

구글맵을 이용한 위치 추적 서비스를 제공하는 6LoWPAN 테스트베드 구현

Implementation of 6LoWPAN Testbed: Location Tracking Service Based on Google Map

김 계 원*
Kyewon Kim

서 재 완**
Jaewan Seo

황 대 준***
Dae-joon Hwang

추 현 승****
Hyunseung Choo

요 약

유비쿼터스 사회에서 u-서비스를 구현하기 위한 핵심기술 중의 하나인 무선센서네트워크는 대규모망에서 관리가 어렵고 안정성 및 이동성이 취약하다는 단점을 가진다. 이러한 문제점을 해결하기 위하여 최근 센서네트워크와 IP망과의 연동을 위한 6LoWPAN에 대한 연구가 활발히 진행 중에 있다. 본 논문에서는 6LoWPAN을 이용한 위치 추적 시스템 LTSGM(Location Tracking Service Based on Google Map)을 제안한다. LTSGM 시스템은 IP 인프라 서비스인 구글맵과 연동하여 센서노드의 위치를 시각적으로 제공함으로써 대규모 센서네트워크에서의 유지, 보수, 관리를 보다 용이하게 한다. 또한 모바일노드의 위치를 추적할 수 있으므로 향후 각종 재난, 범죄 등의 응용서비스에 활용될 수 있을 것으로 기대한다. 본 논문에서 구현한 LTSGM 시스템은 향후 6LoWPAN연구를 위한 시험적인 플랫폼이 될 수 있을 것이다.

ABSTRACT

Wireless Sensor Networks (WSNs) is one of the key technologies that make it possible to provide u-service in a ubiquitous society, however, it has disadvantages such as difficulty of managing sensor nodes and weaknesses of stability or mobility in large-scale WSNs. In order to solve these problems, 6LoWPAN that integrates with WSNs and IP networks has been studied widely. In this paper, we propose a Location Tracking Service Based on Google Map (LTSGM) system using 6LoWPAN. Since LTSGM system provides visual location information of sensor nodes through Google Map, it is possible to makes it easier than ever to manage sensor nodes in large-scale WSNs. Moreover, LTSGM can be used for various services such as applications for disaster or crime because it can trace the location of mobile nodes. Implementation of LTSGM system will be a test platform for 6LoWPAN.

☞ KeyWords : 6LoWPAN, Wireless Sensor Networks, Testbed, TCP/IP, Location Tracking

1. 서 론

무선센서네트워크는 앞으로의 유비쿼터스 사회에서 통신 인프라를 선도적으로 구축하여 각종

u-서비스를 구현하기 위한 핵심기술 중의 하나이다[1][2]. 하지만 기존의 무선센서네트워크는 소규모 망에 적합하도록 설계되었기 때문에 대규모 망에서 관리가 어렵고 다양한 서비스와 접목 시 추가적인 비용 발생과 안정적인 시스템 구축에 어려움이 따른다. 이러한 문제점이 대두됨에 따라 최근 센서네트워크와 IP망의 연동을 다루는 6LoWPAN이 최적의 솔루션으로 각광받고 있으며, 활발한 연구가 이루어지고 있다[3]. IETF의 6LoWPAN Working Group에서 기술 표준을 제정하고 있는 6LoWPAN은 구체적으로 IPv6을 IEEE

* 준 회 원 : 삼성테크윈 SIS사업부 선임연구원
sound79@skku.edu

** 준 회 원 : 성균관대학교 일반대학원 휴대폰학과(석사)
todoll2@skku.edu

*** 정 회 원 : 성균관대학교 정보통신공학부 교수
djhwang@skku.edu

**** 중신회원 : 성균관대학교 정보통신공학부 부교수
choo@ece.skku.ac.kr

[2008/12/10 투고 - 2008/12/13 심사 - 2009/03/04 심사완료]

802.15.4 기반 저속 WPAN에 적용하는 방안과 센서네트워크를 구축 후 관리의 유연성과 모바일노드의 이동성 지원 서비스에 관한 연구를 진행 중이다[4].

또한 최근에 주목받고 있는 센서네트워크를 이용한 위치추적 기술은 핸드폰과 같은 모바일노드를 소지한 사용자의 위치를 실시간으로 파악하여 사용자의 위치에 적합한 정보와 서비스를 제공하거나 혹은 키즈케어(Kids-care)와 같은 위치기반 응용서비스에 적합한 기술로 대두되고 있다. 이러한 위치추적 기술에 기반한 응용서비스를 제공하기 위한 방안으로서 센서네트워크와 인터넷과 같은 기존 인프라와의 연동이 필수적인 요건으로 부각되고 있다.

본 논문에서는 센서노드에 6LoWPAN을 탑재하여 IP기반 응용서비스인 구글맵을 이용한 LTSGM(Location Tracking Service Based on Google Map) 시스템 구조를 제안한다. LTSGM은 6LoWPAN이 탑재된 센서노드간의 무선통신과 이를 수집하는 서버와 데이터베이스로 구성된다. 서비스를 제공받는 사용자는 웹브라우저가 탑재된 단말기를 통해서 본 시스템에 접근하여 지역에 기반한 센서노드의 측정값과 모바일노드의 위치 추적 서비스를 제공받을 수 있다. 이는 모두 구글맵을 통해서 시각적으로 제공됨으로써 사용자가 보다 이해하기 쉽고 용이하게 데이터를 확인 가능할 것으로 기대한다.

본 논문은 다음과 같이 구성된다. 2절은 무선센서네트워크의 플랫폼 및 표준들을 살펴보고 이를 응용한 테스트베드의 연구사례를 살펴본다. 제안하는 서비스 시나리오는 3절에서 다룬다. 4절에서는 본 논문에서 제안하는 LTSGM시스템 구조를 소개하며 5절에서는 제안한 시스템의 구현방법 및 동작 예시를 보인다. 마지막으로 6절에서는 본 논문의 결론과 향후 과제를 설명한다.

2. 관련 연구

2.1 무선센서네트워크 플랫폼 분석

무선센서네트워크의 응용서비스로는 환경 모니터링, 홈 오토메이션, 빌딩 자동화, 군사적 감시 정찰, 헬스케어, 지능형교통시스템(ITS), 보안, 재고 관리 등이 있다. 이러한 응용서비스를 구현하기 위해서는 센서노드간에 TinyOS와 같은 센서노드용 OS를 사용하거나 ZigBee나 6LoWPAN과 같은 표준 프로토콜이 구현된 스택을 사용하고 있다. TinyOS는 UC 버클리 대학에서 개발, 유지·보수되고 있는 센서노드용 운영체제이다. Event-Driven 방식으로 적은 메모리 사이즈에서도 동작이 가능하도록 구현되었으며 Component 기반의 설계가 가능한 구조이다. 하지만 TinyOS를 이용하지 않는 센서노드와의 통신을 지원하지 않으며 IEEE 표준스펙을 따르지 않는다. 근래에는 이러한 문제점을 보완하기 위해서 ZigBee와 6LoWPAN과 같은 표준 프로토콜도 지원하는 방향으로 나아가고 있는 중이다[5].

