

클러스터 트리 라우팅 프로토콜 연구

(Study of Cluster Tree Routing Protocols)

조무호*

(Moo-Ho Cho)

요 약

클러스터 기반의 계층적 트리 구조는 대형화가 용이하면서 라우팅 프로토콜이 단순하다는 점에서 장점을 가진다. 본 논문에서는 무선 센스 네트워크에서 클러스터 트리 네트워크 관리를 효과적으로 지원할 수 있는 방안과 라우팅 프로토콜의 데이터 모음과 지연에 대해 분석한다. 수치 해석의 결과에서 보면 중간 노드에서의 데이터 모음을 통한 통신 메시지 수를 줄임으로 센스 노드의 전력 소모를 크게 절약할 수 있지만, 이에 따른 메시지 전달 지연이 커짐을 알 수 있었다. 또한 클러스터 헤드를 무작위로 선정하게 되면 메시지 지연이 매우 많이 증가될 수 있는 것이 확인되었다.

Abstract

A hierarchical tree structure of clusters has advantages for the network design due to its scalability and simple routing protocol. In this paper, the cluster tree routing protocol is studied for the wireless sensor network. From the numerical analysis results, the data aggregation in the intermediate nodes reduces the number of communication message and saves the energy of sensor nodes, but it may result in increased data traffic latency. And also the selection of cluster head can increase the relaying hops very high.

Key Words : IEEE 802.15.4, WSN, Sensor, Adhoc

1. 서 론

센서 네트워크이란, 사물과 환경의 변화를 실시간 감지 또는 추적하기 위하여 센서, 안테나, 집적회로 등을 하나의 칩으로 만든 장치를 사물에 집어넣고, 이를 네트워크로 연결하여 구성한 무선 네트워크이

다. 이러한 네트워크는 “언제 어디서든 어떤 기기를 통해서도 컴퓨팅할 수 있는 것”을 의미하는 유비쿼터스 컴퓨팅을 실현하는데 핵심적인 기능이며, 미래 정보통신의 기본 인프라로 발전하게 될 것으로 전망된다. 센서 네트워크는 임의의 형태로 필요할 때 형성되는 ad-hoc 네트워크이며, 네트워크의 형성이나 유지 관리가 인간의 관여 없이 자동적으로 이루어지는 자가 구성(self-organizing) 네트워크이다. 무선 센서 네트워크로부터의 정보 전달을 위해서는 센서 노드들간의 정보 라우팅 기능이 요구되는데, 기존의

* 주저자 : 경주대학교 컴퓨터멀티미디어공학부 조교수
Tel : 054-770-5277, Fax : 054-744-5293

E-mail : mhcho@kju.ac.kr
접수일자 : 2005년 10월 24일
1차심사 : 2005년 10월 27일
심사완료 : 2005년 11월 10일

무선 ad-hoc 네트워크에서 제안된 라우팅 프로토콜은 센서 네트워크가 가지는 한정된 자원과 성능으로 인해 적용하기가 적합하지 않다. 그러므로 무선 센서 네트워크의 자가 구성 능력, 제한된 전력, 데이터 중심적인 특성을 고려한 네트워크 구성과 라우팅 프로토콜이 요구된다[1,2].

센서 네트워크에 적용될 무선 ad-hoc 라우팅 기법은 크게 평면 라우팅(flat routing)과 계층적 라우팅(hierarchical routing) 프로토콜로 나뉜다. 평면 라우팅은 네트워크 전체를 하나의 영역으로 간주하여 모든 노드들이 동등하게 라우팅에 참여할 수 있고 멀티 홉 라우팅을 특징으로 한다. 반면에 계층적 라우팅은 네트워크를 클러스터링을 기반으로 한 다수의 영역으로 분할하여 각각의 영역내의 특정 노드에 헤드의 역할을 부여하여 라우팅을 수행하도록 한다.

센서 네트워크의 특징을 고려할 때 기존의 무선 ad-hoc 평면 라우팅 프로토콜을 채택할 경우, 수백에서 수천 개의 노드들이 경로를 찾기 위한 경로 요청 메시지들이 발생하게 되어 많은 전력이 소비되고, 많은 지연이 발생하게 된다. 테이블 기반 방식에서는 네트워크 내에 있는 수많은 노드까지의 경로를 한정된 자원을 가진 센서 노드가 유지하는데 한계가 존재한다. 즉 센서 노드의 메모리 용량이 수백 킬로바이트 정도이므로, 수백 개에서 수천 개에 이르는 노드들에 대한 정보를 가진 라우팅 테이블을 유지하고 관리하는 것은 사실상 불가능하다. 그러므로 이를 고려하여 속성 기반 어드레싱과 데이터 모음등을 고려한 간단한 평면 라우팅 프로그램이 요구된다[3].

