

계층적 그룹관리와 신뢰성을 위한 동적인 변형 P2P 시스템 설계 및 구현

이 석 희* · 조 상** · 김 성 열**

요 약

현재 많이 사용되고 있는 P2P 개념으로는 순수 P2P와 변형 P2P 구조가 있다. 순수 P2P의 모델에는 Gnutella와 Ktella 등의 형태가 존재하고 변형 P2P로는 무수히 많은 형태가 존재한다. 순수 P2P 모델의 경우에는 정보 공유에서 연결성을 장점으로 Gnutella의 형태를 응용한 형태로 많이 사용되고 있지만 정보를 검색하거나 제공하기 위해 많은 트래픽을 소모하게 된다. 이와는 달리 변형 P2P 모델들 중 정보 공유 모델들이 존재하는데 이 모델들은 사용자에게 효율적이고 빠른 검색과 색인을 제공하기 위해 기존의 서버/클라이언트 형태를 취하고 있지만 확장성에 있어서 제공하는 서버의 능력에 의존할 수 밖에 없다. 파일공유 모델의 Peer들에 대해 연결성 유지를 위한 많은 부하와 사용자에 있어서 그룹에 대한 형태의 문제점 그리고 서버의 Fail로 인한 비 연결성에 대한 문제점을 해결하기 위해 본 논문에서는 라우팅 프로토콜 기법에서의 접근과 계층적 구조를 적용하고 Backup 시스템을 포함해서 대규모 네트워크에서 효율적인 그룹관리와 동적인 서버의 지정으로 신뢰성을 유지하기 위한 시스템을 설계 구현하여 그 우수성을 검증하였다.

A Design and Implementation of Dynamic Hybrid P2P System with Hierarchical Group Management and Maintenance of Reliability

Seok-Hee Lee* · Sang Cho** · Seong-Ryeol Kim**

ABSTRACT

In current P2P concept, pure P2P and Hybrid P2P structures are used commonly. Gnutella and Ktella are forms of pure P2P, and forms of Hybrid P2P are innumerable. File searching models exist in these models. These models provide group management for file sharing, searching and indexing. The general file sharing model is good at maintaining connectivity. However, it is defective in group management. Therefore, this study approaches hierarchical structure in file sharing models through routing technique and backup system. This system was designed so that the user was able to maintain group efficiency and connection reliability in large-scale network.

키워드 : 변형 P2P(Hybrid P2P), 분산 시스템(Distributed System), 계층 구조(Hierarchical Structure), 백업(Backup)

1. 서 론

초기 인터넷은 오늘날의 네트워크보다 더 개방적이고 자유로웠다. 하지만 1980년대를 접어들어 네트워크에 부정하게 접근하는 것을 막는 방화벽이 등장하고 사용자들은 서버/클라이언트 관계에서 사용되는 FTP와 텔넷과 같은 서비스를 사용하지 않게 되었다. 1994년에 인터넷이 폭발적으로 성장하면서 인터넷의 모습은 거대한 대중매체로 급속히 변화하였다. 이 과정에서 사람들은 웹 서버를 사용하며 자신의 자원들을 사용자로 하여금 접근할 수 있게 하고 ADSL이나 케이블 모뎀 같은 비대칭 네트워크 연결의 성장을 통해 자연히 서버/클라이언트로서의 모델을 자리잡았으며 각종 어

플리케이션들이 이와 같은 비대칭적인 모델을 갖게 되었다[1].

P2P모델은 1970년대부터 존재하였지만 1990년대 후반에 들어와 PC는 속도와 처리 능력 면에서 점점 향상되고 소프트웨어 개발자들은 PC에 서버 소프트웨어를 개발할 수 있게 되었고 정보를 직접 공유할 수 있는 P2P시스템이 대두되었다.

