

Gnutella Protocol 기반 네트워크에서 P2P응용을 위한 캐싱 기법

김병룡^o 김기창

인하대학교 전자계산공학과, 인하대학교 정보통신공학과
dooryn@super.inha.ac.kr^o, kchang@inha.ac.kr

A Caching Technique for P2P Applications in Network based on Gnutella Protocol

ByungRyong Kim^o KiChang Kim

Dept. of Computer Science & Engineering, Dept. of Information & Communication Engineering

요 약

Gnutella Protocol 기반 네트워크에서 상호간 통신 및 검색과 응답과정에서 릴레이와 브로드캐스팅 되는 Query 메시지로 인해 많은 통신 부하를 초래한다. 또한 멀티 홉을 통한 질의 응답과정의 통신 오버헤드로 인해 통신 효율성이 저하된다. 따라서 본 논문에서는 Zipf's law 을 기반으로한 검색어 캐싱기법을 통해, 검색어 및 검색된 응답 문자열의 빈도와 랭킹, 그리고 검색된 파일의 크기와의 상관 관계를 얻어 캐싱 정책을 수립하고, 이에 따라 캐싱함과 동시에 캐싱정책을 이웃 peer에게 전파함으로써 전체적인 네트워크에서의 P2P 응용 성능을 향상 시킬수 있는 캐싱 기법을 제안 한다.

1. 서 론

인터넷 사용량의 증가로 인해 네트워크 환경이 급속도로 변화하고 있고, 그리고 인터넷 이용자들이 요구하는 자료의 양과 그 범위가 거대해지고 다양해지고 있다. 현재 이러한 자료의 저장 및 검색은 중앙집중 관리 형태로써, 클라이언트가 중앙서버에게 검색을 요청하면 서버가 적절하게 클라이언트에게 서비스를 제공하는 클라이언트-서버 모델이 주를 이루고 있다. 그러나 이로 인해 중앙서버가 담당하는 자료처리의 양이 많아지게 된다.

Gnutella Protocol은 순수 P2P구조로써 중앙서버의 간섭을 배제시켜 각 Peer가 독립적인 권한과 리소스를 통해 자료검색과 교환 및 관리와 이로 인해 발생하는 기타 제반사항을 독립된 Peer가 책임지고 관리한다. 독립적인 리소스를 통해 네트워크를 형성하고 전체 네트워크에서 발생하는 부하를 각 peer에게 분산시켜 진폭적인 효율을 향상시킬 수 있다. 그러나 이에 따르는 문제점 역시 존재한다. 중앙에서 자료의 저장과 검색을 하던 것을 이제 각 peer가 수행하게 되는데, 이때 peer-to-peer 통신을 위한 peer연결 정보 수집과 자료 검색과정에서 브로드 캐스팅 되는 패킷들로 인해 통신 대역폭을 떨어뜨릴 수 있고, 상대적으로 컴퓨팅 파워가 약한 peer에게 많은 양의 자료요청이 있을 경우에도 서비스 지연 및 네트워크 장애가 발생할 수 있다.

본 논문에서는 이러한 Gnutella Network에서의 문제점을 극복하기 위해서 Pong과 QueryHit 패킷을 캐싱하고, Zipf's law 을 기반으로 요청된 자료를 캐싱하여 peer간 통신 능력을 향상시키고 서비스의 질을 향상시킬 수 있는

캐싱 기법을 소개한다.

2. 기반 연구

2.1 Gnutella Networking

순수 P2P구조인 Gnutella Network에서는 그림1 과 같이 peer가 다른 peer로의 연결 정보를 얻기 위하여 Ping 메시지를 현재 연결된 모든 peer에게 전송하게 되면, 수신한 peer들은 응답값으로 자신의 연결정보와 리소스를 Pong 메시지로 전송하고, 또한 수신한 Ping메시지를 자신과 연결된 peer들에게 다시 전송하게 되고, 응답값이 오면, 원래의 peer에게 릴레이 시켜주게 된다. 또한 패킷 중복및 네트워크 자원 낭비를 줄이기 위해 GUID(global unique identifier)값과 TTL값을 이용한다^[1,2].

