

개선된 노드 분산율을 위한 적응적 P2P 검색 알고리즘

김분희*, 이준연**

An Adaptive Peer-to-Peer Search Algorithm for Reformed Node Distribution Rate

Boon-Hee Kim*, Jun-Yeon-Lee**

요약

제한된 통신 환경 상에서 P2P 어플리케이션에 의해 발생하는 과도한 트래픽은 네트워크 대역폭 문제와 연관된다. 또한 P2P 시스템은 피어들의 P2P 오버레이 네트워크에의 연결성이 매우 약한 약결합 시스템으로 검색 단계에서 원하는 자원을 찾는다 하여도 다운로드 단계에서 반드시 그 피어에서 자원을 다운 받을 수 있다는 보장이 없다. 이전 검색 알고리즘[1]에서는 이러한 문제를 해결하기 위해 Random Walks 기반의 휴리스틱 피어 선택 기법을 적용하였다. 본 논문에서는 각 피어의 컴퓨팅 파워에 영향 받는 노드 분산율을 개선하기 위해 이전 알고리즘[1] 기반의 적응적 P2P 검색 알고리즘을 제안하였다. 또한 트래픽 양을 줄이기 위해 부가적으로 질의율 기반의 차별적 리플리케이션 기법을 채택했다. 제안한 시스템의 성능 평가 결과 검색의 방향성과 통신 발생의 분포 측면에서 적절한 타협점에서 동작함을 확인하였다.

Abstract

Excessive traffic of P2P applications in the limited communication environment is considered as a network bandwidth problem. Moreover, Though P2P systems search a resource in the phase of search using weakly connected systems(peers' connection to P2P overlay network is very weakly connected), it is not guaranteed to download the very peer's resource in the phase of download. In previous P2P search algorithm[1], we had adopted the heuristic peer selection method based on Random Walks to resolve this problems. In this paper, we suggested an adaptive P2P search algorithm based on the previous algorithm[1] to reform the node distribution rate which is affected in unit peer ability. Also, we have adapted the discriminative replication method based on a query ratio to reduce traffic amount additionally. In the performance estimation result of this suggested system, our system works on a appropriate point of compromise in due consideration of the direction of searching and distribution of traffic occurrence.

▶ Keyword : P2P(Peer-to-Peer), 검색 알고리즘(Search Algorithm), 객체 리플리케이션(Object Replication)

• 제1저자 : 김분희 교신저자 : 이준연

• 접수일 : 2005.06.21, 심사완료일 : 2005.07.28

* 동명정보대학교 멀티미디어공학과 전임강사, ** 동명정보대학교 멀티미디어공학과 조교수

I. 서론

최근 P2P 응용프로그램의 네트워크 트래픽이 압도적으로 증가하고 있다. KT는 18일 초고속인터넷인 메가패스와 기업전용회선의 지난 2004년 2월 국내 트래픽을 분석한 결과 P2P 트래픽의 비중이 88.8%로 집계됐다고 밝혔다. 우리나라와 외국 간에 발생하는 인터넷 트래픽 중에서 각종 P2P 프로그램을 통한 파일 교환 트래픽이 전체의 33.7%, 메시지를 통한 P2P 트래픽이 42.7%로 집계됐다[2][10]. 이 중 파일 교환 이전에 파일을 찾기 위한 자원 검색 메시지가 발생하는 트래픽량이 간과할 수 없는 수준이 되었다. 현존하는 P2P 모델은 피어간의 네트워킹 방법과 역할에 따라 순수한 P2P, 간단한 조회 기능 서버를 가진 P2P[3], 조회 서버와 록업 서버를 가진 P2P, 조회/록업/컨텐츠 제공 기능의 서버를 가진 P2P로 분류할 수 있다[4]. 이 중에서 P2P 시스템의 탄생 배경에서 볼 수 있듯이 사용자 피어들이 각각 P2P 시스템에서 독립된 자원 제공 및 제공 받는 작업의 주체로써 하나의 피어의 기능 마비가 전체의 P2P 시스템에 영향을 주지 않는 순수 P2P 시스템이 궁극적인 P2P 시스템의 모델이다.

자원 검색 기법으로 P2P 모델과 부합되어 서버의 기능이 있는 중앙집중식 디렉토리 모델, 순수한 P2P 환경에서 제안된 검색 모델로 검색 메시지를 전달하는데 있어 피어들의 참여여부에 따라서 모델들 구분할 수 있는 브로드캐스트 요청 모델, 순수한 P2P 네트워크 기반의 검색 모델로 공유된 자원의 ID를 기반으로 자원 검색 도큐먼트 라우팅 모델로 구분할 수 있다. 중앙집중식 디렉토리 모델로 Napster[5], 브로드캐스트 요청 모델로 Modified-BFS, Random Walks, GUESS, DRLP, 도큐먼트 라우팅 모델로 DHT[8], VBST 등이 있다. 각각의 분야들 모두 현재 활발히 연구가 진행되고 있다.

순수한 P2P 모델 기반의 브로드캐스팅 모델에 속하는 검색 기법들은 검색 성공률 측면에서는 객관적인 성능이 상당히 보장되어 실제의 P2P 시스템에서 유용하게 이용되는 방법이다. 그러나 제한된 통신 환경에서 P2P 어플리케이션에 의해 발생하는 과도한 트래픽은 네트워크 대역폭 문제와 연관된다. 또한 P2P 시스템은 피어들의 P2P 오버레이 네트워

크에의 연결성이 매우 약한 약결합 시스템으로 검색 단계에서 원하는 자원을 찾는다 하여도 다운로드 단계에서 반드시 그 피어에서 자원을 다운 받을 수 있다는 보장이 없다. 이전 검색 알고리즘[1]에서는 이러한 문제를 해결하기 위해 Random Walks 기반의 휴리스틱 피어 선택 기법을 적용하였다. 본 논문에서는 각 피어의 컴퓨팅 파워에 영향 받는 노드 분산율을 개선하기 위해 이전 알고리즘[1] 기반의 적응적 P2P 검색 알고리즘을 제안하였다. 또한 트래픽 양을 줄이기 위해 부가적으로 질의율 기반의 차별적 리플리케이션 기법을 채택했다. 제안한 시스템의 성능 평가 결과 검색의 방향성과 통신 발생의 분포 측면에서 P2P 약결합 시스템을 위한 솔루션으로 적합하지 검증한다.

