

무선 센서네트워크에서 동적 예비 클러스터 헤드를 이용한 효율적인 토폴로지 관리 방안에 관한 연구

정회원 김재현*, 이재용*, 김석규**, 도윤미***, 박노성***

A Dynamic Pre-Cluster Head Algorithm for Topology Management in Wireless Sensor Networks

Jaehyun Kim*, Jaiyong Lee*, Seoggyu Kim**, Yoonmee Doh***, Noseong Park*** *Regular Members*

요약

무선 애드 혹/센서 네트워크에서 제안된 일반적인 클러스터링 기반의 계층적 토폴로지 관리 기법들은 빈번한 토폴로지 변화에 따라 자주 클러스터링을 재구성하게 되고 네트워크 관리에 필요한 오버헤드가 증가하게 된다. 본 논문에서는 이러한 재클러스터링 문제와 부하 분산을 위하여 동적 예비 클러스터 헤드 기법을 사용하는 멀티 홉 클러스터링 알고리즘을 제안한다. 제안하는 기법은 이동성과 전원 레벨로 구성된 가중치 맵을 사용하여 예비 클러스터 헤드를 선출하고 멀티 홉 클러스터를 구성한다. 클러스터 헤드는 이러한 가중치 맵과 임계값을 사용하여 헤드의 역할을 예비 클러스터 헤드에게 넘겨주게 된다. 실험결과, 제안하는 알고리즘이 네트워크의 오버헤드를 줄이고 부하 분산을 제공하며, 토폴로지 변화에 무관하게 적절한 클러스터와 멤버를 관리할 수 있음을 확인 하였다.

Key Words : Wireless Sensor Network, Clustering, Pre-Clusterhead, Load Balancing, Prediction Weight MAP

ABSTRACT

As the topology frequently varies, more cluster reconstructing is needed and also management overheads increase in the wireless ad hoc/sensor networks. In this paper, we propose a multi-hop clustering algorithm for wireless sensor network topology management using dynamic pre-clusterhead scheme to solve cluster reconstruction and load balancing problems. The proposed scheme uses weight map that is composed with power level and mobility, to choose pre-clusterhead and construct multi-hop cluster. A clusterhead has a weight map and threshold to hand over functions of clusterhead to pre-clusterhead. As a result of simulation, our algorithm can reduce overheads and provide more load balancing well. Moreover, our scheme can maintain the proper number of clusters and cluster members regardless of topology changes.

I. 서론

무선 애드 혹/센서 네트워크에서 클러스터링에

기반한 계층적 라우팅 기법이 많은 장점을 가진다. 우선, 하나의 클러스터 구조는 시스템 용량이 증가함에 따라서 자원의 재사용이 용이하게 만든다. 예

※ 본 연구는 한국전자통신연구원의 RFID/USN용 센서태그 및 센서노드 기술개발 사업(2005-8-0784) 지원 및 정보통신부/정보통신 연구진흥원의 대학 IT 연구센터 육성·지원사업(IITA-2005-C1090-0502-0012)의 연구결과로 수행되었음.

* 연세대학교 전기전자공학과 유비쿼터스 네트워크 연구실 (jaykim@nasla.yonsei.ac.kr, jyl@yonsei.ac.kr)

** 안동대학교 전자정보산업학부 정보통신전공학 (sgkion@andong.ac.kr)

*** 한국전자통신연구원 USN시스템 연구팀 (ydoh@etri.re.kr, behack@etri.re.kr)

논문번호: KICS2006-01-039, 접수일자: 2006년 1월 21일, 최종논문접수일자: 2006년 2월 15일

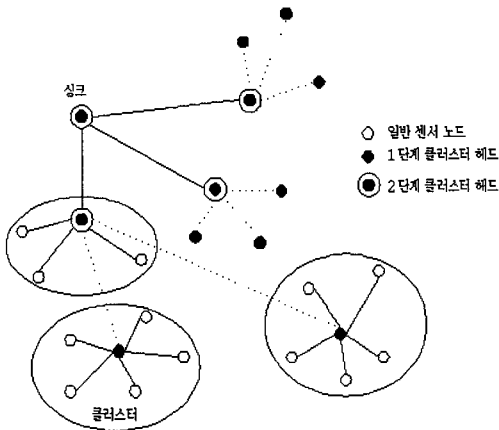


그림 1. 클러스터링 기반의 계층적 토폴로지 관리

를 들어 두 개의 독립적인 클러스터인 경우 동일한 주파수 및 코드를 각각의 클러스터 영역 안에서 사용할 수 있다. 또한, 클러스터 안에서 멤버들과 클러스터 헤더 사이에 TDMA 기반의 데이터 송수신 메커니즘으로 인하여 일반적인 경쟁 기반의 데이터 전송과 비교하여 데이터 충돌에 따른 자원 낭비를 줄일 수 있다. 결국, 클러스터링의 목적은 클러스터 기반 구조가 크고 동적인 네트워크에서 보다 자원을 효율적으로 사용하고 동적인 환경에 적응할 수 있도록 네트워크의 구조를 변경 관리하기 위함이다.^[1] 이러한 계층적 관리 구조는 클러스터 헤더라고 불리는 특정한 노드와 관리 구조를 이루는 요소 및 변수에 형성되는 클러스터 그리고 클러스터 멤버로 구성된다. 계층적 관리 구조의 클러스터 내의 모든 기능은 클러스터 멤버 간에 직접 수행되거나 혹은 클러스터 헤더를 통해 수행된다. 그러나 클러스터 외부와의 통신이나 라우팅 및 관리 기능은 클러스터 헤더를 통해서만 이루어진다. 따라서 클러스터 헤더가 네트워크의 병목점이 되는 단점이 있으나 계층적 관리 구조는 확장성이 용이하다. 하지만, 무선 센서 네트워크에서 자가 구성(self-organization) 및 다양한 동적 토폴로지 변화에 의하여 재클러스터링이 빈번하게 발생하게 된다. 이러한 빈번한 재클러스터링은 전체 네트워크의 수명 및 성능에 악영향을 미치게 되며, 재클러스터링 구축에 따른 오버헤드가 증가하고 이에 따른 에너지 소모 또한 증가하게 된다. 결국 전체 네트워크의 수명도 줄어들게 된다. 한편, 클러스터링 기반의 계층적 구조는 클러스터 헤더에게 부하가 집중되는 병목 현상 문제를 발생시킨다. 특히, 센서 네트워크에서의 에너지 효율성 측면에서 빈번한 데이터 송수신 및 클러

