

센서 네트워크를 위한 클러스터 기반의 에너지 효율적인 라우팅 프로토콜

(A Cluster-based Power-Efficient Routing Protocol for Sensor Networks)

권기석[†] 이승학[†] 윤현수^{**}
(Ki Suk Kweon) (Seung Hak Lee) (Hyun Soo Yun)

요약 센서 네트워크는 어떤 현상을 감지하기 위해서 관찰 지역 내에 뿌려진 센서 노드들로 구성된 다. 각 센서 노드들의 수명은 전체 센서 네트워크의 수명에 큰 영향을 미친다. 하나의 센서 노드가 배터리 수명을 다하여 죽게 되었을 때 이는 센서 네트워크의 분할을 가져 올 수도 있다. 각 센서 노드들의 수명은 각 노드들의 배터리 용량에 달려있다. 그러므로 네트워크에 있는 모든 센서 노드들이 공평하게 오래 사는 것이 전체 네트워크의 수명을 길게 하는 것이다.

이 논문에서 우리는 클러스터 기반의 에너지 효율적인 라우팅 프로토콜을 제안한다. 이 프로토콜은 여러 개의 이동성 싱크가 존재하는 센서 네트워크에서 효율적인 데이터 전송을 지원한다. 기존의 제안된 Directed Diffusion과 두 계층 데이터 전송 라우팅 프로토콜(TTDD)은 다수의 이동성 싱크와 다수의 소스가 존재하는 네트워크를 지원하기 위해서 많은 컨트롤 패킷들을 네트워크에 플러딩해야 한다. 이는 센서 노드들의 많은 배터리 소모를 야기시킨다. 이에 본 논문에서는 센서 노드가 자신의 위치를 알뿐만 아니라 변경하지 않는다는 사실을 이용하여 하나의 영구적인 그리드 구조를 만들어 네트워크에 플러딩 되는 컨트롤 패킷의 수를 줄인다. 이는 센서 노드들의 배터리 소모를 줄이고 전체 네트워크의 수명을 연장 시킨다. 제안한 라우팅 프로토콜의 성능평가를 위해서 두 계층 데이터 전송 라우팅 프로토콜과 비교 분석 하였다. 결과는 제안한 라우팅 프로토콜이 두 계층 데이터 전송 라우팅 프로토콜 비해서 더 에너지 효율적이라는 결과를 보여준다.

키워드 : 센서 네트워크, 에너지 효율적인 라우팅 프로토콜, 위치 기반의 라우팅 프로토콜

Abstract Sensor network consists of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. The life time of each node in the sensor network significantly affects the life time of whole sensor network. A node which drained out its battery may incur the partition of whole network in some network topology. The life time of each node depends on the battery capacity of each node. Therefore if all sensor nodes in the network live evenly long, the life time of the network will be longer.

In this paper, we propose Cluster-Based Power-Efficient Routing (CBPER) Protocol which provides scalable and efficient data delivery to multiple mobile sinks. Previous routing protocols, such as Directed Diffusion and TTDD, need to flood many control packets to support multiple mobile sinks and many sources, causing nodes to consume their battery. In CBPER, we use the fact that sensor nodes are stationary and location-aware to construct and maintain the permanent grid structure, which makes nodes live longer by reducing the number of the flooding control packets. We have evaluated CBPER performance with TTDD. Our results show that CBPER is more power-efficient routing protocol than TTDD.

Key words : Sensor Network, Power-Efficient Routing Protocol, Location-Aware Routing Protocol

* 본 연구는 첨단정보기술 연구센터를 통하여 과학기술부/한국과학재단의 지원을 받았고 정보통신부 및 정보통신연구원진흥원의 대학 IT연구센터 육성, 지원사업의 연구결과로 수행되었음

[†] 비회원 : 한국과학기술원 전산학과

kskweon@camars.kaist.ac.kr

shlee@camars.kaist.ac.kr

^{**} 종신회원 : 한국과학기술원 전산학과 교수

hyoon@camars.kaist.ac.kr

논문접수 : 2004년 5월 13일

심사완료 : 2005년 10월 26일

1. 서론

최근 무선 통신 기술과 전자 장비의 발달로 인하여 다양한 센서기능을 가진 센서 노드(sensor node)의 개발이 가능하게 되었다. 센서 노드는 다양한 센싱 작업(sensing task)을 수행하기 위한 센싱 장치(sensing unit), 센싱 되어진 데이터를 처리하기 위한 처리 장치(processing unit), 센서 노드들 간의 단거리 무선 통신을 위한 무선 송수신 장치(transceiver unit), 센서 노드의 전력 공급을 담당하는 전력 장치(power unit) 등으로 구성되어있다[1]. 센서 노드를 설계 할 때는 몇 가지 고려해야 할 사항이 있다. 먼저 센서 노드는 저전력을 소모하도록 설계되어야 한다. 일반적으로 센서 노드는 한번 전개(deployment)된 후에는 다시 전력 공급을 받지 못 한다. 그렇기 때문에 센서 노드의 배터리 파워를 효과적으로 사용하여 최대한 센서 노드의 수명을 길게 해야 한다. 두 번째로 고려할 사항은 센서 노드의 크기이다. 센서 노드는 다양한 응용(application) 분야에 사용된다. 응용 분야에 따라서 센서 노드의 크기가 제약 받는 경우가 있다. 예를 들어 조류의 생태계를 관찰 하려고 할 때 센서 노드는 관찰 하려는 조류에게 직접 부착 되어야 한다. 이런 경우에는 센서 노드의 크기가 커서는 안 된다. 이처럼 센서 노드를 설계 할 때는 센서 노드가 사용되는 응용분야에 따라 센서 노드의 크기도 고려해야 한다. 마지막으로 센서 노드는 낮은 가격으로 제작되어야 한다. 센서 네트워크는 일반적으로 수 천에서 수 만개의 센서 노드들로 구성되어있다. 이처럼 많은 수의 센서 노드들을 전개 해야 하기 때문에 센서 노드의 제작 가격이 비싸면 안 된다.

센서 네트워크(sensor network)는 어떤 현상(phenomenon)을 감지하기 위해서 관찰 지역 내에 뿌려진 센서 노드들로 구성된다. 그림 1.1은 센서 필드에 뿌려진 센서 노드들을 보여주고 있다. 센서 필드에서 감지된 정보는 싱크 노드로 전송되고 싱크 노드는 인터넷이나 인공위성 등을 통해서 사용자에게 정보를 전송 한다[1]. 센서 노드들은 현상을 감지하고 데이터를 생성하여 사용자(user)에게 전송하기 위해서 무선 송수신 장치를 이용한다. 하지만 무선 송수신 장치의 송신 거리(transmission range)가 무척 짧기 때문에 센서 노드에서 멀리 떨어져 있는 사용자까지 데이터를 직접 전송하는 것은 불가능하다. 센서 네트워크를 구성하는 센서 노드들은 멀리 떨어져 있는 사용자에게 데이터를 전송하기 위해서 서로 협력해야 한다. 센서 노드들은 현상을 감지하는 센싱 작업뿐만 아니라 다른 센서 노드의 데이터를 사용자까지 재전송 해주는 라우터(router) 역할까지 한다. 즉 센서 노드들은 멀티 홉(multi-hop) 데이터 전송

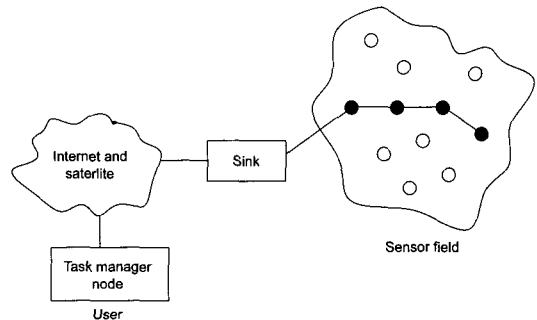


그림 1 센서 필드에 형성된 센서 네트워크

을 통해서 송신 거리보다 더 멀리 떨어져 있는 사용자에게 데이터를 전송할 수 있다.

센서 네트워크에는 싱크(sink)라는 특별한 노드가 존재한다. 싱크 노드의 역할은 센서 네트워크에서 생성된 데이터를 수집하는 것이다. 센서 네트워크에서 센서 노드들은 데이터를 목적지 노드(destination node)까지 재전송 할 뿐 직접 데이터를 요구 하지는 않는다. 즉 센서 네트워크에서 목적지 노드는 싱크 노드 뿐이다. 싱크 노드는 데이터를 요구하는 사용자 자신이 될 수도 있고 아니면 데이터를 수집하여 인터넷이나 인공위성과 같은 외부 네트워크를 통해서 사용자에게 데이터를 전송 해주는 게이트웨이(gateway) 노드일 수도 있다. 일반적으로 싱크 노드는 센서 노드 보다 좀 더 강력한 컴퓨팅(computing) 능력과 더 많은 메모리를 가지고 있다.

현재 센서 네트워크는 군대, 가정, 공장 등 다양한 분야에서 널리 쓰이고 있다. 특히 센서 네트워크는 유독한 화학 물질이 감염된 지역이나 전장과 같이 사람들이 직접 현상을 관찰 할 수 없는 위험한 곳에 설치하여 데이터를 수집 할 수 있다. 센서 네트워크의 예로서 침입 탐지 시스템이 있다. 침입 탐지 시스템이 설치되어 있는 지역에 침입이 발생하면 처음 침입을 감지한 센서 노드는 데이터를 생성하게 된다. 싱크 노드는 센서 네트워크로부터 어느 곳에서 침입이 발생 했는지에 대한 데이터를 수집한다. 소스 노드가 생성한 데이터는 침입이 어느 곳에서 발생 했는지에 대한 정보를 포함해야 한다. 즉 센서 노드는 자신의 위치 정보(location information)를 가지고 있어야 한다. 이러한 위치 정보는 GPS(Global Positioning System)을 센서 노드에 부착 하거나 사용자가 직접 센서 노드에 입력해서 알 수가 있다. 이러한 위치 정보를 라우팅 프로토콜(routing protocol)에 이용하면 좀 더 나은 성능을 가진 라우팅 프로토콜을 기대할 수 있다.

