

클러스터 기반 모바일 P2P 알고리즘의 최적 클러스터 구성

Optimal cluster formation in cluster-based mobile P2P algorithm

우혁*, 이동준*

Hyuk Wu*, Dong-Jun Lee*

요 약

최근 모바일 기기의 사용 급증에 따라 모바일 애드혹 네트워크(MANET, Mobile Ad-hoc Networks)에서 사용하는 모바일 P2P(Peer to Peer)에 대한 관심이 높아지고 있으며 그에 대한 연구도 활발히 이루어지고 있다. P2P 방식은 구조적인(Structured) P2P 방식과 비구조적인(Unstructured) P2P 방식으로 분류할 수 있다. 모바일 애드혹 네트워크에서 구조적인 방식의 P2P 알고리즘은 정보의 인접성을 고려하지 않아 컨트롤 트래픽이 증가하는 문제가 있으며, 비구조적인 방식의 P2P 알고리즘은 노드 수가 많아질 경우 확장성에 문제가 있다. 그래서 구조적인 방식의 P2P 알고리즘과 비구조적인 방식의 P2P 알고리즘의 장점을 취합한 하이브리드(Hybrid) 방식의 P2P 알고리즘이 제안 되었다. 본 논문에서는 하이브리드 P2P 방식 중 클러스터 기반(Cluster-based)의 P2P 알고리즘의 컨트롤 트래픽을 분석하고, 이에 관한 최적 클러스터 구성을 도출하였다. 이러한 최적 클러스터 구성 사용시, 비구조적인 방식의 P2P 알고리즘 중 대표적인 Gnutella 보다 더 좋은 성능을 나타냄을 보였다.

Abstract

Mobile P2P(Peer-to-Peer) protocols in MANET(mobile ad-hoc networks) have gained much attention recently. Existing P2P protocols can be categorized into structured and unstructured ones. In MANET, structured P2P protocols show large control traffic because they does not consider the locality of P2P data and unstructured P2P protocols have a scalability problem with respect to the number of nodes. Hybrid P2P protocols combine advantages of the structured and unstructured P2P protocols. Cluster-based P2P protocol is one of the hybrid P2P protocols. Our study makes an analysis of the cluster-based P2P protocol and derives the optimal cluster formation in MANET. In the derived optimal cluster formation, the cluster-based P2P protocol shows better performance than Gnutella protocol with respect to control traffic.

Key words : Cluster-based P2P, Gnutella, MANET, Mobile P2P.

I. 서 론

최근 모바일 기술의 비약적인 발전과 다양한 모바

일 장치가 소개되면서 모바일 네트워크에 대한 관심이 높아지고 있으며 그에 대한 연구도 활발히 이루어지고 있다. 모바일 네트워크 중에서 모바일 애드혹

* 한국항공대학교 항공전자 및 정보통신공학부

(School of Electronics, Telecommunications and Computer Engineering, Korea Aerospace University)

· 제1저자 (First Author) : 우혁

· 투고일자 : 2011년 1월 25일

· 심사(수정)일자 : 2011년 1월 26일 (수정일자 : 2011년 3월 22일)

· 게재일자 : 2011년 4월 30일

네트워크(MANET)는 네트워크 유지를 위한 기반 시설 없이 네트워크가 가능한 단말들이 스스로 네트워크를 구성하는 방식이다. 모바일 애드혹 네트워크를 구성하는 단말들은 자신의 통신 가능 거리 밖의 다른 단말과도 통신이 가능하며, 이는 직접적인 통신이 불가능한 송신 단말과 수신 단말 사이에 중간 단말들이 위치하여 데이터를 전달(Relay)함으로써 가능하다. 모바일 애드혹 네트워크는 기지국(Base station), 라우터(Router), AP(Access Point) 등 별도의 기반 시설 없이 단말들 스스로 네트워크 구성이 가능하며, 네트워크의 토폴로지 변화에 쉽게 대처가 가능한 것이 특징이다.

P2P 네트워크는 비교적 소수의 서버에 기능을 집중하기 보다는 네트워크 구성에 참여하는 단말들에게 기능을 분산하여 구성하는 네트워크이다. P2P 네트워크는 일반적으로 단말들의 규모가 큰 애드혹 네트워크로 연결하는 경우에 이용하며 단말들 간의 연결을 전송 계층 네트워크 상위에 가상의 오버레이(Overlay) 네트워크 기반에서 이루어진다. 오버레이 네트워크는 전송 계층 상위에 존재하는 가상의 네트워크 계층이다. 이 오버레이 네트워크에서 이웃 단말은 물리적인 이웃 단말이 아니라 논리적인 이웃 단말이다. 예를 들면 토폴로지 상에서 이웃하여 있는 단말이 실제로는 다른 대륙에 있는 단말일 수도 있다.

