

데이터 전송 지연 최소화 기법

서창수^o, 고영배, 김재훈
 아주대학교 정보통신전문대학원
 changsu@dmc.ajou.ca.kr^o, {youngko, jaikim}@ajou.ac.kr

A Method for Minimizing Data Delay in Energy Efficient Sensor MAC protocols

ChangSu Suh^o, Young-Bae Ko, Jai-Hoon Kim
 Graduate School of Information and Communication, Ajou University

요 약

최근 무선 통신 기술과 전자 기술의 진보로 지능형 센서(smart sensor)를 이용한 무선 센서 네트워크 구축이 가능해졌고, 이러한 센서 네트워크는 최근 유비쿼터스 컴퓨팅과 Ad-Hoc 네트워크에서 많이 연구되고 있는 새로운 이슈이다. 최근 센서관련 논문에서는 각 센서 노드의 제한된 에너지를 효율적으로 사용하기 위해 데이터 전송이 없을 경우 노드의 통신 모듈을 sleep 시킴으로써 에너지를 절약 한다. 이를 위해서 모든 센서 노드들은 동기화를 맞추어 listen과 sleep를 주기적으로 반복해야 한다. 이러한 프로토콜들에서 발생하는 가장 큰 문제는 에너지 효율성과 데이터 전송 지연간의 반비례 관계이다. 에너지의 효율성을 위해 sleep 시간을 길게 할수록, sleep 시간에서 발생한 데이터의 지연 시간도 길어지게 되므로, 데이터의 지연을 고려하여 sleep 시간을 적당히 조정해야 필요가 있다. 본 논문에서는 에너지 효율성을 위해 listen/sleep 주기를 사용하면서도 데이터 지연을 최소화 할 수 있는 효율적 sleep 시간에 대해 논의 해 보려 한다.

1. 서 론

최근 무선 통신 및 전자 기술의 발전은 통신 기능과 컴퓨팅 기능을 동시에 갖는 스마트 센서의 개발을 가능하게 하였고, 이러한 센서들을 이용한 무선 센서 네트워크는 환경, 의료, 군대, 홈 네트워크 등 여러 응용분야에서 광범위하게 사용되고 있다. 이러한 센서 네트워크는 일반적으로 많은 수의 센서 노드들로 구성되며, 각 노드는 멀티 홉 형식의 무선 통신을 하고, 건전지 전력을 기반으로 동작한다 [3].

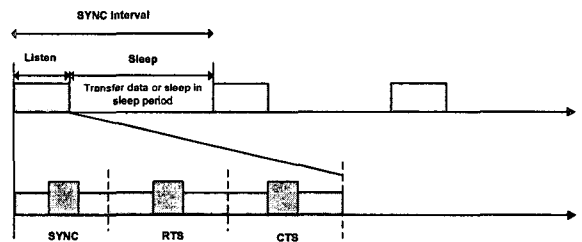
네트워크 계층 구조에서 통신에 기본이 되는 계층은 MAC 계층이다. 무선 네트워크에서의 MAC 프로토콜들은 최대 전송량에 초점이 맞추어지지만, 무선 센서 네트워크의 MAC 프로토콜들은 최소 에너지 소모에 초점이 맞추어진다. 일반적으로 MAC 계층에서의 주요 에너지 소모 요소들은 다음과 같다. [4].

- Collision: 충돌에 의한 재전송은 에너지 소모 및 지연을 유발한다.
- Overhearing: 다른 노드에게 보낸 패킷을 받을 경우 불필요한 수신 에너지가 소모 된다.
- Control 패킷: 정교한 데이터 송수신을 위한 복잡한 컨트롤 패킷의 사용은 노드의 에너지를 낭비시킨다.
- Idle listening: 수신 데이터의 유무를 떠나, 계속해서 채널을 감시하는 것은 불필요한 에너지 소비를 의미한다.

MAC 계층에서의 주요 에너지 소모 요소 중, 가장 많은 에너지를 소모하는 것은 idle listening 이다. 기존의 논문(S-MAC [1])에서는 listen/sleep 주기를 통해, 송수신에 참여하지

listening 문제를 해결하였다. 다른 주요 에너지 소모 요소인 overhearing과 collision들은 RTS/CTS 기술을 사용하여 해결할 수 있다.

그림 1은 idle listening 문제를 처리하기 위해서 사용되는 S-MAC의 listen/sleep 주기이다. S-MAC의 주기는 크게 컨트롤 패킷 전송하기 위한 시간인 listen 시간과 데이터 전송 혹은 sleep을 위한 sleep 시간으로 나누어진다. listen 시간 동안 모든 노드들이 깨어서 컨트롤 패킷(SYNC, RTS, CTS)을 통해 동기화 및 데이터 존재유무를 확인하고, sleep 시간에는 데이터의 존재가 있을 경우 데이터를 전송하고, 없을 경우는 sleep을 하므로써 에너지를 절약한다. 그림 1과 같은 주기는 센서 네트워크의 S-MAC뿐만 아니라, 무선 랜의 802.11 에너지 절약 모드(power saving mode)에서도 유사하다.



