



## 저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#) 

2011년 8월  
석사학위논문

무선 센서 네트워크에서 블록응답을  
이용한 에너지 효과적인 맥 프로토콜

조선대학교 대학원

정보통신공학과

박용태

2015년 8월

석사학위논문

무선 센서 네트워크에서 블록응답을 이용한 에너지 효율적인 맥 프로콜

박용태

무선 센서 네트워크에서 블록응답을 이용한 에너지  
효과적인 맥 프로토콜

An Energy Efficient MAC Protocol using  
Block ACK for the Wireless Sensor Network

2011년 8월 00일

조 선 대 학 교 대 학 원

정보통신공학과

박 용 태

무선 센서 네트워크에서 블록응답을 이용한 에너지  
효과적인 맥 프로토콜

지도교수 변 재 영

이 논문을 공학석사학위신청 논문으로 제출함.

2011년 8월 00일

조 선 대 학 교 대 학 원

정보통신공학과

박 용 태

# 박용태의 석사학위 논문을 인준함

위원장 조선대학교 교수 박 종 안 印

위 원 조선대학교 교수 한 승 조 印

위 원 조선대학교 교수 변 재 영 印

2011년 8월 00일

조 선 대 학 교 대 학 원

# 목 차

그림 목 차 .....	ii
표 목 차 .....	iii
ABSTRACT .....	iv
I. 서 론 .....	1
A. 연구의 배경 및 목적 .....	1
B. 연구의 목표 .....	2
C. 연구의 목표 .....	2
II. 무선 센서 네트워크와 블록응답 기법 .....	3
A. 무선 센서 네트워크 .....	3
B. 802.11e의 블록 응답 기법 .....	13
III. 제안하는 프로토콜의 설계 .....	16
A. 기존 조각화 전송의 문제점 .....	16
B. 제안하는 프로토콜의 조각화 전송 .....	17
C. 프로토콜의 패킷 구조 .....	18
D. Duration .....	19
E. 손상된 조각의 재전송 .....	20
IV. 실험 및 고찰 .....	22
A. 시뮬레이션 설정 .....	22
B. 시뮬레이션 및 결과 고찰 .....	27
V. 결론 및 향후 연구 과제 .....	34
참고문헌 .....	35

# 그림 목 차

그림1. 무선 센서 네트워크의 동작 개념 .....	4
그림2. 센서 노드의 구성 .....	5
그림3. S-MAC의 구간과 프레임 포맷 .....	8
그림4. S-MAC의 listen/sleep 주기 및 동작 상태 .....	9
그림5. 데이터 전송 시의 TEEM의 동작 .....	10
그림6. 데이터 전송이 없을 시의 TEEM의 동작 .....	10
그림7. INS 프로토콜의 기본 동작 .....	12
그림8. 무선 센서 네트워크의 MAC 프로토콜 분류 .....	12
그림9. Block ACK 동작 과정 .....	13
그림10. Block ACK 즉시 전송 모드 .....	14
그림11. Block ACK 지연 전송 모드 .....	15
그림12. 전형적인 조각화 전송 방법 .....	16
그림13. 제안하는 프로토콜의 기본 동작 .....	17
그림14. 제안하는 프로토콜의 패킷 구조 .....	17
그림15. 손실 된 데이터에 대한 재전송 방법 .....	19
그림16. 손실 된 ACK에 대한 재전송 방법 .....	21
그림17. S-MAC 프로토콜과 제안하는 프로토콜과의 ACK 전송량 비교 .....	27
그림18. TEEM 프로토콜과 제안하는 프로토콜과의 ACK 전송량 비교 .....	28
그림19. INS 프로토콜과 제안하는 프로토콜과의 ACK 전송량 비교 .....	28
그림20. 조각 수의 증가에 따른 전송 지연 시간. 에러율 0% .....	29
그림21. 조각 수의 증가에 따른 전송 지연 시간. 에러율 5% .....	30
그림22. 송신 노드의 에너지 소비량 .....	31
그림23. 중간 노드의 에너지 소비량 .....	32
그림24. 목적지 노드의 에너지 소비량 .....	33
그림25. 세 노드의 에너지 소비량 평균 값 비교 .....	33



## 표 목 차

표 1. 상용화 센서 노드 하드웨어의 주요 규격 .....	6
표 2. 새롭게 추가된 에너지 모델 파라미터 .....	23
표 3. 시뮬레이션을 위한 tcl 파라미터 .....	26
표 4. 프로토콜 동작 관련 파라미터 .....	26
표 5. 에러 없는 환경에서의 이상적인 ACK 전송량 .....	27

# ABSTRACT

## An Energy Efficient MAC Protocol using Block ACK for the Wireless Sensor Network

Park, Yong-Tae

Advisor : Prof. Pyun, Jae-Young, Ph.D.

Department of Information Communication Engineering

Graduate School of Chosun University

The reduction of power consumption and network's life expansion are crucial for a wireless sensor network (WSN). This is typically performed by introducing low duty cycle in a sensor node. A special challenge in WSNs is the transmission of a large amount of data, like images and video, that is becoming more and more required in various applications. A wireless network is very much prone to communication errors. Thus, for the efficient design, large data messages are broken down into smaller fragments and those smaller fragments are transmitted. But this approach introduces the burden of exchanging redundant control packets increasing the energy consumption and the transmission delay. In this paper, I propose a data fragmentation scheme using a block acknowledgment (BA) mechanism to minimize the number of the control packets and delayed transmission caused by fragmentation. I implemented the proposed scheme in typical WSN protocols and compared it with the original protocols through ns-2 simulation. The simulation results verify that our scheme can decrease energy consumption as well as end to end delay.

# I. 서 론

## A. 연구의 배경 및 목적

일반적인 무선 센서 네트워크는 낮은 대역폭에서 센서 디바이스로부터 감지한 데이터를 전송하여 수집한다. 최근에는 무선 네트워킹 기술과 저 전력 고성능 프로세서의 개발, 저렴해진 하드웨어로 인하여 카메라와 마이크 등을 장착하여 정지영상이나 음성, 비디오 영상의 멀티미디어 데이터의 전송 까지도 가능하게 되었고 그 응용 분야 또한 재난 방재, 환경 감시, 물류관리, 실시간 보안, 헬스 케어 등의 분야로 넓어지고 있다.

센서 노드는 배터리 전력을 사용하여 동작 한다. 이러한 방식은 배터리의 교환이나 재충전이 어렵다. 그렇기 때문에 센서 노드의 동작은 에너지 효율적이어야 한다. 무선 센서 네트워크의 노드들은 노드들 간에 스스로 네트워크를 구성하고 주변 노드들과 협업하여 데이터를 교환한다. 따라서 노드 간 통신에서 가장 많은 에너지를 소비하게 된다[1]. 그러므로 무선 센서 네트워크의 MAC 프로토콜은 노드간의 통신이 필요할 때만 무선 통신 모듈을 켜으로써 에너지를 절약하도록 설계되었다. 무선 센서 네트워크에서 에너지 절약을 위한 많은 노력이 있지만 대부분은 듀티 사이클을 수정하는 형태로 한정되어왔다.

최근의 무선 센서 네트워크의 응용은 멀티미디어 데이터의 전송을 요구한다. 그러나 현재 제안된 무선 센서 네트워크의 MAC 프로토콜들은 멀티미디어 데이터와 같은 대용량의 데이터 전송에 적용하기에 적합하지 못하다. 유선의 통신 환경과는 다르게, 무선 채널 환경은 에러가 빈번하게 발생한다. 긴 데이터 메시지는 짧은 메시지보다 상대적으로 손실되기 쉽다. 따라서 멀티미디어 데이터들은 본질적으로 부피가 크기 때문에 네트워크에서 빈번한 재전송을 요구할 것이 분명하다. 단 몇 비트의 에러로 인하여 전체 데이터를 처음부터 재전송을 해야 한다면, 노드가 떠안아야 할 에너지 소모량은 엄청날 것이며, 이를 위해 사용되는 긴 시간의 불공정한 채널의 점유는 다른 노드의 전송을 방해하여 네트워크의 역할을 지연 시키고, 심지어 네트워크 전체 수명의 감소를 초래할 수 있다. 이러한 상황에서 큰 데이터 패킷을 전송하는 것은 비효율적이다. 따라서 긴 데이터의 전송에 적합한 MAC 프로토콜의 개발이 필요하다.

## B. 연구 목표

본 논문에서는 기존의 센서 네트워크의 환경에서는 물론이고 멀티미디어 데이터의 수집과 전송을 요구하는 환경에서의 사용가능한 MAC 프로토콜을 제안한다.

재전송시에 발생하게 될 에너지 소비와 과도한 채널의 점유를 줄이기 위해서 센서노드들은 큰 데이터 패킷을 여러 개의 더 작은 조각으로 나누어 전송하는 조각화 전송 기법을 적용하였다. 또한 조각화 전송 기법을 사용 할 때 발생하는 제어 패킷의 오버헤드를 줄이기 위해서 Block ACK 기법을 적용하였다. 조각화 전송과 Block ACK 기법을 사용함으로써 노드의 에너지 효율성을 높이고 전송 지연 시간 감소 또한 기대할 수 있다. 또한 기존의 센서 네트워크 MAC 프로토콜에 적용할 수 있어 구현의 편리성을 기대할 수 있다.

## C. 논문의 구성

본 논문의 구성은 다음과 같다. 2장 관련 연구에서는 이론적 배경으로써 무선 센서 네트워크와 통신상에서 발생하는 에너지 낭비 요인에 대해 간략히 언급한다. 또한 관련 연구로서 무선 센서 네트워크 MAC 프로토콜의 종류와 IEEE802.11e에서 제안한 Block ACK 기법에 대해 언급한다. 3장에서는 조각화 전송과 수정된 Block ACK 기법과 재전송 방법에 대하여 설명한다. 4장에서는 성능 평가로 시뮬레이션 모델 및 결과를 분석하고 마지막 장에서 요약 후, 향후 연구 방향을 제시하면 결론을 맺는다.

## II. 무선 센서 네트워크와 블록응답 기법

### A. 무선 센서 네트워크 (Wireless Sensor Networks)

#### 1. 무선 센서 네트워크 개념과 응용

정보의 전달하는 기술은 매우 중요하다. 실제로 컴퓨터, 네트워크를 기반으로 하는 정보 기술은 기업의 업무와 생산, 국방, 의료, 교육, 복지 등 인류의 생활에 많은 변화를 가져왔다. 특히 인터넷은 사람과 사람간의 연결해주는 매개체로 우리 삶의 일부분이 되었다. 최근에 대두되고 있는 유비쿼터스 컴퓨팅의 기술은 사람과 사물, 사물과 사물 간의 정보 교환이 가능하게 만들었다. 유비쿼터스 컴퓨팅 기술에 의해 새롭게 바뀐 사물들은 스마트해지고 인간은 이러한 사물로부터 원하는 지역의 환경 상황을 정확하게 인지할 수 있게 되었다. 즉, 유비쿼터스 사회가 도래한 것이다.

무선 센서 네트워크는 유비쿼터스 사회를 만들어 가는데 필요한 기술 중에 하나이다. 무선 센서 네트워크는 센서디바이스를 장착하고 광범위한 지역에 설치되어 있는 대규모의 노드들이 노드 주변 환경데이터를 수집하고 외부 네트워크의 사용자에게 전달하는 기술이다. 그림 1처럼 특정 목적을 위해 여러 지역에 분포된 센서들이 응용 계층에서 요구하는 어떤 현상의 정보를 수집한다. 그림처럼 어떤 이벤트가 발생한 경우에 노드들은 자신의 데이터 또는 이벤트 알람을 다른 노드들을 통해 싱크 노드로 전달한다.

싱크 노드는 외부 네트워크와 연결되어있는 센서 노드로써 직접적인 노드들의 정보를 외부 네트워크로 전달해주는 역할을 한다[2-5].

무선 센서 네트워크의 연구는 초기에 적군의 움직임을 감시하기 위한 군사적인 목적으로 시작되었다. 적과 아군의 이동을 감시하기 위해 땅위, 물속, 공중의 다양한 환경에서 통신 가능한 센서들을 배치하고, 이들 센서로부터 정보를 얻고자 하는 형태의 응용을 개발하고자 하였으나 당시의 기술로는 센서를 비롯한 주요 부품들의 소형화가 이루어지지 않고 크기에 비해 낮은 성능으로 센서를 배치하고 정보를 수집하기 위해서는 많은 자원의 소비 되었다.

