클라이언트 측 연접 저장 결과 재생을 위한 MP4 기반 DASH 세그먼트에 대한 조건

*김용한 **박민규 ***유성열 ***황서영
*,**서울시립대학교 ***삼성전자
*yhkim@uos.ac.kr

Conditions on the MP4-based DASH Segments for the Replay of Client-side Concatenated Storage Results

*Kim, Yong Han **Park, Minkyu ***Sungryeul Rhyu ***Seo-young Hwang
*,**University of Seoul ***Samsung Electronics Co., LTD

요약

DASH(Dynamic Adaptive Streaming over HTTP)는 MPEG(Moving Picture Experts Group)에서 표준화 중인 HTTP를 이용한 적응형 비디오 스트리밍 기술이다. 일반적으로 HTTP를 이용한 적응형 비디오 스트리밍에서 클라이언트가 사용할 수 있는 기능은 파일 전체 또는 그 일부를 다운로드하는 것이다. 따라서 전체 콘텐트를 작은 파일 조각 즉, 세그먼트(segment)로 분할하여 세그먼트들을 순차적으로 다운로드함으로써 마치 스트리밍이 달성되는 것과 같은 효과를 낼 수 있다. 네트워크의 상태에 따라 전송률을 조절하기 위해서는 서버에 서로 다른 비트율로 부호화된 세그먼트들을 함께 보관해 두어 클라이언트로 하여금 선택할 수 있게 한다. DASH에서는 MPEG-2 TS 또는 MPEG-4 파일 포맷(MP4)를 기반으로 하는 두 가지 형태의 서버 콘텐트를 제공할 수 있다. MP4 기반 DASH의 경우, 클라이언트가 수신한 세그먼트들을 순차적으로 이어 붙여 저장한다 하더라도 특별한 조건을 만족하는 경우를 제외하고는 MP4 파일을 재생할 수 있는 기존 미디어 플레이어가 이를 재생하지 못한다. 본 논문에서는 이러한 특별한 조건을 제시하고 이를 검증하였다. 이 조건은 서버에서 제공하는 세그먼트들에 대한 조건이다.

1. 서론

DASH(Dynamic Adaptive Streaming over HTTP)는 MPEG(Moving Picture Experts Group)에서 표준화 중인 HTTP를 이용한 적응형 비디오 스트리밍 기술이다[1]-[5]. DASH 표준화는 2010년 4월 드레스덴에서 개최된 제94차 MPEG 회의에서 기술제안요청서[2]를 공표하였고, 같은 해 7월 제네바에서 개최된 제95차 MPEG 회의에서 제안서들을 접수하였으며, 같은 해 10월 광저우에서 개최된 제96차 회의에서 CD(Committe Draft)가, 2011년 1월 대구에서 개최된 제97차 회의에서 DIS(Draft International Standard)[5]가 승인되었으며, 올해 7월 토리노에서 개최될 제99차 회의에서 FDIS(Final Draft International Standard)가 승인됨으로써 기술적 내용이 확정될 예정이다. DASH는 MPEG-B의 Part 6이며, 공식 표준 번호는 ISO/IEC 23001-6이다.

일반적으로 HTTP를 이용한 적응형 비디오 스트리밍에서 클라이언트가 사용할 수 있는 기능은 HTTP get을 이용하여 파일 전체를 다운로드하거나 또는 HTTP partial get을 이용하여 파일 일부(지정된 바이트 위치부터 일정 바이트 수)를 다운로드하는 것뿐이다. 따라서 클라이언트가 수신하여야 하는 전체 콘텐트를 작은 파일 조각 즉, 세그먼트(segment)로 분할하고 이들 세그먼트들을 순차적으로 다운로드함으로써 마치 스트리밍이 달성되는 것과 같은 효과를 낼 수 있다. 네트워크의 상태에 따라 전송률을 조절하기 위해서는 서버에 서로 다른 비트율로 부호화된 세그먼트들을 함께 보관해 두어 클라이언트로 하여금 선택할 수 있게 한다. 서버는 세그먼트들의 URL, 속성 등의 정보를 담은 메타데이터 파일인 매니페스트(manifest) 파일과 세그먼트들을 저장하고 있으며, 클라이언트는 먼저 메니페스트 파일을 다운로드하여 분석한 후 네트워크 상황에 맞게 세그먼트들을 순차적으로 다운로드한다. DASH에서 매니페스트 파일은 MPD(Media Presentation Description)라 불린다. 스트리밍 제어에 대한 모든 지능성은 클라이언트에 집중되어 있다. DASH에서는 MPEG-2 TS[6] 또는 MPEG-4 파일 포맷(MP4)[7]를 기반으로 하는 두 가지 형태의 서버 콘텐트를 제공할 수 있다.

