

---

# HTTP상에서 동적 적응적 스트리밍 시스템 구현

정선철\* · 반태학\*\* · 정회경\*\*\*

An Implementation of Dynamic and Adaptive Streaming System over HTTP

Sun-Chul Jeong\* · Tae-Hak Ban\*\* · Hoe-Kyung Jung\*\*\*

## 요 약

최근 QoS/QoE 기술의 일환으로 HTTP에 기초한 적응적 스트리밍 기술이 주목받고 있다. 이에 본 논문에서는, HTTP에 기초한 동적 적응적 스트리밍 기술에 대해 알아본다. 이를 기반으로 HTTP상에서 동적 적응적 스트리밍 시스템을 설계 및 구현 하였다. 이 시스템은 MPEG2-TS 파일의 비트율별 변환, Segment 분할, MPD(Media Presentation Description)의 생성과 서버와 클라이언트 간의 동적이고 적응적인 네트워크 환경에서 MPD File의 분석을 통한 비트율별 플레이어로 구성된다. 이는 다양한 네트워크 환경에서 지속적이고, 원활한 영상의 재생을 위해 다양한 멀티 미디어 분야에 활용될 것이다.

## ABSTRACT

Recently, QoS/QoE technology as a part of the HTTP-based adaptive streaming technology has attracted attention. In this paper, we find out HTTP-based adaptive and dynamic streaming technology. It is based on a dynamic adaptive streaming system over HTTP, which is designed and implemented. The system consist of conversion by the bit rate of MPEG2-TS files, division of segment, creation of MPD(Media Presentation Description) and bitrate's player through analysis of MPD File between servers and clients in network environments of dynamic and adaptive. This diverse network environments, continuous and smooth playback of video will be used in various multimedia fields.

## 키워드

DASH, HTTP 스트리밍, MPD, QoS

## Key word

DASH, HTTP Streaming, MPD, QoS

---

\* 정회원 : 배재대학교  
\*\* 준회원 : 배재대학교  
\*\*\* 종신회원 : 배재대학교 (교신저자, hkjung@pcu.ac.kr)

접수일자 : 2012. 01. 09  
심사완료일자 : 2012. 02. 15

I. 서 론

멀티미디어 콘텐츠들이 늘어나고 다양한 서비스가 멀티플랫폼 환경에서 제공되면서 서비스에 대한 QoS 문제가 대두되고 있다. 이러한 문제를 해결하기 위해 네트워크를 효율적으로 활용하고, 콘텐츠 품질을 향상하기 위한 시스템이 필요하게 되었다[1,2].

기존 HTTP상에서의 스트리밍 서비스는 서버에서 제공하는 한가지의 화질 영상만을 서비스 받아왔다. 또한, 스트리밍 방식은 RTSP(Real-Time Streaming Protocol) 등과 같은 전용 스트리밍 프로토콜을 사용하거나 전용회선을 이용하여 현 시점에서 필요한 영상을 미디어 서버로부터 받는 방식이었다. 그러나, 최근에는 YouTube, Video 등에서는 웹 표준 프로토콜인 HTTP를 이용하여 미디어 콘텐츠를 버퍼링하고 플레이하는 프로그레시브 다운로드 방식이 널리 이용되고 있다. 기존의 스트리밍 방식과는 달리 웹 표준 프로토콜을 이용한 전송방식은 HTTP를 지원하는 종래의 구조적 체계를 그대로 이용하여 언제 어디서나 인터넷을 이용하여 미디어 서비스에 접근이 가능하다는 점과 인터넷에 존재하는 캐시 또는 프락시 서버들을 이용하여 네트워크를 효율적으로 사용할 수 있다. 또한, RTP/RTSP와 같은 전송 프로토콜의 경우처럼 NAT/Firewall과 같은 이슈가 발생하지 않는다는 장점이 있기 때문에 오늘날 웹상의 비디오 전송 방식으로 널리 사용되고 있다. 그리고, 최근에는 HTTP 프로토콜의 장점과 필요한 영상을 바로 서버로부터 요청하고 서비스를 받아 플레이하는 스트리밍 방식의 장점을 살린 하이브리드 전송 방식으로 HTTP 적응적 스트리밍(HTTP Adaptive Streaming, HAS) 기술이 주목받고 있다 [1-5].

