

# 모바일 애드 혹 네트워크에서 이동성을 고려한 안정적인 클러스터링 기법

준회원 김혁수, 정회원 황준호, 종신회원 유명식\*

## A Stable Clustering Scheme Based on Node Mobility for Mobile Ad Hoc Networks

Hyeksu Kim Associate Member, Junho Hwang Regular Member, Myungsik Yoo\*<sup>o</sup> Lifelong Member

### 요약

모바일 애드 혹 네트워크는 기존의 유선망과 같은 고정적인 네트워크 인프라 없이 이동 노드들만으로 구성된 통신망으로써 노드의 잦은 이동으로 인하여 변화하는 네트워크 토폴로지를 안정적으로 유지하는 것이 매우 어렵다. 이에 본 논문에서는 노드의 이동성을 고려한 안정적인 클러스터 헤드 선정을 통해 클러스터 내 토폴로지를 유지, 관리하는 클러스터링 기법을 제안한다. 제안하는 방법은 모든 노드가 자신의 주변 노드 수 변화를 통해 이동성을 측정하여 가장 낮은 이동성을 갖는 노드를 클러스터 헤드로 선정하며, 선정된 클러스터 헤드는 자신의 클러스터 이동성을 측정하고 이를 통해 클러스터 별 Hello 메시지 전송 주기를 적응적으로 조절한다. 제안하는 방법의 성능 평가를 위해 모의실험을 수행하였으며, 그 결과 기존 방법들보다 클러스터 헤드의 재선정 횟수와 네트워크 부하를 감소시키고 네트워크 토폴로지를 안정적으로 유지시킴을 입증하였다.

**Key Words :** Mobile ad-hoc network; Cluster; Node mobility; Hello message period.

### ABSTRACT

A mobile ad-hoc network is an autonomous collection of wireless mobile nodes that organizes a temporary network without any network infrastructure. Due to node mobility, it is a challenging task to maintain the network topology. In this paper, we propose a stable clustering algorithm that uses node mobility for cluster formation. In the proposed algorithm, the node mobility is measured by counting the number of nodes entering into/ leaving from its transmission range. The node having the lowest mobility is selected as a cluster head. For topology maintenance with reduced control overhead, the cluster head adaptively controls the broadcasting period of hello message to the measured node mobility. Through computer simulations, it is verified that the proposed algorithm outperforms previous clustering algorithms in terms of control overhead, the rate of node mobility changes and the number of cluster head changes.

### I. 서론

모바일 애드 혹 네트워크(Mobile Ad Hoc Network

: MANET)는 기반 네트워크 시설 없이 이동 노드 (e.g. 핸드폰, 노트북, PDA)들만으로 임시적인 통신망 구성이 가능하기 때문에 기존 네트워크 인프라를

※ 본 연구는 지식경제 프론티어기술개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅 및 네트워크 원천기반기술개발사업의 09C1-C1-20S 과제에 지원된 것임.

\* 숭실대학교 정보통신공학부 통신망설계 및 분석연구실(° : 교신저자)

논문번호 : #KICS2008-09-423, 접수일자 : 2008년 09월 29일, 최종논문접수일자 : 2009년 4월 22일

설치할 수 없는 지역에 유용하며 전시, 긴급 상황, 교육, 회의 등과 같은 다양한 분야로 응용되어 사용될 수 있다<sup>11)</sup>. 그러나 MANET에서는 노드들의 잦은 이동으로 인한 네트워크 토폴로지(topology)의 변화와 가용 대역폭의 제약 그리고 노드들의 한정적인 전력량 때문에 빈번한 제어 메시지 및 정보의 신뢰적인 전달이 어렵다. 이러한 한계점을 극복하기 위해 MANET에서는 이동 노드들을 그룹 단위로 나누어 관리함으로써 네트워크 자원을 효율적으로 사용할 수 있고 이동성 관리가 용이하며 제어 메시지의 오버헤드(overhead)를 줄일 수 있는 클러스터(cluster) 적용 기법이 활발히 연구되고 있다<sup>2)3)4)</sup>.

그림 1은 일반적인 MANET 환경에서의 클러스터 구조를 도시하고 있다. 그림에서 볼 수 있듯이 클러스터는 클러스터 멤버 노드(A, B, E, G, J)를 관리하는 클러스터 헤드(C, F, I)와 두 개 이상의 클러스터와 접하고 있는 게이트웨이 노드(D, H)로 구성된다. 이와 같은 MANET에서의 클러스터 구성은 클러스터 헤드와 클러스터 멤버 노드들 간의 최대 통신 가능 경로 길이에 따라 1-홉 클러스터와 k-홉 클러스터로 구분된다<sup>5)6)</sup>.

먼저, k-홉 클러스터의 경우 다중 홉 환경 구축이 가능하기 때문에 네트워크 확장이 매우 용이하다. 하지만 넓은 범위의 네트워크를 유지, 관리하기 위한 제어 메시지의 증가가 불가피하기 때문에 오버헤드, 자원 낭비 등의 문제를 야기한다. 반면, 1-홉 클러스터의 경우 클러스터 헤드와 멤버 노드간의 직접적인 통신이 가능하여, 소규모 통신에 효율적이며 클러스터 구성 및 유지에 안정적이다<sup>7)8)</sup>.

그러나 1-홉 클러스터의 경우에도 클러스터 헤드의 이동에 의한 클러스터 헤드의 재선정 과정이 빈

번하게 발생되고 이로 인한 클러스터 헤드의 선출 지연이 발생하게 된다. 또한 이와 같이 클러스터가 재구성 되어야 함에 따라 토폴로지가 불안정해짐으로써 네트워크 성능을 저하시키는 요인으로 작용한다.

