論文
4-12-1

ISP별 IP분포를 고려한 비구조적 Peer-to-Peer에서의 Lookup 기법

An Efficient Lookup Mechanism for Unstructured Peer-to-Peer Considering IP Distribution

김홍일 (Hong-IL Kim)[1] 신판섭(Pan-Seop Shin)[2]

요약

비구조적 Peer-to-peer 환경에서의 소속 네트워크에 포함된 호스트를 찾는 Lookup 기능은 P2P 환경의 네트워크 트래픽에 상당한 영향을 주는 요소이다. 특히 사용자 접속이 임의로 가입과 탈퇴가 빈번하고, 유동 IP를 사용하는 사용자들로 주로 구성된 P2P 자료 공유 환경에서는 Lookup 기능에서 현저하게 많은 자원이 소요된다. 본 논문에서는 국내의 ISP(Internet Service Provider)별 보유 IP를 분류한 ISP Key 값과 공유된 정보에 대한 Key를 SHA-1 해싱을 통하여 추출된 값을 조합하여 Lookup을 찾는 방법을 사용한다. 제안된 방법은 ISP들의 Routing을 고려하여 Lookup을 수행하기 때문에 자료의 검색 및 조회에 있어서 효율성을 기대할 수 있다. 또한 기존에 널리 사용되고 있는 emule P2P 공유 시스템과의 연동을 통한 시뮬레이션 환경을 제시하고 실험한다.

ABSTRACT

Lookup is one of important factors for network traffic on peer-to-peer network. Especially, large amount of network resources are required for lookup process in p2p file sharing system that clients use the dynamic IP and they import to p2p network in random. In this paper, we employ efficient lookup mechanism which searches lookup with the combined information of ISP key values and the extracted key values from SHA-1 hashing function for shared contents id. The proposed mechanism efficiently searches files on the network because the mechanism processes P2P lookup with consideration the routing information of ISP. We suggest the adapted simulation environment based upon the famous file sharing system of emule P2P and experiment the proposed system.

논문접수 : 2003. 11. 20.
심사완료 : 2003. 12. 5.

1) 정회원 : 대진대학교 컴퓨터공학과 조교수
2) 정회원 : 대진대학교 컴퓨터공학과 전임강사

1.서론

P2P(peer-to-peer) 방식은 자료의 공유 및 검색, 콘텐츠 통합등의 응용에서 매우 매력적으로 동작하는 인터넷 응용중 하나이기 때문에 공유 용량, 성능, 인접 서버의 선택, 검색, 인증 등의 많은 분야에 대한 활발한 연구가 진행 중이다.[1]

일반적인 P2P 자료 공유 방식에서는 사용자 질의를 수행하기 위하여 공유서버에 직접 질의를 요구하거나 IP 서버로부터 제공받은 공유 네트워크에 포함된 노드에 질의를 의뢰하는 방식을 사용한다. Gnutella와 같은 비구조적(Unstructured) P2P 시스템의 경우, 별도의 서버 시스템 없이 공유 네트워크에 포함된 노드들이 개별적인 servent(server + client)로 동작하여 사용자 질의를 처리하는 방식을 이용하고 있다.[2][3] 비구조적 P2P의 각 노드들은 공유 네트워크에 포함된 이웃한 servent들에 대한 정보를 얻기 위하여 lookup 기능을 수행하게 된다. Gnutella의 경우 인접한 노드들에게 질의를 전달하고 해당 노드가 다시 질의를 그 노드에 인접한 노드들에게 전이하는 방식으로 질의가 전달된다. 또한 수시로 변경되는 네트워크 환경을 반영하기 위한 제어 정보들의 전달까지 포함하면 네트워크 유지 및 질의 수행을 하기 위한 lookup 과정에서의 트래픽량이 매우 높다는 문제점을 가지고 있다.[1][2][3] 비구조적 P2P에서의 lookup 부하를 줄이기 위한 연구로는 chord 시스템이 있다. chord에서는 lookup을 찾기 위한 공간을 줄이기 위해 ID 정보 값과 IP값을 SHA-1 방식으로 요약하여 이를 키 값으로 하는 해싱 테이블을 구성하고, 구성된 해싱 범위 안에서 자료를 찾는 방법으로 사용자 질의 수렴 속도를 개선하는 방안을 제시하였다.[1][5] Chord에서 IP 주소를 해싱 연산에 포함시킨 것은 계층 구조를 가지는 IP 주소의 연계성을 이용하여 routing과 실제 자료를 다운 받을 때, 인접한 노드를 우선하여 선택할 수 있다는 장점을 가지고 있다. 그러나 국내의 경우, 각 인터넷 서비스 공급업체(ISP:Internet Service Provider)가 보유하고 있는 IP주소 값은 동일한 ISP의 경우에도 거의 대부분의 주소 공간이 연계성 없는 무작위 순서이며, IPv4 주소 공간의 고갈이 인접하면서 기존의 IP 주소 배정 체계를 따르지 않고 분배된(A class에 해당됨에도 C class 단위로 부여) 주소 공간을 다수 내포하고 있는 실정이다. Chord와 같이 IP주소에 대하여 SHA-1으로 산출하여 구성된 bit값은 국내의 경우 연계성을 기대할 수 없다. 또한, P2P 공유 서비스가 유동 IP를 사용하는 가정용 회선 사용자들이 주로 사용하기 때문에 IP 주소 값이 가지는 의미는 고정 IP에 대비하여 상대적으로 희박하다. 따라서 국내의 주소 공간 환경과 빈번하게 변동되는 유동 IP 주소의 노드에 대한 인덱싱을 함께 고려한 비구조적 P2P 시스템에서의 효율적인 lookup 체계가 필요하다.

본 논문에서는 현재(2003년 8월) 국내의 77개의 ISP들이 보유하고 있는 IP 주소공간을 몇 개의 ISP 그룹으로 별도 분류하여 이에 대한 분류코드를 배정하고, IP 주소 변동 시에도 콘텐츠의 동일성을 추측할 수 있는 콘텐츠 id에 대한 SHA-1 요약 값을 조합하여 lookup 전이의 요소로 사용하고자 한다. 따라서 lookup 검색은 1차적으로 ISP 분류 코드에 대하여 분류된 소규모 네트워크에서 콘텐츠 id에 대한 SHA-1 해싱을 통하여 추출하는 방식을 이용한다. 제안된 방법은 기존의 DHT(Distributed Hash Table) 이용한 lookup 검색에 비하여 작은 범위 내에서 검색을 수행하기 때문에 검색 효율이 높을 뿐만 아니라 ISP 그룹별 검색을 수행하기 때문에 lookup에 소요되는 지연(latency)도 작게 나타난다. 또한, 축소된 공간에서의 검색으로 인한 lookup 검색 실패 시에는 좀더 규모가 큰 상위 네트워크에 대한 검색으로 검색 공간을 확장하여 원하는 정보가 네트워크에 존재하는 경우에는 성공적인 검색이 수행될 수 있도록 알고리즘을 설계하였다.

