

## HTTP with Peer to Peer (HTTP3)

추 성 호, \*박 흥 성

강원대학교 대학원 제어계측공학과, \*강원대학교 전기전자정보통신공학부

전화 : 033-256-6501, \*033-250-6446

## HTTP with Peer to Peer (HTTP3)

Seongho Choo, \*Hong Seong Park

Dept. of Control & Instrumentation Eng., Grad. School, Kangwon Nat'l Univ.,

\*Dept. of Electrical & Computer Eng., Kangwon Nat'l Univ.

E-mail : somebody@control.kangwon.ac.kr, \*hspark@cc.kangwon.ac.kr

### Abstract

For reducing network traffic between long distance network or WAN, we supplement the Peer-to-Peer technology to HTTP. The large file transmission have the biggest traffic on HTTP. If downloading a large size file from a near computer not from the original location, we will get it speedier. For this, we propose an extended HTTP, named HTTP3. In this paper, we describe the HTTP3, the HTTP3 agent to support this protocol, and its security.

### I. 서론

인터넷의 폭발적인 사용 증가로 인해 인터넷 망의 데이터 전송량은 기하급수적으로 늘어나고 있다. 특히 World Wide Web (이하, WWW, Web 또는 웹) 관련 프로토콜이 대역폭의 대다수를 차지하고 있다. WWW을 지원하기 위한 HTTP는 멀티미디어 정보를 많이 실어나르기 때문에 더욱더 네트워크 대역폭을 많이 차지하고 있다. 네트워크 대역폭의 부족을 해결하기 위해 초고속 통신 망 등 차기 네트워크 시스템의 구축에 대한 연구 및 개발과 국가간 전송 선로의 대역폭 확대, 국가 기간망의 확충 등을 통해 모든 네트워크 시스템이 네트워크 성능 향상에 많은 자원을 투자하고 있는 실정이다.

현재 인터넷 환경의 다른 특징은 WWW의 저변확대로 인해 근거리 네트워크 내의 데이터 전송량도 증가했지만 원거리 네트워크 또는 국가간, 대륙간 네트워크에서의 데이터 전송량이 상당히 증가하게 되었다. 원거리 네트워크의 네트워크 성능 향상에는 근거리 네트워크에 비교할 수 없을 정도의 많은 물적, 시간적 자원이 필요하게 된다. 일반적으로 그런 자원 부담으로 인해 원거리 네트워크의 네트워크 대역폭은 근거리 네트워크 보다 협소할 수 밖에 없고 이로 인해 원거리 네트워크에서의 데이터 전송에 있어서 병목 현상이 발생하고 있다.

병목 현상과 더불어 심지어는 새로운 소프트웨어의 발표 시, 유명한 동영상 등의 멀티미디어 정보, 운영체제의 서비스 팩 등이 출시됐을 때, 네트워크 부하를 못 이긴 서버들의 다운 현상이 빈번하게 발생하고 있으며, 출시 직후의 원거리 네트워크의 네트워크 성능은 극도로 저하된다. 이러한 문제를 해결하기 위해 여러 가지 접근 방법들이 있었다.

HTTP의 성능 향상을 통한 사용의 편의를 추구하기 위한 연구는 많은 연구자들의 관심 속에서 광범위한 연구가 이루어져 오고 있다.

프로토콜 특성을 파악하여 실제 HTTP/TCP의 성능 측정에 관한 연구[2], HTTP/TCP 연결과 흐름에 대한 실험적인 방법 대한 연구 결과[4], HTTP/1.1의 전송량 예상에 대한 연구[3] 등이 있었으며 이로 인해 우리는 HTTP의 특성과 사용자들의 사용 패턴, 전송 데이터들의 특성을 파악할 수 있었다.

