LEACH 프로토콜 기반 망 수명 개선 알고리즘

정회원 추 영 열*°, 최 한 조*, 권 장 우**

Algorithm Improving Network Life-time Based on LEACH Protocol

Young-yeol Choo*°, Han-jo Choi*, Jang-woo Kwon** *Regular Members*

요 약

본 논문에서는 환경 감시 등 무선 센서네트워크 응용을 위한 LEACH 프로토콜 기반의 망 수명 개선 알고리즘을 제안한다. 첫 째, LEACH 프로토콜에 따른 클러스터 구성시 각 클러스터에 노드 수를 균등하게 배분한다. 둘 째, 클러스터 형성시 각 클러스터별로 헤더 역할을 담당할 노드의 순서를 설정한다. 이후, 정해진 순서에 따라 헤더가 일정 수의 패킷을 수신후 다음 노드에게 헤더 역할을 양도한다. 이렇게 함으로써 각 노드의 에너지 소비를 균등하게 하여 망 전체의 수명이 증대되도록 하였다. 시뮬레이션 결과 망 수명은 LEACH에 비해 두 배 증가하였고 망 전체의 에너지 소비는 1/4로 감소됨을 보여주었다.

Key Words : Clustering Algorithm, WSN, LEACH, Routing Algorithm

ABSTRACT

This paper proposes an algorithm to improve total network lifetime for Wireless Sensor Network (WSN) application such as environmental condition monitoring systems based on LEACH protocol. Firstly, the algorithm had equal number of nodes allocated to each cluster at cluster set-up phase where it abided by LEACH protocol. Secondly, at cluster set-up phase, each node was determined the order to be cluster header of the cluster which it had joined. After then, the role of a cluster head delivers to the next node according to determined order when the cluster head has received certain number of packets. With above method energy consumption of each node made equal and overall network lifetime was increased. Simulation results shows that overall network lifetime of proposed algorithm was increased two times than that of LEACH and total energy consumption was one forth of that of LEACH protocol.

Ⅰ. 서 론

무선 센서 네트워크(Wireless Sensor Network)는 RFID와 더불어 최근 부상하는 유비쿼터스 네트워크의 핵심 기술로, 환경 감시나 목표물 추적, 고속도로 교통 정보 관리, 건물 내부 감시, 위험물 및 화재 감시등의 다양한 잠재적인 응용 분야를 가지고 있다. 무선 센서 네트워크는 센서 모듈과 네트워크 모듈을 갖는 센서 노드들로 이루어지며 많은 수의 센서 노드들이 목표 지역에 배치되어 유기적으로 동작하며 하나의 네트워크를 형성한다. 센서노드(Sensor Node)는 제한된 배터리와 메모리, 연산 처

※ "본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (NIPA-2010-(C-1090-1021-0006))

* 동명대학교 컴퓨터공학과(yychoo@tu.ac.kr, hanjo1001@nate.com) (° : 교신저자)

** 경원대학교 컴퓨터공학과(jwkwon@kyungwon.ac.kr)

논문번호 : KICS2010-02-074, 접수일자 : 2010년 2월 11일, 최종논문접수일자 : 2010년 7월 19일

리 속도 그리고 통신거리 등의 제약이 있으며 이런 제약을 가지고 있음에도 장기간 동작하는 것을 필요로 하므로 에너지를 최대한 절약하는 것이 필수적이다. 이러한 이유 등으로 인해 유선 네트워크와는 다른 라우팅 프로토콜의 설계가 요구 된다. 이동식 크레인과 같은 설비 고장진단의 경우 네트워크의 Life-time을 최대화하는 것이 큰 문제점이다. 현재에도 무선 센서 네트워크 전체 센서노드의 Life-time을 최대화하기 위한 연구가 많이 이루어지고 있다[1]-[6].

계층적 혹은 클러스터 기반의 라우팅은 기존에 유선 네트워크에 비해 확장성과 효율적인 통신에 관련된 장점을 가진 기술로 알려져 있다. 또한, 전체 시스템의 확장성, 수명 및 에너지 효율적인 면에서도 큰 장점을 가지고 있다[2]-[7]. 본 논문에서는 무선 센서 네트워크의 Life-time을 기존의 클러스터링 기반 라우팅 알고리즘보다 더 연장할 수 있는 방법을 제안 한다. LEACH (Low Energy Adaptive Clustering Hierarchy)[8],[9]의 경우 클러스터링 기반의 가장 대표적인 라우팅 프로토콜로서 가장 많은 비교대상으로 사용되고 있으며 라우팅 프로토콜의 정확한 소스(Source)와 다양한 정보를 제공하고 있다. 본 논문에서도 LEACH를 비교대상으로 선택하였다.

제안하는 라우팅 알고리즘은 LEACH 에서 불균등한 에너지 소비를 최소화 하고, 네트워크 전체의 비효율적인 에너지 소비를 막아 네트워크 전체의 Life-time을 보다 더 연장하는데 목적이 있다. LEACH를 비롯한 대부분의 클러스터링 기반의 라우팅 알고리즘은 각 클러스터에 서로 다른 개수의 클러스터 멤버가 연결되어 있어 각 클러스터들은 불균등하게 에너지가 소비 되고, 매 라운드마다 클러스터 헤더를 재 선출하는 과정에서 많은 에너지가 소비되며 이로 인해 네트워크 전체의 Life-time도 짧아진다.