이후 2000년대에 들어서 MEMS(micro electro mechanical systems) 기술과 IC 및 VLSI 기술을 바탕으로 저가의 저 전력 센서와 프로세서의 개발이 가능해지면서 무선

센서 네트워크 기술 개발이 가속화 되었다. MEMS 기술을 이용한 대표적인 구현 사례가 미국 UC 버클리 대학의 Smart Dust 프로젝트이다. 크기가 수 mm에 불과한 작은 센서에 온도·습도·조도 센서 및 통신 기능을 구현했다.

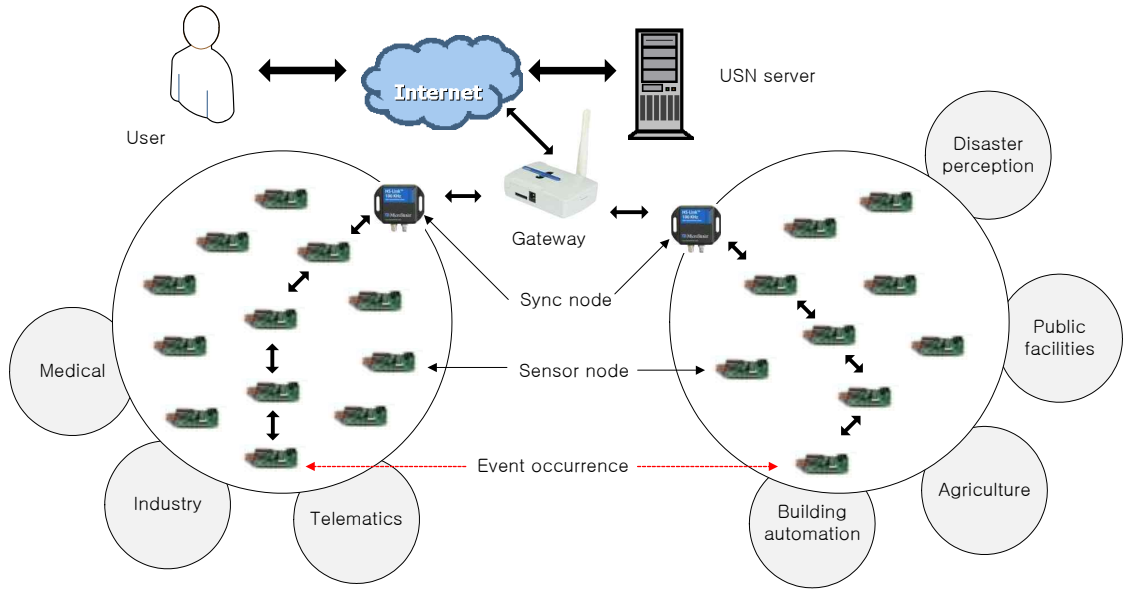


그림 1. 무선 센서 네트워크의 동작 개념.

무선 센서 네트워크가 적용될 수 있는 분야는 다양하다. 도시의 빌딩 곳곳에 설치된 센서들이 실시간으로 공기의 상태, 빛의 세기, 온도, 화재 감지 등을 하여 빌딩의 환경을 쾌적하고 안전하게 유지하게 해준다. 농장에 무선 센서 네트워크를 구축하면 온도나 습도 등 여러 환경 데이터를 감시하고 이에 따라 농작물에 물을 공급하거나 주변 온도를 제어하고 빛의 양을 조절하여 최상품의 농산물을 재배 할 수 있게 해준다. 댐이나 교량, 발전소, 하천 등의 구조물에 센서들을 적용하여 시설물의 이상 상태 등을 신속하게 파악하여 사고를 예방 할 수 있다. 이 밖에도 물속이나 산악지대 감시 초소 등에 적의 이동에 대한 감시를 센서 노드로 대체 할 수 있으며, 최근에는 텔레매틱스 (Telematics) 분야나 u-헬스케어(Ubiquitous Health Care) 분야를 위한 연구와 응용이 활발하게 진행되고 있다.

## 2. 무선 센서 네트워크의 노드의 구성

센서 노드의 하드웨어 구조는 노드의 사용 목적에 따라 달라질 수 있지만, 일반적으로 그림 3과 같이 센서모듈, 송수신장치, 전원공급 장치, 프로세서의 4가지 요소로 구성된다. 위치 정보가 반드시 필요한 응용분야에서는 GPS와 같은 위치측정 장치가 추가되며, 노드의 움직임이 요구되면 유동체를 포함한다[6, 7].

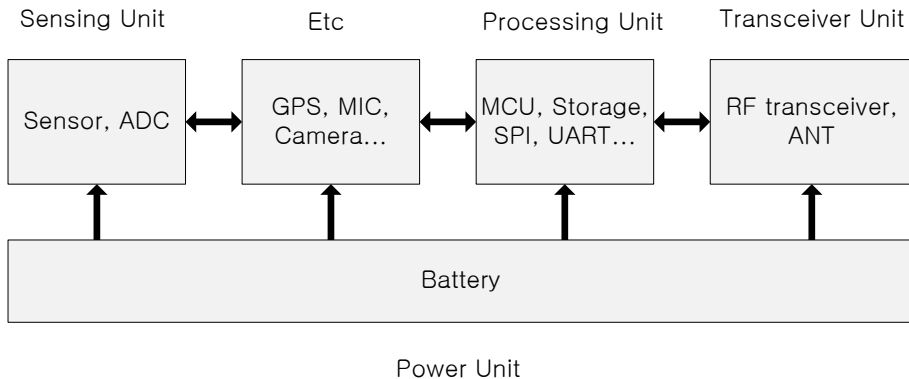


그림 2. 센서 노드의 구성

센서 모듈은 센서와 ADC(Analog to Digital Converter)로 구성된다. 센서는 온도, 습도, 조도, 적외선, 초음파, 가속도, 압력, 중력 센서 등이 사용 될 수 있으며 일 부 센서는 측정 한 값이 아날로그 신호로 나타나기 때문에 이를 디지털 신호로 변환시키기 위하여 ADC를 사용한다.

송수신장치는 라디오 전송 칩(RF transceiver chip)과 안테나로 구성된다. 라디오 전송 칩에서 센서 노드가 전송할 디지털 데이터를 라디오 전송 칩에서 무선 신호로 변환하여 안테나를 통해 전송한다.

전원공급 장치는 무선 센서 네트워크를 구성하는 노드가 대량으로 사용되고 사용한다는 비용적인 문제와 작은 크기 노드를 위해서 소형의 배터리를 사용한다. 응용에 따라서 재충전이 가능한 배터리나 외부 전력원을 연결하는 경우도 있다.

센서 노드의 프로세서는 노드에 부착된 센서의 데이터 처리 및 외부 인터페이스를 위한 SPI, UART, I2C, timer, power control 등으로 구성된다. 프로세서는 센서 노드에 부착된 센서의 종류 및 노드의 사용 목적에 따라 다양한 성능의 프로세서를 필요로 한다. 기존에 사용된 8bit MCU에서 센싱 된 단순 데이터를 인식 및 감지하는 것이

아닌, 데이터 용량이 큰 영상정보의 처리, 음성 신호 처리에 대한 분석을 가능하게 하기 위한 고성능의 16bit, 32bit 프로세서가 사용되기도 한다.

표1. 상용화 센서 노드 하드웨어의 주요 규격[8]

Node	Mica	Mica2	MicaZ	Tmote Sky	iMote	zigbeX
Clock (Mhz)	4	7.37	7.37	8	12	16
CPU	ATmega 103L	ATmega 128L	ATmega 128L	MSP430F	ARM7	ATmega 128L
Program memory (KB)	128	128	128	48	512	128
Data rate(kbps)	40	38.4	250	250	723.2	250
RF module	TR1000	CC1000	CC2420	CC2420	Zeevo TC2001	CC2420
Modulation	ASK	FSK	O-QPSK	O-QPSK	GFSK	O-QPSK
OS	TinyOS	TinyOS	TinyOS	TinyOS	TinyOS	TinyOS

### 3. 무선 센서 네트워크에서의 에너지 낭비 요인

무선 센서 네트워크의 MAC 프로토콜 설계의 가장 큰 목적은 노드들 간의 통신에 사용되는 소비전력을 줄여서 네트워크의 생존시간을 최대화 하는 것이다. 그래서 MAC 프로토콜의 연구는 이러한 목적을 달성하기 위해서 다음과 같은 4가지의 에너지 낭비 요인을 얼마나 효율적으로 최소화 시키는가에 있다.

- Collision
- Overhearing
- Control packet overhead
- Idle listening



첫 번째 에너지 낭비 요인은 충돌(collision)이다. 전송된 패킷이 hidden terminal 및 exposed terminal 문제로 인해 충돌이 발생하고 그 결과로 손상되어 재전송이 요구된다. 둘째, 다른 노드를 목적지로 하는 패킷을 엿듣게 되는 overhearing 문제. 셋째, 불필요한 제어 패킷의 전송에 따른 overhead. 마지막으로 현재 자신이 전송해야 할 데이터는 없지만, 언제 이웃노드로부터 데이터를 전송할지 모르기 때문에 항상 수신 상태를 유지하는 idle listening이 있다. 이러한 에너지 낭비 요인을 해결하기 위한 많은 MAC 프로토콜의 연구가 진행되어져 왔으며, 크게 두 가지 방식으로 나눌 수 있다. TDMA(Time Division Multiple Access) 기반의 프로토콜은 스케줄 기반의 프로토콜이라고도 불리며, 각 노드마다 타임 슬롯을 할당해주고, 자신의 정해진 시간에만 통신이 가능하도록 하는 방식이다. 이 방식의 프로토콜은 idle listening과 충돌 문제를 해결할 수 있다. 또한 RTS/CTS 패킷교환 방식[9]을 사용하지 않기 때문에 경쟁기반 프로토콜보다 제어 패킷의 오버헤드가 줄어드는 장점이 있다. 하지만 이 방식은 멀티 홉 환경에서는 데이터의 지연(delay)이 발생할 수도 있다. 또한 노드 간 동기화 및 정밀한 스케줄링을 요구해서 구현이 어렵다. 그리고 동적인 무선 센서 네트워크의 토폴로지 변화에 적응(scalability)하기 어렵다는 단점이 있다. 경쟁기반(contention based)의 프로토콜은 랜덤 액세스(Random Access) 방식으로 노드간의 전송 충돌과 매체 접근을 위한 채널 감지로 에너지 소모가 많은 방식이다. 하지만 RTS/CTS와 같은 제어패킷을 사용하여 통신 시에 충돌을 회피하고, idle listening 문제를 해결하기 위해 듀티 사이클을 사용하여 활성 구간 및 비활성 구간으로 나누어 활성구간에서만 스케줄 교환 및 RTS/CTS 패킷의 교환이 가능하도록 함으로써 에너지 소비를 줄이는 방법이다. TDMA 방식의 프로토콜에 비해서 데이터 수율(throughput)과 지연에서 이점을 가지며 상대적으로 구현이 쉽다.

## 4. 무선 센서 네트워크의 MAC 프로토콜

### a) S-MAC

S-MAC[10, 11]은 고정된 listen/sleep 구간을 가지는 대표적인 경쟁 기반 프로토콜이다. S-MAC이 경쟁 기반 방식의 최초의 프로토콜은 아니지만, 간단하면서도 효율적인 에너지 절감 성능을 보임으로써 상당수의 경쟁 기반 프로토콜은 S-MAC에서부터 파생되었다.

S-MAC은 listen과 sleep라는 두 구간으로 나뉜다. 이 구간은 주기적으로 반복되며, listen과 sleep 한 cycle을 프레임이라고 부른다. sleep 구간에서는 통신을 하지 않는 모든 노드들이 자신의 무선 모듈을 끄으로써 필요하지 않는 idle listen 상태에서의 에너지 낭비를 줄일 수 있다. listen 구간은 다시 두 개의 구간으로 나눌 수 있다. 첫 번째 구간은 SYNC구간으로 SYNC 패킷의 전송을 위해 설계되었다. SYNC 패킷은 브로드캐스트 방식의 패킷으로 주변 노드와의 동기화 문제를 해결하기 위해 사용된다. 두 번째 구간은 Data구간으로 data packet의 전송을 위해 설계되었다. 이 구간에서는 데이터 전송을 위해 사용 되는 RTS와 CTS 패킷이 사용된다. S-MAC의 프레임 포맷은 그림 3에 나와 있다.