MP4 기반 DASH의 경우, 클라이언트가 수신한 세그먼트들을 순차적으로 이어 붙여 저장한다 하더라도 특별한 조건을 만족하는 경우를 제외하고는 MP4 파일을 재생할 수 있는 기존 미디어 플레이어가 이를 재생하지 못한다. 본 논문에서는 이러한 특별한 조건을 제시하고 이를 실험을 통해 검증하였다. 이 조건은 서버에서 제공하는 세그먼트들에 대한 조건이다.

2. MP4 파일 포맷 기반 DASH 개요

DASH에서는 네트워크 상황에 따라 다른 비트율로 부호화된 미디어 압축 세그먼트로 전환하는 것을 비트스트림 스위치(bitstream

switch)라 부른다. 또한 어떤 콘텐트를 압축부호화한 데이터로부터 얻은 세그먼트 집합을 표현(representation)이라 하고, 비트스트림 스위치 가능한 여러 표현들의 집합을 표현 그룹(representation group)이라 한다.

MP4 파일 포맷은 크게 파일 내 데이터 구성 및 속성을 알려주는 메타데이터 부분('moov' 박스)과 실제 압축된 미디어 데이터를 저장하는 부분('mdat' 박스)으로 구성된다. MP4 파일 내용 중 'moov' 박스의 내용은 초기화 세그먼트(Initialization Segment, IS)로 구성하고, 'mdat' 박스의 내용을 여러 미디어 세그먼트(Media Segment, MS)로 분할한다. MP4 파일 포맷에서는 압축된 미디어 데이터를 저장하는 방법으로 두 가지를 제공한다. 하나는 'mdat' 박스 내에 비디오 압축 데이터를 한꺼번에 모두 먼저 넣고 이후에 이에 부수된 오디오 압축 데이터를 한꺼번에 넣도록 하는 방법이며, 다른 하나는 동기화되어 함께 재생하여야 하는 미디어들 즉, 비디오와 이에 부수된 오디오의 압축 데이터를 시간 순에 맞게 일부씩만 함께 담은 미디어 파편(media fragment)들로 나누어 순차적으로 여러 'moof' 박스들에 넣는 방법이다. 후자는 실시간으로 미디어를 저장할 때 유용한 포맷이다. DASH에서도 상기 후자에 해당하는 MP4 파일 포맷을 기반으로 'mdat' 박스의 내용물을 여러 세그먼트로 분할한다.

MP4 파일 포맷 기반 DASH를 지원하는 DASH 클라이언트는 어떤 콘텐트를 스트리밍하고자 할 때, 가장 먼저 MPD를 다운로드 받아 IS와 MS들의 URL, 표현 그룹의 수 등을 파악하고, IS를 다운로드 받아 분석함으로써 미디어 구성요소들의 부호화 조건을 알아낸다. 이후 미디어 세그먼트 파일들을 순차적으로 다운로드 받아 미디어를 재생하는데, 이 때 네트워크 상태를 조사하여 필요하면 동일한 표현 그룹 내의 다른 세그먼트를 다운로드하는 비트스트림 스위칭을 시행한다. MPD에 동일 표현 그룹 내의 다른 세그먼트로 비트스트림 스위칭할 수 있는지 여부를 나타내는 플래그가 있는데, 이는 아래 3장의 조건을 만족할 때, 'true'로 설정된다. 이를 bitStreamSwitchingFlag라 부른다.

bitStreamSwitchingFlag가 'true'로 설정된 경우, 사용자가 추후 기존 미디어 플레이어(예를 들어 애플사의 퀵타임 플레이어)에서 재생하기 위해 수신된 내용을 저장하기를 원하면 이를 저장하여 사용할 수 있다.

