유비쿼터스 센서네트워크 기술 개발 동향

이 상 학　　김 대 환　　유 준 재

◈ 목 차 ◈

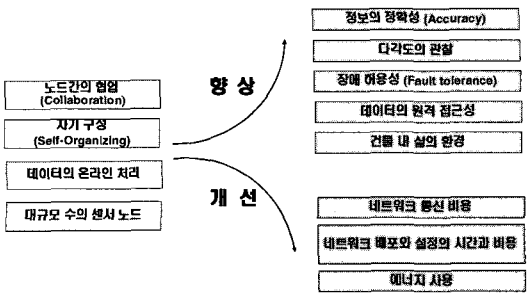
1. 서 론

저전력 무선 통신의 발전과 다기능, 저가의 스마트 센서는 원격에서 상태정보를 감지할 수 있는 센서네트워크의 실현이 가능하도록 하였다. 센서 네트워크는 광범위하게 설치되어 있는 유무선 네트워크 인프라에 상황인지를 위한 다양한 센싱 디바이스를 통합하여 감지된 환경데이터를 응용서비스 서버와 연동하는 기술이다. 이 기술을 사용하여 군사, 지능형 물류관리, 모바일 헬스케어, 실시간 시큐리티 분야 등에 적용을 시도하고 있다 [2].

센서 네트워크는 관찰하려는 지역의 내부나 그에 매우 근접한 거리에 수많은 센서 노드들을 조밀하게 배치하여 구성한다. 이 때 센서 노드의 위치는 미리 설정되지 않으며, 이는 접근이 불가능한 지역이나 재난지역에 무작위 배치가 가능하게 한다. 이를 위해 센서 네트워크의 알고리즘과 프로토콜이 자기 구성 (Self-organizing) 기능을 가져야 하며, 센서 노드간의 협력 처리 (cooperative processing) 기능이 보다 지능적으로 동작되어야 한다 [5]. 이 같은 무선 센서네트워크는 기존 유선 센서네트워크에 비해 그림 1과 같은 장점을 지닌다.

센서 네트워크의 응용이 구현되기 위해서는 무선 애드 혹(ad hoc) 네트워크 기술을 필요로 하지만 기존에 많은 무선 애드 혹(ad hoc) 네트워크 프로토콜과 알고리즘이 제안되었으나, 이들은 센서 네트워크의 독특한 특성과 응용분야의 요구사항에 적합하지 않다. 센서 네트워크가 기존 네트워크와 차별되는 특징은 다음과 같다 [1].

- 센서 네트워크의 노드 수는 기존 애드 혹(ad hoc) 네트워크의 노드 수보다 수십 배에서 수백 배 많을 수 있다.
- 센서 노드는 조밀하게 배치된다.
- 센서 노드는 고장 나기 쉽다.
- 센서 네트워크의 토폴로지(topology)는 매우 빈번하게 변경된다.
- 대부분의 애드 혹(ad hoc) 네트워크는 점 대 점(point-to-point) 통신에 기반 하지만, 센서 노드는 주로 브로드캐스트(broadcast) 통신을 이용한다.



(그림 1) 무선센서네트워크의 특징 및 장점

* 전자부품연구원

- 센서 노드는 전력, 계산 능력, 메모리에 제한 받는다.
- 센서 노드는 센서의 수와 오버헤드 때문에 전역(global) 식별자(ID)를 가질 수 없다.

센서네트워크의 성능은 크게 다음 세 가지 범주로 분류할 수 있다. ① 에너지 효율성, ② 감지 데이터의 정확도, ③ 서비스 품질 (Quality of Service) [3]. 이 중 가장 중요한 문제는 에너지 효율성이다. 에너지 효율성을 간단히 정의하면 네트워크가 동일한 에너지를 소모하며 발생한 이벤트의 손실 없이 얼마나 오랜 시간 동작하느냐이다 [4]. 본 논문에서는 네트워크의 에너지 효율성을 높일 수 있는 클러스터 기반의 네트워크 구조에 대한 연구와 플랫폼 개발 동향에 대해 기술하고 클러스터링 알고리즘을 소개한다.

본 논문은 다음과 같이 구성되어 있다. 2장은 센서네트워크의 클러스터링 기반 관련 연구에 대해, 3장은 플랫폼 개발 관련 연구에 대해 기술한다. 4장에서는 논문에서 제안한 클러스터링 알고리즘을 상세 기술하고, 5장은 성능평가에 대해 기술한다. 마지막으로 6장에서는 결론 및 향후 연구 방향으로 끝을 맺는다.

2. 클러스터링 관련 연구

클러스터 기반의 망의 형태 (Topology) 구성 및 유지 방법은 노드의 이동성이 많은 무선 애드 혹 네트워크에서 많은 연구가 이루어져 왔다. 클러스터링을 통해 얻을 수 있는 장점은 라우팅 설정 시 오버헤드를 줄이고 라우팅 테이블의 크기를 줄일 수 있고, 네트워크 망의 형태의 안정성을 확보할 수 있다. 이 밖에 매체 액세스 시 자원 관리나 대역폭 할당 등을 용이하게 하고 노드의 위치 관리나 송신 전력 관리를 가능하게 한다.