P2P에는 순수 P2P와 변형 P2P형태가 존재하는데 순수 P2P는 분산형태를 취하고 대표적인 것이 Gnutella와 국내 Ktella가 있다. Gnutella는 분산형태로 순수 P2P인데 Gnutella는 연결성에 있어서 장점을 지니고 있지만 사용자가 원하는 정보를 찾기 위한 방법으로 논리적인 라우팅을 사용하고 Broadcast를 사용하기 때문에 수많은 트래픽을 발생시킨다 [2]. 이러한 형태의 유지와 변형으로 많은 공유 프로그램들이 사용되고 있지만 아직도 필요 없는 트래픽에 대한 문제

* 준 회 원 : 청주대학교 대학원 컴퓨터정보공학과
** 정 회 원 : 청주대학교 컴퓨터정보공학과 교수
논문접수 : 2003년 7월 14일, 심사완료 : 2004년 4월 28일

점이 존재한다.

변형 P2P로는 음악정보를 공유하기 위한 Napster와 외계 신호를 분석하는데 사용되고 있고 전산학과 산업, 과학분야에 응용이 되고 있는 SETI@Home 프로젝트를 들 수 있다. 이러한 중앙 집중형의 변형 형태는 서버에서 특정한 서비스와 검색, 색인 등의 기능을 제공할 때 주로 사용하는데 이 형태는 확장성에서도 서버의 능력에 의존한다는 단점을 지니고 있다.

본 논문에서 제시한 변형 P2P의 형태는 계층형태와 중앙 집중형태를 혼합하였고 이로써 대규모 네트워크에서 Peer들에 대해 효율적인 관리가 가능하고 계층 형태의 그룹을 형성하여 그룹의 서버 역할을 하는 노드를 지정하는 기능과 그룹 서버의 물리적, 논리적 Fail에 대비해 Backup-노드를 지닌 신뢰성을 제공하는 시스템을 구현하였다.

그룹관리와 연결성의 신뢰를 보장하기 위한 시스템의 구체적인 사항에 있어서 많은 부분이 동적 라우팅 프로토콜에서의 형태와 구조에서 근거를 했으며, 갱신정보의 유지와 다중 그룹 지정기능과 그룹관리를 위한 방법도 변형 P2P 구조인 파일공유 모델의 관점에서 볼 때 라우팅 프로토콜에서의 필요한 점을 재구성하여 모델의 특성과 대규모 네트워크의 연결성과 관리적인 측면에 중점을 두어 설계하고 그 우수성을 검증하였다.

2. 관련 연구

2.1 P2P의 기본적인 형태별 특성

중앙 집중형은 서버가 처리해야 할 일들을 각 클라이언트에 나누어주는 기술을 가진 SETI@Home에서 사용하고 있고 Napster의 검색 기술에서 볼 수 있다. 관리하는데 편하고 정보의 일관성이나 혹은 결합력에 있어서 뛰어나다. 하지만 내구성에 있어서 서버가 다운되면 아무것도 할 수 없고, 확장성에서는 중앙 노드의 한계가 있다.

원형은 일반적으로 분산 서버들의 집단에 대한 형태로서 동일한 기능을 가진 집단에 적절하다. 트래픽의 집중화와 서버가 연결성을 잃었을 때에 대한 대책으로 효과적으로 사용된다. 관리하기 용이하고 보안에도 강하다고 볼 수 있다. 내구성 또한 강해서 중앙 집중형과도 함께 혼합형으로 이용된다.

계층형태의 가장 커다란 장점은 확장 가능성이다. 가장 대표적인 예로 DNS(Domain Name System)와 NTP(Network Time Protocol), Usenet을 들 수 있다. 계층이 시작되는 루트와 하위 계층을 잇는 서버의 Fail에 대해서는 여전히 서버/클라이언트 모델과 같은 문제점을 갖고 있다.

