

TinyOS 기반의 USN 응용 개발 분석

김영락* · 박찬흠** · 김성호*** · 최용준**** · 한덕구***** · 김종근*****

1. 서 론

유비쿼터스 센서 네트워크(USN)는 정보통신부의 IT839 전략의 3대 첨단 인프라 중 하나로 그 중요성이 날로 증가하고 있다. 인텔도 기술 로드맵 중에 센서 네트워크를 기반 기술로 예측할 정도로 미래 유비쿼터스 사회를 구현하기 위한 핵심 기반 구조로 주목받고 있다.

유비쿼터스 센서 네트워크란 기존 인간과 컴퓨터간의 커뮤니케이션에 일상생활에 산재된 사물과 물리적 대상을 추가시켜 협력 네트워크를 구성하고, 필요로 하는 모든 곳에서 센서 노드들을 통하여 자율적으로 정보를 수집, 관리 및 제어하는 시스템이다. 즉 물리 공간에 빛, 소리, 온도, 움직임 같은 물리적 데이터를 센서 노드에서 감지하고

측정하여 중앙으로 전달하고 처리하는 구조를 가진 네트워크이다.

센서 네트워크의 핵심 기술 요소로는 하드웨어 플랫폼과 노드에 들어가는 초소형 운영체제 기술을 들 수 있다. 하드웨어 플랫폼은 크게 센싱 데이터를 수집하기 위한 센서 노드, 데이터를 처리 전달하기 위한 싱크노드 등으로 나눌 수 있다. 현재 하드웨어 플랫폼은 국내외에서 다양하게 개발되고 있다. 또한 해당 하드웨어 플랫폼에 들어가는 센서 네트워크용 OS 개발도 활발히 이뤄지고 있다. 센싱 정보를 처리하고 노드들 간의 통신 등을 위해서는 센서 노드를 위한 시스템 환경이 요구된다. 제한된 메모리, CPU 자원 및 전력을 가진 센서 노드의 특성상 센서 노드 OS는 극소형의 크기를 가진다. 이러한 특징을 수용하는 센서 OS의 경우 버클리의 TinyOS나 한국전자통신연구원의 나노 Qplue 등이 개발되어 사용되고 있다.

※ 교신저자(Corresponding Author): 김종근, 주소: 김종근, 주소: 경북 경산시 대동 영남대학교(712-749), 전화: (053) 810-2555, FAX: (053)810-4630, E-mail: cgkim@yu.ac.kr

* 영남대학교 컴퓨터공학과 박사수료 (E-mail: yrkim@ynu.ac.kr)

** 영남대학교 컴퓨터공학과 박사수료 (E-mail: shem@ynu.ac.kr)

*** 영남대학교 컴퓨터공학과 석사과정 (E-mail: j9915526@ynu.ac.kr)

**** 영남대학교 컴퓨터공학과 특임교수 (E-mail: yjchoi@yu.ac.kr)

***** (주)포맨정보시스템 연구소장 (E-mail: dkhan@forman.co.kr)

***** 중신회원, 영남대학교 컴퓨터공학과 교수

※ 본 연구는 산업자원부와 한국산업기술재단의 지역혁신 인력양성사업으로 수행된 연구 결과임.

2. TinyOS 기반 센서 네트워크

2.1 하드웨어 플랫폼 기술 동향

USN의 가장 큰 장점은 센서 네트워크 노드들이 독자적으로 무선 네트워크 기술을 이용해 애드 혹(ad hoc) 네트워크를 구성한다. 하지만 센서 네트워크의 특성상 센서 노드는 가격과 유지 보

수 문제등으로 고급 컴퓨팅 장비를 사용할 수 없다. 그래서 센서 노드를 위한 하드웨어 플랫폼은 다음과 같은 특징을 가져야 한다.

- 한번 배치되면 유지보수가 어렵기 때문에 간단한 구조에 저전력으로 설계되어야 한다.
- 다양한 응용 분야에서 효과적으로 사용될 수 있도록 유연성과 모듈성을 갖추고 있어야 한다.
- 유비쿼터스 컴퓨팅 환경에 맞게 무선 네트워크 기술을 지원해야 한다.
- 많은 센서 노드가 산재되기 위해 가격이 저렴하고 크기가 작아야 한다.

국내 USN 기반산업은 아직 초기 단계로 서비스 모델 발굴이 이뤄지는 정도다. 하지만 국내 USN 시장규모가 해마다 70% 이상 성장하고 있고, 기술수준도 경쟁력이 높은 편이어서 국민소득 3만달러 시대를 이끌 중추적 역할을 할 것으로 기대 된다.

이와 같은 특징을 만족시키기 위해 많은 하드웨어와 노드들이 개발되고 있다.

2.1.1 센서서 노드 하드웨어 플랫폼

본고에서는 TinyOS를 지원하는 국내외 사용 플랫폼에 대해 설명한다.

Crossbow Mica Series : 미국 국방성의 DARPA 프로젝트의 스폰서를 받아 개발된 미국 버클리 대학 Mote 시리즈는 미국 정부와 관련 대학과 기업의 노력으로 가장 널리 사용되는 하드웨어이다. 이는 하드웨어뿐만 아니라 TinyOS라는 센서 네트워크용 OS와 각종 시뮬레이터 및 공개 애플리케이션을 통한 인프라가 구축되었기 때문이다.