P2P 방식은 크게 구조적(Structured) P2P, 비구조적(Unstructured) P2P, 하이브리드(Hybrid) P2P 방식으로 구분할 수 있다.

구조적인 P2P는 일정한 모양의 토폴로지를 갖고 있고 파일의 배치도 규칙적이기에 검색 비용이 낮은 장점이 있다. 그러나 일정한 토폴로지를 갖기 위해 단말들이 서로에 대한 정보를 교환하여야 한다. 또한 일정한 토폴로지를 형성하기 위한 부가적인 컨트롤 메시지 교환이 필요하며 단말들이 상호의존적이다. 그러므로 네트워크에 새로운 단말이 연결을 하거나 이탈하게 되면, 그것을 복구하기 위한 부가적인 컨트롤 메시지 교환이 발생하게 된다. 또한 검색된 정보의 인접성이 고려되지 않는다. 이는 검색된 자료가 물리적으로 가까이 위치하더라도 오버레이 네트워크 상에서 그렇지 않을 수 있다는 뜻이다. 대표적으로 CAN(Content Addressable Network)[1], Chord[2],

Pastry[3]가 있다.

비구조적인 P2P는 연결이 일정한 토폴로지를 갖고 있지 않으며 단말들 사이의 의존성이 낮아 네트워크에 새로운 단말이 연결을 하거나 이탈하여도 다른 단말들은 큰 영향을 받지 않는다. 또한 서로에 대한 정보를 거의 갖고 있지 않고 순수하게 메시지 플러딩(Flooding)에 의해서 검색을 하기 때문에 검색 비용이 높은 단점이 있다. 무선 환경에서는 동적인 토폴로지 변화에 적응하기 위한 측면으로 볼 때 일정한 구조를 가지지 않는 비구조적인 P2P가 더 적당하다고도 볼 수 있으며 대표적인 알고리즘으로는 Gnutella가 있다 [4].

하이브리드 P2P는 구조적인 P2P 방식과 비구조적인 P2P 방식을 함께 사용하는 방식으로써 슈퍼피어를 이용한 클러스터 기반의 P2P가 대표적이다[5, 6]. 클러스터 기반의 P2P는 전체의 네트워크를 가상의 다수 클러스터로 구분하며, 클러스터 내에서는 구조적인 P2P 방식으로 동작하며, 클러스터 간에는 비구조적인 P2P 방식으로 동작한다. 클러스터는 한 개의 슈퍼피어와 슈퍼피어에 자신의 파일 목록을 등록하는 다수의 서브피어로 구성한다. 슈퍼피어는 같은 클러스터에 존재하는 서브피어의 주소와 파일 목록 등의 정보를 테이블로 관리하며, 서브피어는 자신이 등록되어 있는 슈퍼피어를 통해 네트워크에 참여한 후, 자신의 슈퍼피어에게만 검색을 요청하기 때문에 전체 네트워크 트래픽이 기존 방식보다 적게 발생한다. 클러스터 기반의 P2P 프로토콜에서는 클러스터 구성에 따라 컨트롤 트래픽이 많이 변화하게 되는데, 기존에 이에 대한 연구가 없었다.

본 논문에서는 클러스터 기반 P2P 프로토콜에서 컨트롤 트래픽에 대한 수학적 분석을 하였다. 최적의 클러스터의 개수보다 적은 개수로 클러스터를 구성할 경우, 슈퍼피어까지의 평균 홉 수가 증가하여 라우팅 트래픽이 증가한다. 최적의 클러스터 개수보다 많은 개수로 구성할 경우, 클러스터 크로스오버가 더 많이 발생하여 파일 목록 등록 트래픽이 증가하게 되어, 최적의 클러스터 개수로 구성할 경우보다 더 많은 트래픽이 발생한다. 이를 고려하여 최적의 클러스터 개수를 수학적으로 도출하였다.

본 논문의 구성은 다음과 같다. II장에서는 클러

스터 기반의 P2P 알고리즘에 대하여 소개한다. III장에서는 클러스터 기반의 P2P 알고리즘의 트래픽을 수학적으로 분석하고, IV장에서는 이러한 수학적 분석에 기반한 수치 결과를 보이고, 마지막으로 V장에서는 결론을 맺는다.

