

# 단거리 데이터 전달 무선 센서네트워크 라우팅 기법

안 광 선<sup>†</sup>

요 약

무선센서네트워크는 센서필드 안에서 어떤 이벤트를 센싱하는 수많은 센서노드들로 구성된다. 무선센서네트워크에서 이벤트를 인식한 소스노드와 이 값을 처리하는 싱크노드간에 데이터가 전송되며 싱크노드의 이동성을 고려한 에너지 효율적인 라우팅 기법이 필요하다. 본 연구에서는 데이터패스 템플릿을 이용하여 미리 정해진 방향으로 소스노드로부터의 데이터공고, 싱크노드로부터의 데이터요구 메시지를 전송한다. 다음 단계로 센싱한 데이터를 단거리계산법을 사용하여 싱크노드에게 전송한다. 그리드 개념과 이동성 싱크가 존재하는 센서네트워크에 대한 기존의 TTDD 라우팅 프로토콜과 CBPER 프로토콜을 본 연구에서 제안한 SPDR 프로토콜과 시뮬레이션을 통하여 비교한 결과, 제안된 프로토콜이 기존의 TTDD나 CBPER보다 향상된 에너지 효율성을 가졌음을 보여준다.

키워드 : 무선센서네트워크, 클러스터헤드, 데이터패스 템플릿, 그리드, 단거리계산법

## A Short Path Data Routing Protocol for Wireless Sensor Network

KwangSeon Ahn<sup>†</sup>

ABSTRACT

Wireless sensor networks have many sensor nodes which response sudden events in a sensor fields. Some efficient routing protocol is required in a sensor networks with mobile sink node. A data-path template is offered for the data announcement and data request from source node and sink node respectively. Sensed data are transferred from source node to sink node using short-distance calculation. Typical protocols for the wireless networks with mobile sink are TTDD(Two-Tier Data Dissemination) and CBPER(Cluster-Based Power-Efficient Routing). The proposed SPDR(Short-Path Data Routing) protocol in this paper shows more improved energy efficiencies from the result of simulations than the typical protocols.

Key Words : Wireless Sensor Network, Cluster Head, Data Path Template, Grid, Short-Distance Calculation

### 1. 서 론

센서네트워크는 통상적으로 특정 지역에 소형의 센서노드를 설치하여 주변 정보 또는 특정 목적의 정보를 획득하고, 베이스 스테이션이 정보를 수집하여 이를 활용하기 위한 서비스 환경을 말한다. 센서네트워크가 기존의 네트워크와 구분되는 점은 기본 목적이 상호간의 정보 전달보다는 자동화된 원격 정보의 수집에 있다는 것이다. 즉, 각 센서노드가 특정 목적을 위해 필요한 주변정보를 센싱하고, 센싱된 정보를 센서노드 간의 무선통신을 이용하여 특정 지점으로 자동화된 방식으로 전달함으로써, 사용자가 센서필드 주변의 정보를 원격으로 수집하여 활용할 수 있다는 것이다.[1] 이러한 센서네트워크의 전통적인 개념은 무선의 센서필드 개념을 중심으로 불특정 공간에 배포된 센서로부터 수집된 정

보를 일괄적으로 활용하는 무선 센서네트워크(wireless sensor network : WSN)를 의미한다.[2]

센서네트워크에 사용되는 센서는 무인으로 동작되거나 사람이 접근하기 힘든 환경에서 동작하는 경우가 대부분이다. 또한 다량의 센서를 유포하여 센서네트워크를 형성하는 경우가 많아 센서노드의 크기가 작고 가격이 저렴해야 하므로 노드의 소모 전력량과 데이터 처리 능력에 제한이 있게 된다. 이런 센서의 제약 사항들과 센서네트워크의 용도에 따라 네트워크에서 원활한 통신을 위해 충족되어야 할 설계 요구조건 및 적합한 라우팅 알고리즘이 다르게 된다. 센서네트워크의 주요 네트워크 토폴로지 및 특징은 센서노드의 제한된 배터리 등으로 인해 대부분의 경우에 원 홉 간의 통신이 아니라 멀티 홉 라우팅을 통하여 산재해 있는 센서들 간 통신이 이루어지게 되고, 최종적으로 싱크노드를 통해 센서로부터 취득한 데이터를 취합하게 된다. 여기서 싱크노드는 센서네트워크에서 이루어지는 센싱 정보를 위한 통신의 최종 목적지로서, 기존의 전통적인 유무선 네트워크와

※ 본 논문은 경북대학교 BK21 사업단의 지원 사업에 의해 수행되었음

† 종신회원 : 국민대학교 수학과 부교수

논문접수 : 2007년 5월 28일, 심사완료 : 2007년 7월 16일

센서네트워크의 게이트웨이 역할을 한다[3].