분산형은 Gnutella 프로토콜에서 사용하며 가장 순수한

P2P프로토콜이라고 볼 수 있다. 대표적인 예로 Freenet과 Ocean Store등이 있다. 라우팅 기술로는 BGP(Border Gateway Protocol)에서 서로 다른 AS(Autonomous System)들간의 라우팅에서 사용된다[4,5]. 이러한 형태는 확장 가능성과 연결성에 있어서 탁월한 장점을 갖지만 관리적인 측면에서와 보안적인 측면 그리고 일관성에서는 중계자 역할을 하는 서버가 존재하지 않기 때문에 나타나는 문제점들이 많다.

2.2 Gnutella 분석

Gnutella방식의 순수 분산형 P2P의 장점은 연결성에 있다. 또한 Gnutella에서 가장 주목되는 특징으로 볼 수 있는 것은 논리적인 라우팅을 사용한다는 것이다. 회선중심의 물리적인 네트워크 시스템들의 라우팅은 호스트와 호스트가 연결되어 있는 물리적인 연결성을 바탕으로 데이터를 라우팅하는데 반해서 Gnutella에서는 라우터가 하는 행동을 노드 자신이 가지고 있고 경유하는 데이터를 노드 자신과 인접한 노드들에게 정보를 전달하는 방식이다. 이러한 메시지를 전달하거나 논리적인 라우팅을 하기 위해 Gnutella에서는 TCP중심의 Broadcast를 사용한다[1, 2].

이러한 분산형 P2P의 특징 중 검색에 대한 요청메시지와 이에 대한 응답 메시지를 전송하기 위한 구조적인 형태에서 검색 요청을 위한 패킷의 범위를 지정하거나 논리적인 라우팅을 위한 방법으로서 TTL(Time To Live)을 사용하는데, 이러한 TTL의 값이 커질수록 응답 메시지가 지연되는 검색 요청 데이터의 지연 문제를 해결하기 위하여 TTL을 제한하는 방법도 있다.

또 Garbage를 갖는 ping 메시지로 인해 쓰레기 정보가 Broadcasting 되었을 때를 위해 ping 메시지를 검사하여 막는 방법이 있는데 이러한 방법들은 요청과 응답 메시지에 대한 보안과 관련이 있다. 사용자가 이러한 프로토콜을 사용하는 응용프로그램에서 프로토콜을 인지하고 Garbage 정보를 포함 할 수 있기 때문에 프로토콜에 대한 보안과 암호화가 필요하다.

형태의 분석에서도 볼 수 있듯이 연결성과 확장 가능성이 가장 높은 이러한 순수형 P2P는 상대적으로 관리적인 측면과 보안성에 있어서는 형태의 특성상 단점으로 나타나고 있다.

3. 시스템 설계

3.1 시스템 형태

형태적인 측면에서는 부분적인 그룹관리가 필요로 하기 때문에 관리적인 측면에서 부분적으로 장점을 갖고 있는

계층형태를 전체의 형태로 두고 각 계층의 노드는 여러 개의 노드가 존재하는 클라이언트/서버형태를 혼합하여 (그림 1)과 같이 구성하였다.

결정보 및 그룹정보를 관리한다. BDP는 DP다음으로 우선 순위가 큰 것을 지정하고 BDP는 각 Peer들에 대한 상태정보와 연결 정보들을 DP로부터 받고 Cache로 가지고 있지만 DP가 기능들을 수행하고 있는 동안에는 아무런 기능을 갖지 않는다.

(그림 1) 시스템의 형태도

이러한 형태에서 각 단말 계층의 많은 노드들을 관리하는 그룹의 서버 형태로 존재하고 이러한 서버는 상위 계층과의 연관성에 있어서 계층의 특성과 세부적인 특성에 의해 나누어지게 되며, 이러한 형태는 그룹별로 특수 목적에 따라 연결성을 지니고 있기 때문에 사용자의 목적에 따른 분류로서 인터넷의 검색 방법 중 카테고리를 선택하여 원하는 정보의 서버를 찾아가는 형태를 취하도록 설계하였다.

