

메시지 수신 성공률을 이용한 클러스터 기반의 에너지 효율적인 라우팅 프로토콜

(Cluster-based Energy-Efficient
Routing Procotol using Message
Reception Success Rate)

장 유 진 [†] 최 영 호 ^{**}
(YouJin Jang) (YoungHo Choi)

장 재 우 ^{***}
(JaeWoo Jang)

요 약 기존 무선 센서 네트워크에서의 클러스터 기반 라우팅 기법은 첫째, 임의의 헤더 선출로 인하여, 일부 클러스터에 노드가 편중되는 문제점이 발생한다. 둘째, 실제 환경에서의 통신 범위를 고려하지 않기 때문에, 라우팅 경로의 신뢰도가 저하된다. 마지막으로, 헤더 선정을 위해 모든 센서 노드 정보를 전송하기 때문에, 데이터 전송 오버헤드가 증가한다. 이를 해결하기 위해 본 연구에서는 메시지 수신 성공률을 이용한 클러스터 기반 라우팅 프로토콜을 제안한다. 제안하는 기법은 첫째, 노드 편중도를 해결하기 위하여 노드의 밀집도 및 연결성을 이용하여 클러스터 헤더를 선정하고, 분할 및 병합을 수행한다. 둘째, 라우팅 경

로의 신뢰도 향상을 위하여, 실제 환경에 적용 가능한 메시지 수신 성공률을 기반으로 데이터 전송 경로를 설정한다. 마지막으로 데이터 전송 오버헤드의 감소를 위하여, 모든 센서 노드는 자신의 이웃 노드 정보만을 이용하여 헤더 선정 및 클러스터 구성 작업을 수행한다.

키워드 : 센서네트워크, 라우팅프로토콜, 메시지수신성공률

Abstract The existing cluster-based routing protocols have some problems. Firstly, because of selecting cluster head at random, they occur a node concentration problem. Secondly, they have a low reliability for data communication due to the less consideration of node communication range. Finally, data communication overhead is greatly increased because of sending all sensor node information to sink node for constructing clusters. To solve these problems, we in this paper, propose a cluster-based routing protocol using message reception success rate. Firstly, to solve the node concentration problem, we design a cluster head selection algorithm based on node connectivity and devise cluster splitting/merging algorithms. Secondly, to guarantee data communication reliability, we use message reception success rate. Finally, to reduce data communication overhead, we use only neighbor nodes information at both cluster construction and cluster head selection.

Key words : Sensor Network, Routing Protocol, Message Reception Success Rate

1. 서 론

최근 무선 센서 네트워크 통신 기술은 유비쿼터스 컴퓨팅 시대의 핵심 기반 기술로 각광받고 있다[1]. 무선 센서 노드는 한정된 에너지 자원을 지니기 때문에, 에너지 효율성을 향상시키는 것이 필수적이다. 센서 노드의 에너지 소비는 통신 에너지가 매우 큰 비중을 차지한다. 따라서 통신 에너지를 감소시키는 것이 가장 중요하다[2]. 이를 위해 메시지 통신량과 횟수를 감소시키기 위한 클러스터 기반 라우팅 기법이 제안되었다[3]. 그러나 기존의 클러스터 기반 라우팅 기법들은, 첫째, 클러스터 헤더를 임의로 선정함으로써 특정 클러스터에 센서 노드들이 편중되는 문제점이 존재한다. 둘째, 센서 노드 통신 범위를 고려하지 않기 때문에, 실제 환경에 적용할 경우 라우팅 경로의 신뢰도가 저하된다. 마지막으로, 클러스터 구성을 위해 모든 센서 노드의 정보를 수집하기 때문에, 데이터 전송 오버헤드가 급격히 증가한다.