스터 관리에 따른 에너지 소모가 클러스터 헤더의 경우 급격히 증가하게 되어 결국 에너지 고갈 문제가 발생하게 된다. 따라서 적절히 클러스터 헤드를 선택 변경 해 주어야만 한다.^[10]

본 논문에서는 이러한 무선 센서 네트워크에서 클러스터링 기법 기반의 효율적인 계층적 구조 관리를 위하여 동적 예비 클러스터 헤드 개념을 이용한 효율적인 토폴로지 관리 기법을 제안한다. 제안하는 기법은 재클러스터링에 따른 오버헤드 및 에너지 소모를 최소화 하는 동시에 클러스터 헤더의 부하 분산을 잘 수행하여 전체 네트워크의 성능 및 수명을 향상시킨다. 시뮬레이션을 통하여 제안하는 알고리즘의 성능을 분석한다.

II. 관련 연구

클러스터링 기법은 단일 홉 혹은 다중 홉에 의한 클러스터링 방식으로 크게 나뉜다. 단일 홉 방식으로 제안된 알고리즘의 대표적인 방법은 *lowest_ID* 방법과 *highest_degree* 방법이 있고 다중 홉에 의한 방법은 높은 이동성을 갖는 환경에서 노드 밀도를 이용하여 다중 홉 클러스터를 구성하는 방법이 있다.^[10]

Lowest_ID 방법은 주소기반 클러스터링 방법이라고도 하며, 그림 2에서처럼 각각의 노드에 유일한 주소(ID)를 할당하고 가장 작은 주소를 가진 노드를 클러스터 헤드로 선출하는 단일 홉 토폴로지 관리 방법이다.^[11] 따라서 클러스터 헤더의 이웃 노드 주소들은 클러스터 헤더보다 항상 큰 주소를 가지고 있으며, 클러스터 헤더는 자신의 클러스터 안에 최소의 주소를 가진 노드가 새로 존재한다면 자신의 헤더 역할을 위임한다. 클러스터 헤드 노드의 급격한 에너지 소모 문제를 해결하기 위해 노드의 남은 전력을 고려하여 주기적으로 노드의 주소를 변경한다. *Highest_degree* 방법은 연결기반 클러스터링 방법으로 알려져 있으며, 그림 3에서처럼 노드 상호간의 전송 범위 안에 있는 노드의 밀도를 이용한 단일 홉 클러스터링 방법이다.^[2] 예를 들면, 노드 x 가 노드 y 의 전송범위 안에 있다면 노드 x 는 노드 y 의 이웃 노드가 되는데, 노드 중 이웃 노드 수가 가장 많은 노드(최대 밀도 혹은 최대 연결도)를 가진 노드가 클러스터 헤드로 선출되며 클러스터 헤더의 이웃 노드는 그 클러스터의 멤버가 되며 클러스터 헤드 선정 절차에 더 이상 참여하지 않는다. 클러스터 헤드들은 직접적으로 연결되어 있지

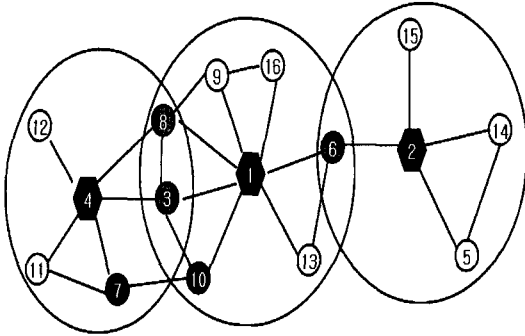


그림 2. Lowest_ID 방법의 클러스터링 구성

않기 때문에 클러스터마다 하나의 클러스터 헤드가 허용된다. Highest_drgree 방법도 lowst_ID와 마찬가지로 노드의 밀도 큰 노드가 클러스터 안으로 이동해 오는 경우 클러스터 재구성을 통해 클러스터 헤드를 재선정하고 노드는 자신의 이웃 노드 정보를 갱신하여 브로드캐스트 한다. 반면에, 현재 클러스터 헤드는 자신의 이웃 노드 하나와의 연결이 해제된다 하더라도 클러스터를 재구성하지 않는다. AMCS(Adaptive Multihop Clustering Scheme in highly mobile ad hoc networks)는 애드 혹 네트워크에서 노드의 이동성을 고려한 다중 홉 클러스터링 알고리즘이다.^[4] 이 방법은 노드의 연결도(밀도)를 이용한 계층적인 토폴로지 관리 방법으로 클러스터는 클러스터 헤더와 클러스터 멤버로부터 단일 홉 혹은 다중 홉 거리에 있는 클러스터 멤버로 구성된다. 이동 노드는 클러스터의 멤버가 될 경우 이웃 클러스터 멤버와 주기적으로 정보를 교환하며, 동일한 클러스터 내에 있는 멤버와는 제어 패킷을 통해 토폴로지 정보를 교환한다. 또한 이 알고리즘은 클러스터 멤버의 수를 제한적으로 운영하여 클러스터를 관리 하여 클러스터 헤더의 오버헤드를 분산시키고 만약 제한된 멤버의 수를 초과하게 되는 경우 클러스터를 분리하여 자동적으로 적당한 클러스터 멤버의 수를 유지하여 관리하게 된다. MOBIC은 이동하는 특히 고려한 클러스터링 방법으로써 그 클러스터 구조에서 이동하는 노드들의 이동성 패턴을 결정하여 서로 비슷한 이동성을 가지는 노드들로 클러스터링을 구성하여 최대한 클러스터 멤버간의 변동이 가장 적어지도록 만들어 주는 클러스터링 알고리즘이다.^[5] MOBIC에서는 이웃 노드들로부터 각각 노드의 이동하는 스피드의 분산을 계산한 정보를 받아 이동성을 예측하게 되고 이동성이 가장 작은 노드를 클러스터 헤드로 선출하게 된다. LEACH는 무선 센서 네트워크를 위한 클