이에 본 논문에서는 1-홉 클러스터를 고려한 무선 애드 혹 네트워크에서 노드의 이동으로 인해 발생하는 잦은 토폴로지 변화의 문제점을 해결하고자 한다. 이를 위해 주변 노드의 입/출입 횟수에 따라 노드의 이동성을 판단하여 가장 안정적인 노드를 클러스터 헤드로 선정하는 클러스터 형성 기법과 클러스터 별 이동성에 따라 Hello 메시지 주기(Hello message Period: HP)를 적응적으로 조절하는 유지 기법을 적용한 클러스터링 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 클러스터 헤드 선정 및 유지 기법에 대해 설명하고, 이어 3장에서는 본 논문에서 제안하는 클러스터링 기법을 설명한다. 4장에서는 본 논문에서 제안하는 기법의 성능 평가를 수행하고, 마지막으로 5장에서 결론을 맺는다.

## II. 관련 연구

클러스터는 계층적으로 하나의 노드가 다수의 노드를 관리하는 구조를 가지고 있다. 따라서 클러스터를 구성함에 있어 클러스터 헤드 선정과 선정된 클러스터 헤드를 통해 클러스터를 유지할 수 있는 방법이 가장 중요한 기술적 요소이다.

### 2.1 클러스터 헤드 선정 기법

현재 MANET을 위한 1-홉 클러스터 헤드 선정 기법에는 LID(Lowest ID) 기법<sup>9)</sup>, HD(Highest Degree) 기법<sup>10)</sup> 등이 있다. 먼저, LID 기법은 노드의 고유한 ID를 사용하여 클러스터 헤드를 선출하는 특징을 가지고 있어 주소 기반 클러스터링 기법이라고 불린다. LID에서는 클러스터 헤드를 선정하기 위해 각 노드가 자신의 ID를 주기적으로 방송(broadcast)하며 가장 낮은 ID의 노드가 클러스터 헤드로 선정된다. 따라서 클러스터 헤드는 자신보다 높은 ID를 갖는 노드들을 자신의 클러스터 멤버로서 관리하게 되고 자신보다 낮은 ID를 갖는 노드의 가입을 인지하게 되면 자신의 헤드 역할을 위임하여 가장 낮은 ID를 가진 노드가 항상 클러스터 헤드의 역할을 수행할 수 있도록 한다.

한편, HD 기법은 노드의 연결 상태를 고려하여 클러스터 헤드를 선출하기 때문에 연결 기반 클러

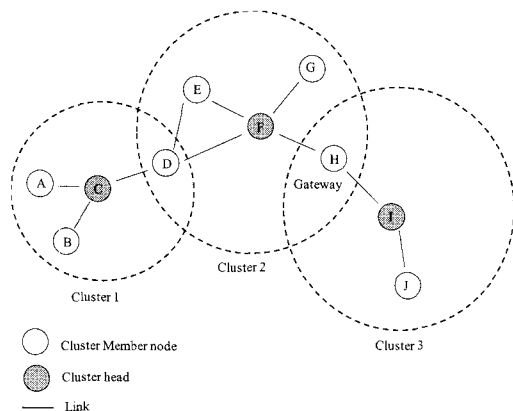


그림 1. MANET 환경에서의 클러스터 구조

스터링 기법으로 불린다. 각 노드는 자신의 이웃 노드 정보를 동일한 주기로 브로드캐스팅하며 가장 많은 이웃 노드(밀도)를 가지는 노드가 클러스터 헤드가 된다. 이후 클러스터로 유입되는 다른 노드의 밀도가 현재 클러스터 헤드의 밀도보다 클 경우 해당 노드가 새로운 클러스터 헤드로 선정된다.

이와 같은 기존의 클러스터 헤드 선정 기법들은 노드의 이동성을 고려하지 않고 클러스터 헤드를 선정하기 때문에 클러스터 헤드의 빈번한 교체로 인해 클러스터의 안정성이 저하된다. 따라서 클러스터를 형성함에 있어 노드의 이동성을 고려하여 이동성이 적은 노드를 통해 안정적으로 클러스터를 형성할 수 있는 방법이 요구된다.

## 2.2 클러스터 유지 기법

클러스터 유지 방안으로 기존의 클러스터 기법들은 주변 노드의 인지를 위한 Hello 메시지를 고정적인 주기로 설정하여 사용하였다. 그러나 이는 클러스터 헤드가 클러스터 멤버 노드들의 이동에 유기적으로 대처하지 못하게 되며, 이로 인해 전체 네트워크 토폴로지 유지 및 네트워크 운용에 불필요한 자원이 소모되는 결과를 초래한다. 예를 들어 도서관과 같이 노드의 이동성이 적은 장소에서는 네트워크 토폴로지가 안정적임에도 불구하고 잦은 Hello 메시지 전송으로 인해 불필요한 네트워크 오버헤드가 발생되며, 백화점과 같이 노드의 이동이 빈번한 장소에서는 잦은 네트워크 토폴로지 변화에 유기적이지 못한 Hello 메시지 주기 설정으로 인해 전체 네트워크 토폴로지를 안정적으로 유지하지 못하는 문제점이 발생된다.