또한, 본 논문에서는 제안된 시스템의 가상 실험을 위하여 현재 공개 소프트웨어로 source code가 공개되어 있고 많은 사용자 층을 가지고 있는 emule P2P 공유 프로그램을 배경으로 실험 환경을 제시하고, 제안된 알고리즘은 적용하여 시뮬레이션 하였다.

2.관련연구

P2P 기반 정보 공유로는 Napster가 가장 널리 알려져 있다. Napster에서는 노드 정보와 공유 자료 정보 등의 관련 참여자 정보를 중앙 서버에 모

아 관리하는(찾기, 할당, 수정, 가입, 탈퇴 등) 중앙 집중식 방법을 사용한다. 각기 다른 곳에 분산되어 있는 자원을 이용하려는 사용자는 먼저 중앙서버에 연결하여 원하는 자원에 관한 속성(attribute)를 질의하면 현재 연결 가능한 노드의 명단과 공유 목록을 볼 수 있으며, 사용자가 해당 노드를 선택하면 직접 연결되어 해당 자원을 조회할 수 있다. 파일 공유의 경우, 사용자가 요청한 해당 파일이 요구 노드의 적절한 위치로 복사된다. Napster는 찾기에 소요되는 시간은 최소이나 중앙 서버에 집중화되는 가용성 측면의 취약성을 지니고 있다.

P2P 기반의 공유 형태 중, 두 번째로는 현재 널리 사용되는 edonkey 공유 서비스 형태가 있다. edonkey에서는 네트워크 전체를 여러 개의 부 네트워크로 분리하고 각각의 부 네트워크를 관장하는 IP 서버(edonkey 서버)를 둔다. 각 IP 서버는 해당 서버에 접속한 노드들에 대한 IP를 데이터베이스화하여 관리한다. edonkey에서의 검색은 IP 서버로부터 제공받은 부 네트워크의 IP 정보를 이용하여 개별의 노드들이 상대 노드에게 검색 질의를 전달하고 검색 질의를 받은 노드가 해당 질의를 수행한 결과를 질의 발생 노드로 반환하는 형태로 이루어진다. 이때 반환되는 결과의 내용이 다수인 경우 사용자가 미리 설정한 초기값에 따라 먼저 결과가 도착한 노드들에게 정보를 나열하는 방식으로 질의가 수행된다. (edonkey와 동일한 서버 체제를 가지는 emule의 경우, 반환 결과 값을 200개로 한정한다)

edonkey 공유 서비스가 가지는 또 다른 특징으로는 공유 자원을 여러 개의 분할된 조각으로 나누고 각각의 조각들을 동일한 콘텐츠를 보유한 서로 다른 노드로부터 분산시켜 제공받을 수 있는 특징을 가지고 있다. 따라서 특정 콘텐츠에 대한 완벽한 정보가 존재하지 않아도 현재 네트워크 상에 공유된 자료 부분을 여러 개의 분산된 노드들로부터 제공 받을 수 있다. edonkey와 같이 별도의 IP 서버를 중심으로 복수개의 네트워크로 구성된 서비스는 현재 P2P 자료공유에서 매우 많이 응용되고 있다. edonkey에서 실행되는 서버는 자체 서버와 일부 P2P 마니아가 자발적으로 운영하는 서버 형태로 나뉘어 진다.

Gnutella는 위의 두 가지 형태와 다르게 어떠한 서버도 존재하지 않는 순수(pure) P2P 응용 시스템이다. Gnutella와 같이 어떠한 서버도 존재하지 않는 P2P 형태를 비구조적(unstructured) P2P라 명명한다. Gnutella에서는 분산된 자원에 관한 정보를 중앙집중식으로 관리하지 않고 각기 분산된 형태 그대로 이용한다. 사용자의 검색 요청 질의는 다음과 같이 수행된다. 먼저, 찾기 요청을 받은 노드는 자기가 해당 자원을 가지고 있는가를 검사하

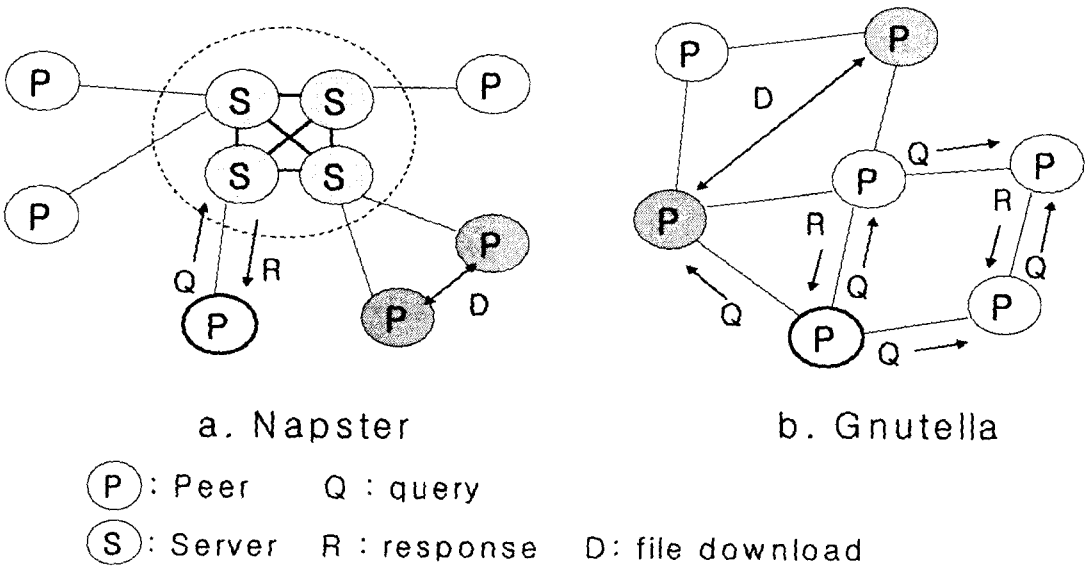

그림 1. Napster와 Gnutella의 비교

고, 그렇지 않을 경우, 자신이 알고 있는 다른 노드로(또는 직접 연결된) 찾기 요청을(flooded queries) 동시에 전송한다. 이 중, 하나의 노드에서 찾기 결과가 반환되면 그 결과를 이용하고 이후에 도착하는 찾기 결과는 무시한다. 최악의 경우는 가장 먼 곳의 위치한 단 한 곳에 자원이 존재하는 경우이며, 이럴 경우 수많은 네트워크 자원(bandwidth, 사용되는 메시지 수 등)을 낭비하게 된다. 이 경우에는 O(n)의 메시지가 찾기 연산 시 필요하게 되어 단일 경로를 순차적으로 거쳐서 해당 자원의 위치를 파악하는 순차적인 찾기 방법과 동일하게 된다.