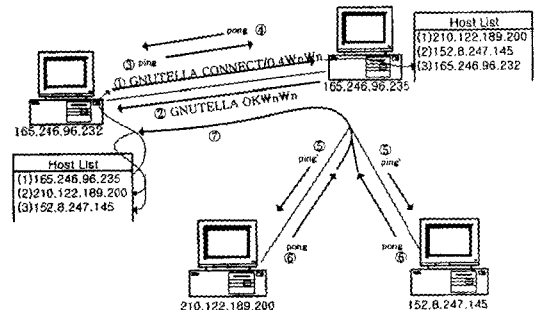


그림1 Gnutella Protocol의 기본 알고리즘

Gnutella Network에 참여하고 있는 peer들은 자료 검색을 위해 검색 쿼리를 Ping/Pong과정에서 얻어진 모든 연결가능한 Peer에게 전송하게 된다. 이 과정에서도 마찬가지로 검색 Query를 수신한 peer는 요청한 자료가 있는지 검색하여 그 응답값으로 QueryHit 메시지를 되돌려주게 되는 동시에 현재 자신과 연결된 모든 peer에게 수신한 Query메시지를 릴레이 시켜주게 된다. 그리고 릴레이시킨 Query에 대한 응답값인 QueryHit메시지가 되돌아오면 원래의 peer에게 다시 릴레이 시켜줌으로써, 자신과 직접 연결되지 않은 peer의 자료도 검색 가능하다^[1]. 이때 발생하는 패킷 중복과 네트워크 자원 낭비를 줄이기 위해 GUID와 TTL을 사용한다. 최종적으로 원하는 자료가 있을 경우 P2P로 Http를 이용하여 자료를 요청한다^[1].

2.2 Zipf's 법칙과 Zipf 분포

Zipf's 법칙은 자연어에서 단어의 rank와 frequency에 관련된 계산적 언어학의 실험적 사실을 바탕으로 한다. Zipf's 법칙은 표1과 같이 표현된다.

표 1

$$r * Fr = \text{Constant} \quad (r : \text{rank}, Fr : \text{frequency})$$

Rank와 Frequency와의 관계는 어떤 포괄적인 텍스트에 대해서도 유효하고 정확하게 일반적인 경험적인 법칙으로 표현된다는 것이다. 그리고 수많은 검색과정이 수행되는 인터넷망에서도 접속빈도 및 조회수와 이에따라 순번을 매긴 rank와의 관계도 Zipf's law의 법칙을 따른다^[3,4,5]. 따라서 Web Request들은 Zipf 분포를 따른다고 할 수 있다^[3,4,5]. 그리고 자료의 조회수와 그 자료의 크기와의 관계 및 서버의 접속빈도와 접속빈도를 통해 만들어진 랭킹과의 관계도 이와 비슷한 결과를 가져온다^[3,4,5].

3. Gnutella Network를 위한 캐싱 기법

3.1 Ping/Pong 및 Query/QueryHit 메시지의 캐싱

Gnutella Network에서 peer들은 많은 다른 peer들과 Network을 형성하기 위해 peer의 연결정보와 리소스, 그리고 연결 지속성 여부를 판단하기 위해 주기적으로 Ping/Pong메시지를 주고 받는다. 이때 peer는 자신과 현재 연결된 모든 peer들에게 ping메시지를 전송하고, 이를 수신한 peer는 pong메시지로 응답하는 동시에 수신한 Ping메시지를 자신과 연결된 peer들에게 전송하게 된다. GUID값 및 TTL값으로 중복메시지를 제거할 수 있으나, 그 메시지의 양은 상당하다. 따라서 그림2와 같이 일정 수준의 peer의 정보가 수집이 된 다음, ping메시지를 릴레이해줘야 할 peer들이 접속가능(Live)하다면 현재 자신과 연결된 peer정보를 이용하여 ping/pong메시지가 캐

시된 것 처럼 메시지를 생성하여 릴레이 하지 않고 바로 해당 peer에게 pong메시지를 생성하여 되돌려주게 되면, 많은 양의 메시지를 줄일 수 있다. Query와 QueryHit메시지도 똑같은 기법으로 효과적으로 사용될 수 있다.