이에 본 논문에서는 HTTP 상에서 동적 적응적 스트리밍 시스템을 설계 및 구현하였다. HTTP상에서 동적 적응적 스트리밍이란 동영상을 플레이하여 재생할 때, 통신망이나 트래픽 등의 간섭을 고려하여 현재의 네트워크 상황에 맞는 비트율의 화질을 서비스하여 제공하는 것을 말한다. 따라서, 본 논문에서는 MPEG DASH(Dynamic Adaptive Streaming over HTTP)에 관련된 연구를 하였으며, basic on-demand profile에 기초한 DASH 서버 및 클라이언트 시스템을 설계 및 구현하였다.

II. 관련 연구

본 장에서는 DASH 관련 기술을 분석하였다.

2.1. DASH(Dynamic Adaptive Streaming over HTTP)

HTTP에 기초한 동적 적응적 스트리밍의 DASH는 DASH 호환 클라이언트에 표준 HTTP 서버에서 미디어 콘텐츠를 제공하여 활성화 하고 표준 HTTP 캐시의 콘텐츠를 캐싱 할 수 있도록 형식을 지정한다. 또한, DASH MPD(media presentation description)는 정의된 자원의 스키마에 지정된 프로토콜을 통해 세그먼트(segment)에 액세스하여 사용자에게 스트리밍 서비스를 제공하기 위한 것이다. DASH 형식의 MPD에 액세스 할 수 있도록 클라이언트에 충분한 정보를 제공한다[6,7].

그림 1은 DASH에서 서버와 클라이언트의 동작과정 부분을 설명한다. 서버에서는 세그먼트 파일들이 저장되어 있고, 클라이언트에서는 MPD 파일에 정의되어 있는 정보를 분석하여 HTTP를 이용해 서버에 저장되어 있는 세그먼트를 전달받는 과정이다.

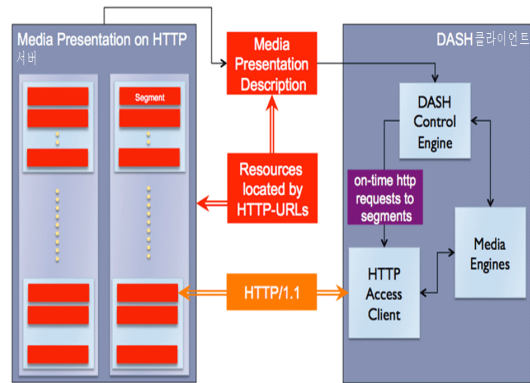


그림 1. DASH 흐름도  
Fig. 1 Diagram of DASH

2.2. MPD(Media Presentation Description)

MPD는 DASH에서 실질적으로 서버와 클라이언트에서 전송과 수신을 위한 매개체이다. MPD 문서의 전체적인 구조는 MPD, 피리어드(Period), 리프리젠테이션(Representation), 세그먼트 등의 4개 항목으로 구분되어진다. 각 항목에 대해서 아래에 자세히 기술한다.

MPD는 DASH 클라이언트에서 액세스하는 데이터 구조이다. DASH 클라이언트 요청과 사용자에게 스트리밍 서비스를 제공하기 위해 MPD에서 검증된 다운로드 정보를 제공한다. MPD에서는 가능한 모든 업데이트 정보가 포함되어야 한다.

### 2.3. 세그먼트(Segment)

스펙에서 세그먼트는 크게 MPEG2-TS와 ISOBMFF를 지원하며 다음과 같은 미디어 세그먼트 특징을 가진다.

- MPD 문서에 URL 요소를 할당
- MPD에서 제공하는 문서 표현의 시작에 상대적으로 시작 시간을 할당하며, 클라이언트는 정기적인 플레이 아웃 모드 또는 추후 적절한 세그먼트를 다운로드할 수 있다.
- 하나 이상의 랜덤 액세스 포인트를 포함
- 임의 HTTP로 일부분을 요청할 때, GET을 사용하여 세그먼트의 sub segments에 액세스 가능
- Discription의 첫 번째 경우에는 문서의 시작 시간에 상대적으로 MPD 타임 라인에서 프레젠테이션 시간이 0을 할당
- ISOBMFF와 같은 하나 이상의 동영상 조각이 포함 되어야 한다.