무선 센서네트워크에서는 위의 장점 이외에도 클러스터 헤드에서 클러스터 멤버 노드의 데이터를 병합할 수 있도록 하고 노드 증가에 따른 네트워크의 범위성 (scalability)을 용이하게 한다. 따라서 노드의 수가 애드 혹 네트워크에서 비해 수 백배에서 수 십만 배까지 많은 센서네트워크에서 망의 형태 관리와 데이터 병합을 위해 클러스터 기반의 구조를 기반으로 하는 것은 타당한 접근 방법이라 할 수 있다.

클러스터링 알고리즘의 기본적 요구사항은 클러스터링 후 모든 노드는 클러스터 헤드이거나 단 하나의 클러스터에 속해야 한다는 것이다. 이를 위해 필요로 하는 메시지와 시간의 오버헤드는 최소화되어야 하며 클러스터링의 목표를 만족해야 한다. 클러스터링의 목표는 안정적 망의 형태의 유지, 라우팅, 네트워크 효율성, 에너지 소비의 최소화 등이다.

클러스터 헤드 선정 문제는 그래프 컬러링, 셋 커버링 (Set Covering) 문제와 같이 최적 값을 구하기 위해서는 NP-hard 의 복잡도를 지니기 때문에 전역 최적 값을 구하기 매우 어렵다. 따라서 결정적 (deterministic) 알고리즘은 존재하지 않기 때문에 지금까지 제안된 알고리즘은 대부분 휴리스틱을 기반으로 하고 있다. 지금까지 제안된 클러스터 헤드 선정 알고리즘을 분류하면 세 가지 범주로 나눌 수 있다. 첫째, 노드의 식별자 (ID: identifier) 기반 클러스터 헤드 선정, 둘째, 노드의 연결성 (Connectivity) 기반 클러스터 헤드 선정, 셋째, 노드의 가중치 (Weight) 기반 클러스터 헤드 선정 알고리즘이다.

Baker와 Ephremides [6]는 클러스터 반경 내에서 가장 낮은 식별자가 클러스터 헤드가 되는 LCA (Linked Cluster Algorithm)를 Gerla와 Parekh [7, 8]는 가장 높은 연결도를 가진 노드가 클러스터 헤드가 되는 클러스터링 알고리즘을 제안하였다. Basagni *et al.* 은 노드의 클러스터 헤드 적합도를 가중치로 나타낸 DCA (Distributed Clustering Algorithm)와 DMAC (Distributed Mobility Adaptive Clustering Algorithm) [9]을 제안하였다.

기존 무선 애드 혹 네트워크는 망의 형태의 안정성을 확보하고 라우팅의 오버헤드를 줄이기 위해 클러스터를 구성한 반면 센서네트워크에서는 에너지 효율성이 중요한 설계 요소이기 때문에 노드의 잔여 에너지와 연결도 등이 클러스터 헤드 선정에 기준이 된다.

D. Estrin *et al.* [11] 은 두 레벨의 클러스터링 알고리즘을 제안했다. 모든 노드는 특정 계층과 대

응되고 이 계층은 통신 반경이 연계된다. 초기 모든 노드는 가장 낮은 레벨 0에서 시작하고 자신의 계층 레벨, 부모 노드 ID, 잔여 에너지를 알리고, 자신의 통신 반경과 비례하게 대기 시간 (Wait time)을 가진다. 대기 시간이 완료되면 승격 시간 (Promotion time)을 시작한다. 승격 시간은 노드의 잔여 에너지와 레벨 0의 노드로부터 수신한 노드의 수에 반비례한다. 이는 보다 밀집된 자리에 위치하고 잔여 에너지가 높은 노드의 승격 시간이 짧다. 승격 시간이 종료되면 노드는 레벨 1로 승격하고 이전에 수신한 레벨 0 노드들이 자식 노드의 후보가 된다. 이를 자식 후보들에게 알리고 이 메시지를 수신한 레벨 0 노드들은 가장 근접한 레벨 1의 노드를 부모 노드로 선택하고 승격 작업을 멈춘다. 승격 후 레벨 1의 노드는 보다 넓은 통신 반경으로 대기 시간을 시작한다. 대기 시간 종료 후 레벨 1 노드는 자신에게 포함되는 자식 노드가 없거나 잔여 에너지가 자식 노드들보다 일정 임계치 이하 (자식 노드 중 최대 에너지의 50%) 이면 레벨 0으로 낮아진다. 모든 레벨 0과 1 노드들은 주기적으로 대기 상태로 들어간다.

GAF [11], SPAN [12], ASCENT [13]는 멀티 홉 인프라에 최소 수의 노드만이 참여할 수 있도록 클러스터 헤드를 선정하였다. 이는 나머지 노드들은 무선 통신의 전원을 주기적으로 끄도록 하여 네트워크의 생존시간을 연장하기 위함이다.

센서네트워크의 구조의 특성상 다수의 네트워크 노드에서 싱크로 데이터를 전송하면 싱크에 인접한 노드들은 전송량이 많아져 다른 지역의 노드보다 먼저 에너지 소모가 이루어질 수 밖에 없다. LEACH (Low-Energy Adaptive Clustering Hierarchy) 프로토콜에서는 네트워크 노드간의 에너지 소모를 균등하게 하여 네트워크 생존시간을 최대화하기 위해 분산된 환경의 클러스터 기반의 네트워크 구조로 데이터 전송을 수행한다 [10]. 클러스터 헤드는 식 1의 확률 함수에 의해 결정된다.

$$P_i(t) = \begin{cases} \dfrac{k}{N - k * \left(r \bmod \dfrac{N}{k} \right)} & : C_i(t) = 1 \\ 0 & : C_i(t) = 0 \end{cases} \quad (1)$$

위 식에서 i는 노드의 식별자, t 는 시각, N은 전체 노드의 수, k는 클러스터의 수, r은 라운드를 나타낸다. $C_i(t)$ 는 최근 r mod (N/k) 라운드 동안 클러스터 헤드였다면 0이고, 아니라면 1이다.