또한 WWW 환경에서 원거리 데이터 전송량을 줄이기 위해 캐쉬(cache) 기술을 이용한 미러링(mirroring)

기술과 프록시(proxy) 기술들이 사용되어 왔다. 미러링 기술은 사전에 캐쉬할 데이터를 미리 정해 사용하는 기술이고, 프록시는 위치에 따라 브라우저(browser) 내부의 캐쉬 뿐만 아니라 프록시 형태로 구현하면서 개인용 컴퓨터에 위치하는 퍼스널(personal) 프록시, 기업 및 단체 단위, 국가, 대륙 간의 프록시 서버에 대한 연구[1]가 많이 진행되었으며 인터넷 초창기에는 많이 이용되었지만, 현재는 별도의 기능에 더 초점을 맞추어 네트워크 하위 레벨에 설치되는 추세이다. 또한 프록시 자체 성능을 향상시키기 위해 캐쉬의 hit ratio를 높이는 방법, 메모리 관리를 통해 캐쉬의 성능을 향상시키는 방법 등에 대한 연구[5]도 상당히 많이 있어왔고 현재도 계속 진행되고 있다. 하지만 앞서 기술했던 바와 같이 현실적인 여러 가지 제약이 있으므로, 현재 상황에서 프록시는 캐쉬 기능 보다는 보안 등의 다른 이유로 더 많이 사용되어지고 있다.

이 모든 방법들은 모두 중앙집중적인 방법을 통해 원거리 네트워크의 부하를 줄이려는 시도였다. 하지만 이미 개인의 컴퓨터에 저장되어 있는 많은 데이터들을 이용하면 분산 환경의 장점, 즉 고용량 서버의 불필요, 부하 분산으로 병목 현상 방지, 거대한 저장 용량 및 지역 네트워크 대역폭의 적절한 안분 활용 등을 모두 취할 수 있게 된다. 우리는 이런 제반 문제를 해결하는 데에는 물리적인 확장, 소프트웨어 확장 뿐만 아니라 네트워크 프로토콜 자체의 효율성 제고도 있어야 한다고 생각한다. 그 중 가장 적당하다고 생각하는 방법 중의 하나가 제안하려고 하는 Peer-to-peer 기술을 기존의 WWW 환경에 접목시키는 방법이다. 현재 Peer-to-peer 기술을 응용한 서비스들은 별도의 전용 클라이언트 프로그램을 사용하여야 하며 검색 서버에 접속하여 일단 검색을 하여야 하는 부분적인 중앙집중적인 방법에 의해 운영되어지고 있다. 이것은 검색 서버의 의존성이 크기 때문에 문제가 있으며 별도의 프로그램은 일반인들에게 거부감을 줄 수 있는 문제이다.

우리는 일반인들이 쉽게 널리 쓰이는 웹 브라우저에 HTTP를 확장한 프로토콜을 적용함으로써 근본적인 네트워크 부하의 분산화를 추구하며 편리한 사용자 환경을 구축하고자 한다. Peer-to-peer 기술의 근본적인 장점에 의해 원거리 네트워크에서의 부하도 획기적으로 줄어들 것으로 생각한다. 우리는 이 방법을 현실로 옮기기 위해 HTTP에 Peer-to-peer 기술을 접목, 보완한 HTTP3 (HTTP with Peer to Peer) 프로토콜을 제안한다.

## II. HTTP3의 구조

### 2.1 네트워크 전송 데이터 특성

HTTP 프로토콜의 대역폭은 인터넷 통신에서 상당히 많은 대역폭을 차지하고 있다. 그 중 FTP와 더불어 대용량 파일의 전송도 상당히 많은 대역폭을 차지한다. 또한 동일한 파일의 경우도 많은 빈도로 전송이 이루어지고 있다. 일반적으로 비슷한 관심사와 연령대를 가지는 집단, 예를 들어, 학교나 기업 등의 근거리 네트워크에서 동일한 데이터를 원거리 네트워크로부터 전송 받는 확률은 상당히 높다고 볼 수 있다.

그러므로 우리는 가장 가까운 네트워크, 즉 서브넷(subnet)부터 대용량 파일을 찾고 되도록 가까운 컴퓨터에서 그 파일을 전송 받음으로써 네트워크 효율을 극대화하는 방안을 제안한다.

