무선 센서 네트워크에서의 Max k-Cut 기반의 클러스터링 알고리즘

(Max k-Cut based Clustering Algorithm for Wireless Sensor Networks)

김 재 환 † 장 형 수 ††
(JaeHwan Kim) (HyeongSoo Chang)

요 약 본 논문에서는 Wireless Sensor Networks에서 Max k-Cut Problem을 기반으로 위치 정보를 사용하지 않고 클러스터 헤드를 적절히 분산하여 선출함으로써 에너지 효율적인 클러스터링을 하는 중앙처리 방식의 새로운 알고리즘 "MCCA : Max k-Cut based Clustering Algorithm for Wireless Sensor Networks"을 제안한다. MCCA는 이웃 노드와의 상대적이고 근사적인 거리 정보만을 사용하여 효율적으로 클러스터링을 하고 에너지가 적은 노드는 클러스터 헤드 선출에서 일정 기간 제외되는 방법을 사용함으로써 LEACH, EECS보다 에너지 효율이 증대됨과 GPS를 사용한 BCDCP와 에너지 효율이 비슷함을 실험을 통하여 보인다.

키워드 : 무선 센서 네트워크, 클러스터링, Max k-Cut 문제

Abstract In this paper, we propose a novel centralized energy-efficient clustering algorithm, called "MCCA : Max k-Cut based Clustering Algorithm for Wireless Sensor Networks." The algorithm does not use location information and constructs clusters via a distributive Max k-Cut based cluster-head election method, where only relative and approximate distance information with neighbor nodes is used and nodes, not having enough energy, are excluded for cluster-heads for a specific period. We show that the energy efficiency performance of MCCA is better than that of LEACH, EECS and similar to BCDCP's by simulation studies.

Key words : Wireless Sensor Networks, Clustering, Max k-Cut Problem

1. 서 론

Wireless Sensor Networks(이하 WSNs)는 일반적으로 적은 비용, 낮은 에너지, 작은 장치, 짧은 통신 거리 등의 특징을 가지고 있다. WSNs에서 각 노드들은 에너지원으로 베터리를 사용하기 때문에 추가적인 전원 공급이 되지 않는다. 따라서 각 노드의 전력 사용은 WSNs에서의 가장 큰 논점이라고 할 수 있다. 효율적인 전력 사용은 각각의 노드의 생존 기간을 연장시켜 줄 뿐 아니라 전체 네트워크의 lifetime을 연장시켜 줄 수 있기 때문이다.

센서 노드의 에너지 사용은 크게 두 부분에서 이루어진다. 하나는 감지 에너지이며 또 하나는 통신 에너지이다. 일반적으로 센서 노드가 감지를 위해서 사용하는 에너지는 통신을 위해서 사용하는 에너지에 비해 극히 미미하다고 할 수 있다. 따라서 통신 에너지를 줄이는 것이 중요하다.

감지된 데이터를 base station에 보내기 위한 방법은 크게 세 가지 방법으로 나눌 수 있다[1]. 첫 번째는 각 노드가 base station과 직접통신(direct communica-

· 이 논문은 2007년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2007-000-10511-0)
· 이 논문은 2008 한국컴퓨터종합학술대회에서 '무선 센서 네트워크에서의 Max k-Cut 기반의 클러스터링 알고리즘'의 제목으로 발표된 논문을 확장한 것임

† 정 회 원 : 서강대학교 컴퓨터공학과
greensun00@sogang.ac.kr
†† 정 회 원 : 서강대학교 컴퓨터공학과 교수
hschang@sogang.ac.kr

논문접수 : 2008년 8월 27일
심사완료 : 2008년 11월 15일

tion)을 하는 것이다. 두 번째 방법은 base station과 멀리 떨어져 있는 노드가 base station과 가까이에 있는 다른 노드를 거쳐서 통신하는 방법(Minimum Transmission Energy(MTE) protocol)이다. 마지막은 클러스터링을 하여 클러스터 헤드가 대표로 데이터를 보내는 방법이다. 통신 에너지는 통신 거리의 제곱에 비례하기 때문에 마지막 방법이 가장 에너지를 적게 사용한다는 것을 알 수 있다[1].

클러스터링을 통하여 데이터를 전송하는 방법에 대하여 모든 노드들이 random한 방식으로 돌아가면서 클러스터 헤드로 선출되는 LEACH(Low-Energy Adaptive Clustering Hierarchy)[1] 알고리즘이 WSNs에서 대표적인 방법으로 사용되었다. LEACH는 WSNs에서 고정된 클러스터 헤드를 사용할 경우 클러스터 헤드로 선출된 노드만 많은 에너지를 소비하므로 전체적인 lifetime이 줄어드는 문제를 해결하였다.

LEACH는 분산적인 에너지 사용으로 뛰어난 성능 향상을 가져왔지만 확률적으로 클러스터 헤드를 선출하는 방식이기 때문에 클러스터 헤드들의 위치가 균일하지 않게 분포되어 선출될 수도 있다. 또한 선출되는 클러스터 헤드의 수가 일정하지 않을 가능성도 있다. 따라서 비효율적인 클러스터를 구성할 가능성이 있다.

이를 해결하기 위한 방법으로 중앙처리 방식인 LEACH-C[2]가 고안되었다. 이 알고리즘은 각 노드에서 위치와 에너지 정보를 base station으로 보내고 base station에서는 이 정보를 받아 어떠한 노드가 클러스터 헤드로 적당한지 결정하게 된다. BCDCP(Base-Station Controlled Dynamic Clustering Protocol)[3] 또한 위치 정보를 이용하여 네트워크 공간을 적절한 클러스터의 수만큼 균일하게 분할하여 클러스터링을 한 뒤 멀티 홉 방식을 이용하여 데이터를 전송한다. 이러한 방법은 WSNs가 가장 최적의 상태에서 통신할 수 있다는 장점은 있으나 각 노드의 위치 정보를 알기 위하여 GPS를 사용해야 한다. 따라서 추가 비용이 발생하게 된다.

GPS를 사용하지 않고 균일한 분포로 클러스터 헤드를 선출하는 방법으로 EECS(Energy Efficient Clustering Scheme)[4]가 있다. EECS는 필요한 수 이상의 클러스터 헤드의 후보자들을 선출하여 이들끼리 경합하여 클러스터 헤드를 선출하는 방식을 사용하고 있다. 하지만 이 방법도 확률적인 방법이므로 선출된 클러스터 헤드들의 분포가 균일하지 않을 가능성이 존재한다.