클러스터 기반의 계층적 트리 구조는 대형화가 용이하면서 라우팅 프로토콜이 단순하다는 점에서 장점을 가진다. 즉 로컬 클러스터를 형성함으로써 인접한 지역에서 발생한 사건에 대한 유사한 정보를 클러스터 헤드로 전송하고 클러스터 헤드가 데이터 모음을 수행하여 보다 전력 소모가 효율적인 라우팅을 가능하게 한다. 또한 요청된 질의에 대한 클러스터 헤드에 의한 전달로 비효율적인 플러딩 되는 질의들을 막을 수 있다.

본 논문에서는 이러한 점을 고려하여 무선 센서 네트워크에서 클러스터 트리 네트워크 관리를 효과적으로 지원할 수 있는 방안에 대해 기술한다.

2. 클러스터 트리 네트워크

센서 네트워크에서도 다른 통신 네트워크와 마찬가지로 네트워크 규모를 대형화할 수 있는 속성은 매우 중요하다. 단일 계층(tier)의 네트워크는 센서 노드들이 네트워크에 추가되어 증가하게 되면 게이트웨이 노드의 부하가 가중되게 된다. 이러한 과부하는 통신 지연을 초래하게 되고, 사건 발생을 추적하는데 지장을 초래하게 된다. 또한 통상적으로 센서 노드의 한정적인 전력으로 인해 통신 능력이 장거리 통신에 적합하지 않기 때문에 단일 계층으로 대상 지역이 넓고 많은 노드들을 서비스 하는 규모로 커지게 하는 것은 불가능하다. 이러한 추가되는 노드들에 의한 부하를 감당하고, 광범위한 지역의 서비스를 가능하게 하기 위해서 네트워크를 클러스터링 하는 방안이 대두 연구되기 시작 했다. 클러스터 네트워크의 주된 목적은 클러스터 내에서 멀티 홉 통신과 클러스터 헤더에서의 데이터 모음과 통합을 통하여 싱크로 가는 메시지 수를 줄임으로 센서 노드의 전력 소모를 효율적으로 유지 감소시키는 것이다[4].

센서 노드들은 클러스터 헤드를 선택하고 자가 구성 방법으로 클러스터를 형성한다. 클러스터 형성 과정에서 클러스터 헤더는 각 멤버 노드들에게 고유한 노드 식별 번호를 부여한다. 자가 구성된 클러스터들은 Designated Device(DD)에 의해서 서로 연결된다. DD는 다른 노드들보다 우수한 계산 능력과 대용량 메모리를 보유한 특별한 디바이스이다. 대부분의 응용에서 DD는 센서 네트워크와 인터넷 사이에서 게이트웨이 역할을 수행한다. DD는 각 클러스터에 고유한 클러스터 식별번호를 부여한다[5].

