

다중 홉 클러스터 센서 네트워크에서 속성 기반 ID를 이용한 효율적인 융합과 라우팅 알고리즘

이 보 형(李寶馨), 이 태 진(李泰珍)
성균관대학교 정보통신공
전화/팩스 : (031) 290-7982
H.P 번호 : 019-208-7831

Efficient Aggregation and Routing Algorithm using Local ID in Multi-hop Cluster Sensor Network

Bo Hyeong Lee, Tae-Jin Lee
Dept of Information and Communication Engineering,
SungKyunKwan University
E-mail : {shaak,tjlee}@ece.skku.ac.kr

Abstract

Sensor networks consist of sensor nodes with small-size, low-cost, low-power, and multi-functions to sense, to process and to communicate. Minimizing power consumption of sensors is an important issue in sensor networks due to limited power in sensor networks. Clustering is an efficient way to reduce data flow in sensor networks and to maintain less routing information. In this paper, we propose a multi-hop clustering mechanism using global and local ID to reduce transmission power consumption and an efficient routing method for improved data fusion and transmission

I. 서론

센서 네트워크는 저전력, 저가격, 소형, 근거리 무선 통신이 가능한 다기능 노드를 사용하여, 센싱, 데이터 프로세싱, 통신을 할 수 있도록 구성된 네트워크를 말한다[1]. 센서 네트워크의 설계에 있어 가장 중요하게 고려되는 것은 네트워크의 생존 시간이다. 한정된 에너지를 갖는 센서로 구성된 네트워크에서 하나의 센서라도 전력을 모두 소비하면 네트워크의 연결이 단절되어 데이터 전송의 신뢰성을 보장받지 못할 수 있다. 따라서 센서 네트워크에서는 노드가 소비하는 전력을

최소화하고 네트워크 생존 시간을 늘리기 위한 알고리즘이 많이 연구되고 있다[1],[2],[3],[8].

센서 네트워크에서는 각 노드의 센서가 얻을 수 있는 제한된 정보만을 전송하기 때문에 인접한 노드들에서 발생한 데이터들은 서로 비슷한 경우가 많게 된다. 이러한 데이터의 유사한 성질을 모아 전송할 데이터의 양을 줄이는 것을 데이터 융합(data fusion)[2] 또는 통합(aggregation)이라 부른다. 데이터 융합을 사용하였을 경우에는 전체 네트워크에서 전송데이터의 양이 데이터 융합을 사용하지 않았을 경우보다 많이 줄어들기 때문에 전력소비가 적어져 네트워크의 생존 시간이 더 길어질 수 있다. 일반적으로 데이터 융합을 전체 네트워크에 대해 실행할 경우 효율적으로 하기 어렵기 때문에 노드들을 클러스터 단위로 묶고 그 안에서 데이터 융합을 하는 방법(그림 1 참조)이 선호된다[3].

클러스터링은 데이터 융합뿐 만아니라, 라우팅에 필요한 전체 메시지의 수를 줄일 수 있지만, 네트워크의 규모가 커지면 커질수록 이런 장점은 사라지게 된다. 클러스터링을 계층적으로 구성하여 클러스터를 확장하는 방법도[1] 있지만, 이런 경우는 단계(level)만큼의 ID와 라우팅 테이블 추가와 데이터의 중복되는 전송으로 인해 에너지 소모가 증가하게 된다.

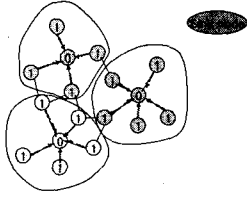


그림 1 클러스터링과 데이터 융합

본 논문에서는 네트워크 내의 Cluster Head(CH) 개수를 줄이고 중복되는 전송을 줄이기 위해 계층적인 클러스터를 구성하지 않고 다중 홉 클러스터링을 하는 방법과 클러스터 내 노드의 ID를 global ID+지역/속성 ID로 작성하여 이 ID를 가지고 클러스터 단위의 스케줄링을 통해 데이터 융합과 라우팅을 효율적으로 할 수 있는 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2절은 클러스터링 및 ID설정 방법, 그리고 3절은 라우팅 설정 방법에 대해, 4절은 데이터 융합 과정, 5절에서 성능분석을 제시하고, 6절에 결론을 기술한다.