현재까지 센서네트워크를 위한 다양한 종류의 라우팅 프로토콜이 제안 되었다[4]. 그 중에서도 TTDD (Two-Tier Data Dissemination)와 CBPER(A Cluster Based Power Efficient Routing) 프로토콜은 모바일 싱크노드를 가진 센서네트워크 환경의 문제점을 해결하기 위해서 제안되었다. TTDD는 관심 사건이 발생하면 그리드를 만들어 데이터공고(data announcement) 메시지를 전송하고 그리드 구조를 이용하여 센싱한 데이터를 전달한다. 하지만 소스노드가 많아지면 각 소스노드의 그리드를 형성하는데 오버헤드가 발생한다[5]. CBPER프로토콜에서는 데이터공고 메시지가 세로축으로, 데이터요구 메시지는 가로축으로 보내며, 두 메시지를 가진 헤더 노드를 통하여 소스노드에서 싱크노드로 데이터를 전송한다. 하지만 TTDD와 마찬가지로 데이터공고, 데이터요구(data request), 데이터전송을 그리드의 가로, 세로축으로만 전송하는 문제점이 있다[6]. 본 논문에서는 제한된 에너지를 효율적으로 사용하기 위해서 데이터패스 템플릿(data-path template)과 단거리계산법(short distance calculation)을 이용하여 에너지 효율적인 라우팅 프로토콜을 제안한다.

## 2. 관련 연구

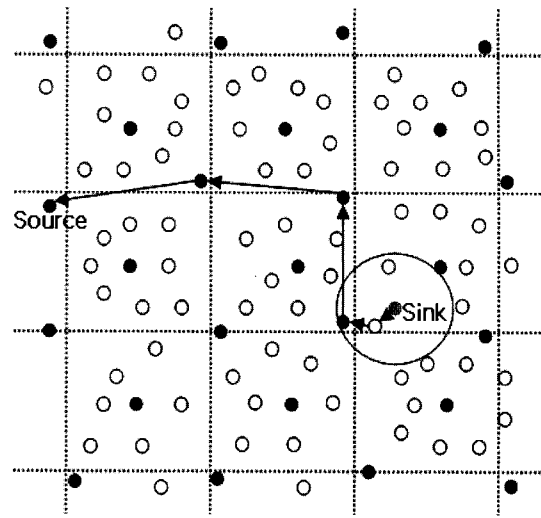
본 장에서는 모바일 싱크노드를 가진 센서네트워크 환경에서 동작하는 라우팅 프로토콜인 두 계층 데이터 전송 방식인 TTDD 라우팅 프로토콜과 클러스터 기반의 에너지 효율적인 라우팅 프로토콜인 CBPER 에 대해 설명하고 두 라우팅 프로토콜은 다음과 같은 가정과 요구사항을 갖는다.

- 넓은 센서필드에서 동작한다.
- 각 센서노드와 싱크노드는 자신의 위치를 알아야 한다.
- 각 센서노드들은 이동성이 없다.
- 여러 개의 이동성 싱크노드가 존재한다.

### 2.1 TTDD 라우팅 프로토콜

이동성 싱크 문제를 해결하기 위한 TTDD(Two - Tier Data Dissemination) 라우팅 프로토콜은 소스노드와 센서노드들이 고정되어 있고 싱크노드에 대해서만 이동성을 지원하는 알고리즘이다.

동작을 살펴보면, 일단 이벤트가 발생 하면 그 주변의 센서노드들이 집단적으로 이벤트를 센싱한다. 그리고 그 중 하나의 센서노드가 대표로 소스노드가 되어 이벤트를 처리한다. 그리드구조는 소스노드를 시작으로 전체 센서필드를  $\alpha \times \alpha$  정사각형 형태의 클러스터셀로 구성된다. 처리 과정에서 소스노드는 자신과 이웃하고 있는 4개의 교차점(dissemination points)의 위치를 계산하고, 각각의 교차점들은 같은 방법으로 확장되어 전 센서필드를 그리드 구조로 나눈다. TTDD는 두 가지 계층인 이벤트를 센싱하는 계층과 그리드를 통해 데이터를 전달하는 계층을 가진다.



(그림 1) TTDD의 센싱한 데이터 전송 경로

싱크노드가 소스노드를 찾는 과정은 (그림 1)과 같다. 싱크노드는 자기와 가장 가까이에 있는 그리드의 교차점의 교차점노드(dissemination node)에게 소스노드의 위치 정보를 가지고 그리드의 교차점을 통하여 멀티홉으로 소스노드로 연결된다. 다음 과정에서 소스노드는 역방향으로 싱크노드에게 데이터를 보낸다. 싱크노드가 이동 할 경우 같은 방법으로 데이터요구를 하여 데이터를 받는데, 여기서 이전에 전달된 데이터의 정보를 가지고 있는 그리드의 교차점을 만나는 경우, 다시 소스노드까지 갈 필요 없이 바로 그 그리드의 교차점에 있는 데이터 정보를 받을 수 있다. TTDD는 동작방식에서 알 수 있듯이 싱크의 이동성을 효과적으로 지원해 줄 수 있는 알고리즘이다.

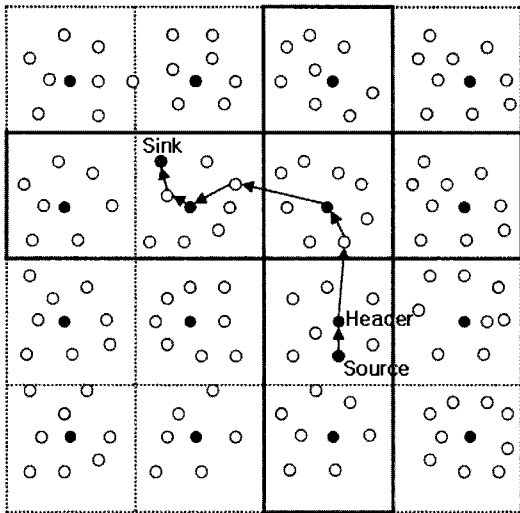
### 2.2 CBPER 프로토콜

CBPER(A Cluster Based Power Efficient Routing) 프로토콜은 라우팅을 위해 전송되는 패킷의 수를 줄여 센서노드들이 에너지를 절약한다. 이 프로토콜은 그리드와 클러스터 개념을 사용함으로써 특정 센서노드에게만 데이터전송 정보를 보낸다. CBPER은 클러스터헤드 선출, 데이터공고, 데이터요구 그리고, 데이터 전달로 동작한다.

