

Normal Play Time 정보를 활용한 다윈 스트리밍 미디어 동기화 알고리즘

정태준, 고명필, 서광덕
 연세대학교 컴퓨터정보통신공학부
 kdseo@yonsei.ac.kr

Darwin Streaming Media Synchronization Algorithm Using Normal Play Time Information

Tae-jun Jung, Yuhee Jang, Kwang-deok Seo
 Division of Computer and Telecommunications Engineering, Yonsei University

요 약

본 논문에서는 IP 망을 통한 다윈 스트리밍 미디어 전송 서비스에서 압축된 미디어를 RTP 패킷화하여 전송할 때 RTP 패킷의 헤더에 기록될 타임스탬프 (timestamp) 정보로부터 유도해 낼 수 있는 Normal Play Time 정보를 활용하여 비디오와 오디오 간에 미디어 동기화 지원 방법을 제안한다. 모의실험을 통해 제안된 미디어 동기화 알고리즘을 적용함으로써 서로 다른 미디어 간에 정확한 동기화가 제공됨을 확인할 수 있었다.

1. 서론

최근 몇 년 사이에 인터넷은 다양한 멀티미디어 서비스를 수용하고 있는 가장 대중적인 망으로 성장하였다 [1]. 다윈 스트리밍 서버(DSS: Darwin Streaming Server)는 스트리밍 미디어를 산업 표준인 RTP(Real-time Transport Protocol)와 RTSP (Real-time Streaming Protocol)를 사용하여 인터넷을 통해 클라이언트에게 전송하는 Apple 사의 QuickTime 스트리밍 서버 기술의 오픈 소스 버전이다 [2]. DSS 는 유연한 코덱의 선택이 가능한 장점을 가지고 있으며 다른 QuickTime 툴들과의 통합도 지원된다.

DSS 에서는 스트리밍을 위한 시그널링과 서비스 제어를 위해 RTSP (Real-time Streaming Protocol)를 채택하며, 미디어 전송을 위해서 RTP 및 RTCP 프로토콜을 적용한다[3]. 그림 1 은 DSS 기반의 스트리밍 서비스를 위한 전체 동작 흐름도를 나타낸다. 표 1 은 그림 1 의 각 단계별로 이루어지는 세부 절차를 설명한다.

표 1. 스트리머 서비스 단계별 세부 절차.

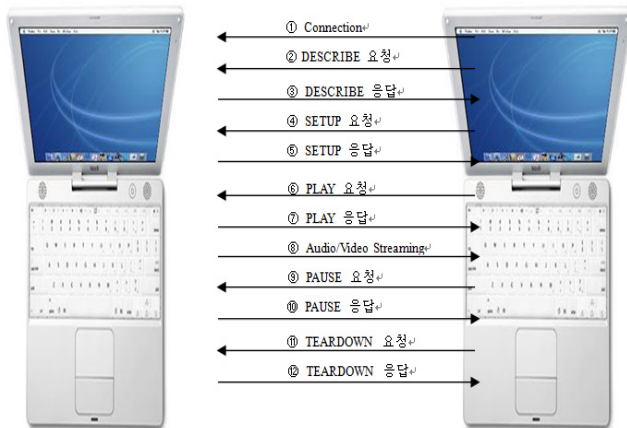


그림 1. DSS 기반의 미디어 스트리밍 서비스 시스템 동작 흐름도.

	Content
① Connection	RTSP 통신을 위한 TCP 연결
② DESCRIBE 요청	서버에게 특정 매체에 관한 정보를 요청
③ DESCRIBE 응답	서버의 DESCRIBE 응답
④ SETUP 요청	수용 가능한 전송 방식 요청
⑤ SETUP 응답	서버의 SETUP 응답
⑥ PLAY 요청	매체 전송 요청
⑦ PLAY 응답	서버의 PLAY 응답
⑧ Video Streaming(Play)	Client에서 Play버튼을 누르게 되면 Server에서는 네트워크 환경과 사용자의 환경을 근거로 하여 적절히 적응 변환된 비디오 비트스트림을 실제로 Client에 RTP 를 통해 전송하게 되고, 클라이언트에서는 전달받은 비트스트림을 실시간 디코더를 사용하여 디코딩하여 렌더러를 통해 Client GUI에 있는 비디오 창을 통해 비디오를 재생.
⑨ PAUSE 요청	일시 중지 요청
⑩ PAUSE 응답	서버의 PAUSE 응답
⑪ TEARDOWN 요청	세션 해제 요청
⑫ TEARDOWN 응답	서버의 TEARDOWN 응답