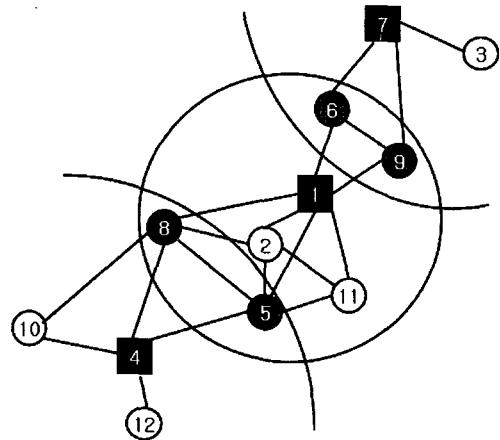


그림 3. Highest_degree 방법의 클러스터링 구성

러스터링 기반 라우팅 기법으로 클러스터 헤드가 멤버들로부터 데이터를 수집하여 데이터 퓨전(data fusion)을 통해 데이터를 모아서 싱크 노드에게 전달한다.^[6] 이 기법의 특징은 네트워크에 있는 모든 센서 노드들에게 에너지 소비를 공평하게 분산시키기 위해 클러스터 헤드를 랜덤하게 순환시켜 모든 노드가 클러스터 헤드로써 선정 되는 기회를 공평하게 부여되며, 매 라운드(round) 마다 일정한 수의 클러스터를 구성하고 클러스터 헤드가 고르게 배치하도록 한다. LEACH는 라운드라는 시간 단위로 이루어지는데 클러스터 헤드가 선출되는 설정 단계와 TDMA 기반의 스케줄로써 클러스터 헤드가 멤버들과 데이터를 송수신하는 안정화 단계로 이루어 동작한다. TEEN은 센서 노드들이 주기적으로 전송할 데이터를 가지지 않는다는 점을 제외하고 LEACH와 유사하게 동작한다.^[7] 단지 클러스터 헤드가 미리 설정된 하드 임계값(hard threshold)과 소프트 임계값(soft threshold)에 따라 변경되어 진다. APTEEN은 TEEN과 같이 시간에 민감한 데이터를 처리하고 반응 동작을 위해서, 센서 노드가 카운트 시간동안 데이터를 전송하지 않는 경우 데이터를 감지해서 클러스터 헤드로 전송하도록 하여 TEEN의 단점을 개선하였다.^[8]

III. 동적 예비 클러스터 헤드 알고리즘

3.1 문제 제기

클러스터링 기법의 대표적인 Lowest_ID와 Highest_degree 방법은 노드의 주소 혹은 노드의 밀도를 변수로 하여 클러스터링을 하는 방법으로써 노드의 이동 및 새로운 노드의 네트워크 참여 등에 따른

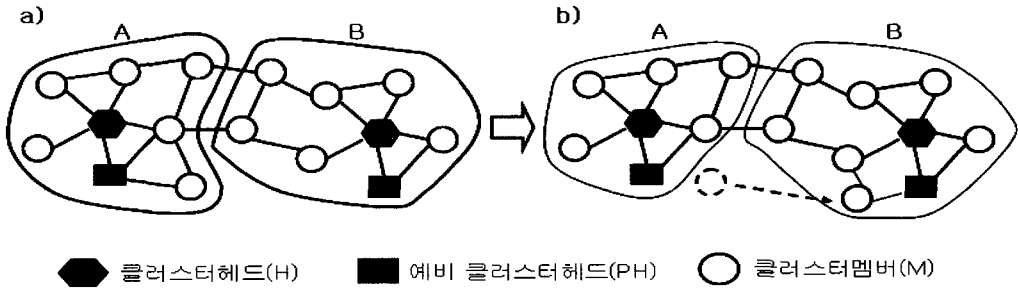


그림 4. 노드의 클러스터 구성

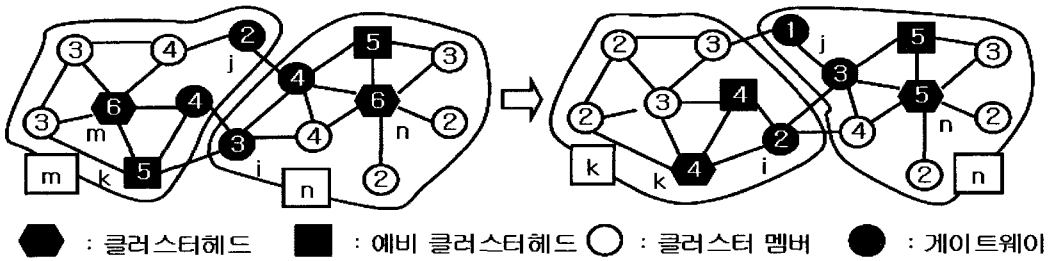


그림 5. 클러스터의 자동 관리

토폴로지 변경 시 클러스터 헤더의 재선출 및 클러스터를 재구성해야 한다. 그러므로 클러스터 헤더의 재선출에 따른 네트워크의 프로세싱 오버헤드와 선출 지연이 발생되고, 작은 클러스터의 재구성으로 토폴로지가 불안정하게 관리되며 경로 설정, 자원할당 및 데이터의 손실을 초래하게 된다. 따라서, 작은 클러스터 재구성으로 인한 불안정한 토폴로지는 네트워크 성능에 많은 영향을 미치고, 네트워크 오버헤드와 각 노드의 프로세싱 오버헤드에 의해 네트워크 수명에도 영향을 미친다.

