

일반논문 (Regular Paper)

방송공학회논문지 제25권 제1호, 2020년 1월 (JBE Vol. 25, No. 1, January 2020)

<https://doi.org/10.5909/JBE.2020.25.1.94>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

P2P 통신 병용 DASH 시스템의 피어 부하 분산 방안 연구

서 주 호^{a)}, 김 용 한^{a)*}

A Peer Load Balancing Method for P2P-assisted DASH Systems

Ju Ho Seo^{a)} and Yong Han Kim^{a)*}

요 약

현재 유무선 인터넷을 통한 미디어 소비는 대부분 ISO/IEC MPEG(Moving Picture Experts Group)에 의해 표준화된 미디어 스트리밍 방식인 DASH(Dynamic Adaptive Streaming over HTTP) 또는 이와 유사한 형태의 적응형 미디어 스트리밍(adaptive media streaming) 기술에 의해 시행되고 있다. 이들은 모두 ISP(Internet Service Provider)가 웹서비스를 원활하게 제공하기 위해 충분히 설치할 수밖에 없는 HTTP 캐시(cache)에 크게 의존한다. 결과적으로 미디어 스트리밍 사용자 증가에 따라 CDN(Contents Delivery Network) 사업자의 서버 증설 부담 대신 ISP의 HTTP 캐시 증설 부담이 커지게 되었다. 이로 인해, ISP들은 이러한 HTTP 증설 비용을 보전하기 위해 CDN 사업자에게 미디어 트래픽 비용을 부과하게 되었다. 최근 이러한 비용을 줄이고자 P2P(Peer-to-Peer) 통신을 함께 사용하는(P2P-assisted) DASH 방식이 제안되었다. 또한 이러한 P2P 통신 병용 DASH 시스템의 효율을 극대화하여 CDN 사업자의 비용을 최대한 절감시키는 피어 선택 알고리즘도 연구되었다. 그러나 이 알고리즘은 선택된 피어에게 부담을 집중시키는 경향이 있다. 본 논문에서는 피어의 부담을 여러 피어들에게 분산시키면서도 CDN 사업자의 비용 절감 수준을 적절하게 유지하는 새로운 피어 선택 알고리즘을 제안하고, 이를 WebRTC(Web Real-Time Communication) 표준 API를 활용한 웹기반 스트리밍 시스템에 구현한 후, 실험을 통해 제안한 알고리즘의 효용성을 검증하였다.

Abstract

Currently media consumption over fixed/mobile Internet is mostly conducted by adaptive media streaming technology such as DASH (Dynamic Adaptive Streaming over HTTP), which is an ISO/IEC MPEG (Moving Picture Experts Group) standard, or some other technologies similar to DASH. All these heavily depend on the HTTP caches that ISPs (Internet Service Providers) are obliged to provide sufficiently to make sure fast enough Web services. As a result, as the number of media streaming users increases, ISPs' burden for HTTP cache has been greatly increased rather than CDN (Content Delivery Network) providers' server burden. Hence ISPs charge traffic cost to CDN providers to compensate for the increased cost of HTTP caches. Recently in order to reduce the traffic cost of CDN providers, P2P (Peer-to-Peer)-assisted DASH system was proposed and a peer selection algorithm that maximally reduces CDN provides' traffic cost was investigated for this system. This algorithm, however, tends to concentrate the burden upon the selected peer. This paper proposes a new peer selection algorithm that distributes the burden among multiple peers while maintaining the proper reduction level of the CDN providers' cost. Through implementation of the new algorithm in a Web-based media streaming system using WebRTC (Web Real-Time Communication) standard APIs, it demonstrates its effectiveness with experimental results.

Keywords : Media streaming, DASH, P2P-assisted, WebRTC

Copyright © 2020 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

I. 서론

현재 유무선 인터넷을 통한 미디어 소비는 대부분 적응형 미디어 스트리밍(adaptive media streaming) 방식에 의해 시행되고 있다. 이러한 방식에는 ISO/IEC MPEG(Moving Picture Experts Group) 표준인 DASH(Dynamic Adaptive Streaming over HTTP) 또는 이와 유사한 기술이 사용된다. 이러한 기술들은 모두 HTTP를 사용하여 서버와 클라이언트 간의 일대일 통신에 의해 미디어를 분배한다. 따라서 이 기술들은 기본적으로 ISP(Internet Service Provider)들이 빠른 웹 서비스를 제공하기 위해 설치할 수밖에 없는 HTTP 캐시(cache)에 크게 의존한다. 이에 따라 미디어 스트리밍 서비스 사용자가 증가할수록 ISP는 더 많은 HTTP 캐시를 설치해야 하는 부담을 지게 되었다. 원래 이 비용은 미디어 스트리밍 서비스로 인해 급증한 것이므로, 이를 제공하는 CDN(Content Delivery Network) 사업자가 부담하도록 ISP들이 CDN 사업자들에게 트래픽 비용을 부과하기에 이르렀다. 이로 인해 사용자가 증가함에 따라 CDN 사업자들이 부담해야 하는 트래픽 비용이 그 만큼 더 커지게 되었다.

최근 P2P(Peer-to-Peer) 통신을 함께 사용하여 동일한 콘텐츠를 소비하는 피어(peer)들이 이미 CDN 서버로부터 획득한 콘텐츠의 DASH 세그먼트(segment)(이하 ‘세그먼트’로 약칭)들을 공유하게 함으로써 CDN 서버로의 스트리밍 요청을 줄여서 CDN 사업자의 트래픽 비용(이하 ‘CDN 비용’으로 약칭)을 절감시키는 P2P 통신 병용(P2P-assisted) DASH 시스템이 제안되었다^[1]. [1]에서 이 시스템을 ‘Hive’라 불렀으므로, 본 논문에서도 같은 이름을 사용하도록 한다. Hive가 처음 제안되었을 때에는 어떤 클라이언트가 필

요로 하는 세그먼트를 갖고 있는 피어들이 여럿 있을 때, 무작위로 그 중 하나를 선택하는 ‘무작위 피어 선택 알고리즘’을 사용하였다. [2]에서는 Hive의 피어 선택 알고리즘을 개선하여 P2P 통신을 병용함으로써 얻을 수 있는 CDN 비용 절감 효과를 극대화할 수 있게 하였다. 그러나 [2]의 알고리즘은 클라이언트 입장에서 세그먼트 획득 성공률을 최대한 높이기 위해 자신과 통신이 가장 원활한 피어에게 지속적으로 세그먼트를 요청하는 경향이 있기 때문에, 이러한 피어에게 부담을 집중시키는 단점이 있다. 이렇게 한 피어에게 부하가 집중되면, 해당 피어의 자원이 과도하게 사용될 뿐만 아니라, 피어 간 통신이 유료인 경우, 해당 피어의 소유자는 CDN 비용을 절감시키기 위해 자신이 P2P 트래픽 비용을 부담하게 되므로 불만을 갖게 될 수 있다. 따라서 이와 같이 극단적으로 한 피어에게 부담을 지게 하기 보다는 적절한 수준에서 여러 피어에게 부담을 분산시키는 것이 바람직하다.

