

SPEED-MAC: 무선 센서 네트워크 환경에서 전송성능을 향상시키기 위한 에너지 효율적인 MAC 프로토콜

이현준[○] 이상훈 최린
고려대학교

kelly1104@korea.ac.kr[○], smile97@korea.ac.kr, lchoi@korea.ac.kr

SPEED-MAC: Speedy and Energy Efficient Data Delivery MAC Protocol for Wireless Sensor Network

Hyunjun Lee[○] Sanghoon Lee Lynn Choi
Korea University

요 약

본 논문에서는 휴면 지연에 의한 전송지연을 줄이는 효과적인 방법으로 staggered schedule 방식에 기반하여 동작하는 SPEED-MAC 프로토콜을 제안하고자 한다. SPEED-MAC은 이벤트 발생에 따른 정보를 발생사실과 세부정보로 나누어 각각 이벤트 공지 구간과 데이터 전송 구간에 싱크 노드로 전송함으로써 이벤트 발생 사실을 싱크 노드에게 최단시간에 전송할 수 있다. 또한 제어 패킷의 오버헤드를 줄이기 위해 이벤트 공지 구간에서 signal 패킷을 사용함으로써 데이터 패킷을 전송할 때 발생 가능한 충돌을 예측하였고 제어 패킷을 동적으로 사용하여 노드의 에너지 효율성을 추가로 개선할 수 있게 되었다. 본 논문에서는 SPEED-MAC을 MICA MOTE에서 구현하였으며 기존의 MAC 프로토콜에 비하여 메시지 전송 지연시간을 대폭 단축시킬 수 있었으며 또한 저 전력을 구현할 수 있는 것을 확인 하였다.

1. 서 론

센서 네트워크는 Ad-Hoc 네트워크와 매우 유사하지만 각각의 노드의 크기가 작기 때문에 배터리와 메모리 크기가 작고 그 용량에 한계가 있다[1]. 대부분의 센서 네트워크는 노드가 배치된 후 자가 구성으로 망을 형성하며 장기간의 동작을 요구하므로 에너지 효율성은 센서 네트워크 설계에 있어서 가장 중요한 고려사항 중 하나이다. 대부분의 센서 네트워크 MAC 프로토콜은 sleep 모드를 주기적으로 주입하여 idle listening으로 인한 에너지 낭비 문제를 개선하였다. 하지만 낮은 듀티 사이클은 패킷의 전송 지연을 급격히 증가시킨다. 즉, 전송 성능과 에너지 효율성은 반비례 하는 것으로 알려져 있다. 그런데 이벤트의 발생을 가능한 빨리 보고해야 하는 산 불탐지와 같은 응용에서는 전송 성능을 보장해야 할 뿐만 아니라 센서노드의 배터리교체가 어렵기 때문에 장시간의 동작을 보장해야 한다. 기존의 센서네트워크의 MAC프로토콜은 상호 배타적인 것으로 알려진 전송 성능과 에너지 효율성을 동시에 만족하기 어려운 문제가 있다.

이러한 문제를 해결하기 위해 staggered schedule 방식을 제안한 D-MAC[2]은 고정된 노드들이 데이터 수집 트리의 depth에 따라 순차적으로 wakeup하여 데이터를 전송한다. 이는 노드의 전송 순서 및 wakeup 시간을 미리 맞추어놓음으로써 휴면 지연에 의한 전송지연을 제거하고자 하는 방법이다. 하지만 D-MAC은 장애물이 존재하지 않는 실제 환경에는 적용하기 어려운 이상적인 환경을 가정하였다.

본 논문에서는 staggered schedule을 사용하여 휴면 지연으로 인한 전송지연 문제와 이벤트 발생 사실과 그에 따른 세부 정보를 분리하여 전송함으로써 제어 패킷 네트워크 상황에 따라 동적으로 사용하여 제어 패킷의 오버헤드 문제를 해결하고 D-MAC의 문제점인 hidden terminal 문제를 해결한 SPEED-MAC이라는 MAC프로토콜을 제안한다.