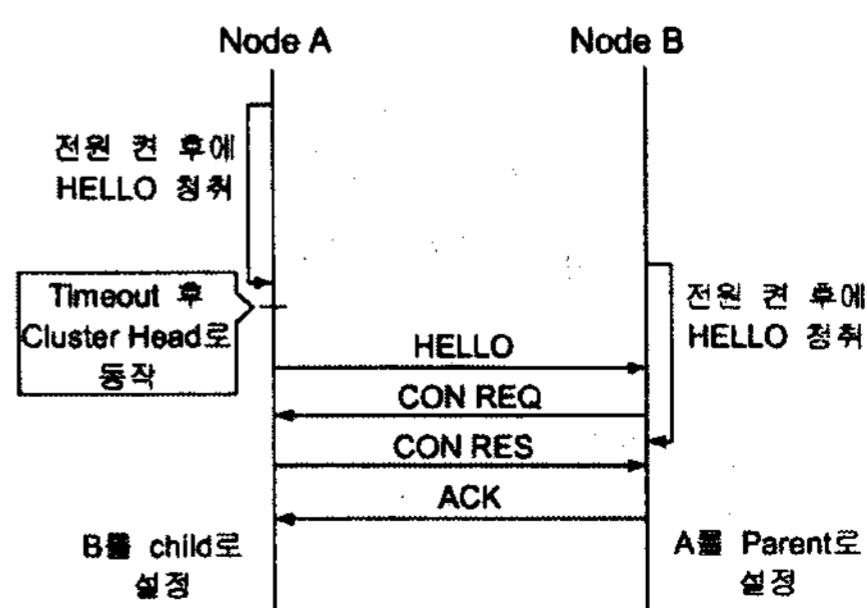


그림 1. 클러스터 헤더 선택 및 노드 추가
Fig. 1. Cluster head selection and node addition

2.1 클러스터 헤더 선택 과정

클러스터 형성은 클러스터 헤더 선택으로부터 시작된다. 어느 한 노드의 동작이 시작되면 그림 1과 같이 다른 노드로부터 송출되는 HELLO 메시지를 기다린다. 만약 HELLO 메시지를 어느 기간 동안 기다린 후에도 들을 수 없으면 인접에 클러스터 헤더가 없는 것으로 간주하고 자신이 클러스터 헤더가 되어 HELLO 메시지를 송출한다. 이 새로운 클러스터 헤더는 일정 시간 동안 이웃으로부터 응답을 기다린다. 만약 어떠한 연결 요구도 받을 수 없으면 다시 일반 노드로 돌아가 HELLO를 수신하게 된다.

클러스터헤더는 전송범위, 전원 능력, 계산 능력 또는 위치 정보등과 같은 각 노드의 파라미터를 근거로 선택될 수 있다.

2.2 단일 홉 클러스터

어느 한 노드가 클러스터 헤더로 선택되어 주기적으로 HELLO 메시지를 방송하게 된다. 여기에는 클러스터 헤더의 MAC 주소와 클러스터 헤더 식별 번호를 나타내는 Node ID(NID) 0번이 포함되어 있다. 이 HELLO 메시지를 받은 노드는 클러스터 헤더로 연결 요구(CON REQ) 메시지를 송출한다. 이 메시지를 받은 클러스터 헤더는 요구한 노드를 위한 NID 번호가 포함된 연결응답(CON RES) 메시지로 응답한다. NID는 클러스터 내에서 유일해야 하며, 클러스터 헤더가 유일한 NID를 할당하는 책임을 담당하게 된다. NID를 할당 받은 노드는 클러스터 헤더로 ACK 메시지로 응답하여 링크 연결을 완료한다.

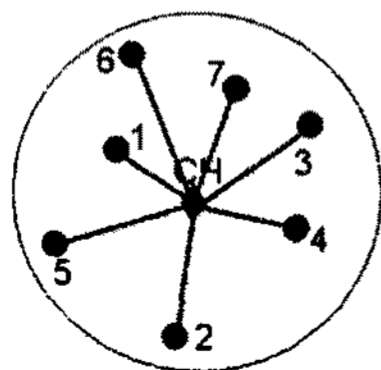


그림 2. 단일 홉 클러스터 구조
Fig. 2 Single hop cluster structure

링크 연결 후 각자의 이웃목록(neighbor list)에 상대방을 parent와 child로 설정하는데, 클러스터 헤

더는 새로이 추가된 노드를 child로 설정하고, 새로이 추가된 노드는 클러스터 헤더를 parent로 설정한다. 만약 모든 노드들이 클러스터 헤더의 서비스 영역인 1홉 내에 위치하게 되면 접속 형태(topology)는 그림 2와 같이 성형(star)이 된다. 클러스터 내에 존재할 수 있는 노드 수는 방송과 임시 NID를 제외하면 최대 254개가 된다.

2.3 멀티 홉 클러스터

클러스터는 각 노드들이 다중 연결을 지원하면 멀티 홉 구조로 확장할 수 있다. 비록 통신 지연은 증가하게 되지만 서비스 커버리지는 하나의 클러스터 서비스 영역에서 확장될 수 있다.