(그림 2) 그룹간의 DP와 BDP의 지정과 역할

3.2.2 DP와 BDP의 재지정

네트워크 상에서 중앙 집중형 모델은 여러 다른 지역에 걸쳐 분산되어 있는 프로그램들을 연결시켜주는 편리한 수단을 제공한다. 하지만 이러한 모델은 서버가 연결을 잃었을 때 클라이언트들의 요청에 응답을 해줄 수 없다. 서버가 존재하고 그 서버에 의해 연결정보를 관리한다는 의미에서는 변형 P2P의 접속 서버와 같다. 변형 P2P에서 접속 서버나 DP의 끊김 현상이 일어나면 그 서버에 평면적이거나 단층구조로 연결되어있던 Peer들의 접속이 동시에 끊기게 된다. 이를 해결하기 위한 복구 시스템으로서 Peer들 중 DP를 선정할 때와 같은 방법으로 BDP가 선출된다.

3.2 시스템의 계층 관리 및 체계

3.2.1 DP(Designated Peer)와 BDP(Backup DP)의 지정

라우팅 프로토콜인 OSPF(Open Shortest Path First)에서는 AS내의 그룹에 대한 정보의 전달을 위해 각 그룹에 DR(Designated Router)과 BDR(Backup DR)을 지정한다. DR은 BDR간의 Master-Slave관계를 형성하여 계층의 상태 정보를 주고 받음으로써 각각의 경로에 대해 동기화 한다. BDR도 DR처럼 모든 정보를 받지만 DR이 동작중인 경우에는 정보에 대한 기능을 수행하지 않는다[7-9].

계층형 관리 구조에서 (그림 2)과 같이 그룹에서 OSPF의 LSA(Link State Advertisement)에 해당하는 Peer의 상태정보와 연결정보를 받는 DR의 역할을 하는 DP를 지정하는데 DP를 결정하는 우선순위는 정보를 통합하는 Peer로서의 신뢰성을 고려하여 접속한 시간으로부터의 Age를 두어 계산하여 가장 큰 것으로 지정한다.

이렇게 DP를 지정하면 DP는 Peer로부터 상태정보와 연

(그림 3) DP와 BDP의 재지정 과정

BDP는 각 그룹의 계층의 상태 정보와 갱신 정보들을 그

룹 내에 속해 있는 DP로부터 전달 받고 이 정보들을 Cache로 지니고 있지만 DP가 동작 중에는 Cache를 사용하는 일은 없다. 예의 끊김 현상이 물리계층이나 응용계층에서 발생하였을 경우에만 동작을 한다.

3.3 프로토콜 설계

(그림 3)에서 DP의 끊김이 발생하였을 경우 먼저 DP의 끊김을 발견하는 시점에서 BDP는 Peer들로부터 DP와 동일하게 전달 받은 그룹내의 Peer정보들을 가지고 DP의 권한을 갖고 동작을 하고 상위 계층의 DP가 지닌 하위 계층의 정보에서 기존의 DP에 대한 정보가 수정되고 자신이 DP가 되었음을 통보한다. 둘째로 계층 그룹의 Peer들에게 바뀐 DP가 자신의 IP 정보를 Broadcasting 한다.

이렇게 DP가 재지정되고 나면 BDP가 부재하게 되고 이를 위해 BDP가 선정되는 과정처럼 (그림 3)과 같이 BDP였던 Peer가 DP가 되는 과정에서 계층 Peer들인 P1, P2, P3들 중 우선순위가 가장 큰 것을 선택한다. (그림 3)에서는 P1이 가장 큰 것이기 때문에 P1이 BDP로 선정되었고 갱신정보가 모든 Peer들에게 전달된다.

3.4 접속 상태의 확인과 전달

OSPF에서 ASBR(Autonomous System Boundary Router)이 다른 AS와의 통신을 위해 Backbone-라우터에 연결된 Area의 ABR을 통해 Area에 대한 LSP(Link State Packet)을 주고 받으며 LSA Type에 맞는 정보를 갱신한다[7-9].