CBPER은 TTDD 프로토콜이 이벤트가 발생 했을 때 싱크노드에게 데이터를 전송하기 위해서 소스노드가 능동적으로 그리드를 만드는 것과 달리 센서필드에 대하여 고정된 하나의 그리드의 셀 구조를 형성하고 각 셀(cell)에 아이디를 부여한다. 센서필드에 산재된 센서노드들은 그리드 상의 자신이 속한 셀의 아이디를 안다.

각 셀에서는 초기 클러스터헤더를 선출하기 위해 플러딩을 한다. 이 플러딩의 패킷에는 셀 아이디, 가용 에너지, 패킷이 전송된 시간과 헤더 지속 시간이 있는데 그 중 패킷이 전송된 시간이 가장 빠른 노드가 클러스터헤더가 된다.

센서필드에서 센싱한 데이터를 가지는 소스노드가 발생하면 그 소스노드는 해당 클러스터헤더 노드까지 데이터공고 메시지를 전송한다. 클러스터헤더 노드는 자신이 위치한 가



(그림 2) CBPER의 센싱한 데이터 전송 경로

로축에 있는 클러스터헤더 노드들에게 데이터공고 메시지를 전송한다. 가로축에 있는 클러스터헤더들은 그 데이터공고 메시지를 자신의 캐쉬에 저장한다. 그 데이터공고 메시지에 소스노드의 위치, 셀 아이디 그리고 데이터를 생성한 시간이 기록되어 있다.

싱크노드가 센서필드에 들어와서 센싱한 데이터를 요구할 때 가장 가까운 인접중계노드(immediate agent node)로 통하여 해당 셀의 클러스터헤더에게 데이터요구 메시지를 보낸다. 클러스터헤더는 자신의 세로축에 위치한 클러스터헤더들에게 데이터요구 메시지를 전송한다. 데이터요구 메시지를 받은 세로축의 클러스터헤더들은 캐쉬에 데이터공고 메시지를 확인하고 그 메시지가 있다면 기록된 데이터 생성 시간의 유효성을 판단하여 가로축의 소스노드의 위치로 데이터요구 메시지를 전송한다

데이터요구 메시지를 받은 소스노드는 싱크노드의 위치정보를 확인하여 데이터요구 메시지가 전송되어온 경로의 역방향으로 센싱한 데이터를 전송한다. (그림 2)는 소스노드가 센싱한 데이터를 싱크노드에게 전송하는 경로를 보여준다.

### 3. SPDR 프로토콜

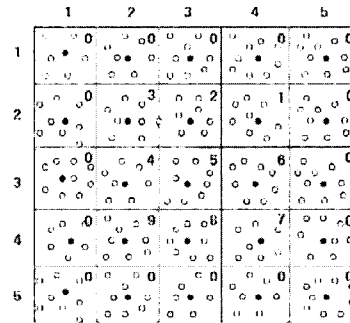
본 장에서는 단거리 데이터 전달 무선센서네트워크 라우팅 기법(A Short Path Data Routing Protocol for Wireless Sensor Network : SPDR)을 다음의 가정을 기반으로 하여 설명한다.

- 넓은 센서필드에 수많은 동질의 센서노드들이 배치된다. 그 센서들 중에 거리가 가까운 센서끼리는 원 홉 통신을 하여 데이터를 전송하며, 거리가 먼 센서끼리는 멀티 홉을 사용하여 데이터를 전달한다.
- 각 센서노드는 GPS나 수동적인 방법으로 자신의 위치를 알고 있다. 하지만 모바일 싱크(사용자)는 자신의 위치를 알 수도 모를 수도 있다.

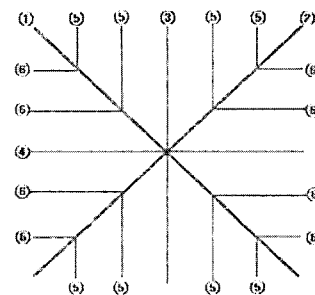
- 각 센서노드는 자신의 가용에너지를 알고 있다.
- 여러 개의 이동성 싱크가 존재한다.

#### 3.1 그리드 구조

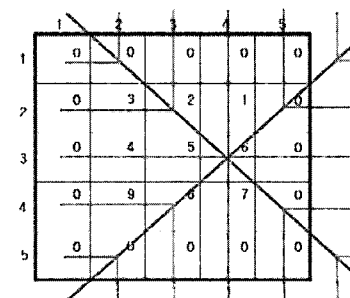
각 센서노드가 센서필드에 전개되기 전에, 절대 좌표를 이용해서 센서필드에 고정된 그리드를 만들고 각 그리드의 셀에 고유한 아이디를 부여한다. 센서필드의 각 셀은 [가로, 세로]의 정수 아이디를 부여하고 센서필드 중 경계지역을 제외한 셀을 중앙클러스터라고 하는데 그 각 셀에는 부번호를 부여한다. 부번호는 중앙클러스터가 아닌 셀에는 0을 부여하고 중앙클러스터의 셀들에게는 순차적인 번호를 부여한다. 중앙클러스터의 부번호로 중앙 헤더의 위치가 일정한 시간간격으로 바뀐다. (그림 3)은 그리드의 구조, 각 셀의 아이디 그리고 부번호를 보여준다. 센서노드가 센서필드에 산재되면 각각의 센서 노드는 자신의 클러스터셀 아이디를 갖게 되고 그 값을 저장한다. 이 아이디는 절대적인 좌표이며 바뀌지 않는다.



