

Unchoked Peer 개수에 따른 BitTorrent 성능 분석

정태중⁰, 한진영, 김현철, 권태경, 최양희

서울대학교 컴퓨터공학부

Analysis on BitTorrent Performance Based on the Number of Unchoked Peers

Taejoong Chung⁰, Jinyoung Han, Hyunchul Kim, Taekyoung “Ted” Kwon, Yanghee Choi

School of Computer Science and Engineering, Seoul National University

{tjchung, jyhan, hkim}@mmlab.snu.ac.kr, {tk, yhchoi}@snu.ac.kr

요 약

최근 파일 공유를 위해 널리 사용되는 BitTorrent 의 장점은 BitTorrent 의 핵심 메커니즘인 peer 선택 기법에 기인한다. Peer 선택 기법은 해당 host 가 업로드 할 peer 를 선택하는 것인데, 이 때 자신에게 각 peer 가 업로드 한 비율을 비교해서 가장 많이 업로드 한 peer 들에게 업로드를 하게 된다. 그런데 BitTorrent 에서는 업로드할 peer 의 개수 (이하 unchoke 개수)를 4 개로 고정하고 있다. 하지만 peer 들의 개수나 다운로드 속도, 업로드 속도와 같은 공유집단 (이하 swarm)의 특성은 계속 변하기 때문에 unchoke 개수를 고정하는 것은 효율적이지 못할 것이다. 본 논문에서는 BitTorrent 에서 네트워크 자원인 업로드 대역폭을 몇 명의 peer 에게 배분하는 것이 효과적일지에 대해 고찰한다. 즉, BitTorrent 에서 unchoke 할 peer 개수가 고정된 4 개가 아니라 변할 때 전체 시스템 성능에 어떠한 영향이 끼치는지 분석한다. 분석을 위해 본 논문에서는 서울대학교 캠퍼스 망에 테스트베드를 설치하였고, 실제로 널리 사용되는 오픈 소스 BitTorrent 클라이언트를 수정해서 사용하였다. 테스트베드에서의 다양한 실험을 통해 BitTorrent 의 성능이 unchoke 개수에 따라 달라지게 되는 것을 보였고, 이에 따라 swarm 의 상황을 고려해서 unchoke 개수를 적절하게 조절하는 메커니즘이 필요하다는 것을 보였다.

1. 서론

BitTorrent [1]는 파일을 공유하기 위한 peer-to-peer (이하 P2P) 소프트웨어로 많이 사용되고 있다. IPOQUE 에서 최근에 발간한 보고서에 따르면 현재 인터넷 트래픽의 약 27-55%를 BitTorrent 가 차지하고 있다 [2]. 이렇게 BitTorrent 가 널리 사용되는 이유는 BitTorrent 가 기존의 다른 P2P 에서 해결하지 못한 문제점들을 해결하였기 때문이다. 첫 번째로, BitTorrent 에서는 인기도가 높은 파일에 대한 요구가 몰리는 flash crowd 상황에서도 효과적으로 파일이 전송된다. 두 번째로, 기존의 P2P 에서는 파일을 공유하는 공유집단 (이하 swarm)에 기여하지 않고 받기만 하는 free-rider 문제가 심각한데 BitTorrent 에서는 이 문제가 효과적으로 방지된다.

이러한 BitTorrent 의 장점들은 BitTorrent 의 핵심 메커니즘인 peer¹ 선택 기법에 기인한다. Peer 선택 기법은, 자신이 업로드 할 peer 를 선택하는 것으로 BitTorrent 에서는 tit-for-tat 전략에 의해 peer 를 선택

하게 된다. tit-for-tat 전략이란 자신에게 각 peer 가 업로드한 비율을 비교해서 가장 많이 업로드한 peer 를 선택하는 전략이다. 이때, 업로드 하도록 선택된 peer 들은 unchoke 되었다고 하고, 선택되지 않은 peer 들은 choke 되었다고 한다. 일반적으로 BitTorrent 에서는 unchoke 할 peer 의 개수를 4 개로 고정하고 있다.

