

무선 센서네트워크에서 대표경로를 이용한 클러스터기반 라우팅 프로토콜의 설계 및 구현

Design and Implementation of Cluster based Routing Protocol using Representative Path in Ubiquitous Sensor Network

장유진, 김아름, 장재우
전북대학교 컴퓨터공학과

You-Jin Jang(yjjang@dblab.chonbuk.ac.kr), Ah-Reum Kim(kar@dblab.chonbuk.ac.kr),
Jae-Woo Chang(jwchang@chonbuk.ac.kr)

요약

유비쿼터스 컴퓨팅 시대의 핵심 기반 기술로서 무선 센서 네트워크 통신 기술에 대한 연구가 점차 확산되고 있다. 특히 센서 노드는 에너지 자원이 한정되어 있기 때문에, 에너지 효율적인 라우팅 기법을 통해 통신 에너지를 감소시키는 것이 가장 중요하다. 기존의 클러스터 기반 라우팅 프로토콜은 헤더를 임의로 선정함으로써 효율적으로 클러스터 헤더를 선정하지 못하는 문제점을 지닌다. 또한 기존의 클러스터 기반 라우팅 프로토콜은 대규모 크기의 네트워크를 지원하지 못하기 때문에 다양한 응용에서 활용될 수 없다. 따라서, 본 논문에서는 기존 클러스터 기반 라우팅 기법의 문제점을 해결하기 위하여, 대표경로를 이용한 새로운 클러스터 기반 라우팅 프로토콜을 제안한다. 제안하는 기법은 홉 수 기반의 대표경로를 생성함으로써 분산된 클러스터 헤더를 지닌 효율적인 클러스터를 구성한다. 또한, 클러스터 멤버 노드와 클러스터 헤더간의 데이터 통신 및 클러스터 헤더간의 통신을 위하여 멀티 홉 통신을 지원한다. 마지막으로 LEACH와 Mutihop-LEACH와의 성능 비교를 통해 제안하는 기법이 신뢰성 및 확장성 측면에서 성능이 우수함을 제시한다.

■ 중심어 : | 라우팅 프로토콜 | 센서네트워크 | 클러스터기반 라우팅 프로토콜 | 클러스터 헤더 선정 |

Abstract

A wireless sensor network communication technique has been broadly studied with continuous advances in ubiquitous computing environment. Especially, because the resource of the sensor node is limited, it is important to reduce the communication energy by using an energy-efficient routing protocol. The existing cluster-based routing protocols have a problem that they cannot select a cluster head efficiently by randomly choosing a head. In addition, because the existing cluster-based routing protocols do not support the large scale of network, they cannot be used for various applications. To solve the above problems, we, in this paper, propose a new cluster-based routing protocol using representative paths. The proposed protocol constructs an efficient cluster with distributed cluster heads by creating representative paths based on hop count. In addition, a new routing protocol supports multi-hop routing for data communication between a cluster member node and a cluster head as well as between cluster heads. Finally, we show that our protocol outperforms LEACH and Multihop-LEACH in terms of reliability and scalability.

■ keyword : | Routing Protocol | Sensor Network | Cluster-based Routing Protocol | Selection of Cluster Header |

* 본 연구는 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임.

* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. 2010-0000202)

접수번호 : #100219-001

심사완료일 : 2010년 05월 10일

접수일자 : 2010년 02월 19일

교신저자 : 장재우, e-mail : jwchang@chonbuk.ac.kr

I. 서론

무선 센서네트워크 기반의 응용 프로그램은 설치가 용이하고 관리비용이 저렴하여 상황인식 및 다양한 응용 서비스를 지원할 수 있다. 이러한 유비쿼터스 컴퓨팅 시대의 핵심 기반 기술로써 무선 센서 네트워크 통신 기술에 대한 연구가 점차 확산되고 있다. 무선 센서 네트워크 통신을 지원하기 위해서는 먼저 센서 노드, 싱크 노드 그리고 서버를 통한 네트워크 구축이 필요하다. 센서 노드는 관찰 지역 내의 환경이나 상황 정보를 수집하여 싱크 노드로 전송하고, 싱크 노드는 각 센서 노드에서 수집한 데이터를 수신하여 서버로 전송한다. 서버는 응용시스템을 통해 사용자에게 수집된 정보를 제공한다.

특히 무선 센서 네트워크 기반 응용 시스템에서 주로 사용되는 소형 센서 노드는 에너지 자원이 한정되어 있기 때문에, 응용 서비스를 안정적으로 지원하기 위해서는 에너지 효율성을 높이는 것이 필수적이다. 센서 노드의 에너지 소비는 크게 수집 에너지, 통신 에너지, 데이터 처리 에너지 세 부분으로 구성되는데, 통신에너지가 가장 큰 비중을 차지한다. 이때 통신 에너지는 전송하는 통신 거리의 제곱에 비례하여 증가한다[1]. 따라서 무선 센서 네트워크에서는 에너지 효율적인 라우팅 기법을 통해 통신 에너지를 감소시키는 것이 가장 중요하다[2]. 이를 위해 메시지의 통신량과 홉수를 감소시키기 위한 클러스터링 기반 라우팅 기법이 제안되었다. 하지만, 기존의 연구들은 클러스터 헤더를 임의로 선정함으로써 일정 지역에 배치된 각 노드의 위치를 고려하지 못하기 때문에 효율적인 헤더를 선정하지 못하는 문제점을 지닌다. 또한 소형 센서 노드가 지니는 한정된 통신 범위를 고려하지 않기 때문에, 네트워크 내의 모든 센서 노드가 서로 통신이 가능한 소규모 지역에 한정된 네트워크만을 지원하며, 다양한 응용에서 활용할 수 없다.

따라서, 본 논문에서는 기존 클러스터 기반 라우팅 기법의 문제점 해결을 위하여, 대표경로를 이용한 클러스터 기반 라우팅 프로토콜을 제안한다. 제안하는 기법은 홉 수 기반의 대표경로를 생성함으로써 분산된 클러스터

헤더를 선정하여 효율적인 클러스터를 구성한다. 또한, 클러스터 멤버 노드와 클러스터 헤더간(cluster member-cluser header)의 데이터 통신 및 클러스터 헤더간의 통신을 위하여 멀티 홉 통신을 지원한다.

본 논문의 구성은 다음과 같다. 2장에서는 대표적인 라우팅 프로토콜 기법에 대해 분석한다. 다음으로 3장에서는 대표 경로를 이용한 클러스터 기반 라우팅 알고리즘에 대해 상세히 기술한다. 4장에서는 다양한 환경과 조건에서의 실험을 통해 알고리즘을 검증한다. 마지막으로, 5장에서는 결론 및 향후 연구에 대하여 기술한다.

II. 관련 연구

무선 센서 네트워크 환경에서 센서 노드의 에너지 효율성을 높이기 위한 다수의 연구가 진행되었다[3]. 특히, 클러스터링 기반 라우팅 기법은 클러스터 단위로 네트워크를 관리함으로써 통신 메시지의 양과 홉수를 감소시켜 센서 노드의 수명을 연장시킨다. 클러스터링 기반 라우팅 기법은 다수의 센서 노드를 하나의 도메인(domain)으로 구성하여 하나의 클러스터를 형성함으로써, 전체 네트워크를 수개의 클러스터 단위로 관리한다. 각 클러스터를 대표하는 대표 노드를 선택하여 클러스터 헤더로 선정하고, 클러스터 헤더는 자신의 클러스터 영역에 포함된 클러스터 멤버 노드로부터 데이터를 수집 및 집계 처리하여 싱크 노드로 전송한다.

이러한 클러스터 기반 계층적 라우팅 기법은 LEACH(Low-Energy Adaptive Clustering Hierarchy)[4], LEACH-C(Centralized LEACH)[5], Multihop-LEACH[6]가 대표적이다.