(그림 3) 그리드



(그림 4) 데이터패스 템플릿



(그림 5) 그리드+데이터패스 템플릿

3.2 데이터패스 템플릿 구성

데이터공고, 데이터요구 그리고 데이터전송 경로를 짧게 하기 위하여 데이터패스 템플릿을 제안한다. 이 템플릿의 구조는 센서필드 내에 구성된 클러스터 셀들 중에서 선택되는 중앙클러스터에 적용되며 이 템플릿이 제시하는 방향으로 데이터공고, 데이터요구 메시지가 전송된다. 그리고 데이터패스 템플릿의 중심은 중앙 헤더에 항상 위치한다. 기존의 TTDD와 CUPER 기법에서는 데이터공고, 데이터요구, 전송경로가 가로와 세로만의 방향을 갖는다. 그러나 본 연구에서 제안한 기법은 기존의 가로와 세로만의 경로 대신, 적용 가능한 구간에 대해 대각선의 방향을 적용함으로써 전체적인 데이터 서비스 경로를 짧게 한다.

데이터패스 템플릿을 만드는 순서 (그림 4)와 같이 한다. 선택된 중앙 헤더에서 대각선 ①, ②번 패스를 만든 다음에 가로와 세로의 ③, ④번 패스를 만든다. 그 외의 대각선을 지나며 가로축과 세로축에 평행한 ⑤, ⑥번 패스를 만든다. (그림 5)는 가로 5, 세로 5의 그리드 구조에서 중앙 헤더의 아이디가 [4, 3]이고 중앙 클러스터가 6번에 대한 템플릿 위치를 보여준다. 만약, 센서 필드가 넓어진다면 그리드 셀을 늘려주고 중앙 클러스터의 셀들에게 순차적인 번호를 부여한다. (그림 5)의 중앙클러스터는 1번부터 9번까지 9개이며, 본 기법은 클러스터의 에너지 소비를 분산시키기 위하여 중앙 클러스터의 위치를 순차적으로 변경시키면서 서비스를 진행한다.

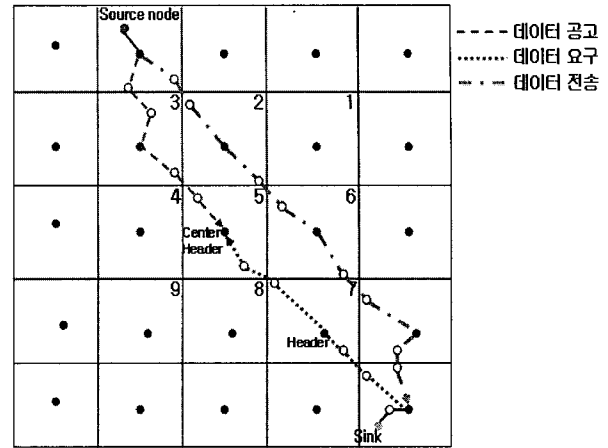
3.3 클러스터 헤더 및 중앙 헤더 선출

센서필드에 센서노드들이 전개되어 자신의 위치에 해당되는 셀의 아이디를 확인하면 각 셀에서는 클러스터헤더를 선출하기 위해 CUPER과 같은 방법으로 지역적인 플러딩을 한다. 이 플러딩의 패킷에는 클러스터셀 아이디, 가용 에너지, 패킷이 전송된 시간과 헤더 지속 시간이 있다. 패킷을 받은 센서노드는 자신의 아이디와 비교하여 다르면 패킷 드랍(drop)을 하고, 패킷을 가장 먼저 전송한 센서노드가 클러스터헤더가 된다. 클러스터헤더의 가용 에너지가 임계치 미만일 경우, 지속시간 안에 새로운 클러스터헤더를 가용 에너지를 고려하여 선출하고 기존의 클러스터헤더는 일반 센서노드로 돌아간다[6].

각 셀의 클러스터헤더가 선출되면 중앙의 클러스터들(부번호가 0이 아닌 셀) 중, 하나의 클러스터헤더가 중앙클러스터의 대표 헤더가 되어 부 번호의 순차적인 번호로 동일한 특정시간동안의 중앙 헤더(center header)의 역할을 한다. 소스노드는 특정시간에 해당되는 중앙 헤더에게 데이터공고 메시지를 보낸다. 중앙 헤더는 데이터공고 메시지를 받아 자신의 캐쉬에 저장하고, 싱크노 특정시간에 해당되는 중앙 헤더에게 데이터요구 메시지를 전송한다.

3.4 데이터 전송

데이터공고와 데이터요구는 데이터패스 템플릿을 통해 이루어진다. 템플릿이 제공하는 데이터 패스는 소스노드와 싱



(그림 6) 질의 및 데이터 전송 경로

크노드의 위치에서 선택된 중앙 헤더까지의 최단거리이고 센싱된 데이터들은 소스노드의 위치와 싱크노드의 위치 정보를 사용하여 최단거리 패스를 선택한다. 센싱한 데이터 전송은 소스노드의 위치와 싱크노드의 위치를 사용하여 최단거리 패스를 선택, 전송한다. (그림 6)은 소스노드가 [2,1]의 셀에, 중앙 헤더가 [3, 3]의 셀에, 그리고 싱크가 [5,5]의 셀에 위치했을 때의 질의 및 데이터 전송 경로이다.