센서 네트워크에서는 센서 노드들이 협력을 해서 데이터를 싱크 노드까지 전송하기 때문에 각 센서 노드의

수명이 센서 네트워크의 통신망 자체의 수명을 결정 짓는 중요한 요소가 된다. 통신망의 구조(topology)에 따라서 어떠한 경우에는 단지 하나의 센서 노드가 자신의 배터리를 모두 소모하고 동작을 멈추었을 경우 자칫 통신망의 분할(partition)을 가져 올 수도 있다. PDA나 노트북과 같은 일반 무선 기기(wireless device)들은 배터리 전원이 방전 되었을 시에 사용자들이 각각의 장비들의 배터리를 재충전시켜 사용 할 수 있지만, 센서 노드들은 사용자에게서 독립적으로 무수한 노드들이 이곳 저곳에 배치되기 때문에 한번 배터리가 방전된 센서 노드를 재충전하는 것은 불가능하다. 이처럼 센서 노드의 수명은 각 센서 노드가 가지고 있는 배터리에 의존하므로 될 수 있는 한 소모하는 에너지량을 줄여 모든 센서 노드가 골고루 오래 살 수 있도록 하는 것이 전체 통신망의 수명을 늘릴 수 있는 방안이 된다.

본 논문에서 고려하는 센서 네트워크에서는 여러 개의 이동성(mobility)을 가진 싱크 노드들과 위치가 고정되어 있는 센서 노드들이 존재한다. 센서 노드는 비와 바람 또는 지형의 변화에 따라서 위치가 변화 될 수 있다. 하지만 이 논문에서는 위와 같은 자연적인 변화에 의해서 센서 노드의 위치 변화가 일어나지 않는 것을 가정한다. 각 센서 노드들은 GPS나 수동적인 위치 정보 입력을 통해서 자신의 위치를 알고 있으며 또한 자신의 이용 가능한 에너지 양(Power Availability)도 감지하고 있다. 센서 노드들이 감지해야 할 관심 사건(interest event)들은 센서 필드 여러 곳에서 동시에 다발적으로 발생 될 수도 있다.

현재 센서 네트워크를 위한 다양한 종류의 라우팅 프로토콜들이 제안되어 있다. 이러한 프로토콜에는 Directed Diffusion[2], Declarative Routing Protocol, GRAB 등이 있다. 하지만 이러한 라우팅 프로토콜들은 이동성 싱크 노드를 가진 큰 규모의 센서 네트워크에서 그리 좋은 성능을 나타내지 못 하고 있다. 위의 프로토콜들에서는 싱크 노드가 모든 센서 노드들에게 계속적으로 자신의 위치 정보를 전송 해야 하고 센서 노드들은 싱크 노드가 보낸 위치 정보를 가지고 후에 자신이 싱크 노드에게 보낼 데이터가 있을 때 데이터 전송 방향을 결정 할 수 있다. 그러나 여러 개의 싱크 노드로부터 전송되는 잦은 위치 정보 업데이트는 센서 노드들의 배터리를 소모 할 뿐만 아니라 무선 통신 채널의 많은 대역폭(bandwidth)을 소비한다. 이러한 싱크 노드 지향 데이터 전송 방식(Sink-Oriented Data Dissemination approach : SODD)은 모바일(mobile) 싱크 노드를 가진 큰 규모의 센서 네트워크를 위한 라우팅 프로토콜로는 적합하지 않다.

모바일 싱크 노드 문제점을 해결하기 위해서 두 계층

데이터 전송 방식(Two-Tier Data Dissemination approach : TTDD)이 제안 되었다[3]. 이 방식에서는 모든 센서 노드들이 싱크 노드에게 데이터를 전송하기 위한 데이터 회송 정보(data forwarding state)를 가지는 것이 아니라 그리드(grid) 교차점에 위치한 센서 노드들만 데이터 회송 정보를 갖는다. 센서 노드가 관심 현상을 감지하게 되면 데이터를 싱크 노드에게 전송하기 위해서 센서 필드에 그리드 구조를 만든다. 그리드의 각 교차점에 가장 가까이 위치한 노드들은 전송 노드(dissemination node)가 되어 데이터 회송 정보를 가지게 된다. 싱크 노드가 데이터를 전송 받기 위해서는 가장 가까이에 있는 전송 노드에게 데이터 요구 쿼리 패킷(query packet)을 전송하면 된다. 싱크 노드는 가장 가까이에 있는 전송 노드를 찾기 위해서 하나의 셀(cell) 크기 정도의 지역에 플러딩(flooding)을 하게 된다. 데이터 요구 쿼리 패킷을 받은 전송 노드는 자신이 가지고 있는 데이터 회송 정보를 이용하여 이 쿼리 패킷을 소스 노드까지 전송하게 된다. 결국 데이터는 쿼리 패킷이 전송 노드에서 소스 노드까지 간 패스(path)를 역으로 하여 싱크 노드까지 도달하게 된다. 이 프로토콜에서는 센서 노드가 현상을 발견 할 때마다 그리드 구조를 만들게 된다. 그리드 구조를 만들는데 사용되는 제어 패킷(control packet)의 수는 전체 센서 네트워크에 플러딩 하는 패킷의 수와 비슷하다. 만약 관심 사건이 여러 곳에서 자주 발생하게 된다면 이 프로토콜 또한 에너지 효율적인 라우팅 프로토콜로는 적합하지 않다.

본 연구에서는 그리드(grid)와 클러스터(cluster) 개념을 라우팅 프로토콜에 접목 시킨 무선 센서 네트워크를 위한 클러스터 기반의 에너지 효율적인 라우팅 프로토콜(Cluster-Based Power-Efficient Routing Protocol for Wireless Sensor Networks : CBPER)을 제안 하였다.

그림 2는 센서 필드를 그리드로 나눈 제안된 라우팅 프로토콜의 전체적인 모습을 보여주고 있다. 각 셀은 하나의 헤더 노드와 셀 아이디를 가지고 있다. 제안된 라우팅 프로토콜은 SODD 방식에서 문제가 되었던 모바일 싱크 문제와 TTDD 방식에서 문제가 되었던 관심 사건 다중 발생 문제를 모두 해결 하였다. 제안된 프로토콜에서는 TTDD 방식과는 달리 하나의 그리드 구조만을 사용한다. 즉 발생하는 관심 사건의 수와 그리드의 개수와는 아무런 관계가 없다. 또한 모바일 싱크 문제를 해결하기 위해서 TTDD 방식과 비슷하게 센서 노드가 현상을 감지하면 능동적으로 데이터 회송 정보를 자신이 포함된 세로축에 있는 클러스터 헤더 노드들에게 저장 한다. 싱크 노드가 데이터를 전송 받기 원하면 자신이 포함되어 있는 클러스터의 헤더 노드에게 데이터 요

구 쿼리 패킷만 전송 하면 된다. 데이터 요구 쿼리 패킷을 전송 받은 클러스터 헤더 노드는 자신이 포함된 가로축에 있는 클러스터 헤더 노드들에게 데이터 요구 쿼리 패킷을 재전송 한다. 이 패킷을 데이터 회송 정보를 가지고 있는 클러스터 헤더 노드가 받으면 자신의 데이터 회송 정보를 이용하여 소스 노드에게 패킷을 전달하게 된다. 결국 데이터는 패킷이 전달된 패스의 역 방향으로 싱크 노드에게 전송된다. 위에서 살펴본 것처럼 제안된 프로토콜은 데이터 패킷과 제어 패킷의 수를 줄여서 센서 노드들의 에너지 사용을 줄였다. 또한 각 센서 노드의 가능한 배터리 과워를 고려해 클러스터 헤더 노드를 선택함으로써 어떤 하나의 특정 센서 노드가 배터리를 빨리 소모하는 것을 막고 주변의 다른 센서 노드들에게 일을 분산시킴으로써 전체 통신망의 수명을 연장 시켰다.

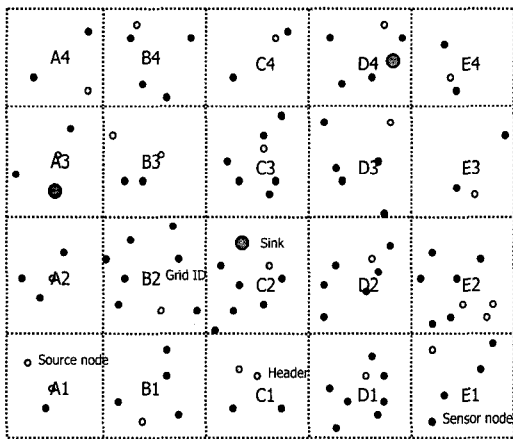


그림 2 클러스터 기반의 에너지 효율적인 라우팅 프로토콜

본 문은 구성은 다음과 같다. 2장에서는 센서 네트워크에서의 네트워크 계층(network layer)과 두 계층 데이터 전송 방식에 대하여 설명한다. 3장에서는 본 논문에서 제안하고 있는 클러스터 기반의 에너지 효율적인 라우팅 프로토콜에 대하여 설명한다. 4장에서는 기존의 라우팅 프로토콜과 본 논문에서 제안한 라우팅 프로토콜을 통신 비용(communication overhead) 측면에서 분석하여 비교 한다. 5장에서는 시뮬레이션 환경과 본 논문에서 제안한 프로토콜에 대한 성능 평가 결과를 기술 한다. 마지막으로 6장에서 결론을 내린다.

2. 연구 배경

본 장에서는 본 논문에서 제안하는 클러스터 기반의 에너지 효율적인 라우팅 프로토콜을 이해하기 위해 필

요한 배경 지식인 센서 네트워크에서의 네트워크계층(network layer)과 두 계층 데이터 전송 방식에 대하여 소개한다. 또한 본 논문에서 제안하는 프로토콜을 위한 용어 설명을 한다.

