

## 배터리 소비 특성을 고려한 센서네트워크 라우팅 알고리즘

이호준<sup>○</sup> 차호정  
연세대학교 컴퓨터과학과  
{hjlee<sup>○</sup>, hjcha}@yonsei.ac.kr

### Battery-aware Adaptive Routing Algorithm in Wireless Sensor Networks

Hojoon Lee<sup>○</sup>, Hojung Cha  
Department of Computer Science, Yonsei University

#### 요 약

센서네트워크에서 라이프타임에 관한 기존 연구는 다음 노드까지 데이터를 전송하는데 필요한 에너지 혹은 데이터의 생성 속도 등 미리 정해진 몇몇 값에 의존하는 경향이 있어 동적인 네트워크의 상황을 제대로 반영하지 못한다. 본 논문에서 제시하는 라우팅 알고리즘은 주기적으로 배터리의 전압값을 읽고 시간당 전압 감소율을 계산하여 각 노드의 작동 가능한 시간을 계산한다. 또 하드웨어에 따른 배터리 소비 특성을 데이터베이스화하여 참고하기 때문에 보다 정확한 라이프타임의 예측이 가능하다. 이렇게 계산된 결과를 바탕으로 라이프타임이 큰 노드만 선택하여 데이터를 전송하기 때문에 전체 네트워크가 동작하는 시간을 연장시킬 수 있다.

#### 1. 서 론

센서네트워크에 주로 사용되는 하드웨어는 제한된 자원을 가지고 특정 목적을 위해 네트워크를 구성한다. 이런 경우 센서네트워크의 특성상 인력으로 관리가 어려운 많은 수의 하드웨어를 이용하여 넓은 지역에 네트워크를 구성하는 경우가 많다. 때문에 네트워크가 구성될 당시 부여된 역할을 추가적인 관리 없이 좀 더 오래 수행할 수 있도록 라이프타임을 증가시키기 위한 연구가 진행되어 왔다.

라이프타임을 증가시키는 연구에는 크게 두 가지 분류가 있다. 첫 번째는 topology를 조정하는 방식이다[1]-[2]. 이 방식은 네트워크가 관리하는 영역 내에 필요한 최소한의 노드만 작동시키는 방법으로, 이벤트를 감지할 수 있는 영역이 겹치는 노드를 번갈아가면서 sleep시키거나 각각의 time table을 바탕으로 sleep/wake를 반복하면서 작동시간을 최소화하는 알고리즘이다. topology를 조정하는 방식의 경우 전체적인 에너지 소비량은 감소하지만 sleep하는 노드로 인해 데이터 전달에 delay가 발생할 수도 있다는 단점을 가지고 있다. 두 번째는 에너지를 고려한 라우팅 프로토콜이다[3]-[4]. 에너지 소모를 고려한 라우팅 알고리즘은 데이터의 흐름을 분산시켜 특정 노드에 트래픽이 집중되지 않고 전체 노드가 고르게 에너지를 소비하는 방법을 사용한다. 따라서 에너지의 소비가 특정 노드에 치우치지 않도록 라우팅 경로를 찾아내고, 데이터를 전달하는데 필요한 에너지가 적은 방향으로 packet의 전달이 이루어진다. MTE[5]에서는 다음 노드까지 데이터를 전송하는데 필요한 에너지를 계산하여 더 적은 양의 에너지를 소비하는 라우팅 경로를 사용하는 방법을 제시하고 있다. 그리고 MMRE[6]에서는 라우팅 테이블에 있는 여러 노드의 현재 남은 에너지 살펴보고 가장 많은 에너지를 경로를 선택한다. [7][8]에서는 이 두 방식을 각각 적용해서는 최적의 라우팅 결정을 할 수 없음을 보이고 이를 혼합한 알고리즘을 제안하고 있다. [9]에서는 각 노드에서 발생하는 시간당 데이터양을 고려하여 트래픽이 분산되도록 스페인트리 구조를 구성하는 방법을 수학적으로 풀어내고 있다. 이러한 방식은 각 노드에서 시간당 발생하는 데이터의 양이나 노드별 데이터 전달에 필요한 에너지 등을 사전에 정의한 상태로 동작

하기 때문에 임의로 이벤트가 발생하는 지역이나 localization 등에는 적용하기가 어려운 면이 있다. 또 하드웨어로 측정하기 어려운 요소를 참고하는 방법은 시뮬레이션이 아닌 실제 네트워크에서는 사용하기 힘들다.

