

# 네트워크 식별자 기반 P2P 파일 공유 네트워크

박찬모\*, 이정기\*, 배일호\*, 정성옥\*\*, 이 준\*\*\*

\* 조선대학교 컴퓨터공학과

\*\*광주여자대학교 컴퓨터과학과

\*\*\* 조선대학교 컴퓨터공학부

## P2P File Sharing Network based on Network Identify

Chan-Mo Park\*, Jeong-Ki Lee\*, Il-Ho Bae\*, Sung-Ok Jung\*\*, Joon Lee\*\*\*

\* Dept. of Computer Engineering, Chosun Univ.

\*\* Dept. of Computer Science, Gwangju Women's Univ.

\*\*\* School of Computer Engineering, Chosun Univ.

### 요 약

최근 수 년간 클라이언트/서버 모델에서 발생하는 문제인 서버 병목 현상, DoS(Denial of Service) 공격, 그리고 시스템의 확장성에 따르는 비용 증가 등의 문제를 해결하기 위한 방법으로 P2P (Peer-to-Peer) 방식이 활발히 연구되고 있다. P2P 방식을 사용하는 모델은 기존의 클라이언트/서버 모델이 적용되었던 여러 분야에 이용되고 있다. 그 중 사용자간 파일 공유하는 분야에 적용된 P2P가 가장 널리 알려져 있다.

본 연구는 P2P 파일 공유 어플리케이션을 위한 효율적인 네트워크 생성 및 검색 프로토콜을 제안한다. 이것은 동일 네트워크 식별자를 통한 그룹화로 전체 논리 노드 수를 감소시킴으로써 물리적인 네트워크의 라우팅에 근접해 가도록 한다.

### 1. 서론

P2P(Peer-to-Peer) 컴퓨팅은 노드들이 가진 자원을 공유하며 다른 노드들이 공유 자원들을 직접 액세스한다. 또한 모든 노드들은 서버와 클라이언트 기능을 함께 갖고 있는 것으로 정의한다[9]. P2P 컴퓨팅의 한 분야인 파일 공유는 가장 널리 사용되는 분야이다.

중앙 집중식 모델은 중앙 서버에 병목 현상을 야기하고 DoS(Denial of Service) 공격에 대한 취약성 및 서비스의 확장 비용 증가 등의 문제가 발생한다. 이를 해결하기 위해서 순수 분산형 P2P 파일 공유에 대한 연구들이 진행되었다. 이들은 중앙의 서버를 가지지 않고 임의의 연결을 통해 완전히 분산되어 있으므로 네트워크 연결 및 탐색에 이웃 노드 전체에 메시지를

보내는 Flooding 방식을 사용하였다. 이것은 중복 메시지로 인한 네트워크 트래픽의 증가를 가져왔다. 이로 인해 P2P 네트워크의 유지 및 탐색의 비용을 증가시키게 되어 확장성의 문제가 대두 되었다.

Flooding으로 인한 문제를 해결하기 위한 방법으로 분산 P2P 네트워크에 구조적인 특성을 추가하여 전체 네트워크에 대한 정보를 가지도록 하는 연구들이 진행되었다. 이들은 네트워크를 구성할 때 이웃 노드들이 일정한 규칙을 가지도록 하였다. 또한 각 노드들은 검색의 효율성을 위해 일정한 범위의 파일들에 대한 디렉토리 서비스를 제공하도록 하였다. 탐색은 파일에 대한 디렉토리 서비스를 하는 노드를 연결함으로써 가능하다. 노드 위치를 지정하기 위한 방법으로 분산 해쉬 테이블이 사용된다. 하지만 이들 연구들에서 검색

및 메시지 라우팅의 효율성은 증명되었지만 다른 문제점들이 지적되고 있다. 논리적으로 연결된 P2P 네트워크에서 메시지 라우팅 및 검색의 성능은 향상되었지만 물리적인 네트워크의 라우팅과 차이를 갖는다는 것이다. 또한 hotspot으로 인한 로드 밸런싱의 문제로 특정 노드의 과부하가 문제가 되고 있다. 전자는 물리적으로 근접한 노드나 RTT(Round Trip Time)가 적은 노드를 인접 존에 두는 방식을 통해 해결하려고 하고 있다. 후자는 캐시를 사용하거나 과부하된 노드의 인접 노드에 복제를 하는 방식을 사용하여 해결하려고 하고 있다.[1, 6, 10, 11]

본 논문은 P2P 파일 공유 네트워크의 논리적 네트워크 라우팅이 물리적 네트워크 라우팅에 근접하도록 하기 위해 네트워크 식별자를 사용, 동일한 네트워크상의 노드들을 그룹화하여 논리적인 네트워크상의 노드 수를 감소시킴으로써 라우팅 Hop이 감소되도록 하여 성능을 향상시키도록 한다. 제안하는 네트워크 topology는 CAN과 유사한 방법을 사용하여 형성한다.

