

센서 네트워크의 노드 응용 레벨에서 에너지 소모 계획

모델을 위한 연구*

조용만[○] 이승재 김창화 김상경 강태원

강릉대학교 컴퓨터공학과

{ymcho[○], silveree, kch, skkim98, twkang}@kangnung.ac.kr

A Study on Energy Consumption Scheduling Algorithm of Sensor Networks at the Node's Application Level

Yongman Cho[○], Seungjae Lee, Changhwa Kim, Sangkyung Kim, Tae-Won Kang
Department of Computer Science and Engineering, Kangnung National University

요 약

센서 네트워크에서 에너지의 제한은 이전의 다른 많은 네트워크 시스템과 구별되는 특징을 가지게 한다. 따라서 센서 네트워크에서의 가장 중요한 연구 주제는 에너지를 절약하도록 하는 것에 초점이 맞춰진다. 기존의 센서 네트워크에서는 주로 Network, Mac, Phy 계층에서 연구가 진행되었으며, 또한 사용자가 센서 네트워크의 Lifetime을 미리 정할 수 없었다. 하지만, 어떤 센서 네트워크에서는 어떤 특정한 시간까지 생존할 필요가 있을 것이다. 이 논문에서는 미리 정해진 Lifetime을 보장하는 노드의 응용 수준에서의 에너지 소모 모델을 제공한다.

이와 같은 일을 하기위해서, 첫 번째로, 센서 네트워크에서의 응용을 6가지로 구분하고, 응용에서 필요로 되는 5가지 연산을 정의했다. 두 번째로 이러한 5가지의 연산을 포함하는 에너지 소모 모델을 만들었으며, 이것을 가지고 각각의 연산에서 소모되는 에너지를 통계적인 기법을 통해서 계산한다. 마지막으로 노드가 미리 정해진 시간까지 생존하도록 각각의 노드에서 응용의 에너지 소모 계획을 다시 작성한다. 예를 들면, 응답주기와 센싱주기의 변경을 통하여 에너지의 소모를 줄일 수 있다. 결과적으로 이러한 일들은 노드가 정해진 시간까지 생존하도록 보장해 준다.

1. 서 론

최근 무선 통신 기술과 컴퓨팅 기술의 발달로 인하여 초소형의 센서 노드들이 개발되고 있다. 센서 노드들은 다양한 센싱 작업을 수행하는 센서와, 센싱되어진 데이터를 처리하기 위한 처리장치(Processing unit)와 센서 노드들 간의 단거리 무선 통신을 위한 무선 송수신 장치(Transceiver unit), 센서 노드의 전력을 공급하는 전력장치(Power unit) 등으로 구성되어 진다[1]. 센서 네트워크는 이전의 네트워크와 많은 부분에서 차이를 보이고 있는데, 한정된 CPU 속도, 메모리 크기, 파워, 그리고 대역폭 등을 포함한 많은 제약사항을 가진 분산된 컴퓨팅 플랫폼으로 생각할 수 있다. 센서 네트워크는 센서들과 구동장치들을 통해서 물리적인 환경과 매우 강하게 상호 작용하는 것이 기존의 네트워크와 다르다[2].

센서 네트워크의 가장 큰 제약점은 센서 노드들이 한정된 에너지를 가지고 있다는 것이다. 즉, 다수의 초소형센서 노드들이 매우 넓은 지역에 배치되어 있고, 이들 센서 노드들은 배터리로 에너지를 공급받는다. 따라서 센서 노드로 에너지를 지속적으로 공급하거나, 또는 배터리 재충전과 같은 방법으로의 에너지 공급이 현실적으로 현재의 기술로는 거

의 불가능하다. 그러므로 현재 대부분의 센서네트워크분야에서는 한정된 에너지를 극복하기위한 에너지 절약 기법을 연구하는데 많은 노력을 기울이고 있다.

지금까지의 에너지 절약연구는 최소의 에너지를 소모하는 라우팅 연구, 데이터 취합을 통하여 최소 전력을 소모하도록 하기위한 클러스터링 연구, 에너지 효율적인 MAC 연구, 에너지 소모를 줄이는 보고 주기 연구 등 센서네트워크 시스템의 모든 레벨에서 연구되어지고 있다. 이러한 연구의 기본 플랫폼은 다수의 센서 노드들이 특별한 네트워크 위상을 가지는 하나의 큰 네트워크 시스템으로 본다. 그리고 전체적인 네트워크 수준에서의 에너지 소비를 줄일 수 있는 다양한 방법들을 연구하였다. 따라서 전체 네트워크 시스템 단위에서는 매우 좋은 결과를 보이고 있다.

대부분의 연구가 센서 네트워크의 생존시간을 최대한 늘리는 것에 목적을 두었다면, 본 연구의 목적은 센서 네트워크도 일반적인 제품처럼 내구연한을 가지도록, 생존시간을 미리 정하고 그 정해진 시간동안 생존하기 위한 에너지 소모계획을 만드는 것이다. 이 소모계획은 중앙집중식이 아니라 개별 노드 자체에서 수행되도록 한다. 이는 전체 네트워크 시스템에서 추가적인 메시지 발생을 막아 에너지의 낭비를 줄인다. 다음은 센서 네트워크에서 에너지 소모를 줄이기 위한 기존의 연구에 대해서는 2장에서 살펴보고, 3장에서는 본 논문의 목적을 성취하기위한 기본 전략에 대해서 기술한다. 4장에서는 이러한 전략을 만족하는 에너지 소모 알고리즘을 소개한다. 마지막으로 5장에서 본 논문의 결론을 짓는다.