제안하는 라우팅 알고리즘은 최초 클러스터링이 형성될 때 각각의 클러스터 형성에 참여했던 클러스터의 멤버들의 개수가 동일하게 하여 Join하고, 멤버들은 클러스터 헤더와 연결 시 정해진 순서대로 다음 클러스터 헤더가 됨으로써 클러스터 헤더와 클러스터 멤버들 간의 클러스터 형성 시 소모되는 에너지를 최소화 하여 네트워크 전체의 생존시간을 최대화로 할 수 있다. 헤더 재 선출 방법은 모든 클러스터 헤더는 클러스터 멤버들로부터 받은 패킷의 개수를 체크하여 클러스터 헤더가 사전에 정의되어 있었던 임계값 이상의 패킷을 받게 되면 사전에 정해져 있는 멤버 노드에게 헤더의 권한을 넘김으로써 클러스터 헤더가 재 선출된다.

본 논문은 다음과 같이 구성된다. 2장에서는 무선센서 네트워크에서 사용하는 클러스터링 기반의 라우팅 프로토콜에 대해 소개하고, 3장에서는 LEACH에서 개선된 라우팅 알고리즘을 제안한다. 4장에서 제안된 라우팅 알고리즘과 LEACH의 성능을 시뮬레이션을 통해 성능을 평가하고 분석하고, 5장에서 결론을 맺는다.

Ⅱ. 관련 연구

제안하는 라우팅 알고리즘과 비교분석이 될 LEACH는 클러스터 기반의 대표적인 프로토콜로서 분산된 다수의 클러스터를 가진다. Base Station은 센서노드들과는 떨어진 지정된 위치에 있으며, 모든 노드들은 동일한 속성을 가진다. LEACH의 특성은 클러스터 형태가 계속해서 변경되고 센서노드들 간에 스스로 클러스터 헤더(CH)를 선정하는 것이다. 그래서 선택된 클러스터 헤더 노드의 에너지 소모를 줄이고, 네트워크 전체 에너지 소모 부하를 센서노드 간에 분산하여 나눌 수 있다. 그리고 데이터는 병합하여 Base Station에 전송하기 때문에 전송되는 데이터양도 적다. 먼저, Set-up 단계에서는 클러스터 헤더를 선출한다. CH의 선출은 각 노드가 스스로 선출을 한다. CH가 선출되고 나면, CH는 광고 메시지를 방송하여 클러스터 멤버(CM)노드들을 모은다. Steady 단계에서는 Set-up 단계에서 설정된 클러스터의 스케줄링에 따라 멤버노드가 CH에 데이터를 전송한다. 전 멤버노드들은 평상시에 Sleep 상태로 있다가 자신의 전송시간이 되면 깨어나 데이터를 전송하고 다시 Sleep 상태로 돌아가기 때문에 전송 이외 필요 없는 에너지 소모를 줄일 수 있다[3]-[5].

또한, 기존의 계층적 기반 라우팅뿐만 아니라 클러스터 헤더 재 선출기법들도 여러 가지가 있다. 선점형 스케줄링 방식의 Round-Robin을 이용해 LEACH에서 정해진 순서대로 클러스터 멤버가 클러스터 헤더로 재 선출되는 기법[10], LEACH에서 클러스터 멤버의 통신 참여횟수 반영하여 가장 적은 통신 횟수를 가진 멤버가 다음의 헤더로 재 선출되는 기법[11]등이 있다.

그림 1은 LEACH의 타임라인과 구조를 보여주고 있다. Set-up 단계와 Steady 단계를 합쳐 한 라운드 로 정의 한다. 한 프레임은 클러스터 멤버가

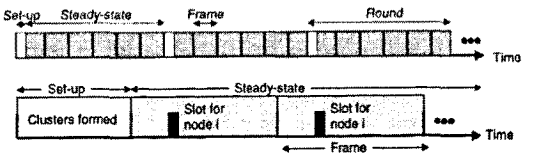


그림 1. LEACH의 타임라인(Time Line) & 구조

클러스터 헤더에게 데이터를 보내도록 분할되어 있는 것으로 정의 한다.

Ⅲ. 제안하는 알고리즘

본 논문은 LEACH을 기반으로 하여 다음의 두 가지 문제점에 대한 개선 알고리즘을 제안한다.

첫째, 기존 클러스터링 기반의 알고리즘에서는 각각의 클러스터들은 클러스터에 Join 되어 있는 클러스터 멤버 수가 달라 네트워크 전체 에너지 소비가 불균등하고, 비효율적 이였다. 이를 개선키 위해 본 논문에서는 각각의 클러스터에 동일한 개수의 클러스터 멤버들을 Join 함으로써 네트워크 전체 에너지 소비를 균등하게 함과 동시에 에너지 소비의 효율성을 높이고자 한다.

둘째, 기존의 클러스터링 기반의 라우팅 알고리즘과 달리 클러스터 헤더 재 선출시 에너지 소모가 많은 투표 단계 없이 헤더를 선정한다. 제안하는 라우팅 알고리즘은 최초에 클러스터링이 형성될 때 각각의 클러스터링에 참여 하였던 클러스터 멤버들이 정해진 순서대로 각 클러스터의 헤더로 선출된다. 클러스터 헤더의 선출은 각각의 헤더가 가지고 있는 임계값(Threshold) 이상의 패킷을 받게 되면 헤더는 사전에 정해져 있던 다음 노드에게 헤더의 권한을 넘김으로써 새 클러스터 헤더가 선출된다. 이를 통해 헤더 선출 과정에서 소모되는 에너지를 최소화 할 수 있어 네트워크 Life-time 연장을 기할 수 있다. 상세 과정은 다음과 같다.

3.1 Set-up 단계

본 논문에서는 Set-up 단계에서 헤더 선출과 클러스터 형성까지 하는 것으로 구성한다. 또한 제안하는 라우팅 알고리즘의 최초의 헤더 선출 방법은 LEACH와 동일한 방법으로 한다.

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

P : 클러스터 헤더의 구성 비율
r : 현재의 라운드
G : 지난 $1/P$ 라운드들에서 클러스터 헤더가 아니었던 노드들의 집단

$T(n)$[1]은 노드 n이 클러스터헤드가 되기 위한 임계치 값으로 클러스터 형성 단계는 다음과 같다.

