제25회 한국정보처리학회 춘계학술발표대회 논문집 제13권 제1호 (2006. 5)

멀티 싱크 센서 네트워크 환경에서의 에너지 효율적인 플러딩 기법

마용재*, 윤성민*, 이종협*, 송주석*
*연세대학교 컴퓨터과학과 정보통신연구실
e-mail : neverbass@emerald.yonsei.ac.kr

Heuristic Energy-Efficient Flooding in Multi-Sink Sensor Networks

Yong-Jae Ma*, Sung-Min Yoon*, Jong-Hyup Lee*, Joo-SeokSong*
*Dept. of Computer Science, Yonsei University

요 약

무선 센서 네트워크에서 하나의 싱크가 아닌 여러 개의 싱크(multi-싱크)인 경우에는 데이터의 전송에 있어 특정 라우팅 기법을 이용하는 것 보다 플러딩방식이 매우 적합한 전송기법으로 알려져 있다. 그러나 유선 환경 등의 기타 네트워크에서 사용되던 플러딩 방식을 무선 센서 네트워크에 그대로 사용하는 것은 적절하지 않다. 무선 센서 네트워크에서는 에너지의 효율성이 무엇보다 중요하며 기존의 플러딩방식은 불필요한 패킷의 중복전송 등 해결해야 할 문제점들이 많이 남아있기 때문이다.

우리는 이웃 노드들의 노드 리스트를 적절히 관리하여 불필요한 패킷의 재전송을 막아주며, 노드 리스트를 관리하는 오버헤드를 최소화 하여 에너지 효율성을 높인 무선 센서 네트워크의 특성에 맞는 플러딩방식인 HEEF 기법을 제안한다.

1. 서론

센서 네트워크는 밀집해 뿌려진 많은 수의 센서 노드들로 구성된다. 센서 노드의 위치는 미리 정해질 필요 없이 임의적으로 뿌려진다. 이것은 접근이 쉽지 않은 지역에 대한 탐사를 가능하게 하는 등 많은 장점을 가지지만 반면에 센서 네트워크의 프로토콜과 알고리즘들은 자기 자신이 설정을 가능하게 해야 한다.

Broadcast 는 센서 네트워크에서 자주 사용되는데 플러딩 방식으로 구현된 broadcast 는 모든 노드들이 받은 메시지를 재전송하는 문제점을 갖고 있다. 비록 플러딩이 간단한 기법이라 하더라도, 그것은 많은 양의 메시지가 중복 전송 됨으로써 많은 네트워크 자원을 소모시킨다. 이것은 센서 네트워크에서 broadcast storm 이라 불리는 심각한 중복, 충돌 문제를 만들어낸다. 따라서 우리는 효율적으로 센서 네트워크에서 중복된 메시지를 줄일 수 있는 새로운 플러딩 기법을 제안한다. 이 기법은 단지 근거리의 node 들에 대한 정보만을 이용하여 불필요한 재전송을 피하도록 해주며 따라서 에너지의 효율성을 높일 수 있게 해 준다. 시뮬레이션에는 다른 플러딩 기법들과 비교하여 수행하였다.

2. 배경 지식

A . Self pruning

Self pruning 에서 각 노드는 이웃한 노드들과 인접한 노드의 리스트를 교환한다. 이 기술을 사용하기 위해서 우리는 모든 노드들이 인접한 노드들을 알고 있다고 가정한다. 이 가정은 대부분의 센서 네트워크에서 불가능하지 않다. 각 노드들은 주기적으로 'Who I am' 패킷을 이웃 노드들에 전송하여 자신의 존재를 알린다.

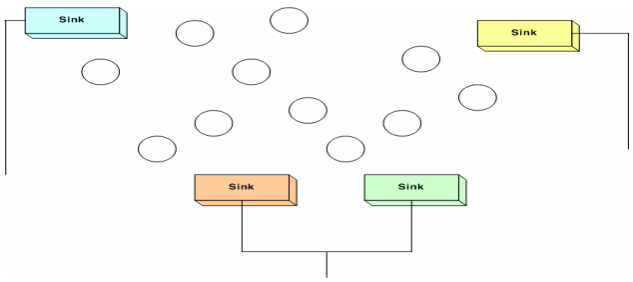
본 연구는 정보통신부 및 정보통신연구진흥원의
대학 IT 연구센터 육성・지원사업의 연구결과로 수행되었음

패킷을 전달하려는 노드 v_i는 인접 노드들의 리스트인 $N(v_i)$를 피기백한다. 패킷을 받는 노드 v_j 는 집합 $N(v_j) - N(v_i) - \{v_i\}$가 공집합인지를 검사한다. 만약 그것이 공집합이라면, 노드 v_j 는 모든 이웃노드들이 다 받은 것으로 간주하고 전달을 멈춘다. 그렇지 않을 경우에만 v_j 는 패킷을 전달한다.