세그먼트 생성과정은 그림 2와 같이 원본 파일을 서버에서 인코딩 과정을 거쳐 여러 종류의 비트 스트림 파일을 생성하고, 이 파일 등이 세그먼트를 거쳐 세그먼트 파일들이 만들어진다. 만들어진 세그먼트 파일들은 서버에 저장되고 이에 기초한 MPD 문서를 함께 생성한다. 클라이언트에서 요청이 들어오면 HTTP를 통해 전달된다.

## III. DASH 시스템 설계

본 장에서는 DASH 시스템을 클라이언트와 서버로 구분하여 설계하였다. 서버는 기존 HTTP 서버를 이용하지만 이 서버에 입력되는 멀티미디어 콘텐츠를 DASH 형식으로 전환하여 올리는 작업을 DASH 서버라 한다. 클라이언트는 웹 브라우저가 아닌 별도의 프로그램으

로 MPD 문서를 분석하고 적용적으로 미디어 데이터를 요청하여 이를 재생하는 것을 말한다. 그림 3에 DASH 서버와 클라이언트 흐름을 보인다.

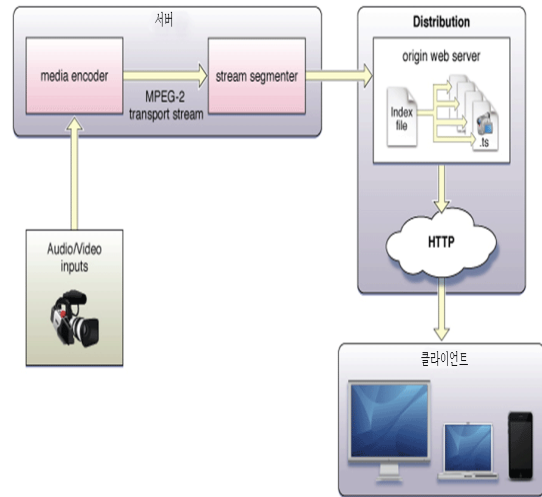


그림 2. 세그먼트 생성 과정  
Fig. 2 Process of segment generation

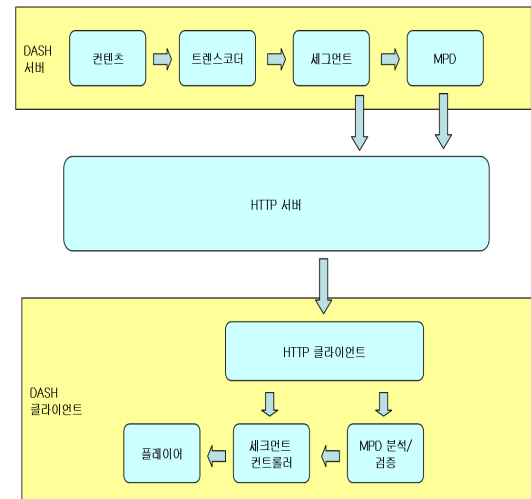


그림 3. DASH 서버와 클라이언트 구성도  
Fig. 3 Daigram of DASH server and client

### 3.1 DASH 서버 시스템 설계

이는 비트 스트림 생성 단계, 세그먼트 생성 단계,

MPD 문서 생성 단계로 구분된다. 비트 스트림 생성 단계에서는 리프리젠테이션 레벨에서 필요한 코덱 정보 및 비트 레이트, 해상도, 프레임레이트 등의 정보를 입력 받아 비트 스트림을 생성한다. 세그먼트 생성 단계에서는 세그먼트 레벨에서 사용되는 세그먼트 시간 간격 및 저장 장소 URL을 제공 받아 일정한 시간 간격으로 미디어 데이터를 세분화 하고, 재생에 필요한 초기화 세그먼트와 미디어 세그먼트를 생성한다. MPD 문서 생성 단계에서는 위의 내용을 기반으로 DASH 스키마에 맞추어 문서를 생성하고 이를 검증하여 HTTP 서버에 저장한다. DASH 서버의 흐름도를 그림 4에 보인다.