Chord는 MIT에서 고안한 DHT(Distributed Hash Table) 방식의 P2P lookup 기법이다. Chord에서는 routed queries 방식을 사용하는데, 이는 노드가 라우터처럼 각자가 지닌 부분적인 찾기 정보를(위치와 해당 정보) 이용하고 해당 데이터가 다른 곳에 있는 경우, 찾기 요청을 다른 노드에게 전송한다. 찾기 요청을 받은 다른 노드는 다시 로칼 노드에서 요청받은 자원이 있는가를 확인하고, 해당 자료가 없는 경우는 또 다른 노드에게 찾기 요청을 전송한다. Chord는 찾기 때마다 평균 O(log(n))의 메시지를 사용하고 노드마다 O(log(n))의 정보를 관리하게 된다. 또한 네트워크가 분할되는 경우에도 재빠르게 분할된 부분에 위치한 노드들의 finger table을 수정하여 전체 찾기에 차질이 없도록 하고 있다.

Chord에서는 다음 그림 2와 같이 node id와 key id가 동일한 ID 공간에 오도록 키와 node id를 매핑 한다.

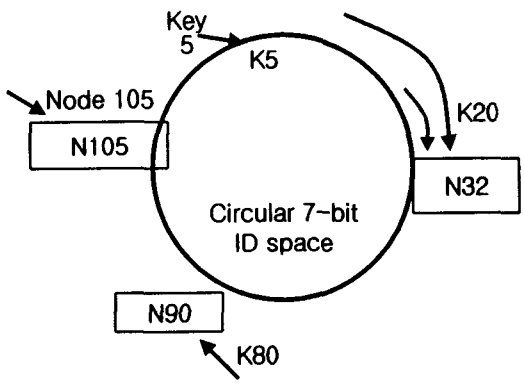

그림 2. Chord ID 공간

그림 2에서 key와 node는 균일하게 분포되어 있으며, 해당하는 key의 node가 존재하지 않을 경우, key는 다음 node에 저장된다. 그림 2의 key값 5와 20은(k5, k20) 동일한 node인 N32에 저장된다. 각 노드마다 최대 log(n)만큼의 routing 정보를(finger table) 가지고 있으며 해당정보를 이용하여 키를 찾을 때까지 검색하게 된다.

만일 이러한 routing 정보를 가지고 있지 않으면 단순 검색이 되고, 그 경우에는 최대 n(참여하고 있는 node 수)만큼 검색을 하게 된다. 다음의 그림 3과 같이 단순 검색을 하는 경우에는 원하는 키를 지닌 노드를 검색할 때까지 계속하여 순차적으로 질의를 보내고 응답을 기다리는 방식이 된다.

Chord의 finger table에는 최대 log(n)만큼의 routing 정보를 저장하고 있으며 해당 i 번째 i인덱스는 해당 노드에서 2i 만큼 위치에 있는 노드를 나타내고 있다. 예를 들어 현재 참여하고 있는 노드 수가 128개라 가정하면, node80(그림 4에서 N80)은 finger table에 6개의 인덱스를 지니게 되고, 각 인덱스는 2i 만큼의 거리를 가지게 된다. 그림 3에서 각 노드들은 N80에서 시작하여 전체 id 공간상에 1/2, 1/4,..,만큼 떨어져 있는 노드들로 구성된다. 각 노드마다 지니고 있는 finger table의 정보는 그림 4.와 같이 index, interval, 그리고 해당 interval에 참여하고 있는 노드를 나타낸 successor를 지니고 있다.

finger table에 있는 정보를 이용하여 검색을 하는 경우에는 해당키가 있는 노드를 찾기 위하여 먼저 검색을 요구한 node의 finger table을 살펴서 해당키 값이 어디에(interval) 있는지 확인한 후, 해당 구간의 successor를 파악하여 해당 노드에 키 검색을 요청한다. 동일한 방식으로 검색을 요청하면 최대 log(n)만큼의 검색 요청이 발생되고 해당 키를 찾게 된다.

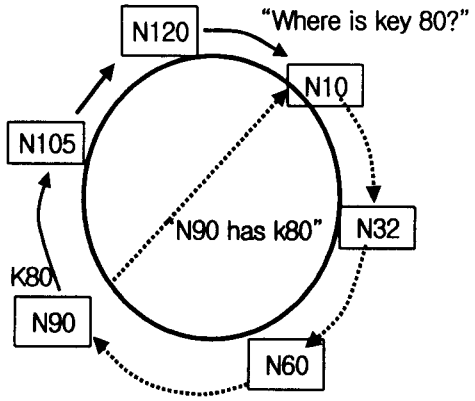

그림 3. 순차적 검색에 대한 예

그림 5는 새로운 검색 방식을 나타낸다. 새로운 노드가 join하는 경우는 앞에 있는 노드의 successor를 수정하고 해당 노드에서 보관하여야 할 키를 옮기는 (successor나 predecessor에서) 작업을 수행한다. 또한, 일부 노드에서 해당 node를 포함하는 interval과 successor 정보를 변경한다. 어느 경우에나 소요되는 시간이 O(log(n))이므로 Chord는 순차적인 검색에 비하여 효율성이 높게 나다난다.