II. 클러스터 기반 P2P 알고리즘

클러스터 기반 P2P 알고리즘은 노드들을 클러스터 내에서 슈퍼피어와 서브피어의 두 그룹으로 분류하며, 한 클러스터는 한 개의 슈퍼피어와 다수의 서브피어로 그림 1과 같이 구성한다[6]. 슈퍼피어는 서브피어를 위한 제어 기능을 갖고 있으며, 같은 클러스터 내의 서브피어들에게 중앙 집권적인 서버로써 동작한다. 특정 노드가 파일을 찾고자 할 경우 자신이 속한 클러스터의 슈퍼피어에게 검색 메시지를 전송한다. 슈퍼피어가 검색 하고자 하는 파일에 대한 정보를 갖고 있을 경우에는 해당 파일이 존재하는 노드의 정보를 검색을 요청한 노드에게 알려준다. 만약 자신이 속한 클러스터의 슈퍼피어가 검색 파일에 대한 정보를 갖고 있지 않을 경우에는 다른 클러스터의 슈퍼피어에게 검색 메시지를 전달하여 파일 검색을 하게 된다. 검색 메시지를 전달 받은 클러스터에도 파일이 존재하지 않을 경우, 주변의 또 다른 클러스터에게 검색 메시지를 전달하며, 검색 파일을 찾을 때 까지 이 과정을 반복한다.

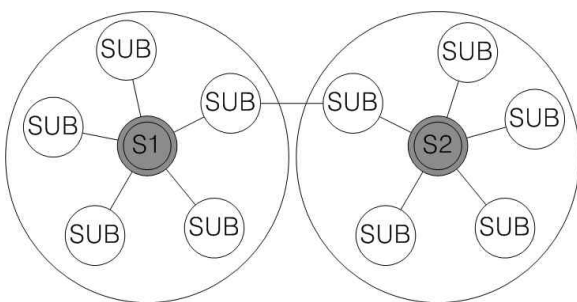


그림 1. 클러스터 기반 P2P 구조.
Fig. 1. The structure of cluster-based P2P.

모든 노드는 일정 주기로 자신의 존재를 알리기 위해 Hello 메시지를 자기 주변으로 전파(broadcast)한다. 그리고 클러스터에 속하기 위해서는 해당 클러스

터의 슈퍼피어에 자신의 파일 목록을 등록하여야 한다. 그리고 자신의 파일 목록을 등록하여도 파일 생성, 삭제로 인하여 파일 목록이 변경될 수 있다. 이럴 경우 변경된 내용을 슈퍼피어에 목록을 업데이트 하게 된다. 특정 노드가 파일 검색을 요청할 경우, 우선 자신이 속한 클러스터의 슈퍼피어에게 검색 요청 메시지를 전송하고, 슈퍼피어가 파일을 찾지 못하였을 경우, 다른 클러스터에 검색 요청 메시지를 전달하여 파일을 찾게 된다.

III. 클러스터 기반 P2P 알고리즘 분석

1. 수학적 분석을 위한 가정

네트워크상에 존재하는 모든 노드는 논리적으로 구분된 가상의 클러스터에 속한다. 각 클러스터의 크기는 동일하며, 각 클러스터 내에는 동일한 수의 노드가 위치한다고 가정한다. 노드의 이동 속도 분포는 최저 0부터 최고 V_{max} 까지 균등 분포(uniform distribution)이며, 네트워크 안에서 노드는 정해진 속도로 일정하게 움직인다고 가정한다.

클러스터 기반 P2P 알고리즘의 분석을 위하여 사용한 주요 파라미터는 표1과 같다.

표 1. 파라미터.
Table 1. parameters.

Symbol	Description
N	전체 노드 수
N_i	한 클러스터의 노드 수
N_c	클러스터의 수
V_{max}	노드의 최대 속도
V_{min}	노드의 최저 속도
V_a	노드의 평균 속도
r_h	Hello 메시지 전송 주기
r_u	파일 변경 사항을 업데이트 주기
r_f	파일 검색 메시지를 전송 주기

S_h	Hello 메시지 크기
S_f	파일 목록 등록 메시지 크기
S_u	파일 목록 업데이트 메시지 크기
S_q	파일 검색 메시지 크기
S_r	라우팅 경로 메시지 크기
P_i	클러스터 내부에 검색하고자 하는 파일이 존재할 확률
H	클러스터 중심에서 클러스터의 경계까지의 홉 수
H_a	클러스터 중심에서 노드들까지의 평균 홉 수
η_c	노드가 다른 클러스터로 이동할 확률
R_r	라우팅 경로 탐색에 의하여 발생하는 라우팅 트래픽 크기

2. 클러스터 기반 P2P 트래픽 분석

본 논문에서는 헬로우 메시지, 파일 목록 등록 메시지, 파일 목록 업데이트 메시지, 파일 검색 메시지를 구분하여 분석하며, 각 경우의 경로 검색(라우팅)이 필요한 경우에 해당 메시지의 경로 검색 트래픽도 분석 한다.