이와 같은 기존 클러스터링 기법의 문제점을 해결하기 위해 제안된 LIDAR(Lowest-ID with Adaptive ID Reassignment)기법<sup>[11]</sup>은 LID와 동일한 방법으로 초기 클러스터 구성에 필요한 헤드를 선정하고 이후 노드들의 이동성을 고려하여 클러스터 헤드를 재선정한다. 이때, 선정된 클러스터 헤드는 각 클러스터 멤버 노드들로부터 측정된 이동성을 산술 평균화하여 자신의 관리하는 클러스터의 이동성이라 판단하고, 이에 따라 Hello 메시지 주기를 조절한다.

그러나 각 클러스터 멤버의 이동성을 산술 평균화하여 클러스터의 이동성을 판단하는 것은 자칫 잘못된 클러스터 이동성 정보를 제공할 수 있다. 즉, 각 클러스터 멤버는 자신의 이동성 계산을 위해 자신의 통신 영역 내의 주변 노드 입출입 상태를 파악한다. 따라서 해당 클러스터 멤버가 클러스터 헤드의 통신 영

역 외곽에 존재할 경우 클러스터 헤드가 관리하는 통신 영역 밖에 노드의 이동성이 반영되게 되는 것이다. 이러한 클러스터 멤버의 이동성 측정 방법으로 인해 결과적으로 클러스터 헤드는 자신의 통신 영역 밖에 있는 노드의 이동성 영향이 반영되게 되는 것이다. 따라서 클러스터의 이동성 측정에 있어 외부 노드의 영향을 최소화하여 클러스터의 이동성을 판단할 수 있는 방법이 필요하며, 더불어 측정된 이동성 정보를 토대로 적응적으로 Hello 메시지 전송 주기를 조절할 수 있는 방법이 필요하다.

## III. 이동성을 고려한 안정적인 클러스터링 기법

MANET 환경은 노드의 이동성으로 인해 빈번한 토폴로지 변화가 발생하고, 이는 클러스터 형성에 있어서도 빈번한 헤드 선정 및 잦은 토폴로지 변화의 문제점을 야기한다. 따라서 클러스터 형성에 있어서 가장 안정적인 노드의 선정과 안정적인 토폴로지 유지가 가능하면서 오버헤드는 감소시킬 수 있는 방법이 요구된다. 이를 위해 본 논문에서는 다음과 같이 노드의 이동성 측정, 클러스터 헤드 선정과 클러스터 유지로 구성된 클러스터링 기법을 제안한다.

### 3.1 노드의 이동성 측정

일반적으로 MANET 환경에서 노드의 이동성은 GPS를 통해 측정된 노드의 이동 속도를 의미한다. 그러나 모든 이동 단말들이 GPS 측정을 위한 하드웨어적인 요구사항을 만족하기 어려우며<sup>[12]</sup>, 단순히 노드의 이동 속도만을 측정하는 것은 노드들과의 링크 상태를 직접적으로 반영하지 못하므로 클러스터 헤드 선정을 위한 기준 지표로 적용하기에 문제점을 야기한다. 따라서 본 논문에서는 노드의 이동성을 단순히 노드의 이동속도로 판단하는 것이 아니라 주변 노드들과의 연결 상태를 고려하는 이동성 측정 방법을 사용한다.

본 논문에서 정의하는 노드의 이동성은 주변 노드의 통신 영역 가입 노드와 탈퇴 노드의 수로 산출된다. 즉, 가입 노드는 임의의 이동 노드 자신의 전송범위 안으로 새로운 노드가 유입됨으로써 자신의 주변 노드로 판단되는 노드 수를 의미하고, 탈퇴 노드의 경우 기존에 자신의 주변 노드였으나 통신 영역 밖으로 이동함으로써 자신과의 링크 상태가 끊기는 노드 그리고 신호의 감쇄 등으로 인해 정상적인 신호를 수신 받지 못하는 노드를 포함한다. 이

를 토대로 해당 노드는 자신의 주변 노드의 변화 수가 많은 경우 이동성이 높다고 판단하고, 주변 노드의 변화 수가 적은 경우 이동성이 낮다고 판단한다.

이러한 이동성을 측정하기 위해서 모든 노드들은 자신만의 고유 ID를 가지고 있으며, 고유한 ID를 포함한 Hello 메시지를 방송한다. 따라서 임의의 노드는 수신한 Hello 메시지 내 고유 ID의 비교를 통해 자신의 주변 노드의 수를 측정할 수 있다.

먼저, 임의의 노드는 자신의 주변 노드들의 고유 ID 목록을 주변 노드 테이블(Neighbor Node Table : NNT)에 저장하고, 매 Hello 메시지 전송 주기마다 수신되는 Hello 메시지 내 ID에 따라 테이블을 갱신하여 해당 테이블 내 ID 정보가 변경되는 횟수를 토대로 자신의 이동성을 판단한다. 위와 같은 과정을 수행하기 위한 노드의 이동성 측정은 식 1과 같다.

$$M^i(t) = \frac{N_{IN}^i(t) + N_{OUT}^i(t)}{N_D^i(t-1)} \quad (1)$$

이때,  $M^i(t)$ 는 현재 시간  $t$ 에서 임의의 노드  $i$ 의 이동성을 나타내고,  $N_{IN}^i(t)$ 는 노드  $i$ 에 새로 가입하는 노드 수,  $N_{OUT}^i(t)$ 는 노드  $i$ 에서 탈퇴하는 노드 수를 의미한다. 또한  $N_D^i(t-1)$ 는 현재 시간  $t$ 의 바로 전 주기인 과거  $(t-1)$ 시간에서 노드  $i$ 의 총 주변 노드 수를 의미한다.