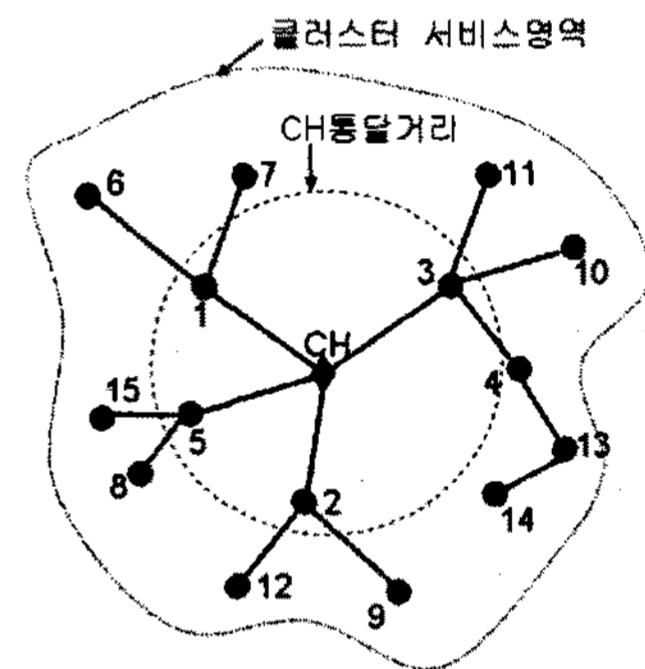


그림 3. 멀티 홉 클러스터 구조
Fig. 3 Multi-hop cluster structure

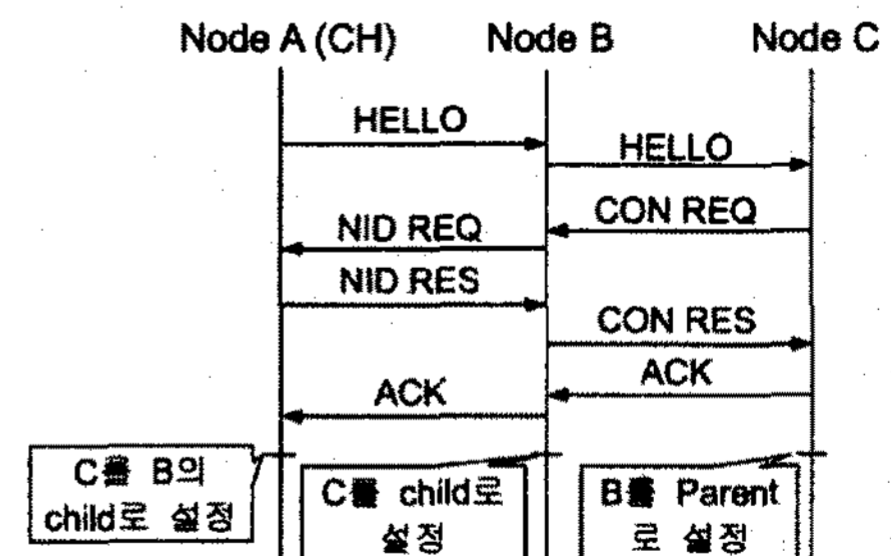


그림 4. 멀티 홉 클러스터 형성 절차
Fig. 4 Multi hop cluster setup and node addition

그림 4에서와 같이 노드 B는 클러스터 헤더와의 링크가 설정된 후에 클러스터 헤더로부터 전송되는 HELLO 메시지를 중계하게 된다. 노드 C가 노드 B로부터 HELLO를 받게 되면 연결 요구 메시지를 송

출한다. 노드 B는 클러스터 헤드로 노드 C의 NID 요구(NID REQ)하고, 새로운 NID 할당(NID RES) 받아 노드 C로 응답한다. 링크가 연결된 후에는 노드 B를 노드 C의 parent로, 노드 B는 노드 C를 child로, 클러스터 헤드는 C를 B의 child로 설정한다. 이후에 노드 C는 이웃을 위하여 HELLO 메시지를 중계하게 된다. 만약 어느 노드가 여러 개의 HELLO 메시지를 받게 되면 가장 빨리 도착한 메시지에 응답한다. 이 경우에 일반적으로는 최단 경로가 되지만, 클러스터 헤더로의 경로가 최적일 수도 있으며, 추후에 최적화가 이루어진다. 만약 클러스터 헤더가 NID를 할당할 식별번호를 모두 사용하거나 다른 제약에 의해서 새로운 노드의 연결 요구를 거절해야 하는 경우에는 NID를 254로 사용한다. NID 254를 받은 새로운 노드는 클러스터 식별번호 (CID)를 저장하고 한동안 이 CID에 속한 노드들로 연결 요구를 전송하지 않는다.

3. 네트워크 관리 방안

3.1 링크 상태 보고

클러스터 헤더는 주기적으로 HELLO 메시지를 멤버 노드들에게 전송한다. 멤버 노드는 HELLO 메시지를 받게 되면 그들의 이웃을 위해 다시 방송하게 된다. 각 노드는 HELLO 메시지가 방송될 때 그들의 이웃 노드 정보를 인식하여 이웃목록에 기록한다. 특정 시간 동안 이웃 노드에 대한 HELLO 정보가 갱신되지 않으면 목록에서 제거된다. 만약 다른 클러스터에 속한 노드로부터 HELLO 메시지를 받으면 클러스터 번호를 이웃목록에 저장한다.