2. 관련 연구

2.1 D-MAC

D-MAC은 data gathering tree를 통해 데이터를 전송하는 센서 네트워크에서 높은 전송률과 에너지 효율성을 동시에 제공하고자 제안되었다. D-MAC의 가장 큰 특징은 고정된 노드들로 구성된 센서 네트워크에서 각 노드가 싱크 노드로부터의 hop수에 따라 순차적으로 wakeup하는 staggered schedule 방식을 사용하는 것이다. 그림 1에서와 같이 각 노드는 자신의 depth에 맞추어 순차적으로 수신 구간과 전송 구간에 동기화 되어 wakeup함으로써 매 hop마다 발생하는 휴먼 지연을 제거하였다[3]. D-MAC은 staggered schedule을 사용함으로써 동일 메시지 전송에 의한 채널 획득 경쟁을 제거할 수 있는 장점을 가지고 있으며, 채널 획득을 하기 위해 사용되는 RTS/CTS와 같은 제어 패킷을 제거하여 노드의 에너지 효율성을 높이고자 하였다.

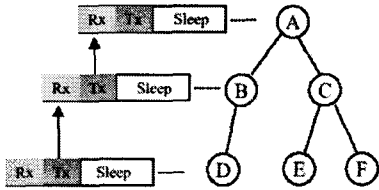


그림 1. D-MAC의 schedule

2.2 D-MAC의 문제점

무선 네트워크에서는 전송 채널의 특성상 유선 네트워크에 비해 높은 에러 발생률을 보인다. 에러는 noise에 의해 발생하는데, 그 원인에 따라 다수의 노드에서 전송된 신호들이 서로 간섭을 일으켜 발생하는 Multiple-access interference에 의한 것과 주위 환경에 따라 발생 빈도와 형태가 달라지며 그 세기 또한 일정하지 않은 Natural or artificial noise에 의한 것으로 구분할 수 있다.

2.2.1 Multiple-access interference

다수의 노드들이 동시에 패킷을 전송하고자 하는 경우 노드들 사이에서 채널을 획득하기 위한 conflict가 발생한다. 센서 네트워크에서 발생하는 conflict는 발생한 노드의 역할에 따라 contention과 collision으로 분류할 수 있다.

먼저 collision은 둘 이상의 송신 노드가 동일한 carrier sensing range에 속해 있을 때 발생하는 것으로 채널 경쟁에 참여한 노드들 중 채널을 획득한 노드만 전송을 하고 나머지 노드들은 전송을 하지 못한다. D-MAC과 같이 staggered schedule을 사용하는 경우에는 노드가 채널 획득에 실패하면 다음 cycle까지 기다려야하는 문제가 발생한다. D-MAC에서는 data prediction기법과 MTS(more to send)패킷을 이용하여 이러한 문제를 해결 하였다.

Collision은 수신 노드에서 발생하는 것으로 둘 이상의 송신 노드가 서로의 존재를 확인할 수 없을 때 발생한다. 그림 2에서 보이는바와 같이 두 송신 노드는 carrier sensing range 밖에 위치하고 있으므로 carrier sensing 후 패킷 전송을 시작한다. 그러나 수신 노드가 두 송신 노드의 전송 범위에 포함되므로 두 송신 노드가 전송한 패킷이 수신 노드에서 collision을 일으킨다. D-MAC은 control 패킷을 제거하여 에너지 효율성을 높이고자 하였으나 그림 2와 같은 경우의 collision문제를 해결하지 못하는 단점이 있다. D-MAC에서는 센서 노드의 transmission range와 carrier sensing range가 1:2의 비율을 가지는 이상적인 상황을 가정하고 있으나, 실제 센서네트워크 환경은 장애물로 인해 무선 채널 환경의 특성이 일정하지 않으므로 패킷 충돌 문제가 발생할 수 있다[4]. 특히 D-MAC에서는 충돌이 발생하면 해당 주기에 패킷을 전송하지 못하고 다음 전송 구간에 전송을 시도하는데 이 때에도 동일한 충돌이 발생하므로 결국 패킷이 정상적으로 전달되지 않는 문제가 있다.