1. LEACH 및 LEACH-C 프로토콜

A. Woo와 D. Culler가 제안한 LEACH의 수행 흐름도는 [그림 1]과 같다. 첫째, 각 센서 노드 및 싱크 노드를 초기화하여 라우팅을 구성하기 위한 준비 단계를 수행한다. 둘째, 임의의 클러스터 헤더를 선정한다. 셋째, 클러스터 단위로 네트워크를 구성하기 위하여

RSS(Received Signal Strength)를 통해 각 센서 노드에서 가까운 헤더 노드를 탐색한다. 마지막으로, 주기적으로 헤더를 재 선출한다.

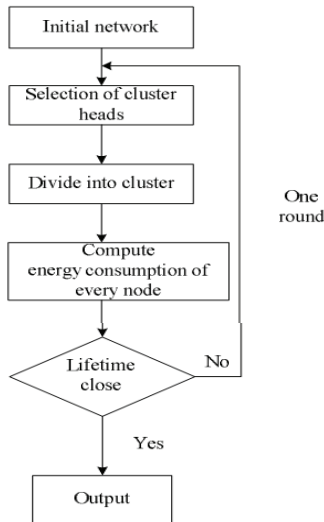


그림 1. LEACH의 수행 흐름도

[그림 2]는 LEACH의 경로 설정 예이다. 먼저 임의로 클러스터 헤더(붉은 색 노드)를 선정한다. 선정된 헤더 노드 이외의 센서 노드들은 자신과 근접한 헤더 노드를 선택하여 클러스터를 구성한다. 클러스터 구성 완료 후, 클러스터에 포함된 모든 센서 노드는 자신이 속한 클러스터의 헤더에게 메시지를 전송하고, 클러스터 헤더는 전송받은 데이터를 수집하여 싱크 노드(푸른 색 노드)로 전송하는 경로를 생성한다.

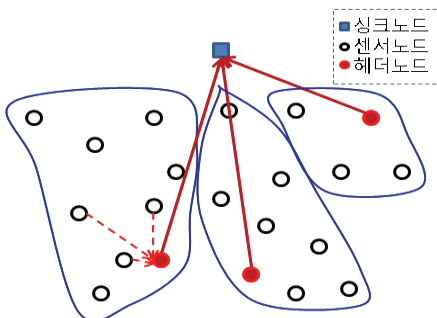


그림 2. LEACH의 전송 경로 예

이때 발생하는 헤더 노드의 통신 오버헤드를 줄이기 위하여 식 (1)을 통해서 클러스터를 헤더를 주기적으로 교체한다.

$$T(n) = \begin{cases} \frac{P}{1 - P(r \bmod \frac{1}{P})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

(1)에서 P 값은 전체 노드 수에 대한 클러스터 헤더의 수에 대한 백분율을 나타낸다. 노드에서 0에서 1사이 임의로 선택된 T 값이 (1)에 의한 임계값(threshold) 이하이면 현재 라운드의 헤더로 선정된다. 클러스터 헤더를 주기적으로 클러스터 내의 다른 노드로 교체함으로써 클러스터 헤더가 가지는 통신 오버헤드를 분산시킨다. 집합 G 는 $1/P$ 라운드 동안 클러스터 헤더로 선정되지 않은 노드들의 집합이다. 따라서 라운드가 증가함에 따라 모든 노드는 클러스터 헤더로 선출되기 때문에, 헤더로 선정된 특정 센서 노드로의 집중된 에너지 소비를 분산시켜 에너지를 균등하게 사용할 수 있다. 하지만 LEACH에서 클러스터 헤더의 임의 선출 방식은 헤더의 위치가 편중되기 때문에, 효율적인 경로를 구성하지 못하는 문제점을 지닌다.

이러한 문제점을 해결하기 위해 W. B. Heinzelman이 제안한 LEACH-C는 각 센서 노드의 위치 정보를 이용하여 중앙 처리 방식으로 확장한 방법이다. LEACH-C 알고리즘은 다음과 같다. 첫째, 각 센서 노드는 자신의 위치 정보를 싱크 노드에게 전송한다. 싱크 노드는 전송받은 위치 정보를 이용하여 전체 네트워크 영역을 클러스터 수만큼 분할하고, 각 영역에서 헤더를 선정하여 각 센서 노드에게 알린다. 둘째, 헤더 이외의 센서 노드는 자신과 가장 근접한 헤더 노드를 선택하여 메시지를 전송한다. 마지막으로 헤더 노드는 헤더-싱크 사이의 전송 경로를 연결함으로써, 전체 메시지 전송 경로 설정이 완료된다. [그림 3]은 LEACH-C 알고리즘을 통해 구성된 전송 경로를 나타낸다. 이 기법은 모든 센서 노드의 위치 정보를 알고 있기 때문에, 효율적인 헤더를 선택하여 편중된 클러스터가 생성되는 것을 방지하여 특정 클러스터 헤더의 에너지 소모를 줄이는 장점을 가진다. 반면, 각 노드의 정확한 위치 정보가 요구되기 때문에, GPS등의 위치 측정 장비가 추가적으로 필요하다.

또한 LEACH 및 LEACH-C는 선정된 헤더 노드와 싱크 노드간 통신에 대해 멀티 홉을 지원하지 않는 문제점이 존재한다. 이는 네트워크 내의 모든 센서 노드가 서로 통신이 가능한 소규모 지역에 한정된 네트워크만을 지원하기 때문에 다양한 응용 서비스를 지원하지 못한다.

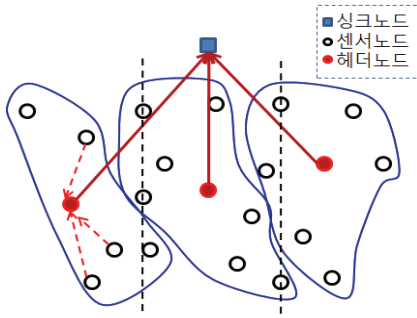


그림 3. LEACH-C의 전송 경로 예

[그림 5]는 Multihop-LEACH의 경로 구성 예이다. LEACH와 동일한 방법으로 클러스터 헤더를 선정하고, 클러스터 헤더가 아닌 센서 노드들은 헤더 노드로 데이터를 전송하는 경로를 생성한다. 헤더 노드는 싱크 노드 방향에 위치하는 헤더 노드를 통해 싱크 노드로 메시지를 전송하는 멀티 홉 경로를 구성한다.

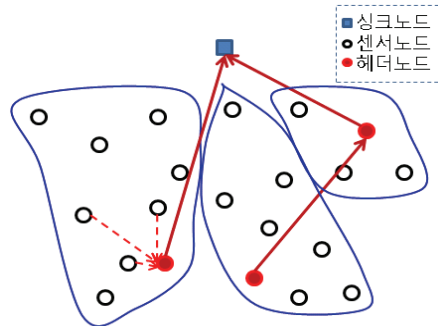


그림 5. Multihop-LEACH의 경로 구성 예

2. Multihop-LEACH

F. Xiangning와 S. Yulin이 제안한 Multihop-LEACH는 LEACH 및 LEACH-C의 헤더 노드와 싱크 노드간 통신에 멀티 홉을 지원하지 않는 문제점을 해결한다. 이 기법은 LEACH의 알고리즘을 바탕으로 임의로 생성된 클러스터 헤더로 클러스터를 구성하고, 헤더 노드와 싱크 노드 간의 멀티 홉 구성을 지원한다([그림 4]).

