

에너지 효율 제약조건을 가진 센서 네트워크 모니터링 시스템 구현

이기욱[†], 심창규^{**}

요 약

최근 유비쿼터스 컴퓨팅 환경에 관한 연구가 활발하게 이루어지고 있으며, 특히 센서 네트워크 기술은 유비쿼터스 컴퓨팅 환경의 핵심 기술이다. 무선 센서 네트워크는 주변의 관심 데이터를 감지하고 수집하기 위해서 물리적 공간에 배치된 센서 노드들로 구성된다. 센서 네트워크를 구성하는 각 센서 노드는 제한된 자원을 이용하여 필요한 서비스를 수행할 수 있어야 한다. 또한 무선 센서 네트워크를 구축함에 있어서 제한된 자원을 이용하는 센서 노드의 에너지 소비를 효율성 있게 사용해야만 전체 네트워크의 수명을 연장할 수 있다. 본 논문에서는 효율적으로 에너지를 사용하여 특정 지역의 환경 데이터를 수집하고 전송하여 원격지에서 컴퓨터를 통해 실시간으로 변화되는 환경 정보를 모니터링할 수 있는 시스템을 구축한다.

Implementation of Sensor Network Monitoring System with Energy Efficiency Constraints

Ki Wook Lee[†], Chang Gyu Seong^{**}

ABSTRACT

As the study of ubiquitous computing environment has been very active in recent years, the sensor network technology is considered to be a core technology of it. This wireless sensor network is enabled to sense and gather data of interest from its surroundings by sensor nodes applied in physical space. Each sensor node structuring the sensor network is demanded to execute the required service using limited resources. This limited usage of resources requires the sensor node to energy-efficiently perform in building wireless sensor network, which enables to extend the entire network life. This study structures a system able to monitor changing environment data on a real-time basis using a computer remotely as it energy-efficiently gathers and sends environment data of specific areas.

Key words: Wireless Sensor Network(무선 센서 네트워크), Monitoring System(모니터링 시스템), Energy-Efficient(에너지 효율), Rembeer(램버어)

1. 서 론

최근 정보기술은 초소형 컴퓨터를 주변 사물에 내장하여 물리공간과 네트워크 기반의 가상공간을 융합하여 새로운 유비쿼터스 컴퓨팅 세계를 창출하여

인간의 생활을 보다 안전하고, 생산적이며, 풍요하게 변화시키고 있다. 보이지 않는 컴퓨터에 의해 사물은 지능화되고 인간은 실세계의 상황을 보다 정확하게 인지할 수 있다[1]. 무선 센서 네트워크 유비쿼터스 컴퓨팅을 위한 기반 기술로, 광범위하게 설치되어 있

※ 교신저자(Corresponding Author): 이기욱, 주소: 부산시 남구 신천로 179번지(608-711), 전화: 051)629-1432, FAX: 051)629-1419, E-mail: kwlee@tu.ac.kr
접수일: 2009년 5월 25일, 수정일: 2009년 8월 6일

완료일: 2009년 9월 14일

[†] 정회원, 동명대학교 향단물류학부 부교수

^{**} 정회원, 해양대학교 컴퓨터공학과
(E-mail: astroboy007@nate.com)

는 네트워크 인프라를 통해 상황 인지를 위해 개발된 다양한 센서 디바이스가 감지한 환경 데이터를 응용 서비스와 연동하는 기술이다[2]. 무선 센서 네트워크 기술을 응용하여 의료, 군사, 산업, 스마트 홈, 재난 방지, 환경 감시, 생태 조사, 교통 정보, 건축물 관리, 실시간 보안, 생산물 유통 및 물류 등 다양한 분야에 활용될 수 있다[3].

무선 센서 네트워크와 같이 많은 수의 센서 노드가 조밀하게 분포되어 센서 노드간의 협업을 통하여 정보를 모으고 전달하는 유기적인 시스템에서는 가능한 많은 노드들이 오랫동안 생존하는 것이 네트워크의 수명에 중요한 요인이 된다.