본 논문에서는 GPS와 같이 각 노드의 위치 정보를 사용하지 않으면서 클러스터 헤드의 위치가 네트워크 공간에 균일하게 분포되도록 클러스터 헤드를 선출하는 알고리즘 MCCA(Max k-Cut based Clustering Algorithm for Wireless Sensor Networks)를 제안하고자 한다. MCCA는 중앙 처리 방식을 사용하여 Max k-Cut Problem을 풀어 클러스터 헤드로 선출되는 노드 사이에 일정한 거리를 유지시키도록 한다. 또한 에너지를 많이 소비한 노드를 일정 기간 동안 클러스터 헤드에서 제외시키는 방법을 사용하여 모든 노드들이 비슷하게 에너지를 소비하도록 한다.

본 논문의 2장에서는 논문에서 가정하고 있는 네트워크 환경과 논문의 목적에 대하여 설명한다. 3장에서는 Max k-Cut Problem을 이용한 새로운 알고리즘인 MCCA에 대하여 소개한다. 4장에서는 새로운 알고리즘을 실험을 통해 LEACH, EECS, BCDCP 알고리즘과 비교 분석한다. 마지막으로 5장에서 논문의 결론을 내리고 앞으로의 연구 방향을 제시한다.

2. 문제 정의

2.1 네트워크 모델

WSNs는 구축하는 방법에 따라서 여러 가지 다양한 환경을 만들어 낼 수 있다. 본 논문이 가정하고 있는 네트워크 환경은 다음과 같다.

- 모든 센서 노드는 배치된 이후로 움직이지 않는다. 즉, 고정되어 있다.
- 모든 센서 노드의 에너지는 제한되어 있으며, 각 노드가 가지고 있는 에너지와 통신 능력은 모두 동일하다.
- 센서 노드들은 자발적으로 무선 통신이 가능하며 base station과 직접 통신도 가능하다.
- Base station은 1개이며 외부에서 에너지를 공급받기 때문에 에너지 제약이 없다.
- 센서 노드는 transmission power level을 조절할 수 있다. 즉, 전송 세기를 이용하여 전송 거리를 조절할 수 있다.
- 각 센서 노드의 위치를 알지 못한다. 즉, 위치 정보를 알 수 있는 GPS를 사용하지 않는다.

2.2 연구 목적

WSNs에서 전체 네트워크를 하나의 그래프 $G=(V,E)$로 나타내면 각 노드는 vertex라고 할 수 있다. 각각의 노드를 vertex v라고 한다면 n개의 노드들은 vertex의 집합 $V=\{v_1, v_2, \cdots, v_n\}$로 표현할 수 있다. $edge(u,v)\in E$ $(u,v\in V)$는 vertex u와 v의 연결 관계이며 u와 v가 통신 가능함을 나타낸다.

클러스터링을 사용하는 WSNs는 모든 노드를 partition하는 것과 같다. 즉, s가 n보다 작거나 같은 임의의 정수일 때 $V_1, V_2, \cdots, V_s \subset V$이며 $V_1 \cup V_2 \cup \cdots \cup V_s = V$이고 $V_i \cap V_j = \varnothing$, $V_i \neq \varnothing$, $\forall i,j\in\{1,\cdots,s\}$, $i\neq j$이다. $V_1, V_2, \cdots, V_s$가 하나의 클러스터라면 각 클

러스터마다 대표적으로 통신을 하는 클러스터 헤드가 있다. 이를 $H(V_i) \in V_i, i = 1,2,\cdots,s$와 같이 나타낼 수 있고 $H(V_i)$는 V_i 클러스터를 대표하는 클러스터 헤드이다.

모든 클러스터 헤드가 자신의 클러스터에 속하는 노드의 데이터를 모두 모아서 정확히 한 번 전송하는데 걸리는 시간을 t라고 하자. 본 논문에서는 이러한 t를 정해진 횟수 f(임의의 상수)번 반복하는데 걸리는 시간 $f \cdot t$를 한 round로 정의한다. 그렇다면 V_i^T를 T round에 i번째 cluseter에 속하는 센서 노드들의 집합으로 나타낼 수 있다.

본 논문의 목적은 WSNs의 lifetime을 연장시키는 것이다. lifetime은 모든 노드가 작업을 시작한 후 첫 번째로 에너지를 전부 소진하여 죽는 노드가 발생하는 때까지의 시간을 의미한다. 각 노드가 가지고 있는 에너지는 제한되어 있기 때문에 lifetime을 연장시키기 위해서는 한 round에 각 노드가 사용하는 통신 에너지를 줄여야 한다. 통신을 위해서 사용되는 에너지는 노드간의 거리의 제곱에 비례한다. 따라서 lifetime을 줄이기 위한 문제는 매 round마다 구성되는 클러스터들에서 클러스터 헤드 노드와 클러스터 헤드가 아닌 노드 사이의 거리를 일정하게 만드는 문제로 귀결된다. 처음으로 에너지를 다 소비하는 노드가 발생하여 통신 불가능한 노드가 생겼을 때까지의 진행된 round수를 Q라고 하고 $dist(edge(u,v))$ $(s.t. u,v \in V)$를 $edge(u,v)$의 거리라고 했을 때 본 논문에서는 $\frac{1}{Q}\sum_{T=1}^{Q}\sum_{i=1}^{s}\sum_{v_j \in V_i^T, v_j \neq H(V_i^T)} dist(edge(H(V_i^T), v_j))^2$의 값을 최소화 시키는 문제를 풀고자 한다. 이는 클러스터의 개수 s가 정해져있고 모든 노드가 에너지가 충분하여 다른 노드와 통신이 가능하다고 가정한다면 매 round마다 구성되는 각 클러스터에서 클러스터 헤드와 클러스터 헤드가 아닌 노드 사이의 거리를 일정하게 만드는 것과 같다.

3. Max k-Cut based Clustering Algorithm

3.1 기본 개념

본 논문의 목적은 매 round마다 구성되는 각각의 클러스터에서 클러스터 헤드와 클러스터 헤드가 아닌 노드 사이의 거리가 일정하도록 만드는 것이다. 또한 클러스터 헤드는 에너지 소비가 심하기 때문에 매 round마다 에너지가 더 많은 노드로 교체를 해주는 것이 바람직하다. 이를 위해서는 클러스터 헤드로 선출된 노드의 위치가 전체 네트워크 공간에 적절히 분산되고 각 round마다 선출되는 클러스터 헤드의 수가 일정하도록 만드는 것이 좋다. 클러스터 헤드로 선출되는 노드가 전체 네트워크 공간에 적절히 분산되도록 하기 위해서는 한 round에 선출되는 클러스터 해드들 사이를 일정한 거리 이상으로 유지시켜주는 것이 중요하다. 일정 수의 클러스터 헤드가 서로 일정 거리를 유지하면서 선출되려면 전체 네트워크에 고르게 분산되어 위치해 있어야 한다.

본 논문에서 센서 노드가 transmission power level을 조절할 수 있다고 가정하였기 때문에 각 센서 노드는 전송 세기를 조절하여 이웃 노드의 상대적인 거리를 파악할 수 있다. 즉, transmission power level을 조금씩 높여가면서 자신과 가까운 이웃 노드와 먼 이웃 노드를 구분할 수 있다. 가까운 이웃일수록 같은 round에 클러스터 헤드로 선출되지 않도록 한다면 클러스터 헤드들 사이에 일정 거리를 유지할 수 있다.