다윈 스트리밍 서비스에서는 미디어 동기화를 위해서 RTP 와 RTCP 패킷에 포함되어 있는 타임스탬프

정보를 활용한다. RTP 패킷에 포함되어 있는 타임스탬프 정보만으로는 미디어간 동기화 지원이 불가능하므로 RTCP 패킷에 포함되어 있는 Wall-clock 정보인 NTP (Network Time Protocol) 시간 정보를 활용해야 한다. 이를 위해서 RTCP 패킷을 주기적으로 전송하여야 하며, 그 결과 인터넷 망에 추가적인 트래픽 증가의 부작용이 발생할 수 있다 [4]. 이를 해결하기 위해서 본 논문에서는 기존의 RTP/RTCP 프로토콜 기반의 미디어 동기화를 대체할 수 있는 새로운 방법을 제안한다.

2. Normal Play Time 기반의 스트리밍 미디어 동기화 알고리즘 및 성능 결과

2.1 비디오 Time Stamp 와 Normal Play Time

일반적인 비디오의 샘플링 클럭은 90 kHz 이고 최대 30 fps 를 지원하므로, 한 프레임의 클럭 간격은 3000 이며 시간단위로 환산하면 1/30 초에 해당한다. 서버에서 TL (temporal level) 값을 이용해 생성된 각 화면에 대한 타임 스탬프는 비디오 부호화 특성상 일반적으로 순서적이지 않고 불규칙적이다. 따라서 클라이언트에서는 동기화를 위하여 각 화면에 대한 타임스탬프를 정렬하여 보관할 필요가 있는데, 이는 디코딩한 데이터가 화면에 출력될 때 시간 순서대로 출력된다는 점을 착안한 것이다. 첫 화면의 타임스탬프 값을 알고 있으면 현재 출력되는 화면의 Normal Play Time (NPT)를 RTP 타임스탬프를 통해 알 수 있게 된다.

$$\text{출력 영상의 NPT} = (\text{현재 출력 영상의 타임스탬프} - \text{첫 화면의 타임스탬프}) / \text{샘플링 율}$$

4.2 오디오 Time Stamp 와 Normal Play Time

mpeg4-generic 의 AAC-hbr 모드에서 AU (access unit)의 간격은 한 프레임을 나타낸다. 한 프레임당 클럭 간격은 1024 이고 타임스탬프 역시 한 RTP 페이로드에 포함된 AU 의 수에 따라 (1024 * AU 의 수)만큼 증가하게 된다. 비디오와는 달리 조금 더 복잡적으로 현재 디코딩되고 있는 오디오 데이터의 타임스탬프 값과 wave 출력 버퍼 한 개당 플레이 시간, 그리고 초기 버퍼링 시간을 알고 있으면 현재 출력되는 오디오 데이터의 타임스탬프를 알 수 있게 되고, 이를 통해 현재 출력되는 오디오 데이터의 NPT 를 구할 수 있다.

$$\text{초당 프레임 수} = \text{오디오 샘플링 율} / 1024$$

$$\text{현재 출력 오디오의 타임스탬프} = \text{현재 디코딩되고 있는 오디오 데이터의 타임스탬프} - (\text{디코딩 시작시간} - \text{플레이 시작시간}) * \text{초당 프레임 수}$$

$$\text{현재 출력 오디오의 NPT} = (\text{현재 출력 오디오의 타임스탬프} - \text{첫 출력 오디오의 타임스탬프}) / \text{샘플링 율}$$