<그림 1 SMAC의 주기>

위와 같은 주기를 갖는 기술들의 문제점은 sleep 시간 동안 발생한 데이터의 전송 지연이다. 한번 sleep을 하면 다음 listen 시간 동안 sleep 상태를 유지해야 하므로, 어쩔 수 없이 다음 listen이 되어야만 데이터를 전송할 수 있다. 센서 네트워크에서 효율적 에너지 소모를 가장 중요시 한다 하더라도, 데이터 전송 지연 역시 무시할 수 없는 주요 요소이다. 이에 본 논문에서는 listen/sleep 주기를 사용하면서도 최소의 데이터 지연을 갖도록 하는 sleep 시간은 어떤 것인지에 대해 연

* 본 연구는 정보통신연구진흥원의 국제공동연구 지원사업과 한국학술진흥재단의 지원사업(KRF-2003-003-D00375) 연구결과로 수행 되었음.

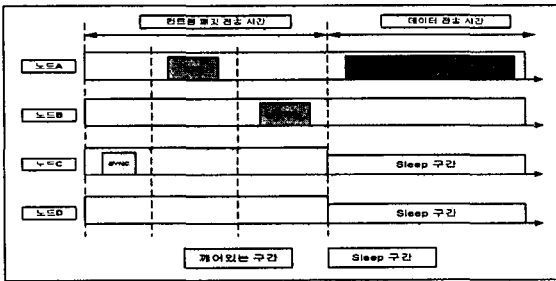
구하였다.

2. 관련 연구

센서 네트워크의 MAC 디자인에서 가장 중요한 이슈는 에너지 효율적 사용이며, 이를 위해서는 listen/sleep의 주기성은 꼭 필요하다. 하지만, sleep에 의해 발생하는 데이터 지연은 무시할 수 없는 문제이다.

2.1 S-MAC

앞에서 설명했듯이, S-MAC에서는 sleep기술을 이용하여, 불필요한 에너지 낭비를 최소화한다. 만약 4개의 노드 중 노드 A가 노드 B에게 데이터를 전송한다고 가정했을 때, S-MAC에서의 노드 상태는 아래 그림과 같다.



<그림 2 S-MAC>

Listen 시간 즉, 컨트롤 패킷 전송시간에 처음 전송되는 패킷은 노드들의 주기를 맞추기 위한 SYNC 패킷이다. 그 후, 노드 A가 노드B에게 데이터를 전송하기 위해 먼저 RTS 패킷을 보내고, RTS 패킷을 받은 노드 B는 전송을 허락하는 의미에서 노드 A에게 CTS를 보낸다. CTS 전송을 끝으로, 전송에 참여하는 노드 A, B를 제외한 나머지 노드는 sleep 향으로써 에너지를 절약한다.

2.2 Duty Cycle

Duty cycle이란 말은 한 주기(SYNC interval)에서의 listen 시간이 차지하는 비율을 의미하며, 수식 표현은 아래 식(1)과 같다. listen time을 구성하는 SYNC, RTS, CTS 전송 시간은 컨트롤 패킷과 contention window의 크기에 의해 고정되어진다. 이 고정된 값을 통해 listen time을 구할 수 있고, 이 listen time과 Duty cycle을 이용하여 SYNC interval 역시 구할 수 있다. sleep time은 SYNC interval에서 listen time을 빼서 구해진다. (식2)

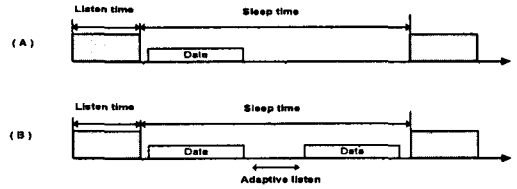
$$\begin{aligned} \text{Duty cycle} &= \text{Listen time} * 100 / \text{SYNC interval} && \text{식(1)} \\ \text{Sleep time} &= \text{SYNC interval} - \text{Listen time} && \text{식(2)} \end{aligned}$$

Duty cycle이 높을수록, 한 주기에서 차지하는 listen time의 비율이 증가하고, sleep time의 감소한다. 반대로 Duty cycle이 낮을수록, sleep time은 증가된다. sleep time은 데이터 지연 문제와 깊은 관련이 있기 때문에, 결국 Duty cycle을 어떻게 선택하냐에 따라 데이터 지연의 크기도 변화된다. 결국 유저는 센서가 설치된 환경 및 application을 고려하여 적당한 Duty cycle를 선택해야 한다.