본 논문에서는 효율적으로 에너지를 사용하기 위하여 데이터를 빠르게 전송하는 것 보다는 전체 네트워크의 수명에 중점을 두었다. 일부 센서 노드의 에너지 고갈로 인한 네트워크의 심각한 문제를 해결하기 위하여 네트워크 전체적인 에너지 효율성을 고려하는 라우팅 프로토콜을 적용한 무선 센서 네트워크를 구현하였다. 제한된 시스템은 외부 환경으로부터 발생할 수 있는 온도와 조도에 대한 환경 정보를 현장에 배치된 센서 노드들이 감지, 수집한 데이터를 분석하여 사용자가 실시간 상황 대처 및 적절한 대응 방안을 수행할 수 있다.

2. 관련 연구

2.1 무선 센서 네트워크

무선 센서 네트워크(Wireless Sensor Network; WSN)는 매우 작은 크기의 독립된 무선 센서들을 건물, 산림, 도로, 인체 등의 물리적 공간에 배치하여 주위의 온도, 조도, 습도, 가속도, 기울기 등의 정보를 무선으로 실시간 감지 및 관리할 수 있는 기술이다[4].

그림 1은 센서 노드, 싱크 노드, 전송 네트워크 및

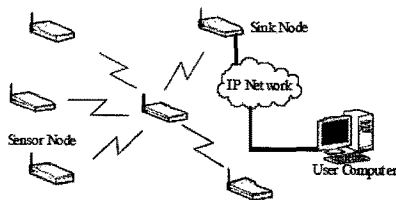


그림 1. 무선 센서 네트워크 시스템의 개념도

컴퓨터로 구성된 무선 센서 네트워크 시스템이다[5].

무선 센서 네트워크 시스템은 일반적으로 그림1처럼 센서노드, 싱크 노드, 전송 네트워크 및 컴퓨터로 구성되어 있다. 무선 센서 네트워크 시스템에서 센서 노드에 의하여 수집된 정보를 무선 네트워크를 활용하여 동일한 형태의 이웃하는 센서 노드를 통하여 수집 노드(sink node)로 전송한다. 그리고 무선 센서 네트워크는 센서 노드의 제한된 전송범위로 인해 멀티 홉(Multi-hop) 통신을 해야 한다[6].

무선 센서 노드는 일반적인 무선 노드와 달리 소형화 및 경량화에 따른 제한된 컴퓨팅 및 전력 자원을 갖게 된다. 따라서 이 노드들로 구성된 무선 센서 네트워크는 일반 무선 네트워크와 다음과 같은 차이점을 갖게 된다.

- 무선 센서 네트워크는 주변 환경 상태를 측정하기 위해서 다수의 센서 노드들로 구성되어야 한다.
- 제한된 컴퓨팅 및 전력 자원으로 인해 무선 센서 네트워크는 일반 무선 네트워크에 상대적으로 높은 실패율을 가질 수 있다.
- 긴급 상황이나 위급한 센싱 데이터의 경우 빠른 전송이 필요하다.
- 배터리를 통해 전원을 공급받기 때문에 효율적인 에너지 소모 기술이 필요하다.
- 다수의 센서 노드가 배치되었을 경우, 센서 노드 스스로 동작하여 네트워크를 형성할 수 있는 능력이 필요하다.

무선 센서 네트워크의 구축의 대표적 사례로는 UC Berkeley의 GDI(Great Duck Island) 프로젝트로 바다제비의 서식지 굴속에 4인치 크기의 센서 노드를 설치하고 게이트웨이에서 센서 정보를 수집하여 수백 미터에 걸쳐 실시간으로 데이터를 수집하였다.

레드우드(Redwood) 프로젝트는 삼나무 전체에 80개 정도의 센서노드를 설치하여 전체 나무들 주위의 온도와 습도를 측정하여 세밀한 지점의 기후를 감시하여 전체 나무의 상태를 파악할 수 있다.

2.2 무선 센서 플랫폼

2.2.1 하드웨어

무선 센서 네트워크를 위해 개발된 센서 네트워크 하드웨어는 1999년 미국 UC Berkeley 대학에서 WeC 라는 첫 번째 플랫폼을 개발한 후, 매년 RENE, DOT,

MICA, MICA2, MICA2 DOT, MICAZ와 같은 센서 노드 하드웨어가 개발되었다. 이들 하드웨어 플랫폼은 미국의 Crossbow란 회사에 의해서 상업화되어 시장에 공급되고 있다[7]. 그 외에도 인텔의 iMote, ETH의 btNode, Moteiv의 Telos, 한백전자의 Zigbex, 옥타컴사의 Nano-24 등이 개발되어 공개되었다. 센서 노드들은 최근 MEMS(Micro-Electro-Mechanical System) 기술을 이용하여 초소형화와 저가격화, 초집적화된 형태로 계속 발전해 나가고 있다. 센서 노드는 센싱 장치(Sensing Unit), 처리 장치(Processing Unit), 통신 장치(Communication Unit), 그리고 전력장치(Power Unit)로 구성되어 있다. 그림 2는 센서노드의 플랫폼 구조를 보여준다.