먼저, Base Station에서 전 노드를 대상으로 Hello 메시지를 브로드캐스팅 한다. Hello 메시지는 전 노드들을 Sleep 상태에서 Wake-up 상태로 깨우면서 노드가 통신이 가능한지 확인하는 역할을 한다. Hello 메시지를 받은 노드들은 식 (1)[8],[9]을 통해 본인이 클러스터 헤더인지 아니면 일반 노드(Non_CH)인지에 대한 정보를 Base Station에게 알려 준다. 이 정보를 받은 Base Station은 클러스터 헤더의 개수와 일반 노드의 개수를 확인한다. Base Station은 아래의 식 (2)를 이용해 하나의 클러스터 헤더에 Join 될 수 있는 클러스터 멤버의 수를 정해 클러스터 헤더에게 *Member_Threshold*의 값을 보낸다.

$$Member\_Threshold = \lceil NCH/Total\_CH \rceil + 1 \quad (2)$$

NCH (Non Cluster Header)는 일반노드의 개수를, *Total_CH*은 클러스터 헤더의 전체 개수를 의미한다. 식 (2)에서 뒤에 1을 더하는 것은 헤더에 Join 된 클러스터 멤버들의 수가 *Member_Threshold* 값을 넘지 않게 하기 위해서 이다. 이렇게 함으로써 모든 클러스터 헤더에는 *Member_Threshold* 개 이하의 클러스터 멤버가 Join 하게 된다.

Set-up 과정은 다음과 같다.

① Base Station은 모든 노드들에게 살아 있는지, 헤더 선출을 위해 Hello 메시지를 Broadcasting 함.
② Base Station은 전체 일반노드들의 개수와 클러스터 헤더의 개수를 확인한 후, *Member − Threshold* 값을 연산해 클러스터 헤더에게 알려줌.
③ 클러스터 헤더로 선출된 노드는 주변의 모든 노드들에게 AD_Messag를 보내 본인이 클러스터 헤더임을 알리고 주변의 일반노드들로부터 Join_Message를 기다림.
④ 클러스터 헤더로부터 AD_Message를 받은 일반노드들은 자신과 가까운 클러스터 헤더에게

클러스터 멤버가 되겠다는 Join_Message와 Node_id를 보냄.

⑤ 클러스터 헤더는 주위 노드들로부터 Join_Message와 Node_id를 받은 노드 순서대로 다음번 클러스터 헤더로 정의하고 순서를 부여 함.

⑥ 클러스터 헤더는 Next_CH_order 테이블에 다음 클러스터 헤더 순서대로 Node_id를 저장.

⑦ 저장된 Next_CH_order 테이블을 자신의 클러스터 멤버들에게 전달함.

Set-up 단계를 의사코드로 그림 2에 표시하였다.

본 논문에서 제안하는 라우팅 알고리즘은 클러스터 형성 시 클러스터 헤더는 클러스터 헤더의 선출 순서에 대한 정보 테이블을 가진다. 정보 테이블은 헤더 재 선출을 위한 다음과 같은 정보를 포함한다.

- *Node_id* : 클러스터 멤버들의 고유 노드 id. 클러스터 헤더는 자신에게 Join되어 있는 클러스터 멤버들의 id를 저장.
- *Header_Threshold* : 헤더가 멤버와 통신할 수 있는 최대 패킷 수. 클러스터 헤더가 *Header_Threshold* 번 이상의 패킷을 받게 되면 클러스터 헤더 재 선출.
- *Member_Threshold* : 하나의 클러스터 헤더가 가질 수 있는 최대 클러스터 멤버들의 개수. *Member_Threshold* 값은 식(2)에 의해 구해짐.

```
1: r <- Current Round
2: M_Th <- Member_Threshold
3: H_Th <- Header_Threshold
4: for(i=0 to Total_Nodes) do // 헤더를 선출 한다.
5: if(random< T(n) = { P / (1 - P*(r mod 1/P)) } AND node_i battery
   > 0) then
6:       node_i <- CH // 헤더가 선출 되었다.
7:       Total_CH = CH++ // 헤더의 전체 수를 더한다.
8:    end if
9:  end for
10: Total_CH++ // BS에서 전체 CH수를 Counting
11: Total_Nodes++ // BS에서 전체 Node 수를 Counting
12: Total_NCHs = Total_Nodes - Total_CH
13:  M_TH = ⌈(Total_NCHs/Total_CH)⌉ +1
14: // Member_Threshold 값을 정함.
15: for(j=0 to Total_Nodes) do
16:    if(node_i == CH!) then // 멤버로 선정.
17:       node_i <- CM
18:       for(k<0 to M_Th) do
19:       CH_join = CM++ // 헤더에 Member_Threshold
          개 까지 join.
20:       end for
21:    end if
22: end for
23: if(node_i == CH) then
24:    CHnode_i.H_Th = CM++ // 해당 헤더의 Header_
       Threshold값 설정.
25: end if
```

그림 2. Set-up 단계의 의사코드

표 1과 같이 정보테이블의 예를 통해 전술된 Set-up 단계를 설명 한다.

