# 학술대회 중계용 MPEG-DASH 기반 HTTP 적응적 스트리밍 서비스

정은영, 김남태, 서봉석, 유동호, 김동호
서울과학기술대학교
{jeunyoung, rlaskaxoek, sbs91, youdongho, dongho.kim }@seoultech.ac.kr

# HTTP adaptive streaming service based on MPEG-DASH for conference.

Jeong Eunyoung, Namtae Kim, Bong-seok Seo, Dongho You, Dong Ho Kim
Seoul National University of Science and Technology

## 요약

학술대회는 동 시간대에 각기 다른 주제의 여러 세션이 운영되기 때문에 시간적으로 선택의 제약이 존재한다. 따라서 일반적으로 학회 참가자는 선택적으로 세션을 청취해야한다. 본 논문에서는 이러한 제약을 해결할 수 있는 학술대회 중계용 MPEG-DASH 기반의 HTTP 적응 스트리밍 서비스를 구현하였고, 그 결과를 보여준다.

## Ⅰ. 서론

학술대회는 동 시간대에 여러 세션이 운영된다. 따라서 학회 참가자들은 선택적으로 세션을 청취해야하기 때문에 선택의 제약이 존재한다. 또한 세션장의 크기가 제한되어있기 때문에 자리가 부족하거나, 발표자로부터 먼 자리의 경우 발표자와 소통이 어렵다. 이러한 제약을 극복하기 위해 대개 학회에서는 이동형 저장장치에 논문을 저장하여 학회 참가자에게 전달한다. 하지만 논문만 보는 것은 실제 발표자의 설명을 듣는 것과 비교하여 저자의 의도를 온전히 파악하기 어렵다. 이에 본 논문에서는 이러한 제약을 해결할 수 있는 학술대회 중계용 MPEG-DASH (Dynamic Adaptive Streaming on HTTP) 기반의 HTTP 적응 스트리밍 서비스를 제안하고, 그 결과를 보여준다. 이 서비스를 통해 위에서 언급한 학술대회의 제약을 해결할 수 있다.

본론에서 MPEG-DASH 에 관련한 연구를 하고, 논문에서 제안하는 서비스의 구현과정에 대해 설명한다. 결론에서 본 논문에서 제안한 학술대회 중계용 MPEG-DASH 기반 HTTP 적응 스트리밍 서비스의 결과를 보인다.

## Ⅱ. 본론

### 가. MPEG-DASH

멀티미디어 전송 방식은 크게 다운로드 방식과 스트리밍 방식으로 구분된다. 다운로드 방식은 재생 전에 데이터를 완전히 다운로드 받는 방식으로 데이터 다운로드가 완료되기 전까진 재생이 불가능한 방식이다. 또 다른 방식은 스트리밍 방식이다. 기존 스트리밍 방식은 일반적으로 RTSP (Real-Time Streaming Protocol) 과 같은 정적인 프로토콜을 사용하였다. 이는 서버와 클라이언트 간 접속이 이루어지면 세션을 생성한다. 생성된 세션은 클라이언트의 제생 상태를 유지하는데 사용하며 세션이 살아있는 동안 서버는 작은 패킷 단위로 미디어 데이터를 전송한다 [1]. 하지만 이는 세션의 관리와 흐름 제어로 인한 개발의 복잡도가 증가하고, 서버의 부하가 발생하며 방화벽이나 라우터에서 막히는 경우가 많다는 단점이 있다 [2].

