

Gnutella Protocol을 기반한 P2P Web Service 개발

김병룡⁰ 김기창
인하대학교 전자계산공학과
mydarling@lovebuffer.com kchang@dragon.inha.ac.kr

The Development of P2P Web Service Based on Gnutella Protocol

Byung-Ryong Kim⁰ Ki-Chang Kim
Dept. of Computer Science and Engineering, Inha University

요 약

오늘날 인터넷은 중앙 집중식 웹서버로 콘텐츠를 서비스하는 방식을 주로 택하고 있으며, 이용자들은 저마다의 홈페이지 서버를 갖길 원하고 있다. 그러나 이런 중앙 집중식 서비스방식을 사용했을 경우 비용적인 측면과 유지 보수 관리적인 문제로 인해 사용자들의 욕구를 제대로 해결해주지 못하고 있다. 본 논문에서는 중앙집중형 웹 콘텐츠 서비스가 아닌 순수한 Peer to Peer 방식의 웹 서비스를 할 수 있는 P2P 웹서버의 개발을 소개하고자 한다. 기본 알고리즘은 Gnutella Protocol을 기본으로 하였으며, 여기에 웹서버를 두고 Gnutella Protocol에서 사용되는 Header 외에 웹서버의 존재유무를 파악할 수 있도록 별도로 정의된 Header를 사용하여 쉽게 웹서버의 기능유무 및 위치추적을 가능하게 하였으며, 또한 기존의 Gnutella Protocol과도 호환성을 지니고 있다. 위치 추적 및 검색엔진은 Gnutella Protocol과 유사한 방식을 택했으며, 멀티미디어 파일의 검색 및 업/다운로드가 가능하고, 웹문서의 검색후 원하는 문서를 서비스한다. 즉 Peer가 검색엔진과 클라이언트 및 웹서버의 역할을 동시에 수행하는 순수 P2P 웹 서비스를 개발한다.

1. 서 론

모자이크가 웹의 제2의 전성기를 맞게 해주었고, 인터넷 이용자 또한 폭발적으로 증가하고 있다. 자연적으로 이용자들의 욕구의 증가 및 수준이 향상되었으며 능력에 따라서 자신만의 웹서버를 갖고자 한다. 그러나 기존의 중앙집중식 웹서비스 방식으로는 업데이트 비용증가 및 자체 인프라 확장의 문제로 인해 Dynamic IP를 가지고 있는 수많은 사용자들을 만족시키지 못하고 있다.

본 논문은 위의 문제점을 보완하기 위해 현재 음반산업 및 킬러 어플리케이션 시장에서 최대 이슈로 떠오르고 있는 P2P Network의 Gnutella P2P 어플리케이션을 이용하여 기존의 중앙 통제식 Client-Server 컴퓨팅을 보완할 수 있는 Gnutella Network와 호환가능한 P2P 웹서버의 개발에 대해서 살펴본다.

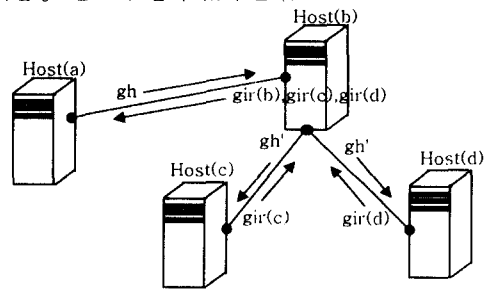
2. 관련 연구

중앙 집중식 웹서비스의 형태는 중앙서버에서 콘텐츠 및 Network를 관리하는 형태이다. 이와는 다르게 현재 분산 컴퓨팅에서 주목을 받고있는 P2P 서비스 방식은 Server와 Client 역할을 동시에 수행가능한 형태로서 컴퓨팅 객체의 의존성에 따라 중앙서버 없이 Peer가 동등한 조건을 갖는 Pure Distributed Computing Architecture와 기능이 강화된 Peer가 개입하여 다른 Peer를 관리하는 Hybrid Distributed Computing Architecture로써 현재 많은 연구가 되고 있으며, 또한 Peer간 콘텐츠 검색 능력을 강화시켜 콘텐츠를 인덱싱 할 수 있는 Index서버를 두는 방안에 대한 연구도 되고 있다.