본 논문에서는 기존 연구의 문제점을 해결하기 위하여, 메시지 수신 성공률을 이용한 에너지 효율적인 클러스터 기반 라우팅 프로토콜을 제안한다. 제안하는 기법은 첫째, 클러스터 노드 편중도 해결을 위해, 노드의 밀집도 및 연결성을 이용하여 클러스터 헤더를 선정하고,

- 본 연구는 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임
- 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. 2010-0000202)
- 이 논문은 2010 한국컴퓨터종합학술대회에서 '메시지 수신 성공률을 이용한 클러스터 기반의 에너지 효율적인 라우팅 프로토콜'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 텔리언 연구팀

yjiang@dblab.chonbuk.ac.kr

^{**} 학생회원 : 전북대학교 컴퓨터공학과

yhchoi@dblab.chonbuk.ac.kr

^{***} 종신회원 : 전북대학교 컴퓨터공학과 교수

hwchang@chonbuk.ac.kr

논문접수 : 2010년 8월 5일

심사완료 : 2010년 10월 13일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제16권 제12호(2010.12)

분할 및 병합을 수행하여 편중도를 해결한다. 둘째, 라우팅 경로의 신뢰도 향상을 위하여, 메시지 수신 성공률을 기반으로 경로를 설정한다. 마지막으로 데이터 전송 오버헤드의 감소를 위해, 이웃 노드 정보만을 이용하여, 헤더 선정 및 클러스터 구성 작업을 수행한다.

2. 관련 연구

무선 센서 네트워크 환경에서 대표적인 클러스터 기반 라우팅 기법으로는 LEACH[5], LEACH-C[6], Multihop-LEACH[7], CBRP[8]가 존재한다.

첫째, LEACH[5]는 임의로 클러스터 헤더를 선정하고 주기적으로 교체하여, 에너지 소비를 분산시켜 에너지를 균등하게 사용한다. 그러나 임의 선출 방식에 의해 클러스터 헤더를 선정하기 때문에, 효율적인 경로를 구성하지 못한다. 이러한 문제점을 해결하기 위해 둘째, LEACH-C[6]는 위치 정보를 이용하여 헤더를 선정한다. 이 기법은 모든 센서 노드가 자신의 위치 정보를 싱크 노드에 전송하여 클러스터 헤더를 선택한다. 하지만, GPS등의 위치 측정 장비가 필요하고, 메시지 전송 오버헤드가 크다. 또한 LEACH 및 LEACH-C는 헤더 노드와 싱크 노드간의 멀티 홉을 지원하지 않는다. 셋째, Multihop-LEACH[7]는 LEACH와 LEACH-C가 멀티 홉을 지원하지 않는 문제점을 해결한다. 이 기법은 LEACH의 알고리즘을 바탕으로 임의로 생성된 클러스터 헤더로 클러스터를 구성하고, 멀티 홉 구성을 지원한다. Multihop-LEACH는 멀티 홉 경로를 구성하여, 비용을 많이 소모하는 장거리 전송을 피하는 장점을 가진다. 반면, LEACH와 동일한 방법으로 클러스터 헤더를 임의로 선출하기 때문에, 헤더의 위치가 편중된다. 또한, 클러스터 내에서는 멀티 홉을 지원하지 않기 때문에, 장거리 전송이 존재한다. 마지막으로, CBRP[8]에서는 기존 연구의 한정된 통신범위를 고려하지 않는 문제점과 헤더의 임의 설정으로 인한 클러스터 편중 문제를 해결하기 위하여, 통신 범위를 고려한 홉 수 기반의 네트워크를 구성하고, 대표 경로를 이용한 분산된 헤더를 선정한다. 반면 클러스터 구성을 위해 모든 센서노드의 정보를 수집하기 때문에, 데이터 전송 오버헤드가 급격히 증가하는 문제점이 존재한다.

3. 메시지 수신 성공률을 이용한 클러스터 기반의 에너지 효율적인 라우팅 알고리즘

기존의 클러스터 기반 라우팅 기법인 LEACH, LEACH-C, Multihop-LEACH 및 CBRP의 연구는 네트워크 구성 시, 벽과 같은 장애물로 인해 메시지 전송 경로 구성이 원활하지 못한 문제점을 지닌다. 또한 기존 연구 중 LEACH와 Multihop-LEACH는 임의로 헤더를 선정하여 일부 클러스터에 센서 노드들이 편중되는 문

제점이 있으며, 센서 노드가 지나는 한정된 통신 범위를 고려하지 않기 때문에, 대규모 센서 네트워크에서 사용이 불가능하다. LEACH-C 및 CBRP의 연구는 모든 센서 노드의 정보를 싱크 노드에서 관리하기 때문에, 노드 정보 전송에 따른 커다란 메시지 전송량이 단점이다.