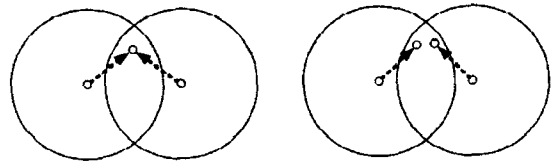


그림 2. 수신 노드에서의 충돌

2.2.2 Natural or artificial noise

다양한 장애물이 존재하는 무선 환경에서는 natural or artificial noise에 의한 bit error 발생 확률이 유선망에 비해 높다. D-MAC에서는 에러 발생시 송신 노드에서 해당 패킷을 재전송 하는 방식인 ARP를 사용하는데, bit error가 발생한 경우 다음 wakeup구간에 해당 패킷을 재전송한다. 그러나 단일 패킷을 전송하는 도중 에러가

발생하거나 다수의 패킷으로 이루어진 메시지를 전송할 때 첫 번째 패킷에서 에러가 발생하는 경우 전송 경로상의 노드들이 추가적인 wakeup slot을 할당받지 않고 다음 주기까지 long sleep 상태에 들어가므로 패킷지연이 급격히 증가하는 문제가 있다. 즉, 에러 발생 확률이 높은 환경에서 D-MAC은 빈번한 전송 실패로 인해 latency가 증가할 수 있으며, 이로 인해 실제 환경에서는 응용이 요구하는 전송 성능을 지원하지 못할 수 있다.

3. SPEED-MAC

본 논문에서는 D-MAC의 staggered schedule 기법에 기초하여 SPEED-MAC(SPeedy and Energy Efficient Date Delivery MAC)이라는 새로운 MAC 프로토콜을 제안한다. SPEED-MAC에서는 D-MAC에서 해결하지 못한 conflict 및 에러 문제를 해결함과 동시에 이벤트 발생을 최단시간에 알리고 에너지 소비량을 줄이기 위해 fast announcement와 selective wakeup 기법을 사용한다.

Fast announcement 기법은 이벤트를 감지한 노드가 전송 경로상의 노드들에게 signal 패킷을 사용하여 이벤트 발생 사실을 미리 알려줘서 이후에 데이터를 전송하는데 있어서 전송지연을 줄이도록 하기 위한 방법이다. 이를 위해 노드의 cycle time을 이벤트 발생을 알리는 이벤트 공지 구간과 데이터 전송 구간으로 분리한다. 이와 함께 selective wakeup 기법은 이벤트 발생을 알리는 signal의 수신 여부에 따라 데이터 전송 구간에서의 wakeup 여부를 결정하는 방식으로 노드의 idle listening을 최소화 할 수 있도록 하는 기법이다.

3.1 Basic scheme

3.1.1 Cycle time

SPEED-MAC에서는 이벤트의 발생을 알리는 fast announce 기법과 그에 따른 데이터 전송 구간에서 동작을 제어하는 selective wakeup 기법을 위해 이벤트 공지 구간과 데이터 전송 구간에서 다른 cycle time을 사용한다. 그림 3은 SPEED-MAC의 cycle time을 보여준다. 각 구간은 signal 슬롯과 data 슬롯 단위로 설정되는데, signal 슬롯의 수는 전체 네트워크 내의 최대 depth에 따라 설정되며 data 슬롯의 수는 최대 depth와 응용의 latency tolerance 및 메시지의 크기에 따라 설정된다. 따라서 D-MAC에서는 고정된 노드로 구성된 트리 형태의 라우팅 프로토콜을 사용하는 것으로 가정한다. cycle

