

# 무선 센서 네트워크상에서 시간지연에 민감한 데이터 전송을 위한 에너지 효율적인 MAC 프로토콜 설계

준회원 오형래\*, 정회원 송황준\*

## Design of Energy Efficient MAC Protocol for Delay Sensitive Application over Wireless Sensor Network

HyungRai Oh\* Associate Member, Hwangjun Song\* Regular Member

### 요약

본 논문은 센서 네트워크상에서 지연시간에 민감한 데이터 전송을 위한 에너지 효율적인 MAC 프로토콜을 제안한다. 비 동기식 센서 네트워크에서는 소비되는 에너지와 지연시간이 채널 모니터링 주기(Channel Monitoring Interval)와 데이터 센싱 주기(Sensing Data Interval)에 따라 변화한다. 이러한 사실을 기반으로 본 논문에서는 오버히어링(Overhearing)으로 발생하는 에너지 소비를 줄이기 위해 자신의 채널 모니터링 주기를 이웃노드에 알리는 새로운 프리앰블 구조와 낮은 연산 복잡도를 가지는 채널 모니터링 주기를 결정하는 알고리즘을 제안한다. 실험결과에서는 다른 센서 MAC 프로토콜과 제안한 MAC 프로토콜과의 성능비교를 보여준다.

**Key Words** : Wireless Sensor Network, Sensor MAC Protocol, Energy, Delay

### ABSTRACT

This paper presents an energy efficient MAC protocol for delay-sensitive data transmission over wireless sensor network. In general, energy consumption and delay depend on Channel Monitoring Interval and data sensing period at each sensor node. Based on this fact, we propose a new preamble structure to effectively advertise Channel Monitoring Interval and avoid the overhearing problem. In order to pursue an effective tradeoff between energy consumption and delay, we also develop a Channel Monitoring Interval determining algorithm that searches for a sub-optimal solution with a low computational complexity. Finally, experimental results are provided to compare the proposed MAC protocol with existing sensor MAC protocols.

### 1. 서론

센서 네트워크는 마이크로 컨트롤러, 적은양의 메모리, 배터리, 무선 송수신장치와 센싱 하드웨어로 구성된 많은 수의 노드들로 구성되며, 싱크 노드(sink node)로 센싱된 데이터를 보내기 위하여 자가 구성 및 멀티 홉 연결을 지원해야만 한다. 이러한

센서 네트워크는 군사, 홈 네트워크, u-헬스 등과 같은 여러 분야에서 광범위하게 활용되고 있다. 센서 네트워크에서 가장 중요한 문제는 적은 양의 에너지로 가능한 오랫동안 동작할 수 있는 에너지 효율성이다. 이를 위해 많은 센서 MAC 프로토콜들이 무선 송수신시 발생하는 에너지 소모를 고려하여 제안되었다.

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (NIPA-2009-C1090-0902-0006)

\* 포항공과대학교 컴퓨터공학과 멀티미디어통신시스템 연구실(raibest@postech.ac.kr, hwangjun@postech.ac.kr)  
 논문번호 : KICS2009-08-334, 접수일자 : 2009년 8월 4일, 최종논문접수일자 : 2009년 10월 21일

센서 MAC 프로토콜은 크게 동기화 방식과 비동기화 방식이 존재한다. 동기화 방식의 센서 MAC 프로토콜은 모든 센서 노드가 동시에 wake-up과 sleep상태로 전환한다. S-MAC<sup>[1]</sup>에서는 주기적으로 모든 노드가 sleep과 wake-up을 수행하여 에너지 소모를 줄이고 있다. 송신 노드는 깨어있는 동안 CSMA-CA방법을 이용하여 센싱된 데이터를 전송한다. 또한 에너지 소모를 줄이기 위해 저전력 RTS-CTS, 메시지 패싱과 동기화를 위한 SYNC 메시지를 사용한다. P-MAC<sup>[2]</sup>에서는 PRTF와 PETF를 통한 sleep 기간을 변경하여 에너지 소모를 줄이고 있다. PRTF는 센싱된 데이터를 전송하기 위해 사용되며, PETF는 변경된 duty cycle을 설정하기 위해서 사용되어진다. T-MAC<sup>[3]</sup>은 listen만 하고 있는 상태를 최소화하기 위해 시간제한을 이용하고 있다. STEM<sup>[4]</sup>과 같은 경우는 wake-up과 데이터를 전송하기 위해 서로 다른 채널을 이용한다. 송신노드는 wake-up 채널을 통해서 보낼 데이터가 있음을 알리고 ACK 메시지를 받으면 데이터를 전송하는 방법을 이용하여 에너지 소모를 줄이고 있다. Z-MAC<sup>[5]</sup>은 CSMA와 TDMA를 이용하여 에너지 효율, 채널 활용, 지연시간, 신뢰성 및 확장성을 향상시키고 있다. 802.15.4<sup>[6]</sup>는 현재 많이 활용되고 있는 MAC 프로토콜이다. 각 디바이스는 FFD(Full Fuction Device)와 RFD (Reduced Funtion Device)로 구분되어지며 FFD는 Personal Area Network (PAN)의 조정노드(Coordinator), 조정노드 또는 네트워크 디바이스로 동작될 수 있으며, RFD는 네트워크 디바이스로만 동작될 수 있다. 802.15.4 MAC 프로토콜이 다른 MAC 프로토콜과 가장 다른 점은 GTS(Guaranteed Time Slots)를 제공함으로써 종단간의 지연시간을 최소화 할 수 있다는 점이다. 하지만 다른 노드들과의 동기화를 위한 Beacon 메시지의 충돌문제가 발생 할 수 있다.