II. 클러스터링 및 ID설정 방법

어떤 노드를 CH로 결정하는가에 따라 네트워크 생존 시간에 큰 영향을 미치지만 이 문제는 어떠한 클러스터링 알고리즘을 사용할 것인가에 주로 좌우한다. 본 논문의 목적은 클러스터 내의 노드들에 지역ID를 배정하고 이를 바탕으로 해서 효율적인 데이터 융합과 라우팅에 이용하는 방법을 구현하는 것이기 때문에 클러스터링 과정에서 CH 노드가 되는 조건은 각 노드의 현재 에너지 보유량에 따라 결정하는 것으로 가정한다.

센서 노드들이 배치되어 있는 상태에서 전원이 켜진 시점에서 각 노드는 자신이 가진 에너지를 기초로 하여 CH가 될지를 결정한다. 많은 전력을 가진 노드를 CH가 되는 랜덤 시간을 짧게 하여, 에너지가 많은 노드가 될수록 먼저 CH가 되도록 한다. CH가 되기로 결정을 한 노드는 자신의 global ID와 그 클러스터의 스케줄이 포함된 CH 신호(그림 2 참조)를 주변에 브로드캐스트한다. 클러스터의 스케줄은 CH노드의 CH 시작 시간을 기준으로 구성되며 각 노드는 스케줄을 보고 해당 과정에서 자신의 단계에 맞는 전송시간을 알 수가 있다.(그림 3 참조)

CH ID	node ID(그림 4 참조)	schedule(그림 3 참조)
-------	------------------	-------------------

그림 2. CH signal 구조

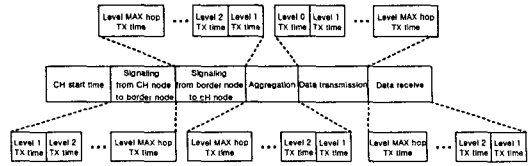


그림 3. 스케줄 구조

CH ID	남은 전력	CH와의 hop 수 (단계 : level)	CH로의 최단 경로 비용	count
-------	-------	----------------------------	------------------	-------

그림 4. 제안된 node ID 구조(클러스터 내의 node ID)

CH 신호를 받은 주변 노드는 그 신호에 포함된 정보를 정보 테이블(그림5.(a) 참조)에 기록하고 자신은 CH가 되는 것을 포기한다. 이 노드는 일정시간 동안 신호를 받아들이다가 그 중에서 CH와의 홉 수가 적고, 가장 가까운 노드의 CH ID를 선택한 다음, 선택한 CH노드에 대한 정보와 자신의 속성을 기반으로 하여 자신의 ID를 설정한다(그림 4의 node ID 구조 참조). ID가 결정된 노드가 초기에 결정된 CH와의 최대 홉 수 내에 있다면 일정시간이 지난 뒤 CH 신호를 다시 주변에 브로드캐스트한다. 이 신호를 받은 노드는 위와 같은 nodeID 설정 과정을 반복하게 된다.

III. 라우팅 경로 설정

3.1 CH를 목적으로 하는 인트라 클러스터 라우팅 경로 설정

CH 노드에서 외곽 노드로 신호를 보내며 각 노드의 ID를 설정할 때에는 ID에 CH로 향하는 최단 경로에 대한 비용을 기록하여 클러스터 내의 외곽 노드에서 CH 노드까지의 라우팅 경로를 결정하게 된다. 이때는 distributed Bellman-Ford(DBF)와 유사한 방법으로 최단 라우팅 경로를 설정하게 되는데 기존 DBF와 다른 점은 단계가 같거나 낮은 노드는 최단경로 설정 시 제외하여 센서 네트워크와 같이 노드 밀도가 높은 환경에서 경로와 상관없는 노드에 대한 불필요한 라우팅 계산을 줄일 수 있도록 한 점이다.

CH ID	주변 node ID	주변 노드로부터의 수신전력	CH로의 최단경로 비용
-------	------------	----------------	--------------

(a) 클러스터내의 각 노드가 가지고 있는 한 홉 거리의 노드들에 대한 정보 테이블

NC	next	NC의 최외곽노드	NC의 최외각	NC CH로부터 현재 노
CH ID	node ID	의 홉 수	노드 level	드까지의 최단경로 비용

(b) CH로부터 NC의 최외각 노드까지의 경로 노드가 가지고 있는 NC 라우팅 테이블

CH	CH	reverse path를 위	forward path를 위	홉	sequence
source	destination	한 이전 CH 노드	한 다음 CH 노드	수	#