ZigBee 프로토콜은 IEEE 802.15.4를 바탕으로 ZigBee Alliance에서 상위 네트워크, 보안, 어플리케이션의 표준을 제정하고 있다[6]. 현재 ZigBee-Pro 2007까지 표준 프로토콜이 제정되었으며, 벤더에게 공개되어 있는 상태이다. ZigBee의 특징은 원격 감지, 감시, 제어, 모니터링의 응용과 가격을 최대한 낮추고 전력소모를 최소화하는 방향으로 설계되었다. 하지만 IP를 고려하지 않은 설계로 인해 IETF에서는 6LoWPAN이라는 프로토콜을 제정하였다.

표 1은 ZigBee와 6LoWPAN의 차이점을 보여주고 있다. 6LoWPAN은 IPv6를 기반으로 하기 때문에 네트워크 계층, 트랜스포트 계층, 어플리케이션 계층을 단대단(End-to-End)으로 연결할 수 있다[7][8]. 따라서 게이트웨이에서 별도의 라우팅이 필요하지 않으며, ZigBee에 비해 데이터 전송 지연이 적다는 장점을 가진다. IP를 사용함으로써 IP망과 같은 기존 인프라를 그대로 활용할 수 있기

때문에 비용절감의 효과가 있을 뿐만 아니라 신뢰성과 안정성을 제공할 수 있다. 또한 IPv6 주소 체계의 사용으로 대규모의 센서네트워크 구축이 용이하며, 향후 All-IP를 지향하는 유비쿼터스 세계의 핵심기술로 주목받고 있다.

맵을 통해서 시각적으로 제공됨으로써 사용자가 보다 이해하기 쉽고 용이하게 데이터를 확인할 수 있을 것으로 기대한다.

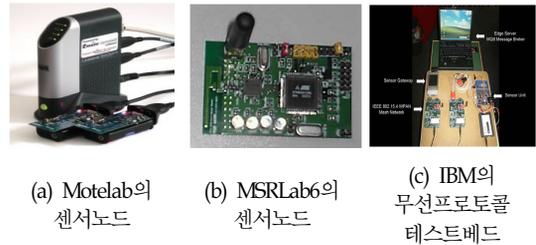
(표 1) ZigBee와 6LoWPAN의 비교

	ZigBee	6LoWPAN
PHY/MAC	IEEE 802.15.4	IEEE 802.15.4
프로토콜 스택 범위	전체	Adaptation 계층
네트워크 주소 및 범위	16비트로써 하나의 PAN내에서만 유일	128비트 IPv6 주소이며 전세계적으로 유일
게이트웨이 기능 범위	응용 계층	네트워크 계층
서비스 발견	로컬	글로벌
통합망 관리	어려움	간단함(SNMPv3)

또한 최근에 주목받고 있는 센서네트워크를 이용한 위치추적 기술은 핸드폰과 같은 모바일노드를 소지한 사용자의 위치를 실시간으로 파악하여 사용자의 위치에 적합한 정보와 서비스를 제공하거나 혹은 키즈케어(Kids-care)와 같은 위치기반 응용서비스에 적합한 기술로 대두되고 있다. 이러한 위치추적 기술에 기반한 응용서비스를 제공하기 위한 방안으로서 센서네트워크와 인터넷과 같은 기존 인프라와의 연동이 필수적인 요건으로 부각되고 있다.

본 논문에서는 센서노드에 6LoWPAN을 탑재하여 IP기반 응용서비스인 구글맵을 이용한 LTSGM(Location Tracking Service Based on Google Map) 시스템 구조를 제안한다. LTSGM은 6LoWPAN이 탑재된 센서노드간의 무선통신과 이를 수집하는 서버와 데이터베이스로 구성된다. 서비스를 제공받는 사용자는 웹브라우저가 탑재된 단말기를 통해서 본 시스템에 접근하여 지역에 기반한 센서노드의 측정값과 모바일노드의 위치 추적 서비스를 제공받을 수 있다. 이는 모두 구글

2.2 테스트베드 구현 사례



(그림 1) 센서네트워크 테스트베드 사례

무선센서네트워크를 위한 플랫폼 및 표준 프로토콜의 응용을 위해서 테스트베드를 구축하고 이를 응용한 연구가 활발하게 수행되고 있다. 먼저 TMote Sky의 H/W에 TinyOS를 응용한 테스트베드로는 그림 1 (a)의 Motelab이 있다. Motelab은 대규모 센서노드를 지원하기 위해서 웹기반으로 설치된 센서노드를 모두 관리할 수 있는 시스템이다. 인터넷을 매개로 하여 각 센서노드를 제어, 수정 및 모니터링이 가능하므로 원거리에 있는 사용자도 직접 센서노드를 관리할 수 있도록 구현되었다. 또한 MoteTrack, CodeBlue로 명명된 프로젝트에서는 위치추적과 이동성을 가진 기기간의 Ad-hoc 인프라에 응용 되었다[9].

Hue, Hongwei 등이 구현한 IPv6기반의 테스트베드에 이용된 MSRLab6의 센서노드(MSR6680)는 그림 1 (b)와 같다. 센서노드에 IPv6을 응용하기 위한 테스트베드로써 센서노드의 H/W는 ATmega128L과 CC2420으로 구성되어 있다. 저용량의 센서노드에 IPv6을 지원하기 위해서 Tiny IPv6 프로토콜 스택을 개발하였다. Tiny IPv6는 코드사이즈를 줄이기 위해서 uIP와 같은 방법을 사용하여 8비트 MCU에서도 동작이 가능하도록 하였다. IPv6의 패킷을 IEEE 802.15.4의 프레임에 캡슐화하여 센서노드들간에 데이터를 주고 받으며,

자체 라우팅 프로토콜을 설계하여 센서네트워크에 응용한 시스템이다[10].

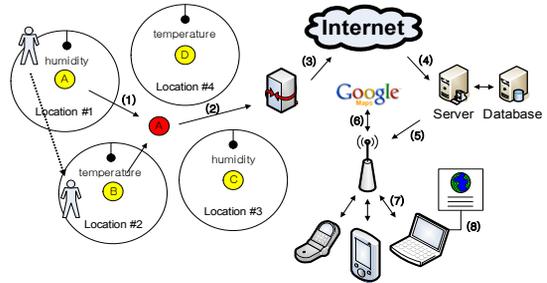
그림 1 (c)는 IBM의 Zurich 연구센터에서 구현한 무선센서네트워크 테스트베드이다. 근거리 무선통신의 송수신 프로토콜로써 많이 사용되고 있는 IEEE 802.15.4/ZigBee, Bluetooth, IEEE 802.11b을 실험하기 위한 테스트베드로써 종단 센서노드와 사용자의 응용프로그램간의 통신에 초점을 맞춘 사례이다. 센서노드, 게이트웨이, 미들웨어, 응용프로그램의 시스템 구조를 가지고 있으며 MQTT (MQ telemetry transport) 프로토콜 메시지 모델을 제안하였다. MQTT 메시지 모델은 분산된 환경의 응용프로그램 메시지 처리에 관한 프로토콜로써 원격의 다량의 데이터를 수집하여 이를 응용하기 위한 서비스에 적합하도록 설계되었다[11].