본 논문에서는 CDN 비용 절감 수준을 적절히 유지하면서도 여러 피어에게 부담을 분산시키는 새로운 피어 선택 알고리즘을 제안하고, 이를 Hive에 구현한 후 실험을 통해 그 효용성을 검증하였다.

본 논문의 구성은 다음과 같다. II장에서는 P2P 통신 병용 DASH 시스템의 개요에 대해 설명하고, III장에서는 제안하는 피어 선택 알고리즘을 [1] 및 [2]의 피어 선택 알고리즘과 비교하여 설명한다. IV장에서는 실험 환경 및 실험 방법을 설명하고 V장에서는 실험 결과를 통해 제안하는 알고리즘의 효용성을 보이며 VI장에서는 결론을 제시한다.

II. P2P 통신 병용 DASH 시스템의 개요

이 절에서는 P2P 통신 병용 DASH 시스템 즉 Hive의 개요에 대해 설명한다.

기본적인 DASH 시스템의 경우, 클라이언트는 CDN 사업자의 미디어 서버로부터 세그먼트들을 직접 획득한다. 물론 미디어 서버와 이로부터 획득 가능한 세그먼트들의 URL은 DASH MPD(Media Presentation Description)(이하 ‘MPD’로 약칭)라고 하는 메타데이터 파일에 적시되어 있으므로 클라이언트는 서비스 제공자가 제공하는 별도의 서

a) 서울시립대학교 일반대학원 전자전기컴퓨터공학과(Department of Electrical and Computer Engineering, Graduate School, University of Seoul)

‡ Corresponding Author : 김용한(Yong Han Kim)

E-mail: yhkim@uos.ac.kr

Tel: +82-2-6490-2330

ORCID:https://orcid.org/0000-0001-9470-6060

※ 이 논문은 2018년도 서울시립대학교 연구년교수 연구비에 의하여 연구되었음.(This work was supported by the 2018 sabbatical year research grant of the University of Seoul.)

· Manuscript received December 23, 2019; Revised January 14, 2020; Accepted January 14, 2020.

버로부터 또는 다른 통신 방법을 통해 이를 먼저 획득하여야 한다. MPD를 제공하는 서버와 미디어 서버는 물리적으로 동일한 서버일 수도 있다.

Hive는 웹기반 미디어 스트리밍 시스템으로서 클라이언트가 콘텐츠 정보를 알려 주는 웹페이지에 접속함으로써 서비스를 받기 시작한다. 이 웹페이지에는 스트리밍 받을 콘텐츠에 상응하는 MDP의 URL과 클라이언트 측의 Hive 기능을 수행하게 하는 자바스크립트가 포함되어 있다. 이 자바스크립트는 WebRTC(Web Real-Time Communication) 표준 API^[3]를 활용하여 구현되어 있다. 따라서 Hive는 WebRTC 표준을 지원하는 웹브라우저에서만 동작한다. 크롬을 비롯한 주요 웹브라우저에서 WebRTC 표준이 지원되고 있다.

그림 1은 Hive의 구성을 보여 준다. Hive는 웹페이지 서버, 미디어 서버, 트래킹(tracking) 서버, 시그널링(signaling) 서버 등의 CDN 사업자가 제공하는 서버들과 여러 클라이언트들로 구성된다. 트래킹 서버는 트래커(tracker)라 불리기도 한다. 본 논문에서는 어떤 한 클라이언트를 기준으로 Hive의 동작을 설명하도록 하며, 이를 다른 클라이언트들과 구분하기 위해 이를 제외한 다른 클라이언트들은 피어라고 부르기로 한다. 따라서 그림 1의 피어 입장에서서는 클라이언트가 자신의 피어에 해당할 수 있다.

클라이언트가 CDN 사업자가 제공하는 웹페이지에 접속하여 거기에 포함된 콘텐츠를 스트리밍하고자 하면 해당 웹페이지에 포함된 자바스크립트에 의해 우선 트래커에 접

속하여 자신을 등록하게 된다. 따라서 동일한 웹페이지의 콘텐츠를 스트리밍 받는 모든 클라이언트들을 트래커가 목록화하여 관리할 수 있다. 이 목록을 ‘피어 목록’이라 부르기로 한다.

CDN 비용을 줄이려면 클라이언트는 미디어 서버 측으로 세그먼트 요청을 하기 전에 먼저 다른 피어에게 해당 세그먼트를 요청하여야 한다. 이를 위해 트래커는 주기적으로 피어 목록에 있는 피어들 중 무작위로 일정 수의 피어들을 선택하여 이들에 대한 정보를 클라이언트에게 제공하고 이 정보를 바탕으로 클라이언트는 일정 수의 피어들과의 P2P 통신 채널을 수립한다. 피어들과의 P2P 통신 상태는 동적으로 변하기 때문에 클라이언트는 P2P 통신이 원활하지 못한 피어는 배제시키고 트래커로부터 주기적으로 제공되는 피어들의 정보를 기반으로 새로운 피어와 P2P 통신 채널을 수립하는 과정을 지속적으로 수행함으로써, 가능한 일정 수의 피어와 P2P 통신 채널이 수립되어 있는 상태를 유지하고자 노력한다. 물론 피어 목록에 등록된 피어의 수가 상기 일정 수 보다 작은 경우에는 상기 일정 수 보다 작은 수의 피어와 P2P 통신 채널을 수립할 수밖에 없을 때도 있다.