표 1은 클러스터 헤더의 정보 테이블로 전체 20개의 노드들 중에 3개의 클러스터 헤더가 선출되어 클러스터가 형성된 것을 보여 준다. CH_1, CH_2, CH_3은 클러스터 헤더의 번호를 의미한다. *Node_id*는 노드의 고유 *id*이고, *Next_CH_order*는 클러스터 내에서 헤더가 되는 순서를 의미한다. 표 1의 경우 $Non_CH = 17, CH = 3$로 구성되어지고, *Member_Threshold* 값은 식 (2)에 따라 '6'이 된다. *Header_Threshold* 값은 클러스터 헤더에 Join된 클러스터 멤버의 수가 그 값으로 된다. CH_1의 경우 *Member_Threshold* = 6 이지만 Join된 클러스터 멤버는 5개인 관계로 *Header_Threshold* = 5로 설정된다. 이는 최초 클러스터 헤더 선출 시 클러스터 헤더로 먼저 선출된 노드들부터 *Member_Threshold*값의 개수만큼 클러스터 멤버와 연결하기 때문에 마지막에 선출된 헤더는 *Member_Threshold* 값 보다 적거나 같은 수의 클러스터 멤버가 Join

표 1. 클러스터 헤더의 정보 테이블 예

No	CH_1		CH_2		CH_3	
	Node_id	Next_CH_order	Node_id	Next_CH_order	Node_id	Next_CH_order
1	0	1	1	5	4	4
2	3	3	2	2	5	5
3	9	2	6	4	8	3
4	10	4	13	1	16	2
5	18	5	14	3	17	1
6			15	6	19	6
	Header_Threshold=5		Header_Threshold=6		Header_Threshold=6	
	Member_Threshold=6		Member_Threshold=6		Member_Threshold=6	

될 수 있다.

3.2 Steady 단계

일반적으로 LEACH를 비롯해 클러스터링 기반의 라우팅 알고리즘의 대부분은 Base Station과 클러스터 헤더간의 통신이 끝나고 나면 헤더 재 선출을 위해 다시 Set-up 단계로 가서 헤더를 재 선출 한다. 하지만 제안하는 알고리즘은 Set-up 단계를 거치지 않고 헤더가 *Header_Threshold* 값만큼 데이터를 수신하면 첫 번째 Set-up 단계에서 정해진 순서에 따라 다음 노드가 헤더가 된다.

먼저 그림 3은 LEACH에서의 라운드(Round)의 구조를 보여 준다. 그림 4는 본 논문에서 제안 하는 라우팅 알고리즘의 라운드 구조이다. Set-up 단계는 최초 클러스터링이 형성 될 때만 실행되고 이후부터는 Steady 단계만 실행된다. 클러스터 헤더와 멤버는 매 라운드의 마지막에는 노드들이 살아 있는지 확인 하는 *ACK+Flag* 신호를 주고받는다.

최초의 Set-up 단계를 실시해 클러스터가 형성된 이후 각각의 센서노드들 간(CH-to-CM, CH-to-BS) 클러스터 헤더의 재 선출시 까지 데이터 통신을 할 뿐 재차 Set-up 단계는 실시되지 않는다. 매번의 라운드는 클러스터 헤더가 재 선출되는 시점을 1 라운드로 정의하고 Frame은 클러스터 멤버가 클러스터 헤더에게 데이터를 보내는 것을 1 Frame으로 정의 한다. 매 라운드마다 클러스터 헤더는 해당 클러스터 주위의 센서노드들에게 Flag 메시지를 보내 기존에 Join 되어있던 노드가 살아있는지 확인한다. 그리고 클러스터 주위에 기존의 노드이외의 다른 노드가 있는지 확인하도록 한다. 이 Flag 메시지를 받은 기존의 클러스터 멤버들은 본인이 살아있다면 ACK 메시지를 보냄으로써 클러스터를 유지하고 기존의 클러스터에서 계속 통신참여를 한다. 또한, 통신 중간에 링크(Link)가 끊어져 통신이 되지 않았던 기존의 클러스터 멤버들도 이 때 원래의 클러스터에 다시 Join 할 수 있다.

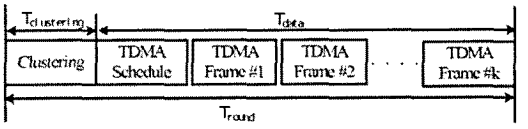


그림 3. LEACH에서의 라운드 구조

3.3 Reconfiguration 단계

본 논문에서 제안하는 라우팅 알고리즘의 Reconfiguration 단계는 Set-up 단계를 다시 하여 전체의 클러스터를 다시 구성하는 것이 아니라 Steady 단계에서 빠졌거나 소속된 클러스터와 물리적으로 멀리 있을 경우 클러스터를 변경 할 수 있게 하는 단계이다. 이 단계는 Round가 끝난 시점에만 가능하며 헤더는 매 라운드마다 헤더의 신호 범위 내에 AD_Msg를 *Flag* 메시지에 담아 발송한다. 클러스터 헤더는 한 라운드 중 소속된 클러스터에서 빠져나간 노드나 이중으로 클러스터에 물려 있는 노드를 알아차리게 된다. 통신에 참여 하지 않거나, 클러스터를 옮기려고 하는 노드는 참여 하고자 하는 헤더에게 Join_Msg를 보내고 *node_id table* 의 마지막부분에 추가 하여 클러스터에 Join 하도록 한다. 또한, 클러스터 헤더가 죽어 클러스터 헤더로부터 *Flag* 메시지가 오지 않으면 사전에 선정되어 있던 노드가 클러스터 헤더로 재 선출된다. 본 단계는 하나의 클러스터가 완전히 파괴되어 더 이상 클러스터형성이 힘든 노드들을 다시 하나의 클러스터로 재구성 하는데 목적이 있다. 기존에 Set-up 단계에서 *Member_Threshold* 값만큼 만 Join이 가능 하도록 되어 있기 때문에 클러스터 멤버 노드의 이동성(Mobility)으로 인한 클러스터의 이동이 아닌 경우에는 *Member_Threshold* 값을 유지 하도록 한다.