제시하는 알고리즘은 ADC를 통하여 배터리의 전압값을 읽어내고 시간당 전압 감소율을 계산한다. 이를 통해 보다 정확한 노드의 라이프타임을 예측하고 더 오래 동작할 수 있는 노드를 다음 라우팅 경로로 선택하게 된다. 또 전압의 감소율을 이용하기 때문에 라디오나 CPU를 통한 에너지 소비뿐만 아니라 하드웨어의 다른 부분에서 소비하는 에너지 역시 고려할 수 있다. 뿐만 아니라 하드웨어별 에너지 소비 특성을 사전에 저장한 후 컴파일 타임에 이를 이용하기 때문에 서로 다른 하드웨어에서는 차별화된 기준이 적용된다. 따라서 기존에 생각하지 않았던 하드웨어의 특성까지 반영할 수 있다는 장점이 있다. 이러한 방식은 사전에 네트워크의 속성이나 임의의 값을 정의하여 참고하는 부분이 없기 때문에 네트워크에 독립적으로 적용할 수 있다.

#### 2. 배터리를 고려한 라우팅 알고리즘

##### 2.1 Background

그림 1은 서로 다른 오버헤드에 따라 배터리의 전압 감소량이 얼마나 달라질 수 있는 가를 보여주고 있다. 실험은 Mica2를 사용하였으며 두 노드에는 서로 다른 CPU, radio 오버헤드를 주고 6시간 동안 ADC를 통하여 전압값을 측정함 그래프이다. 6시간 동안의 전압차이는 아주 서서히 벌어지지만 노드의 작동시간이 100시간이 된다고 했을 때 그 차이는 아주 커질 수밖에 없다. 따라서 정해진 시간마다 배터리의 값이 얼마나 감소되는지를 측정하면 각 노드가 현재 얼마나 많은 에너지를 소모하는지 예측할 수 있고, 이를 통하여 노드가 앞으로 작동 가능한 시간을 계산할 수 있다.

실험에 사용된 Mica2에 장착된 Atmega 128은 active mode에서 Telos의 msp430보다 11배의 에너지를 소비한다. 또 sleep 모드에서는 10배가 넘는 에너지를 필요로 한다. 하드웨어가 동작할 수 있는 최소 전압도 서로 다른데 Mica2같은 경우 89mW의 전력을 소비하면서 2.7V이상에서만 동작하는 반면 Telos는 41mW의 에너지를 사용하며 1.8V이상에서 작동할 수 있다. 이렇게 하드웨어에 따라 서로 다른 에너지 소비 패턴을 가지고,

본 연구는 한국과학재단에서 지원하는 국가지정연구실사업으로 수행하였음 (과제번호 : 2005-01352)

동작할 수 있는 전압이 다르기 때문에 라이프타임을 계산하는 기준도 달라져야 한다. 본 알고리즘에서는 배터리 전압값의 감소율에 따라 세 개의 state로 구분하는데 하드웨어가 다를 경우 이러한 state의 전환 시점이 다르기 때문에 어떠한 하드웨어를 사용하느냐에 따라 서로 다른 기준이 적용되어야 한다. 또 라이프타임을 계산하는데 사용되는 노드가 작동할 수 있는 최소 전압 역시 하드웨어에 따른 차별화를 적용하여야 정확한 예측이 이루어질 수 있다. 따라서 하드웨어별 실제 측정된 데이터를 미리 저장한 후 컴파일 타임에 이를 이용한다.