2.1 센서 네트워크에서의 네트워크계층(Network Layer)

네트워크계층은 전송계층(transport layer)의 세그먼트(segment)를 호스트(host)에서 다른 호스트로 이동시키는 호스트간의 통신 서비스를 제공한다. 송신 호스트에서 전송계층의 세그먼트가 네트워크계층으로 전달 되면 이 세그먼트를 목적지 전송계층의 프로토콜 스택(protocol stack)까지 전달하는 것이 네트워크계층의 몫이다. 송신자에서 수신자까지 패킷을 전달할 때 네트워크계층은 패킷의 경로를 결정해야 한다. 이러한 경로를 계산하는 알고리즘을 라우팅 알고리즘이라고 한다. 라우팅 알고리즘의 목적은 간단하다 : 주어진 라우터 집합에서 라우터를 연결하는 링크들과 근원지에서 목적지까지 좋은 경로를 찾는다. 일반적으로 좋은 경로란 최소 경비를 갖는 것이다. 경비는 라우팅 프로토콜이 쓰이는 네트워크에 따라서 다르다. 일반적인 네트워크에서는 경로의 거리가 가장 짧은 경로나 혼잡이 적은 경로를 최소 경비를 갖는 경로라고 한다. 하지만 센서 네트워크에서는 에너지의 소모가 가장 적은 경로를 최소 경비를 갖는 경로라고 한다.

위에서 살펴 본 것처럼 센서 네트워크는 일반 네트워크에 비해 약간 특별한 라우팅 프로토콜을 필요로 한다. 이는 센서 네트워크가 일반 무선 네트워크에 비해서 많은 제약 사항을 가지고 있기 때문이다. 그렇기 때문에 센서 네트워크를 위한 라우팅 프로토콜을 제안 할 때는 몇 가지 고려해야 할 사항이 있다. 첫 번째로, 센서 네트워크는 에너지를 효율적으로 사용해야 한다. 앞에서 설명 했듯이 센서 노드는 에너지에 크게 제약을 받는다. 그렇기 때문에 패킷의 경로를 설정할 때 패킷의 전송 시간이 지연되더라도 에너지가 적게 드는 경로를 설정해야 한다. 두 번째로 센서 네트워크에서는 가능하면 많은 데이터 병합(data aggregation)을 해야 한다. 이 기법은 같은 현상을 감지하고 데이터를 보내는 여러 센서 노드들의 데이터 패킷을 하나의 데이터 패킷으로 병합함으로써 네트워크내의 패킷의 수를 줄일 수 있다. 세 번째로 센서 네트워크는 데이터 중심적이다. 데이터 중심적인 라우팅에는 두 가지 접근 방식이 있다. 첫 번째 방식은 싱크 노드가 데이터의 요구를 센서 네트워크에 브로드캐스트(broadcast)하는 방식이고 또 다른 하나는 데이터를 가지고 있는 센서 노드가 자신의 데이터를 광고(advertisement)하는 방식이다. 네 번째로 센서 네트워크에서는 특성 기반의 어드레싱(addressing)을 한다. 즉 싱크 노드는 현상의 특성을 이용하여 쿼리(query)를

실행한다. 예를 들어 “침입이 발생된 지역이 어디인가?”와 같은 쿼리를 실행 할 수 있다.

2.1.1 에너지 효율적인 경로

에너지 효율적인 경로를 결정하는 요소는 크게 두 가지가 있다. 첫 번째 요소는 각 센서 노드에 남아있는 배터리 양이다. 센서 네트워크는 모든 센서 노드들의 협력을 통해서 통신망이 이루어 진다. 그렇기 때문에 모든 센서 노드가 공평하게 오래 같이 사는 것이 중요하다. 각 센서 노드의 가용 에너지(Power Availability: PA)를 고려해서 경로를 설정할 때는 센서 노드의 PA가 큰 노드들 중심으로 경로를 설정해야 한다. 두 번째 고려해야 할 요소는 두 센서 노드 사이에 패킷을 전송 할 때 소모되는 전송 에너지(transmission power : α)이다. 이 전송 에너지는 두 노드 사이의 거리나 기타 다른 요소들에 의해서 결정된다. 그러나 일반적으로 전송 에너지는 전송 거리에 비례한다. 에너지 효율적인 경로를 설정하기 위해서는 전체 전송 에너지도 고려해야 한다.

그림 3은 근원지 노드(S)에서 싱크 노드까지의 다양한 경로들을 보여주고 있다. 에너지 효율적인 경로는 다음의 접근 방법 중 하나의 의해서 선택 된다[1].

- 최대 PA 경로(Maximum PA route) - 이 접근 방법은 경로에 포함된 노드들의 전체 PA가 최대가 되는 경로를 선택한다. 그림 3에서는 경로 3이 전체 PA 6로써 최대 PA를 갖는다.
- 최소 에너지 경로(Minimum energy route) - 이 접근 방법은 전체 전송 에너지를 최소로 소모하는 경로를 선택한다. 그림 3에서는 전체 전송 에너지 3을 소모하는 경로 1이 선택 된다.
- 최소 홉 경로(Minimum hop route) - 이 접근 방법은 모든 경로 중에서 최소의 홉을 가진 경로를 선택

한다. 만약 센서 노드 사이의 전송 에너지가 모두 같다면 최소 홉 경로와 최소 에너지 경로는 같은 결과를 보여 줄 것이다. 그림 3에서는 홉 수가 2인 경로 2가 선택 된다.

- 최대 최소 PA 노드 경로(Maximum minimum PA node route) - 이 접근 방법은 각 경로에 속한 최소 PA 노드들 중에서 그 값이 최대가 되는 노드를 포함하는 경로를 선택한다. 즉 경로 1에서의 최소 PA 노드의 PA 값은 2, 경로 2에서는 3, 경로 3에서는 1이다. 그러므로 최소 PA 노드를 최대 PA 값을 가진 경로 2가 선택 된다.

2.2 두 계층 데이터 전송 라우팅(Two-Tier Data Dissemination Routing)

두 계층 데이터 전송(TTDD) 라우팅 프로토콜[3]은 관심 사건(interest event)이 발생 했을 때 싱크 노드에 데이터를 전송하기 위해서 소스 노드(source node)가 능동적으로 그리드를 만든다. 그리드의 각 교차 점에는 회송 정보(forwarding information)을 가지고 있는 보급 노드(dissemination node)가 존재한다. 두 계층 데이터 전송 라우팅은 데이터를 전송하기 위해서 두 개의 전송 계층을 사용한다. 하위 계층(low-tier) 전송은 한 셀 내에서의 지역적인 통신을 위해서 사용되고 상위 계층(high-tier) 전송은 셀(cell)과 셀 사이의 통신을 위해서 사용된다.

싱크 노드는 데이터를 전송 받기 위해서 데이터 요구 패킷(data request packet)을 셀 내에 플러딩(flooding)한다. 데이터 요구 패킷을 받은 보급 노드는 이 패킷을 자신의 상류 보급 노드(upstream dissemination node)에게 재전송 하게 된다. 이런 방식으로 재전송된 데이터 요구 패킷은 결국 소스 노드까지 전송 된다. 본 논문에서는 소스 노드로의 방향을 상류(upstream)로 정의한다. 소스 노드는 데이터 요구 패킷이 전송된 경로의 역 방향으로 데이터를 전송 한다. 두 계층 데이터 전송 라우팅 프로토콜은 그리드를 만드는 것(grid construction)과 데이터 패킷과 쿼리 패킷의 두 계층 회송(two-tier query and data forwarding)으로 이루어져 있다.

2.2.1 그리드 제작(Grid Construction)

관심 사건이 발생 했을 때 소스 노드는 능동적으로 그리드를 생성하게 된다. 그리드를 생성하기 위해서 소스 노드는 데이터 공고(data announcement) 패킷을 그리드의 각 교차점(dissemination point)에 보낸다. 소스 노드의 위치가 $L_s = (x, y)$ 라고 했을 때 각 교차점의 위치는 $L_p = (x_i, y_j) \{xi = x + i \cdot \alpha, yj = y + j \cdot \alpha; i, j = \pm 0, \pm 1, \pm 2 \dots\}$ (α 는 셀 한 변의 길이)이다. 소스 노드가 지리적 회송(geographical forwarding)

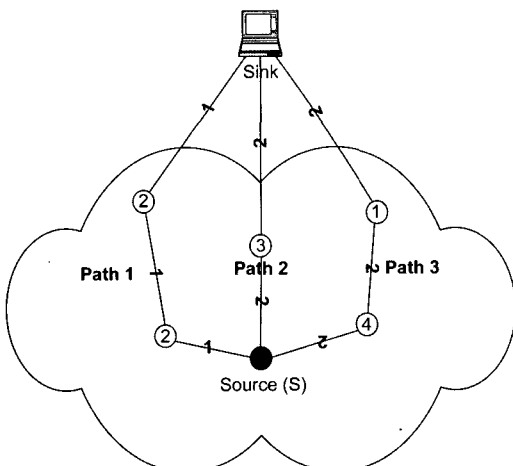


그림 3 다양한 경로들의 에너지 효율도

을 통해서 각 교차점으로 데이터 광고 패킷을 전송하면 각 교차점에 가장 가까이에 있는 노드가 보급 노드(dissemination node)가 된다. 보급 노드들은 데이터 광고 패킷과 자신의 교차점의 위치 그리고 자신에게 데이터 광고 패킷을 전송해준 상류 보급 노드(upstream dissemination node)의 위치를 저장한다.

그림 4는 소스 노드 B가 생성한 그리드를 보여준다. 그리드를 구성하는 셀의 한 변의 길이는 a 가 되고 각 교차점에는 보급 노드들이 존재한다. 싱크 노드에서 소스 노드로의 방향이 Upstream으로 역방향인 Downstream으로 정의된다.

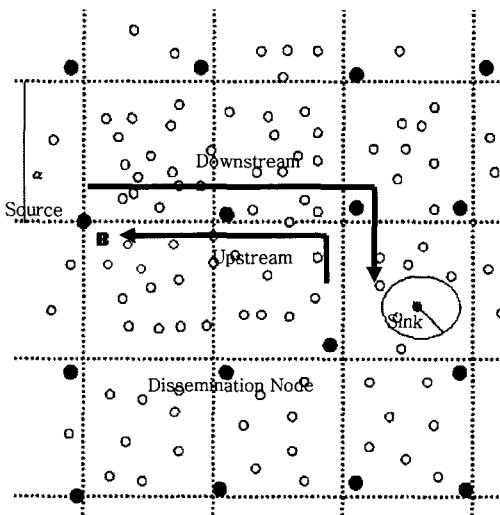


그림 4 소스 노드 B가 생성한 그리드

2.2.2 데이터와 쿼리의 두 계층 회송(two-tier query and data forwarding)

싱크 노드는 데이터가 필요 할 때 주변에 있는 지역 보급 노드(immediate dissemination node)를 찾기 위해서 지역적 플러딩을 한다. 쿼리 패킷을 받은 지역 보급 노드는 상류 보급 노드에게 쿼리 패킷을 전송한다. 상류 보급 노드는 쿼리 패킷을 전송 해준 하류 보급 노드(downstream dissemination node)의 위치를 저장한다. 이런 과정을 통해서 데이터 요구 쿼리 패킷은 소스 노드에게 전달 된다. 소스 노드는 데이터를 자신의 하류 보급 노드에게 전송 한다. 데이터를 받은 하류 보급 노드는 또 다시 자신의 하류 보급 노드에게 데이터를 전송 한다. 이런 과정을 통해서 데이터는 싱크 노드가 머무르고 있는 셀의 보급 노드까지 오게 된다.

