

# P2P환경에서 Gossip 프로토콜 기반의 메시지 구성 및 중복 제어

김병관\*, 구태완\*, 이성룡\*, 정연진\*\*, 이광모\*

\*한림대학교 컴퓨터공학과

\*\*춘천정보대학 멀티미디어과

e-mail: [kevin1975@hallym.ac.kr](mailto:kevin1975@hallym.ac.kr)

## Construction and Redundancy Control of Messages Using Gossip-based Protocol on P2P Environments

Byung-Kwan Kim\*, Tae-Wan Gu\*, Sung-Ryong Lee\*,

Yeon-Jin Jung\*\*, Kwang-Mo Lee\*

\*Dept. of Computer Engineering, Hallym University

\*\*Dept. of Multimedia, Chuncheon College of Information Technology

### 요 약

초고속 통신망의 발달과 개인 컴퓨터의 고급화가 이루어지면서 분산 환경에서 네트워크에 연결된 이런 고급화된 컴퓨터들의 유휴자원을 이용하여 대규모의 데이터를 처리하기 위한 연구와 기존의 클라이언트-서버 환경에서의 부하를 각 컴퓨터들에게 분산시키는 연구가 활발하게 진행되고 있다. 본 논문에서는 Peer-to-Peer환경에서 메시지 제어를 위한 mesh 기반의 네트워크를 구성하고 Gossip 프로토콜을 이용한 각 peer간의 효율적인 메시지 전송 기법을 제안한다.

### 1. 서론

인터넷의 활용이 급속도로 확산 되어지면서 그에 따른 네트워크 환경과 개개인의 PC도 점점 고급화 되어가고 있다. 이러한 사용환경의 고급화에 따라 P2P(Peer-to-Peer)환경을 이용한 분산 네트워크에 연결된 peer들의 유휴자원을 이용하여 대규모의 데이터를 처리하기 위한 연구가 활발하게 진행되어 오고 있다[1].

P2P는 크게 순수 P2P와 혼합형 P2P로 나누어지며 순수 P2P란 어떠한 서버의 개입 없이 peer들끼리 연결된 네트워크 환경으로써 대표적인 것으로, Gnutella와 Freenet이 있다. 혼합형 P2P는 peer들 사이의 상호작용을 원활하게 해주기 위해서 서버가 약간의 개입을 하는 형태의 네트워크 환경이며 대표적인 예로 Napster가 있다[3].

Napster는 모든 검색을 서버에서 처리를 하고 검색된 결과를 peer에게 전송하며[8], Gnutella는 검색을 위하여 브로드캐스트(Broadcast)를 사용하여 모

든 peer에게 메시지를 전송한다[12]. 하지만 Napster처럼 모든 검색을 서버에서 처리를 하게되면 서버에 너무 많은 과부하가 발생하게 되고, Gnutella에서처럼 검색을 위해 브로드캐스트를 사용하면 확장성의 문제를 가진다[13].

본 논문에서는 메시지 전송의 신뢰성 향상을 위해 mesh를 기반[4]으로 구성된 네트워크에서 Napster에서의 서버 과부하 문제와 브로드캐스트 프로토콜이 가지는 네트워크 확장성의 문제를 해결하고 peer간 발생하는 통신의 빈도 수를 줄이고자 Gossip 프로토콜[5,6]을 이용하여 메시지의 효율적인 전송과 중복되어 전송 받은 메시지의 처리, 그리고 메시지의 구성에 대한 방법을 제시한다.

### 2. 관련 연구

#### 2.1 Napster

Napster는 혼합형 P2P 방식 네트워크로 중앙에 서버를 두어 사용자들의 공유 파일 목록 등을 제어

하는 방법이다. 이러한 방법을 이용하면 검색의 속도나 검색 성공률을 높힐 수 있다. 그리고 실제적인 공유 파일의 전송은 클라이언트들 사이에서 발생한다[8,9].

Napster는 클라이언트들이 공유하고 있는 목록을 서버에서 관리하기 때문에 클라이언트가 자료 검색을 위해서 서버로 직접 검색을 요청한다. 검색 요청을 받은 서버는 저장된 목록을 기반으로 응답 가능한 클라이언트를 검색하고 해당 클라이언트의 정보를 최초 요청자에게 응답하게 된다. 이때 실제 자료의 전송은 클라이언트들 사이에서 이루어진다[8,9].