Intel Mote(iMote) : 버클리대의 산업체 파트너인 인텔에서 개발된 Mote 시리즈로 ARM 기반의

마이크로 프로세서를 사용한다. 이는 센서 네트워크의 응용분야를 조사하고 다양한 응용을 위해 성능을 향상시킨 디자인이다. 센싱 정보의 복잡한 계산이나 라우팅 등의 상위 레벨의 정보 처리, 보안 등에 더 치중하기 위해 Intel의 iMote는 강력한 32bit ARM7TDMI CPU를 사용했다.

한백전자 ZigbeX Mote : MCU는 ATmega128 CPU를 사용하고 RF는 Chipcon의 CC2420을 사용한다. 특징으로는 기본으로 온도, 조도, 습도 센서, RTC를 모드에 장착하고 있다.

다양한 옵션 센서 모듈을 지원하는데 기상센서, 방범센서, 홈 오토메이션 센서 모듈, 위치추적 센서, 구조역학 센서, 바이오 센서, 자동차 센서 등의 응용 모듈을 제공한다.

하이버스 Hmote-zigbee : MCU는 TI MSP-430 프로세서를 사용하고, RF Chip은 CC2420을 사용하여 IEEE802.15.4 스택을 지원한다. USB 인터페이스를 통해 프로그래밍이 가능하고, 확장 슬롯을 통해 센서 보드가 연결되며, 센서보드는 파트별로 다양한 센서들을 연결하여 사용이 가능하다.

2.2 TinyOS(운영체제)의 구조 및 특징

UC 버클리에서 진행했던 WEBS(Wireless Embedded System) 프로젝트에서 자율적인 센싱과 통신 플랫폼 능력을 갖춘 초소형 모트(Mote)인 스마트 더스트(Smart Dust)와 함께 개발된 초소형 네트워크 임베디드 시스템 운영체제가 바로 TinyOS이다.

TinyOS[1]는 핵심 OS 코드가 4000바이트 이하, 데이터 메모리가 256바이트 이하의 초소형 운영체제이다. 매우 적은 메모리로 멀티 테스킹을 지원하는 이벤트 기반 멀티 테스킹 방식으로 동

작하며, 이벤트 발생이 없는 시간 동안 CPU를 Sleep 모드로 전환함으로써 효율적인 CPU 사용을 통해 저전력을 실현한다.

TinyOS는 컴포넌트 기반의 구조, 단순 이벤트 기반의 동시 모델과 함께 단계별로 분리되어 실행된다. TinyOS는 각각의 서비스를 재사용이 가능한 일련의 컴포넌트로 제공하며, 어플리케이션은 이러한 컴포넌트들 간의 연결로 구성되며, 그림 1과 같이 스케줄러를 포함한 Main 컴포넌트와 사용자 컴포넌트, 그리고 재사용가능한 일련의 컴포넌트들을 독립적인 연결방법을 이용하여 연결하여 TinyOS를 구성한다. 각각의 어플리케이션은 사용하고자 하는 일련의 컴포넌트를 주문형으로 만든다. 컴포넌트를 분해함으로써 사용되지 않는 서비스를 어플리케이션으로부터 제외할 수 있다.

TinyOS는 간단한 FIFO 구조를 가지는 소형 태스크 스케줄러와 재사용이 가능한 일련의 컴포넌트로 구성되며 컴포넌트는 상위 컴포넌트로부터 내려오는 non-blocking request에 응답하여 커맨드를 수행하는 커맨드 핸들러, 하위 컴포넌트로부터 신호되어지는 하드웨어 이벤트를 처리하는 이벤트 핸들러, 동적 할당 시 오버헤드를 방지하기 위한 고정 크기의 프레임, 스케줄러에 의해 실행되는 작업을 정의하는 태스크로 구성이 된다.

TinyOS의 프로세스들은 크게 태스크와 이벤트(하드웨어 인터럽트)로 나뉘며 스케줄링은 간편성을 위해 단순한 2 레벨 스케줄링 기법을 사용한다. 태스크는 다른 태스크에 의해 선점되지 않지만, 이벤트에 의해서는 선점된다. 이벤트는 특정 하드웨어 인터럽트나 특정 조건을 만족했을 경우 호출되는 프로세스로서, 다른 태스크 보다 먼저 실행되는 특징이 있다.

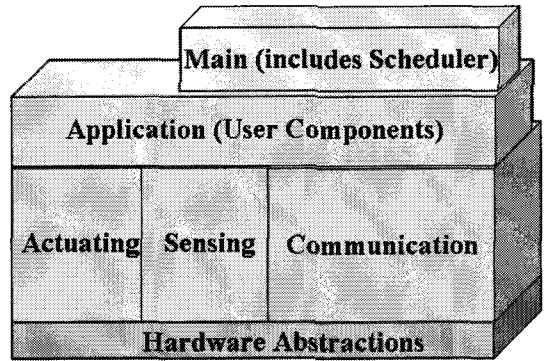


그림 1. TinyOS 구조

2.3 NesC 응용 개발 환경

2.3.1 NesC

NesC[2]는 TinyOS를 자체를 프로그래밍하고 센서노드의 기능을 개발하기 위한 언어이다. C 언어에 기반한 확장을 통해 만들어 졌지만, NesC는 새로운 특징을 가진다. 우선 NesC 프로그램은 안정성을 위해 전체 프로그램에 대한 분석을 통해 최적화를 수행한다. 또한 NesC는 동적메모리를 할당하지 않는 정적 언어이다. 즉 컴파일 타임에 모든 콜 그래프가 완전히 알려지게 된다. 이러한 제한 사항은 전체 프로그램 분석과 최적화를 쉽게 하고 더욱 정확하게 만들며 태스크와 이벤트에 기반한 동시성 모델을 정의하고 컴파일 타임에 데이터 경쟁을 검출한다. NesC는 컴포넌트 개념에 기반하고 있으며 직접적으로 TinyOS의 이벤트 기반 동시성 모델을 지원한다.