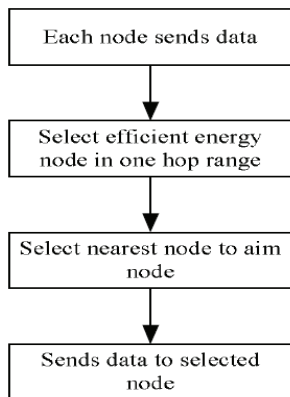


그림 4. Multihop-LEACH의 라우팅 구성

Multihop-LEACH는 헤더 노드에서 싱크 노드 사이의 멀티 홉의 경로를 구성하여, 비용을 많이 소모하는 장거리 전송을 피하는 장점을 가진다. 반면에, 클러스터 구성 시에 LEACH와 동일한 방법으로 클러스터 헤더를 임의로 선출하기 때문에 헤더의 위치가 편중되어 효율적인 경로를 구성하지 못한다. 또한, 클러스터 내의 멀티 홉을 지원하지 않기 때문에 클러스터 멤버 노드에서 헤더로의 에너지 소모가 큰 장거리 전송이 필요한 경우가 존재한다.

III. 대표 경로를 이용한 클러스터 기반 라우팅 프로토콜 알고리즘

대표적인 클러스터 기반 라우팅 기법 연구인 LEACH, LEACH-C와 Multihop-LEACH는 에너지를 효율적으로 사용하기 위해 클러스터를 구성하고 클러스터 헤더를 주기적으로 교체한다. 하지만 모든 센서 노드가 클러스터 헤더 후보 집합으로 설정되어, 센서 노드의 위치를 고려하지 않고 임의로 선출하기 때문에 클러스터 헤더의 위치가 편중될 가능성이 높다. 센서

노드의 에너지 소비에서 가장 중요한 부분을 차지하는 통신 에너지는 노드 간 통신 거리의 제곱에 비례한다 [1]. (2)는 센서 노드의 통신 에너지를 구하는 식이다.

$$E_{Tx}(k, d) = E_{Tx_elec}(k) + E_{Tx_amp}(k, d) \\ = E_{elec} * k + E_{amp} * k * d^2 \quad (2)$$

따라서 헤더 노드가 균일하게 분포되도록 선출함으로써 통신 거리를 감소시킬 수 있다. 또한 기존 연구 중 LEACH와 LEACH-C는 선정된 헤더 노드가 싱크 노드와 직접 통신 가능해야 하기 때문에 센서 노드와 싱크 노드의 라우팅 설정에 있어서 멀티 홉을 지원하지 않는다. 아울러, Multihop-LEACH는 헤더와 싱크 노드 사이의 멀티 홉을 지원하지 않지만, 클러스터 내의 멤버 노드와 헤더 노드 사이의 멀티 홉을 지원하지 않기 때문에 노드 간 통신 반경을 바탕으로 넓게 배치되는 실제 응용 시스템에 적용할 수 없다.

따라서 에너지 효율적인 클러스터링 라우팅 프로토콜을 설계할 시 고려사항은 다음과 같다. 첫째, 각 센서 노드의 통신 범위를 고려하여 홉 수 기반으로 네트워크를 구성한다. 둘째, 싱크 노드로부터 대표 경로를 통하여 효율적으로 클러스터 헤더가 분산 될 수 있도록 선출함으로써 에너지 효율성을 높인다. 셋째, 각 센서 노드는 싱크 노드와 직접 통신 할 수 없기 때문에 이웃 노드를 통해 통신 가능한 경로를 구성한다.

전체 노드에 대한 통신 경로를 구성하기 전에 먼저 싱크 노드에서 가장 멀리 있는 센서 노드까지 생성된 하나의 경로를 대표 경로라 명명한다. 대표 경로는 센서 노드의 밀도가 높고, 통신량이 많은 노드를 기준으로 경로를 생성한다. 이때 선정된 노드는 대표 경로를 구성하는 앵커 노드라 정의한다. 대표 경로를 통해 클러스터 헤더를 선정하고 클러스터링 함으로써 다음과 같은 장점을 가진다. 첫째, 싱크(sink) 노드 방향 및 이웃하는 노드들의 방향을 인식함으로써 우회하지 않는 효율적인 라우팅 경로를 구성한다. 둘째, 앵커(anchor) 노드 중에서 클러스터 헤더를 선정하기 때문에 네트워크에서 클러스터 헤더 위치 및 클러스터 영역이 균일하게 분포된다.

대표경로를 이용한 클러스터 기반 라우팅 프로토콜의 알고리즘은 크게 네트워크 정보 생성, 대표 경로 생

성, 클러스터 구성, 클러스터 관리의 4단계로 구성된다 ([그림 6]). 첫째, 네트워크 정보 생성 단계에서는 싱크 노드에서 메시지 전파(flooding)기법을 통해 각 노드의 홉 수와 이웃 노드를 설정한다. 둘째, 대표 경로 생성 단계에서는 센서 노드의 밀도가 높고 통신량이 많은 노드를 앵커 노드로 가지는 대표 경로를 생성한다. 셋째, 클러스터 구성 단계에서는 앵커 노드 중에서 클러스터 헤더 노드를 선정하고 클러스터 멤버를 구성한다. 마지막으로 클러스터 관리 단계에서는 주기적인 헤더 교체 기법과 장애 복구를 위한 전체 클러스터 재구성능을 수행한다.

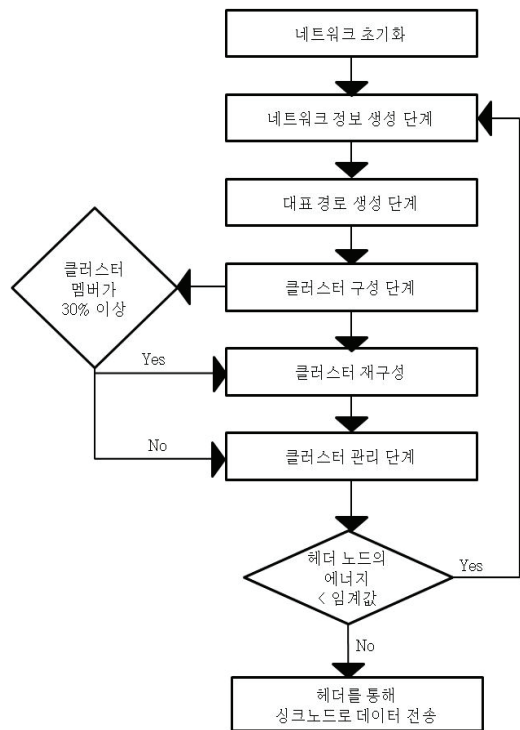


그림 6. 대표경로를 이용한 라우팅 프로토콜 알고리즘

1. 네트워크 정보 생성 단계

먼저 각 센서 노드에 대해 메시지 전파(flooding)기법 [1]을 통하여 싱크 노드로부터의 홉 수를 설정하여 네트워크 정보 테이블을 생성한다. 네트워크 정보를 생성하는 알고리즘은 [그림 7]과 같다. 네트워크 정보 생성 알고리즘은 각 센서에서 네트워크 정보를 생성하여 싱

크 노드로 전송하기 위해 센서 노드에서 수행하는 부분(1~22)과 네트워크 테이블 생성 및 홉카운트 메시지 최초 플러딩을 수행하는 싱크 노드 부분(25~31)으로 구성된다. 수신한 메시지에 홉 수가 설정되어 있으면(2), 현재 센서 노드에 설정된 홉수와 비교하여 더 작은 값일 경우 홉 수를 갱신하고(4~6), 메시지를 전송한 이웃 노드를 부모 노드로 저장한다. 갱신된 홉 수 정보로 메시지를 플러딩시킨다(10). 또한 설정된 네트워크 정보는 부모 노드를 통해 싱크 노드로 전송한다(11~15). 싱크 노드는 홉카운트를 설정하기 위하여 최초로 플러딩 메시지를 전송하고(26), 각 센서 노드에서 보내온 메시지를 통해 네트워크 테이블을 생성한다(28~30).