### 2.2 Mesh기반 네트워크

Mesh 기반 네트워크는 하나의 네트워크를 구성하고 있는 각각의 모든 컴퓨터가 별도의 케이블을 통해서 다른 모든 컴퓨터에 연결되는 것이다. 이렇게 연결된 네트워크는 트리 기반의 네트워크보다 좀더 향상된 잠복(latency) 감소의 성능을 제공한다[4]. 또한 전체의 peer가 n개일 경우 최소 1개에서 최대 n-1개의 peer들과 연결되어 서로 메시지를 주고받기 때문에 신뢰성 향상에 이점으로 작용한다[4]. 본 논문에서는 이러한 mesh 기반 네트워크의 장점을 이용하여 네트워크를 구성한다.

### 2.3 Gossip 프로토콜

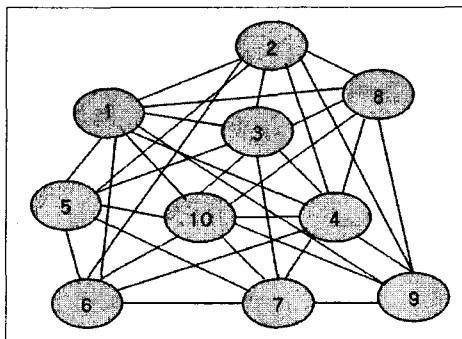
Gossip 프로토콜은 하나의 메시지를 전송 받은 peer는 자신 주변에 있는 임의의 peer에게 메시지를 전송하는 것이다. 이러한 Gossip 프로토콜은 메시지 전송의 높은 신뢰성을 제공하면서 확장성을 제공하기 위해 개발되었다[6,10]. Xerox Corporate Internet에서 복사된 데이터베이스 일관성 관리를 위해 처음 개발된 Gossip 프로토콜은 신뢰적인 multicast뿐만 아니라 실패 감지를 구현하려고 개발되고 있다. Gossip 프로토콜은 네트워크 상의 peer들의 개수에 독립적이고 단지 고정된 peer에게 메시지를 보내고 일반적인 multicast 프로토콜처럼 많은 동기화를 요구하지 않기 때문에 확장성을 가진다[5,7].

## 3. 본론

### 3.1 네트워크의 구성

본 논문에서 네트워크의 구성은 Napster와 같이 중간에 하나의 서버가 존재한다고 가정한다. 하지만 Napster처럼 모든 검색이나 검색 결과의 전송이 서

버를 통하지 않고 서버는 단순히 peer들의 목록만을 관리하여 새로운 peer의 등록에만 관여하게 된다. 그러므로 peer들은 처음 네트워크에 접근 할 때에만 서버에 확인 작업을 요청하고 peer들의 검색 요청이나 결과 전송은 peer들 사이에서 직접 발생하게 된다.



[그림 1] peer들의 구성 및 연결 상태

### 3.2 메시지의 구성

각 peer들 간에 주고 받는 모든 메시지는 XML 문서의 형태를 가진다. XML은 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준 마크업 언어로, 인터넷 상에서 가장 많이 사용하는 HTML의 단순함을 극복하고 SGML의 복잡성을 해결함으로써 다양한 형태의 정보 공유에 효과적이라는 커다란 이점이 있다[11].

먼저 각각의 peer는 최초 등록 시 각자의 컴퓨터에 하나의 Compare파일을 생성하는데, 이때 Compare파일은 로컬 IP 주소, Date, 그리고 Time으로 구성된 테이블을 가진다. 생성된 파일은 해당 peer가 다른 peer들로부터 메시지를 받아서 메시지 중복 여부를 판단 할 때 사용된다.

```
<?xml version="1.0" encoding="UTF-8"?>
<?ELEMENT MessageInfo (Header, Body)*>
<?ELEMENT Header (Originator, CreateTime)*>
<?ELEMENT CreateTime (Date, Time)+>
<?ELEMENT Body (Object)+>
<?ELEMENT Originator (#PCDATA)>
<?ELEMENT Date (#PCDATA)>
<?ELEMENT Time (#PCDATA)>
<?ELEMENT Object (#PCDATA)>
```

[그림 2] 새로운 peer등록을 위한 메시지

```

<?xml version="1.0" encoding="UTF-8"?>
<?ELEMENT MessageInfo (Header, Body)*>
<?ELEMENT Header (Originator, CreateTime)*>
<?ELEMENT CreateTime (Date, Time)+>
<?ELEMENT Body (Object, Retrieval)+>
<?ELEMENT Originator (#PCDATA)>
<?ELEMENT Date (#PCDATA)>
<?ELEMENT Time (#PCDATA)>
<?ELEMENT Object (#PCDATA)>
<?ELEMENT Retrieval (#PCDATA)>
    
```

[그림 3] 검색을 위한 메시지

[그림 2]와 [그림 3]은 메시지의 구성을 나타내는 DTD 문서를 보여준다. 메시지는 Header와 Body 두 가지 부분으로 구성이 된다. Header는 다시 Originator와 CreateTime으로 구성되는데 Originator는 최초 메시지 생성자의 IP 주소를 나타내며 CreateTime은 메시지의 생성된 날짜와 시간을 나타낸다. 메시지의 Body부분에는 메시지를 생성한 peer가 새로 등록된 peer인지 아니면 어떠한 검색이나 처리 결과를 원하는 판단하기 위한 내용이 들어간다.