NesC에서 어플리케이션은 정의된 컴포넌트와 양방향성을 갖는 인터페이스로 만들어진다. 즉, 한 개 이상의 연결된 컴포넌트로 구성된다. 컴포넌트는 인터페이스를 제공하고 이를 통하여 다른 컴포넌트와 연결된다. 인터페이스는 컴포넌트에 접근할 수 있는 양방향의 통로이며 인터페이스 제공자가 반드시 구현해야 하는 커맨드와 인터페

이스 사용자가 구현해야 하는 이벤트라고 불리는 함수 집합을 정의한다. 하나의 컴포넌트는 여러 개의 인터페이스를 사용 및 제공 할 수 있으며, 같은 인터페이스에 대해 여러 개의 인스턴스를 생성 할 수 있다.

NesC에서는 Module과 Configuration 두 타입의 컴포넌트가 있다. Module은 어플리케이션 코드를 제공하고 한 개 혹은 그이상의 인터페이스를 구현한다. Configuration은 특정 컴포넌트에 의해 사용된 인터페이스를 다른 컴포넌트에 의해 제공된 인터페이스에 연결하고 다른 컴포넌트를 함께 결합하는데 사용된다. 이러한 과정을 Wiring이라고 한다. 그림 2에 예를 보인다.

NesC 응용 어플리케이션을 개발하기 위해서 우선 어플리케이션에서 사용할 서비스에 대한 명세를 작성해야한다. 서비스 명세를 작성함으로써 어플리케이션에 포함될 컴포넌트들을 선정할 수 있다. 어플리케이션에서 사용될 컴포넌트가 선정된 후 어플리케이션의 실행 구조를 표현한 상태천이도를 작성한다. 각각의 컴포넌트는 하나의 상태에 대응되며 커맨드 호출과 이벤트 시그널에 의해 상태가 천이되어 해당 컴포넌트의 동작이 실행된다. 어플리케이션의 실행구조에 적합한 상태천이도를 작성한 후, 사용자 컴포넌트와 사용될 컴포넌트를 상태천이도를 바탕으로 Wiring 한다. 각 컴포넌트를 독립적인 인터페이스로 연결하고, 인터페이스에 선언된 커맨드, 이벤트를 실행구조에 맞게 정의하고, 사용자 컴포넌트의 테스트를 작성하여 어플리케이션을 완성한다.

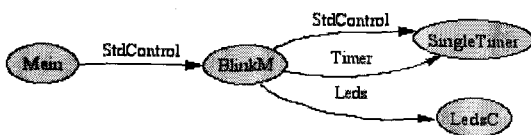


그림 2. Blink 어플리케이션 wiring

2.3.2 Java tools

TinyOS 개발 환경에는 데이터 교환 및 PC용 응용을 위한 자바 도구들이 구현되어 있다. 가장 대표적인 모트와 PC간의 데이터 전달을 위한 프로그램이 SerialForwarder이다.

SerialForwarder 프로그램은 PC의 프로그램에서 센서 네트워크와 통신을 하기위해, 시리얼 포트로부터 패킷 데이터를 읽고 인터넷 연결으로 전송하는데 사용하는 서버 프로그램이다. 프로그램이 작동되면, TCP 포트 9001번을 통해서 시리얼로부터 전달된 데이터를 네트워크 클라이언트 접속으로 포워딩을 해주고, 반대의 역할도 수행한다. 다수의 응용프로그램이 동시에 SerialForwarder에 연결될 수 있다.

2.3.3 Cygwin

일반적으로 센서 노드를 위해 사용되는 운영체제인 TinyOS는 리눅스 환경에서 개발한다. 윈도우 기반에서 TinyOS를 개발할 경우 Cygwin 프로그램을 사용한다. Cygwin은 리눅스가 아닌 윈도우 기반 운영체제에서 리눅스와 같은 환경을 제공하는 에뮬레이터이며, 리눅스의 모든 명령어를 사용할 수 있다.

cygwin 환경에서 TinyOS 응용어플리케이션을 make 후 생성된 실행이미지는 In-system programming 보드를 통해서 센서노드에 포팅한다. 베이스노드로 전송된 센싱 데이터를 PC에서 확인하기 위해서 Listen 또는 ListemRaw java 어플리케이션을 cygwin 환경에서 실행하여 확인할 수 있다. 또한 GUI가 구현된 java 어플리케이션 경우 cygwin에서 실행시키면 새로운 윈도우에서 실행이 되며 데이터를 확인할 수 있다.

2.3.4 웹 서비스

USN을 웹으로 서비스하기 위해서는 대량의

센싱 정보를 저장하고, 처리할 수 있는 DBMS와 웹 서버 그리고, DBMS와 연동이 가능한 서버용 스크립트 언어가 필요하다. PHP와 Mysql을 이용하면 USN 웹 서비스를 쉽게 구현할 수 있다.

PHP는 멀티 플랫폼을 지원하며, html 문서 내부에 포함되어 웹 서버에서 실행될 수 있는 스크립트 언어이다. PHP는 웹 서버가 실행이 될 때 해석 엔진자체가 같이 실행되면서 메모리로 로드되는 웹 서버 모듈로 동작을 하며, 다양한 종류의 데이터베이스에 집적 연결하는 함수를 제공하며, 사용자 수의 증가에 상관없이 인터프리터를 해주는 엔진 프로세스가 하나이기 때문에 실행 속도가 떨어지지 않고, 서버부하도 적다.