1) 헬로우(Hello) 트래픽

모든 노드는 자신의 주변 노드들에게 주기적으로 Hello 메시지는 전파한다. 그 트래픽의 양은 다음과 같다.

$$C_h = r_h N S_h. \quad (1)$$

2) 파일 목록 등록 트래픽

서버피어가 기존 클러스터에서 새로운 클러스터로 이동한 경우, 그 노드는 자신의 파일 목록을 새로운 클러스터의 슈퍼피어에게 등록해야 한다. 이 파일 목록 등록 트래픽을 계산하기 위하여 우선 노드의 클러스터 이동이 얼마나 자주 발생하는지 클러스터 크로스오버 비율(cluster crossover rate)를 계산해야 한다. 한 클러스터 내에서 서버피어의 클러스터 크로스오버 비율은 다음과 같다[7].

$$\eta_c = \frac{n_s V_{avg} N_i}{\pi l_s} = \frac{V_{avg} \sqrt{N d_n} 4(N_c - \sqrt{N_c})}{N_c^{3/2} \pi}. \quad (2)$$

여기서 dn 은 노드 밀집도(density)이고 l_s 는 클러스터의 한 변의 길이이다. 한 노드의 주변에는 8개의 노드가 있다고 가정한다[8, 9].

클러스터 중앙에 위치하는 슈퍼피어로부터 동일 클러스터에서 가장 멀리 위치하는 노드까지의 최대 홉 수는 다음과 같다[7].

$$H = \frac{-1 + \sqrt{1 + N_i}}{2} \approx \frac{\sqrt{N_i}}{2} = \frac{\sqrt{N}}{2\sqrt{N_c}}. \quad (3)$$

클러스터의 중앙에 위치하는 슈퍼피어와 같은 클러스터에 존재하는 서버피어 간의 평균 홉 수는 다음과 같다.

$$H_a = \sum_{h=1}^H 8h^2 / N_i. \quad (4)$$

서버피어가 슈퍼피어에게 파일 목록을 등록함으로써 인해 발생하는 파일 목록 등록 트래픽은 다음과 같다.

$$C_f = \eta_c H_a S_f N_c. \quad (5)$$

파일 목록 등록을 위하여 서버피어가 슈퍼피어의 경로 획득을 위하여 발생시키는 경로 검색 트래픽은 다음과 같다.

$$C_{rf} = \eta_{lc} R_r N_c. \quad (6)$$

3) 파일 목록 업데이트 트래픽

서버피어가 자신의 파일 목록을 슈퍼피어에게 등록한 이후, 서버피어의 파일 삭제, 생성으로 인하여 파일 목록에 변화가 있을 수 있다. 이 경우 서버피어는 슈퍼피어에게 변경된 목록을 업데이트해야 한다. 이 트래픽의 양은 다음과 같다.

$$C_u = r_u H_a S_u N_i N_c. \quad (7)$$

파일 목록 업데이트를 위하여 서브피어가 슈퍼 피어의 경로를 획득하기 위하여 발생시키는 경로 검색 트래픽은 다음과 같다.

$$C_{ru} = r_u R_T N_i N_c. \quad (8)$$

4) 파일 검색 트래픽

특정 노드가 같은 클러스터에 존재하는 파일을 슈퍼피어에게 검색 요청한 경우, 슈퍼피어는 클러스터내의 모든 파일 정보를 알고 있으므로 클러스터 내의 어떤 서브피어가 파일을 갖고 있는지 별도의 메시지 전송 없이 알 수 있다. 이 경우 파일 검색 트래픽의 양은 다음과 같다.

$$C_{qin} = r_f P_i H_a S_q N. \quad (9)$$

여기서 P_i 는 동일 클러스터에 검색 파일이 존재할 확률이다. 그 P_i 는 다음과 같이 유도할 수 있다.

$$P_i = \frac{N_i}{N} = \frac{1}{N_c}. \quad (10)$$

같은 클러스터의 슈퍼피어로 파일 검색 요청 메시지를 전송하기 위한 경로 획득을 위한 트래픽의 양은 다음과 같다.

$$C_{rqin} = r_f P_i R_T N. \quad (11)$$

동일 클러스터에 파일이 존재하지 않을 경우, 내 클러스터의 슈퍼피어는 주변 클러스터의 슈퍼피어들에게 파일 검색 요청 메시지를 전파한다. 파일 검색 요청 메시지를 수신 받은 다른 클러스터의 슈퍼피어도 파일을 찾지 못하였다면, 또 다른 그 주변의 클러스터의 슈퍼피어에게 전파한다. 검색 파일이 존재하는 클러스터를 찾을 때까지 이 과정을 반복하며, 그 트래픽의 양은 다음과 같다.