아래 그림 1과 같이 네트워크 공간에 균일하게 분포되어 있는 센서 노드들 중에 검정색으로 표시된 노드를 기준으로 전송 세기를 늘려가면서 level에 따라 이웃을 설정하였다고 하자. 모든 노드들이 이러한 방법으로 이웃 노드를 설정하고 한 round에 선출되는 클러스터 헤드의 수가 일정하다면 가까운 이웃 노드를 피하기 위하여 그림 2와 같이 클러스터 헤드가 선출될 것이다.

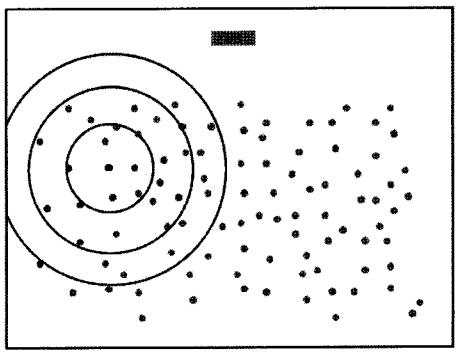


그림 1 전송 세기에 따른 이웃 노드 설정

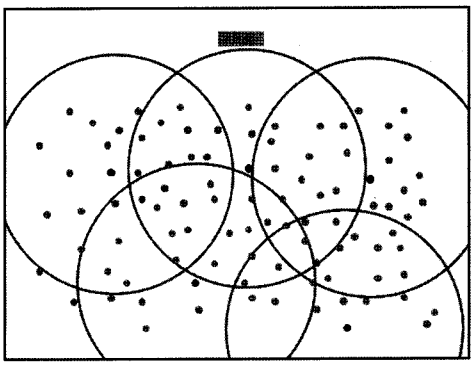


그림 2 서로의 이웃이 아닌 노드들의 배치

물론 모든 노드들이 돌아가면서 클러스터 헤드로 선출되는 것이 바람직하기 때문에 클러스터 헤드를 선출할 때 항상 이웃 노드가 아닌 노드들로만 선출될 수가 없다. 하지만 좀 더 멀리 떨어져 있는 노드들로 클러스터 헤드를 선출하는 것이 더 좋은 방법이다. 따라서 그림 1과 같이 전송 세기에 따라서 자신의 이웃 노드를 구분하였으므로 자신과 가까운 이웃과는 같은 round에 클러스터 헤드로 선출되는 것을 피해야할 것이다. 이를 위해 전송 세기의 level에 따라서 노드 사이에 가중치를 정해준다. 가중치가 level에 반비례하게 정해지면 가까운 노드끼리는 높은 가중치를 가지게 되고 먼 노드끼리는 낮은 가중치를 가지게 된다. 물론 이웃 노드가 아니라면 가중치는 0으로 설정된다. 이러한 상황에서 가중치가 높지 않은 노드끼리 같은 round에 클러스터 헤드로 선출되는 것이 좋다.

이 문제를 해결하기 위하여 본 논문에서는 Max k-Cut Problem을 사용한다. 위와 같은 가정 하에 WSNs를 가중 그래프 G로 나타낼 수 있다. 방향성이 없는 가중 그래프 $G=(V,E)$에서 V는 vertex들의 집합이며 WSNs에서의 노드가 되고 E는 edge들의 집합이며 WSNs에서의 이웃 노드와의 연결 관계가 된다. Edge는 두 노드가 이웃 노드로 설정되었을 때 두 노드 사이의 연결 관계이며 level에 반비례하게 가중치가 주어진다. n은 vertex의 개수이다. 본 논문에서는 이러한 그래프 G에 대하여 Max k-Cut Problem에 적용하여 클러스터 헤드가 분산되어 선출되도록 한다.

V에 대하여 가능한 모든 partition의 집합을 Π라고 할 때 Max k-Cut Problem이란 그래프 $G=(V,E)$에서 식 (1)을 만족하는 k개의 V의 부분집합으로 구성된 partition $P=P_0,P_1,...P_{k-1}(P\in\Pi)$을 만드는 것이다[5].

$$\max_{P\in\Pi}\sum_{0\le r<s\le k-1}\sum_{i\in P_r, j\in P_s} w_{ij}$$

where w_{ij}=nonnegative weight for $i,j\in V$ (1)

이는 가중치가 높은 vertex끼리는 다른 집합에 속하도록 partition하는 것이 목적이다. 즉, 같은 집합에 속하는 vertex끼리의 가중치를 낮추는 것이 목적이다. 이를 이용하여 Max k-Cut Problem을 다시 정의하면 다음과 같다.

$$\min_{P\in\Pi}\sum_{s=0}^{k-1}\sum_{i,j\in P_s, i\neq j} w_{ij}$$

where w_{ij}=nonnegative weight for $i,j\in V$ (2)

이와 같은 조건을 만족하는 partition P를 찾는다면 같은 집합 $P_i(i=\{0,1,...k-1\})$에 속하는 vertex들은 일정한 거리를 유지할 수 있게 된다. 따라서 하나의 집합 $P_i(i=\{0,1,...k-1\})$에 속하는 센서 노드들을 같은 round에 클러스터 헤드로 선출하면 매 round마다 돌아가면서 선출되는 클러스터 헤드의 위치는 고르게 분포된다.

매 round마다 선출되는 클러스터 헤드가 고르게 분포되어야 하는 것 이외에 각 round에 선출되는 클러스터 헤드의 수도 일정해야 한다. 따라서 Max k-Cut Problem에서 다음의 조건을 고려하여 partition을 나누어야 한다.

$$\max_{P\in\Pi}\prod_{s=0}^{k-1}|P_s| \quad (3)$$

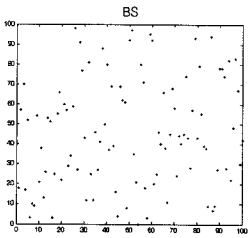
이는 Max k-Cut Problem을 통해 나누어진 집합 P_i $(i=\{0,1,...k-1\})$들의 개수가 일정해야함을 의미한다.

Max k-Cut Problem을 통하여 나누어진 partition P는 클러스터 헤드를 선출하는데 사용된다. Max k-Cut Problem을 통해 얻어진 k개의 집합 $P_0,P_1,...,P_{k-1}\subset V$은 $P_0\cup P_1\cup\cdots\cup P_{k-1}=V$이고 $P_i\neq\emptyset$, $P_i\cap P_j=\emptyset$, $\forall i,j\in\{0,\cdots,k-1\}, i\neq j$이다. t번째 round에 집합 P_x에 속하는 노드들이 클러스터 헤드로 선출되었다면 $t+1$번째 round에는 집합 $P_{(x+1)\bmod k}$에 속하는 노드들이 클러스터 헤드로 선출되게 된다. 같은 집합 P_x에 속하는 노드들끼리의 가중치가 최소가 되도록 하였으므로 이 노드들의 거리는 최대한 멀리 떨어지게 되고 partition P의 집합들에 속하는 노드들의 개수는 일정하기 때문에 전체적으로 클러스터 헤드의 고른 분포를 얻을 수 있다.