그림 3은 대표적 센서 하드웨어인 MICA Mote로서 메인보드와 센서보드로 구성되어 있다. 메인 보드는 마이크로 컨트롤러와 무선 인터페이스를 내장하고 있고, 센서 보드는 온도, 조도, 자기 센서 등과 같은 센서를 내장하고 있다.

2.2.2 TinyOS

센서 네트워크를 구성하는 센서 노드는 마이크로 컨트롤러를 내장한 소형 컴퓨터 시스템으로 센싱 응용 처리와 노드 간의 통신 등을 위해 운영체제가 필수적으로 요구된다. 표 1은 무선 센서 네트워크에서 사용되는 운영체제들을 비교하였다.

TinyOS는 오픈 소스 운영체제로 무선 센서 네트워크와 같은 소형 임베디드 시스템을 위해 미국 UC Berkeley 대학에서 개발된 대표적인 무선 센서 네트

표 1. 센서 네트워크 운영체제 비교

	TinyOS	MANTIS	SOS	나노Qplus
멀티 태스킹	불가	가능	불가	가능
기반 언어	nesC	C	C	C
동적 메모리할당	없음	있음	있음	있음
실행모델	이벤트 드리븐	쓰레드 드리븐	이벤트 드리븐	쓰레드 드리븐

워크 전용 운영 체제이다. 일반 운영체제와 프로그래밍 언어는 소형의 무선 센서 노드 플랫폼에 적용하기 부적합하므로 TinyOS와 NesC 같은 센서 네트워크 전용 운영체제와 언어를 제공한다[8]. TinyOS의 특성 및 요구사항은 적은 코드 크기, 효과적인 리소스 이용, 저전력 소모, 강한 동작, Ad-Hoc 센서 네트워크 프로토콜, 분산처리 등이다. TinyOS의 어플리케이션의 구조는 그림 4와 같이 어플리케이션 부분만 개발자가 구현하고, 나머지 부분은 TinyOS에서 제공하는 컴포넌트를 이용한다. 이 구조는 개발자에게 메인(Main)과 서브(Sub)의 기능들을 서로 연결(wiring)하는 방식으로 어플리케이션을 구현하고 라이브러리로 인해 소스코드의 재사용성을 높인다.

TinyOS는 제한된 전력을 사용하기에 이벤트가 발생했을 경우에만 하드웨어가 동작하는 방식(Event-driven)을 사용한다. 그래서 무선 센서 네트워크 작동을 위해서는 세 가지 중요 인터럽트인 Timer, Sesing, Communication의 인터럽트가 발생되어야만 하드웨어를 활성 상태(Active mode)로 전환할 수 있고, 나머지는 휴면 상태(Sleep mode)로 반복하여 에너지 소모를 최소화시키면서, 메시지를 효율적으로 처리한다.

