

# 유비쿼터스 센서 네트워크에서 효율적인 토폴로지 제어를 위한 클러스터 설정 기법\*

김중헌<sup>0</sup> 김은교 김동신 이원준  
고려대학교 컴퓨터학과  
wlee@korea.ac.kr

## Adaptive Cluster Configuration Scheme for Topology Control in Ubiquitous Sensor Networks

Joongheon Kim<sup>0</sup> Eunkyo Kim Dongshin Kim Wonjun Lee  
Dept. of Computer Science and Engineering, Korea University

### 요 약

센서 네트워크에서 에너지 효율성을 향상시키기 위한 방안 중 하나는 계층적 클러스터링 기법이다. 이 기법은 센서 노드가 이웃 노드와 통신할 필요 없이 오직 클러스터 헤드 노드와 통신하게 함으로써 에너지 소비량을 줄이게 된다. 하지만 이러한 클러스터링 기법은 클러스터 반경 크기에 따라 성능이 크게 좌우된다. 본 논문에서는 클러스터 헤드 노드의 잔류 에너지양에 따라 동적으로 클러스터 반경 크기를 조절함으로써 에너지 소비를 최적화시킬 수 있는 기법을 제안한다.

### 1. 서론 및 관련 연구

최근 센서 네트워크 [1] 기술을 유비쿼터스 환경에 접목하여 모든 사물이 서로 통신하며 정보를 주고받도록 하는 연구가 활발히 진행 중이다. 센서 네트워크는 수많은 센서들이 수집한 정보를 싱크(sink)라 불리는 노드에게 전송하는 방식으로 동작하며 각 센서는 에너지 제약적이며 기본적인 기능만을 수행할 수 있다. 따라서 센서 네트워크 설계 시 이러한 제약 사항들을 고려해야 한다.

에너지 효율성 증대시키기 위한 토폴로지 제어 기법 중 하나는 클러스터링 [2, 3]이다. 일반적으로 각 센서 노드는 이웃 노드를 통하여 수집한 정보를 싱크 노드로 전송한다. 하지만 클러스터링 기법을 이용하면 각 센서 노드는 이웃 노드가 아닌 자신이 포함된 지역의 클러스터 헤드 노드에게 정보를 전송하게 되고, 클러스터 헤드 노드는 데이터를 수집, 처리하여 싱크 노드에게 전송한다. 이 기법에서 클러스터의 반경이 작다면 클러스터 헤드의 에너지 소비량은 줄일 수 있으나 어떤 클러스터에도 포함되지 않는 센서들이 생길 수 있다. 반대의 경우에는 센서들이 모두 특정 클러스터에 속해 있으나 각 클러스터 헤드의 에너지 소비량은 증가한다. LEACH[4]는 클러스터링 기법을

사용하고 있는 대표적인 프로토콜로 에너지 부담을 줄이기 위하여 지역적인 클러스터 헤드를 임의로 설정한다. LEACH의 경우에는 클러스터 헤드를 구성하는 다른 종류의 노드는 존재하지 않고 센서 중에서 클러스터 헤드를 선출한다. 클러스터링 기법을 적용한 다른 프로토콜인 [5]은 지역 모니터링을 위한 센서 노드와 다른 종류의 노드를 클러스터 헤드로 사용하지만 클러스터 헤드 변경을 필요에 따라 조절하는 것이 아니라 임의의 값으로 고정해 두고 있다. 본 논문에서는 [5]에서 제안한 계층적 클러스터링을 기반으로, 센서 네트워크 전체를 커버하는 동시에 센서 네트워크 수명을 최대화 할 수 있는 클러스터 최적 반경 설정 기법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 클러스터 반경 제어 알고리즘을 NLP 기반 제어 방식과 벡터 연산 기반 제어 방식 두 가지로 나누어 제시한다. 3장에서는 수치적인 분석을 통해 제안한 방식의 성능을 평가한다. 마지막으로 4장에서 결론과 향후 연구 방향을 제시하고 본 논문의 끝을 맺는다.