본 개발에서 기반으로 한 Gnutella Protocol(Program)은 중앙서버가 없는 순수한 P2P의 형태를 취하고 있다. Gnutella Program은 기본적으로 TCP/IP프로토콜을 기본으로 삼고 있다. 대부분의 인터넷이 Server-Client 모델을 가지고 서비스를 수행하는 반면 Gnutella Program은 서버와 클라이언트

의 역할을 동시에 모두 수행하고, IP주소 및 파일검색에 필요한 6개의 Header를 사용하고 있으며 HTTP Protocol을 사용하여 데이터를 전송하게 된다.

그림 1의 host(a)에서 Gnutella Network로 연결되기 위한 필수 조건은 Gnutella Protocol을 사용하는 하나의 host(b)의 IP주소를 알고 있어야 한다는 것이다. host(a)는 host(b)로 접속하여 연결이 성립이 되면 host(a)는 Gnutella Header(gh)를 host(b)에게 보내게 되고, host(b)는 응답값으로 Gnutella Init Response(gir:IP, Port, File정보)을 보낸다. 다시 host(b)는 자신과 연결된 host(c,d)에게 host(a)가 보낸 gh값을 TTL값과 Hops수를 조정하여 보내면 host(c,d)는 응답값으로 gir(c,d)를 host(b)에게 보낸다. 여기서 host(b)는 캐쉬된 Host Information을 이용하여 host(a)을 찾아 gir(c,d)을 host(a)에게 넘겨주게되면, 결국 host(a)는 host(b)에게만 접속을 시도하였으나 host(b)가 relay역할을 수행하여 host(b,c,d)의 IP주소, Port번호, 공유된 파일정보를 모두 알 수 있게 된다.



[그림 1] Gnutella Protocol의 동작 원리

host에 대한 정보(IP, Port, File정보)을 파악하고 난뒤 각 host에게 원하는 검색어를 Query로 보내주면 각 host들은

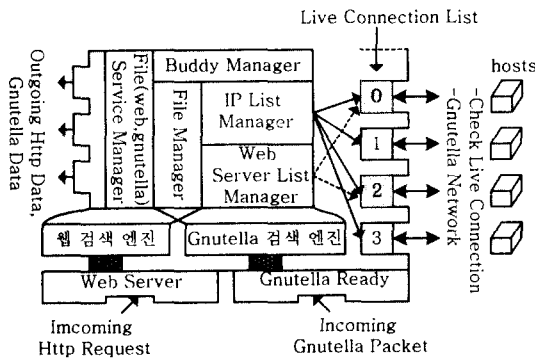
그것에 맞는 응답값을 요청한 host에게 넘겨주고 그중에서 원하는 데이터를 특정 host에게 요청하면 Http Protocol을 이용하여 데이터를 전송하게 된다

3. P2P Web Service 설계 및 개발

본 논문에서의 P2P Web Service는 순수한 P2P Web Application인 Gnutella Program(Protocol)을 기반으로 하여 여기에 자체 검색엔진 및 웹서버의 역할을 추가하고 강화시킨 P2P 웹서버를 개발하여 Dynamic IP를 가진 Host들에게 웹서버의 기능을 할 수 있도록 하는 것이다.

3.1 System 구성

시스템 구성은 그림 2와 같은 구조로 구성된다. Gnutella Protocol을 이용하여 host의 정보(Ip, Port, File정보)를 추적해내고, 입수한 정보는 List형태로 Buddy Manager가 관리하고, 그 정보들 중에서 Web Server가 탑재된 host의 정보를 가려내고 Live Connection 여부를 주기적으로 판단한다. 여기에 웹서버를 장착하여 Peer로의 http요청을 받아들이고 로컬 검색엔진이 해당 콘텐츠를 검색하여 검색결과 및 데이터를 서비스한다.