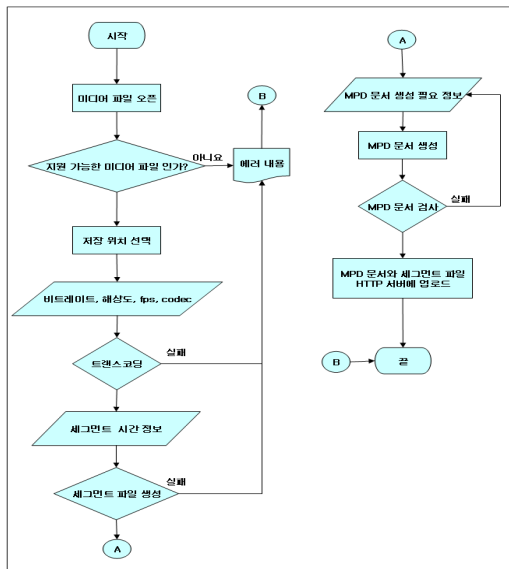


그림 4. DASH 서버 흐름도  
Fig. 4 Flowchart of DASH server

### 3.2. DASH 클라이언트 시스템 설계

이는 3단계로 구성되는데 첫째, 메타데이터 분석 단계로 MPD 문서가 올바른 것인지 검사하고, 클라이언트에서 지원 가능한 MPD 문서인지 판별한다. 이때, DASH 서버에 존재하는 세그먼트의 구조 및 위치, 그리고 서비스 형태 등의 정보를 수집한다. 둘째로 세그먼트 컨트롤러에서는 네트워크 상태의 분석을 통해 망의 상태에 가장 적당한 비트율의 세그먼트에 대해 HTTP-URL을 이용하여 요청한다. 마지막으로, 전달받은 세그먼트들이 끊김 없이 플레이 될 수 있도록 시간

정보의 체크 및 세그먼트 순서 등을 조정하여 미디어 플레이어에게 재생한다. DASH 클라이언트 흐름도를 그림 5에 보인다.

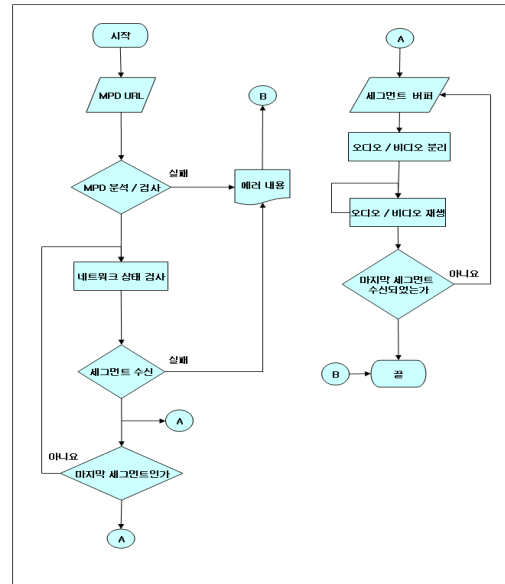


그림 5. DASH 클라이언트 흐름도  
Fig. 5 Flowchart of DASH client

## IV. DASH 시스템 구현

### 4.1. 구현환경

구현 내용은 VoD 콘텐츠를 기반으로 MPEG2-TS 파일과 MP4 파일의 두 종류의 비디오 포맷 파일 중에서 관리자가 원하는 종류의 비디오 포맷 사용이 가능하다. 비디오 코덱은 H.264를 사용하였으며, 오디오 코덱은 MPEG-4 ACC를 이용하였다. 비디오 리프리젠테이션은 3개의 리프리젠테이션으로 구성하였는데, 첫 번째 부분은 128, 256, 512를 이용하였고, 두 번째는 512, 1024, 2048을 이용, 세 번째는 1024, 2048, 4096을 이용하였다. 세 부분의 리프리젠테이션 단위는 “kbps”이다. 콘텐츠에 대해 일정 시간 단위로 분할하는 세그먼트의 단위는 10초로 설정하였다.