Sensor Node MakeNetInfo

input : msg //노드 정보를 담은 메시지
msgType //메시지의 타입

1. If msgType = HopCount then
2. Set parentID, recHopCnt from msg
3. if(curHopCnt > recHopCnt + 1)
4. curHopCnt = recHopCnt + 1
5. else
6. break;
7. Parents.add(parentID)
8. Flooding(curHopCnt, nodeID)
9. NetInfo.id=nodeID
10. NetInfo.curEntry = 0
11. NetInfo.Parent[NetInfo.curEntry]=parentID
12. NetInfo.HopCnt[NetInfo.curEntry]=-1
13. Send Message(parentID, NetInfo)
14. If msgType = NetInfo then
15. Set NetInfo from msg
16. NetInfo.curEntry++
17. NetInfo.parent[NetInfo.curEntry]
 =Parents.First
18. NetInfo.parent[NetInfo.curEntry]
 =NetInfo.HopCnt[NetInfo.curEntry]-1
19. Send Message(Parents.First, NetInfo)

Sink Node MakeNetInfo

1. Flooding(initHopCnt, sinknodeID)
2. wait until receiving message
3. If Receive Message(NetInfo) from Sensor Node
4. For each entry i of NetInfo
5. NetInfoTable[NetInfo.id][parent[i]]=
 NetInfo.HopCnt[i]

그림 7. 네트워크 정보 생성 알고리즘

예를 들어, 싱크 노드로부터 메시지를 받은 노드의 홉 수를 1로 설정하고, 홉 수가 설정된 노드는 플러딩 메시지를 전송한다. 홉 수가 1인 노드로부터 메시지를 수신한 센서 노드는 2홉으로 설정한다. 즉, 메시지를 수신한 센서 노드는 메시지를 전송한 센서 노드의 다음 홉 수로 설정한다. 이와 같은 과정을 반복 수행하면서 전체 센서 노드의 홉 수를 설정한다. 여러 개의 센서 노드로부터 메시지를 중복하여 수신한 센서 노드는 가장 작은 홉 수를 저장하게 된다. 가장 작은 홉 수의 메시지를 보낸 이웃 노드를 부모 노드로 설정한다.

[그림 8]은 센서 노드 20개를 배치한 네트워크를 나타내며 싱크 노드로부터 최대 홉 수는 5이다. 이때 각 센서는 1홉 내에 통신 가능한 하나 이상의 센서 노드를 이웃 노드로 가진다. 이때 이웃 노드 중에서 싱크 노드로부터의 홉 수를 기준으로 자신보다 홉 수가 작은 노드를 in-coming 노드, 자신보다 홉 수가 큰 노드를 out-going 노드라고 정의한다. [그림 9]는 [그림 8]에 대해 싱크 노드로부터의 홉 수에 따른 노드 아이디와 이웃 노드의 수를 표현한 것이다. 예를 들어, 2홉에 있는 4번 노드를 기준으로 할 때, 1홉에 존재하는 1번 노드는 in-coming 노드이고 3홉에 존재하는 3, 7, 8번 노드는 out-going 노드이다. 홉 수 정보가 설정된 센서 노드는 부모 노드를 통해 싱크 노드로 네트워크 정보를 전송한다. 서버에서 이를 이용하여 멀티 홉으로 통신 가능한 모든 센서 노드 간 홉 수 정보를 생성한다.

[표 1]은 생성된 네트워크 정보 테이블의 예이다. 생성된 네트워크 정보를 통해 이웃 노드에 대한 정보를 알고 있기 때문에, 주변 이웃 노드를 탐색하기 위한 추가적인 통신비용이 소모되지 않는다. SN(Source Node)은 네트워크 테이블에서의 기준 노드의 식별자이고, DN(Destination Node)은 목적지 노드의 식별자를 나타낸다. 테이블의 값은 SN에서 DN까지 메시지를 전송하기 위해 필요한 홉 수와 방향성 정보를 저장한다. 싱크 노드로부터 홉 수는 싱크 노드와 멀수록 큰 값을 가지기 때문에, SN을 기준으로 DN이 싱크 노드 방향이면 음수 값, 싱크 노드 반대 방향이면 양수 값을 갖는다.

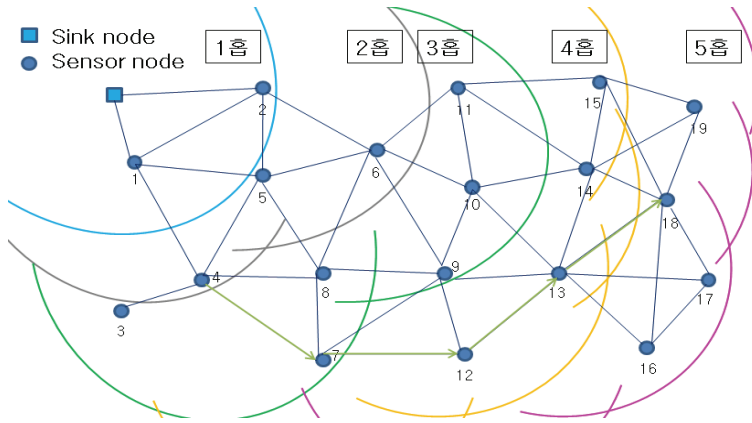


그림 8. 홉 수 설정 예제 (노드 20개)

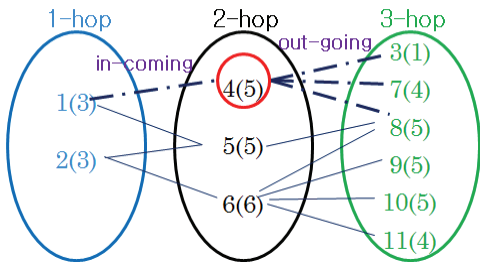


그림 9. 이웃 노드 정보 그래프 표현

표 1. 네트워크 정보 테이블 예제

SN \ DN	0	1	2	3	4	5	...
0	-	1	1	3	2	2	
1	-1	-	1	2	1	1	
2	-1	1	-	3	2	1	
3	-3	-2	-3	-	-1	-2	
4	-2	-1	-2	1	-	1	
5	-2	-1	-1	2	1	-	
...							

2. 대표 경로 생성 단계

대표 경로는 싱크 노드로부터 홉 수를 기준으로 생성하며, 최대로 생성 될 수 있는 대표 경로의 수는 1홉 내의 노드의 수이고, 노드가 편중되게 배치된 경우 최소 1개가 생성 될 수 있다. 설정된 이웃 노드 정보를 대표 경로 생성 시에는 out-going을 고려하여 효율적으로

메시지를 전파시킬 수 있도록 한다. 또한 클러스터 멤버에서 헤더로 메시지를 전송하는 라우팅 경로 구성 시에는 in-coming을 고려하여 싱크 노드 방향으로 효율적인 헤더를 선택할 수 있다. [그림 10]은 대표 경로를 생성하는 알고리즘을 보여준다. 첫째, 싱크 노드와 1홉 내에 노드 중에서 다수의 이웃 노드를 지나는 노드에서부터 다음 홉으로 더 이상 확장할 수 없을 때까지 대표 경로를 생성한다(1~2). 둘째, 다음 홉에 있는 이웃 노드 중, 이웃 노드의 수가 가장 많고(5), out-going 노드가 가장 많은 노드를 탐색한다(15~21). 셋째, 만일 기존 경로와 겹침이 발생하지 않는다면, 해당 노드로 확장하며 경로를 저장한다(10~13). 아울러, 두 개 이상의 대표 경로를 생성할 때에는 이미 생성된 대표 경로를 고려해야 한다. 대표 경로를 생성하기 위해 확장하려는 노드가 이미 생성된 경로의 앵커 노드와 1홉 이내이면, 그 노드를 제외한 노드 중에서 재탐색을 수행한다. 기존에 생성된 대표 경로 영역에서 클러스터 영역으로 3홉 이내에 포함된 노드는 멀티 홉으로 포함 가능한 영역이기 때문에 대표 경로 생성에서 제외된다. 마지막으로 재탐색 수행 시 확장을 중지한 노드와 싱크 노드로부터의 홉 수가 같은 이웃 노드 중에서 이웃 노드 수가 가장 많고 out-going 노드가 가장 많은 노드를 재탐색 하여 다시 확장을 계속한다(24~30).

```

Sink Node MakeRPPath
input : NetInfo //네트워크의 정보
1. For each node i within 1hop from Sink Node
2. CurrNode = i
3. MaxNN=0
4. res = FindNext(CurrNode,CurrHopCnt,NetInfo)
   //대표 경로를 구성하는 다음 노드 탐색
5. while res = TRUE and NN of CurrNode < NN of
   FindNode from NetInfo and CurrHopCnt < FindNodeCnt
   from NetInfo then
6.   CurrNode = FindNode
7.   MaxNN = NN of FindNode
8.   If CheckPath() = False then
9.     AddPath(*PathList[PathCnt])
10.    PathCnt++
11.   else
12.    ReFindNext(CurrNode,CurrHopCnt, NetInfo) //대표
   경로 구성하는 다음 노드 재탐색
    
```