AMCS 방법의 경우도 다중 홉 기반의 클러스터링 기법으로써 기본적으로 노드 밀도(연결도)에 기반한 클러스터링 방법으로써 클러스터 멤버의 최소/최대 제한 값에 의해 비교적 안정적인 클러스터를 구성 관리 하게 되나, 이동 노드의 증가와 제한 값 초과에 따른 재클러스터링 및 클러스터헤드 재선출에 의한 오버헤드 문제 및 성능 저하 현상은 동일하게 발생된다. 센서 네트워크를 위한 클러스터링 기법인 LEACH의 경우 랜덤하게 클러스터 헤드를 선출하여 매 round마다 클러스터링을 재구성하게 되는데 적절한 클러스터 헤드 선출하는 방법이 어려우며, 비록 에너지 효율성은 향상되지만, 매 round마다 클러스터링을 재구성하는데 소요되는 오버헤드가 크다. TEEN과 APTEEN의 경우 기본적으로 LEACH와 동일한 클러스터링 알고리즘을 적용한것이므로 LEACH와 거의 유사한 문제점을 가지

고 있다.

3.2 동적 예비 클러스터 헤드 기법

본 논문에서 제안하는 알고리즘은 동적 예비 클러스터 헤드 기법의 다중 홉 클러스터링 기법으로써 노드의 추정된 가중치를 기반으로 하는 맵(MAP)중심의 다중 홉 클러스터링을 통해 이루어진다. 노드의 가중치는 노드의 이동성과 노드의 에너지를 고려한 함수 값으로 구해지며, 노드의 이동성은 노드간 평균 속도를 나타내며, 전력은 노드에 남아 있는 배터리의 파워이다. 계층적 토폴로지 구조가 갖는 클러스터들은 클러스터 헤드, 예비 클러스터 헤드, 클러스터 멤버, 게이트웨이 노드로 구성되며 클러스터 멤버는 클러스터 헤드로부터 단일 홉이거나 다중 홉 거리에 있다.

클러스터 헤드는 데이터 전송 및 클러스터를 관리하는 역할을 수행한다. 클러스터 멤버들의 이동성과 에너지 값을 예측하여 가중치 맵을 작성하고 이를 바탕으로 예비 클러스터 헤드를 선출한다. 예비 클러스터 헤드는 클러스터 헤드에 의해 선정되며, 일반적인 상황에서는 클러스터 멤버와 동일하나, 클러스터 헤드가 갑작스런 동작 중단이나 부하 분산 임계값 이하가 될 경우 클러스터 헤드 역할을 수행한다. 게이트 노드는 클러스터 간의 연결을 보장한다. 클러스터 멤버 노드는 클러스터 헤드에 의해 관리 되는 일반 노드이다. 계층적 구조를 보다 효율적

으로 구성하고 유지하기 위해 이동 노드는 기본적으로 이웃 노드의 정보를 통해 클러스터 헤드 및 게이트웨이와 같은 역할을 수행한다. 다시 말해, 이동 노드는 자동적으로 클러스터의 멤버로서 역할을 수행한다. 또한 이동 노드는 자동적으로 클러스터를 구성하고 클러스터는 자동적으로 클러스터 자신을 유지하는 것이다. 이동 노드가 패킷을 전송하기 위해 목적지 노드까지의 경로를 설정하였으나, 노드의 이동으로 인해 목적지 노드까지의 경로를 설정하였으나, 노드의 이동으로 인해 설정된 경로를 통해 패킷을 전송하지 못하는 경우가 종종 발생한다. 클러스터링의 경우 그러한 상황이 자주 발생한다. 그렇기 때문에, 센서 네트워크에서 계층적인 구조를 유지하는 것은 클러스터가 노드의 이동성에 무관하게 유지될 필요가 있기 때문이다. 제안하는 동적 예비 클러스터 헤드 알고리즘은 이러한 노드 이동에 따른 토폴로지 변경 시 재클러스터링을 최소화하고 클러스터의 안정성을 보장해 준다. 그림 4는 이동 노드가 자동적으로 클러스터를 구성하는 예를 보여 준다. 그림 4 (a)와 같이 네트워크에 두 개의 클러스터 A, B가 있다고 가정한다. 클러스터 A의 멤버 노드 중 하나의 노드가 클러스터 B로 이동한다고 가정하면 그림 4 (b)와 같이 토폴로지가 변경된다. 클러스터 A는 클러스터 멤버가 클러스터로부터 떠났지만 처음부터 이동해 간 클러스터 멤버가 없었던 것처럼 정상적으로 동작한다. 또한 클러스터 B는 클러스터 A로부터 노드가 이동해 왔지만 시작부터 그 노드가 마치 있었던 것처럼 동작한다. 제안하는 알고리즘은 네트워크에 참여하는 노드의 가중치 (이동성과 에너지의 함수 값)에 의해 클러스터 헤드를 선출하고 다중 홉으로 구성된 클러스터를 형성한다. 클러스터의 크기는 클러스터 헤드로부터 홉 수를 고려하여 클러스터 상호 적절한 크기를 유지한다. 클러스터 헤드는 일정 기간 후 추정된 가중치에 의해 한계 값에 도달하게 되면 자신의 단일 홉 이웃 노드 중 가중치가 가장 큰 노드를 예비 클러스터 헤드로 선출하고 일정 기간 후 예비 클러스터 헤드가 클러스터 헤드가 된다. 그림 5 (a)의 경우와 같이 두 개의 클러스터가 있다고 가정하면 노드 m과 n이 클러스터 헤드(클러스터 주소도 m과 n)이고 예비 클러스터 헤드는 k와 l이다. 일정 시간이 지난 후 클러스터 m의 헤드가 예비 클러스터 헤드 k로 변경된 경우 노드 k가 클러스터 헤드 역할을 수행한다. 또한 노드 k로부터의 홉 수를 계산하여 노드 l은 클러스터 k의 멤버가 되고 노드 j는 클러스

터 n의 멤버로 역할을 한다. 이러한 절차를 통해 클러스터는 자동적으로 유지된다. 선출된 클러스터 헤드는 이동성과 에너지 값에 의한 가중치 맵을 바탕으로 예비 클러스터헤드 선출 임계값(Pre-cluster-head selection threshold)이하로 가중치 값이 떨어지게 되면 다음으로 가장 가중치가 큰 노드를 예비 클러스터 헤드로써 선출한다. 이후, 클러스터 헤드는 예비 클러스터 헤드가 선출되었음을 멤버들에게 알린다. 가중치가 부하 분산 요소 임계값(Load balancing factor: LBF threshold)보다 이하로 떨어지면 클러스터 헤드는 예비 클러스터 헤더에게 클러스터 헤드의 역할을 넘겨주게 된다. 다음 그림 6은 제안하는 알고리즘의 동작 순서도이다.