$$C_{qout} = r_f (1 - P_i) \{H_a S_q + (N_c - 1) 2HS_q\} N. \quad (12)$$

여기서 $2H$ 는 이웃한 두 클러스터의 슈퍼피어 간의 평균 홉 수이다. 그리고 위 경우의 라우팅 트래픽의 양은 다음과 같다.

$$C_{rqout} = r_f (1 - P_i) R_T N. \quad (13)$$

여기서 클러스터 슈퍼피어 간에는 잦은 메시지 교환으로 인하여 경로가 유지되어 경로 획득을 위한 별도의 라우팅 트래픽이 없다고 가정한다.

파일 검색 요청에 대한 트래픽의 양은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} C_q &= C_{qin} + C_{qout} \\ &= r_f P_i H_a S_q N + r_f (1 - P_i) \{H_a S_q + (N_c - 1) 2HS_q\} N \\ &= r_f [P_i H_a S_q + (1 - P_i) \{H_a S_q + (N_c - 1) 2HS_q\}] N. \end{aligned} \quad (14)$$

위 경우의 라우팅 트래픽의 양은 다음과 같다.

$$\begin{aligned} C_{rq} &= C_{rqin} + C_{rqout} \\ &= r_f \{P_i R_T + (1 - P_i) R_T\} N = r_f R_T N. \end{aligned} \quad (15)$$

5) 전체 트래픽

Hello, 파일 목록 등록, 파일 목록 업데이트, 파일 검색에 대한 P2P 컨트롤 트래픽의 양은 다음과 같다.

$$\begin{aligned} C_p &= C_h + C_f + C_u + C_q \\ &= r_h N S_h + \eta_c H_a S_f N_c + r_u H_a S_u N_i N_c \\ &\quad + r_f [P_i H_a S_q + (1 - P_i) \{H_a S_q + (N_c - 1) 2HS_q\}] N. \end{aligned} \quad (16)$$

P2P 컨트롤 트래픽에 대한 라우팅 트래픽은 파일 목록 등록, 파일 목록 업데이트, 파일 검색 메시지 전송 시에 발생하며 그 경우의 트래픽을 합한 라우팅 전체 트래픽은 다음과 같다.

$$\begin{aligned}
C_r &= C_{rf} + C_{ru} + C_{rq} \\
&= \eta_c R_T N_c + r_u R_T N_i N_c + C_{rq} \\
&\quad + r_f \{P_i R_T + (1 - P_i) R_r\} N. \quad (17)
\end{aligned}$$

P2P 컨트롤 트래픽과 라우팅 트래픽을 더한 전체 트래픽은 다음과 같다.

$$\begin{aligned}
C &= C_p + C_r \\
&= r_h N S_h + \eta_c (H_a S_f + R_T) N_c + r_u (H_a S_u + R_T) N_i N_c \\
&\quad + r_f \left[\begin{array}{l} P_i (H_a S_q + R_T) \\ +(1 - P_i) \{H_a S_q + R_T + (N_c - 1) 2 H S_q\} \end{array} \right] N. \quad (18)
\end{aligned}$$

최적 클러스터 개수는 다음 식으로부터 구할 수 있다.

$$N_c = \arg \min_x C(x). \quad (19)$$

여기서 C는 식(18)에서 트래픽의 양이며, x 는 클러스터의 개수이다. 트래픽의 양이 최소일 때의 x 를 구한다.

3. 라우팅 트래픽 분석

여기서는 식 (17)의 RT를 도출한다. 메시지 전송을 위해서는 목적지까지의 경로 획득 절차가 필요하다. 만약 프로액티브(proactive) 라우팅 알고리즘을 사용한다면, P2P 메시지 전송에 의한 추가적인 라우팅 트래픽은 발생하지 않는다. 리액티브(reactive) 라우팅 알고리즘을 사용한다면, P2P 메시지 전송 시 목적지 노드의 경로를 획득하기 위한 라우팅 컨트롤 메시지에 의한 트래픽이 발생한다. 리액티브 라우팅 알고리즘은 자신의 이웃 노드들에게 메시지를 전파하여 경로를 검색한다. 만약 메시지를 수신 받은 노드가 목적지가 아닐 경우 해당 메시지를 또 다시 전파한다. 이 과정을 목적지를 찾을 때 까지 반복한다. 이웃 노드 수가 8이라고 가정할 경우, 전체 네트워크에서 목적지 노드의 경로를 획득하기 위한 라우팅 트래픽은 다음과 같다.

$$R_T = \sum_{h=0}^{H-1} 8^h S_r = \frac{8^H - 1}{7} S_r. \quad (20)$$

본 논문에서는 리액티브 라우팅을 가정하고 다음 장의 수치 결과를 도출 하겠다.