II. 관련 연구

2.1 P2P 기반 검색 기법

이 장에서는 P2P 시스템의 구축을 위한 대표적인 모델에 대해 알아보고, 현재 논의되고 있는 주요 검색 시스템을 분류하여 각 검색 메커니즘의 특징을 살펴보겠다.

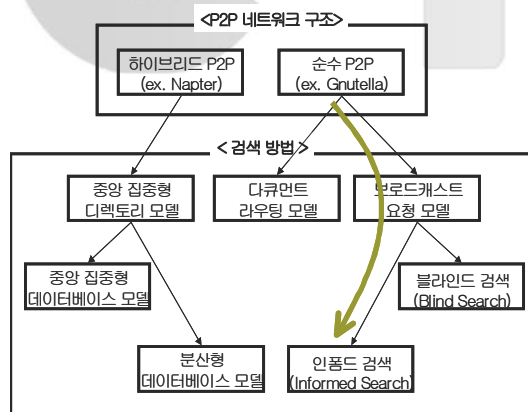


그림 1. P2P 검색 알고리즘 분류
Fig 1. Classification of P2P search algorithm

(그림 1)에서와 같이 P2P 시스템에서 자원의 검색 기법은 P2P 네트워크 구조에 따라 분류 할 수 있다. 먼저 P2P 네트워크 구조는 피어간의 네트워킹 방법과 역할에 따라 순

수 P2P(Pure Peer-to-Peer), 간단한 조회 기능 서버를 가진 P2P(P2P with a simple discovery server), 조회 서버와 룩업 서버를 가진 P2P(P2P with discovery and lookup servers), 조회/룩업/컨텐츠 제공 기능의 서버를 가진 P2P (P2P with discovery, lookup and content servers) 모델로 분류할 수 있는데, 여기서 순수한 P2P 모델을 제외한 나머지 서버의 기능이 있는 모델을 일반적으로 하이브리드 P2P로 묶어서 분류한다. 하이브리드 P2P 구조에 속한 검색 기법은 중앙집중식 모델로써 대변되는데, 피어의 연결정보를 보유한 중앙 서버가 검색 대상 데이터의 보유 여부에 따라 중앙집중형 데이터베이스 모델과 분산형 데이터베이스 모델로 나뉜다. 순수한 P2P 모델의 경우는 검색 기법의 특징에 따라 브로드캐스트 요청 모델, 도큐먼트 라우팅 모델로 나뉜다. 실제 가장 많이 이용되는 모델인 브로드캐스트 요청 모델은 검색 메시지의 전달에 특정 정보의 이용 여부에 따라 인포드 검색[과 블라인드 검색 기법으로 나뉜다[11][12][15].

순수 P2P 컴퓨팅 환경을 기반으로 하는 검색 기법 가운데, Gnutella[6][7]는 자원 검색을 위해 사용하기 가장 쉬운 방법으로 불규칙적으로 연결된 네트워크 구조에 적용하기 용이하다. 그러나 검색 기법 가운데 가장 많은 네트워크 트래픽 양을 나타낸다. 변형된 BFS 기법은 Gnutella 알고리즘과 비교해 보면 검색 메시지에 의한 네트워크 트래픽 양이 적지만 인접한 모든 피어에게 질의 메시지를 전송하는 것이 아니므로 적중률이 낮다. Random Walks는 검색에 의해 발생하는 메시지의 수를 제한하여 전송하는 방법으로 변형된 BFS 알고리즘에 비해 메시지 발생량이 적다. 그러나 검색 적중률이 낮다. 이러한 검색 알고리즘의 특징은 검색 성공률과 트래픽 양 측면에서 효과적인 결과를 얻고자 연구의 방향이 진행되고 있음을 알 수 있다. 이와 더불어 P2P 환경의 기본 특징인 약결합 환경에 적응적인 기법이 보안되어야 한다.

2.2 P2P 기반 객체 리플리케이션 기법

P2P 시스템은 기본적으로 피어의 연결과 해체가 자유로운 피어들 간의 결합의 상태 변화가 심한 약결합 환경이다. 이러한 약결합 환경에서 자원을 검색하고, 검색된 자원을 보유한 피어에게서 다운로드 받는 과정을 거치게 된다. 이러한 과정에서 다운로드 대상 피어의 연결 해제로 인해 검색의 주체인 피어는 원하는 자원을 완전히 다운로드 받지 못하는 경우가 발생할 수 있다. 이러한 과정에서 검색 주체 피어는 해당 P2P 시스템에 신뢰하기 힘들다.

검색 주체 피어에 신뢰성을 향상시키기 위한 다양한 방법이 제시되고 있는데, 대표적인 연구의 흐름은 먼저 여러 피어로부터 동시에 다운로드 받는 방법이다. 이는 하나의 자원 제공 피어가 P2P 시스템에서 연결 해제를 하여도 다른 참여 피어들 가운데 하나의 피어가 그 역할을 책임지면 되므로, 검색 주체 피어는 다운로드 과정에서의 피어의 연결 해제에 대해 고려하지 않아도 되는 것이다. 이 방법은 신뢰성 향상이라는 측면에서는 상당한 성능을 보이지만, 네트워크 트래픽 측면에서 트래픽 발생량이 다른 방법에 비해 네트워크 성능과 연관지어 볼 때 영향력이 높다.

