

# 토렌트 응용 설정에 따른 트래픽 전송 방식에 대한 연구

유종현, 권재범, 이수강, 김명섭

고려대학교 컴퓨터정보학과

{dididaboa, lhmagic, sukanglee, tmskim} @ korea.ac.kr

## 요 약

최근 토렌트와 같은 P2P 사용자가 늘어남에 따라 전 세계적으로 많은 양의 P2P 서비스 트래픽 또한 늘어나고 있다. 기존의 서버-클라이언트 구조의 일대일 파일 전송 방식과 다르게 각 Client 가 직접 다수의 세션을 생성하기 때문에 사용자의 인터넷 환경의 최대 대역폭까지 다운로드 속도가 증가한다. 이로 인해 네트워크 사용에 대한 효율이나 비용 측면에서 여러 가지 부작용이 나타나고 있다. 최근 이런 문제들을 해결하기 위해 토렌트 트래픽 제어 관련 연구가 다양하게 이루어지고 있으나, 토렌트 응용은 설정 변경을 통해 다양한 방식의 다운로드가 가능하기 때문에 사실상 제어가 어려운 현실이다. 본 논문에서는 Snort 기반의 패킷 차단 프로그램을 통해 토렌트 트래픽의 Payload 내 특정 시그니처를 찾고, 이를 활용하여 파일 다운로드를 제어하였다. 최종적으로 토렌트 응용의 설정 변경에 따라 다양한 방식으로 파일을 다운로드 하는 것을 확인하였다.

## 1. 서론

오늘날 많은 사용자들이 토렌트를 사용하여 파일을 공유한다. 다수의 접속을 통해 여러 곳에서 동시에 파일을 가져오는 방식을 사용하여 빠른 속도로 원하는 파일을 받을 수 있기 때문이다. 그러나 다수의 세션을 생성하는 만큼 많은 트래픽을 유발하거나 저작권 문제에 저촉되는 등 여러 분야에서 토렌트 트래픽 제한에 대한 필요성이 요구되고 있다.

시장조사기관 인비저널(Envisional)의 조사 결과에 따르면, 토렌트로 인해 발생하는 트래픽이 전 세계 인터넷 트래픽의 약 17.9%를 차지하고 있는 것으로 추정되며, 그 중 약 2/3 가 저작권 침해인 것으로 나타났다[1]. 이로 인해 토렌트 트래픽을 제한하기 위한 다양한 방법론이 제시되고 있는데, Client 와 Tracker 사이에 이루어지는 HTTP 프로토콜 통신(Tracker Request)을 막는 방법이 대표적이다. 그러나 토렌트는 응용의 설정에 따라 파일을 여러 방법으로 공유할 수 있기 때문에 많은 사용자들이 해당 네트워크의 토렌트 응용 제한을 우회하여 사용하고 있는 것이 현실이다.

본 1 논문에서는 Snort 기반의 패킷 차단 프로그램을 통해 HTTP 프로토콜 통신을 막고, 다양한 토렌트 응용 설정을 통해 토렌트 트래픽 제한을 우회하는 방법을 알아본다.

본 논문의 구성은 다음과 같다. 2 장에서는 토렌트 응용의 동작 원리와 HTTP 프로토콜 통신 제어를 위한 시그니처에 대해 알아보고, 토렌트 응용의

설정 변경에 따른 다양한 파일 전송을 보인다. 3 장에서는 결론 및 향후 연구를 제시한다.

## 2. 본론

본 장에서는 토렌트 응용의 동작 원리에 대해 간략히 알아본다. 또한 Client 와 Tracker 간 HTTP 프로토콜 통신 시 발생하는 Tracker Request 의 Payload 내 시그니처를 이용하여 Peer 요청을 제한한 후, 응용 설정에 따른 통신 방식을 살펴본다.