모든 노드들은 클러스터 헤더로 링크상태보고(Link State Report)를 사용하여 이웃목록을 주기적으로 또는 요청에 의해서 보고한다. 링크상태보고는 클러스터 트리의 최종 말단에서 시작되며, 클러스터 헤더로 가는 경로의 중간에 있는 노드들은 이 메시지를 중계하면서 트리 하부 노드들에 대한 정보를 얻게 된다. 중간 노드들은 여러 child들로부터 수신한 정보를 종합하여 링크상태보고 메시지를 parent로 송부한다. 이때 트리 하부에 직접 통신이 안 되는 노드들

의 경로를 링크상태보고를 송신한 child 노드로 표시하여 라우팅시 트리를 따라서 찾아갈 수 있게 단순화시킨다. 이러한 과정은 링크 상태 보고가 최종 클러스터 헤더에 도달할 때까지 반복되며, 클러스터 내에 존재하는 어떠한 노드라도 최종적으로 클러스터 헤더에서는 라우팅 경로를 알 수 있게 된다.

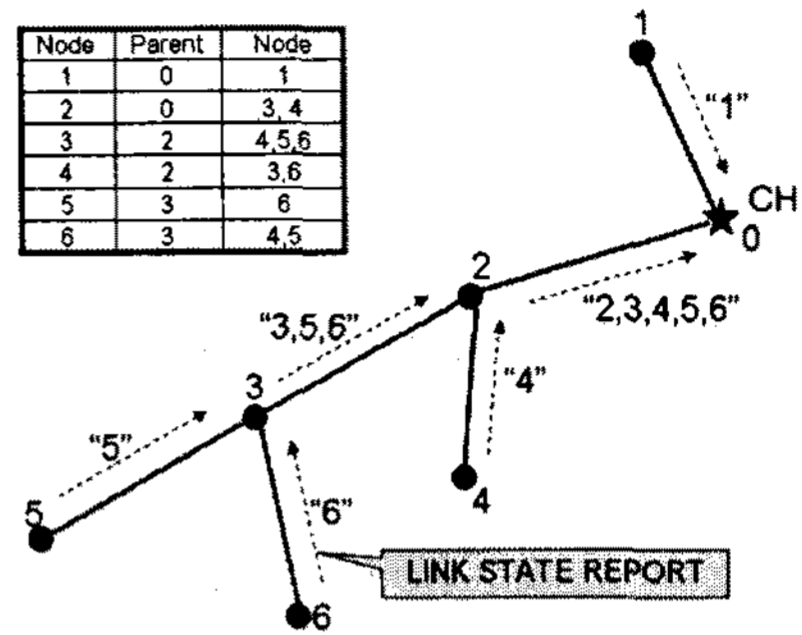


그림 5. 링크상태보고 메시지 전달
Fig. 5. Link state report message delivery

각 노드는 직접 통신이 가능한 이웃 노드들과는 직접 통신할 수 있다. 만약 노드의 통신 범위는 벗어나지만 자신의 라우팅 테이블에 존재하는 목적지 즉, 하부 트리에 존재하는 목적지는 트리를 따라 전송 가능하다. 자신의 라우팅 테이블에 존재하지 않는 목적 노드는 parent로 보내어 최종 목적지로 메시지 중계를 요청한다.