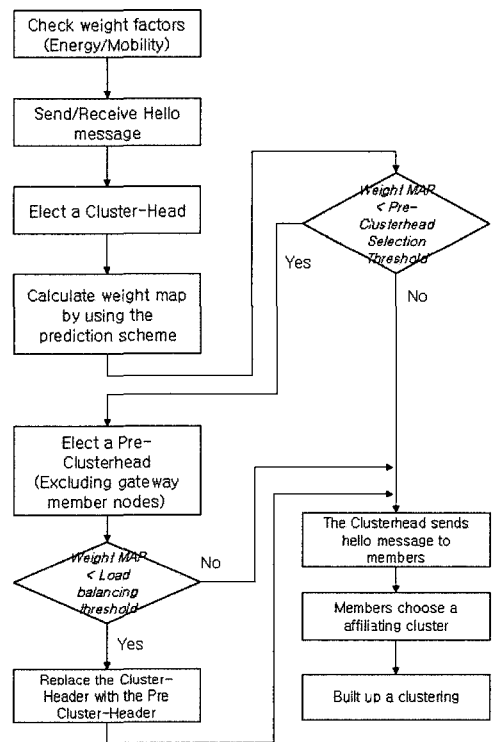


그림 6. 동적 예비 클러스터 헤드 알고리즘 동작 순서도

3.3 이동성과 에너지 예측

노드의 에너지를 예측하기 위해 클러스터 헤더는 각각의 클러스터 멤버가 어떤 상태로 얼마간의 시간 동안 있었는지를 알고 있어야 한다. 클러스터를 형성하게 되면 TDMA 방식을 이용하여 클러스터 헤더와 멤버가 통신을 하게 된다. 이렇게 되면 클러스터 헤더는 각각의 노드가 얼마나 오랫동안 통신을 했는지를 알 수 있고, 이를 노드가 평균 송신 파워와 곱하면 노드가 통신하는데 사용한 에너지

량을 알 수 있다. 그리고 노드가 수면(sleep) 상태에서 사용하는 평균 에너지 량도 고려한다면 더 정확한 에너지 예측이 가능하다. 이를 수식적으로 나타내면 다음과 같이 나타낼 수 있다.

$$E_{consume} = P_{avg_tx} \times T_{tx} + P_{avg_idle} \times T_{idle} \quad (1)$$

$E_{consume}$: 노드가 소모한 총 에너지 예측 량

P_{avg_tx} : 평균 전송 에너지

T : 클러스터 헤더가 선출된 이후부터의 총 시간

T_{tx} : 클러스터 헤더가 측정한 노드의 전송 시간

$T_{idle} = T - T_{tx}$: 노드의 idle 시간

이동성을 예측하여 가중치 맵을 만들기 위해서는 기본적으로 각각의 노드들은 자신의 이동 속도와 방향을 알고 있어야 한다. 이러한 정보는 클러스터 헤더와 동기를 맞출 때 클러스터로 전달되며 이 정보를 기록해 놓았다가 클러스터 헤더가 평균 이동 속도와 분산을 구하여 가중치를 계산하는데 사용한다. 이러한 정보는 정확한 이동성을 예측하지는 못하지만 가중치 맵을 만드는 데는 충분한 효과를 발휘할 수 있다. 예를 들면 클러스터 헤더로부터 멀어지는 방향으로 이동하는 노드나 이동속도가 빠른 노드들은 클러스터 헤더를 선출하는데 적합하지 않기 때문에 이들 노드에 대해서는 가중치를 낮게 주면 된다. 반면 이동속도가 느리고 클러스터 헤더 방향으로 움직이는 노드들은 클러스터 헤더로서 적합하기 때문에 이러한 노드들은 가중치를 높게 해주면 된다. 그리고 각각의 노드의 분산을 구해서 분산이 작은 노드가 급격한 움직임이 적다고 볼 수 있으므로 이러한 노드들의 가중치를 높게 하면 좋은 효과를 얻을 수 있다. 이러한 수준의 이동성 예측만으로도 가중치 맵을 만들어 클러스터 헤더를 선출하게 되면 많은 효과를 볼 수 있다.

IV. 실험 환경 및 성능 평가 방법

본 논문의 제안 알고리즘 성능 평가를 위해 ns-2를 사용하였다⁹⁾. 이동 노드는 50, 75, 100개를 기반으로 하는 센서 네트워크를 구성하였다. 이동 노

드의 초기 위치는 500m × 500m 크기의 면적 위에 단일 랜덤 분포(uniform random distribution)를 통해 선정하였고 랜덤 위치 모델(random way-point model)을 사용하였다. 랜덤 위치 모델이란, 이동 노드는 지정된 평균 속도로 정해진 위치로 이동하고 위치에 도착하면 정해진 시간 동안 머무르게 된다. 이 시간을 휴지 시간(pause time)으로 정의한다. 본 논문의 실험에서는 휴지 시간을 '10'으로 설정하였다. 그러므로 이동 노드가 지정된 위치에 도착하자마자, 새로운 목적지를 선택하고 다시 선택된 위치로 이동을 시작한다.