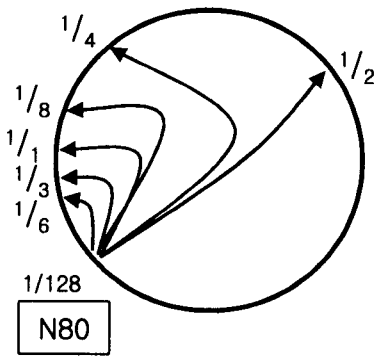

그림 4. Chord의 Modular 2 연산에 의한 이진 검색

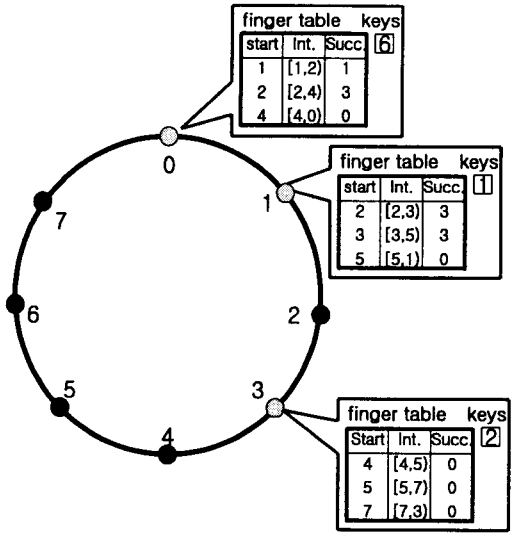

그림 5. Chord의 finger table

3. 다단계 해싱을 통한 lookup

비구조적 P2P에서의 Lookup 과정은 사용자의 질의 수행과 네트워크의 일관성 유지를 위한 접속 상태 확인을 위한 점검에서 매우 자주 사용되기 때문에 chord 시스템과 같이 해싱 방법을 통하여 검색의 효율성을 높이는 것은 매우 바람직하다. 그러나 chord의 경우 다음과 같은 문제점을 지니고 있다.

3.1 Chord 해싱의 문제점

관련연구에서 비교하였듯이 비구조적 P2P 환경에서 chord는 다른 시스템에 비하여 매우 효율적인 lookup 검색 기능을 가지고 있으나 아래와 같은 두 가지 문제점을 가지고 있다.

첫째, Chord가 사용하는 IP에 대한 SHA-1 해싱은 네트워크 IP가 불규칙하게 분포되어 있는 경우 실제적인 라우팅 정보를 충분히 반영하지 못한다.

둘째, 빈번하게 발생하는 갱신에 대하여 순차적인 finger table의 갱신은 매우 비효율적이어서 대량의 정보를 공유하는 P2P 공유 시스템에서는 실제 응용이 어렵다.

첫 번째 문제점은 매우 명백하다. 인터넷은 미국에서 시작되었고 IP주소 체제도 미국을 중심으로 하여 각국에 배정되어 있는 상황이다. 한국인터넷정보센터의 자료에 의하면 현재 (2003년 8월 25일 기준) 국내에는 77개의 ISP(Internet Service Provider)가 등록되어 있다.[9] 또한 국내의 ISP 중 APNIC으로부터 A class 급 IP 주소를 배정 받은 기관은 존재하지 않으며, B또는 C class급으로 배정받은 IP 주소의 경우에도 동일 ISP임에도 불구하고 매우 불규칙한 주소 공간을 사용하고 있는 실정이다. 특히 IPv4 주소 공간의 고갈에 따라 A class급 주소의 일부를 B 또는 C class급 주소로 재배정하게 되었고 이에 따라 IPv4의 주소공간은 기존의 주소 공간 설계와는 서로 상이한 구성을 가지고 있다. 따라서 단순한 함수의 조합으로 산출된 결과를 이용하여 IP 주소에 따르는 라우팅 정보를 검출한다는 것은 매우 부적절한 방법이라 할 수 있다. SHA-1(Secure Hash Algorithm-1)은 264 bit이하의 메시지를 받아들여 160bit(20byte)의 요약 코드를 산출하는 해싱 방법이다.[6] 국내와 마찬가지로 미국을 제외한 대부분의 나라에서도 IP 주소에 대한 SHA-1 요약으로 산출된 결과는 실제적인 라우팅 정보를 충분하게 반영하지 못한다는 단점을 지닌다.

Chord의 두 번째 문제점은 lookup을 좀더 빠른 방법으로 찾기 위하여 운영되는 finger table의 유지 관리에서 나타난다. 대부분의 P2P 공유에서는 1만개에 가까운 노드들이 한 네트워크를 구성하게 되고 각 노드가 공유하는 콘텐츠의 양은 그의 몇 배를 지니게 된다. 따라서 각 노드들이 유지하여야 하는 finger table의 크기는 매우 방대하며, 빈번하게 발생하는 가입과 탈퇴 갱신 동작은 각 노드가 보유한 finger table의 갱신을 순차적으로 유발하게 된다. 특히 가입자가 늘어나면서 발생하는 네트워크 분리 동작에서는 각 노드가 보유하고 있는 finger table 전체를 갱신하는 문제점을 가지고 있다.

3.2ISPIP분배를고려한다단계해싱방법

(1) IP 배정 현황과 ISP 그룹 코드

한국인터넷정보센터 자료에 의하면 현재 (2003년 8월 25일 기준) 국내에는 77개의 ISP(Internet Service Provider)가 등록되어 있다. 그중 6개 기관은 2048(/24) 보다 많은 IP를 보유 하고 있으며, 14개 이상의 기관이 256(/24)의 IP를 보유한 것으로 나타나 있다. 그 외의 ISP는 32(/24) 이하의 비교적 소규모의 ISP들로 구성되어 있다. 본 연구에서는 이들 ISP를 16개의 기본 분류로 나누고 이를 다시 16개의 세부 분류로 배정하였다. 따라서 각 분류마다 4bit로 구성된 분류 코드를 부여하고, 다음 절에서 언급할 확장 해싱 키로 이용한다. 각 분류에 속하는 ISP에 대한 세부 명칭은 한국인터넷정보센터의 회원 ISP 현황표를 참조하길 바란다.[9]

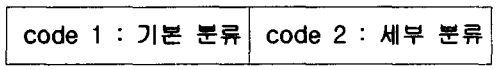

그림 6. ISP 코드의 구조

그림 6의 분류 코드는 P2P 네트워크에 속한 모든 노드들이 자신이 속한 네트워크의 정보를 알아내기 위하여 ISP별 IP 배정표를 통하여 추출한다. 따라서 동일한 ISP에 속한 노드들은 동일한 ISP 분류 코드를 얻게 된다. 이렇게 추출된 ISP에 대한 분류 코드는 확장 해싱를 통하여 콘텐츠 id에 대한 SHA-1 요약 코드에 대한 검색 공간을 선택하는데 활용된다. 따라서 동일한 ISP에 속한 콘텐츠에 대해서만 검색하고자 하는 경우, 위의 분류 코드에 따른 검색 공간의 축소를 얻어낼 수 있다.

(2) 확장 해싱 방법을 이용한 다단계 해싱

콘텐츠에 대한 id와 IP 주소에 의한 단순한 SHA-1 요약에 따른 해싱은 위에서 밝힌 바와 같이 IP 주소 자체가 불규칙하게 분포된 국내의 경우에는 라우팅 정보를 충분하게 반영하지 못한다. 따라서 본 논문에서는 아래의 그림 7과 같은 2단계의 해싱 구조를 제안한다.