### 3.2 클러스터 형성

본 논문에서 제안하는 클러스터링 기법은 각 노드가 자신의 주변 노드의 변화를 토대로 자신의 이동성을 판단하고, 주변 노드와의 이동성 비교를 통하여 가장 낮은 이동성을 갖는 노드가 클러스터 헤드로 선정되는 특징을 가지고 있다. 이를 위해 본 논문에서는 Hello 메시지 전송 단계, 노드의 이동성 판단 단계, 클러스터 헤드 선정 단계를 포함한 클러스터 형성 기법을 사용한다.

#### 3.2.1 Hello 메시지 전송 단계

모든 노드는 이동성을 판단하고 일정 주기로 그림 2와 같은 형태의 Hello 메시지를 방송한다. 주기적으로 방송되는 Hello 메시지 내에는 MH\_ID, CH\_ID, K, 그리고 HP 필드가 각각 8bit로 구성되는데, 이들은 각각 자신의 고유 ID, 자신이 속한 클러스터 헤드의 ID, 자신의 이동성 측정값 그리고 Hello 메시지 주기를 의미한다.

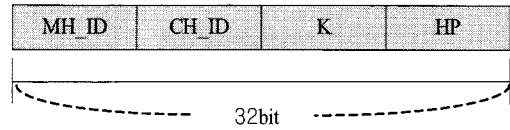


그림 2. Hello 메시지 포맷

#### 3.2.2 노드의 이동성 판단 단계

각 노드는 자신의 이동성을 Hello 메시지 내 K 필드에 저장하여 Hello 메시지 주기마다 방송한다. 이때, Hello 메시지의 K 필드에는 각 노드가 식 1의 방법을 이용하여 Hello 메시지 주기마다 자신의 이동성을 측정하고 이를 식 2와 같이 가중치 평균  $M_{WA}^i(t)$ 으로 계산하여 기입한다.

$$M_{WA}^i(t) = \alpha \times M^i(t) + (1 - \alpha) \times M_{WA}^i(t-1) \quad (2)$$

이때,  $M_{WA}^i(t)$ 은 현재  $t$ 시간에서 노드  $i$ 의 가중치 평균 이동성을 나타내며,  $\alpha$ 값은 임의의 변수로  $0 < \alpha < 1$ 의 값을 갖는다.

#### 3.2.3 클러스터 헤드 선정 단계

클러스터 헤드 선정 단계는 다음과 같은 순서에 따라 결정된다.

1. 모든 노드는 Hello 메시지를 최초 설정된 Hello 메시지 주기마다 방송한다.
2. 각 노드는 모든 주변 노드들의 ID를 테이블에 저장하고 이전 테이블 정보와의 비교를 통해 K값을 계산한다.
3. 자신의 K값과 모든 주변 노드들의 K값을 비교하여 클러스터 헤드와 멤버 노드를 선정한다.
  - if 자신의 K값 > 주변 노드들의 K값 중 최소 then 해당 노드 = 클러스터 멤버
  - if 자신의 K값 < 주변 노드들의 K값 중 최소 then 해당 노드 = 클러스터 헤드
  - if 자신의 K값 = 주변 노드들의 K값 중 최소 then 가장 낮은 ID 노드 = 클러스터 헤드

이와 같은 클러스터 헤드 선정 단계를 보다 자세히 설명하기 위해 그림 3과 같이 10개의 노드로 구성된 MNAET 환경이 존재하고, 노드 5번과 3번이 이동하는 상황에서 두 주기 동안 각 노드의 이동성은 측정한다고 가정하자. 이때 각 노드의 이동성 측정값(K)은 수식 1, 2를 통해 표 1과 같이 계산된다.

표 1. 노드의 이동에 따른 K값 계산 예

| MH_ID | K    | MH_ID | K    |
|-------|------|-------|------|
| 1     | 0.2  | 6     | 0    |
| 2     | 0.25 | 7     | 0.4  |
| 3     | 0.5  | 8     | 0    |
| 4     | 0    | 9     | 1    |
| 5     | 2.5  | 10    | 0.33 |

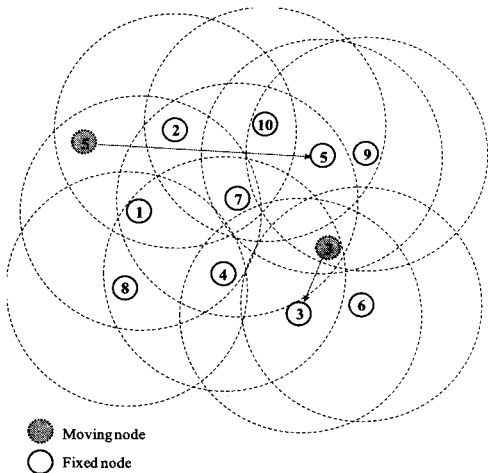


그림 3. 클러스터 형성 전 노드의 이동

표 1과 같은 이동성 측정값을 토대로 클러스터 형성 과정의 예를 살펴보면 그림 4와 같다.

먼저, 그림 4의 (a)는 시간  $t$ 에서 클러스터 형성 과정을 나타내고 있다. 그림에서 볼 수 있듯이 4번( $K=0$ )과 6번( $K=0$ )은 주변 노드의 이동성을 표 1을 토대로 판단하여 자신이 주변 노드보다 낮은 이동성을 가지고 있기 때문에 클러스터 헤드의 역할을 수행해야 함을 인지한다. 이때 8번 노드는 4번 노드의 주변 노드 이면서 동일한  $K$ 값을 가지고 있지만 높은 노드 ID로 인해 클러스터 헤드가 될 수 없다.