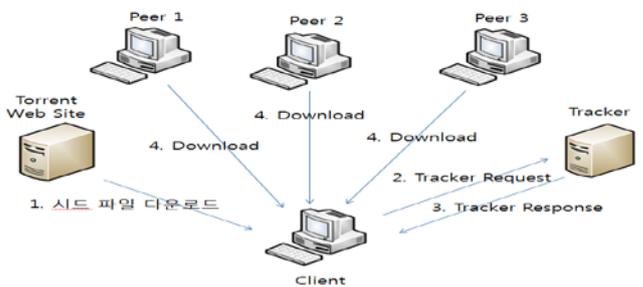


그림 1. Torrent 응용의 동작 원리

그림 1 은 토렌트 응용의 동작 원리를 도식화한 것이다. 토렌트 응용을 이용하여 데이터를 다운로드 하기 위해서는 먼저 시드(seed) 파일이나 자석(magnet)이라 불리는 링크주소가 필요하다. 그 중 시드 파일은 데이터를 다운로드 받고자 하는 Client 에게 필요한 파일의 Hash 값 및 Tracker 의 URL 주소가 포함된 메타 파일이다.

Client 가 시드 파일을 실행하면 파일에 포함된 정보를 기반으로 Tracker 에게 Peer(제공자) List 를 요청(Tracker Request)한다. 요청을 받은 Tracker 는 해당 Peer 들의 정보가 담긴 List 를 Client 에게 응답

이 논문은 2012 년 정부(교육과학기술부)의 재원으로 한국연구재단(2012R1A1A2007483) 및 2013 년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보.컴퓨팅기술개발사업(2010-0020728)의 지원을 받아 수행된 연구임.

(Tracker Respond)하는데, 이 때 HTTP 프로토콜을 이용하여 Client 와 Tracker 사이의 통신이 이루어진다. Tracker 를 통해 Peer List 를 받은 Client 는 해당 Peer 들과 BitTorrent 프로토콜을 이용하여 통신을 시작한다. Client 와 Peer 는 파일 공유를 위해 서로 가진 정보를 교환하며, 가지고 있는 데이터 조각을 확인한 후 주기적으로 조각을 공유한다.[2]

토렌트를 이용한 파일 공유를 제어하기 위해서는 Client 가 Tracker 에게 요청하는 HTTP 프로토콜 통신을 막거나, Tracker 와 Client 의 통신 이후에 이루어지는 Client 와 Peer 사이의 통신을 막아야 한다. Tracker Request 는 Client 가 다운로드 하려는 파일의 고유 Hash 값을 시드 파일에 포함되어 있는 URL 로 보내는 HTTP 프로토콜로, Tracker 는 이 때 받은 Hash 값을 이용하여 같은 파일을 소유한 Peer 들을 찾을 수 있다. "GET /announce?info\_hash=" 의 형태로 시작하기 때문에 Snort 기반의 패킷 차단 프로그램의 제어 시그니처로 info\_hash 를 사용하였다.[3]

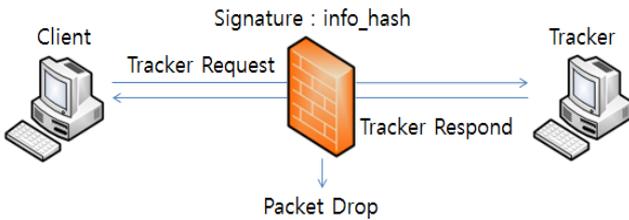


그림 2. Snort 프로그램을 이용한 패킷 차단

그림 2 는 실험에 사용한 Snort 기반의 패킷 제어 프로그램의 동작 방식을 나타낸 것이다. Client 가 Tracker Request 를 보낼 때 전송하는 패킷의 Payload 내에 info\_hash 를 찾아내 특정 패킷만 차단하는 방식이다. Tracker Request 가 차단되면 Client 는 Tracker 로부터 파일을 소유한 Peer 의 정보를 얻지 못해 파일을 다운로드할 수 없다. 그러나 토렌트 응용의 설정 중 하나인 망 내 피어 찾기를 활성화하면 Tracker Request 가 없어도 파일 다운로드가 가능하다. Client 가 Tracker 를 찾지 못할 경우 같은 네트워크의 Peer 들을 우선 연결하여 로컬 공유를 시작한다. 멀티캐스트의 특성상 하나의 패킷이 연결된 모든 Peer 들에게 전송되기 때문에 빠른 파일 공유가 이루어지나 과도한 트래픽을 유발하는 문제가 있다.