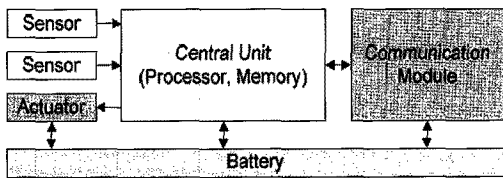


그림 2. 센서 노드의 구조

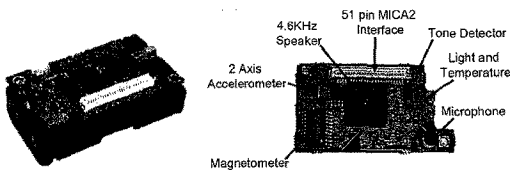


그림 3. MICA 하드웨어 플랫폼

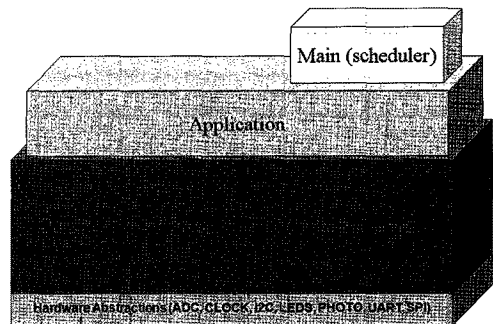


그림 4. TinyOS 어플리케이션 구조

2.3 라우팅 프로토콜

무선 센서네트워크의 라우팅은 일반적인 유·무선 네트워크의 라우팅과 다른 방향으로 접근해야 한다. 일반적인 유·무선 네트워크의 목표는 데이터나 자원을 공유하는 것이지만 센서네트워크는 정보 획득을 위한 작업을 수행하는 것이다. 그리고 기존 네트워크는 노드의 식별자(Identifier)를 기반으로 하는 통신을 수행하지만, 센서네트워크는 각 노드에서 수행되는 데이터 및 위치 등의 노드가 지니는 속성(attributer)에 기반으로 한다.

센서 네트워크의 라우팅 프로토콜은 네트워크 구조에 의해 크게 평면 기반 라우팅(flat based routing), 계층 기반 라우팅(hierarchical based routing) 그리고 위치 기반 라우팅(location based routing)으로 나눌 수 있다[9,10]. 평면 기반 라우팅은 모든 노드들이 동등한 입장에서 공통된 하나의 라우팅 기법을 사용하는 방식으로 주요 라우팅 프로토콜로는 Directed diffusion, SPIN 등이 있다. 계층 기반 라우팅은 각 노드들이 일정 집합을 구성하고 임의의 헤더(header)를 선출하여 헤더가 직접 혹은 다른 헤더와의 통신을 통해 센싱 데이터를 수집 노드에게 전달하는 방식이다. 주요 라우팅 프로토콜로는 LEACH 등이 있다. 위치 기반 라우팅은 센서 노드의 위치 정보를 기반으로 라우팅 경로를 설정하는 방식으로 센서 노드들이 이미 자신의 위치를 알고 있다고 가정한다. 각각의 노드는 자신의 이웃 노드의 정보만을 유지하고, 네트워크 전체에 플러딩 사용이 없어 네트워크의 확장성이 좋다. 주요 라우팅 프로토콜로는 GPSR, LAR 등이 있다.

3. 시스템의 구현

3.1 시스템 구성

센서네트워크 모니터링 시스템은 환경 데이터를 감지하고 센싱 하기 위한 센서노드, PC와 센서 노드 간 연결을 위한 싱크 노드, 그리고 센싱한 환경 데이터를 사용자에게 가시화 하고 분석할 수 있는 모니터 컴퓨터로 구성된다. 그림 5는 센서 모니터링 시스템 구조이다.

물리적 관심 영역에 배포된 센서 노드들은 무선 센서 네트워크를 형성하며 싱크 노드에는 각 노드에

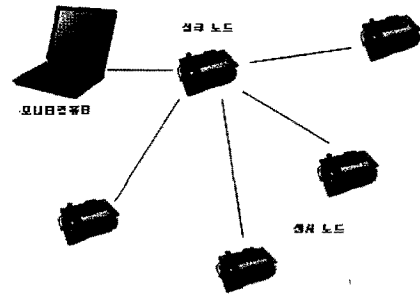


그림 5. 센서 모니터링 시스템 구조

서 센싱된 온도 및 조도 데이터가 전송된다. 싱크 노드는 센서 노드로부터 전송된 데이터를 수집하여 모니터링 컴퓨터에 직렬 통신을 통해 데이터를 전송할 수 있다. 모니터 컴퓨터는 실제로 센서네트워크에서 전달되는 데이터를 실시간으로 받을 수 있으며 그의 다양한 응용 프로그램을 개발할 수 있는 플랫폼이 되고, 구현된 모니터링 시스템은 센서 노드에서 전송된 온도와 조도 정보의 변화를 가시화하였다.

3.1.1 하드웨어 구성

구현된 센서 모니터링 시스템의 하드웨어 플랫폼은 Crossbow사의 Micaz 계열의 mote를 기반으로 조도와 온도 데이터 감지하여 라우팅을 통하여 송신하기 위해 사용된 다수의 센서 노드와 센서 정보를 수집하여 모니터 시스템으로 보내는 하나의 싱크 노드(게이트웨이)로 구성되어 있다. 싱크 노드와 모니터 시스템과 RS232 통신으로 연결하여 테스트하였다.

그림 6는 구현된 센서 네트워크 모니터링 시스템에 사용되는 하드웨어 구성이다.

