

QUIC 프로토콜을 이용한 360VR 송출 플랫폼 개발

*송민정 *유성근 *박상일

서울과학기술대학교

*mi0_0ng@naver.com *{orcogre, sangilparkmail}@gmail.com

A Study on 360VR Transmission Platform using QUIC protocol

*Song, Minjeong *Yoo, Seong-geun *Park, Sangil

Seoul National University of Science and Technology

요약

최근 정보통신 기술의 발달로 사용자에게 원격 공간의 실재감을 느낄 수 있게 하는 실감미디어의 발전이 두드러지고 있다. 이 중 360VR의 경우, 시청자의 시청 영역이 넓어지기 때문에 높은 전송률과 넓은 대역폭을 요구하여 전송 지연과 영상 끊김 등의 여러 문제점이 나타나고 있는 상황이다. 이에 본 논문은 360VR 영상을 지연이 발생하지 않고 전송하기 위해 구글에서 제작한 Chromium 프로젝트의 소스코드를 이용하여 QUIC 서버를 제작하였고 QUIC 프로토콜과 HTTP/2를 이용하여 Chromium 브라우저 클라이언트에 송출하였다. 또한 기존 Node.js 서버의 HTTP 1.1와 비교를 통해 QUIC 프로토콜을 이용한 360VR 전송 속도가 빠르다는 것을 확인하였다.

1. 서론

최근 정보통신 기술의 발달로 사용자에게 원격 공간의 실재감을 느끼게 해주는 실감미디어의 발전이 두드러지고 있다. 특히 사용자가 원하는 방향을 능동적으로 선택하여 시청할 수 있는 VR(Virtual Reality) 비디오 서비스가 급속히 확대되고 있으며, 이에 따라 관련 기술 개발 및 표준화도 활발히 진행되고 있다[1].

360VR 비디오는 시청영역이 고정적인 기존 비디오에 비해 시청자가 시청할 수 있는 영역이 넓어지므로 고해상도의 영상을 중점으로 제작이 되고 높은 전송률과 넓은 대역폭을 요구하고 있다[2]. 이와 같은 환경에서는 클라이언트 측에서 전송 지연(Latency)과 끊김 현상이 발생할 수 있다.

이에 본 논문은 구글의 Chromium 프로젝트를 이용하여 QUIC 프로토콜을 생성하고 이를 이용해 360VR을 송출하는 플랫폼 시스템을 제안한다. 제안 시스템은 전송 지연 솔루션이 요구되는 360VR 플랫폼에 응용될 수 있다.

2. QUIC 프로토콜

(1) QUIC(Quick UDP Internet Connection)

QUIC은 TCP(Transmission Control Protocol)에 대응하여 데이터 송신 시 대기 시간을 줄이는 전송 프로토콜이다. 표면적으로는 UDP(User Datagram Protocol)에서 구현된 TCP + TLS + HTTP/2와 유사하다. 하지만 TCP는 운영체제 커널 및 middlebox 펌웨어에 구현되므로 이를 변경하는 것은 불가능하다. 이에 반해 QUIC은 UDP위에 구축되기 때문에 제한이 없다는 장점이 있다[3].

그림 1과 같이 TCP+TLS 방식은 서버와 2~3회 왕복 통신을 필요로 했기 때문에 불필요한 전송 시간이 생성된다. 하지만 QUIC은 클라이언트가 이전에 지정된 서버와 통신한 경우 Round-trip없이 데이터를 전송할 수 있으므로 웹 페이지가 더 빨리 로드될 수 있도록 설계되었다[4].

또한 QUIC의 장점은 기존의 통신 방식보다 향상된 혼잡 제어 및 손실 복구이다. 어떤 패킷이 수신되었는지에 대한 모호함을 피할 수 있고 재발신 Time out을 방지할 수 있다. 결과적으로 QUIC은 열악한 네트워크 환경에서 TCP를 능가하여 웹 페이지 로드 시간을 단축한다. 이러한 이점은 비디오 서비스에서 두드러진다. QUIC을 이용해 동영상을 시청할 때 기존에 비해 버퍼 용량이 감소되고 시청자들은 이를 통해 한층 질 높은 서비스를 이용할 수 있다[4].

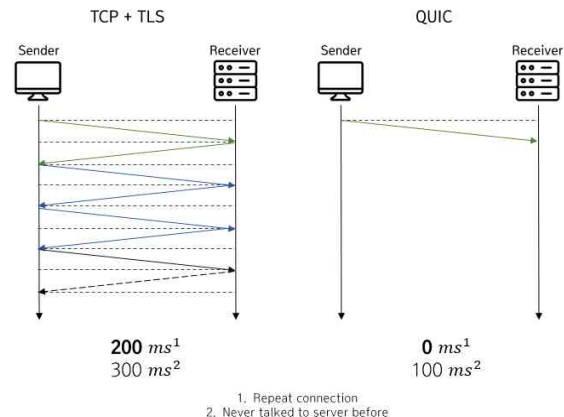


그림 1. TCP+TLS, QUIC의 비교

(2) Chromium

Chromium은 구글에서 개발한 모든 인터넷 사용자가 웹을 경험하며 보다 안전하고 빠르게 그리고 안정적인 방법을 구축하는 것을 목표로 하는 오픈 소스 브라우저 프로젝트이다. 구글은 Chrome 브라우저의 소스 코드를 제공하기 위해 제작하였으며 Chrome과 Chromium은 대부분의 코드와 기능을 공유하고 있다. Chromium OS는 구글 운영체제의 오픈 소스 개발 버전 운영체제이다. 이는 Chromium 브라우저를 주 사용자 인터페이스로 사용한다. 결과적으로 Chromium OS는 웹 응용 프로그램을 지원하고 있다[5].