time의 길이를 결정짓는 요인 중 최대 depth는 센서 노드의 배치가 완료된 후 생성되는 라우팅 경로에 따라 달라진다. 그러므로 cycle time을 결정하기 위해서는 네트워크 설치 후 depth 정보를 수집하고, 최대 depth를 네트워크 내의 노드들에게 알려주어야 한다. 이는 네트워크 초기화 과정에 진행되는데 라우팅 트리 구축, depth 정보 수집 및 최대 depth의 전파의 세 단계로 진행된다. 싱크노드에 의해 라우팅 트리가 구축이 된 후 depth 정보를 수집하는 단계에서는 트리 내의 모든 리프노드들이 자신의 depth 정보를 받았음을 확인하고, depth 정보를 전송하는 횟수를 줄이기 위해 각 노드는 자신의 모든 자식노드의 depth 보고가 끝난 뒤 그 중 가장 큰 값을 부모 노드에게 전달한다. 싱크 노드는 자신의 모든 자식노드로부터 depth 보고를 받으면 정보 수집이 완료되었음을 확인하고 최대 depth 값을 staggered schedule의 초기 시작 시간과 함께 네트워크 내의 모든 노드들에게 전파함으로써 각 노드가 동일한 시간에 staggered schedule에 따른 동작을 시작할 수 있도록 한다.

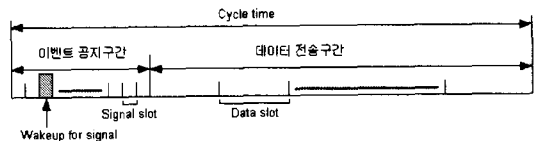


그림 3. SPEED-MAC의 사이클 시간 구성

3.1.2 Fast announcement

이벤트 공지 구간은 signal 패킷을 전송함으로써 이벤트의 발생유무와 데이터 전송 구간에서 충돌상황을 미리 예측하여 동적으로 제어 패킷을 사용하기 위한 구간이다. 각 노드는 signal을 수신하기 위해 이벤트 공지 구간에서 자신의 depth에 맞추어 적절한 signal 슬롯에 wakeup하여 자식 노드로부터 전송되는 signal 패킷을 수신한다. 이 때 이벤트 발생 보고는 리프노드에서 싱크노드로 향하므로 depth가 큰 노드부터 낮은 노드로 순차적으로 wakeup 구간을 가진다. 이벤트를 감지한 노드는 자신의 signal 슬롯에서 signal 패킷을 브로드캐스트 할 수 있으며 자신의 wakeup 구간 다음이 signal 브로드캐스트 슬롯이 된다. 이 슬롯에 노드는 자신의 자식 노드로부터 전송된 signal을 전송하거나 자신이 감지한 이벤트에 대한 signal을 전송한다. Signal은 단순히 이벤트의 발생을 알리기 위한 수단으로 사용되기 때문에 signal을 전송하는 노드의 주소만을 포함하는 짧은 패킷을 이용한

다. 이벤트 공지 구간의 사용은 이벤트 발생 사실과 이벤트의 세부적인 정보를 분리하는데 또 다른 의의가 있다. 화재 발생 혹은 침입감지와 같은 응용은 이벤트 발생 사실을 최대한 빠른 시간내에 알려야 하는데, 이벤트 발생에 따라 보고되는 데이터의 크기가 큰 경우에는 데이터 전송이 빠르게 전달하지 않을 수 있다. 그러므로 이벤트 발생 사실과 이벤트에 대한 세부 정보를 분리함으로써 싱크노드에게 이벤트 발생을 알리는 시간을 줄일 수 있다.

3.1.3 Selective wakeup

데이터 전송 구간에서 노드의 wakeup은 signal의 수신 여부에 따라 결정된다. 즉, 트래픽이 없는 경우에는 signal이 전송되지 않으므로 이벤트 공지 구간에만 활동하며, 트래픽이 발생한 경우에는 signal을 송수신한 노드들은 데이터 전송 구간에도 활동하여 데이터 패킷을 전달한다. 이러한 방식을 selective wakeup 기법이라 하며, 이를 통해 노드의 idle listening으로 인한 에너지 낭비를 줄일 수 있다. 데이터 전송 구간에서의 메시지 전송이 끝나면 전송 경로상의 노드들은 다음 이벤트 공지 구간까지 sleep 상태로 들어가야 하는데, 이를 위해 데이터 패킷 헤더에 end 플래그를 포함하여 소스노드가 경로상의 노드들에게 마지막 패킷임을 알려주도록 한다.