그림 10. 대표 경로 생성 알고리즘

3. 클러스터 구성 단계

LEACH의 실험[4]에 따르면 클러스터 헤더의 수는 전체 센서 노드 수의 5% 이상으로 설정하고 수행한다. 가장 효율적인 클러스터 헤더를 선정하기 위해 각 대표 경로에 대해 클러스터 헤더 선정 시 비용을 미리 계산한다. 첫째, 클러스터 헤더 선정 위치에 따라 경로에 대한 통신비용이 달라지기 때문에 전체 대표 경로에 대해 선정될 수 있는 클러스터 헤더의 모든 조합을 고려한다.

```

Sink Node MakeCluster
input : NetInfo,
   *PathList[PathCnt] //대표 경로에 포함된
   노드들의 리스트
   NumPath //생성된 대표 경로 수
1. TotalHead = NumNodes*0.05
2. ChooseHead[TotalHead]
3. For each Path p from NumPath
4.   Combination(PathCnt, TotalHead, PathCnt,
   ChooseHead) //최적의 클러스터 헤더 선출을 위한 모든
   헤더 조합
5.   Comm = CalComm(ChooseHead) //클러스터 헤더 선
   출 시의 비용 계산
6.   If Comm < MinComm then
7.     MinComm = Comm
8.   FindNNHead(ChooseHead, NetInfo)
    
```

그림 11. 클러스터 구성 알고리즘

[그림 11]은 클러스터링 구성단계에 대한 알고리즘을 나타낸다. 먼저, 네트워크 정보 생성 단계에서 설정한 NetInfo 정보를 활용하여 대표경로 생성 단계에서 설정한 대표 경로를 기반으로 헤더를 선택한다(1~4). 다음, Combination 함수를 통해 발생할 수 있는 모든 헤더의 조합을 생성하고(12~22), 각 조합에 대한 통신 비용을 미리 계산한다. 통신 비용 계산시에는 추가적인 통신을 수행하지 않고 생성된 네트워크 정보 테이블인 NetInfo의 값을 이용하여 통신 비용을 계산한다(5~6). 마지막으로 통신비용이 가장 작을 때의 헤더 노드 정보를 통해 가까운 헤더로 클러스터를 구성한다(7~10).

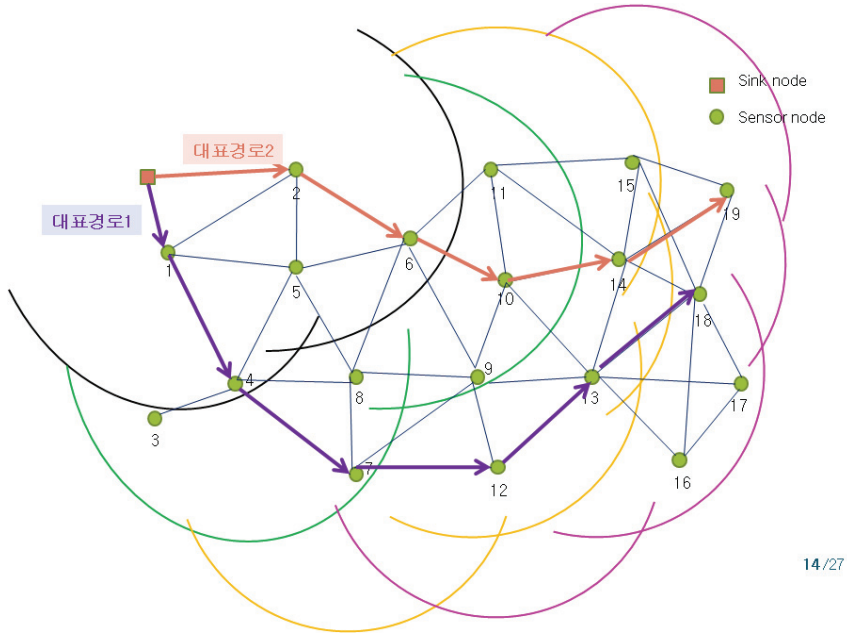
[그림 12]는 전체 2개의 대표 경로가 선정된 예제이고 앵커 노드의 노드 식별자를 나타낸다. 전체 노드의 수가 60개라면 그중 5%인 총 3개의 클러스터 헤더를 선출해야 한다고 할 경우, 각 대표 경로에 대해 선출할 수 있는 헤더에 대한 경우의 수는 {(3,0),(2,1),(1,2),(0,3)} 이다. 괄호의 첫 번째 원소는 대표 경로 1에 대해 선출할 수 있는 헤더의 수이고, 괄호의 두 번째 원소는 대표 경로 2에 대해 선출할 수 있는 헤더의 수이다. 즉 (2,1)은 헤더 노드의 수가 대표 경로 1에서 2개, 대표 경로 2에서 1개에 선출되는 조합을 의미한다. 즉, [그림 9]에서 1-4, 1-7, 1-12, 1-13, 4-7, 4-12..., 12-13 노드가 헤더 노드로 선택될 수 있는 경우이다. 또한 대표 경로 2에서는 5개 노드 중에서 1개의 헤더 노드를 선출하는 조합의 경우의 수이다. 대표 경로 1과 대표 경로 2에서의 헤더 선출에 대한 사건은 서로 독립 관계이다. 따라서 위의 예제에서 클러스터 헤더가 선정되는 조합에 대한 경우의 수는 다음과 같다.

$$(5C3 * 5C0) + (5C2 * 5C1) + (5C1 * 5C2) + (5C0 * 5C3) = 120$$

즉, 대표 경로가 두 개인 경우, 클러스터 헤더 선정 경우의 수는 다음과 같이 정의 할 수 있다.

$$C(x,y) = xCy = xCombination y$$

$$\sum_{n=0}^{CHcnt} C(anc - cnt_{route1}, CHcnt) \times C(anc - cnt_{route2}, n)$$



14/27

그림 12. 대표 경로 설정 예제

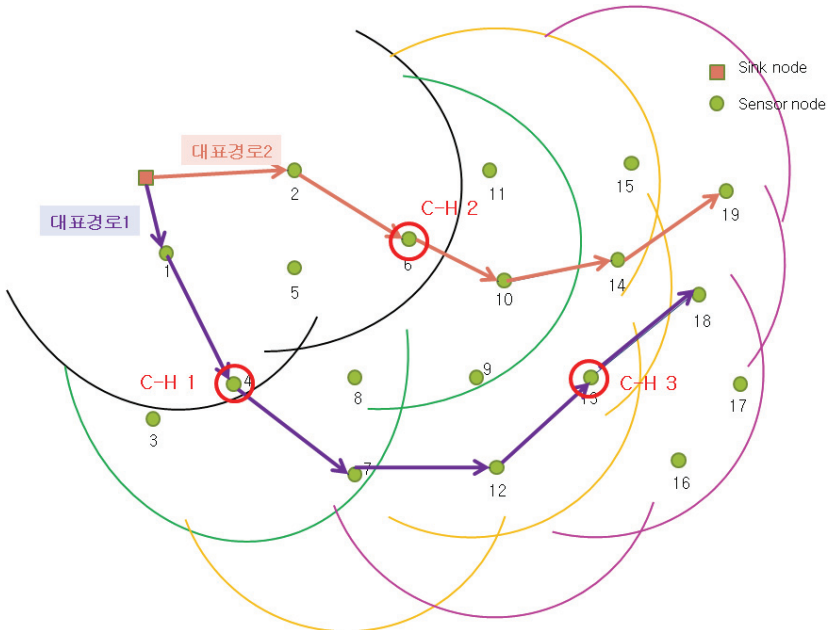


그림 13. 헤더 선출 예시 (3개의 헤더 노드 선출 : 4, 6, 13번 노드)

대표 경로 1을 구성하는 총 앵커 노드의 수는 $anc-cnt_{route1}$, 대표 경로 2를 구성하는 총 앵커 노드의 수는 $anc-cnt_{route2}$, CHcnt는 선출한 클러스터 헤더수를

나타낸다.

