

클러스터링을 이용한 효율적인 라우팅에 관한 연구

양환석*, 최호진*, 이용기**

*조선대학교 전산통계학과

**조선대학교 전산통계학과

e-mail:badhack@shinbiro.com

A Study on Efficient Routing using Clustering for Ad-Hoc Network

Hwan-Seok Yang*, Ho-Jin Choi*, Woong-Ki Lee**

*Dept of Computer Science and Statistics, Graduate School,
Chosun University

**Dept of Computer Science and Statistics, Chosun University

요 약

Ad hoc network는 유선 백본없이 이동 노드들로 구성된 네트워크로서 이동 노드들의 움직임이 빈번히 발생하기 때문에 클러스터간 노드들의 재가입이 계속되므로 네트워크의 안정성을 유지하기 어렵다. 그러므로 최대한 네트워크 토폴로지의 안정성을 유지하는 것이 필요하다. 본 논문에서는 ad hoc network를 구성하는 노드들을 1-level 클러스터로 형성하여 효율적인 라우팅 방법을 제안한다. 그리고 클러스터 리더의 변화를 최소화하기 위해 novel 방법을 사용하고 플리딩을 줄이기 위해 제한된 broadcast 메커니즘을 이용하였다.

1. 서론

Ad hoc network에서 네트워크안에 존재하는 노드들은 제한된 무선 전송 범위 때문에 서로에게 패킷 전송을 도와준다. 어떤 송신 노드로부터 목적 노드까지의 네트워크 경로는 송신자부터 목적지까지의 multihop 경로를 생성하기 위해 패킷을 전달해주는 많은 중간 노드가 필요하다. Ad hoc network는 중앙식 관리나 기지국이나 access point 같은 고정된 네트워크 인프라스트럭처가 필요하지 않고 필요할 때 빠르고 낮은 비용으로 구성할 수 있다. 이러한 특징 때문에 ad hoc network는 긴급 구조 상황, 전쟁 상황, 법 집행 과정, 상업용과 교육용 그리고 유선 컴퓨팅이 불가능한 경우에 이용되고 있다[5].

Ad hoc network에서 라우팅 프로토콜의 역할은 multihop 경로를 중간 노드들이 알 수 있도록 해주는 것이다. 네트워크 안에 있는 노드들은 움직임이 빈번히 발생하고 무선 전송 여건이 수시로 변화하기 때문에 라우팅 프로토콜은 이러한 변화에 대응해야만 하고 연결을 유지하기 위해 새로운 경로를 알아

야만 한다. 한 개 이상의 경로를 형성하는 노드가 다른 곳으로 이동함으로써 해당 경로를 무효화하기 때문에 통신연결이 매우 취약하다[1]. 이동 노드들은 이동에 따른 경로의 계산이나 수정에 대한 많은 시간을 소비하거나 자신의 클러스터 리더와의 연결이 끊겨서는 안 된다. 따라서 잘 정의된 클러스터링 및 라우팅 기법이 필요하다[2]. 노드들의 이동성 때문에 네트워크를 나눈 클러스터들 사이에서 자유롭게 탈퇴하거나 등록함으로써 네트워크의 안정성이 떨어질 수밖에 없다.

현재까지의 클러스터링에 관한 연구에는 노드들 간의 연결성이 가장 좋은 위치의 노드를 클러스터 리더로 선택하는 Highest-Degree Heuristic, 이동속도가 가장 낮은 노드를 클러스터 리더로 선택하는 Node-weight Heuristic 그리고 조합 가중치 클러스터링 알고리즘 등이 있다. 이 알고리즘은 여러 가지 시스템 요소들에 가중치를 주어 가장 가중치가 작은 노드를 클러스터 리더로 선택하는 방법이다.

본 논문에서는 ad hoc network를 구성하는 노드

들은 1-level 클러스터로 형성하여 라우팅하는 방법을 제안하고 네트워크상의 제어 메시지의 플러딩을 줄이기 위하여 제한된 broadcast 매커니즘을 이용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 ad hoc network에서의 클러스터링에 대하여 설명하고, 3장에서는 제안한 클러스터 라우팅 방법에 대하여 설명한다. 4장에서는 제안한 라우팅 방법의 성능을 평가하고, 마지막으로 5장에서는 결론을 맺는다.

2. 관련연구

Ad hoc network를 구성하는 노드들을 클러스터로 분할하고 각각의 클러스터를 책임질 리더를 선정하는 것을 클러스터링이라 한다. 클러스터 리더는 스케줄링 결정, 타임 슬롯과 코드들의 공간적 재사용 강화, 가상 경로의 지원 등의 역할을 수행한다.