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2006-C1090-0603-0044)

2. 관련 연구

전체 센서 네트워크에서 에너지 소모를 최소화하기 위한 방법으로는 센서 네트워크의 구조를 계층으로 나누고 각 계층별 에너지 효율적인 프로토콜을 연구하는 방법과 각 계층의 모든 것을 종합적으로 고려한 크로스 레이어 디자인에 관한 연구로 구분할 수 있다.

먼저 각 계층별로 에너지 효율적인 프로토콜 연구를 살펴보면 다음과 같다. 에너지 효율적인 MAC연구로는 일반적으로 노드들이 통신에 참여하지 않는 동안에 송/수신 장비들의 전력을 차단하여 에너지 소모를 최소화 하도록 함으로서 센서 네트워크의 생존시간을 높이는 방법들이다[3][4]. 이 연구들은 다른 계층과의 연계 없이 MAC계층 내에서만 에너지 효율을 어떻게 높일 것인지에 관한 연구이다. 에너지 효율적인 라우팅으로는 데이터의 흐름을 분산하여 특정 노드가 집중적으로 에너지를 소모하여 전체적인 네트워크의 에너지 불균형을 해소하는 방향으로 라우팅 경로를 찾고, 데이터를 전달하는데 있어서 필요한 에너지가 적은 경로로 전달이 이루어진다. 이런 라우팅 프로토콜의 예로 MMRE [5]는 라우팅 테이블에 있는 여러 노드의 현재 남아 있는 에너지를 살펴보고 가장 많은 에너지를 가진 노드 방향으로 경로를 선택한다. MTE[6]에서는 다음 노드까지 데이터를 전송하는데 드는 에너지를 계산하여 더 적은 양의 에너지를 소비하는 라우팅 경로를 이용한다. [7][8]에서는 이 두 가지 방식을 각각 적용해서는 최적의 라우팅 경로를 결정할 수 없음을 보이고 이 둘을 혼합한 알고리즘을 제안하였다. 그리고 위상을 조정함으로써 네트워크가 관리하는 영역 내에 필요한 최소한의 노드만 작동 시키고, 이벤트를 센싱할 수 있는 영역이 겹치는 노드를 번갈아 가면서 sleep시키거나 각자의 time table을 바탕으로 sleep/wake를 반복하면서 작동시간을 최소화한다[9][10].

크로스 레이어 기법으로는 다음과 같은 연구들이 있다. 먼저 데이터 어그리게이션(Data aggregation)은 주기성과 유사성을 가지는 취합 혹은 축약을 통한 망 자원 확보, 에너지 절약, 트래픽 매니지먼트의 용이성, QoS 처리용이 등의 이유로 센서 네트워크에서 중요성이 부각되고 있다[11]. 최근의 Object 탐지와 트래킹에 대한 Dynamic Cluster구조 [12]는 센서 네트워크상에서 이동성을 가진 커다란 범위의 continuous objects에 대한 boundary 정보로 물체의 움직임을 파악하기 위한 효율적이고 효과적인 구조를 제안하고 있다. LEACH[13] 프로토콜은 자기 구성과 제한된 전원 문제에 있어 유용한 해결 방안의 하나로, 일정한 수의 클러스터를 구성함으로써 자기 구성을 수행한다.

3. 에너지 소모 알고리즘을 위한 기본 전략

기존의 연구에서는 전체 센서노드들의 에너지가 고르게 소모하도록 중앙에서 개별 노드들의 에너지 정보를 체크하여 중앙에서 관리한다. 하지만 이러한 방법은 센서 네트워크에 부가적인 메시지 교환을 발생시키거나, 전송량을 증가시켜 많은 에너지가 낭비될 수 있다. 그래서 가능하면 개별 센서 노드에서 스스로 에너지 소모를 조절할 수 있는 방법이 매우 중요하다. 그러기 위해서는 노드에서 에너지를 소모하는 요인이 무엇인지부터 파악해야 한다. 그리고 기존의 연구에는 포함되지 않았던 내용으로서, 센서 네트워크의 생존시간(Life-time)을 미리 정하는 것이다. 이렇게 생존시간을 미리 정하게 되면 기존의 에너지 절약 방식처럼 모든 노

드들이 에너지를 공평하게 소모하고 있는지를 알기위해서 서로 정보교환을 할 필요성이 없어진다. 왜냐하면 이미 정해진 시간까지 각자의 개별노드에서 그 시간까지 살아가기 위한 절약을 개별적으로 수립하면 되기 때문이다.

이와 같은 에너지 소모 계획 모델을 세우기 위해서는 그림 1에서와 같은 기본적인 전략이 필요하다. 우선, 에너지를 소모하는 요인이 노드에서 실행되는 응용의 종류에 따라 다르다고 보고 응용을 전송주기와 센싱주기에 따라 먼저 구분하였다. 또한 이 응용들을 수행하는데 있어서 시간의 흐름에 따라 에너지 소모량이 다를 수 있는데, 이렇게 에너지의 소모가 다른 상태를 여러 개의 연산으로 구분하여 표현한다. 이렇게 식별된 연산은 각각의 응용에 따라 소모되는 에너지를 통계적인 방법을 통해서 구할 수 있고, 구해진 각 연산들의 주기를 변경함으로써 미리 정해진 센서 네트워크의 생존 시간까지 개별 노드들이 생존하게 할 수 있다. 결과적으로 제안한 알고리즘에 의해서 센서 네트워크가 정해진 시간까지 생존함을 보장할 수 있다. 다음의 절에서 이들 응용의 분류와 연산의 식별에 대해서 기술한다.