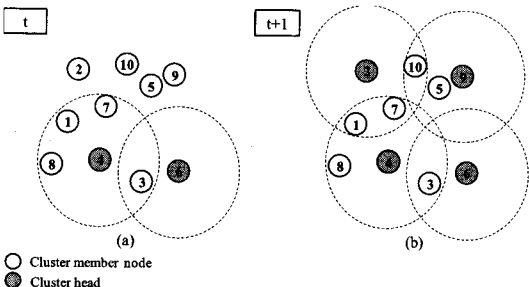


그림 4. 클러스터 형성 과정의 예

이와 더불어 2번 노드는 자신의 주변에 1, 7, 10번 노드가 있으며, 이중 1번 노드가 2번 노드보다 낮은 이동성 값을 가지고 있으나 1번 노드는 이미 4번 노드를 클러스터 헤드로 인식하고 있기 때문에 더 이상의 클러스터 헤드는 형성되지 않는다. 이후 그림 4의 (b)에서와 같이 시간  $t+1$ 에서는 다시 주변 노드의 이동성 정보 갱신을 통해 2번과 9번이 클러스터 헤드의 역할을 수행해야함을 인지한다. 이때, 1, 7, 10번 노드는 두 개의 클러스터 헤드와 접하고 있기 때문에 게이트웨이 노드로서 동작할 뿐만 아니라, 하나의 클러스터 멤버노드로서 일반적인 동작을 수행한다. 이때, 게이트웨이 노드는 두 개의 클러스터 헤드로부터 Hello 메시지를 전송 받지만 그중 이동성이 낮은 클러스터 헤드를 선택하여 일반 클러스터 멤버와 마찬가지로 하나의 클러스터 멤버로 동작한다.

이와 같은 클러스터 형성은 클러스터 헤드의 손실이나 이동성의 변화로 인해 클러스터 헤드의 재선정이 일어날 경우에도 최초 클러스터 헤드를 선정하는 방법과 동일한 과정을 수행하게 된다. 클러스터 재선정 과정을 자세히 살펴보면 다음과 같다.

예를 들어 그림 4의 (b)와 같은 상황에서 4번 노드가 이동 또는 전력 소모로 인하여 더 이상의 클러스터 헤드의 역할을 수행하지 못할 경우 4번 노드의 클러스터 멤버인 1번 노드와 7번 노드는 다음 Hello 메시지 수신 주기에 4번 노드로부터 두 번 이상 Hello 메시지를 수신 받지 못하면 클러스터 헤드가 손실되었다고 판단하여 주변 클러스터 헤드인 2번 노드의 클러스터 멤버로 동작한다.

마찬가지로 3번 노드의 경우도 동일한 과정을 통해 6번 노드의 클러스터 멤버로 동작한다. 이때, 새로이 편입되는 클러스터 멤버 노드의 Hello 메시지 주기는 클러스터 헤드의 전송 주기를 따른다. 그러나 8번 노드의 경우 자신의 통신 영역 내에 클러스터 헤드가 없기 때문에 자신의 Hello 메시지 주기를 리셋(reset)하고 자신이 클러스터 헤드의 역할을 수행한다.

표 2는 그림 4의 클러스터 형성 과정에 있어 시간의 흐름에 따른 각 이동 노드의 Hello 메시지 내 CH\_ID의 변화를 정리한 표이다.

### 3.3 클러스터 유지

앞서의 클러스터 형성 방법을 통해 안정적으로 클러스터를 형성한 후 클러스터 헤드는 주기적으로 방송되는 Hello 메시지를 통해 클러스터 멤버들의 존재 여부를 확인하게 된다. 하지만 클러스터 헤드가 측정된 이동성이 높을 경우에는 토폴로지를 안정적으로

표 2. 시간에 따른 노드의 클러스터 헤드 판단의 예

| Node_ID | CH_ID |     |
|---------|-------|-----|
|         | t     | t+1 |
| 1       | 4     | 4   |
| 2       | 1     | 2   |
| 3       | 4     | 4   |
| 4       | 4     | 4   |
| 5       | 10    | 9   |
| 6       | 6     | 6   |
| 7       | 4     | 4   |
| 8       | 4     | 4   |
| 9       | 10    | 9   |
| 10      | 2     | 2   |

유지시키기 위해 Hello 메시지 주기를 감소 시켜야 하며, 이동성이 작은 경우 Hello 메시지 주기를 증가시켜 불필요한 오버헤드를 감소시킨다.

이를 위해 본 논문에서는 다음과 같은 2단계의 적응적인 Hello 메시지 주기 조절 과정으로 구성된 클러스터 유지 방법을 사용한다.

### 3.3.1 클러스터 이동성 판단 단계

클러스터 이동성 판단 단계에서는 선정된 클러스터 헤드가 자신의 클러스터에 대한 이동성을 판단한다. 이때 클러스터의 이동성은 앞서 클러스터 헤드를 선출하는 비교 기준인 노드의 이동성 가중치 평균값  $M_{HA}(t)$ 으로 결정된다. 즉, 모든 노드들은 주변 노드와의 비교를 통해 가장 낮은  $M_{HA}(t)$  값을 갖는 노드를 클러스터 헤드로 선출하고 클러스터 헤드로 선출된 노드는 자신의  $M_{HA}(t)$  값을 자신의 클러스터 이동성으로 설정한다.