3.2 Error problem

D-MAC과 마찬가지로 SPEED-MAC도 multiple-access 및 natural or artificial noise에 의한 signal 혹은 데이터 손실 문제가 발생한다. 하지만 SPEED-MAC에서는 이벤트 공지 구간의 signal 패킷의 정보를 이용하여 동적으로 제어 패킷을 사용함으로써 D-MAC의 문제점을 해결하였다. 또한 collision에 의한 에러와 noise에 의한 에러를 구별함으로써 불필요한 제어 패킷의 사용으로 인한 에너지 소모와 전송 지연 시간을 줄일 수 있었다.

3.2.1 Error signal

이벤트 공지 구간에 에러가 발생하면 에러 발생 직후 이후의 노드는 데이터 전송 구간에 활동을 하지 못하게 된다. 이를 해결하기 위해 이벤트 발생을 알리는 signal에 에러 발생을 표시하기 위한 error bit을 추가한다. 이벤트 공지 구간에 에러를 감지한 노드는 다수의 노드가 동시에 signal을 전송하거나 noise에 의한 에러가 발생

한 것으로 간주한다. 특히 multiple-access interference에 의한 에러가 발생한 경우에는 동일한 형태의 에러가 데이터 전송 구간에도 발생할 것이다. 그러므로 error signal을 받은 노드들은 데이터 전송 구간에 RTS/CTS 제어 패킷을 사용함으로써 conflict를 회피할 수 있도록 한다. 그림 4는 collision이 발생한 경우 각 노드의 동작을 보여준다. collision을 감지한 노드는 몇 개의 signal이 충돌을 일으켰는데 모르므로 데이터 전송 구간에 몇 개의 메시지가 전송할 것인지를 모른다. 그러므로 collision이 발생한 노드 이후의 전송 경로에 속하는 노드들은 미리 설정된 타이머가 만료될 때까지 추가로 전송할 데이터 패킷의 유무를 확인한다.

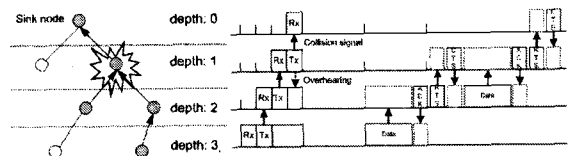


그림 4. 충돌이 발생한 경우의 노드의 동작

3.2.2 Collision과 Noise에 의한 에러 구분

SPEED-MAC에서는 이벤트 공지 구간에서 signal 패킷의 에러 종류를 collision에 의한 에러와 noise에 의한 에러로 구분함으로써 데이터 전송 구간에서 불필요한 제어 패킷의 사용을 제거함으로써 제어 패킷의 사용으로 인한 여러 가지 오버헤드를 감소시켰다. 수신된 패킷 신호강도를 수집하여 임계값 이상의 신호강도에 대해서는 collision에 의한 에러라고 구별하였고, 그 외에는 natural or artificial noise에 의한 에러라고 구별하였다. 두 종류의 에러를 구별함으로써 데이터 전송 구간에서 signal 패킷에 에러가 발생한 경우 collision에 의한 에러에 한해서만 데이터 전송 구간에 제어 패킷을 사용하면 되므로 전송 지연 및 에너지 소모를 줄이는 효과를 확인할 수 있었다.

4. 실험 및 분석

Hop 수에 따른 메시지 지연시간과 메시지 간격에 따른 에너지 소모량을 판단하기 위하여 MICA MOTE를 이용하여 실험하였다. 네트워크는 최대 hop 수를 동일하게 하기 위해 S-MAC의 경우 일직선으로 놓여진 7개의 노드로 구성하였고, SPEED-MAC과 D-MAC의 경우 3개의