[그림 13]에서 각 앵커 노드 수가 5개인 2개의 대표 경로에서 3개의 클러스터 헤더를 선출하는 경우의 수

는, 총 120회의 경로 비용 산출이 필요하다는 것을 알 수 있다. 모든 조합을 고려하기 때문에 경로 비용 횡수가 많지만 최적의 클러스터 헤더를 선정할 수 있고, 각 센서 노드가 해당 클러스터 헤더에 포함될 때의 비용은 미리 계산된 네트워크 정보를 참조할 수 있기 때문에 간단히 계산할 수 있다. 또한 가장 작은 경로 비용 값을 임계값으로 가지치기를(pruning) 하여 계산 비용을 효과적으로 감소시킬 수 있다.

둘째, 각 클러스터 헤더에서 싱크까지의 전송비용은 대표 경로를 이용함으로써 경로를 재탐색을 수행해 하는 추가 비용 없이 구성할 수 있다. 해당 클러스터 헤더가 포함된 대표 경로 상에서 싱크 노드 방향으로 가장 가까운 앵커 노드를 중계 노드로 설정한다.

최종적으로, 대표 경로에 대한 통신비용은 각 센서 노드에서 선택된 클러스터 헤더 노드까지의 전송비용과 헤더에서 가장 가까운 중계 노드까지의 전송비용의 합이다. 각 앵커 노드 중 클러스터 헤더 선정에 대한 모든 조합을 고려하여 가장 통신비용이 가장 적을 때의 클러스터 헤더 조합을 선택한다. 모든 센서 노드는 가장 가까운 한 개의 클러스터 헤더 노드를 갖는다. 이때, 실험을 통해 전체 노드 수의 30%이상이 동일한 헤더 노드로 설정되어 부하가 발생할 경우, 인접하는 헤더 노드 중 클러스터 멤버를 적게 가지는 헤더 노드로 클러스터 병합 및 재분할을 수행하여 헤더 노드의 부하를 분산시킨다.

4. 클러스터 관리 단계

클러스터 헤더로 선정된 노드는 다른 노드에 비해 에너지 소모가 많다. 따라서 이를 주기적으로 교체함으로써 전체 네트워크 수명을 연장시킬 수 있다. 기존 LEACH에서는 클러스터 헤더를 임의로 선정하여 교체함으로써 헤더 노드의 위치가 편중되기 때문에, 에너지 소모량이 증가되는 문제점이 있었다. 하지만 본 논문에서는 대표 경로를 이용하여 클러스터 헤더를 균일하게 분포되도록 설정하였기 때문에, 그 주변 노드 중에서 헤더를 주기적으로 교체하는 기법을 통해 에너지 효율성을 높일 수 있다.

[그림 14]는 클러스터 관리 단계 알고리즘이다.

Header Update Interval로 설정된 임의의 헤더 변경 주기를 이용하여 현재 노드의 라운드를 체크하여, 변경 주기가 되면 헤더를 변경한다(2~5). 노드의 에너지가 일정 임계값 이하로 내려가면 라우팅 테이블을 재구성하는 메시지를 전송한다(6~9).

Sensor Node ManageCluster

input : NetInfo

9. RoundCheck = CurrRound%/Header Update Interval
10. If RoundCheck = 0 then
11. Header = ChangeHeader()
12. If Energy of Header < Energy of Threshold then
13. Type of Message = ReConstruction of Routing Table;
14. MakeNetInfo(Message)

그림 14. 클러스터 관리 알고리즘

대표 경로의 통신비용 계산을 통해 결정된 최적의 클러스터 헤더와 1홉 내인 주변 노드로 일정 라운드마다 교체함으로써 에너지를 보다 균일하게 소모시킬 수 있다. 또한 교체된 헤더 노드의 에너지 잔여량이 임계값 이하일 경우 전체 클러스터를 재구성함으로써 장애 발생 시 복구를 지원한다. 전체 클러스터 재구성시 대표 경로 생성은 에너지 잔여량을 고려하여 확장함으로써 에너지 소모가 적은 센서 노드를 클러스터 헤더로 선정하여 에너지 효율성을 높일 수 있다. 클러스터 재구성 시에도 30% 이상 편중이 발생한 헤더 노드는 인접하는 헤더 중 클러스터 멤버 노드의 편중이 적은 헤더 노드를 선택하여 재분배한다.

IV. 성능평가

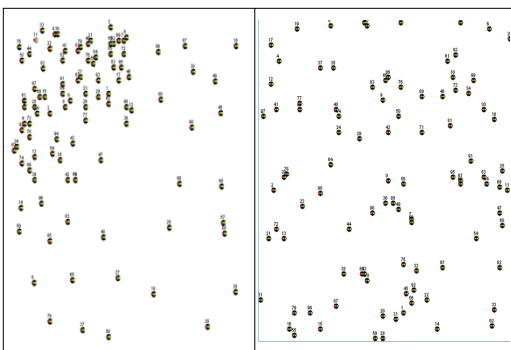
1. 실험환경

제안하는 대표 경로를 이용한 클러스터링 기반 라우팅 프로토콜의 성능을 검증하기 위하여 가장 대표적인 TinyOS[7] ver.1.1.15 기반의 이벤트 구동식(Event Driven) 언어인 NesC (network embedded systems C) 언어로 구현하였고, TinyOS에서 제공하는 무선 센서 네트워크 전용 시뮬레이터인 TOSSIM[8]을 통해 성능

평가를 수행하였다. 성능평가에 사용된 어플리케이션은 TinyOS에서 기본 제공하는 Surge 어플리케이션으로써, 모든 센서 노드가 매 주기마다 메시지를 생성하고, 이를 싱크노드에 전송하는 간단한 질의를 수행한다. 성능평가는 100X100m의 정방형의 네트워크 영역에서 총 노드 수 100개를 가지고 수행하였다. 아울러 다양한 상황에 대한 성능을 평가하기 위해서, 싱크 노드의 위치 및 전체 센서 노드의 배치를 변경하며 실험하였다. 먼저 싱크 노드의 위치는, 첫째, 네트워크의 중심에 있는 경우와 둘째, 네트워크의 바깥에 있는 경우를 실험하였다. 다음 전체 센서 노드의 배치는, 임의의 배치(random), 비대칭형 배치(skewed), 격자 배치(uniform)의 세가지 배치 유형을 사용하였다. [표 2]는 성능 평가에 사용된 노드 배치의 유형과 특징을 나타낸다. [그림 15]는 TinyOS에서 제공하는 GUI인 TinyViz를 이용하여 [표 2]의 노드 배치 유형을 나타낸 것이다. 한편, 성능 평가 요소는 클러스터당 멤버수 편차, 클러스터 멤버 노드 간 통신거리, 노드 평균 수명, 전체 네트워크 수명 및 센서 노드 통신 거리 변화에 의한 클러스터 구성 여부로써, 이를 통해 구성된 클러스터의 효율성을 측정한다.

