

에너지 및 메모리 효율성을 개선한 비트맵기반 라우팅 프로토콜

최 해 원* · 김 상 진** · 류 명 춘***

Bitmap-based Routing Protocol for Improving Energy and Memory Efficiency

Choi, Hae Won · Ryoo, Myung Chun · Kim, Sang Jin

〈Abstract〉

This paper proposes a improved bitmap routing protocol, which finds the best energy efficient routing path by minimizing the network overheads and prolongs the overall network lifetime. Jung proposed a bitmap scheme for sensor networks. His scheme uses a bitmap table to represent the connection information between nodes. However, it has a problem that the table size is depends on the number of nodes in the sensor networks. The problem is very serious in the sensor node with a limited memory. Thereby, this paper proposes a improved bitmap routing protocol to solve the problem in Jung's scheme. Proposed protocol over the memory restricted sensor network could optimize the size of bitmap table by applying the deployed network property. Proposed protocol could be used in the diversity of sensor networks due to it has minimum memory overheads.

Key Words : Sensor Network, Energy Efficiency, Memory Efficiency, Routing Protocol, LEACH

I. 서론

센서 네트워크는 재난탐지, 군사, 홈 네트워크, 헬스케어 등의 다양한 응용에 사용되어 우리 생활과 밀접한 관계를 가지며 유비쿼터스 인프라 구축에 있어서 빠질 수 없는 중요한 부분으로 인식되고 있다[1-5].

지금까지 센서네트워크에 적용될 무선 애드혹(Ad-hoc) 라우팅 기법은 크게 평면 라우팅(Flat Routing)과 계층적 라우팅(Hierarchical Routing) 프로토콜로 분

류된다[6-8]. 평면 라우팅은 네트워크 전체를 하나의 영역으로 간주하여 모든 노드들이 동등하게 라우팅에 참여할 수 있고 멀티 홉 라우팅을 특징으로 한다. 계층적 라우팅은 네트워크를 클러스터링을 기반으로 한 다수의 영역으로 분할하여 각각의 영역 내 특정 노드에 헤드의 역할을 부여하여 라우팅을 수행하도록 한다.

평면 라우팅프로토콜의 대표적인 기법으로 DD(Directed Diffusion)를 들 수 있다[9, 10]. DD는 질의(Interest)에 기반을 둔 데이터 중심 라우팅 기법이다. DD에서 질의를 처리하기 위해서 싱크는 네트워크 전체에 질의를 플러딩한다. 이때 데이터는 다중경로를 통해 각 노드들에게 전송된다. 더 이상의 플러딩을 막기 위해

* 경운대학교 컴퓨터공학과 전임강사

** 경운대학교 컴퓨터공학과 조교수

*** 경운대학교 컴퓨터공학과 부교수(교신저자)

경로들 중 전송 품질이 좋은 몇 개의 경로들이 선택되어, 선택된 경로를 통해 데이터가 전송된다. 하지만 이 프로토콜은 경로를 단지 한번만 사용하는 형태의 응용에는 적합하지 않다.

계층형 라우팅 프로토콜은 로컬 클러스터를 형성함으로써 인접한 지역에서 발생한 사건에 대한 유사한 정보를 클러스터 헤드로 전송하고, 클러스터 헤드는 데이터 통합(Aggregation)을 수행한다. 이렇게 함으로써 평면 라우팅 보다 에너지 효율적인 라우팅을 가능하게 하고, 요청 질의를 클러스터 헤드에 의해 클러스터 내의 각 노드에게 전달함으로써 비효율적인 질의의 플러딩을 막을 수 있다. LEACH는 계층형 라우팅 프로토콜의 대표적인 기법이다[11]. LEACH는 하나의 시간단위인 라운드(Round)를 기반으로 동작한다. 각 라운드는 크게 클러스터 형성을 위한 설정 단계(Setup)와 여러 개의 TDMA(Time Division Multiple Access) 프레임으로 구성되는 지속상태(Steady)단계로 구성된다. 이 기법의 특징은 네트워크에 있는 모든 센서 노드들에 에너지 소비를 공정하게 분산시키기 위해, 에너지 집약적인 역할을 수행하는 클러스터 헤드를 무작위로 순환 시키고, 전체적인 통신비용을 줄이기 위해 클러스터 헤드가 클러스터 내의 데이터를 통합한다. 하지만 이 기법에는 클러스터 헤드의 배치분포가 고르게 배치되는 것을 보장할 수 없다는 단점이 있다.