3.1.2 소프트웨어 구성

구현된 센서 모니터링 시스템의 소프트웨어의 플랫폼은 노드용 어플리케이션, 싱크용 어플리케이션

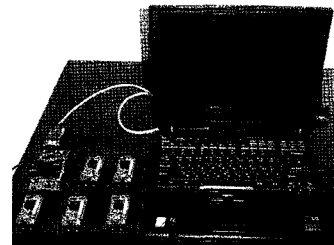


그림 6. 하드웨어 실험 환경

표 2. RF 패킷 구성

구분	크기(byte)	설명
Dest Address	2	목적지 주소
Handler ID	1	메시지 핸들러
Group ID	1	그룹 구분자
Message Length	1	메시지 길이
Source Address	2	보내는 노드 ID
Hop Count	2	홉 수
Channel	2	데이터 채널
ADC Data	20	데이터 값

및 모니터 어플리케이션으로 구성하였다.

노드용 어플리케이션은 센서 노드에서는 환경 데이터를 수집하여 RF 통신으로 싱크노드로 데이터를 전송하도록 구현하였다. RF 패킷의 구성은 표 2와 같다.

온도와 조도를 센싱하기 위해서 ADC에 DemoSensorC와 Temp 컴포넌트를 연결하여 동시에 센싱할 수 있도록 하였다. 또한 Rembeer 라우팅을 이용하여 네트워크를 구성하도록 하였다. 그림 7은 노드용 어플리케이션의 컴포넌트들 간 연결(wiring)에 대한 구성도이다.

시스템의 초기화 및 스케줄러 구동을 위한 Main, RF 통신을 위한 GenericComm, LED 제어를 위한 LedsC, 온도 데이터를 수집하기 위한 Temp, 조도 데이터를 수집하기 위한 DemoSensorC, 무선 라우팅을 위한 HMRembeerRouter, 타이머 제어를 위한 TimerC 컴포넌트로 구성되어 있다.

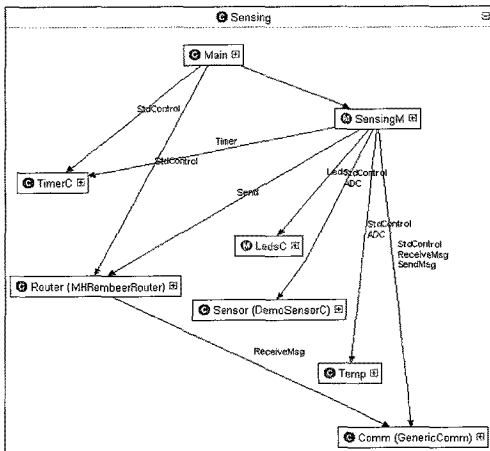


그림 7. 센싱 어플리케이션 컴포넌트 연결도

싱크용 어플리케이션은 무선 통신 및 RS232 통신을 제어하여 센서 노드로부터 보낸 데이터를 PC로 전송하는 동작을 한다. 그림 8은 싱크용 어플리케이션의 컴포넌트들 간 연결(wiring)에 대한 연결도를 표현한 것이다. 무선 통신을 위한 RadioCRCPacket, RS232 통신을 위한 UART 컴포넌트로 구성되어 있다.

모니터 어플리케이션은 센서 노드들이 보내는 데이터를 화면에 출력하여 실제로 센싱한 온도와 조도 값을 확인할 수 있도록 하였다. 또한 센싱된 데이터를 기반으로 그래프로 표현하여 사용자에게 데이터 값의 변화가 가시화될 수 있도록 하였다. 실험 환경은 실내에서 노드간에 2-3M 간격으로 거리를 두고 5개의 센서 노드를 배치하여 모니터링 하였고 임의적으로 센서노드에 조도 및 온도 변화를 주어 테스트 하였다. 구현된 센서 네트워크 모니터링 시스템은 그림 9처럼 정상적인 데이터 송수신이 원활하게 이루어짐을 확인할 수 있다.