보통 클라이언트는 NAT(Network Address Translation)을 통해 인터넷에 연결되는 경우가 많아서 피어의 사설(private) IP 주소를 알 수 없기 때문에 P2P 통신 채널 수립 과정에는 각 피어들의 IP 주소 정보를 별도로 받아서 피어간의 연결 수립 도와주는 시그널링 서버가 필요하다. 시그

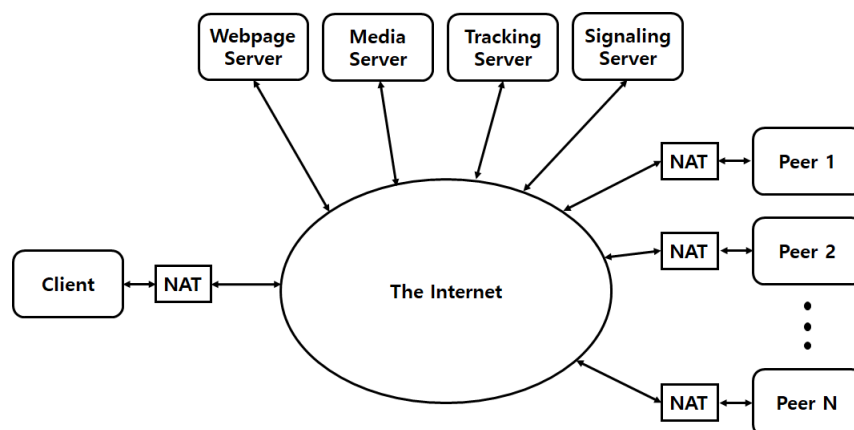


그림 1. Hive의 구성
Fig. 1. The Structure of Hive

널링 서버의 도움을 받아 P2P 통신 채널을 수립하는 과정은 매우 복잡하다. 이에 대한 상세한 내용은 [4]를 참고하기 바란다. 단, WebRTC 표준 API를 사용하는 경우, 이 과정에 대한 구현을 크게 단순화시킬 수 있다.

클라이언트를 포함한 모든 피어들은 새로운 세그먼트를 획득할 때마다, 자신과 P2P 통신 채널이 수립된 피어들에게 해당 세그먼트의 식별자 즉 세그먼트 ID(Identifier)를 알려준다. 어떤 피어를 기준으로 그것과 P2P 통신 채널이 수립되어 있는 피어들의 집합을 ‘이웃(neighbor)’이라 부른다. 모든 피어는 이웃으로부터 전달받은 세그먼트 ID를 ‘세그먼트 인덱스(index)’라 불리는 목록에 추가한다. 물론 어떤 피어가 세그먼트 버퍼로부터 어떤 세그먼트를 삭제하게 되면 이를 이웃에게 통보해 주며 이웃은 자신의 세그먼트 인덱스로부터 해당 세그먼트 ID를 삭제한다. 이러한 과정을 통해 클라이언트는 세그먼트 인덱스에 세그먼트 ID와 그에 해당하는 세그먼트를 보유한 이웃의 정보를 지속적으로 관리하며, 새로운 세그먼트를 요청할 때 세그먼트 인덱스를 참조하여 해당 세그먼트를 보유한 이웃이 있으면 그 중 한 피어에게 세그먼트 요청을 보낸다. 물론 해당 세그먼트를 보유한 이웃이 없으면, 미디어 서버에게 세그먼트 요청을 보낸다.

클라이언트는 스트리밍 서비스를 개시할 때 트래커에 자신을 등록하고 자신에게 IP 패킷을 전달할 수 있는 정보를 시그널링 서버로 보낸다. 이후 일정 수의 피어와 P2P 통신 채널을 수립한 후 미디어 서버로 세그먼트를 요청하기 전에 1회에 한하여 같은 세그먼트를 이 중 한 피어에게 요청한다. 요청이 성공하여 피어로부터 세그먼트를 받게 되면, 해당 세그먼트 크기만큼의 CDN 비용을 절감한 것이 된다. 만약 요청이 실패하면 클라이언트는 미디어 서버로 해당 세그먼트를 요청한다. 이 때 다른 피어에게 다시 한 번 세그먼트를 요청하는 과정은 시행하지 않는다. 이는 피어로부터 세그먼트를 획득하는 과정이 연속적으로 실패할 때 발생할 수 있는 지연시간 누적으로 인해 DASH 플레이어의 미디어 버퍼를 채우는 속도가 느려지는 것을 피하기 위한 것이다. 즉 피어에로의 요청 실패로 인한 지연시간은 1회 실패 분만큼으로 제한된다.

따라서 각 세그먼트에 대해 미디어 서버가 아니라 피어

로부터 이를 획득할 수 있는 기회는 단 한 번뿐이므로, 현재 P2P 통신 채널이 수립되어 있는 피어들 중에서 어느 피어를 선택하여 세그먼트를 요청하느냐에 따라 CDN 비용 절감 효과가 결정된다. CDN 비용 절감 효과에 대한 측도는 (1)과 같은 ‘CDN 비용 절감비’에 의해 측정한다.

$$\text{CDN 비용 절감비 (\%)} = \frac{\text{P2P 통신을 통해 받은 데이터 양}}{\text{받은 데이터 총량}} \times 100 \quad (1)$$

CDN 비용 절감비에 가장 큰 영향을 끼치는 피어 선택 알고리즘에 대해서는 III장에서 설명한다. 본 장에서는 지면의 제한으로 인해 본 논문의 결과를 이해할 수 있을 정도로만 Hive에 대해 개괄적으로 설명하였다. 클라이언트의 프로토콜 스택, 트래커와 피어들 간에 주고받는 메시지, P2P 채널 수립 과정, WebRTC 표준 API를 활용한 P2P 통신 채널 수립 및 통신 과정 등에 대한 보다 더 상세한 설명에 대해서는 [1]~[4]를 참고하기 바란다.

III. 피어 부하 분산을 위한 피어 선택 알고리즘

세그먼트를 요청할 피어를 선택하는 알고리즘으로서 [1]에서 사용한 방법은 클라이언트가 관리하고 있는 세그먼트 인덱스를 참조하여 현재 요청할 세그먼트를 갖고 있는 이웃 중 하나의 피어를 무작위로 선택하는 ‘무작위 피어 선택 알고리즘’이다. 이 알고리즘은 피어들의 부담을 최대한 분산시킨다. 반면 이렇게 선택된 피어들 중에는 통신 상태가 나빠서 세그먼트를 주어진 제한시간 내에 전달하지 못하는 피어들도 있기 때문에 이 알고리즘은 CDN 비용 절감비를 극대화하지는 못한다.

[2]에서는 P2P 통신 이력(history)을 기반으로 세그먼트 전달 가능성이 높은 피어가 높은 우선순위를 갖도록 관리하고, 세그먼트 요청 시에는 우선순위가 가장 높은 피어들 중에서 한 피어를 무작위로 선택하는 ‘이력 기반 피어 선택 알고리즘’을 제안하고 실험을 통해 그 효용성을 보였다. 이 알고리즘은 통신 상태가 좋은 피어의 우선순위를 계속 높게

유지하는 특성이 있기 때문에 세그먼트 요청이 특정 피어에게 집중되는 경향을 보이며 세그먼트 획득 확률이 높다. 이에 따라 CDN 비용 절감비가 극대화되는 특성이 있다.