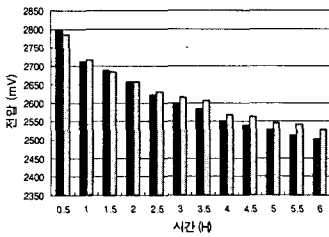


그림 1 서로 다른 오버헤드를 가진 두 노드의 전압 변화량

2.2 라우팅 알고리즘

그림 2의 실선은 Mica2를 이용하여 작동을 멈출 때까지 전압값의 변화추세를 나타낸 것이다. 전압값의 감소량은 기울기에 따라서 세 부분으로 구분이 가능하다. 각 부분을 각각 1,2,3 state라고 하면 라우팅 결정은 우선 1번 state에 있는 노드를 우선적으로 선택한다. 1번 state의 노드는 다른 state에 노드보다 더 많은 양의 에너지를 가지고 있기 때문에 더 오래 동작할 수 있는 가능성이 있기 때문에 각 노드가 어떠한 state에 있는지를 첫 번째로 고려한다.

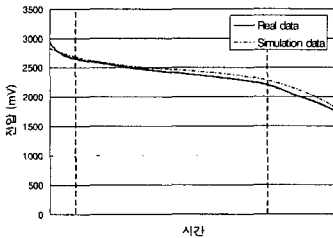


그림 2 시뮬레이션과 실제 하드웨어에서의 전압 변화량

state의 전환에는 미리 저장된 하드웨어별 데이터가 이용된다. 하드웨어별 소모하는 에너지의 양이 다르기 때문에 전압 감소량이 차이가 나며, 동작할 수 있는 최소 전압도 상이하기 때문에 라이프타임 계산에 이용되는 기준값이 똑같이 적용되어서는 정확한 예측을 할 수 없다. 따라서 대표적인 하드웨어인 Mica 시리즈와 Telos의 관련 데이터를 실제 하드웨어를 이용하여 미리 측정하고 컴파일 타임에 적용될 수 있도록 하였다.

$$V_n - \frac{dV}{dt} \times T_p = V_f$$

$$\therefore T_p = \frac{(V_n - V_f) \times dt}{dV} \quad \dots(1)$$

만약 state가 같은 노드가 두 개 이상 존재할 경우 그 중 가장 오래 동작할 수 있는 노드를 선택한다. 이때 각 노드에 저장된 전압값을 바탕으로 시간당 전압 감소율을 구하고 이를 이용하여 노드가 동작할 수 있는 시간을 구한다. 동작할 수 있는 최소 전압을  $V_f$ , 현재 전압값을  $V_n$ , 노드의 수명은  $T_p$ 이라 하면 (1)과 같은 식이 성립한다. 각 state별 현재 전압 감소율을 이용하여 노드가 동작할 수 있는 시간  $T_p$ 를 구하는 것이다. 따라서

각 노드는 현재와 같은 오버헤드로 인해 전압이 감소할 경우 얼마나 오래 동작할 수 있는지를 알아낼 수 있다. 동일한 state에 있는 경우라도  $T_p$ 가 큰 노드를 라우팅 경로로 선택함으로써 데이터송수신이나 이벤트 혹은 또 다른 이유로 배터리 소모가 클 경우 이를 라우팅 경로에서 제외시킬 수 있다.

```
//at every time interval f
UpdateNodeState
Input : history table of voltage value (H)
Output : updated history table
Begin
  Vn = ReadADC();
  write Vn to history table;
  Tp = GetLifetime(Vn); // by function (1)
  S = GetState(H); // by pre-stored criteria
  Broadcast(S, Tp);
End

//whenever a node receive control message
FindParent
Input : routing table(R), new state(S), new lifetime(Tp)
Output : new parent node, updated routing table
Begin
  find node K in routing table
  update S, Tp of node K
  update transmission rate of node K
  find new parent Cp
  Output = Cp
End
```

그림 3 라우팅 알고리즘 Pseudo code

제안하는 라우팅 알고리즘을 적용시킬 경우 전송할 데이터를 가진 노드는 우선적으로 남은 에너지가 많은 노드를 라우팅 경로로 선택하게 되고, 배터리가 많은 노드지만 에너지 소비율이 큰 노드를 라우팅 경로에서 제외시킬 수 있다. 또 배터리가 많다 해도 현재 사용하는 에너지가 많아서 배터리 소모가 클 경우 라이프타임 계산을 통한 예측이 가능하기 때문에 좀 더 정확한 라우팅 결정을 내릴 수 있다.