클러스터 헤드로 선출되는 노드는 클러스터 헤드로 선출되지 않은 노드에 비하여 에너지를 많이 소비하게 된다. 또한 base station에서 멀리 떨어진 노드가 클러스터 헤드로 선출된다면 가까이에 있는 노드보다 에너지 소모가 많을 것이다. 각 노드의 에너지 소모가 균일하지 않다면 다른 노드보다 먼저 에너지를 전부 소비해 통신이 불가능한 노드가 발생하고 전체 네트워크의 상태가 불안정하게 된다. 따라서 상대적으로 에너지가 다른 센서 노드보다 적은 노드는 Max k-Cut Problem을 사용한 클러스터 헤드 선출시 제외시킨다면 전체 센서 노드들의 에너지 소비를 비슷하게 유지시킬 수 있고 WSNs의 lifetime을 연장할 수 있다.

3.2 알고리즘

알고리즘의 전체적인 진행 과정은 다음과 같다. 우선 그림 3(a)와 같이 원하는 지역에 센서 노드가 배치된다. 모든 노드는 자신의 주변 노드에 대한 상대적이고 근사적인 거리 정보를 base station에 보낸다. Base station에서는 이 정보를 통합하여 그림 3(b)와 같은 가중치 테이블을 만들고 이 테이블을 토대로 Max k-Cut



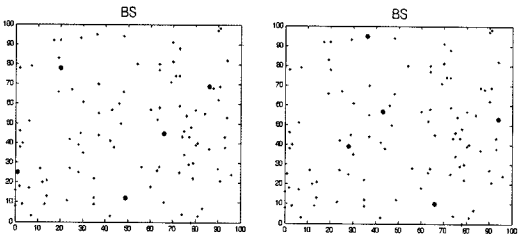
(a) 초기 노드의 배치

	1	2	3	4	5		96	97	98	99	100
1	0	16	0	12	20		20	0	0	25	0
2	16	0	25	0	100		33	14	16	20	16
3	0	25	0	11	20	...	0	25	50	0	20
4	12	0	11	0	0		0	0	12	0	20
5	20	100	20	0	0		33	12	16	25	16
			⋮			⋱			⋮		
96	20	33	0	0	33		0	0	0	33	0
97	0	14	25	0	12		0	0	33	0	16
98	0	16	50	12	16	...	0	33	0	0	25
99	25	20	0	0	25		33	0	0	0	0
100	0	16	20	20	16		0	16	25	0	0

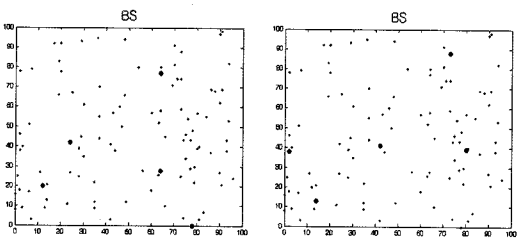
(b) 각 노드 사이의 가중치 테이블

	ID of node				
P_0	17	23	54	42	5
P_1	34	67	2	44	89
⋮			⋮		
P_{k-2}	1	76	79	25	12
P_{k-1}	95	58	31	22	66

(c) Max k-Cut Problem을 통하여 얻은 partition P



(1) P_0에 속하는 노드 (2) P_1에 속하는 노드



(3) P_{k-2}에 속하는 노드 (4) P_{k-1}에 속하는 노드

(d) P_0에서 P_{k-1}까지 클러스터 헤드로 사용하면서 알고리즘 진행

그림 3 MCCA의 진행 과정

Problem을 해결한다. Max k-Cut Problem을 통하여 나온 결과는 partition $P=P_0,P_1,...,P_{k-1}$이고 그림 3(c)와 같다. k round 동안 $P_i(i=\{0,1,...,k-1\})$에 속하는 노드들이 순차적으로 클러스터 헤드의 역할을 하고 클러스터링 및 감지와 데이터 전송을 한다. 그림 3(d)는 각 round가 진행될 때의 클러스터 헤드가 선출된 모습이며 그림 3(d)(1)은 P_0에 속하는 노드들이 클러스터 헤드로 선출된 모습이다. k round 이후 모든 노드들은 자신의 남은 에너지를 base station으로 보내고 base station은 다시 Max k-Cut Problem을 통하여 그림 3(c)와 같은 partition P를 얻고 다시 그림 (d)의 과정을 진행한다. 이러한 과정을 에너지를 전부 소모하여 더 이상 통신을 할 수 없는 노드가 일정 수 이상이 될 때까지 반복하여 진행한다.

3.2.1 이웃 노드 설정 및 전송 단계

MCCA의 핵심은 전송 세기를 이용한 이웃 노드와의 상대적이고 근사적인 거리 정보를 이용하는 것이라고 할 수 있다. 각 노드는 자신의 이웃과의 근사적인 거리 정보를 얻기 위하여 전송 세기를 점차 늘려가면서 자신과 이웃과의 가중치를 다르게 설정한다. 가장 작은 전송 세기의 level이 1이라고 하고 그 때 전송되는 면적을 a라고 한다면 level 2에서는 $2a$의 면적을 전송할 수 있도록 전송 세기를 설정한다. 이런 식으로 level i에 비례하여 $i \cdot a$의 면적만큼 전송할 수 있도록 전송 세기를 점차 늘려나간다. 전체 노드의 수를 n, 클러스터링을 사용하였을 때 통신 에너지를 가장 적게 사용하는 클러스터 헤드의 수를 CH_{opt}라고 하자. 이 때 자신의 이웃이 되는 노드의 수를 더해나가면서 하나의 클러스터가 가지는 노드 수(n/CH_{opt})이상이 된다면 이웃 설정을 종료한다. 모든 노드가 이웃 노드 설정을 마치면 각 노드는 이 데이터를 base station으로 전송한다. 모든 노드는 움직이지 않는다고 가정하였으므로 자신의 이웃으로 설정된 노드와의 거리는 변하지 않는다. 따라서 이 과정은 전체 MCCA 알고리즘에서 단 한번만 이루어지며 정보를 전송하는데 사용되는 overhead는 크지 않다.

하나의 level 1에서 전송되는 면적 a값의 변화가 알고리즘의 성능에 크게 영향을 미치지는 않지만 이웃 노드 사이의 가중치 차이를 두기 위해서는 a의 값은 전체 네트워크 면적 A에 대하여 A/CH_{opt}보다는 작아야 한다. 만약 노드가 균일하게 분포된 상황에서 a의 값이 A/CH_{opt}보다 크다면 level 1에서 이웃 설정을 종료할 가능성이 크고 그렇다면 이웃 노드끼리의 가중치가 같아지기 때문이다.