Mysql은 여러 플랫폼에서 동작을 하며, 커널 스레드를 이용한 멀티 스레드를 지원한다. SQL 함수들은 최적화된 클래스 라이브러리를 이용하여 구현되어 매우 빠르게 동작하며, 불필요한 메모리 할당을 하지 않는다. 권한과 암호 시스템은 유연성이 있고 보안이 잘되어 있다. 원격의 서버에 접속할 때 모든 암호들은 암호화 되어 전송된다. 크기가 큰 데이터베이스도 다룰 수 있으며 메모리 누수가 없다.

2.3.5 TOSSIM(TinyOS Simulator)

TOSSIM[3]은 TinyOS 기반의 센서 네트워크에서 TinyOS의 동작을 검증하기 위한 시뮬레이터이다. 실제 Mote에서 동작하는 NesC코드를 PC에서 시뮬레이션 하여 검증할 수 있다. TinyOS 컴포넌트 그래프에 기반한 discreat-event simulator 이며, Bit-level에서 TinyOS의 네트워크 stack을 시뮬레이션 할 수 있다. 따라서 Low-level의 네트워크 프로토콜 검증할 수 있다. TOSSIM은 표준 TinyOS release에 포함 되어 있으며, TOSSIM의 시뮬레이션 상황을 GUI 환경으

로 보여주는 자바 도구인 TinyViz도 제공이 된다.

2.3.6 MoteWorks 2.0

Moteworks 2.0은 Crossbow에서 제공하는 센서 네트워크 S/W 개발 플랫폼이다. MoteWorks 2.0 패키지는 Mote에 프로그래밍을 할 수 있는 MoteConfig와 xmesh, xsensor, xsniper 소스일체 및 sample 프로그램을 포함한 Cygwin INS-TALL, packet sniper와 같은 기능의 Xsniper, CVS 업데이트 기능의 putty, TortoiseCVS, 그리고 NesC 작성, 수정, 빌드를 손쉽게 할 수 있는 Programmers Notepad2를 포함하고 있다. 그 외에도 데이터베이스와 연동하여 포트로 들어오는 값을 저장하고, 모니터를 통해 그래프로 나타내주는 Moteviw 1.4와 Java .net Framework 1.0을 설치 할 수 있다.

3. USN 미들웨어

센서 네트워크 미들웨어[4]는 다양한 센서 노드로 구성된 하드웨어 계층과 운영체제 상에 존재하며, 응용 소프트웨어에 추상화된 인터페이스를 제공한다. 센서 네트워크는 일반적으로 데스크톱 환경이나 모바일, 무선 환경과는 또 다른 특징을 가지고 있어 다양한 센서 네트워크의 특징들을 만족시킬 수 있도록 미들웨어가 설계되어야 한다.

3.1 USN 미들웨어 특징

센서 네트워크는 제한된 컴퓨팅 능력과 에너지, 낮은 대역의 특성상 설치되는 환경의 영향을 많이 받는다. 또한, 노드 위치의 변화, 센서 네트워크 일부의 유실 등 센서 네트워크의 전체 혹은 부분이 동적으로 변화하기 쉽다. 이러한 문제점들을 효율적으로 처리하기 위해 미들웨어 개발에는

다음과 같은 주요 특징들이 고려되어야 한다.

- 지역적인 알고리즘(localized algorithm) : 일정 범위 내에 속한 이웃 노드들만의 통신에 의해서 달성되는 분산된 알고리즘을 바탕으로 한다. 이는 증가하는 네트워크 크기에 따른 그룹화, 확장, 네트워크의 분할이나 노드의 장애도 쉽게 극복할 수 있게 한다.
- 데이터중심통신(data-centric communication) : 센서 노드들에 의하여 만들어지는 새로운 방식의 통신이다. 예를 들어, 특정 노드들에 감지되는 온도 데이터를 단순히 요구하고 수신하는 형태가 아니라, 특정 온도가 메시지에 포함되면서 전달돼 이를 넘어서는 지역의 위치 정보를 발생시키도록 노드들에게 요구한다. 센서 네트워크의 미들웨어에서는 기존의 미들웨어와는 달리 통신뿐만 아니라 응용부분의 데이터를 처리하는 기능이 통합적으로 제공되어야 한다.
- 품질 조절 가변 알고리즘(adaptive fidelity algorithm) : 센서 네트워크에서 센싱된 정보를 처리하는 노드들의 신뢰성은 제한된 기능을 가지는 센서와 네트워크의 불확실성으로 인해 각각의 노드들이 제공하는 센싱값들을 신뢰할 수 없는 문제가 있다. 따라서 주어진 문제에 대한 답의 품질이 제한된 자원의 활용을 조절하여 제공되도록 대표 변수들이 지정되거나 조절을 가능하게 하는 알고리즘이 광범위하게 제공되어야 한다.
- 응용정보에 기반한 설계 및 운용(application knowledge in nodes) : 기존의 네트워크에서는 응용에 대한 정보를 반영하지 않고 다양한 영역의 응용을 지원할 수 있도록 만능 미들웨어의 설계가 요구되었지만, 센서 네트워크에서는 이러한 정보를 네트워크 기반 구조

나 노드 자체에 주입할 수 있는 방법까지 미들웨어에서 제공하도록 요구한다.