우리가 제안하는 방법에 의해 가장 이상적인 것은 지역 네트워크에 있는 여러 에이전트(agent)들이 상호 협력하여 잘 동작하여 hit ratio가 1이 되는 경우이다. 이 경우 동일한 대용량 파일의 전송이 원거리에서 일어나는 경우가 처음의 한번만 발생하게 되므로 위의 실험 결과치에 적용하여 계산할 경우, 원거리 전송 선로에 있어서 엄청난 네트워크 대역폭 감소 효과를 나타내게 된다.

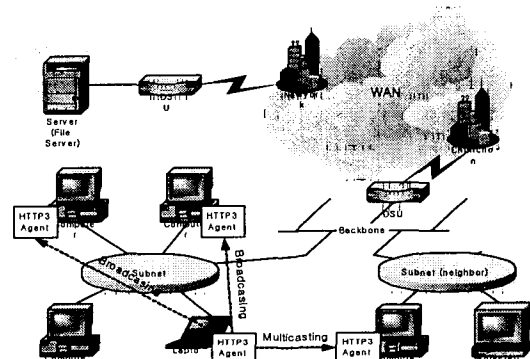


그림 1. 네트워크 구성과 HTTP3 질의 동작

### 2.2 HTTP3를 위한 네트워크 구성

HTTP3는 HTTP를 확장하고 있으나 당연히 아직 일반 브라우저에서는 지원을 안 하므로 HTTP만 지원하는 브라우저 기능의 확장을 위해 HTTP3 프로토콜 에이전트를 내장한 HTTP 프록시가 브라우저가 위치한 각 컴퓨터에서 동작하도록 설계했다.

## HTTP with Peer to Peer (HTTP3)

### 2.3 HTTP3 Agent

HTTP3 에이전트의 주요 기능은 브라우저와 웹 서버와의 HTTP 통신을 모니터 하다가 큰 파일을 받아야 할 경우에 인터셉터하여 HTTP3를 기동시키는 것이다. 이 후 주위의 에이전트로부터 파일을 전송 받도록 한다. (master mode)

자신에게 어떤 파일이 있는 지 다른 에이전트가 물어왔을(query) 경우 거기에 적절한 응답을 하고, 파일 전송 요구 시 서비스를 하는 역할을 한다. (slave mode)

또한 브라우저가 웹 서버로부터 일정 크기 이상의 파일을 전송 받는 경우, 그 URL과 자신의 컴퓨터 상의 위치를 저장하여 관리한다.

에이전트가 관리해야 할 것들은 다음 두 가지이다. 가지고 있는 파일 리스트의 원래 URL과 자신의 컴퓨터에서의 위치(URL)의 리스트, 사용자가 미리 설정해 둔 주변의 에이전트들의 IP 주소 리스트이다.

### 2.4 HTTP3 동작을 위한 파라미터

얼마만한 크기 이상의 파일을 일반 HTTP가 아닌 HTTP3 모드로 전송 받을지를 결정하기 위한 것은 다음과 같은 절차로 결정이 된다.

브라우저가 처음 서버와 연결을 시도할 때 서버의 응답 시간을 검사하여 서버와의 전송 속도를 유추한다. 이 때 에이전트는 서버와의 ping 프로토콜을 이용하여  $S_p$  크기의 패킷을 보냈을 때의 평균 응답 시간  $T_p$ 을 알 수 있다. 이 응답 시간의 1/2는 일단 브라우저와 서버 간의 전송 속도  $B_{wide}$ 라고 볼 수 있다.