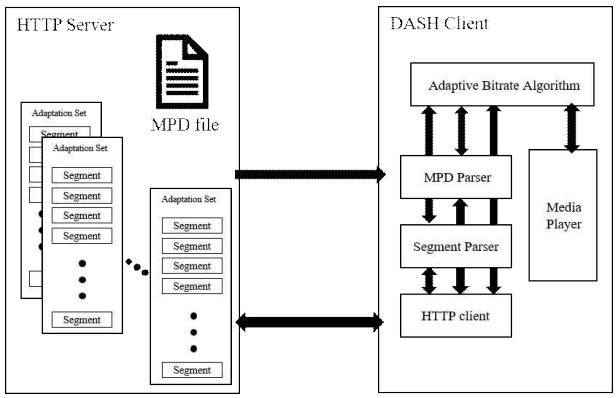



그림 1. MPEG-DASH 동작 흐름도

이러한 단점을 해결하기 위하여 HTTP 적응 스트리밍 기술이 개발되었다. 이는 표준 HTTP를 이용하여 클라이언트의 상황에 적응할 수 있도록 콘텐츠를 초 단위의 세그먼트로 쪼개어 전송하는 기술이다. 클라이언트의 네트워크 상황이 좋은 경우에는 고화질로 전송한다. 모든 세그먼트를 전송할 때까지 계속해서 클라이언트의 네트워크 상황을 관측하여 전송 도중 네트워크나 단말의 환경이 열악하면 저화질로 영상을 전송하여 버퍼링이 필요하지 않다 [3]. HTTP 적응 스트리밍 기술에는 대표적으로 Apple의 HLS (HTTP Live Streaming). Microsoft의 Smooth Streaming, Adobe의 HTTP Daynamic Streaming 이 있다. 이에 MPEG (Moving Picture Experts Group)은 DASH (Dynamic Adaptive Streaming on HTTP) 라는 그룹을 만들어 HTTP 적응 스트리밍과 관련된 패키지 포맷에 대한 표준화를 진행하였고, 2011년 11월 국제 표준이 되었다.[4]

MPEG-DASH의 기본 동작은 그림 1 과 같다. 서버에 비디오 스트림을 각기 다른 비트율의 여러 버전(Adaptation Set)으로 인코딩 한

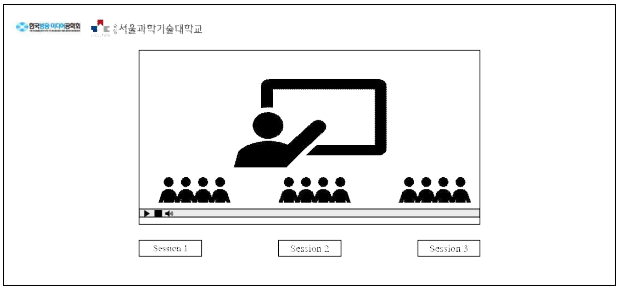



그림 2. 제안하는 서비스의 구성도

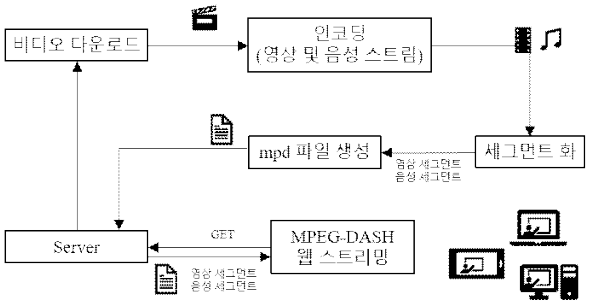



그림 3. 제안하는 서비스의 블록도

뒤 초 단위의 세그먼트로 분할하여 저장한다.