이상의 구현사례를 살펴보면 단순한 제어목적의 테스트베드가 아닌 센서노드와 인터넷 혹은 IP와의 연동을 고려하기 위한 방향으로 연구가 진행되고 있음을 알 수 있다. 향후 센서네트워크를 응용한 서비스를 제공할 경우에도 기존 IP 기반의 널리 사용되고 있는 서비스와 융합화가 이루어질 것으로 예상되며, 이는 실제 대규모 센서네트워크 인프라와 인터넷 환경 그리고 다양한 서비스 시나리오에 기반한 시스템으로 사용자에게 다가올 것이다.

3. 서비스 시나리오

유비쿼터스 시대에서 무선센서네트워크는 단순히 정보를 전송하는데 그치지 않고 개개인의 요구 또는 위치에 따른 차별화된 서비스를 제공해야 한다. 특히 대규모의 센서네트워크를 구축할 때 센서노드의 위치를 제공하는 것은 네트워크 유지, 보수, 관리 측면에서 매우 중요하다. 또한 센서네트워크에 속하는 모바일노드의 위치를 판별함으로써 재난 혹은 불상사에 신속하게 대처할 수 있는 다양한 응용서비스 제공이 가능하다. 따라서 본 논문에서는 6LoWPAN을 이용하여 대규

모 센서네트워크를 관리하고, 다양한 응용서비스를 지원하기 위한 맵기반 위치 추적 시스템을 구축한다.

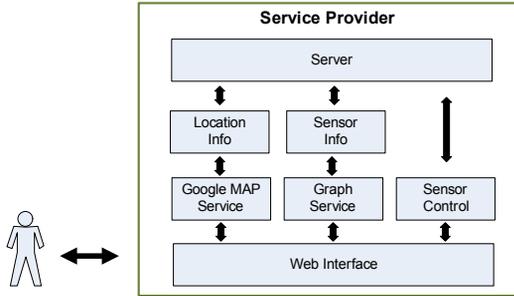


(그림 2) 서비스 시나리오

그림 2는 제안하는 서비스 시나리오의 전체적인 모습이다. 시스템은 1차적으로 지역에 기반한 환경모니터링의 역할을 수행한다. 각 지역의 센서노드들은 싱크노드를 중심으로 무선네트워크를 형성하고 있다. 센서노드가 주기적으로 측정하는 정보는 센서네트워크와 인터넷을 거쳐서 서버로 전송되고 데이터베이스에 저장된다. 사용자는 관심있는 지역의 환경 정보를 획득하기 위하여 Wi-Fi가 탑재된 노트북이나, 인터넷이 가능한 휴대폰이나 PDA를 이용하여 시스템에 접속할 수 있다. 접속된 사용자에게 제공되는 정보는 실시간으로 측정되고 있는 센싱정보와 센서노드의 위치를 인터넷 맵의 형태로 보여준다. 이로써 사용자는 센서노드가 측정한 정보뿐만 아니라 센서노드의 위치까지 친숙한 맵과 그래프의 형태로 제공받음으로써 보다 쉽고 빠르게 센서의 정보를 인지할 수 있다.

시스템은 2차적으로 위치 추적 서비스를 제공한다. 모바일노드를 소지하고 있는 사용자 혹은 개체의 움직임이 센서노드에 감지되었을 경우 센서노드는 이를 인식하여 모바일노드의 정보를 서버로 전달한다. 센서네트워크를 모니터링하고 있는 사용자는 모바일노드의 위치가 감지되면 시스템에 모바일노드의 위치를 추적할 수 있도록 요청할 수 있다. 사용자로부터 위치 추적 요청이 시

시스템에 전달되면 시스템은 모바일노드 위치의 변경이 발생할 때 마다 위치 정보를 갱신하고 사용자에게는 위치 추적맵을 통해서 실시간으로 모바일노드의 동선을 제공해 준다.

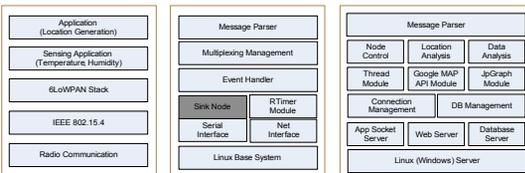


(그림 3) 테스트베드에서 제공하는 서비스 구성도

그림 3과 같이 서비스를 제공받는 표준 웹 인터페이스를 통해서 서비스 제공자와 상호작용을 한다. 제공되는 정보는 모바일노드의 위치 정보와 센서가 측정한 데이터이다. 이는 실좌표에 기반한 구글맵의 특정위치 형태와 함께 제공된다. 또한 데이터는 텍스트 기반이 아닌 그래프 형태로 표현됨으로써 사용자는 현재 센서네트워크의 상황 인지를 보다 효과적으로 수행할 수 있다. 또한 구글맵과 같은 오픈 API를 적용함으로써 향후 오픈 API를 제공하는 다양한 서비스와 접목·확장이 가능하다.

4. 시스템 개요

4.1 시스템 구조 및 모듈

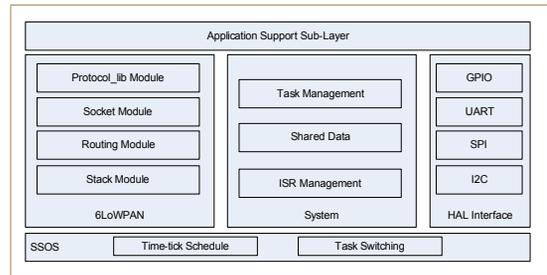


(a) 센서노드 (b) 게이트웨이 (c) 서버

(그림 4) LTSGM 시스템 구조

본 논문에서 제안하는 LTSGM(Location Tracking Service Based on Google Map) 시스템의 구조는 그림 4와 같다. 이는 센서노드, 게이트웨이, 서버로 구성되며, 구글에서 제공하는 구글맵 오픈 API와 연동되는 형태이다. 센서노드의 각 모듈은 위치 생성과 센싱에 관련된 어플리케이션 모듈, 6LoWPAN 통신 스택을 구현하기 모듈, Radio Communication인 CC2420과의 PHY과 MAC의 구현을 위한 IEEE 802.15.4 모듈로 구성된다 [12]. 게이트웨이는 리눅스 커널을 기반으로 하며 싱크노드와의 시리얼 통신연결 인터페이스, 네트워크 인터페이스, 타이머, 이벤트 핸들러, 메시지 파서 모듈로 구성된다.

서버는 32비트 프로세스 운영체제에서 모두 구동이 가능하며, 기능상으로 소켓 서버, 웹 서버, 데이터베이스 서버로 분류된다. 각 서버에서 제공되는 기능은 센서노드로부터 수집되는 데이터의 분석 및 저장, 모바일노드의 위치 분석, 센서노드 컨트롤이다. 상기 기능을 위해서 서버는 쓰레드모듈, 구글맵 오픈 API 모듈, 그래프 모듈로 구성되며, 각 서버와의 연결성을 위해서 연결 관리 인터페이스가 제공된다.



(그림 5) 센서노드 스택 구조

센서노드의 스택 구조는 그림 5와 같다. 센서노드의 개별적인 기능을 수행하는 모든 모듈 및 단위 개체들은 SSOS(Simple Sensor Operating System)를 기반으로 구동된다. 본 테스트베드를 위해 개발한 SSOS는 Single Stack기반의 8비트 마이크로 컨트롤러에서 동작한다. 타이머 기반의 태

스크 스케줄링(Task Scheduling)을 제공하며, 매우 작은 코드크기를 가지고 있어서 저용량의 센서노드에 적합하도록 설계되었다. SSOS를 근간으로 하는 상위 계층은 6LoWPAN, 시스템, HAL 인터페이스로 구성된다.