그러나 비록 계층적 라우팅 프로토콜에서 클러스터 헤드를 이용함으로써 평면 라우팅에 존재하는 비효율적인 플러딩을 해결하였다고 하지만, 이들 라우팅 프로토콜들은 여전히 라우팅을 형성 하는데 있어서 많은 메시지 요구량을 필요로 한다. 하지만, 센서 네트워크에서 대부분의 에너지 소모는 메시지의 송수신에서 발생한다는 점을 고려할 때 이들 프로토콜의 문제점을 해결하기 위한 다양한 연구는 효율적인 유비쿼터스 컴퓨팅 환경을 위한 선행연구로서 꼭 필요하다.

이러한 문제를 해결하기 위해서 최근에 에너지 효율성을 제시하기 위한 다양한 연구가 제시되었다[12-14].

이들 중 비트맵 테이블(Bitmap Table)을 이용한 기법(Scheme)은 노드간의 연결 정보를 1비트로 표현한다[14]. 비트맵 테이블 기법에서 요구 메시지를 처리한 각 노드는 동일한 메시지의 요청에 대해서는 비트맵 테이블에 대한 갱신만 수행하도록 함으로써 라우팅 시 전체 메시지의 플러딩 수를 제한 할 수 있었다. 즉, 각 노드들은 단 한 번의 플러딩으로 라우팅을 형성함으로써 기존의 클러스터링 기법보다 메시지 수를 줄일 수 있었다. 하지만, 이 기법은 비트맵 테이블을 사용함으로써 발생하는 센서 네트워크의 메모리 제약에 대한 문제를 완전히 해결하지 못하고 있다. 즉 네트워크를 구성하는 노드수가 일정 수 이상이 되는 경우에는 적용이 불가능하다.

본 논문에서는 이러한 문제점을 해결하기 위한 개선된 비트맵 기반 라우팅 프로토콜을 제안한다. 제안하는 라우팅 프로토콜은 기존의 비트맵 방식의 장점은 유지하면서 문제점을 효율적으로 해결할 수 있다. 특히, 메모리 제약을 가진 센서네트워크에서 비트맵 테이블의 크기를 네트워크의 크기와는 관계없이 적용 가능하도록 함으로써 네트워크 전체 노드 수에 상관없이 효율적인 네트워크의 확장성을 제시할 수 있다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 기존 연구로서 비트맵 기법에 대한 간략화 된 설명을 제시하고, 문제점을 분석한다. 3장에서는 2장에서 제시한 기존 기법의 문제점을 해결하기 위한 새로운 개선된 비트맵 라우팅 프로토콜을 제안한다. 그리고 4장에서는 본 논문에서 제안된 라우팅 프로토콜과 관련된 프로토콜간의 비교와 분석을 제시한다. 마지막으로 5장에서는 결론을 제시한다.

II. 관련연구

본 장에서는 본 논문의 관련연구로서 기존의 비트맵 테이블 기법에 대해서 요약하여 살펴보고, 문제점을 분석한다[14].

2.1 비트맵 테이블 기법

비트맵 테이블 기법은 노드의 한정된 용량을 고려하여 라우팅 테이블을 비트맵으로 표현하고, 전체 메시지의 수를 줄이기 위해 싱크노드에서 감지작업(Sensing Tasks)에 대한 질의(Interest)를 발송할 때, 비트맵 테이블을 단 한번만 전송하게 함으로서 경로 설정 시 전체 메시지의 수를 줄이는 방법을 이용한다. 본 절에서는 이러한 기법을 설명하기 위해서 먼저 비트맵 테이블의 구성에 대해서 살펴보고, 이를 이용한 전송경로 설정 과정에 대해서 다룬다. 마지막으로 비트맵 테이블의 크기에 대한 분석을 제시한다.

