

# TinyOS기반의 Ad-Hoc네트워크 상에서 BaseNode의 데이터 수신성능에 관한 연구

한병희<sup>o</sup> 김문기 이수용 김지홍 김용현 홍윤식  
인천대학교 컴퓨터공학과

{ byunghhee<sup>o</sup>, moonki2003, bluegenii, riot999, yh-kim, yshong }@incheon.ac.kr

## A Study on the Throughput of Receiving Using TinyOS in Ad-hoc Network

Byunghhee Han<sup>o</sup>, Moonki Kim, SooYong Lee, Jihong Kim, YongHyun Kim and YounSik Hong  
Dept. of Computer Science and Eng. University of Incheon

### 요 약

무선 센서노드 사용한 데이터 계측 및 제어 기술은 특히 홈 네트워크 분야에 널리 적용되고 있다. 본 논문에서는 Ad-Hoc네트워크의 말단인 Base node와 인접한 노드들 사이에 조도 값을 실시간으로 전송하는 시스템을 구현하였다. 이는 OscilloscopeRF라고 하는 TinyOS에서 제공되는 프로그램이 사용하는 메시지 구조를 분석하여, 각 노드의 ID 및 계측 데이터를 추출하였다. 이렇게 추출된 계측 데이터를 센서 네트워크의 Base node에서 데이터 통제 센터(DOC)로 효율적으로 전송하기 위해 TCP 기반 네트워크 프로그래밍을 구현하였다. 실험 결과 센서 노드 수 및 샘플링 주기에 상관없이 안정적으로 계측 데이터 수신이 이루어짐을 확인하였다. 또한, IEEE802.11a/g 기반 무선 네트워크를 통해 계측 상황을 휴대용 단말기인 PDA에서 확인할 수 있도록 이를 구현하였다.

### 1. 서 론

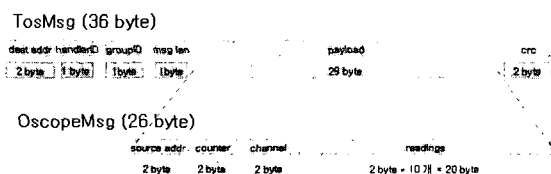
최근 외국어에 대한 관심이 높아지고 있다. 읽기, 말하기, 듣기 등의 영역으로 나누어져 있는데 그 중 정확한 상대방의 의견을 받아들이고 파악해야 하는 듣기가 중요하다.

본 논문에서는 TinyOS기반의 센서모듈 사이의 네트워크 연결에서 다른 센서들이 송신하는 데이터를 수신하는 센서모듈의 RF수신능력을 이야기 하기 앞서서, 기반 OS가 되는 TinyOS와 사용되는 어플리케이션에 관해서 알아보고, 테스트를 위한 개발 환경에 관해서 설명한 다음, 개발된 어플리케이션을 이야기 하고, 테스트 내용을 결론으로 다루려고 한다.

### 2. 연구 배경

UC Berkeley 대학에서 센서 네트워크를 위해서 개발된 소형 OS형태인 TinyOS는 기능별로 작은 컴포넌트를 만들어 두고 이것들의 연결 방식을 설정해 주어서 하나의 어플리케이션을 구성하는 방식으로 되어 있다[1].

TinyOS 1.1 버전을 설치하면 기본으로 제공되는 ListenRaw라는 분석 도구를 이용하여 다음과 같은 패킷 구조를 알 수 있었다.



OscilloscopeRF (37 byte)



[그림 1] ListenRaw로 본 각 메시지의 구조

여기서 OscilloscopeRF란 센서노드에 포팅하여 사용하는 TinyOS에서 기본 제공하는 어플리케이션으로 센서노드가 현재 받고 있는 밝기(조도)를 10개씩 'readings'에 채워서 TosMsg의 payload 부분에 넣고, 완성된 메시지를 같은 groupID를 가진 불특정 다수의 노드에게 브로드캐스팅하는 프로그램이다[2].