다른 설정으로 UDP Tracker 지원을 활성화하면 시드 파일의 Tracker URL 중 UDP 주소를 가진 URL 을 기반으로 통신이 이루어진다. UDP 프로토콜은 기존의 HTTP 프로토콜이 사용하는 대역폭의 50% 이하를 사용하고, 라우터 또는 서버가 처리할 수 있는 TCP 연결 수 제한이 적용되지 않는다.[4] Client 는 Hosts File 에 해당 URL 이 포함되어 있는지 확인 후 포함되어 있지 않으면 자신의 DNS Server 로 DNS Query 를 보내 Tracker 의 IP 주소를 알아낸다. 이후 다시 Tracker Request 와 Tracker Respond 를 주고 받는데, 이 때 Client 와 Tracker 는 connection\_id 와 action, transaction\_id[5]를 이용하여 통신을 시작한

다. 표 1 은 위의 내용을 표로 나타낸 것이다.

	Name	description
Client	Connection_id	네트워크 바이트 순서로 초기화, 프로토콜 식별
	Action	connect(0), announce(1), scrape(2), error(3)
	Transaction_id	Client 가 무작위로 결정
Tracker	Connection_id	Tracker 와 추가 정보 교환 시 사용, 프로토콜 식별
	Action	connect(0), announce(1), scrape(2), error(3)
	Transaction_id	Client 에서 보낸 transaction_id 와 일치해야 한다.

표 1. UDP Tracker 통신에 사용되는 값

실험을 통해 기존의 HTTP 프로토콜 시그니처를 이용하여 Tracker Request 를 차단하여도 망 내 피어 찾기, UDP Tracker 지원 활성화로 UDP Tracker 를 통한 파일 공유가 가능함을 확인하였다.

### 3. 결론 및 향후 연구

현재까지 토렌트 트래픽을 줄이기 위한 방법으로 토렌트 Web Site 접속 제한 및 Tracker 와 Client 사이의 HTTP 프로토콜 통신을 제한하는 방법이 제안되었다. 본 논문에서는 토렌트 응용의 Client 가 Tracker 에게 Peer List 를 요청하는 Tracker Request 의 Payload 내 시그니처를 사용하여 파일 공유를 차단하였다. 그러나 토렌트 응용은 설정에 따라 다양한 방식으로 동작, 제안된 방법을 우회하여 통신하였다. 토렌트 응용이 야기하는 네트워크 및 저작권 문제를 해결하기 위해서는 기존의 Client 와 Tracker 사이에 이루어지는 통신뿐만 아니라, UDP Tracker 를 이용한 파일 공유 제어에 대한 연구가 필요하다. 향후 연구로는 토렌트 응용의 UDP 통신을 제어하기 위해 응용 내 UDP 프로토콜 시그니처를 사용한 트래픽 차단에 대한 연구를 진행할 계획이다.

### 4. 참고 문헌

- [1] Envisional Ltd, "Technical report: An Estimate of Infringing Use of the Internet", version 1.8, pp.2-14, Jan. 2011
- [2] NMC Consulting Group, "BitTorrent 프로토콜의 동작원리", Netmanias 기술문서, 2011년 5월
- [3] Wiki Theory, "BitTorrent Protocol Specification v1.0", Retrieved april., 14, 2014, from <http://wiki.theory.org/BitTorrentSpecification>
- [4] Olaf van der Spek, "UDP Tracker Protocol for BitTorrent", [http://www.bittorrent.org/beps/bep\\_0015.html](http://www.bittorrent.org/beps/bep_0015.html)
- [5] Arvid Norberg's specification, for rasterbar.com: [http://www.rasterbar.com/products/libtorrent/udp\\_tracker\\_protocol.html](http://www.rasterbar.com/products/libtorrent/udp_tracker_protocol.html)