2.1.1 비트맵 테이블 구성

비트맵 테이블을 구성하기 위해 각각의 센서 노드는 유일한 식별자를 가진다. 비트맵 테이블은 센서 노드들의 식별자를 이용하여 노드간의 연결정보를 1비트로 그림 1과 같이 표현한다.

$$\text{map}[i][j] = \begin{pmatrix} 7 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 4 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 5 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 6 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 7 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

<그림 1> 비트맵 테이블 구성

그림 1은 임의의 노드 7번에 대한 비트맵 테이블의 예를 보여준다. 그림 1에서 $\text{map}[0][0]$ 은 노드 자신의 식별자 정보(ID)를 제시하고, i 와 j 의 교차점은 노드들 간의 직접적인 링크의 존재여부에 따라 존재하면 1, 존재하지 않으면 0으로 설정된다. 싱크노드로부터 질의가 풀러딩 될 때 이 비트맵 테이블도 함께 전송된다.

2.1.2 전송경로 설정

싱크가 특정한 질의에 대한 요구 메시지를 풀러딩한

다. 질의를 받은 노드는 자신이 요구를 수행할 수 있는지 확인한 후, 처리할 수 없는 작업인 경우 본 소절에서 제시하는 전송경로설정 과정을 실행 한 후, 이웃노드에게 질의와 비트맵 테이블을 포함하는 메시지를 풀러딩한다. 이러한 작업은 질의를 처리할 수 있는 노드를 발견 할 때까지 계속 수행한다. 기 처리된 요구 메시지를 받은 노드는 이 요구 메시지에 대한 풀러딩을 수행하지 않고, 단지 자신의 비트맵 테이블의 정보만 갱신한다. 비트맵 테이블 기법에서 전송경로 설정을 위해 다음의 4단계를 수행한다.

단계 1 : 요구 메시지를 수신한 노드는 자신이 질의에 대한 처리를 수행할 수 있는지의 여부를 판단한다.

단계 2 : 질의를 처리할 수 없는 노드는 다음과 같이 비트맵 테이블을 갱신한다.

if(수신된 노드의 식별자가 비트맵 내에 존재 ?)

{수신된 테이블과 자신의 테이블을 식별자가 동일한 열끼리 OR 연산 수행}

else

{테이블의 i 번째 행, 열 위치에 식별자 삽입; 자신의 식별자 행과 i 번째 열, 자신의 식별자 열과 i 번째 행을 1로 표시}

end if

단계 3 : 정보의 갱신이 완료 된 후, 노드는 질의 요청과 비트맵 정보를 포함하는 메시지를 단 한번 풀러딩한다.

단계 4 : 풀러딩을 수행한 후, 기 처리된 요구 메시지를 받으면, 노드는 단계 2에서와 같이 단지 자신의 비트맵 테이블 정보만 갱신한다.

이러한 전송경로 설정과정이 끝나면 자신의 노드로부터 싱크노드까지의 경로 정보를 포함함 비트맵 테이블을 유지하게 된다.

2.1.3 비트맵 테이블 기법의 문제점

그림 1의 비트맵 테이블에서 $map[0][0]$ 은 노드 자신의 식별자 정보를 저장할 수 있어야 하므로 노드의 개수가 2^8 이하의 개수를 고려할 경우 1바이트로 설정될 수 있다. 즉, 하나의 비트맵 테이블에 있어서 0열은 모든 노드들의 식별자를 저장할 수 있어야 하므로 최악의 경우 각 노드는 $2^8 \times 2^8$ 비트(28바이트)의 크기가 식별자를 위해서 필요하다. 이때 연결정보를 위한 테이블의 크기는 $2^8 \times 2^8$ 비트가 되므로, 전체 비트맵 테이블의 크기는 $2 \times (2^8 \times 2^8)$ 비트(65,536바이트, 64KB)가 된다.