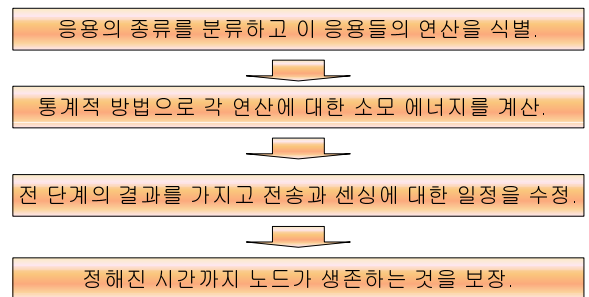


그림 1 에너지 소모 계획을 위한 전략.

3.1. 정해진 lifetime이 보장되는 센서 네트워크를 위한 응용 모델

센서 노드의 구조를 일반적으로 그림 2의 왼쪽에서 보는 바와 같이 크게 5가지 계층으로 볼 수 있으며, 각 계층별 에너지 절약의 프로토콜 접근이 가능하다고 볼 수 있다. 하지만, 이 논문에서는 응용계층에서의 접근을 시도한다. 먼저 응용계층에서의 접근을 위해서 응용계층에서 수행되는 응용을 전송 주기와 센싱주기에 따라서 6가지로 구분하였다.

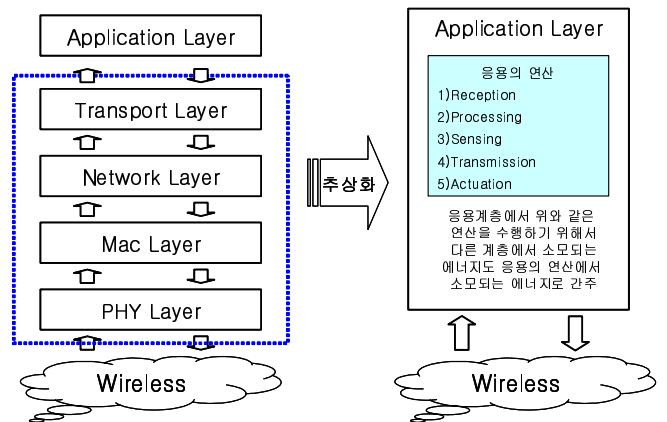


그림 2 센서 노드의 응용 계층에서의 에너지 소모에 대한 추상화

센서 네트워크의 메시지 전송 구조는 크게 단방향과 양방향으로 볼 수 있는데, 단방향 구조는 중앙의 컨트롤 센터에서 센서 노드로 어떠한 제어권도 가지고 있지 않고, 처음에

센서 노드에 초기화된 센싱 주기와 전송주기를 가지고 있다. 이에 반해 양방향 구조는 중앙의 컨트롤 센터로부터 지시사항을 받아서 센서 노드들이 처리하는 구조로 볼 수 있다. 이 센서 노드들이 받는 지시사항은 여러 가지가 있을 수 있다. 예를 들면, 센서 네트워크의 동기화를 위한 메시지, 클러스터링을 위한 메시지, 라우팅을 위한 메시지 또는 데이터의 전달을 하기위한 메시지가 있을 수 있고, 수순하게 센서 노드의 센싱 값을 알기 위한 쿼리 메시지가 있을 수 있다. 전자를 명령(command)라면, 후자는 질의(query)라고 볼 수 있다. 이 논문에서 명령은 반드시 센서 네트워크 시스템 유지를 위해 필요한 요소로서 노드에서 임의로 수행여부를 결정할 수 없는 반드시 수행할 임무로 본다. 하지만, 쿼리는 요청하는 중요도와 센서 노드의 현재 에너지 상태에 따라 조정이 가능한 것으로 본다. 이렇게 센서 노드에서 도착한 쿼리는 응용레벨에서 수행되어지므로 응용계층에서는 이 쿼리를 응용으로 본다. 다음은 중앙의 컨트롤 센터에서 센서 노드로 요청될 수 있는 쿼리(응용)를 전송(응답)주기와 센싱주기에 따라 나눈 분류이다.