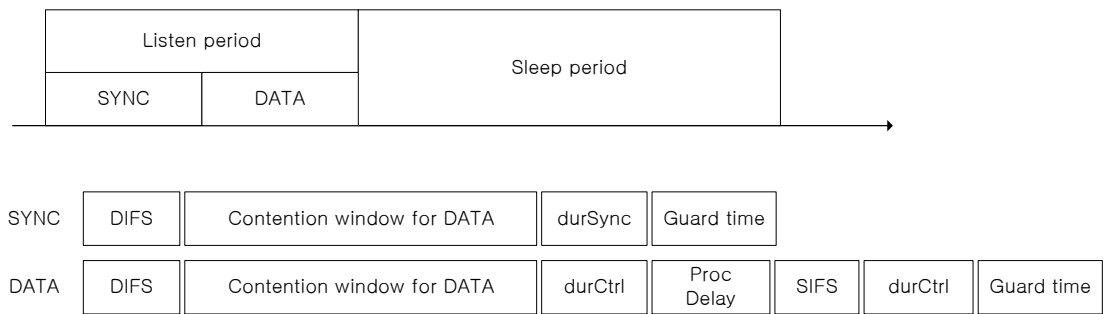


그림 3. S-MAC의 구간과 프레임 포맷.

S-MAC 프로토콜의 기본 동작은 그림 5와 같다. listen 구간에서는 모든 노드들이 깨어나 동기화를 위한 SYNC 와 데이터 전송에 필요한 제어 패킷인 RTS 와 CTS 패킷을 교환 한다. 기본적으로 SYNC 패킷은 10cycle 주기로 교환된다. 만약, 한 개 이상의 노드들이 다른 노드들과 통신을 하려고 한다면, listen 구간에서는 RTS와 CTS 교환을 통해 매체 접근에 대한 경쟁을 하게 된다. RTS와 CTS를 교환한 두 노드만이 채널에 접근하여 통신을 할 수 있으며, 나머지 노드들은 현재의 프레임이 끝나도 다음 프레임의 listen 구간이 시작 될 때 까지 sleep모드로 들어간다. 데이터 교환이 모두 끝난 두 노드는 곧바로 sleep 상태로 전환 하지 않고 다음 프레임 까지 Idle 상태에서 대기하게 된다.

S-MAC의 듀티 사이클은 listen 구간과 sleep 구간의 비율을 결정한다. 기본적으로 설정된 듀티 사이클은 10%로 고정되어있다. 노드가 깨어 있는 시간의 90% 시간을 sleep 상태로 있기 때문에 트래픽이 적은 상태에서는 에너지를 절약 할 수 있지만, 트래픽이 큰 경우에는 긴 sleep 시간으로 인하여 데이터 지연 문제가 발생하게 된다. 이런 문제점

은 듀티 사이클 방식을 채택하고 있는 모든 프로토콜에 해당된다.

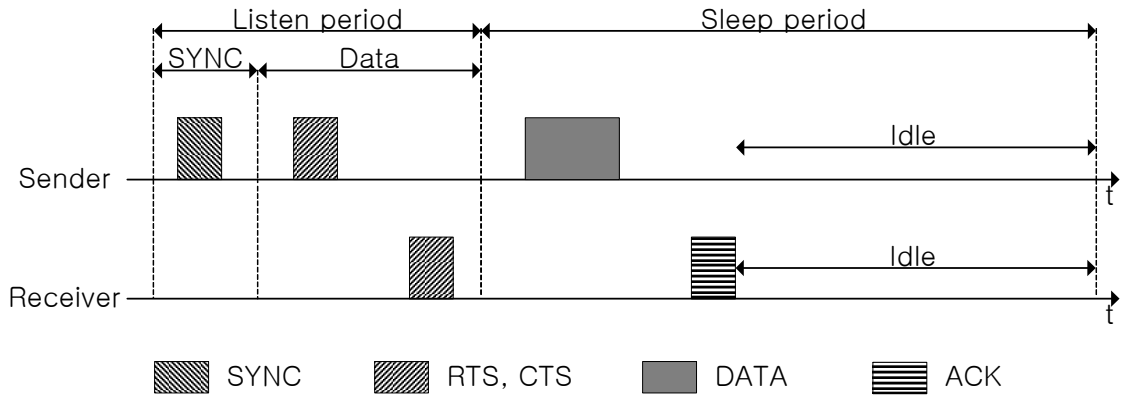


그림 4. S-MAC의 listen/sleep 주기 및 동작 상태.

### b) TEEM

TEEM [12]은 S-MAC으로부터 파생된 프로토콜이다. 이 프로토콜은 동기화를 위한 SYNC 패킷과 데이터 전송을 위한 RTS패킷을 결합한 SYNCrts라는 새로운 패킷 형태를 사용한다. Listen 구간은 두 구간으로 구성되어있으며, 첫 번째 구간인 SYNCdata 구간은 데이터 전송 시에 사용된다. 데이터 전송이 없을 경우에는 순수한 SYNC 패킷만 전송하는데, 두 번째 구간인 SYNCnodata 이뤄진다.

S-MAC 프로토콜이 자신의 전송이 끝나도 남아있는 프레임 시간까지는 idle 상태를 유지하는 것에 반해, TEEM은 자신의 전송이 모두 끝나면 프레임 종료 시간이 남아있더라도 sleep 상태로 전환 하면서 idle 상태에서의 에너지를 절감한다.

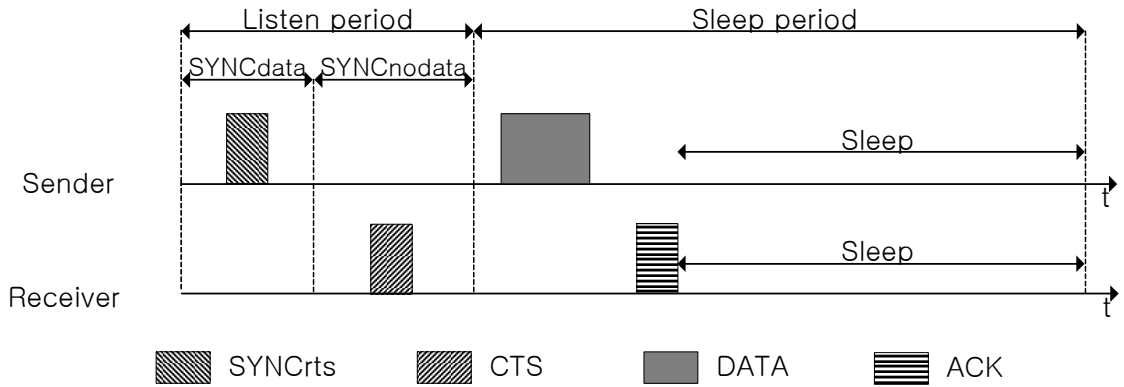


그림 5. 데이터 전송 시의 TEEM의 동작.

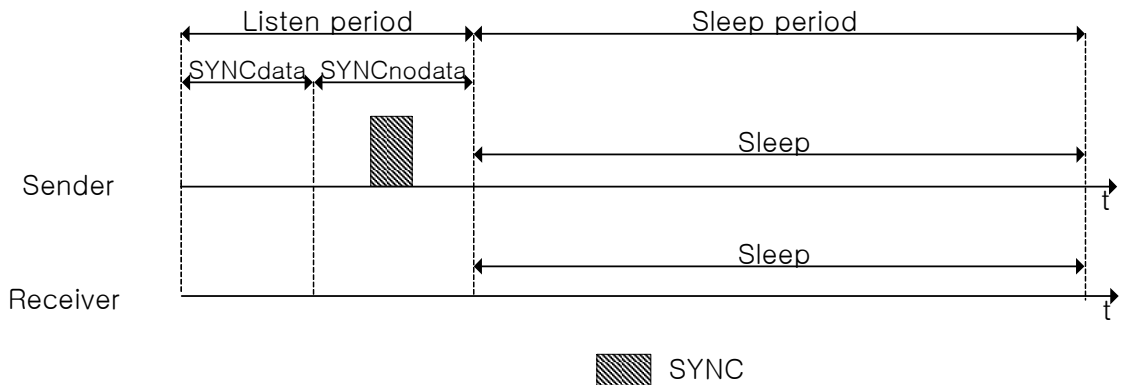


그림 6. 데이터 전송이 없을 시의 TEEM의 동작.

### c) WiseMAC

WiseMAC [13]은 각 노드의 idle listening에 의한 전력 소모를 감소시키기 위하여 이웃 노드가 송신 직전에 송신하는 프리앰블이 있는지 주기적으로 들어보는 프리앰블 샘플링 기법을 사용한다. 송신 노드는 자신의 데이터 패킷을 보내기 전에 미리 긴 프리앰블을 주변 노드가 듣도록 송신함으로써 이웃 수신노드에게 패킷 전송이 임박했음을 통보한다. 이 통보는 주변의 노드들이 각자의 정해진 시간마다 깨어나 채널을 샘플링 할 때 감지될 수 있도록 한다. 만약 이 프리앰블에 의해 채널이 busy하면 계속 active 상태를 유지하면서 패킷의 수신을 완료한 후에 sleep 상태로 들어간다. WiseMAC에서 노드는 이웃노드가 wake up 스케줄을 알아야하며, hidden node 문제를 감는다.

#### d) DSMAC

DSMAC [14]은 데이터 전송 지연 문제를 해결하기 위해 동적인 듀티 사이클을 제안하였다. 일반적으로 S-MAC이나 listen/sleep의 주기를 갖는 프로토콜은 듀티 사이클을 미리 정의하여 통신을 하는데 비해, 데이터 트래픽을 고려해 동적으로 듀티 사이클을 변화시킴으로써 데이터 지연을 줄일 수 있는 기법을 제시하였다. 하지만 2홉 이내에서만 전송 지연 효과가 발생하므로 2홉 이상의 멀티 홉 네트워크에서는 여전히 전송 지연 문제가 남게 된다. 또한 지연 시간 및 듀티 사이클 정보를 저장하기 위하여 SYNC 패킷 및 데이터 패킷을 저장하는 영역이 증가하는 단점을 갖는다.

#### e) T-MAC

S-MAC을 기반으로 하여 만들어진 MAC이기 때문에 S-MAC이 가지고 있는 특성을 그대로 가지고 있다. Idle listening 시간을 최소화하기 위하여 각 노드는 일정 시간 TA(Timeout) 동안 아무런 트래픽 이 없다고 감지하면, 규정된 listen 구간의 만기 이전에라도 미리 sleep mode로 동작함으로써, 듀티 사이클이 트래픽 상황에 따라 적응하면서 감소될 수 있는 특징을 가지고 있다[15].

#### f) INS

기존의 듀티 싸이클 구조의 MAC 프로토콜이 데이터 전송을 위해 RTS와 CTS 전송을 위한 listen 구간의 개선에 집중한 반면, INS 프로토콜은 SYNC 패킷 전송을 위한 listen 구간을 개선하였다. 노드는 SYNC 패킷을 교환한 이후 아홉 번의 사이클 동안 SYNC를 교환하지 않는다. 즉 매 10번째 사이클의 SYNC 구간에서 모든 노드들이 자신의 듀티 사이클 동기를 인접한 주위 노드들에게 브로드캐스팅 한다. 그림 8의 INS 프로토콜에서는 인접한 노드들의 SYNC 주기를 상호 교환하고, 이를 이용하여 미리 주위 노드들의 data패킷 전송을 예측한다. 따라서 data 패킷 송신이 예측될 경우를 제외하고는 바로 sleep 상태를 유지할 수 있어 에너지 소모를 줄일 수 있는 특성이 있다[16].

