 모든 클라이언트의 재생 위치 차이의 최솟값과 최댓값의 차를 계산하여 평균값을 구한 것을 그룹 동기화의 정확도로 정의하였다. 위치 차이의 최솟값은 서버의 재생 위치를 기준으로 가장 빠르게 재생되고

있는 기기와 재생 위치 차이를 의미하고, 최댓값은 가장 느리게 재생되고 있는 기기와 서버 간의 재생 위치 차이를 의미한다. 수식 (6)에서 CG_j 는 각 클라이언트 기기별 재생 위치 차이의 집합인 $\{C_1G_j, C_2G_j, \dots, C_nG_j\}$ 을 의미하며 정수 0은 서버의 재생 위치로서 동기화 기준 값을 나타낸다.

$$Group\ Sync\ Accuracy = \frac{\sum_{j=1}^n |MAX(0, CG_j) - MIN(0, CG_j)|}{n} \quad (6)$$

그림 6은 수식 (6)을 이용하여 비트전송률별 그룹 동기화의 정확도를 측정하여 나타낸 그림이며, 표 2는 그림 6에서 측정 비트전송률 당 재생 위치 차이의 평균과 표준편차를 계산한 것이다. 실험결과는 각 비트 전송률에서 평균 37.5~38.7ms 사이의 재생 위치 편차가 발생하였다. 선행음 효과에 따르면 이 값들은 모두 50ms 미만으로 인간이 인지하기 어려운 시간 차이이며 정밀한 동기화 재생이 이루어지고 있음을 알 수 있다.

S. Moriyasu의 연구⁹⁾에서는 동기화 성능 측정결과 120ms 미만의 정확도를 가지는 것을 보였고 미디어의 비트전송률 증가와 시스템 성능에 따라 동기화 정확도가 급격히 감소한다. 본 논문에서는 비트전송률에 적응적인 동기 지연시간을 사용하기 때문에 큰 폭의 비트전송률 증가에도 불구하고 위치 차이의 평균들이 거의 변화 없이 일정한 값을 유지하고 있다. 따라서 본 논문에서 제안한 동기화 기법은 비트전송률의 증가가 동기화 정확도에 미치는 영향이 매우 작음을 알 수 있다.

F. Boronat의 연구¹⁰⁾에서 제시한 동기화 정확도의 평균 오차는 미디어 소스 별로 각각 1283.66ms², 1912ms²으로 측정되었고, 이를 밀리 초로 환산 시 평균 39.78ms로 본 논문의 정확도 실험결과와 유사하였

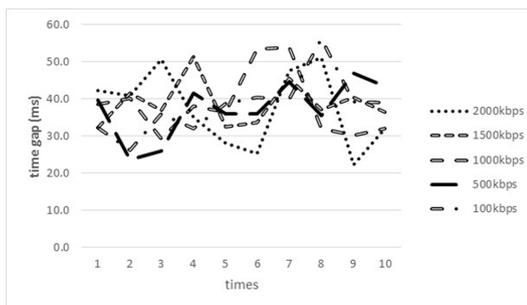


그림 6. 미디어 비트전송률별 그룹 동기화 정확도 측정결과 Fig. 6. Play time gaps between the server and the client group for each bit rate

표 2. 비트전송률별 동기화 그룹 재생 위치 차이 Table 2. Play time gaps for each bit rate of device group

Media Bit Rate	Average	Standard Deviation
2000kbps	37.5ms	10.57ms
1500kbps	38.7ms	6.07ms
1000kbps	38.4ms	8.87ms
500kbps	37.4ms	7.63ms
100kbps	38.0ms	7.86ms

다. 하지만 동기화 제어 패킷으로 인한 부하 관점에서 F. Boronat의 연구는 15,300개의 데이터 패킷이 전송될 때 192비트를 가진 364개의 제어 패킷이 발생하였고 이것은 2.38% 비율에 해당된다. 본 논문에서 제안하는 동기화 재생 기법은 222,900개의 데이터 패킷이 전달될 때 120비트의 제어 패킷이 410개 전송되며 이는 데이터 패킷 수 대비 0.18%의 비율이다. F. Boronat의 연구는 동기화를 시작하는 재생 위치 임계값으로 80ms를 사용하였기 때문에 비트전송률이 높은 미디어를 재생하여 재생 위치 차이가 더 빈번하게 임계값을 초과하게 되면 제시된 실험결과보다 제어 패킷의 수가 더욱 급격히 증가하는 단점이 존재한다. 본 논문에서 제안하는 기법은 500ms 간격으로 동기화 제어 패킷을 전송하며 재생 위치 임계값이 75ms 이상이 되었다고 판별되면 클라이언트 기기에서 동기화를 수행한다. 따라서 본 논문에서 제안하는 기법은 미디어의 비트전송률에 따른 제어 패킷 수의 증가 없이 매우 적은 수의 제어 패킷만으로도 높은 동기화 정확도를 유지한다.

V. 결론 및 향후 연구

본 논문에서는 모바일 스마트 기기들 간에 스트리밍 중인 미디어의 재생 위치를 동기화하여 미디어의 재생을 공유할 수 있는 동기화 재생 기법을 제안하였다. 이 동기화 기법은 재생하고자 하는 미디어의 비트 전송률에 적응적인 동기 지연시간을 적용하여 비트 전송률의 증가에 따른 동기화 정확도의 변화가 미미한 강건한 기법이다. 또한 클라이언트 기기의 시각 정보와 재생 경과시간을 기반으로 현재 서버의 재생 위치를 추정하여 동기화를 수행함으로써 서버의 재생 위치 정보를 담고 있는 동기화 제어 패킷의 잦은 수신 없이도 정밀한 수준의 동기화 유지가 가능하다.