HEED [14] 프로토콜은 클러스터 헤드의 선정을 개별 노드에서 분산 처리를 통해 결정하는 알고리즘을 제안했다. 노드의 잔여 에너지를 이용하는 헤드 선정 확률 함수는 식 2 와 같다.

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{\max}} \quad (2)$$

E_{max}는 노드의 초기 에너지, $E_{residual}$은 노드의 잔여 에너지, C_{prob}는 전체 네트워크 노드 중 클러스터 헤드 노드의 비율을 나타낸다. 이 밖에 클러스터 내의 통신비용을 두 번째 헤드 선정 기준값으로 이용하여 잔여 에너지 값이 같은 후보 노드가 있는 경우 헤드 선정을 위해 이용하였다. 통신비용은 이웃 노드의 근접성이나 클러스터의 밀집도이다. 이 알고리즘은 초기에 CH_{prob}와 절대값 P_{min} ($=10^{-4}$) 중 큰 값으로 시작하여 노드 자신의 CH_{prob}이1이 될 때까지 CH_{prob}를 2배씩 증가시키거나 1이된 이웃 노드로부터 메시지를 수신할 때까지 반복하고 이들이 클러스터 헤드가 되도록 한다. 이러한 방법은 클러스터의 크기에 관계없이 일정 시간 내에 알고리즘이 종료되도록 하며 이웃 노드의 위치를 고려하지 않아도 되는 장점을 지닌다.

ACE (Algorithm for Cluster Establishment) [15]는 노드의 연결도를 주 매개변수로 하여 고정된 횟수의 반복을 통해 최소 수의 클러스터 구성을 이루는 알고리즘을 제안하였다.

T. J. Kwon *et al.* [16] 은 클러스터 구성에 따른 오버헤드 (메시지와 시간) 가 없는 수동 클러스터링 (Passive Clustering) 을 제안하였다. 이는 이전의 클러스터링 구성 방법이 일정 시간 주기로 클러스터 구성을 위한 제어 메시지를 필요로 하므로 이를 없애기 위해 데이터 전송이 발생하면 이 때 클러스터 구성 메시지를 데이터에 부가 (Piggyback) 하는 방식으로 클러스터를 구성하고자 하였다.

3. 플랫폼 개발 동향

본 장에서는 다양한 접근을 통해 시도되고 있는 센서네트워크 하드웨어 플랫폼 가운데 대표적인 몇 가지를 살펴보도록 한다. 일반적인 센서네트워크의 배치 지역이 사람이 접근하기 힘든 열악한 환경의 지역이기 때문에 하드웨어의 물리적 크기, 안전성, 신뢰성, 저전력 등의 요구사항이 대두되며 이를 만족시키기 위한 노력이 이루어지고 있다.

3.1 UC Berkeley 의 'Smart Dust'

UC Berkeley 대학의 'Smart Dust' 프로젝트는 1 mm^3이내의 극초소형의 자율 센서네트워크를 만들기 위한 목표로 진행 되었다[17]. 이를 위해 네트워크, 소프트웨어, 시스템 디자인, MEMS 등의 기술들에 대한 연구가 수행되었다.

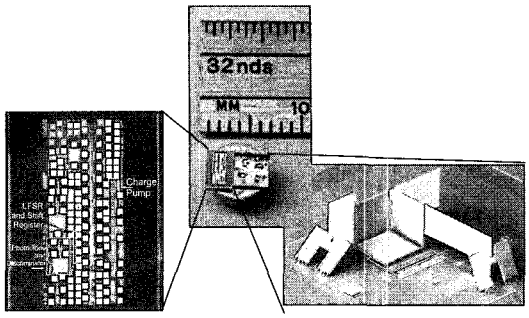
무선 주파수(Radio Frequency) 의 상용 부품을 이용한 플랫폼 외에 무선 광통신(Optical Communication)을 사용하고 MEMS(Micro-ElectroMechanical Systems)[2] 기술을 접목하여 눈에 보이지 않는 플랫폼을 만들었다. 이를 통해 2001 년에 개발된 결과 시제품이 그림 2와 같다.

위 플랫폼은 1Mbps 광 통신 수신기, 8bit 아날로그 디지털 변환기 (Analog to Digital Converter: ADC), 광 센서, 디지털 제어기 등이 0.15mm^3 크기의 CMOS (Complementary Metal Oxide Semiconductor)[3] ASIC 에 포함되어 있다.

센서네트워크에 대한 초기 연구에 선도적인 역할을 수행한 본 연구는 2001년 종료되었으며, 그 이후 Webs NEST, CENS 등의 프로젝트가 계속 연구를 수행 중이며, Kris Pister 교수는 Dust Inc. 를 설립하여 상용화하는 작업을 계속 중이다.

2) MEMS는 반도체 칩에 내장된 센서, 밸브, 기어, 반사경, 그리고 구동기 등과 같은 아주 작은 기계장치와 컴퓨터를 결합하는 기술.