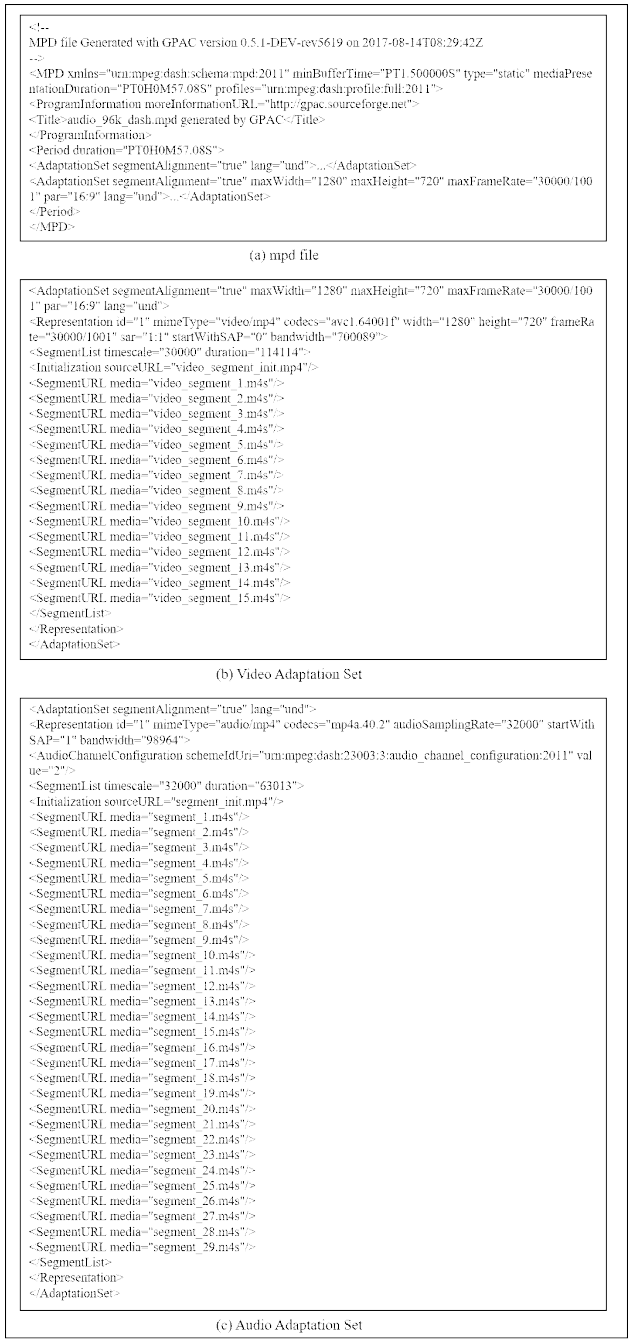
서버 내 저장된 비디오 스트림의 정보를 나타내는 XML 문서인 MPD (Media Presentation Description)을 저장한다. 클라이언트는 MPD 파일을 요청하고 받아와 그 내용을 읽어(Parsing) 원하는 세그먼트를 요청하여 받아온다. 그 후 자신의 상황에 따라 비트율 적응 알고리즘을 수행하여 네트워크 가용 대역폭을 계산하여 다음 세그먼트의 비트율을 결정하고 요청한다. 클라이언트의 비디오 플레이어는 현재 단말 네트워크의 상태와 QoE (Quality of Experience)를 고려하여 적절한 비디오 화질을 선택하여 끊김 없는 서비스를 제공한다 [5].

### 나. 제안하는 서비스 구현 과정

서비스의 구현은 Video player를 포함한 html 페이지 제작, 영상 및 음성 스트림 인코딩, 세그먼트 화, MPD 파일 작성의 네 단계로 이루어진다.

본 논문에서 제안하는 서비스의 html 페이지는 그림 2 와 같이 구성한다. 세션 1, 2, 3 버튼을 제작하고 클릭 이벤트가 발생하면 해당 세션의 비디오 및 오디오를 HTTP 적응적 스트리밍 재생 한다. 본 논문에서 제안하는 서비스의 블록도는 그림 3과 같다.

서버 내 원본 비디오를 다운로드 받아 ffmpeg 프로그램을 이용하여 영상과 음성 스트림을 각각 인코딩한다. ffmpeg은 디지털 영상 및 음성 스트림을 인코딩하는 프로그램으로, 명령어를 입력하여 원하는 종류와 형태로 인코딩한다. 인코딩 된 영상 및 음성 스트림은 MP4Box 프로그램을 통해 초 단위로 세그먼트 화 한다. MP4Box는 GPAC에서 개발한 오픈 소스 멀티미디어 패키징 도구로서 DASH 세그먼트 MP4와 관련 MPD 파일을 만들 수 있는 프로그램이다. 본 논문에서는 4초의 세그먼트로 영상과 음성 스트림을 세그먼트 화 하였다. 다음으로 세그먼트 화 된 영상과 음성 스트림의 MPD 파일을 각각 생성하고, 음성



```
<!--
MPD file Generated with GPAC version 0.5.1-DEV-rev5619 on 2017-08-14T08:29:42Z
-->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT0H0M57.08S" profiles="urn:mpeg:dash:profile:full:2011">
 <ProgramInformation moreInformationURL="http://gpac.sourceforge.net">
  <Title>audio_96k_dash.mpd generated by GPAC</Title>
 </ProgramInformation>
 <Period duration="PT0H0M57.08S">
  <AdaptationSet segmentAlignment="true" lang="und">...</AdaptationSet>
  <AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720" maxFrameRate="30000/1001" par="16:9" lang="und">...</AdaptationSet>
 </Period>
</MPD>
```

