일반논문 (Regular Paper)
방송공학회논문지 제23권 제4호, 2018년 7월 (JBE Vol. 23, No. 4, July 2018)
https://doi.org/10.5909/JBE.2018.23.4.549
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

MPEG-DASH 기반 저지연 라이브 360 VR 분할영상 스트리밍 서버 구현

김 현 욱[a)‡], 최 우 성[b)], 양 성 현[a)]

Implementation of MPEG-DASH based Low-Latency Live 360 VR Tiled Video Streaming Server

Hyun Wook Kim[a)‡], U Sung Choi[b)], and Sung Hyun Yang[a)]

요 약

360 VR(Virtual Reality) 비디오와 같이 고품질의 비디오를 기존 케이블 망에서 IPTV 또는 OTT(Over the Top) SettopBox과 같은 저사양 미디어 서비스 기기를 통해 저지연 라이브 스트리밍 서비스를 할 수 있도록 MPEG DASH(Dynamic Adaptive Streaming over HTTP) 기반의 스트리밍 서버를 설계하고 구현하였다. 또한, 서버 응답 지연 시간을 줄이기 위해 스트리밍 영상 파일(MPD, Segment Files)을 메모리상에 캐싱 하여 서비스 할 수 있는 관리 프로세스를 설계하고 적용하였다. 그리고 실험을 통해 bitrate가 50,000kbps 이상이고, 8K@60P 급의 고품질 분할 영상 스트리밍을 지원하는 것을 확인하였다.

Abstract

We designed and implemented streaming server based on MEPG DASH, which is able to provide high quality video with low-latency live streaming service like 360 VR video on the existing cable network via low-spec media service devices such as IPTV and OTT(Over the Top) SettopBox. We also designed and applied management process which is cable of supporting services by cashing streaming video file(MPD, Segment Files) to reduce the server response delay time. Further more, we confimred that it is also able to provide high quality of tiled video streaming with over 50,000kbps bitrate and 8K@60P through the experiment.

Keyword : MPEG-DASH, Low-Latency Live Streaming, Streaming Cache, Tiled Video

a) 광운대학교 전자공학과(Department of Electronic Engineering, Kwangwoon University)
b) ㈜유일씨엔티(Yoolicnt Co., Ldt.)
‡ Corresponding Author : 김현욱(Hyun Wook Kim)
E-mail: khw@kw.ac.kr
Tel: +82-2-940-5751
ORCID: https://orcid.org/0000-0001-9725-6050

※ 이 논문은 2017년도 정부(과학기술정통부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00307, 고품질 VR 콘텐츠 실시간 서비스를 위한 분할영상 스트리밍 기술 개발).
※ This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government (MSIT) (No.2017-0-00307, Development of Tiled Streaming Technology for High Quality VR Contents Real-Time Service).
· Manuscript received June 4, 2018; Revised July 6, 2018; Accepted July 6, 2018.

Ⅰ. 서 론

컴퓨터 하드웨어 및 네트워킹 기술의 발전으로 4K, 8K 급의 높은 대역폭을 요구하는 비디오 스트리밍 서비스가 가능해지면서 멀티미디어 서비스 환경은 사용자에게 단순히 2D 비디오 영상을 보여주는 환경에서 3D 또는 360 VR(Virtual Reality) 비디오를 스트리밍 할 수 있는 가상 환경으로 진화하고 있다[1][2]. 하지만 아직까지 가상환경에서 멀티미디어 서비스를 제공하는 VR HMD(Head Mount Display) 장치의 스트림 대역폭이 제한적이기 때문에, 8K 이상 또는 높은 Bitrate을 갖는 고품질 비디오 영상을 스트리밍 하는 것은 한계가 있다[3]. 이러한 한계를 극복하기 위해 최근에는 스트리밍 대역폭에 적응적이고, 공간 관계 정보 활용이 가능한 MPEG DASH(Dynamic Adaptive Streaming over HTTP) SRD(Spatial Relationship Description)를 적용하여 HMD 장치의 뷰포트(View Port)는 고품질의 Tile들로, 뷰포트를 제외한 영역은 저품질의 Tile들을 사용하여 HMD로 장치로 전달되는 전체 영상의 스트리밍 대역폭을 낮추는 방법[4][5]이 현실적인 방안으로 대두되고 있다. 하지만 HMD 장치의 뷰포트가 Tile들의 공간 경계에 걸쳐 있다면, 다수의 고품질 Tile들을 필요로 하기 때문에 상황에 따라 높은 스트리밍 대역폭을 요구한다는 단점을 가지고 있다. 이를 해결하기 위해서는 작은 공간 단위의 Tile로 영상을 분할하여 해결 할 수 있지만, 늘어나는 Tile의 수만큼 HTTP Connection 역시 증가하기 때문에 통신 지연이 발생한다. 클라우드 환경과 같이 대량의 자원을 기반으로 하는 온라인 콘텐츠 서비스 시스템에서는 스트리밍 되는 영상의 Tile Set을 미리 메모리상으로 Caching 하여 전송 지연 시간을 단축시키는 방법으로 해결 할 수 있겠지만, 기존 케이블망에서는 IPTV 셋톱박스, OTT(Over the Top) 단말과 같이 제한적인 네트워크 대역폭과 H/W 제원을 갖는 장치에서 라이브 스트리밍 서비스를 제공하기 때문에 영상의 모든 세그먼트(Segments)를 메모리에 캐싱(Caching)하여 처리 것은 한계가 있다.