각 노드의 전송 범위는 100m이고 각 노드들은 자신의 전송 범위 내에 있는 모든 노드들과 아무때나 통신할 수 있다. 실험 시간은 5000 sec로 하였고 클러스터링 알고리즘에 의해 생성된 초기 계층적인 구조를 보장하기 위해 10 sec 동안은 정보를 수집하지 않았고 5000 sec에 이르면 모든 노드의 이동을 멈추게 하였다. 노드의 평균 이동 속도는 0.1 m/s, 0.5 m/s, 1 m/s, 2 m/s 이다.

제안 알고리즘의 성능을 평가하기 위해 우선 적절한 부하 분산 요소(Load Balancing Factor: LBF)를 모의실험을 통하여 찾아본다. LBF는 클러스터 헤더의 역할을 예비 클러스터 헤더에게 넘겨 주기 위한 기준 역할과 더불어 클러스터 헤더의 부하를 분산하는 역할을 하게 된다. 이러한 LBF를 바탕으로 제안하는 알고리즘과 주기적인 클러스터링과의 네트워크 오버헤드와 제안하는 알고리즘의 적응력(Adaptability)와 부하 분산 능력(Load Balancing)에 따른 성능을 평가한다.

V. 결과 분석

4.1 부하 분산 요소(Load Balancing Factor: LBF)

적절한 LBF를 구하기 위해 이동 노드는 100개, 전송 범위는 100m, 노드는 고정된 위치에서 기능을 수행토록 하였고 LBF를 0.6 ~ 0.9까지 0.05 간격으로 설정하여 5000초 동안 실험하였다. 다음 그림 7과 표 1은 적절한 LBF를 찾기 위한 실험 결과이다. 제안하는 알고리즘에 대한 실험 결과, 그림 7을 살펴보면 LBF = 0.9 인 경우에 전체 노드에 대한 전력의 분산이 작고 일정하므로 전체 노드에 대한 Load balancing이 적절하게 이루어진다고 볼 수 있다. 환원 하면, LBF ≈ 1 일수록 클러스터 헤더의 역할 분담을 위한 한계 값(Threshold value)이 이상

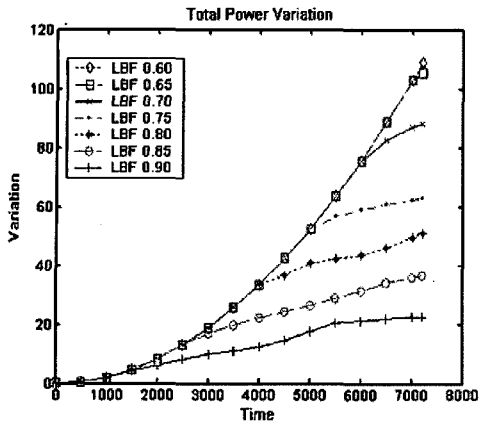


그림 7. LBF 값의 변화에 따른 노드 전력의 분산

표 1. LBF값에 따른 클러스터 헤더 및 노드의 클러스터 변동 수

	0.60	0.65	0.70	0.75	0.80	0.85	0.90
헤더 변동 수	8	9	11	12	13	15	43
노드의 클러스터 변동 수	17	20	25	27	30	32	105

적으로 설정된다고 말할 수 있다. 그러나 표 1의 결과와 같이 LBF가 커질수록 클러스터 헤더의 변동 수가 증가하게 되고 이에 따라 클러스터 멤버의 클러스터 변경 수도 증가하게 된다. 클러스터 헤더의 변동 수와 클러스터 멤버의 클러스터 변화 수가 LBF 0.6에서 0.85까지는 일정하기 증가됨을 확인할 수 있으나, 0.9 이상에서는 그 수가 급격히 증가하는 것을 볼 수 있다. 클러스터 헤더가 자주 변하게 되면 전체 클러스터 멤버의 토폴로지가 자주 변하게 되고 이에 따른 노드의 처리 오버헤드가 증가하게 되므로 결국 전체적인 네트워크의 오버헤드가 증가한다고 볼 수 있다. 그러므로 제안하는 알고리즘의 모의실험 결과, 전체적인 부하 분산을 원활하게 하고 네트워크 오버헤드를 최소화 할 수 있는 적절한 LBF 값을 0.85로 선정하였다.

4.2 네트워크 오버헤드(overhead)

네트워크의 오버헤드 실험을 위하여 예비 클러스터 헤더를 선출하는 기법의 제안하는 알고리즘과 일정 주기를 가지고 클러스터 헤더를 선출하는 알고리즘을 비교 하였다. 기본 실험 환경은 앞의 LBF 실험 환경과 동일하다. 일정 주기로 클러스터를 선출하는 알고리즘은 30초의 주기를 가지고 클러스터 헤더를 계속해서 선출하도록 하였다.

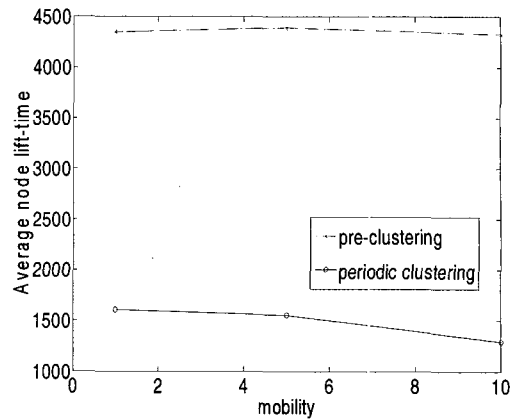


그림 8. 주기적인 클러스터 헤더 선출 알고리즘과 예비 클러스터 헤더 선출 알고리즘간의 노드 평균 수명 비교

그림 8에서 시뮬레이션 결과가 나타나 있다. 위에서 보면 한눈에 알 수 있듯이 주기적으로 클러스터 헤더를 선출하는 알고리즘의 경우 각각의 노드들의 평균 수명이 예비 클러스터 헤더를 선출하는 알고리즘의 경우와 비교해 보면 약 1/3의 평균 수명을 가지고 있다. 그리고 주기적으로 예비 클러스터 헤더를 선출하는 알고리즘은 노드의 이동성이 증가함에 따라서 노드의 평균 수명이 점차 감소하는 추세를 보여주고 있다. 하지만, 예비 클러스터 헤더를 선출하는 알고리즘의 경우에는 노드의 이동성이 증가하더라도 노드의 평균 수명에는 거의 영향을 주지 않는다는 것을 보여주고 있다. 이러한 결과가 나타나는 이유는 예비 클러스터 헤더를 선출할 때 별도의 헬로우(Hello) 메시지를 주고받지 않고 예측기법에 기반 한 가중치 맵을 통해서 선출하기 때문에 추가적인 오버헤드가 발생하지 않기 때문이다. 따라서 이동성이 높은 네트워크 일수록 노드의 수명차이는 점차 벌어지게 된다.

