

# P2P 협업 및 사용자 콘텐츠 이용 정보 기반의 네트워크 프로토콜 설계

## Design of Network Protocol based on P2P Collaboration and User's Content Using Information

남 의 석\*  
(Eui-Seok Nahm)

**Abstract** - In these days, the big-size and high resolution multimedia file is widely used through networks. To transfer and service effectively, the internet network technology is necessary to substitute broadcasting. Normally Content Delivery Network(CDN) is widely used in conventional internet for multimedia services. But it has a small bandwidth to service. So to solve this problems, many researchers have suggest the protocol for download, content distribution/saving, server synchronization, caching, pushing rate, and streaming etc. But all of these has some defects like low resolution, packets loss and delay, real application implementations etc. So, this paper suggests a new method of network protocol based on P2P collaboration and user's content using information. And it evaluated the performance of suggested method. As the results, it showed the effectiveness of 4 performances indices : download speed, decreasing rate of connected user in same time, adaptive hit ratio, traffic decreasing rate.

**Key Words** : P2P, Network, Protocol, CDN, Caching

### 1. 서 론

오늘날 VR, UHD 등과 같은 고해상도 대용량 멀티미디어 파일들이 네트워크를 통하여 많이 확산되고 있는 추세이다. 이러한 대용량 파일의 효율적인 전송을 위해서는 방송 네트워크를 넘어 인터넷 기반으로 대용량 파일들을 서비스 할 수 있는 기술들이 필수적이다. 기존 인터넷에서는 이러한 대용량 멀티미디어 전송을 위해 Content Delivery Network(CDN) 등을 이용하고 있으나 대용량 멀티미디어 전송을 위한 대역폭을 지원하는 데는 한계가 있다. 따라서, 대용량 멀티미디어 파일을 효율적으로 전송하기 위해서는 대용량 멀티미디어 파일 다운로드에 특화된 프로토콜 기술, 콘텐츠 분산 저장/관리를 위한 서버 및 동기화 기술, 콘텐츠 전송을 위한 스트리밍 기술 등이 필요하다.

프로토콜 기술의 경우, 기존 인터넷에서는 대용량의 멀티미디어 파일을 콘텐츠를 전송하는 경우 패킷 지연/손실 등의 이유로 화질 저하가 발생할 수 있고, 또한 기존 미국의 Akamai 솔루션, Limelight 솔루션, 국내의 CDNetworks 솔루션의 경우도 충분한 대역폭을 제공하지 못하는 실정이다. 또한 서버 기술의 경우도 사용자 수에 비례한 서버 부하의 증가의 문제가 있다.

대용량 멀티미디어 파일을 전송을 위한 기존 기술로는 파일의 전송시 pushing 비율을 조절하는 방식에 제안되었다[1]. 그러나 이 방법은 pushing의 전략이 파일 콘텐츠의 접근 패턴, 네트워크 연결성 등이 고려되지 않은 문제점이 있다. 또한 비디오 파일을

여러 개의 작은 콘텐츠로 분할하여 사용자들에게 pushing 하는 기법도 제안되었다. 이 방법은 콘텐츠 분할 방식에 따라 성능이 좌우되는 문제점이 있다[2]. BitTorrent에서는 다수의 콘텐츠를 다운받을 때, 협업자들의 인센티브 제공 및 대역폭 할당 기법들도 연구되었다[3,4,5]. 또한, 최근의 연구에서는 사용자들의 사회적 관계를 고려한 콘텐츠 전달 가속화 기법, 서비스 이용자의 사회적 관계 정보를 이용한 콘텐츠 전달 경로와 방법을 결정 기법들도 부분적인 문제 해결을 위한 방법으로 제안되었다[6,7]. 그밖에 대역폭 변화에 따른 화질 조절, 서브 스트림 인코딩, 캐싱 등의 방법들도 제시되었다. 이러한 방식들은 특정한 환경에 한정적으로 적용되는 것으로 의미는 있으나 서비스와 환경 예측이 어려운 실제 서비스에서는 한계가 있다[12].

따라서, 본 논문에서는 1) P2P 협력 기반 사용자들의 네트워크 동안 협력 수행 및 서버 부하에 따른 선택적 사용자 자원 적응 기술과 2) 사용자의 콘텐츠 접근 패턴, 네트워크 연결성, 파일 콘텐츠 특성 등의 정보를 고려한 device-to-device 전송 기술들을 제안한다. 제안된 기술의 검증은 위하여 평균 콘텐츠 다운로드 속도, 콘텐츠별 동시 접속 수 감소율, 적응형 캐시 적중 비율, 스트리밍 서버 트래픽 감소율을 시험한 결과를 통하여 제안된 기법의 타당성을 검증한다.