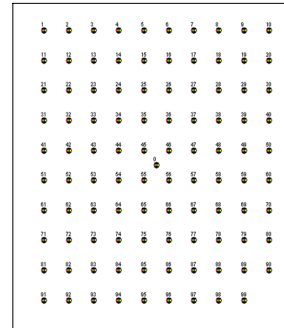
표 2. 노드 배치의 유형

유형	특징
skewed	노드의 70% 이상이 특정 지역에 밀집된 배치
random	노드를 전체적으로 임의로 배치
uniform	노드를 일정한 격자 형태로 배치



(a) skewed

(b) random



(c) uniform

그림 15. 노드의 배치 예제

LEACH-C는 성능평가에서 배제한다. 그 이유는 본 논문에서 제안하는 클러스터 기반 라우팅 기법은 저가의 센서 노드를 기반으로 GPS를 사용하지 않는 라우팅 기법이기 때문에, GPS를 이용하여 위치를 측정하는 LEACH-C는 제외한다.

가장 대표적인 클러스터 기반 라우팅 프로토콜인 LEACH, Multihop-LEACH에 대해 제안하는 알고리즘(Clustering Based routing protocol using Representative Path: CBRP)과 비교 분석한다.

2. 노드 배치 유형에 따른 클러스터 성능평가

효율적인 클러스터 구성 및 헤더 선출을 검증하기 위하여, [그림 16][그림 17]와 같이 클러스터 당 멤버 노드 수의 편차, 클러스터 멤버 노드의 통신 거리를 측정하였다.

[그림 16]은 각 클러스터에 포함되는 클러스터 멤버 노드 수에 대한 편차를 나타낸다. LEACH의 경우 표준편차가 평균 약 9.54, Multihop-LEACH는 약 9.48으로 나타나는 반면, 제안하는 알고리즘 약 8.07로 기존 기법에 비해 작은 표준편차를 보인다. 그 이유는 제안하는 알고리즘이 생성된 대표 경로 상에 존재하는 앵커 노드를 기반으로 클러스터 헤더 노드를 선출하기 때문에 균일하게 헤더가 선출되어, 각 클러스터가 수직 및 집계 처리해야 할 클러스터 멤버 노드의 수가 다른 프로토콜에 비해, 균일하게 분포되기 때문이다.

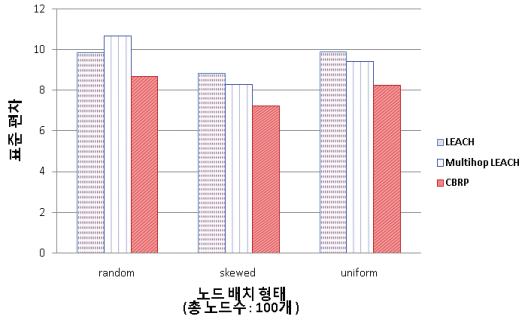


그림 16. 클러스터당 멤버 노드수의 편차

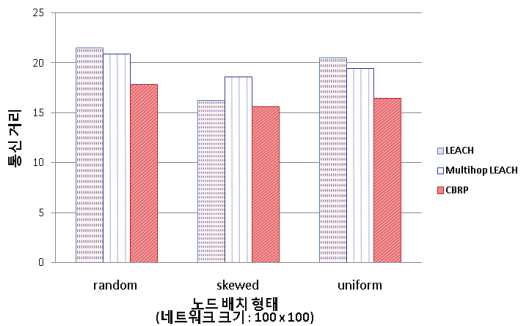


그림 17. 클러스터 멤버 노드당 통신거리

[그림 17]은 각 클러스터 멤버 노드의 통신 거리를 나타낸다. 센서 노드의 총 에너지 자원은 대기 : 수신 : 송신 이 1 : 2 : 2.05 로써, 송수신 에너지가 전체 에너지의 80% 이상을 차지한다. 특히, 센서 노드가 소모하는 통신 에너지는 통신거리의 제곱에 비례하기 때문에 효율적인 위치에 존재하는 헤더를 선정함으로써 에너지 효율성을 높일 수 있다. LEACH의 경우는 평균 19.47m, Multihop-LEACH의 경우는 19.68m인 반면, 제안하는 알고리즘은 16.67m로 약 3m의 통신 거리를 감소시킨다. 이는 LEACH와 Multihop-LEACH는 모두 헤더를 임의로 선출하고, 헤더 노드가 편중된 위치로 선출되는 경우가 많아 각 클러스터 멤버 노드 당 통신 거리가 긴 반면에, 제안하는 알고리즘은 클러스터 헤더 선정 시 네트워크 정보를 생성하여 미리 통신비용을 계산하고 최적의 헤더를 선택하기 때문에 각 클러스터 멤버 노드가 전송해야 할 통신 거리가 짧다. 또한 멀티 홉을 지원하기 때문에 가까운 이웃 노드를 통해 멀티 홉으로 구성되기 때문에, 노드 배치가 편중된 skewed 데이터의

경우에도 LEACH는 16.30m, Multihop-LEACH는 18.64m인 반면, 제안하는 알고리즘은 15.64로 기존 기법보다 짧은 통신 거리를 지닌다.

3. 노드 배치 유형에 따른 에너지 효율성 성능평가

한정된 에너지 자원을 가지는 센서 노드를 지원하기 위하여 [그림 18][그림 19]와 같이 각 센서 노드의 평균 수명과 네트워크 수명을 측정하였다. 초당 1회씩 데이터를 수집하도록 질의가 설정되어 있을 때, 전체 센서 노드가 싱크 노드로 데이터를 전송하는 1주기가 1round이다. [그림 18]은 센서 노드에 대한 평균 수명을 나타낸다. LEACH의 경우 노드의 평균 수명은 55.85round인 반면에, Multihop-LEACH는 78.99round이다. 이는 Multihop-LEACH의 경우 클러스터 헤더 노드에서부터 싱크 노드까지 멀티 홉을 지원하기 때문에, LEACH에 비해 노드 평균 수명이 약 30% 증가하였다. 제안하는 알고리즘은 노드의 평균 수명이 약 142.15round로, 앞서 실험한 것에서도 알 수 있듯이 클러스터당 멤버 노드의 수가 균일하게 분포되고, 각 멤버 노드당 통신거리가 짧기 때문에 노드의 평균 수명이 LEACH에 비해서 약 3배, Multihop-LEACH에 비해 약 2배 증가함을 알 수 있다.

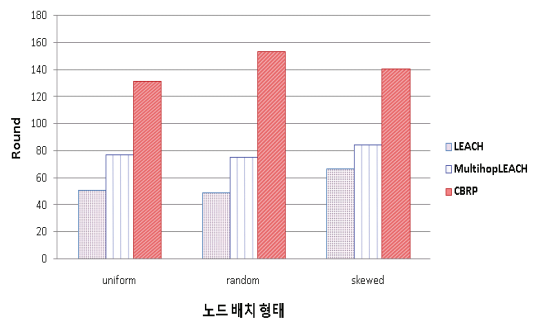


그림 18. 노드 평균 수명

[그림 19]는 네트워크 수명을 측정 한 그래프이다. 배터리가 소진된 노드수가 50% 이상일 경우는 센서 노드의 데이터 수집이 원활하게 이루어질 수 없기 때문에 해당 네트워크는 유효하지 않다[9]. 따라서 전체 센서 노드의 50% 이상이 배터리가 모두 소진되어 재기능을

수행하지 못하는 경우를 실험하였다. LEACH와 Multihop-LEACH는 50round 이전에 전체 센서 노드 수 20%가 배터리가 소진된다. 하지만 제안하는 알고리즘은 130round에서 전체 센서 노드 수 20%가 배터리가 소진되는 것을 알 수 있다. 네트워크가 유효하지 않은 정상적으로 동작하는 센서 노드가 전체 노드의 50%일 경우에 제안하는 알고리즘은 150round, Multihop-LEACH는 100round, LEACH는 60round 이다. 이는 제안하는 알고리즘이 노드의 평균 수명 뿐만 아니라 네트워크 수명도 150%이상 성능 향상을 나타냄을 알 수 있다.