3) 인버터 (Inverter) 회로에 p-채널 트랜지스터와 n-채널 트랜지스터를 같이 구성하여 동작 속도(연산속도)는 늦지만 소비 전력이 아주 작은 반도체. 포켓 계산기나 손목시계 등의 휴대용 제품에 많이 사용된다.



(그림 2) UC Berkeley의 Micro Mote

3.2 Crossbow사의 MICAx

Crossbow [18]사는 UC Berkeley 대학의 센서네트워크 운영체계인 TinyOS[19] 와 Mote를 기반으로 센서네트워크의 연구개발 및 애플리케이션을 위한 MICA 시리즈의 상용 플랫폼을 제공하고 있다.

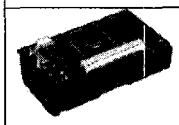
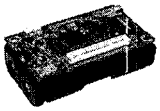
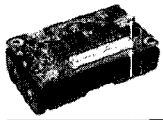
TinyOS는 UC Berkeley 대학에서 초기 연구를 진행하여 현재는 오픈 소스 프로젝트로 계속 발전되고 있다. 센서네트워크를 위해 컴포넌트(Component) 기반의 프로그래밍 언어인 NesC를 기반으로 태스크(Task)의 동시(Concurrency) 작업 지원에 초점을 두고 있다. 이를 위해 하위 레벨의 메시지 전달과 실행 배정(Dispatching) 프리미티브(Primitive)를 제공한다. TinyOS가 탑재되는 플랫폼의 프로세서나 메모리가 극히 제한적이기 때문에 코드 사이를 최소화하며 기능 동작을 수행한다.

Berkeley Mote 를 토대로 발전되어 온 Crossbow의 Mote는 현재 MICA, MICA2, MICA2DOT, MICAz 등의 여러 유형의 플랫폼이 개발된 상태이다. 그 외에 각종 센서 보드와 네트워크 게이트웨이(Gateway) 플랫폼 등이 있다. 표 1 은 MICA 시리즈의 플랫폼을 정리한 내용이다.

이들은 저가의 8bit 프로세서를 사용하고 주파수 사용 문제를 피하기 위해 ISM (Industrial Scientific and Medical)[4] 대역의 무선 통신을 위한 송수신기

4) 공업용, 과학용, 의료용 등으로 사용하는 무선통신 대역. 사용 주파수에 관한 제한은 없지만 국제 전기 통신 협약에 부속된 전파 규칙에 의한 주파수대가 지정되어 있고,

(표 1) Crossbow의 MICA 시리즈 플랫폼

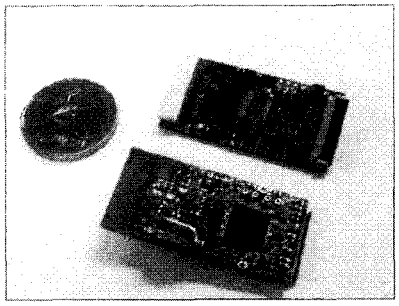
사 진	이 름	주파수 범위 (MHz)	라디오 송수신기
	MICA	902~928	RFM TR1000
		433~435	
	MICA2	868~870; 902~928	Chipcon CC1000
		433~435	
		315~316	
	MICA2DOT	868~870; 902~928	Chipcon CC1000
		433~435	
		315~316	
	MICAz	2400~ 4283.5	Chipcon CC2420 (IEEE 802.15.4 호환)

를 부착하였다. 가장 최근에 출시될 예정인 MICAz는 작년에 표준이 완료되어 무선 센서네트워크에 적용이 기대되는 IEEE 802.15.4 [20] 표준과 호환하는 MICAz이다. 현재까지 실제 애플리케이션에 적용되기에는 높은 가격과 배터리의 사용시간의 제약이 심하지만 전세계 연구개발 분야의 활용도는 매우 높다.

3.3 전자부품연구원의 TIP

국내 전자부품연구원(Korea Electronics Technology Institute: KETI)에서 무선 센서네트워크를 위한 무선통신 칩셋, 플랫폼, 네트워크 프로토콜, 시스템 소프트웨어, 응용 애플리케이션에 대한 연구를 수행 중이다. 무선통신 칩셋 설계와 병행하여 상용 부품을 사용한 플랫폼 시제품인 Tiny Interface for Physical World(TIP)을 개발하였다 [21].

개발 완료된 플랫폼의 사진은 그림 3와 같다. 주



(그림 3) 전자부품연구원의 TIP 플랫폼

요 부품으로는 8MIPS 저전력 프로세서, 3V 배터리 전원부, 914~915MHz 대역의 최대 56Kbps의 무선 통신 송수신기, 호스트 혹은 콘솔 인터페이스를 위한 시리얼, JTAG 포트, 그리고 온도, 습도, 조도 센서가 탑재되어 있다.

향후 무선통신 칩셋의 개발이 완료되면 현재 사용 중인 상용 부품을 대체할 예정이며, 네트워크 프로토콜, 시스템 소프트웨어, 응용 애플리케이션에 대한 연구를 병행하고 있다.

현재 국내 무선 센서네트워크에 대한 연구는 미국, 일본 등에 비해 뒤쳐져 있으나 인터넷, 이동통신 등 보급률이 매우 높고 PC, PDA, 텔레매틱스 등의 단말기 분야도 경쟁력을 갖추고 있기 때문에 이를 적용하기 위한 인프라가 잘 갖추어진 상태이다. 특히 신기술의 적용을 통한 응용 서비스 창출이 매우 뛰어나기 때문에 활발한 기반기술 연구활동과 응용분야의 가치 창출을 이룬다면 향후 기술적 우위를 확보할 수 있을 것으로 예상된다.