하지만 BitTorrent 의 swarm 상황은 계속 변하기 때문에 unchoke 할 peer 의 개수를 4 개로 고정하는 것은 효율적이지 못하다. 예를 들어서 swarm 에 참여하는 peer 의 개수나 데이터의 완전한 사본을 가지고 있는 peer (이하 seed)의 개수, 그리고 데이터를 download 하고자 하는 peer (이하 leecher)의 개수는 계속해서 변하게 된다. 이때, 해당 호스트가 다운로드를 받기 위해서는 (즉, 다른 peer 가 업로드를 해주기 위해서는) 다른 peer 로부터 tit-for-tat 전략에 의해 선택을 받아야 한다. 다른 peer 로부터 선택을 효과적으로 받기 위해서는 해당 host 가 swarm 의 상황에 따라 소수의 peer 에게 많은 양의 대역폭으로 업로드를 하는 것이 유리할 수도 있고, 많은 peer 에게 적은 양의 대역폭으로 업로드를 하는 것이 유리할 수도 있다. 하지만 현재의 BitTorrent 의 peer 선

¹ 본 논문에서 peer 는 파일을 함께 공유하는 host 중 에서 자신과 연결이 되어 있는 host 를 지칭한다.

택 기법은 고정된 unchoke 할 peer 개수를 두게 해서 이러한 swarm의 상황을 반영하지 못하고 있다.

본 논문에서는, BitTorrent 에서 네트워크 자원인 업로드 대역폭을 몇 명의 peer 에게 배분하는 것이 효과적일지에 대해 고찰한다. 즉, BitTorrent 에서 unchoke 할 peer 개수가 고정된 4 개가 아니라 변할 때, 전체 시스템 성능에 어떠한 영향이 끼치는지 분석한다. 분석을 위해 본 논문에서는 서울대학교 캠퍼스 망에 테스트베드를 설치하였고, 실제로 널리 사용되는 오픈 소스 BitTorrent 클라이언트를 수정해서 사용하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 실험을 위한 테스트베드 및 실험 설정에 대해 설명한다. 3 장에서는 다양한 시나리오에서 unchoke 할 peer 개수 변화에 따른 성능 측정을 하고 마지막으로 4 장에서 결론을 맺는다.

2. 테스트베드 설정

본 논문에서 실험을 위해 그림 1 과 같이 서울대학교 캠퍼스망에 테스트베드를 설치하였다. 본 테스트베드에는 총 9 개의 host 가 있다. 이 중에서 8 대는 파일을 다운로드 받고자 하고 (즉, leecher), 나머지 1 대는 공유할 파일을 미리 갖고 있는 seed 이다. 또한 실험을 위해 하나의 테스트베드에 하나의 Tracker 도 설치하였다. 각 host 에서 동작할 BitTorrent 클라이언트로 현재 많이 사용되고 있는 오픈 소스 기반의 Azureus [3] 를 사용하였다. 각 host 의 상태를 매 시간마다 측정 및 기록하고, unchoke peer 의 개수를 변화시키기 위해서 Azureus 클라이언트의 소스를 고쳐서 사용하였다. 실험을 위한 환경 설정은 <표 1>에 나타나 있다.

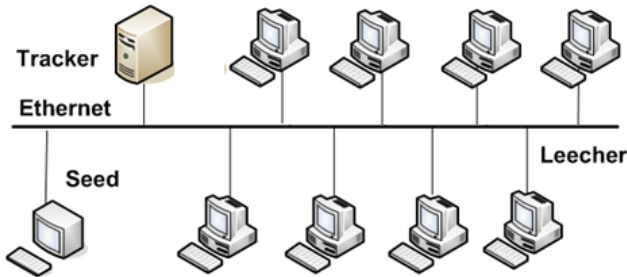


그림 1. 테스트베드

<표 1> 실험 환경 설정

항목	값
파일의 크기	298 MB
Seed 의 개수	1 개
Seed 의 업로드 peer 개수	2 개
Leecher 의 개수	8 개
대역폭	100Mbps