### 3.3.2 Hello 주기 조절 단계

Hello 주기 조절 단계에서는 클러스터 헤드에 의해 측정된 클러스터 이동성에 따라 Hello 메시지 주기를 조절한다. Hello 메시지 전송 주기는 노드의 이동성과 반비례 관계에 있으나, 네트워크 토폴로지 안정성과도 밀접한 상관관계를 갖는다. 즉, 이동성이 낮다고 하여 전송 주기를 길게 설정하면 Hello 메시지로 인한 네트워크 오버헤드는 줄일 수 있지만 빈번하게 변화하는 네트워크 토폴로지를 안정적으로 유지하기 어렵다.

반면, Hello 메시지 주기를 짧게 설정할 경우 빈번한 네트워크 토폴로지 변화를 빠르게 인지할 수 있지만, 토폴로지 유지를 위한 많은 오버헤드를 야기한다. 따라서 이러한 문제점을 해결하기 위해 본 논문에서는 식 3과같이 Hello 메시지 주기를 클러스터 이동성에 따라 적응적으로 조절한다.

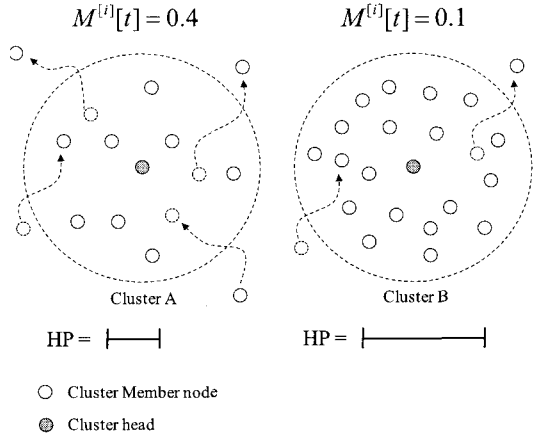


그림 5. 클러스터의 이동성과 Hello 메시지 주기의 관계

$$HP(t+1) = \frac{M_{HA}^{[i]}(t-1)}{M_{HA}^{[i]}(t)} \times HP(t) \quad (3)$$

그림 5에서는 두 클러스터 A와 B의 이동성에 따른 Hello 메시지 전송 주기와의 관계를 나타내고 있다. 그림에서 볼 수 있듯이 수식 1을 통해 측정된 클러스터 A와 B의 이동성은 각각 0.4와 0.1이며, 이러한 클러스터의 이동성과 Hello 메시지의 전송 주기가 서로 반비례하게 설정되는 것을 볼 수 있다.

## IV. 모의실험

본 논문에서는 제안하는 클러스터링 기법의 성능을 평가하기 위해 C++ 언어 기반으로 다음과 같은 모의 실험 환경을 구축하였다. 본 모의 실험에서는 무선 채널 환경에서 발생하는 신호의 감쇄를 위해 AWGN (Additive White Gaussian Noise) 채널 모델을 사용하였으며, 모든 이동 노드들의 좌표, 이동속도 및 방향은 1600m×1600m의 평면상에서 랜덤하게 발생하도록 하였다.

총 이동 노드의 수는 최소 20에서 최대 120개이고 각 노드의 최대 전송 반경은 250m이라 가정하였다. 또한 모의실험 시간은 초기 설정한 Hello 메시지 주기인 2sec를 기준으로 100번 수행할 수 있는 시간으로 200sec를 한번 모의실험 수행 시간으로 설정하였고, 50번의 실험 결과를 평균화하여 결과 그래프를 도출하였다. 표 3은 이와 같은 모의실험 환경에서 사용한 세부 파라미터를 정리한 것이다.

이와 같은 모의실험 환경을 토대로 본 논문에서 제안하는 클러스터링 기법의 성능 평가를 위해 Hello 메시지 전송 횟수, 노드의 이동성 변화율 그리고 클러스

표 3. 모의실험에 사용된 파라미터

| Parameters        | Value          |
|-------------------|----------------|
| 네트워크 크기           | 1600m×1600m    |
| 노드의 전송 반경(=R)     | 250m           |
| 전체 이동 노드의 수       | 20~120 개       |
| 최초 Hello 메시지 주기   | 2sec           |
| 노드 위치             | Uniform Random |
| 노드 이동 속도 및 방향     | Uniform Random |
| $\alpha$ (가중치 변수) | 0.4            |
| 모의 실험 반복 횟수       | 50번            |
| 총 모의 실험 시간        | 200sec         |

터 헤드 재선정 횟수 측면에서 기존 LID, LIDAR 방법들과 비교 분석하였다.

그림 6은 노드의 이동 속도에 따라 모든 노드에서 전송한 총 Hello 메시지 수를 나타내고 있다. 그림에서 볼 수 있듯이 LID의 경우 클러스터의 이동성을 반영하지 않고 고정적인 Hello 메시지 주기를 사용하기 때문에 Hello 메시지 전송 횟수가 일정하다. 반면, LIDAR와 제안 알고리즘은 클러스터 내 이동성에 따라 Hello 메시지 전송 주기를 조절하기 때문에 노드의 이동 속도 변화에 따라 Hello 메시지 전송 주기가 변하는 것을 볼 수 있다. 하지만 LIDAR는 이동성 계산 시 클러스터 밖의 주변 노드의 변화에도 영향을 받아 이동성이 높게 측정되어 Hello 메시지의 전송 주기가 짧아지기 때문에 많은 수의 Hello 메시지를 전송하게 된다. 이와 달리 제안 알고리즘의 경우 클러스터 내 이동성 계산 시 주변 노드의 영향을 받지 않기 때문에 적은 양의 Hello 메시지를 전송하게 된다. 그러나 단순히 Hello 메시지의 전송 횟수를 감소시켰다고 해서 제안 알고리즘의 우수성을 입증하기에는 무리가 따른다. 따라서 적은 전송 횟수로도 안정적으로 토폴로지를 유지할 수 있어야 한다.