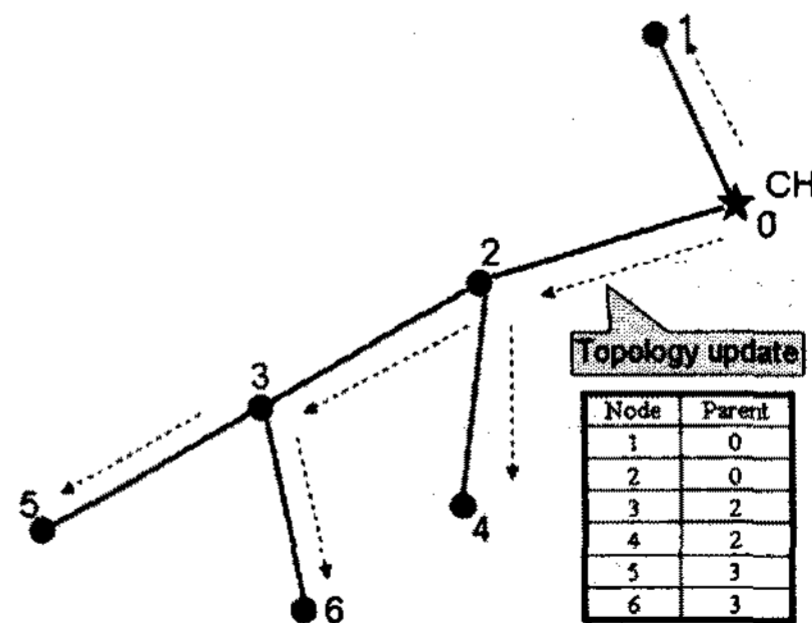


그림 6. 토폴로지 수정 메시지 전달
Fig. 6. Topology update message delivery

3.2 토폴로지(Topology) 수정

클러스터 헤더는 모든 노드들로부터 송출되는 링

크 상태 보고를 근거로 각 멤버 노드들과의 최적 경로를 주기적 계산하여, 토폴로지 수정 메시지를 통해 각 노드들에게 통보한다. 최적 경로 선정 방법은 가장 작은 홉수를 가지는 경로를 근거로 한다. 만약 동일한 홉수를 가지는 다수개의 경로가 존재하면 parent 노드의 식별번호가 가장 최소의 NID를 가지는 경로를 선택한다. 클러스터의 멤버 노드는 parent 노드가 변경되어 명시된 토폴로지 수정 메시지를 수신하게 되면 메시지에 명시된 것과 같이 parent 노드를 변경한다.

3.3 노드 고장으로부터 링크 회복

멤버 노드 중에 하나가 고장이 발생하여 통신이 불가능한 경우에는 클러스터 트리 경로는 재구성되어야 한다. 만약 어느 노드의 동작이 중단되면 인접 노드들로부터 보고되는 링크상태보고의 이웃목록에서 제외된 것으로 보고된다. 클러스터 헤더는 각 노드에서 송출되는 링크상태보고를 참조하여 고장난 노드를 검출한다. 그리고 고장난 노드로 인해 단절된 이하 노드들을 연결하기 위해 링크상태보고의 이웃목록에서 가능한 경로를 찾고, 그 경로를 따라 토폴로지 수정 메시지를 송출하여 단계적으로 링크를 회복시킨다.

만약 클러스터 헤더에 문제가 발생하면 HELLO 메시지의 송출이 중단되어 모든 멤버들이 알게 된다. 이 경우에는 클러스터 초기 형성과 동일한 방법으로 재구성이 이루어져야 한다.

4. 수치해석

멀티 홉 클러스터 네트워크에서는 중간 노드들에서의 데이터 모음과 통합을 통하여 클러스터 헤드로 가는 메시지 수를 줄임으로 센서 노드의 전력 소모를 감소시키는 것이다. 그림 7은 개별 전송과 데이터 모음의 차이를 나타내고 있다. 노드 레벨(Node Level)은 클러스터 헤드로부터의 홉수를 나타내며, 클러스터 헤더는 레벨이 0이고, 클러스터 헤더의 child는 레벨 1이 된다. 그림 7은 한 노드에서 child를 4개씩 가지면서, 노드 레벨 5까지 각 노드에서 클러스터 헤드

로의 개별 전송할 때의 전체 홉수와 데이터 모음을 수행했을 때의 홉수를 비교하여 나타내고 있다. 개별 전송에 비해 데이터 모음을 수행했을 때의 홉수가 약 1/5로 감소함을 나타낸다. 데이터 모음에 요구되는 전력에 비해 데이터 전송에 요구되는 전력이 상대적으로 아주 큰 점을 고려하면 멀티홉 클러스터 네트워크에서의 데이터 모음은 센스 노드의 전력 소모 절약에 매우 중요하다는 것을 나타내고 있다.

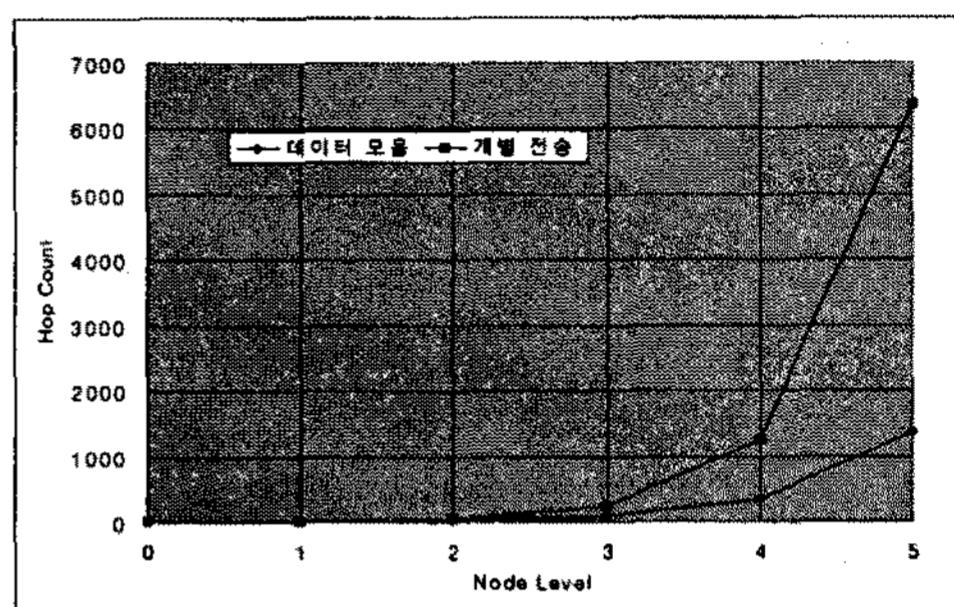


그림 7. 개별 전송과 데이터 모음의 홉수 비교
Fig. 7. The hop differences between individual transmission and data aggregation