IV. 수치 결과

주요 수치 분석 환경은 다음과 같다.

표 2. 파라미터 설정값.

Table 2. Parameters values.

symbol	value
N	100~4000
V_{\max}	20 m/sec
V_{\min}	0 m/sec
r_h	0.5 message/sec
r_u	0.01 message/sec
r_f	0.02 message/sec
S_h	20 Byte
S_f	64012 Byte
S_u	140 Byte
S_q	140 Byte
S_r	12+ α Byte

전체 네트워크의 기본적인 노드 수(N)는 100~4000이고 노드의 최저 이동 속도는 0 m/sec이고 최고 이동 속도는 20 m/sec이다. 각 노드의 이동 속도는 균등 분포를 갖는다. 파일 목록 등록 메시지의 크기는 헤더 12바이트이고 각 노드는 파일명 128바이트의 파일 700개를 갖고 있다고 가정하여 총 89612바이트로 설정 한다. 파일 검색 요청 메시지의 크기는 헤더 12바이트이고 검색하고자 하는 파일명 128바이트를 추가하여 기본적으로 140바이트의 크기를 갖는다. 그러나 파일 검색 요청 메시지의 경우 파일을 검색하지

못하였을 경우 다른 노드로 메시지가 재전송 되므로 거쳐 간 노드들의 주소 4바이트 기록하기 위하여 재 전송시마다 검색 요청 메시지에 4바이트씩 추가된다. 위 파라미터들의 기본적인 설정들은 [6]을 기본으로 한다. 트래픽을 최적화하기 위한 변수로 전체네트워크에서 클러스터의 수(Nc)가 있다. 만약 이에 대한 특별한 언급이 없다면 트래픽을 최적화하기 위한 Nc를 적용하였다고 가정한다.

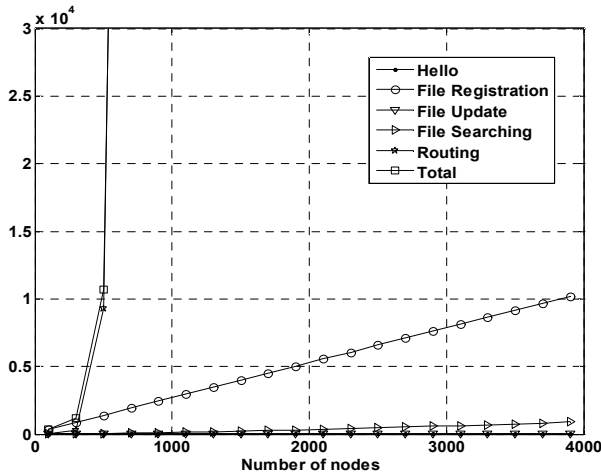


그림 2. 클러스터 기반의 P2P 알고리즘에서 클러스터가 4개일 경우 노드 수에 따른 트래픽.

Fig. 2. The amount of traffic in the cluster-based P2P with respect to the number of nodes when there are 4 clusters.

그림 2는 전체 네트워크가 가상의 클러스터 4개로 구성되어 있을 경우의 노드 수에 따른 트래픽이다. 노드의 수가 증가함에 따라 네트워크의 크기는 증가하게 되며, 각 클러스터의 특정 노드에서 슈퍼피어까지의 평균 홉 수 또한 증가하게 된다. P2P 컨트롤 트래픽은 노드 수에 비례하여 선형적으로 증가하지만, 경로 검색을 위한 라우팅 트래픽은 라우팅 경로가 증가함에 따라 기하급수적으로 증가하게 된다. 그러므로 라우팅 경로 검색을 위한 트래픽을 감소시키고, 트래픽을 최적화시키기 위하여 전체 네트워크를 최적의 가상 클러스터 개수로 구분하여 구성할 필요가 있다.

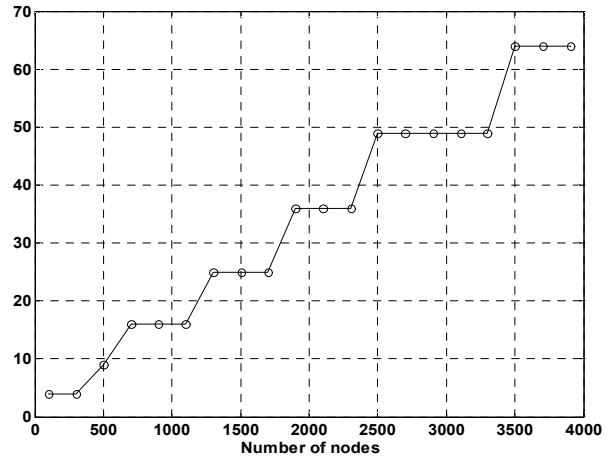


그림 3. 전체네트워크에서 노드 수에 따른 최적의 클러스터 개수.