### 2. 대용량 CDN 전송 시스템 설계

#### 2.1 P2P 협업 네트워크

본 논문에서는 비디오 스트리밍 서비스에 특화된 P2P 프로토콜 기반의 프로토콜을 구현한다. 기존의 파일 공유에 사용되는

\* Corresponding Author : Dept. of Aviation and IT Convergence,  
Far East University, Korea.  
E-mail: nahmes@kdu.ac.kr

Received : December 26, 2016; Accepted : February 16, 2017

순수 P2P 프로토콜의 문제점은 파일의 희소성에 기반하여 다운로드 받을 조각을 결정하므로, 순서대로 조각을 다운로드 받을 수 없는 단점이 있다. 따라서 다운로드 속도는 높아지지만, 실제로 파일의 다운로드가 끝나기 전까지는 비디오를 재생할 수 없다는 문제가 있다. 본 논문에서는 이를 해결하기 위해서는 순차적으로 조각을 전송하면서도 다운로드 성능을 높일 수 있는 프로토콜 설계 구현하였다. 본 논문의 비디오 스트리밍 프로토콜은 1) 기존의 일반적 P2P 프로토콜과 같이 사용자간 협업 네트워크를 구성하여 고용량 콘텐츠의 전송을 가능하게 하며, 2) 연속적이고 순차적으로 데이터를 소비하는 비디오 스트리밍 서비스의 데이터 소비 형태에 부합하고, 3) 단말의 자원을 효율적으로 이용하여 콘텐츠 서비스 제공자의 네트워크 자원 이용을 최소화할 수 있도록 설계하였다. 대용량 비디오 스트리밍을 위한 협력 네트워크 프로토콜의 구조는 그림 1과 같다[12].

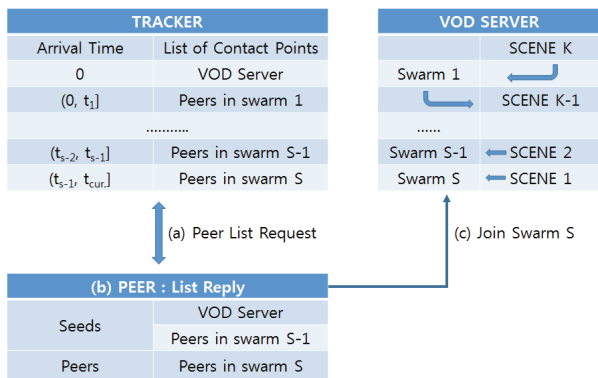


그림 1 협력 기반의 스트리밍 프로토콜  
Fig. 1 Streaming Protocol based Cooperation

본 논문의 프로토콜에서는 대용량 비디오를 균등한 크기의 장면(Scene)으로 쪼개며, 사용자의 그룹을 스웜(Swarm)이라는 단위로 쪼개서 관리한다. 장면은 다시 전송 및 에러 수정의 효율성을 위해 조각(chunk)로 쪼개져서 전송된다.

(a) 사용자(Peer)는 특정 멀티미디어 콘텐츠를 시청하기 위해 트래커(Tracker) 서버에 다른 사용자의 리스트를 요청하고 트래커 서버는 운영의 편리성 및 서비스의 안정성을 위해서 당사에서 직접 로컬에서 운영함. 또한 트래커에서는 사용자들의 시스템 사용 패턴을 확인하기 위하여 각종 기록을 로그로 저장한다.

(b) 트래커 서버는 비디오 콘텐츠 서버와 사용자가 비디오 다운로드를 위해 협력하게 될 다른 사용자인 Peer 클라이언트의 리스트를 보내준다. 이 때, 트래커 서버는 VoD Server 서버의 부하를 고려한 다음, 서버가 단독으로 서비스를 제공해도 충분한 상황에서는 안정적인 서비스를 위해서 서버의 주소만을 사용자에게 넘겨준다. 만약 서버의 부하가 큰 경우에는 서버와 같은 장면을 시청하고 있는 다른 사용자의 리스트를 사용자에게 넘겨준다. 비디오 서버의 경우, 사용자와 가까이 배치되는 것이 서비스 품질 면에서 유리하며, 콘텐츠의 숫자가 지속적으로 증가할 것을 고려하여 클라우드 상에 배치하여 운영한다[12].

(c) 사용자는 트래커가 알려준 스웜 그룹에 가입을 하고, 비디오 콘텐츠를 서로 공유하면서 시청을 시작한다. 이 때 그룹 내에서 파일 공유를 위해서는 빠른 다운로드를 위해 제안 프로토콜을 사용한다. 즉, 순차적인 전송을 위해서 장면을 쪼개 대신, 장면 내에서는 제안 방식으로 다운로드 성능을 높이고자 한다. 그룹 내에서 특정 데이터 조각이 존재하지 않는 경우에는 비디오 서버를 통해서 데이터 조각을 다운로드 할 수 있다. 이를 위해서 비디오 서버도 사용자 협력 프로토콜의 일부로서 동작한다.