본 논문에서는 기존 케이블에서 360 VR 스트리밍 서비스를 라이브로 IPTV 셋톱박스, OTT 단말과 같이 제한적인 자원으로 구성된 장치를 통해 다수의 유저에게 제공하기 위한 MPEG-DASH 라이브 스트리밍 서버를 설계하고 구현하였다. 그리고 전송 지연을 줄이기 위해 제한 적인 자원에서 활용할 수 있는 MPEG-DASH 세그먼트 캐시(Segment Cache)를 설계하고 구현하여 적용하였다. 그리고 실험을 통해 대싱(Dashing)된 세그먼트들의 전송 지연 시간이 단축 되는 것을 확인하였다.

본 논문의 구성은 2장에서 사전 연구로 기존 케이블 망 기반의 고품질 360VR영상을 스트리밍하기 위한 시스템 구조를 확인한다. 그리고 3장에서 MPEG DASH 기반의 스트리밍 서버를 설계하고 구현하였으며, 지연 시간을 줄이기 위한 Dashing 파일들에 대한 Memory Cache 구조를 설계하고 적용하였다. 또한 MPEG DASH 기반의 라이브 서비스를 적용하기 위한 Index 기능을 추가하였다. 4장에서는 실험을 통해 스트리밍 서버의 처리 지연 시간을 측정하였다. 그리고 마지막으로 5장에서 결론을 맺는다.

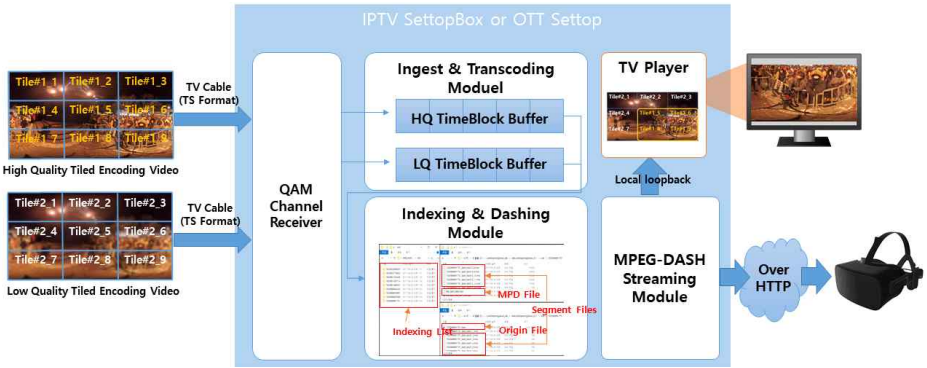

그림 1. 기존 케이블 망 기반 IPTV, OTT 셋탑박스 용 360 VR Live 스트리밍 서비스 구조도
Fig. 1. 360 VR Live Streaming Service Structure based on existing cable network for IPTV or OTT Settop

Ⅱ. 사전 연구

1. 케이블 망 기반 IPTV&OTT SettopBox 환경 고품질 360 VR 영상 스트리밍 시스템

케이블 망 기반의 고품질 360 VR 영상 스트리밍을 위한 전체 시스템은 위 그림 1과 같다. 네트워크 대역폭에 대응적인 MPEG-DASH 기반의 스트리밍 서비스를 위해서는 우선 케이블망에서 고품질, 저품질 영상을 수신한다[6]. 그리고 수신되는 영상을 IPTV 또는 OTT 셋탑에서 일정 시간 단위로 Segmentation을 하고 Segmentation 된 영상을 다시 Dashing 및 Indexing한다. Dashing된 파일들을 Timestamp 기반의 Index에 따라 TV 또는 HMD 장비들로 스트리밍 서비스를 제공하는 구조로 이루어져 있다.

Ⅲ. 본 론

이번 장에서는 케이블 방송 망 기반의 IPTV, OTT Settop 환경에서 360 VR 분할 영상을 라이브로 스트리밍 서비스하기 위한 MPEG-DASH 라이브 스트리밍 서버를 설계하고 구현하였다. 그리고 스트리밍 서버에서 HTTP 응답 지연 시간을 줄이기 위해 MPD, Segment File을 효율적으로 메모리 상에서 관리할 수 있는 캐시 구조를 설계하고 적용하였다.

1. MPEG-DASH 기반 분할 영상 라이브 스트리밍 서버 설계 및 구현

본 논문에서 설계 및 구현한 스트리밍 케이블 방송망으로부터 전송되는 영상신호를 MPEG-DASH 기반으로 Client에 스트리밍 하기 위해 일정 시간단위로 분할하여 클라이언트로 전송하도록 설계하였다. 또한 스트리밍까지의 전 과정을 기능 별(Indexing, Dashing, Caching 등)로 모듈화 하였다. Indexing 수신되는 영상 신호를 순서대로 스트리밍하고, 빠른 검색을 위해 Timestamp 단위로 관리 한다. Dasing 모듈은 케이블로부터 수신받은 영상을 타일 단위로 분할하는 역할을 수행하도록 설계 하였고 Caching 모듈은 각각의 전송파일들을 빠르게 응답하기 위해 MPD, Segment 파일을 메모리에 캐싱(Caching) 하여 클라이언트의 요청에 최대한 빠르게 응답할 수 있도록 구성하였다. 아래 그림 2는 MPEG-DASH기반 저지연 라이브 스트리밍을 위한 서버 구조도이다.