따라서 에너지 효율적인 라우팅 프로토콜을 설계할 시 고려사항은 다음과 같다. 첫째, 메시지 수신 성공률을 이용하여 연결 신뢰성이 높은 노드로 경로를 구성한다. 둘째, 각 센서 노드 이웃노드와의 메시지 수신 성공률에 따른 홉 수 기반으로 네트워크를 구성한다. 셋째, 각 센서 노드는 싱크 노드와 직접 통신할 수 없기 때문에 이웃 노드를 통해 통신 가능한 경로를 구성한다. 넷째, 클러스터 헤더 선정을 위한 정보를 인접한 이웃 노드 사이의 통신을 통해 헤더를 선정한다.

제안하는 라우팅 알고리즘은 크게 네트워크 정보 생성 단계, 클러스터 헤더 선정 단계, 클러스터 구성단계, 클러스터 관리의 4단계로 구성된다.

첫째, 네트워크 정보 생성 단계에서는 메시지 수신 성공률을 이용하여 초기 네트워크 정보를 생성한다. 각 센서 노드는 자신의 이웃 노드를 탐색하고, 메시지를 전송하여 수신 성공률을 측정한다. 메시지 수신 성공률이 높은 노드를 통해 초기 경로를 구성하고 자신의 네트워크 정보를 생성한다. 센서 노드의 네트워크 정보는 각 노드에서 인접한 이웃 노드간의 통신으로 생성된다. 자신과 싱크 노드까지의 홉 차이를 홉 수로 저장하고, 이웃 노드들 중 싱크 노드 방향에 존재하고, 메시지 수신 성공률이 가장 높은 노드를 부모 노드로 선정한다. 또한, 자신을 부모 노드로 선정하는 노드를 자식 노드로 저장한다. 각 노드에 생성되는 정보는 표 1과 같다.

둘째, 클러스터 헤더 선정 단계에서는 센서 노드의 밀도가 높고, 연결 신뢰성이 높은 노드를 클러스터 헤더로 선정한다. LEACH의 실험[5]에 따르면 최적의 클러스터 헤더 수는 전체 센서 노드의 5%이다. 본 연구에서는 최적의 클러스터 헤더 수를 5%, 클러스터 멤버 수를 전체 노드의 20%를 목표로 한다. 최적의 클러스터 제한 멤버 수를 위한 실험은 4장에서 제시한다.

표 1 센서 노드의 네트워크 정보

NodeID	센서 노드의 아이디
HopCount	싱크 노드와의 홉 차이
NeighborID	센서 노드의 통신 범위 안에 포함되는 노드들의 아이디
InitialNeighborCount	초기 이웃 노드 수
NeighborMegSuccRate	이웃 노드의 메시지 수신 성공률
InitialChildnodeCount	초기 자식 노드 수
InitialChildnodeID	초기 자식 노드 아이디
InitiaParentnodeID	초기 부모 노드 아이디

클러스터 헤더는 멤버 노드들의 데이터를 집계하여 싱크 노드로 전송한다. 클러스터 내의 모든 노드들이 헤더를 통해 데이터를 전송하기 때문에, 에너지 소모를 줄이는 효율적인 헤더 선정이 필수적이다. 클러스터 헤더 선정을 위한 고려사항은 다음과 같다. 첫째, 이웃 노드 수가 많아 노드의 밀집도가 높아야한다. 둘째, 메시지 수신 성공률이 높아 연결 신뢰성이 좋아야한다. 마지막으로, 최하위나 최상층에 헤더가 편중되지 않고, 네트워크의 중심에 위치하기 위해 홉 수를 고려해야한다. 클러스터 헤더 선정을 위한 가중치는 정의 1과 같다. α , β , γ 는 클러스터 헤더 선정을 위한 임계값이며 합은 1이다. 최적의 임계값에 대한 실험은 4장에서 제시한다.