일정 시간이 지난 뒤에 도착하는 사용자는 새로운 스웜 그룹에 가입하게 된다. 이러한 그룹 멤버십 관리는 트래커 서버를 통해서 이루어진다. 따라서 시간이 지남에 따라 스웜 그룹은 중첩된 구조를 가지게 된다. 선행 그룹은 후행 그룹보다 많은 비디오 콘텐츠를 소비하였으며, 따라서 후행 그룹이 원하는 데이터를 가지고 있는 그룹이 된다. 선행 그룹은 여유가 있는 경우 후행 그룹을 도와서 전체적인 성능 향상을 가져올 수 있다. 이러한 중첩 구조를 통해서 다운로드 효율성과 전송의 순차성을 동시에 만족시킬 수 있다. 그림 2는 그룹간의 협력 동작 과정을 보여준다.

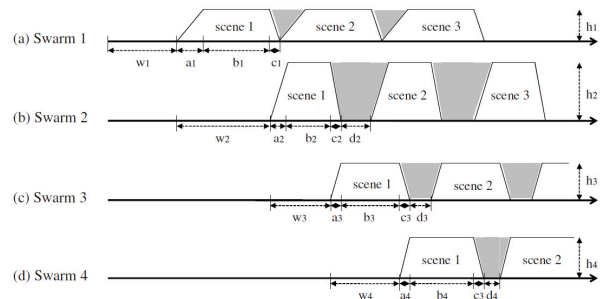


그림 2 사용자 협력 프로토콜 동작  
Fig. 2 User Cooperation Protocol Operation

스웜 그룹1은 가장 먼저 비디오를 시청하는 그룹이다. 사각형의 기울기는 다운로드 속도를 나타낸다. 그룹 내의 사용자들은 하나의 장면을 동시에 다운로드 받고, 다운로드 받은 데이터를 서로 공유함으로써 다운로드 속도를 높인다. 또한 서버 상에서의 트래픽 전송량을 줄일 수 있다. 스웜1은 장면1을 먼저 다운로드 한 뒤 시청하며, 이 때 후발 그룹인 스웜2는 전 그룹인 스웜1으로부터 콘텐츠를 제공받아서 그룹 내에서 공유한다. 후발 그룹인 스웜2에 속한 사용자들은 스웜1에 속한 사용자보다 이전 영상을 시청하는 상황이다. 그림 2의 회색부분은 해당 스웜이 후발 스웜에게 데이터를 “공유”해주는 양을 나타낸다. 이 때, 그룹 내에서 남는 네트워크 자원을 활용하여 후발 그룹을 도와주는 형태로서, 네트워크 유휴자원을 활용하여 다운로드 속도를 높일 수 있다. 마찬가지로 스웜2는 장면1을 시청하게 되고, 스웜3에 장면1의 데이터를 “공유”해 준다. 그림 2에서 볼 수 있듯이 이전 그룹에서 데이터를 많이 공유받게 되면, 다운로드 속도 (e.g., h1, h2, h3, h4)가 높아질 수 있다. 안정적인 속도 제공을 위해서 비디오 서버가 참여하게 되며, 속도 보장을 위해서 참여하는 비디오 서버의 숫자를 증가시킬 수 있다. 이러한 결정은 트래커 서버가 수행

한다.

## 2.2 비디오 콘텐츠 캐싱 및 분산 저장 알고리즘

기존 YouTube, IPTV 스트리밍 서비스 이용자들의 미디어 콘텐츠 소비 행태를 분석한 논문 등의 자료에 따르면 (M. Char, et al, I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System, ACM IMC 2007), YouTube, 다음 동영상과 같은 대부분의 콘텐츠의 인기도는 Zipf's Law를 따르게 되어있음. 즉, 콘텐츠 요청의 대다수는 인기 있는 상위 콘텐츠 몇 개에 집중되며, 대부분의 콘텐츠에 대한 요청은 그 비율이 굉장히 낮게 나타난다. 따라서 캐싱 및 분산 저장을 결정할 때, 전체 콘텐츠에 대한 캐싱 및 분산 저장을 고려하지 않고, 인기 있는 상위 콘텐츠에 대해서만 캐싱 및 분산 저장을 수행하면 된다.

실제 운영상에서 콘텐츠의 특성 및 인기도 등의 지표를 빠르게 판단하여 캐싱을 수행하기 위해서는 최소의 오버헤드를 통해서 인기도를 파악하는 것이 중요하다. 장기적으로는 로그 분석을 통해서 인기도를 파악하고, 단기적으로는 최소의 오버헤드를 통해서 인기도를 파악하는 두 가지 전략을 사용해야 한다. 본 논문에서는 이를 위해 최단시간 안에 인기도를 파악하기 위해 먼저 콘텐츠에 대한 요청 횟수를 기반으로 캐싱할 콘텐츠를 결정하는 알고리즘을 개발하여 적용하였다. 즉, 콘텐츠에 대한 요청 횟수를 기록하고 있다가, 일정 Threshold를 넘으면 캐싱 여부를 결정하는 알고리즘을 결정하여 단기적으로 사용한다. 장기적으로는 트래커 및 각 Peer 클라이언트에서 저장 로그를 분석하여, 다운로드 요청 통계를 제공한 뒤에 상위 10개의 콘텐츠에 대해서 캐싱 및 분산 저장을 수행하도록 알고리즘을 수정한다.