그림 7은 확장 해싱과 콘텐츠 id에 대한 SHA-1 요약 값으로 구성된 2단계 해싱 구조를 나타낸다. 그림 7에서 정보 검색을 수행하고자 하는

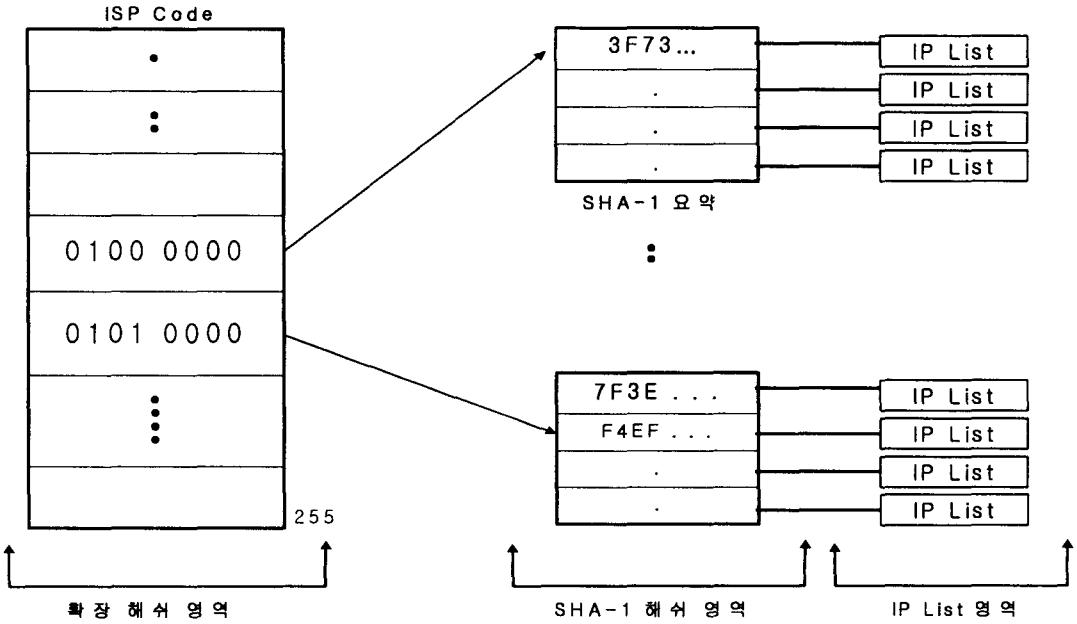

그림 7. 다단계 해싱 구조

노드는 자신이 속한 네트워크를 확인하기 위하여 해당 노드의 IP를 이용하여 그림 5와 같은 ISP 분류 코드 중 1개를 추출한다. 확장 해싱을 통하여 얻은 포인터 값을 이용하여 콘텐츠 id에 대한 검색 공간을 선택하게 된다. 그림 6의 확장 해싱에 대한 매핑은, 확장 키 값 크기의 bit 연산을 통해 이루어진다. 따라서 전체 네트워크의 크기가 작은 경우에는 초기치가 "1" 이었을 때 좌측 bit를 기준하여 매핑이 이루어진다. 네트워크의 크기가 늘어나는 경우에는 확장 키 값 key를 증가시켜 매핑에 반영되는 bit 수를 증가 시킨다. 네트워크 확장에 따른 네트워크 분할과 합병은 4장에서 다루고자 한다. 그림 7에서 ISP 분류 코드 부분은 그림 6에 의한 본류 기준에 의거하여 인코딩 된 8bit 확장 해싱 키이다. 각 ISP에 대한 인코딩은 해당 ISP의 보유 IP를 기준으로 하여 한국인터넷정보센터의 자료를 참고하여 부여하였다. ISP가 보유한 IP의 변화량은 비교적 적은 편이며, 실제 구현 시에는 프로그램에 대한 update patch에서 충분히 갱신 가능하다.

그림 7의 다단계 해싱 자료구조를 운영하기 위한 삽입, 갱신, 삭제에 대한 알고리즘은 그림 8과 같다. 그림 8은 어떤 노드 n에서 특정 공유 노드 IP가 보유한 콘텐츠 m에 대한 정보로, 노드 n에 대한 확장 해싱 값은 프로그램의 시작과 동시에 해당 호스트의 IP 주소를 획득하여 산출한다. 알고리즘에서 n'은 노드 n에 대한 확장키이고, s는 콘텐츠 m에 대한 SHA-1 요약 값이며, k는 확장 해싱을 반영하기 위한 bit값으로 네트워크 분할 합병 알고리즘에 의하여 정해진다.

```
n.insert()
    SHA.entry = SHA.table(n',k);
    if s ∈ SHA.entery thenRETURN;
    make SHA.entry(s);
    SHA.entry(s).code = s;
    SHA.entry(s).ip_poiter = ip;
    SHA.entry(s).counter =1;

n.update()
    if s ∈ SHA.entry then
        increase SHA.entry.counter;
        SHA.entry(s).ip_pointer(SHA.entry(s).counter) = ip;
    else n.insert();

n.delete()
    find ip from ip_pointer();
    delete ip form ip_pointer();
    decrease SHA.entry(s).counter;
    if SHA.entry(s).counter = 0 then
        delete SHA.entry(s);
```

그림 8. 콘텐츠 m에 대한 삽입, 갱신, 삭제 알고리즘

알고리즘 비교 : 다단계 해싱 방법과 Chord 두 가지 모두 콘텐츠에 대한 검색은 그림 3과 같이 modular 2 연산을 이용한 이진검색을 바탕으로 실행된다. 따라서 두 알고리즘이 지니는 복잡도는 O(log n)에 비례하여 동작한다. 단, chord의 경우에는 메시지 id와 ip 주소가 동일한 SHA-1 요약 코드 공간에 존재하기 때문에 동일한 공간에서의 이진 검색을 수행한 결과로 검출할 수 있다. 다단계 해싱 방식에서는 IP 주소와 콘텐츠 id가 이원화된 구조이고 IP 주소에 대한 검색은 순차적이다. 알고리즘의 복잡도는 동일하게 O(log n)이나 다단계 해싱 방법에서는 노드의 추가와 삭제 시, 해당되는 ip_pinter()의 값만 변경하지만 chord에서는 finger-table을 순차적으로 갱신하는 문제점을 가지고 있다. 다단계 해싱 방법에서 테이블을 갱신하는 경우는 동일한 값을 가지는 SHA.entry(s).counter 값이 "0"인 즉, 해당 콘텐츠를 공유한 ip가 네트워크에 없을 때만 발생하기 때문에 테이블 갱신에 따르는 overhead가 감소된다.

4. 네트워크 분할과 합병 및 실험

4.1 네트워크의 분할과 합병

P2P lookup의 성능은 다음에서 언급하는 3가지 척도로 측정한다.