Ⅳ. 실험 결과 및 성능 비교분석

4.1 실험 변수

이번 장은 본 논문에서 제안하는 라우팅 알고리즘과 LEACH와 성능을 비교 분석을 해보았다. 실

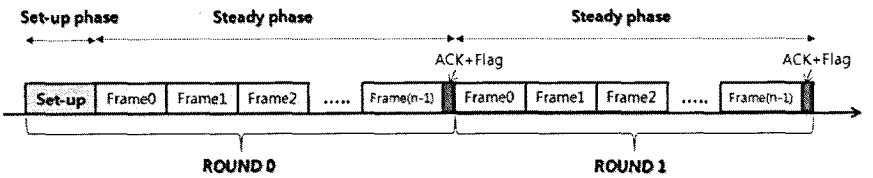


그림 4. 제안한 알고리즘의 라운드 구조

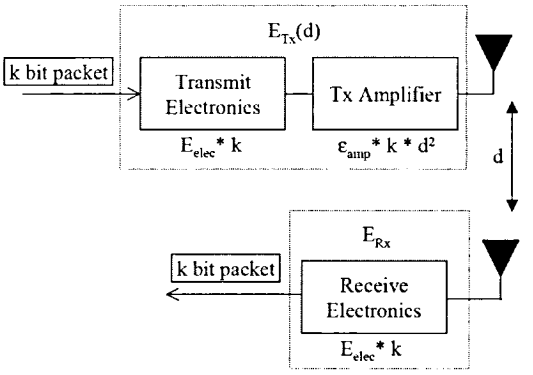


그림 5. 1:1 Data 전송 시 소비되는 에너지

험에서 사용되는 에너지 소비는 아래와 같다.

일반적으로 RF(Radio Frequency)를 이용해 Data 전송할 때 소비되는 에너지는 그림 5와 같다. Data를 보내는 노드에서는 k bit의 Packet를 보낼 때 k bit에 대한 전기적 에너지와 Data를 받는 노드까지의 거리 제곱(d^2) 만큼 신호를 증폭하면서 에너지를 소비한다. 반면 Data를 받는 노드는 k bit의 Packet을 처리하는 전기적 에너지만 소비된다.

본 논문에서는 클러스터 생성 시 전체 노드들은 1-Hop으로 통신한다고 가정 한다. 각각의 노드들은 k bit의 데이터를 d 거리 전송을 하고 노드들은 $M \times M$ 구역에 l 개의 클러스터가 있다고 가정하고 소모되는 에너지의 수식을 구한다. 아래 식 (3)은 두 센서노드 사이에서 전송 노드가 k bit를 보낼 때 소비되는 에너지를 나타내며 이때 소비되는 전기적 에너지(E_{elec})와 k bit를 d거리까지 보내기 위해 신호세기를 증폭한 전기적 에너지의 합으로 표현된다. 식 (4)는 k bit를 수신하는 노드에서 소비되는 에너지이다[8].

$$E_{Tr}(k,d) = E_{Tx\_elec}(k) + E_{Tx\_amp}(k,d)$$
$$E_{Tx}(k,d) = E_{elec} * k + E_{amp} * k * d^2 \quad (3)$$

$$E_{Rx}(k) = E_{Rx\_elec}(k)$$
$$E_{Rx}(k) = E_{elec} * k \quad (4)$$

여기서 $E_{Tx}(d)$는 d거리의 두 노드에서 데이터를 전송할 때 소비되는 에너지를, E_{Rx}는 데이터를 수신할 때 소비되는 에너지를 나타낸다. 그리고 E_{Tx_amp}은 데이터 전송 시 라디오의 신호세기(Signal Strength)를 증폭시키는데 소비되는 에너지를 의미한다.

식 (5)[11]에서 첫 번째 식은 하나의 클러스터 내 있는 노드들 간의 소비되는 에너지로 주로 클러스터 헤더와 클러스터 멤버(CH-to-CM)간의 통신시 발생한다. 두 번째 식은 클러스터 헤더와 Base Station (CH-to-BS) 간에 소비되는 에너지이다.

$$E_{Tr}(k,d) = \begin{cases} E_{elec} * k + E_{fs} \frac{M^2}{2\pi l} * k, & among\ sensor\ nodes \\ E_{elec} * k + E_{mp} * k * d^4, & CH-BS \end{cases} \quad (5)$$

식 (5)에서 E_{elec}은 노드의 전기적인 에너지, E_{fs}은 클러스터 구성에 필요한 신호를 보낼 때 라디오 신호세기를 증폭할 때(Free Space Channel Model) 소비되는 에너지이고, E_{mp}은 싱크노드로 수집된 데이터를 전송하는데 사용되는 신호세기를 증폭할 때(Multi-Path Channel Model)소비되는 에너지를 의미 한다. 그리고 N 개 센서노드들 중에 l 개의 클러스터 가지고 있을 때 각각의 클러스터에는 N/l 개의 노드들이 하나의 클러스터로 구성된다. 즉, l 개의 클러스터와 각각의 클러스터에는 $(N/l)-1$ 개의 클러스터 멤버로 구성된다. 따라서 식 (6)은 하나의 클러스터 헤더가 클러스터 멤버들로부터 메시지를 받을 때 소비되는 에너지를 의미하고, 식 (7)은 각각의 클러스터 헤더가 클러스터 헤더 재 선출 시 다음의 헤더에게 헤더의 역할을 넘길 때 각각의 클러스터에서 소비되는 에너지를 나타낸 식이다[11].

$$E_{Rx}(k) = E_{elec}\left(\frac{N}{l} - 1\right) * k \quad (6)$$

$$E_{Sx}(k) = E_{elec} \frac{N}{l} * k \quad (7)$$

위 식 (5), (6), (7)에서 보면, Set-up 단계에서 클러스터 헤더와 클러스터 멤버(CM)간의 통신, Base Station과 클러스터 헤더간의 통신에서 서로 다른 에너지 소비 형태를 가지게 됨을 알 수 있다. 위의 식들을 이용하여 Set-up 단계 전체 에너지 소비를 계산하면 Set-up 단계에서 클러스터 헤더가 소비하는 에너지와 클러스터 멤버가 소비하는 에너지는 각각 식 (8), (9)와 같다[11].