(그림 4) Peer의 상태 정보의 확인과 전달에 대한 리스트를

접속 관리 서버는 ASBR과 같은 계층적 위상에 속한다. (그림 4)와 같이 새로운 Peer의 접속이나 기존 접속되어 있던 Peer의 상태 종료에 대한 정보를 받아서 DP는 Peer들의 정보를 갱신하고 통합하여 지니고 있다.

새로운 사용자가 접속 관리서버를 통해 접속하기 위해

등록하고 계층정보를 받아 원하는 계층에 접속하기 위한 프로토콜은 <표 1>과 같다.

<표 1> 접속을 위한 프로토콜

DP와 BDP, Peer의 상태에서 해제를 하거나 접속을 끊을 때에는 <표 2>와 같은 프로토콜을 사용한다.

<표 2> 접속 해제를 위한 프로토콜

DP와 BDP, Peer들의 상태에 대한 동적인 변환에 사용되는 프로토콜은 <표 3>과 같다.

<표 3> 상태 재지정을 위한 프로토콜

4. 시스템 구현

4.1 동작 과정

4.1.1 on-demand 방식

그룹에서 선택한 계층 정보에 소속된 그룹의 Peer들에 대한 리스트를 보여주는 과정은 DP가 Peer들의 목록을 Cache로 관리하고 있다가 새로운 Peer의 가입과 동시에 가입한 Peer에게로 Peer들의 목록 정보인 Cache의 내용을 전송하며 (그림 5)와 같다.

본 시스템의 접속 관리 서버에서는 각 계층별 정보와 DP 정보만을 유지하고 접속을 처음 시도하는 Peer들에 대해 DP정보를 전달하는 역할을 맡고 있다. DP가 계층별 Peer들에 대한 상태정보를 가지고 있는 것은 서버에 많은 노드들을 관리하는 중앙 집중형 공유 모델보다 트래픽을 분산

시킬 수 있다.

(그림 5) On-Demand 방식

4.1.2 상태의 변경과 기능

내부에서는 프로그램이 시작되고 접속 관리 서버와의 접속이 일어 나고 난 이후부터 상태에 대한 변화가 가능하기 때문에 서버 소켓을 열어서 쓰레드를 이용하여 상태 변경에 관련된 메시지가 오는지에 대한 검사가 이루어진다. 상태의 변화와 각각의 메시지에 대한 기능을 하기 위한 중첩 쓰레드가 실행된다. 계층정보를 보고 자신이 원하는 그룹에 가입하는 순간 자신의 상태가 결정된다. DP가 가입한 그룹에 없으면 DP가 되는 것이고 DP는 존재하는데 BDP가 존재하지 않을 경우에는 BDP가 되는 것이다. 이 경우를 제외하고는 Peer의 상태가 되는 것이다. Peer의 상태에서는 SetBDP 메시지가 오기 전까지는 항상 자신의 Peer상태에 맞는 기능을 수행한다. DP로부터 Ping 메시지가 오면 Pong 메시지를 보낸단든지 하는 Peer의 기능만을 수행한다. 만약 SetBDP 메시지가 오면 자신의 상태를 BDP로 바꾸게 된다. 같은 방법으로 상태가 BDP일 때 UnsetDP 메시지가 오면 자신을 DP로 설정하고 소속된 계층의 모든 Peer들에게 Broadcast로 ResetDP 메시지를 보내며 접속 관리 서버에게 ResetDP 메시지를 보낸다. 하지만 DP가 물리적으로 연결이 해제되거나 예외적인 종료로 인해서 연결성을 잃으면 DP의 종료를 알 수가 없다.

BDP는 이러한 DP에 대해 Ping 메시지의 전송으로 그 상태를 검사하게 된다. (그림 6)은 BDP상태에서의 DP 연결성 검사와 상태 변경 과정이다.