- 상대 지연률(RDP: Relative Delay Penalty) : 두 노드 사이에서의 상대적인 지연률로 각

IP의 지연 거리로 측정.

- 물리 네트워크의 Hop 수 : 실제 두 노드 사이의 Hop의 수
- 실패 lookup의 수 : lookup 정보 획득에 실패 횟수

위의 3가지 척도를 기준으로 평가하였을 때, 성능 향상을 위해서는 네트워크에 포함되는 노드의 개수를 일정한 수준으로 유지하는 것이 가장 중요하다. Gopal P.등은 P2P 공유에서 적정 수준의 노드를 제한하는 것에 대한 이론적 수치를 제안하였다.[11] Gopal P. 등은 어떤 네트워크에 진입하는 노드의 수가 포아슨 분포로 증가하였을 때 방출되는 노드의 수와 비례한 전체 네트워크의 상대적 크기를 수치적으로 증명하였다. 그러나 Saroiu S. 등의 연구에 의해 Napster나 Gnutella와 같은 일반적인 P2P의 노드는 일정한 분포를 따르지 않는다는 사실이 실제 측정으로 알려졌으므로 노드 증가수를 포아슨 분포로 한정하였을 때의 이론적 추론은 무의미 하다. Saroiu 등의 연구에 의하면 gnutella 네트워크에서 20% 이상의 노드들은 680ms 이상의 느린 응답 시간을 가지는 것으로 측정되었다. 따라서 네트워크가 일정 수준 이상으로 커진 경우 해당 네트워크의 분리는 필요한 조치다. 그림 9는 Gnutella 네트워크에 포함된 노드 수에 따라 증가되는 네트워크 hop 수를 나타낸다. 그림 9에서 알 수 있듯이 10,000개 이상의 노드가 중첩된 경우에는 hop 수가 10 이상의 높은 값을 가지게 된다. 노드의 개수가 4000개 이하 일 때 까지는 일정한 기울기를 가지며 네트워크 지연도가 높아지고 7000개상의 경우와 10000개 이하의 경우에는 hop 수의 편차가 적은 것으로 나타났다. 네트워크 성능에 대한 또 다른 연구로는 Jung 등의 연구가 있다. Jung 등은 실질적인 DNS lookup에서의 TCP/IP 환경에서의 hop 수 변화에 대한 연구를 진행하였다. Jung 등의 분석에 의하면 lookup의 효율적인 hop 수는 8 이하의 hop인 경우에 가장 효율적인 것으로 분석되었다.[12]

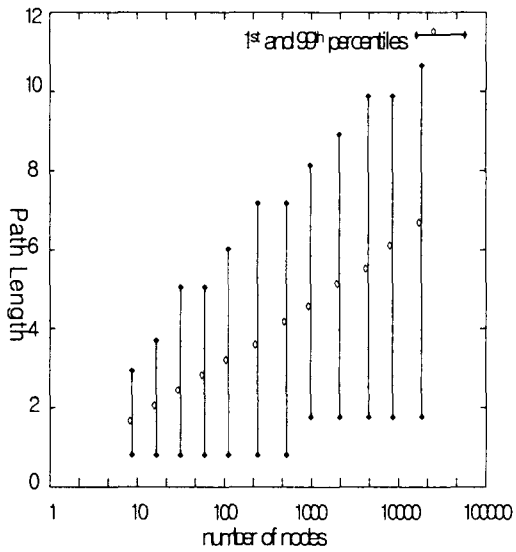

그림 9. 노드 수에 따른 네트워크 지연도

본 논문에서 실험 대상인 네트워크는 동일 ISP의 범위에서 lookup이 진행되기 때문에 실질적인 lookup의 hop 수는 Gnutella의 경우에 비하여 양호하다. 따라서 네트워크의 분리가 이루어지는 기준을 해당 네트워크의 노드수가 10,000개 이상이었을 때를 기준으로 하여 2개의 네트워크로 분류하는 방식을 이용하여 구현한다.

그림 10은 ISP 분류 코드에 의하여 네트워크가 분류되는 것에 대한 도시이다. 그림 10에서 초기 네트워크는 ISP 코드 중 1 bit를 기준으로 하여 두 개의 네트워크로 분리된 상태로 출발하여 네트워크에 참여하는 노드의 수가 증가될수록 확장 해싱에 사용되는 키 bit를 증가하는 방식으로 역 이진 트리 형태의 증가 감소를 반복하게 된다.

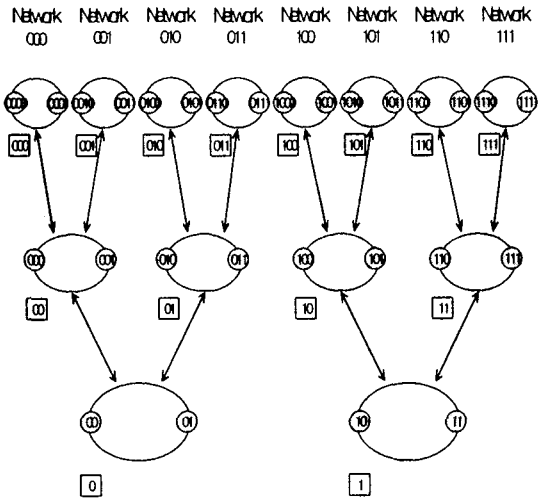

그림 10. 확장키를 이용한 네트워크 분할과 합병

```
// k : extensible hash key //
if k ≠ 1 then
    for k = k downto 1
        //reference another SHA table for search//
        lookup(m) within upper size network (k-1);
        get IP_list that match with extended SHA.table;
else return(fail);
```

그림 11. 가상으로 확대된 네트워크에서의 SHA.table 검색을 위한 알고리즘

4.2실패lookup에 대한처리

제안된 다단계 해싱에 의한 lookup 방식에서는 성공적으로 lookup을 찾는 경우와 네트워크를 분할 합병하는 부분에 대해서는 매우 효율적이나, 아래와 같은 문제점을 가지고 있다.

- 해당 네트워크 내에서의 lookup 검색에 실패한 경우.
- IP_list에 포함된 ip 중 lookup 검색 request에 응답하지 않는 노드

따라서 이러한 문제점을 해결하기 위하여 다음과 같은 알고리즘을 제시한다.