다른 연구 방법으로 자원의 발견할 수 있는 빈도를 높여 주는 대안이 있다. 이는 해당 자원의 분포를 넓힘으로써 자원 검색시 적중될 확률을 높여주고, 약결합 환경에서의 다운로드 대상 피어의 연결 해제로 인한 영향력을 줄여줄 수 있다. 이러한 방법을 리플리케이션 기법이라 한다[9][13]. 일반적으로 분산시스템과 같은 공동 작업과 관련한 분야에서 발생할 수 있는 결함을 허용하기 위해 제공하는 서비스로 사용되는데, 여기서는 리플리케이션된 객체 간의 일관성 유지 및 결정적인 동작이 중요하다. 이는 공동 작업중인 상태에서 주기적으로 객체를 리플리케이션 함으로써 수행중인 컴퓨터의 결함으로 대체될 필요가 있을 때 중요하게 작용되기 때문이다. 하지만 여기서 언급되는 P2P 시스템의 신뢰도 향상을 방안으로써의 리플리케이션 기법은 완전한 파일 형태의 자원을 피어간에 다운로드 받는데 있어 중도에 자원 제공 피어의 연결해제로 인한 검색 주체 피어의 자원 다운로드 측면에서의 신뢰성을 보장하기 위해 P2P 네트워크 상에서 인기 자원의 분포를 높여주는 방법을 말한다. 즉, 리플리케이션 될 대상의 선정 방법과 리플리케이션 될 대상의 리플리케이션 수를 결정하는 부분이 주요 관심사인 것이다.

자원을 요청하는데 있어 그 자발성에 따라 리플리케이션 기법을 분류할 수 있는데, 확률적 리플리케이션 기법은 자원에 대한 요청이 있을 경우만 그 자원이 리플리케이션의 대상이 되는 경우이다. 대표적인 예로 Gnutella가 이에 해당한다. 다음으로 자발적 리플리케이션 기법은 자원에 대한 요청이 없음에도 불구하고 리플리케이션이 이루어지는 자발적인 정책을 이용하는 경우로 Freenet이 대표적인 예이다.

또 다른 리플리케이션 분류 방법으로 리플리케이션 대상 자원의 개수를 정하는 기준에 따라 나눌 수 있는데, 단순 리플리케이션 기법의 경우 질의 빈도수와 무관하게 리플리케이션 대상으로 선택된 자원을 일정한 수로 리플리케이션 하는 방법이다. 이에 비해 비례 리플리케이션 기법의 경우 질의 빈도수에 따라 리플리케이션 대상 자원의 수를 차별

적으로 정하는 방법이다. 이는 신뢰성 측면에서 적극성 측면이 부족하지만 알고리즘의 복잡성 및 트래픽 양의 측면에서 앞선 다중 피어로부터의 다운로드 기법에 비해 적절성이 인정되고 있다.

III. 제안한 검색 알고리즘

이 장에서는 본 논문에서 제안한 노드 분산을 개선을 위한 브로드캐스팅 기반 검색 알고리즘을 제시한다. 제안한 순수 P2P 모델 기반 검색 알고리즘은 K Walks 기반 P2P 시스템의 자원 검색 과정에서 발생하는 네트워크 트래픽 양을 최소화한 것과 P2P 네트워크를 구성하는 각 피어의 컴퓨팅 파워에 영향 받는 노드 분산율의 개선을 목표로 하며, 기본적인 P2P 시스템의 성능에 미치는 영향에 대해 평가하고자 한다. 제안한 검색 기법은 이전의 LTO[1]와 마찬가지로 운영 환경 상에서의 몇 가지 가정하에 설계되었다. 먼저 자원 요청을 위한 검색 진행 메시지의 발생 및 진행의 과정에서 초기 이웃 피어의 선택과정은 K Walks의 기본 원리를 따른다. (그림 2)의 LTO는 각 피어들이 저장하는 연관성 확률값은 해당 자원을 보유한 피어의 이전 피어에 저장되는데, 메시지 발생 피어로부터 다음 피어로 메시지가 전개되는데 있어 선택 기준이 된다. 이러한 연관성 확률값을 자원 보유 피어로부터 떨어진 홉 수에 따라 차등적으로 적용하여 차별화 한 것이 LTO 검색 알고리즘이다.

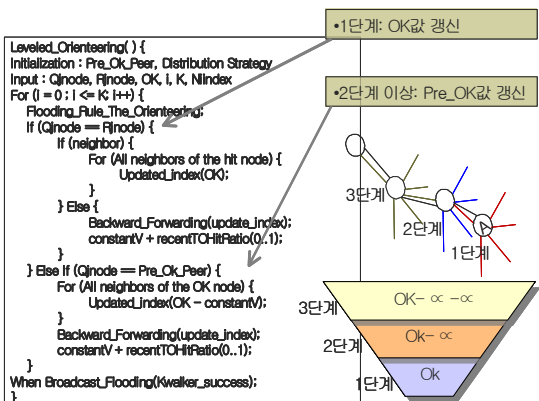


그림 2. LTO 알고리즘 도식화
Fig 2. A diagram for LTO algorithm

브로드캐스트 검색을 지원하는 P2P 시스템들의 메시지 발생량을 줄이기 위해 해당 피어 보유한 이웃 피어와의 차수 가운데 일부분을 선택하여 검색이 진행되는 연구가 요구되지만 어렵게 찾아진 자원 보유 피어에 대한 정보를 다수의 피어의 다양한 방향에서의 접근될 수 있도록 근접성을 넓히는 것은 다양하게 검색 성능에 영향을 준다. 제안한 브로드캐스팅 기반 검색 알고리즘의 근접성을 개선하기 위한 모델을 제시한다. 또한 검색 트래픽에 미치는 영향을 최소화하기 위한 적합성 측면을 검토한다. 피어의 자원 보유 피어로의 근접성을 개선하기 위해 해당 검색 메시지 발생량이 증가됨에 따른 걱정 수준의 개선 모델을 제시하고자 한다.

3.1 기본 구조

이 절에서는 제안한 검색 알고리즘의 근접성을 개선하기 위한 검색 모델로 확장하고자 한다. 검색 진행 메시지의 발생 및 진행의 과정에서 초기 이웃 피어의 선택 과정은 K Walks의 기본 원리를 적용하며, (그림 3)에 제안한 알고리즘을 표현하고 있다. 기본적으로 검색을 진행하고자 하는 주체 피어는 찾고자 하는 자원에 대해 하나의 엔트리로 구성된 지역 인덱스 유지하며, 검색의 과정을 거치게 되는 해당 이웃 피어들도 각각의 이웃 피어에 대한 지역 인덱스 값을 유지하게 한다. 또한 이 지역 인덱스는 각 자원 당 하나로 배당되며, 인덱스에 저장되는 값은 이웃피어와의 연관성 확률 값을 반영한 값이다. 이러한 초기화 작업을 거쳐서 검색의 과정을 진행하는 Walker를 기반으로 실제적인 검색 과정이 시작되며, 질의의 분석 과정에서 지역 인덱스의 연관성 확률 값을 바탕으로 우위에 있는 이웃피어를 선택하여 질의가 전파된다.