6LoWPAN은 스택 모듈을 기반으로 프로토콜, 소켓, 라우팅 모듈로 이루어져 있으며, 각 모듈은 레이어 구조를 가짐으로써 레이어간의 통신을 가능하게 한다. 시스템은 태스크 관리, 공유 데이터, ISR(Interrupt Service Routine)의 관리를 하는 개체로 구성된다. 이는 SSOS의 시스템 요소로서 역할을 수행한다. 다음으로 HAL 인터페이스는 센서노드의 하드웨어적인 정합을 위한 GPIO, UART, SPI, I2C 등이 있다. SPI는 CC2420 RF-IC와의 인터페이스이며, I2C는 센서노드의 온·습도 센서인 SHT11과의 하드웨어 인터페이스이다. 어플리케이션 서포트 서브레이어(Application Support Sub-Layer)는 상위 어플리케이션의 구동을 위한 파라메타 설정 및 다중 어플리케이션을 구동하기 위한 서브 레이어의 역할을 담당한다.

4.2 시스템 동작 과정

LTSGM의 센싱 정보를 사용자에게 제공하는 과정은 다음과 같다. 센서노드는 미리 설정된 타임아웃 시간마다 주변의 온도, 습도값을 측정한다. 측정된 데이터는 헤더 압축과 같은 과정을 통해 6LoWPAN의 패킷 포맷으로 변환되어 전송을 요청한다. 이는 Radio Communication의 CC2420의 칩과 IEEE 802.15.4의 프레임 구조를 통해서 게이트웨이의 싱크노드에 전달된다. 싱크노드는 시리얼통신을 이용하여 수신한 패킷을 게이트웨이에 전달하며, 게이트웨이는 이벤트 핸들러를 통해서 패킷을 수신했음을 인식한다. 이벤트 핸들러는 게이트웨이의 멀티플렉싱 구조를 통해서 싱크노드와의 시리얼 통신과 네트워크 인터페이스를 통한 서버와의 통신을 동시에 수행 가능하도록 구조화되어 있다.

게이트웨이는 데이터의 생성시점을 기록하기

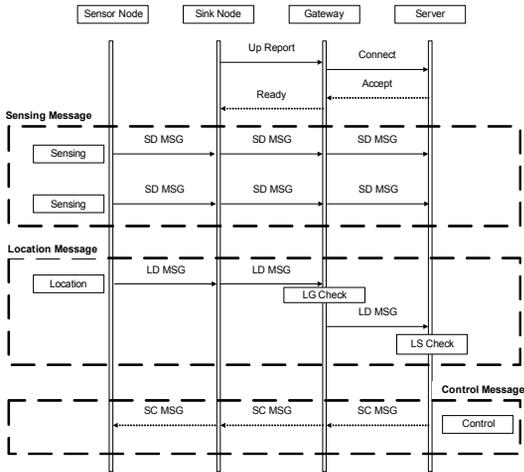
위해서 프레임의 시간 필드를 현재 시간으로 설정하고 데이터를 서버로 전송한다. 소켓 서버는 게이트웨이로부터 수신한 데이터를 필터링과 분석 과정을 거쳐 데이터베이스에 저장한다. 이후 사용자는 웹 서버로 해당 데이터를 분석 요청을 하게 되면 구글맵 API 모듈은 데이터베이스에 저장된 위치정보를 바탕으로 사용자에게 구글맵의 센서노드 위치를 표시하고 JpGraph 모듈은 저장된 데이터를 그래프형태로 가공하여 사용자에게 제공해 주는 역할을 한다.

모바일노드의 위치추적을 위한 처리과정은 다음과 같다. 센서노드로부터 모바일노드의 신호가 감지되면 센서노드는 Location Generation 모듈을 통해 모바일노드의 현위치 정보를 생성하고 이를 싱크노드로 전송한다. 싱크노드는 RSSI(Received Signal Strength Indicator)에 기반하여 특정 임계치 이하의 데이터는 필터링을 수행하며 임계치 이상의 데이터는 게이트웨이를 통해서 서버로 전달한다. 위의 과정을 통해 서버는 센서노드로부터 수신한 모바일노드의 위치정보를 분석 및 가공하여 데이터베이스에 저장하고 이후 사용자의 위치추적 요청에 해당 데이터를 제공할 수 있다. 위치 추적 또한 실제 좌표에 해당하는 구글맵의 위치 정보 형태로 제공한다.

5. 구 현

5.1 메시지 흐름도

본 테스트베드의 메시지 흐름도는 그림 6과 같으며 시나리오를 구현하기 위한 메시지의 처리과정은 Sensing Message, Location Message, Control Message의 3가지로 구분된다. Sensing Message는 종단 센서노드에서 측정된 센싱값을 주기적으로 서버까지 SD(Sensor Data) MSG 포맷으로 송신하는 과정이다. MSG 포맷은 헤더와 데이터로 구성되며 송, 수신자는 헤더를 기준으로 MSG를 구분하고 처리를 수행한다.



(그림 6) 테스트베드의 메시지 흐름도

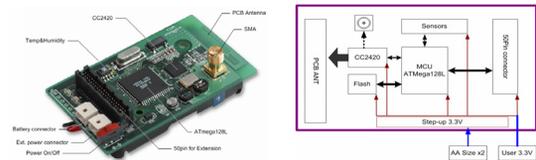
Location Message는 모바일노드의 위치 추적을 위한 과정이며 센서노드에서 모바일노드를 인식하면 LD(Location Detection) MSG를 생성하여 싱크노드에게 전송한다. LD MSG의 데이터에는 모바일노드의 고유아이디와 센서노드의 어드레스 정보를 포함한다. LD MSG를 수신한 싱크노드는 게이트웨이로 이를 포워딩하고 게이트웨이는 LGD Proc(Location Gateway Decision Procedure)의 과정을 수행한다. LGD Proc의 과정은 수신한 LD MSG의 모바일노드 고유아이디와 센서노드의 어드레스를 이전 데이터와 비교·분석하여 모바일노드의 이동 혹은 현재 위치에 대기를 판단한다. 판단 알고리즘은 이전 같은 모바일노드를 인식한 센서노드의 어드레스와 새롭게 인식한 센서노드를 비교하여 같을 경우, 이동이 아니라 현재의 센서노드 위치에 모바일노드가 대기하고 있다고 판단한다.

센서노드의 어드레스가 다를 경우 이는 기존 센서노드의 위치에서 현재의 센서노드의 위치로 이동했음을 의미한다. 모바일노드의 이동을 판단한 게이트웨이는 LD MSG 데이터 필드의 이동 플래그를 설정한 후 서버로 전송한다. 이를 수신한 서버는 LSD Proc(Location Server Decision Procedure)를 수행한다. LSD Proc의 과정은 게이

트웨이의 이동을 판단한다. 지역적인 그룹으로 설정이 되었을 경우 이는 특정 그룹을 이탈 혹은 진입했음을 설정할 수 있다.