$$B_{wide} = \frac{1}{2} \cdot \frac{S_p}{T_p}$$

이 시간에 의해 파일 크기의 한계치(threshold)  $S_{th}$ 가 계산되어 질 수 있다. HTTP3로 주위의 다른 에이전트에서 파일을 받아야 할 때 미리 사용자가 정해 놓은 주위 에이전트의 응답을 기다리는 최대 시간  $T_{WaitMax}$ 과 브라우저가 속해 있는 로컬 네트워크의 전송 속도  $B_{local}$ 가 있다면,  $S_{th}$ 가 다음과 같은 조건식을 만족할 때는 HTTP3를 이용하여 주변의 에이전트에 그 파일을 가지고 있는 지 질의(query)를 한다.

$$\frac{S_{th}}{B_{local}} + T_{WaitMax} < \frac{S_{th}}{B_{wide}}$$

$$S_{th} < \frac{T_{WaitMax} \cdot B_{local} \cdot B_{wide}}{B_{local} - B_{wide}}$$

즉, 전송 받아야 할 파일이 HTTP로 받을 것인지 HTTP3로 받을 것인지를 결정하는 파일 크기  $S_{th}$ 가 결정되어진 후에, HTTP로 통신을 하면서 그 크기 이상의 파일을 받아야 할 경우는 HTTP3 모드를 시작하게 된다.

먼저 브라우저는 HTTP HEADER를 통해 서버로부터 받으려는 파일의 MD5 hash value를 가져온 후, HTTP3 Agent를 통해 받은 파일의 hash value와 비교한다.

### 2.5 HTTP3 Packets

HTTP3 패킷은 다음과 같다.

Method (1 byte)	Data Length (2 bytes)	Data
--------------------	--------------------------	------

Method field에는 다음과 같은 내용이 적용된다.

- 0x01 : Query
- 0x02 : Query Response
- 0x03 : File Refresh

Data Length field는 그 다음에 올 데이터의 길이를 2 bytes로 표현하게 된다.

#### (1) Query Method

HTTP3 모드가 시작되면 브라우저는 서브넷에 질의 Method 패킷을 broadcasting하게 된다. 그 후 좀 더 질의 대상 에이전트를 확장하기 위해, 즉, hit ratio를 높이기 위해 자신의 서브넷에 속해있지 않는 에이전트 리스트로 multicasting도 한다. 이 에이전트 리스트는 사용자가 임의로 지정할 수 있다. 자동으로 찾는 것은 좀 더 연구가 진행되어야 할 것이다.

그 때 전송하는 query method 패킷은 다음과 같다.

0x01	URL의 크기	찾고자 하는 파일의 URL
------	---------	----------------

#### (2) Query Response Method

만약 query method 패킷을 받은 에이전트가 자신이 보유하고 있는 파일 목록에서 해당 파일이 있고, 그 파일 목록의 자신 컴퓨터의 URL에 정발 그 파일이 있는 지를 확인했다면, 다음과 같은 응답을 보낸다.

0x02	Index의 크기	자신이 소유한 파일 리스트에 있는 파일의 Index	파일의 hash value
------	-----------	------------------------------	----------------

브라우저의 에이전트는 가장 먼저 온 query response method 패킷의 hash value를 확인하고 그 패킷을 보내온 에이전트와 파일 전송을 시작한다.

(3) File Response Method

파일 전송이 끝난 후, 미리 원 서버에서 받았던 hash value와 전송 받은 file의 hash value를 비교하여 확인을 하게 되는데, 만약 서로 다른 경우, 해당 agent에 다음과 같은 패킷을 보냄으로서 이 사실을 알려준다.

0x03	Data Length	Refresh할 file의 index	Refresh할 file의 URL
------	-------------	----------------------	--------------------

이 패킷을 받은 agent는 해당 index를 자신의 리스트에서 삭제한다.

2.6 보안

다른 에이전트로부터 받은 파일이 원래 서버로부터 받으려는 파일에 불순한 코드가 포함되어 있을 지 모르므로 MD5 해쉬를 하여 서버로부터 받은 해쉬값과 동일할지를 항상 확인한다.

물론 DOS(Denial Of Service) 등의 공격에 대비한 일정 시간 내에 일정 량 이상의 쿼리가 오는 경우 자동으로 막는 등의 네트워크 시각에서의 보안도 필요하다.