- 전송(응답)주기를 갖는 쿼리(쿼리의 수행 기간이 정해져 있지 않음)
 - 메시지 형태 : query (interval=10m, sensor_type = temp)
 - 내용 : 10분마다 온도 정보를 노드가 살아있는 동안 전송하시오.
 - 쿼리의 수행 기간을 갖는 쿼리(전송 주기가 정해져 있지 않음)
 - 메시지 형태 : query (time(20070201:20071231), sensor_type = all)
 - 내용 : 모든 센싱 정보를 2007년 2월 1일에서 2007년 12월 31일까지 전송하시오.
 - Event Driven 쿼리(전송 주기를 가질 수 없고 쿼리의 수행 기간이 정해져 있지 않음)
 - 메시지 형태 : query (sensor_type = oxygen, threshold=5.0mL/L)
 - 내용 : 용존 산소가 5.0mL/L이상인 경우만 정보 전송하시오.
 - 한번만 전송하는 쿼리(쿼리의 전송 주기와 수행시간을 정할 수 없다.)
 - 메시지 형태 : query(sensor_type = all)
 - 내용 : 모든 센싱 데이터를 한번만 보내시오.
 - 전송 주기와 쿼리의 수행기간을 모두 갖는 쿼리
 - 메시지 형태 : query (time(20070201:20070630), interval=10m, sensor_type = temp)
 - 내용 : 2007년 2월 1일에서 2007년 06월 30일까지 10분마다 온도 데이터를 전송하시오.
 - 쿼리의 수행 기간을 갖는 Event Driven 쿼리
 - 메시지 형태 : query (time(20070201:20070630), sensor_type = oxygen, threshold=5.0mL/L)
 - 내용 : 용존 산소가 5.0mL/L이상인 경우만 데이터를 전송하시오.
- 위 분류에서 쿼리의 수행기간은 쿼리를 수신한 노드에서 쿼리의 생존시간을 의미한다. 즉 언제까지 도착한 쿼리에 대한 응답을 할 것인가를 말한다. 다음은 이렇게 분류된 응용들에서 에너지를 소모하는 연산에는 어떤 것들이 있는지 설명한다.

3.2 응용을 구성하는 연산

센서 노드에서 응용들이 수행될 때 에너지를 소모하는 몇

가지 연산들이 있다. 따라서 이러한 연산들을 정의하고, 노드가 정해진 Lifetime까지 생존하기 위해서 필요한 에너지를 줄이기 위해서 이러한 연산들을 조절해야 한다. 노드들이 응용 메시지를 수신해서 응답하기까지의 여러 상태들에 대해서 그림 3에 표현하였다. 이 그림에서 보이는 바와 같이 응용을 구성하는 연산을 다음의 5가지로 분류하여 설명할 수 있다.

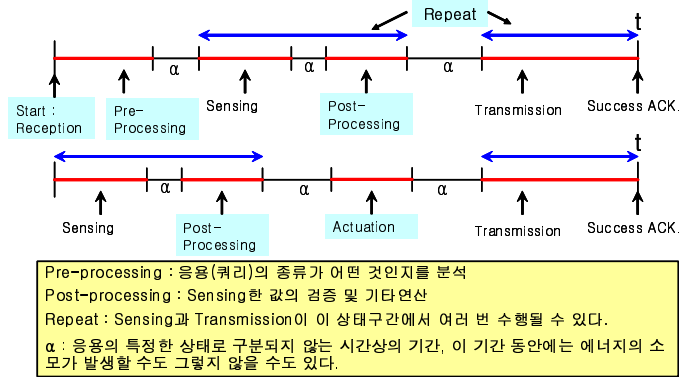


그림 3 센서 노드에서 응용의 수행에 있어서 시간흐름에 따라 변화되는 에너지 소모 상태를 연산으로 정의

- Transmission or Response(전송 또는 응답) : 센서 노드에서 센싱한 결과를 sink node나 control center로 보내기 위해서 라우팅 경로 상에 있는 이웃 노드로 데이터를 전송하기위해서 필요한 연산이다. 이 연산을 수행하는데에는 주로 Transmitter에서 소모되는 에너지가 주가 될 것이며, 응용 계층에서 구해진 메시지를 네트워크 레이어 및 Mac layer, phy Layer를 거치는데 소모되는 processing 비용은 이 연산에 종속되는 것으로 본다.
- Reception(수신) : 노드에서 쿼리를 수신한 경우에 필요한 수행하는 연산이다. 이 연산의 경우 receiver에서 주로 에너지를 소모할 것이며, 이것이 receiver에서 윌 단계를 거쳐 응용계층까지 오는데 들이간 processing의 에너지도 이 연산을 수행하는데 소모된 에너지로 본다. 이 연산은 거의 노드에서 control 되지 않을 것으로 본다.
- Sensing(센싱) : 노드에서 센서들을 통해서 외부의 상황을 센싱하여 데이터로 변환되는 과정에 필요한 연산이다. 이 연산에서는 주로 센서에서 데이터를 읽어서 노드로 가져오는데 주로 에너지가 소모되며, 이후 calibration등의 processing에서의 에너지 소모도 이 연산에서 소모된 에너지로 본다.
- Processing : 노드에서 응용의 프로세싱 연산은 응용의 종류에 따라 에너지 스케줄링을 위해서 수행되는 알고리즘의 처리 과정에 필요한 연산이다.
- Actuation : 노드에서 actuator들을 기동하는데 필요한 연산이다. 이 연산은 주로 LED 또는 모터를 기동하는데 주로 많은 에너지를 소모하게 된다.

다음은 응용의 상태 천이 그래프(STG)를 그림4에서 나타내고, 각 상태 변화에 대한 설명을 기술한다.

- ① 쿼리를 수행하기 위해서 우선 쿼리의 종류를 분석하는 Pre-processing 상태로 천이한다.
- ② 쿼리의 종류에 따른 정보를 가지고 Sensing 상태로 천이.
- ③ Sensing 된 결과를 가지고 Post-processing 상태로 천이.