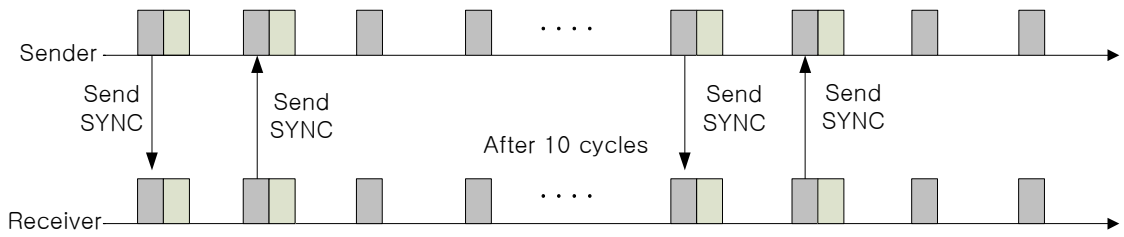


그림 7. INS 프로토콜의 기본 동작

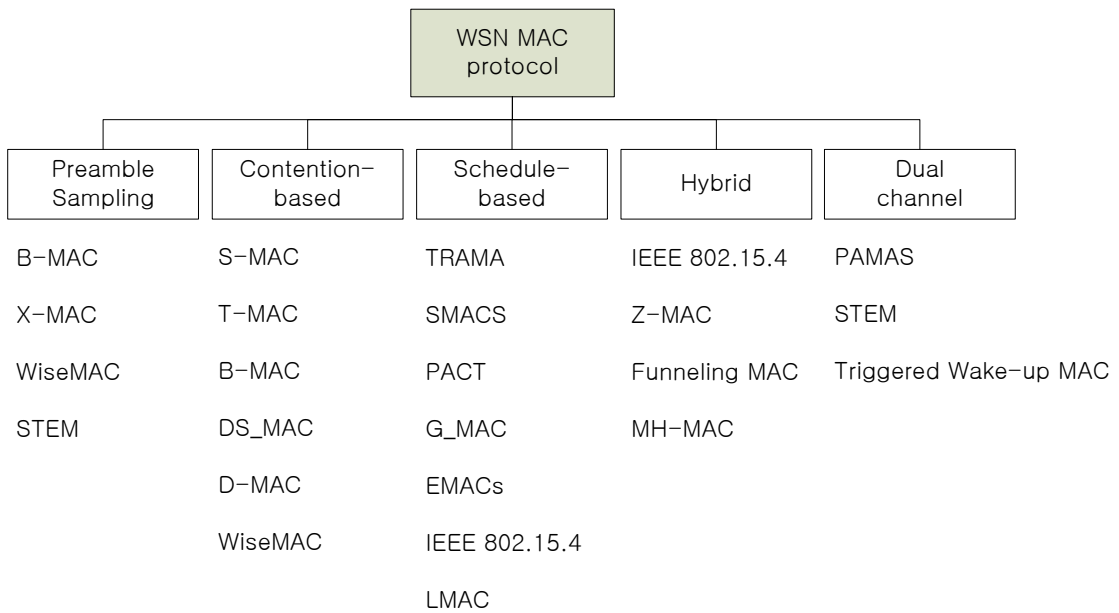


그림 8. 무선 센서 네트워크의 MAC 프로토콜 분류.

## B. IEEE 802.11e 블록 응답 기법

IEEE 802.11e MAC은 QoS의 향상을 위해서 Block Acknowledgment라는 기법을 사용한다[17]. Block ACK 기법에서는, 모든 프레임 각각의 ACK를 교환하는 대신에, ACK의 모든 데이터를 합하여, 모든 조각을 위한 하나의 Block ACK패킷을 사용한다.

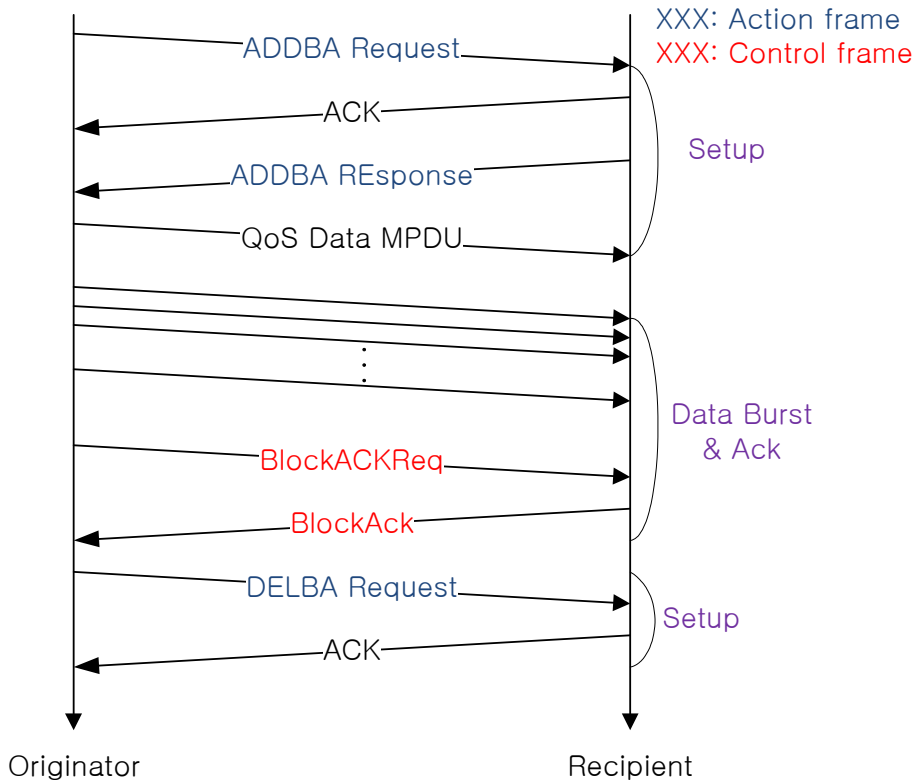


그림 9. Block ACK 동작 과정.

그림 9은 Block ACK의 동작 과정을 보여 준다. 크게 Action frame과 Control frame으로 나눌 수 있다. Action frame 프레임은 Block ACK 방식을 사용하기 위한 셋업 과정과 해제 과정에 관한 것이고 Control frame은 보낸 데이터에 대한 Block ACK의 요청과 Block ACK에 대한 것이다.

IEEE 802.11e에서 Block ACK 전송 기법은 옵션으로 존재하기 때문에 이를 사용하기 위해서 송신 단에서는 ADDBA Request 라는 요청 패킷을 보낸다. 수신 단에서는

ADDBA Request 에 대한 수신 확인 응답과 Block ACK 모드로 수신 하겠다는 ADDBA Response 패킷을 수신한다.

송신지는 모든 조각을 보낸 후에 BlockAckReq라는 패킷을 보내고 수신 측에서는 이를 받은 후에, Block ACK 패킷을 전송한다. BA 기법은 단 한번의 ACK 패킷 전송으로 채널의 QoS를 높일 수 있다.

Block ACK 전송 방식을 해제하기 위해서 송신 단은 DELBA Request 라는 패킷을 보냄으로써 이뤄진다.

그림 10와 그림 11는 각각 BA 기법의 두 가지 전송 타입인 immediate mode와 delay mode을 보여준다. Immediate mode는 높은 대역폭과 낮은 지연 트래픽에 적합한 반면, delay mode는 지연에 민감하지 않은 응용에 적합하다.

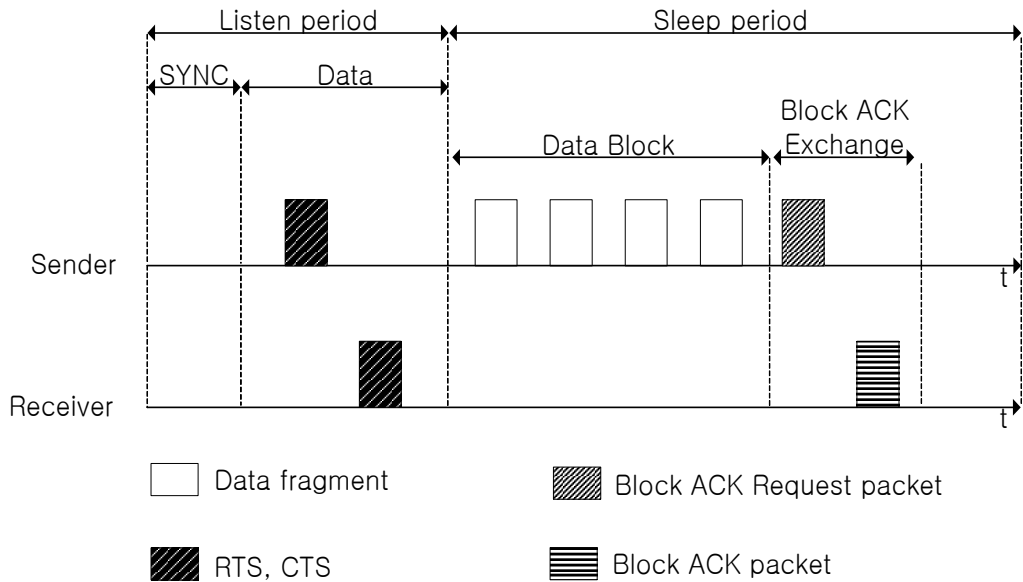


그림 10. Block ACK의 immediate mode



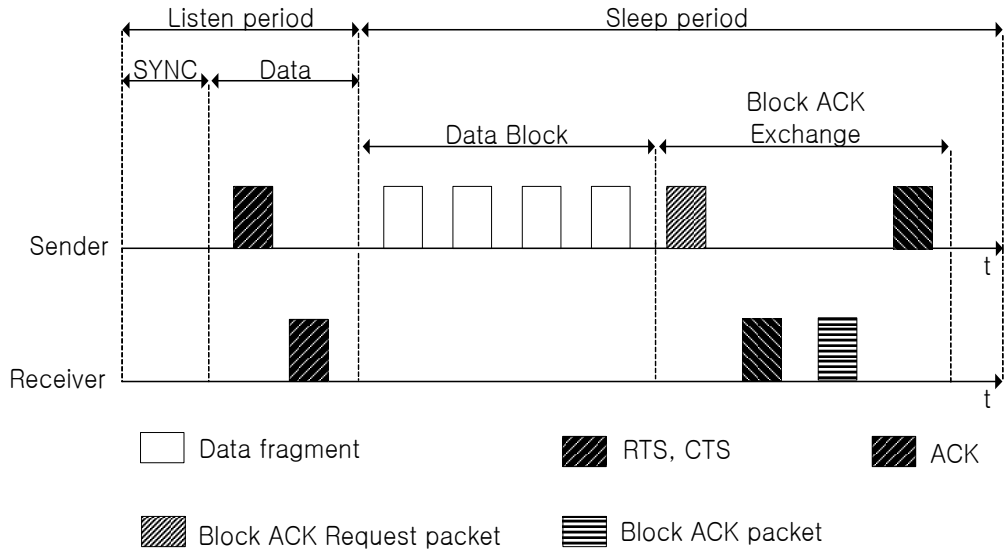


그림 11. Block ACK의 delay mode

### III. 제안하는 프로토콜의 설계

#### A. 기존의 조각화 전송과 문제점

최근의 무선 센서 네트워크는 멀티미디어 데이터를 전송하는 응용이 시도 되고 있다. 노드들은 서로 큰 데이터를 주고받는다. 긴 메시지를 한 번에 전송하는 것은 비효율적인 방법이다.

그림 12은 데이터 패킷이 손상되었을 때의 문제점을 보여준다. 송신 노드 A는 수신 노드 B에게 하나의 긴 메시지를 보낸다. 여기서 에러가 발생했다고 가정해보자. 이때 수신 노드 B는 수신된 전체 메시지를 버린다. 송신 노드 A는 메시지가 성공적으로 수신될 때 까지 계속해서 하나의 긴 메시지를 재전송하게 된다. 이제 조각화 전송을 사용하여 긴 메시지를 전송하는 방법을 살펴보자. 그림과 같이 송신노드C는 긴 메시지를 작은 조각들로 쪼개어 한 번에 하나 씩 전송한다. 수신 노드 D는 메시지 조각을 받을 때마다 ACK 패킷을 보냄으로써 수신이 잘 이뤄지고 있음을 알린다. 만약 전송 중에 어떤 조각이 에러가 발생했다면 송신 노드 C는 단지 에러가 발생한 조각 일부만 전송하면 된다.

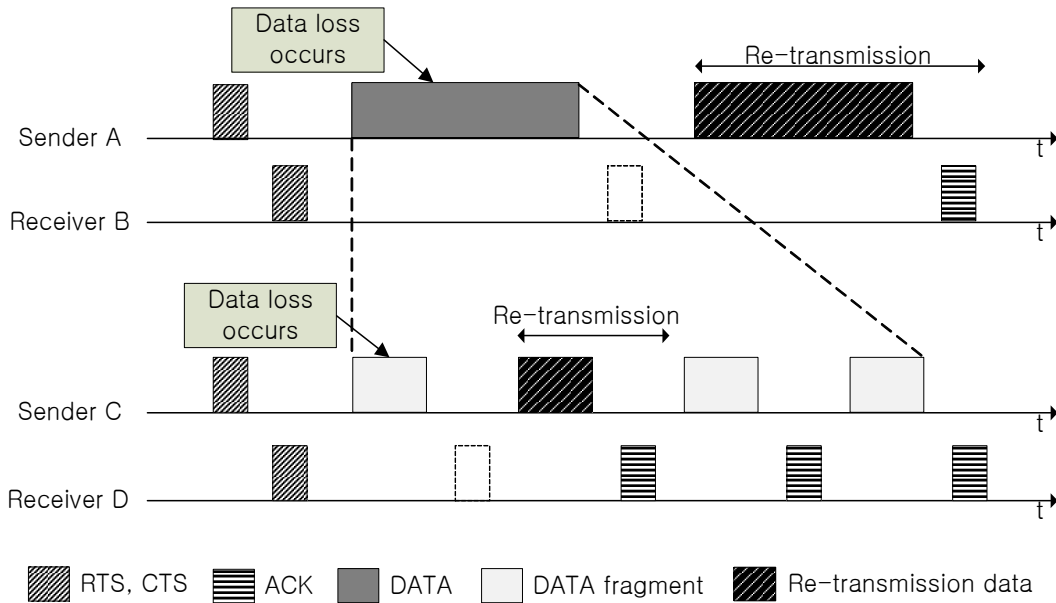


그림 12. 전형적인 조각화 전송 방법.