서버는 Location Analysis 모듈을 이용하여 LD MSG를 분석하여 데이터베이스의 Location 테이블에 모바일노드의 현위치 정보를 업데이트한다. 세번째로 Control Message는 서버의 웹 인터페이스를 통한 사용자의 컨트롤 메시지를 중단 센서노드까지 전달하는 역할을 수행한다. 메시지의 형태는 SC(Sensor Control) MSG 형태로써 처리된다.

5.2 센서노드 구현



(그림 7) 센서노드의 H/W 모습 및 내부 모듈 구성도

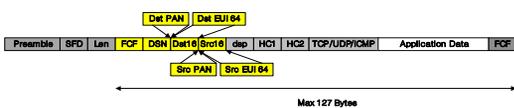
무선센서네트워크를 구성하는 센서노드는 기본적으로 데이터를 수집하고 수집된 데이터를 처리한 후 무선통신을 이용하여 싱크노드에 전송하는 역할을 한다. 본 테스트베드에서 사용된 센서노드의 하드웨어는 한백전자의 ZigBeX을 이용하였다[13]. ZigBeX은 ATmel사의 ATmega128L과 TI사의 CC2420 RF-IC를 적용한 모듈이며, 기능 구성 및 블록도는 그림 7과 같다. ATmega128L은 8Bit 마이크로 컨트롤러로서 현재 저속 무선통신 솔루션의 2-Chip 기반으로는 TI사의 MSP430과 더불어 가장 많이 사용되는 칩중 하나이다. 128KByte 내부 프로그램 메모리와 4KByte 내부 SRAM으로 구성되어 있으며 중요한 정보를 위한 부가적인 비휘발성 메모리인 EEPROM(Electrically Erasable Programmable Read-Only Memory)을 가지고 있다[14].

CC2420 RF-IC는 IEEE 802.15.4와 호환되며 2.4GHz ISM 밴드를 사용하는 싱글 트랜시버(Transiver)이다[15]. 센서는 Sensirion사의 SMT 타

입의 SHT11이 적용되어 있으며 온도와 습도를 동시에 측정할 수 있는 Multi-Sensor Device로써 ATmega128L과 I2C 통신을 이용하여 현재의 측정값을 디지털 형태로 출력한다[16]. ZigBeX는 TinyOS를 기본적으로 지원하는 플랫폼을 가지고 있지만 TinyOS를 이용하지 않는 센서노드와의 연동 및 확장성 문제와 표준 프로토콜 지원 등의 단점이 있다.

본 테스트베드에서는 TinyOS를 사용하지 않고 AVR에서 구동되는 가벼운 센서용 OS인 SSOS를 설계하여 센서노드에 적용하였다. SSOS 태스크 스위칭은 비선점형 기반이며 우선순위가 없는 순차적인 태스크 스케줄링과 일정한 시간간격을 미리 시스템에 설정하여 태스크들이 수행되는 구조를 가지고 있다. 구현은 ATmega128L의 타이머1 오버플로우 이벤트를 설정하여 타이머 Tick이 설정된 카운팅 숫자를 오버하면 타이머의 인터럽트가 발생한다. 이는 설정된 태스크의 스위칭 시간과 비교 대상이 되고 일정한 시간이 경과하면 태스크간의 스위칭이 이루어지는 간단한 형태이다. 센서노드에서 수행되는 태스크는 6LoWPAN 스택 태스크와 어플리케이션 태스크가 구동된다.

6LoWPAN 스택 태스크에서는 그림 8와 같은 6LoWPAN의 메시지 포맷을 바탕으로 데이터를 센서노드사이에 주고 받을 수 있는 구조를 가지고 있다. IEEE 802.15.4의 프레임 헤더 구조인 FCF, DSN, PAN Address, Source와 Destination Address의 필드는 최소 9Bytes에서 최대 23Bytes를 가진다. 6LoWPAN의 포맷은 메시지 구분자인 dsp와 헤더 압축을 위한 HC1, HC2를 구성하고 이후 트랜스포트 레이어의 TCP/UDP/ICMP의 구분자를 설정한다. 어플리케이션의 데이터 구조는 본 테스트베드에서 구동이 되는 센서측정 데이터와 위치 추적 데이터가 포함된다.



(그림 8) 6LoWPAN 프로토콜

본 테스트베드의 6LoWPAN 스택 태스크는 이러한 메시지의 포맷을 따르며 어플리케이션 태스크의 경우에는 두가지의 역할을 수행하도록 구현되었다. 주기적으로 SHT11 센서의 온도, 습도 측정과 RSSI를 기반으로 하는 위치 추적이다. SHT11는 I2C통신을 수행하므로 ATmega128L에서 측정 커맨드를 전송한 후 일정 시간의 지연이 발생하며 온도, 습도값을 얻을 수 있다. Location Tracking을 구현하기 위해서 CC2420의 RSSI을 이용하였다. CC2420의 RSSI 값은 디지털 형태로 RSSI.RSSI_VAL 레지스터로부터 가지고 올 수 있으며 다음과 같은 등식으로 표현된다.

$$RSSI = RSSI.RSSI_VAL + RSSI_OFFSET (dBm)$$

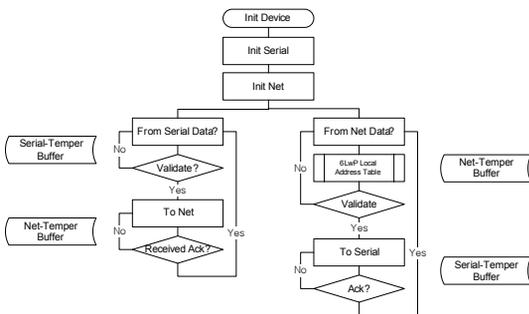
RSSI_OFFSET은 대략 -45로 설정이 된다. 본 논문에서는 미리 Threshold(임계치)값을 설정하여 센서노드로부터 임의의 패킷을 수신한 RSSI의 값이 Threshold값을 상회하면 이는 특정한 거리까지 모바일노드가 접근한 것을 의미한다. 또한 Threshold값 이하로 내려가면 이는 모바일노드가 그 지역의 센서노드로부터 이격되었음을 의미하고 센서노드는 싱크노드로 메시지를 전송한다 [17].

5.3 게이트웨이 구현

리눅스 커널기반의 게이트웨이는 커널소스가 오픈되어 있고 커널레벨의 수정 및 확장이 가능하므로 향후 게이트웨이의 네트워크단에서 메시지 포워딩을 수행할 수 있는 기능을 추가할 수 있다. 본 테스트베드에서 구현된 게이트웨이는 6LoWPAN과 인터넷망과의 연동을 위해서 어플리케이션에서 메시지 포워딩 작업을 수행하였다. IETF의 6LoWPAN에서 권고하는 게이트웨이의 옵션은 Data Link 계층에서 6LoWPAN의 Adaption을 수행하도록 하고 있다. 하지만 이는 실제 OS를 지원하는 시스템의 경우 추가적인 커널레벨의 작업과 표준화의 정확한 규약이 정해지지 않았다. 그

리므로 본 테스트베드에서는 어플리케이션에서 시리얼로 들어오는 6LoWPAN 데이터와 인터넷을 통하는 데이터의 메시지 변환 작업을 수행하여 작업을 하도록 한다.