3. 성능 평가 및 분석

3.1 Swarm 내 모든 peer 의 unchoke 개수가 동일 할 때

첫 번째 실험은 swarm 내에 모든 peer 가 동일한 unchoke 개수를 가지고 파일을 공유할 때 효율적인 unchoke 개수를 알아보기 위한 것이다. 그림 2 에서 swarm 내의 unchoke 개수가 각각 2, 4, 6, 8 개 일 때 시간에 따른 다운로드 속도를 나타내고 있고, 표 2 에서 이에 따른 파일 다운로드 종료시간을 나타내고 있다. unchoke 개수가 8 개일 때, unchoke 개수가 2 개와 4 개일 때 보다 각각 34%, 21% 종료 시간이 단축 되었다. 즉 unchoke 개수가 늘어남에 따라 전체 swarm 에서의 peer 간의 업로드의 수와 양이 늘어나 되고, 이에 따라 swarm 전체적으로 다운로드 속도가 빨라지게 된 것이다. 그러므로 전체 swarm 측면에서는 대역폭이 충분할 때, unchoke 개수가 늘어날수록 swarm 의 전체적인 성능이 향상됨을 알 수 있다.

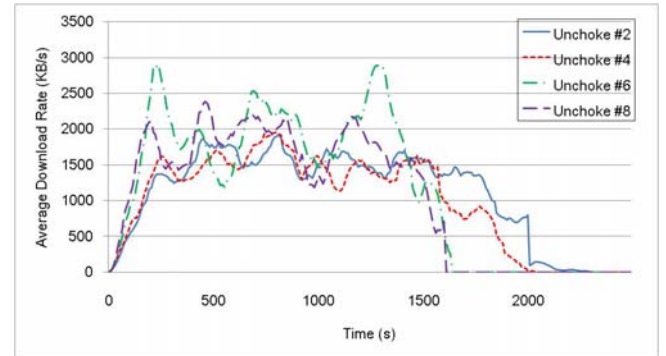


그림 2. 8 개 host 의 unchoke 개수가 각각 2 개, 4 개, 6 개, 8 개 일 때의 Average Download Rate

<표 2> Unchoke 개수에 따른 다운로드 종료시간

Unchoke 개수	다운로드 종료시간
2 개	251 초
4 개	207 초
6 개	165 초
8 개	162 초

3.2 Swarm 내 peer 의 unchoke 개수가 다를 때

다음 실험은 한 swarm 내에 동일하지 않은 unchoke 개수를 지닌 host 들이 파일을 공유할 때의 성능 차이를 알아보기 위한 것이다. 각각의 host 들은 서로 다른 상황에서 파일을 공유하고 있고 (예: 높은 대역폭 사용 vs. 낮은 대역폭 사용), 또한 각자 swarm 에 업로드로 공헌하려는 정도가 모두 다르다. 따라서 3.1 장의 결과처럼 모든 host 들이 많은 unchoke 개수를 사용하는 것은 전체 swarm 성능에 도움이 되지만 일부 host 들은 많은 unchoke 개수를 사용하려 하지 않을 수 있다. 따라

서 본 실험에서는 이와 같은 상황을 가정해서 unchoke 개수가 2 개인 4 개의 host 와 unchoke 개수가 5 개인 4 개의 host 가 존재 할 때 시간에 따른 다운로드 속도를 측정하였다.