(a) mpd file

```
<AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720" maxFrameRate="30000/1001" par="16:9" lang="und">
 <Representation id="1" mimeType="video/mp4" codecs="avc1.64001f" width="1280" height="720" frameRate="30000/1001" sar="1:1" startWithSAP="0" bandwidth="700089">
  <SegmentList timescale="30000" duration="114114">
   <Initialization sourceURL="video_segment_init.mp4"/>
   <SegmentURL media="video_segment_1.m4s"/>
   <SegmentURL media="video_segment_2.m4s"/>
   <SegmentURL media="video_segment_3.m4s"/>
   <SegmentURL media="video_segment_4.m4s"/>
   <SegmentURL media="video_segment_5.m4s"/>
   <SegmentURL media="video_segment_6.m4s"/>
   <SegmentURL media="video_segment_7.m4s"/>
   <SegmentURL media="video_segment_8.m4s"/>
   <SegmentURL media="video_segment_9.m4s"/>
   <SegmentURL media="video_segment_10.m4s"/>
   <SegmentURL media="video_segment_11.m4s"/>
   <SegmentURL media="video_segment_12.m4s"/>
   <SegmentURL media="video_segment_13.m4s"/>
   <SegmentURL media="video_segment_14.m4s"/>
   <SegmentURL media="video_segment_15.m4s"/>
  </SegmentList>
 </Representation>
</AdaptationSet>
```

(b) Video Adaptation Set

```
<AdaptationSet segmentAlignment="true" lang="und">
 <Representation id="1" mimeType="audio/mp4" codecs="mp4a.40.2" audioSamplingRate="32000" startWithSAP="1" bandwidth="98964">
  <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
  <SegmentList timescale="32000" duration="63013">
   <Initialization sourceURL="segment_init.mp4"/>
   <SegmentURL media="segment_1.m4s"/>
   <SegmentURL media="segment_2.m4s"/>
   <SegmentURL media="segment_3.m4s"/>
   <SegmentURL media="segment_4.m4s"/>
   <SegmentURL media="segment_5.m4s"/>
   <SegmentURL media="segment_6.m4s"/>
   <SegmentURL media="segment_7.m4s"/>
   <SegmentURL media="segment_8.m4s"/>
   <SegmentURL media="segment_9.m4s"/>
   <SegmentURL media="segment_10.m4s"/>
   <SegmentURL media="segment_11.m4s"/>
   <SegmentURL media="segment_12.m4s"/>
   <SegmentURL media="segment_13.m4s"/>
   <SegmentURL media="segment_14.m4s"/>
   <SegmentURL media="segment_15.m4s"/>
   <SegmentURL media="segment_16.m4s"/>
   <SegmentURL media="segment_17.m4s"/>
   <SegmentURL media="segment_18.m4s"/>
   <SegmentURL media="segment_19.m4s"/>
   <SegmentURL media="segment_20.m4s"/>
   <SegmentURL media="segment_21.m4s"/>
   <SegmentURL media="segment_22.m4s"/>
   <SegmentURL media="segment_23.m4s"/>
   <SegmentURL media="segment_24.m4s"/>
   <SegmentURL media="segment_25.m4s"/>
   <SegmentURL media="segment_26.m4s"/>
   <SegmentURL media="segment_27.m4s"/>
   <SegmentURL media="segment_28.m4s"/>
   <SegmentURL media="segment_29.m4s"/>
  </SegmentList>
 </Representation>
</AdaptationSet>
```

(c) Audio Adaptation Set

그림 4. 생성한 영상 및 음성의 mpd 파일

스트림의 MPD 파일 내 Adaptation Set을 가져와 영상 스트림의 MPD 파일 내 Adaptation Set 으로 추가하여 하나의 MPD 파일로 생성한다. 본 논문에서 생성한 MPD 파일은 그림 4 와 같다.

### 다. 제안하는 서비스 구현 결과

해당 서비스의 구현 동작은 로컬호스트에서 확인하였다. 제작한 html 페이지는 그림 5 와 같다. session 1, 2, 3 버튼을 생성하고 클릭하면 해당 세션장의 영상을 MPEG-DASH로 스트리밍 재생한다.