싱크 노드는 두 개의 에이전트 노드를 가지고 있다: 프라이머리 에이전트 노드(primary agent node)와 이미디어트 에이전트 노드(immediate agent node). 싱크 노

드는 자신의 한 홉에 있는 노드들 중에서 프라이머리 에이전트 노드를 선택 한다. 초기엔 프라이머리 에이전트 노드와 이미디어트 에이전트 노드는 같다. 싱크 노드는 데이터 요구 쿼리를 지역 보급 노드에게 전송 할 때 프라이머리 에이전트 노드의 위치도 함께 보낸다. 그러므로 싱크 노드의 위치는 프라이머리 에이전트 노드에 의해서 지역 보급 노드에게 투명하게 된다. 싱크 노드가 셀 내에서 이동을 하여 프라이머리 에이전트 노드와 더 이상 한 홉 통신을 할 수 없게 되면 자신의 한 홉 이웃(neighbor)들 중에서 새로운 이미디어트 에이전트 노드를 선택하게 된다. 그리고 새로 선택한 이미디어트 에이전트 노드의 위치를 프라이머리 에이전트에게 보내게 된다.

데이터가 지역 보급 노드에 도착하게 되면 지역 보급 노드는 싱크 노드의 프라이머리 에이전트 노드에게 데이터를 전송한다. 만약 싱크 노드가 새로운 이미디어트 에이전트 노드를 선택하지 않았다면, 즉 프라이머리 에이전트 노드와 이미디어트 에이전트 노드가 같다면 프라이머리 에이전트 노드는 데이터를 싱크 노드에게 바로 전송한다. 만약 싱크 노드가 새로운 이미디어트 에이전트 노드를 가지고 있다면 프라이머리 에이전트 노드는 데이터를 이미디어트 에이전트 노드에게 전송하고 결국 싱크 노드에게까지 전송 된다.

2.3 용어 설명

제안한 라우팅 프로토콜을 위한 용어 설명을 한다.

싱크 노드(sink node)

싱크 노드는 센서 네트워크에서 생성된 데이터를 수집하는 이동성을 가진 노드이다. 싱크 노드는 데이터를 요구하는 사용자 자신이 될 수도 있고 아니면 데이터를 수집하여 인터넷이나 인공위성과 같은 외부 네트워크를 통해서 사용자에게 데이터를 전송 해주는 게이트웨이(gateway) 노드 일 수도 있다. 일반적으로 싱크 노드는 센서 노드에 비해서 좀 더 긴 통신 거리를 가지고 있다. 본 논문에서는 여러 개의 이동성 싱크 노드를 가정한다.

헤더 노드(header node)

헤더 노드는 셀(cell) 내에서 상위 계층 통신(high-layered communication)을 담당한다. 헤더 노드는 회송 정보를 가지고 있으며 셀 내에서 발생되는 데이터를 병합 한다. 가장 많은 에너지를 소모하는 노드이며 셀 내의 센서 노드들이 돌아 가면서 담당한다.

소스 노드(source node)

소스 노드는 관심 이벤트를 감지하고 데이터를 생성하는 센서 노드이다.

그리드(grid)

그리드는 절대 좌표를 이용해서 만들어 진다. 그리드의 각 셀들은 두 개의 숫자로 이루어진 고유 의 아이디

를 부여 받고 각 셀마다 하나의 클러스터(cluster)가 존재한다. 센서 노드와 싱크 노드는 자신의 위치 정보를 가지고 자신의 셀을 알 수가 있다.

두 계층 통신(two-layered communication)

제안하는 라우팅 프로토콜에서는 패킷을 전송 할 때 두 계층 통신을 한다. 셀 내에서 센서 노드나 싱크 노드가 헤더 노드에게 패킷을 전송 할 때는 하위 계층 통신(low-layered communication)을 이용하여 전송을 하고 셀 외부로의 통신을 할 때는 상위 계층 통신(high-layered communication)을 이용해서 전송을 한다. 하위 계층 통신에서는 각 센서 노드들이 헤더 노드의 위치를 알기 때문에 거리에 맞는 적당한 전송 파워를 사용하여 전송한다.

3. 클러스터 기반의 에너지 효율적인 라우팅

본 장에서는 클러스터 기반의 에너지 효율적인 라우팅(Cluster-Based Power-Efficient Routing) 프로토콜에 대하여 설명한다. 3.1에서는 본 논문에서 제안하는 프로토콜의 개요에 대해서 소개한다. 3.2에서는 제안되는 라우팅 프로토콜을 위한 가정과 요구 사항에 대하여 설명을 하고 3.3에서 제안하는 라우팅 프로토콜의 동작에 대하여 설명한다.

3.1 클러스터 기반의 에너지 효율적인 라우팅 프로토콜의 개요

클러스터 기반의 에너지 효율적인 라우팅 프로토콜(CBPER)은 라우팅을 위해 전송되는 패킷의 수를 줄여 센서 노드들의 에너지를 절약한다. 이 프로토콜은 그리드(grid)와 클러스터(cluster) 개념을 사용함으로써 특정 센서 노드에게만 회송 정보(forwarding information)을 전송한다.

두 계층 데이터 전송 라우팅 프로토콜은 사건이 발생할 때마다 동적으로 그리드를 생성해야 한다. 즉 여러 개의 사건이 동시 다발적으로 발생하면 그 수만큼의 그리드 생성이 필요하다. 센서 노드가 그리드를 생성하고 유지하기 위해서는 컨트롤 패킷을 센서 네트워크에 플러딩해야 한다. 센서 네트워크는 센서 노드의 개수가 굉장히 많기 때문에 플러딩 할 때 많은 양의 패킷이 발생한다. 제안된 라우팅 프로토콜은 여러 개의 사건이 발생해도 미리 만들어진 하나의 그리드만 사용한다. 또한 그리드를 유지하기 위해서 제어 패킷을 전송 할 필요도 없다.

제안된 라우팅 프로토콜은 3개의 기본적인 동작을 한다. 처음 관심 사건이 발생하면 그 사건을 제일 처음 감지한 센서 노드는 소스 노드가 되어 데이터를 생성한다. 데이터의 발생을 센서 네트워크에 알리기 위해서 데이터 공고(data announcement)를 그리드의 세로축으로

한다. 싱크 노드가 데이터가 필요하면 데이터 요구(data request)를 그리드의 가로축으로 한다. 데이터 공고 패킷을 가진 헤더 노드가 데이터 요구를 받게 되면 자신의 캐쉬에 있는 데이터 공고 패킷을 보고 소스 노드에게 데이터 요구를 한다. 소스 노드는 데이터 패킷을 생성하여 싱크 노드에게 전송하게 된다.

3.2 제안된 라우팅 프로토콜을 위한 가정과 요구 사항

본 논문에서 제안한 라우팅 프로토콜은 다음의 가정과 요구 사항을 갖는다.

각 센서 노드와 싱크 노드는 자신의 위치를 알아야 한다.

제안된 라우팅 프로토콜은 각 센서 노드와 싱크 노드가 GPS(Global Positioning System)나 혹은 사용자의 수동적인 위치 입력을 통해서 자신의 위치를 안다고 가정한다. 특히 위치 정보는 센서 네트워크에서 데이터를 생성하는데 필요하다. 예를 들어 침입 탐지를 위한 센서 네트워크를 가정해 보자. 침입 탐지 시스템을 설치한 지역에 침입이 발생 했을 경우 침입을 탐지한 센서 노드는 데이터를 생성한다. 이때 이 데이터에서 가장 중요한 것은 '침입이 어디서 일어 났는가?' 이다. 이처럼 센서 노드의 위치 정보는 라우팅 프로토콜을 위해서 뿐만 아니라 생성되는 데이터를 위해서도 필요하다.

각 센서 노드들은 이동성이 없다.

제안된 라우팅 프로토콜은 각 센서 노드들이 자신의 위치를 바꾸지 않는 것을 가정 한다. 실제로 센서 노드가 센스 필드에 전개(deployment)된 후에는 자연적인 현상에 의해서 센서 노드의 위치가 변 할 수 있다. 하지만 일반적으로 센서 노드가 자체적인 이동 장치를 가지는 경우는 거의 없다. 센서 노드가 장마나 지진과 같은 자연적인 현상에 의해서 자신의 위치가 변경이 된 경우라도 자신의 위치 정보를 GPS 와 같은 장비를 통해서 다시 얻게 된다면 제안된 라우팅 프로토콜을 이용할 수 있다. 하지만 이 논문에서는 센서 노드들의 위치 이동은 없는 것으로 가정한다.

각 센서 노드들은 자신의 배터리 양을 알고 있다.

제안된 라우팅 프로토콜은 각 센서 노드들이 자신의 남아 있는 배터리 양을 알고 있는 것을 가정한다. 센서 네트워크는 센서 노드들의 협동(collaboration)을 통해서 통신 망을 이룬다. 그렇기 때문에 모든 센서 노드들이 다 같이 오래 사는 것이 중요하다. 제안된 프로토콜은 각 센서 노드들이 자신의 남아 있는 배터리 양을 감지하고 얼마나 오래 헤더 노드 역할을 할 것인지 결정 한다.

여러 개의 이동성 싱크 노드가 존재한다.

제안된 라우팅 프로토콜은 여러 개의 싱크 노드가 센서 필드내에서 이동하는 것을 가정한다. 싱크 노드는 센서 필드내를 이동하면서 주기적으로 데이터 요구 패킷

을 전송하게 된다.