- ④ Pre-processing과 Sensing에서 받은 정보를 가지고 Event Driven 쿼리인 경우라면, Event 발생 유무에 따라서 Transmission상태(⑤)와 Sensing상태(④)로 천이하고, 결과의 오류여부에 따라서 Sensing상태(④)와 Transmission상태(⑤)로 천이 한다.
- ⑥ 전송실패로 여러 번의 Transmission을 할 수 있다.
- ⑦ 쿼리의 수행 시간이 만료되지 않으면 Sensing 상태로 천이한다.
- ⑧ 쿼리의 수행 시간이 만료되었으면 End로 천이한다.

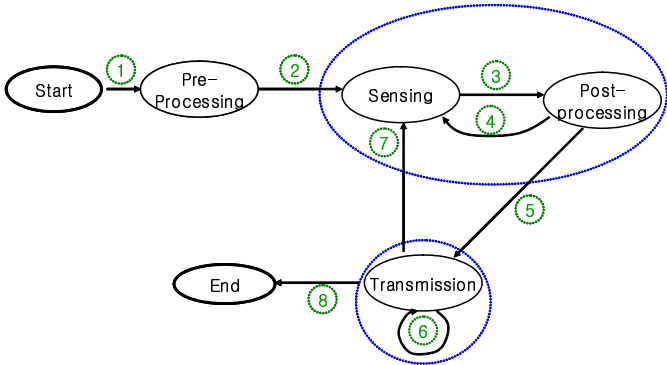


그림 4 응용의 상태 천이 그래프(STG : Status Transition Graph)

그림 4의 상태 천이 그래프에서 Actuation 연산은 포함되어 있지 않다. 이유는 대부분의 센서 네트워크 시스템에서 Actuation은 에너지를 많이 소모하는 것으로 보고 독립적인 전력 공급 구조를 택한다. 따라서 이 Actuation 연산은 다른 연산과 연관성을 배제하였다. 다음은 위에서 정의한 연산에서 소모된 에너지를 측정하기 위한 에너지 모델을 정의한다.

3.3 응용의 에너지 모델

에너지 소모 계획 알고리즘은 노드에서 쿼리를 수행할 때, 노드의 Lifetime을 보장하기 위해서 센서 노드의 현재 에너지 잔량을 기준으로 쿼리를 수행하는데 소모되는 에너지를 계획한다. 따라서 위에서 나눈 6종류의 응용들을 수행하는데 소모되는 에너지 중에서 조절 가능한 에너지 소모 요소로 쿼리를 구분

- 전송(응답)주기를 조절할 수 있는 쿼리
 - 전송 주기를 갖는 쿼리, 쿼리의 수행 시간을 갖는 쿼리, 전송주기와 쿼리의 생존시간을 모두 갖는 쿼리
- 센싱 주기를 조절할 수 있는 쿼리
 - Event Driven 쿼리, 쿼리의 수행 시간을 갖는 Event Driven 쿼리

3.1절에서 정의한 6가지 종류의 응용들에 대하여 에너지를 절약할 수 있는 전략이 다를 수밖에 없다. 다음은 각각의 쿼리에 따른 에너지 절약 전략을 설명하고 있다.

- 전송(응답) 주기를 갖는 쿼리(쿼리의 수행 시간이 정해져 있지 않음)
 - 원하는 응답 주기를 맞추기에 에너지가 충분치 않다면, 주기를 길게 잡아서 메시지 전송 횟수를 줄인다. 이럴 경우 proxy node에서 생략된 주기의 응답을 추정해서 보고할 수 있도록 보완한다.
- 쿼리의 수행 시간을 갖는 쿼리(전송 주기가 정해져 있지 않음)
 - 이 경우 노드 자체에서의 에너지의 분배를 가장 자율적

으로 처리할 수 있는 형태의 쿼리이다. 따라서 현재 남은 에너지를 고려해서 메시지 전송 주기를 정해진 기간 동안 정할 수 있다.

- Event Driven 쿼리(전송 주기를 가질 수 없고 쿼리의 수행 시간이 정해져 있지 않음)
 - 이 경우에는 다른 쿼리에 비해서 수행하는데 있어서 많은 에너지가 소모 되지 않을 수 있지만, 너무 낮은 문턱치 값을 가질 경우 매우 빈번한 전송이 발생해서 많은 에너지를 소모할 수 있다. 따라서 이 쿼리의 경우에도 한번 전송을 수행한 후 다음 전송까지의 최소한이 term을 두어서 빈번한 전송이 발생하지 않도록 한다.
- 한번만 전송하는 쿼리(쿼리의 전송 주기와 수행시간을 정할 수 없다.)
 - 현재 가지고 있는 에너지가 극히 적어서, 메시지 전송을 수행하지 못할 정도가 아니라면 특별히 노드 자체에서 고려할 사항은 없다.
- 전송 주기와 쿼리의 수행시간을 모두 갖는 쿼리
 - 원하는 기간 동안 원하는 주기를 맞추기에 에너지가 충분치 않다면, 주기를 길게 잡아서 메시지 전송 횟수를 줄인다. 이럴 경우 proxy node에서 중간 term의 센싱 값을 추정해서 보고할 수 있도록 한다.
- 쿼리의 수행 시간을 갖는 Event Driven 쿼리
 - 이 경우도 Event Driven 쿼리의 처리와 마찬가지로 다른 쿼리에 비해서 수행하는데 있어서 많은 에너지가 소모 되지 않을 수 있지만, 너무 낮은 문턱치 값을 가질 경우 매우 빈번한 전송이 발생해서 많은 에너지를 소모할 수 있다. 따라서 이 쿼리의 경우에도 한번 전송을 수행한 후 다음 전송까지의 최소한의 term을 두어서 빈번한 전송이 발생하지 않도록 한다.