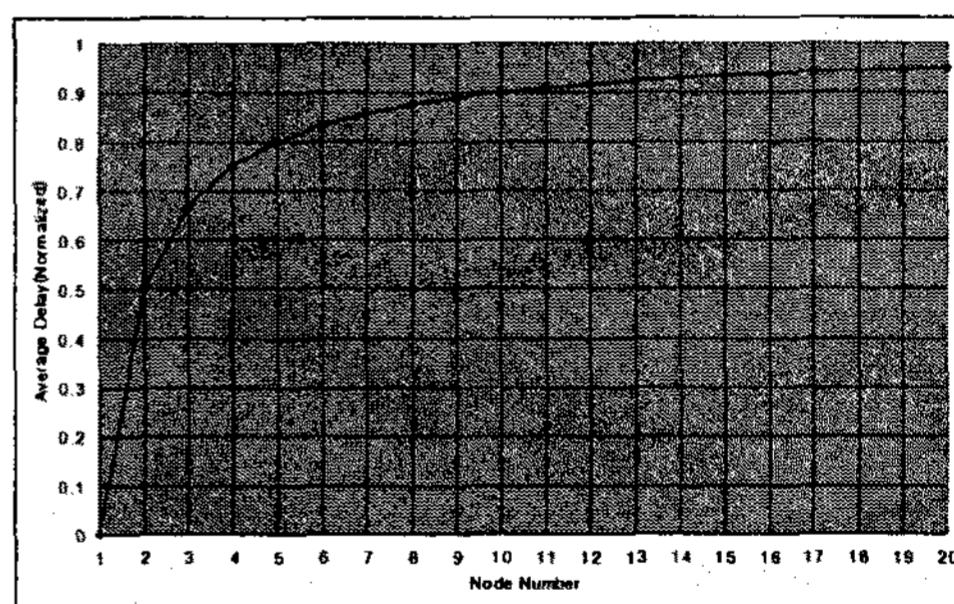


그림 8. 정규화된 데이터 모음의 평균지연
Fig. 8. Normalized average delay for data aggregation

데이터 모음을 수행하기 위해서는 필연적으로 지연이 수반된다. 그림 8은 센스 노드 수에 따라 데이터 모음을 위해 중간 노드에서 기다리는 평균 지연을 정규화 시킨 것이다. 노드의 개수가 증가될수록 기다리는 시간이 길어지는 것을 나타내고 있다. 거의 99.9%가 비활성 상태에 있는 센서 노드의 특성을 고려하면 데이터 모음에 지연을 최소화 시킬 수 있는 방안이 연구되어야 한다.

멀티 홉 클러스터 네트워크에서 해결해야 할 또

하나의 중요한 문제점은 클러스터 헤드의 전력 소모 문제이다. 클러스터 헤드는 멀티 홉 네트워크 내의 모든 통신이 집결되고, 처리되므로 일반 노드에 비해 전력 소모가 많다. 이를 해결하기 위해 클러스터 헤드를 특정 시간 마다 무작위로 순환 시키는 LEACH(Low-Energy Adaptive Clustering Hierarchy) 방안이 제안되어 있다[6]. LEACH 방안은 각 노드에 전력 소모를 균등하게 배분하는 것은 해결했지만, 클러스터 헤더가 무작위로 선정됨에 따른 문제점들을 안고 있다. 멀티홉 클러스터 네트워크에서는 현재의 헤드를 중심으로 최적화되게 트리 형태로 형성되어 있다. 만약 무작위로 선정되는 클러스터 헤더가 트리의 가장 말단 노드로 선정되게 되면 최악의 지연이 발생하는 결과를 초래한다.