(c) CH 노드가 관리하는 인트라 클러스터 라우팅 테이블

그림 5. 라우팅 테이블 구조

3.2 인접 클러스터를 목적지로 한 클러스터링의 라우팅 경로 설정

인접 클러스터와 연결되어 있는 클러스터의 외곽 노드는 클러스터 간의 라우팅을 하는데 있어 중요한 역할을 담당한다. 클러스터 외곽 노드는 클러스터링이 끝난 후 외부 클러스터에서 신호가 왔다면 다시 자신의 CH에게 그 클러스터에 관한 정보를 전송한다. 이 때 클러스터 외곽노드와 CH 노드 중간에서 이 정보를 전달해 주는 노드들은 그 신호에서 인접 클러스터(Neighbor cluster)에 관한 정보와 홉 수 등을 NC 라우팅 테이블(그림 5.(b) 참조)에 입력하여 CH 노드가 다른 클러스터와의 통신할 때 필요한 라우팅 경로를 구축한다. 이 때의 라우팅 경로는 인접 클러스터에 대한 정보를 가진 외곽 노드에서 CH 노드 방향으로 3.1와 같은 방법으로 구축된다.

3.3 sink 노드를 목적지로 하는 인터 클러스터 라우팅 설정

융합된 데이터를 sink 노드에게 전송하기 위한 클러스터간의 라우팅은 클러스터를 하나의 노드로 치환하여 기존 ad-hoc에서 쓰이는 알고리즘을 그대로 사용하였다. 사용되는 알고리즘에 따라서 클러스터 간의 라우팅 테이블의 구조는 달라진다(그림 5.(c) 참조). 인터 클러스터 라우팅 테이블은 CH 노드에서 관리를 하며 이 CH 노드를 통해서 sink 노드에게 전송하기 위한 다음 클러스터에 관한 정보를 알 수가 있다.

IV. 데이터 융합과정

클러스터 내의 노드들은 주기적으로 주변 정보들을 센싱을 하고 그 정보가 유효한 것인지를 따져 CH 노드에게 그 정보를 전달한다. 다중 홉 클러스터의 경우 클러스터링을 할 때 정한 최대 홉 수 거리에 있는 노드에서부터 CH 노드를 향해 데이터를 전송하며, 중간 노드들은 그 데이터와 자신의 데이터를 융합하며 CH 노드를 향해 전송을 하는 것이 효율적이다. 이 때 그 클러스터에 속한 노드들은 한번만 전송을 하여야 전력낭비를 줄일 수 있다. 전력 소비 측면에서 가장 좋은 데이터 융합방법은 세일즈맨 문제를 해결하는 방법과 유사하다[4]. 하지만, 클러스터 내의 모든 노드들이 그 클러스터에 관한 모든 정보를 알아야 하고 이런 융합경로를 찾는 것은 NP-complete 문제이기 때문에 적은 메모리를 갖고 느린 CPU를 지닌 센서 노드들이 처리하기에는 어렵다.

논문[4]에서 제안한 방법처럼 sink 노드에게 최종적으로 융합한 데이터를 보낼 노드가 주위 노드에게 token을 전달하여 자신으로의 데이터 융합 경로를 지정하는 방법이 있지만, 이 방식 역시 한 노드가 전체 네트워크에 대한 정보를 가지고 있어야 하고 token을 전달하는데 걸리는 지연 시간의 문제 때문에 효율적이지 못하다. 또한, 본 논문에서는 각 노드가 가지는 다른 노드에 대한

정보는 자신과 한 홉 거리에 있는 노드들로 제한이 되므로 위와 같은 방법으로는 데이터 융합을 할 수 없다. 따라서, 제한된 정보를 가지는 노드들의 효율적인 데이터 융합을 위해서 다음과 방법을 제안한다.

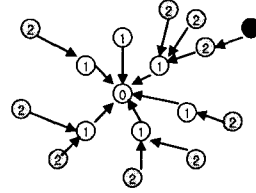


그림 6. level schedule에 따른 전달

4.1 level 스케줄링

데이터 융합을 할 때 클러스터의 최대 홉 수 거리에 있는 노드에서부터 CH 노드 쪽으로 전송을 한다. 예를 들면, 단계 D의 노드들은 자신이 보낼 수 있는 기간이 되었을 때 단계 D-1의 노드 중에서 적절한 노드(예: 최단경로)를 선택하여 데이터를 전송하고 단계 D-1의 노드는 그 노드들의 데이터와 자신의 데이터를 융합하여 단계 D-1이 전송 가능한 시간에 다시 단계 D-2의 노드에게 전송을 한다. 이런 방식으로 데이터 융합 기간 중에 클러스터 외부에서부터 각 단계에 해당하는 시간에 만 하위 단계의 노드에게 데이터 송신이 가능하게 하여, 각 노드가 단 한번 전송을 하여 그 클러스터 내부의 모든 데이터가 CH로 올 수 있도록 하는 방법을 제안한다. (그림 6 참조, 원안의 숫자는 CH로부터의 홉 수 의미)