비동기화 MAC 프로토콜의 경우 각 노드들은 비동기식으로 wake-up을 하고, 채널의 상태를 살펴게 된다. 만약 채널이 busy하다면 센서노드는 수신 상태를 유지하게 되며, 채널이 idle하다면 sleep상태로 가거나 전송할 데이터를 보내게 된다. 비동기식 MAC 프로토콜들 중에서 가장 널리 알려진 BMAC<sup>[7]</sup>은 일반적인 동기화 방식의 MAC 프로토콜보다 단순한 구조를 가지고 있다. 하지만 BMAC의 가장 큰 문제점은 오버히어링이다. 오버히어링은 센서 노드에서의 추가적인 에너지 소모를 초래하며, 이 오버히어링을 해결하기 위해 여러 가지 MAC

프로토콜이 제안되었다. BMAC에서 오버히어링 문제를 줄이기 위해 제안된 BMAC+<sup>[8]</sup>는 연속된 짧은 프리앰블(Short Preamble)을 이용하여 오버히어링을 감소시켰다. X-MAC<sup>[9]</sup>에서는 짧은 프리앰블과 ACK을 이용하여 오버히어링을 감소시켰다. 이러한 MAC+[8]는 연에너지 소비를 최소화하기 위하여 제안되어졌으나, 센서네트워크에서 종단간 지연시간을 고려하고 있지 못하다는 문제점을 가지고 있다.

본 논문에서는 비동기식 MAC 프로토콜에서 에너지 소비와 지연시간이 채널 센싱 주기에 따라 변화되는 점을 이용하여 지연시간에 민감한 응용을 위한 에너지 효율적인 센서 MAC 프로토콜을 제안한다.

## II. 관련 연구

일반적으로 BMAC이 센싱 주기가 길거나 센서노드들의 밀도가 낮은 경우에 효과적인 것으로 알려져 있다. 그림 1은 BMAC 동작방식을 보여주고 있다.  $T_{pre}$ ,  $T_{CMI}$ 와  $F_{shift}$ 는 각각 프리앰블의 길이, 채널 모니터링 주기(Channel Monitoring Interval)와 최대 주파수 변화량(Maximum Frequency Shift)을 나타낸다. 센서 MAC 프로토콜이 정상 동작하기 위해서는 프리앰블의 길이가 CMI와 최대 주파수 변화량의 합보다는 커야한다. BMAC의 문제점중 하나는 그림 1에서 보여주는 것과 같이 수신노드가 아닌 노드가 listen 상태를 유지함에 의해 발생하는 오버히어링 문제이다. 오버히어링 문제는 오버히어링노드에서 추가적인 에너지 소비를 발생시킨다. 이 문제를 해결하기 위해서 BMAC+가 제안되었다. BMAC+의 경우 그림 2와 같이 반복된 짧은 프리앰블을 이용하여 오버히어링 문제를 해결한다. 짧은 프리앰블은 Preamble, Delimiter, Destination Address와 Peamble의 Countdown으로 구성된다. 만약 짧은 프리앰블을 받으면 수신노드가 아닌 노드들은 자신이 오버히어링하고 있음을 알게되 sleep 상태로 가게된다. 그리고 수신노드는 짧은 프리앰블의 Countdown 값을 보고 언제 데이터가