그림 2는 본 논문에서 [2]의 이력 기반 피어 선택 알고리즘을 개선한 알고리즘을 보여 준다. [2]에서는 우선순위 조정을 위해 ‘세그먼트 획득 속도(segment acquisition speed)’ 측정치에 대한 2개의 문턱값을 사용하고 그 문턱값을 정할 때 다소간 임의로 정하였으나, 그림 2의 개선된 알고리즘에서는 1개의 문턱값만을 사용하며, 이 문턱값은 스트리밍할 콘텐츠의 MPD에 담겨 있는 여러 표현(representation) 중 최고 화질을 갖는 표현의 비트율과 동일하게 설정하도록

하였다. 세그먼트 획득 속도는 하나의 세그먼트를 P2P 통신을 통해 획득하는 동안 다음과 같이 측정된다.

(2)에서 세그먼트 획득 속도의 분모는 세그먼트 전달 시작 시점부터가 아니라 세그먼트 요청 시점부터 세그먼트 전달 완료 시점까지의 소요 시간이므로, 세그먼트의 첫 바이트부터 마지막 바이트까지의 전달에 소요되는 시간으로 세그먼트 크기를 나눈 것으로 정의되는 통상의 전달 속도(transfer speed)에 비해 세그먼트 획득 속도가 더 작다. 한 세그먼트를 P2P 통신을 통해 획득하는 동안 측정한 세그먼트 획득 속도가 문턱값 보다 크다는 것은 해당 콘텐츠의 최고 화질 DASH 표현의 비트율 보다 빠른 속도로 DASH

$$\text{세그먼트 획득 속도} = \frac{\text{세그먼트 크기}}{\text{세그먼트 요청 시점부터 세그먼트 전달 완료 시점까지의 시간}} \quad (2)$$

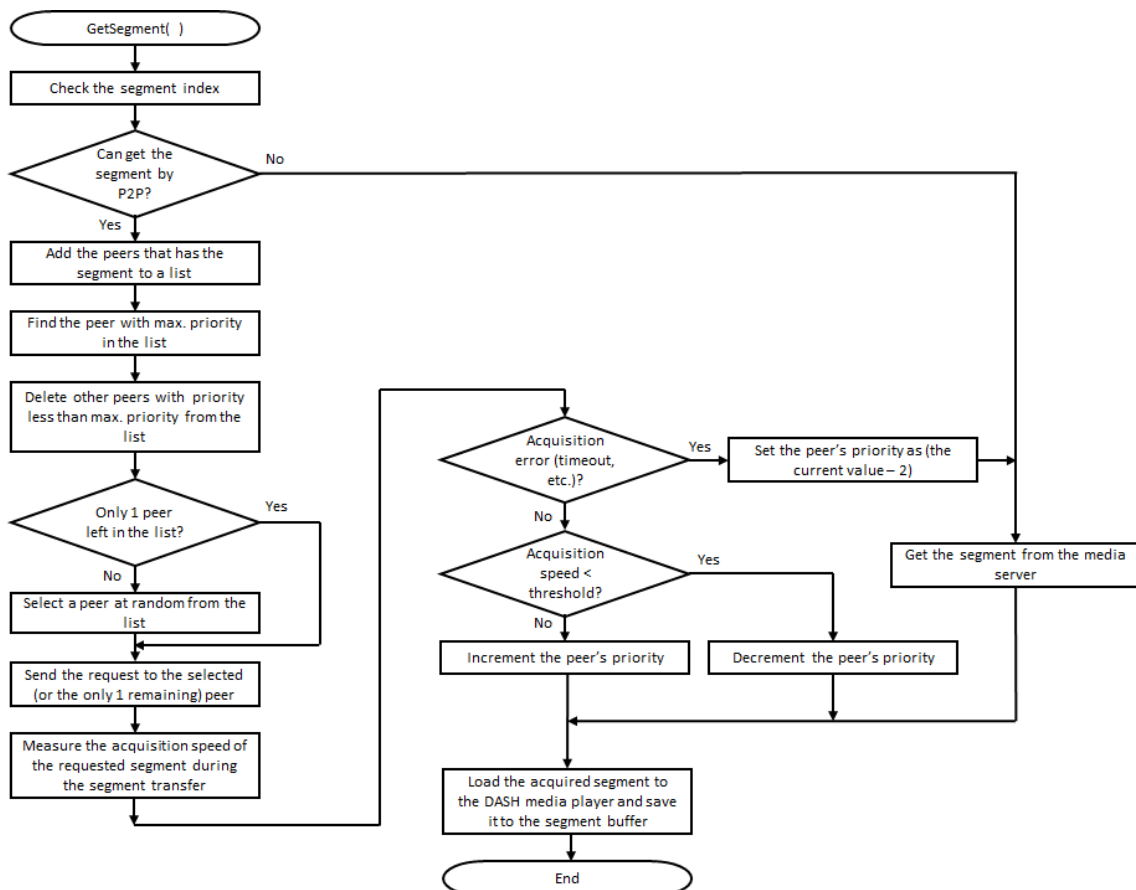


그림 2. 세그먼트 전달 이력 기반 피어 선택 알고리즘의 흐름도

Fig. 2. The flowchart of the peer selection algorithm based on segment transfer history

플레이어의 미디어 버퍼를 채울 수 있다는 것을 의미하므로, 해당 피어의 P2P 통신 채널 상태가 스트리밍을 지원하기에는 충분히 양호하다는 것을 뜻한다.

그림 2에 보인 이력 기반 피어 선택 알고리즘은 어떤 피어를 세그먼트 인덱스에 추가할 때, 우선순위 초기값을 3으로 설정하고, 매번 세그먼트를 P2P 통신을 통해 획득할 때마다 세그먼트 획득 속도를 측정하여 그 결과가 문턱값 보다 크면 그 우선순위를 하나 증가시키고 작으면 하나 감소시킴으로써 피어들의 우선순위를 재설정한다. 만약 세그먼트를 전달 받는 중에 오류가 발생하거나 제한시간이 경과(timeout)하면 해당 피어의 우선순위를 2만큼 감소시키고 P2P 통신을 통해 세그먼트를 얻어오는 것을 포기하고 미디어 서버로부터 세그먼트를 받는다. 이 때 우선순위 결정에 P2P 통신 채널

상태의 최근 단기 이력만을 반영하기 위해 우선순위의 상한과 하한 값을 각기 5와 1로 제한한다. 추후 세그먼트 요청을 할 때, 해당 세그먼트를 가진 피어들 중 우선순위가 가장 높은 피어를 선택하여 요청을 보낸다. 이 때 만약 우선순위 값이 가장 큰 피어들이 여러 개 있으면 이 중 하나를 무작위로 선택하여 요청을 보낸다. 그림 2의 흐름도의 초반에 보였듯이, 물론 세그먼트 인덱스를 참조하여 필요한 세그먼트를 가진 피어를 검색했을 때, 그러한 피어가 하나도 없으면 즉시 미디어 서버로부터 세그먼트를 받는다.