이 절에서는 검색 주체 피어가 해당 자원이 있는 피어로의 근접성의 정도를 더욱 높이기 위한 정책으로써 단계별 OK 정책을 제안한다. (그림 3)에서와 같이 OK 정책의 단계를 정함에 따라 자원에 대한 근접율을 기존의 방법에 비해 향상시키고자 한다. 검색의 과정이 시행되기 이전에 OK 값의 단계를 정하는 작업이 선행되며, 기존의 정책은 단계별 OK 정책을 인용하면 1단계 OK 정책을 구현한 것이다. 2단계 이상의 OK 정책을 선택함으로써 비로소 (그림 3)의 과정을 거치게 된다. (그림 3) 알고리즘에서 표현된 모듈의 의미는 다음과 같다.

- K : Walker 수를 나타낸다.
- QP : 질의가 발생된 피어이다.

- RP : 해당 자원이 있는 피어이다.
- QList : QP가 RP 찾을 때까지 해당 경로 저장하고 있다.
- U : 연관값의 갱신 단위이다. 해당 노드의 연관값은 질의 적중 시 U만큼 증가된다.
- OK : 연관값의 최고값이다.
- PL : RP로부터 떨어진 거리에 따라 연관값을 차등 부여할 때, 차등 부여 대상 피어의 범위이다.

제안 알고리즘의 1-48번 라인에 대해 각 라인별 설명을 하면 다음과 같다. 검색 알고리즘이 동작되기 이전에 P2P 내부 구조를 구성 부분은 기본 조건이다. P2P 네트워크 구조적인 측면에서 순수 P2P 모델이며, 오버레이 네트워크 토폴로지는 랜덤 그래프 모델로 초기화 된다. 6-14번 라인은 알고리즘에 이용되는 인수로써 K는 질의 발생 노드로부터의 이웃 노드로 전달될 질의 발생 수, U는 연관값의 갱신 단위를 나타낸다. 17번 라인에서 자원 발생 노드로부터 전파되는 메시지의 수 K만큼 Walker를 발생하면서 18-46번 라인과 같이 각 Walker들은 동일한 과정을 거치게 된다.

▶17번 라인 → For (i = 0 ; i <= K; i++) {

18번 라인의 조건문은 QP가 보유한 해당 자원에 대한 연관값은 모두 동일한 값인지를 의미한다. 모두 초기값일 때도에 해당한다. 15-17번 라인은 검색 메시지를 전파할 노드를 선택하고, 선택된 노드를 QList에 추가함으로써 질의 파생의 라우트를 기록한다. 21번 라인은 이웃 노드 선택에 있어서 Random Walks 룰을 따라 진행함을 의미한다. Random Walks는 이웃 노드 선택에 있어서 임의의 노드에 대해 이미 정해진 개수만큼 선택해서 질의를 파생시킴을 의미한다. 23번 라인은 QList값을 살펴 본 결과 모두 동일한 값이나 모두 초기값으로 셋팅된 경우가 아니라면, 24-26번 라인의 모듈을 진행하게 된다.

- ▶18번 라인 → If (QP가 보유한 연관값은 모두 동일한 값인가?) {
- ▶19번 라인 → 검색 메시지를 전파할 노드 선택;
- ▶20번 라인 → QList에 선택된 노드 추가;
- ▶21번 라인 → 임의의 이웃노드에 메시지 전파;

23번 라인에 조건에 부합할 경우 보유한 연관값 중에서 가장 큰 값을 지닌 노드를 선택하고, 선택된 노드를 QList에 추가함으로써 질의 파생의 라우트를 기록한다. 그리고 실제 선택된 이웃노드에 메시지를 전파하게 된다.

▶24번 라인 → 연관값 중 가장 큰 값의 노드 선택;

25-35번 라인이 변형된 주요 부분으로 25번 라인에서 질의에 맞는 RP를 찾았을 경우에서 시작된다. 질의에 맞는 RP를 찾게 되면 가장 먼저 27번 라인에서와 같이 RP의 이웃노드에 대해 28번과 같이 연관성 확률값의 최고치인 OK 값을 갱신한다. 다음으로 RP 노드의 이웃노드에 대해서 이후에 이용되기 위해 RPList에 해당 노드를 추가하는 작업이 이어진다.

- ▶25번 라인 → If (질의에 맞는 RP를 찾았는가?) {
- ▶27번 라인 → If (RP의 이웃노드 인가?)
- ▶28번 라인 → OK값으로 연관값 갱신;
- ▶29번 라인 → RPList에 선택된 노드 추가;

RP 노드의 이웃노드에 연관값 갱신 작업이 이어진 후 정해진 PL의 단계만큼 33-35번 라인을 반복하게 된다. 33번은 RPList의 해당 이웃노드에 연관값 갱신작업이 수행되고, 35번 라인에서는 연관값이 갱신된 노드들에 대해 각각의 이웃노드에 대한 리스트를 새롭게 구성하게 된다. 이렇게 구성된 RPList를 이용하여 PL의 다음 단계에서 갱신 대상인 RPList로 이용하게 된다. LTO는 QList를 이용하여 연관값 갱신 대상이 결정되었는데, MLTO (Modified LTO)는 갱신 단계에서 구성된 RPList를 이용하여 연관값을 갱신하게 되어 갱신 대상 피어의 결정이 동적이며, QList의 라우트와의 거리가 넓어지게 되는 특징을 보인다. 이러한 부분에 대해서 MLTO가 성능에 미치는 영향은 실험 평가 절에서 다루도록 하겠다.