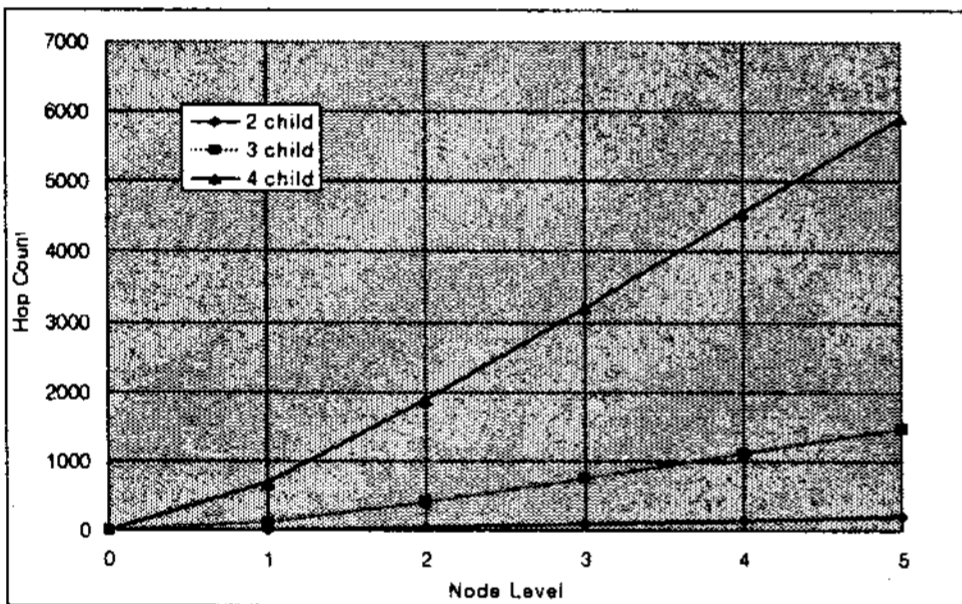


그림 9. 클러스터 헤더 변경에 따른 추가되는 홉수
Fig. 9. The increased hops of cluster head rotation

그림 9는 무작위로 선정되는 클러스터 헤드의 위치에 따라 추가되는 홉수를 나타내고 있다. 4개의 child를 가지는 경우에 노드 레벨 5 즉, 헤더로부터 5홉 하위의 노드가 선정되면 노드 레벨 1에 선정되는 것에 비해 약 8.7배의 홉수가 늘어나게 된다. 이러한 결과는 LEACH에서 얻을 수 있는 4~8배의 효과를 상쇄시키는 결과를 초래한다.

5. 결 론

본 논문에서는 클러스터 센스 네트워크에서 네트워크 관리를 효과적으로 지원할 수 있는 방안에 대해 살펴보았다. 클러스터 네트워크는 계층적 트리 구조로 대규모로 구성하기가 용이하면서 라우팅 프

로토콜이 단순하다는 점에서 장점을 가진다. 또한 중간 노드에서의 데이터 모음을 통한 통신 메시지 수를 줄임으로 센스 노드의 전력을 크게 절약할 수 있지만, 중간 노드에서 메시지 전달 지연이 커지는 단점이 있다. 그리고 클러스터 헤더가 무작위로 선정되면 센서 노드들의 전력 소모는 균등하게 배분되지만 지역적인 말단에 선정되는 경우에는 이에 따른 메시지 지연이 크게 증가하는 문제점이 있다.

앞으로의 연구 방향은 클러스터 트리 네트워크에서 트리 형성의 최적화 방안과 메시지 지연이 최소화되는 클러스터 헤더의 전력 균등 소모 방안에 대해 지속적인 연구가 계속되어야 한다.

이 논문은 경주대학교 교내 학술연구비 지원을 받아 수행된 연구임.

References

- [1] Ian F. Akiyldiz et al., "A survey on Sensor Networks", IEEE Communications Magazine, pp. 102-114, Aug. 2002.
- [2] K. Sahrabi, et al., "Protocols for self-organization of a wireless sensor network," IEEE Personal Communication, Vol. 7, No. 5, pp. 16-27, Oct. 2000.
- [3] 배정숙, 김성희, "무선 센서 네트워크에서의 라우팅 프로토콜", 한국전자통신연구원 주간기술동향, 1140권호, 2004.04.
- [4] K. Akkaya and M. Younis, "A Survey of Routing Protocols in Wireless Sensor Networks," in the Elsevier Ad Hoc Network Journal (to appear).
- [5] Ed Callaway, "Cluster Tree Network", IEEE P802.15 Working Group for WPANs, April 2001.
- [6] W. Heinzelman, et al. "Energy-efficient communication protocol for wireless sensor networks," Proceeding of the Hawaii International Conference System Science, January 2000.

◇ 저자소개 ◇

조무호 (趙茂昊)

1957년 7월 18일생. 1980년 2월 경북대학교 전자공학과 졸업. 1989년 8월 청주대학교 대학원 전자공학과 졸업(석사). 1998년 2월 충북대학교 대학원 정보통신과 졸업(박사). 1983년 3월~2000년 2월 한국전자통신연구원(ETRI) 책임연구원. 2000년 3월~현재 경주대학교 컴퓨터멀티미디어공학부 조교수.