제안한 동기화 기법의 성능 평가를 위해서 Android 미디어 동기화 재생 애플리케이션을 구현하고 성능

실험을 수행한 결과, 기기별 동기화 정확도는 평균 26ms 차이 미만, 그룹 동기화 정확도는 평균 39ms 차이 미만으로 매우 정밀한 재생 동기화가 가능함을 보였다. 선행음 효과에 따르면 이 실험결과는 인간이 인식하기 매우 어려운 수준이다.

향후 연구로는 재생하고자 하는 미디어의 비트전송률에 의한 고정된 동기 지연시간이 아닌 네트워크의 혼잡 상태를 반영한 동적인 지연시간을 적용하는 동기 지연시간 최적화에 관한 연구가 필요하다. 또한 서버 기기에서 클라이언트 기기의 동기화 정확도 수준을 판단하여 동기화 제어 패킷의 수를 더욱 최소화할 수 있는 연구가 수행되어야 한다.

References

[1] F. Boronat, J. Lloret, and M. Garcia, "Multimedia group and inter-stream synchronization techniques: A comparative study," *J. Inf. Syst.*, vol. 34, no. 1, pp. 108-131, Mar. 2009.

[2] H. Stokking, R. van Brandenburg, F. Boronat, and M. Montagud, "Standardization of inter-destination media synchronization," in *Proc. Media Synchronization Workshop 2012 (MediaSync 2012)*, Berlin, Germany, Oct. 2012.

[3] D. Suh, I. Jang, and S. Pack, "A video bitrate adaptation algorithm for DASH-based multimedia streaming services to enhance user QoE," *J. KICS*, vol. 39B, no. 06, pp. 341-349, Jun. 2014.

[4] S. Son, H. Lee, Y. W. Kwag, H. Yang, and J. Nam, "Design and implementation of network adaptive streaming through needed bandwidth estimation," *J. KICS*, vol. 35, no. 3, pp. 380-389, Mar. 2010.

[5] S. Hong and Y. Won, "Cross-layer design of packet scheduling for real-time multimedia streaming," *J. KICS*, vol. 34, no. 11, pp. 1151-1168, Nov. 2009.

[6] S. H. Kim, C. Lee, S. Kang, K. Seo, and T. Jung, "Timing control for synchronizing multimedia streaming over heterogeneous networks," in *Proc. Int. Conf. Advanced Commun. Technol. 2013 (ICACT 2013)*, pp.

260-263, Pyeongchang, Korea, Jan. 2013.

[7] M. A. Mughal, G. Zoric, and O. Juhlin, "Frame rate exclusive sync management of live video streams in collaborative mobile production environment," in *Proc. Mobile Video Delivery Workshop 2014 (MoViD 2014)*, no. 2, Singapore, Mar. 2014.

[8] D. Lee, S. Lee, D. Sim, and H. Lee, "Stereo-video synchronization for 3D video transmission," *J. KICS*, vol. 34, no. 4, pp. 349-359, Apr. 2009.

[9] S. Moriyasu, K. Tajima, K. Ohshima, and M. Terada, "Group synchronization method with fast response time for VoD Services," in *Proc. Int. Conf. Inf. Netw. 2012 (ICOIN 2012)*, pp. 182-187, Bali, Indonesia, Feb. 2012.

[10] F. Boronat, M. Montagud, and J. C. Guerri, "Multimedia group synchronization approach for one-way cluster-to-cluster applications," in *Proc. IEEE Conf. Local Comput. Netw. (LCN 2009)*, pp. 177-184, Zurich, Switzerland, Oct. 2009.

[11] Helmut Haas, "The influence of a single echo on the audibility of speech," *J. Audio Eng. Soc. (JAES)*, vol. 20, no. 2, pp. 146-159, Mar. 1972.

권 동 우 (Dongwoo Kwon)



2010년 2월 : 계명대학교 컴퓨터공학과 졸업
 2012년 2월 : 계명대학교 컴퓨터공학과 석사
 2013년 9월~현재 : 계명대학교 컴퓨터공학과 박사과정
 <관심분야> 미디어 스트리밍, 인터넷 침입 예측, 네트워크 관리 및 보안

옥 기 수 (Kisu Ok)



2015년 2월 : 계명대학교 컴퓨
터공학과 졸업
2015년 3월~현재 : 계명대학교
컴퓨터공학과 석사과정
<관심분야> 미디어 스트리밍,
미디어 그룹 동기화 재생,
네트워크 관리

주 흥 택 (Hongtaek Ju)



1989년 8월 : 한국과학기술원 전
자계산학과 졸업
1991년 8월 : 포항공과대학교 컴
퓨터공학과 석사
1997년 8월 : 대우통신종합연구
소 선임연구원
2002년 2월 : 포항공과대학교 컴
퓨터공학과 박사
2002년 9월~현재 : 계명대학교 컴퓨터공학부 교수
<관심분야> 네트워크 및 시스템 관리, IoT 관리,
SDN 네트워크 관리, 인터넷 침입 예측, 네트워크
보안

김 현 우 (Hyeonwoo Kim)



2010년 8월 : 계명대학교 컴퓨
터공학과 졸업
2012년 8월 : 계명대학교 컴퓨
터공학과 석사
2012년 9월~현재 : 계명대학교
컴퓨터공학과 박사과정
<관심분야> 방화벽 정책 추론
및 관리, 네트워크 관리 및 보안