3.1절에서 정의한 6가지 종류의 응용들이 3.2절에서 정의한 다섯 가지의 연산을 통해서 노드의 응용 계층에서 수행된다. 따라서 응용에서 소모되는 에너지를 조절하기 위해서는 우선 이러한 연산에서 소모되는 에너지의 양을 알아야 한다. 그러기 위해서 응용들의 에너지 모델을 정의해야 하며, 이러한 에너지 모델은 기본적으로 다섯 가지의 연산으로 구성되어 질 수 있다. 이 에너지 모델은 각 응용에 따라 다양한 속성 값을 가질 것이다.

에너지 모델은 노드에서 소모되는 에너지의 양을 계산하기 위해서 필요하다. 하지만, 궁극적으로는 소모되는 에너지 중에서 노드를 정해진 생존시간까지 생존하도록 하기위해서 노드에서 조절 가능한 에너지 소모 요소의 에너지 값을 구하는 것이다. 우선 노드에서 에너지의 소모를 시간 t 에 대한 변화 함수로 보면, 이 에너지 소모 함수를 다음과 같이 수식 (1)로 표현할 수 있다.

$$E = E(t) \tag{1}$$

즉, E 는 소모된 에너지를 의미하는데, t 시간 동안 소모된 에너지로 볼 수 있다. 그리고 소모되는 에너지는 노드의 여러 상태에서 에너지를 소모할 수 있는데, 이들 상태를 구분하지 않고 모든 상태에서 소모되는 전체 에너지를 E_{β} 로 생각한다면, 아래와 같이 수식 (2)로 표현된다.

$$E = E_{\beta} \tag{2}$$

위 식의 E_{β} 에서, 조절 가능한 에너지 소모 요소의 대상 관심을 응용에서 소모한 에너지로 보기 위해서는 다음처럼 수식 (3)으로 표현할 수 있다.

$$E = E_{\beta''} + \sum_{i=0}^n ECA_i \quad (3)$$

즉, ECA (Energy Consumption of Application)는 노드에서 수행되는 응용으로 인해서 소모되는 모든 에너지를 의미한다. 한 노드에 여러 개의 응용들이 존재한다면, 수식 (3)처럼 여러 개의 응용들이 소모하는 에너지의 합으로 표현될 수 있다. 위 수식 (3)에서 ECA 는 응용의 순환주기에서 정의한 상태를 가지고 표현될 수 있다. 따라서 각 상태별로 에너지 소모를 분류할 수 있고, 순환주기에서 개별적인 상태로 정의되지 않은 부분일지라도 어느 정도의 에너지를 소모한다고 생각된다. 그래서 이런 에너지를 α 로 표기한다면, 다음처럼 수식 (4)로 표현될 수 있다.

$$ECA = n_T \times E_T + n_S \times E_S + n_R \times E_R + \alpha \quad (4)$$

수식 (4)에서 E_T 는 전송(Transmission)연산에서 소모하는 에너지를 의미하고, E_S 는 센싱(Sensing)연산에서 소모하는 에너지를 의미하고, E_R 는 수신(Reception)연산에서 소모하는 에너지를 의미한다. 그리고 n_T 는 t시간 동안의 전송 회수를 나타내며, n_S 는 t시간 동안의 센싱(Sensing) 회수를 나타내며, n_R 는 t시간 동안의 수신 회수를 나타낸다. 결과적으로 α 도 $E_{\beta''}$ 과 마찬가지로 에너지를 소모하는 정확한 상태들로 구분할 필요가 없는, 즉 노드에서 조절 가능한 에너지 소모 대상으로 보지 않는다. 따라서 이 두 변수를 합쳐서 E_{β} 로 표현한다. 그리고 전송과 센싱 및 수신 등의 연산은 셀 수 있는 연산이나 E_{β} 는 셀 수 있는 성질의 것이 아니라 어느 시간 구간 동안에 소모된 에너지로 볼 수 있다. 그러므로 일정한 시간을 정하여 측정해야 한다. 이 일정한 시간 구간을 단위시간이라 한다. 따라서 n_T, n_S, n_R 등도 단위 시간동안에 수행된 개별 연산들의 회수로 정의될 수 있다. 그리고 위 수식 (4)에서 $ECA - \alpha$ 는 응용 순환 주기에서 개별적인 연산으로 분리될 수 있는 상태들에서 소모된 에너지의 합으로 보고, 이를 $ECSSA$ (Energy Consumption of Split States in Application)로 표기한다. 그럼 수식 (3)과 (4)는 다음의 유도과정을 통해서 수식 (5)로 표현된다.

$$ECSSA = n_T \times E_T + n_S \times E_S + n_R \times E_R$$

$$ECA = \sum_{i=0}^n (ECSSA_i + \alpha_i)$$

$$E = E_{\beta''} + \sum_{i=0}^n (ECSSA_i + \alpha_i)$$

$$E = E_{\beta''} + \sum_{i=0}^n ECSSA_i + \sum_{i=0}^n \alpha_i \quad (5)$$

$$E_{\beta} = E_{\beta''} + \sum_{i=0}^n \alpha_i$$

$$E = E_{\beta} + \sum_{i=0}^n ECSSA_i$$

다음은 응용에 따른 에너지 소모 계획 알고리즘에 대하여 설명한다.