[그림 2] P2P Web Service 구성도

초기화시에 File Manager는 현재 공유하고 서비스할 디렉토리 및 파일들을 가려내고 구조화하여 서비스 요청에 대비하고, Gnutella Packet이 오면 대기하던 Gnutella Daemon은 자신의 정보(Ip, Port, File정보)를 해당 Peer에게 서비스 해주는 동시에 요청한 Peer의 정보를 획득하여 그 결과를 Buddy Manager에게 넘겨준다. Buddy Manager는 host정보를 Live Connection List에 저장하고 현재 Live한 IP Connection List들을 관리하고 주기적으로 Live한 Connection과 header를 주고 받으면서 Live여부를 조사한다. 또한 이때에 Web Server List Manager는 해당 List중에서 Web Server를 탑재한 host들을 가려낸다. 자체 웹 검색엔진은 Peer에서 Search Query가 왔을때 File Manager와 연계하여 해당 파일의 정보를 돌려주게 되면, 검색여부에 따라 http요청이 왔을 때 Web Server는 File(web) Service Manager를 통해서 http Protocol에 맞게 Encapsulation 한뒤에 전송하게 된다. 중요한것은 http요청을 하는 Client나 http서비스를 하는 Server나 똑같은 구조를 갖는다는 것이다.

3.2 Host주소 및 Port정보 추적

개발한 프로그램은 Gnutella를 기반으로 개발되었기 때문에 host의 IP, Port정보를 입수하는 것은 기본적으로 그림 1과 같은 동일한 접근방식을 사용했다. 그러나 본 논문에서 개발된 프로그램

은 Gnutella에서 사용하는 파일 공유용 Port 외에 웹서버 탑재 유무 및 대기하고 있는 웹서비스 Listen port을 알아내어 그 정해진 Port로 웹서비스를 요청한다.

Gnutella에서 사용된 Header 및 본 프로그램에서 정의된 Header의 역할은 표 1과 같다.

[표 1] P2P Web Server에서 사용된 Header

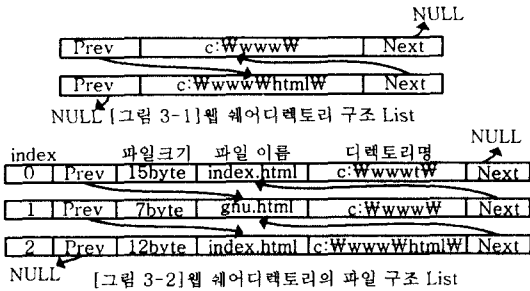
Header 이름	기능
GnutellaHeader(gh)	Gnutella Network에 참여한 Host에게 Ip, Port, File정보를 요구
GnutellaInitResponse(gir)	gh에 대한 응답값으로 자신의 IP, Port, File 정보를 담아 요청한 host로 gh와 함께 전송
GnutellaQuery(gq)	연결된 host에게 검색어를 전송할때 사용. 받은 host는 Query와 파일이름이 동일하면 일치한 것으로 간주한다.
GnutellaQueryResponseHeader(gqrh)	Query를 보낸 host에게 검색 결과를 보내줄때 자신의 IP, Port를 함께 보내준다.
GnutellaQueryResponseEntry(gqre)	현재 공유된 File List중에서 검색된 파일의 Index번호와 파일 Size정보.
(Ws)GnutellaPushRequest(gpr.wgpr)	검색된 파일이 있는 host가 Fire wall같은 것으로 숨겨있는 경우 정확성 및 무결성 확보를 위해 해당 host로 전송. 응답값으로 동일한 mySessionID값이 왔을때만 서비스 요청 시도.
WsGnutellaHeader(wgh)	Gnutella Network에 참여하는 Host에게 WebServer의 유무파악 요청
WsGnutellaInitResponse(wgir)	wgh에 대한 응답값으로 host의 IP, WWW Listen Port 정보를 전송
WsGnutellaQuery(wgq)	웹문서 검색요청 Query
WsGnutellaQueryResponseHeader(wgqrh)	wgq에 대한 응답으로써 서비스중인 웹문서중에서 검색된 문서의 총개수 및 Ip, Port, 접속유형을 요청한 host에게 전송
WsGnutellaQueryResponseEntry(wgqre)	서비스중인 웹문서 List중에서 검색된 해당 파일의 Index번호 및 Size.