III. 결론

본 논문에서는 인터넷 환경에서의 네트워크 대역폭의 물리적 확장도 빠르고 편리한 인터넷 사용을 위해 중요하지만, 가장 많이 쓰이고 가장 많은 대역폭을 차지하는 HTTP에 Peer-to-peer 기술을 접목시킨 HTTP3 프로토콜을 제안하여 보다 근본적인 원거리 전송 선로의 데이터 전송량을 줄이는 방안을 마련하고자 했다.

이를 위해 HTTP3를 정의하고 그 프로토콜을 지원하기 위한 HTTP3 에이전트의 기능과 역할을 정의했다. 또한 현재 환경에서 이 프로토콜을 지원하여 실험하기 위해 HTTP3 에이전트를 포함한 HTTP 프록시를 구현하였다.

HTTP3 에이전트의 핵심 기능은 서버와의 응답 시간에 대비해하는 일정 크기 이상이 되는 파일을 전송 받아야할 경우, 주위에 있는 다른 HTTP3 에이전트에게 그 파일의 URL을 물어 만약 존재하면 그곳에서 전송을 받도록 하는 것이다. 이 때의 query는 서브넷에 broadcasting을 하고 부가적으로 에이전트가 가지고 있는 서버넷을 넘어서는 네트워크에 위치한 에이전트 리스트

트로 multicasting을 하게 된다. 이를 위해 에이전트는 자신의 컴퓨터에 가지고 있는 일정 크기 이상의 파일의 리스트와 URL을 관리하며, 또한 주위 에이전트의 위치 정보를 관리한다.

일반적으로 많은 대역폭을 차지하면서 응답 속도 영향 미치는 데이터 전송은 원거리에 있는 대용량 파일의 전송일 경우가 많으며, 일정 집단은 보통 같은 파일을 반복해서 받는 경우가 많다는 데에 착안하여 이를 Peer-to-peer 기술과 접목하여 원거리 전송 선로의 대역폭을 절약하자는 것이다.

HTTP3의 효율성을 높이는 핵심은 HTTP3 에이전트가 근거리 네트워크에 많이 설치되어 있는 것이다. HTTP3 에이전트 수가 많으면 많을수록 원거리 전송 선로를 이용한 대용량 데이터의 전송이 일어나지 않을 확률이 높아지기 때문이다.

이 프로토콜이 광범위하게 쓰이게 될 경우 집단이 크면 클수록 더욱더 원거리 전송 선로의 대역폭을 줄일 수 있을 것이며, 이것은 비용적인 면에서 매우 큰 효과를 도출해 낼 수 있다.

향후 에이전트 동작에 있어서의 자체 보안은 더욱 연구해야할 분야이다.

참고 문헌

- [1] Joo-Yong Kim, Kyoung-Woon Cho and Kern Koh, Computer Software and Application Conference, 1999. COMPSAC '99. Proc. The 23th Annual International, 1999, p.p. 278 ~ 283
- [2] Go Hasegawa, Masayuki Murata and Hideo Miyahara, "Performance evaluation of HTTP/TCP on asymmetric networks," International Journal of Communication Systems, 19, 1999, p.p. 281 ~ 296
- [3] Balachander Krishnamurthy and Jennifer Rexford, En Passant, "Predicting HTTP/1.1 Traffic," Global Telecommunication Conference '99 Proceedings, 1999, p.p. 1768 ~ 1773
- [4] Joachim Charzinski, "HTTP/TCP connection and flow characteristics," Performance Evaluation and International Journal vol. 42, 2000, p.p. 149 ~ 162
- [5] Anirban Mahanti, Derek Eager and Carey Williamson, "Temporal locality and its impact on Web proxy performance," Performance Evaluation an International Journal vol. 42, 2000, p.p. 187 ~ 203
- [6] RFC2616, "Hypertext Transfer Protocol -- HTTP/1.1," Network Working Group, June 1999