2.1 Highest-Degree Heuristic

네트워크를 구성하는 노드들 중에 연결성이 가장 좋은 노드를 클러스터 리더로 선정하는 알고리즘이다. 각각의 노드들은 다른 노드들과의 거리에 근거해 자신의 전송 범위 안에 놓인 노드들의 개수를 계산한다. 이 알고리즘은 클러스터 리더의 업데이트 횟수는 적은 반면에 각 클러스터는 각각의 멤버들 간에 공유되는 자원들을 할당받으므로 노드들의 수가 늘어날수록 클러스터 리더의 처리량이 늘어나게 된다. 따라서 시스템의 성능도 떨어진다.

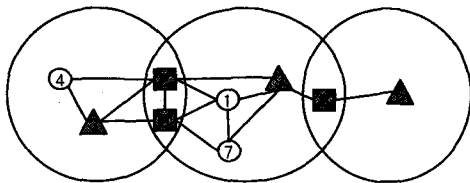


그림 1. Highest-Degree Heuristic

위 그림 1에서 사각형으로 표시된 노드가 클러스터 리더이고, 원은 일반 노드 그리고 삼각형으로 표시된 노드가 게이트웨이이다.

2.2 Lowest-ID Heuristic

모든 노드들이 자신만의 고유한 번호를 부여받고, 그중 가장 작은 노드번호를 가진 노드를 클러스터 리더로 결정하는 알고리즘이다. 만약 클러스터 리더 노드가 클러스터 밖으로 이동을 할 경우 그 다

음 작은 번호를 가진 노드가 클러스터 리더가 되는 간단한 알고리즘이다. 이 알고리즘은 간단하고 계산량이 적은 장점이 있지만 작은 노드번호를 가지고 있는 노드가 클러스터 리더가 될 확률이 높기 때문에 전체 노드간에 부하를 균일하게 배분할 수 없고 노드들에게 부하가 많이 걸려 배터리 소모량이 많다는 단점을 가지고 있다.

2.3 Node-Weighted Heuristic

이동이 활발한 노드는 클러스터 리더로 적합하지 않다. 왜냐하면 그만큼 자주 업데이트가 필요하기 때문이다. 따라서 이동성이 적을수록 클러스터 리더로서 적합할 것이다. 이동성이 적을수록 큰 가중치를 할당하게 된다[3]. 노드 가중치가 가장 높은 노드가 클러스터 리더가 되며 가중치가 같을 때는 더 작은 노드번호를 가진 노드가 클러스터 리더가 된다. 이 알고리즘은 다른 알고리즘에 비해 업데이트 횟수가 비교적 적지만 노드 가중치가 계속 달라지므로 계속해서 주기적으로 가중치 값을 계산해야 하는 단점을 가지고 있다. 따라서 클러스터 리더의 계산 비용이 많이 들고 노드 연결성이나 전력제어와 같은 시스템 파라미터들에 대한 고려가 없는 것이 단점이다[4].

3. 제안한 방법

클러스터링 프로토콜은 동적인 네트워크에서 경로 설정의 강건함을 증가시키기 위해 IP 계층과 라우팅 프로토콜 사이에서 인터페이스 역할을 수행한다. 고정된 주소를 갖는 네트워크에서 경로는 hop-by-hop에 의해 저장된다. 유효한 경로가 끊어진 상태에서 두 개의 노드를 연결하기 위해서는 끊어진 연결에 대한 수정 절차가 필요하게 된다. 반면에 클러스터 네트워크에서는 계층적으로 주소를 부여하고 경로는 클러스터 리더-by-클러스터 리더로 기록이 된다. 그림 2는 이러한 계층적 주소를 보여주고 있다.

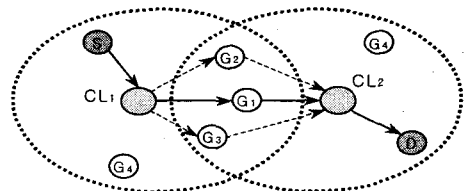


그림 2. 계층적 주소

그림 2에서 소스 노드(S)와 목적지 노드(D) 사이의 hop-by-hop 경로는 다음과 같이 저장된다.

$$S \rightarrow CL_1 \rightarrow G_1 \rightarrow CL_2 \rightarrow D$$

그런데 계층적 주소를 이용한 경로는 다음과 같다.

$$S \rightarrow CL_1 \rightarrow CL_2 \rightarrow D$$

두 가지 방법에서 첫 번째 차이는 경로 CL_1 과 CL_2 에 있다. 만약 CL_1 과 CL_2 의 연결이 끊어진다면 고정된 주소 구조는 수정된 연결이 필요하다. 그러나 계층적 주소는 클러스터 리더 CL_1 과 CL_2 사이의 어떤 게이트웨이 노드가 두 개의 클러스터 사이의 경로를 위해 사용될 수 있다. 그러므로 만약 게이트웨이 G_1 이 클러스터 밖으로 이동한다면, 게이트웨이 G_2 또는 G_3 가 대신 사용된다. 게다가 게이트웨이가 CL_1 과 CL_2 의 범위 밖으로 이동한다면, G_4 또는 G_5 와 같은 두 개의 클러스터를 위한 게이트웨이로 된다. 그 때문에 경로의 강건함은 유지된다.