- 기존 네트워크와의 연계(interaction with the traditional networks) : 센서 네트워크는 이용 가능한 자원이 제한되어 있어 자원 집약적으로 요구되는 기능이나 대규모 정보의 저장을 네트워크 외부의 다른 네트워크나 장치에서 수행하도록 하는 것이 적합하다.

3.2 USN 미들웨어 요구 기능

센서 네트워크를 위한 미들웨어는 기본적으로 네트워크 요소의 개발, 센서 노드의 배포, 유지 및 관리, 감지 기반 응용의 수행을 원활하게 하는 것을 기본으로 한다. 응용부분에서 요구한 문제에 적절한 답을 제공하기 위해 미들웨어는 통신의 흐름에 따라 표면적으로는 상위 레벨 태스크를 형성하고, 이들 태스크와 무선 센서 네트워크간의 통신을 실시한 후, 형성된 태스크를 노드 각각의 역량에 맞게 분할해 각 노드에 배포한다. 계속해서 감지된 센서 데이터를 조합해 상위 레벨에 적합한 정보로 만들기 위한 데이터 융합(data fusion) 및 전달을 실시한 후, 최종적으로 사용자에게 데이터를 보고하는 기능을 제공한다. 미들웨어에 요구되는 기능을 요약하면 표 1과 같다.

3.3 센서 네트워크 미들웨어 모델구조

센서 네트워크 미들웨어는 제한된 자원을 가진 노드상에서 장시간에 걸쳐 고장 및 요구사항의 변경에 적응하여야 하며, 물리적인 세계에서 발생하는 센서 데이터를 데이터 사용자에게 높은 신뢰도로 전달해 주는데 필요한 공통 서비스를 제공하여야 한다.

표 1. 센서 노드 미들웨어의 요구 기능

구분	주요 기능
전력 관리	센서 노드의 저전력 특성을 반영한 프로세싱, 하드웨어 관리
위치인식 (Localization)	위치기반서비스 제공을 위한 위치정보 제공, 노드 정보 파악
시간 동기화 (Time Synchronization)	다양한 센서 노드간 정확한 협력 작업을 위한 시간 동기화
소프트웨어 배포와 자동 갱신	이동성이 있거나 지리적으로 넓게 분포된 센서 노드의 기능 수정과 신규 기능 부여
센싱 정보 저장과 정리	센서에서 측정된 센싱 데이터의 효율적 생성·전달·저장·관리
데이터 분배와 복제	센싱 데이터와 공통 데이터를 최적의 위치에 저장함으로써 데이터 접근성 증대, 중복 데이터 전송에 의한 에너지 소모 최소화
센서 노드와 네트워크 보안	센서 노드에서 신뢰하지 못하는 코드가 수행되지 못하도록 미들웨어 수준의 보안 기능
장애 관리 (Fault Tolerance)	신뢰성, 가용성이 낮은 환경에서 안정적 동작을 위한 장애 극복 능력
센서노드, 네트워크 네이밍	수많은 센서 노드의 효과적 구분·관리를 위한 식별체계
기존 네트워크와 연동	IP 네트워크의 비 IP 네트워크 상호 연동 기능

- 데이터 중심 구조 : 센서 네트워크 미들웨어는 사용자가 원하는 데이터를 수집하고 전달하는 과정을 효율적이고 신뢰성 있게 진행하는데 필요한 공통 기능들을 서비스로 제공하는 것이 필요하다. 수집된 데이터는 단독으로 존재하는 독립 데이터와 여러 장소에서 수집된 측정 데이터를 처리하여 얻어지는 협동 데이터가 있다. 대표적인 미

들웨어로는 코넬 대학의 Cougar[5]이라는 USN용 분산 데이터 처리 시스템이 있고 SINA[6]도 있다.

- 이벤트 중심 구조 : 어떤 센서 데이터는 사용자가 원하는 순간에 존재하지 않고 임의의 순간에 스스로 발생하여 사용자에게 비동기적으로 전달된다. 비동기적 이벤트 발생 및 전달에 초점을 맞추어, 센서 이벤트를 발생시키고 통보를 원하는 사용자에게 전달하는데 필요한 상위 기능들을 제공하는 이벤트 중심 구조의 미들웨어가 필요하다. 대표적인 미들웨어로는 일정 패턴으로 정의되는 복합된 이벤트를 기본 프로그래밍 기능으로 제공하는 DSWAR[7]이 있다.
- 이동 에이전트 지원 구조 : 센서 네트워크는 한번 설치되면 오랜 기간 자율적으로 동작하여야 한다. 따라서 네트워크 용도변경으로 인해 소프트웨어나 프로토콜들이 바뀌어야 할 때 소프트웨어 모듈의 동적다운로드와 기존의 응용프로그램의 대체를 지원하는 플랫폼이 필요하다. 이 개념을 확장시킨 것이 이동 에이전트 모델로서 응용 프로그램, 미들웨어 및 운영체제 구성모듈을 자동적으로 업데이트할 수 있게 하는 구조가 필요하다. 대표적인 미들웨어로는 USN의 효율적인 활용과 프로그래밍을 위한 제안된 프레임워크로 SensorWare[8]이 있으며, Mate[9]라는 초경량의 통신 중심 가상 머신 형태의 미들웨어도 있다.
- 제한자원 보호 구조 : 센서 네트워크는 제한된 에너지를 가진 배터리를 사용하여 장기간 동작하는 것을 전제로 하기 때문에 에너지를 최대한 절약하는 방안이 모든 소프트웨어 모듈들에 적용되어야 한다. 따라서 에너지 절

감을 포함하여 부족한 자원을 효율적으로 사용하는데 필요한 각종 기능들과 에너지 절감형 공통 서비스 모듈들을 제공하는 자원보호 미들웨어가 필요하다. 대표적인 미들웨어는 프린스턴 대학에서 ZebraNet 프로젝트의 일환으로 시작된 Impala[10]가 있다.