4. 분산 클러스터링 알고리즘

본 논문에서 제안한 클러스터링 알고리즘은 노드의 잔여 에너지, 노드간 거리, 그리고 클러스터 내 노드의 수를 기반으로 클러스터 헤드를 결정하는 분산(Distributed) 처리 알고리즘이다.

노드의 잔여 에너지, 노드간 거리, 그리고 클러스터 멤버 수를 기반으로 하는 이유는 노드의 잔여에너지는 노드 간 에너지 소모의 균형을 이루기 위한 중

이 주파수대의 누설 전계 강도 (electric field strength) 는 제한이 설정되어 있지 않다.

요한 요소이기 때문에 클러스터 헤드 선정에 가장 큰 기준이 되고, 노드간 거리와 클러스터 멤버 수 역시 이전 절에서 살펴본 바와 같이 클러스터가 구성되었을 때 클러스터간 균형을 이루었는지를 평가할 수 있는 요소이기 때문이다.

네트워크 프로토콜은 클러스터링 단계와 데이터 전송 단계로 이루어지고, 클러스터링 단계 (T_{CP}) 는 초기화 (Init), 병합 (Merge), 분할 (Partition) 단계로 구성되고 이후 데이터 전송 단계 (T_{DT}) 가 이어진다.

초기화 단계에서 각 노드는 확률 함수 식 3으로 클러스터 헤드를 스스로 결정한다.

$$P_i(t) = \frac{E_i(t)}{\sum_{j=1(j\in N(i))}^{n} E_j(t)} \times \frac{n}{N} \times k \qquad (3)$$

$E_i(t)$ 는 노드 i의 잔여 에너지, n 은 클러스터 반경 내의 이웃 노드의 수, k 는 목표 클러스터의 수이다. $P_i(t)$가 (0, 1) 사이의 임의 난수보다 크면, 노드 i는 클러스터 헤드 후보이다. 클러스터 헤드 후보는 이를 이웃 노드에게 공지하고 센서노드는 가장 근접한 클러스터 헤드 후보에 결합한다. 클러스터 헤드 후보로부터 메시지를 수신하지 못한 센서 노드는 스스로 클러스터 헤드 후보가 된다. 클러스터 헤드 후보는 멤버 노드들의 클러스터 내 비용값을 식 4와 같이 계산한다.

$$D_j = \sum_{k\in Cluster(i)} ((x_j - x_k)(x_j - x_k) + (y_j - y_k)(y_j - y_k))$$
$$\arg\min_{j\in Cluster(i)} D_j \qquad (4)$$

클러스터 헤드 후보는 멤버 중 비용값이 가장 작은 노드를 새로운 클러스터 헤드로 승급한다. 그림 4는 초기화 단계의 유사부호이다.

확률적 클러스터 헤드 선정은 목표 클러스터와 잘 분산되는 것을 보장하지 못하기 때문에 병합과 분할 단계를 거친다. 초기화 단계 이후 클러스터의 멤

```
PROCEDURE   Init ;
begin
   send E_i in Cluster Radius
   receive E_j in Cluster Radius
   P_i(t) = E_i / (sum_{j=1}^{n} E_j) × n/N × k ;
   if (P_i(t) > random(0,1 )) then begin
      CH_CANDIDA  TE = TRUE;
      send ADV_CH_CAN  DIDATE( i) in Cluster Radius
      receive REQ_JOIN( j,i) in Cluster Radius
      Cluster( i) = Cluster( i) ∪ { j}
      select min arg j∈Cluster (i) Intra _ Cluster _ Cost j
      if (i != j) then begin
         send PROMOTE_CH  _CANDIDATE (i, j) to j
      end
   else
      CH_CANDIDA  TE = FALSE;
      receive ADV_CH_CAN  DIDATE( j) in Cluster Radius
      CH( i) = CH( i) ∪ { j}
      if (CH( i) != φ) then begin
         send REQ_JOIN( i, j, E_i) to min arg j∈CH (i) D_{i,j}
      else
         CH_CANDIDA  TE = TRUE;
      end
   end
end;
```

(그림 4) 초기화 단계의 유사부호

```
PROCEDURE Merge;
begin
   if (CH_CANDIDATE = TRUE ∧ |Cluster(i)| < Thresh_Lower) then begin
      send REQ_MERGE(i)
      receive ACK_MERGE(j,i)
      CH(i) = CH(i) ∪ {j}
      if (CH(i) != φ) then begin
         CH_CANDIDATE = FALSE;
         send REQ_CH_MERGE(i, j, Cluster(i)) to min arg_{j∈CH(i)} D_{i,j}
      else
         if (|Cluster(i)| == φ) then begin
            if (Pi(t)> random(0,1)) then begin
               CH_CANDIDATE = TRUE;
            else
               CH_CANDIDATE = FALSE;
            end
         else
            if (0.5> random(0,1)) then begin
               CH_CANDIDATE = TRUE;
            else
               CH_CANDIDATE = FALSE;
            end
      end
   end
end;
```