게이트웨이의 프로그램 흐름도는 그림 9와 같다. 게이트웨이는 싱크노드와는 시리얼 통신을 수행하고 서버와는 소켓 통신을 수행한다. 게이트웨이의 초기화 과정은 시리얼과 소켓의 처리로 이루어진다. 먼저 시리얼의 초기화 과정은 게이트웨이의 COM포트 혹은 MODEMDEVICE를 오픈하고 입력과 출력에 관련된 시리얼 파라메타를 설정한다. 시리얼 초기화 과정이 끝난 후 소켓 초기화 과정을 수행한다. 소켓 초기화 과정은 서버의 주소와 포트를 설정하고 연결 요청을 서버에 보낸다. 이상의 초기화 과정이 완료되면 게이트웨이는 서버에 ALARM 메시지를 전송하여 서버에게 게이트웨이의 초기화 과정이 완료되었음을 알린다.



(그림 9) 게이트웨이 S/W 순서도

게이트웨이는 select()와 파일 디스크립터를 이용한 멀티플렉싱 기법을 사용하여 시리얼과 소켓의 이중화 작업을 동시에 수행하도록 구현되었다. 멀티플렉싱은 디스크립터 설정, 검사범위 설정, 타임 아웃 설정이 되어야 한다. 다중의 입력과 출력을 지원하기 위해서는 시리얼과 소켓의 디스크립터를 fd_set 구조체의 파일 디스크립터에 설정한다. 이후 파일 디스크립터의 변화가 생기면 어떤 디스크립터의 변화가 생긴 지를 확인한 후 해당 루틴을 처리한다.

먼저 시리얼로 발생한 이벤트일 경우 이는 센서 네트워크로부터 발생한 메시지이므로 메시지의 포맷을 검사한 후 서버로 전송한다. 서버에서 ACK 메시지를 받을 때까지 임의의 타임 아웃 동안 기다린 후 ACK 메시지를 받지 못하면 재전송하는 메커니즘을 수행한다. 두번째의 경우는 파일 디스크립터의 변화가 소켓의 디스크립터일 경우이다. 이는 서버로부터 전송된 메시지임을 나타낸다. 게이트웨이는 싱크노드로 전송하기 전에 어드레스 변환 테이블을 참조하여 Adaptation 레이어 역할을 수행하는 모듈에 의해서 6LoWPAN 프로토콜로 변환되고 시리얼을 통해서 싱크노드로 전송한다.

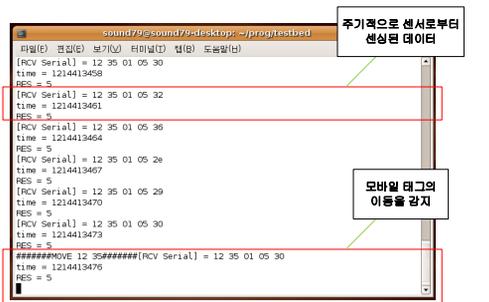
5.4 서버 구현

서버는 동시에 많은 접속요청을 처리하고 센싱된 데이터를 가공, 저장하는 시스템이다. 본 테스트베드의 서버를 기능상으로 분류하면 Socket 서버, Web 서버, Database 서버로 구분 할 수 있다. Socket 서버는 C언어로 프로그래밍되어 있으며 BSD 타입의 Socket System-Call을 사용하여 네트워크 통신을 수행한다. Socket 서버와 게이트웨이와의 연결 모델은 서버-클라이언트 모델 구조를 따르며 다중의 클라이언트, 즉 게이트웨이의 연결 설정을 위해서 Socket 서버에서는 POSIX(Portable Operating System Interface) 기반의 멀티스레드 형태로 구현되었다. 스레드는 fork()나 프로세스보다 적은 오버헤드를 가지며 커널에서 추가적인 메모리 공간 및 파일 기술자 등의 복사본을 새로 생성하지 않으므로 스레드 생성이 프로세스 보다 빠르다는 장점을 가진다.

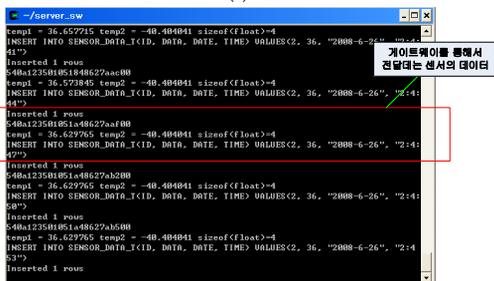
Web 서버를 구현하기 위해서 Apache2와 Database 서버인 MySQL를 사용하였다. Apache2는 HTTP Server로써 사용자의 웹 브라우저를 통해서 서버의 정보를 웹페이지 형태로 제공한다. 또한 기본적인 HTML 구문 분석과 더불어 동적인 페이지를 위해서 서버 측 스크립트 언어인 PHP를 사용하였으며 그래프 표현과 빠른 렌더링을 위해서 PHP Extension인 Zend Optimizer, gd library를

추가하였고, 오픈소스인 JpGraph를 사용했다. JpGraph는 PHP상에서 다양한 그래프를 만들 수 있는 라이브러리를 제공하며 인스톨 과정 없이 서버에 압축을 풀어서 사용이 가능하다[18]. JpGraph를 사용함으로써 향후 센서의 타입에 따른 추가적인 그래프 표현시 소스의 수정 및 업데이트가 비교적 간단하다는 장점을 가질 수 있다.

센서노드의 위치를 시각적으로 표현하기 위해서는 구글에서 제공하는 구글맵 API와의 연동 또한 필수적이다. 이는 Ajax (Asynchronous JavaScript and XML)라는 비동기 통신기반의 대화형 웹페이지의 기술을 이용함으로써 가능하다. Ajax의 특징은 매번 웹서버에서 클라이언트의 요청에 대하여 중복되는 HTML 코드를 다시 전송하면서 생기는 대역폭의 낭비를 없애고 클라이언트에서 데이터의 처리 작업을 수행함으로써 웹서버의 처리량을 줄이는 효과를 가지면서 고속으로 화면을 전환할 수 있다[19].



(a)

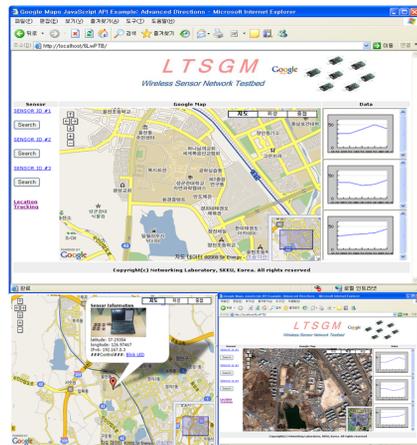


(b)