(1) Lookup 실패 처리

현재의 네트워크에서 lookup 검색이 실패한 경우, 다음의 알고리즘을 통하여 현재의 네트워크 보다 확대된 네트워크에서의 검색을 수행한다. 만약 현재의 네트워크가 초기 분할 이전의 상태이거나 상위 네트워크를 검색한 이후에도 획득된 IP_list가 없는 경우에는 lookup 검색에 실패한 것으로 간주한다. 아래의 알고리즘에서 상위 네트워크에 대한 SHA.table의 정보는 다음에 소개되는 successor-list의 순차적인 전이에 의하여 갱신된다.

(2) 무응답 IP의 처리

Sariou 등의 또 다른 측정에 의하면 Gnutella 네트워크의 60%에 해당되는 노드들이 네트워크에 접속할 수 없는(shutdown) 노드들인 것으로 측정되었다.[10] 이들 노드들에 대한 효율적인 관리가 진행되지 않는 경우에는 검색 overhead를 초래하게 된다. 따라서 주기적인 검색을 통하여 도달 불가능 IP들에 대한 축출 작업을 지속적으로 수행하여야 한다. lookup 검색을 통하여 응답이 있는 노드와 일정시간 이후 응답이 없는 노드들이 판별되었을 때, 무응답 노드들은 테이블에서 제거되어야 한다. 본 논문에서는 이들 노드들에 대한 정보를 IP_list로부터 제거하도록 설계하였다. 또한 갱신된 자료를 저장하고 있는 새로운 SHA.table을 lookup 질의에 응답해 준 노드들에게 전달하는 방식을 통하여 기존의 IP_list를 갱신한다. 위와 같은 방법은 순차적인 갱신을 유발하기 때문에 수렴 속도가 느려질 수 있는 단점을 가지고 있으나 P2P 파일 공유 시스템에서의 lookup 검색은 매우 빈번하게 발생되기 때문에 검색 이후의 갱신을 통하여 모든 활성화된 노드들에 대한 점진적인 갱신이 가능하다.

4.3시스템시뮬레이터

P2P 공유 프로그램은 다수의 클라이언트를 확

보하여야 하기 때문에 별도의 시스템 구현을 통한 실험이 매우 곤란하다. 따라서 별도의 시뮬레이션 환경을 설정한 후, 이를 바탕으로 수치 해석적 접근 방법이 Gopal 등[11]을 통하여 시도되었으나 Saroiu 등[10]의 실측에 의한 실험 결과, 매우 불규칙한 형태로의 노드 접속과 네트워크 유지가 이루어지기 때문에 수식에 의한 분석 방법은 부적절한 접근 방법으로 판단된다. 따라서 본 논문에서는 현재 실행되는 P2P 공유 시스템과의 연계를 통한 실제 시뮬레이션을 수행하는 방법을 이용한다.

현재 다수의 사용자 층을 확보하고 있으면서 소스 프로그램이 공개된 P2P 공유 프로그램으로는 emule 프로그램이 있다.[13] emule은 국제적으로 널리 사용되는 edonkey 공유 서비스의 다른 형태이고 edonkey가 사용하고 있는 네트워크와 공유 자원을 공유한다. 따라서 edonkey와 동일한 크기의 P2P 네트워크를 구성하고 있는 서비스이다. emule의 경우에는 각 네트워크마다 별도의 edonkey 서버를 보유하고 있고 서버가 접속된 클라이언트들에 대한 정보를 유지 관리하는 기법을 이용하는 분산 IP 서버 형태의 P2P 공유 시스템이다.

본 논문에서는 emule의 특정 서버에 접속하여 클라이언트에 대한 정보를 획득하고 이에 대하여 별도의 다단계 해싱 테이블을 구성한 후, 해당 해싱 테이블에 대한 DHT lookup 검색을 수행 하는 형태로 실험을 진행하였다. 위를 위한 시뮬레이션 구성도는 그림 12와 같다.

그림 12의 시뮬레이션 환경에 본 논문에서 제안된 알고리즘을 반영하기 위하여 환경 변환 조건을 표 1에 제시하였다. 표 1의 환경 설정에 따른 prototype 프로그램의 실행 예는 그림 13과 같이 나타난다. 네트워크 분할과 합병은 시뮬레이션을

수행하는 클라이언트에서만 이루어진다. 본 논문은 P2P lookup에 대한 검색 질의 효율성 향상에 관한 것으로 이 부분에 대한 실험은 추후 연구 과제로 남기고자 한다. 검색 질의를 전달하는 방법은 emule의 프로토콜을 이용하여 전달하는 방식을 이용하였기 때문에 검색 질의 처리에 대한 편차는 존재하지 않았다.

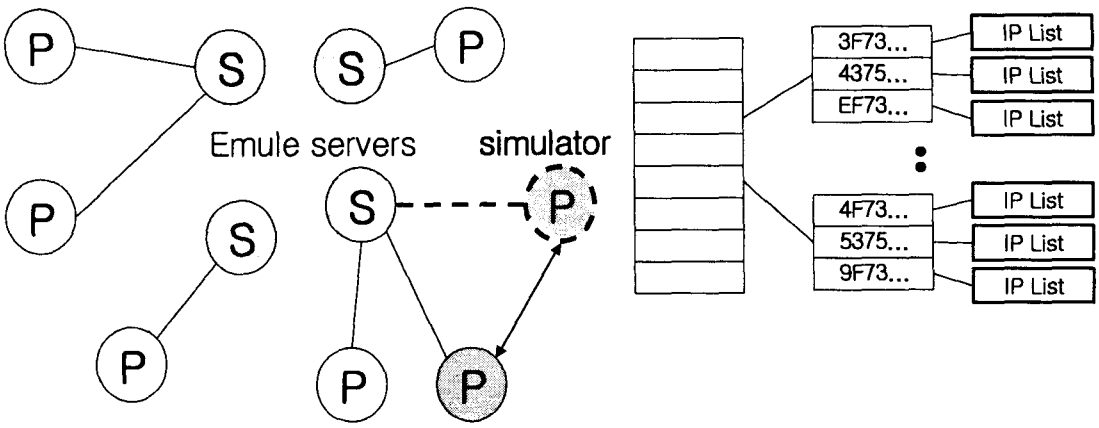

그림 12. emule 서버와의 연동을 통한 시뮬레이션 환경

표 1. Emule 네트워크와의 시뮬레이션을 위한 설정

Condition	Emule client	Simulated Servent
Server 접속	IP 서버에 클라이언트 등록	초기 테이블 구성을 위하여 조회
Lookup	검색시 서버로부터 IP 획득이후 가용 IP에 대하여 flooding	위에서 구축된 테이블을 통한 hash
네트워크 분할	다른 서버로의 접속 이전엔 수행되지 않는다	2000개 이상의 노드가 축적되면 2진 분류
ISP별 분류	해당 서버에 접속된 모든 노드	확장 해싱를 통한 분류
lookup 실패	IP lookup 실패는 존재하지 않는다	상위 네트워크로의 질의 확장 수행
무응답 IP 처리	일정 시간 경과 이후 서버로의 응답이 없는 IP를 정리	servent 자체에는 반영가능, 타 클라이언트로의 전이 불가