따라서, 본 논문에서는 인기 있는 콘텐츠를 캐싱/분산 저장하기 위한 간단한 알고리즘에 의해 콘텐츠를 캐싱하고 앞 절의 P2P 협업 네트워크 프로토콜과 연동하도록 설계하였다. 콘텐츠 캐싱을 위한 부가적인 정보의 유지를 최소한으로 하기 위해서 콘텐츠에 대한 요청 횟수를 캐싱의 지표로 삼았으며, 콘텐츠를 전송하는 네트워크 프로토콜과 밀접하게 연관되어 동작할 수 있도록 캐시 서버가 Peer로 동작하도록 프로토콜을 수정하였다. 캐싱을 사용하는 경우 네트워크 내에 비디오 캐시 서버와 같은 저장소가 널리 퍼져있을 때에는 효과가 더욱 높게 나타난다.

## 3. 시뮬레이션 결과

### 3.1 실험 환경

본 논문의 성능 검증을 위해서는 대규모 사용자가 접속하는 환경에서 실험할 필요가 있다. 소규모 사용자가 접속하는 환경을 에뮬레이션하는 경우에는 로컬 네트워크 환경에서 실험용 장비를 연결한 뒤에 성능을 검증할 수 있으며, 이 경우 실험 환경을 완벽하게 제어할 수 있다는 장점이 있다. 그러나 1) 실험에 참여하는 노드의 숫자가 적다는 점, 2) 실험 네트워크 환경이 실제 환경과 다르다는 점, 3) 외부 트래픽의 유입이 불가능하다는 점 등

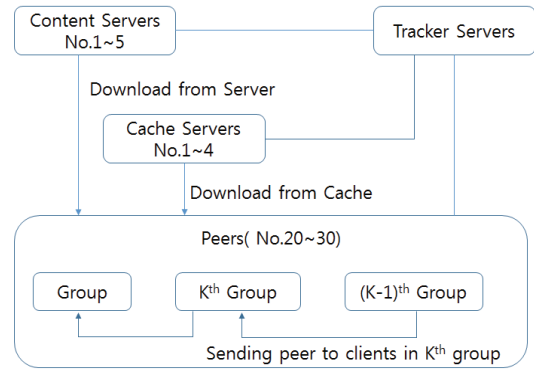


그림 3 Environment 실험 환경

Fig. 3 Environment of Test

의 문제가 발생한다. 따라서 기술의 성능을 실제와 유사한 환경에서 검증하는 것이 반드시 필요하며, 최근 널리 사용되는 클라우드 환경을 활용하면 실제 환경과 가장 유사한 상황에서 성능 평가가 가능하다. 이러한 이유 때문에 그림 3과 같은 클라우드 환경에서 본 논문의 비디오 스트리밍 프로토콜의 성능 검증을 진행하였다.

클라우드의 가상 서버를 임대하여 서비스에 접속하는 사용자를 에뮬레이션 하였으며, 접속하는 사용자의 숫자는 최대 30개까지 사용하였다. 대규모 사용자가 접속하는 환경을 에뮬레이션하기 위해서는 더 많은 숫자의 서버가 필요하였으나, 1) 프로토콜의 특성 상 동시에 접속이 활성화되어 다운로드에 참여하는 Peer의 숫자가 5개 내외인 점, 2) 클라우드 서비스의 특성 상 동시에 30대 이상을 제어하기 어려운 문제가 존재하여, 최대 30개의 사용자를 사용하여 실험을 수행하였다. 그러나 앞서 언급한 대로 프로토콜의 특성상 동시에 다운로드가 약 5개 내외의 Peer로부터 진행되므로, 실제 환경에서의 성능과는 큰 차이가 없을 것으로 기대된다.

대용량 고품질 콘텐츠를 보관하는 콘텐츠 서버 및 콘텐츠 캐시 서버는 1개에서 5개까지의 가상 인스턴스를 사용하였고, 콘텐츠를 다운로드하는 Peer 클라이언트 서버는 20개~30개의 인스턴스를 사용하였다. 로그 수집 및 분석을 위한 서버는 1개의 인스턴스를 사용하여 총 36개의 인스턴스를 생성하여 실험을 진행하였다. 콘텐츠 서버는 비디오 스트리밍 서비스를 위해서 사업자가 유지/관리하는 서버로 클라이언트가 요청한 콘텐츠를 직접 전송한다. 서비스를 위해서 최소 1대의 콘텐츠 서버가 필요하며, 콘텐츠 서버의 개수가 증가함에 따라 성능이 좋아지는 것을 확인하였음. 따라서, 실험에 따라 콘텐츠 서버를 1대에서 5대로 조절하면서 성능을 측정하였다. 성능 향상을 위해서 콘텐츠 서버의 대용량 콘텐츠를 임시로 저장하기 위한 캐시 서버를 운영하였다. 캐시 서버는 콘텐츠 서버로부터 받은 파일 중 클라이언트로부터 요청이 온 Chunk를 전송하여 콘텐츠 서버의 트래픽을 분산하는 역할을 한다. 캐시 서버의 숫자에 따른 성능 향상을 확인하기 위해서 실험에서는 상황에 맞춰서 캐시 서버를 1대에서 5대까지 변화시키면서 성능을 측정하였다. 실험에 사용되는 비디오 콘텐츠는 실제 약 16GB의 콘텐츠를 이용하여 수행하였다. 콘텐츠는