긴 데이터는 한 번에 보내는 것 보다 조각화 전송 방법이 보다 훨씬 더 효율적이지만, 조각화 전송 역시 몇 가지 문제점을 가지고 있다. 모든 조각들에 대한 ACK 패킷을 전송해야 될 필요성이 있기 때문에, 조각의 양이 커질수록 제어 패킷의 오버헤드를 발생 시키고 에너지 소비량이 증가하게 된다.

## B. 제안하는 프로토콜의 조각화 전송

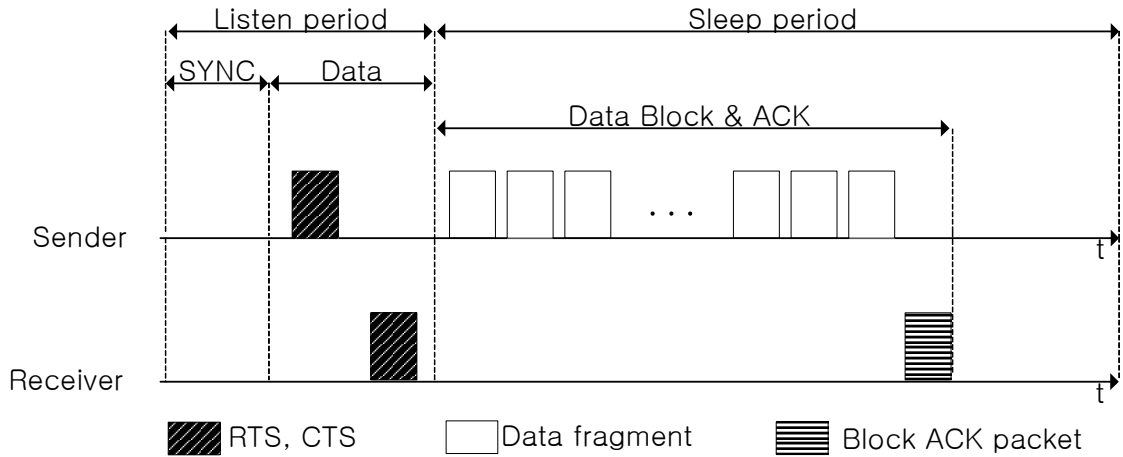


그림 13. 제안하는 프로토콜의 기본 동작.

앞에서 언급한 것처럼 긴 길이의 메시지는 여러 개의 조각들로 나눈다. 조각의 크기는 네트워크의 상황에 따라 변할 수 있다. 얼마나 많은 조각이 전송 될지 정해진 후에 그 정보는 RTS패킷을 통해 수신 노드에게 전달된다.

RTS와 CTS 패킷은 매 조각 마다 전송 되는 것이 아니고 맨 처음 한번의 RTS/CTS 교환에 의해 조각들이 전체 전송되는 시간을 예약한다.

ACK 패킷 또한 조각 들이 전송되는 동안에 단 한번의 전송을 한다. 전송 중에 패킷이 손실이 되든지 아니든지 간에, ACK 패킷은 모든 조각들을 수신한 후에 송신 노드에게 전송된다. ACK 패킷에는 수신 성공한 조각들의 정보가 비트맵 형태로 저장되어져있다. 이 비트맵 크기는 모든 데이터 조각들의 송수신 성공 여부를 저장할 수 있을 정도로 충분히 커야한다. 만약 10 개의 조각으로 나뉘어 전송이 되었다면 ACK는 송수신 여부를 위해 10 비트의 비트맵 공간을 생성하여 전송한다.

리슨 구간은 싱크 구간과 data 구간으로 이뤄져있다. 리슨 구간의 시간은 다음과 같이 결정된다.

$$T_{listen} = T_{sync} + T_{data} \quad (1)$$

$$T_{sync} = t_{difs} + t_{slot} \times CW_{sync} + TX_{sync} + t_{guard} \quad (2)$$

$$T_{data} = t_{difs} + t_{slot} \times CW_{data} + TX_{ctrlpkt} + t_{guard} \quad (3)$$

$t_{difs}$ 는 difs 프레임 시간 값으로 802.11에서와 같이 가장 평범한 프레임 간격을 갖는다.  $t_{slot}$ 은 타임 슬롯 하나의 시간을 나타내며  $CW_{sync}$ 와  $CW_{data}$ 은 각각의 경쟁 윈도우에서 사용되는 타임 슬롯의 개수를 나타낸다.  $TX_{sync}$ 와  $TX_{ctrlpkt}$ 은 동기화 패킷과 제어패킷이 전송되는데 필요한 시간을 나타낸다.

## C. 프로토콜의 패킷 구조

조각화 전송을 하게 되면 하나의 데이터는 여러 조각으로 나뉘어서 보낸다. 조각화 전송을 하려면 송신 노드와 수신 노드 간에 몇 가지 정보가 추가로 교환 되어야 한다. 첫 번째로 나누어지져 전송하게 될 전체 조각의 수이다. 이 정보는 송신 노드에서 패킷을 조각화 시킨 후에 RTS 패킷을 통해 전송하게 된다. 다음으로 추가 되는 정보는 데이터 조각의 번호이다. 현재의 조각이 몇 번째 조각인지에 대한 정보를 가지고 있어야 수신 단에서의 메시지 재조립을 할 수 가 있다. 따라서 데이터를 보낼 때 현재 보내는 조각의 번호를 헤더에 추가하여 전송하게 된다. 마지막으로 추가되어야 하는 중요한 정보는 Block ACK가 포함하고 있는 조각들의 전송 성공여부이다. 제안하는 프로토콜은 여러 조각의 ACK의 정보를 하나로 합쳐서 전송하게 된다. 이 정보는 비트맵 형태로 저장되며 재전송이 필요한 조각은 1이라는 값이 기록된다.

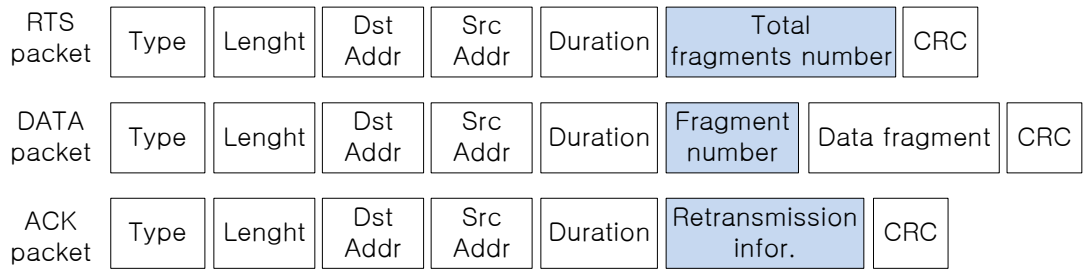


그림 14. 제안하는 프로토콜의 패킷 구조

## D. Duration

그림 14에서 나와 있듯이 각 패킷에는 duration 이라는 필드를 가지고 있다. 이 필드 값은 전송 시간이 계산된 값으로 패킷을 수신한 노드는 패킷으로부터 자신 또는 이웃의 전송이 언제 끝나는지 알게 된다.

RTS가 전송될 때 RTS의 duration의 값은 CTS부터 마지막 ACK의 수신까지 걸리는 시간이 계산되어 있다. RTS에 포함된 duration 값은 다음과 같은 방식으로 계산된다.

$$\text{duration} = 2 * \text{durCtrlPkt}_ + (\text{number of fragments}) * \text{durDataPkt}_ \quad (4)$$

durCtrlPkt\_ 값은 제어 패킷이 전송되는데 소요되는 시간이다. 2를 곱하는 이유는 전송 중에 CTS와 ACK가 사용되기 때문이다. durDataPkt\_ 값은 데이터 조각이 전송되는데 소요되는 시간이며 조각의 수만큼 값이 증가한다.

durCtrlPkt\_ 값과 durDataPkt\_ 값은 물리계층의 값들에 의해 계산된다. 시뮬레이션에서 사용하는 값처럼 9byte의 컨트롤 패킷과 50byte의 데이터 사용할 때 각각 11ms와 43ms 값을 가진다.

## E. 손상된 조각의 재전송

두 노드의 전송 중에는 제어 패킷과 데이터들이 교환된다. 이 때 그 패킷들은 신뢰성이 떨어지는 무선채널의 특성 때문에 손상되거나 손실 될 수 있다. 즉, 수신 노드에 게서 받은 ACK 패킷은 데이터의 전송의 성패를 알려주는데 있어서 매우 중요하기 때문에 ACK 패킷의 오버헤드를 줄이는 과정에서 조심스럽게 다뤄져야 한다.

재전송 절차는 다음과 같이 수행된다. 두 개의 노드가 서로 데이터를 교환하고 있다. 송신 노드는 모든 조각들을 보내고 나서 ACK 패킷을 수신 받기를 기다리고 있다. 수신 노드는 조각들을 받을 때마다 조각들의 번호를 검사하여 성공적으로 수신된 조각들을 번호를 기록한다. 만약 중복되어 수신한 조각들이 있다면 나중에 수신한 조각은 버리게 된다. 마지막 조각을 수신하게 되면 ACK에 성공적으로 수신된 조각들의 정보를 포함하여 ACK를 전송한다. 송신 노드는 ACK를 수신 받아 만약 전송에 실패한 조각이 있다면 그 조각들을 재전송한다. 그림 15에서 이 과정이 나타나있다.

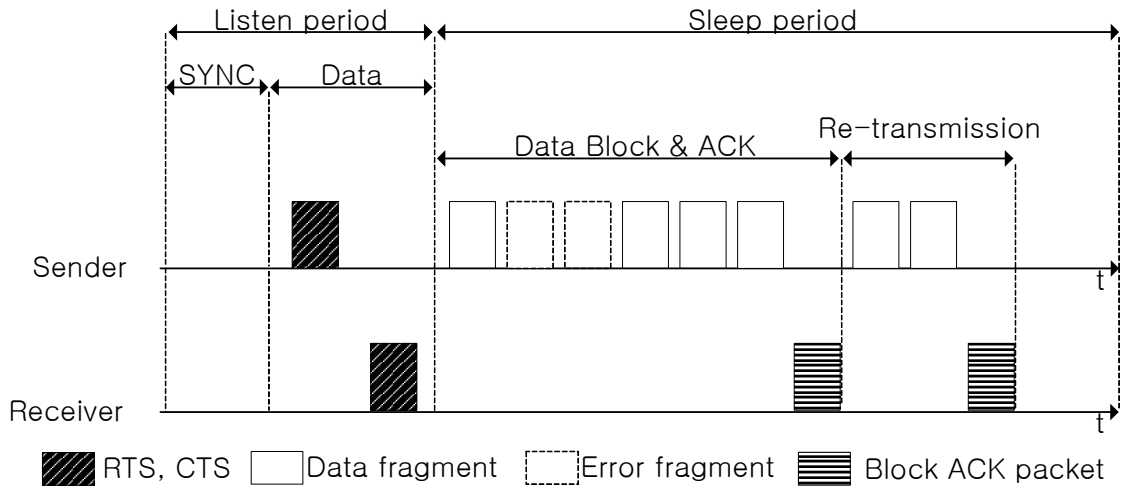


그림 15. 손실 된 데이터에 대한 재전송 방법.

만약, ACK 패킷이나 마지막 조각이 손실 된다면 통신은 그림 16처럼 진행되어야 한다. 송신 노드 ACK 패킷을 받지 못 했다면, 마지막 조각을 재전송 하게 된다. 수신 노드가 ACK 패킷을 보낸 이후에 마지막 조각을 다시 받으면 수신 노드는 ACK 패킷이 전송 중에 손실 된 것으로 파악하고 다시 ACK 패킷을 재전송한다.