이력 기반 피어 선택 알고리즘이 CDN 비용 절감비를 극대화하는 장점은 있지만, 특정 피어의 자원이 과도하게 소모되고 특정 피어의 P2P 트래픽 비용이 과도하게 증가되는 경우를 허용하기 때문에 이는 사용자의 불만 사항이 될 수

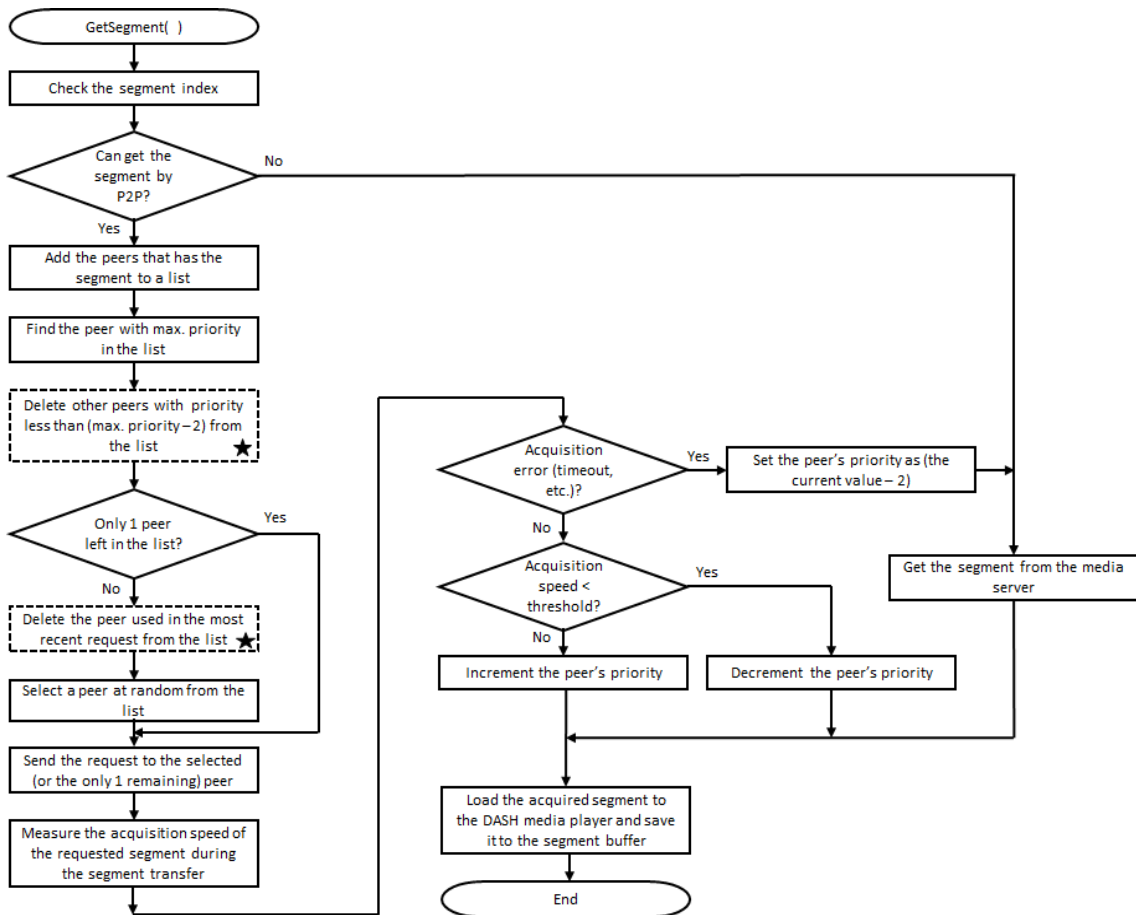


그림 3. 피어 부하 분산을 위한 피어 선택 알고리즘의 흐름도

Fig. 3. The flowchart of the peer selection algorithm for peer load balancing

있다. 따라서 CDN 비용 절감비를 크게 낮아지게 하지 않으면서도 가능한 한 여러 피어에게 부담을 균등하게 분산시키는 알고리즘이 필요하다.

그림 3은 본 논문이 제안하는 ‘피어 부하 분산을 위한 피어 선택 알고리즘’을 보여 준다. 그림 3에서 그림 2와 다르거나 추가된 부분에 대해서는 점선으로 표기하고 별표를 추가로 기입하였다. 그림 3에 보인 알고리즘에서는 우선순위가 가장 높은 피어를 선택하는 것이 아니라 가장 높은 우선순위를 갖는 피어와 그 우선순위 보다 1단계 및 2단계 아래의 우선순위를 갖는 피어들까지 모두 포함하는 리스트를 구성하고 이로부터 무작위로 하나의 피어를 선택하도록 하였다. 이 때 만약 구성된 리스트에 직전의 세그먼트 요청을 받은 피어를 포함하여 여러 피어가 리스트에 포함되어 있으면 직전의 세그먼트 요청을 받은 피어는 리스트로부터 제외함으로써 어떤 한 피어가 연속적으로 2개의 세그먼트 요청을 받게 되는 경우를 가능한 한 피하도록 하였다. 물론 리스트에 남아 있는 피어가 하나뿐이라면 비록 그것이 직전의 세그먼트 요청을 받은 피어라 할지라도 세그먼트 요청을 보내도록 한다. 그림 3의 나머지 부분은 그림 2의 알고리즘과 동일하다.

그림 3의 알고리즘에서 세그먼트 요청을 여러 피어에게 분산시키기 위해 우선순위가 가장 높은 피어를 기준으로 2단계 아래까지의 우선순위를 갖는 피어들을 선택 대상으로 삼은 것은 우선순위의 초기값이 3이고 상한값이 5이기 때문에, 처음으로 세그먼트 인덱스에 등록된 피어에게도 최소 한 번의 세그먼트 요청을 보내어 P2P 통신 채널 상태를 점검할 수 있는 기회를 가질 수 있게 하기 위함이다. 즉 보통 P2P 통신 상태가 양호한 상태로 일정 기간 유지되는 피어의 우선순위는 상한값인 5까지 쉽게 도달하기 때문에, 무작위 선택 과정을 통해 우선순위가 3인 피어까지는 선택될 수 있는 기회를 부여하기 위함이다.