DASH 구현에서 서버의 구성은 일대일 구성과 다중 게이트웨이로 구성된다. 한 대의 서버를 설정하여 클

라이언트는 한 대의 서버를 통해서 세그먼트 데이터를 전송받을 수 있고, 여러 대의 분산되어 있는 서버를 통해서도 세그먼트 데이터를 전송받을 수 있는 것이 가능하다.

#### 4.2. DASH 서버 구현

서버 구현 화면 중 트랜스코딩 실행 화면과 세그먼트 실행 화면을 그림 6과 7에 보인다.

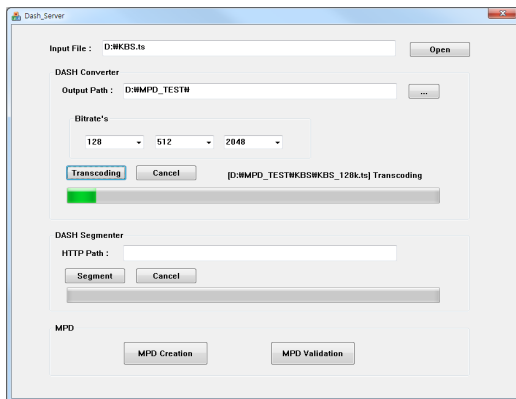


그림 6. 트랜스코딩 실행 화면  
Fig. 6 Screen of transcoding execution

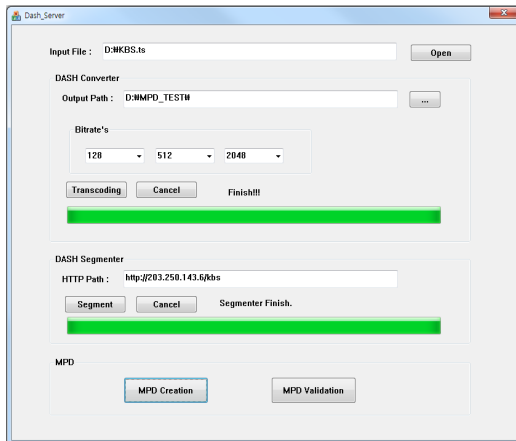


그림 7. 세그먼트 실행 화면  
Fig. 7 Screen of segment execution

#### 4.3. DASH 클라이언트 구현

클라이언트 재생 화면을 그림 8에 보인다.

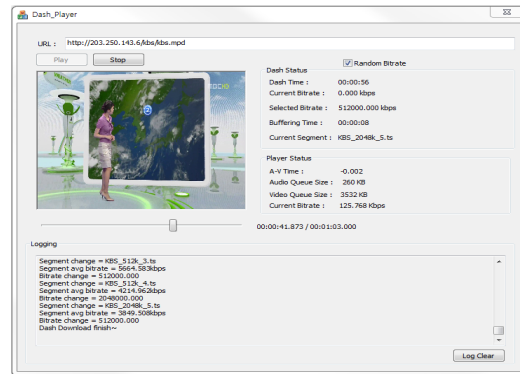


그림 8. DASH 클라이언트 재생 화면  
Fig. 8 Screen of DASH client playing

### V. 고찰 및 결론

기존 스트리밍 서비스에서는 서버에서 하나의 콘텐츠를 클라이언트로 전송하는 단순한 방식이었으나 본 논문에서는 HTTP를 이용한 동적 적응적 스트리밍 기술을 활용하여 실시간으로 망 간섭과 통신환경에 맞는 끊임 없는 스트리밍 기술을 구현하였다. 기존 한가지의 정해진 비트율에 맞는 콘텐츠를 제공하던 방식과는 달리 서버에서 관리자가 하나의 콘텐츠를 망 환경에 맞는 비트율의 콘텐츠들로 변환하고 변환된 콘텐츠들을 각각의 세그먼트로 분할하여 서버에 저장해 두고 분할된 정보에 대해 MPD 문서로 만들어서 서버에서는 클라이언트에게 MPD 문서만 제공한다. 클라이언트는 MPD 문서를 파싱하여 서버에 저장되어 있는 세그먼트를 현재의 네트워크 상태를 확인하여 현 상황에 맞는 비트율의 세그먼트 콘텐츠를 제공받아 제한된 환경이라도 끊임 없는 스트리밍 서비스를 가능하게 만들었다.