- 공통 센서기능 제공 구조 : 센서 네트워크가 제공하여야 할 공통 서비스들을 모아서 미들웨어에서 제공하는 것은 모든 미들웨어의 기본적인 사양에 해당된다. 특히 센서 네트워크가 제공할 수 있는 공통 서비스로 시간 동기화 서비스, 노드 위치 인식 서비스 등이 있다. 대표적인 미들웨어는 남가주 대학에서 센서 네트워크 미들웨어로서 확장성을 가진 협동 구조로 설계된 SCADDS (Scalable Coordination Architectures for Deeply Distributed Systems)가 있다.

4. USN 기반 교량 관리 시스템

TinyOS기반으로 간단한 교량관리 시스템을 구현한 예를 보인다.

4.1 시스템 개요

모든 시설물은 시간의 경과와 함께 성능도 점차 저하하며, 어떤 경우에는 예기치 않은 외부하중이나 재난에 의해 손상을 받아 시설물의 성능이 갑작스럽게 저하되기도 한다. 이 때문에 사회간접시설물은 확충 못지않게 구조물에 대한 “수명간 총 기대비용”을 고려해 시행하는 수명관리 중요성이 대두된다. 또한, 시설물의 효율적인 유지관리를 위한 정밀안전진단 기법 또는 상시모니터링 및 평가기술이 절실히 요구되고 있다. 상기 기술개발을 위해서는 다양한 추정기법을 중심

으로 시설물 안전감시에 필요한 구조 특성값 도출을 위해 자료 수집, 분석 기법과 목적데이터의 종류에 대한 데이터 형식 연구가 선행된다. 여기에는 필요한 구조 특성값을 도출할 수 있는 적절한 센서의 위치선정과 센싱 데이터를 목적데이터로 해석 및 분석하는 기법의 알고리즘 연구가 동반된다. 이는 이종의 다양한 센싱 데이터를 효율적으로 취합하고 처리할 수 있는 USN 라우팅 기법 연구의 근간을 이룬다. USN기반의 실시간 안전감시 시스템 구축에 대한 전반적인 개요는 그림 3과 같다.

현재 교량, 터널, 댐 등 주요 사회기반시설에서 행하는 실시간 모니터링은 유선 센서 시스템에 의존하고 있으며, 무선 통신 기술을 이용하고 있다 하더라도 네트워크에 대한 기술력은 여전히 미비하다. 따라서 효율적인 지리적 특성과 같은 국내 여건에 적합한 관리 시스템을 구축하기 위해서는 시스템 체계와 센서검교정, 데이터값 보정, 데이터 이상값 제거 등 안정성에 관련되는 제반 알고리즘에 관한 연구로 기존 유/무선 센서 시스템에서는 얻을 수 없었던 많은 장점을 얻어야 한다.

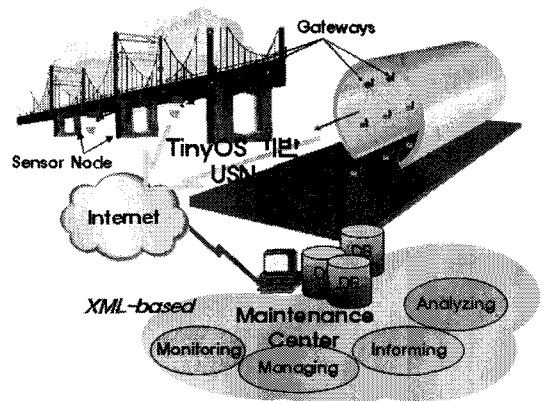


그림 3. USN기반의 실시간 교량관리 시스템 구축 기술 개요

4.2 미들웨어 설계

각 센서노드에서 센싱된 데이터는 게이트웨이 노드로 취합이 되고 취합된 데이터는 미들웨어로 전송이 이루어진다. 미들웨어에서는 센서 노드들의 채널, 그룹ID, 센싱 정보 등을 토대로 센싱 데이터를 필터링하여 관리 시스템으로 전송을 한다. 비동기적 센서 이벤트 발생에 의해서 수집된 데이터인 경우 정기적인 센싱 데이터보다 우선적으로 미들웨어에서 필터링되어 관리 시스템으로 전송이 된다. 앞 장에서 언급하였듯이 미들웨어는 전력 관리, 위치인식, 시간 동기화, 소프트웨어 배포 와 정보 갱신, 데이터 분배와 복제, 장애 관리 그리고 기존 네트워크와 연동을 고려하여 미들웨어가 설계되어야 된다.

본 시스템에서는 Java 처리 환경을 배제하기 위해 독자적인 Serialforwarder와 미들웨어를 구현한다.

그림 4는 직접 개발한 넷미들웨어로서 센싱된 데이터를 게이트웨이 노드로부터 제공받아 관리 시스템으로 데이터를 전송해주는 serialforwarder와 미들웨어의 위치를 보인다. 노드의 프로그램은 NesC로, 전용 serialforwarder 및 미들웨어는 visual C로 웹기반 관리시스템은 PHP로 구현된다.