정의 1. 클러스터 헤더 선정을 위한 가중치

$$Weight = \alpha * InitialNeighborCount + \beta * \sum (NeighborMsgSuccRate) + \gamma * HopCount$$

그림 1은 클러스터 헤더 선정 알고리즘을 나타낸다. 클러스터 헤더 선정은 노드가 자식 노드가 없는 경우(2-7번째 줄)와 존재하는 경우(8-20번째 줄)로 구성된다. 자식 노드가 없는 경우는 최하위 노드로 설정하고, 하위 노드를 카운트한다(3번째 줄). 또한 가중치는 0으로 설정하고(4번째 줄), 부모 노드에게 자신의 ID, 가중치, 카운트와 홉 수를 전송한다(5-7번째 줄). 자식 노드가 존재하는 경우는 자식 노드의 정보를 전송받아(9번째 줄), 노드 카운트와 가중치를 계산한다(10-11번째 줄). 계산된 가중치를 자식 노드의 가중치와 비교하여 가장 높은 값의 노드 가중치 아이디를 저장한다(12-15번째 줄). 노드 카운트가 클러스터 제한 멤버수를 만족하지 않으면 부모 노드로 메시지를 전송하고(16-19번째 줄), 만족하면 가중치 값이 가장 높은 노드의 아이디를 클러스터 헤더로 설정한다(20번째 줄).

```

1. SelectClusterHead(){
2.   if (InitialChildNodeCount == 0){
3.     nodeCnt=1;
4.     weight=0;
5.     child_weight=weight;
6.     ChildnodeID=NodeID;
7.     sendMsg(ChildnodeID,child_weight,nodeCnt, HopCount);
8.   }else{
9.     receiveMsg(NodeID, child_weight, nodeCnt, HopCount);
10.    nodeCnt++;
11.    weight=calc_weight(InitialNeighborCount,
NeighborMsgSuccRate, HopCount);
12.    max_weight=compare(weight, child_weight);
13.    if(max_weight == weight){
14.      candi_ch=NodeID;
15.    }else{ candi_ch=ChildnodeID; }
16.    if(ClusterMemCnt > nodeCnt){
17.      ChildNodeID=NodeID;
18.      child_weight=max_weight;
19.      sendMsg(candi_ch,ChildNodeID,max_weight,
nodeCnt,HopCount);
20.    }else{ ClusterHead=candi_ch; } }
21. }

```

그림 1 클러스터 헤더 선정 알고리즘

셋째, 클러스터 구성 단계에서는 초기 클러스터를 구성하고, 최적의 클러스터를 구성하기 위해 생성된 클러스터의 분할 및 병합을 수행한다. 클러스터 구성 알고리즘은 크게 초기 클러스터 구성, 노드 편중 클러스터 분할, 그리고 클러스터 병합의 세 단계로 구성된다. 먼저 초기 클러스터 구성에서는 클러스터 헤더 선정 단계에서 선정된 클러스터 헤더를 대표로 하는 클러스터를 형성한다. 자식 노드가 존재하지 않는 라우팅 경로의 최하위 단계에서부터 탐색을 시작하여 클러스터 제한 멤버수를 만족할 때까지 상위 노드로 확장하여 클러스터를 구성한다. 초기 클러스터 구성 후, 최적 클러스터 멤버수를 만족시키지 못하는 잔여 노드가 발생할 수 있다. 이러한 잔여 노드들은 인접 클러스터를 선택한다. 이때 인접 클러스터를 선정하기 위해, 클러스터 헤더와의 홉 수 차이 및 클러스터 내 멤버 수를 고려한다. 한편, 노드 편중 클러스터 분할 단계에서는, 클러스터 내의 멤버 수가 임계값 δ 이상일 경우, 클러스터를 분할한다. 클러스터 분할시, 클러스터 헤더와의 홉 차이 및 라우팅 경로 유사도를 이용하여 새로운 클러스터 헤더를 선정한다. 정의 2는 클러스터 분할 시, 클러스터 헤더 선정을 위한 가중치를 나타낸다. α_1 , β_1 , γ_1 는 새로운 클러스터 헤더 선정을 위한 임계값이고, α_1 , β_1 , γ_1 의 합은 1이다. HopDiff는 기존 헤더 노드와의 홉 수 차이를 나타낸다.

정의 2. 분할을 위한 클러스터 헤더 선정 가중치

$$Weight = \alpha_1 * InitialNeighborCount + \beta_1 * \sum (NeighborMsgSuccRate) + \gamma_1 * HopDiff$$

마지막으로 클러스터 병합 단계에서는 클러스터 멤버수가 ε 이하일 경우, 클러스터를 인접한 클러스터에 병합한다. 이때 각 클러스터 헤더들 사이의 홉 차이 및 라우팅 경로 유사도를 이용하여 병합 대상 클러스터 후보 집합을 선정한다. 아울러 멤버 노드 수를 비교하여, 최적의 병합 대상 클러스터와 병합한다. 정의 3은 병합하기 위한 후보 클러스터를 선택하기 위한 가중치를 나타낸다. α_1 , β_1 는 병합 후 클러스터 헤더 선정을 위한 임계값이고, α_1 , β_1 의 합은 1이다. HopDiffbetweenCH는 클러스터 헤더 간의 홉 수 차이이고, Similaritypath는 라우팅 경로 유사도를 나타낸다.