512KB의 Chunk단위로 분할되어 있으며, Peer 클라이언트는 콘텐츠 서버 혹은 다른 Peer 클라이언트로 Chunk 요청을 보내고, 요청을 받은 인스턴스가 해당 Chunk 부분을 전송한다. 따라서, 한 클라이언트는 한 번의 실험에서 약 3만 번의 요청을 하게 되고, 전체 실험 규모로 볼 때 총 90만 번 (3만 \* 30개 클라이언트)의 요청이 수행됨. 결론적으로, 해당 실험은 성능 검증을 위해서 충분한 수의 콘텐츠 요청이 존재하도록 설계되었다.

실험을 통해 평균 콘텐츠 다운로드 속도 (Mbps), 콘텐츠 서버의 동시 접속자 수, 캐시 트래픽 비율, 콘텐츠 서버의 트래픽 제공양 등을 측정하였다.

### 3.2 평균 다운로드 속도

본 논문의 대용량 콘텐츠를 지원하기 위한 비디오 스트리밍 네트워크 프로토콜의 성능을 검증하기 위해서 먼저 평균 다운로드 속도를 측정하였다. 모든 클라이언트에 대해서 다운로드의 시작부터 완료될 때까지의 평균 다운로드 속도를 측정하였다. 비디오 콘텐츠의 스트리밍 서비스를 원활하게 지원하기 위해서는 20Mbps 이상의 다운로드 속도 이상이 보장되어야 하며, 비디오 코덱의 압축률 등을 고려하여도 최소 10Mbps 이상의 다운로드 속도를 제공해야 한다. 앞 절의 실험 환경에서 대용량 비디오 콘텐츠의 다운로드 속도를 측정하였다. 콘텐츠 서버가 1대 존재할 때의 Peer 클라이언트의 평균 다운로드 속도를 나타내는 것으로 하였다. 모든 Peer 클라이언트가 최소 30Mbps 이상의 다운로드 속도를 보였으며, 2대의 클라이언트를 제외하고는 모두 40Mbps 이상의 속도로 콘텐츠를 다운받을 수 있었다. 이것으로 볼 때, 100Mbps의 대역폭을 가진 1개의 콘텐츠 서버만으로도 30대의 클라이언트에 대용량 콘텐츠의 스트리밍 서비스를 제공할 수 있을 것으로 기대된다. 사용자의 숫자가 더욱 큰 경우에도 Peer 클라이언트들은 자신들의 스왈 네트워크를 구축하여, 그 내부에서 콘텐츠 조각을 공유하므로, Peer 클라이언트의 숫자가 더욱 많아진다고 하여도 다운로드 속도의 저하는 크지 않을 것으로 예상된다. 실제로 상단의 실험 중 실제 Peer 는 평균 5개 정도만 연결되어 데이터를 공유하였다. 또한, 다수의 실험 수행을 통해, 약간의 성능 저하를 보인 2대의 클라이언트는 클라우드 상에서 보이는 일시적인 현상임을 파악하였다.

### 3.3 서버 동시 접속자 수

서비스 상용화에 있어서 가장 중요하게 고려해야 할 사항 중 하나는 콘텐츠 서버의 안정성이다. 서비스 측면에서의 안정성을 보장하기 위해서는 많은 Peer 클라이언트가 시스템에 접속하는 상황에서도 콘텐츠 서버가 직접적으로 서비스하는 Peer 클라이언트의 수를 최소화하고 Peer 클라이언트끼리 콘텐츠 서버로부터 전송받은 콘텐츠를 공유하는 수를 극대화하는 것이 필요하다. 이러한 동작을 바탕으로 서버의 동시 접속자 수를 늘릴 수 있고, 이는 결국 서비스 제공 가능 용량을 극대화할 수 있다. 이를 측정하기 위해 30대의 Peer 클라이언트 중에서 콘텐츠 서버 1대가 직접적으로 콘텐츠를 전송한 Peer 클라이언트의 수가 몇 대인지를 측정하였다. 본 실험에서는 콘텐츠 서버가 최대 5대 존재하기