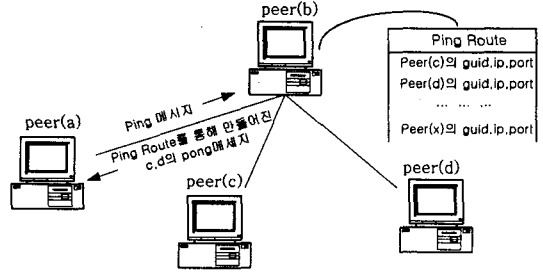


그림2 PingRoute를 이용한 Pong메시지 생성

자료검색에서 발생하는 Query/QueryHit메시지에 대한 캐싱도 PingRoute와 같이 Guid값과 검색된 자료를 가지고 있는 peer의 Session ID값^[1]을 이용하여 QueryHit메시지를 생성하여 되돌려주게 되면, 릴레이되는 Query메시지와 해당 peer에서의 응답값인 QueryHit메시지를 효과적으로 줄일 수 있다.

3.2 Zipf's 법칙에 근거한 파일 캐싱

Gnutella Network에서 원하는 자료를 검색하고 얻기 위해서는 가능한 많은 접속정보를 얻어야 하고, 그리고 많은 검색 메시지가 네트워크를 통해 브로드캐스팅 된다. 이때 peer들은 자신과 TTL값에 기반하여 근접하게 접속되어 있는 많은 peer로 부터 많은 직접 받거나, 릴레이된 검색 메시지를 받게 되고, 또한 릴레이해준 검색 메시지에 대한 검색 결과값을 받게 된다. 이때, peer들은 자신이 요청한 검색결과와 다른 peer가 요청한 검색메시지, 그리고 다른 peer가 요청한 검색메시지에 대한 검색결과를 받게 된다. 이렇게 수집된 검색어와 검색 결과를 가지고 Zipf's 법칙을 적용할 수 있다.

수집된 검색어들과 검색결과 문자열을 통하여 표1의 식을 통하여 수치적인 상수값을 얻을 수 있다^[3]. 즉, 검색어와 검색결과에서의 문자열을 가지고 순번을 매기고, 이를 주기적으로 업데이트 하면서, 정규화된 근사적인 상수값을 얻을 수 있다. 그림3-1과 같이 peer(b)는 수집된 정보를 가지고 통계적인 수치를 생성하고, 상위 랭킹에 대해서는 캐싱을 준비한다. 그 이후에 peer(a)가 peer(b)에게 검색을 요청하고, 다시 Query를 peer(c)에게 릴레이 해준다. 이때 검색 결과를 peer(b)가 되돌려받게 된다. 여기서 peer(b)는 현재 요청된 Query와 QueryHit값이 상위

랭킹 이라면 peer(b)는 마치 proxy와 같이 동작하게 된다. 따라서 검색 결과를 받은 peer(a)는 http를 이용하여 peer(b)에게 자료를 요청하게 되고 peer(b)는 peer(c)의 자료를 캐싱하는 동시에 peer(a)에게 요청한 자료를 서비스한다. 그러면 peer(d)가 과거에 peer(a)가 요청했던 자료를 peer(b)에게 요청한다면, peer(b)는 peer(c)에게 Query 메시지를 릴레이 시키지 않고 QueryHit 메시지를 생성하여 되돌려주게 되고, 결국 peer(b)는 캐싱된 자료를 peer(d)에게 서비스할 수 있다. 또한 QueryHit 메시지를 통해 자료를 서비스할 peer의 콘텐츠 서비스 품질을 알 수 있기때문에^[1] 자신의 리소스와 비교하여 서비스 품질이 낮을경우 캐싱한다면 더 효율적일 수 있다.