표시	크기	내용
start	1	0x7E 메시지의 시작
ack	1	0x42는 ack 신호를 다시 보낼 필요가 없음을 의미
dest addr	2	0xFFFF는 브로드캐스트
handlerID	2	0x0A는 OscscopeMsg
groupID	1	0x7D는 디폴트 그룹
msglen	1	메시지의 길이
counter	2	메시지를 보내면 카운터 증가
source addr	2	노드 번호
channel	2	채널
reading	20	노드가 수신한 조도 값. 0x0242와 같이 2byte씩 데이터를 저장
crc	2	데이터의 유효성 판별
end	1	0x7E로 데이터의 끝

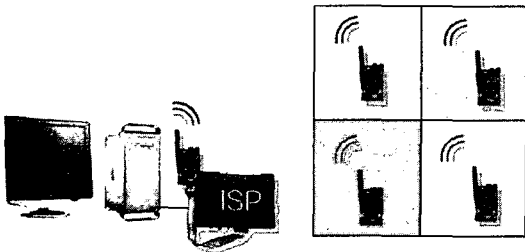
[표 1] OscilloscopeRF의 메시지 구조

\*본 논문은 2005년도 산업자원부지정 인천대학교 멀티미디어연구센터 지원에 의한 것임.

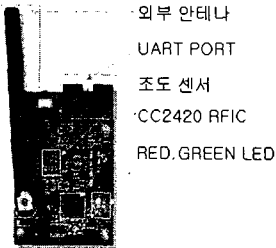
### 3. OscilloscopeRF를 이용한 RF테스트 시스템 구현

이번 RF테스트에서 테스트 하고자 하는 것은 센서모들간의 통신 방식 중에서 Ad-hoc방식의 소규모를 생각해 해서 센서모들간의 통신상태가 1-hop관계이다[3].

이때 사용한 센서노드의 마이크로컨트롤러는 ATmega 128L이 사용되고, RFIC로는 CC2420 칩을 사용하였는데 CC2420은 ISM대역이라고 하는 2.4GHz의 무선 통신 주파수 영역을 사용한다[4].



[그림 2] 테스트의 구성



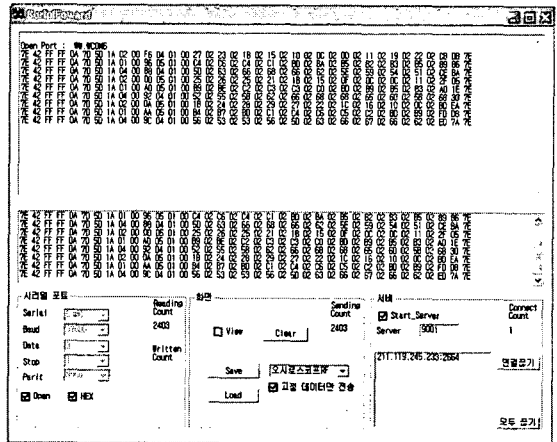
[그림 3] 센서 노드의 상세 설명

전체적인 테스트의 구성은 [그림 2]에서와 같이 각 센서노드들은 각자의 환경에 위치 하고 있고, 환경 정보를 수집하여, 주변있는 볼특정 다수노드에게 브로드캐스팅을 한다. 이 때 정보의 수집을 담당하는 Base node는 이 데이터들을 수집하고, 자신이 연결되어 있는 ISP에 UART인터페이스로 데이터를 전송하게 된다. 이때 ISP의 역할은 UART로 입력된 데이터를 시리얼인터페이스로 출력해주는 역할을 하고, 다른 한편으로 Base node에게 전원을 공급해주는 역할을 하고 있다.

#### 3.1 서버 프로그램

ISP가 연결된 PC는 서버역할을 하고, 내부 기능으로 SERIAL로 데이터를 입력 받는데, 입력받은 데이터는 앞뒤 구분 없는 연속적인 16진수의 데이터이기 때문에 앞서 말한 OscilloscopeRF 메시지의 구조를 이용해서, 한개의 센서노드에서 보낸 데이터의 집합인 하나의 '패킷 데이터' 단위로 가공 처리 하기위한 파싱작업을 하는 클래스를 구성하였다. 파싱의 결과물은 네트워크로 전송이 가능하도록 구성하였다. PC클라이언트나 PDA 클라이언트가 서버로 접속을 요청할 경우 서버는 현재 접속한 클라이언트에게 버퍼에 저장되어 있는 데이터를 전송