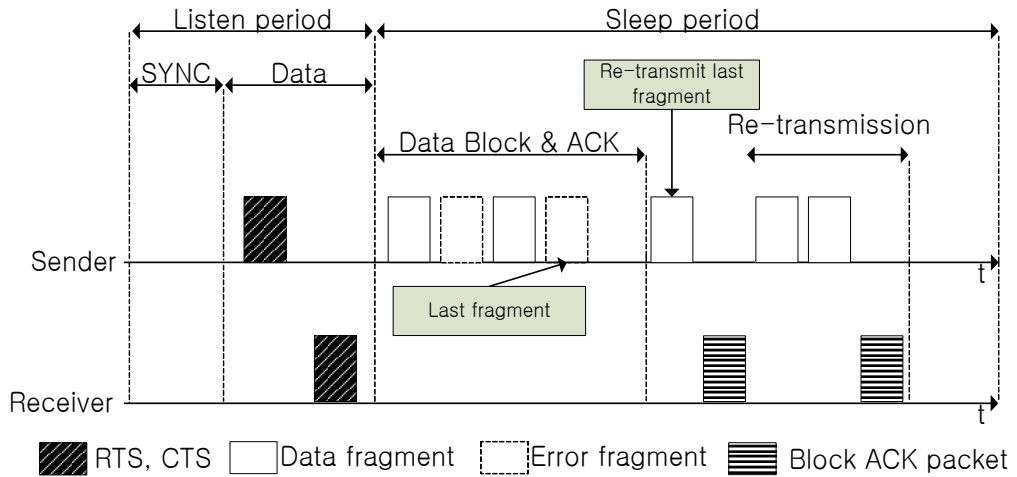


그림 16. 손실된 ACK에 대한 재전송 방법.

송신 노드가 ACK를 받지 못하는 이유는 두 가지 경우로 나누어 볼 수 있다. 첫 번째는 ACK 패킷 자체가 손실 된 경우이고, 두 번째 경우는 마지막 조각이 손실되어 수신 노드가 마지막 조각의 수신을 기다리고 있는 경우이다.

마지막 조각을 ACK를 요청하는 역할로 사용한다면, 마지막 조각이 손실 된 경우에 마지막 조각의 재전송과 ACK패킷의 요청하는 두 가지 이득을 얻을 수 있다. 이것이 IEEE 802.11e BA 기법처럼 송신 노드가 모든 조각을 보낸 이후에 ACK를 요청하는 메시지로써 BlockAckReq 패킷을 사용하지 않고 마지막 조각으로 ACK를 요청하는 이유이다.

## IV. 실험 및 고찰

### A. 시뮬레이션 설정

#### 1. NS-2 네트워크 시뮬레이터

본 논문에서 제안한 프로토콜의 성능을 평가하기 위하여 제안하는 프로토콜과 비교하기 위한 센서 네트워크 프로토콜들을 네트워크 시뮬레이터인 NS-2에서 구현하고 테스트 하였다[18]. 제안한 방식이 얼마나 효율적으로 동작하는지 평가하기 위해서 무선 센서 네트워크의 에너지 효율적인 MAC 프로토콜로 제안된 S-MAC, TEEM과 INS에 제안하는 프로토콜의 알고리즘을 적용하였다. 제안하는 프로토콜은 S-MAC 프로토콜을 기반으로 만들어진 프로토콜로 S-MAC BA 라는 이름으로 명명하였고, TEEM과 INS와의 성능 비교를 위해 제안하는 프로토콜의 알고리즘을 TEEM과 INS에 적용하여 TEEM BA, INS BA 라는 이름으로 실험하였다.

##### a) 에너지 모델

시뮬레이션에서 노드의 에너지 소비량을 측정하기 위해 에너지모델을 적용하였다. 에너지 모델을 적용하기 위해서는 다음과 같이 시뮬레이션 tcl 파일의 node-config 항목에 적용하면 된다.

```
$ns_ node-config -adhocRouting $opt(adhocRouting) W
...
-energyModel $opt(energymodel) W
-idlePower 0.014 W
-rxPower 0.014 W
-txPower 0.036 W
-sleepPower 0.000015 W
-transitionPower 0.028 W
-transitionTime 0.002 W
-initialEnergy $opt(initialenergy) W
```



기존의 NS-2의 에너지 모델은 sleep상태를 지원하지 않았고 버전 2.29에서부터 릴리즈 되었다. sleep 상태를 지원하기 위해 추가된 항목은 표2와 같다.

표2. 새롭게 추가된 파라미터.

항목 명	단 위	설 명
sleepPower	Watt (W)	sleep 상태에서의 에너지 소비량
transitionPower	Watt (W)	active 상태에서 sleep 상태로 전환시에 소비되는 에너지양
transitionTime	Second	active 상태에서 sleep 상태로 전환 되는 시간

## b) NOAH (No Ad-Hoc Routing Agent)

NS-2에는 무선 네트워크를 시뮬레이션 하기 위하여 특별한 routing 프로토콜들을 지원한다. 기본적으로 DSDV (Destination-Sequenced Distance-vector), DSR (Dynamic Source Routing), TORA (Temporally Ordered Routing Algorithm), AODV (Ad hoc On-demand Distance Vector) 4개의 ad-hoc routing 프로토콜을 지원한다.

이 중 어떤 라우팅 프로토콜도 시뮬레이션에 사용하는데 문제없지만 routing 프로토콜의 성능에 좌우되지 않고 순수하게 MAC 프로토콜의 성능을 평가하기 위해서 새로운 NOAH[19] 라우팅 프로토콜을 사용하였다. 이 routing 프로토콜은 노드들을 정적인 멀티 홉 라우팅을 하도록 만들어준다.

다음은 NS2 시뮬레이터에 NOAH를 적용 및 생성하기 위해 해당 에이전트를 추가하는 코드이다.

```
...
Simulator instproc create-noah-agent { node } {
    # Create a noah routing agent for this node
    set ragent [new Agent/NOAH]

    # setup address (supports hier-addr) for noah agent and mobile node
    set addr [$node node-addr]

    $ragent addr $addr
    $ragent node $node

    if [Simulator set mobile_ip_] {
        $ragent port-dmux [$node demux]
    }
    $node addr $addr
    $node set ragent_ $ragent
    return $ragent
}
...
```

다음은 시뮬레이션 tcl 파일에 NOAH를 적용하기 위한 코드이다.

```
...
# setup static routing for line of nodes
for {set i 0} {$i < $val(nn) } {incr i} {
    set cmd "[$node_($i) set ragent_] routing $val(nn)"
    for {set to 0} {$to < $val(nn) } {incr to} {
        if {$to < $i} {
            set hop [expr $i - 1]
        } elseif {$to > $i} {
            set hop [expr $i + 1]
        } else {
            set hop $i
        }
        set cmd "$cmd $to $hop"
    }
    eval $cmd
}
...
```

### c) 에러 모델

제안하는 프로토콜에서 재전송 기능이 원활 하게 동작하는지 확인하기 위해서 유니 폼 에러모델(Uniform Error Model)을 적용하였다. 유니 폼 에러 모델은 가장 단순한 형태의 오류 모델로써, 무선 채널을 시뮬레이션 할 때 많이 쓰이는 오류 모델이다. 유니 폼 오류 모델의 특징은 오류가 일어나는 패킷을 평균오류 율을 바탕으로 임의로 결정한다. 에러 모델을 추가하기 위해서는 다음과 같은 코드를 tcl 파일에 추가한다.

```
set opt(energymodel) EnergyModel ;
proc UniformErrorProc {} {
    set err [new ErrorModel]
    $err set rate_ 0.01
    return $err
}
```

**d) 시뮬레이션 파라미터**

시뮬레이션을 하기 위해 표 3과 같이 파라미터 값을 적용하였다. 이 값들은 시뮬레이션 환경을 설정하기 위한 값이다.

표3. 시뮬레이션을 위한 tcl 파라미터 값.

파라미터	값 (단위)
Idle Power	14 (mw)
Reception power	14 (mw)
Transmission power	36 (mw)
Sleep power	15 ( $\mu$ w)
Transition power	28 (mw)
Transition time	0.002 (s)
Initial energy	100 (J)
Error model	UniformErrorProc
Packet interval	10 (s)
Maximum number of packet to generate	100

**e) MAC 프로토콜 파라미터**

다음의 표 4의 값들은 프로토콜의 동작을 결정하는 중요한 역할을 하는 파라미터들이다.

표 4. 프로토콜 동작 관련 파라미터

파라미터	값 (단위)
Synchronization period	10cycle
Duty Cycle	10 (%)
Channel bandwidth	20 (kbps)
Cycle time	1433 (ms)
Sleep time	1289.8 (ms)
listen time	143.2 (ms)
Sync time	55 (ms)
Data time	88.2 (ms)
durData	43 (ms)
durCtrl	11 (ms)

## B. 시뮬레이션 및 결과 고찰

### 1. Control packet overhead

본 논문에서 에너지 소비를 줄이기 위해서 데이터 조각들이 전송 되는데 사용되는 컨트롤 패킷의 수를 줄이는 방법을 제안했다. 제어 패킷의 전송량을 측정하기 위해서 에러 발생 확률을 0%로 설정하고, 3개의 노드를 일렬로 배치하였다. 5초 단위로 생성되는 100 개의 트래픽을 각 각 10개의 조각으로 나누어서 전송하였다. 결과는 표 5와 같이 나타났다.

표 5. 에러 없는 환경에서의 이상적인 ACK 전송량

S-MAC	TEEM	INS	S-MAC BA	TEEM BA	INS BA
2000	2000	2000	200	200	200

에러가 발생했을 경우에 증가하는 ACK의 전송량을 측정하기 위하여 같은 조건하에 에러 전송률을 1%에서 5%까지 변경해가면서 ACK의 전송량을 측정하였다. 그림 17, 그림 18과 그림 19는 각각 S-MAC, TEEM, INS의 프로토콜과 제안하는 프로토콜을 비교한 결과이다.

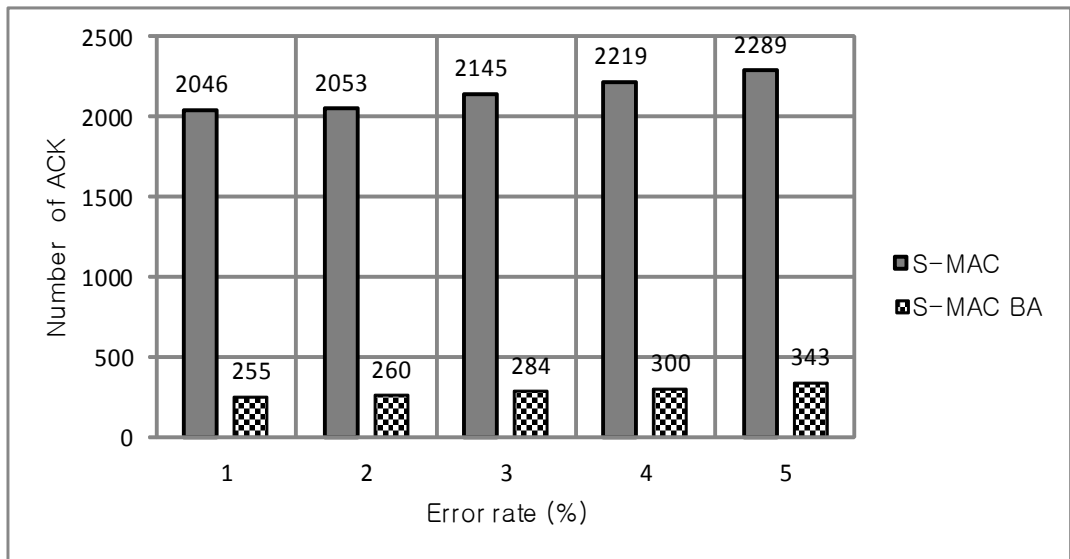


그림 17. S-MAC 프로토콜과 제안하는 프로토콜과의 ACK 전송량 비교.