$$E_{Setup\_CH} = E_{elec} * k + E_{mp} * k * d^4 + E_{fs} \frac{M^2}{2\pi l} * k + E_{Rx}(k) + E_{Sx}(k) \quad (8)$$

$$E_{Setup\_NonCH} = E_{elec} * k + E_{fs} \frac{M^2}{2\pi l} * k + 2E_{Rx}(k) \quad (9)$$

식 (8)과 (9)로부터 Set-up 단계에서 소비되는 전체 에너지는 식 (10)과 같이 된다.

$$E_{Setup\_Total} = l * (E_{Setup\_CH} + E_{Setup\_NonCH}) \quad (10)$$

Steady 단계에서 클러스터 헤더의 에너지 소비량과 클러스터 멤버의 에너지 소비량은 식 (11), (12)로 표현된다.

$$E_{Steady\_CH} = E_{Rx}(k)\left(\frac{N}{l} - 1\right) + E_{Sx}(k)\frac{N}{l} + E_{Tx}(k,d) + E_{Tx}(h,d) \quad (11)$$

$$E_{Steady\_NonCH} = E_{Tx}(k,d) + E_{Tx}(h,d) \quad (12)$$

여기서 h는 매 라운드가 끝나고 $ACK+Flag$ 메시지를 보낼 때 쓰이는 bit를 의미한다. 제안하는 라우팅 알고리즘은 헤더의 재 선출이 Steady 단계에서 이뤄지기 때문에 네트워크 전체 에너지 소비가 Steady 단계에서 가장 많이 발생한다.

따라서 네트워크의 전체 에너지 소비량은 Set-up 단계의 식 (8), (9)과 Steady 단계의 식 (11), (12)로 알 수 있다. 이로부터 네트워크 전체의 에너지 소비량을 구하면 식 (13)과 같다. 식 (13)은 한 번의 Set-up 단계와 n번의 Steady 단계에서 소비되는 에너지의 양을 보여준다.

$$E_{total} = (E_{Setup\_CH} + E_{Setup\_NonCH}) + n * (E_{Steady\_CH} + E_{Steady\_NonCH}) \quad (13)$$

4.2 실험 환경

본 실험에서 제안하는 라우팅 알고리즘과 LEACH를 비교한 항목들과 그 의미를 표 2에 나타내었다. 총 네트워크 에너지, Transmit Nodes / Round, Round / 1-Node, 한 라운드 당 CH와 NCH의 에너지 소비량 등 총 4가지 항목에 대해 시뮬레이션을 통해 비교 분석하였다.

시뮬레이션에 사용된 주요 파라미터와 그 값들을 표 3에 요약하였다. 표 3의 파라미터 중 d 는 노드 간 거리이다. 실험에서는 노드들이 랜덤으로 배치되기 때문에 특정하게 노드간의 거리는 정의 하지 않았다. 그리고 Power는 노드 하나당 배터리(에너지)를 말한다. 노드 하나당 3.0V 로 정의한다. Sch_Msg는 최초 클러스터링 시 클러스터 헤더가 전 센서노드들을 대상으로 AD_Message를 송신 할 때 사용하도록 정의 하였다.

표 2. 실험 항목

실험항목	내 용
총 네트워크 에너지	실험 60시간 동안 네트워크 전체에 남아있는 에너지
Transmit nodes/ round	실험 네트워크 전체에서 매 라운드 마다 죽지 않고 통신에 참여하는 노드 수
Round / 1-Node	하나의 노드가 통신에 참여하는 라운드 횟수
한 라운드 당 CH와 NCH의 에너지 소비량	하나의 클러스터에서 한 라운드에 클러스터 헤더와 클러스터 멤버가 소비하는 에너지

표 3. 시뮬레이션에 사용된 주요 파라미터들

Parameter	Value	Unit	Parameter	Value	Unit
E_{elec}	50	nJ/bit	k	2048	bit
E_{fs}	10	pJ/bit/ m^2	h	8	bit
E_{mp}	0.0013	pJ/bit/ m^4	M	200	m
E_{Rx}	10	nJ/bit/ signal	d	Random	m
E_{Sx}	10	nJ/bit/ signal	*Power*	3.0	power /node
N	200	개	*Sch_Msg*	64	bit

4.3 실험 결과 및 분석

본 논문에서 제안하는 라우팅 알고리즘과 LEACH의 성능평가를 위해 시뮬레이션을 수행하였다.

먼저 네트워크 전체의 소비되고 있는 에너지의 양에 대한 시뮬레이션 결과값을 그림 6에 그래프로 나타내었다. 그래프에서 보이는 것처럼 본 논문에서 제안한 라우팅 알고리즘이 LEACH 보다 망 전체의 Life-time에 있어 약 2.7배 향상된 것으로 나타났다. 즉, 시뮬레이션 결과는 60시간이 지난 후에 제안한 라우팅 알고리즘을 사용하면 LEACH 보다 망 전체로 약 2.7배 이상 더 많은 에너지를 가짐을 보여준다.