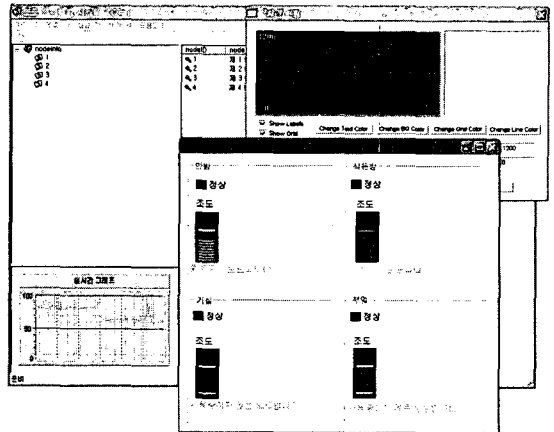
한다. 현재 전송되고 있는 네트워크 패킷의 원시적인 데이터의 모습(Base node에서 수집한)과 네트워크로 전송하는 패킷의 모습도 화면에서 볼 수 있게 하였고, 이중 원시 데이터의 리스트는 텍스트파일로 저장할 수 있게 하였다.



[그림 4] 서버 프로그램의 동작화면

#### 3.2 클라이언트 프로그램

서버에서 수집된 데이터는 TCP/IP의 형태로 접속한 클라이언트에게 데이터를 전송하는데, 일반 PC와 PDA 클라이언트 두 가지 형태가 존재한다.

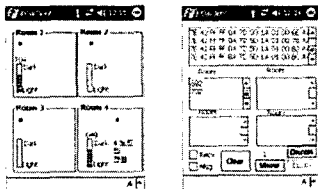


[그림 5] PC클라이언트의 동작화면

클라이언트 프로그램은 서버에서 받아온 데이터를 센서노드의 아이디인 'source'와 환경변수 'reading'을 부분을 이용해서 동작한다. [그림 5]는 클라이언트 프로그램이다. 클라이언트는 센서의 노드(4개)의 개수 만큼 영역을 가지고 있고, 수신 패킷중 'source'로 센서ID를 구별해서 어떤 센서에서 보낸 데이터 인지 판별한 후 'reading'의 데이터 부분의 값을 10진수로 변환한 다음 그래프로 표현된다. 그래프는 세로 프로그래스바 형태로

되어 있고, 기준 값을 정해서 기준값 보다 높은 수치를 나타낼 경우 '붉은색'으로 그래프를 표시하도록 하였고, 센서가 동작 도중 오류, 배터리의 방전, 고장 등으로 인하여 동작을 멈춘 경우를 대비 해서 각 센서 노드의 변화를 확인해서 변화가 없이 2000 ms 이상 지속된 경우에는 동작이상을 표시하도록 했다.

PDA 클라이언트는 가장 문제 되었던 화면 크기와 Processing 때문에 필수적인 모니터링 기능만을 넣어서, 간단히 서버에 접속하고 데이터를 받고, 수신된 데이터를 세로방향 프로그래스와 이상 유무를 나타내는 표시 등으로 이루어져 있다.



[그림 6] PDA 클라이언트 프로그램의 동작화면

구분	CPU	OS	기타
서버(PC)	Pentium 4 2.4GHz	Windows Server2003	RAM: 1GB LAN: 10/100Mbps RFIC: CC2420 (IEEE 802.15.4)
Node	ATmega 128L	TinyOS	Data Transfer Rate: 250Kbps RAM: 1GB LAN: 10/100Mbps
클라이언트 (NoteBook)	Pentium M 1.73GHz	Windows XP	RAM: 1GB LAN: 10/100Mbps
클라이언트 (PDA)	PXA270 624MHz	Pocket PC 2003	RAM: 128M LAN: 802.11b

[표 2] 서버와 센서 노드 클라이언트의 구성

#### 4. Base node의 데이터 수신 능력

1-hop 관계에 있는 노드들 간의 통신중 Base node의 수신 능력과 안전한 데이터의 수신이 가능한지 측정 하기 위해서 실험을 하였다. 각각의 환경에 위치에서 조도 값을 수집하는 센서 노드는 일정주기에 의해 10번의 환경변수를 수집하고 이 데이터를 브로드캐스팅 하는데, 이때 환경변수를 담은 데이터가 서버 컴퓨터에 까지 도달 하는데 가지 걸리는 시간은