본 논문에서는 ad hoc network를 구성하는 노드들을 1-level 클러스터 형태로 구성한다. 그리고 클러스터 헤더의 오버헤드를 줄이기 위해 제한된 broadcast 메커니즘을 이용하였다. 클러스터 헤더와 게이트웨이와의 거리는 1 hop 떨어져있고, 두 개의 클러스터는 그들 사이의 여러 개의 게이트웨이를 가질 수 있다. 그리고 노드들은 두 개의 클러스터 사이에서 joint 게이트웨이로서의 역할을 할 수 있다. 또한 각 노드는 다른 클러스터의 구성원이 될 수 있고 두 개의 노드는 직접 통신할 수도 있다. 그림 3은 게이트웨이와 joint 게이트웨이의 차이점을 보여주고 있다.

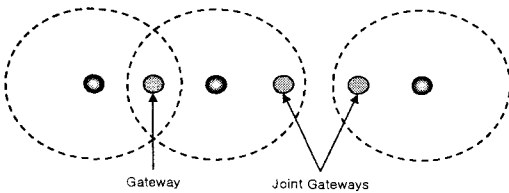


그림 3. 클러스터 게이트웨이

3.1 클러스터링 테이블

노드들은 자신의 상태에 따라서 한 개 또는 두 개의 클러스터링 테이블을 유지한다. 노드 테이블은

다음과 같은 필드로 구성된다.

Node IP Address	Cluster Leaders
-----------------	-----------------

노드가 게이트웨이인 경우에는 클러스터 헤더의 필드는 각 클러스터 헤더의 IP 주소를 포함하고 있으며, 만약 그렇지 않다면 이 필드는 비어있게 된다. 그리고 neighbor 테이블은 다음과 같은 필드로 이루어져 있다.

Cluster Leader IP Address	Gateways
IP Gateway	Flags

Gateway 필드는 클러스터 리더 IP address 필드에서 가리키고 있는 클러스터 리더를 위한 게이트웨이 노드의 리스트를 나타낸다. 그리고 IP 게이트웨이는 IP 라우팅 테이블에 저장된 현재의 노드이다. 만약 클러스터리더가 다른 클러스터리더의 통신 환경이내에 존재한다면 IP 게이트웨이는 클러스터리더 자신이 된다. Flag 필드는 클러스터리더와 이웃하는 클러스터리더와 직접 통신하는지 그렇지 않은지를 나타낸다. 만약 neighbor 테이블 엔트리가 monitor 플래그로 설정되었다면 클러스터리더는 이웃 클러스터의 일부분이 되는지 어떤지를 결정하기 위해 자신의 클러스터를 감시해야만 한다.

노드는 Hello 메시지를 통하여 자신의 존재여부를 알리게 된다. 노드의 상태와 Hello 메시지의 소스의 상태에는 다음과 같은 네 가지 경우가 있다.

- 노드로부터 Hello 메시지를 수신한 클러스터 리더
- 다른 클러스터 헤더로부터 Hello 메시지를 수신한 클러스터 헤더
- 클러스터 헤더로부터 Hello 메시지를 수신한 노드
- 다른 노드로부터 Hello 메시지를 수신한 노드