제안하는 스트리밍 서버의 구조는 위 그림 1과 같이 방송 케이블로부터 영상을 수신하고, 수신된 영상을 Dashing, Indexing하는 수신부와 Client로 영상을 스트리밍하기 위해 Dashing된 영상을 Caching하고, Client의 요청에 따라 MPD, Dash Segment를 전송하는 스트리밍 서비스부(송신부)로 구성되어 있다.

1.1 케이블 영상 수신부

케이블 영상 수신부에서는 다중 QAM 신호를 수신 카드를 통하여 복조하고 고품질, 저품질, Metadata를 추출한다. 그리고 MPEG-TS 포맷으로 추출된 영상은 시간단위로 분할하여 Dashing 작업을 수행하면서 각각의 영상들을 Indexing 한다. 그림 3은 케이블로부터 수신 받은 신호를 수신 카드에서 처리되는 과정을 구성한 것이다.

고품질의 8K영상을 송수신 하기 위해서는 케이블 방송

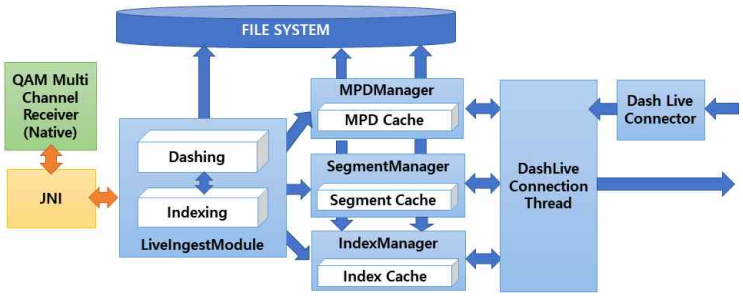

그림 2. MPEG-DASH 기반 라이브 스트리밍 서버 구조
Fig. 2. MPEG-DASH base Live Streaming Server Structure

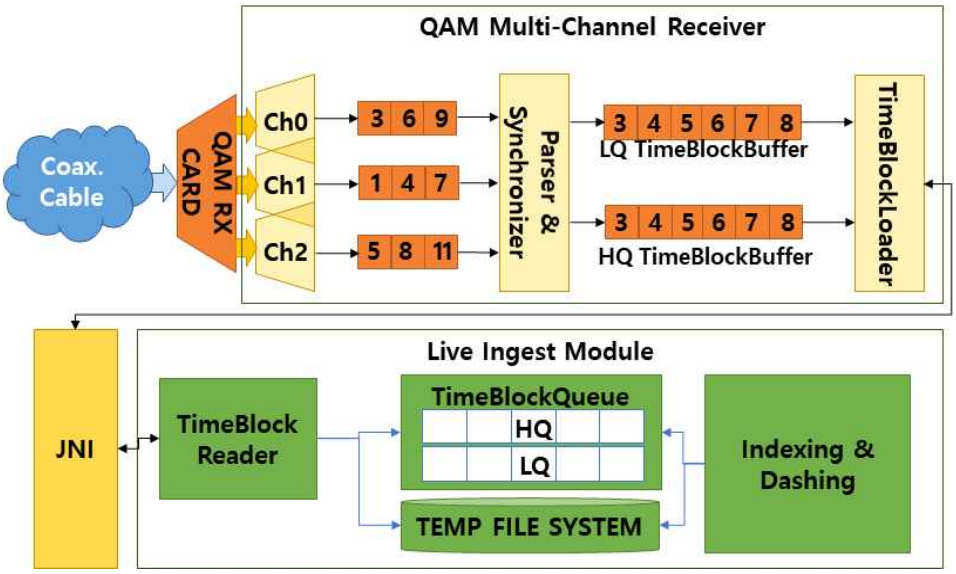

그림 3. 라이브 미디어 수신부 프로세싱 모듈 구조
Fig. 3. Live Media Receive Processing Module Structure

망에서 bitrate에 따라 80.9~116.4 Mbps 대역폭이 요구된다. 본 논문에서는 안정적인 8K영상을 방송하기 위해서 3개의 Multi-Channel을 결합하여 전송하고 수신 받도록 설계 하였다.

연속적으로 수신 받는 영상을 끊김 없이 Client로 제공하기 위해서 실시간으로 수신되는 영상을 우선 TimeBlock-Buffer에서 버퍼링 하였고, 이를 비동기식으로 스트리밍 서버에 TimeBlockQueue에 일정 단위만큼 다시 적재한다. 버퍼는 영상의 크기에 따라 최소 1000ms에서 최대 5000ms 크기로 할당이 가능하며 LiveIngestModule를 통해 Client로 전송이 가능하도록 Dashing 처리를 한다.

1.2 저지연 라이브 스트리밍 서비스 부

스트리밍 서비스 부 는 Index와 Client 정보를 관리하기 위한 IndexManager, Dashing된 Segment파일을 관리하기 위한 SegmentManager, Dashing된 MPD파일을 관리하기 위한 MPDManager, 스트리밍 Connection(Response, Request)를 처리 하기 위한 DashLiveConnector, DashLiveConnectionThread 모듈로 설계 하였다.