IV. 실험 환경 및 실험 방법

1. 실험 환경

실험을 위해 사용한 웹 브라우저는 현재 WebRTC 표준

API를 가장 잘 지원하고 있는 크롬 브라우저(버전 71.0.3478.80)이다. 실험에 사용된 미디어 콘텐츠는 DASH-IF (Industry Forum)에서 제공하는, 크기 91MB, 재생시간 260s인 Envivio사의 DASH 콘텐츠^[5]이다. 이 DASH 콘텐츠를 안내하는 웹페이지와 [5]로부터 다운로드 받은 해당 콘텐츠를 실험실 내의 웹서버를 통해 제공하도록 하였다. 즉 웹페이지 서버와 미디어 서버는 물리적으로 동일한 서버로 구현하였다. 이 웹페이지에 포함된 자바스크립트는 III장에 설명한 알고리즘을 포함하여 Hive의 클라이언트 측 기능을 수행한다. 피어들의 네트워크 환경을 에뮬레이션하기 위해 NetBalancer(버전 9.12.9)^[6]를 사용하였다. 피어의 수는 클라이언트를 포함하여 10개로 하였는데, 이들은 편의상 모두 한 실험실 내의 인트라넷에 속해 있지만, 이들 간의 P2P 통신 채널의 상태를 에뮬레이션하기 위해 상기 NetBalancer를 모든 피어에 설치하여 사용하였다. 단, 이 실험실은 미디어 서버가 있는 실험실과는 다른 실험실로서 같은 건물 내 3개 층 떨어진 곳에 있다.

콘텐츠 웹페이지의 자바스크립트, 트래커 프로그램, 시그널링 서버 프로그램 등은 Hive 시스템 오픈 소스^[7]의 내용을 수정하여 사용하였다. 단, 제안하는 피어 선택 방법의 성능을 기존 논문 [1] 및 [2]와 같은 조건 하에서 비교하기 위해, 트래커로부터 피어 목록을 전달 받는 주기(15초), 이 때 전달 받는 피어 목록의 최대 피어 수(5개), 각 피어가 P2P 연결을 유지하는 최대 피어의 수(10개), 피어에게 세그먼트를 요청한 이후의 타임아웃 검출용 제한시간(5초), DASH 플레이어의 초기 버퍼링 양(10초) 등 각종 파라미터는 Hive 시스템의 설정과 같게 하였다.

2. 실험 방법

III장에서 설명한 3가지 피어 선택 알고리즘 간의 성능 차이를 측정하기 위한 비교 실험을 시행하였다. 실제 실험에서는 P2P 통신 속도가 느린 피어 수가 늘어남에 따른 성능 변화를 분석할 수 있는 실험을 시행하였다. 하나의 클라이언트는 9개의 피어를 이웃으로 가지며, 각 피어들은 미리 콘텐츠를 플레이함으로써 해당 콘텐츠의 모든 세그먼트들을 갖고 있도록 하였다. 이처럼 피어들이 미리 세그먼트를 갖도록 한 이유는, 피어들이 VOD(Video On Demand)를

시청하며 클라이언트가 피어들 중 가장 늦게 VOD를 시청하는 경우를 실험 상황으로서 상정했기 때문이다. 또 모든 세그먼트를 갖도록 한 이유는 실험의 편의상 모든 실험에서 이웃들의 세그먼트 보유상황이 같도록 하기 위함이다. 실제 상황에서는 클라이언트 피어 보다 늦게 VOD를 시청하는 피어들이 존재할 수 있지만, 이러한 피어들은 클라이언트가 필요로 하는 세그먼트를 제때에 보유하게 될 가능성이 거의 없기 때문에, 클라이언트를 기준으로 볼 때, CDN 비용을 줄이는 데에 기여하지 못한다. 또한 실제 상황에서는 클라이언트보다 VOD 시청을 먼저 시작하였으나, VOD 시청을 완료하지 못한 피어들이 존재할 수 있지만, 이러한 피어들이 세그먼트를 획득하는 시간의 지터(jitter)로 인해 CDN 비용을 절감하는 데에 기여하는 정도가 균일하지 않을 수 있으므로, 실험 결과를 해석할 때의 편의를 위해 그러한 피어들은 실험에 포함시키지 않았다.

‘정속 피어’의 경우, 2017년 기준의 LTE 평균 속도^[8] 즉, 다운로드 속도 15.85MB/s와 업로드 속도 4.05MB/s가 되도록 에뮬레이터를 설정하였고, ‘저속 피어’의 경우, 실험을 위해 임의의 속도를 정하였는데, 다운로드 및 업로드 속도 모두 200KB/s가 되도록 에뮬레이터를 설정하였다. 해당 저속 피어의 속도는 실험에 사용된 Envivio 콘텐츠의 최고 화질 비트율인 375KB/s 즉 III장에서 제시한 알고리즘의 세그먼트 획득 속도에 대한 문턱값 보다 낮은 속도이다. 실제 세그먼트 획득 속도는 상기 다운로드 속도보다 더 작기 때문에, 저속 피어가 선택되는 경우에는 측정된 세그먼트 획득 속도가 문턱값 보다 작아서 피어의 우선순위가 1만큼 감소하거나 제한시간 내에 세그먼트 전달에 실패하며 피어의 우선순위가 2만큼 감소하게 된다.

실험에서는 9개의 이웃들 중 각기 0, 2, 4, 6, 8개가 저속 피어가 되도록 네트워크 에뮬레이터를 설정한 5가지 실험을 시행하되, 각 실험은 3회 시행하여 CDN 비용 절감비의 평균값을 산출하였다.

V. 실험 결과

그림 4는 III장에서 설명한 3가지 피어 선택 알고리즘의 비교 실험 결과를 보여준다. 그림 4에서 무작위 피어 선택

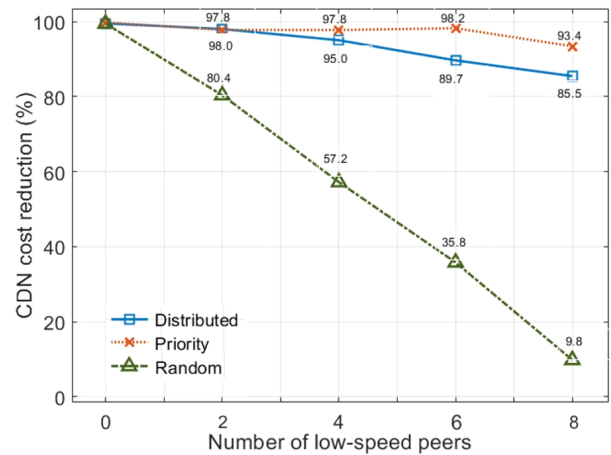


그림 4. 3가지 피어 선택 알고리즘 간의 CDN 비용 절감비 실험 결과
Fig. 4. The experimental results of the CDN cost reduction for the 3 peer selection algorithms

방법, 이력 기반 피어 선택 방법, 제안하는 피어 부하 분산을 위한 피어 선택 방법의 결과는 각기 ‘Random’, ‘Priority’, ‘Distributed’로 표기하였다.