3.3 클러스터 기반의 에너지 효율적인 라우팅 프로토콜의 동작

3.3.1 헤더 노드의 선출(Selection of Header node)

센서 필드에 센서 노드들이 전개되면 각 센서 노드는 자신의 위치 정보를 가지고 자신의 셀을 결정한다. 각 셀은 두 개의 정수로 된 고유 아이디를 가지고 있으며 셀 내의 센서 노드들은 자신의 셀 아이디를 기억하고 있다. 최초의 헤더 노드를 선출 하기 위해서 센서 노드들은 전개 된 후 램덤(random) 시간 동안 헤더 공고 패킷(header announcement packet)를 지역적으로 플러딩(flooding)한다. 지역적 플러딩은 각 센서 노드가 패킷을 전송 받을 경우 패킷 내의 셀 아이디가 자신의 셀 아이디와 다를 경우 패킷 드랍(drop)을 하면서 이루어진다. 헤더 공고 패킷에는 패킷이 전송된 시간과 헤더 지속 시간이 있어서 가장 먼저 패킷을 전송한 센서 노드가 헤더 지속 시간 동안 헤더 노드 역할을 한다. 헤더 지속 시간은 자신의 남아 있는 배터리 파워에 따라서 달라진다. 각 센서 노드들은 초기에 헤더 공고 패킷을 플러딩하면서 자신의 한 홉 이웃 노드(neighbor node)들의 위치와 헤더 노드의 위치를 저장한다.

헤더 노드는 자신의 헤더 지속 시간이 경과하면 다시 센서 노드로 되돌아 간다. 그렇기 때문에 현재 헤더 노드의 헤더 지속 시간이 끝나기 전 다음 헤더 노드를 선출 해야 한다. 셀 내의 현 헤더 노드를 제외한 모든 노드들은 자신의 배터리 파워 값을 측정한다. 노드들 중 배터리 파워 값이 일정 값 이상인 노드들은 헤더 지속 시간을 계산을 하여 헤더 공고 패킷을 만들어 현재 헤더 노드의 헤더 지속 시간이 끝나기 전 지역적으로 플러딩 한다. 헤더 노드의 선출은 최초 헤더 노드를 선출 하는 방식과 같다. 위와 같은 방식으로 헤더 노드는 배터리 파워 값을 토대로 하여 동적으로 선출된다.

```

Header_Selection
{
  if (HeaderTime_Expired - a) {
    if (BatteryPower > PowerThreshold) {
      HeaderAnnouncementPacket.Duration = BatteryPower / β
      HeaderAnnouncementPacket.Time = CurrentTime
      HeaderAnnouncementPacket.CellID = CellID
      broadcast (HeaderAnnouncementPacket)
    }
    while (a)
      Header = min (HeaderAnnouncementPacket.Time)
  }
}
    
```

Pseudo code 1. Header Selection

3.3.2 데이터 공고(Data Announcement)

센서 필드에서 관심 사건(interest event)이 발생하면 그 관심 사건을 제일 먼저 감지한 센서 노드는 소스 노드(source node)가 된다. 소스 노드는 데이터 공고 패킷(data announcement packet)을 생성하여 지리적 회송(geographical forwarding)을 통해서 헤더 노드까지 전송을 한다. 지리적 회송은 이웃 노드들 중에서 헤더 노드와의 거리가 가장 짧은 노드를 선택하여 패킷을 전송함으로써 이루어 진다. 이때 각 센서 노드는 전송거리에 맞는 전송 파워를 사용하여 패킷을 전송한다. 데이터 공고 패킷에는 소스 노드의 위치, 셀 아이디 그리고 데이터를 생성한 시간이 기록되어 있다. 만약 같은 관심 사건을 감지한 여러 개의 소스 노드가 데이터 공고 패킷을 헤더 노드에게 전송 한다면 헤더 노드는 이 패킷들을 병합한다.

헤더 노드가 데이터 공고 패킷을 전송 받으면 패킷을 자신의 캐쉬에 저장한다. 헤더 노드는 데이터를 센서 네트워크에 공고(announcement)하기 위해서 상위 계층 통신을 이용하여 자신의 위, 아래 셀에 있는 헤더 노드에게 데이터 공고 패킷을 전송한다. 데이터 공고 패킷을 전송 받은 위, 아래에 있는 헤더 노드는 자신의 캐쉬에 패킷을 저장한다. 위쪽에 위치한 헤더 노드는 데이터 공고 패킷을 계속해서 위쪽 셀에 있는 헤더 노드들에게 전송하고 아래쪽에 위치한 헤더 노드는 패킷을 아래쪽에 있는 헤더 노드에게 전송한다. 결국 소스 노드를 포함하고 있는 셀의 세로축에 있는 헤더 노드는 데이터 공고 패킷을 전부 자신의 캐쉬에 저장한다. 그림 5는 센

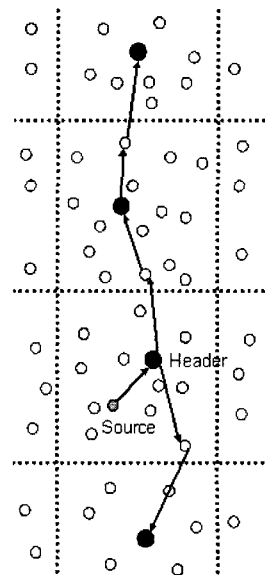


그림 5 센서 네트워크에 데이터 공고(Data Announcement)

서 네트워크에서 사건을 감지한 센서 노드가 소스 노드가 되어 데이터 광고 패킷을 만들어 헤더 노드에게 전달한 뒤 헤더 노드가 데이터 광고 패킷을 자신의 위쪽과 아래 쪽 셀에 있는 헤더 노드에게 회송 함으로써 그리드의 세로축으로 데이터 광고를 하는 것을 보여 준다.

```

Data_Announcement
{
  if (Event_Detected) {
    DataAnnouncementPacket.Location = Location
    DataAnnouncementPacket.CellID = CellID
    DataAnnouncementPacket.Time = CurrentTime
    lowLayerSend (Header, DataAnnouncementPacket)
  }
  if (Is_Header) {
    saveCache (DataAnnouncementPacket)
    if (LocalHeader || UpperHeader)
      highLayerSend (LowerHeader, DataAnnouncementPacket)
    if (LocalHeader || LowerHeader)
  }
}
    
```

Pseudo code 2. Data Announcement

3.3.3 데이터 요구(Data Request)

싱크 노드가 데이터가 필요할 때 자신의 한 홉 이웃 센서 노드에게 데이터 요구 패킷(data request packet)을 전송한다. 싱크 노드는 자신의 이웃 노드 중에서 이미디어트 에이전트 노드(immediate agent node)를 선택 한다. 이미디어트 에이전트 노드는 이웃 노드들 중에서 싱크 노드에 가장 가까운 노드로 선택 한다. 이는 싱크 노드가 자신의 위치를 옮기더라도 이미디어트 에이전트 노드와 계속 통신을 할 수 있는 확률을 높이기 위함이다. 만약 싱크 노드가 자신의 위치를 크게 옮겨 이미 선택한 이미디어트 에이전트 노드와 더 이상 통신을 할 수 없을 경우 싱크 노드는 자신의 이웃 센서 노드들 중에서 새로운 이미디어트 에이전트 노드를 선택 한다.

싱크 노드는 데이터 요구 패킷을 이미디어트 에이전트 노드에게 전송한다. 데이터 요구 패킷에는 이미디어트 에이전트 노드의 위치 정보와 셀 아이디를 포함하고 있다. 이미디어트 에이전트 노드는 지리적 회송을 통해서 헤더 노드에게 패킷을 전송 한다. 헤더 노드는 패킷을 자신의 캐쉬에 저장하고 상위 계층 통신을 통해서 자신의 셀 좌측, 우측에 있는 헤더 노드에게 데이터 요구 패킷을 전송한다. 우측에 있는 셀의 헤더 노드는 전송 받은 데이터 요구 패킷을 자신의 캐쉬에 저장하고 패킷을 다시 우측으로 전송한다. 좌측에 있는 셀의 헤더 노드도 패킷을 자신의 캐쉬에 저장하고 좌측으로 전송

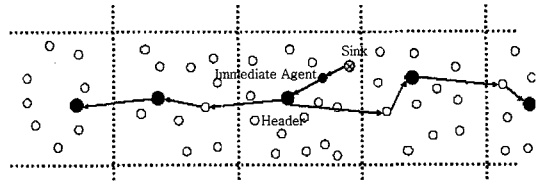


그림 6 싱크 노드의 데이터 요구(Data Request)

한다. 결국 싱크 노드가 있는 셀의 가로축에 있는 모든 헤더 노드는 데이터 요구 패킷을 자신의 캐쉬에 저장하게 된다. 그림 6은 데이터 요구 패킷을 그리드의 가로축으로 전송하는 것을 보여 준다. 싱크 노드가 데이터 요구 패킷을 자신의 이미디어트 에이전트 노드에게 보내고 이미디어트 에이전트 노드는 그 패킷을 다시 자신의 셀에 있는 헤더 노드에게 보낸다. 헤더 노드는 소스 노드에게 데이터 요구를 하기 위해서 자신의 왼편과 오른편에 있는 셀의 헤더 노드에게 패킷을 회송하면서 그리드의 가로축으로 데이터 요구를 하게 된다.

```

Data_Request
{
  : Sink
  if (Data_Needed) {
    DataRequestPacket.CellID = AgentCellID
    DataRequestPacket.Location = AgentLocation
    lowLayerSend (ImmediateAgent, DataRequestPacket)
  }
  : Node
  if (Is_Agent)
    lowLayerSend (Header, DataRequestPacket)
  if (Is_Header) {
    saveCache (DataRequestPacket)
    if (LocalHeader || RightHeader)
      highLayerSend (LeftHeader, DataRequestPacket)
    if (LocalHeader || LeftHeader)
      highLayerSend (RightHeader, DataRequestPacket)
  }
}
    
```

Pseudo code 3. Data Request

3.3.4 데이터 전송(Data Forwarding)

데이터 광고 패킷을 가지고 있는 헤더 노드가 데이터 요구 패킷을 전송 받으면 자신의 캐쉬에서 데이터 광고 패킷을 꺼낸다. 헤더 노드는 데이터 광고 패킷의 생성된 시간을 보고 데이터가 유효한지를 검사 한다. 만약 유효한 데이터 정보라면 헤더 노드는 데이터 광고 패킷에 있는 셀 아이디를 보고 상위 계층 통신을 통해서 소스

노드가 있는 셀의 헤더 노드에게 데이터 요구 패킷을 전송 한다.