3. 연접 저장 결과 재생을 위한 DASH 세그먼트 조건

아래에서 별도의 언급이 없는 박스 이름, 필드 명칭 등에 대해서는 ISO 기본 파일 포맷 표준[8]을 참조하기 바란다.

DASH 클라이언트가 수신한 세그먼트들을 IS부터 마지막 MS까지 연접하여 저장한 결과 파일이 MP4 파일을 재생할 수 있는 기존 미디어 플레이어에 의해 재생될 수 있는 조건은 다음과 같다.

(조건 1) 우선 모든 MS는 DASH DIS[5]의 segmentAlignmentFlag가 'true'로 설정될 조건을 만족하여야 한다. 즉 모든 표현에서 모든 미디어 구성요소의 세그먼트 시작 시간과 종료 시간이 같아야 한다.

(조건 2) 동일한 표현 그룹 내의 모든 표현은 동일한 IS를 가져야 한다.

(조건 3) IS는 해당 표현 그룹 내의 모든 표현을 복호하는 데 필요한 샘플 설명 정보를 포함하여야 한다. 즉 어떤 미디어 구성요소가 서로 다른 비트율로 압축된 여러 표현이 있다면, 각 표현에 대한 모든 부호화 정보를 해당 미디어 구성 요소의 트랙(track) 박스에 넣어야 한다. 이러한 정보는 트랙 박스 내의 'stsd' 박스에 들어가는데, 이 박스 내의 각 'sampleEntry'는 각 표현의 부호화 정보에 해당하며, 'stsd' 박스 내의 'entry_count'는 해당 미디어 구성요소에 대한 서로 다른 표현의 수를 나타낸다. 동일한 미디어 구성요소에 대해 여러 가지 표현이 존재하지만 모든 표현에 대한 부호화 정보가 하나의 트랙 박스에 포함되므로, 모든 표현에 있어 해당 미디어 구성요소를 지칭하기 위해서는 동일한 'track_ID'를 사용하여야 한다.

(조건 4) 동일한 표현 그룹에 속하는 MS 내의 모든 트랙 파편에 대해서는 'moof' 박스 내의 'traf' 박스 내의 'tfhd' 박스 내의 'track_ID' 값을 IS 내의 해당 미디어 구성요소 트랙에 해당하는 트랙 박스의 'track_ID' 값과 동일하게 설정하여야 한다.

(조건 5) 어떤 미디어 구성요소의 트랙 파편 내의 'tfhd' 박스 내의 'sample_description_index' 값은 해당 미디어 구성요소 트랙의 'stsd' 박스 내의 인덱스 중 이에 대응되는 'sampleEntry'의 인덱스로 설정하여야 한다.

(조건 6) 'moof' 박스 내에서는 상대적인 주소를 사용하여야 한다. 절대적인 바이트 오프셋을 사용하면 안 된다. 더 자세한 사항은 ISO 기본 파일 포맷 표준[8]의 8.8.4.-8.8.8.의 내용을 참조하기 바란다.

4. 연접 저장 결과 재생 조건을 만족하는 예

하나의 표현 그룹 내에 '표현 1'과 '표현 2'가 있다고 하자. '표현 1'과 '표현 2'의 비디오는 각기 500 kbps 및 100 kbps로 압축되어 있고, 오디오는 두 표현 모두 동일하게 96 kbps로 압축되어 있다고 하자. 예제 콘텐트의 재생 시간은 전체적으로 60 초라 하자. 한 MS 파일의 재생 시간은 5 초로 하여 모두 12 개의 MS로 전체 압축 데이터 콘텐트를 분할하였다고 하자. 또 한 MS는 10 개(하나의 파편이 비디오와 오디오를 모두 포함하는 경우) 또는 20 개(하나의 파편에 오디오 또는 비디오의 단일 미디어 구성요소만 포함시킨 경우)의 파편으로 분할하였다고 하자. 따라서 한 파편의 재생 시간은 0.5 초이다. 파편의 시작 샘플(접근 단위, access unit)은 RAP(random access point)이다. 비트스트림 스위칭은 그림 1과 같이 60 초 구간 내에서 15 초, 30 초 45 초 시점에서 시행된다고 하자.