정의 3. 병합을 위한 후보 클러스터 선정 가중치

$$Weight = \alpha_2 * HopDiff_{betweenCH} + \beta_2 * Similarity_{path}$$

한편, 기존 연구의 노드 편중도 문제점을 제안하는 클러스터 분할 및 병합 기법을 이용하여 해결한다. 그림 2는 클러스터 구성 알고리즘이다. 선정된 클러스터 헤더를 이용하여 초기 클러스터를 구성하고(2번째 줄), 초기 클러스터에 포함되지 않는 잔여 노드는 클러스터 헤더와의 홉 차이 및 클러스터 내 멤버 수를 비교하여 인접

```

1. ConstructCluster()
2. IsCluster=MakeCluster(ClusterHead);
3. if(IsCluster == False){
4.     ClusterHead = compare (unclusteredNode, ClusterHead);
5.     IsCluster = True; }
6. if(ClusterMemCnt >= 6){
7.     DivideCluster(); }
8. if(ClusterMemCnt <= e){
9.     MergeCluster(); }
10. }
    
```

그림 2 클러스터 구성 알고리즘

한 헤더로 포함된다(3-6번째 줄). 클러스터 멤버 수가 분할 임계값인 δ 이상일 경우, DivideCluster 함수를 통해 클러스터를 분할한다(7-9번째 줄). 그리고 클러스터 멤버 수가 병합 임계값인 ϵ 이하인 경우, MergeCluster 함수를 통해 클러스터를 병합한다(8-9번째 줄).

넷째, 클러스터 관리 단계에서는 주기적인 헤더 교체 및 클러스터 재구성을 수행한다. 클러스터 헤더는 다른 노드에 비해 에너지 소모가 많다. 따라서 이를 주기적으로 교체함으로써 네트워크 수명을 연장시킨다. 이를 위해 최적의 클러스터를 생성하여 멤버 수를 균일하게 분포하고, 이웃 노드 연결성과 홉 수를 이용하여 연결성이 높은 헤더를 선출한다. 또한, 교체된 헤더 노드의 에너지 잔여량이 임계값 이하일 경우, 클러스터를 재구성한다. 클러스터 재구성시 대표경로 생성은 에너지 잔여량을 고려하고, 에너지 소모가 적은 센서 노드를 클러스터 헤더로 선정함으로써 에너지 효율성을 높인다. 클러스터 재구성시, 클러스터 멤버수를 만족하지 못하는 클러스터에 대한 분할 및 병합을 수행하여 재분배를 수행한다. 그림 3은 클러스터 관리 단계 알고리즘이다. Header Update Interval로 설정된 임의의 헤더 변경 주기를 이용하여 현재 노드의 라운드를 체크하고, 변경 주기가 되면 클러스터 헤더를 교체한다(2-4번째 줄). 노드의 에너지가 일정 임계값 이하로 내려가면, 라우팅 경로 재구성 메시지를 전송한다(6-7번째 줄).

```

1. ManageCluster()
2. RoundCheck = CurrRound % / Header Update Interval;
3. if(RoundCheck = 0){
4.     ClusterHead=ChangeHeader(); }
5. CH_EnergyCnt=0;
6. if (Energy of Header <= Energy of Threshold){
7.     SendMsg(reconstruction); }
8. }
    
```

그림 3 클러스터 관리 알고리즘

4. 성능 평가

제안하는 라우팅 프로토콜의 헤더 선정에 사용되는 가중치 최적 임계값 및 초기 클러스터 구성 시에 사용되는 최적의 클러스터 멤버 비율을 선정하기 위해 실험

을 수행한다. 또한, 클러스터 구성 시, 클러스터 분할 및 병합 기법에 대한 균등한 분포성을 증명하기 위해, LEACH와 클러스터 멤버 수의 표준편차에 대한 성능평가를 수행한다. 정방형의 네트워크 필드 100*100m에서, 센서 노드의 통신 범위는 MicaZ[9]의 평균 통신 거리인 25m로 설정하고, 총 노드 수 100개에 대해 실험하였다.