비트맵 테이블의 크기를 분석한 결과, 기존의 비트맵 테이블 기법은 센서네트워크를 구성하는 노드의 개수에 따라 메모리의 요구사항이 변함을 확인할 수 있다. 즉, 센서네트워크를 구성하는 노드의 크기가 작은 네트워크는 비트맵 테이블 기법을 이용해서 라우팅이 가능하지만, 네트워크를 구성하는 전체 노드 수가 각 노드의 메모리 요구량을 초과할 경우에는 비트맵 테이블 기법을 사용할 수 없다는 문제점이 존재한다. 즉 한정된 숫자의 노드로 구성된 네트워크에만 적용가능하다는 점이다.

두 번째 문제점은 각 노드의 잔존 에너지 량에 대한 고려를 제시하고 있지 못하므로 네트워크의 분할(Network Partitioning) 문제나 센싱 홀(Sensing Hole)이 발생 할 수 있다. 그러므로 네트워크의 크기의 변화에 따른 요구사항을 만족하는 네트워크 확장성을 완벽하게 제시할 수 있고, 각 노드의 에너지 잔류량을 고려할 수 있는 개선책이 필요하다.

III. 개선된 비트맵 기반 라우팅 프로토콜

본 장에서는 2장에서 설명한 기존의 비트맵 테이블 기법의 문제점을 해결할 수 있는 개선된 비트맵 기반 라우팅 프로토콜을 제안한다. 먼저 제안하는 프로토콜의 자료구조를 제시하고, 이를 기반으로 라우팅 설정 단계와 라우팅 단계로 나누어서 설명한다.

3.1 자료구조

센서 네트워크에서 필요한 라우팅 테이블을 구성하기 위해서 각 센서노드는 자신의 식별자를 포함한 이웃노드들의 식별자와 각 노드간의 연결정보를 유지하여야 한다. 그림 2 (a)는 노드 식별자를 위한 자료구조를 보여주고, 그림 2 (b)는 비트맵 테이블을 보여준다. 그림 2 (a)에서 $Nodes$ 테이블의 0번째 인덱스에는 노드 자신의 ID와 에너지 잔류량 ER 이 저장되고 나머지 인덱스에는 자신의 이웃 노드들의 정보가 저장된다. 그림 2 (b)에서 $Bitmap$ 은 자신의 노드와 이웃노드와의 연결정보를 1이나 0으로 표시한다.

$$Nodes[i] = \begin{bmatrix} 0 & 1 & \dots & i & \dots & n-1 \\ ID_0 & ID_1 & \dots & ID_i & \dots & ID_{n-1} \\ ER_0 & ER_1 & \dots & ER_i & \dots & ER_{n-1} \end{bmatrix}$$

<a> 노드 식별자 자료구조

$$Bitmap[i] = \begin{bmatrix} 0 & 0 & 1 & 2 & 3 & 4 & \dots & n-1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ n-1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

 비트맵 테이블 자료구조

<그림 2> 자료구조

테이블을 이용하는 라우팅 프로토콜에 있어서 메모리 사용량은 상당히 중요한 요소이다. 그림 2의 자료구조를 사용하는 센서네트워크 라우팅 프로토콜에 있어서 가장 기본적인 노드 식별자와 비트맵 테이블을 위한 자료구조의 크기는 전체 노드의 개수와 연관 관계가 있을 것이다.