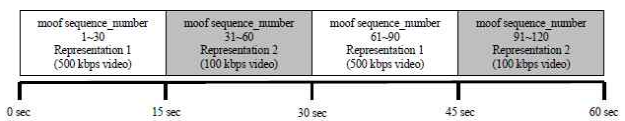

그림 1 DASH 비트스트림 스위칭의 예

그림 2~그림 5에 보인 예에서 MS 내에 'moof' 박스의 실제 저장 순서는 moof_sequence_number(일부 그림에서 seq로 줄여 표기)에 따른다. 또 그림 2(b)에서는 'moof' 박스 아래에 실제 압축 데이터가 저장

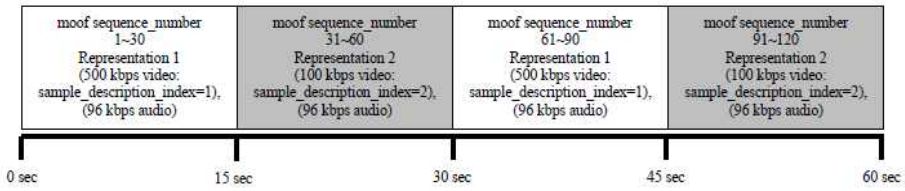

그림 2(a) 예1의 시간 순서도

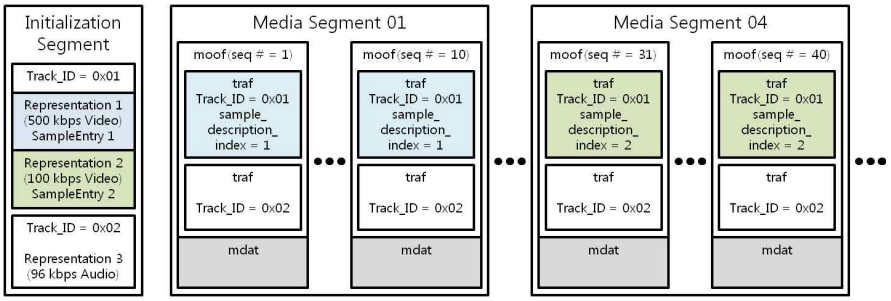

그림 2(b) 예1의 세그먼트들

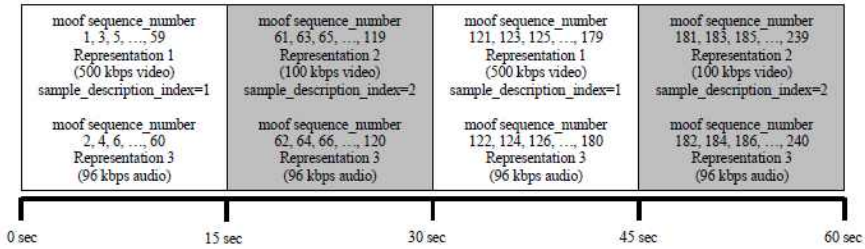

그림 3(a) 예2의 시간 순서도

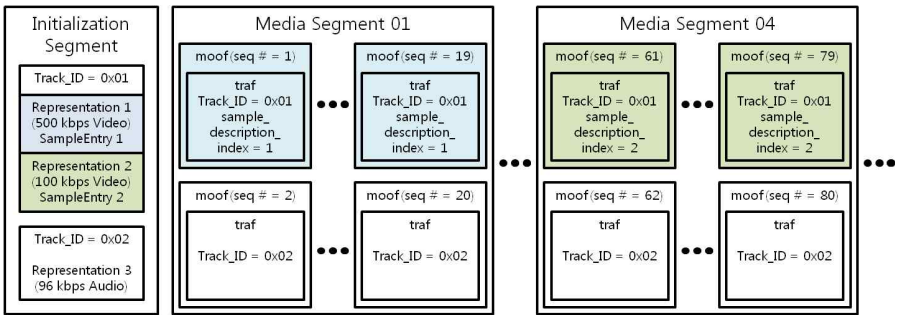