(그림 6)은 DP에 대한 연결성을 검사하고 BDP가 자신 스스로 DP로 설정하며 하위 Peer들에 대해 우선순위가 가장 큰 Peer에게 SetBDP 메시지를 전송하면, 메시지를 받은 Peer는 자신의 상태를 BDP로 설정하고 기능을 수행하게 된다. 정상적인 Peer의 상태 변경은 Peer에서 BDP로, BDP에서 DP의 순서대로 상태가 변경된다.

(그림 6) BDP에서의 DP연결성 검사와 상태 변경 과정

4.2 프로토콜 암호화

P2P 시스템에서 공유정보와 처리하려는 특수한 목적의 행위를 하기 위한 주체는 클라이언트가 된다. 최종적으로 P2P 시스템에서는 연결하려는 Peer들이 각각 서버와 클라이언트가 되어 메시지를 전송하고 원하는 정보를 교환하게 되는 것이다. 본 논문에서 제안하고 구현한 암호화 방법은 이렇게 서로 연결하기 전에 노드들이 각자 생성한 공개키를 교환하여 정보를 암호화할 수 있게 키 교환이 이루어진다. 공유 정보를 지닌 노드에 연결을 요청하는 노드들은 클라이언트가 되는 것이기 때문에 공유정보를 지닌 노드는 일대다(1:m)를 지원하므로 각각의 클라이언트들에 대한 키 테이블을 관리하며 매칭시켜 요청한 클라이언트들에 맞는 암호화를 시키고 전송하게 된다.

정보를 요구한 Peer는 클라이언트 입장이 되고 정보를 지닌 Peer는 서버가 되므로 전자를 Pc(Peer Client), 후자를 Ps(Peer Server), Pc가 가진 비밀키를 S_{pc} , 공개키를 P_{pc} 로 나타내고 Ps가 가진 키를 S_{ps} , P_{ps} 로 나타내어 키 교환이 이루어지는 과정은 (그림 7)과 같다. 처음 프로그램이 초기화될 때 공개키 기반의 암호화 알고리즘을 통해 공개키와 비밀키가 생성되고 키를 검증하기 위한 프로세스가 실행되어 얻어낸 키들을 Peer들 모두가 가지고 있는 형태를 지닌다[11].

Ps는 Pc의 공개키를 획득하였을 때 키 관리 프로세스를 통해 저장하거나 변경하고 인증과 키 교환이 이루어지면 상대방의 공개키에 의해 자신의 메시지를 암호화하여 통신이 이루어진다.

루게 된다.

제시한 형태는 계층형태에 중앙집중형태를 혼합한 형태이므로 TTL과 깊이인 L 값이 같은 의미로 해석되었고 제시한 P2P에서는 Gnutella와 같이 Broadcast 방법을 사용하지 않고 Unicast 방법을 사용하기 때문에 인접한 노드들이 Gnutella에서 전달받은 노드는 포함되지 않는 형태로 함수식을 도출하였다.

함수식 $f_g(n, x, y)$ 와 $f_d(n, x, y)$ 를 이용하여 각 네트워크의 도착 가능한 노드 수를 비교한 것이 (그림 8)이다.

(그림 7) 키 교환과 암호화 메시지 전송 과정

5. 성능 평가

5.1 깊이와 TTL의 변화율에 따른 성능 비교

Gnutella의 전송 메시지와 동작 원리에 있어서 Broadcasting을 사용하기 때문에 트래픽이 많이 발생한다. 이러한 트래픽을 계산하고 구현된 시스템의 성능을 평가하기 위해 <표 4>와 같이 수학적 접근을 통하여 Gnutella와 본 시스템을 분석하였다.

<표 4> 시스템의 성능 평가에 사용된 변수와 함수식

Gnutella에서 n값과 t값의 변화율로서 Gnutella가 가질 수 있는 전체의 도달 가능한 노드 수를 구하는데 사용되는 함수식이 $f_g(n, x, y)$ 이다[2].