3.4.1 데이터공고 및 데이터요구

센서노드가 특정 데이터를 센싱 했을 때, 그 노드를 소스노드라고 한다. 소스노드는 데이터공고 메시지를 자신의 지역 클러스터셀의 클러스터헤더에게 전송한다. 소스노드의 클러스터헤더는 데이터 센싱을 알리는 데이터공고 메시지를 데이터패스 템플릿을 이용하여 중앙 헤더에게 전송한다. 데이터공고 메시지에는 소스노드의 위치, 클러스터셀 아이디, 데이터를 센싱한 시간으로 구성된다. 만약 여러 개의 소스노드가 같은 데이터를 센싱하여 지역셀의 클러스터헤더에게 보냈다면 클러스터헤더는 그 데이터를 병합한다.

싱크노드는 자신의 위치에서 가장 가까운 센서노드를 인접중계노드로 선택하여 자신이 포함된 클러스터헤더에게 데이터요구 메시지를 전송한다. 클러스터헤더는 데이터요구 메시지를 받아서 데이터패스 템플릿을 이용하여 중앙 헤더에게 데이터요구 메시지를 전송한다. 싱크노드와 중앙 헤더 사이의 데이터요구 메시지를 전달하는 클러스터헤더들도 이 데이터요구에 관련된 정보들을 자신들의 캐쉬에 저장한다. 데이터요구 메시지는 싱크의 위치, 클러스터셀 아이디, 질의한 시간으로 구성된다. 데이터공고 메시지를 저장 하고 있는 중앙 헤더가 데이터요구 메시지를 받으면 데이터공고 메시지의 데이터 센싱 시간을 확인하여 데이터가 유효한지 확인한다. 데이터가 유효하다면 중앙 헤더는 데이터공고 메시지의 셀 아이디를 이용하여 소스노드에게 데이터공고 메시지가 전달되었던 경로의 역방향으로 데이터요구 메시지를 전송한다.

### 3.4.2 데이터 전송

데이터요구 메시지를 받은 소스노드는 센싱한 실질적인 데이터를 싱크노드에게 전송한다. 전송 경로는 소스노드와 싱크노드 사이의 단거리계산법을 이용한다. (그림 7)은 소스노드가 있는 클러스터헤더에서 싱크노드가 있는 클러스터헤더까지의 단거리 경로를 계산하는 슈도코드이다. 슈도코드는 DATATRANSMIT 함수와 SIGN 함수로 구성된다. DATATRANSMIT 함수는 싱크노드의 Sink(x, y)에서 소스노드의 SourceNode(x, y)를 뺀 Distance(x, y)의 거리값을 (0, 0)을 만들어 가면서 싱크노드가 있는 클러스터헤더까지 찾아 간다. 소스노드가 있는 클러스터헤더가 데이터요구 메시지를 가로축과 세로축의 클러스터헤더에게 받았으면 대각선으로, 대각선의 클러스터헤더에게 받았으면 세로축이나 가로축으로 싱크노드까지 데이터전송을 한다. SIGN 함수는 Distance(x, y)에서 b, x, y 값의 부호를 이용하여 단거리전송 경로의 방향을 결정한다.

싱크노드는 자신의 위치를 일정한 주기로 인접중계노드를 통하여 해당 클러스터 헤더에게 알리므로 한 셀 내에서 싱크노드의 이동은 싱크노드가 위치한 셀의 클러스터 헤더에 의해 관리된다. 싱크노드가 주변 셀로 이동하였을 경우 기존의 셀에서의 마지막 인접중계노드가 해당 클러스터 헤더에게 싱크노드의 부채를 알린다. 해당 클러스터 헤더는 마지막 인접중계노드의 위치와 인접한 셀의 클러스터 헤더에게 데이터를 전달한다.

```

TargetClusterHeader(x, y) = SourceNode(x, y)
Distance(x, y) = Sink(x, y) - SourceNode(x, y)

DATATRANSMIT{
  if(RequestMessageLastpath.x == SourceNode.x
  or RequestMessageLastpath.y == SourceNode.y){
    while(Distance.x==0 or Distance.y==0){
      sign(x), sign(y)
    }
  }else{
    while(Distance.|x| == Distance.|y|){
      if(Distance.|x|>Distance.|y|){
        sign(x)
      }else{
        sign(y)
      }
    }
    while(Distance.x == 0 and Distance.y==0){
      sign(x), sign(y)
    }
  }
}

SIGN(a){
  if(a>0){
    Distance.a = Distance.a-1
    TargetClusterHeader.a
    =TargetClusterHeader.a+1
  }else if(a<0){
    Distance.a = Distance.a+1
    TargetClusterHeader.a
    = TargetClusterHeader.a-1
  }
}

```

(그림 7) 데이터 전송 슈도코드

## 4. 성능분석

본 장에서는 단거리 데이터 전달 라우팅 기법을 이용한

SPDR 프로토콜을 TTDD 라우팅 프로토콜과 CBPER 프로토콜과 비교하여 통신비용을 분석한다. 전체 통신비용은 각 라우팅 프로토콜의 데이터공고, 데이터요구, 그리고 데이터 통신에 대한 비용이다. 데이터 처리, 데이터 병합, 그리고 그리드 유지에 관한 비용은 고려하지 않는다. 또한 SPDR 프로토콜의 중앙 헤더의 위치는 계산의 편의를 위해서 아이디 [3, 3]의 셀로 하였다. 센서필드(A)는 한 변의 길이가 a인 셀로 나누어진다. 각 셀에는  $n = \frac{Na^2}{A}$  개의 센서 노드들이 존재하며 셀 한변의 길이는  $\sqrt{n}$ 으로 나타낼 수 있다. 각 라우팅 프로토콜을 분석하기 위해 다음과 같이 분석모델[5]을 정의한다.