각 노드는 정해진 시간마다 새로운 라이프타임과 자신이 속한 state를 one-hop neighbor에 전송한다. 이 데이터는 라우팅 테이블을 구성하는 동시에 라우팅 결정을 내릴 정보를 구성한다. 전송할 데이터가 있는 노드는 자신의 라우팅 테이블을 확인하고 state가 가장 낮은 노드를 첫 번째 후보로 선택한다. 만약 후보가 두 개 이상 존재할 경우 각 노드의 라이프타임을 비교하여 가장 오래 동작할 수 있는 노드를 라우팅 경로로 선택한다. 데이터를 만들어 낸 노드부터 싱크노드까지 각 노드에서 이러한 결정을 하게 되며, greedy한 방식으로 one-hop마다 라우팅 경로를 결정해 나가는 것이다.

3. 실험

라이프타임에 실제 실험은 시간적인 어려움이 있기 때문에 TinyOS 시뮬레이터인 TOSSIM을 이용하였다. 기존의 TOSSIM에서는 ADC 포트를 이용하여 전압값을 읽는 것이 지원되지 않기 때문에 새로운 모듈을 추가하였다. 실제 전압값의 변화와 유사한 값을 구할 수 있는 3차 방정식을 정의하고 이를 이용하여 전압 값을 읽을 수 있는 ADC 포트에 10초마다 값을 써나감으로써 어플리케이션에서 이를 이용할 수 있도록 하였다. 또 서로 다른 오버헤드에 따라 다른 전압 감소율을 적용하여 매 3분마다 오버헤드에 따라 전압감소율을 변화시킴으로써 실제 하드웨어에서의 실험과 같은 환경을 최대한 반영할 수 있도록 하였다. 그림 2에서 볼 수 있듯이 같은 오버헤드의 어플리케이션을 실제 하드웨어와 시뮬레이터에서 동작 시킨 후 전압값의 변화가 약간의 차이는 있지만 유사한 패턴을 보이는 것을 볼 수

있다.

이를 이용하여 25 feet 간격으로 25개의 노드를 정사각형으로 배열하고 한쪽 모서리의 24번 노드에서 반대쪽 모서리인 0번 노드까지 매초 데이터를 전송하였다.(그림 4) 라우팅에서는 각자의 위치정보를 알고 있는 geographical한 알고리즘이 사용되었으며 25개의 노드 중 처음으로 배터리를 모두 소모하는 노드가 발생할 때까지 실험을 실시하였다.

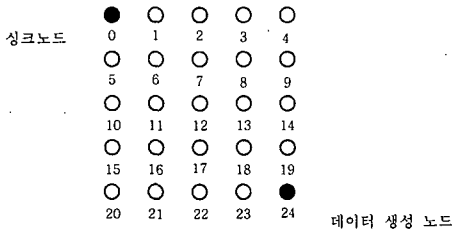
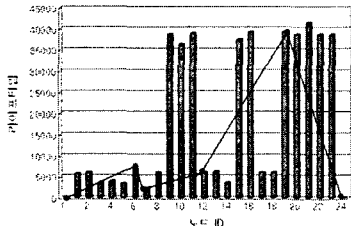
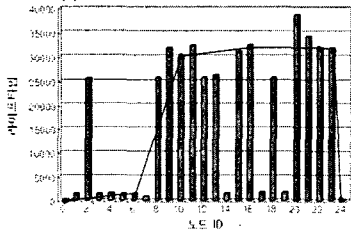


그림 4 노드의 배치

그림 5의 (a)는 시뮬레이션이 시작되고 일정 시간이 지난 후의 라우팅 경로(실선)와 라이프타임(막대)을 나타낸다. 24번 노드에서 전송한 데이터는 각 노드별로 최상의 경로라고 파악되는 노드를 따라 0번 노드까지 전송된다. 이 경우 22번 23번 등 배치에서 왼쪽 편에 있는 노드는 이전에 많은 양의 데이터를 처리하여 state가 더 높아진 노드들이다. 때문에 라이프 타임이 더 큰 경우라도 선택되지 않은 것이다. 그림 5(b)는 (a)에서 모든 노드가 새로운 라이프타임을 한번 씩 업데이트 한 후의 경로변화를 보여준다. 19번 노드의 배터리 소모가 심했기 때문에 새로운 라이프타임은 아주 작게 계산되었으며, 때문에 24번 노드는 라이프타임이 더 큰 23번 노드를 거치는 새로운 경로로 데이터 전송을 시작하였다.