소스 노드와 1개의 싱크 노드, 6개의 중계 노드로 이루어진 2진 트리 형태로 구성하였다. MICA MOTE 실험에 사용된 파라미터 값으로 데이터 패킷은 100Bytes, 제어 패킷은 10Bytes, signal 패킷은 6Byte, 싱크 패킷과 데이터 패킷을 위한 contention window의 크기는 각 15, 31 슬롯으로 설정하였으며, 실험시간은 각 20분으로 하였다. 그림 5는 hop 수에 따른 평균 지연 시간을 보여준다. 10% 듀티 사이클로 동작하는 S-MAC은 홉 수가 증가함에 따라 전송지연이 확연하게 드러나는데 비해서 staggered wakeup schedule을 사용하는 D-MAC과 SPEED-MAC이 전송 지연이 적은 것을 확인할 수 있었다. 또한 signal 패킷의 latency가 가장 적게 나오는 사실로부터 이벤트의 발생 사실을 최단시간에 알려줄 수 있음을 확인할 수 있었다. 그림 6은 메시지 간격에 따른 에너지 소모량을 보여 주는데 그림에서 보이는 바와 같이 collision발생 시 동적으로 제어 패킷을 사용하여 메시지 재전송 횟수를 줄이는 SPEED-MAC이 D-MAC에 비해서 에너지 소모가 적음을 확인할 수 있었다.

5. 결론 및 향후과제

SPEED-MAC은 D-MAC과 같이 staggered wakeup schedule을 사용함으로써 노드의 sleep으로 인한 전송 성능 저하를 해결한다. 또한 이벤트 발생에 대한 fast announcement기법과 selective wakeup기법을 통해 이벤트 보고 시간을 단축하고, 노드의 에너지 소비를 줄일 수 있다.

D-MAC은 실제 환경에서 발생할 수 있는 패킷 충돌 문제를 해결하지 못하였고, 이로 인해 실제 응용에는 적용하기 어렵다는 단점이 있었다. SPEED-MAC은 D-MAC의 이러한 문제를 해결하고, 추가로 이벤트 보고 시간을 단축하고 에너지 효율성을 높일 수 있는 방법을 제안한다. 특히 패킷 충돌 문제를 해결함으로써 실제 응용에 적용 가능한 MAC 프로토콜이라는 점에서 의의를 가진다.

6. 참고문헌

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankara-subramaniam, and Erdal Cayirci, Georgia Institute of Technology, "A Survey on Sensor Networks," *IEEE Communication Magazine*, Aug.2002.
 [2] Gang Lu, Bhaskar Krishnamachari, Cauligi S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks," in *IProceedings of the 18th International Parallel and Distributed Processing Symposium(IPDPS'04)*
 [3] Jeremy Elson, Lewis Girod and Deborah Estrin "Fine-Grained Network Time Synchronization using Reference Broadcasts," in *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation(OSDI2002)*.
 [4]Gang Zhou, Tian He, John A. Stankovic,Tarek Abdelzaher, "RID: Radio Interference Detection in Wireless Sensor Networks," *IEEE Infocom 2005*.

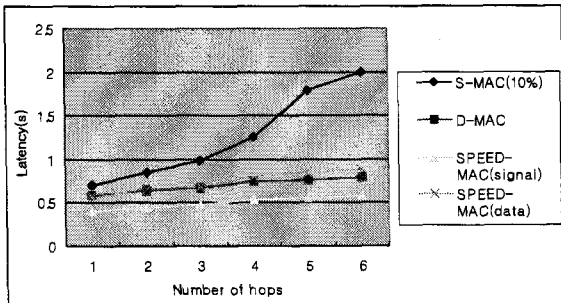


그림 5. hop수에 따른 평균지연시간

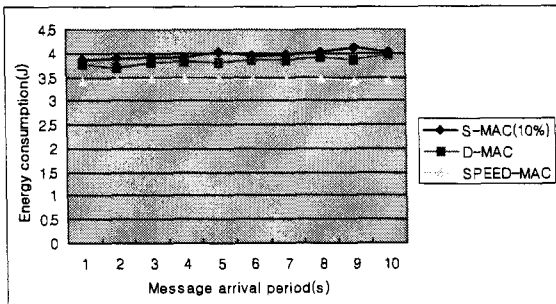


그림 6. 메시지 간격에 따른 평균에너지 소모량