- ▶31번 라인 → For (정해진 PL단계 만큼 반복) {
- ▶33번 라인 → If (RPList에 해당하는 노드의 이웃노드 인가?)
- ▶34번 라인 → OK값에서 U*PL만큼 뺀 값으로 연관값 갱신;
- ▶35번 라인 → 새로운 RPList 구성;

▶36번 라인 → If (해당 PL에 해당하는 노드의 이웃노드 인가?)

▶38번 라인 → OK값에서 U*PL만큼 뺀 값으로 연관값 갱신

39번 라인은 연관성 확률값의 갱신 대상 노드들 모두가 갱신되도록 반복함을 의미한다. 갱신의 방향은 자원 보유 피어로부터 질의 발생 피어로 진행된다.

42번 라인은 QList에 포함된 노드로써 RP의 이웃노드 이외의 갱신 과정을 거쳐야 하는 해당 노드들을 의미한다.

43번 라인은 연관성 확률값의 갱신 대상 노드들 모두가 갱신되도록 반복함을 의미한다. 갱신의 방향은 자원 보유 피어로부터 질의 발생 피어로 진행된다.

```

1 Modified_LTO ( {
  [가정]
  ...
  -알고리즘의 시작시점은 QP로부터 검색 질의가 발생된 이후이다.
  -K의 수, PL의 수, U, 검색 알고리즘이 수행 될 P2P 환경은 사전에 결정되어 있다.
6  /** -K : Walker 수를 나타낸다. -OK : 연관값의 최고값이다.
    -QP : 질의가 발생된 피어이다. -RP : 해당 자원이 있는 피어이다.
    -U : 연관값의 갱신 단위이다. -RPList : RP의 이웃노드 리스트이다.
    ...
    해당 노드의 연관값은 질의 적중 시 U만큼 증가된다.
    -QList : QP가 RP 찾을 때까지 해당 경로 저장하고 있다.
    -PL : RP로부터 떨어진 거리에 따라 연관값을 차등 부여할 때,
    차등 부여 대상 피어의 범위이다. *****/
14
15 Do {
16 //QP는 K개의 이웃피어에게 검색 메시지를 전파
17 For (i = 0 ; i <= K; i++) {
18   If (QP가 보유한 연관값은 모두 동일한 값인가?) {
19     검색 메시지를 전파할 노드 선택;
20     QList에 선택된 노드 추가;
21     임의의 이웃노드에 메시지 전파;
22   }
23   else { //QP가 보유한 연관값들에 차이가 있을 때
24     연관값 중 가장 큰 값의 노드 선택;
25     QList에 선택된 노드 추가;
26     해당 이웃노드에 메시지 전파;
27   }
28   If (질의에 맞는 RP를 찾았는가?) {
29     Do {
30       If (RP의 이웃노드 인가?)
31         OK값으로 연관값 갱신;
32       RPList에 선택된 노드 추가;
33     } While (RP의 이웃노드들이 모두 갱신될 때 까지)
34     For (정해진 PL단계 만큼 반복) {
35       Do {
36         If (RPList에 해당하는 노드의 이웃노드 인가?)
37           OK값에서 U*PL만큼 뺀 값으로 연관값 갱신;
38         새로운 RPList 구성;
39       } While (이웃노드들이 모두 갱신될 때 까지)
40     }
41     Do {
42       If (QList에 포함된 노드인가?) 기존의 연관값에 U만큼 증가시킴;
43     } While (연관값 갱신 대상 노드들이 모두 갱신될 때까지)
44     Modified_LTO 검색 알고리즘 종료;
45   }
46 } //end of For-statement
47 } While (자원 검색 과정이 끝날 때 까지)
48 }

```

그림 3. 제한한 알고리즘
Fig 3. This Algorithm

▶42번 라인 → If (Qlist에 포함된 노드인가?) 기존의 연관값에 U만큼 증가시킴;

▶43번 라인 → } While(연관값 갱신 대상 노드들이 모두 갱신될 때까지)

47번 라인은 질의 발생 피어로부터 발생될 K Walker에 대해 K 개의 Walker가 조건이 다할 때까지 검색 과정을 모두 마쳐야 함을 의미한다.

▶47번 라인 → } While(자원 검색 과정이 끝날 때 까지)

3.2 기본 동작 모듈

이 절에서는 본 검색 알고리즘을 이용해 실제적인 P2P 시스템을 구현하는데 있어 기본적으로 필요한 모듈을 정의하고, 제안한 검색 알고리즘이 위치하는 검색 모듈을 설계한다. (그림 4)는 피어의 연결 요청에 대한 처리 시 요구되는 서버 소켓 관련 요소의 구성이다. P2P 응용 프로그램에서 이용하는 별도의 포트 번호로의 연결이 들어오면 연결수락 작업이 이어지게 된다.