### 2. 클러스터 반경 제어 알고리즘

제안하는 알고리즘은 크게 초기 단계, 클러스터 반경 제어 단계, 반복 단계로 나누어져 있다. 본 알고리즘은 다음과 같은 가정 하에 동작한다. 첫째, 본 논문에서 제안하는 센서 네트워크는 계층적 구조를 가지고 있다. 둘째, 싱크는 클러스터 헤드의

\* 본 연구는 한국학술진흥재단 선도연구자 사업(KRF - 2003 - 041 - D00509)의 지원으로 수행되었음.

위치를 알고 있다. 셋째, 센서는 밀도 높게 뿌려져 있다.

2.1 초기 단계

제안하는 알고리즘의 초기 단계에서는 센서 네트워크 상의 임의의 위치에 뿌려져 있는 클러스터 헤드들끼리 삼각형을 구성한다. 삼각형을 구성하는 목적은 그 삼각형 내부에 각 클러스터 헤드들이 가장 에너지 소모를 작게하는 반경을 설정할 수 있는 점(CRDP: Cluster Radius Decision Point)을 선출하여 삼각형의 각 꼭지점을 이루는 점들과 그 CRDP까지의 거리를 각 클러스터의 반경으로 책정하기 위함이다 [그림 1].

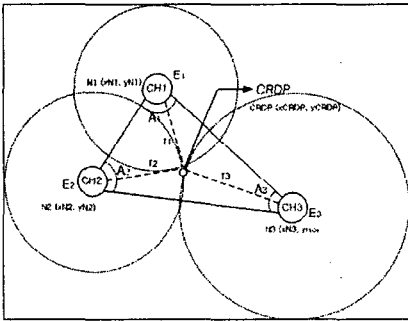


그림 1 클러스터 헤드 반경 제어의 개념도

삼각형을 구성하는 방법은 Delaunay 삼각법을 이용한다. Delaunay 삼각법을 이용하는 이유는 본 방법이 주어진 점들 사이에서 정삼각형에 가장 가까운 모양을 보장해 주기 때문에 각 클러스터 헤드의 공평한 에너지 소모를 최대한 가능하게 해 줄 수 있기 때문이다. 이는 궁극적으로 네트워크 생존시간을 늘릴 수 있다.

2.2 클러스터 반경 제어 단계

클러스터의 반경을 제어하는 것은 "2.1 초기 단계"에서 설정한 CRDP의 위치를 조절함으로써 가능하다. CRDP의 위치 조절은 비선형계획법 (Non-Linear Programming; NLP)을 기반으로 매번 CRDP를 결정할 때마다 에너지 소모를 고려하는 NLP 기반 제어 방식과 일차적으로 CRDP를 설정한 후에 그 후에는 클러스터 헤드들 간의 에너지 소모를 고려하여 CRDP의 벡터 좌표 값을 조절하는 벡터 연산 기반 제어 방식의 두 가지 방법이 있다.

2.2.1 NLP 기반 제어 방식

이 방법은 시스템 모델에서 에너지를 고려하여 반복 단계마다 비선형계획법 [6]을 수행하여 CRDP를 결정하는 방법이다.

위의 그림 1에서 보듯이 각도(A1, A2, A3)를 제외한 다른 변수 값들은 패킷에 실려 싱크에게 전달될 수 있는 정보이나 각도 값은 각각의 클러스터 헤드마다 서로의 위치 정보를 교환해야 하므로 각 클러스터끼리 분산적으로 결정하는 것 보다는 싱크에서 계산하는 것이 오버헤드가 적다. 즉, 싱크가 클러스터 헤드로부터 전달되는 패킷에 의해서 얻어지는 정보는 자신의

현재 에너지 값과 자신의 좌표 값이다. 각도 값은 싱크에서 계산하는 코사인 제 2법칙에 의해서 알 수 있다. 위에서 얻어진 값들로부터 에너지를 고려한 최소 에너지 사용 클러스터 반경 (r1, r2, r3) 설정 목적식은 다음과 같다.

$$\begin{aligned}
 & \text{Minimize :} \\
 & f(r_1, r_2, r_3, A_1, A_2, A_3, E_1, E_2, E_3) \\
 & = \sum_{i=1}^3 \left\{ \left( \frac{A_i}{2} \right) \left( \frac{E_i}{\sum_{j=1}^3 E_j} \right) \times r_i^2 \right\} - S_{\text{triangular}} \\
 & \text{s.t. } r_i^2 = (x_{\text{CRDP}} - x_i)^2 + (y_{\text{CRDP}} - y_i)^2
 \end{aligned} \tag{1}$$

위의 계산은 "2.3 반복 단계"에서 제시될 주기마다 수행한다.