또한, 그래프의 기울기를 보면 네트워크 전체에서 에너지 소비가 균등하게 이뤄지고 있는지 알 수 있다. 제안하는 라우팅 알고리즘의 경우 일정한 기울기를 보여줌으로써 에너지가 균등하면서 일정하게 소비되고 있음을 알 수 있다. 반면에 LEACH의 경

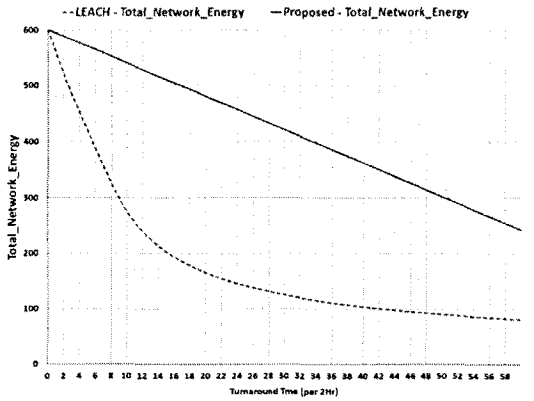


그림 6. 총 네트워크 에너지 결과 그래프

우 각각 클러스터마다 소비되는 에너지 양이다. 이로 인해 네트워크 전체의 에너지 소비가 불균등해지면서 그림 6의 LEACH와 같이 급격하게 낮아지는 그래프의 기울기를 볼 수 있다. 제안하는 라우팅 알고리즘의 경우 첫 번째 라운드 외에는 Set-up 단계가 다시 발생 하지 않고 Steady단계만 지속되어 일정한 에너지의 소비 형태를 보이는 것과 달리 LEACH의 경우에는 매 라운드 때 마다 발생하는 Set-up단계와 각각의 클러스터에서 불균등하게 소비되는 에너지로 인해 급격하게 에너지 소비가 증가하는 것을 볼 수 있다.

그림 7은 노드 전체의 Life-time을 보여 준다. 즉, 네트워크상에서 살아서 통신이 가능한 노드의 수를 나타낸 그래프이다. 위의 그래프 역시 제안하는 라우팅 알고리즘의 노드들이 더 오랜 Life-time을 가진다. 제안하는 라우팅 알고리즘의 경우 전체 노드들이 공평하게 그리고 균등하게 에너지를 소비하기 때문에 거의 동시에 노드들이 죽는 것 을 볼 수 있다. Set-up 단계의 에너지 소모가 없는 이유로 각각의 노드들이 더 많은 에너지를 Steady 단계에서 소비 할 수 있다.

그림 7을 보면 LEACH의 경우 대략 10000 라운드까지는 200개의 노드 모두가 살아 있지만 그 이후부터는 노드의 수가 점점 줄어들고 있는 결과를 볼 수 있다. 그림 7의 그래프를 보면 제안하는 라우팅 알고리즘의 네트워크 전체 Life-time이 LEACH 보다 약 2배 이상 유지되고 있는 것을 알 수 있다.

그림 8은 한 노드가 평균 몇 번의 라운드에 참여하는지를 보여 주는 그래프 이다. 그림 8을 보면 LEACH의 경우 한 노드 당 57.5번의 라운드에 참여 하지만 본 논문에서 제안한 라우팅 알고리즘은 160.5번의 라운드에 참여한다. 이는 한 노드가 참여하는 라운드의 횟수에 있어 제안된 알고리즘이 LEACH 보다 대략 2.8배 이상 수명이 증가함을 의미한다. 본 그래프를 통해 제안한 라우팅 알고리즘이 LEACH 보다 에너지 소모가 더 균등하고 효율적으로 사용함을 알 수 있다.

마지막으로 하나의 클러스터가 매 라운드 때 마다 소비 되는 클러스터 헤더와 클러스터 멤버 전체의 에너지를 표 4에 나타내었다. 본 논문에서 제안하는 라우팅 알고리즘이 LEACH 보다 매 라운드 약 1/3 정도의 에너지만 소비됨을 알 수 있다. 제안하는 라우팅 알고리즘의 클러스터 헤더 경우 LEACH의 클러스터 헤더에 비해 약 1/4 정도의 에너지만 소비하고, 클러스터 멤버 경우 약 1/3 정도의 에너지만 소비하는 것을 보여 주고 있다. 전체적으로는 제안하는 알고리즘이 LEACH 보다 약 1/4 정도의 에너지만 소비해 LEACH 보다 향상된 성능을 보여 주고 있다.

기존의 클러스터링 기법이나 클러스터 헤더 재

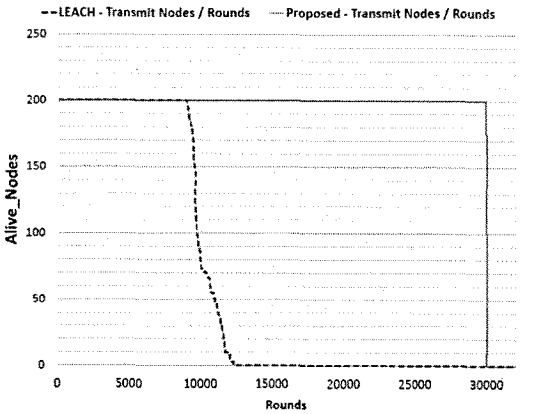


그림 7. Transmit Nodes / Round 결과 그래프

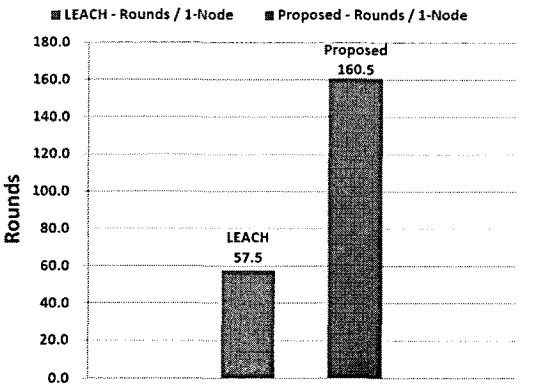


그림 8. Rounds / 1-Node 결과 그래프

표 4. 하나의 클러스터 에너지 소비량

소비량	LEACH (A)	제안된 알고리즘(B)	비교 ((A/B)x100)
CH-node	5.20×10^{-3}J	1.16×10^{-3}J	22.3 %
NCH-nodes	1.46×10^{-4}J	0.53×10^{-4}J	36.3 %
Total	5.34×10^{-3}J	1.21×10^{-3}J	22.7 %

선출 기법들은 네트워크 전체의 에너지 소비를 줄임으로 해서 네트워크 전체의 Life-time을 연장하였으나, 각각의 클러스터에서 소비되는 에너지의 양이 불균등한 결과를 보여 왔다. 본 논문에서 제안하는 라우팅 알고리즘은 네트워크 전체의 에너지 소비를 줄임과 동시에 각각의 클러스터에서 소비되는 에너지도 균등하게 함으로써 기존의 기법들보다 더 우수한 에너지 효율성을 보였다.