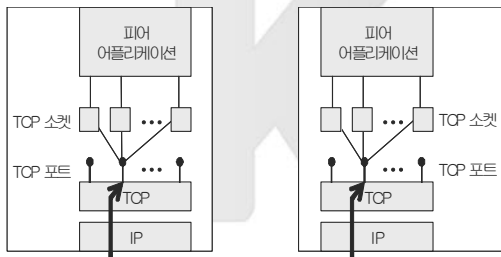


그림 4. 연결 요청 모듈
Fig 4. Connection Request Module

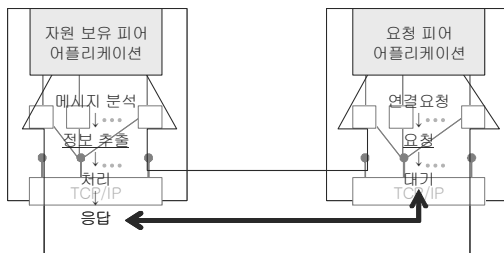


그림 5. 요청 모니터링 모듈
Fig 5. Request Monitoring Module

연결 요청을 수락한 후 연결자의 메시지가 올바른 형태인지 분석하는 과정이 필요하고, 올바른 메시지에 대하여 정보로써 추출하고, 이를 처리할 수 있는 해당 모듈로 보내게 된다. (그림 5)은 요청 피어의 P2P 응용프로그램에서 자원을 보유한 피어에 자원에 대한 다운로드 요청을 하기에 앞서 연결요청 작업에 들어가고, 응답을 기다리게 되며, 해당 자원 보유 피어는 요청 메시지를 분석 및 처리과정을 거쳐 요청 피어에 응답을 하게 되는 구조이다. 이러한 과정에서 자원 보유 피어는 요청을 수용할 수 있는 연결의 최대 개수를 정할 필요가 있다. 하나의 피어에 작업량이 수용 범위를 넘어설 수 있기 때문인데 연결할 수 있는 범위 내에서 연결 수락이 이뤄질 수 있도록 수용 가능 연결 개수를 정하게 된다. 연결할 수 있는 범위를 넘어선 요청에 대해서는 대기 상태를 두는 것보다는 연결 요청에 대한 거부를 하는 방법이 효율적인 것으로 나타난다. 여기서 연결은 의미 있는 데이터가 그 피어에 있음을 알고 있는 상태에서 행해지므로, 대기상태로 처리되는 경우 연결 요청자는 대기리스트에 머물 것인지, 두 번째 후보 피어에게 요청을 할 것인지를 결정 할 필요가 있고, 이때 피어는 기다릴 예상 시간과 대기가 순번을 알려 줄 필요가 있으며, 이러한 과정에서 요청 피어의 목적 달성에 있어 지연의 부담이 매우 크다. 검색 모듈은 자원 요청 피어의 질의를 시작으로 동작되는데, 검색 알고리즘이 실제로 P2P 시스템에 적용될 수 있는 모듈이다. 메시지 전파 과정은 개개의 피어들 간에 자발적으로 검색 과정이 발생하므로 연결되어 있는 피어들을 근간으로 검색 메시지를 보내게 된다. 검색 진행이 시작되면 검색 메시지를 받은 피어에 의해서 자체적으로 자신이 가지고 있는 자원과 검색 메시지와 일치하는 내용이 있는지 확인하는 과정이 일어난다.

이 경우 요청을 받은 피어의 검색 속도만을 측정한다면 로컬에서 일어나는 작업이므로 다른 어떠한 방법보다 빠른 처리가 가능하겠지만, 검색 수행 단계와 동일시기에 연결되어 있는 피어들 간에서만 검색이 일어난다. 하지만 일단 요청 정보와 동일한 자원을 지닌 피어를 발견한 이후에는 더 이상 검색이 이뤄질 필요가 없으므로 검색 범위 측면에서 장점이 있다. 검색의 대상이 되는 피어와 자원 요청 피어는 모두 서버가 존재하지 않는 순수한 P2P 모델을 기반으로 하고 있기 때문에 다른 피어를 찾는 데 있어 로그온 한 사용자에게 동적으로 요청하는 형태로 진행된다.

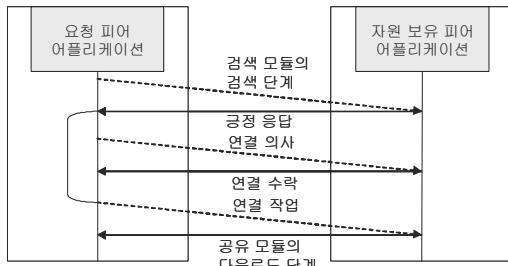


그림 6. 질의 결정 모듈
Fig 6. Query Decision Module

P2P 시스템에서 사용자가 피어 검색 과정을 끝낸 후 해당 피어에게 자원에 대한 요청의 과정은 프로토콜에 따라 다르지만 일반적으로 연결 수립 과정을 거쳐 다운로드 단계로 발전하게 된다. 이렇듯 검색 알고리즘에 따라 여러 가지 방법으로 질의가 전파되고, 그 질의에 해당하는 공정응답이 발생된 피어와 질의 발생 피어 사이에 연결이 수립되어 해당 목표를 달성하게 되는데, (그림 6)에서와 같이 요청을 받은 피어가 해당 질의에 만족하는 자원을 가지고 있을 경우 공정 응답의 과정을 거쳐 연결 수립과정으로 진행된다.

P2P 시스템에서 자원 공유의 방법과 역할이 각 모델별로 구분되는데, 순수한 P2P 모델의 경우 요청 피어는 다른 피어에게 요청 메시지를 보내는 역할에서 끝나고, 이 요청을 받아들인 피어에서 전적으로 자원 공유의 전반 사항을 책임지게 된다. 질의 결정 모듈에서 자원 요청 정보를 확인한 후 부합되는 정보가 있을 경우 그 접속을 열어주고, 공유 모듈에서 자원을 제공하는 역할을 담당하는데, 모두 요청을 받아들인 피어에게 그 역할이 있다. 자원의 공유 정책에 따라 자원 제공 피어의 역할 분담이 이루어질 수 있는데, 동일 자원 보유 피어와의 작업을 분담하는 경우 동일한 자원을 보유한 피어에 대해 사전 정보가 있어야 하지만 분담 정책에 따라 신뢰성 측면의 향상된 성능이 기대된다. 일반적인 단일 피어 전담 형의 경우 공유의 역할이 전적으로 자원 제공 피어에게 일임된다.