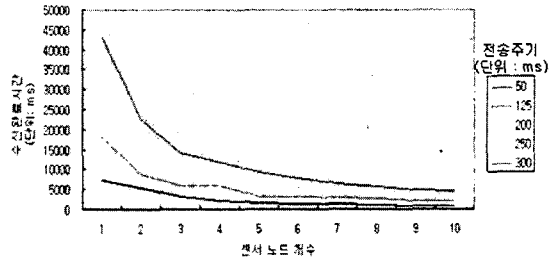
$$T = (\text{센싱주기} \times 10) + W + S + \alpha$$

여기서 T는 서버PC에 도착시간, W는 환경데이터를 센서 노드에서 수집하고 패킷의 'reading'에 넣고 CC2420 RFIC로 브로드캐스팅을 마칠 때 까지의 시간이고, S는 Base node에서 데이터를 수집하고 ISP를 통해서 서버 PC에 수신이 완료되는 시간을  $\alpha$ 는 방해요소들로 인한 지연을 나타낸다. 서버 PC로 전송된 데이터는 센서에서 보낸 데이터를 기준으로 100개를 수신할 때 까지의 시간을 측정하였고 이때 데이터의 크기는 20Byte  $\times$  100 개로 2,000 Byte가된다.

센싱주기의 간격을 50, 125, 200, 250, 300 ms 단위로 변화를 주어 실험했는데 이 값은 TinyOS를 포팅하기

전 설정하고 포팅한 후에는 고정되어 있는 특성이 있다. 그리고 센싱주기의 간격을 늘리는 것과 동시에 센서의 개수에도 변화를 주어서 Base node의 수신능력을 테스트 하였다.

센서 노드 개수와 수신 시간 관계



[그림 7] 센서 노드 개수와 수신 시간 관계 분석

위의 [그림 7]의 그래프에서 처럼 전송 주기를 고정시키고 센서의 개수를 증가시킴에 따라 데이터를 수신이 완료 되는데 까지 걸리는 시간이 증가하는 것이 아니라 반대로 줄어드는 것을 볼 수 있다. 이것은 Base node의 수신 대역폭 내에 있다는 것을 알 수 있다. 연구 결과를 이용해서 어떤 센서 네트워크의 구성을 하기전에 그 규모에 따라서 전송주기를 설정해 줌으로써 보다 안전한 데이터 수신이 가능하도록 할 수 있다.

#### 5. 결론

TinyOS기반의 센서네트워크를 구성할 수 있는 노드에 OscilloscopeRF 프로그램을 이용해서 Ad-Hoc네트워크에서 Base node와 그 인접 노드들 간의 수신능력(수신대역폭)을 확인하였다. 센서네트워크에서의 통신은 Wired-Network TCP/IP에서의 통신과는 달리 목표 지점에게 직접 데이터를 전달하는 방식이 아닌 브로드캐스트를 통한 라우팅방식을 이용하기 때문에 Base node와 1-hop의 관계에 있는 센서노드들에서 주고 받는 데이터는 2-hop이상의 거리에서 브로드캐스트되는 데이터들간의 손실되는 데이터를 RFIC수신능력(대역폭)으로 해결할 수 있다고 가정하고 수신대역폭에 관한 연구를 진행하였다. 이는 센서네트워크를 구성하는 센서노드의 수가 많아짐에 따라 발생하는 데이터 수집 손실을 줄이기 위해 센서의 개수에 적합한 전송 주기를 찾아내는 것이 그 목적이라 할 수 있다.

향후 연구 과제로는 RFIC의 출력에 따른 전송거리와 편중되지 않는 데이터의 수신을 위한 연구와 전송주기와 RFIC출력을 병행하여 에너지 효율 증가를 위한 Ad-Hoc네트워크 연구를 진행할 예정이다.

#### 참고문헌

- [1] TinyOS Community forum, <http://www.tinyos.net/>
- [2] TinyOS Oscilloscope Tutorial Part, <http://www.tinyos.net/tinyos-1.x/doc/tutorial/>
- [3] C.Siva Ram Murthy and B.S. Manoj, Ad Hoc Wireless Networks, PRENTICE HALL, 2004
- [4] Chipcon, <http://www.chipcon.com/>
- [5] Feng ZHAO, Leonidas GUIBAS, WIRELESS SENSOR NETWORKS, ELSEVIER, 2004
- [6] 정완영, 이현창, 권성열, 유비쿼터스 컴퓨팅을 위한 센서&인터페이스, 성안당, 2005