소스 노드가 있는 셀의 헤더 노드가 데이터 요구 패킷을 전송 받으면 지리적 회송을 이용해서 소스 노드에게 데이터 요구 패킷을 전송 한다. 소스 노드는 데이터를 포함한 데이터 패킷(data packet)을 생성하여 헤더 노드에게 전송한다. 헤더 노드는 데이터 요구 패킷에 있는 셀 아이디를 보고 데이터 패킷을 싱크 노드가 있는 셀의 헤더 노드에게 상위 계층 통신을 통해서 전송한다.

데이터 패킷이 싱크 노드가 있는 셀의 헤더 노드에 전송 되면 헤더 노드는 데이터 패킷을 이미디어트 에이전트 노드에게 보낸다. 이미디어트 에이전트 노드는 한 홉에 있는 싱크 노드에게 데이터 패킷을 전송한다. 만약 싱크 노드가 자신의 위치를 옮겨서 새로운 이미디어트 에이전트 노드를 선택 했다면 그 위치를 예전 이미디어트 에이전트 노드에게 알려줘서 예전 이미디어트 에이전트 노드에게 온 데이터 패킷을 새로운 이미디어트 에이전트 노드에게 재전송 하게 한다. 그림 7은 센서 네트워크에서 소스 노드로부터 싱크 노드까지 데이터 패킷을 전송하는 것을 보여준다. 데이터 공고 패킷과 데이터 요구 패킷을 모두 받은 헤더 노드는 데이터 요구 패킷을 소스 노드가 있는 셀의 헤더 노드에게 회송한다. 헤더 노드는 데이터를 소스 노드에게 요구하고 소스 노드는 데이터 패킷을 만들어 헤더 노드에게 보낸다. 헤더 노드는 데이터 요구 패킷의 경로를 역방향으로 해서 데이터 패킷을 싱크 노드가 있는 헤더 노드에게 보내고 그 헤더 노드는 이미디어트 에이전트 노드를 통해서 싱크 노드에게 데이터 패킷을 보내게 된다.

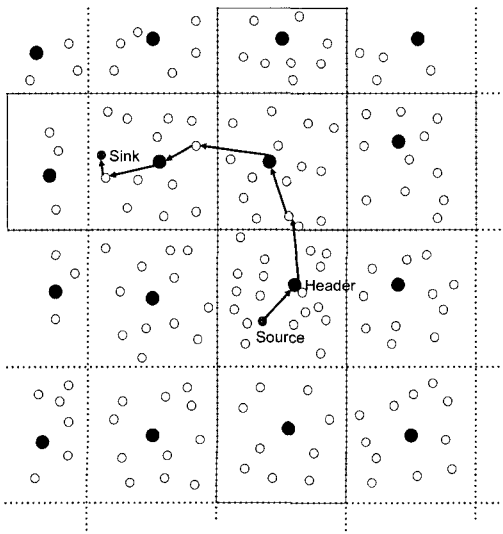


그림 7 센서 네트워크에서 데이터 전송

```

Data_Forwarding
{
  if (Is_Header & Cached_DataAnnouncementPacket) {
    retrieveCache (DataAnnouncementPacket)
    if (DataAnnouncementPacket.Time > AvailableTime)
      highLayerSend (DataAnnouncementPacket.CellID, DataRequestPacket)
  }
  if (Is_SourceNode) {
    DataPacket = Data
    lowLayerSend (Header, DataPacket)
  }
  if (Is_Agent) {
    if (AvailableAgent)
      lowLayerSend (Sink, DataPacket)
    else
      lowLayerSend (NewAgent, DataPacket)
  }
}
    
```

Pseudo code 4. Data Forwarding

4. 성능 분석

본 장에서는 본 논문에서 제안하는 클러스터 기반의 에너지 효율적인 라우팅 프로토콜의 통신 비용(communication overhead)을 분석한다. 분석을 위해서 분석 모델을 정의하고 제안하는 라우팅 프로토콜의 성능을 나타내기 위해서 기존 라우팅 프로토콜과 비교를 한다. 본 장에서는 분석을 용이하게 하기 위해서 데이터와 쿼리의 병합 그리고 각 라우팅 프로토콜을 유지하는데 들어가는 비용은 고려하지 않는다.

4.1 분석 모델

본 장에서는 본 논문에서 제안하는 라우팅 프로토콜(CBPER)과 기존의 라우팅 프로토콜인 두 계층 데이터 전송(TTDD)과 싱크 노드 지향 데이터 전송(SODD)을 비교 분석 한다. 분석을 위해서 먼저 분석 모델을 정의 한다. 분석 모델은 넓이가 A인 센서 필드에 N개의 센서 노드가 균등하게 뿌려져 있는 센서 네트워크이다. 센서 네트워크에는 k개의 유동성 싱크 노드가 존재한다. 이 싱크 노드들은 v의 속도로 T 시간 동안 m개의 셀을 지나며 d개의 데이터 패킷을 소스 노드로부터 받게 된다. 그러므로 싱크 노드가 한 셀에 있는 동안 받을 수 있는 데이터의 양은 $\frac{d}{m}$ 개 데이터 패킷이다. 데이터 패킷은 단위 크기이고 데이터 요구 쿼리 패킷(data request query packet)과 데이터 공고 패킷(data announcement packet)의 크기는 모두 1이다.

CBPER과 TTDD 라우팅 프로토콜에서는 센서 필드가 한 변의 길이가 α 인 셀(cell)로 나누어진다. 각 셀에는 $n = \left(\frac{N\alpha^2}{A}\right)$ 개의 센서 노드들이 존재하며 셀 한 변의

길이는 \sqrt{n} 으로 나타낼 수 있다. 그림 8은 CBPER과 TTDD 라우팅 프로토콜의 분석 모델을 보여준다.

이 분석 모델[3]에서는 플러딩(flooding)을 할 때 사용되는 패킷의 수는 전체 센서 노드의 수에 비례하고 지리적 회송(geographical forwarding)은 전송 경로에 포함된 센서 노드의 수에 비례 한다.

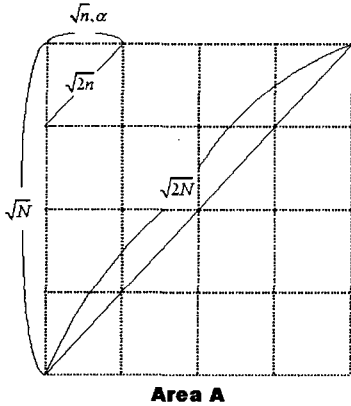


그림 8 CBPER과 TTDD 라우팅 프로토콜의 분석 모델

4.2 통신 비용 분석

이 분석에서는 싱크 노드가 데이터를 소스 노드에게 요구하고 전송 받을 때까지 사용되는 패킷의 수를 구한다. TTDD 라우팅 프로토콜에서는 그리드를 생성할 때 사용되는 패킷의 수가 더해지고 CBPER 라우팅 프로토콜에서는 데이터의 발생을 알릴 때 사용되는 패킷의 수가 더해진다. 분석을 간단하게 하기 위해서 패킷의 병합과 그리드를 유지하기 위해서 사용되는 통신 비용은 고려하지 않는다.

먼저 SODD 라우팅 프로토콜의 통신 비용을 구해본다. SODD 라우팅 프로토콜에서 싱크 노드는 데이터 회송 정보를 모든 센서 노드에게 알리기 위해서 센서 네트워크에 플러딩을 한다. 이때 사용되는 패킷의 수는 전체 센서 노드의 수에 비례한다.

$$Nl \quad (1)$$

데이터 회송 정보를 받은 소스 노드는 데이터를 지리적 회송을 통해서 싱크 노드까지 전송하게 된다. 한 셀에서 싱크 노드가 받는 데이터 패킷은 $\frac{d}{m}$ 개 이고 싱크 노드에서 소스 노드까지의 최대 직선 거리는 센서 필드의 대각선이므로 $\sqrt{2N}$ 이 된다. 그러므로 소스 노드에서 싱크 노드까지 데이터를 보내기 위해서 사용되는 패킷의 수는

$$(c\sqrt{N})\frac{d}{m} \quad (0 < c \leq \sqrt{2}) \quad (2)$$

이다.

k개의 싱크 노드가 m개의 셀을 지날 때의 SODD 라우팅 프로토콜의 총 통신 비용은

$$\begin{aligned} km(Nl + (c\sqrt{N})\frac{d}{m}) \\ = kmNl + kcd\sqrt{N} \end{aligned} \quad (3)$$

이다.

다음은 TTDD 라우팅 프로토콜의 통신 비용을 구해 보겠다. 싱크 노드는 데이터 요구 패킷을 소스 노드에게 전송하기 위해서 회송 정보를 가지고 있는 지역 보급 노드(immediate dissemination node)를 찾는다. 싱크 노드는 이를 위해서 지역적인 플러딩을 한다.

$$nl \quad (4)$$

데이터 요구 패킷을 받은 지역 보급 노드는 회송 정보를 이용하여 데이터 요구 패킷을 소스 노드까지 전송한다. 지역 보급 노드에서 소스 노드까지의 최대 직선 거리는 $\sqrt{2N}$ 이지만 TTDD 라우팅 프로토콜에서는 패킷을 직선으로 보내지 않고 가로 방향, 세로 방향으로 돌려 보낸다. 그렇기 때문에 직선 거리에 $\sqrt{2}$ 을 곱해야 한다.

$$\sqrt{2}(c\sqrt{N})l \quad (0 < c \leq \sqrt{2}) \quad (5)$$

데이터를 요구 받은 소스 노드는 데이터 패킷을 지역 보급 노드까지 전송을 한다. 지역 보급 노드는 데이터를 싱크 노드에게 전송하기 위해서 에이전트 노드에게 보내고 결국 싱크 노드에게 전달된다.

$$(\sqrt{2}(c\sqrt{N}) + c\sqrt{n})\frac{d}{m} \quad (0 < c \leq \sqrt{2}) \quad (6)$$

k개의 싱크 노드가 m개의 셀을 지날 때의 TTDD 라우팅 프로토콜의 통신 비용은

$$\begin{aligned} km \cdot (nl + \sqrt{2}(c\sqrt{N})l + (\sqrt{2}(c\sqrt{N}) + c\sqrt{n})\frac{d}{m}) \\ = kmnl + kc(ml + d)\sqrt{2N} + kdc\sqrt{n} \end{aligned} \quad (7)$$

이다.