IndexManager에서는 Timestamp기반의 파일 Index를 생성하고, Client별 IndexMap을 생성하여 ClientID마다 Last Index 관리를 수행한다. Client로 스트리밍하기 위해서 각각의 Client정보와 스트리밍 되는 영상의 Segment파일을 관리하기 위한 ClientIndexManager, SegmentIndexManger를 그림 4와 같이 구성하였다. 각각의 ClientIndexMap은 최종 스트리밍 된 Last Segment Index를 저장하고 관리하여 스트리밍 흐름제어 및 관리에 적용할 수 있도록 설계 하였다.

Dashing된 MPD를 MPDCache에 적재하고, Cache 가 Full일 경우 시간별로 MPD를 Cache에서 순차 삭제 하도록

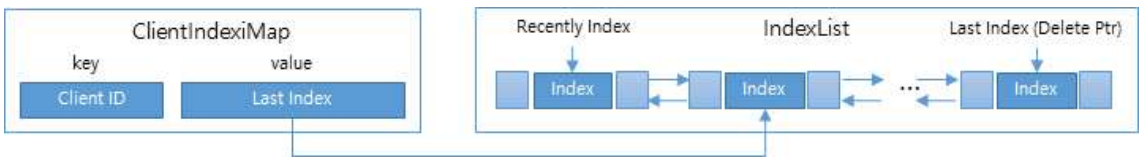

그림 4. ClientID 별 Last Index Map 구조
Fig. 4. ClientID and Last Index Map Structure

구성하였다. 또한 빠른 응답속도를 위해 클라이언트에서 MPD파일의 요청이 있을 경우 필요한 MPD 파일을 Cache 또는 파일에서 Load하여 Client로 전송한다. 만약 MPD 파일의 요청이 있을 경우 우선 Cache에서 탐색을 수행하고, Cache에 존재하지 않을 경우 파일 시스템에서 파일을 Cache에 적재하도록 설계 하였다. MPDCache는 아래 그림 5과 같이 Key/Value 형식의 HashMap으로 구성하였다. 여기서 Key는 해당 MPD의 Index이며, Value는 MPD파일의 내용을 Byte Array(byte[])으로 저장한다. IndexList를 사용하여 MPD파일의 Caching 여부를 검색하고, IndexList의 마지막 번지의 해당하는 Index를 Delete Ptr로 정의하여, Cache가 Full인 경우 Delete Ptr의 Index에 해당하는 key/value Cache Data를 HashMap에서 삭제하도록 설계하였다.

Client로 영상을 전송하기 위한 SegmentManager는 Dashing된 Segment 파일들을 Segment에 적재하고, Cache가 Full일 경우 시간별로 Cache의 관리하도록 설계하고 구현하였다. 또한 클라이언트에서 Segment 파일의 요청이 있을 경우 해당 Segment 파일을 Cache 또는 파일 시스템에서 Load하여 제공하는 역할을 수행한다.

Segment Cache는 아래 그림 6과 같이 Key/Value 형식의 HashMap으로 구성되어 있다. 여기서 Key는 Index이며, Value는 다시 Key/Value형식의 HashMap 구조를 가지며, 그 구조는 Segment Data Structure와 같이 Key는 Filename, Value는 Byte Array(byte[]) 구조의 Data로 구성된다. 그리고 IndexList를 사용하여 Segments파일의 Caching 여부를 검색하고, IndexList의 마지막 번지의 해당하는 Index를 Delete Ptr로 정의하여, Cache가 Full인 경우 Delete Ptr의 Index에 해당하는 key/value를 삭제 하도록 설계 하였다.

HTTP 기반으로 Client와 통신하기 위한 DashLive Connector는 그림 7과 같이 설계 하고 구현 하였다. Client로부터 스트리밍 요청을 받은 시점부터 DashLiveConnection Thread를 생성하고 요청된 파라미터에 따라 ClientID, IndexList, MPD 데이터와 영상 Segment 전송을 시작하도록 설계 하였다. 각각의 데이터는 별도의 Thread 모듈로 설계 하여 다수의 영상 Segment와 Client 관리가 용의 하도록 설계 하였다.