클러스터 헤더 선정시 고려되는 요소는 이웃노드 수, 이웃 노드의 메시지 수신 성공률, 노드의 홉 수가 존재한다. 각각의 가중치를 α , β , γ 라 하고, 합은 1이다. 각 클러스터에 포함된 노드 멤버수가 균일할수록 세 가중치의 최적값이다. 이를 측정하기 위해 각 클러스터 노드 멤버수를 표준편차로 나타내며, 표준편차가 작을수록 클러스터가 균일하게 분포됨을 의미한다. 이를 위해, 세 가중치의 대표적 조합을 36개 선정하여 실험하였다. 그림 4는 각 조합에 대한 클러스터 멤버 수 표준편차를 나타낸다. x축의 세자리 수는 α , β , γ 값을 의미하며, 예를 들어 118은 α 의 값이 0.1, β 의 값이 0.1, γ 의 값이 0.8임을 의미한다. 실험 결과 가장 작은 표준편차는 3.6이며, 이때의 최적 조합은 α 는 0.3, β 는 0.1, γ 는 0.6이다. 실험 결과 홉 수 가중치, 이웃노드수 가중치, 메시지 수신성공률 가중치 순으로 중요도가 측정되었다. 첫째, 홉 수 가중치가 높은 이유는, 헤더 노드의 위치를 적절하게 선정함으로써 센서 네트워크 필드를 적절하게 분할할 수 있기 때문이다. 둘째, 이웃 노드 수의 경우, 특정 지역에 발생한 노드 편중도를 이웃 노드 수를 고려함으로써 해결하기 때문이다. 마지막으로 메시지 수신 성공률의 경우, 실제 환경에서는 장애물에 의해 큰 영향을 받지만, 실험 환경에서는 거리에 의해서만 영향을 받기 때문에 큰 비중을 차지하지 못했다. 본 성능평가는 센서네트워크 환경에서 가장 많이 사용되는 배치 형태인 임의 배치(random) 유형을 사용하였다. 한편 특정한 환경의 센서 노드 배치인, 비대칭 배치 또는 격자 배치 등의 특정 센서 네트워크 환경을 고려한 최적의 가중치 조합을 찾기 위한 실험이 요구된다.

그림 5는 초기 클러스터 구성 시에 사용되는 최적의 클러스터 멤버 비율에 대한 표준편차이다. 초기 클러스터 제한 멤버 비율은 20%이고, 클러스터 헤더 비율은

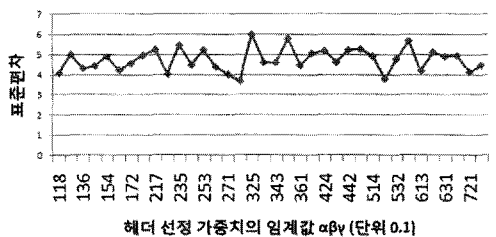


그림 4 헤더 선정 가중치 최적 임계값

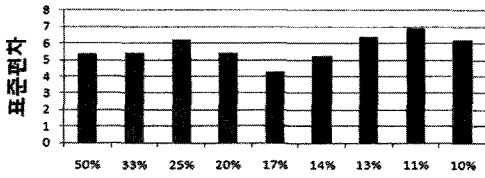


그림 5 최적 클러스터 멤버 비율

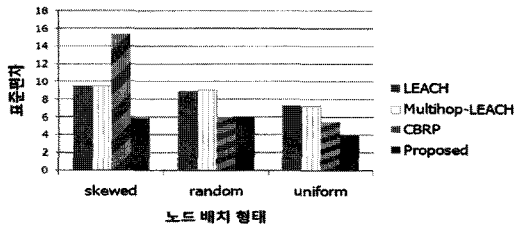


그림 6 클러스터 멤버수 표준편차

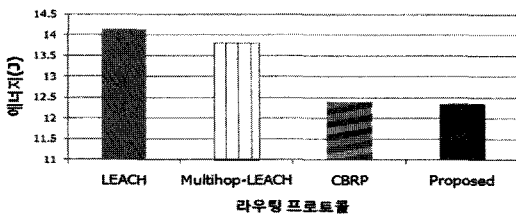


그림 7 센서 노드의 에너지 소모량

2%에서 10%까지 증가시켰다. 클러스터 멤버 비율이 17%일 경우, 표준편차가 4.2로 가장 작은 값을 나타낸다. 이때 클러스터 멤버 수가 가장 균일함을 나타낸다.