V. 결 론

본 논문에서는 한정된 자원을 가진 센서노드의 제약사항을 고려하여 에너지를 보다 효율적이면서 균등하게 사용함으로써 네트워크 전체의 생존시간을 늘릴 수 있는 기법을 제안하였다. 각각의 클러스터 헤더에 동일한 개수의 클러스터 멤버로 클러스터를 형성함으로써 센서 네트워크 전체의 에너지가 균등하고 효율적으로 소비된다. 또한, 최초 클러스터링 형성시, 각각의 클러스터링에 참여 하였던 클러스터 멤버들이 정해진 순서대로 각각의 클러스터의 헤더로 선출됨으로써 클러스터 헤더 재 선출로 소모되는 에너지를 최소화 하여 네트워크 전체 노드들의 생존시간을 향상시켰다. 제안한 라우팅 알고리즘의 성능을 시뮬레이션을 통하여 LEACH와 비교 분석하였다. 시뮬레이션 결과 본 논문에서 제안한 라우팅 알고리즘이 기존의 LEACH 에 비해 1/4 정도의 에너지를 소모하였고, 노드들의 Life-time 역시 2배 이상 향상되었다.

참 고 문 헌

[1] W. B. Heinzelman, J. Kulik and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proc. of the 5th ACM/IEEE MCN*, pp.174-185, August, 1999.

[2] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. of the Hawaii International Conference on System Sciences*, pp.1-10, January, 2000.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, 40(8), pp.102-114, August, 2002.

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks : A Survey," *IEEE Computer*, Vol.38, No.4, pp.393-422, March, 2002.

[5] S. C. Ergen, "IEEE 802.15.4 Summary," *Technical Report, Advanced Technology Lab of National Semiconductor*, August, 2004.

[6] 정훈, 이종오, 이종영, 박노성, 진광자, 김봉수, "센서 네트워크 기술 동향," *ETRI, 전자통신동향분석*, 제22권 제3호, pp.80-89, June, 2007.

[7] S. Bandyopadhyay and E. Coyle, "An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proc. of IEEE INFOCOM*, pp.1713-1723, April, 2003.

[8] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, 1(4) : 660-670, October, 2002.

[9] V. Raghunathan, C. Schurgers, S. Park and M. Srivastava, "Energy Aware Wireless Microsensor Networks," *IEEE Signal Processing Magazine*, 19(2) : 40-50, March, 2002.

[10] D. Nam and H. Min, "An Energy-Efficient Clustering Using a Round-Robin Method in a Wireless Sensor Network," *Proc. of the 5th SERA*, pp.54-60, October, 2007.

[11] Jinsuk Baek, Sun Kyong An, Paul S. Fisher, and Elva J. Jones, "Dynamic Cluster Header Selection with Self-incentive for Wireless Sensor Networks," *Proceedings of the 2009 IEEE Sarnoff Symposium (IEEE Sarnoff 2009)*, pp.1-5, Princeton, NJ, March-April, 2009.

[12] Al-Karaki, J. N. and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks : A Survey," *IEEE Wireless Communications*, Vol.11, No.6, December, 2004, pp.6-28.

[13] M. Ye, C. Li, G. Chen and J. Wu, "EECS : An Energy Efficient Clustering Scheme in Wireless Sensor Networks," *Proc. of IEEE IPCCC 2005*, pp.366-379, April, 2005.

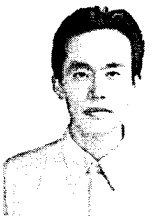
추 영 열 (Young-yeol Choo) 정회원

1985년 2월 서울대학교 제어계측 공학과 학사
1988년 8월 서울대학교 제어계측 공학과 석사
2002년 8월 포항공과대학 컴퓨터공학과 박사
1988년 6월~1994년 6월 포항산업과학기술연구원 선임연구원
1994년 7월~2002년 8월 posco 기술연구소 책임연구원
2002년 9월~현재 동명대학교 컴퓨터공학과 부교수
2005년 1월~2005년 6월 독일 Fraunhofer 연구소 (IESE) Visiting Scientist
<관심분야> 컴퓨터 통신, WSN, 공장자동화, 네트워크 보안

최 한 조 (Han-jo Choi) 정회원

2008년 2월 동명대학교 컴퓨터공학과 학사
2010년 2월 동명대학교 컴퓨터미디어학과 석사
2010년 3월~현재 M2M Korea 조명연구소 연구원
<관심분야> USN, 임베디드 시스템, LBS

권 장 우 (Jang-woo Kwon) 정회원



1990년 인하대학교 전자공학과 학사
1992년 인하대학교 전자공학과 석사
1996년 인하대학교 전자공학과 박사
1996년~1998년 특허청 심사관
1998년~현재 동명대학교 컴퓨터공학과 부교수
2006년~현재 정보통신산업진흥원 인력양성단장
2010년~현재 경원대학교 컴퓨터공학과 부교수
<관심분야> USN, U-health care, Biological Information system등