4.3 시스템 구현 결과

웹기반의 교량 모니터링 시스템의 특징은 언제 어디서나 네트워크가 연결되어 있고 인터넷이 가

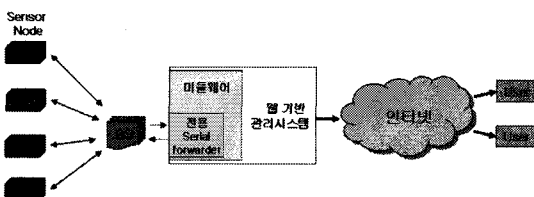


그림 4. 시스템 구성

능한 곳이라면 실시간으로 교량의 특정한 부분뿐만 아니라 교량의 모든 상태 정보를 얻을 수 있다. 즉 센서 노드로부터 센싱된 데이터를 교량관리 시스템과 연동된 웹 서버로부터 실시간 서비스를 받을 수 있다. 교량 관리 시스템은 교량관련 업무뿐만 아니라 기록 및 보고용에서 시설물의 유지 관리에 필요한 이전 정보의 열람도 가능하며, 센서 노드로부터 센싱된 데이터에 대한 판단기능이 제공되어 교량의 위험 상태를 알려줄 수 있는 통합 시스템이라 할 수 있다. 또한 웹기반 교량 모니터링 시스템은 업무 시간의 단축 및 정보 검색 절차의 단순화 그리고 센서 노드로부터 센싱된 데이터의 신속한 판단으로 업무 효율성을 높일 수 있으며, 유지관리 비용의 절감은 물론 일관성 있는 교량 시설물의 유지관리를 가능하게 할 것이다.

다음은 구현된 웹 기반의 교량 모니터링 시스템을 예를 보인다.

그림 5의 웹기반의 모니터링 시스템 실험환경의 예를 살펴보면 센서 노드로부터 취합한 데이터를 메인서버에서 필터링을 하고 인터넷을 통하여 일반 사용자 및 관리자에게 서비스를 제공하게 된다. 관리 시스템은 웹기반으로 관리 명령어를 통하여 센서노드를 제어하게 되며, 웹 매니저를 통하여 센서노드의 상태 변화를 그림 6, 7과 같이 실시간으로 알려주게 된다.

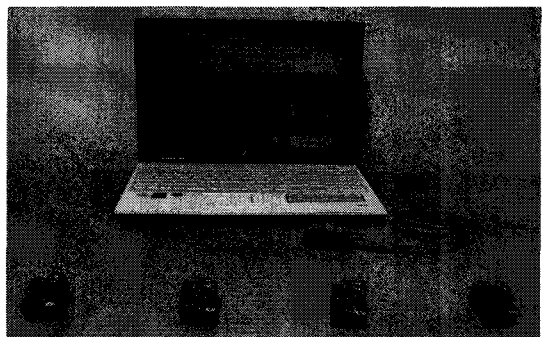


그림 5. 웹기반의 교량 모니터링 시스템 예

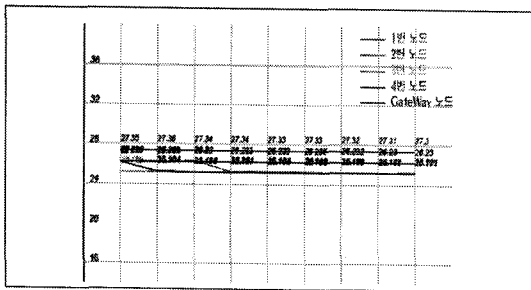


그림 6. 센싱 데이터를 그래프로 표현

Node 1번					Node 2번				
현재 시간	노드 ID	온도	습도	진동	현재 시간	노드 ID	온도	습도	진동
27:05	1	25.1	55.2	0.1	27:05	2	26.5	58.3	0.2
27:10	1	25.5	56.1	0.1	27:10	2	27.0	59.4	0.2
27:15	1	26.0	57.0	0.1	27:15	2	27.5	60.5	0.2
27:20	1	26.5	58.0	0.1	27:20	2	28.0	61.6	0.2
27:25	1	27.0	59.0	0.1	27:25	2	28.5	62.7	0.2

Node 3번					Node 4번				
현재 시간	노드 ID	온도	습도	진동	현재 시간	노드 ID	온도	습도	진동
27:05	3	25.2	55.3	0.1	27:05	4	26.8	58.4	0.2
27:10	3	25.7	56.2	0.1	27:10	4	27.3	59.5	0.2
27:15	3	26.2	57.1	0.1	27:15	4	27.8	60.6	0.2
27:20	3	26.7	58.0	0.1	27:20	4	28.3	61.7	0.2
27:25	3	27.2	58.9	0.1	27:25	4	28.8	62.8	0.2

그림 7. 센서 노드의 로그 정보

교량에 설치된 각각의 센서들이 센싱한 온도, 습도, 변형률 및 진동 등의 정보를 데이터베이스에 저장하며 실시간으로 표와 그래프를 이용하여 확인할 수 있다.

5. 결 론

무선 센서 네트워크 기술이 사회기반시설 및 물류/유통, 가정 전반에 걸쳐 다양하게 이용되어 짐으로써 인간 사회에 적용될 수 있는 광범위함은 우리 모두가 인식하고 있다. 그리고 이러한 사회 현상을 뒷받침하기 위하여 센서 네트워크 운영 체제를 통한 응용프로그램의 손쉬운 개발, 그리고 저전력 센서 네트워크 지원 및 유연한 센서 노드들의 관리가 필수적인 기술로 떠오르게 되었음을 알 수 있다. 그리하여, 본고에서는 센서 네트

워크에서 TinyOS 기반 센서 네트워크, 센서 네트워크의 기술동향, 그리고 USN 미들웨어 특징 및 요구 기능에 대해 전반적으로 알아보았다. 이를 통하여 우리는 무선 센서 노드를 이용하여 교량의 상태와 정보를 습득하여 관리자에게 교량의 상태 및 관리정보를 GUI로 제공하는 USN기반 실시간 IMS구축 시스템의 구현 예를 보였다.