생성자의 IP 주소는 서로 다른 각 peer들이 메시지 Body의 내용에 따라 새로운 peer의 등록을 알 수 있거나, 받은 메시지 내용을 처리한 후 그 결과를 최초의 생성자에게 보내줄 때 사용된다. 메시지의 생성된 날짜와 시간은 해당 메시지가 각 peer에 저장되어 있는 Compare파일과 비교하여 메시지의 중복 여부를 판단하기 위해 사용된다.

Body부분에는 해당 메시지의 실질적인 내용이 들어가는 부분으로 [그림 2]는 새로 등록된 peer를 나타내고, [그림 3]은 메시지 생성자가 자료나 정보를 검색 하기 위한 내용을 담고 있다. Body부분의 내용에 따라 결과 값의 전송을 하게 된다.

### 3.3 메시지의 전송

최초 서버에 자신을 확인한 후 mesh를 기반으로 구성된 네트워크는 전체 peer의 개수가 n개일 경우 적게는 1개에서 많게는 n-1개의 peer들과 서로 메시지를 주고 받으며 통신을 하게 된다.

[그림 1]에서 보이는 형태로 구성되면 메시지 손실을 줄이는 것과 같은 신뢰성을 향상시킬 수 있다. 하지만 하나의 메시지를 네트워크 상의 모든 peer에게 전송하고 또 모든 peer들로부터 메시지를 계속하여 받게되기 때문에 네트워크 상에 많은 트래픽이 발생할 수 있다. 또한 전송된 메시지에 대한 빈번한 확인 응답도 네트워크 상에 많은 트래픽을

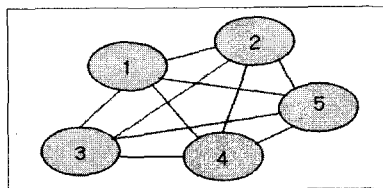
발생시킬 수 있다.

본 논문에서는 mesh 기반의 신뢰성을 보장하고 좀 더 향상된 확장성을 가질 수 있도록 메시지 전송을 위해 Gossip 프로토콜을 사용한다. Gossip 프로토콜을 사용하여 아래의 방법으로 수행한다.

- (1) 각 peer는 생성된 메시지의 전송을 위해 랜덤하게 몇 개의 peer들을 선택한다.
- (2) 선택한 peer들에게 메시지를 보내고 해당 메시지가 제대로 전송이 되었는지를 관여하지 않는다. 단순히 메시지를 전송하고
- (3) 일정 시간이 경과한 후 원하는 검색의 결과에 대한 응답이 없을 때에는
- (4) 다시 Gossip 프로토콜을 사용하여 해당 메시지를 재전송한다. 메시지를 받은 peer는 역시 마찬가지로 랜덤하게 peer를 선택하여 메시지를 전송하고
- (5) 메시지 생성자가 아닌 peer는 해당 메시지에 대한 처리 결과 응답을 기다리지 않는다. 단지 전송 받은 메시지에 대한 중계 역할만을 담당하게 된다.

### 3.4 메시지의 중복성 여부 판단

3.3에서 설명된 방법으로 메시지를 전송 받은 각 peer들은 해당 메시지의 Header와 자신의 컴퓨터에 있는 Compare파일을 아래의 방법을 이용하여 비교하게 된다.