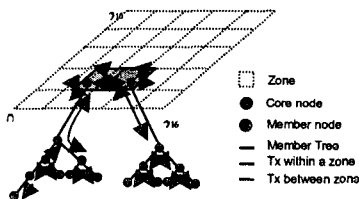
## 2. 네트워크 식별자 기반 P2P 파일 공유 프로토콜

### 2.1 프로토콜 개관

본 논문은 CAN과 같이 구조적인 네트워크를 생성한다. CAN과 제안하는 프로토콜과의 차이점은 다음과 같다.

- 노드의 존 매핑을 위해 2차원 좌표 공간을 사용한다.
- 노드의 존 매핑 위치는 임의로 지정되지 않고 네트워크 식별자를 사용하여 결정한다.
- 존 내에는 코어 노드와 멤버 노드로 구성된다. 코어 노드는 존을 관리하고 이웃 존과 메시지를 교환한다.
- 존 내의 코어와 멤버 노드들은 공유 이진 트리를 통해 메시지 전송을 한다.
- 존 분할은 동일한 크기로 분할하지 않는다.

그림 1은 제안 프로토콜에 따라 생성되는 전체 네트워크의 구성을 보여주며 하나의 존에는 하나의 코어와 다수 개의 멤버 노드들로 구성됨을 보여준다. 또한 존 내의 멤버 노드와 코어 노드는 공유 이진 트리로 메시지를 통해 전송함을 보여준다.

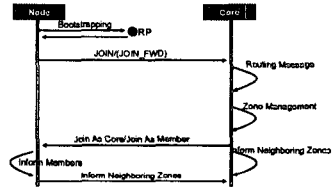


[그림 1] 전체 구성도.

### 2.2 네트워크 생성

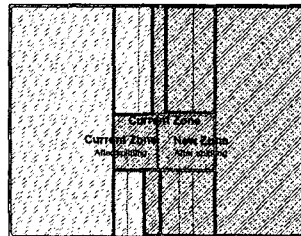
새로운 노드가 네트워크에 연결을 하는 경우 그림

2에 보여지는 과정에 따른다. 연결을 원하는 노드는 란데부(Rendezvous Point)가 되는 이미 알려진 노드로부터 현재 네트워크에 참여 중인 노드의 IP 주소와 포트 번호에 대한 정보를 얻는다. 참여 중인 노드 정보를 사용하여 연결 요청을 자신의 네트워크 식별자와 함께 보낸다. 연결 요청을 수신한 노드는 자신이 존의 코어가 아닌 경우 존의 코어 노드에 직접 전달하여 요청자의 네트워크 식별자를 포함하는 존으로 메시지가 라우팅(message routing) 되도록 한다. 영역을 관리하는 존의 코어가 연결 요청을 수신하면 자신의 네트워크 식별자와 비교하여 동일한 경우, 현재 존의 멤버 노드로 연결을 하도록 한다. 그렇지 않은 경우, 자신이 관리하는 영역을 분할하여 새로운 존을 생성하고 자신의 존을 조정한다. 존 분할이 발생하면 이웃 존과 현재 존의 멤버 노드들에게 존 변경을 알린다. 분할로 생긴 존은 연결 요청을 한 노드가 관리하며 자신의 이웃 존들에게 자신의 영역 정보와 코어 노드 정보를 알린다.



[그림 2] 새로운 노드의 조인

### 2.3 존의 분할 및 공유 트리 관리

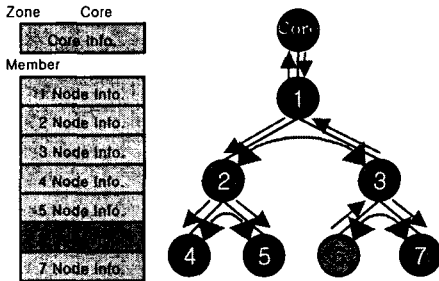


[그림 3] 존 분할.

연결을 요청하는 노드의 네트워크 식별자가 존의 영역 내에 속하는 경우, 존은 분할된다. 존 분할은 X와 Y축의 방향으로 분할된다. 분할 방향의 선택은 현재 존의 네트워크 식별자와 각각 X와 Y축의 차를 구한 후 절대값이 큰 방향으로 결정된다. 또한 존 분할 경계는 두 값의 중간 값으로 정해진다. 존이 분할되면 유지하는 이웃 존은 재 배정된다. 재 배정된 결과에 따라 이웃 존에 현재 존 및 새로 생성된 존의 정보가 전달되어 변경을 반영토록 한다. 또한 현재 존은 변경된 정보를 현재 존의 멤버 노드들에게 알려 존 영역 정보를 변경토록 한다. 그림 3은 존 분할 및 분할 후 이웃 존에 대한 재 배정을 나타낸다. 분할로 새로 생성된 존에서는 새로 네트워크에 연결된 노드가 코어가 된다. 새로 네트워크에 참여하는 노드가 현재 존의 영

역 내에 있고 네트워크 식별자가 동일한 경우 존의 멤버 노드로서 연결된다.