때문에, 한 개의 콘텐츠 서버 당 평균적으로 몇 대의 Peer 클라이언트를 서비스하는지를 측정하였다. 이러한 측정방식을 통해 클라이언트의 수, 콘텐츠 서버의 수, 콘텐츠 서버의 평균 동시접속자 수 간의 상관관계를 예측할 수 있으며, 이에 기반하여 안정적인 서비스를 위한 콘텐츠 서버의 개수를 추정할 수 있다. 콘텐츠 서버의 개수 변화에 따른 콘텐츠 서버 당 평균 동시접속 Peer 클라이언트 수를 시험하였다. 나타냄, 콘텐츠 서버가 1대만 존재할 경우에도 2대 미만 (약 1.5)의 Peer 클라이언트에게 비디오 데이터를 전송하는 것만으로도 30대의 Peer 클라이언트 전체에 스트리밍 서비스를 제공할 수 있음을 확인하였다. 특히 콘텐츠 서버의 개수가 늘어날수록 한 콘텐츠 서버 당 서비스해야 하는 동시접속 Peer 클라이언트의 수는 지속적으로 감소하였다. 이는 P2P 프로토콜의 Peer 간 데이터 공유 프로토콜 특성에 기인한 결과로, 콘텐츠 서버 1대만을 운영하는 경우에도 획기적인 동시접속 클라이언트 숫자의 절감 효과를 가져올 수 있었다.

결론적으로, 기존의 클라이언트-서버 모델이 클라이언트의 수 만큼 동시접속자를 관리하고, 자원을 사용해야 하는 것에 비해 약 5%의 클라이언트만큼의 동시접속자를 관리하는 것만으로 30대의 클라이언트 모두에 스트리밍 서비스를 제공할 수 있으며, 콘텐츠 서버의 개수를 늘리는 것으로 더 많은 규모의 클라이언트를 지원할 수 있을 것으로 예상된다.

### 3.4 캐시 적중 트래픽 비율

캐시 서버는 콘텐츠 서버가 발생하는 트래픽의 부하를 덜어준다는 점에서 필수적으로 존재해야 하는 요소이다. 즉, 캐시 서버가 Peer 클라이언트에 제공하는 트래픽의 양만큼 콘텐츠 서버의 트래픽 비용 지출은 감소하게 된다, 특히, 앞서 언급한 바와 같이 콘텐츠의 인기도는 Zipf's Law를 따르므로, 인기 있는 콘텐츠 몇 개만을 캐시 서버에 저장하는 경우에도 캐시 적중 비율이 높으므로 상당히 높은 트래픽 감소 효과를 이룰 수 있다. 네트워크 내에 캐시 서버의 존재로 인한 콘텐츠 서버의 부하 감소량을 측정하기 위해 콘텐츠 캐시 서버 4개를 추가하여 실험을 진행하였다. (콘텐츠 서버는 1대를 배치하였음.) 실험 상에서 콘텐츠 서버와 캐시 서버에서 제공한 트래픽의 총량을 측정하여 총 트래픽의 비율을 측정하였다. 측정 결과 같이 캐시 서버와 콘텐츠 서버의 트래픽 비율은 20%정도로 유사한 경향을 보였다. 이것은 모든 캐시 서버가 동일하게 콘텐츠 서버의 트래픽을 분산하여 서비스했다는 것을 의미하며, 이로 인해 콘텐츠 서버의 부하는 약 20% 정도로 감소하게 된다. (즉 서버 부하 감소율은 80%). 이러한 성능 향상 결과는 본 논문의 스트리밍 프로토콜이 콘텐츠 서버 및 캐시 서버를 데이터 조각 (Chunk) 전송 시 균등하게 참여시키기 때문에 발생한 결과로, 캐시 서버가 콘텐츠 서버의 부하를 동등한 수준에서 감소 시켜줄 수 있음을 의미한다.

### 3.5 콘텐츠 서버 트래픽 감소율

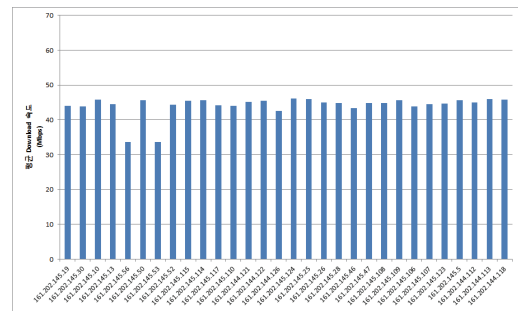
앞의 실험 결과에서 알 수 있듯이, 네트워크 상에 콘텐츠 서버나 캐시 서버를 많이 배치할수록 사용자는 더 짧은 대기시간과