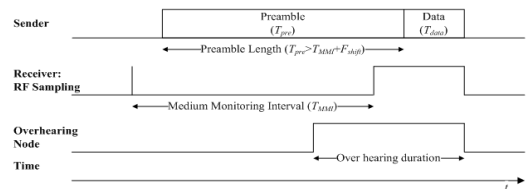


그림 1. BMAC 동작 예

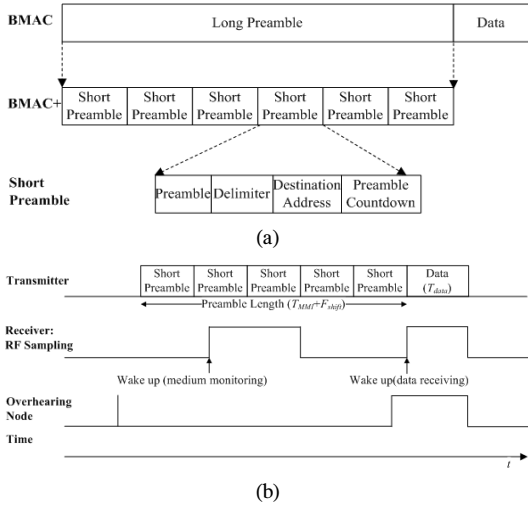


그림 2. Short preamble 구조와 BMAC+ 동작 방식  
(a) BMAC+ preamble 구조 (b) BMAC+ 동작방식

전송되는지를 확인링 문데이터 전송시기까지 sleep을 링 문데이터가 전송될 때 깨어나서 데이터를 받게 된다. 하지만 BMAC+ 역시 완벽하게 오버히어링을 회 피하지 못한다. 그림 2. (b)에서 보듯이 오버히어링 노 드가 데이터를 전송하기 바로직전에 wake up을 하게 될 때는 데이터가 전송이 완료될 때까지 오버히어링 을 하게 된다. 이 경우 만약 데이터의 전송시간이 긴 경우 추가적인 에너지 소모가 많이 발생하게 된다. 또 한 BMAC과 BMAC+ 모두 지연시간을 고려하지 않 고 있는 문제점을 가지고 있다. BMAC과 BMAC+ 모두 고정된 CMI를 사용하여 트래픽이 증가할 때 충돌 (Collision)이 발생하고 지연시간이 급격히 증가하게 된다. 본 논문에서는 CMI를 변화시켜 지연시간을 고 려할 뿐 아니라 에너지 소비를 최소화하기 위한 새로 운 MAC 프로토콜을 제안한다.

### III. 제안하는 센서 MAC 프로토콜

센서 네트워크에서 에너지 소비와 지연시간은 중 요한 문제점들이며, CMI 및 오버히어링과 밀접한 관련을 가지고 있다. 본 논문에서는 새로운 프리엠 블 구조와 CMI 선택 기법으로 구성된 새로운 MAC 프로토콜은 제안한다.

#### 2.1 제안하는 프리엠블 구조

그림 3은 제안하는 프리엠블 구조를 보여주고 있 다. 프리엠블은 반복된 짧은 프리엠블로 구성되며, 각 짧은 프리엠블은 다시 Preamble, Delimiter,

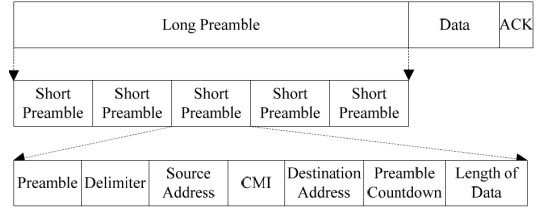


그림 3. 제안하는 프리엠블 구조

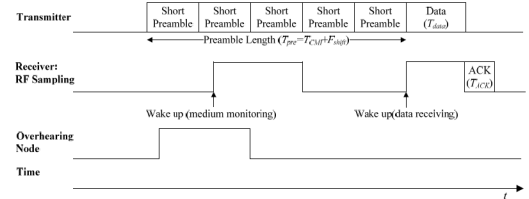


그림 4. 제안하는 MAC 프로토콜의 동작예

Source Address, CMI, Destination Address, Preamble Countdown과 Data Length 필드로 구성 된다. 이러한 구조를 가지고 있는 짧은 프리엠블은 오버히어링 문제점과 데이터의 오버히어링 문제점을 두 가지 방식으로 동시에 해결하고 있다. 수신노드 는 자신에게 보내지는 데이터인지 아닌지를 Destination Address로 판단할 수 있으며, 오버히어 링 노드들은 Countdown과 Data Length 값을 이용 하여 전송이 끝나는 시점을 정확히 알 수 있으므로 오버히어링 문제점을 해결할 수 있다. 그림 4는 제 안하는 MAC 프로토콜의 동작 예를 보여주고 있다.

#### 2.2 문제 수식화

본 논문에서는 CMI를 동적으로 변화시켜 지연시 간을 고려할 뿐 아니라 에너지 소비를 최소화하기 위한 새로운 MAC 프로토콜을 제안한다. 주어진 문 제는 다음과 같이 정의한다.

$$\begin{aligned} & \sum_{i=1}^N Energy_i, \\ \text{subject to } & \sum_{i=1}^{N_j^{hop}} Delay_i^j < Delay_{max} \text{ for all } j \quad (1) \end{aligned}$$

여기서  $N$ 은 센서 노드의 개수이며,  $N_j^{hop}$ 은  $j$ 번째 경로에서의 전체 홉 수를 의미한다.  $Energy_i$ (mJ)는  $i$  번째 노드에서의 초당 에너지 소비를 나타내며,  $Delay_i^j$ (second)는  $j$ 번째 경로에서  $i$ 번째 링크의 지 연시간을 나타내며,  $Delay_{max}$ 는 최대 허용가능한 지

연시간이다. 하지만  $j$ 번째 경로에서의 중단간 지연시간을 계산하기 위해서는 경로상의 모든 센서노드에서의 지연시간을 예측해야 하므로 오버헤드가 매우 큰 문제점을 가지고 있다. 그러므로 위의 문제점의를 다음과 같이 단순화 한다.

$$\alpha \cdot Energy_i + (1 - \alpha) \cdot Delay_i \quad (2)$$

여기서  $Delay_i$ 는  $i$ 번째 노드에서의 지연시간을 의미한다. 그리고 본 논문에서 이 수식을 비용함수로 정의한다.