4. 응용에 따른 에너지 소모 계획 알고리즘

앞에서 정의한 응용과 응용의 연산을 응용의 에너지 모

델을 통해서 각 연산들의 값들을 구할 수 있음을 설명하였다. 하지만, 이들 연산중에서 현재 에너지 소모 계획을 통하여 조절 가능한 연산은 전송(Transmission)과 센싱(Sensing)이 두 가지로 여겨진다. 왜냐하면, 수신(Reception)연산은 응용을 수행하기 위해서 반드시 필요한 연산이므로 임의로 수신을 거절하거나 하여 에너지 소모를 줄이기 위하여 변경할 수 있는 성질의 연산이 아니다. 그리고 처리(Processing)연산의 경우 다른 모든 연산과 의존관계가 있고, 응용을 처리하는데 필요한 처리 연산 또한 에너지 소모를 줄이기 위하여 변경할 수 있는 성질의 연산이 아니다. 물론 다른 연산과 의존관계에 있으므로 처리 연산에 소모되는 에너지가 다른 연산의 에너지 소모량에 대부분 흡수될 수 있도록 처리한다. 또한 구동(Actuation)연산 또한 아직 모든 센서 네트워크의 응용에 반드시 필요한 연산이라 보기 어렵고, 또한 이 구동 연산을 수행하기 위한 에너지를 분리해서 공급할 수도 있으므로 이 연구에서는 구동 연산 또한 고려대상에서 제외하였다. 따라서 전체 5가지 연산중에서 2개의 연산(전송, 센싱)을 제외한 나머지 3개의 연산은 E_{β} 로 귀속되는 것으로 보고, 앞의 에너지 모델을 통해서 전송과 센싱연산에 소모 되는 단위시간의 에너지를 구하게 된다. 즉, 앞의 수식 5는 다음과 같이 표현된다.

$$E = E_{\beta} + n_T \times E_T + n_S \times E_S \quad (6)$$

따라서 수식 (6)을 풀기위해서는 이미 측정되어서 알고 있는 E, n_T, n_S 등의 값을 제외하고, E_{β} 와 E_T 와 E_S 의 값을 알아야한다. 그래서 이 수식은 3원 1차 연립방정식을 풀면 된다. 따라서 서로 동치가 아닌 3개의 식이 필요한데, 만약 생성된 식이 동치가 되면 새로운 식을 얻기 위하여 측정을 다시 해야 한다. 즉, 3개의 동치관계가 성립되지 않는 3개의 식이 필요하다. 또한 모든 연산에는 에너지가 소비된다는 전제를 가지고 있어야 한다. 즉 E_T, E_S 의 값이 0이 아니라는 것이다. 그런데, 응용에서 전송주기와 센싱주기가 서로 의존관계를 가지고 있다면, 이 수식은 좀 더 단순해질 수 있다. 예를 들면, 3번의 센싱을 수행한 후 결과를 전송하라는 응용이 있다면 전송주기와 센싱주기는 서로 의존관계를 가지고 있다. n_T 에 대한 상대적인 센싱 회수를 r_{SN} 이라고 한다면 다음과 같이 수식 (7)로 표현하는 것이 가능하다. 그리고 $(E_T + r_{SN} * E_S)$ 을 E_{TS} 로 간략하게 표현하면 식을 좀 더 간략하게 표현할 수 있고, 2원 1차 연립방정식의 해를 구하면 된다.

$$E = E_{\beta} + n_T \times (E_T + r_{SN} * E_S)$$

$$E_{TS} = (E_T + r_{SN} * E_S) \quad (7)$$

$$E = E_{\beta} + n_T \times E_{TS}$$

만약, 응용이 Event driven의 경우라면, 수식 (6)에서 E_T 는 조절 가능한 값이 아니므로 아래의 수식 (8)처럼 E_T 는 E_{β} 로 포함시킬 수 있다. 예를 들면, 산불 감시와 같은 응용에서는 산에 불이 났는지에 대한 센싱을 수행하다가, 산불이 난 것으로 판단된 경우에만 데이터를 전송할 것이다. 이 경우 센싱 주기는 변경이 가능하지만, 전송주기 즉 산불이 난 것으로 판단한 경우에 대한 전송은 반드시 이루어져야 한다. 따라서 센싱에 소모된 단위 에너지에 대한 수식으로 표현 가능하다.

$$E = E_{\beta} + n_S \times E_S \quad (8)$$

이 수식의 값을 계산하기 위해서는 2개의 식만 있어도 각각의 값들을 구할 수 있다. 따라서 이 경우 센서 노드에서 정해진 시간까지 살기 위한 에너지의 소모 계획은 매우 간단하게 작성될 수 있다. 이 경우에는 센서 노드에 남은 에너지(E_{REM})와 앞으로 전송할 주기(T_{cycle})와 센싱할 주기(S_{cycle})에 관한 식을 작성하면 된다. 앞의 7, 8의 식을 통하여 E_{β} 와 E_{TS} 와 E_S 의 값들은 구할 수 있다. 이렇게 구해진 값들은 측정을 단위 시간(T_{UNIT})에 따라 수행을 하였다. 그러므로 단위 시간당 소요되는 값들이다. 즉, 단위 시간동안 전송 연산이나 센싱 연산과 관련 없는 에너지를 E_{β} 로 놓고, E_{TS} 와 E_S 는 한번 전송하거나 센싱을 하는데 소요되는 에너지의 예측 값들이다. 센서 노드가 앞으로 생존해야 할 시간을 T_{REM} 라 하면, T_{cycle} 와 S_{cycle} 에 관한 식은 다음과 같이 작성된다. 여기에서 N_{TS} 는 남은 Lifetime동안에 최대 전송할 수 있는 회수이다. 그리고 N_S 는 남은 Lifetime동안에 최대 센싱할 수 있는 회수이다.