본 논문의 DASH 시스템 설계 및 구현을 통해 네트워크 상황에 맞는 스트리밍 서비스를 제공하는 것을 확인하였다.

본 논문에서는 기존의 한 가지 원본 영상으로 스트리밍 서비스를 제한된 네트워크 환경에서도 각각의 환경에 맞는 끊임 없는 스트리밍 서비스를 제공하기 위해 여러 화질의 비트율로 변환하여 네트워크 상황에 맞는 비트율로 적응적으로 플레이하는 것에 중점을 두고 개발 및 테스트되어, 서비스가 가능함을 검증하는데 초점을

맞추었다. 이러한 HTTP에 기초한 동적 적응적 스트리밍 서비스는 제공받는 클라이언트에게는 양질의 서비스 제공을 통한 만족도 향상과, 서버측 면에서는 한곳에 중점적으로 몰리는 과부하의 단점도 충분히 보완할 수 있고, 향후 IP-TV와 Smart TV의 사업 분야에도 무한한 가능성의 발판이 될 것으로 판단된다.

향후 연구 과제로는 현재의 PC 환경이 아닌, 클라우드 컴퓨팅이나 셋톱박스 등에 적용하여 디지털 콘텐츠의 양질화에 대한 연구가 필요하고, 비디오 및 오디오 인코더의 성능 최적화를 통한 “Live” 형식의 서비스와, 콘텐츠 보호를 위한 연구가 필요하다.

### 참고문헌

- [1] R. S. Cruz, M.S. Nunes, and J.P. E. Goncalves, “A Personalized HTTP Adaptive Streaming WebTV”, in Proceedings of the First International Conference on User Centric Media Workshop, UCMedia ‘09 Venice, 2009. 12.
- [2] H. Schulzrinne et. al, “RTP:A Transport Protocol for Real-Time Application”, IETRF RFC 3550, 2003. 7.
- [3] X. Cheng, C. Dale, J. Liu, “Understanding the characteristics of internet short video sharing” You Tube as a case study“, Technical report, Cornell Univ., 2007. 7.
- [4] A. Zambelli, “IIS Smooth Streaming Technical Overview”, Microsoft Corporation, 2009. 3.
- [5] Information technology=MPEG systems technologies-Part 6:Dynamic adaptive streaming over HTTP(DASH), ISO/IEC CD 23001-6: 2010. 10.
- [6] ISO/IEC, “Information technology-Coding of audio-visual objects-part3: Audio, ISO/IEC 14496-3 (MPEG-4), 2005.
- [7] ISO/IEC, “Information technology-Coding of audio-visual objects-part11: Scene description and application engine”, ISO/IEC 14496-11(MPEG-4), 2005.

### 저자소개



**정선철(Sun-Chul Jeong)**

1990년 한국해양대학교  
기관공학과(공학사)  
1999년 충남대학교  
기계공과(공학석사)

1995년~2011년 한국전력공사 원자력 R&D 연구  
2011년~현재 배재대학교 컴퓨터공학과 박사과정  
※관심분야: XML, Web Services, Semantic Web, USN, SVG, DASH, MPEG-21



**반태학(Tae-Hak Ban)**

2011년 배재대학교  
컴퓨터공학과(공학사)  
2011년~현재 배재대학교  
컴퓨터공학과 석사과정

※관심분야: XML, EDI, Web Services, USN, DASH, MPEG-21



**정희경(Hoe-Kyung Jung)**

1985년 광운대학교  
컴퓨터공학과(공학사)  
1987년 광운대학교  
컴퓨터공학과(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)  
1994년~현재 배재대학교 컴퓨터공학과 교수  
※관심분야: 멀티미디어 문서정보처리, XML, SVG, Web Services, Semantic Web, MPEG-21, Ubiquitous Computing, USN