존 내의 모든 노드들은 현재 멤버 노드들에 대한 정보를 가지고 있으므로 연결 노드는 현재 존의 정보를 받아 자신을 포함한 트리를 재구성한다. 트리를 구성하기 위해 IP를 정렬하여 이전 트리로 구성한다. 이 트리는 존 내에 모든 노드들이 공유한다. 존 공유 트리에서 메시지는 상위와 하위 노드로 전달되며 메시지를 수신한 경우, 수신 노드를 제외한 방향으로 메시지 전송을 한다. 그림 4에서 새로운 노드 6이 네트워크에 멤버로 참여하는 경우 트리 생성 및 새로운 노드 참여 정보의 전달을 실선 방향으로 이루어짐을 보여주고 있다. 또한 다른 노드에서 수신된 메시지 전달은 코어 노드로부터 점선을 따라 전달된다.

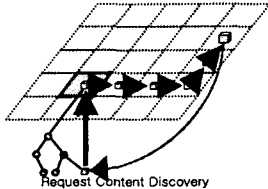


[그림 4] 존 내부의 공유트리.

연결요청 메시지는 연결 노드의 네트워크 식별자를 포함하는 존으로 전달되어야 한다. 이를 위해 연결 요청 메시지를 수신한 존의 코어는 이웃 존 정보를 이용하여 이를 전달한다. 메시지의 전달은 항상 X축 방향을 전달하며 네트워크 식별자의 X축 값을 만족하는 경우 X축 값을 포함하는 Y축으로 전달이 이루어진다.

2.4 검색

파일을 검색하기 위해서는 파일 이름을 해쉬한 값(즉, 공간상의 좌표 값)을 포함하는 메시지를 파일 공유 네트워크에 전송한다. 요청하는 노드가 존의 멤버 노드인 경우 현재 존의 코어 노드에 전달한다.



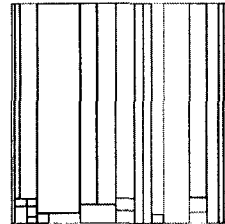
[그림 5] 검색

각 코어 노드는 메시지에 포함된 좌표 값과 이웃 존의 비교를 통해 전달된다. 해쉬 값의 X를 관리하는

존에 도달할 때까지 X축을 따라 전달된 후 Y축으로 전달된다. 그림 5와 같이 해당 파일을 관리하는 노드가 메시지를 수신한 경우 직접 요청 노드에 응답 메시지를 보낸다.

3. 실험 및 결과

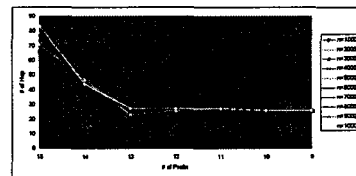
실험을 위해 본 논문에서는 GT-ITM[7]을 사용하여 인터넷을 모델링하기 위해 10,100개의 노드를 생성하고 임의로 추출된 노드들을 사용하여 실험하였다. 추출된 노드들의 P2P 네트워크 생성 및 검색을 위해 시뮬레이터를 구현하였다. 시뮬레이터 구현은 위스콘신 대학의 myns[8]을 기반으로 구현하였다. 광대부 포인트 노드는 노드 0을 지정하여 사용하였다. 각 노드들이 맵핑 되는 2차원 공간은 실험의 복잡성을 줄이기 위해 X와Y축 모두 8비트를 사용하였다. 따라서 실험에 사용되는 노드들의 IP 주소는  $2^{16}$  범위 내 노드들이다.



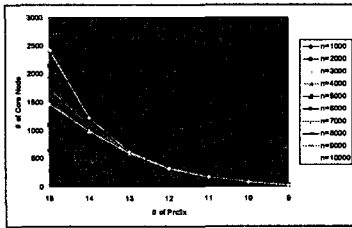
[그림 6] 노드 공간 분할 (30개의 존).

그림 6은 30개의 노드를 임의로 선택하고 각 노드의 prefix값을 15비트를 사용하여 P2P 네트워크를 형성한 후의 2차원 공간 분할 결과를 보여준다. 분할 공간은 28개의 존으로 구성되어 있으며 나머지 2개 노드는 멤버 노드로 참여하였음을 알 수 있다.