### 2.3 에너지 소비와 지연시간 예측

에너지 소비와 지연시간을 예측하기 위하여 데이터가 수집 전달되는 과정은 그림 5와 같이 센서 노드는 자식노드의 센싱 데이터를 데이터 병합 없이 부모 노드로 전달한다고 가정한다.

에너지 소비는 receiving, transmitting, listening과 sleep 상태에서 발생한다. 그러므로 1초당 소비되는 에너지는 다음과 같은 수식으로 표현할 수 있다.

$$E = E_{short} + E_{rx} + E_{tx} + E_{listen} + E_{sleep} \quad (3)$$

여기서  $E_{short}$ ,  $E_{rx}$ ,  $E_{tx}$ ,  $E_{listen}$  과  $E_{sleep}$  은 각각 짧은 프리엠블을 수신할 때, 패킷을 수신할 때, 프리엠블과 패킷을 송신할 때, 채널 상태를 확인할 때와 sleep상태에서 소비되는 에너지를 나타낸다. 짧은 프리엠블을 수신할 때 소비되는 에너지를 계산하기 위하여 평균적으로 짧은 프리엠블을 수신하는데 걸리는 시간을 예측하면 다음과 같다.

$$\overline{t_{short}} = \sum_{i=1}^n r_i \cdot (T_{short}/2 + T_{short}) \quad (4)$$

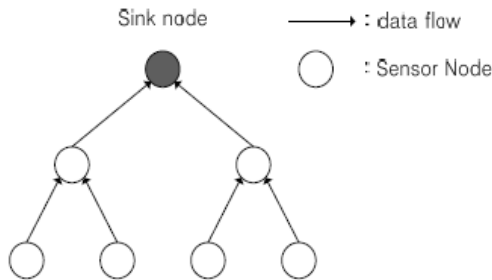


그림 5. 데이터 경로 예

여기서  $n$ 과  $r_i$ 는 이웃노드의 수와  $i$ 번째 노드에서의 초당 센싱 횟수를 의미한다. 그리고  $T_{short}$ 은 짧은 프리엠블의 길이를 나타낸다. 그러므로 짧은 프리엠블을 수신하기 위해 소비되는 에너지는 다음과 같이 예측할 수 있다.

$$\overline{E_{short}} = \overline{t_{short}} \cdot I_{rx} \cdot V \quad (5)$$

여기서  $I_{rx}$ 는 패킷 또는 프리엠블을 수신하는 동안의 전류량을 나타내며,  $V$ 는 전전지의 전압을 의미한다. 그리고 데이터 패킷을 수신하는데 필요로 되는 초당 시간량은 다음과 같이 예측할 수 있다.

$$\overline{t_{rx}} = \sum_{i=1}^m r_i \cdot T_{packet} \quad (6)$$

여기서  $m$ 은 자신에게 데이터를 전송하는 노드(자식노드)의 수를 나타내며,  $T_{packet}$ 은 패킷의 길이를 의미한다. 그러므로 평균 에너지 소비량은 다음과 같이 계산할 수 있다.

$$\overline{E_{rx}} = \overline{t_{rx}} \cdot I_{rx} \cdot V \quad (7)$$

센서 노드는 자신의 센싱 데이터와 자식 노드들의 센싱 데이터를 모두 전송해야 하므로 패킷을 전송하는 초당 시간량은 다음과 같이 계산 할 수 있다.

$$t_{tx} = \sum_{i=0}^m r_i \cdot (T_{pre} + T_{packet}) \quad (8)$$

여기서  $r_0$ 는 자기 자신의 초당 센싱 횟수를 나타내고  $T_{pre}$ 는 전체 프리엠블의 길이를 의미한다. 그러므로 송신시에 소비되는 에너지는 다음과 같이 계산할 수 있다.

$$E_{tx} = t_{tx} \cdot I_{tx} \cdot V \quad (9)$$

각 센서 노드들은 CMI 마다 깨어나서 채널 상태를 체크하게 된다. 그러므로 초당 listen 상태로 소비되는 시간은 다음과 같이 나타낼 수 있다.

$$t_{listen} = (T_{rinit} + T_{ron} + T_{rx/tx} + T_{sr}) / T_{CMI} \quad (10)$$

여기서  $T_{rinit}$ ,  $T_{ron}$ ,  $T_{rx/tx}$  와  $T_{sr}$ 은 각각 송수신기를 초기화하는데 걸리는 시간, 송수신기를 키는데 걸리는 시간, RX또는 TX로 전환하는데 걸리는 시간과 채널을 체크하는데 걸리는 시간을 의미한다. 그리고 소비되는 에너지( $E_{monitoring}$ )은 약  $17.3\mu J^{[10]}$  이다. 그러므로 초당 listen 상태에서 소비되는 에너지는 다음과 같이 계산할 수 있다.