표 2. 마지막 홉 노드 증가에 따른 에너지 소비

노드 수 /전송범위	노드 이동 속도 (m/s)				
	0.1	0.5	1	2	
50	50	2.96 (5.15)	2.97 (5.16)	2.94 (5.21)	3.02 (5.22)
	75	2.46 (4.56)	2.42 (4.51)	2.53 (4.63)	2.54 (4.66)
	100	2.01 (3.84)	2.03 (3.91)	2.05 (3.95)	2.10 (4.01)
75	50	2.85 (4.95)	2.89 (4.97)	2.91 (4.99)	2.90 (5.01)
	75	2.51 (4.43)	2.50 (4.47)	2.53 (4.46)	2.54 (4.49)
	100	2.13 (4.05)	2.18 (4.09)	2.20 (4.13)	2.22 (4.17)
100	50	3.21 (5.02)	3.22 (5.09)	3.23 (5.11)	3.25 (5.07)
	75	2.75 (4.64)	2.77 (4.62)	2.75 (4.68)	2.81 (4.67)
	100	2.27 (4.10)	2.30 (4.17)	2.32 (4.22)	2.33 (4.28)

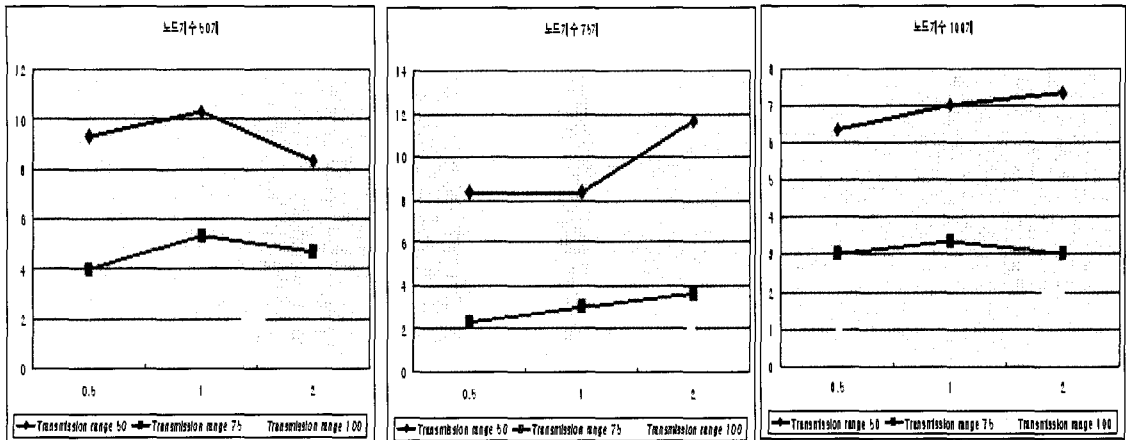


그림 9. 클러스터 헤드의 평균 변화 수

표 2는 각 클러스터에 따른 클러스터 헤더에서 클러스터 멤버까지의 평균 거리의 평균을 나타낸다.

최대 거리의 평균은 괄호() 안에 표시된 값이다. 이 표의 각 항목은 노드의 전송범위와 노드의 이동 속도에 따른 결과를 보여준다. 노드의 이동속도가 0.1m/s에서 2m/s까지 다양하게 변할지라도 평균 거리와 최대 평균 거리의 평균은 거의 비슷하게 유지된다. 이 실험 결과는 노드의 이동속도에 무관하게 토폴로지가 적절하게 유지 관리됨을 보여준다.

4.3 적응력(Adaptability)과 부하 분산(Load Balancing) 실험

센서 네트워크에서 계층적인 구조를 구성하고 유지하기 위해서는 각 노드의 역할이 노드의 이동 등에 따른 변화에 잘 적응하여야 한다. 이를 확인해 보기 위해서 제안하는 기법이 동적인 환경에서 적절하게 적응하는가의 여부를 이동성에 따른 클러스터 헤더의 숫자 변화를 확인해 보았다.

그림 9는 노드의 이동속도(0.5, 1, 2m/s)와 전송 범위(50, 75, 100m)에 따른 클러스터의 평균수를 나타낸다. 그림 9의 경우 제안하는 기법이 전송범위에 무관하게 노드의 이동속도에 영향을 받지 않는 것을 보여준다. 또한 전송 범위에 따른 클러스터의 수는 전송 범위가 커질수록 클러스터의 수가 감소하는 것을 보여준다. 그림 9의 실험결과 제안하는 기법은 노드의 이동속도가 변하더라도 비슷한 수의 클러스터와 멤버를 유지하는 것을 보여주며 전송범위에 따라 각각의 값이 변경되는 것을 보여준다. 따라서 제안하는 기법은 노드의 이동 등에 따른 토폴로지 변경 시 클러스터를 재구성하지 않고 이에 무관하게 적절하게 토폴로지를 관리 할 수 있다.

VI. 결론

본 논문에서는 애드 혹/무선 센서 네트워크를 위하여 연구되고 제안된 계층적인 관리 구조를 갖는 토폴로지 관리 기법에 대해서 살펴보고, 제안된 기법들의 문제점인 토폴로지 변경 시 클러스터를 재구성함으로써 클러스터 헤더의 재 선출에 의한 선출 지연, 재구성에 따른 정보손실과 네트워크의 부하 분산에 대한 문제도 살펴보았다.