3.2.2 Max k-Cut Problem 해결 단계

각 노드로부터 이웃 노드와의 관계에 대한 정보를 전

송받은 base station은 이 정보를 통합하여 방향성이 없는 가중 그래프를 만들 수 있다. 이 그래프에서 Max k-Cut Problem을 통하여 각 노드의 클러스터 헤드 선출 시기를 결정할 수 있다.

Max k-Cut Problem을 WSNs에 적용하기 위해서는 나누어지는 집합의 수 k를 정해야 한다. 이는 전체 센서 노드의 수 n을 적절한 클러스터 헤드의 수 CH_{opt}으로 나누어주면 된다. 이 때 에너지가 다른 노드에 비해 상대적으로 적은 노드들은 클러스터 헤드로 선출되는 것에서 제외시키고 제외되지 않은 노드의 수를 n'라고 한다면 Max k-Cut Problem에 적용되는 노드의 수는 n'가 된다.

전체 노드의 집합을 N이라고 하면 $|N|=n$이다. $R(X)$를 집합 X의 노드들이 가진 평균 에너지라고 하고 $r(x)$를 노드 x의 현재 에너지라고 하자. 클러스터 헤드 선출에서 제외되는 노드의 집합 $\overline{N}$는

$$\overline{N}=\{r(x)<c\times R(N)|x\in N\} \quad (4)$$

이고 c는 전체노드의 평균 에너지와 제외될 노드의 에너지의 비율을 나타낸 값으로 1이하의 상수이다. 따라서 클러스터 헤드 선출에 참여하는 노드의 집합 N'는

$$N'=N-\overline{N} \quad (5)$$

이다. $|N'|=n'$이고 집합의 수 k는 n/CH_{opt} 대신 n'/CH_{opt}로 정해진다.

모든 인자가 결정되면 Max k_Cut Problem을 이용하여 N'을 k개의 부분으로 partition을 해야 한다. 하지만 Max k-Cut Problem은 잘 알려진 NP-hard problem이다. 이러한 NP-hard problem을 해결하기 위해서 Genetic 알고리즘과 같은 metaheuristic 방법을 사용할 수 있다. 하지만 metaheuristic 방법은 엄청난 계산량과 시간을 요구한다. WSNs와 같은 네트워크 환경에서 이렇게 많은 시간을 계산해야 할 경우 발생하는 delay는 감당하기 쉽지 않다. 그렇기 때문에 본 논문에서는 polynomial 시간 복잡도를 가지는 approximation 기법을 기반으로 하는 알고리즘을 사용한다.

Cho et. al.[6]은 Maxcut과 k-Coloring에 대한 approximation 알고리즘에 대하여 연구하였다. Maxcut Problem을 사용하여 k-Coloring Problem을 해결하였는데 같은 방법으로 Maxcut Problem을 사용하여 Max k-Cut Problem을 해결할 수 있다. 방향성이 없는 가중 그래프 $G=(V,E)$에 대하여 집합 X는 $X\subset V$이고 $X\neq\varnothing$이고 집합 $\overline{X}$는 $\overline{X}=V-X$, $\overline{X}\neq\varnothing$일 때 Maxcut Problem의 정의는 다음과 같다[6].

$$\max_{X,\overline{X}} \sum_{i\in X, j\in\overline{X}} w_{i,j}$$

where w_{ij}=nonnegative weight for $i,j\in V$ (6)

이는 X에 속하는 vertex와 $\overline{X}$에 속하는 vertex 집합 사이의 가중치가 최대가 되도록 두 집합으로 나누는 것이다. 서로 vertex를 공유하지 않는 edge들의 집합을 M이라고 하자. 이 논문[6]에서는 Maxcut Problem을 풀기 위한 approximation 방법으로 집합 M에 속하는 edge에 대한 두 vertex는 각각 다른 집합$(X,\overline{X})$에 속하되 현재까지의 알고리즘의 수행을 통해 집합에 포함되어 있는 vertex들과 자신의 가중치의 합을 구하여 그 값이 작은 곳으로 포함되도록 한다. 즉 M에 속하는 edge에 대한 두 vertex를 u,v라고 하고 $W(X:x)$를 집합 X에 속하는 vertex들과 vertex x와의 가중치의 합이라고 한다면 알고리즘은 다음과 같이 수행된다.

$$\begin{cases} u\in X,\ v\in\overline{X} & W(X:u)+W(\overline{X}:v)\le W(X:v)+W(\overline{X}:u) \\ v\in X,\ u\in\overline{X} & otherwise \end{cases} \quad (7)$$

또한 M에 속하는 edge에 대한 수행을 마치면 남아있는 vertex v는 $W(X:v)$와 $W(\overline{X}:v)$ 중에 값이 작은 집합에 속하게 된다.

k-Coloring Problem은 그래프의 edge에 가중치를 1로 바꾸어서 Maxcut Problem에 대한 approximation 알고리즘을 반복적으로 사용하여 해결한다. Max k-Cut Problem을 해결하기 위해서는 그래프의 edge에 가중치를 그대로 두고 Maxcut Problem에 대한 approximation 알고리즘을 반복적으로 사용하면 된다. 이 때 반복적으로 나누다가 집합의 개수가 k개가 되면 알고리즘을 종료하게 된다.

이 알고리즘은 집합이 tree 형태로 나누어지기 때문에 집합에 속하는 원소의 개수가 균일하지 않게 된다. 하지만 본 논문에서는 나누어진 k개의 집합의 원소 개수가 균일하기를 원하기 때문에 원소의 개수가 많은 집합 X_i에서 $W(X_i:v)$ $s.t.\ v\in X_i, i\in 1,2,...,k$의 값이 큰 v를 골라 원소의 개수가 작은 집합으로 옮겨주는 작업을 추가로 수행한다. 이 작업은 Max k-Cut Problem의 performance에 변화를 줄 수도 있지만 본 논문에서 원하는 목적에 부합하게 만들어 준다.

3.2.3 클러스터 헤드 선출 및 클러스터링 단계

Base station에서 Max k-Cut Problem을 통하여 각 노드를 k개의 집합으로 분류한 뒤 이 정보를 이용하여 클러스터 헤드 선출 시기와 선출된 클러스터 헤드에 따른 클러스터에 속하는 노드들을 미리 정할 수 있다. 이렇게 base station에서 스케줄링을 해주는 방법은 클러스터링 과정에서 소모되는 에너지를 줄일 수 있게 된다.