- 센서 필드에 k개의 유동성 싱크노드가 있다.
- 싱크노드는 v의 속도로 m개의 셀을 지나며 d개의 데이터 패킷을 받는다.
- 싱크노드는 한 셀에서  $\frac{d}{m}$  개의 데이터를 받는다.
- 셀 한 변의 길이는  $\sqrt{n}$ 이다.
- 센서필드의 한 변의 길이는  $\sqrt{N}$ 이다.
- 데이터공고, 데이터요구의 패킷수는 l 이다.
- 센서노드와 센서노드 사이의 거리상수는 c 이다.

c 는 하나의 셀이나 전체 센서필드에서 두 노드사이의 거리가 가장 가까운 경우 겹치지는 않으므로 0보다는 크고, 가장 멀리 있을 경우  $\sqrt{2}$ 이다. ( $0 < c \leq \sqrt{2}$ )

### 4.1 데이터공고

TTDD 라우팅 프로토콜은 데이터공고 메시지를 보내기 전에 센서필드에 있는 모든 센서들이 먼저 플러딩을 한다. 그리고 센서필드에서 이벤트를 감지한 소스노드는 그리드를 만들어 데이터공고 메시지를 교차점노드에게 보낸다.

$$\frac{4N}{\sqrt{n}}l \tag{1}$$

CBPER 프로토콜은 이벤트가 발생하여 소스노드가 생기면 해당 클러스터헤더에게 데이터공고 메시지를 전송하고 그 클러스터헤더는 자신의 가로축에 있는 클러스터 헤더들에게 데이터공고 메시지를 전송한다.

$$(c\sqrt{n} + \sqrt{N})l \tag{2}$$

SPDR 프로토콜은 이벤트를 감지한 소스노드가 자신의 해당 셀의 클러스터헤더에게 데이터공고 메시지를 보내고 그 클러스터헤더는 데이터패스 템플릿을 이용하여 중앙 헤더에게 데이터공고 메시지를 전송한다.

$$(c\sqrt{n} + \frac{\sqrt{2N}}{2})l \tag{3}$$

4.2 데이터요구

TTDD는 싱크가 데이터요구 메시지를 인접중계노드에게 전송하기 위해서 지역적인 플러딩을 한다. 인접중계노드는 교차점노드를 통해 데이터요구 메시지를 소스노드까지 전송한다.

$$nl + \sqrt{2}(c\sqrt{N})l \tag{4}$$

CBPER은 싱크가 데이터요구 메시지를 인접중계노드에게 전송한다. 인접중계노드는 해당 지역셀의 클러스터헤더를 통해 자신의 세로축에 있는 클러스터헤더에게 데이터요구 메시지를 전송한다. 만약 세로축에 있는 클러스터헤더의 캐쉬에 데이터공고 메시지가 있다면 그 메시지 정보를 이용하여 소스노드에게 데이터요구 메시지를 전송한다.

$$2c\sqrt{n}l + \sqrt{2}(c\sqrt{N})l \tag{5}$$

SPDR은 싱크가 데이터요구 메시지를 인접중계노드에게 전송한다. 인접중계노드는 데이터패스 템플릿을 이용하여 중앙클러스터헤더에게 데이터요구 메시지를 전송한다. 중앙클러스터헤더에 데이터공고 메시지가 캐쉬에 있다면 그 메시지 정보를 이용하여 소스노드에게 데이터 요구 메시지를 전송한다.

$$2c\sqrt{n}l + (c\sqrt{N})l \tag{6}$$

4.3 데이터전송

TTDD는 소스노드가 데이터요구 메시지를 받으면 싱크가 위치한 교차점노드까지 데이터전송을 한다. 교차점노드는 인접중계노드를 통해 싱크노드까지 데이터전송을 한다.

$$(\sqrt{2}(c\sqrt{N})+c\sqrt{n})\frac{d}{m} \tag{7}$$

CBPER은 데이터요구 메시지의 역방향으로 실질적인 데이터전송을 한다.

$$(2c\sqrt{n})\frac{d}{m} + \sqrt{2}(c\sqrt{N})\frac{d}{m} \tag{8}$$

SPDR은 소스노드가 데이터요구 메시지를 받으면 해당 셀의 클러스터헤더에게 데이터전송을 하고 그 클러스터헤더는 단거리계산법을 이용하여 싱크노드에게 데이터전송을 한다.

$$(2c\sqrt{n})\frac{d}{m}+(c\sqrt{N})\frac{d}{m} \tag{9}$$

4.4 전체통신비용

각 라우팅 프로토콜의 총 통신비용은 k개의 싱크 노드가

m개의 셀을 지나는 경우에 다음과 같다.