따라서 이러한 문제를 효과적으로 해결하기 위해 무선 센서 네트워크에서의 동적인 예비 클러스터 헤드 기법을 이용한 다중 홉 클러스터링 기법을 제안 하였다. 클러스터 헤드는 예측 기중치 맵을 사용하여 에너지와 이동성 예측을 수행한다. 이러한 예측 기법을 통해 동적인 예비클러스터 헤드를 선출하게 되며, 이러한 예비클러스터 헤드를 이용하여 토폴로지 변경 시 클러스터를 재구성하지 않고 현재 클러스터 구조를 유지하도록 하였다. 또한 기존의 클러스터링 알고리즘의 경우 주기적인 클러스터 정보를 교환함으로써 발생하는 오버헤드에 따른 에너지 소모 등의 문제가 발생하는데 비해 제안하는 알고리즘에서 사용한 예측 기법의 경우 주기적인 클러스터 정보 교환이 필요 없어 오버헤드를 최소화하여 결과적으로 에너지 효율성을 증대시켰다. 실험 결과에서 보여지듯이 노드의 이동 등으로 인하여 토폴로지가 변경된다 하더라도 제안하는 기법은 클러스터를 재구성하지 않고 각 클러스터의 크기를 비슷하게 유지하고 클러스터 헤더와 예비 클러스터 헤더간의 적절한 역할 분담을 통해 클러스터링이 잘 유지되고 전체 네트워크의 부하가 적절히 분산되어 결과적으로 네트워크 수명이 최대화됨을 알

수 있다. 따라서 제안하는 기법은 동적인 환경을 갖는 애드혹/센서 네트워크에서 노드의 이동 및 이상 등에 따른 토폴로지 변화에 무관하게 토폴로지를 잘 유지하므로 계층적 구조를 갖는 토폴로지 관리에 효과적으로 적용될 수 있을 것으로 기대된다. 추후에 부하 분산과 예측 가중치 맵의 정확성 및 최적화를 위한 분석 및 성능 평가 등이 필요할 것으로 여겨진다.

참고 문헌

[1] 김동완, "이동 Ad-Hoc 망 기술개요," TELECOMMUNICATIONS REVIEW, 2000.
 [2] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," in Proc. IEEE, vol. 75, 1987, pp. 56-73.
 [3] M. Gerla and J. T. Tsai, "Multiuser, Mobile, Multimedia Radio Network," Wireless Networks, vol. 1, Oct. 1995, pp. 255-65.
 [4] T. Ohta, S. Inoue, and Y. Kakuda, "An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks," in Proc. 6th ISADS'03, Apr. 2003.
 [5] P. Basu, N. Khan, and T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," in Proc. IEEE ICDCSW'01, Apr. 2001, pp. 413-18.
 [6] W. B. Heinzelman, A. P. Chandrakasan, Hari Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on wireless communications, vol. 1, No. 4, October 2002.
 [7] Arati Manjeshwar et al., "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," Proc. Second International Workshop Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, 2001.
 [8] Arati Manjeshwar et al., "APTEEN: A Hybrid Protocol for Efficient Routing and

Comprehensive Information Retrieval in Wireless Sensor Networks," IEEE Proc. of the International Parallel and Distributed Processing Symposium (IPDPS'02), April, 2002, pp.195-202.

[9] <http://www.isi.edu/nsnam/ns/index.html>
 [10] Shin Bond Deug, "Efficient Topology management using Pre-Clusterhead in Ad hoc Networks," 제21회 한국정보처리학회 춘계학술대회 논문집 제11권 제1호, 2004.5.14~5.15, pp. 1537-1540.

김 재 현 (Jaehyun Kim)

정회원



1997년 2월 광운대학교 전자공학
학과 졸업
2003년 8월 연세대학교 전기전
자공학과 석사
2003년 9월~2004년 4월 SK 텔
레텍 연구원
2004년 9월~현재 연세대학교 전
기전자공학과 박사과정

<관심분야> Ubiquitous Sensor Network, Wireless Routing/MAC Protocol Design, Mobile/Reliable Multicast, IPv6

이 재 용 (Jaiyong Lee)

정회원



1977년 2월 연세대학교 전자 공
학과 졸업
1884년 5월 IOWA State Univer-
sity 공학석사
1987년 5월 IOWA State Univer-
sity 공학박사
1987년 6월~1994년 8월 포항공
과대학 교수

1994년 9월~현재 연세대학교 전자공학과 교수
 <관심분야> Protocol Design for Wired/Wireless QoS Management, Ubiquitous Sensor Network, Wireless Multimedia Support Protocol

김 석 규 (Seoggyu Kim)

정회원



1990년 2월 연세대학교 전자 공
학과 졸업
1992년 8월 연세대학교 전자 공
학과 석사
1997년 8월 연세대학교 전자 공
학과 박사
1997년 9월~2004년 3월 SK 텔

레콤 선임연구원

2004년 3월~2006년 3월 연세대학교 전기전자공학과
IT 연구단 연구 교수

2006년 4월~현재 안동대학교 전자정보산업학부 정보
통신공학전공 조교수

<관심분야> USN, 유무선 통합망에서 QoS Architecture,
B3G Convergence Network, TCP 성능 분석, IP기
반 유무선 통합 네트워크

박 노 성 (Noseong Park)

정회원

2003년 숭실대학교 컴퓨터 학부 졸업

2005년 한국정보통신대학교 공학부 석사

2005년~현재 한국전자통신연구원 텔레매틱스··USN
연구단 연구원

<관심분야> 유비쿼터스 센서 네트워크

도 윤 미 (Yoonmee Doh)

정회원

1989년 경북대학교 전자공학과 졸업

1991년 경북대학교 전자공학과 석사

2003년 University of Florida 컴퓨터공학 박사

1991년~1997년 한국전자통신연구원 선임연구원

2001년~2003년 Arizona State University 컴퓨터공학
과 방문 연구원

2003년~2004년 한국정보통신대학교 연구교수

2004년~현재 한국전자통신연구원 텔레매틱스··USN
연구단