(그림 5) 병합 단계의 유사부호

```
PROCEDURE Partition;
begin
   if (CH_CANDIDATE = TRUE ∧ |Cluster(i)| > Thresh_Upper) then begin
      j = FindBestPartitioner(i, Cluster(i));
      if (E_j > AVERAGE(E_j∈Cluster(i))) then begin
         send PROMOTE_CH_CANDIDATE(i, j) to j
      end
   end
end;

PROCEDURE FindBestPartitioner;
begin
   for j to |Cluster(i)| begin
      min arg_j ||Cluster(i)| - |Cluster(j)||
   end
   if (|Cluster(i)| < Thresh_Lower ∧ |Cluster(j)| < Thresh_Lower) then begin
      return j
   end
end;
```

(그림 6) 분할 단계의 유사부호

버 수가 하한 임계치 ($Thresh_{Lower}$) 이하이면 병합을 수행한다. 병합을 수행해야 하는 클러스터 헤드 노드는 병합 요청 메시지 (REQ_MERGE) 를 보내고 이를 수신한 병합을 수행하지 않는 클러스터 헤드는 응답 메시지 (ACK_MERGE) 를 보낸다. 이후 병합 메시지 (REQ_CH_MERGE) 를 보내 병합을 완료한다.

만일 응답 메시지를 수신하지 못했다면, 두 경우로 나누어 분기한다. 멤버 노드가 없는 경우, 이는 초기화 단계에서 모든 노드의 $P_i(t)$가 임의 난수보다 작은 경우이므로 (0, 1) 사이의 난수를 다시 발생 해 클러스터 헤드 여부를 재결정한다. 멤버 노드가 있는 경우, 확률값 $P_i(t)$ 를 0.5로 고정하여 클러스터 헤드를 재결정한다. 이는 모든 클러스터의 멤버 수가 하한 임계치 미만인 경우이다. 그림 5 는 병합 단계의 유사 부호이다.

병합 단계 후 멤버 노드의 수가 상한 임계치 ($Thresh_{Upper}$) 이상이면 클러스터의 분할을 수행한다. 클러스터 헤드는 멤버 노드 중 가장 균등하게 분할할 수 있는 새로운 클러스터 헤드를 선정한다. 추가의 클러스터 헤드 선정 함수는 식 5와 같다.

$$CH_{PART} = \min \arg_{j \in cluster(i)} \| Cluster(i) | - | Cluster(j) \| \quad (5)$$

$|Cluster(i)|$는 클러스터 i의 멤버 수이다. 만일 분할한 클러스터의 수가 하한임계치 이하이면 분할을 수행하지 않는다. 최악의 경우 네트워크에 클러스터가 하나만 있을 경우 k개의 클러스터를 생성하기 위해서는 $log_2 k$ 의 반복을 수행한다. 그림 6은 분할 단계의 유사부호이다.

초기화, 병합, 분할 단계를 거치는 분산 클러스터링 알고리즘은 N개의 노드로 구성된 네트워크에서 k 개의 클러스터가 잘 분산되는 것을 보장한다. 다음 장에서 실험을 통해 이를 보인다.

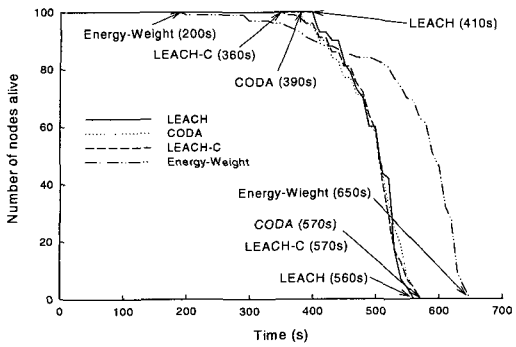
(표 2) 시뮬레이션 매개 변수

Parameter	Value
Network grid	(0, 0) × (100, 100)
Sink	(50, 175)
Threshold distance (dcrossover)	87 m
εelec	50 nJ/bit
εfriss amp (<dcrossover)	10 pJ/bit/m2
εtwo ray amp (≥ dcrossover)	0.0013 pJ/bit/m4
εaggregation	5 nJ/bit
Data packet size	500 bytes
Packet header size	25 bytes
Initial energy	2 J
Number of nodes (N)	100
Number of clusters (k)	5
ThreshLower	10 (N/k - (N/k)/2)
ThreshUpper	30 (N/k + (N/k)/2)

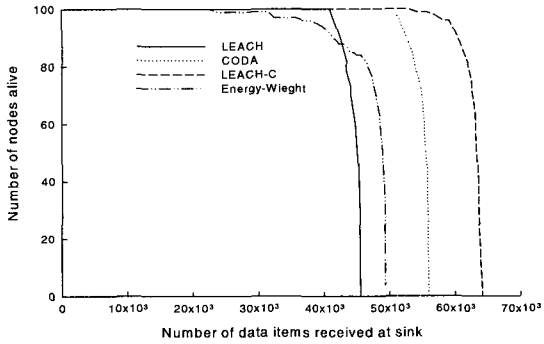
5. 성능평가

이번 장에서는 제안한 알고리즘의 성능 평가 결과를 보인다. 시뮬레이션 환경은 [10]의 매개 변수를 동일하게 사용한다. 이는 표 2와 같다. 100개의 노드로 구성된 네트워크에서 5개의 클러스터 구성을 목표로 한다. 클러스터 구성 후에는 센서노드는 클러스터 헤드로 데이터를 전송하고 모든 멤버 노드로부터 데이터를 수신한 클러스터 헤드는 데이터를 병합한 후 결과값을 싱크로 전송한다.