3.1.3 개발 환경

센서 네트워크 모니터링 시스템을 구현하기 위한

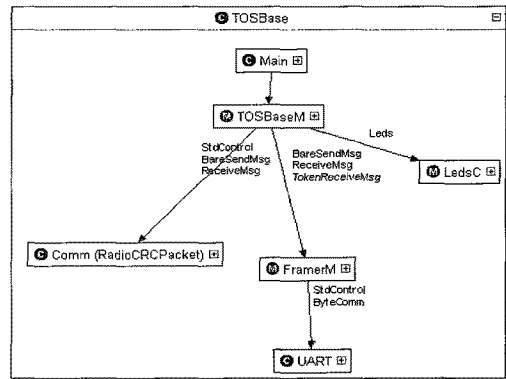


그림 8. 싱크 어플리케이션 컴포넌트 연결도

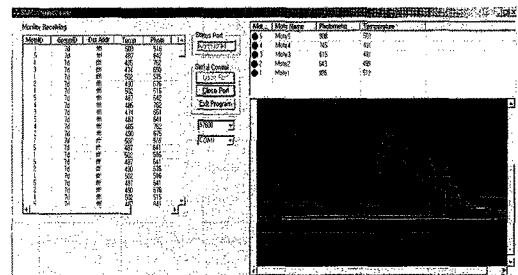


그림 9. 센서 모니터링 어플리케이션

표 3. 모니터링시스템 구현 환경

구분	구성 요소	구성 내용
H/W	센서 플랫폼	Crossbow Micaz Mote
	모니터 컴퓨터	PC
S/W	운영체제	TinyOS 1.1.11 Window XP
	개발 언어	NesC Visual C++
	개발 도구	Eclipse

환경은 표 3과 같다.

실험에서 사용한 센서 노드 및 싱크 노드를 위한 하드웨어 플랫폼은 Crossbow사 Micaz Mote를 사용하였고, 싱크 노드와 연결하여 데이터를 가시화하기 위해서 Windows XP 기반의 PC를 사용하였다. 소프트웨어로는 센서 노드와 싱크 노드용 어플리케이션을 개발하기 위해 TinyOS 1.x 기반의 NesC를 사용하였고, NesC개발 환경은 Eclipse를 이용하였다. 모니터 어플리케이션 개발은 Visual C++로 개발되었다.

4. 성능 평가

무선 센서 네트워크는 센서 노드의 제한된 전송범위로 인해 멀티 홉 통신을 해야 한다. 따라서 효율적인 라우팅 프로토콜은 전반적인 네트워크 성능에 큰 영향을 미친다. 센서 네트워크에서 라우팅 프로토콜을 고려하여 동작하는 센서 노드의 작업 수행에 제한된 에너지를 가진 배터리 사용을 효율적으로 수행하여 에너지를 절약해야 한다.

그래서 NS-2 네트워크 시뮬레이터를 이용하여 시스템 구현에 사용된 REMBEER 라우팅 프로토콜에 대한 밸런싱과 패킷 량에 대해서 실험 및 평가하였다.

REMBEER 라우팅 프로토콜은 지역적으로 에너지 레벨을 고려해서 주변 노드로의 경로를 선택하는 알고리즘으로 데이터를 빠르게 전송하는 것보다 전체 네트워크의 수명에 중점을 두었다. 만약 일부분의 센서 노드들만 에너지를 모두 사용한 경우라도 센서 노드에 의해 센서 네트워크 전체가 동작 못할 수 있으며, 이런 원인으로 네트워크에 심각한 문제가 발생할 수 있다.

그리고 일부 센서 노드의 오동작으로 인해 새로운 센서 노드를 투입하는 경우, 동작되지 않는 노드들의 정확한 위치를 알지 못하면 센서 노드를 임의적으로 배분되어야 하기 때문에 때로는 예상하지 못한 문제가 발생할 수 있다. 그래서 이런 문제를 해결하기 위해서 모든 센서 노드의 에너지가 최대한 균일하게 소비되도록 조절하는 것이 바람직하다.

그림 10은 NS-2의 xgraph를 통하여 전체 노드의 에너지 소모정도에 대한 시뮬레이션 결과이다. X축은 노드 번호이고 Y축은 노드가 소모한 에너지 정도를 나타내며, 특정 노드를 제외하면 전체적으로 노드 밸런싱이 안정된 형태로 나타난다.

그림 11은 최초의 경로 설정과 단 한 번에 보내진 데이터에 대하여 모든 노드들이 사용한 전체 패킷량을 다른 라우팅 프로토콜과 비교하였다. X축은 노드의 총 개수로 네트워크의 크기를 의미하고 Y축은 전체 패킷량의 증가도를 보여준다. 다른 라우팅과의 패킷량 비교에서 directed diffusion은 노드 수가 증가할수록 전체 패킷량이 많이 증가되고, AODV는 패킷량 증가가 directed diffusion보다 현저하게 낮고, 본문에서 사용한 REMBEER은 AODV에 비해 조금 더 우수한 것으로 나타난다.