(그림 10) 게이트웨이와 서버의 Tracing 화면

이상의 서버 구조는 대규모의 센서네트워크를 구축함에 있어 센서노드의 시각화와 기존 기술과

의 연동성 및 대규모 시스템 구축시에 적은 오버헤드에 초점을 맞추어서 구현한 시스템 구조이다. 게이트웨이와 서버의 포트에 들어오는 메시지의 Tracing 화면은 그림 10과 같다. 그림 10 (a)는 게이트웨이의 2가지 역할을 보여주고 있다. 첫째 센서네트워크를 통해서 3초에 한번씩 센싱되는 데이터는 싱크노드에게 전달이 되고 싱크노드와 시리얼 인터페이스로 연결되는 게이트웨이에 5Byte의 데이터가 전달되는 모습을 보여주고 있다. 게이트웨이는 해당 메시지에 현재 시간을 추가한 후 서버에 전달한다. Tracing의 하단 부분은 모바일노드의 이동이 감지된 모습을 보이고 있다. 모바일노드의 이동은 게이트웨이에서 이동 플래그를 셋팅한 후 서버로 해당 메시지를 전달한다. 그림 10 (b)는 서버에서 수신한 메시지의 Tracing 화면이다. 수신한 데이터는 서버에서 필터링과 일련의 과정을 수행한 후 데이터베이스에 저장하기 위해서 SQL 쿼리 구문으로 변환된다. 데이터베이스에 정보를 저장하기 위한 테이블의 구조는 SENSOR_DATA_TABLE 과 SENSOR_MOVE_TABLE로 구분된다. 센서노드로부터 전달된 메시지는 INSERT구문으로 변환되어 데이터베이스에 저장되고 사용자는 이를 SELECT하여 정보를 확인한다. 그런 다음에 사용자는 마지막 데이터를 가지고 모바일노드의 최근 위치를 확인할 수 있다.



(그림 11) 사용자에게 제공되는 LTSGM 화면구조

사용자는 웹 브라우저를 통해서 본 테스트베드 시스템에 접근할 수 있고 메인화면은 그림 11의 위측과 같다. 메인화면의 구성은 좌측에서부터 센서정보, 구글맵, 센서데이터의 3단으로 구성되어 있다. 센서정보는 설치된 센서노드의 고유 아이디를 기본정보로 하여 설치된 센서노드의 개수만큼 표시된다. 사용자가 센서노드의 고유 아이디에 해당하는 Search 버튼을 클릭하면 센서노드에 미리 설정된 좌표에 해당하는 위치가 중앙의 구글맵에 붉은색 마커로 표현된다. 그림 11의 좌측아래는 구글맵에 3개의 센서노드들이 마커로 표현되어 있으며 마커를 더블클릭할 경우 센서노드의 세부 정보 화면이 팝업되는 것을 보여준다. 팝업의 내용은 센서노드의 사진, 위도, 경도, IP 어드레스, 센서 컨트롤 등이 표시된다. 센서노드의 사진은 초기 센서를 등록할 때 주변 환경 등을 미리 촬영하여 저장할 수 있으며 센서컨트롤은 센서노드에 미리 정해진 명령을 수행하도록 한다. 여기서는 LED를 블링크하여 주변 모바일노드에게 위험 신호 및 알람 신호를 알리기 위해 사용한다. 구글맵은 상세정보를 제공해 주는 맵화면과 인공위성 사진을 제공해 주는 2가지 형태이다. 국내에서는 인공위성 사진만 서비스되었지만 2008년 후반부터 맵화면도 제공하고 있다. 메인화면의 우측에는 센서노드가 측정하고 있는 센서데이터를 센서노드의 개수만큼 그래프형태로 나타내고 있다. 그래프는 시간단위에 해당하는 데이터의 값으로 표시된다.

위치 추적을 수행하는 웹인터페이스는 사용자에게 현재 센서네트워크상에서 모바일노드의 위치를 실시간적으로 보여준다. 앞 절의 서비스 시나리오에서 설명한 것처럼 위치추적이 필요한 모바일노드가 감지되었을 경우, 사용자는 본 테스트베드의 위치추적 시스템을 구동하여 해당 모바일노드의 위치를 실시간으로 확인할 수 있다. 모바일노드의 위치가 변동될 때마다 본 테스트베드 시스템에서 사용자에게 보여주는 화면은 그림 12와 같다. 마커로 표시된 센서노드 주위의 붉은색

원은 모바일노드가 현재 해당 지역에 위치하고 있음을 나타낸다. 그림 12의 3개의 화면은 모바일노드가 지역 A-B-C를 이동한 모습을 보여주고 있으며 하단의 표시되는 시간은 모바일노드가 감지된 시간을 나타내고 있다. 이는 측정된 시간에 모바일노드가 이동 혹은 현 위치에서 대기 상태를 보여줄 수 있다.

Location Tracking System



2009-02-03 21:05:19

Location Tracking System



2009-02-03 21:05:50

Location Tracking System



2009-02-03 21:06:25

(그림 12) 사용자에게 제공되는 위치 추적 화면

상기 위치추적의 구현 방법은 다음과 같다. A, B, C의 각 지역에 고유 Cell-ID값이 설정되어 있는 센서노드가 설치되어 있으며, 모바일 노드는 주기적으로 고유 ID 신호를 송신하면서 이동한다. 센서노드가 인식할 수 있는 전파 반경에 모바일 노드가 이동해 오면 센서노드는 해당 지역의 Cell-ID와 모바일 노드의 고유 ID를 서버에 전달한다. 서버는 Cell-ID에 매칭되는 절대 좌표값을 데이터베이스로부터 가지고 와서 Location API의 파라미터로 구글맵에 전달하게 되며, 구글맵은 자동으로 좌표에 해당되는 지역을 사용자에게 보여준다. 이와같이 모바일 노드의 위치를 추적하기 위해서 이동경로에 Cell-ID 기반의 센서노드를 설치함으로써 사용자는 맵을 통한 모바일 노드의 각 이동 경로를 쉽게 파악할 수 있다.

본 시스템은 업데이트 시간이 3초로 설정되어 있으며 업데이트 간격이 조정이 가능하다. 이는 신속한 모바일노드의 동선 파악이나 빠른 응답이 필요한 경우 시간 간격을 조정하여 응용할 수 있다. 또한 서비스를 제공하는 사용자 혹은 클라이언트의 경우 Ajax를 지원하는 인터넷 브라우저를 이용하여 고속으로 화면을 전환할 수 있고, 부드러운 화면을 서비스 받을 수 있다.

6. 결 론

본 논문에서는 무선센서네트워크와 IP 서비스의 연동을 위한 6LoWPAN 기반 테스트베드를 구축하고, 이를 통해 사용자가 보다 쉽고 빠르게 정보를 전달받을 수 있는 시스템구조를 제안하였다. 제안하는 시스템은 LTSGM으로서 센서노드부터 주변환경 정보 획득과 Cell-ID기반 위치인식 알고리즘을 통해 모바일노드의 위치를 제공한다. 또한 Web2.0의 주요한 어플리케이션으로 각광받고 있는 맵기반 서비스인 구글맵과 그래프 라이브러리의 연동으로 인해 보다 시각적인 정보제공이 가능하다. 사용자는 웹 브라우저를 구동할 수 있는 노트북, PDA, 휴대폰과 같은 단말기를 이용

하여 센서네트워크의 정보를 확인할 수 있다. 이를 통해 무선센서네트워크의 서비스를 제공받는 사용자의 입장에서 정보의 시각화와 접근성을 최대화하였으며, 모바일노드의 위치를 제공함으로써 이를 응용할 수 있는 다양한 분야로의 확장성 뿐만 아니라 다양한 작업을 수행할 수 있는 멀티 무선센서네트워크 시스템의 가능성도 확인할 수 있었다. LTSGM 플랫폼은 단일 PC상에서 시뮬레이션이 아닌 실제 환경의 다양한 요소들을 반영함으로써 이론에 기반한 LTSGM 플랫폼을 검증하였다. 센서네트워크는 정보의 정확한 수집과 더불어 수집된 정보를 사용자에게 효과적으로 제공하는 것이 중요하며, 본 논문에서 구현한 테스트베드는 이를 실생활에 적용할 수 기본 플랫폼이 될 것이라고 기대한다. 추후에는 본 논문에서 제안하는 시스템 구조를 확장하여 다양한 프로토콜과 서비스를 지원할 수 있는 테스트를 수행할 예정이며, IP와의 연동 시 보다 효율적인 기법과 6LoWPAN을 탑재한 게이트웨이에서 네트워크 계층의 IP 패킷 릴레이에 대한 연구를 수정, 보완할 계획이다.