2.2.2 벡터 연산 기반 제어 방식

비선형계획법에 따른 계산 복잡도를 줄이기 위하여 비선형계획법에 따른 CRDP의 계산은 첫 번째 단계에서만 수행하고 나머지 반복 단계에서는 정해진 CRDP의 위치를 벡터 기반으로 연산하여 움직이게 하고자하는 알고리즘이 벡터 연산 기반 제어 방식이다. 초기 상태에서 클러스터 헤드의 에너지는 100% 충전 상태를 가정하므로 비선형계획법에서는 에너지를 고려하지 않는다. 클러스터 반경 설정 목적식은 다음과 같다.

$$\begin{aligned}
 & \text{Minimize :} \\
 & f(r_1, r_2, r_3, A_1, A_2, A_3) \\
 & = \sum_{i=1}^3 \left\{ \left( \frac{A_i}{2} \right) \times r_i^2 \right\} - S_{\text{triangular}} \\
 & \text{s.t. } r_i^2 = (x_{\text{CRDP}} - x_i)^2 + (y_{\text{CRDP}} - y_i)^2
 \end{aligned} \tag{2}$$

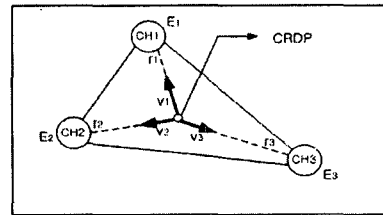


그림 2 CRDP의 벡터 변화 연산 개념도

위와 같은 방식으로 초기 CRDP의 값을 정한 후 "2.3 반복 단계"마다 각 클러스터 헤드들의 에너지 소모를 고려하여 CRDP의 위치를 이동시킨다. 이를 기반으로 각 클러스터의 반경을 조절함으로써 각 클러스터 헤드의 공평한 에너지 소모를 가능케 한다.

각 클러스터 헤드는 자신이 관장하고 있는 센서 노드의 개수에 따라 에너지 소모가 다르다. 그리고 각 클러스터 반경이 클 경우 자신이 관리해야하는 센서 노드의 개수가 많아지므로 에

너지 소모가 크다. 따라서 에너지 소모가 큰 쪽으로 CRDP의 위치를 이동시켜 주어 클러스터의 반경을 줄여주는 방법으로 삼각형의 세 점 내부에서 최적의 CRDP의 위치 설정한다. 위와 같이 CRDP의 위치를 설정하는 방정식은 그림 2의 개념도에 따라 모델링하면 다음과 같다.

$$CRDP_{k+1} = - \left( \frac{\sum_{i=1}^3 \frac{E_i}{\sum_{j=1}^3 E_j} \times \frac{v_i}{|v_i|} \right) \times CRDP_k \quad (3)$$

s. t.  $CRDP_k = (x_{CRDP_k}, y_{CRDP_k})$

위의 방정식을 통해 에너지 소모가 많은 클러스터 헤드가 있는 방향으로 CRDP가 이동한다.

### 2.3 반복 단계

클러스터 반경이 커질 경우에는 해당 클러스터 헤드가 관리해야 할 센서 노드 수가 많아지므로 클러스터 헤드의 에너지소모가 크다. 따라서 시간이 지남에 따라 에너지 소모가 많은 클러스터 헤드는 반경을 줄여주어야 한다. 이렇듯 시간이 지남에 따라 여러 발생할 수 있는 상황을 고려하여 이에 맞게 클러스터 헤드의 반경을 조절할 필요가 있다. 따라서 위에서 제시한 두 가지 방법을 반복수행 한다. 또한 이러한 반복 수행에 있어서 일정한 expire time을 설정하여 특정 이벤트가 존재하지 않더라도 일정시간이 지나 각각 클러스터 헤드의 에너지가 불균등해지는 상황을 방지한다.