그림 3(b) 예2의 세그먼트들

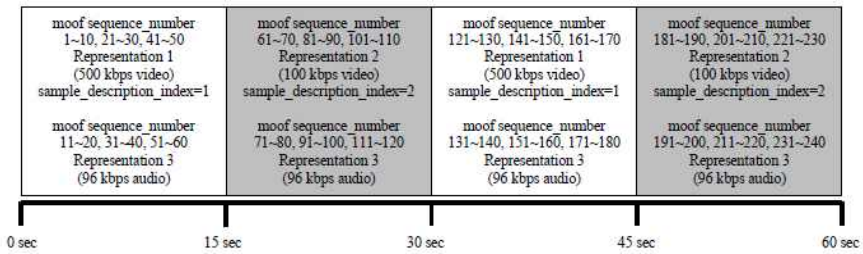

그림 4(a) 예3의 시간 순서도

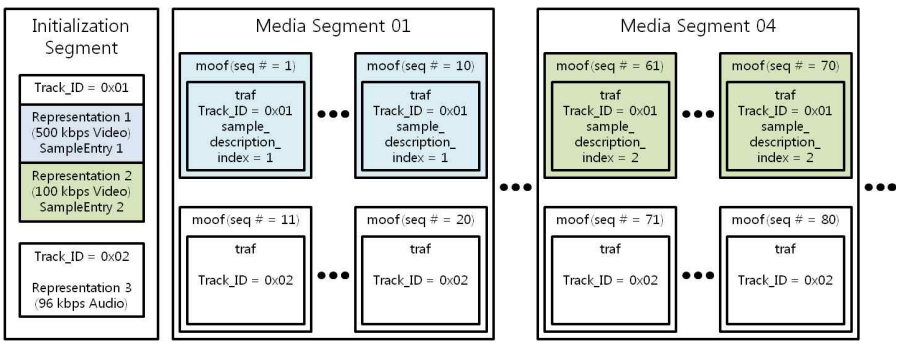

그림 4(b) 예3의 세그먼트들

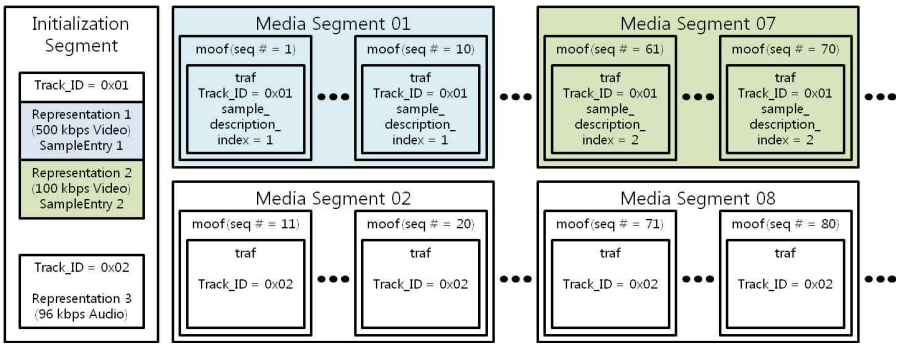

그림 5 예4의 세그먼트들

되는 'mdat' 박스를 표기하였으나, 그림 3(b), 그림 4(b), 그림 5에서는 편의상 이를 생략하였다.

그림 2에 보인 예1에서는 하나의 'moof' 박스 내에 비디오 트랙 파편과 오디오 트랙 파편을 모두 포함한다.

그림 3에 보인 예2에서는 하나의 'moof' 박스 내에 비디오 또는 오디오 중 한 가지만 포함하되 한 MS 내에서는 비디오 파편과 오디오 파편이 번갈아 등장한다.