IV. 실험 및 성능 평가

이 장에서는 P2P 시뮬레이터를 이용한 일련의 실험들을 통해 제안한 검색 알고리즘의 성능을 평가한다. 제안한 검색

색 알고리즘의 성공적인 검색 연산을 확인하기 위해 관련 알고리즘 분석을 통해 조사된 K Walks, APS, Int-BFS과의 성능 비교를 통해 제안한 검색 알고리즘의 P2P 약결합 환경에서의 적합성을 검증한다. J2SDK 1.4.2 버전의 Java 언어를 기반으로 한 PeerSim 시뮬레이터는 P2P 환경의 검색 알고리즘을 실험하기 위해 설계되었다. 노드의 수가 100만 이상에 달하는 P2P 시스템들이 동작되는 가운데, 네트워크 트래픽의 상당부분을 차지하는 P2P 시스템은 노드의 생성 및 삭제가 빈번히 일어나는 동적인 환경이다. 이러한 P2P 약결합 시스템에서 PeerSim 시뮬레이터는 다양한 검색 성능을 실험해 볼 수 있다. 또한 다양한 운영환경을 지원하여, 본 논문에서는 Window Server 2000 운영체제 하에 J2SDK 1.4.2의 개발 환경을 기반으로 소프트웨어 플랫폼을 구성했고, 시스템 사양은 Intel Pentium III 871MHz, 메인 메모리 256 Mb, 하드디스크 40 Gb 환경에서 실험하였다. 구성된 실험 환경 하에서 실험 대상의 기본 인수들을 설정해야 한다. 시뮬레이션 인수 가운데 P2P 모델은 순수한 P2P 모델, 그래프 모델은 랜덤 그래프 모델, 노드 수는 1000개의 노드, 평균 노드의 차수(degree)는 20, K Walker의 수는 15, TTL은 6, 객체 수는 100, 질의 분포는 Zipf를 기본으로 하고 비교 대상으로 Power-Law 및 Watts-Strogatz 분포를 대상으로 하며, 질의 발생 피어의 수는 1000으로 한다. 그래프 모델로써 랜덤 그래프를 실험 인수로 이용하였는데, 다수의 P2P 관련 검색 알고리즘의 네트워크 토폴로지로서 랜덤 그래프 모델을 선정하여 실험 환경에서 피어 간의 연결 관계를 맺고 있다. 또 질의 분포는 실제 P2P 시스템에서의 분포와 가장 유사하다고 알려져 있는 Zipf 분포를 바탕으로 주요 실험이 이루어졌다.

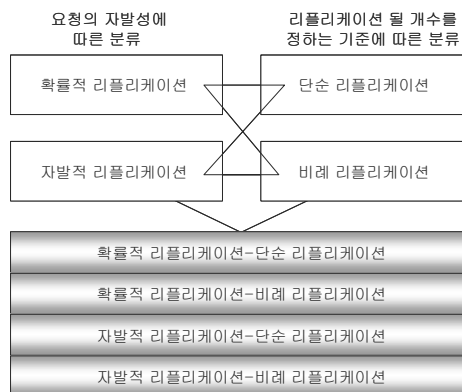


그림 7. 리플리케이션 기법의 확장
Fig 7. Extension of replication techniques

표 1. 리플리케이션 기법에 따른 성공률
Table 1. Success rate according to replication techniques

리플리케이션 기법		평균 성공률 (K=1..15)		
		Random Walks	MLTO	APS
확률적 리플리케이션	비례 리플리케이션	46.6%	81.4%	77.1%
	단순 리플리케이션	40%	74%	74.2%
자발적 리플리케이션	비례 리플리케이션	42.1%	75.3%	73%
	단순 리플리케이션	37%	68.2%	68%

특정 자원에 대한 리플리케이션은 리플리케이션에 대한 요청의 자발성에 따라 확률적 리플리케이션과 자발적 리플리케이션으로 나눌 수 있다. 확률적 리플리케이션은 요청 피어의 자원에 대한 요청이 있을 경우만 그 자원이 리플리케이션의 대상이 되는 것으로, Gnutella가 확률적 리플리케이션 기법을 적용한 대표적인 예이다. 자발적 리플리케이션은 요청 피어의 자원에 대한 요청이 없음에도 불구하고 리플리케이션이 자발적으로 발생하는 정책을 이용하는 것으로, Freenet이 자발적 리플리케이션 기법의 대표적인 예이다.

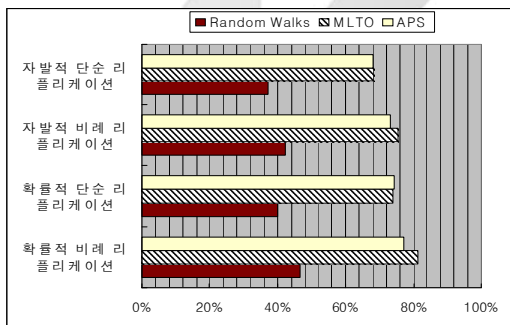


그림 8. 리플리케이션 기법에 따른 성공률
Fig 8. Success rate according to replication techniques

리플리케이션 될 개수를 정하는 기준에 따라 단순 리플리케이션과 비례 리플리케이션으로 나눌 수 있다. 단순 리플리케이션은 모든 자원에 대해 일정한 규칙에 의해 대칭적으로 리플리케이션 하는 기법으로 자원의 질의율 및 기타 다양한 측정 단위의 결과와 상관없이 단순 규칙에 의해 자원의 리플리케이션이 결정된다. 비례 리플리케이션은 단순 리플리케이션과 반대로 자원의 질의율과 같은 결과에 따라서 리플리케이션 비율을 차별적으로 정하는 정책이다. 따라서 질의율이 높은 자원에 대한 P2P 네트워크 내의 분포를 늘려서 분산된 로드와 성공률 관점에서 질의율에 따라 보장 정도를 달리하는 차별적 리플리케이션 정책이 된다.

본 논문에서는 (그림 7)과 같이 요청의 자발성과 리플리케이션 될 개수를 정하는 기준에 따라 분류된 리플리케이션의 기법들을 각각 대응시켜, 제안한 검색 알고리즘과의 조화에 따라 성능의 시너지가 가장 높은 관계를 제시하고자 한다. 즉, 확률적 리플리케이션-단순 리플리케이션, 확률적 리플리케이션-비례 리플리케이션, 자발적 리플리케이션-단순 리플리케이션, 자발적 리플리케이션-비례 리플리케이션 기법을 적용하여 그 적용성에 따른 성능의 변화를 측정해보고자 한다. 요청의 자발성과 리플리케이션 될 개수를 정하는 기준에 따라 분류된 리플리케이션의 기법들을 각각 대응시켜, Random Walks, TO, APS 각각과의 조화에 따른 성공률에 대해 <표 1>에 그 결과값을 표현했다. 표기된 성공률은 K가 1에서 15까지 각각의 값에 대한 성공률의 평균값을 나타내며, 리플리케이션 될 위치는 자원이 있는 피어에서 3홉 내로 한정하였으며, 자발적 리플리케이션 하에 비례 리플리케이션은 그 비례 정도를 2배로 한정하였다. 이와 같은 환경에서 확률적 리플리케이션 하의 비례 리플리케이션 기법을 조화시켰을 때 TO는 가장 좋은 성공률을 보이며, 자발적 리플리케이션 하의 단순 리플리케이션 기법을 조화시킨 Random Walks의 평균 성공률이 가장 낮았다.