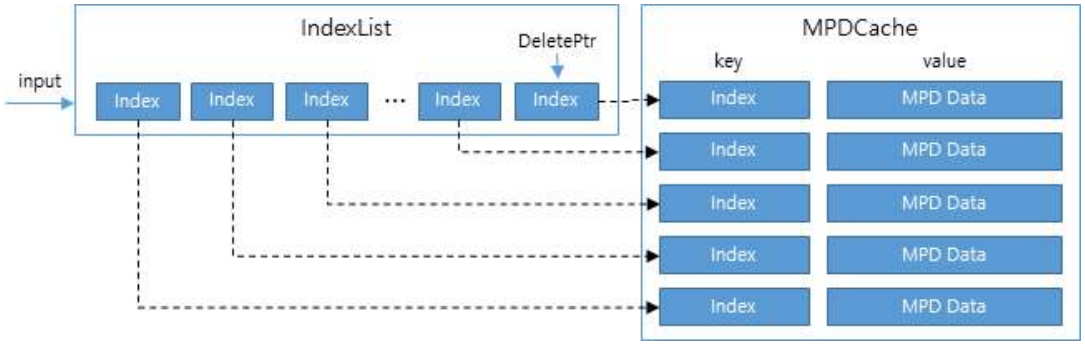

그림 5. MPD Cache 관리 구조
Fig. 5. MPD Cache Management Structure

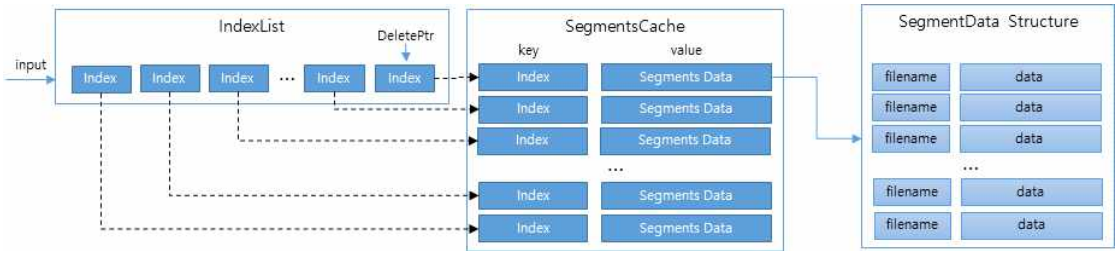

그림 6. Segment Data Cache 구조
Fig. 6. Segment Data Cache Management Structure

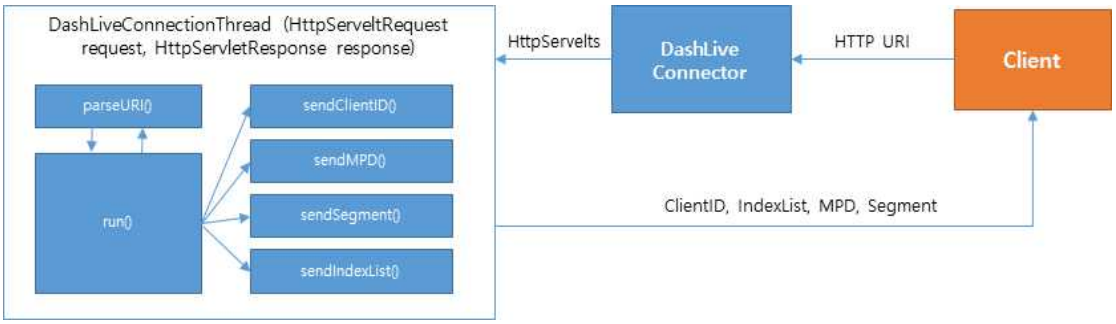

그림 7. 스트리밍 연결 부 모듈 상세 구조
Fig. 7. Streaming Connection Management Module Structure

1.3 저지연 스트리밍을 위한 MPD, Segment 파일 탐색 기법

Client에서 MPD를 요청하였을 경우 DashLiveConnection Thread에서는 IndexManager를 통해 MPD를 요청한 Client의 마지막 MPD Index를 확인하고, Next Index를 요청한다. 이때 Next Index가 존재하지 않을 경우 기존 Index를 최신 Index로 판단하여 기존 Index에 있는 MPD 파일을 전송한다. 만약 IndexManager에서 해당 Client의 ID가 존재 하지 않을 경우 Recently Index의 MPD를 전달하고, 위 그림 4와 같이 신규 ClientIDIndexMap을 생성한다.

Client에서 Segment 파일을 요청하였을 경우 ClientID를 기준으로 IndexManager에서 ClientIndexMap에서 현재 요청하는 Index를 확인하고, SegmentManager에 해당 Index와 Segment 파일 정보를 함께 전달하여 Segment 파일을 요청한다. SegmentManager에서는 Cache 탐색 시 Index를 기준으로 SegmentList를 탐색하여 Cache에 존재 유무를 확인하고, SegmentFilename을 기준으로 다시 Cache에서 Caching된 Segment Data를 탐색한다. 우선 SegmentList에서 Cache에 존재유무를 확인하는 이유는 Cache에 적재되어 있지 않을 경우 바로 파일에서 Load하기 위함이다.

HTTP기반 MPEG-DASH Live 스트리밍을 하기위해서 시간별로 연속적으로 들어오는 데이터를 연속적으로 스트리밍 하기위해 DashLiveConnectionThread에서는 Index Manager를 통해 MPD를 요청한 Client의 마지막 MPD Index를 확인하고, Next Index를 요청하도록 설계 하였다. 또한 다수의 Client가 각기 다른 시간의 스트리밍을 요청할 경우 설계된 IndexManager와Cache Manager를 활용하여 가장최신 스트리밍 데이터를 유지하고 제공하도록 설계 하였다.

2. 저지연 라이브 스트리밍 프로토콜

저지연 라이브 스트리밍을 위해 MPEG-DASH 기반 Streaming Protocol에 기반 하여 클라이언트와 스트리밍 서버 간 통신하기 위한 Request URL에 해당하는 HTTP 1.1 Get Method 방식의 URI 구조와, 라이브 메시지 프로토콜을 설계 하고 구현하였다.

2.1 Request URL 정의

클라이언트에서 스트리밍 서버로 ClientID, MPD, Segment File을 요청하는 URL의 GET URI 구조를 설계하고 구현하였다.