$$N_{TS} = (E_{REM} - (T_{REM} \div T_{UNIT}) \times E_{\beta}) \div E_{TS} \quad (9)$$

$$T_{cycle} = T_{REM} \div N_{TS}$$

$$N_S = (E_{REM} - (T_{REM} \div T_{UNIT}) \times E_{\beta}) \div E_S \quad (10)$$

$$S_{cycle} = T_{REM} \div N_S$$

식 9는 센싱주기가 전송주기에 의존적인 연산의 경우에 필요한 계산이며, 식 10은 센싱주기에 의존적인 연산의 경우에 필요한 계산이다. 이렇게 구해진 T_{cycle} 와 S_{cycle} 은 앞으로 남은 생존시간동안 몇 분 또는 몇 초 마다 전송을 할지와 센싱을 할 것인지를 나타낸다. 따라서 남은 생존 시간을 보장하는 새로운 전송 주기와 센싱 주기가 작성되었다. 위 수식 9와 10에서 T_{REM} 의 단위와 T_{UNIT} 이 같은 값이라면 아래와 같이 수식을 단순화 시킬 수 있다.

$$N_{TS} = (E_{REM} - (T_{REM} \times E_{\beta})) \div E_{TS} \quad (11)$$

$$T_{cycle} = T_{REM} \div N_{TS}$$

$$N_S = (E_{REM} - (T_{REM} \times E_{\beta})) \div E_S \quad (12)$$

$$S_{cycle} = T_{REM} \div N_S$$

5. 결론

본 논문에서는 기존의 연구와 달리 생존 시간이 정해진 센서 네트워크를 가정하여, 중앙 집중식의 에너지 소모 계획에서 발생될 수 있는 부수적인 메시지 발생을 막음으로써 매우 제한된 에너지를 갖는 센서 노드의 에너지 소모를 줄일 수 있다고 본다. 그래서 본 논문에서는 센서 노드 수준에서의 에너지 소모 계획 알고리즘을 제안하였다. 그리하여 센서 노드의 응용 계층에서 응용의 종류를 분류하고 각 응용에서 에너지를 소모하는 연산들을 정의하였으며, 응용의 에너지 모델을 제안하여 각 응용의 연산에서 소모되는 에너지들을 통계적 기법을 이용하여 계산할 수 있음을 논리적으로 전개하였다. 또한 본 논문에서 제안한 에너지 소모 계획 알고리즘은 정해진 시간까지 센서 노드가 존재할 수 있음을 보장함으로써 전체 센서 네트워크가 정해진 생존 시간까지

존재하도록 하였다. 따라서 본 논문의 결과가 환경감시, 재해예방, 의료, 교통, 국방 등 매우 많은 응용영역에서 사용될 센서 네트워크의 발전에 상당한 기여를 할 것으로 기대한다. 앞으로의 추가적인 연구는 본 논문에서 제안한 알고리즘의 신뢰성을 좀 더 높이기 위해서 NS-2와 같은 네트워크 시뮬레이터를 통한 실험 계획을 가지고 있다.

6. 참고 문헌

- [1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramanian, and Erdal Cayirci, "A Survey on Sensor Networks", IEEE Communications Magazine 2002.
- [2] John A. Stankovic, Tarek F. Abdelzaher, Chenyang Liu, Lui Sha, Jennifer C. Hou, "Real-Time Communication and Coordination in Embedded Sensor Networks", Proceedings of the IEEE, Vol. 91, NO. 7, July 2003
- [3] T. V. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks", in ACM SenSys'03, Nov. 2003.
- [4] Budhadiya Deb, Sudeept Bhatnagar and Badri Nath, "A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management", Technical Report DCS-TR-441, Department of Computer Science, Rutgers University, May 2001. Submitted for Publication.
- [5] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," presented at the ATIRP Conf., College Park, MD, Mar. 2000.
- [6] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "Power-aware routing in wireless packet networks," Mobile Multimedia Communication (MOMUC), San Diego, November 1999.
- [7] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," Mobile Computing and Networking (MobiCom), Rome, July 2001.
- [8] C. K. Toh, "Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks," IEEE communication magazine, pp 138-147, June 2001
- [9] Ram Ramanathan and Regina Hain, "Topology control of multihop wireless networks using transmit power adjustment," in INFOCOM, 2000.
- [10] Vikas Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in INFOCOM, 2003.
- [11] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", In Proceedings of 5th ACM/IEEE Mobicom Conference, 1999
- [12] Xiang ji; Hongyuan Zha; Metzner, J.J.; Kesidis, G.; Communications, 2004 IEEE International Conference on Volume 7, 20-24 June 2004 Page(s)
- [13] W. Heinzelman, A. Chandrakasan, H. Balakrishnan: Energy-efficient routing protocols for wireless micro sensor networks. In Proc. 33rd Hawaii Int. Conf. System Sciences (HICSS), Maui, HI (2000)