그림 6은 클러스터 멤버의 표준편차를 나타낸다. 센서 노드의 배치는 임의 배치(random), 비대칭 배치(skewed), 격자 배치(uniform)의 유형을 사용하였다. LEACH는 8.5, Multihop-LEACH는 8.6 CBRP는 8.9를 나타내는 반면, 제안한 기법의 경우 5.3으로 기존 기법에 비해 작은 표준편차를 보인다. 이는 제안한 기법이 기존기법보다 40%의 효율성을 보인다.

그림 7은 노드 에너지 소모량을 나타낸다. LEACH, Multihop-LEACH, CBRP 및 제안한 기법의 에너지 소모량은 각각 14.1J, 13.8J, 12.4J, 12.2J이다. CBRP 및 제안한 기법의 경우, LEACH에 비해 약 12%, Multihop-LEACH에 비해 약 11% 향상된 성능을 나타낸다. 이는 두 기법이 센서 노드의 지역적 유사성 혹은 연결성을 기반으로 헤더를 선정하기 때문에, 노드 간 통신 거리가 다른 기법들에 비해 감소하기 때문이다.

5. 결론 및 향후 연구

본 논문에서는 센서 노드의 메시지 수신 성공률을 이용한 클러스터 기반의 에너지 효율적인 라우팅 기법을 제안

하였다. 제안하는 기법은 클러스터에 센서 노드들이 밀집되는 문제를 해결하기 위해, 클러스터의 분할 및 병합을 통해 최적화를 수행한다. 또한, 노드 정보 전송에 사용되는 불필요한 전송량을 감소시키기 위해, 인접한 이웃 노드 사이의 통신을 통해 클러스터 헤더를 선정한다. 아울러, 장애물로 인한 전송 경로 구성이 원활하지 못한 문제점을 해결하기 위해, 메시지 수신 성공률을 이용하여 네트워크 신뢰성을 향상시킨다. 성능평가를 통해, 클러스터 헤더 선정에 사용되는 최적의 가중치의 임계값과 초기 클러스터 구성 시의 최적의 클러스터 멤버 비율을 도출하였다. 아울러 LEACH와의 성능평가를 통해 제안하는 기법의 클러스터가 보다 균일한 분포를 지니는 보였다.

향후 연구는 실제 무선 센서 네트워크 플랫폼에 적용하여, 응용 서비스 지원시의 신뢰성을 검증하는 것이다. 아울러 비대칭 배치 또는 격자 배치에 대한 최적의 가중치 조합을 찾기 위한 실험을 수행하는 것이다.

참고 문헌

- [1] 김봉수, 황소영, 정훈, 주성순, "유비쿼터스 센서 네트워크 라우팅 기술", *인터넷 정보학회지*, vol.9, no.1, pp.13-21, 2008.
- [2] K. Akkaya, M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol.3, no.5, pp.352-349, 2005.
- [3] A. A. Abbasi, M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communication*, vol.30, no.14-15, pp.2826-2841, 2007.
- [4] Koungh-Jin Choi et al., "ECS : Energy efficient Cluster-head Selection algorithm in Wireless Sensor Network," *Korea Information and Communications Society*, vol.32, no.6, pp.342-349, 2007.
- [5] A. Woo, D. E. Culler, "A transmission control scheme for media access in sensor networks," *The ACM/IEEE International Conference on Mobile Computing and Networking*, pp.221-235, 2001.
- [6] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *The IEEE International Conference on Transactions on Wireless Communications*, vol.1, no.4, pp.660-670, 2002.
- [7] F. Xiangning, S. Yulin, "Improvement on LEACH protocol of wireless sensor network," *International Conference on Sensor Technologies and Applications, IEEE*, pp.260-264, 2007.
- [8] You-Jin Jang et al., "Design and Implementation of Cluster based Routing Protocol using Representative Path in Ubiquitous Sensor Network," *The 32th KIPS Fall Conference 2009*, vol.16, no.2, pp.75-76, 2009.
- [9] MICAz data sheets, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf, Crossbow Technology, 2006.