TTDD 라우팅 프로토콜에서는 관심 사건(interest event)이 발생 할 때 마다 그리드를 생성해야 한다. 다음은 하나의 그리드를 생성하기 위한 통신 비용이다.

$$\frac{4N}{\sqrt{n}}l \quad (8)$$

마지막으로 CBPER 라우팅 프로토콜의 통신 비용에 대해서 알아 보겠다. 우선 싱크 노드는 데이터 요구 패킷을 자신의 에이전트 노드에게 전송한다. 패킷을 전송 받은 에이전트 노드는 데이터 요구 패킷을 지리적 회송을 통해서 지역 헤더 노드에게 재전송하게 된다.

$$c\sqrt{nl} \quad (0 < c \leq \sqrt{2}) \quad (9)$$

지역 헤더 노드는 회송 정보를 이용하여 소스 노드가 있는 셀의 헤더 노드에게 데이터 요구 패킷을 전송한다. CBRPER 라우팅 프로토콜도 TTDD 라우팅 프로토콜 방식과 똑같이 패킷을 직선으로 전송하지 않고 가로, 세로 방향으로 돌려서 전송한다.

$$\sqrt{2}(c\sqrt{N})l \quad (0 < c \leq \sqrt{2}) \quad (10)$$

데이터 요구 패킷을 받은 헤더 노드는 자신의 셀에 있는 소스 노드에게 패킷을 재전송 한다.

$$c\sqrt{nl} \quad (0 < c \leq \sqrt{2}) \quad (11)$$

소스 노드는 데이터를 싱크 노드에게 전송한다. 이때, 데이터의 전송 경로 길이는 데이터 요구 패킷의 전송 경로 길이와 같다.

$$(\sqrt{2}(c\sqrt{N}) + 2c\sqrt{n})\frac{d}{m} \quad (0 < c \leq \sqrt{2}) \quad (12)$$

k개의 싱크 노드가 m개의 셀을 지날 때의 CBRPER 라우팅 프로토콜의 통신 비용은

$$km(2(c\sqrt{n})l + \sqrt{2}(c\sqrt{N})l + \sqrt{2}(c\sqrt{N})\frac{d}{m} + 2(c\sqrt{n})\frac{d}{m}) \\ = \sqrt{2}kc(ml + d)\sqrt{N} + 2kc(ml + d)\sqrt{n} \quad (13)$$

이다.

CBRPER 라우팅 프로토콜에서는 관심 사건이 발생하면 데이터를 공고해야 한다.

$$(c\sqrt{n} + \sqrt{N})l \quad (0 < c \leq \sqrt{2}) \quad (14)$$

세 라우팅 프로토콜의 전체 통신 비용은 다음과 같다.

- SODD 라우팅 프로토콜

$$kmNl + kcd\sqrt{N} \quad (15)$$

- TTDD 라우팅 프로토콜

$$\frac{4N}{\sqrt{n}}l + kmnl + kc(ml + d)\sqrt{2N} + kdc\sqrt{n} \quad (16)$$

- CBRPER 라우팅 프로토콜

$$(c\sqrt{n} + \sqrt{N})l + \sqrt{2}kc(ml + d)\sqrt{N} + 2kc(ml + d)\sqrt{n} \quad (17)$$

위의 비교에서 CBRPER 라우팅 프로토콜은 이전 라우팅 프로토콜에 비해서 데이터 전송에 필요한 패킷의 수를 N(전체 센서 노드의 개수)의 배수에서 \sqrt{N} 의 배수로 줄였다. 센서 네트워크는 일반적으로 노드의 숫자가 수 천에서 수 만개까지 되기 때문에 CBRPER 라우팅 프로토콜은 기존의 라우팅 프로토콜에 비해서 굉장히 많은 패킷의 수를 줄일 수 있다. 이는 결과적으로 센서 노드의 에너지를 절약하는 효과를 가져온다.

다음은 CBRPER 라우팅 프로토콜과 TTDD 라우팅 프로토콜에서 다중 소스 노드가 존재 할 때의 통신비용을 비교 분석한 결과이다.

TTDD 라우팅 프로토콜

$$\left(\frac{4s}{\sqrt{n}}l + \frac{\alpha^2}{A}l\right)N + \sqrt{2}cs(l+d)\sqrt{N} + scd\sqrt{n} \quad (18)$$

CBRPER 라우팅 프로토콜

$$s(\sqrt{2}cl + \sqrt{2}cd + l)\sqrt{N} + sc(2l + sd + l)\sqrt{n} \quad (19)$$

위의 결과에서도 TTDD 라우팅 프로토콜은 소스 노드의 개수의 N배로 패킷의 수가 늘어나고 CBRPER 라우팅 프로토콜은 \sqrt{N} 배로 패킷의 수가 늘어난다.

다음은 센서 노드에서의 대략적인 배터리 파워 소모량을 분석한 결과이다.

TTDD 라우팅 프로토콜 배터리 파워 소모량

$$N \times 1.055 W \quad (20)$$

CBRPER 라우팅 프로토콜 배터리 파워 소모량

$$\sqrt{N} \times 1.055 W \quad (21)$$

즉, CBRPER 라우팅 프로토콜이 TTDD 라우팅 프로토콜에 비해서 훨씬 더 에너지 효율적인 라우팅 프로토콜이라는 것을 보여주고 있다.

5. 실험 및 결과

본 장에서는 클러스터 기반의 에너지 효율적인 라우팅 프로토콜의 성능 평가를 위한 실험을 수행한다. 4.1에서는 실험 환경에 대하여 설명하고, 4.2에서는 두 계층 데이터 전송 라우팅 프로토콜과의 비교를 통한 실험 결과를 제시하고 분석한다.

5.1 실험 환경

본 논문에서 제안된 라우팅 프로토콜은 NS-2 (Network Simulator 2)[4-6]을 이용하여 구현 되었다. TTDD 라우팅 프로토콜과 에너지 효율성을 비교하기 위해서 NS-2에 구현된 에너지 모델(energy model)을 사용하였다. 에너지 모델에서 센서 노드가 패킷을 전송할 때 사용되는 에너지와 패킷을 수신 할 때 사용되는 에너지는 각각 0.66 W, 0.395 W의 에너지가 소모 된다. 본 실험에서는 MAC 계층으로 802.11을 사용하였다.

본 실험에서 설정된 센서 필드의 크기는 1,000 m × 1,000 m이며 센서 필드 안에 전개된 센서 노드의 개수는 100개이다. 이는 센서 네트워크가 다른 무선 네트워크에 비해서 노드의 밀도가 높기 때문에 위와 같이 설정 하였다. 소스 노드의 개수와 싱크 노드의 개수는 수행되는 실험에 따라서 각각 1개부터 8개, 2개부터 8개까지 설정 하였다. 또한 싱크 노드의 속도는 5 m/s에서 20 m/s이며 5초간의 pause 시간을 갖는다. 소스 노드는 초당 1개의 데이터 패킷을 싱크 노드에게 전송한다. 셀 사이즈는 200 m²으로 설정되었다. 실험은 각각 5개의 랜덤 네트워크 topology에서 수행하여 평균을 냈으며

각 실험은 200초 동안 수행되었다. 본 실험에서 사용된 컨트롤 패킷과 데이터 패킷의 크기는 각각 36 바이트와 64 바이트이다.

제안된 라우팅 프로토콜의 성능을 평가하기 위해서 2개의 측정 기준을 사용하였다. 첫 번째 측정 기준은 통신 활동(패킷의 전송과 수신)에 소비된 에너지(energy consumption)이다. 두 번째 측정 기준은 데이터 패킷의 송, 수신율(delivery ratio)이다. 이는 소스 노드에서 생성된 데이터 패킷의 수를 싱크 노드가 수신한 데이터 패킷의 수를 나눈 값이다.

5.2 실험 결과

다음에서 수행된 실험의 결과를 제시하고 분석한다. 4.2.1에서는 소스 노드의 수와 싱크 노드의 수에 따른 제안된 라우팅 프로토콜의 성능 평가 4.2.2에서는 싱크 노드의 속도에 따른 제안된 라우팅 프로토콜의 성능 평가를 하였다.

5.2.1 소스 노드 수와 싱크 노드 수에 따른 제안된 라우팅 프로토콜의 성능 평가

본 실험에서는 소스 노드의 수를 변화 시키면서 두 라우팅 프로토콜의 성능을 비교 분석 하였다. 싱크 노드는 4개로 고정을 하고 소스 노드의 수를 1, 2, 4, 8로 변화 시킨다. 싱크 노드는 5초의 pause 시간을 가지면서 10 m/s로 이동한다.

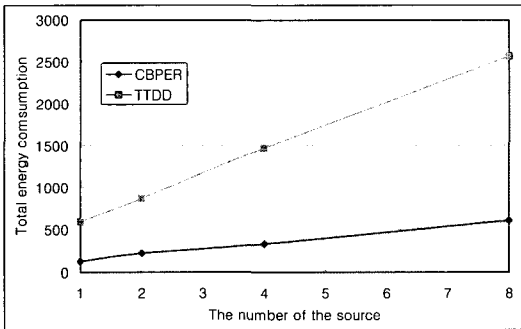


그림 9 소스 노드 개수에 따른 소비된 에너지

그림 9는 소스 노드의 개수에 따른 통신에 소비된 에너지를 보여 주고 있다. 위 실험 결과는 CBPER 프로토콜이 TTDD 프로토콜에 비해서 약 5배의 에너지를 절감하는 것을 보여준다. 이는 TTDD 프로토콜이 소스 노드의 수에 비례해서 그리드를 생성해야 하고 또한 각 소스 노드와 각 싱크 노드 사이의 경로를 유지하기 위해서 주기적으로 업데이트 패킷을 전송하기 때문이다. 이에 반해 CBPER 프로토콜은 소스 노드의 수에 상관 없이 하나의 그리드만 생성하고 소스 노드와 싱크 노드 사이의 경로를 유지 하지 않는다. 또한 데이터 전송에

있어서 TTDD 프로토콜은 데이터 패킷의 병합을 위해서 소스 노드와 싱크 노드 사이의 최단거리를 사용하지 않고 x 축, y 축으로 돌아가는 경로를 택한다. 이는 소스 노드와 싱크 노드 사이의 경로를 계속 유지하기 때문에 가능 하다. 이에 반해 CBPER 프로토콜은 소스 노드와 싱크 노드 사이의 최단 거리를 사용하여 데이터 패킷을 전송한다.