받은 패킷을 전달할지 말지를 결정하기 위해서, v_j 는 $N(v_i)$에 속하는 모든 v에 대해 $N(v_j) - \{v_i\}$로부터 v를 찾아 제거하는 과정을 반복해야 한다. 만약 모든 엘리먼트들이 $N(v_j) - \{v_i\}$로부터 제거되었다면, vj는 패킷을 전달하지 않고 모두 제거된 것이 아니라면, v_j 는 패킷을 전달한다. 그러므로 self pruning 기법의 시간복잡도는 A가 트리의 최대 단계일 때 O(A)가 된다.

B . multi-sink environment

우리는 우리의 broadcast 알고리즘에서 싱크노드를 여러개 가지는(Multi-싱크) 센서 네트워크를 가정하였다.



(그림 1) Multi-sink sensor network environment

3. HEEF

A . 기존 플러딩 기법이 가지는 문제점

기존 플러딩 기법이 가지는 문제점은 대표적인 문제점은 무분별한 전송으로 인한 네트워크 전체적인 성능 저하이다. 특히 네트워크의 기반이 유선에서 무선으로 넘어오면서 무분별한 패킷의 반복적인 전송이 네트워크 성능에 미치는 영향이 더욱 더 커지게 되었다. 즉, 노드들이 밀집해 있는 경우 한 노드의 한번의 패킷 전송으로 인해 주변의 노드들이 한꺼번에 패킷을 이미 전송 받았음에도 불구하고
이를 확인하지 못하여 각 노드는 불필요한 플러딩을 수행하게 되는데 이러한 현상을 broadcast storm 이라고 부른다.
이 broadcast storm 에 의해서 야기되는 문제점은 단순히 네트워크의 성능 저하 뿐만이 아니라 잦은 패킷 전송을 통하여 전력 소비가 중요한 factor 로 작용하는 센서 네트워크에서 더욱 심각한 문제로 작용하게 된다.

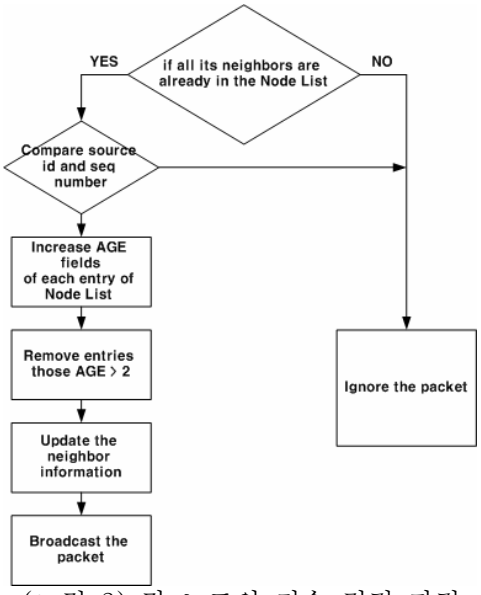
B . 제안하는 아이디어

앞에서 언급한 사항들을 고려하여 우리는 센서 네트워크를 고려한 새로운 플러딩 기법을 제안한다. 제안하는 기법은 기존의 플러딩 기법을 수정하여 사용한다. 이 기법에서는 Self-pruning 기법을 바탕으로 하여 주변 노드의 정보를 이용한다. 각 노드는 노드의 부팅과 동시에 주변 노드의 beacon 신호를 탐지하여 주변 노드의 존재와 주변 노드의 ID 정보를 알아낸다. 주변 노드의 정보는 각 노드의 Neighbor IDs 테이블에 저장하여 보관한다.

플러딩에 사용되는 패킷은 모두 Sequence number 와 Source node ID 항목을 가진다. 이 두 항목을 이용하여 각 노드는 한번 수신한 적이 있는 패킷이 재전송 되는 경우를 방지 할 수 있다. 또한 이 두 항목에 추가하여 각 패킷은 Node list 항목을 가진다. 이 항목의 각 entry 는 노드 ID 와 AGE 필드로 구성되고 이는 soft-state 특성을 가지는데, 이는 최근에 이 패킷을 받았을 것으로 예상되는 노드들의 ID 의 리스트로 플러딩을 시작하는 노드에서 생성하여 패킷을 받는 각 노드에서 정보를 갱신한다.
플러딩의 과정은 다음과 같다.