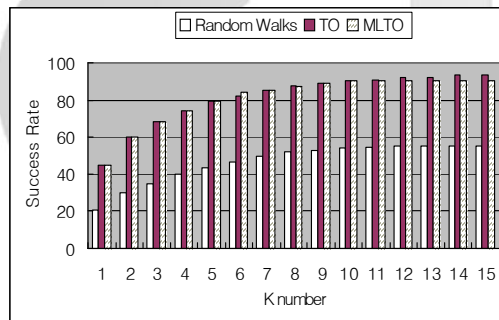


그림 9. MLTO의 성공률 비교
Fig 9. Comparison of MLTO's success rate

MLTO는 LTO의 갱신단계의 변형 구조로써 실험 단계의 증가에 따른 검색 성공률 측면에서 실험한 결과 (그림 9)에서와 같이 TO와 유사한 결과를 보여주고 있다. MLTO와 Random Walks와의 비교 결과는 35-58 정도의 큰 차이를 보이고 있다. 6단계까지 5% 이상의 급격한 상승률을 보이는데, 이는 연관값의 학습 효과에 의한 결과로 판단된다. TO와 유사한 결과를 보이는 MLTO는 K 수가 11이 넘어가면서 소폭 하락하고 있는데, 이는 검색과정의 라우트에 의존하지 않고, 자원을 찾은 이후 갱신단계에서의 독자적인 방식의 갱신에 의한 영향으로 판단된다.

V. 결론

P2P 시스템의 이용이 급증하면서 P2P 어플리케이션에 의해 발생하는 과도한 트래픽은 네트워크 대역폭 문제를 유발하고 있으며, P2P 시스템은 피어들의 P2P 오버레이 네트워크에의 연결성이 매우 약한 약결합 시스템으로 검색 단계에서 원하는 자원을 찾는다 하여도 다운로드 단계에서 반드시 그 피어에서 자원을 다운받을 수 있다는 보장이 없다. 기존 LTO[1]는 이러한 문제를 해결하기 위해 Random Walks 기반의 휴리스틱 피어 선택 기법을 적용하였다. 본 논문에서는 이러한 LTO 기반 하에 각 피어의 컴퓨팅 파워에 영향 받는 노드 분산을 개선을 위한 적응적 P2P 검색 알고리즘을 제안하였다.

제안한 시스템의 성능 평가 결과 검색의 방향성과 통신 발생의 분포 측면에서 적절한 타협점에서 동작함을 확인하였으며, 시뮬레이션 단계의 증가함에 따른 질의 결과의 높은 성과를 보여주었고, 질의율에 따른 차별적 리플리케이션 기법의 적용에 따른 유효성을 검증하였다. 또한 노드 분산율을 높임에 따라 각종 성능 저하 요인에 대한 영향이 거의 없어 본 알고리즘의 유효성을 간접적으로 나타내고 있다.

향후 연구로 P2P 네트워크를 이루고 있는 노드들의 일반적인 환경에 적용적인 이형 분산 시스템[14]기반 검색 알고리즘의 확장 가능성에 대해 적용하고, 급증하는 연구 배경이 되고 있는 유비쿼터스 환경에 적용적인 P2P 검색 알고리즘으로 연구 영역을 넓히고자 한다.

참고문헌

- [1] 김분희, 이준연, "단계별 OK 기법 기반 효과적 P2P 검색 알고리즘", 한국컴퓨터정보학회 논문지, 10권 2호, 2005. 5.
- [2] KT, P2P Traffic Investigation, 2004.
http://news.naver.com/hotissue/daily_read.php?section_id=105&office_id=015&article_id=000071
- [3] Boon-Hee Kim, Young-Chan Kim, "Tops Index Server for Advanced Search Performance of P2P System with a Simple Discovery Server,"

LNCS 3032, Springer-Verlag, pp. 285-291, 2004.

- [4] D. S. Milojevic, et al., "Peer-to-Peer Computing," Technical Report (HPL-2002-57), Hewlett-Packard Company, 2002.
- [5] Napster website: <http://www.napster.com>.
- [6] S. Daswani and A. Fisk, Gnutella UDP extension for scalable searches (GUESS) v0.1
- [7] Gnutella website: <http://gnutella.wego.com>
- [8] P. Felber, E. Biersack, L. Garces-Ercce, K.W. Ross, G. Urvoy-Keller, Data Indexing and Querying in P2P DHT Networks, ICDCS 2004, Tokyo, Japan.
- [9] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," ACM (Supercomputing), June 2002.
- [10] Rüdiger Schollmeier, Gero Schollmeier, "Why Peer-to-Peer (P2P) Does Scale: An Analysis of P2P Traffic Patterns," P2P'02.
- [11] B. Yang, H. Garcia-Molina, "Improving search in peer-to-peer networks," ICDCS'02, pp.103-113, 2002.
- [12] D. Tsoumakos and N. Roussopoulos, "Analysis and Comparison of P2P Search Methods," Technical Report(CT-TR-4451), Maryland Univ., 2003.
- [13] 박종선, 장용철, 오수열, "적응적 중복 객체 알고리즘을 이용한 객체 복제본 관리 연구," 한국컴퓨터정보학회, pp.51-59, 2003.3.
- [14] 장연세, "이형 분산 컴포넌트 플랫폼간 상호 운영성 보장에 대한 연구," 한국컴퓨터정보학회, pp.40-45, 2002.12.
- [15] Y. Lee, et al., "Intelligent Knowledge Discovery in Peer-to-Peer File Sharing," IKM'02, pp. 308-315, 2002.

저 자 소 개



김 분 희

2005년 2월 중앙대학교 컴퓨터공학과 공학박사
2005년~현재 동명정보대학교 멀티미디어공학과 전임강사



이 준 연

2000년 2월 중앙대학교 컴퓨터공학과 공학박사
2000년~현재 동명정보대학교 멀티미디어공학과 조교수