3.3 P2P Web Server

본 프로그램이 초기화될 때 Gnutella share용 port 및 웹서버 Port를 초기화 한다. 그리고 첫 사용자는 Share할 Directory 및 Web Service Home Directory, 접속상황 및 검색에 대한 질의 기준등을 설정하게 되고, 이 정보는 레지스트리에 저장되어 다시 초기화될 때는 레지스트리 값을 기준으로 초기화를 시도한다. 그리고 중요한 것은 Gnutella Network로 진입하기 위해서는 1개의 Live한 IP주소를 알아야 한다는 것인데, 만약 나중에 Live한 IP를 알지 못했을 경우 이에 대해서 대처할 방법이 전혀 없다. 따라서 본 개발 프로그램에서 기본으로한 Gnutella 프로그램에서는 최후의 프로그램이 종료될 때 접속했었던 모든 Live한 IP주소와 Port정보를 host에 file의 형태로 저장시켜 놓는다. 따라서 프로그램이 초기화되면서 파일로부터 host정보들을 읽어들이고 각 호스트로 접속을 시도한다.

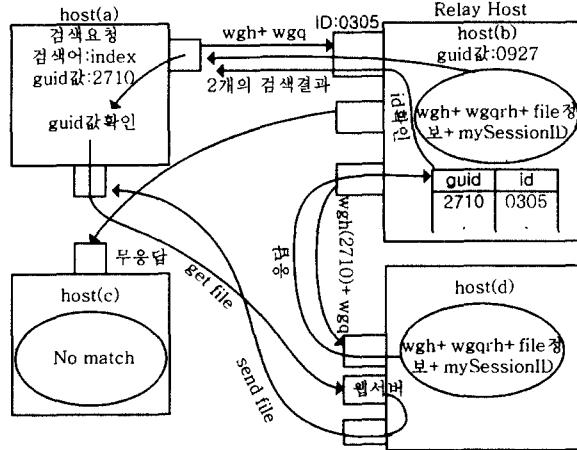
마지막으로 Gnutella Share Directory List 및 Home Directory File List 구조는 그림 3과 같이 Double Linked List로 구성하여 host들의 검색 요청에 효율적이고 신속하게 응답할 수 있도록 하였다. 검색 요청에 대한 응답값으로

WebServer Gnutella Query Response Entry(wgqre)를 보내는 데 여기에 사용되는 것이 index값과 파일크기이다.



3.4 P2P 웹 문서의 검색

P2P Network에서 웹문서의 검색은 gh, gir로 우선 host의 정보를 수집한 뒤에 wgh, wgr로 현재 host중에서 웹서버가 탑재된 host의 정보를 가려낸다. 그리고 난뒤에 해당 host로 웹문서 검색 Query를 보내게되면 로컬 검색엔진이 매치된 파일의 정보를 그림 4와 같이 검색 결과로 보내주게 된다.