ClientID를 요청하는 Request URL의 GET URI를 'GET_CLIENT_ID'로 정의하였다. ClientID는 최초 스트리밍서버에 Connection시 부여되는 고유번호로 36자리 UUID을 기반으로 한다. MPD를 요청하는 Request URL의 GET URI는 'GET_MPD?client_id={client_id}'으로 구성하였다. ClientID별로 LastIndex를 관리하기 때문에 매개변수로 'clinet_id'를 삽입하였다.

Segment File을 요청하는 Request URL의 GET URI는 '{segment_file_name}?client_id={client_id}'으로 정의하였다.

Client관리를 위하여 요청되는 Client의 ID를 통해 관리하도록 설계 하였으면 서버는 요청되는 클라이언트 별로

그림 8. 서버 클라이언트 간 라이브 스트리밍 프로토콜 시퀀스 다이어그램
Fig. 8. Server & Client Live Streaming Protocol Sequence Diagram

고유번호 UUID를 할당하여 관리 하도록 구성하였다. 최초 서버에 접속하고 스트리밍 통신을 하기위해서는 최초 MPD파일을 수신 받고 클라이언트의 요청에 따라 영상을 제공하게 된다.

2.2 Live Message 교환 프로토콜 정의

클라이언트과 스트리밍 서버 간 Message 교환 프로토콜 시퀀스는 위 그림 8과 같다. 클라이언트에서 스트리밍 서버로 초기 연결 시 Client ID를 획득하고, 그 이후 ClientID를 매개변수로 하여 HTTP 1.1 기반의 GET방식 URL을 사용하여 클라이언트에서 스트리밍 서버로 MPD, Segment 파일을 요청 하도록 설계 하였다.

Ⅳ. 실험 및 결과

1. 저지연 라이브 스트리밍 서버 기능 확인

1.1 저지연 라이브 스트리밍 서버 수신부 기능 구현 확인

이번 장에서는 본 논문에서 설계하고 구현한 스트리밍 서버의 기능 동작을 확인하고, 성능을 측정하였다.

우선 수신부에서 수신되는 영상들에 대한 Dashing 및

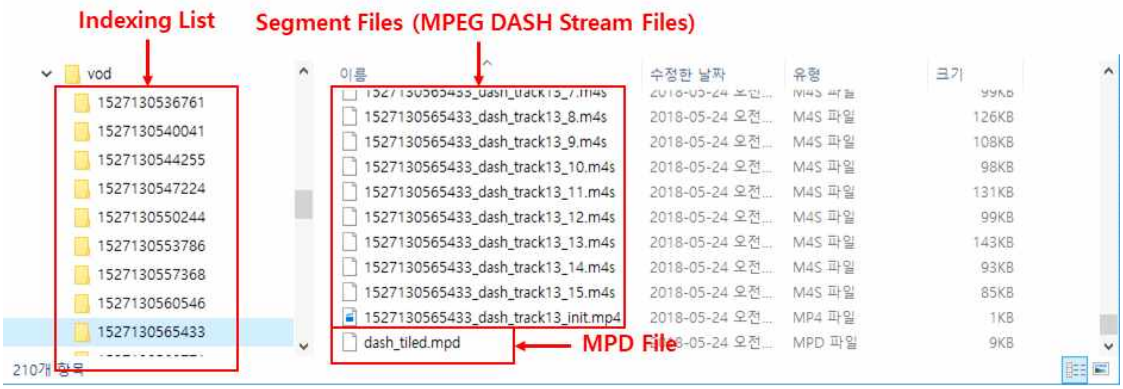

그림 9. 라이브 Indexing 및 Dashing 결과
Fig. 9. Live Indexing and dashing Result

Indexing 처리가 정상적으로 이루어지는지 확인하였고, 그 결과는 위 그림 9와 같다. 위 그림 9는 약 3초 간격으로 수신한 영상을 Indexing 및 Dashing 한 결과를 파일로 저장하여 확인한 결과이다.

1.2 저지연 라이브 스트리밍 서버 스트리밍 부 기능 구현 확인

MPEG-DASH 기반의 스트리밍 서비스 기능을 확인하기 위해 아래 그림 10과 같이 플레이어에서 영상이 서비스 되는 것을 확인하였다. 상단은 TV에 플레이되는 영상이고, 아래 2개 영상은 HMD에 렌더링 되는 화면이다.

2. 저지연 라이브 스트리밍 서버 성능 측정 및 비교

메모리 캐싱을 통한 성능 향상을 확인하기 위해 같이 스트리밍 서버에서 메모리 캐싱 기능을 사용하였을 때와 사용하지 않았을 때 클라이언트로부터 Request Message를 수신한 시점부터 Segment 파일을 클라이언트에 전송하기까지의 서버 프로세싱 시간을 아래 그림 11과 같이 측정하고 평균을 내어 비교 하였다.

시간 비교 측정을 위해 사용한 서버 및 클라이언트 환경은 아래 표 1과 같이 구성하였다.