하지만, 이러한 기본적인 자료구조를 사용한다면 메모리 제약성이 강한 센서노드에서는 적용되기 어려울 것이다. 이러한 문제를 해결하기 위해서 본 논문에서는 노드의 메모리 제약성을 고려하여 노드 식별자와 비트맵 테이블 자료구조의 크기를 결정한다. 즉, 일반적인 경우의 테이블은 센서노드의 수를 n 이라고 할 경우, $2n \log n$ 비트의 크기를 가지고 비트맵 테이블은 n^2 비트의 크기를

가진다. 하지만, 본 논문에서 사용하는 테이블은 센서네트워크의 모든 노드에 관한 정보를 저장하지 않고, 각 노드가 항상 $2k \log k + k^2 < \delta$ 인 노드의 연결정보만을 저장할 수 있도록 설정한다. 여기서, k 는 $k < n$ 제한을 가진 센서노드의 수이고, δ 는 센서노드의 메모리 요구사항을 고려하기 위한 속성 값이다.

3.2 라우팅 설정 단계

본 소절에서는 라우팅 시 경로를 선택하는데 사용될 *Bitmap*(라우팅 테이블)을 생성한다. *Bitmap*을 생성하기 위한 라우팅 설정단계는 다음과 같다.

단계 1 : 싱크노드는 { ID_i , ER_i , Interest }를 송신한다 (여기서 ID_i 는 노드의 식별자, ER_i 는 노드의 에너지 잔류량, Interest는 질의 정보이다.)를 플러딩한다.

단계 2 : 요구 메시지를 송신한 이웃 노드는 자신이 질의를 처리할 수 있는가를 판별한다. 만약 처리할 수 없다면 아래와 같이 처리한다.

```

if( $(2k \log k + k^2 < \delta)$ )
if(노드의 식별자가 비트맵 내에 존재 ?)
{수신된 테이블과 자신의 테이블을 식별자가 동일한 열끼리 OR 연산 수행;}
else
{식별자 테이블 Nodes에 식별자와 에너지 잔류 정보를 테이블의 마지막 위치 j에 삽입; 자신의 식별자 행과 j번째 열, 자신의 식별자 열과 j번째 행을 1로 표시;}
end if
else
{노드는 현재까지의 정보를 저장; 받은 요구 메시지의 목적노드를 자신의 노드로 설정하여 자신이 마치 제 2의 싱크 노드의 역할을 수행;}

```

단계 1로 이동해서 싱크노드가 요구 메시지를 보낸 것과 동일한 방법으로 수행)

단계 3 : 정보의 갱신이 완료 된 후, 노드는 질의 정보와 노드의 ID 정보 및 ID 테이블을 포함하는 메시지를 단 한번 플러딩한다.

단계 4 : 플러딩을 수행한 후, 기 처리된 요구 메시지를 받으면, 노드는 메시지를 플러딩 하지 않고 다만 자신의 식별자 테이블과 비트맵 테이블 정보를 갱신만 한다.

단계 2에서 노드의 메모리 요구량을 체크하는데 있어서 현재 메모리 가용량 뿐만 아니라 식별자 테이블의 크기 및 비트맵 테이블의 크기도 함께 고려되어야 한다. 이러한 이유는 현재 노드의 메모리 가용량이 δ 를 만족한다고 하더라도 다음 노드는 이전 노드의 정보를 포함한 다른 노드들의 정보를 갱신하게 될 것이고, 이러한 과정에서 노드의 메모리 고갈의 우려가 있기 때문이다. 즉, 본 논문에서 제안하는 비트맵 기반 라우팅 프로토콜은 이와 같이 메모리 가용량을 고려함으로써 기존 기법의 문제점을 해결할 수 있다.

3.3 라우팅 단계

제안하는 프로토콜은 라우팅 테이블 설정 단계에서 생성된 각 노드의 *Nodes*과 *Bitmap*을 기반으로 각각의 BFS(Breadth First Search)기반의 라우팅을 수행한다. 그 과정은 다음과 같다.

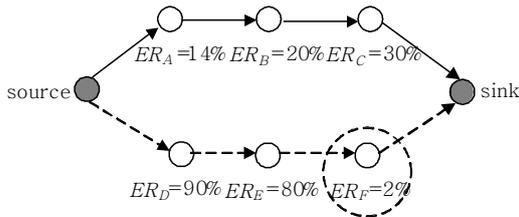
단계 1 : 수신 메시지의 요구사항을 만족하는 노드는 라우팅 테이블 설정단계에서 생성된 *Bitmap*에 기반하여 싱크노드까지 경로를 BFS알고리즘을 이용하여 선택한다. 경로를 선택할 때 *Nodes*에 저장된 각 노드의 에너지 잔류량도 함께 고려된다.