2.3 Adaptive listen

논문[2]에서는 데이터 지연 문제를 극복하기 위하여

adaptive listen이라는 시간을 사용하였다. 일반적인 S-MAC(그림 3의 A)에서는 한 주기 동안 하나의 data가 전송되는데 비해, 논문[2](그림 3의 B)에서는 Data 전송이 끝나는 후에 또 전송할 데이터가 존재 할 경우 adaptive listen 시간 동안 RTS와 CTS를 교환한 후에 그 데이터를 전송한다. 노드들은 listen time에서 전송된 RTS나 CTS의 NAV을 통해 첫 데이터 전송이 끝나는 시간을 알 수 있다. 논문[2]는 한 주기 동안 여러 데이터를 전송 가능하도록 하므로써, 어느 정도 데이터 지연 문제를 해결하였다.



<그림 3 adaptive listen>

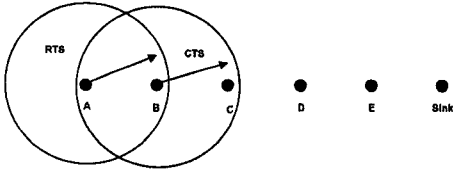
2.4 DSMAC

일반적으로 listen/sleep의 주기를 갖는 프로토콜에서는 duty cycle 미리 정의하여 통신을 하는데 비해, DSMAC[3]는 데이터 트래픽을 고려해 동적으로 duty cycle을 변화시킴으로써 데이터 지연을 줄였다.

3. 효과적인 Sleep time

본 논문에서는 에너지 소모의 효율성을 위해 listen/ sleep 주기를 사용하면서도, 데이터 지연이 최소화 될 수 있는 sleep time을 제시한다. 일반적으로 listen/sleep 주기를 갖는 프로토콜들은 아래와 같은 특징을 갖는다.

- Adaptive listen을 사용하는 프로토콜은 Listen time에서 교환된 RTS와 CTS 패킷의 NAV 시간을 이용하여, 데이터 전송이 끝나는 순간에 일어나서 다른 데이터의 교환을 가능하게 한다.
- RTS/CTS를 사용한다는 것은 유니 캐스트를 의미하며, 결국 특정 라우팅 경로에 의해 sink쪽으로 전달되는 과정임을 말한다. 즉, 유니 캐스트에 의해 다른 노드로부터 받은 데이터는 결국 다른 노드에게 전달되어야 함을 뜻한다.
- Listen time에서 RTS와 CTS에 의해 데이터의 존재여부를 아는 것은 데이터를 발생시키는 노드로부터 2 홉까지이다. (그림 4에서는 노드 C까지이다.)
- Listen time에서 RTS를 전송하는 노드로부터 2 홉 이상 떨어져 있는 노드들은 어떤 컨트롤 패킷도 받지 못하므로 Adaptive listen 시간에 일어날 수 없다. (NAV을 알 수 없으므로)
- Duty cycle의 크기가 동적으로 변화하더라도, Listen time에서 RTS를 전송하는 노드의 데이터의 트래픽에 영향을 받는 건 2 홉까지일 뿐이다.



<그림 4 노드 A에서 데이터 발생>

위 특징들과 그림 4를 연관 시켜 생각해보면, adaptive listen을 사용하더라도 데이터는 A->B, B->C 이렇게 두 번만 전송된다. 이는, Listen time에서 RTS와 CTS를 받을 수 있는 노드가 C까지이므로 나머지 노드들은 NAV 정보가 없어 중간에 깨지 못하기 때문이다. 즉, 두 번의 데이터 전송 시간과 그에 따른 컨트롤 패킷 시간을 고려한 sleep time이 데이터 지연을 고려한 최적의 sleep time이 되는 것이다. DSMAC처럼 duty cycle을 조절하여 sleep 시간을 변화시킨다 하더라도 한 주기에 전송되는 데이터는 2개 이상이 될 수 없기 때문에 데이터 지연에 크게 영향을 주지 못한다. 오히려 duty cycle의 동적인 변화는 노드들간의 동기화에 영향을 미칠 수도 있으므로 주의해야 한다. 그렇기 때문에 에너지 효율성을 위하여 listen/ sleep 주기를 사용하면서도 데이터 지연을 최소화 할 수 있는 최적의 sleep time은 아래와 같다. (한번의 데이터 전송은 RTS-CTS-DATA-ACK 이다.)