평가에 사용된 $f_d(n, x, y)$ 는 $f_g(n, x, y)$ 에 대한 평가식의 도출 방법과 동일하고 단, 제안된 시스템에서의 N은 그룹에서의 DP들만 적용하였다. 형태적으로 분산형태로 지닌 노드들의 집합을 한 노드를 지정해서 그 노드와 인접한 노드, 그리고 그 노드로부터의 TTL값에 해당하는 노드들의 연결 고리 수에 해당하는 값을 통해 전체 도착 가능한 노드 수를 얻어 낼 수 있는데 여기서 한 노드를 지정하여 그 노드로부터 출발하는 것은 다시 말해 그 노드를 루트 노드로 지정하고 그물을 잡아 당겼을 때의 형태는 계층형을 이

(그림 8) Gnutella와 제시한 시스템의 도착 가능한 전체의 노드 수 비교

Gnutella의 TTL값이 8이고 인접하는 노드 수인 n이 8값을 지닐 때 식 $f_g(n, x, y)$ 를 이용하여 얻어낸 도착 가능한 전체의 노드 수는 770만개가 된다. 하지만 본 시스템에서 TTL값과 동일한 계층의 깊이인 l이 8값을 지니고 n이 8값을 지닐 때 식 $f_d(n, x, y)$ 를 이용하여 얻어낸 도착 가능한 전체의 노드 수는 1900만개가 된다. 결과적으로 같은 깊이와 TTL값을 지닐 때 즉, 논리적인 연결의 크기가 같을 때 본 논문에서 제시한 시스템의 네트워크가 더 큰 수용성을 지니고 있다는 것을 증명할 수 있다.

5.2 총 소요 대역폭 비교

시스템에서 차지하는 총 소요 대역폭은 정보 검색에 필요한 요청 메시지와 응답 메시지를 전송하는 모든 노드들의 대역폭을 계산하였다. <표 4>에서 식 $h_g(n, t, s)$ 는 Gnutella에서 검색을 하기 위한 outbound에 소요되는 전체 소요 대역폭을 계산에 적용되고 inbound를 포함한 소요 대역폭의 계산은 식 $i_g(n, t, s)$ 이다[2].

식 $i_g(n, t, s)$ 를 사용해서 inbound와 outbound를 포함한

전체 소요 대역폭을 추정할 수 있고 제안된 시스템의 대역폭은 식 $h_d(n, t, s)$ 에서 outbound를 구할 수 있고 inbound는 outbound와 같은 형태로 이루어지므로 $2 \times h_d(n, t, s)$ 로 나타낼 수 있다. 이러한 총 소요 대역폭은 메시지의 크기를 같게 하고 앞에서 계산된 본 시스템과 Gnutella에서 도착 가능한 전체의 노드 수를 바탕으로 총 소요 대역폭을 계산하였다. 결과 값에 대한 비교는 (그림 9)와 같다.

(그림 9) Gnutella와 제시한 시스템의 총 소요 대역폭 비교

Gnutella 네트워크에서의 소요 대역폭은 TTL이 8이고 패킷의 크기가 83Bytes일 때, 1.2GBytes 정도가 소요되지만 본 논문에서 제시한 동적인 변형 P2P 네트워크에서는 0.4GBytes 정도가 소요된다. 따라서 대규모 네트워크에서 같은 계층 높이와 TTL일 때, 연결 가능한 노드의 수가 본 시스템에서 많음에도 불구하고, 총 소요 대역폭은 1/3정도로 적게 차지하는 것으로 본 시스템의 우수성을 검증할 수 있다. 또한 각 계층의 그룹에서 관리하는 DP와 BDP의 존재로 인해 신뢰성을 제공한다.