Chromium 프로젝트에서 제공되는 소스 코드에는 QUIC 서버가 내장되어 있고 개발자는 이를 통해 편리하게 QUIC 프로토콜을 이용할 수 있다.

3. 제안하는 프레임워크

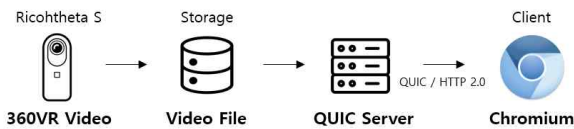


그림 2. 제안하는 프레임워크 구성도

제안하려는 프레임워크는 그림 2와 같다. 리코세타(Ricotheta) S를 이용하여 360VR 영상을 촬영한 후 생성된 비디오 파일을 Storage에 저장한다. 저장된 파일은 Chromium 프로젝트를 빌드하여 생성된 QUIC 서버에 전송되고 최종적으로 클라이언트인 Chromium 브라우저에 QUIC / HTTP/2.0 프로토콜을 이용하여 전송하고 개발자는 이를 이용하여 QUIC 프로토콜을 테스트할 수 있다.

4. 구현 결과

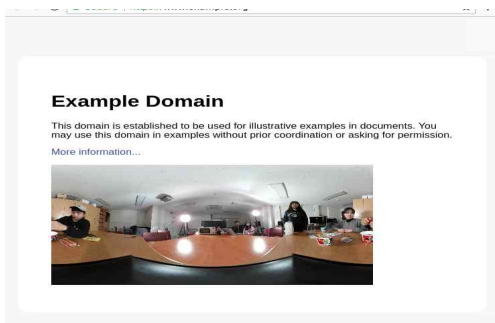


그림 3. 제안 프레임워크로 구성된 360VR 플랫폼

제안한 프레임워크로 구현한 결과는 그림 3과 같다. Javascript 기반의 웹 페이지를 구성하여 리코세타 S로 촬영한 360VR 영상을 삽입하였다.

Node.js 서버의 HTTP/1.1과 Proto-QUIC 서버의 QUIC 프로토콜을 비교한 결과는 그림 4와 같다. HTTP/1.1의 경우 초기 인덱스 페이지 로딩 과정에 여러 번의 요청과 응답이 반복되어 속도가 낮아지고 효율이 떨어진다는 점을 확인할 수 있다. 이에 반해 QUIC 프로토콜은 초기에 로딩 과정에서 초기에 리소스를 한 번씩 로딩하는 방법을 사용

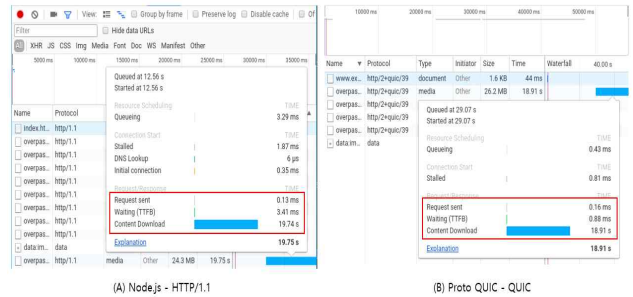


그림 4. HTTP/1.1과 QUIC 프로토콜 테스트 결과

하여 요청이 한 번에 이루어진다. 또한 24.3MB의 360VR 영상을 로딩하는 데 HTTP/1.1은 약 20초가 걸리는 반면 QUIC 프로토콜은 약 19초가 소요되며 약 1초의 로딩 속도 차이를 확인하였다.

5. 결론

본 논문에서는 사용자에게 실재감을 제공하는 실감미디어 중 한 종류인 360VR 영상을 구글의 Chromium 프로젝트와 QUIC 프로토콜을 이용하여 전송하는 플랫폼을 개발하였다. QUIC 프로토콜을 이용해 360VR을 전송한 결과, 기존의 Node.js 서버 기반 HTTP/1.0에 비해 간편한 로딩 프로세스로 구성되어 있기 때문에 약 1초 정도 전송 속도가 빠른 것을 확인했다.

하지만 현재 Chromium 프로젝트로 QUIC 서버를 구현하는 과정 중 이를 컴파일하기 위해 걸리는 시간은 기존의 다른 서버들을 이용하는 시간보다 많이 소요된다. 또한 이를 구축하는 컴퓨터의 사양에 의존되는 점이 많으며 서버 안정화의 과정이 완벽하게 이루어지지 않은 상태이기 때문에 많은 개발자들이 사용하기에는 불편한 점이 존재한다.

추후 진행될 연구에서는 W3C에서 제시된 WebRTC를 이용하여 DataChannel에서 SCTP(Stream Control Transmission Protocol) 대신 QUIC 프로토콜을 적용하여 높은 용량의 실감 미디어를 저지연으로 전송할 수 있는 서비스를 개발할 예정이다.

감사의 글

본 논문은 2018년도 과학기술정보통신부의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. 2018-0061. 제작 편집과 실감 시청 체험 극대화를 위한 개인방송 제작 기술 개발)

참고문헌

- [1] 3GPP TR26.198, "Virtual Reality (VR) Media Services over 3GPP," Jan. 2017
- [2] D.Ochi, Y.Kunita, A.Kameda, A.Kojima, and S.Iwaki, "Live Streaming System for Omnidirectional Video," Proc. of IEEE Virtual Reality (VR), 2015
- [3] <https://www.chromium.org/quic>
- [4] Chromium Blog, "A QUIC update on Google's experimental transport", 2015
- [5] <https://www.chromium.org/Home>