### 3. 성능평가

본 논문에서 제안하는 알고리즘(Type A)과 각 클러스터 헤드가 이웃 클러스터 헤드를 포함할 수 있을 때까지 클러스터 반경을 확장한 경우(Type B)를 수리적으로 비교하면 다음과 같다.

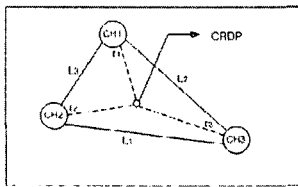


그림 3 Type A와 Type B의 비교 개념도

Type A의 경우에는  $r_1, r_2, r_3$  각각을 각 클러스터 헤드인  $CH_1, CH_2, CH_3$ 의 반경으로 설정한다. Type B의 경우에는  $CH_1, CH_2, CH_3$ 의 반경을  $L_1, L_2, L_3$ 를 각 클러스터 헤드의 반경으로 설정한다. Type A의 경우에 각 클러스터 헤드에서 생기는 클러스터 면적의 합은 다음과 같다.

$$S_{type_A} = r_1^2\pi + r_2^2\pi + r_3^2\pi = \pi(r_1^2 + r_2^2 + r_3^2) = \pi \sum_{i=1}^3 r_i^2 \quad (4)$$

Type B의 경우에 각 클러스터 헤드에서 생기는 클러스터 면적의 합은 다음과 같다.

$$S_{type_B} = L_1^2\pi + L_2^2\pi + L_3^2\pi = \pi(L_1^2 + L_2^2 + L_3^2) = \pi \sum_{i=1}^3 L_i^2 \quad (5)$$

$L_1, L_2, L_3$  값은 항상  $r_1, r_2, r_3$ 보다 큰 값을 가지므로,

$$\frac{S_{type_A}}{S_{type_B}} = \frac{\pi(r_1^2 + r_2^2 + r_3^2)}{\pi(L_1^2 + L_2^2 + L_3^2)} = \frac{(r_1^2 + r_2^2 + r_3^2)}{(L_1^2 + L_2^2 + L_3^2)} \quad (6)$$

$$= \frac{\sum_{i=1}^3 r_i^2}{\sum_{i=1}^3 L_i^2} = \sum_{i=1}^3 \left( \frac{r_i^2}{L_i^2} \right) = \sum_{i=1}^3 \left( \frac{r_i}{L_i} \right)^2 < 1$$

이다. 따라서 본 논문에서 제안하는 알고리즘이 항상 적은 클러스터 반경을 가지므로 하나의 클러스터 헤드에서 관장해야 하는 센서의 수가 적어 에너지 소모가 적으므로 효율적이다. 본 결과에 대한 시뮬레이션은 NS-2로 구현 중에 있다.

### 4. 결론 및 향후 연구

본 논문에서는 NLP 기법을 이용하여 에너지 효율적인 클러스터 반경 최적화 알고리즘을 제안하였다. 실험 결과 제안한 알고리즘을 적용한 경우 네트워크 전체를 모니터링 함과 동시에 네트워크 수명 연장 면에서 성능 향상을 보였다. 향후에는 클러스터 헤드가 이동성을 가지는 경우를 고려하여 연구를 진행해 나아갈 것이며, 이 경우에는 실제 Ad-hoc 네트워크와의 연동도 고려할 것이다.

### 참고문헌

- [1] I. F. Akyildiz, W. L. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Comm. Magz.*, vol. 40, no. 8, pp. 102-114, 2002.
- [2] N. Heo, and P. K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," *In Proc. of IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4576-4581, 2003.
- [3] G. Gupta and M. Younis, "Load-Balanced Clustering of Wireless Sensor Networks," *In Proc. of ICC'03*, vol. 3, pp. 1848-1852, 2003.
- [4] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Tran. on Wireless Communications*, vol. 1, no. 4, pp. 660-670, 2002.
- [5] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology Control for Wireless Sensor Networks," *In Proc. of ACM MobiCom'03*, 2003.
- [6] A. R. Conn, N. I. M. Gould, and P. L. Toint, "Trust-Region Methods," *SIAM*, 2000.