실험 결과는 네트워크 패킷 분석에 유용한 WinPcap 패킷 캡쳐 라이브러리 기반의 패킷 분석 프로그램인 와이어샤크(wireshark)를 이용

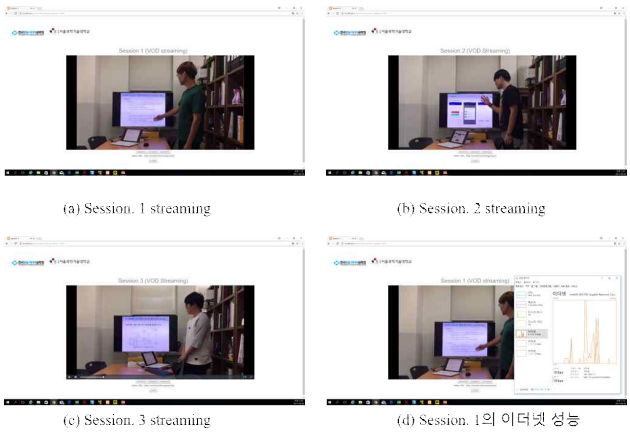



그림 5. 학술대회 중계용 MPEG-DASH 기반 HTTP 스트리밍 서비스 구현 html 페이지

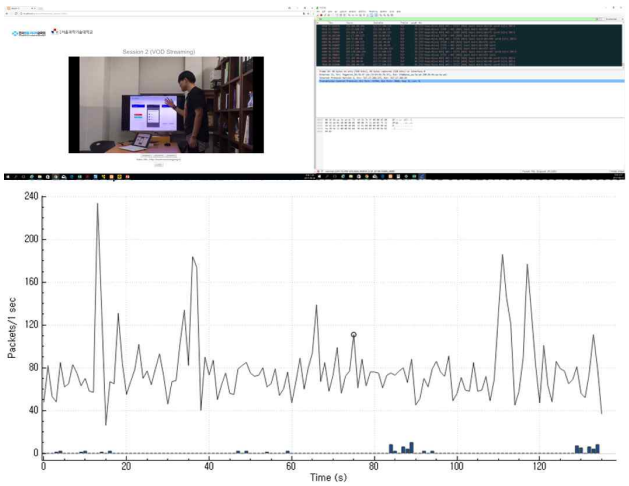



그림 6. 제안한 서비스의 패킷 이동유무 확인 결과

하였다. 와이어샤크를 통해 본 논문에서 제안한 학술대회 중계용 MPEG-DASH 기반 HTTP 적응 스트리밍 서비스의 패킷 이동 유무를 확인하였고, 그 결과는 그림 6 과 같다. 그래프의 x축은 시간, y축은 초당 이동한 패킷의 수를 의미하며 그래프를 통해 매 초마다 패킷이 이동하는 것을 확인하였다.

## Ⅲ. 결론

본 논문에서는 학술대회 중계용 MPEG-DASH 기반 HTTP 적응 스트리밍 서비스를 제안하고 구현하였다. 이는 동 시간대 진행되는 학회 세션의 선택 제약을 해결할 수 있다. 학회에 국한되지 않고, 동 시간대에 여러 세션장이 동시에 운영되는 특성을 가진 프로그램에 응용 될 수 있는 서비스이다.

본 논문에서는 VoD(Video on Demand) 스트리밍의 동작을 로컬호스트에서 확인하였다. 추후 실시간 스트리밍을 구현할 계획에 있다. 더 나아가 발표자와의 실시간 채팅 서비스를 구현하여 질의응답이 가능하게 하는 서비스 또한 구현할 계획에 있다.

## 감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 [2016-0-00099, 제작 편리성과 실감 시청 체험 극대화를 위한 개인방송 제작 기술 개발]

## 참고논문

[1] Sorokopud, Gennady, A. Satt, and L. Langer. *"Real Time Streaming Protocol (RTSP) proxy system and method for its use."* U.S. Patent Application, 2004.

[2] 정훈영, 박지우, 정광수. "무선 DASH 환경에서 QoE 향상을 위한 대역폭 측정 기반의 비디오 품질 조절 기법." 한국정보과학회 학술발표논문집, 2015.

[3] Oyman, Ozgur, and S. Singh. *"Quality of experience for HTTP adaptive streaming services."* IEEE Communications Magazine, 2012.

[4] Sodagar, Iraj. *"The mpeg-dash standard for multimedia streaming over the internet."* IEEE MultiMedia, 2011.

[5] Stockhammer, Thomas. *"Dynamic adaptive streaming over HTTP--: standards and design principles."* Proceedings of the second annual ACM conference on Multimedia systems. ACM, 2011.