예3은 예2와 유사하나, 한 MS 내에서는 비디오 파편과 오디오 파편이 번갈아 등장하지 않고 모든 비디오 파편들이 먼저 저장된 후 오디오 파편들이 저장된다.

예4에 대한 시간 순서도는 예3과 동일하며, 예4는 예3과 유사하나 예3에서의 MS를 비디오 MS와 오디오 MS의 두 개의 MS로 분리한 것이다. 오디오를 별도의 MS로 저장함으로써, 서로 다른 표현에 대해서 동일한 오디오를 사용하므로 DASH 서버의 저장 용량을 줄일 수 있다.

5. 실험 결과

4절에 보인 예 모두에 대해 검증 실험을 시행하였다. 실제 전송은 하지 않았으며, IS 파일과 두 표현에 대한 MS 파일들을 작성한 후, 비트스트림 스위칭 시점에서 각 표현에 해당하는 MS로 바꿔가면서 세그먼트 파일들을 연접하여 하나의 파일을 만든 후, 이를 기존 미디어 플레이어로 재생하였다.

실험에 사용된 운영체제는 마이크로소프트 윈도우즈 7이고, 프로그램은 마이크로소프트 비주얼 C++ 6.0으로 개발하였다. 세그먼트 파일을 만들기 위해서 MP4 파일 포맷 참조 소프트웨어인 libisomedia.dll을 활용하였다. 비디오 압축 방식은 H.264/AVC를, 오디오 압축방식은 AAC를 사용하였다. 기존 미디어 플레이어로는 퀵타임 플레이어(QuickTime 7.6.8)을 사용하였다.

실험 결과, 4절에 보인 예에 해당하는 모든 연접 파일이 기존 미디어 플레이어로 성공적으로 재생되었다.

5. 결론 및 향후 예상

본 논문에서는 DASH 클라이언트가 수신한 세그먼트들을 IS부터 마지막 MS까지 연접하여 저장한 결과 파일이 MP4 파일을 재생할 수 있는 기존 미디어 플레이어에 의해 재생될 수 있는 조건을 제시하고 실험을 통해 이를 검증하였다. 본 논문에서 제시한 조건 및 실험 결과는 2011년 1월 대구에서 개최된 제97차 MPEG 회의에 기고하였다.[10] 회의 결과, 본 논문의 내용은 MPEG-B의 일부로서 작성될 기술보고서(TR, Technical Report)인 DASH 구현 가이드라인에 포함시키기로 결정되었다.

참고문헌

[1] ISO/IEC JTC1/SC29/WG11 N11337, "HTTP Streaming of MPEG Media Context and Objectives," April 2010, Dresden, Germany.

[2] ISO/IEC JTC1/SC29/WG11 N11338, "Call for Proposals on HTTP Streaming of MPEG Media," April 2010, Dresden, Germany.

[3] ISO/IEC JTC1/SC29/WG11 N11339, "Use Cases for HTTP Streaming of MPEG Media," April 2010, Dresden, Germany.

[4] ISO/IEC JTC1/SC29/WG11 N11340, "Requirements on HTTP Streaming of MPEG Media," April 2010, Dresden, Germany.

[5] ISO/IEC JTC1/SC29/WG11 N11749, "Text of ISO/IEC DIS 23001-6 Dynamic Adaptive Streaming over HTTP," January 2011, Daegu, Korea.

[6] ISO/IEC 13818-1:2007, Information technology - Generic coding of moving pictures and associated audio information: Systems, 3rd Edition.

[7] ISO/IEC 14496-14:2003, Information technology - Coding of audio-visual objects - Part 14: MP4 file format.

[8] ISO/IEC 14496-12:3rd Edition, Information technology - Coding of audio-visual objects - Part 12: ISO base media file format.

[9] ISO/IEC 14496-5:2001/Amd.12:2007 Reference Software.

[10] ISO/IEC JTC1/SC29/WG11 MPEG2011/M19360, "Proposals for the change of semantics of bitStreamSwitchingFlag and a set of conditions for DFC by simple concatenation of DASH segments," January 2011, Daegu, Korea.