$$E_{listen} = E_{monitoring} / T_{CMI} \quad (11)$$

마지막으로 노드가 초당 sleep상태를 유지하는 시간은 다음과 같이 예측할 수 있다.

$$\overline{t_{sleep}} = 1 - \overline{t_{over}} - \overline{t_{rx}} - \overline{t_{tx}} - \overline{t_{listen}} \quad (12)$$

그리고 소비되는 에너지는 다음과 같이 예측할 수 있다.

$$\overline{E_{sleep}} = \overline{t_{sleep}} \cdot I_{sleep} \cdot V \quad (13)$$

그러므로 센서 노드에서 초당 소비되는 에너지는 다음과 같이 예측할 수 있다.

$$\overline{E} = \overline{E_{short}} + \overline{E_{rx}} + \overline{E_{tx}} + \overline{E_{listen}} + \overline{E_{sleep}} \quad (14)$$

그리고 각 에너지 소비를 계산할 때  $r_i$ 는 테이블 1에서와 같이 저장된 시간 테이블을 보고 유추할 수 있다. 앞에서 설명하였듯이 종단간 지연시간을 예측하기 위해서는 경로상의 모든 노드에서의 지연시간을 알아야만 하므로 오버헤드가 매우 큰 문제점을 가지고 있다. 그러므로 본 논문에서는 한 노드에서의 지연시간을 예측하여 종단간 지연시간을 최소화 하고자 한다. 먼저 노드에서의 버퍼량이 무한대라고 가정하였을 때

표 1. 에너지 소비와 지연시간을 예측하기 위해 센서 노드에서 저장된 시간테이블 예

Overhearing time (s)	Receiving time (s)	Sensing time (s)
0.850	0.068	0.247
1.350	0.947	0.679
1.850	1.587	1.021
2.350	1.934	1.723
2.850	2.523	2.621

충돌(collision)이 발생할 확률은 다음과 같이 나타낼 수 있다.

$$p = 1 - \prod_{i=1}^n (1 - \tau_i) \quad (15)$$

여기서  $\tau_i$ 는  $i$ 번째 노드가  $t$ 시간에 패킷을 전송할 확률이며 다음과 같이 예측 할 수 있다.

$$\tau_i = \begin{cases} \sum_{j=0}^m r_j \cdot (T_{pre} + T_{data}) & \text{if } \sum_{j=0}^m r_j < \frac{1}{T_{pre} + T_{data}} \\ 1 & \text{otherwise} \end{cases} \quad (16)$$

그리고 센서 노드는 만약 충돌이 발생하면 CMI 만큼 backoff를 수행하므로 평균 backoff 주기는 다음과 같이 계산할 수 있다.

$$\sum_{i=1}^{\infty} p^i \cdot T_{CMI} \quad (17)$$

그러므로 노드에서의 지연시간은 다음과 같이 예측 할 수 있다.

$$\sum_{i=1}^{\infty} p^i \cdot T_{CMI} + T_{pre} + T_{packet} = \frac{\left(1 - \prod_{i=1}^n (1 - \tau_i)\right) \cdot T_{CMI}}{\prod_{i=1}^n (1 - \tau_i)} + T_{pre} + T_{packet} \quad (18)$$

실제 구현상에서 collision이 발생할 확률은 테이블 I에서와 같이 오버헤더링이 발생한 시간과 수신한 시간을 이용하여 간단히 예측할 수 있다.

$$p = 1 - \prod_{i=1}^n (1 - \tau_i) = \begin{cases} \frac{k(T_{pre} + T_{packet})}{t_i - t_1} & \text{if } t_i - t_1 > k(T_{pre} + T_{packet}) \\ 1 & \text{otherwise} \end{cases} \quad (19)$$

### 2.3 채널 모니터링 주기(CMI) 결정 알고리즘

앞에서 설명하였듯이 에너지 소비와 지연시간은 CMI와 센싱 주기에 의해 결정된다. 만약 센싱 주기가 고정되어 있다면 비용함수를 최소화하기 위한 최적의 CMI를 수식적으로 쉽게 찾을 수 있다. 하지만 만약 센싱 주기가 고정되어 있지 않다면 최적의 CMI를 찾

채널 모니터링 주기 결정 알고리즘

Step 1. 예상되는 지연시간과 에너지 소비를 찾아낸다.  
 Step 2. 비용함수를 최소화 하는 CMI를 찾아내고  $T_{Next\_CMI}$  값을 갱신한다.  
 Step 3. 만약 찾아낸 CMI가 현재 CMI보다 크다면, 노드는 자식노드에게 CMI를 알리기 위하여 한번 최대 길이 ( $T_{max}$ )의 프리엠블을 이용하여 데이터를 전송한다. 반대의 경우에는 찾아낸 CMI를 이용하여 바로 데이터를 전송한다.