그림 7과 그림 8은 노드 수를 고정하고 네트워크 식별자 prefix의 크기를 감소시킴으로써 한 영역에 속하는 노드의 수가 증가하도록 했을 때 검색의 Hop수와 코어 노드의 수 변화를 보여준다. 검색은 0번 노드에서 해쉬 값 65653을 사용하여 X와 Y 축의 최대 값이 되는 존으로 라우팅 되도록 했을 때 전체 Hop 수이다.

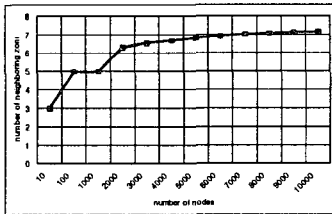


[그림 7] 검색 Hop 수



[그림 8] 코어 노드 수

검색의 경우, Hop수는 네트워크 식별자의 범위가 커짐에 따라서, 즉 코어 노드 수가 감소함에 따라 감소됨을 알 수 있다. 이것은 하나의 존에 많은 노드들이 멤버 노드로 참여하여 전체 존의 수가 감소했기 때문이다. 또한 동일한 네트워크 prefix를 가진다는 것은 동일한 네트워크상에 노드들이 있을 확률이 높기 때문에 물리적인 네트워크의 IP 라우팅의 성능에 가까워진다고 말할 수 있다. 하지만 너무 많은 노드가 하나의 존에 있는 경우 존 내부의 라우팅에 많은 영향을 끼치므로 존은 그림 8과 같이 일정한 수준을 유지하는 정도의 네트워크 prefix를 사용하여 전체 존을 유지하는 것이 중요함을 알 수 있다.



[그림 9] 이웃 존의 평균 수(prefix=15)

그림 9는 각 존이 가지는 평균 이웃 존 수의 변화에 대한 결과를 보여준다. 노드의 수가 증가함에도 불구하고 이웃 존의 수는 평균 7개에 근접해 있음을 보여준다. 이것은 제한하는 프로토콜이 확장성을 가짐을 보여주는 것이다.

#### 4. 결론 및 향후 연구방향

본 논문은 실험에서 살펴본 바와 같이 P2P 파일 공유 네트워크의 논리적 네트워크 라우팅이 물리적 네트워크 라우팅에 근접하도록 하기 위해 네트워크 식별자를 사용해 동일한 네트워크상의 노드들을 그룹화함으로써 논리적인 네트워크상의 노드인 존의 수를 감소시킴으로써 검색 Hop의 수가 감소함을 보였다. 또한 이것은 동일 네트워크 식별자는 동일한 그룹에 속하게 함으로써 물리적인 네트워크의 라우팅에 근접하는 메시지 라우팅을 가능하다고 할 수 있다. 또한 노드 수를 증가시킴에 따른 이웃 존의 수가 7을 유지하기 때문에 확장성을 갖고 있다고 할 수 있다.

하지만 여전히 물리적 라우팅에 근접하기 위한 보다 많은 노력이 필요하다.

향후 연구 방향은 존 내에서 새로운 층을 추가하여 다시 근접성을 측정하여 물리적 라우팅에 근접하도록 하는 지속적인 연구가 필요하다.

#### [참고문헌]

- [1] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron, "Topology-aware routing in structured peer-to-peer overlay networks," in *FuDiCo 2002: International Workshop on Future Directions in Distributed Computing*, Italy, June 2002.
- [2] N. Blundell and L. Mathy, "An overview of Gnutella optimisation techniques," *PGNet2002*, 2002.
- [3] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer file sharing technologies," *Athens University of Economics and Business*, 2002.
- [4] Clarke, O. Sandberg, and B. Wiley, "Freenet: A distributed anonymous information storage and retrieval system. Designing privacy enhancing technologies," in *Proc. International Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009*, Springer, 2001.
- [5] Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in *Proc. ACM SIGCOMM*, 2001.
- [6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. ACM SIGCOMM*, 2001.
- [7] GT-ITM: Georgia Tech Internetwork Topology Models, <http://www.cc.gatech.edu/projects/gtitm>
- [8] myns, <http://www.cs.umd.edu/~suman/research/myns>
- [9] R. Schollmeier, "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications". *IEEE Conference P2P 2001*, Linköping Sweden. August 2001
- [10] Zhichen Xu, Chunqiang Tang, and Zheng Zhang "Building Topology-Aware Overlays using Global Soft-State". *The 23rd International Conference on Distributed Computing Systems*. May 19-22, 2003 Providence, Rhode Island USA
- [11] Marcel Waldvogel and Roberto Rinaldi, "Efficient Topology-Aware Overlay Network", *Hot Topics in Networks I (HotNets-I)*, October 2002. The HotNets-I proceedings will also appear in *Computer Communication Review*, Volume 33, Number 1, January 2003