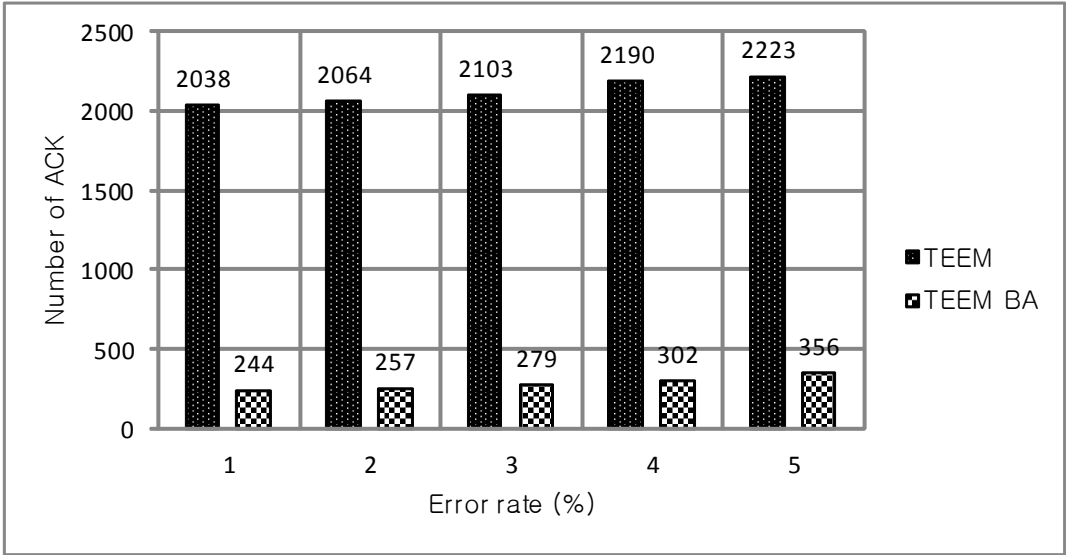


그림 18. TEEM 프로토콜과 제안하는 프로토콜과의 ACK 전송량 비교.



그림 19. INS 프로토콜과 제안하는 프로토콜과의 ACK 전송량 비교.

## 2. End to End Delay

관련연구에서 언급하였듯이, 듀티 사이클을 사용하는 프로토콜은 데이터 지연 문제가 있다. 본 논문에서 제안하는 프로토콜은 역시 데이터 지연 문제가 발생한다. 두 번째 실험에서 측정하고자 하는 것은 ACK패킷의 사용을 최소화 하였을 때 전송 시간이 얼마만큼 변하는가이다.

첫 번째 실험과 동일 한 조건에서 에러율과 데이터 조각 수를 변경해가며 실험하였다. 그림 20는 에러 발생 확률이 0%일 때, 5조각부터 15조각 까지 데이터를 나누어 전송한 실험 결과 값이다.

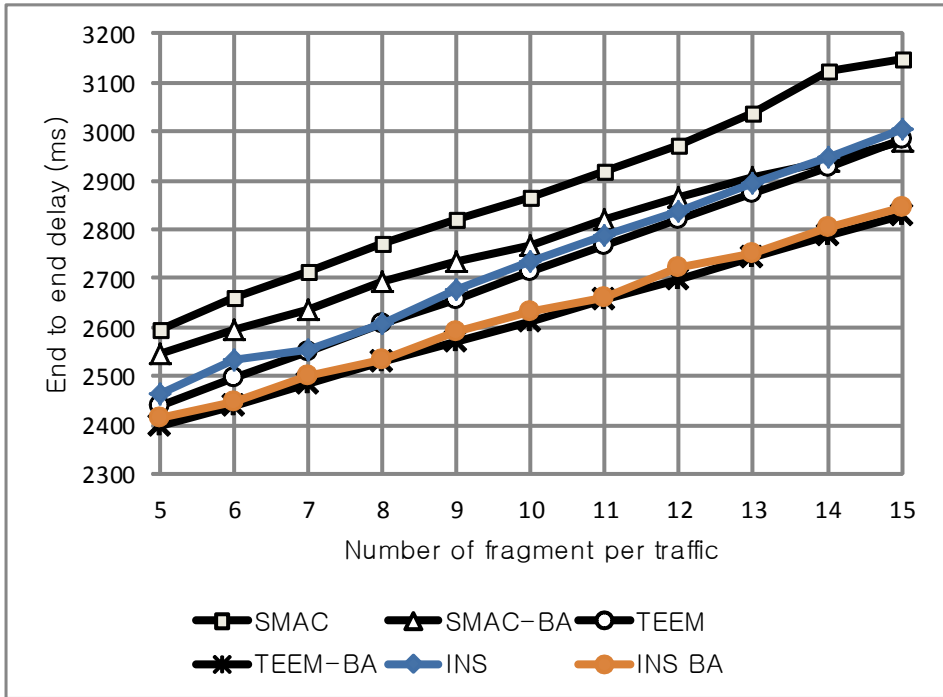


그림 20. 조각 수의 증가에 따른 전송 지연 시간. 에러율 0%.

데이터가 목적지 노드까지 도달하는 시간은 원래의 프로토콜과 제안하는 프로토콜을 비교해 보면 조각 수가 증가할수록 그 차이가 점점 벌어지고 있는 것을 눈으로 확인할 수 있다. 표 4를 보면 durData값과 durCtr값은 데이터가 전송되는 시간과 제어 패킷이 전송되는데 걸리는 시간을 의미한다. 각 각 43ms, 11ms로 설정되어있다. 따라

서 조각 전송과 함께 ACK의 전송도 같이 이루어 S-MAC 프로토콜의 경우 조각 하나의 증가 할 때 마다 54ms 씩 증가하는 셈이다. 반면에 제안하는 프로토콜은 몇 개의 조각이 증가하더라도 하나의 ACK 만이 전송되기 때문에 데이터 조각이 증가 할 때마다 43ms 씩 증가하게 된다. 따라서 이 실험에서 제안하는 프로토콜은 S-MAC과 TEEM 프로토콜보다 (전체 조각 수 -1) \* 11(ms) 의 시간만큼 전송이 빨리 끝나게 된다.

그림 21은 에러 발생 확률이 5% 일 때의 실험 결과이다. 에러에 의해 재전송이 이루어지고 있기 때문에 그래프의 기울기가 일정하지 않지만 조각 수가 증가함에 따라 넓어지는 기울기 값의 차이는 제안하는 프로토콜의 성능이 원래의 프로토콜보다 전송 시간을 단축하는데 있어 효율적인 방식임을 보여주고 있다. 한편 TEEM 프로토콜이 S-MAC 프로토콜에 비하여 전송 시간이 짧은 이유는 S-MAC 보다 짧은 listen 구간을 갖기 때문이다.

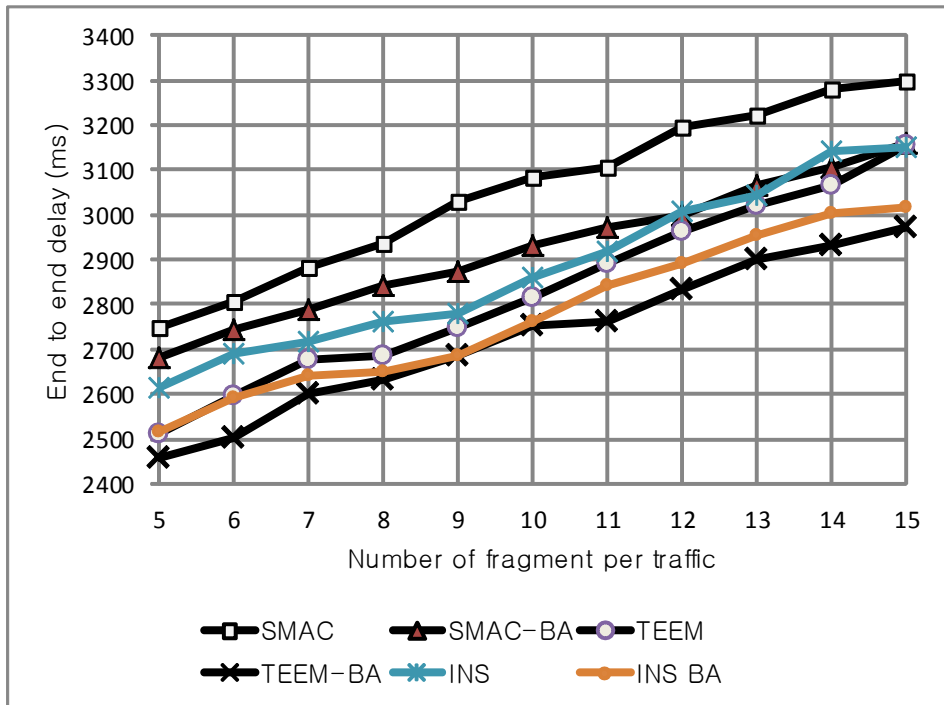


그림 21. 조각 수의 증가에 따른 전송 지연 시간. 에러율 5%



### 3. Energy consumption

서론에서 밝힌 바와 같이 이 논문의 제안 목적은 에너지 효율적인 프로토콜을 설계하는 것에 있다. 에너지 소비량 측정은 두 번째 실험에서 각 노드의 에너지 소비량을 측정하는 것이다.

그림 22의 실험 결과는 송신노드의 에너지 소모량을 측정하는 결과이다. 그래프를 보면 제안하는 프로토콜이나 원래의 프로토콜들 간에 에너지 소비량의 차이가 거의 없다. 송신노드는 제어 패킷을 보내지 않고 받기만 한다. ACK를 수신 하는 시간에는 프로토콜 간에 차이가 있지만 idle 상태에서의 에너지 소비량과 데이터를 수신 할 때의 에너지 소비량은 동일하다. 따라서 에너지 소비량 증가는 조각의 수에 따라 증가하고 있다.

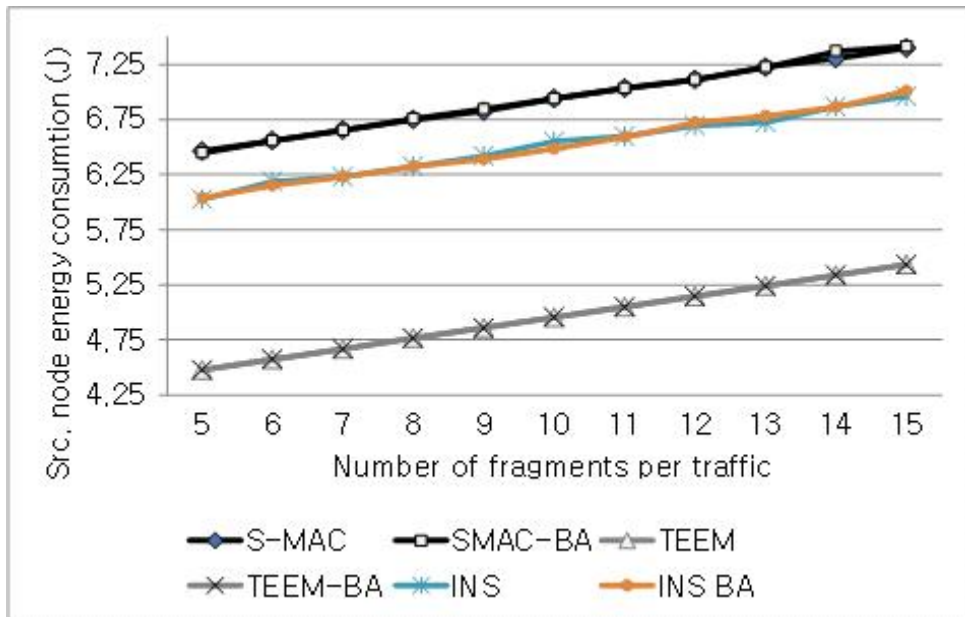


그림 22. 송신 노드의 에너지 소비량.

그림 23는 중간 노드의 에너지 소비량을 측정된 결과이다. 송신 노드보다 훨씬 많은 에너지를 소비하는 것을 알 수 있다. 제안하는 프로토콜의 경우에 S-MAC과 TEEM 보다 훨씬 완만한 기울기 곡선을 보여주고 있다. 송신 노드의 에너지 소비량 측정 결과에서처럼 데이터 조각의 전송에 따른 에너지 소비량 변화 차이는 거의 발생하지 않기 때문에 이러한 기울기의 차이는 ACK의 전송에 의해서 발생한 것임을 알 수 있다.