Fig. 3. Optimal number of clusters with respect to the number of nodes in network.

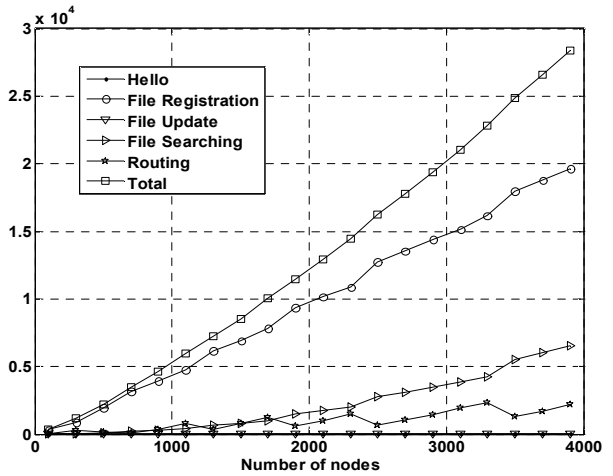


그림 4. 클러스터 기반의 P2P 알고리즘에서 노드 수에 따른 트래픽.

Fig. 4. The amount of traffic with respect to the number of nodes in cluster-based P2P.

그림 3은 노드 수(N)의 변화에 따른 클러스터 기반의 P2P 알고리즘의 최적의 클러스터 개수이다. 식 (19)에서 클러스터 개수를 증가시켜가며 구했다. 전체 네트워크는 정사각형이라고 가정하였고, 전체 네트워크를 가상의 클러스터로 구분할 때 각 클러스터 또한 정사각형이라고 가정하였다. 그래서 클러스터의 개수는 4, 9, 16 등과 같이 제곱수로 증가한다고 가정하였다. 그림 3으로부터 노드의 수가 증가함에 따라 최적의 클러스터의 개수 또한 증가함을 볼 수 있다.

그림 4는 그림 3에서 도출한 최적의 클러스터 개

수를 적용하여 노드 수의 변화에 따른 클러스터 기반의 P2P 알고리즘의 각 메시지별 트래픽을 분석한 것이다. 전체 네트워크의 노드 수에 따라 최적의 클러스터를 적용함으로써 경로 검색을 위한 라우팅 트래픽이 그림 3에 비하여 안정적으로 증가함을 볼 수 있으며, 전체 트래픽에서 차지하는 비중 또한 높지 않음을 볼 수 있다.

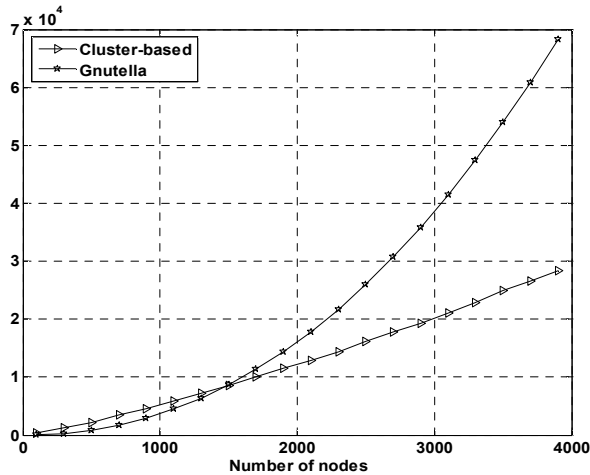


그림 5. 노드 수에 따른 Gnutella와 클러스터 기반 P2P의 트래픽 비교.

Fig. 5. Comparison of the amount of traffic of Gnutella and cluster-based P2P with respect to the number of nodes.

그림 5는 최적의 클러스터 개수를 적용한 클러스터 기반의 P2P 알고리즘과 Gnutella P2P 알고리즘의 트래픽 비교이다. Gnutella P2P 알고리즘은 특별한 등록 절차나 관리 절차 없이 검색 요청이 있을 경우, 전체 네트워크에 검색 요청 메시지를 1홉 이웃노드들에게 전파하며 파일을 찾지 못하였을 경우, 파일을 찾을 때 까지 계속하여 이 과정을 반복하여 전체 네트워크에 검색 요청 메시지를 전파하게 된다. 전체 네트워크의 노드 수가 적을 경우에는 Gnutella P2P 알고리즘이 클러스터 기반의 P2P 알고리즘보다 더 좋은 성능을 보이는 것은 클러스터 기반의 P2P 알고리즘의 Hello, 파일 목록 등록, 파일 목록 업데이트 등의 트래픽이 파일 검색 요청에 의한 트래픽보다 많기 때문이다. 그러나 노드수가 증가함에 따라 Gnutella P2P 알고리즘의 트래픽은 지수적으로 증가하지만, 클러스터 기반의 P2P 알고리즘의 트래픽은 선형적으로