본 실험에서는 싱크 노드의 수를 변화 시키면서 두 라우팅 프로토콜의 성능을 비교 분석 하였다. 소스 노드는 2개로 고정을 하고 싱크 노드의 수를 2, 4, 6, 8로 변화 시킨다. 싱크 노드는 5초의 pause 시간을 가지면서 10 m/s로 이동한다.

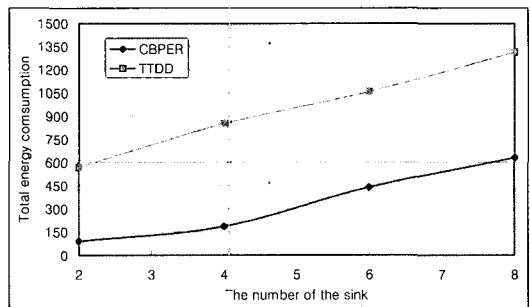


그림 10 싱크 노드 개수에 따른 소비된 에너지

위의 결과에서 싱크 노드의 수가 2개 일 때 CBPER 프로토콜이 TTDD 프로토콜에 비해서 약 6배의 에너지 절감을 가져왔다. 하지만 싱크 노드의 수가 많아 질수록 두 프로토콜 사이의 에너지 소비량 차이는 점점 줄어들고 있다. 싱크 노드의 수가 8개 일 때는 약 2배의 에너지 절감 차이를 보이고 있다. 이는 소스 노드의 수를 2개로 제한 함으로써 TTDD 프로토콜에서 생성되는 그리드의 수를 2개로 고정시켰고 이를 유지하기 위해서 필요한 업데이트 패킷의 양도 비례해서 줄었기 때문이다. 또한 싱크 노드가 많을수록 데이터 패킷을 병합 할 수 있는 기회가 늘어나기 때문에 TTDD 프로토콜이 소비하는 에너지의 양이 줄었다.

위의 두 실험 결과는 소스 노드가 많고 싱크 노드의 수가 적은 응용 분야에서 CBPER 프로토콜이 TTDD 프로토콜 보다 훨씬 에너지를 적게 소모한다는 것을 보여준다.

그림 11는 소스/싱크 노드 개수에 따른 데이터 패킷 전송 성공률이다. 데이터 패킷 전송 성공률은 소스 노드에서 생성한 데이터 패킷의 개수를 싱크 노드가 수신한 데이터 패킷의 개수로 나눈 값이다. 위의 실험 결과는 TTDD 프로토콜의 데이터 전송 성공률에 비해서 약간 낮은 데이터 전송 성공률을 보여준다.

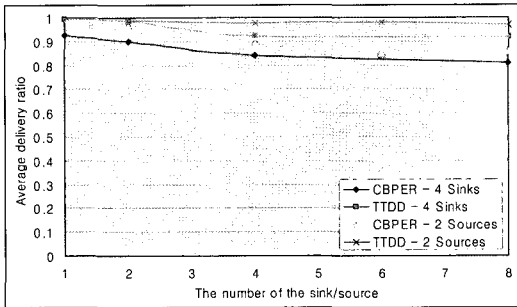


그림 11 소스/싱크 노드 개수에 따른 데이터 전송 성공률

5.2.2 싱크 노드의 속도에 따른 제안된 라우팅 프로토콜의 성능 평가

본 실험에서는 싱크 노드의 속도를 변화 시키면서 두 라우팅 프로토콜의 성능을 비교 분석 하였다. 소스 노드의 수와 싱크 노드의 수는 각각 2개 4개로 고정시켰다. 싱크 노드는 5초의 pause 시간을 가지면서 5, 10, 15, 20 m/s로 이동한다.

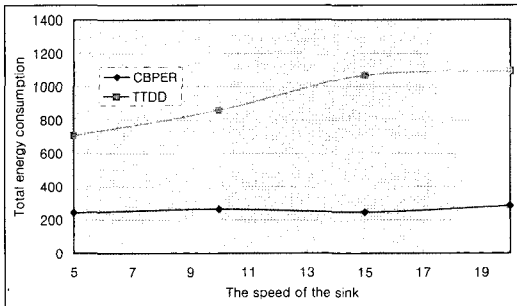


그림 12 싱크 노드의 속도에 따른 소비된 에너지

그림 12는 싱크 노드의 속도를 변화 시켰을 때 소모되는 에너지를 나타낸다. 싱크 노드의 속도가 10 m/s에서 15 m/s 될 때 TTDD 프로토콜에서 에너지의 소비가 약간 증가되었음을 알 수가 있다. 이는 TTDD 프로토콜이 소스 노드와 싱크 노드 사이의 경로를 유지하기 위해서 지속적인 제어 패킷을 전송하기 때문에 싱크 노드의 속도가 빨라지면 그 만큼 더 많은 제어 패킷을 네트워크에 전송해야 하기 때문이다. 위의 실험 결과를 통해서 CBPER 프로토콜이 TTDD 프로토콜에 비해서 싱크 노드의 속도에 덜 영향을 받는다는 것을 알 수 있다.

6. 결론

본 논문에서는 센서 네트워크를 위한 에너지 효율적인 라우팅 프로토콜을 제안 하였다. 센서 네트워크는 다

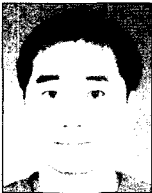
른 무선 네트워크를 보다 훨씬 많은 에너지 제약을 받는다. 그렇기 때문에 센서 네트워크에서 에너지 문제는 제일 중요하게 다뤄왔다. 본 논문에서는 센서 노드들이 싱크 노드로 경로를 설정 할 때 통신에 사용되는 에너지 소비를 줄여 센서 네트워크의 수명을 연장 시켰다.

본 논문에서는 그리드 개념을 이용하여 에너지 효율적인 라우팅 프로토콜을 제안 하였다. 실험을 통해서 기존의 그리드 개념을 이용한 TTDD 라우팅 프로토콜에 비해 에너지를 약 5배에서 6배 정도 덜 소모하는 것을 보였다. 또한 CBPER 프로토콜은 싱크 노드의 수가 적고 소스 노드의 수가 많은 응용 분야에서 좀 더 효율적이라는 것을 실험을 통해서 알 수가 있었다. 실질적으로 센서 네트워크에서는 많은 관심 사건이 발생하며 소수의 싱크 노드가 존재한다. 그러므로 TTDD 라우팅 프로토콜 보다 CBPER 프로토콜이 좀 더 현실적인 센서 네트워크에 적합하다. 하지만 데이터 전송 성공률은 TTDD 프로토콜에 비해서 약 10% 떨어졌다. 이는 앞으로 개선해야 할 문제이다. 싱크 노드의 속도를 달리하면서 에너지 소비를 측정하는 실험에서는 CBPER 프로토콜이 TTDD 프로토콜에 비해서 싱크 노드의 속도에 덜 민감하게 작용한다는 것을 보였다. 그러므로 고속의 싱크 노드에도 CBPER 프로토콜이 적합하다는 것이 실험을 통해서 증명이 되었다.

참고 문헌

- [1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramanian, and Erdal Cayirci, "A Survey on Sensor Networks," IEEE Communications Magazine 2002.
- [2] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, "Directed Diffusion : A Scalable and Robust Communication Paradigm for Sensor Networks," MOBICOM 2002.
- [3] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," MOBICOM 2002.
- [4] Ian Downard, "Simulating Sensor Networks in NS-2," 2003.
- [5] Network Simulator-ns-2, <http://www.isi.edu/nsnam/ns/>.
- [6] The CMU MONARCH Group, "Wireless and mobility extension to ns-2," 1999.
- [7] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," 2000.
- [8] Stefano Basagni, Imrich Chlamtac, and Violet R. Syrotiuk, "A Distance Routing Effect Algorithm for Mobility," MOBICOM 1998.

- [9] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," INFOCOM 2002.
- [10] T. S. Rappaport, "Wireless Communications, principles and practice," Prentice Hall 1996.
- [11] Charles E. Perkins, "Ad Hoc Networking," Addison-Wesley 2001.
- [12] B. Karp and H. Kung, "GPSR : Greedy Perimeter Stateless Routing for Wireless Networks," MOBOCOM 2000.
- [13] Charles E. Perkins and Pravin Bhagwat, "DSDV Routing over a Multihop Wireless Network of Mobile Computers," MOBICOM 1996.
- [14] Yong-Bae Ko and Nitin H. Vaidya. "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," MOBICOM 1998.
- [15] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," MOBICOM 1998.
- [16] F. Ye, S. Lu, and L. Zhang, "GRADient Broadcast : A Robust, Long-lived Large Sensor Network," 2001.
- [17] D. Coffin, D. V. Hook, S. McGarry, and S. Kolek, "Declarative Ad Hoc sensor networking," SPIE Integrated Command Environments 2000.



권기석

2001년 한양대학교 전자, 컴퓨터 공학부 졸업. 2001년~2002년 삼성전자 정보통신총괄 근무. 2004년 한국과학기술원 전산학과 석사. 2004년~현재 한국과학기술원 전산학과 박사과정. 관심분야는 센서 네트워크, 애드혹 네트워크, 셀룰라 네

트워크 등



이승학

2000년 한국과학기술원 전산학과 졸업(학사). 2003년 한국과학기술원 전자전산학과 졸업(석사). 2003년 3월~현재 한국과학기술원 전자전산학과 박사과정. 관심분야는 애드혹 네트워크, 센서 네트워크 등



윤현수

1979년 서울대학교 전자공학과 졸업
1981년 한국과학기술원 전산학과 석사
1988년 미국 오하이오 주립대학 전산학과 박사. 1989년~한국과학기술원 교수
관심분야는 병렬 컴퓨터 구조, 무선 이동통신, 애드혹 네트워크, 정보 보호 등