그림 4에서 무작위 피어 선택 알고리즘에 의한 CDN 비용 절감비는 저속 피어 수에 대체로 반비례함을 볼 수 있다. 이는 무작위 선택으로 인해 세그먼트를 P2P 통신에 할당된 제한 시간 내에 전달하지 못하는 저속 피어가 선택되기 때문이다. 이 경우 클라이언트는 미디어 서버 측으로부터 해당 세그먼트를 가져오며 이로 인해 CDN 비용 절감비가 감소된다. 또한 대체로 저속 피어 수에 비례하여 저속 피어가 선택되는 횟수가 증가하기 때문에, CDN 비용 절감비 또한 저속 이웃 수가 증가함에 따라 비례적으로 급감하게 된다.

그림 4로부터 이력 기반 피어 선택 알고리즘의 경우, 매우 높은 CDN 비용 절감비를 얻을 수 있음 관찰할 수 있다. 이 때 이력 기반 피어 선택 방법 또한 저속 피어 수가 증가함에 따라 약간의 CDN 비용 절감비 감소를 확인할 수 있다. 이는 스트리밍 시작 직후에는 세그먼트 전송 이력이 축적되지 않았기 때문이다. 이에 따라 정속 피어를 찾을 때까지 CDN 비용 절감비가 다소 감소하며 저속 피어 수가 증가함에 따라 정속 피어를 찾는 시점까지 보다 더 많은 전송 이력이 필요하므로 CDN 비용 절감비 감소량이 증가한다. 그럼에도 불구하고 이는 무작위 피어 선택 방법에 비해 매

우 뛰어난 성능이다. 저속 피어 수가 8개일 때, 즉 정속 피어가 하나뿐일 때, 무작위 피어 선택 방법은 약 10%, 이력 기반 피어 선택 방법은 약 93%의 CDN 비용 절감비를 달성함을 관찰할 수 있다.

그림 4로부터, 피어 부하 분산을 위한 피어 선택 알고리즘의 경우, 무작위 피어 선택 알고리즘에 비해서는 매우 높은 CDN 비용 절감비를 얻을 수 있음을 관찰할 수 있다. 하지만 이력 기반 피어 선택 방법 보다는 약간 낮은 CDN 비용 절감비를 얻음을 확인할 수 있다. 저속 피어 수가 8개일 때 무작위 피어 선택 방법에 비해서는 약 75% 높고 이력 기반 피어 선택 방법에 비해서는 약 8% 낮은 CDN 비용 절감비를 얻음을 관찰할 수 있다.

또한 저속 피어 수가 늘어남에 피어 부하 분산을 위한 피어 선택 알고리즘의 CDN 비용 절감비가 감소하며 이력 기반 피어 선택 알고리즘과의 차이가 또한 커짐을 확인할 수 있다. 이는 하나의 정속 피어만을 찾고 해당 피어에게 요청을 집중시키는 이력 기반 피어 선택 알고리즘과 달리 세그먼트 요청을 분산시키기 위해 모든 정속 피어를 찾기 때문에 발생한다. 모든 피어에게 적어도 한 번의 요청을 보낸 후 세그먼트 획득 속도를 측정하여야 하고 이 과정에서 저속 피어 수가 증가함에 따라 세그먼트 요청 오류에 따른 CDN 비용 절감비 감소가 초래된다. 하지만 이러한 CDN 비용 절감비 감소 현상은 스트리밍 초기에 피어들의 세그먼트 획득 속도를 측정할 때만 발생하므로 스트리밍할 콘텐츠의 플레이 시간이 길어질수록 두 방법의 CDN 비용 절감비 차이는 줄어들 것이다. 따라서 피어 부하 분산을 위한 피어 선택 방법은 콘텐츠의 길이가 충분히 길다면 이력 기반 피어 선택 방식에 비해 크게 차이가 나지 않는 CDN 비용 절감비를 얻을 수 있으며 세그먼트 요청도 분산시킬 수 있는 알고리즘이다.

그림 5와 그림 6은 피어 부하 분산을 위한 피어 선택 알고리즘에 대한 실험에서 저속 피어가 4개일 때의 피어 부하 분산 결과이다. 그림 5는 콘텐츠의 모든 세그먼트에 대해 요청한 곳을 보여주는 맵(map)인데, 어떤 과정을 거쳐 정속 피어를 찾게 되는지를 보여 주는 그림이다. 그림 5는 9개의 이웃들 중, 저속 피어가 4개일 때의 첫 비디오 세그먼트(0번 세그먼트)부터 마지막 비디오 세그먼트(64번 세그먼트)까지 순차적으로 요청한 곳을 보여 준다. 그림 5의 가로축

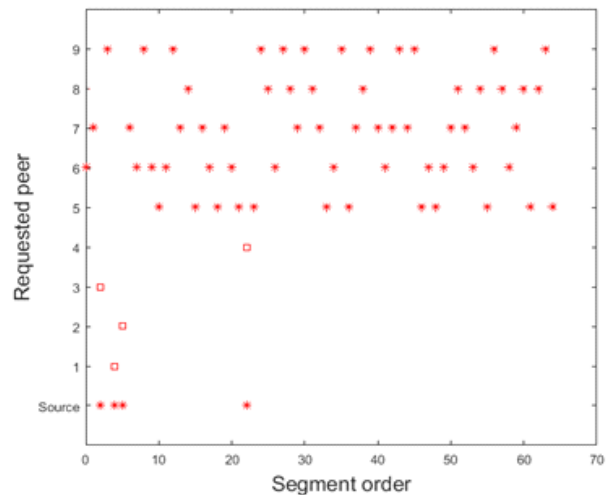


그림 5. 피어 부하 분산을 위한 피어 선택 알고리즘을 사용한 경우, 저속 피어가 4개 있을 때, 세그먼트 별 선택된 피어의 맵(map) ('Source'는 미디어 서버를 나타냄)

Fig. 5. The map of selected peers for each segment when there are 4 low-speed peers and the peer selection algorithm for peer load balancing is used. ('Source' indicates media server.)