현재 진행되고 있는 round가 x이고 Max k-Cut Problem을 통하여 n'개의 노드를 k개의 집합으로 나누었다면 P_0에 속하는 노드들이 한 round 동안 클러스터

헤드가 된다. 다음에는 P_1에 속하는 노드들이 클러스터 헤드가 된다. 이런 식으로 P_{k-1}에 속하는 노드들까지 클러스터 헤드가 된다. 이웃 노드간의 근사적인 거리 정보가 주어져 있기 때문에 이를 토대로 클러스터 헤드가 아닌 노드들은 가장 가까운 클러스터 헤드와 같은 클러스터가 되도록 클러스터링을 할 수 있다. x round부터 $x+k-1$ round까지의 클러스터 헤드 선출과 클러스터링을 마치게 되면 base station은 이 정보를 모든 노드에게 전송하게 된다.

3.2.4 감지 및 데이터 전송 단계

각 노드들이 base station으로부터 round에 따른 클러스터링 정보를 전송받게 되면 해당 round에 각자의 역할을 하게 된다. 이 과정에서도 에너지를 줄일 수 있는 많은 방법이 고안되었지만 본 논문에서는 클러스터링에 대한 알고리즘을 제안하는 것이 목적이므로 새로운 방법을 제시하지 않는다. 4장에서 본 논문의 클러스터링 기법과 LEACH, EECS의 클러스터링 기법을 실험을 통해 비교하고 EECS 또한 LEACH의 방식을 따르고 있으므로 본 논문에서는 감지 및 데이터 전송 단계는 LEACH의 방법을 따른다.

클러스터링에 포함된 노드의 목록을 가지고 있는 클러스터 헤드는 각 노드에 대한 TDMA schedule을 생성하여 각 노드에 이 정보를 전송해 준다[1]. 클러스터 헤드가 아닌 노드는 이 정보를 받아서 자신이 전송해야 할 시간에 전송을 하고 그 이외의 시간에는 에너지 절약을 위하여 통신 radio를 끈다[1].

3.2.5 클러스터 헤드 재선출

k번의 round를 진행하면 집합 N'에 속하는 모든 노드들이 전부 한 번씩 클러스터 헤드를 하게 된다. 그렇다면 다시 Max k-Cut Problem을 통해서 클러스터 헤드 선출과 클러스터를 정해야 한다. 물론 매번 Max k-Cut Problem을 계산할 필요 없이 한 번 정해진 순서대로 계속 반복해서 진행할 수도 있지만 에너지를 많이 소비한 노드는 클러스터 헤드에서 제외시키기 위하여 k번의 round가 진행된 이후에 새롭게 다시 선출한다. 이때 모든 노드들은 자신의 에너지를 클러스터 헤드를 통하여 base station에 전송한다. 3.2.2에서와 같이 k가 정해지고 Max k-Cut Problem을 통하여 클러스터 헤드가 선출되고 클러스터를 구성하여 모든 노드에 전송해 주는 방식을 반복적으로 수행한다.

3.2.6 Pseudocode

전체적인 알고리즘의 흐름을 살펴보면 다음과 같다.

```
MCCA
Input : G=(N,E), CH_opt
Output : transmission sensing data to base station
{
// * 이웃 노드 설정 단계 * //
  for (each node v in N) {
    n(v) = 0;    // n(v) : 노드 v의 이웃 노드 수
    for (i=1 ; i<max_power_level ; i++) {
      for (each node u in N) {
        if (d_{i-1} < d(v,u) ≤ d_i) {
          w_vu = 100/i ;
          n(v)++;
        }
      }
    if (n(v) ≥ |N|/CH_opt)
      break;
    }
  }
// * round 진행 * //
  while (the number of alive node < d * |N|) {
    N' = {v∈N | r(v) ≥ c * R(N)}  for ∀v∈N ; //[3.2.2]참조
    P= Max_k_Cut(G=(N',E), k);
    for (i=0 ; i<k ; i++) {
      CH_x = {v∈P_i | v is xth element of P_i};
      C_x = {v∈V | d(v,CH_x) ≤ d(v,CH_y)} for ∀y={1,2,...,|P_i|};
      sensing and data transmission; //[3.2.4]참조
    }
  }
}
```

max_power_level : 센서 노드가 최대 크기로 전송할 수 있는 level
d_i : i의 power level로 데이터를 전송할 수 있는 거리
$d(v,u)$: vertext v와 u의 거리
CH_x : x번째 클러스터 헤드
C_x : x번째 클러스터 집합
c, d : 0과 1사이의 특정 상수

위 알고리즘에서 호출하는 Max_k_Cut이란 함수의 내용은 다음과 같다[6].

```
Max_k_Cut
Input : G=(N',E), k
Output : partition P= P_1,P_2,...,P_k
{
  ncut = 1;
  nset = 0;
  initqueue(Q);    // Q : queue
  inqueue(Q, G);
  while (ncut < k) {
    g = dequeue(Q);
    Maxcut(g, a, b);   // g,a,b⊂G
    ncut++;
    inqueue(Q, a);
    inquque(Q, b);
  }
  while (empty(Q)≠TRUE) {
    nset++;
    P_nset = dequeue(Q);
  }
  modification(P);  // P의 집합들이 가지는 원소의 수를
                    일정하게 만들어주는 함수
}
```

Max_k_Cut함수에서 호출하는 Maxcut함수는 3.2.2에서 설명한 Maxcut Problem을 해결하는 함수로 g라는 그래프를 입력으로 하여 두 개의 그래프 a, b를 출력으로 한다. a, b는 3.2.2에서 X와 $\overline{X}$에 해당하며 함수의 내용은 [6]의 Algorithm A를 참조하길 바란다.

4. 결과 및 분석

데이터를 보낼 때 보내는 노드는 전송 에너지와 증폭 에너지가 필요하다. 또한 데이터를 받는 노드는 수신 에

너지가 필요하다. 소모되는 에너지를 측정하기 위해서는 radio model을 적용해야 한다. l은 전송되는 데이터, d는 전송되는 거리이고 E_{elec}는 송출되거나 수신될 때 드는 에너지일 때 $E_{T_x}(l,d)$는 노드 x가 데이터를 보내는데 사용되는 에너지이고 $E_{R_x}(l)$는 노드 x가 데이터를 받는데 사용되는 에너지이다. 데이터를 보낼 때 사용되는 에너지는 다음과 같다[5].

$$E_{T_x}(l,d)=\begin{cases} l\times E_{elec}+l\times\epsilon_{fs}d^2, & d<d_{crossover} \\ l\times E_{elec}+l\times\epsilon_{mp}d^4 & d\geq d_{crossover} \end{cases} \quad (8)$$

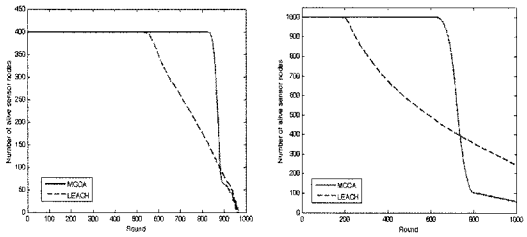
전송되는 거리에 따라서 free space channel model (ϵ_{fs})을 사용할지 multi-path fading channel model (ϵ_{mp})를 사용할지 정해지게 된다[4]. 데이터를 전송받을 때 사용되는 에너지는 $E_{R_x}(l)=l\times E_{elec}$이다.