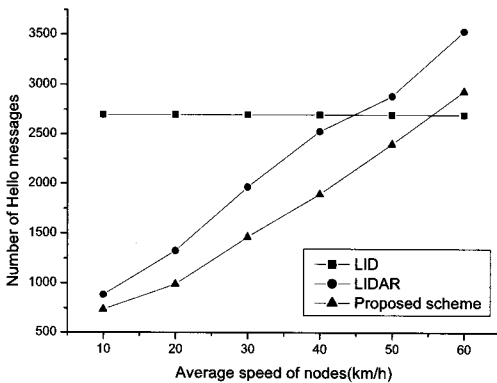


그림 6. 노드의 이동속도에 따른 총 Hello 메시지의 수

이를 위해 다음에서와 같은 노드의 이동성 변화율을 토대로 세 알고리즘의 성능을 비교 분석하였다. 이때 노드의 이동성 변화율은 해당 클러스터의 Hello 메시지 전송 주기 동안 변화한 클러스터 멤버 노드들의 수를 나타낸 것이다. 예를 들어 10개의 클러스터 멤버 노드가 존재하는 클러스터에서 4초의 Hello 메시지 전송 주기 동안에 클러스터 멤버 노드의 입/출입 수가 4였다면(4초 중 처음 2초 동안에 2개의 입/출입 노드 발생, 다음 2초 후에 2개의 입/출입 노드 발생), 이때 노드의 이동성 변화율은  $0.4(=4/10)$ 로 판단한다.

반면, 동일한 노드의 이동성 조건을 가진 클러스터 환경에서 Hello 메시지 전송 주기를 2초로 설정하면, 이때의 노드의 입/출입 수는 2이고, 이에 따라 노드의 이동성 변화율은 0.2로 측정된다. 따라서 노드의 이동성 변화율은 클러스터의 Hello 메시지 전송 주기를 조절하여 클러스터 멤버의 이동성 변화를 적정 수준으로 유지해 줄 수 있는 척도로 사용된다. 이는 노드의 이동성 변화율이 일정 임계치 이상으로 증가할 경우 Hello 메시지의 전송 주기를 현재의 주기보다 빠르게 설정하여 변화율을 낮춤으로써 클러스터 노드 이동성에 알맞은 측정 주기를 선택하여 오버헤드 감소 및 이동성 변화에 적절하게 대처할 수 있게 된다.

그림 7은 노드의 평균 이동 속도에 따른 노드의 이동성 변화율을 나타내고 있다. 그림에서 볼 수 있듯이 LID의 경우 노드의 이동 속도가 증가할수록 노드의 이동성 변화율은 현저히 증가되는 것을 볼 수 있다. 그러나 Hello 메시지 전송 주기를 클러스터의 이동성에 맞춰 조절하는 제안 알고리즘과 LIDAR의 경우 노드의 이동 속도가 증가함에도 불구하고 일정 수준 이상의 토폴로지 변화가 발생하지 않도록 보장해 줌을 확인할 수 있다. 더욱이 단말의 이동속도가 30km/h

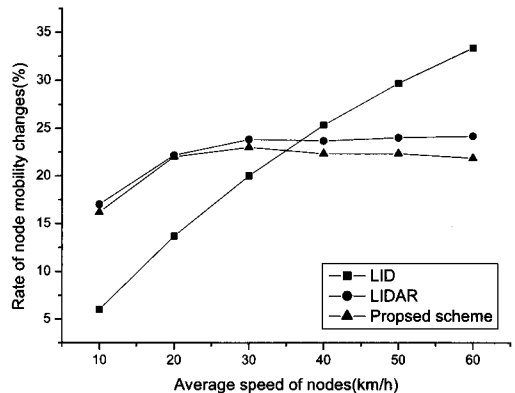


그림 7. 노드의 이동속도에 따른 노드의 이동성 변화율

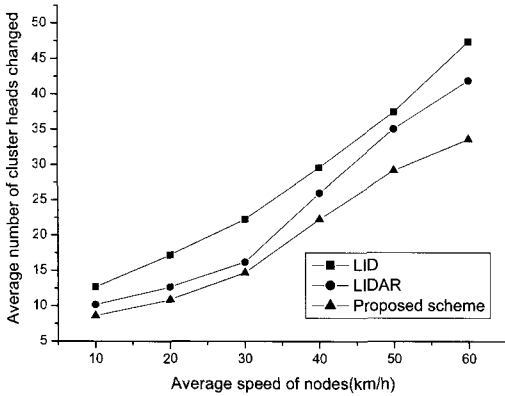


그림 8. 노드의 이동속도에 따른 클러스터 헤드 재선정 횟수

이상부터 제안 클러스터링 기법이 LIDAR 기법 보다 낮은 노드의 이동성 변화율을 제공할 수 있다. 이러한 성능 향상은 노드의 이동 속도가 증가할수록 더 커질 것으로 예상된다.

그림 8은 노드의 이동속도에 따른 클러스터 헤드 재선정 횟수를 나타내고 있다. 그림에서 볼 수 있듯이 LID의 경우 클러스터의 이동성을 고려하지 않고 낮은 ID를 소유한 노드가 클러스터 헤드로 선정되기 때문에 노드의 이동 속도가 증가함에 따라 헤드 재선정 횟수가 급격하게 증가하는 것을 볼 수 있다.