ACKNOWLEDGMENT

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업(NIPA-2009-(C1090-0902-0046))과 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 세계수준의 연구중심대학사업(No. R31-2008-000-10062-0)의 연구결과로 수행되었음. 교신저자 : 추현승

참 고 문 헌

- [1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on sensor networks," IEEE Communications Magazine, vol.40, Issue 8, pp. 102-114, August 2002.

- [2] J. Qiangfeng and D. Manivannan, "Routing protocols for sensor networks," IEEE CCNC, pp. 93-98, January 2004.
- [3] IPv6 over Low Power WPAN Working Group, <http://www.ietf.org/html.charters/6lowpan-charter.html>
- [4] K. Kim, G. Montenegro, S. Park, I. Chakeres, and S. Yoo, "Dynamic MANET On demand for 6LoWPAN(DYMO-low) Routing," draft-montenegro-6lowpan-dymo-low-routing-00.txt, June 2007.
- [5] A. Cunha, A. Koubaa, R. Severino, and M. Alves, "Open-ZB: an open-source implementation of the IEEE 802.15.4/ZigBee protocol stack on TinyOS," IEEE MOBHO, pp. 1-12, October 2007.
- [6] ZigBee Alliance, <http://www.zigbee.org>
- [7] N. Kushalnagar, G. Montenegro, and C. Schumacher, "6LowPAN Overview, Assumptions, Problem Statement and Goals," RFC 4919, August 2007.
- [8] N. Kushalnagar, G. Montenegro, J. Hui, and D. Culler, "6LoWPAN Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, September 2007.
- [9] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: a wireless sensor network testbed," The 4th International Symposium on IPSN, pp. 483-488, April 2005.
- [10] H. Huo, H. Zhang, Y. Niu, S. Gao, Z. Li, and S. Zhang, "MSRLab6: An IPv6 Wireless Sensor Networks Testbed," Signal Processing, ICSP, vol. 4, pp. 16-20, November 2006.
- [11] S. Furrer, W. Schott, H. L. Truong, and B. Weiss, "The IBM wireless sensor networking testbed," TRIDENTCOM, March 2006.
- [12] IEEE Std 802.15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003
- [13] ZigbeX Hardware Reference, <http://www.hanback.com>
- [14] Atmega128(L) data sheet, revision R, updated 6/08.
- [15] CC2420 2.4 GHz IEEE 802.15.4/ZigBee RF Transceiver data sheet, TI (Chipcon), 2005.
- [16] SHT1x/SHT7x Humidity & Temperature Sensor data sheet, Sensirion, March 2007.
- [17] M. Sichitiu and V. Ramaduari, "Localization of wireless sensor networks with a mobile beacon," IEEE MASS, pp. 174-183, October 2004.
- [18] JpGraph, A object-oriented graph creating library for PHP, <http://www.aditus.nu/jpgraph/index.php>
- [19] Google-Map API, Developer documentation for the Google Maps API, <http://code.google.com/apis/maps/documentation>, 2008.

● 저 자 소 개 ●



김 계 원(Kyewon Kim)

2005년 대구대학교 정보통신공학부 졸업(학사)
2009년 성균관대학교 일반대학원 이동통신공학과 졸업(석사)
2009년 ~ 현재 삼성테크윈 SIS사업부 선임연구원
관심분야 : 센서네트워크, 프로토콜, 토폴로지, 모바일 컴퓨팅
E-mail : sound79@skku.edu



서 재 완(Jaewan Seo)

2008년 성균관대학교 정보통신공학부 졸업(학사)
2008년 ~ 현재 성균관대학교 일반대학원 휴대폰학과(석사)
2008년 ~ 현재 지식경제부 ITRC 지능형HCI융합연구센터 연구원
관심분야 : 센서네트워크, 신뢰성, 라우팅 프로토콜
E-mail : todoll2@skku.edu



황 대 준(Dae-Joon Hwang)

1978년 경북대학교 컴퓨터공학과 졸업(학사)
1981년 서울대학교 컴퓨터공학과 졸업(석사)
1986년 서울대학교 컴퓨터공학과 졸업(박사)
1981년 한남대학교 컴퓨터공학과 조교수
1987년 ~ 현재 성균관대학교 정보통신공학부 교수
1990년 7월 ~ 1991년 8월 미국 MIT 컴퓨터과학연구소 교환교수
1993년 11월 ~ 2000년 2월 미국 IBM Thomas J. Watson 연구소 초빙연구원
1997년 9월 ~ 2000년 3월 Open Cyber University 운영위원장
1997년 8월 ~ 2000년 3월 Iville Software Inc., 공동설립 및 이사
1998년 8월 ~ 2000년 8월 미국 World Trade Center Association 교육/연구위원
2001년 3월 ~ 2001년 8월 미국 U.C. Irvine 방문교수
2005년 3월 ~ 현재 미국 CoSN 및 Curriki 자문위원
2006년 8월 ~ 현재 IMS GLC Executive Strategic Council 위원 및 IMS Korea 위원장
2004년 7월 ~ 2008년 4월 한국교육학술정보원장
관심분야 : 첨단학습기술, 모바일컴퓨팅, Web 2.0 응용, 의료응용
E-mail : djhwang@skku.edu



추 현 승(Hyunseung Choo)

1988년 성균관대학교 수학과 졸업(학사)
1990년 University of Texas 컴퓨터공학과 졸업(석사)
1996년 University of Texas 컴퓨터공학과 졸업(박사)
1997년 특허청 심사4국 컴퓨터심사담당관실(사무관)
1998년 ~ 현재 성균관대학교 정보통신공학부 부교수
2001년 ~ 현재 한국인터넷정보학회/한국시뮬레이션학회 이사
2004년 3월 ~ 2006년 8월 대통령직속 교육혁신위원회 전문위원
2004년 8월 ~ 2008년 1월 한국인터넷정보학회 논문지 편집위원장
2005년 1월 ~ 현재 건강보험심사평가원 전문위원
2005년 10월 ~ 현재 지식경제부 ITRC 지능형HCI융합연구센터장
2008년 2월 ~ 현재 한국정보처리학회 이사
관심분야 : 유/무선/광네트워킹, 모바일컴퓨팅, 센서네트워크, 임베디드SW, 그리드컴퓨팅
E-mail : choo@ece.skku.ac.kr