주요 측정값은 네트워크의 생존시간과 데이터 전송률이다. 네트워크 생존시간은 센서네트워크의 응용



(그림 7) 시간 대비 생존 노드의 수



(그림 8) 수신한 데이터 대비 생존 노드의 수

에 따라 FND (First Node Die)와 LND (Last Node Die)로 구분할 수 있으며, 데이터 전송률은 소비한 에너지 대비 싱크에서 수신한 데이터 량이다.

기존에 제안된 LEACH, LEACH-C, 잔여 에너지를 가중치로 계산한 알고리즘과 비교해 보았다.

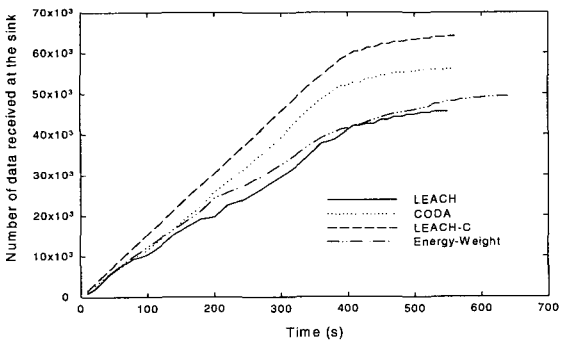
그림 7과 그림 8은 시간 대비 생존 노드의 수와 수신한 데이터 대비 생존 노드의 수를 나타낸다. 본 논문에서 제안한 알고리즘을 CODA (Cluster-based Self-Organizing Data Aggregation) 라 명명하였다.

시간 대비 생존 노드의 수는 다른 클러스터링 알고리즘과 유사한 생존 시간을 보였으며, 효율성을 나타내는 데이터 대비 생존 노드의 수는 분산 클러스터링 중 가장 오랜 생존 시간을 보였다.

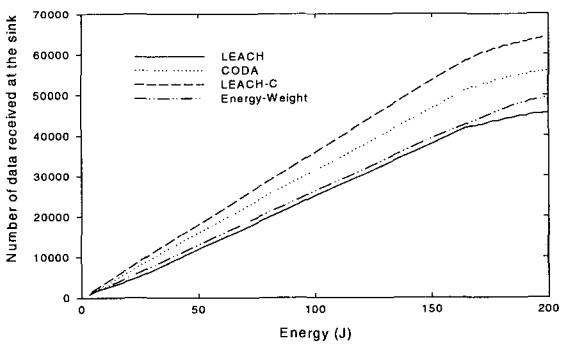
LEACH-C의 경우 분산 클러스터링과는 다르게 싱크에서 모든 노드의 잔여에너지 정보를 수집한 후 휴리스틱 알고리즘인 Simulated Annealing을 통해 근사 최적 클러스터링을 구성하기 때문에 보다 많은 데이터를 수집할 수 있었다. 감지 지역의 데이터의 정확도는 애플리케이션에 따라 다르게 정의될 수 있으므로 일반화할 수 없으므로 실험에서는 보다 많은 데이터를 수집하는 것이 보다 정확한 네트워크의 상황 정보를 인지하는 것으로 볼 수 있다.

그림 9와 10의 시간 대비 수신한 데이터 량과 소비한 에너지 대비 수신한 데이터 량 역시 분산 클러스터링 알고리즘 가운데 제안한 알고리즘이 가장 많은 데이터를 수집하였다.

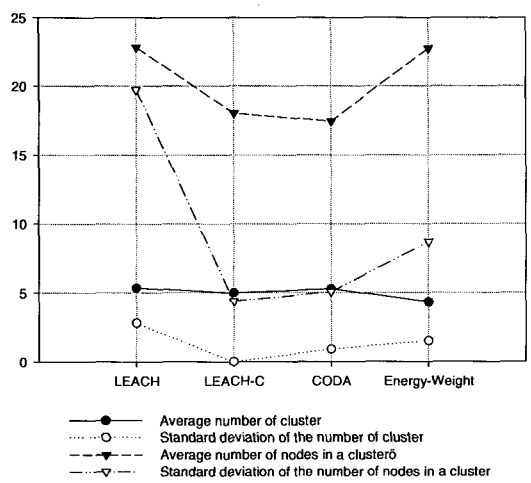
이 같은 알고리즘 간 성능 차이는 그림 11의 클러스터 수와 클러스터의 멤버 수에 대한 평균과 편차를 통해 알 수 있다. 라운드 마다 구성된 클러스터의 수와 이들의 멤버 수를 구한 것으로 100개의 노드를 5개의 클러스터로 구성하기 때문에 클러스터마다 약 20개의 멤버를 포함하는 것이 근사 최적



(그림 9) 시간 대비 수신한 데이터 량



(그림 10) 소비한 에너지 대비 수신한 데이터 량



(그림 11) 클러스터의 수의 평균과 표준편차 / 멤버 수의 평균과 표준편차

의 클러스터라 할 수 있다.

클러스터의 수와 멤버 수에 대한 평균은 비슷한 반면 편차는 LEACH-C와 CODA가 작다. 이는 확률적 접근 방법으로 라운드가 많아지면 평균에 근접하지만 매 라운드 마다 5개의 클러스터를 보장하지 못하기 때문에 발생한다.

