

실시간 라이브 및 비실시간 개인방송 서비스를 위한 적응적 TTL 할당기법

김남태, 유동호, 장정엽, 서봉석, 정은영, 김동호

서울과학기술대학교

{rlaskaxoek, yupjung, youdongho, 10117338, 13184433, dongho.kim}@seoultech.ac.kr

An Adaptive TTL Allocation Scheme for Live and On-Demand Personal Broadcasting Service

Namtae Kim, Dongho You, Jungyup Jang, Bong-seok Seo, Eun-young Jeong, Dong Ho Kim

Seoul National University of Science and Technology (SeoulTech)

요약

본 논문은 CDN(Content Delivery Network)의 동적 캐싱 방식을 기반으로 하는 적응적 TTL(Time-To-Live) 할당기법을 제안한다. 이는 클라이언트가 실시간 개인방송을 시청하는 중 지나간 과거의 특정 장면을 다시 시청할 때 근원(Origin)서버의 부하를 효율적으로 줄일 수 있을 뿐만 아니라 캐시(Cache)서버의 저장 공간도 효율적으로 사용할 수 있는 장점이 있다. 따라서 본 논문에서 제안하는 적응적 TTL 할당기법은 개인방송 시청자들이 지나간 과거의 영상들을 선택적으로 시청할 때 보다 나은 서비스를 제공할 수 있을 것으로 기대된다.

1. 서론

최근 ICT 시장은 스마트폰의 빠른 보급과 네트워크의 브로드밴드화로 모바일 기기를 통한 멀티미디어 콘텐츠의 소비가 증가하면서 1인 미디어의 영향력이 높아지고 있다. 그에 따라 과거부터 꾸준히 개인방송 서비스를 운영하던 아프리카 TV도 호황을 맞이하게 되었으며, 최근에는 SNS(Social Network Service)를 제공하는 Facebook과 세계 최대의 동영상 콘텐츠를 공유하는 YouTube도 실시간 라이브 개인방송을 위한 서비스를 속속 출시하고 있다 [1].

특히 YouTube 같은 경우에는 아프리카TV와는 다르게 과거 지나간 영상 데이터 (예: MPEG-DASH의 세그먼트)를 근원(Origin)서버에 모두 저장함으로써 클라이언트가 실시간 라이브 방송을 보는 중에도 선택적으로 과거 영상을 시청할 수 있다. 하지만 너무 많은 클라이언트가 근원서버에 동시에 지나간 영상을 요청할 경우 근원서버의 과부하를 초래하므로 클라이언트들은 우선적으로 근처에 있는 캐시(Cache)서버에 요청을 하고 캐시서버는 해당 세그먼트가 없을 경우 다시 근원서버에게 요청을 하여 수신한 후 요청한 클라이언트들에게 전송한다. 이와 같은 네트워크 구조를 일반적으로 CDN(Content Delivery Network)이라고 부른다 [2].

이때 근원서버에게 세그먼트를 요청한 캐시서버는 근원서버로부터 해당 세그먼트의 TTL(Time-To-Live)도 할당받게 되는데 [3], 이는 캐시서버가 해당 세그먼트를 실질적으로 저장하는 시간을 나타낸다. 다시 말해 캐시서버는 근원서버가 할당한 TTL 만큼만 세그먼트를 저장하기 때문에 저장 공간에 대한 비용부담도 줄일 수 있다는 장점이 있다. 하지만 TTL이 너무 짧으면 클라이언트가 세그먼트를 요청할 경우 캐시서버가 해당 세그먼트를 가지고 있지 않은 경우가 빈번하여 근