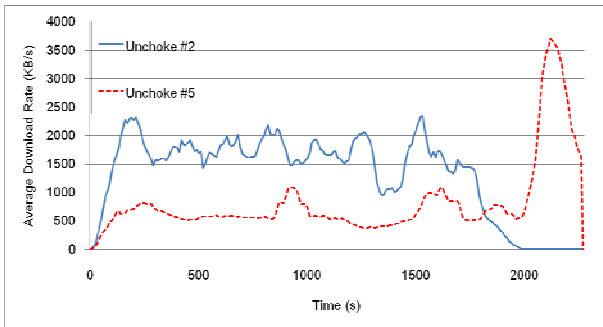


그림 3. Unchoke 개수가 2 개인 host 4 개와 5 개인 host 4 개의 Average Download Rate

그림 3 은 unchoke 개수가 2 개인 host 들과 unchoke 개수가 5 개인 host 들의 평균 다운로드 속도를 나타낸다. 그림 3 에서 보여주듯이 unchoke 개수가 2 개인 host 들이 보다 빠른 다운로드 속도를 보여주었고 이에 따라 약 50 초 정도 빠른 파일 다운로드 종료시간을 나타내었다. 즉 한 swarm 내에 다른 수의 unchoke 개수를 지닌 host 들이 참여하고 있을 때 적은 개수의 unchoke 를 사용하는 host 들이 더 큰 이득을 볼 수 있다. 이것은 같은 양의 대역폭을 더 적은 수의 peer 에게 할당한 unchoke 개수 2 개의 host 들이 각 peer 에게 할당한 업로드 대역폭의 양이 unchoke 개수 5 개 host 들보다 크기 때문에 상대적으로 다른 peer 로부터 선택 (unchoke) 될 확률이 크기 때문이다. 즉, 이와 같은 경우에는 많은 peer 에게 적은 양으로 업로드 하는 것 (즉, 많은 unchoke 개수를 설정하는 것) 보다, 적은 peer 에게 많은 양으로 업로드 하는 것 (즉, 적은 unchoke 개수를 설정하는 것) 이 개개의 host 에게는 더 유리하다. 물론 3.1 장에서 보듯이 개개의 host 가 이러한 선택을 한다면 전체 swarm 성능에는 오히려 좋지 않을 수 있다.

마지막으로 좀 더 다양한 unchoke 개수를 가진 host 들이 파일공유를 할 때의 효율적인 unchoke 개수를 알아보기 위해 2 대씩 host 를 묶은 4 그룹의 unchoke 개수를 각각 2, 3, 4, 5 씩 설정하고 실험을 하였다. 그림 4 에서 보여주듯이 unchoke 개수가 상대적으로 작은 host 들이 unchoke 개수가 큰 host 들보다 빠른 다운로드 속도를 나타내었고 그에 따라 다운로드 종료 시간도 단축됨을 볼 수 있었다. 이것은 그림 3 의 실험과 유사한 결과로, 한 swarm 내에 각기 다른 unchoke 개수를 지닌 host 가 존재할 때, unchoke 개수를 적게 설정할 수록 host 의 성능에 유리함을 보여준다.

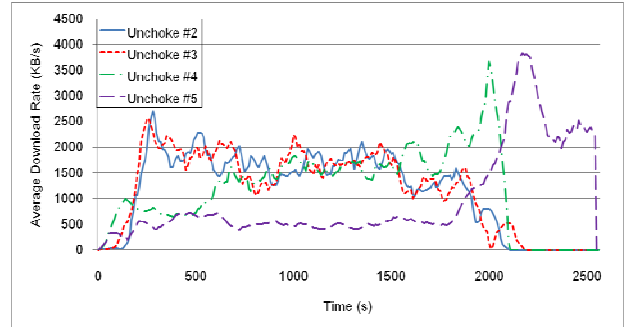


그림 4. 2 개 Host 씩 무리를 지은 4 개의 그룹이 각각 2 개 3 개 4 개 5 개의 unchoke 개수를 가질 때 Average Download Rate

4. 결론 및 향후 과제

본 논문에서는 swarm 의 특성에 따라 unchoke 개수의 변화가 BitTorrent 성능에 영향을 주는 것을 보여 주었다. 전체적인 swarm 의 성능을 고려하자면 unchoke 개수가 많을수록 전체 peer 들의 다운로드 속도와 종료시간에서의 이득을 볼 수 있었다. 하지만 각기 다른 unchoke 개수를 지닌 host 들이 한 swarm 에 참가할 경우 적은 unchoke 개수를 지닌 peer 가 큰 unchoke 개수를 지닌 peer 보다 성능향상에서의 이득을 보았다. 따라서 전체적인 swarm 의 성능과 각 host 의 성능을 동시에 최대화 할 수 있는 unchoke 개수 선택 기법이 필요할 것이다. 본 연구팀에서는 swarm 의 상황에 따라서 unchoke 개수를 조정하는 적응적 unchoke 개수 선택 기법을 후속 연구로 진행하고 있다.

5. 참고문헌

- [1] B. Cohen. Incentives build robustness in bittorrent. In 1st Workshop on Economics of peer-to-peer Systems, 2003.
- [2] The impact of p2p file sharing, voice over ip, instant messaging, oneclick hosting and media streaming on the internet. <http://www.ipoque.com/resources/internet-studies/internet-study-2008> 2009.
- [3] Azureus - now called vuze - open source bittorrent client. <http://azureus.sourceforge.net/>.