안정적인 속도의 스트리밍 서비스를 보장 받을 수 있다. 특히 P2P 프로토콜을 기반으로 하는 스트리밍 프로토콜의 특성 상 부하가 균등하게 분배되는 효과가 존재한다. 하지만 콘텐츠 서버의 개수 및 서비스에 필요한 트래픽의 양은 서비스 제공자가 지불해야 할 자원이기 때문에 최소화 할 필요가 있다. 따라서, 콘텐츠 서버의 개수의 변화에 따른 전체 네트워크 트래픽에서 콘텐츠 서버가 발생시킨 트래픽의 비율 계산을 통해 콘텐츠 서버의 자원 사용 정도를 측정하고자 하며, 이를 위해 전체 발생된 트래픽에 대해 콘텐츠 서버 트래픽 비율을 측정하는 것이 필요하다. 이를 위해 콘텐츠 서버의 개수를 변화시키면서 전체 트래픽 중에서 서버에서 전송하는 트래픽의 비율을 측정하였다. 측정 결과 서버의 개수가 증가할수록 서버가 제공하는 전체 트래픽의 양은 증가하였다. 이는 데이터를 균등하게 공유하는 P2P 네트워크의 특성에 따라 기인하는 현상으로, 콘텐츠 서버의 개수가 늘어날수록 선형적으로 증가하였다. 그러나 한 대의 서버만으로도 30대의 클라이언트에 스트리밍 서비스를 하는 것은 문제가 없지만, 콘텐츠 서버의 양이 증가하면서 클라이언트에 더 많은 양의 콘텐츠를 동시에 전송하기 때문에 전체 트래픽에서 콘텐츠 서버의 트래픽 차지 비중이 높아진다. 비록 서버의 트래픽 관련 비용 지출은 증가하지만, 클라이언트의 대기시간을 낮추고 다운로드 속도를 향상시킬 수 있다는 효과가 있음을 알 수 있다. 또한 5대의 서버가 공격적으로 콘텐츠를 전송하는 상황에서도 클라이언트끼리 공유하는 트래픽의 비율이 압도적으로 높음 (약 84%). 이는 기존 클라이언트 모델에 비해 트래픽 지출이 약 16% 정도로 감소한다는 것을 의미한다.

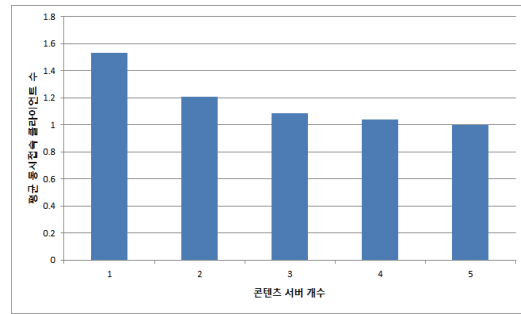
실제 서비스 관점으로 본다면 서버의 트래픽을 최소화하기 위해서 콘텐츠 서버를 1~2대를 운영하는 것보다는, 실제 서비스를 하는데 필요한 트래픽 절감 비율을 정한 뒤에, 이 범위에 맞는 한에서 최대한 콘텐츠 서버의 개수를 늘리는 방향으로 서버를 배치해야한다는 결론을 얻었다.

#### 4. 결 론

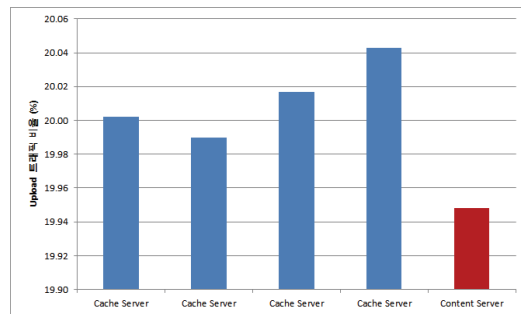
본 논문에서는 1) P2P 협력 기반 사용자들의 네트워크 동안 협력 수행 및 서버 부하에 따른 선택적 사용자 자원 적응 기술과 2) 사용자의 콘텐츠 접근 패턴, 네트워크 연결성, 파일 콘텐츠 특성 등의 정보를 고려한 device-to-device 전송 기술들을 제안하였다. 즉, 본 논문의 비디오 스트리밍 프로토콜은 대용량 비디오를 균등한 크기의 장면(Scene)으로 쪼개며, 사용자의 그룹을 스웜(Swarm)이라는 단위로 쪼개서 관리하고 장면은 다시 전송 및 에러 수정의 효율성을 위해 조각(chunk)로 쪼개져서 전송하는 방식으로 1) 기존의 일반적 P2P 프로토콜과 같이 사용자간 협업 네트워크를 구성하여 고용량 콘텐츠의 전송을 가능하게 하며, 2) 연속적이고 순차적으로 데이터를 소비하는 비디오 스트리밍 서비스의 데이터 소비 형태에 부합하고, 3) 단말의 자원을 효율적으로 이용하여 콘텐츠 서비스 제공자의 네트워크 자원 이용을 최소화할 수 있도록 설계하였다. 또한 인기 있는 콘텐츠를 캐싱/분산 저장하기 위한 간단한 알고리즘에 의해 콘텐츠를 캐싱하고 P2P 협업 네트워크 프로토콜과 연동하고 콘텐츠 캐싱을 위한



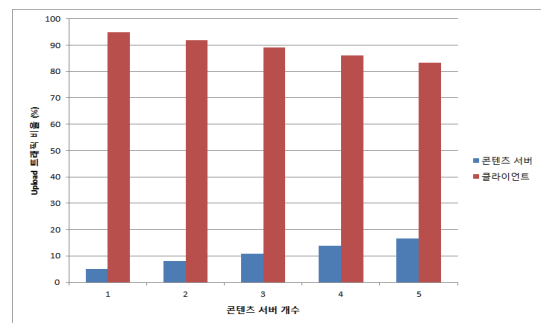
(1) 평균 다운로드 속도(Mbps)



(2) 평균 동시접속 클라이언트 수



(3) 콘텐츠 전송 트래픽 발생 비율



(4) 서버의 트래픽 전송 비율

그림 4 성능지표 결과

Fig. 4 Results of performances

부가적인 정보의 유지를 최소한으로 하기 위해서 콘텐츠에 대한 요청 횟수를 캐싱의 지표로 삼았으며, 콘텐츠를 전송하는 네트워

크 프로토콜과 밀접하게 연관되어 동작할 수 있도록 캐시 서버가 Peer로 동작하도록 프로토콜을 구현하였다.