단계 2 : 경로에 속하는 중간 노드들은 단계 1에서의

방식을 이용하여 소스노드로부터 받은 메시지를 싱크노드까지 포워딩 시킨다. 이러한 과정에서 라우팅 설정 단계의 단계 2에서 노드 메모리 잔류량의 요구사항을 만족하지 못해서 마치 제 2의 싱크 노드로서의 역할을 한 노드들은 기 저장된 *Bitmap*정보를 이용하여 자신의 싱크노드까지 받은 메시지를 포워딩 시킨다.

단계 3 : 이러한 작업은 패킷이 목적지 노드에 도착할 때까지 계속 수행된다.

본 논문에서 제안하는 라우팅 프로토콜은 임의의 경로가 최소의 홉 수를 가진다고 하더라도 단계 1에서 에너지 잔류량이 시스템의 임계값 미만이면 라우팅을 위한 후보에서 제거한다. 이렇게 함으로써 네트워크 전체에 에너지 소비의 분산 효과를 최대화 할 수 있다. 즉, 본 논문에서 제안한 라우팅 프로토콜은 *Nodes* 테이블에 각 노드들의 에너지 잔류량인 *ER*을 저장하고 있으므로 라우팅 시 노드의 에너지 고갈이 될 수 있는 노드를 배제할 수 있어 기존의 라우팅 프로토콜의 문제였던 네트워크의 분할 문제나 센싱 홀과 같은 문제를 그림 3과 같이 효율적으로 해결할 수 있다.



<그림 3> 에너지 잔류량 사용의 필요성

V. 비교분석

본 장에서는 먼저 본 논문에서 제안한 개선된 비트맵 라우팅 프로토콜의 구현에 있어서 필요한 메모리 요구사

항을 MICAZ를 기반으로 분석하고, 시뮬레이션을 통해서 대표적인 계층형 프로토콜인 LEACH와 성능을 비교한다.

4.1 메모리 요구 사항

그림 4는 본 논문에서 제안한 프로토콜의 테스트를 위해서 사용된 MICAZ를 보여주고, 표 1은 MICAZ의 기본적인 사양을 보여준다[15]. 표 1에서 본 바와 같이 MICAZ는 128KB의 프로그램 메모리를 갖는다.



<그림 4> MICAZ

<표 1> MICAZ 기본사양

항 목	내 용
프로세서	칩 : ATmega 128L. 타입 : 7.3MHz, 8bit
메 모 리	Flash Memory : 128KB, SRAM : 4KB
통신대역	2.4GHz
전 송 륜	250KB
배 터 리	3.3-2.7V @ 50mA배터리

제안한 프로토콜의 구현을 위해서 질의는 노드의 온도 체크를 이용하였다. 본 응용에 필요한 전체 메모리는 다음과 같이 사용된다.

- 라우팅 프로토콜 프로그램 : 약 50KB
- *Nodes* 테이블 : $2k \log k$ 비트
- *Bitmap* 테이블 : k^2 비트

즉, 본 논문에서 MICAZ를 기본노드로 하여 센서네트워크를 구성할 경우 한 노드의 가용 메모리는 128KB이고, 프로그램을 위해서 50KB와 테이블들을 위해서 78KB 정도를 이용할 수 있으므로 δ 는 78KB로 설정될 수 있다. 하지만, MICAZ는 8비트 프로세서를 가지므로, 전체적인 δ 의 크기는 워드 단위로 측정하는 것이 적절할 것으로 생각된다. 그러므로 본 논문의 프로토콜 구현에서 적용한 δ 크기는 노드의 개수 $n=256$ 을 고려하여 비트맵의 크기인 65,536바이트와 식별자 배열의 크기인 512바이트를 더한 66,048바이트로 설정한다.