기는 매우 어렵다. 본 논문에서는 비용 함수를 최소화 하기 위한 CMI를 결정 알고리즘을 제시한다. 그림 6는 제안하고 있는 센서 MAC 프로토콜의 전체 순서도를 보여주고 있다. 노드들은 주기적으로 깨어나서 채널 상태를 체크하게 된다. 만약 채널이 busy하다면 센서 노드는 짧은 프리엠블을 수신한다. 그리고 송신 노드가 자신의 부모 노드인 경우에는 송신노드의 부모노드의 CMI를 저장한다(저장된 부모노드의 CMI를 이용하여 부모노드로 데이터를 전송한다). 짧은 프리엠블의 목적지 주소가 자신이 아니면, 노드는 데이터 전송이 끝날때까지 sleep 상태를 유지한다. 반대로, 자

신이 목적지 주소인 경우에는 data전송이 시작될 때까지 sleep상태를 유지하고 다시 깨어나서 데이터를 수신하게 된다. 수신노드는 데이터를 성공적으로 받으면, ACK 메시지를 송신 노드에게 보내게 된다. 만약 채널이 idle하고 보낼 데이터가 있으면 센서 노드는 자신의 다음 CMI( $T_{nextCMI}$ )를 찾아내고, 성공적으로 송신후에 자신의 기존 CMI를 업데이트한다.

#### IV. 실험결과

본 실험에서 성능비교를 위해 다음과 같은 가정을 한다. (1) 각 센서 노드에서 데이터 병합(Data Aggregation)은 사용하지 않는다. (2) 센싱 데이터는 Unicast 방식으로 sink 노드에 전송된다. (3) Routing Protocol은 Tree 방식(Minimum Spanning Tree)을 사용한다. (4) 버퍼 크기는 무한대라고 가정한다. (5) 채널상의 에러 또는 packet drop이 없다고 가정한다. Mica2와 CC1000 송수신기를 사용하였을 때 소비되는 시간과 에너지는 표 2와 같고 실험환경은 표 3과 같으며, 실험에서 사용된 센서 네트워크의 토폴로지는 그림 7과 같다. 그리고 가중치( $\alpha$ )는 사용목적에 따라 사용자가 결정하는 파라미터이며, 본 실험에서는 0~0.5 사이의 값을 이용하였으며, 최대 프리엠블의 길이는 512byte로 설정하였고 초기 프리엠블의 길이는 128byte로 설정하였

표 2. The Mica2 Mote와 CC1000 Transceiver를 사용했을 때 소비 시간과 에너지

Operation	Time (s)	I(mA)
Initialize radio	350E-6	6
Turn on radio (c)	1.5E-3	1
Switch to RX/TX	250E-6	15
Time to sample radio	350E-6	15
Evaluate radio sample	100E-6	6
Receive 1 byte	416E-6	15
Transmit 1 byte	416E-6	20

표 3. 실험 환경

Parameter	Default
Sleep Current (mA)	0.03
Capacity of Battery (mAh)	2500
Voltage	3
Short Preamble Length (bytes)	32
Data Length (bytes)	100

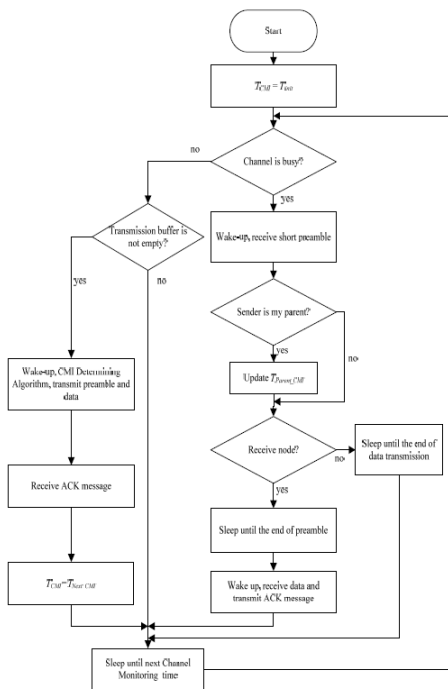


Fig. 8. Flow charts of the proposed sensor MAC protocol.