3.2 제한된 broadcast 메커니즘

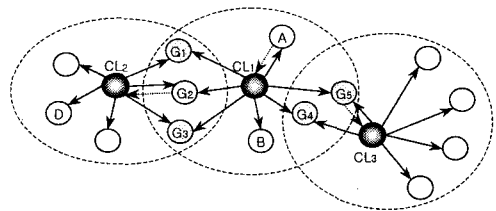


그림 4. 제한된 broadcast

위 그림 4에서 노드 A가 목적지 노드 D에 경로는 찾았다고 가정한다면, 노드 A는 RREQ 메시지를 방

송할 것이다. 클러스터 리더 CL1이 이 패킷을 수신하고 이 패킷을 다른 노드에게로 재방송하게 된다. 이 패킷을 수신한 다른 노드들은 패킷 헤더내에 포함되어 있는 클러스터 리더의 목록을 검사하게 된다. Cluster leader에 의해 재방송된 경우에 이 목록은 비어있다. 만약에 패킷 헤더의 목록에 존재하지 않은 최소한 하나 이상의 클러스터 리더에 도달할 수 있는 패킷을 수신하였다면, 패킷은 버퍼에 저장되고 random timer가 설정된다. 이 시간 동안에 다른 노드에 의해 재방송된 패킷을 주시하게 된다. 만약 자신이 가지고 있는 클러스터 리더가 패킷 헤더의 목록에 있다면, 클러스터 리더는 재방송된 패킷을 수신한 것으로 표시한다. 노드의 타이머가 끝나면 각 클러스터 리더가 이미 패킷을 수신하였는지를 결정하기 위해 재전송 리스트를 검사하게 된다. 여기서 게이트웨이는 패킷을 직접 처리하지 않고 단지 패킷에 재전송이 필요한지 그렇지 않은지 결정을 하게된다. 이와 같은 방법으로 패킷을 방송하게 되면 경로 설정의 강건함과 네트워크 상의 제어 메시지의 수를 상당히 줄일 수 있게 된다.

4. 실험 및 결과

이 장에서는 본 논문에서 제안한 1-level 클러스터를 이용한 라우팅에 대해 ns-2 시뮬레이터를 이용하여 성능을 평가하였다. 성능 평가의 목표는 경로의 효율성과 클러스터당 플러딩 패킷의 수의 성능 평가이다.

40개의 노드가 300m×300m 크기의 지역 내에서 랜덤하게 지역을 이동하게 하였다. 그리고 클러스터는 1초마다 재생성 되며 각 시뮬레이션에는 100초의 시간이 주어졌다.

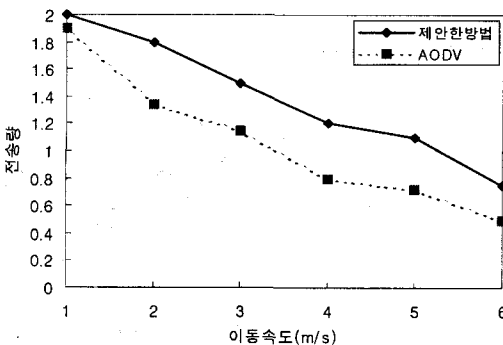


그림 5. 노드가 10개인 경우 패킷 전송량

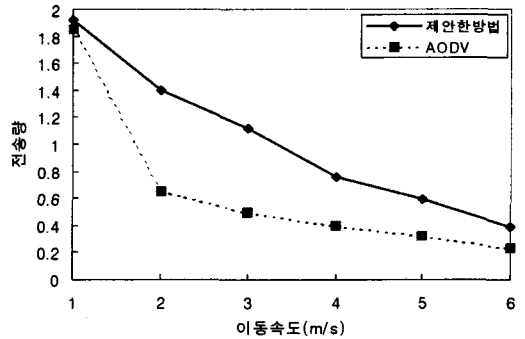


그림 6. 노드가 40개인 경우 패킷 전송량

5. 결론

본 논문에서는 기존의 클러스터링 알고리즘과는 달리 ad hoc network를 구성하는 노드들을 1-level 클러스터로 구성을 하여 라우팅 하는 방법을 제안하였다. 그리고 클러스터헤더의 변화를 최소화하기 위해 novel 방법을 사용하였으며 제한된 broadcast 메커니즘을 이용하여 네트워크 전역의 제어 메시지의 플러딩을 줄일 수 있었다.

향후 연구로는 네트워크의 변화되는 형태에 따라서 계층의 깊이를 동적으로 적응시킬 수 있는 방법에 대해 연구되어야 할 것이다.

참고문헌

[1] J. Broch et. al., "A performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in proc. ACM Mobicom98, Dallas, TX, October 1998.
 [2] Mainak Chatterjee, Sajal K. Das and Damla Trugut, "An On-Demand Weighted Clustering Algorithm(WCA) for ad hoc networks," IEEE GLOBECOM. pp. 1697-1701, 2000.
 [3] S. Basagni, I. Chlamtac, and A. Farago, "A Generalized Clustering Algorithm for Peer-to-Peer Networks," Workshop on Algorithmic Aspects of Communication ICALP Bologna, Italy, July 1997.
 [4] S. Basagni, "Distributed Clustering for Ad Hoc Networks," International Symposium on Parallel Architectures, Algorithms and Networks, pp. 310-315, Perth, June 1999.
 [5] C. K. Toh, "Ad Hoc Network. Wireless Networks," Protocols and System, Prentice Hall PTR, 2002.