(a) 시간 t에서의 라우팅 경로



(b) t에서 모든 노드가 한번 씩 업데이트된 후 수정된 라우팅 경로

그림 5 라우팅 경로 및 각 노드의 라이프타임

그림 6은 노드 1의 시간대별 데이터 전송 횟수(막대)와 라이프타임(실선)의 변화를 보여준다. 그림에서 몇 개의 노드는 라이프타임이 급격히 늘어나거나 줄어드는 것을 볼 수 있다. 라이프타임이 크게 증가하는 경우는 state가 1에서 2로 전환되거나 노드의 오버헤드가 줄어드는 경우이다. state가 1에서 2로 전환될 때와 라우팅 경로에서 제외되어 오버헤드가 사라질 경우 라이프타임이 크게 변화하는 것을 볼 수 있다. 반면 라이프타임이 줄어드는 경우는 state 1에서 2로 전환되기 전이나 state

가 2에서 3으로 전환되는 경우, 또 노드에 트래픽이 발생할 경우이다.

알고리즘을 적용한 경우 24번 노드에서 0번 노드에 이르기까지 균등한 분포를 보여주는 반면 일반 라우팅 알고리즘을 사용할 경우 트래픽이 고르게 분포되지 못할 뿐만 아니라 노드간의 격차도 심한 것을 볼 수 있다. 시뮬레이션 시간은 알고리즘을 사용하지 않았을 때보다 약 6시간 정도 길게 지속되었다. 실험 시간을 줄이기 위해 50배 빠르게 배터리를 감소시켰기 때문에 각 노드의 업데이트 횟수가 그리 많지 못했고, 노드의 시작 시간이 비슷하게 몰려 있기 때문에 긴 시간을 한 경로만 사용한 점을 감안한다면 실제 실험에서는 경로가 자주 변경되면서 트래픽이 고르게 분산되는 더 좋은 결과를 보일 수 있을 것이다.

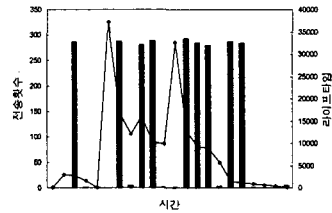


그림 6 노드 1의 시간대별 데이터 전송 횟수 및 라이프타임

#### 4. 결론

본 논문에서는 예측하기 힘든 센서 네트워크의 상황을 고려하여 네트워크에 독립적으로 적용할 수 있는 라우팅 알고리즘을 제시하였다. ADC를 통하여 전압값을 읽고 감소를 계산함으로써 하드웨어의 여러 부분에서 사용하는 에너지까지 반영이 되며, 이를 통하여 좀 더 정확한 라이프타임을 예측할 수 있다. 그리고 각 노드는 라이프타임이 큰 노드를 다음 라우팅 경로로 선택하기 때문에 에너지 소모가 특정 노드에 집중되는 문제를 방지할 수 있다. 이는 기존 방식보다 더 나은 성능을 보여줄 수 없으나 한정된 환경에서만 적용할 수 있는 틀에서 벗어나 제한조건 없이 다양한 상황에서 사용 가능한 알고리즘이라는 것에 의미가 있다. 추후 greedy한 방식에서 벗어나 전체적으로 최적의 경로를 선택하는 프로토콜이나 좀 더 정확성을 높인 TOSSIM 모듈을 연구할 계획이다.

#### 5. 참고문헌

- [1] Ram Ramanathan and Regina Hain, "Topology control of multihop wireless networks using transmit power adjustment," in INFOCOM, 2000.
- [2] Vikas Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in INFOCOM, 2003.
- [3] Sheetal Kumar Doshi and Timothy X Brown, "Minimum energy routing schemes for a wireless ad hoc network," in INFOCOM, 2002.
- [4] Jae-Hwan Chang and Leandros Tassiulas, "Energy conserving routing in wireless ad hoc networks," in INFOCOM, 2000.
- [5] 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.
- [6] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," presented at the ATIRP Conf., College Park, MD, Mar. 2000.
- [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] Guoliang Xing, Ying Zhang, Rober Pless, "Minimum power configuration in wireless sensor networks," MobiHoc 2005