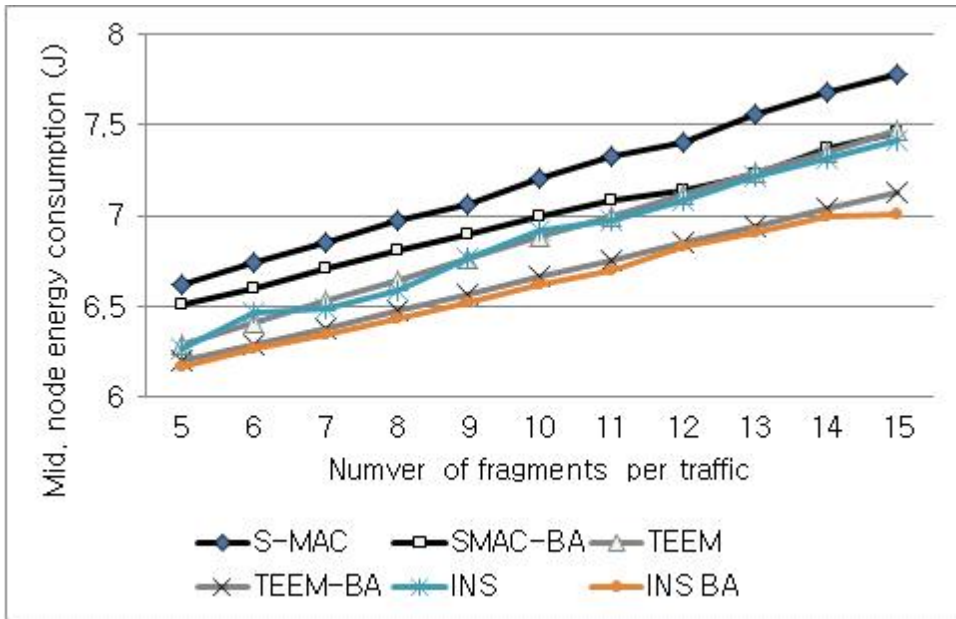


그림 23. 중간 노드의 에너지 소비량.

그림 24은 목적지 노드의 에너지 소비량을 측정된 것이다. S-MAC과 TEEM의 경우에 조각 수의 증가에 따른 에너지 소비량이 증가하지만 제안하는 프로토콜의 경우에 에너지 소비량이 조각 수에 거의 동일한 정도를 유지하고 있다. 이런 결과가 나오는 수신 받는 조각의 수가 증가 하더라도 전송하는 ACK의 수는 한 개로 동일하기 때문이다. 반대로 수신 받는 조각마다 ACK를 전송하는 S-MAC과 TEEM은 수신 받는 조각의 수가 증가 할 때마다 ACK의 전송 횟수도 증가하기 때문에 에너지 소비량 또한 증가하게 된다. 15조각일 때 제안하는 프로토콜에서 에너지 소비가 7% 감소하였다.

그림 25은 세 노드의 에너지 소비량 평균값을 나타낸 것이다. 이 전의 실험 결과와 비슷한 결과를 보이고 있다.

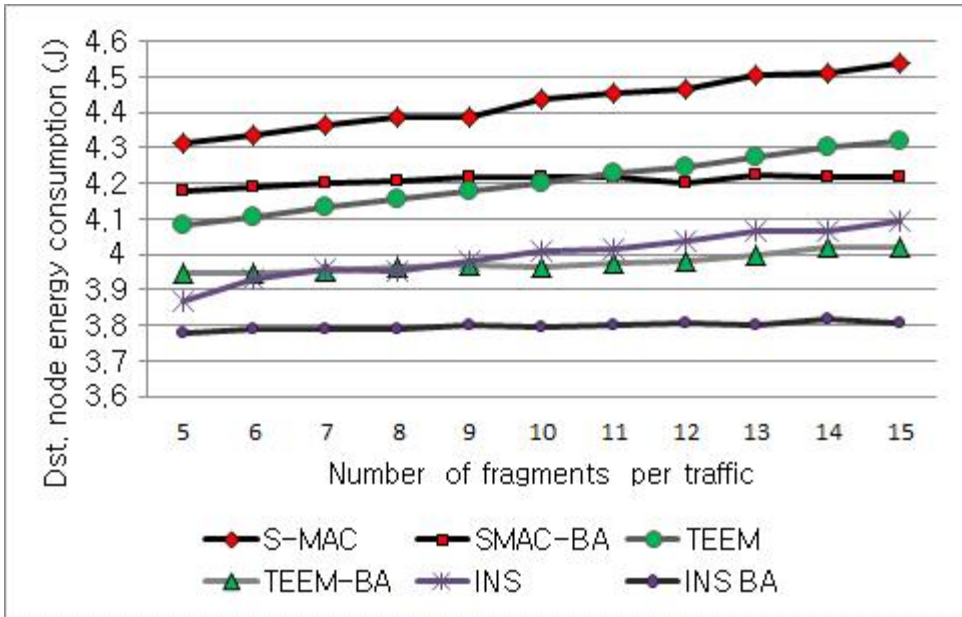


그림 24. 목적지 노드의 에너지 소비량.

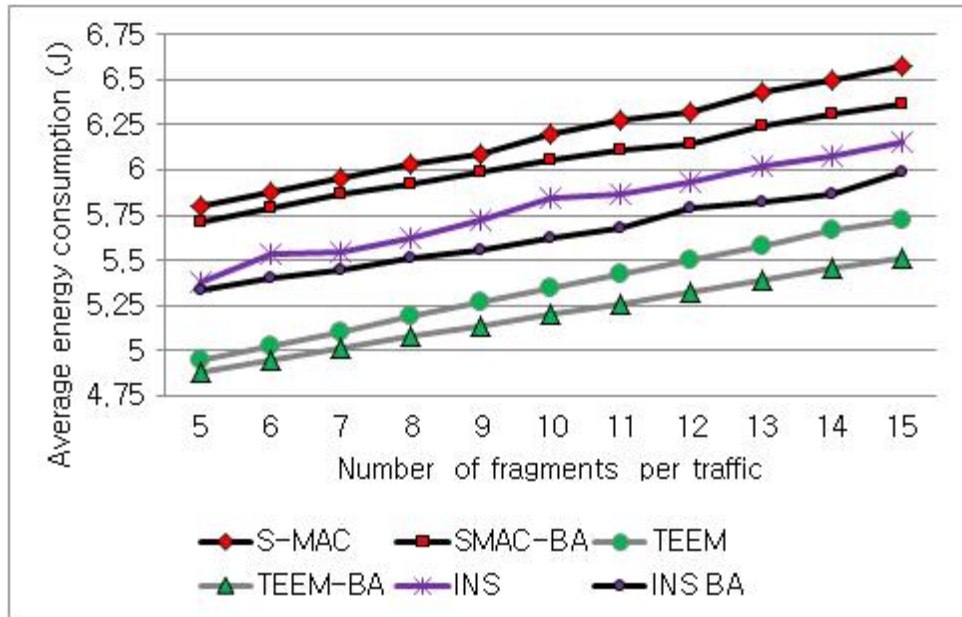


그림 25. 세 노드의 에너지 소비량 평균 값.

## V. 결론 및 향후 연구과제

본 논문에서는 무선 센서 네트워크에서 효율적으로 에너지 소비를 줄이는 맥 프로토콜을 제안하였다. 최근의 무선 센서 네트워크의 응용이 온도·습도·조도 등과 같은 소량의 데이터를 수집하는 것뿐만 아니라 이미지와 음성과 같은 데이터 역시 수집하는 형태로 변화 되어 가고 있다. 특히 무선 센서 네트워크는 배터리로 동작하기 때문에 에너지의 소비에 민감하며 가장 많은 에너지 소비 요인은 노드간의 통신이다. 따라서 얼마나 효율적인 맥 프로토콜을 사용하느냐에 따라 노드의 생존 시간이 달라진다.

기존의 무선 센서 네트워크의 맥 프로토콜은 사이즈가 큰 데이터를 전송하는데 적합하지 않게 설계 되어있다. 본 논문에서 제안된 전송 기법은 데이터를 조각화하고 이를 블록 응답 형태로 전송함으로써 제어패킷으로 인한 오버헤드와 효율적인 재전송 방법을 제안하였다.

본 논문은 실험 결과에서처럼 애러가 없는 환경에서 15조각으로 나누어 전송하였을 때 세 노드의 평균 에너지 소비량은 기존의 프로토콜 대비 약 3.15%의 에너지 감소를 달성하였다. 또한 제어 패킷의 최소화 하여 조각화 전송 시 한 조각 당 11ms의 전송 시간을 단축하였다. 그리고 기존의 여러 맥 프로토콜에 제안하는 아이디어가 적용이 가능함을 보였다.

본 논문에서 제안하는 아이디어는 기존의 맥 프로토콜처럼 센서에서 수집한 데이터만을 전송하는 응용에서도 사용가능하고 영상과 이미지 음성 등을 사용하는 응용에서도 또한 가능하다.

계층 구조를 갖는 프로토콜은 최적화된 성능을 발휘하기 어렵다. 따라서 상위 계층 및 하위 계층의 정보를 바탕으로 좀 더 에너지 효율적인 프로토콜을 만들기 위한 cross layer 방식으로 전환 되어야 할 것이다.

마지막으로 S-MAC 프로토콜 같은 경우는 이미 실제 상용 센서 노드에서 사용할 수 있도록 구현되어있다. 향후 연구 과제에서는 본 논문에서 제안한 프로토콜을 실제의 센서 노드에서 구현하는 노력이 필요하다.

## 참고 문헌

- [1] S. Plancoulaine, A. Bachir and D. Barthel, "WSN Node Energy Dissipation," France Telecom R&D, internal report, July 2006.
- [2] R. Alesii, F. Graziosi, L. Pomante and C. Rinaldi, "Exploiting WSN for Audio Surveillance Applications: the VoWSN Approach," 11th EUROMICRO CONFERENCE on DIGITAL SYSTEM DESIGN Architectures, Methods and Tools(DSD 08), 2008, pp.520-524.
- [3] Quan Zhou, Yongjun Xu and Xiaowei Li, "HTSMAC: High Throughput Sensor MAC for Wireless Video Networks," Wireless Communications, Networking and Mobile Computing, pp.2428-2431, 2007.
- [4] V. Jelicic, V. Bilas, "Reducing Power Consumption of Image Transmission over IEEE 802.15.4/ZigBee Sensor Network," Instrumentation and Measurement Technology Conference (I2MTC), pp.1211-1215, 2010.
- [5] I.F. Akyildiz, T. Melodia, and K.R. Chowdury, "Wireless multimedia sensor networks: A survey," Wireless Communications, IEEE, vol.14, pp.32-39, 2007.
- [6] 한백전자기술연구소, "유비쿼터스센서네트워크시그템," 2006.5
- [7] 전자신문사, "훤히 보이는 RFID/USN" 2008.6
- [8] 김지은, 김세한, 정운철, 김내수, "USN 센서 노드 기술 동향," 전자통신동향분석, 제22권 제 3호, 2007.6
- [9] LAN MAN Standards Committee of the IEEE Computer Society, "Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications," IEEE Std 802.11-1999.
- [10] W. Ye, J. Heidemann, and D. Estrin," An energy-efficient MAC protocol for wireless sensor networks," in IEEE INFOCOM 2002, vol.3, pp.1567-1576, June 2002,
- [11] W. Ye, J. Heidemann, and D.Estrin, "Medium access control with coordinated, adaptive sleeping for wireless sensor networks," IEEE/ACM Transactions On Networking, June 2004.

- [12] Changsu Suh and Young-Bae Ko, "A traffic aware, energy efficient MAC protocol for wireless sensor networks," Proc. of the IEEE International Symposium on Circuits and Systems, vol.3, pp.2975-2978, May 2005.
- [13] A. El-Hoiydi, J-D. Decotignie, C. Enz, and E. Le Roux, "WiseMac, an Ultra Low Power MAC Protocol for the WiseNET Wireless Sensor Networks," In proceedings of ACM SenSys, Los Angeles, CA, November 2003.
- [14] P. Lin, C. Qiao and X. Wang, "Medium Access Control with a Dynamic Duty Cycle for Sensor Networks," In Proceedings of IEEE WCNC, Atlanta, GA, March 2004.
- [15] T. van Dam, K. Langendoen, "An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks," Proc. of ACM SynSyys 03, Los Angeles, California, USA, November 2003.
- [16] P. Sthapit and J.-Y. Pyun, "Intelligent Network Synchronization for Energy Saving in Low Duty Cycle MAC Protocols," Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WowMom), vol. 1, pp.1-6, June 2009.
- [17] LAN MAN Standards Committee of the IEEE Computer Society, "Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications," IEEE Std 802.11e-2005.
- [18] NS-2 website, <http://www.isi.edu/nsnam/ns/> .
- [19] The NO Ad-Hoc Routing Agent  
website, <http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/> .