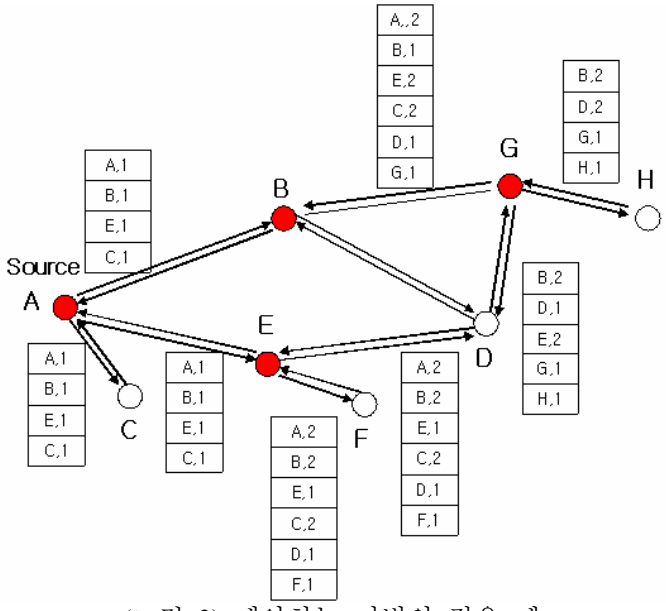
1. 플러딩을 시작하는 노드는 패킷의 Source node ID 필드를 자신의 ID 로 Sequence number 는 자신의 packet count 값으로 채운다. 패킷의 Node list 필드는 자신의 Neighbor IDs 테이블의 값을 이용하여 자신의 주변 노드 ID 를 Node list 에 채우고 각 entry 의 AGE 의 값은 1 로 설정한다. 이렇게 설정된 패킷은 플러딩 기법에 의해 주변 노드들에게 전송된다.
2. 패킷을 받은 각 노드는 먼저 패킷의 내용을 확인하고 자신이 이 패킷을 전송할 필요가 있는지를 판단한다.
   A. 패킷의 Node list 필드와 자신의 Neighbor IDs 테이블을 비교하여 자신의 모든 주변 노드가 Node list 에 포함이 되는지 확인한다. 자신의 주변 노드가 모두 포함되는 경우에는 주변 노드들도 자신과 함께 이 패킷을 받은 경우이므로 자신은 재전송 할 필요가 없다고 생각하고 해당 패킷을 전송하지 않는다.
   B. Node list 에 자신의 모든 주변 노드가 포함이 되지 않은 경우에는 패킷의 Seq. num 와 source node ID 필드를 확인하여 이미 수신한 적이 있는 패킷인지를 확인하고, 그렇다면 패킷을 전송하지 않는다.
   C. 앞의 두 경우에 모두 해당하지 않는다면 해당 패킷은 현재 노드에 의하여 재전송될 필요가 있는 패킷이기 때문에 패킷을 업데이트 하여 전송한다.

패킷의 업데이트 과정은 Node list 필드의 soft-state 특성을 제공하기 위해서 수행된다. 업데이트 과정에서는 해당 패킷의 Node list 필드의 AGE 값을 각각 1 씩 더한다. 그리고 각 entry 를 조사하여 AGE 의 값이 2 보다 커진 entry 를 Node list 에서 삭제한다. 이 과정이 끝나면 자신의 주변 노드 중에서 Node list 에 포함이 되지 않은 노드가 있다면 이 노드를 AGE 값은 0 으로 Node list 에 추가한다.



(그림 2) 각 노드의 전송 결정 과정

Self-pruning 기법을 센서 네트워크에 바로 도입하게 되면 self-pruning 에서 Node list 에 해당하는 패킷 정보가 네트워크가 커짐에 때라 기하 급수적으로 커지게 되는 문제점이 발생하게 된다. 하지만 우리가 제안하는 방법에서는 이러한 업데이트 과정 때문에 Node list 의 각 entry 는 2 hop 이상 유지 되지 않게 된다. 따라서 Node list 는 항상 패킷 주위 노드의 정보만을 유지하면서 불필요한 플러딩을 줄일 수 있게 되어 센서 네트워크의 성능을 향상 시킬 뿐만 아니라 불필요한 패킷의 전송 횟수를 줄이게 되어 효율적인 네트워크를 구성할 수 있게 된다.