증가한다. 그 결과 노드 수가 약 1500명을 기준으로 하여 클러스터 기반의 P2P 알고리즘이 더 좋은 성능을 나타냄을 볼 수 있고, 노드의 수가 증가할수록 클러스터 기반의 P2P 알고리즘이 더욱 좋은 성능을 나타낸다.

V. 결 론

본 논문에서는 클러스터 기반의 P2P 알고리즘을 수학적으로 분석하였다. 클러스터의 개수가 고정적인 경우에 노드의 수가 증가함에 따라 P2P 트래픽과 비교하여 라우팅 트래픽이 기하급수적으로 증가하는 것을 알 수 있었다. 즉, 노드의 수가 증가함에 따라 클러스터의 개수 또한 증가해야 함을 알 수 있었다. 라우팅 트래픽을 고려한 전체 네트워크의 트래픽을 최소화하기 위하여 노드 수에 따른 최적의 클러스터 개수를 구하였으며, 이를 적용한 클러스터 기반의 P2P 알고리즘과 Gnutella P2P 알고리즘의 트래픽을 비교하였다. 노드 수가 적을 경우 Gnutella P2P 알고리즘이 더 좋은 성능을 나타내었지만, 노드 수가 증가할수록 특정 기점 이후에는 최적화된 클러스터 기반의 P2P 알고리즘이 더 좋은 성능을 나타냄을 알 수 있었다.

최근 스마트폰과 태블릿PC 등 많은 모바일 기기가 사용되고 있고, 이 기기들을 위한 모바일 어플리케이션 또한 많이 개발되었다. 그러나 대부분의 모바일 어플리케이션의 구조가 서버-클라이언트 구조를 갖고 있어 서버에서 많은 부하가 발생하고 있고, 그 부하로 인하여 원활한 서비스가 이뤄지지 않는 경우가 있다. 서버-클라이언트 구조를 본 알고리즘을 적용하여 모바일 P2P 구조로 변경할 경우, 트래픽이 분산되어 더욱 원활한 서비스가 이뤄질 수 있을 것이다.

감사의 글

이 논문은 2007년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2007-0053615).

참 고 문 헌

- [1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network", *Proc. ACM SIGCOMM Conf. 2001, San Diego, CA*, pp. 161-172, 2001.
- [2] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and Balakrishnan, H, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", *Proc. ACM SIGCOMM Conf. 2001, San Diego, CA*, pp. 149-160, 2001.
- [3] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Largescale Peer-to-Peer Systems", *Proc. IFIP/ACM Int'l Conf. on Distr. Systems Platforms (Middleware) 2001*, pp. 329-350, 2001.
- [4] Gnutella Forums, <http://www.gnutelliums.com/>
- [5] J. S. Han, K. J. Lee, J. W. Song and S. B. Yang, "Mobile Peer-to-Peer systems using super peers for Mobile Environments", *Proc. ICOIN*, pp. 1-4, 2008.
- [6] B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network", *Proc. ICDE 2003*.
- [7] D.-J. Lee and J. H. Im, "Optimal cluster-based routing scheme considering node mobility in ad hoc networks", *Wireless Networks(online publish)*, Feb. 2011.
- [8] H Takagi and L Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals", *IEEE Tr. on Comm., COM-32*, pp. 246-257, 1984.
- [9] J. H. Im and D.-J. Lee, "Optimal number of 1-hop neighbor nodes considering routing overhead in ad hoc networks", *Proc. IEEE PIMRC*, 2009.

우 혁 (禹赫)



2007년 2월 : 한국항공대학교
정보통신공학과(공학사)
2009년 8월 : 한국항공대학교
정보통신공학과(공학석사)
2009년 9월 ~ 현재: 한국항공대학교
정보통신공학과 박사과정

관심분야 : MANET, P2P, Sensor network

이 동 준 (李東俊)



1994년 2월 : 한국과학기술원 전기 및
전자공학과(공학사)
1996년 2월 : 한국과학기술원 전기 및
전자공학과(공학석사)
2000년 8월 : 한국과학기술원
전자전산학과(공학박사)

2000년 9월~ 2005년 2월: 삼성전자 책임연구원

2005년 3월~ 현재: 한국항공대학교 부교수

관심분야 : P2P, 무선자원 관리, 무선인지통신시스템