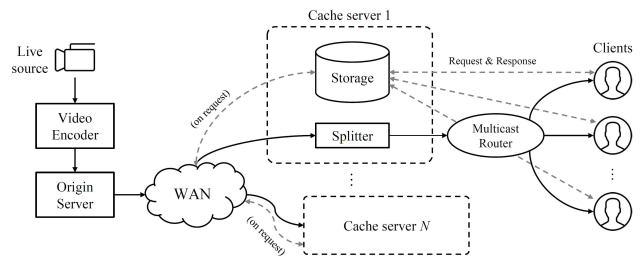


그림 1. 실시간 라이브 및 비실시간 방송 스트리밍을 위한 다이어그램

원서버에게 요청이 물리게 되므로 근원서버의 부하가 증가하게 된다. 반면에 TTL이 너무 길면 캐시서버에서 즉시 데이터를 보내줄 수 있는 캐시 히트율(Hit-rate)이 증가하여 근원서버에 부하를 줄일 수 있지만, 클라이언트의 요청이 많이 없을 때에도 오랫동안 캐시서버에 저장해 두어야 하므로 저장 공간에 대한 부담이 증가하게 되는 트레이드오프(Trade-off)를 가지고 있다.

따라서 본 논문에서는 근원서버가 캐시서버에게 TTL을 할당할 때 각각의 세그먼트의 상태에 따라 적응적으로 TTL을 할당하는 시스템 모델을 제안한다. 이를 통해 클라이언트 요청이 많은 세그먼트들의 TTL은 증가시키고 그렇지 않은 세그먼트들의 TTL은 상대적으로 감소시키므로 위에서 언급한 트레이드오프를 해결할 수 있다.

2. 적응적 TTL 할당 기법

그림 1은 기존의 실시간 라이브 및 비실시간 방송 스트리밍을 위한 전체적인 흐름을 나타낸다. 서론에서 언급한 것과 같이 근원서버는 클라이언트에게 실시간 라이브 방송 서비스를 제공할 때 클라이언트 근