6. 결 론

본 논문에서는 대규모 네트워크에서의 공유 자원에 대한 계층 분류와 그룹관리의 효율성, 그리고 공유지에 대한 신뢰성을 향상시키기 위한 방법으로 계층구조와 서버/클라이언트 형태를 지닌 변형 형태와 백업 시스템을 지닌 동적 변형 P2P 시스템을 설계하고 구현하였다. 구현 시스템을 Gnutella와 비교 평가하여 확장 가능성과 총 소요 대역폭의 탁월한 우수성을 검증하였다.

향후 본 변형 P2P 응용프로그램은 사용자 인증의 개선과

공유하는 자원이 파일이라는 국한적인 의미에서의 공유가 아닌 소프트웨어와 하드웨어 자원 및 정보 가전이나 모바일, PDA 그리고 하드웨어적인 모듈 등으로의 확장에 대한 연구와 그리드 컴퓨팅으로의 확장에 필요한 개선 방안 등이 연구되어야 할 것이다.

참 고 문 헌

- [1] Andy Oram, 'Peer-to-Peer Harnessing the Power of Disruptive Technologies', O'Reilly, September, 2001.
- [2] Jordan Ritter, "Why Gnutella Can't Scale. No Really," <http://www.darkridge.com/~jpr5/doc/gnutella.html>, February, 2001.
- [3] Paul J. Leach, Rich Salz, Certco, "UUIDs and GUIDs," Internet-Draft. August, 1998.
- [4] Nelson Minar, "Distributed Systems Topologies : Part 1", http://www.openp2p.com/pub/a/p2p/2001/12/14/topologies_one.html, December 14, 2001.
- [5] Nelson Minar, "Distributed Systems Topologies : Part 2", http://www.openp2p.com/pub/a/p2p/2002/01/08/p2p_topologies_pt2.html, January, 2002.
- [6] Clip2 Distributed Search Solutions, "The Gnutella Protocol Specification v0.4. Distributed Search Services," <http://dss.clip2.com>, September 2000.
- [7] John Moy, "version 2 of the OSPF protocol", RFC 2328, April, 1998.
- [8] Cisco, 'Open Shortest Path First', Cisco Documentation, Chap 46, February, 2002.
- [9] Ravi Malhotra, 'IP Routing', O'Reilly, January, 2002.
- [10] W. Richard Stevens, "UNIX Network Programming Volume 1 [Second Edition], Vol.1, No.1, pp.3-5, March, 2000.
- [11] B. Kaliski, J. Staddon, 'RSA Cryptography Specifications Version 2.0', RFC 2437, October, 1998.
- [12] Sing Li, 'Peer-to-Peer Computing with Java', Wrox Press, December, 2001.
- [13] Elliotte Rusty Harold, 'Java Network Programming Second Edition', O'Reilly, August, 2000.

이 석 희

e-mail : Netause@cju.ac.kr

2000년 청주대학교 컴퓨터정보공학과
공학사

2003년 청주대학교 전산정보공학과
공학석사

2003년~현재 청주대학교 컴퓨터정보공학과
박사과정

관심분야 : P2P, 그리드 컴퓨팅, 홈 네트워킹, 유비쿼터스 컴퓨팅,
이동 에이전트 S/W

조 상

e-mail : Sangcho@cju.ac.kr
1975년 서울대학교 물리학과 이학사
1986년 Iowa State University 전산학석사
1990년 University of Texas at Dallas
전산학박사
1992년~현재 청주대학교 컴퓨터정보공학과
부교수

관심분야 : 실시간 시스템, 소프트 컴퓨팅

김 성 열

e-mail : Sskim@cju.ac.kr
1982년 숭실대학교 전자계산학과 공학사
1987년 숭실대학교 대학원 전자계산학과
공학석사
1992년 숭실대학교 대학원 전자계산학과
공학박사

1982년~1984년 한국전력공사 전자계산소 근무

1984년~1990년 오산대학 전자계산과 교수

1997년~1998년 호주 QUT ISRC 객원 교수

1990년~현재 청주대학교 컴퓨터정보공학과 교수

관심분야 : 컴퓨터 네트워크, 컴퓨터 보안, 분산 객체 시스템