그림 6. 제안하는 MAC 프로토콜

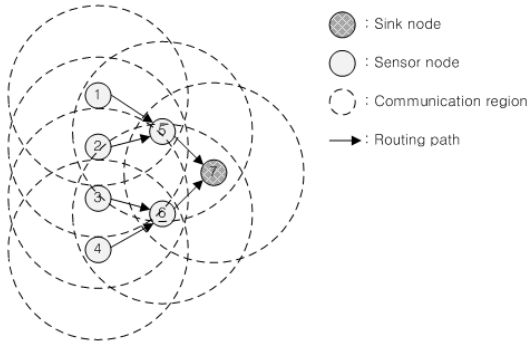


그림 7. 센서 네트워크 토폴로지

다. 실험에서 각 노드는 임의의 센싱 주기로 데이터를 센싱한다. (300초까지 평균 센싱 주기가 42.6초이며, 300~600초 사이는 21.3초이고 600~900초 사이는 106.5초로 설정하였다.) 그림 8은 제안한 MAC 프로토콜에서 가중치 값에 따른 중단간 지연 시간을 보여주고 있다. 가중치 값이 증가함에 따라 에너지 소비는 상대적으로 줄어들고 지연시간은 증가하게 된다. 그림 9는 센싱주기의 변화에 따라서 노드 5와 노드 6의 CMI가 어떻게 변화하는 지를 보여주고 있다. 그림 10은 BMAC, BMAC+와 제안한 MAC 프로토콜의 중단간 지연시간을 보여주고 있다. BMAC과 BMAC+는 고정된 CMI를 사용하므로 collision이 발생(약 230초)하지 않는 한 일정한 값을 가지고 있는 것에 반해, 제안한 MAC 프로토콜은 트래픽의 변화에 따라 지연시간이 변화하는 것을 확인할 수 있다. 그림 11은 제안한 MAC 프로토콜이 효과적으로 오버헤딩을 해결하여 에너지 낭비를 줄이고, 또한 에너지 소비가 적은 CMI를 결

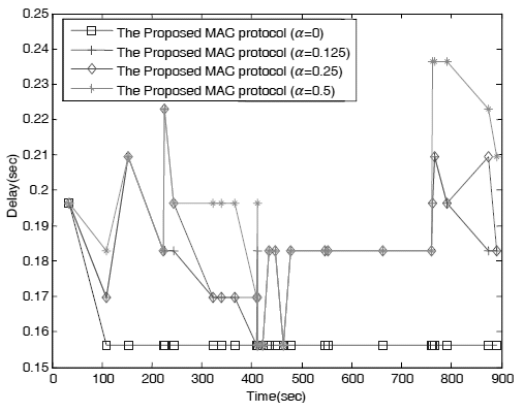


그림 8.  $\alpha$  값 변화에 따른 중단간 지연시간

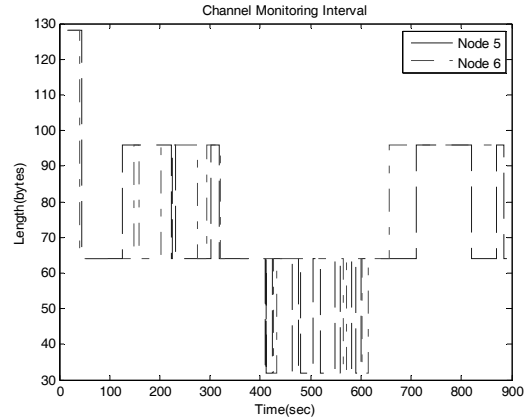


그림 9. 노드5와 6에서 시간에 따른 CMI 변화

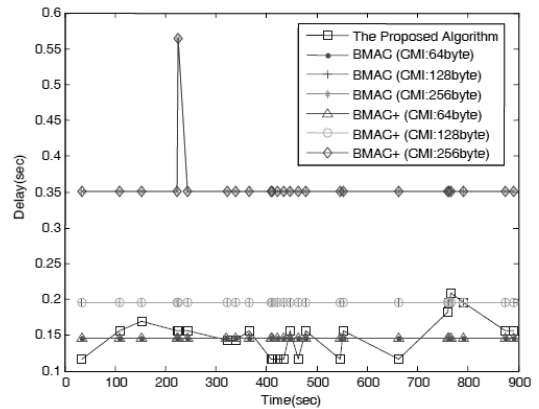


그림 10. BMAC, BMAC+와 제안한 MAC 프로토콜의 중단간 지연시간 비교( $\alpha=0.25$ )

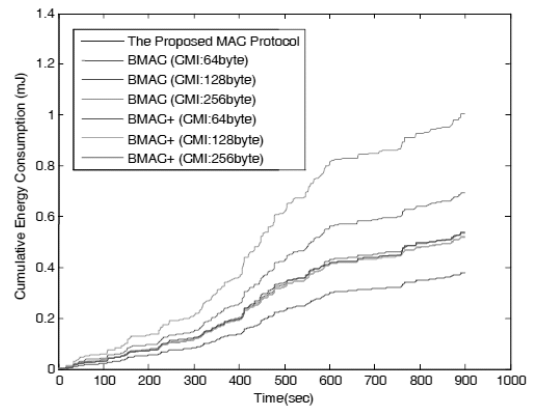


그림 11. BMAC, BMAC+와 제안한 MAC 프로토콜의 에너지 소비 비교( $\alpha=0.25$ )

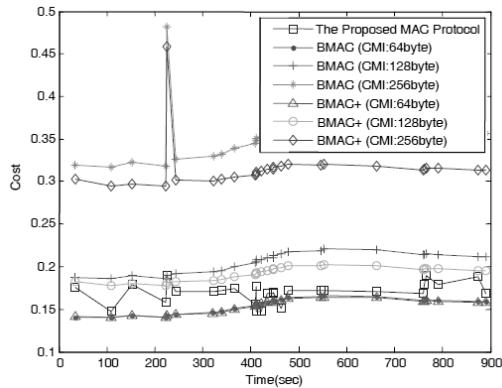


그림 12. BMAC, BMAC+와 제안한 MAC 프로토콜의 비용 비교( $\alpha=0.25$ )