TTDD : (10)

$$\begin{aligned} & \frac{4N}{\sqrt{n}}l + \\ & km(nl + \sqrt{2}(c\sqrt{N})l + (\sqrt{2}(c\sqrt{N})+c\sqrt{n})\frac{d}{m}) \\ & = \frac{4N}{\sqrt{n}}l + kmnl + \sqrt{2}kc(ml+d)\sqrt{N} + kdc\sqrt{n} \end{aligned}$$

CBPER : (11)

$$\begin{aligned} & (c\sqrt{n} + \sqrt{N})l + \\ & km(2c\sqrt{n}l + \sqrt{2}(c\sqrt{N})l + ((2c\sqrt{n}) + \sqrt{2}(c\sqrt{N}))\frac{d}{m}) \\ & = (c\sqrt{n} + \sqrt{N})l + \sqrt{2}kc(ml+d)\sqrt{N} + 2kc(ml+d)\sqrt{n} \end{aligned}$$

SPDR : (12)

$$\begin{aligned} & (c\sqrt{n} + \frac{\sqrt{2N}}{2})l + \\ & km(2c\sqrt{n}l + (c\sqrt{N})l + (2c\sqrt{n})\frac{d}{m} + (c\sqrt{N})\frac{d}{m}) \\ & = (c\sqrt{n} + \frac{\sqrt{2N}}{2})l + kc(ml+d)\sqrt{N} + 2kc(ml+d)\sqrt{n} \end{aligned}$$

즉, SPDR는 TTDD보다 데이터공고에서 N을  $\sqrt{N}$ 으로 줄이고 데이터요구와 데이터전송에서  $\sqrt{2}d$ 에서 d로 줄었다. CBPER보다는 데이터공고에서  $\sqrt{N}$ 을  $\frac{\sqrt{2N}}{2}$ 로 줄였고 데이터요구와 데이터전송에서  $\sqrt{2}d$ 에서 d로 줄었다. 이는 데이터패스 템플릿과 단거리계산법을 이용했기 때문이다.

5. 성능평가

본 장에서는 제안한 라우팅 프로토콜의 성능평가를 위해 C 언어를 이용하여 기존의 TTDD 프로토콜과 CBPER 프로토콜과 비교하였다.

5.1 시뮬레이션 환경

시뮬레이션을 위한 센서 네트워크의 크기는 2000 × 2000m이며, 센서노드의 개수는 200개이다. 센서노드가 패킷을 수신할 때 소요되는 에너지의 양은 0.395w이며, 전송할 때 소요되는 에너지의 양은 0.66w의 에너지가 소모된다. 소스노드의 개수와 싱크노드의 개수는 2의 지수승으로 증가하면서 실험하였다. 싱크노드의 속도는 10m/s이며 5m/s의 pause를 가진다. 셀의 사이즈는 600x600m으로 설정되었다.

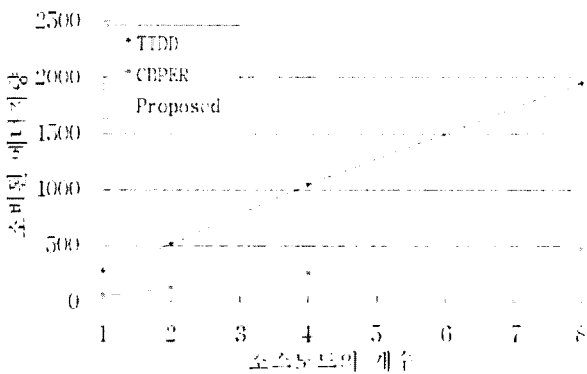
실험은 각각 6개의 네트워크 토폴로지에서 수행하여 평균을 냈으며 각 실험은 200초 동안 수행되었다. 실험에서 사용된 컨트롤 패킷과 데이터 패킷의 크기는 각각 36byte와 64byte이다.

5.2 시뮬레이션 결과 분석

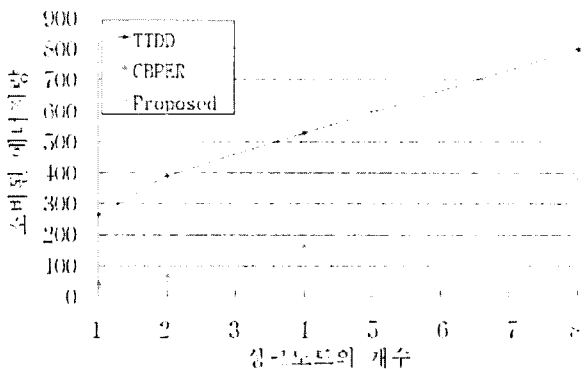
첫 번째 분석으로 소스노드 개수를 증가시키면서 소비된 통신 에너지량을 비교하였다. 싱크노드는 4개로 고정을 하고 소스노드의 개수를 1, 2, 4, 8로 증가시켰다. 싱크노드는 5초의 pause 시간을 가지면서 10 m/s로 이동한다.

(그림 8)은 소스노드 개수 증가에 따른 소비된 에너지량을 보여준다. 소스노드 개수가 8개일 때 제안된 SPDR 프로토콜이 TTDD보다 약 5.5배, CBPER보다 약 1.3배의 에너지 감소를 보여준다. 이는 제안된 SPDR 프로토콜이 고정된 그리드를 사용하며 데이터패스 템플릿을 통해 데이터 요구, 공고 패킷을 줄였기 때문이다. 또한 데이터 전송에 있어서 전송경로는 소스노드와 싱크노드 사이의 클러스터 헤더들이 계산한 최단거리를 계산하여 데이터 패킷을 전송하기 때문이다.