4.2 성능분석

본 소절에서는 본 논문에서 제안한 프로토콜과 기존의 비트맵 테이블 기법이 필요로 하는 전체 메시지의 수를 비교하고 시뮬레이션을 통하여 성능을 분석한다.

센서 네트워크의 라우팅 프로토콜을 설계하는데 있어서 메시지의 수는 센서 노드의 한정된 자원을 얼마나 효율적으로 사용하는가의 여부를 판단하는 중요한 근거이다. 표 2는 각 프로토콜간의 전체 메시지 수를 보여준다. 표 2에서 본 바와 같이 본 논문에서 제안한 프로토콜은 각 노드에서 모든 노드로 라우팅 경로 설정 메시지를 단 한번만 전송하므로, LEACH에 비해서 많은 수의 메시지를 줄일 수 있다.

<표 2> 메시지 수 비교

프로토콜	메시지 수
LEACH	$\sum_{u=1}^{n/c_u} k_u + \sum_{v=v+1}^{n/c_w} k_v$ $+ \dots + \sum_{x=x+1}^{n/c_y} k_x + c$
제안한 프로토콜	$\sum_{i=1}^n k_i$

표 3은 기존의 비트맵 기법과 개선된 비트맵 기반 프로토콜을 비교한다. 표 3에서 보듯이 기존 비트맵 테이블의 크기를 $2n \log n * n^2$ 에서 δ 로 감소시켰음을 알 수 있다. 이는 단순히 메모리의 감소만으로 끝나는 것이 아니라 프로토콜 상에서 δ 를 이용한 확장성을 제공함으로써, 네트워크의 노드 수가 일정 수를 넘어서는 경우 적용이 불가능했던 기존 기법의 문제점을 완벽히 해결한 것이다. 즉, 제안하는 프로토콜에서는 네트워크를 구성하는 노드의 숫자는 더 이상 문제가 되지 않는다. 왜냐하면 항상 제한된 크기의 테이블의 사이즈를 가지는 프로토콜은 시스템 파라메타 δ 를 이용해서 제어되기 때문이다. 또한 네트워크의 센싱홀 문제나 네트워크 분할문제도 해결할 수 있다.

<표 3> 메모리 크기와 센싱홀 문제 비교

프로토콜	라우팅 시 필요한 메모리크기	센싱 홀 문제
기존 비트맵기법	$2n \log n + n^2$	고려하지 않음
개선된 비트맵기반 프로토콜	δ	고려함

IV. 결론

본 논문에서는 센서네트워크에서 네트워크 확장성과 에너지 및 메모리 효율성 개선을 위한 비트맵기반 라우팅 프로토콜을 제안 하였다. 제안한 라우팅 프로토콜은 기존 비트맵 프로토콜의 문제점을 개선하기 위하여, 시스템 파라메타 δ 를 도입해서 테이블의 사이즈를 제어함으로써 네트워크를 구성하는 노드 수에 상관없이 비트맵 테이블을 이용한 라우팅이 가능하도록 하였다. 또한 에너지 고갈이 될 수 있는 노드를 사전에 배제할 수 있도록 했다.

이를 통해서 기존의 비트맵 프로토콜의 문제였던 네

트위크의 노드 수가 일정 수를 넘어서는 경우 적용이 불가능했던 문제점을 완벽히 해결했고, 또한 네트워크 분할 문제나 센싱 홀과 같은 문제도 효율적으로 해결할 수 있었다.