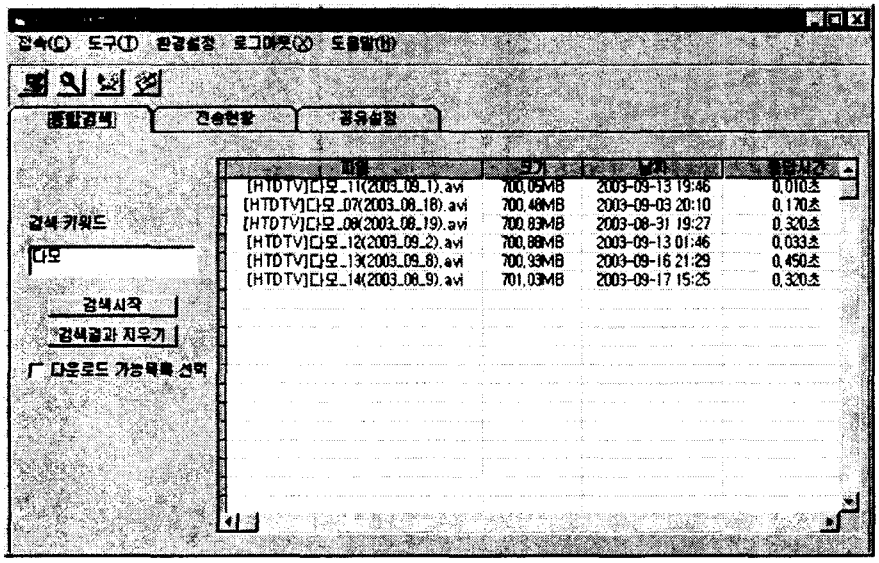

그림 13. emule 네트워크를 통한 시뮬레이션 결과

5.결론

본 논문에서는 MIT의 chord project를 토대로 설계된 DHT 방식의 P2P 시스템에 대한 문제점을 파악하고 이를 개선하기 위한 방안과 알고리즘을 제안하였다. 제안된 방법에서는 국내의 ISP가 보유하고 있는 라우팅 정보를 이용하기 위하여 각

ISP들이 보유하고 있는 IP 주소공간을 몇 개의 ISP 그룹으로 별도로 분류하여, 이에 대한 분류코드를 배정하였다. 또한 IP 주소 변동 시에도 콘텐츠의 동일성을 예측할 수 있는 콘텐츠 id에 대한 SHA-1 요약 값을 조합하여, lookup 전이의 요소로 사용한다. ISP 분류 코드에 대하여 분류된 소규모 네트워크에서, 제안한 lookup 검색은 1차적으로 콘텐츠 id에 대한 SHA-1 해싱을 통하여 노드를 추출한다. 제안한 방법은 기존의 DHT 방식이 이용한 lookup 검색에 비하여 작은 범위 내에서 검색을 수행하기 때문에 검색 효율이 높으며, ISP 그룹별 검색을 수행하므로 lookup에 소요되는 지연도도 낮다. 또한, 축소된 공간상에서 lookup 검색 실패 시, 좀더 규모가 큰 상위 네트워크로 검색 공간을 확장하여 원하는 정보를 검색하도록 알고리즘을 설계하였다.

본 논문에서는 현재 많은 사용자 층을 확보하고 있으며 소스 코드가 공개된 emule P2P 소프트웨어를 기반으로 시뮬레이션 환경을 제시하고 제안된 알고리즘을 적용하여 실험하였다.

참고문헌

[1] Frank Dabek, Emma Brunskill, M. Frans Kaashoek, David karger, Robert Morris, Ion Stoica, Hari Balakrisnan, "Building peer-to-peer systems with Chord, a distributed location service", In Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems (HotOS-VIII) May 2001, pp. 7176.

[2] Jaeyeon Jung, Emil Sit, Hari Balakrishnan, and Robert Morris "DNS Performance and the Effectiveness of Caching",IEEE/ACM Trans. on Networking, October 2002.

[3] Dmitri Loguinov, Anuj Kumar, Vivek Rai, Sai Ganesh, "Graph-Theoretic Analysis of Structured Peer-to-Peer System: Routing Distances and Fault Resilience", In Proceedings of ACM SIGCOMM 2003, PP. 395-406, August 2003.

[4] Nicholas J. A. Harvey, Michaek B. Jones, Stefan Saroiu, Marvin Theimer, Alec Wolman, "SkipNet: A Scalable Overlay Network with Practical Locality Properties", Proceedings of USITS '03, March 2003.

[5] Karger D, Leman E, Leighton F, Levine M, Lewin D, and AND Panigrahy R. "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web", In Proceedings of the 29th Annual ACM Symposium on Theory of Computing (El Paso, TX, May 1997), pp. 654~663.

[6] FIPS 181-1. Secure Hash Standard. U.S. Department of Commerce/NIST, National Technicla Information Service, Springfield, VA April 1995.

[7] Gnutella. http://gnutella.wego.com/.

[8] Napster. http://www.napster.com/.

[9] 한국인터넷정보센터 : http://ip.nic.or.kr/

[10]Saroiu S., Gummasi P. K., and Gribblr S. D. A "measurement Study of Peer-to-Peer File Sharing Systems", In Proceedings of Multimedia Computing and Networking 2002(MMCN '02) January 2002.

[11]Gopal Pandurangan, Prabhakar Raghavan, Eli Upfal, "Building Low-Diameter P2P Networks"

[12]Jaeyein Jung, Emil Sit, Hari Balakrishnan and Robert Morris, " DNS performance and the Effectiveness of Caching",

[13]Emule Project : http://www.emule-project.net

김 홍 일

1986년 홍익대학교
전자계산학과 (이학사)
1989년 인하대학교
전자계산학과 (이학석사)
2000년 홍익대학교
전자계산학과 (이학박사)
1994년 ~ 현재 : 대진대학교
컴퓨터공학과 조교수
관심분야 : IPv6, P2P, 인터넷응용

신 판 섭

1992년 홍익대학교
전자계산학과 (이학사)
1994년 홍익대학교
전자계산학과 (이학석사)
2000년 홍익대학교
전자계산학과 (이학박사)
2002년 ~ 현재 : 대진대학교
컴퓨터공학과 전임강사
관심분야 : 분산 · 객체 데이터베이스,
기억데이터베이스, XML 응용 시스템,
멀티미디어 시스템,.. etc.