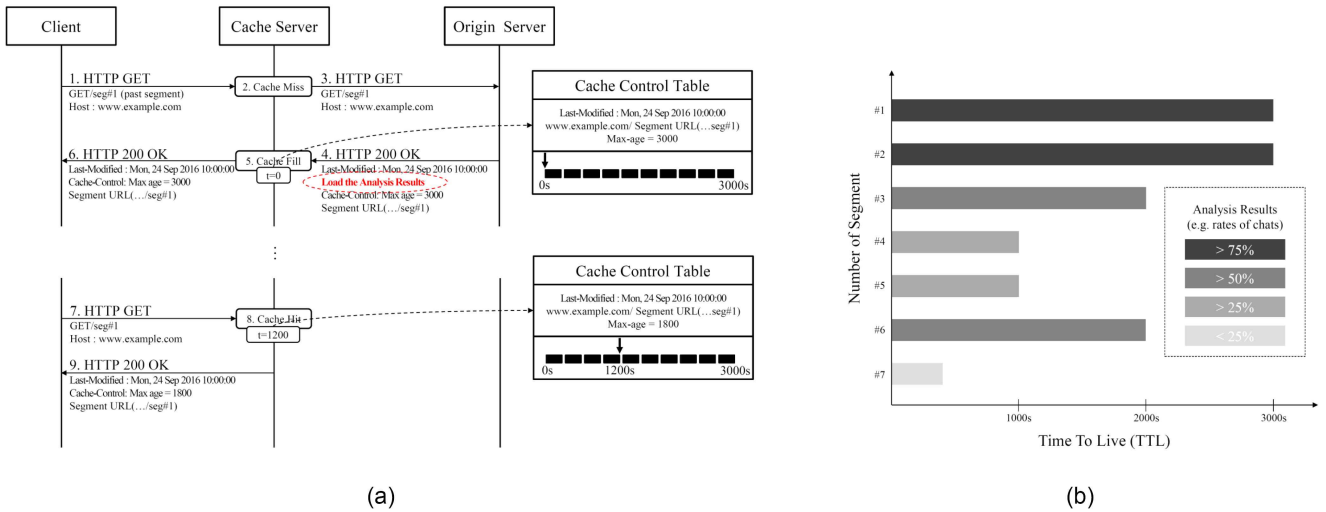


그림 2. 제안하는 적응적 TTL할당을 위한 (a) 동적 캐싱 방법 및 (b) 하이라이트 유무에 따른 적응적 TTL 할당 예시

처에 있는 캐시서버의 스플리터(Splitter)를 통해 전송한다 [2]. 이때 전송된 모든 세그먼트들은 근원서버에 저장되어 있으며, 언제든 클라이언트가 과거 지나간 시점의 세그먼트를 요청할 때 전송해 줄 수 있다. 그러나 클라이언트는 우선적으로 근원서버가 아닌 캐시서버에게 세그먼트를 요청하며, 만약 캐시서버가 해당 세그먼트를 가지고 있지 않다면 캐시서버는 근원서버에게 해당 세그먼트를 요청하여 수신한 후 클라이언트에게 전송해준다. 이때 근원서버는 캐시서버에게 각각의 세그먼트의 TTL을 할당해주는데, 기존에는 모든 세그먼트에게 동일한 TTL을 할당했다.

하지만 본 논문에서 제안하는 방법은 그림 2(a)의 4번째 절차와 같이 캐시서버에게 TTL을 할당하기 전 각각의 세그먼트가 클라이언트의 요청이 많은 하이라이트 장면인지 아닌지를 판단하여 그림 2(b)와 같이 적응적으로 TTL을 할당한다. 여기서 해당 세그먼트가 하이라이트인지 아닌지 판단하는 기준은 실시간 라이브 방송을 보는 시청자들의 동시시간대 시청률, 채팅방의 순간 대화량 및 대화율, 본 세그먼트의 해당하는 장면이 흥미로울 경우 발생하는 ‘좋아요’ 또는 ‘별풍선’의 클릭 수, 그리고 특정시간의 영상을 다시보기 위한 요청 횟수 등이 기준이 될 수 있다. 다시 말해 위에 특정한 기준을 가지고 분석 한 결과를 수단개로 나눠서 특정한 기준점을 넘으면 더 많은 TTL을 할당하고 그 기준점을 넘지 못하면 더 낮은 TTL을 할당하는 방법이다. 이렇게 하면 하이라이트 장면이 있는 세그먼트 같은 경우는 클라이언트의 요청이 잦으므로 캐시 히트율을 높여 클라이언트의 QoS(Quality of Service)를 향상시킬 수 있을 뿐만 아니라 근원서버의 부하도 줄일 수 있게 된다. 반면에 하이라이트 장면을 가지고 있지 않은 세그먼트는 클라이언트의 요청이 뜸하므로 오랫동안 캐시서버에 저장해둘 필요가 없기 때문에 캐시서버의 저장 공간을 효율적으로 사용할 수 있는 장점이 있다.

3. 결론

본 논문에서는 CDN의 동적 캐싱 방식을 기반으로 하는 적응적 TTL 할당기법을 제안했다. 이는 기존의 TTL 할당 방식에 근원서버가 특정한 기준을 판단할 수 있는 시스템 모델을 추가하여 각 세그먼트

마다 적응적인 TTL을 할당할 수 있도록 설계하였다. 따라서 본 논문의 실시간 라이브 및 비실시간 개인방송 서비스를 위한 적응적 TTL 할당기법은 근원서버의 부하를 줄일 뿐만 아니라 캐시서버의 저장 공간을 효율적으로 사용하여 개인방송 시청자들이 지나간 과거의 영상들을 선택적으로 시청할 때 보다 나은 서비스를 제공할 수 있을 것으로 기대된다.

감사의 글

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음.[B0717-16-0012, 제작 편리성과 실감시청 체험 극대화를 위한 개인방송 제작기술 개발]

참고논문

- [1] K. Pires and G. Simon, “YouTube live and Twitch: a tour of user-generated live streaming systems,” in *Proc. ACM Multimedia Systems Conference*, pp. 225-230, Mar. 2015.
- [2] G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman, and Y. A. Reznik, “Video coding for streaming media delivery on the Internet,” *IEEE Trans. Circuits Syst. Video Technol.*, vol 11, no. 3, pp. 269-281, Mar. 2001.
- [3] J. Wang, “A survey of web caching schemes for the Internet,” *ACM SIGCOMM Computer Communication Review*, vol 29, no. 5, pp. 36-46, Oct. 1999.