제안된 기술의 검증에 위하여 평균 콘텐츠 다운로드 속도, 콘텐츠별 동시 접속 수 감소율, 적응형 캐시 적중 비율, 스트리밍 서버 트래픽 감소율을 시험한 결과 콘텐츠 다운로드 속도는 40MB, 콘텐츠별 동시 접속 수 감소율 50% 이상, 적응형 캐시 적중 비율 80%, 스트리밍 서버 트래픽 감소율 70% 이상의 달성의 결과를 보였다. 클라우드 환경하에서 제약을 둔 시험 환경이지만 방법적 측면에서의 실용화 가능성 검증이라는 큰 의미를 가진다고 판단된다.

### 감사의 글

이 연구는 2016년도 극동대학교 교내연구비 지원에 의하여 수행된 것임(FEU2015S01)

### References

[1] Xiaofei Wang, Min Chen, Zhu Han, Ted "Taekyoung" Kwon, and Yanghee Choi, "Content Dissemination by Pushing and Sharing in Mobile Cellular Networks: An Analytical Study," IEEE MASS 2012, Las Vegas, Nevada, USA, Oct. 2012.

[2] Kunwoo Park, Junghoon Kim, Kideok Cho, Ted "Taekyoung" Kwon, Yanghee Choi, and Sangheon Pack, "Waterfall: Video Distribution by Cascading Multiple Swarms," IEEE JSAC, accepted for publication

[3] Jinyoung Han, Seungbae Kim, Taejoong Chung, Ted "Taekyoung" Kwon, Hyun-chul Kim, and Yanghee Choi, "Bundling Practice in BitTorrent: What, How, and Why," ACM SIGMETRICS 2012, London, UK, Jun. 2012.

[4] Seungbae Kim, Jinyoung Han, Taejoong Chung, Hyun-chul Kim, Ted "Taekyoung" Kwon, Yanghee Choi, "Content Publishing and Downloading Practice in BitTorrent," in Proc. IFIP Networking, Prague, Czech Republic, May 2012.

[5] Jaeyoung Choi, Jinyoung Han, Taejoong Chung, Eunsang Cho, Ted "Taekyoung" Kwon, Yanghee Choi, "Bandwidth Allocation for BitTorrent under Multi-Torrent Environments," in Proc. IEEE GLOBECOM 2011, Houston, TX, (US), December 2011.

[6] P. Hui, J. Crowcroft, and E. Yoneki "BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks," IEEE TMC, vol. 10, pp. 1576-1589.

[7] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile Data Offloading through Opportunistic Communications and Social Participation," IEEE TMC, vol. 11, no. 5, pp. 821-834, 2011.

[8] Yago Sánchez de la Fuente, Thomas Schierl, Cornelius Hellge, Thomas Wiegand, Dohy Hong, Danny De Vleeschauwer, Werner Van Leekwijck, and Yannick Le Louédec, "iDASH: improved dynamic adaptive streaming over HTTP using scalable video coding," In Proceedings of the second annual ACM conference on Multimedia systems (MMSys '11). ACM, New York, NY, USA, 257-264.

[9] Arun Venkataramani, Praveen Yalagandula, Ravindranath Kokku, Sadia Sharif, Mike Dahlin, The potential costs and benefits of long-term prefetching for content distribution, Computer Communications, Volume 25, Issue 4, 1 March 2002, Pages 367-375, ISSN 0140-3664, 10.1016/S0140-3664(01)00408-X.

[10] M. Reisslein and K. W. Ross. Join-the-shortest-queue prefetching protocol for VBR video on demand. In Proc. IEEE International Conference on Network Protocols (ICNP), Atlanta, GA, October 1997.

[11] Chung-Ming Huang and Tz-Heng Hsu. 2003. A User-Aware Prefetching Mechanism for Video Streaming. World Wide Web 6, 4 (December 2003), 353-374.

[12] Marvelus, "Development of High Quality/Optimal Video Contents Streaming for Mass" Small & Medium Tech. Development RnD, Final Report, 2015

## 저 자 소 개



### 남 의 석 (Eui-Seok Nahm)

1991년 2월 : 연세대학교 전기공학과(공학사)  
1993년 2월 : 연세대학교 전기공학과(공학석사)  
1998년 2월 : 연세대학교 전기공학과(공학박사)  
1998년~2002년 : LS산전 책임연구원  
2003년 2월~현재 : 극동대학교 항공IT융합학과 교수