4.2 수신 노드 분산

본 논문에서와 제안하는 방법과 같은 클러스터링을 하면 클러스터 내부의 인트라 라우팅 경로는 최소 홉을 기준으로 하여 최소 거리 라우팅 알고리즘으로 구축한 것과 같게 된다. 이러한 라우팅 경로로만 데이터를 전송한다면 경로에 있는 노드나, 수신이 집중되는 노드는 더 빨리 자신의 에너지를 소모 할 수밖에 없다. 본 논문의 3절에서와 같은 라우팅 설정을 하면 모든 노드는 자신의 CH로의 혹은 이웃 클러스터까지의 최소 비용 경로의 다음 홉을 알고 있기 때문에 전체 에너지 소모를 최소화 하기 위해 굳이 최소 비용의 다음 홉을 선택할 필요는 없다.

본 논문에서는 수신 노드 분산을 위해 최단 경로 방법 외에 2가지 방법을 사용한다. 하나는 목적지를 알고 있는 인접노드들 중 하나를 랜덤 선택하는 방법, 다른 하나는 자신과 인접한 노드 중 자신의 단계와 다른 단계에 있는 노드 수, 자신의 에너지, 최소 경비, 에너지를 cost로 하여 수신 노드를 선택하는 방법(load balance)이다. 두 가지 방법 모두 수신 노드를 선택함에 있어 송신노드의 단계를 기준으로 CH로 가야하는 패킷의 경우는 단계-1의 노드, 다른 인접 클러스터로 가야하는 경우는 단계+1의 노드, 또는 인접 클러스터의 외곽 노드 중에서 하나를 선택한다.

V. 성능 평가

제한한 다중 홉 클러스터링과 라우팅의 성능 평가를 위해 시뮬레이션을 수행하였다. 수신 노드 분산의 효과를 알아보기 위해 노드의 남은 에너지와 상관없이 다시 클러스터링을 하지 않고 주어진 라우팅 정보만을 가지고 통신을 할 경우를 가정하고 이 때 하나의 노드라도 가진 에너지를 모두 소모하였을 때까지의 생존 시간, 전력 소비량의 표준 편차 등을 측정하였다.

시뮬레이션에서 사용된 네트워크 환경은 100m x 100m에 50~400개(50씩 증가)의 노드가 랜덤하게 배치되어 있으며 sink 노드는 X= 10 m, Y= 10m 지역에 있다.

클러스터 내부에서의 라우팅은 본 논문에서 제시된 방법을 사용하였으며, 클러스터 간의 라우팅은 기존의 ad-hoc 네트워크에 사용되는 AODV[6] 알고리즘을 사용하였다.

파라메타	값
최대 전송 거리	20 m
클러스터 최대 홉 수	3 홉
초기 전력량	0.01 J
송신 소모 전력	50nJ/ bit
수신 소모 전력	$0.1\text{pJ/bit/m}^2 \times \text{distance}^2$
aggregation 주기	80 timeslot
수행 횟수	1000 회

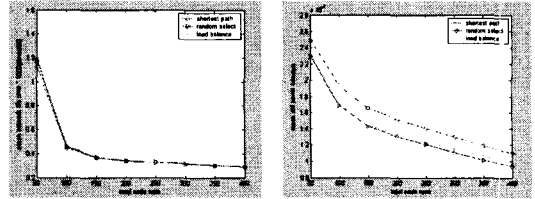
표 1. 시뮬레이션 파라메타

송, 수신 시 에너지가 소비되는 양은 [7]에 나오는 에너지 모델을 참고하였으며, 클러스터링과 라우팅 경로를 찾기 위해 사용되는 신호는 최대 전송 거리만큼 주변에 브로드캐스트를 하도록 하고, 데이터 융합을 위한 전송을 포함한 나머지 실제 데이터 전송에는 수신 노드와의 거리에 따라 전송 전력소비의 변형이 가능하도록 하고 무선랜의 RTS-CTS-data[9]방식을 변형하여 사용했다. 또 전송에 있어 충돌 및 에러는 없다고 가정한다. 시뮬레이션 환경에 대한 자세한 파라메타는 표 1에 나타내었다.