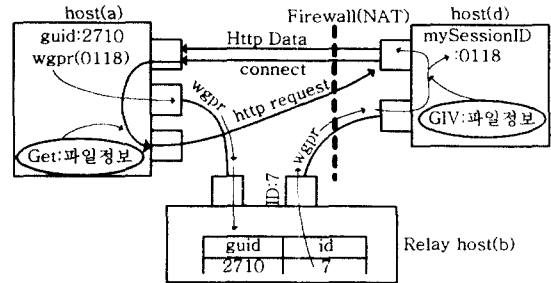
[그림 4] 웹문서 검색 원리

host(a,b,c,d)는 모두 그림 3과 같이 초기화 되어있고, 그림 4와 같이 host(a)는 host(b)의 정보만을 갖고 있으며, host(b)는 host(a,c,d)의 정보를 가지고 있다고 했을 때(추후에 host(a)는 그림 1의 처리로 인해서 host(b,c,d)의 모든 정보를 갖게 된다), host(a)가 검색 Query를 host(b)에게 보내면, host(b)는 파일구조에서 검색된 값(wgh+wgqrh+wgqre 및 파일정보+mySessionID)을 보냄과 동시에 현재 host(a)의 guid값과 자신의 연결id를 기억해 둔다. 그리고 host(c,d)에게 host(a)에서 받은 wgh+wgq를 보낸다. host(c)는 일치된 것이 없어 응답이 없으나, host(d)는 검색 결과를 host(b)에게 전송한다. host(b)는 자신의 guid값과 비교해본 결과 다르다는 것을 알고 기억해둔 guid값(2710)과 연결된 id:0305에 해당되는 연결 정보로 그 검색 결과를 relay시켜주게 된다. host(a)는 2개의 검색 결과를 차례로 받고 자신의 guid(2710)값과 비교하여 맞으면, 자신의 보낸 wgh에 대한 응답값으로 판단하고 host(b,d)의 웹서버 포트에 접속을 시도한다. http요청을 받은 각 호스트의 웹서버들은 요청한 페이지가 있는지 재확인후 새로운 연결을 열고 웹문서를 서비스한다.

결국, 그림 4와 같이 host(a)는 host(b)에게만 검색을 요청했지만 결국은 host(b,d)의 검색결과를 얻을 수 있게 된다.

3.5 host가 숨겨져 있을경우의 처리(firewall)

그림 5에서 host(b)가 firewall로 숨겨져 있을 경우에는 guid및 SessionID를 가진 wgpr을 사용한다.



(그림 5) host가 숨겨진경우의 웹문서 요청

host(a)는 모든 연결 노드에게 wgpr을 보낸다. 따라서 host(b)는 wgpr을 받고 과거의 검색결과에 대한 중계정보를 이용하여 숨겨진 host(d)로 중계시켜준다. host(d)는 SessionID를 확인한뒤 요청한 host(a)로 먼저 연결을 시도하고 연결이 확립이되면 host(a)가 보낸 http request를 host(d)가 받고,이에대한 응답 http Data를 host(a)에게 보낸다.

4. 결론 및 향후 연구

지금까지 Gnutella Protocol을 기반으로 한 Gnutella Network에서의 P2P 웹서버의 개발에 대해서 살펴보았다. 각 host의 정보획득의 원리는 기본적으로 Gnutella Program과 동일하고, 여기에 각 host마다 검색엔진 및 웹서버를 두어 웹문서 서비스를 가능하게 하였으며, firewall인 경우에는 wgpr를 사용하였으며, 호환성을 고려하여 Gnutella Network에서 기존의 Gnutella Program의 기능은 물론 추가적인 웹서비스의 기능을 가능하게 하였다. 본 논문의 주된 골격인 Gnutella P2P의 형태는 순수한 P2P 모델이다. 특정한 네트워크를 형성하고 있는 모든 host들은 리소스에 따라 차이가 있겠지만, Server-Client 모델에 구애받지 않는 동등한 권한을 가지고 있으며, 익명성을 지니고 있는 host들의 file을 공유하는 형태이다. 이와 같은 순수한 P2P모델을 이용하여 P2P환경에서 사용할 수 있는 웹서버를 개발하여 홈페이지를 갖고 싶어하는 많은 이용자들의 욕구를 해소하고, 또한 더 나아가서는 특정 Peer의 역할을 강화시켜 전자상거래 영역인 B2B로의 확장을 목표로 하고 있다.

향후 익명성 보장 및 P2P의 무간섭에 따른 보안에 관한 연구가 되어야 할 것이고, 웹컨텐츠의 종류에 따른 차별적 서비스에 대한 연구가 있어야 할 것이다.

참고 문헌

- [1] John M.A.Roy, "The Decentralized Web", 2000
- [2] Ju Zhang "A Distributed and Decentralized Information Publishing And Exchange System", Austin, Texas, 2000
- [3] Endeavors Technology, "Peer-to-Peer Architectures and the MagiTM Open - Source Infrastructure", 2000
- [4] Andy Oram, "PEER-TO-PEER", O'Reilly, 2001
- [5] <http://gnutella.wego.com/>