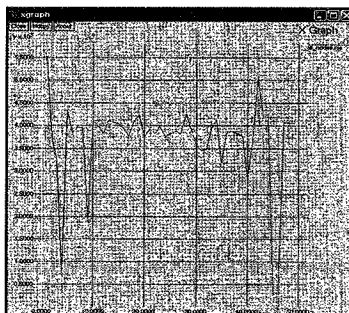


그림 10. 노드 밸런싱

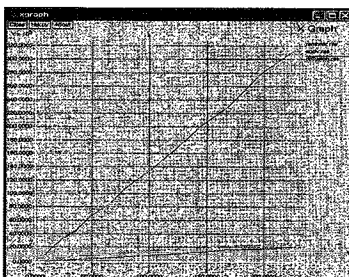


그림 11. 다른 라우팅과의 패킷량 비교

5. 결 론

본 논문에서는 무선 센서 네트워크 환경에서 Micaz 플랫폼의 무선 센서 보드와 TinyOS를 이용하여 관심 영역의 환경 데이터를 센싱하여 수집된 값의 변화를 가시화하여 사용자에게 보여줄 수 있는 시스템을 구현하였다. 또한 실험을 통해 안정적인 데이터의 송·수신이 RF 통신에서 가능함을 확인하였다.

제안한 시스템은 안정적인 데이터 송·수신 뿐 아니라 센서 노드의 수명 연장을 위해서 센서 네트워크 전체의 에너지 효율성을 고려하였고 NS-2를 통하여 시뮬레이션을 하여 검증하였다.

향후 과제로는 효율적인 에너지를 이용하는 특정 분야에 모니터링 시스템 구현이 필요하다.

참 고 문 헌

- [1] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Comm. ACM*, Vol. 36, No.7, pp. 75-84, 1993.
- [2] D. Puccinelli and M. Haengg, "WSN: Applications & Challenges of Ubiquitous Sensing," *IEEE CAS Magazine*, Vol.5, No.3, pp. 19-29, Sep. 2005.
- [3] M. Tubaishat, S. Madria, "Sensor Networks : An Overview," *IEEE Potentials*, Vol.22, No.2, pp. 20-23, April/May 2003.
- [4] Tarik Arici and Yucel Altunbasak, "Adaptive Sensing for Environment Monitoring Wireless Sensor Networks," *IEEE Communications Society*, Vol.4, pp. 2347-2352, 2004.
- [5] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.Culler and K. Pister, "System architecture directions for networked sensors," *ACM SICPLAN Notices*, Vol.35, No.11, pp. 93-104,

Nov. 2000.

- [6] 이기욱, 성창규 "유비쿼터스 센서 네트워크 기반의 상황 정보 모니터일 시스템 구현," 한국컴퓨터정보학회 제11권 제5호, pp. 259-265, 2006.
- [7] <http://www.xbow.com>
- [8] <http://www.tinyos.net>
- [9] Qiangfeng Jiang and D. Manivannan, "Routing Protocols for Sensor Networks," *Proceedings of IEEE Consumer Communications and Networking Conference*, pp. 93-98, 2004.
- [10] Kemal Akkaya and Mohamed Younis, "A survey on routing protocol for wireless sensor networks," *Ad Hoc Networks*, Vol.3, No.3, pp. 325-349, 2005.



이 기 욱

1985년 계명대학교 전자계산학과 (공학사)
 1987년 동국대학교대학원 전자계산학과(공학석사)
 2001년 계명대학교대학원 컴퓨터공학과(공학박사)
 1991년~2006년2월 동명대학 컴퓨터정보처리과 교수

2006년 3월~현재 동명대학교 향만물류학부 부교수
 관심분야 : 유비쿼터스 컴퓨팅, 지능DB



성 창 규

1992년 경성대학교 전산통계 (이학사)
 1998년 경성대학교대학원 컴퓨터과학과(이학석사)
 2002년 한국해양대학교 컴퓨터공학과 박사과정 수료

관심분야 : 유비쿼터스 컴퓨팅, 무선 센서네트워크