위와 같은 radio model을 사용하여 표 1과 같은 parameter를 적용하여 실험을 하였다. 네트워크의 규모에 따라서 normal size area과 large size area로 나누어서 normal size area의 경우에는 100×100의 공간에서 400개의 노드가 위치해 있다고 가정을 하였고 base station은 (50, 200)에 위치해 있다고 가정을 하였다. Large size area의 경우에는 200×200의 공간에 1000개의 노드가 위치해 있다고 가정하고 base station은 (100, 350)에 위치해 있다고 가정하였다. 표 1의 parameter는 일반적인 WSNs에 관한 논문[4,7]을 따른다.

위와 같은 조건에서 본 논문은 MCCA와 WSNs에서의 대표적인 클러스터링 알고리즘을 사용하는 LEACH와의 성능을 비교하였다. 노드들의 위치는 area안에서 uniform random하게 정해지도록 하였다. 또한 CH_{opt}는 LEACH와 같은 5%로 실험하였다. 모든 노드가 작업을 시작한 후 첫 번째로 에너지를 전부 소진하여 죽는 노드가 발생하는 때까지의 시간을 lifetime이라고 한다면 이 기간은 WSNs가 정상적으로 작동하는 기간이라고

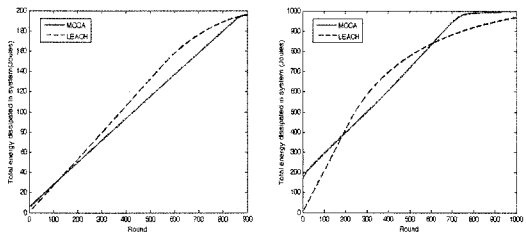
표 1 Parameter

Parameter	Normal size area	Large size area
Area	100×100	200×200
Location of BS	(50, 200)	(100, 350)
Number of node	400	1000
Initial energy	$0.5J$	$1.0J$
E_{elec}	$50nJ/bit$	
ϵ_{fs}	$10pJ/bit/m^2$	
ϵ_{mp}	$0.0013pJ/bit/m^4$	
$d_{crossover}$	$87m$	
Data size	4000 $bits$	
Signal size	200 $bits$	
Packet head size	200 $bits$	



(a) Normal size area (b) Large size area

그림 4 Round에 따른 살아있는 노드의 수

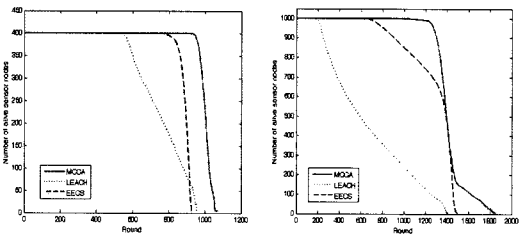


(a) Normal size area (b) Large size area

그림 5 Round에 따른 전체 누적 에너지 소모량

할 수 있다. 따라서 이 기간을 늘리는 것은 매우 중요하며 그림 4와 같이 MCCA는 LEACH보다 lifetime을 연장시켰음을 알 수 있다. 첫 번째 죽는 노드가 발생한 round를 기준으로 성능을 분석한다면 normal size area에서 MCCA는 LEACH보다 약 50%정도 향상되었고 large size area에서는 2배가 넘는 성능 향상을 보인다.

그림 5는 전체 네트워크에서 각각의 노드가 사용하는 에너지의 총합의 누적을 각 round별로 나타낸 것이다. MCCA는 LEACH와는 달리 초기에 이웃 노드를 설정하기 위한 에너지와 자신의 에너지를 base station에 보내는 에너지가 추가로 소모된다. 따라서 그림 5를 보면 MCCA는 첫 round부터 일정 에너지를 소모하고 시작하는 것을 볼 수 있다. MCCA는 초기에 일정 에너지를 소모하지만 효율적인 클러스터링으로 전체적인 에너지의 소모는 적게 한다. 그림 5를 보면 시작할 때 소모한 에너지는 MCCA가 더 많지만 중반에는 LEACH가 소모하는 에너지가 더 많아지게 된다. 중반 이후로 LEACH의 에너지 소모량이 완만해지는 이유는 살아있는 노드의 수가 점차 감소하기 때문이다. 이 시기를 그림4와 비교해서 보면 normal size area에서는 500round 이후, large size area에서는 200round 이후이다. MCCA도 마찬가지로 normal size area에서는 800round 이후, large size area에서는 600round 이후 에너지 소모량이 완만해진다. 모든 노드가 가지고 있는 에너지의 합은 같기 때문에 최종적으로 사용하는 에너지의 합은 비슷해지게 된다. 따라서 MCCA는 효율적인 클러스터링과 에



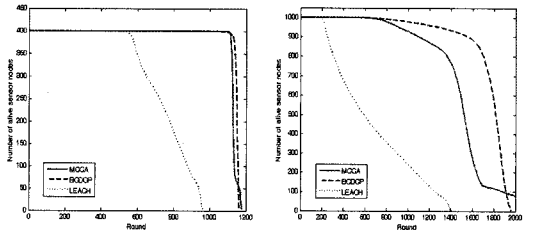
(a) Normal size area (b) Large size area

그림 6 Round에 따른 살아있는 노드의 수

너지가 적은 노드를 클러스터 헤드에서 제외시키는 방법으로 전체적으로 같은 에너지를 소모하면서 WSNs의 lifetime을 더욱 연장시킨다.

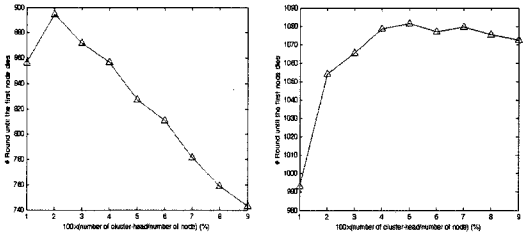
LEACH는 WSNs에서 클러스터링을 사용한 대표적인 논문으로 그 이후에 이를 응용한 많은 논문이 나왔다. 그림 6은 GPS를 사용하지 않는 알고리즘 중 LEACH와 비교하여 많은 성능 향상을 거둔 EECS와의 비교이다. 본 논문은 클러스터 헤드의 수에 따라 성능의 차이가 있으므로 CH_{opt}의 값을 normal size area에서는 2%, large size area에서는 1%로 하고 실험을 하였다. EECS 또한 본문에서 실험한 값과 같은 normal size area에서 k_{opt}는 6, T는 0.2, $R_{compete}$는 26, large size area에서 k_{opt}는 9, T는 0.15, $R_{compete}$는 40으로 놓고 실험을 하였다[4]. 이러한 parameter로 실험을 하게 되면 MCCA와 EECS에서의 선출되는 클러스터 헤드의 수가 비슷하게 된다. 따라서 MCCA가 비슷한 클러스터 헤더를 선출하였을 때 좀 더 효율적인 위치에서 클러스터를 구성하여 에너지를 아낄 수 있다는 것을 의미한다. MCCA는 lifetime을 기준으로 EECS보다 small size area일 때는 20%, large size area일 때는 58%가량의 성능 향상을 가져왔다.