표 1. 서버 및 클라이언트 제원 정보
Table 1. Specification of experiment streaming environment

Item	Streaming Server	360 VR Player Client PC
CPU	Intel Xeon E5-2687W v4 @3GHz (12Core, 24Thread)	Intel Core-i7 6900K @3.2GHz (8Core, 16Thread)
Memory	DDR4 32GB	DDR4 16GB
Ethernet	1Gbps	1Gbps
GPU	-	NVIDIA Geforce GTX 1080Ti

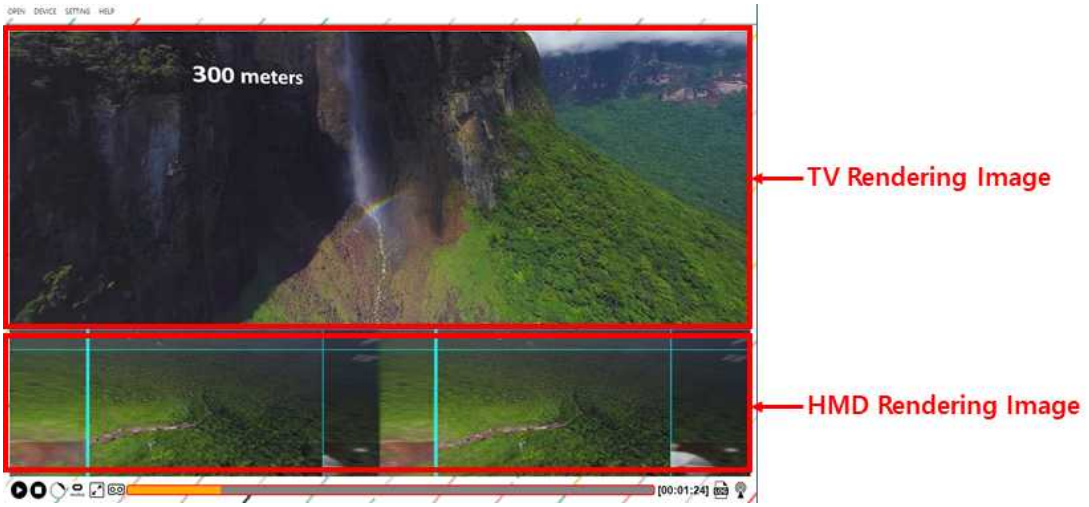

그림 10. MPEG-DASH SRD기반 분할 영상 스트리밍 영상 재생 화면
Fig. 10. MPEG-DASH SRD based Video Streaming Image

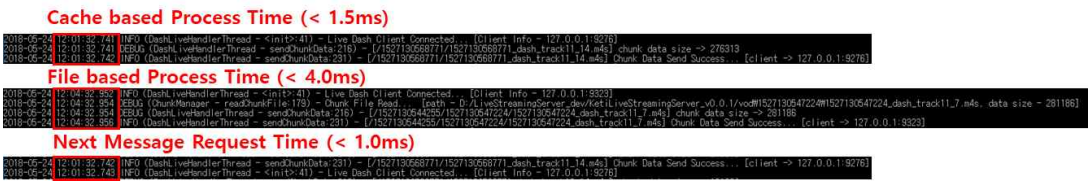

그림 11. 서버 스트리밍 프로세싱 시간 측정 로그
Fig. 11. Sever log for the measurement of Streaming Processing Time

표 2. 시료 영상 스펙
Table 2. Sample Video Specification

Item	KoreaSoccer	Concert	GreenTea	Jongro
resolution	8192*4096	7680x3840	7680x3840	7680x3840
bitrate	72324kbps	78415kbps	52356kbps	79342kbps
frame/sec	25.00	25.00	29.97	29.97
length	56 sec	1min 12sec	33 sec	1min 18sec
Sample Image				

실험에 사용한 시료 영상은 위 표 2와 같이, 3x4 Tile로 분할되어 HEVC로 인코딩된 8K급 고화질 분할 영상을 사용하였다.

각 영상 별 메모리 캐싱 사용 시, 미사용 시 Segment Tile을 전송하기 까지 걸린 서버 프로세싱 시간 측정 결과는 아래 그림 12와 같다. 메모리 캐싱을 사용하였을 때 Segment Tile별 전송 프로세싱 시간이 약 2~3ms 정도 단축되는 것을 확인할 수 있었다.

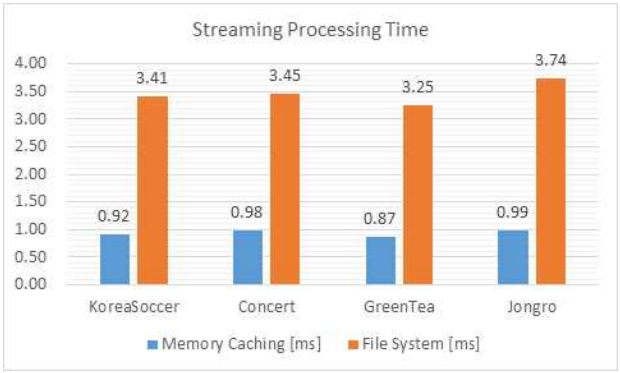

그림 12. Segment Tile 스트리밍 처리 시간
Fig. 12. Segment Tile Streaming Processing Time

본 논문에서 제안한 저지연 스트리밍 서버의 성능 비교 평가를 위해 Dash Low Latency Web Server[7]와 성능을 비교 측정하였다. 측정 방법은 클라이언트 단에서 1개의 화면을 구성하기 위해 Minimum Duration이 1000ms인 6개의 Segment Tiles로 구성된 Adaptation Set을 모두 수신하는데 걸리는 시간을 위 표 2의 각 시료 영상별로 측정하여 비교 분석하였으며, 그 결과는 아래 그림 13과 같다. 성능 측정 환경은 표 1과 동일하다.