두 번째 분석으로는 싱크 노드의 수를 변화 시키면서 성능을 비교 분석 하였다. 소스 노드는 2개로 고정을 하고 싱크 노드의 수를 1, 2, 4, 8로 변화 시킨다. 싱크 노드는 5초의 pause 시간을 가지면서 10 m/s로 이동한다. (그림 9)는



(그림 8) 소스노드의 개수에 따른 소비된 에너지량



(그림 9) 싱크노드의 개수에 따른 소비된 에너지량

싱크노드 개수 증가에 따른 소비된 에너지량을 보여준다. 싱크노드의 수가 4개 일 때 제안된 프로토콜이 TTDD 프로토콜에 비해서 약 4배의 에너지 절감 효과를 가져왔으며, CBPER 프로토콜보다는 다소 적은 에너지 절감을 보였다. 하지만 싱크노드의 수가 많아질수록 CBPER 프로토콜과 제안된 프로토콜 사이의 에너지 소비량 차이는 점점 차이나는 것을 볼 수 있다. 싱크노드의 수가 8개 일 때는 TTDD 프로토콜보다는 3배의 차이를 CBPER보다 약 1.3배의 차이를 보이고 있다.

위 실험으로 소스노드 개수의 증가와 싱크노드 개수가 증가할수록 에너지의 절감이 커짐을 보여준다. 이는 소스노드와 싱크노드의 증가에 따라 제안된 데이터패스 템플릿과 최단거리 계산 이용을 빈번하게 사용하기 때문이다. 데이터패스와 최단거리 계산을 자주 사용함에 따라 기존에 연구된 프로토콜보다 좋은 에너지 효율성을 보임을 알 수 있다.

6. 결 론

네트워크에서 사용되는 센서는 무인으로 동작되거나 사람이 접근하기 힘든 환경에서 동작하기 때문에 에너지와 데이터 처리 능력에 제한이 있다. 이런 센서의 제약 사항들과 센서네트워크의 용도에 따라 원활한 통신을 위해 충족되어야 할 설계 요구조건이 있으며 적합한 라우팅 알고리즘이 필요하다.

이동성 싱크가 존재하는 센서네트워크에 대한 기존의 전형적인 라우팅 프로토콜은 데이터공고, 데이터요구, 그리고 데이터전송의 경로가 수평과 수직의 방향을 갖는다. 본 연구에서 제안한 SPDR 프로토콜은 부분적으로 대각선의 경로를 택하므로 전체 경로의 길이를 줄인다. 제안된 데이터패스 템플릿은 데이터공고와 데이터요구 과정을 단축시켰으며, 단거리계산법은 데이터전송 경로를 단축시킨다. 데이터경로의 단축은 소스노드와 싱크노드간의 거리를 단축시켜서 통신에 사용되는 에너지 소비를 줄여 전체 센서네트워크의 수명을 연장 시킨다.

본 논문에서는 그리드 개념과 이동성 싱크가 존재하는 센서네트워크에 유용하며 에너지 효율적인 라우팅 프로토콜인 기존의 TTDD 라우팅 프로토콜과 CBPER 프로토콜을 본 논문이 제안하는 SPDR 프로토콜과 비교, 평가하였다. 시뮬레이션 결과 제안된 SPDR는 기존의 TTDD보다 평균 4배, CBPER보다 평균 1.3배의 에너지 감소율을 보였다. 그리고 실험 결과로 부터 소스노드와 싱크노드가 많을수록 좀 더 효율적임을 보였다.

참 고 문 헌

[1] S. Maddes, R. Szewczyk, M. J. Franklin, and D. Culler, "Supporting aggregate queries over ad-hoc wireless sensor network," IEEE Workshop on Mobile Computing

Systems and Applications, May 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, pp. 393-422, Dec. 2001.

[3] 남상엽, 송병훈. "무선 센서네트워크 활용", 상학당, pp.21-22, 2005.

[4] J. Qiangfeng, and D. Manivannan, "Routing Protocols for Sensor Networks," in IEEE CCNC, pp. 93, Jan. 2004.

[5] Haiyun Luo, Fan Ye, Jerry Cheng, Songwu Lu, and Lixia Zhang, "A Two-Tier Data Dissemination in Large-scale Wireless Sensor Networks," Wireless Networks, pp. 161-175, 2005.

[6] 권기석, 이승학, 윤현수, "센서네트워크를 위한 클러스터 기반의 에너지 효율적인 라우팅 프로토콜", 정보과학회논문지, 정보통신 33권, 제 1호, pp. 76-90, 2006.

[7] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," Proceedings of Workshop on Sensor Networks and Applications (WSNA), 2002.

[8] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," IEEE Computer Magazine 34(8), pp. 57-66, 2001.

[9] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," Proceeding of ACM International Conference on Mobile Computing and Networking (MOBICOM '00), 2000.

[10] C. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," IEEE Journal on Selected Areas in Communications 15(7), pp. 1265-1275, 1997.



### 안 광 선

e-mail : gsahn@knu.ac.kr

1972년 연세대학교 전기공학과(학사)

1975년 연세대학교 전기공학과(공학석사)

1975년~1976년 스페리유니백 근무

1980년 연세대학교 전기공학과(공학박사)

1976년~현재 경북대학교 교수

관심분야: 유비쿼터스, 센서네트워크, RFID, Embedded System