USN기반의 실시간 관리 시스템 구축은 다양한 응용이 가능한 기술 및 경험을 확보할 수 있으며 안전 감시시스템에 있어 데이터수집, 데이터 전송, 데이터베이스 및 웹연동(인터넷서비스) 등의 요소기술의 기술력도 중요하다.

참 고 문 헌

- [1] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler and Kristofer Pister, "System Architecture Directions for Networked Sensors," April 27, 2000.
- [2] David Gay, Phil Levis, Rob von Behren, Matt Welsh, Eric Brewer, and David Culler. "The nesC Language: A Holistic Approach to Networked Embedded Systems," *In Proceedings of Programming Language Design and Implementation (PLDI) 2003*, June 2003.
- [3] Philip Levis, Nelson Lee, Matt Welsh, David Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Application," *UC Berkeley*. June 26. 2003.
- [4] 김영만, "센서 네트워크 미들웨어 구조 및 연구 현황," *정보과학회지 제 22권 제12호*, pp. 13-20, 12. 2004.
- [5] Y. Yao and J.Gehrke. "The Cougar Approach to In Network Query Processing in Sensor Networks," *SIGMOD Record*, Vol.31, No.3, Sept. 2002.
- [6] C. Shen, C. Srisathapornphat and C. Jaikeo, "Sensor Information Networking Architecture

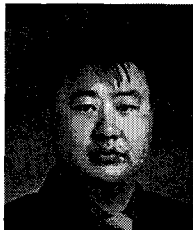
and Applications," *IEEE Personal Communications*, Vol.8, No.4, pp. 52-59, Aug. 2001.

[7] S. Li. Son, and J. Stankovic, "Event Detection Services Using Data Service Middlewar in Distributed Sensor Networks," *Int'l Workshop on Information Processing in Sensor Networks(IPSN'03)*, Palo Alto, CA, Apr. 2003.

[8] A. Boulis, C.C. Han, and M. B. Srivastava, "Design and Implementation of a Framework for Programmable and Efficient Sensor Networks," *In The First International Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, CA, 2003.

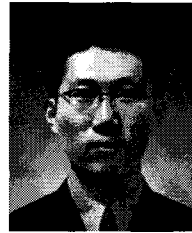
[9] P. Levis and D. Culler, "Mate : A Virtual Machine for Tiny Networked Sensors," *Proc. of ACM Conf. Architectural support for Programming Languages and Operation Systems*, San Jose, CA, Oct. 2002.

[10] T. Liu and M. Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," *Proc. of ACM SIGPLAN symposium on Principles and pactice of parallel program-ming*. pp. 107-118, 2003.



김 영 락

- 2002년 금오공과대학교 응용수학과(이학사)
- 2004년 영남대학교 대학원 컴퓨터공학과(공학석사)
- 2004년~현재 영남대학교 대학원 컴퓨터공학과(박사수료)
- 관심분야 : 컴퓨터네트워크, Mobile Ad Hoc Network, Ubiquitous Sensor Networks 등



박 찬 흠

- 2002년 한국교육개발원 컴퓨터공학과(공학사)
- 2004년 영남대학교 대학원 컴퓨터공학과(공학석사)
- 2000년~2001년 ㈜레이넷 개발총괄팀장
- 2005년~현재 영남대학교 대학원 컴퓨터공학과(박사수료)
- 관심분야 : 모바일/온라인/아케이드 게임개발, 모바일 콘텐츠 자동변환 솔루션, 게임품질관리 등



김 성 호

- 2006년 영남대학교 컴퓨터공학과(공학사)
- 2006년~현재 영남대학교 대학원 컴퓨터공학과(석사과정)
- 관심분야 : 센서네트워크, MAC프로토콜, 라우팅 알고리즘, 네트워크리프로그래밍 프로토콜, TinyOS/NesC 프로그래밍 등



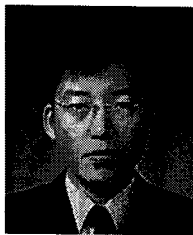
최 용 준

- 1996년 영남대학교 물리학과(이학사)
- 1998년 영남대학교 전산공학과(공학석사)
- 2003년 영남대학교 컴퓨터공학과(공학박사)
- 2001년~2005년 (주)인투정보 대표이사
- 2005년~현재 영남대학교 컴퓨터공학과 특임교수
- 관심분야 : 컴퓨터네트워크, USN, Mobile Networks, 임베디드시스템 등



김 종 근

- 1981년 영남대학교 전자공학과(공학사)
- 1987년 영남대학교 대학원 전자공학과(공학석사)
- 1991년 (일본) 전기통신대학(공학박사)
- 1997년 (미국) Virginia Tech. 연구교수
- 2003년 (미국) UCSC 연구교수
- 1991년~현재 영남대학교 컴퓨터공학전공 교수
- 관심분야 : 컴퓨터 네트워크, 무선 모바일 네트워크, 분산처리, 운영체제, 멀티미디어기반 가상강의 시스템 등



한 덕 구

- 1994년 계명대학교 통계학과(이학사)
- 1994년 (주)포맨정보시스템 입사
- 2007년~현재 (주)포맨정보시스템 연구소장
- 관심분야 : ZigBee, UWB, Network Topology, 홈네트워크, 안전진단, 위치인식, U-Farm 등