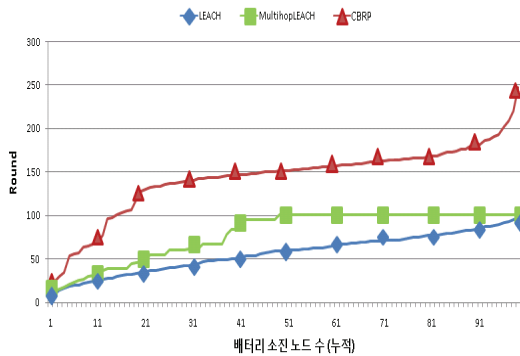


그림 19. 네트워크 수명

4. 센서 노드 통신 범위 변화에 따른 통신 가능 노드 비율 성능평가

각 센서 노드의 통신 범위는 하드웨어적인 요소나 주변 환경의 영향을 받기 때문에 0(최소), 10, 15, 25, ..., 150(최대)로 변화시키면서 실험하였다. 이때 네트워크 필드의 크기는 100X100이다. [그림 20]은 각 센서 노드의 통신 가능한 범위를 ‘통신 가능한 노드 수/전체 센서 노드’ 로 측정하였다. 센서 노드의 통신범위가 15으로 작은 범위 일 경우, LEACH는 3%, Multihop-LEACH는 2%의 노드 만이 통신 가능한 반면, 제안하는 알고리즘은 홉 기반의 멀티 홉으로 경로가 구성되기 때문에 60%이상 통신이 이루어진다. Multihop-LEACH는 각 클러스터 헤더와 싱크 노드까지의 통신에만 멀티 홉을 지원하기 때문에 LEACH보다 통신범위가 넓지만 통신 범위 40이상 되어야 전체 센서 네트워크의 50% 영역만

처리할 수 있다. 반면에, 제안하는 알고리즘은 센서 노드의 통신범위가 전체 네트워크 필드의 1/4(25/100)이 되면 전체 네트워크에 대한 통신이 가능하다.

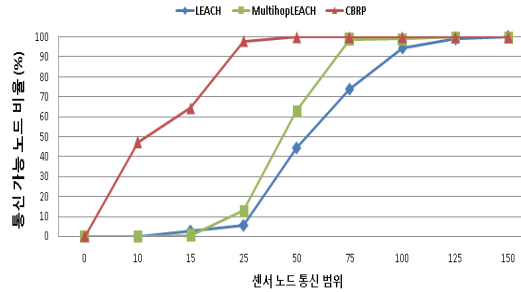


그림 20. 각 센서 노드의 통신 가능 범위

응용 서비스에서 측면에서 살펴보면, 첫째 일반적으로 사용하는 센서 노드의 평균 통신거리는 15~20m를 나타내며, 제안하는 라우팅 기법의 평균 통신거리가 17m이기 때문에 실제 응용에 유용하게 사용될 수 있다. 둘째, 제안하는 라우팅 기법은 짧은 통신 범위를 지닌 센서 노드에서도 전체적으로 네트워크 커버가 가능하기 때문에, 대규모 영역을 관리하는 감시 응용 서비스 지원에 적합하다. 마지막으로 제안하는 라우팅 기법은 효율적인 경로 구성과 주기적인 헤더 노드의 교체를 통해 에너지 효율성을 향상시켰기 때문에, 동적인 헤더 변경을 필요로 하는 응용 시스템에 적합하다.

V. 결론

본 논문에서는 다양한 무선 센서 네트워크 기반 응용 서비스를 지원할 수 있도록 센서 노드의 통신 범위를 고려하여 홉 수 기반의 네트워크 정보를 구성하고, 싱크 노드에서부터 마지막 홉에 이르는 최적의 경로를 설정하여 대표 경로로 정의하였다. 대표 경로를 생성함으로써 자질이 높은 클러스터 헤더를 균등하게 분포되도록 설정하고, 싱크 노드 방향 및 이웃하는 노드들의 방향을 인지함으로써 효율적인 클러스터링을 구성하였다. 이는 기존 라우팅 프로토콜이 가지는 문제점을 해결하여 한정된 에너지를 효율적으로 사용하고, 멀티 홉

통신을 통해 한정된 통신 범위를 극복할 수 있도록 설계 및 구현하였다. 아울러 성능평가를 수행하여 제안하는 기법의 효율성을 검증하였다. 성능평가 결과, 클러스터 멤버 노드 수에 대한 표준 편차가 약 15% 감소하여 클러스터가 다른 프로토콜에 비해 균일하게 분포되었고, 에너지 효율성 측면에서 센서 노드의 평균 수명이 LEACH에 비해 약 3배, Multihop-LEACH에 2배 증가하였고, 네트워크의 수명이 1.5배 이상 향상됨을 보였다. 또한, 통신 범위 변화에 따른 통신 가능 노드 비율에서 작은 범위에서도 전체 네트워크에 대한 통신을 수행할 수 있음을 보였다.

향후 연구로는 설계 및 구현한 라우팅 프로토콜을 실제 무선 센서 네트워크 플랫폼에 적용하여, 응용 서비스 지원시의 효율성을 검증하는 것이다.

참고 문헌

- [1] Y. Panthacha and P. Keeratiwintakorn, "An energy model for transmission in Telos-based wireless sensor networks," International joint conference on computer science & software engineering, 2007.
- [2] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," Ad Hoc Networks, Vol.3, No.5, pp.352-349, 2005.
- [3] 김봉수, 황소영, 정훈, 주성순, "유비쿼터스 센서 네트워크 라우팅 기술", 한국 인터넷 정보학회, 제9권, 제1호, pp.13-21, 2008.
- [4] A. Woo and D. Culler, "A transmission control scheme for media access in sensor networks," ACM/IEEE International Conference on Mobile Computing and Networking, pp.221-235, 2001.
- [5] W. B. Heinzelman, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE International Conference on Transactions on Wireless Communications, IEEE, Vol.1, No.4, pp.660-670, 2002.
- [6] F. Xiangning, S. Yulin, "Improvement on LEACH Protocol of Wireless Sensor Network," International Conference on Sensor Technologies and Applications, IEEE, pp.260-264, 2007.
- [7] J. Hill, R. Szwedczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," ACM, Vol.35 No.11, pp.93-104, 2000.
- [8] P. Levis and N. Lee, "TOSSIM: A Simulator for TinyOS Networks," September, 2003.
- [9] 최경진, 윤명준, 심인보, 이재용, "무선 센서 네트워크에서의 에너지 효율적인 클러스터 헤드 선출 알고리즘", 한국통신학회논문지, 제32권, 제6호, pp.342-349, 2007.

저 자 소 개

장 유 진(You-Jin Jang)

준회원



- 2008년 : 전북대학교 컴퓨터공학과(공학사)
- 2008년 ~ 현재 : 전북대학교 대학원 컴퓨터공학과 석사과정
<관심분야> : 센서네트워크, 공간 데이터베이스

김 아 름(Ah-Reum Kim)

준회원



- 2008년 : 전북대학교 컴퓨터공학과(공학사)
- 2008년 ~ 현재 : 전북대학교 대학원 컴퓨터공학과 석사과정

<관심분야> : 센서네트워크, 공간 데이터베이스

장 재 우(Jae-Woo Chang)

정회원



- 1984년 : 서울대학교 전자계산
기공학과(공학사)
 - 1986년 : 한국과학기술원 전산
학과(공학석사)
 - 1991년 : 한국과학기술원 전산
학과(공학박사)
 - 1996년 ~ 1997년 : Univ. of Minnesota, Visiting
Scholar
 - 2003년 ~ 2004년 : Penn State Univ., Visiting
Professor
 - 1991년 ~ 현재 : 전북대학교 컴퓨터공학과 교수
- <관심분야> : 공간 네트워크 데이터베이스, 센서네트
워크, 하부저장구조