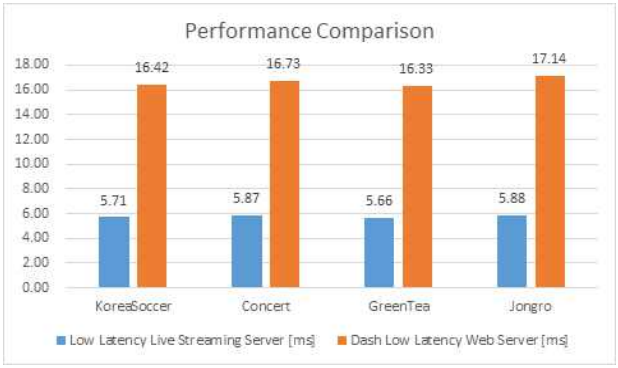

그림 13. 스트리밍 성능 비교 결과
Fig. 13. Performance comparison result

성능 측정 결과 본 논문에서 설계하고 구현한 스트리밍 서버는 50,000kbps, 8K@60P급 이상의 고해상도, 고화질 분할 영상을 충분히 스트리밍 할 수 있음을 확인하였다.

V. 결 론

본 논문에서는 IPTV Settop, OTT Settop과 같은 케이블 방송망 기반의 스트리밍 환경에서 360VR 영상과 같은 고품질 영상을 스트리밍 할 수 있도록 MPEG DASH 기반의 라이브 스트리밍 서버를 설계하고 구현하였다. 그리고 서버 처리 지연 시간을 줄이기 위해 저사양 환경에 맞추어 Dashing된 Segment, MPD 파일과 Index를 Memory에 Caching하여 서비스 하기 위한 구조를 제안하고, 구현하여

적용하였다. 그리고 실험을 통해 8K@60P급의 고해상도, 고품질 분할 영상을 사용하여 성능을 측정하고 검증하였다. 지속적으로 스트리밍 프로세싱 성능을 16K급 초고해상도 영상을 지원하고, 다수의 Client Connection Coverage를 지원할 수 있도록 연구 개발을 수행할 예정이다.

참 고 문 헌 (References)

[1] M. Hosseini, V. Swaminathan, "Adaptive 360 VR Video Streaming: Divide and Conquer!", *IEEE International Symposium on Multimedia,* San Jose, California, USA, 2016.

[2] O.A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, S.Y. Lim, "MPEG DASH SRD - Spatial Relationship Description", *Proceedings of the 7th International Conference on Multimedia Systems,* Klagenfurt, Austria, 2016.

[3] Afshin Taghavi Nasrabadi, Anahita Mahzari, Joseph D. Beshay, Ravi Prakash, "Adaptive 360-Degree Video Streaming using Scalable Video Coding", *Proceedings of the 2017 ACM on Multimedia Conference,* New York, USA, pp. 1689-1697, 2017.

[4] M. Hosseini, V. Swaminathan, "Adaptive 360 VR Video Streaming Based on MPEG-DASH SRD", *IEEE International Symposium on Multimedia 2016*, San Jose, California, USA, pp. 407-408, 2016.

[5] ISO/IEC 23009-1:2014/Amd 2:2015, Spatial relationship description, generalized URL parameters and other extensions

[6] Y. H. Kim, W. S. Cheong, J. J. Lee, "A Study on the Existing Cable Network-based Tiled Streaming Stystem for High Fidelity 360 VR Video Service", *in Proceeding of the 2017 Winter Conference on The Korean Institute of Communications and Information Sciences*, Gangwon-do, Korea, pp. 901-902, 2017.

[7] Dash Low Latency Web Server, https://github.com/gpac/node-gpac-dash (accessed Jul, 18, 2016)

저 자 소 개

김 현 욱

- 2009년 2월 : 광운대학교 컴퓨터공학과 공학사
- 2013년 1월 ~2016년 9월 : ㈜케이사인 정보보안연구소 선임연구원
- 2009년 3월 ~ 현재 : 광운대학교 대학원 전자공학과 석박사통합과정
- ORCID : https://orcid.org/0000-0001-9725-6050
- 주관심분야 : 스마트 홈, 임베디드 시스템, 콘텐츠 응용 기술

최 우 성

- 2005년 5월 : 일본 규수대학교 재료공학 이학박사
- 1993년 3월 ~ 2017년 12월 : 원광대학교 공과대학 정보통신공학과 교수
- 2018년 1월 ~ 현재 : ㈜유일씨엔티 이사
- ORCID : https://orcid.org/0000-0002-4150-2344
- 주관심분야 : 무선통신, 이동통신

양 성 현

- 1993년 2월 : 광운대학교 전지공학(자동제어) 공학박사
- 1991년 3월 ~ 현재 : 광운대학교 전자공학과 교수
- 2005년 7월 ~ 현재 : 광운대학교 Smart H&B Technology Center 센터장
- ORCID : https://orcid.org/0000-0001-8856-764X
- 주관심분야 : 스마트 홈, 디지털 로직, 임베디드