(그림 3) 제안하는 기법의 적용 예

(그림 3)은 제안하는 기법이 적용된 예이다. 그림에서 플러딩은 A 노드에서 시작된다. 노드 A 에서는 자신을 포함하여 자신의 주변 노드의 ID 인 A,B,C,E 를 패킷의 Node list 에 담고 AGE 의 값은 모두 1 로 셋팅하여 전송한다. 전송된 패킷은 노드 B, C, E 에 전달되는데 노드 B 와 E 의 경우에는 Node list 에 자신의 주변 노드가 모두 포함되지 않기 때문에 패킷을 업데이트 하여 재전송하지만 C 의 경우에는 자신의 주변 노드인 A 가 Node list 에 포함이 되어 있으므로 A 는 이미 받았다고 판단하고 패킷을 전송하지 않는다. 노드 D 에서는 노드 B, E, G 에서 패킷을 수신하지만 노드들의 Node list 를 비교해 보고 자신의 주변 노드가 모두 포함 되었으므로 패킷을 전송하지 않는다. Node list 의 경우에도 soft-state 가 이용되기 때문에, A 노드에서 추가 되었던 노드 A 와 C 에 대한 entry 가 패킷이 노드 G 를 넘어가면서부터는 AGE 필드의 값이 2 보다 커지게 되어 사라지는 것을 볼 수 있다. 패킷의 2 hop 차이나는 노드의 정보는 사라져서 불필요한 패킷 크기의 증가를 막을 수 있다. 결과적으로, 제안하는 기법을 이용하면 전체 플러딩 과정에서 색칠된 노드만 전송을 수행하게 된다. 기존의 플러딩기법에서는 네트워크의 모든 노드에서 한번씩 전송이 수행되는 것임을 고려한다면 (그림 3)의 노드 수가 8 개 이므로 8 번의 전송이 있었다면 제안하는 기법에서는 이것이 4 번으로 줄어들게 되어 그림과 같은 네트워크 구조에서는 50%의 전송 횟수가 감소하는 것을 알 수 있다.

4. 성능평가

제안하는 기법의 성능 평가를 위하여 기존의 플러딩 기법과 제안하는 기법에서의 소비되는 에너지의 양을 분석하였다. 분석에 적용하는 네트워크로 Random 네트워크를 가정했을 때, 분석에 필요한 파라미터를 정리하면 다음과 같다.

<표 1> 분석을 위한 random 네트워크에서의 파라미터 정리

파라미터	내용
E_T	한 bit을 전송하기 위해 필요한 에너지의 양 (J/bit)
E_R	한 bit을 수신하기 위해 필요한 에너지의 양 (J/bit)
Δ	주변 노드의 평균 개수 (random 네트워크)
N	센서 네트워크의 총 노드 개수
M	이미 해당 패킷을 받아 본 주변 노드의 평균 개수 (Node list에 이미 포함되어있는 노드)
q	주변 노드와 주변 노드의 주변 노드가 겹칠 확률
S	패킷의 평균 크기 (bits)
i	Node list의 한 entry가 가지는 크기 (bits)

위와 같이 파라미터를 정의 했을 때, 기존의 플러딩에서 소비되는 에너지의 양은 다음과 같다.

$$EC_{CF} = (E_T + E_R\Delta) \times N \times S$$

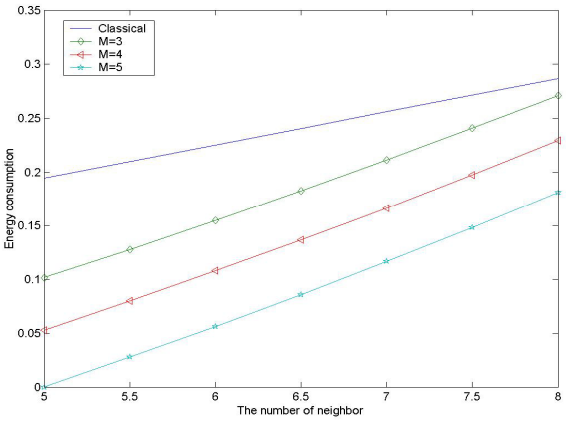
기존의 플러딩 기법에서는 각 노드가 최소 한번씩(총 N 번)의 전송을 수행하고 주변 노드가 그 패킷을 수신해야 하므로 위와 같은 식으로 정리 될 수 있다. 제

안하는 플러딩 기법에서는 자신의 주변 노드가 이미 패킷을 받았을 경우에는 전송을 하지 않기 때문에 대략 $(\Delta - M)/\Delta$ 의 확률로 전송 횟수가 줄어들게 된다. 따라서 제안하는 기법에서 소요되는 에너지의 양은 다음과 같이 표현할 수 있다.

$$EC_{idea} = (E_T + E_R\Delta)\times(\frac{\Delta - M}{\Delta})\times N\times\{S + (\Delta + (1-q)(\Delta-1)M)\times i)\}$$

하지만 제안하는 기법에서는 전송 횟수가 줄어드는 대신 Node list 의 크기만큼 패킷의 크기가 커지기 때문에 이에 대한 고려도 포함되어야 한다. 각 엔트리의 크기는 i 이고 엔트리에는 최소 자기 주변의 노드들과 패킷이 지나온 경로 중 지금 주변 노드와 겹치지 않는 노드들이 포함되기 때문에 Node list 의 엔트리의 크기는 $(\Delta + (1-q)(\Delta-1)M)\times i)$ 와 같이 되어 이 부분이 패킷의 크기에 포함된다.