[그림 4] peer들의 단편

[그림 4]는 5개의 peer로 구성된 하나의 단편이다. 위 그림에서 1번 peer가 메시지를 생성하여 2, 3, 4번의 peer들에게 메시지를 전송한다. 각 메시지를 받은 2, 3, 4번의 peer는 다시 다른 peer로 메시지를 전송한다. 이 때 만약 2번의 peer가 받은 메시지 전송시 3번의 peer에게 메시지를 전송하면 이미 1번 peer에게 받은 메시지가 있으므로 같은 메시지가 중복되어 전송받게 된다. 이 때 생기는 중복의 문제를 다음에 나오는 [그림 5]의 방법에 따라 해결 한다.

```

//메시지 Header 부분의 Originator을 비교
//파일에 해당 IP주소가 있는지 판단
compare(originator of Received MSG, each file);
if(ipNotExist(each file)) { //IP 주소가 없을 경우
//COMPARE에 새로운 IP주소를 추가하여 갱신한다
updateFile(fileName);
sendMessage(); //다른 peer에게 메시지를 전송
} else if(ipExist(each file)) {
//IP 주소가 있을 경우 COMPARE 파일의 Date를 비교
compare(date of Received MSG, each file);

//파일 Date가 메시지 Date보다 크거나 같을 경우
if(compareDate.compareTo(date of Received MSG) >= 0) {
deleteMessage(); //해당 메시지를 삭제
} else { //Date가 메시지 Date보다 작을 경우
updateFile(fileName); //파일의 내용을 갱신
sendMessage(); //다른 peer에게 메시지를 전송
}
}
    
```

[그림 5] 메시지 중복 처리를 위한 의사 코드 (pseudo code)

### 3.5 메시지 처리 결과 전송

3.3과 3.4에서 설명된 방법을 이용하여 메시지의 전송은 각각의 peer들에서 반복적으로 발생하게 되며 메시지를 받은 peer들은 Body부분을 확인하여 해당 메시지를 생성한 peer가 자신의 접속을 알리는 내용의 메시지가 아닌 어떠한 정보나 자료의 검색을 위한 메시지일 경우 자신의 공유 디렉토리를 확인 검색하여 일치되는 자료나 정보가 검색되면 처음 받은 메시지 Header 부분에 있는 Originator 주소로 결과를 직접 전송한다.

## 4. 결론 및 향후 연구과제

본 논문에서는 단순한 메시지 구성의 예와 함께 mesh 토폴로지 개념을 기반으로 하는 peer간의 네트워크 구성과 Gossip 프로토콜을 이용한 메시지의 전송 방법을 제안하였다. 이를 기반으로 Napster의 서버와 클라이언트 사이의 peer 등록, 검색 요청, 검색 결과 전송과 같은 메시지에 의해 발생하는 많은 통신을 줄이고자 하였다. 앞으로는 좀 더 동적인 네트워크 구성을 위한 방법을 연구하고 3.1에서 보여준 메시지 구성보다 좀 더 효율적이고 상세화 시킴으로써 단순한 정보나 자료의 검색과 전송이 아닌 더욱 실질적인 처리 작업을 할 수 있는 것이 필요하다.

### 참고문헌

[1] 이광현, 전형수, 유철중, 장옥배 "서버 부하 감소

를 위한 P2P 기반 데이터 서비스 시스템의 설계 및 구현", 한국정보처리학회 논문지 C, Vol.9-C, No.05, pp. 0615 - 0626.

[2] <http://www.web-biz.pe.kr/web/p2p1.html>

[3] Siu Man Lui, Sai Ho Kwok "Interoperability of Peer-to-Peer File Sharing Protocols", ACM Sigecom Exchanges, Vol.3, No.3, August 2002, Pages 25-33.

[4] Alex c. Snoeren, Kenneth Conley, David K. Gifford "Mesh-Based Content Routing using XML", 18th ACM Symposium on Operating System Principles(SOSP '01).

[5] Meng-Jang Lin, Keith Marzullo "Directional Gossip: Gossip in a Wide Area Network", European Dependable Computing Conference(1999), p. 364 - 379.

[6] Meng-Jang Lin, Keith Marzullo, Stefano Masini "Gossip versus Deterministic Flooding: Low Message Overhead and High Reliability for Broadcasting on Small Networks", Distributed Computing, 14th International Conference, DISC 2000, October, 2000

[7] Ayalvadi J., Anne-Marie Kermarrec, Laurent Massoulie "Peer-to-Peer Membership Management for Gossip-Based Protocols", IEEE Transactions on computers, Vol.52, No.2, February 2003.

[8] Napster Inc.  
<http://www.napster.com>

[9] <http://www.web-biz.pe.kr/web/p2p2.html>

[10] David Kempe, Jon Kleinberg "Protocols and Impossibility Results for Gossip-Based Communication Mechanisms", The 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02) November, 2002.

[11] 최문경, 김행곤 "XML 기반 교수-학생 학습지도 시스템의 설계 및 구현", 정보처리학회논문지D 제9-D권 제6호(2002.12).

[12] <http://www.gnutella.com>

[13] Kenneth P. Birman, et "Bimodal multicast." Cornell University, Department of Computer Science Technical Report TR-98-1665, May 1998.