그림 7의 첫 번째 그래프는 각 수신 노드 분산 방법을 사용하였을 경우의 평균 네트워크 생존 시간을 측정하는 것이다. 수신 노드 분산 방법에 의한 네트워크 생존 시간의 차이는 거의 없다는 것을 알 수가 있다. 이는 모든 패킷은 일단 클러스터에 도착하면 다음 라우팅 경로를 알기위해 CH를 거쳐 가야만 하는 특성 때문에 생기는 문제이다. 결국 네트워크 생존 시간을 늘리기 위해서는 다시 클러스터링을 하여 다른 CH를 선택하여야만 한다[3].

그림 7. 노드 개수에 따른 평균 네트워크 생존 시간과 전력 소비의 표준편차

그림 7의 두 번째 그래프에서 전체 노드의 에너지



소비 표준 편차를 살펴보면 최소 비용 경로로 가는 경우가 다른 경우에 비해서 크다는 것을 알 수가 있었다. 이는 3절에 의해 구축된 라우팅 경로로 모든 패킷의 전송이 이루어져 그 경로를 거쳐 가는 노드들의 에너지 소모가 집중되기 때문이라고 볼 수 있다. 랜덤과 부하 분산 방법의 경우는 수신 노드를 확률적으로 분산하여 결정하기 때문에 최소 경비의 경비보다 여러 노드의 에너지를 골고루 소모하게 된다. sink 노드로 향하는 모든 패킷이 거쳐가는 클러스터에서 라우팅 정보를 얻기 위해 CH를 거쳐 가야만 하는 문제를 해결한다면 랜덤과 부하분산의 경우가 네트워크 생존 시간이 더 길어질 것이라고 예상된다.

VI. 결론

본 논문에서는 센서 네트워크에서 다중 홉 클러스터링과 지역 ID를 사용해서 효율적인 융합과 라우팅을 할 수 있는 방법을 제안하였다. 데이터의 전송방향과 같지 않은 단계의 노드에 대한 라우팅 연산을 고려하지 않으므로서 클러스터 안의 라우팅 계산횟수를 줄여 줄 수 있었고, 클러스터 내부의 라우팅 경로도 CH에서 정해진 홉 수 이상을 넘지 않도록 단순화할 수 있었다. 그리고, CH가 형성될 때 알려진 스케줄을 이용하여 데이터 융합을 할 때 클러스터 내의 노드가 각 한번씩 전송을 하여 불필요한 전송 에너지 낭비를 막을 수 있었다. 또 멀티 홉 클러스터에서 제안된 수신 노드 분산 알고리즘을 검증하기 위하여, 센서 노드가 균일하게 분포한 환경에서의 네트워크 생존 시간과 전력 소비의 표준 편차를 시뮬레이션을 통해 비교하였다. 이 방법은 수신 노드 선택의 편중을 분산시켜 전력소비가 분산되도록 할 수 있다.

참고문헌

- [1] L. F. Akyildiz and W. Su, "A Survey on Sensor Network," IEEE Communications Magazine, vol. 40, pp. 102-114, Aug. 2002.
- [2] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," Proc. of ACM MobiCom, pp. 174-85, 1999.

- [3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proc. of IEEE. Sys. Sci., pp. 1-10, Jan. 2000.
- [4] S. Lindsey and C. S. Raghavendra "PEGASIS: Power-efficient Gathering in Sensor Information System," Proc. of IEEE Aerospace Conference, vol. 3, pp. 1125-1130, Mar. 2002.
- [5] C. Y. Chiu, E. H. Kuang, and G. H. Chen "Stability Aware Cluster Routing Protocol for Mobile Ad-Hoc Networks," Proc. of Ninth International Conference Parallel and Distributed Systems., pp 17-20, Dec. 2002.
- [6] C. E. Perkins and E. M. Royer, "Ad-hoc On Demand Distance Vector Routing," Proc. of IEEE WMCSA, pp. 90-100, Feb. 1999.
- [7] W. Heinzelman, et. al., "Energy-scalable Algorithms and Protocols for Wireless Microsensor Network," Proc. of International Conference on Acoustics, Speech and Signal Processing, vol.6, pp. 3722-3725, Jun. 2000.
- [8] S. Singhal, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," Proc. of ACM MobiCom, vol. 6, pp. 3722-3725, Nov. 1998.
- [9] Phil Karn, "MACA - A New Channel Access Method for Packet Radio," Proc. of 9th Computer Networking Conference, pp. 134-140, 1990.