따라서 논문에서 제안한 클러스터의 구성 시 병합과 분할을 통해 클러스터의 수에 대한 조절과 클러스터 내 비용값을 이용한 클러스터 헤드의 위치 변경은 목표로 하는 클러스터의 수에 근접하고 이들이 네트워크 전체에 잘 분포되도록 하는 것을 실험을 통해 알 수 있었다.

6. 결 론

본 논문에서는 센서네트워크의 망의 형태 구성을 위한 클러스터링 알고리즘에 대한 연구와 플랫폼 개발 동향에 대해 살펴보고, 에너지 효율성 증대를 위한 분산 클러스터링 알고리즘을 제안하였다. 제안한 클러스터링 알고리즘은 클러스터 간 부하 균형을 위해 멤버 노드 수를 균등하게 되도록 하였다. 클러스터링은 초기화, 병합, 분할 단계를 거치며 초기화 단계에서 노드의 잔여 에너지를 기반으로 확률적 클러스터 헤드를 선정하고 클러스터 내 통신 비용으로 클러스터 헤드를 변경한다. 이 후 클러스터 멤버 수를 기초로 클러스터의 병합과 분할 단계를 거쳐 클러스터 구성을 완료한다.

실험을 통한 알고리즘의 성능 평가에서 기존에 제안된 방법들에 비해 보다 오랜 생존시간과 데이터 전송률을 보였다. 지금까지 수행한 연구에서 에너지 효율성을 개선하였으나 이는 단일 홉 전송을 기반으로 하였으며 향후 다중 홉 라우팅과의 연계를 통한 클러스터 구성에 대한 연구와 개발 중인 실제 플랫폼 상에 구현하는 작업을 필요로 한다.

참 고 문 헌

[1] [aky02] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", Communication Magazine IEEE. vol. 40, no. 8, pp. 102-114, Aug. 2002.

[2] National Research Council, Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers, National Academy Press, 2001.

[3] [sai04] H. Saito and H. Minimi, "Performance Issues and Network Design for Sensor Networks", IEICE Trans. On Commun., vol. E87-B, no. 2, pp. 294-301, Feb. 2004.

[4] [gen03] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, "Networking Issues in Wireless Sensor Networks", Journal of Parallel and Distributed Computing(JPDC), Special issue on Frontier in Distributed Sensor Networks, Dec. 2003.

[5] [est99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks", Proc. of the 5th Annual International Conference on Mobile computing and Networks (MobiCOM '99), Seattle, WA.., pp. 263-270,

Aug. 1999.

[6] [bak81] D. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm", IEEE Transactions on Communications, vol. 29, no. 11, pp. 1694 - 1701, Nov 1981.

[7] [lin97] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks", IEEE Journal on Selected Areas in Communications, vol. 15, no. 7, pp. 1265-1275, Sept. 1997.

[8] [ger95] M. Gerla and J. Tsai, "Multicluster, mobile, multimedia radio networks", ACM-Baltzer J. Wireless Networks, vol. 1, no. 3, pp. 255-265, 1995.

[9] [bas99] S. Basagni, "Distributed clustering for ad hoc networks", Proc. of International Symposium on Parallel Architectures, Algorithms and Networks, pp. 310-315, June 1999.

[10] [hei02] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transaction on Wireless Communications, vol. 1, no. 4, pp. 660-670, Oct. 2002.

[11] [xu01] Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing", in Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Rome, Italy, pp. 70-84, July 2001.

[12] [che02] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", ACM Wireless Networks, vol. 8, no. 5, September 2002.

[13] [cer02] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies", in Proceedings of IEEE INFOCOM, New York, NY, June 2002.

[14] [you04] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", IEEE INFOCOM 2004, Mar. 2004.

[15] [cha04] H. Chan and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation", In 2004 European Workshop on Sensor Networks, pp154-171, Aug. 2004.

[16] [kwo03] T. J. Kwon, M. Gerla, V. K. Varma, M. Barton, T. R. Hsing, "Efficient Flooding with Passive Clustering-An Overhead-Free Selective Forward Mechanism for Ad Hoc/Sensor Networks", Proceedings of the IEEE, vol. 91, no. 8, pp. 1210-1220, Aug. 2003.

[17] http://robotics.eecs.berkeley.edu/~pister/SmartDust/

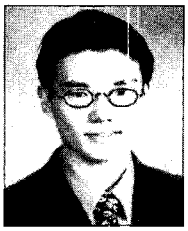
[18] http://www.xbow.com

[19] http://www.tinyos.net

[20] http://www.ieee802.org/15/pub/TG4.html

[21] http://tinyos.keti.re.kr

◐ 저 자 소 개 ◑



이 상 학

1993년 전주대학교 수학과 학사
1997년 경희대학교 대학원 컴퓨터공학과 석사
2000년 경희대학교 대학원 컴퓨터공학과 박사 수료
2000년~현재 전자부품연구원 유비쿼터스컴퓨팅연구센터 선임연구원

김 대 환

1991년 명지대학교 전자공학과 학사
1993년 명지대학교 대학원 전자공학과 석사
1993년~현재 전자부품연구원 유비쿼터스컴퓨팅연구센터 책임연구원

유 준 재

1982년 경북대학교 전자공학과 학사
1997년 아주대학교 대학원 컴퓨터공학과 석사
2004년 충북대학교 대학원 컴퓨터공학과 박사
1992년~현재 전자부품연구원 유비쿼터스컴퓨팅연구센터장