참고문헌

- [1] R. Min, M. Bhardwaj, S. Cho, E. Shih, A. Sinha, A. Wang, A. Chandrakasan, "Low Power Wireless Sensor Network," Proceeding of International Conderence on VLSI Design, 2001, pp. 205-210.
- [2] Ian F. Akyildiz, Weilian Su, Yogesh San Karasubramaniam, Erdal Cayirci, "A Survey on Sensor Network," IEEE Communications Magazine, vol. 40, no. 8, 2002, pp. 102-114.
- [3] K. Sohrabi, J. Gao, V. Ailawadhi, G. J. Pottie, "Protocol for Self-Organization of a Wireless Sensor Network," IEEE Personal Communications, vol. 7, no. 5, 2000, pp. 16-27.
- [4] M. Weiser, "Some Computer Science Issure in Ubiquitous Computing," Comm. ACM, vol. 36, no. 7, 1993, pp. 75-84.
- [5] D. Estrin, D. Culler, K. Pister, G. Sukhatme, "Connecting the Physical World with Pervasive Network," Pervasive Computing IEEE, vol. 1, no. 1, 2002, pp. 59-69.
- [6] Mohammad Ilyas, Lmad Mahgoub, Handbook of sensor Networks: Compact Wireless and Wired Sensing Systems, CRC press, 2005.
- [7] 이상학, 무선센서네트워크기술, 경희대학교출판국, 2005. Conderence on VLSI Design, 2001, pp. 205-210.
- [8] Jie Wu, Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Network, Auerbach publications, 2005.
- [9] C. Inteanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," IEEE/ACM Mobicom, 2000, pp. 56-67.
- [10] D. Estrin, R. Govindan, J. Heidmann, S. Kumar, "Next century Challenges: Scalable Coordination in Sensor Networks," In Mobile Computing and Networking, 1999, pp. 263-270.
- [11] Wendi Rabiner Heinzelman, Anatha Chandrak Asan, Hari Balakrishnan, "Energy Efficient Communication Protocols for Wireless Microsensor Network," in Proc. of Hawaii International Conference on Systems Science, 2000, pp. 8020-8026.
- [12] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, D. Estrin, "Networking Issue in Wireless Sensor Networks," Journal of Parallel and Distributed Computing(JPDC), 2003, pp. 799-814.
- [13] D. Tian, N. D. Georganas, "Energy Efficient Routing with Guaranteed Delivery in Wireless Sensor Network," WCNC 2003, vol. 3, 2003, pp. 16-20.
- [14] 정상준, "무선센서 네트워크에서의 비트맵 브로드 캐스팅 라우팅 알고리즘," 한국통신학회논문지, vol. 32, no. 5, 2006, pp. 459-468.
- [15] J. Hill, Spec mote, <http://www.cs.berkeley.edu/~hjill/spec/index.html>

■ 저자소개 ■



최 해 원
Choi, Hae Won

2005년 9월~현재
경운대학교 컴퓨터공학과 전임강사
2009년 2월 경북대학교 컴퓨터공학과(공학박사)
2000년 2월 경북대학교 컴퓨터공학과(공학석사)
1996년 2월 경일대학교 컴퓨터공학과(공학사)

관심분야 : 유비쿼터스 컴퓨팅, 응용 알고리즘
E-mail : happychw@kw.ac.kr



김 상 진
Kim, Sang Jin

1999년 9월~현재
경운대학교 컴퓨터공학과 조교수
2000년 8월 경북대학교 컴퓨터공학과(공학박사)
1996년 2월 경북대학교 컴퓨터공학과(공학석사)
1994년 2월 계명대학교 컴퓨터공학과(공학사)

관심분야 : 알고리즘, 유비쿼터스 컴퓨팅
E-mail : sjkim@kw.ac.kr



류 명 춘
Ryo, Myung Chun

1997년 3월~현재
경운대학교 컴퓨터공학과 부교수
2009년 2월 영남대학교 컴퓨터공학과(공학박사)
1991년 2월 영남대학교 컴퓨터공학과(공학석사)
1989년 2월 영남대학교 컴퓨터공학과(공학사)

관심분야 : 지능정보 시스템, 바이오인포메틱스, 유비쿼터스 컴퓨팅
E-mail : mcryoo@kw.ac.kr

논문접수일 : 2009년 5월 12일
수 정 일 : 2009년 7월 10일 (1차)
: 2009년 8월 10일 (2차)
게재확정일 : 2009년 8월 26일