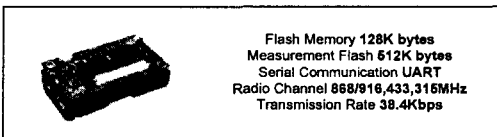
$$Listen\ time = SYNC + RTS + CTS\ 전송시간$$

$$Sleep\ time = DATA + ACK + RTS + CTS + DATA + ACK\ 전송시간$$

본 논문에서는 단순히 duty cycle을 조절하여 sleep time을 정하는 것이 아니라, 데이터 흐름과 컨트롤 패킷의 전달 범위를 고려하여 최적의 sleep time을 정하므로써, end-to-end 지연과 sink 노드에서의 throughput 향상을 가져 올 수 있다. 다음 장에서는 실제 센서 노드에서의 실제로 정의되는 sleep time 값에 대해 알아보겠다.

4. 실제 센서 노드에서의 Sleep time

현재 센서 네트워크에 가장 많이 사용되고 있는 하드웨어는 버클리에서 개발한 Mote Mica이다[5, 6]. 이 센서 노드는 무선 통신 및 컴퓨팅 기능을 동시에 가지고 있으며, TinyOs 라는 경량급 운영체제에 의하여 동작된다.



<그림 6 Mote Mica>

현재 판매되는 최신 Mote Mica에는 S-MAC이 구현되어 있다. 우리는 Mica에 구현 되어 있는 S-MAC을 근거로 시간들을 근거로 최적의 sleep 시간을 사용할 때와 default인 10% duty cycle을 사용했을 때의 데이터 지연을 수식으로 계산해보겠다.

전송 시간	값
SYNC	41 ms
RTS	53 ms
CTS, ACK	21 ms
DATA	83

<표 1 실제 Mote Mica에서의 주요 전송시간들>

위 표1의 시간은 각 패킷을 전송 시간뿐만아니라, 프로세스 시간, delay, guard 시간 등, 여러 값들을 고려한 시간이다. 실제 Mote Mica에서 사용되는 전송시간들을 근거로 Listen 시간과 sleep 시간에 계산해 보면 다음과 같다.

$$Listen\ time = SYNC + RTS + CTS = 115ms$$

$$Sleep\ time = (Listen * 100 / duty_cycle) - listen$$

• duty_cycle이 10%일 때의 sleep time = 1036ms

본 논문에서 제시하는 최적의 sleep time을 계산한 값은 아래와 같다.

$$최적의\ Sleep\ time = DATA + ACK + RTS + CTS + DATA + ACK = 285ms$$

현재 Mote Mica에 구현되어 있는 S-MAC의 sleep 시간 (10% duty cycle)과 비교했을 때, 본 논문의 sleep time보다 무려 700ms 정도의 불필요한 시간을 가지고 있음을 알 수 있다. 이 시간은 결국 데이터 전송 지연을 발생 시킬 것이다.

Mote Mica 실제 센서 노드에서 listen/sleep의 주기를 사용하면 데이터의 지연을 최소화할 수 있는 최적의 sleep 시간은 위의 계산에서 나온 285ms이다.

5. 결론 및 향후 연구

본 논문에서는 listen/sleep의 주기를 사용하는 MAC 프로토콜에서 발생하는 데이터 지연 문제를 다루어 보았다. 한 주기에서 전송될 수 있는 데이터와 컨트롤 패킷의 전송 범위를 고려하여 데이터 지연 문제를 최소화 할 수 있는 효율적인 sleep time의 구하였다. 본 논문에서 제시한 효율적 sleep time을 통해, listen/sleep의 주기를 사용하여 에너지를 효율적으로 사용하면서도 동시에 데이터 지연 역시 최소화 할 수 있었다. 하지만, 본 논문에서 제안한 최적의 sleep time은 데이터의 재전송 등의 예외 상황을 고려하지 못했기 때문에, 예외 상황에서는 최소한의 데이터 지연을 보장할 수 없다. 향후 연구에서는 이러한 상황을 고려해서 좀 더 효율적인 sleep time에 대해 연구해 볼 계획이다.

[참고 문헌]

- [1] W. Ye, J. Heidemann, and D. Estrin. "An energy-efficient MAC protocol for wireless sensor networks," *INFOCOM 2002*, volume 3, pages 1567-1576, June 2002.
- [2] W. Ye, J. Heidemann, and D. Estrin "Medium access control with coordinated, adaptive sleeping for wireless sensor networks," in *IEEE Transactions on Networking*, June 2003.
- [3] P. Lin, C. Qiao, and X. Wang "Medium Access Control With A Dynamic Duty Cycle For Sensor Networks," *WCNC*, Mar 2004.
- [4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci "Survey on Sensor Networks," *IEEE Communications Magazine*, Aug. 2002.
- [5] <http://webs.cs.berkeley.edu/tos/>
- [6] <http://www.xbow.com/>