2.2 Normal Play Time 을 이용한 미디어 동기화

위의 방법을 이용하여 오디오와 비디오의 NPT 를 구할 수 있었는데, 이 NPT 를 이용하여 오디오와 비디오 (A/V) 간의 미디어 동기화를 구현하였다. A/V 동기화에 있어서 동기화의 기준은 오디오이다. 따라서 기본 방법은 오디오 NPT 를 기준으로 비디오의 NPT 를 조절하는 것이다. 이것은 비디오의 디코딩된 데이터를 화면에 출력 시켜주는 타이머의 간격을 조절함으로써 가능한데, 비디오 타이머에서 AVSync() 함수를 호출하여 그 간격을 조절하게 해 주었다. AVSync() 함수의 역할은 현재 출력되기 직전의 오디오와 비디오의 NPT 를 계산하여 그 차이를 구하고, 오디오가 비디오보다 빠를 경우 - 지터를 주어 비디오 타이머 간격을 보다 작게 해주고, 비디오가 오디오보다 빠를 경우 + 지터를 주어 비디오 타이머 간격을 보다 크게 해주었다.

그림 2 는 오디오와 비디오의 NPT 간격의 차를 각 비디오 화면 별로 ms 단위로 계산한 실험 결과 그래프이다. 그림에서 초기에 동기화가 맞지 않는 이유는 SVC 비디오의 화면 전송 구조가 Temporal scalability 지원을 위한 Hierarchical B-picture 에 의한 GOP 구조 때문이다. 첫번째 GOP 의 경우 제일 먼저 전송되는 화면이 Key-picture 이며 이 화면의 타임스탬프 값은 IDR-picture 에 비하여 48,000 이 증가하게 된다. 이는 약 0.5 초에 해당되는 시간이다. 따라서, 초기의 급격한 타임스탬프 값의 상승에 의하여 오디오와의 동기화를 얻기 위해서는 2 초 내외의 수렴하기 위한 시간이 필요하게 된다.

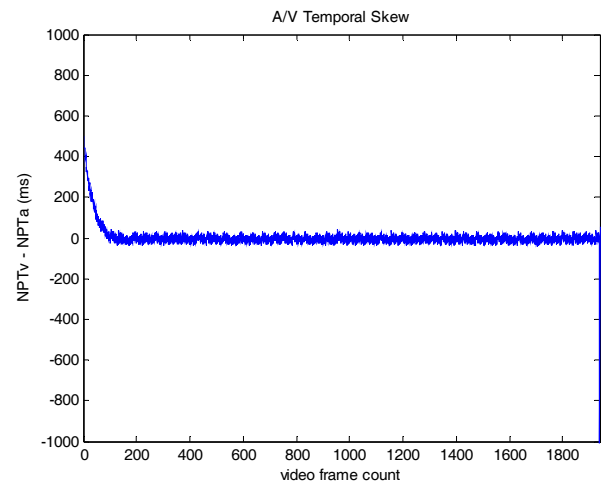


그림 2. 동기화 알고리즘 적용에 의한 A/V 재생 데이터 간의 시간적 차이 실험 결과

3. 결론

본 논문에서는 IP 망을 통한 다원 스트리밍 미디어 전송 서비스에서 압축된 미디어를 RTP 패킷화하여 전송할 때 RTP 패킷의 헤더에 기록될 타임스탬프(timestamp) 정보로부터 유도해 낼 수 있는 Normal Play Time 정보를 활용하여 비디오와 오디오 간에 미디어 동기화 지원 방법을 제안했다. 모의실험을 통해 제안된 미디어 동기화 알고리즘이 정확한 A/V 동기화를 지원할 수 있음을 확인하였다.

감사의 글

본 논문은 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 ‘바이오제조 GMP 기술인력양성사업’의 지원을 받아 수행하였습니다 (N0000961).

참고문헌

- [1] D. Wu, Y. Hou, and Y. Zhang, “Transporting real-time video over the Internet: Challenges and approaches,” *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1855–1877, Dec. 2000.
- [2] Apple Darwin Streaming Server—<http://developer.apple.com/darwin/projects/streaming/>.
- [3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “Real-time transport protocol,” *IETF RFC 3550*, July 2003.
- [4] L. Bertoglio, and P. Migliorati, “Intermedia synchronization for video conference over IP,” *Signal processing: Image Communication*, vol. 15, no. 1, pp. 149–164, 1999.