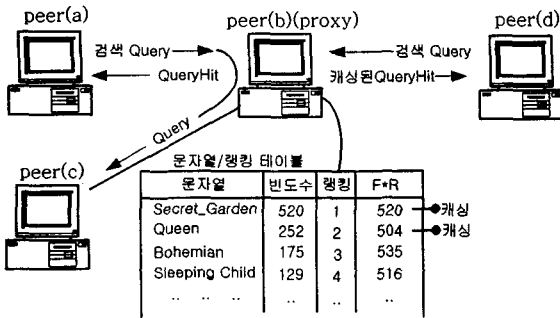


그림 3-1 Zipf's law에 기반한 캐싱 기법

peer(b)는 Query 및 QueryHit 메시지를 통해 랭킹 테이블을 생성하고, 그리고 QueryHit 메시지를 통해 검색된 자료의 크기를 알 수 있다^[1]. 따라서 2.2에서 설명한 자료 조회수와 자료의 크기와의 상관관계를 이용하여 그림3-2와 같이 상위랭킹된 문자열에 대한 자료 크기의 정규값을 생성한다.

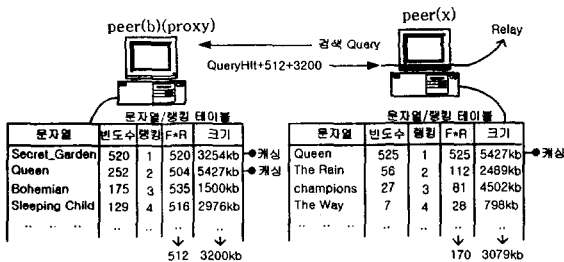


그림 3-2 Zipf's law에 기반한 캐싱 기법

이렇게 만들어진 정규값(512,3200)을 이웃하는 노드로 릴레이 되는 QueryHit 메시지가 있을 경우 QueryHit 메시

지의 Optional QHD Data부분에^[1] 추가하여 전송한다. 이때 이웃노드인 peer(x)는 아직 정규화된 값을 유도하지 못하였지만,현재의 빈도/랭킹 테이블에서 peer(b)에게 받은 상수값과 근사한 부분에 대해서는 향후에도 이웃한 peer들이 요청빈도가 많을것으로 간주하여 캐싱을 시도한다. 이로 인해 peer(b,x)와 인접한 peer들의 근사적인 자료 요구 패턴을 알게되고, 그에 따라 파일을 캐싱함으로써, 서비스 품질을 향상시키고 통신 부하를 감소시킬 수 있다.

4. 결론

본 논문에서는 Gnutella 네트워크에서 연결정보수집과 검색 Query 및 QueryHit 메시지가 인접한 peer로 브로드캐스팅되고 릴레이되므로 인해서 발생하는 통신 대역폭 낭비를 줄이고자 Pong 메시지를 캐싱하고, 그리고 Zipf's law에 기반하여 각 peer와 TTL기준으로 이웃한 peer들의 자료 요구성향을 분석하여 자료를 캐싱한다. 그리고 이웃한 peer들의 요구성향 분석치를 이웃 peer에게 전파함으로써 peer들이 구성하고 있는 Gnutella 네트워크안에서의 자료요구성향을 반영하도록 하였다. 이로 인해 peer간 주고받는 Query가 감소되고, peer간 서비스 품질도 향상될 것이다.

향후 연구과제로는 시간의 흐름에 따라 Zipf's law에 기반하여 각peer에서 생성하는 정규값의 변화를 어떻게 효율적으로 규정하고 적용할지의 연구가 있어야 할것이다.

참고 문헌

- [1] The Gnutella Developer Forum(GDF). "The Annotated Gnutella Protocol Specification v0.4"^[1]. http://dlaikar.de/gdf_files/Dveloppement/GnutellaProtocol-v0.4-r1.8.html. 2003.
- [2] Gnutelliums - Gnutella Download, <http://www.gnutelladev.com/>. 2001.
- [3] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker. "Web Caching and Zipf-like Distributions: Evidence and Implications". IEEE INFOCOM. MONTH 1999.
- [4] Kunwadee Sripanidkulchai. "The popularity of Gnutella queries and its implications on scalability". <http://www-2.cs.cmu.edu/~kunwadee/research/p2p/paper.html>. JAN 2004.
- [5] V.Almeida, A. Bestavros, M.Crovella and A. de Oliveira, "Characterizing reference locality in the WWW," in proceedings of 1996 International Conference on parallel and Distributed.