정함으로써 다른 MAC 프로토콜보다 에너지 소비가 적은 것을 볼 수 있다. 그림 12는 BMAC, BMAC+와 제안한 MAC 프로토콜의 비용을 비교한 것이다. 제안한 MAC 프로토콜이 시간이 지남에 따라 최적의 CMI를 가지는 낮은 비용으로 수렴하는 것을 확인할 수 있다. 780초 부근에서 비용이 다른 MAC 프로토콜보다 높은 이유는 급격한 트래픽 변화에 따라 예측 오차로 잘못된 CMI를 선택하였기 때문이다. 표 4는 900초에서 각 센서 노드에서 소비된 에너지를 비교한 것이다. 본 논문에서 제안한 MAC 프로토콜이 다른 MAC 프로토콜에 비교하여 적은양의 에너지를 소비하는 것을 알 수 있다.

### V. 결론

본 논문에서는 지연시간에 민감한 데이터를 무선 센서 네트워크상에서 전달하기 위한 에너지 효율적인 MAC 프로토콜을 제안하였다. CMI 정보를 이웃 노드에게 전달하기 위하여 짧은 프리앰블 구조에

표 4. 900초에서 각 노드와 전체 노드에서 소비된 에너지 (mJ) 비교

	BMAC (128byte)	BMAC+ (128byte)	Proposed MAC ( $\alpha = 0.25$ )
Node 1	199.379796	135.321475	118.333382
Node 2	275.036021	206.947963	131.392057
Node 3	197.380973	136.013835	105.552696
Node 4	153.138794	125.280607	93.180369
Node 5	325.375429	294.206179	264.072996
Node 6	236.438428	230.858570	191.968609
Sum	1386.749441	1128.628629	904.500109

대해 설명하였으며, CMI 결정 알고리즘을 이용한 새로운 센서 MAC 프로토콜을 제안하였다. 그리고 제안한 MAC 프로토콜은 짧은 프리앰블을 이용하여 오버헤드가 거의 없으며 분산적으로 동작한다. 그리고 실험결과에서는 다른 MAC 프로토콜에 비교하여 에너지 소비와 지연시간에 효율적임을 보여 주고 있다.

### 참고 문헌

- [1] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. Proc. the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies 2002, vol.3, pp.1567-1576.
- [2] T. Zheng, S. Radhakrishnan, and V. Sarangan. PMAC: An adaptive energy-efficient MAC protocol for Wireless Sensor Networks. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium 2005, pp.65-72,
- [3] T. v. Dam and K. Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. SenSys 2003, pp.171-180,
- [4] C. Schurgers, V. Tsiatsis, and M. Srivastava. STEM: Topology Management for Energy Efficient Sensor Networks. IEEE Aerospace Conference 2002, vol.3, pp.1099-1108.
- [5] I. Rhee, A. Warrier, M. Aia, and J. Min. Z-MAC: a hybrid MAC for wireless sensor networks. The 3rd ACM conference on Embedded Networked Sensor Systems (SenSys) 2005, pp.90-101.
- [6] IEEE Std 802.15.4-2006, Part 15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).
- [7] J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems 2004, pp.95-107.
- [8] M. Avvenuti, P. Corsini, P. Masci, and A. Vecchio. Increasing the efficiency of preamble



sampling protocols for wireless sensor networks. Mobile Computing and Wireless Communications international Conference 2006, pp.117-122

- [9] M. Buettner, G. Yee, E. Anderson, and R. Han. X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Sensor Networks. Technical Report CU-CS-1008-06, May., 2006
- [10] K. Paek, J. Kim, U. Song, and C. Hwang. Priority-based medium access control protocol for providing QoS in wireless sensor networks. IEICE Transaction Letters on Information System 2007, vol.E90-D(9), pp.1448-1451.

송 황 준 (HwangJun Song)

정회원



1990년 2월 서울대학교 제어계  
측 공학과(학사)  
1992년 2월 서울대학교 제어계  
측 공학과(석사)  
1999년 5월 Univ. of Southern  
California, EE-Systems(박사)  
2000년 3월~2000년 8월 세종

대학교(전임강사)

2000년 9월~2005년 2월 홍익대학교(조교수)

2005년 2월~현재 포항공과대학교(부교수)

<관심분야> 영상압축, 멀티캐스트, Ad-Hoc

오 형 래 (HyungRai Oh)

준회원



2003년 2월 홍익대학교 전파공  
학과(학사)

2005년 2월 홍익대학교 전파공  
학과(석사)

2005년 8월 포항공과대학교 컴  
퓨터공학과(연구원)

2005년 9월~현재 포항공과대학

교 컴퓨터공학과(박사과정)

<관심분야> Caching, 센서네트워크, P2P 네트워크