이와 더불어 LIDAR의 경우도 노드의 이동 속도가 10~30km/h까지는 제안 클러스터링 기법과 유사한 성능을 보이나 노드의 이동 속도가 40km/h 이상으로 증가할 경우 재선정 횟수가 제안 클러스터링 기법에 비해 증가하는 것을 볼 수 있다. 따라서 본 논문에서 제안하는 이동성 계산 방법을 사용하여 클러스터 헤드를 선정할 경우 보다 안정적으로 클러스터를 유지할 수 있음을 의미한다.

## V. 결론

본 논문에서는 모바일 애드 hoc 무선 네트워크 환경에서 이동성을 고려한 안정적인 클러스터 헤드 선정을 통해 클러스터 내 토폴로지를 안정적으로 유지, 관리하는 클러스터링 기법을 제안하였다. 또한 제안 클러스터링 기법의 성능 분석을 위해 모의 실험 환경을 구축하고 Hello 메시지 전송 횟수, 노드의 이동성 변화율 그리고 클러스터 헤드 재선정 횟수 등을 기존 LID와 LIDAR 기법과의 비교 분석하였다. 그 결과 Hello 메시지 전송 횟수와 낮은 노

드의 이동성 변화율 그리고 클러스터 헤드의 재선정 횟수 또한 감소시킬 수 있음을 확인하였다. 이러한 결과를 토대로 본 논문에서 제안하는 클러스터링 기법을 사용할 경우 노드의 이동성으로 인해 네트워크 유지가 불안정한 MANET 환경에서 보다 안정적인 클러스터 구축 및 유지가 가능할 것으로 기대된다.

## 참고 문헌

- [1] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, 16(4), pp. 11-21, Aug. 2002.
- [2] M. Chatterjee, S. K. Sas and D. Turqu, "An on-demand weighted clustering algorithm (WCA) for Ad Hoc Networks," In Proc. of IEEE GLOBECOM, pp. 1697-1701, 2000.
- [3] M. Jiang, J.Li, and Y.C. Tay, "Cluster Based Routing Protocol(CBRP)," Internet Draft draft-ietf-manet-cbrp-spec-01.txt, Aug. 1999.
- [4] D. Wei, and H. A. Chan, "A Survey on Cluster Schemes in Ad Hoc Wireless Networks," In Proc. of IEEE Mobility Conference, pp. 1-8, Nov. 2005.
- [5] Y. Wang, H. Chen, X. Yang, and D. Zhang, "WACHM: Weight based adaptive clustering for large scale heterogeneous MANET," In Proc. of Communications and Information Technologies, pp. 936 - 941, Oct. 2007.
- [6] Y. Wu, and W. Wang, "MEACA: Mobility and Energy Aware Clustering Algorithm for Constructing Stable MANETs," In Proc. of IEEE MILCOM, pp. 1-7, Oct. 2006.
- [7] G. Chen, F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks," In Proc. of IEEE HICSS, pp. 2450-2459, Jan. 2002.
- [8] C. R. Lin, and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal of Selected Areas in Communications*, 15(7), pp. 1265-1275, Sep. 1997.
- [9] M. Gerla, and J.T.C. Tsai, "Multi-cluster, mobile, multimedia radio network," In Proc. of Wireless Networks, pp. 255-265, 1995.
- [10] Y. J. Yu, and P. H. J. Chong, "A survey of

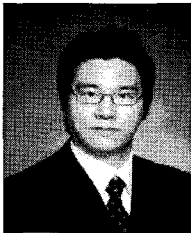


clustering schemes for mobile Ad Hoc Networks," *IEEE Communications Surveys*, 7(1), pp. 32-48, Mar 2005.

- [11] D. Gavalas, G. Pantziou, C. Konstantopoulos, and B Mamalis, "Lowest-ID with adaptive ID reassignment : a novel mobile ad-hoc networks clustering algorithm," In Proc. of Wireless Pervasive Computing, Jan. 2006.
- [12] P. Basu, N. Khan, and Little, T.D.C, "A mobility based metric for clustering in mobile ad hoc networks," In Proc. of IEEE ICDCS, pp. 413-418, Apr. 2001.

**김 혁 수 (Hyeksu Kim)**

준회원



2007년 2월 숭실대학교 정보통신전자공학부 학사  
 2009년 2월 숭실대학교 정보통신전자공학부 석사  
 <관심분야> Mobile Ad hoc Network

**황 준 호 (Junho Hwang)**

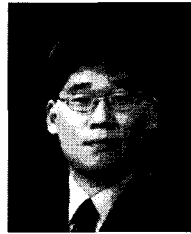
정회원



2004년 2월 숭실대학교 정보통신전자공학부 학사  
 2006년 2월 숭실대학교 정보통신전자공학부 석사  
 2006년 9월~현재 숭실대학교 정보통신전자공학부 박사과정  
 <관심분야> Optical Access Network, 3&4G 무선 이동통신 시스템, MANET

**유 명 식 (Myungsik Yoo)**

종신회원



1989년 2월 고려대학교 전자공학과 학사  
 1991년 2월 고려대학교 전자공학과 석사  
 2000년 6월 SUNY at Buffalo Dept. of EE 박사  
 2000년 9월~현재 숭실대학교 정보통신전자공학부 부교수  
 <관심분야> Optical Network, OBS, EPON, 인터넷 QoS, Wireless MAC Protocol, MANET