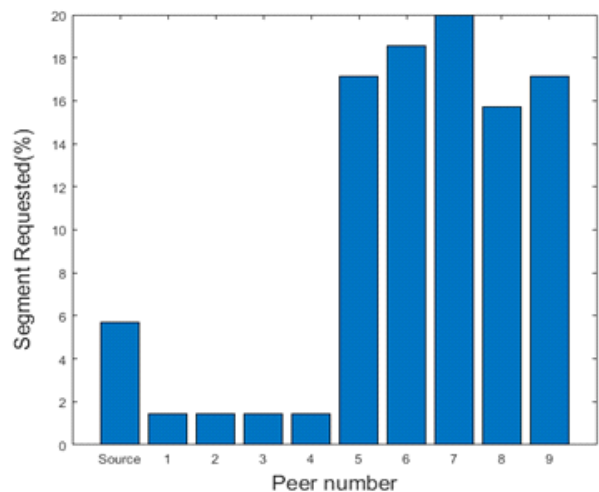


그림 6. 피어 부하 분산을 위한 피어 선택 알고리즘을 사용한 경우, 선택된 피어의 백분율 ('Source'는 미디어 서버를 나타냄)

Fig. 6. The percentage of selected peers for each segment when there are 4 low-speed peers and the peer selection algorithm for peer load balancing is used. ('Source' indicates media server.)

의 1번부터 4번까지 피어는 저속 피어이며, 5번부터 9번까지 피어는 정속 피어이다. 가장 아래에 'source'는 미디어 서버 측으로 세그먼트를 요청한 경우이다. 세그먼트 전달

에 성공한 요청은 ‘*’로, 실패한 요청은 ‘□’로 표기하였다. 그림 5로부터 저속 피어에게도 최소 1회의 세그먼트 요청을 보내지만 이 요청은 세그먼트 획득 속도가 느린 관계로 결국 오류를 초래하여 해당 피어는 그 우선순위가 2만큼 저하되고 이후 피어 선택 과정에서 배제됨을 알 수 있다.

그림 6은 그림 5의 결과를 얻은 실험 결과를 바탕으로 각 피어 및 미디어 서버에 대한 세그먼트 요청 횟수를 백분율로 나타낸 그림이다. 그림 6을 통해 저속인 피어 1부터 피어 4까지의 피어들을 제외한 나머지 피어들 즉 정속 피어들 간에 세그먼트 요청이 분산됨을 확인할 수 있다.

VI. 결 론

본 논문에서는 P2P 통신 채널을 함께 사용하여 피어들 간에 이미 수신한 DASH 세그먼트들을 공유할 수 있게 하는 P2P 통신 병용 DASH 시스템에서 CDN 사업자의 트래픽 비용을 크게 줄이면서도 특정 피어에게 과도한 부하를 지게 하지 않는 새로운 피어 선택 알고리즘을 제안하고 실험을 통해 그 효용을 보였다. 본 논문에서 제안한 새로운 알고리즘은 CDN 사업자의 트래픽 비용 절감 효과를 극대화하는 기존 알고리즘이 특정 피어의 부담을 과도하게 크게 하는 단점을 적은 비용의 상호 절충(trade-off)을 통해 극복할 수 있게 하였다. 또한 기존 알고리즘과 새로운 알고리즘에서 공통으로 사용된 문턱값을 기존의 것 보다 더 합리적인 값으로 대체하였다.

본 논문에서는 CDN 미디어 서버, 웹페이지 서버, 트래커, 시그널링 서버와 클라이언트를 포함한 피어들이 모두 한 건물 내에 있는 상황에서 네트워크 에뮬레이터를 사용하여 실험을 시행하였고, 피어들의 수도 10개로 제한하였다. 향후 여러 서버와 여러 피어들이 지리적으로 멀리 떨어진 실제 환경에서 많은 수의 피어들을 대상으로 하는 대규모의 실험을 시행하여 본 논문에서 제안한 알고리즘의 성능을 검증할 필요가 있다.

참 고 문 헌 (References)

- [1] R. Roverso and M. Höggqvist, “Hive.js: Browser-Based Distributed Caching for Adaptive Video Streaming,” *2014 IEEE International Symposium on Multimedia*, pp.143-146, 2014.
- [2] J. H. Seo, S. H. Choi, S. J. Kim, J. Y. Jeon, and Y. H. Kim, “A transport-history-based peer selection algorithm for P2P-assisted DASH systems based on WebRTC,” *Journal of Broadcast Engineering*, Vol.24, No.2, pp.251-263, Mar. 2019.
- [3] WebRTC 1.0: Real-time Communication between Browsers, <https://w3c.github.io/webrtc-pc/> (accessed Sept. 2018).
- [4] S. H. Choi, *A study on traffic reduction of Web-based video streaming CDN using WebRTC-based P2P data sharing*, Master’s thesis of University of Seoul, Seoul, Rep. of Korea, Dec. 2018.
- [5] <http://dash.edgesuite.net/envivio/dashpr/clear/> (accessed Dec. 12, 2017).
- [6] Netbalancer, <https://netbalancer.com/> (accessed June 24, 2019).
- [7] <https://github.com/Peerlism/hive.js> (accessed Dec. 12, 2017).
- [8] MSIT (Ministry of Science and Information Technology) Press Release, “Quality assessment results of 2017 telecommunication services,” <https://www.msit.go.kr/web/msipContents/contentsView.do?cateId=mssw311&artId=1371275>, (accessed Aug. 26, 2018).

저 자 소 개



서 주 호

- 2017년 8월 : 서울시립대학교 물리학과 학사
- 2017년 9월 ~ 2019년 8월 : 서울시립대학교 전자전기컴퓨터공학과 석사
- ORCID : <https://orcid.org/0000-0002-7140-0816>
- 주관심분야 : 영상통신 및 시스템, P2P 통신, 적응형 스트리밍

저 자 소 개



김 용 한

- 1982년 2월 : 서울대학교 제어계측공학과 (공학사)
- 1984년 2월 : 서울대학교 대학원 제어계측공학과 (공학석사)
- 1990년 12월 : 미국 렌슬리어공대(Rensselaer Polytechnic Institute, RPI) 전기컴퓨터시스템공학과 (Ph.D.)
- 1984년 3월 ~ 1996년 3월 : 한국전자통신연구원 책임연구원(최종)
- 1991년 10월 ~ 1992년 9월 : 일본 NTT 휴먼인터페이스연구소 객원연구원
- 1996년 3월 ~ 현재 : 서울시립대학교 전자전기컴퓨터공학부 교수
- 2017년 1월 ~ 2017년 12월 : 한국방송미디어공학회 회장
- 2014년 1월 ~ 현재 : 미래방송미디어표준포럼(구 차세대방송표준포럼) 의장
- 2000년 1월 ~ 현재 : MPEG뉴미디어포럼(구 MPEG포럼) 운영위원/자문위원
- ORCID : <https://orcid.org/0000-0001-9470-6060>
- 주관심분야 : 영상통신, 디지털 방송, 멀티미디어 부호화 및 전송