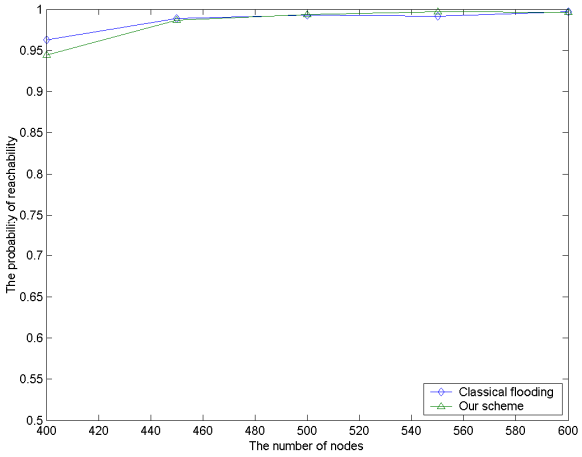
다음은 E_T = 0.8 uJ/bit, E_R = 0.6 uJ/bit, N = 100, S = 64 bytes, i = 2 bytes, q = 0.6 의 값을 가질 때, Δ 값과 M 값의 변화에 따른 에너지 소비량을 나타낸 그래프 이다.



(그림 4) Δ 값과 M 값의 변화에 따른 에너지 소비량

(그림 4)의 그래프에서 볼 수 있듯이 기존의 플러딩기법에 비하여 제안하는 기법이 에너지의 소비가 적은 것을 알 수 있다. 특히 Node list 에 포함되어 있는 주변 노드의 수가 많을수록 에너지 소비가 급격하게 줄어드는 것을 알 수 있다. 이러한 에너지 소비의 효율성은 기존의 플러딩 기법에 비해서 전송 횟수가 줄어들기 때문이지만, 네트워크가 커지면서 node list 의 크기가 커지기 때문에 네트워크의 크기 변화에 따른 M 값과 Δ 값의 동적인 변화를 고려하지 않은 상황에서는 주변 노드의 수가 많아질수록 에너지의 소비가 커지는 것을 확인할 수 있다.

이러한 성능 분석을 바탕으로 하여 우리가 제안한 기법에 대한 시뮬레이션을 수행하였다. 시뮬레이션은 기존의 플러딩 방법과 제안한 방법과의 성능 비교를 목표로 수행되었다.



(그림 5) 노드 수 변화에 따른 패킷도달률

(그림 5) 는 센서 네트워크의 노드 수 변화에 따른 패킷도달률을 나타낸 그래프이다. 두 가지 기법 모두 노드의 수가 많아 질수록 패킷 도달률이 높아지는 것을 알 수 있으며 우리가 제안한 기법이 기존의 플러딩 기법에 비하여 패킷도달률이 감소하지 않음을 확인할 수 있다.

5. 결론

우리가 제안하는 방식에서는 Self-pruning 기법을 기반으로 Node list 를 구성하며 각 Node list 는 soft-state 의 성격을 가지므로 네트워크의 크기에 상관없이 패킷에 주변 정보만을 담게 하여 불필요한 전송을 줄이면서도 무분별한 패킷의 크기 증가를 막을 수 있다.

성능분석과 시뮬레이션을 통한 성능 평가는 우리가 제안하는 기법이 기존의 방법에 비하여 더 나은 에너지 소비를 보이면서도 패킷이 전체 네트워크에 도달하는 비율인 패킷 도달률에서는 기존의 방법과 차이가 없음을 확인 할수 있었다. 하지만, 보다 나은 확장성을 지원하기 위해서는 노드 수 증가에 따른 에너지 소비율의 감소를 제공하기 위해서는 연구의 보완이 아직 필요하다고 생각된다.

참고문헌

[1] W. Peng and X. Lu, “On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks,” Proc. ACM Int’l Symp. Mobile and Ad Hoc 네트워크 ing and Computing, pp. 129-130, 2000

[2] H J. Wu and F. Dai, “Broadcasting in Ad Hoc Networks Based on Self-Pruning,” Proc. Infocom, pp. 2240-2250, 2003