MCCA는 GPS를 사용하지 않고 상대적인 위치 정보로만 클러스터를 구성하지만 에너지 효율 측면에서 GPS를 사용하는 알고리즘과 비슷한 성능을 보인다. 그림 7은 GPS를 사용하는 알고리즘인 BCDCP와 MCCA의 lifetime을 비교한 것이다. BCDCP는 클러스터링 이후



(a) Normal size area (b) Large size area

그림 7 Round에 따른 살아있는 노드의 수



(a) Base station=(50,200) (b) Base station=(50,120)

그림 8 $\frac{100\times CH_{opt}}{|N|}$에 따른 첫 번째로 죽는 노드가 발생하는 round

각 클러스터 헤드에서 base station으로 데이터를 전송할 때 멀티 홉 방식을 사용한다. 이와 같은 조건에서 실험하기 위하여 MCCA도 각 클러스터 헤드에서 base station으로 데이터를 전송할 때 멀티 홉 방식을 적용하여 실험을 하였다. 두 알고리즘에서는 멀티 홉 전송을 하기 위하여 base station과 선출된 클러스터 헤드들 사이에 Minmum Spanning Tree를 구성하는 방식을 사용한다. 그림 7의 Normal size area에서는 MCCA와 BCDCP의 lifetime이 거의 비슷한 것을 볼 수 있다. Large size area에서는 BCDCP가 MCCA보다 좀 더 좋은 에너지 효율을 보이지만 첫 번째로 죽는 노드가 발생하는 round는 비슷한 것을 볼 수 있다. 따라서 MCCA는 GPS를 사용하지 않고도 충분히 효율적인 클러스터링을 구현한다고 할 수 있다.

그림 8은 $100\times CH_{opt}/|N|$(적정 클러스터 헤드의 비율)에 따른 첫 번째로 에너지를 모두 소모하여 죽는 노드가 발생하는 round를 나타내었다. 100×100의 공간에서 400개의 노드가 위치해 있다고 가정을 하였고 base station의 위치는 (a)의 경우 (50, 200), (b)의 경우 (50, 120)라고 가정하였다. (a)와 같이 base station이 센서 노드들과 일정 거리 떨어져 있을 경우는 $100\times CH_{opt}/|N|$가 2%일 때 첫 번째 노드가 죽는 시간이 가장 길다. 하지만 (b)와 같이 base station이 센서 노드들과 가깝다면 LEACH[1]와 같이 $100\times CH_{opt}/|N|$가 5%일 때 첫 번째 노드가 죽는 시간이 가장 길다. 이와 같은 결과를 볼 때 적절한 클러스터 헤드의 수 CH_{opt}는 base station과 각 센서 노드들 사이의 거리에 크게 영향을 받는다고 할 수 있다. 또한 앞에서 실험하였듯이 멀티 홉과 같은 방법으로 데이터를 전송함으로써 base station과 센서 노드들 사이의 거리가 멀더라도 에너지 효율을 증대시킬 수 있다.

5. 결 론

본 논문에서는 GPS를 사용하지 않고 효율적인 클러

스터링을 하는 알고리즘 Max k-Cut based Clustering Algorithm(MCCA)을 제안하였다. MCCA는 base station에서 클러스터 헤드를 지정해 주는 중앙 처리 방식의 알고리즘으로 Max k-Cut Problem을 사용하여 클러스터 헤드의 위치를 전체 네트워크에서 고르게 분포하도록 구성하여 준다. 또한 에너지가 다른 노드들보다 많이 소비된 노드들은 클러스터 헤드 선출에서 제외시킴으로써 에너지를 모두 소비하여 죽는 노드의 발생 시간을 늦추었다. 실험결과 MCCA는 LEACH보다 50%, EECS보다 20% 향상된 성능을 보였다. 또한 GPS를 사용한 알고리즘인 BCDCP와의 비교에서도 비슷한 성능을 보였다.

MCCA는 클러스터 헤드의 선출 비율에 따라서 성능이 많이 달라진다. 따라서 적절한 수의 클러스터 헤드의 수를 찾는 문제도 하나의 중요한 논점이 될 수 있다. 차후 이 문제에 대한 연구가 더 이루어져야 할 부분이다.

참 고 문 헌

[1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Vol.8, pp. 3005-3014, Jan. 2000.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, Vol.1, No.4, pp. 660-670, Oct. 2002.

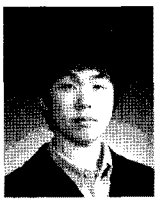
[3] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks," *Communications Magazine, IEEE*, Vol.43, issue 3, Mar. 2005.

[4] M. Ye, C. Li, G. Chen, and J. Wu, "EECS : An Energy Efficient Clustering Scheme in Eireless Sensor Networks," *Performance, Computing and Communications Conference(IPCCC) 24th IEEE International*, pp. 535-540, April 2005.

[5] A. Frieze and M. Jerrum, "Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION," in *Proceedings of the 4th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pp. 1-13, 1995.

[6] J. D. Cho, S. Raje, and M. Sarrafzaden, "Fast Approximation Algorithms on Maxcut, k-Coloring, and k-Color Ordering for VLSI Applications," *IEEE Transactions on Computers*, Vol.47, issue 11, pp. 1253-1266, Nov. 1998.

[7] Y. J. Han, S. H. Park, J. H. Eom, and T. M. Chung, "Energy-Efficient Distance Based Clustering Routing Scheme for Wireless Sensor Networks," *Computational Science and Its Applications - ICCSA 2007*, Vol.4706, pp. 195-206, Aug. 2007.



김 재 환

2007년 서강대학교 컴퓨터공학과 졸업 동대학원 석사(2009). 관심분야는 Wireless Network, System Modeling and Optimization

장 형 수

1994년 미국 Purdue University, 전기 및 컴퓨터공학과 졸업, 동대학원 석사, 박사(1996, 2001). 2001년 9월~2002년 6월 Institute of Systems Research, University of Maryland, College Park, Research Associate, 2002년 6월~2003년 2월 고려대학교 정보통신 기술 공동연구소 연구교수, 2003년 3월~2006년 2월 서강대학교 공과대학 컴퓨터공학과 조교수. 2007년 3월~서강대학교 공과대학 컴퓨터공학과 부교수