

# 유비쿼터스 센서네트워크 기술

□ 김정훈, 황태호, 송병철 / \*전자부품연구원 유비쿼터스연구센터

## I. 서론

새로운 컴퓨팅 패러다임인 유비쿼터스 컴퓨팅 관련 기술 중 센서네트워크 기술은 진입 기술로 중요한 의미를 갖는다. 이런 새로운 컴퓨팅 패러다임의 기술이 성공적으로 발전하기 위해서는 기반이 되는 진입기술의 확산이 중요하다. 본문에서는 현재까지 진행된 유비쿼터스 센서 네트워크 기술 동향과 센서 네트워크에서 사용되는 시스템 OS 기술 및 관련 기술에 대해서 알아본다.

## II. 센서 네트워크 기술

### 1. 개요

저전력 소형 무선 센서 네트워크는 유비쿼터스

컴퓨팅 분야의 핵심 기술이다. 근거리에서 동작하는 RF 매체로 연결된 무선 지능 센서들은 유비쿼터스 컴퓨팅 환경에서 사람과 컴퓨터 사이의 중개 역할을 수행한다. 센서 네트워크(sensor network)는 물리공간의 상태인 빛, 소리, 온도, 움직임 같은 물리적 데이터를 센서노드에서 감지하고 측정하여 중앙의 기본 노드(base-station or sink)로 전달하는, 센서 노드들로 구성되는 네트워크이다. 센서 네트워크는 일반적으로 멀티-홉(multi-hop) 무선 네트워크 형태의 다수의 분산 센서 노드들로 구성된다. 센서 노드들은 하나이상의 센서(온도, 소리, 빛, 가속도, 자기장 등), 액추에이터(actuator), 마이크로 컨트롤러, 수 십 KB 크기의 EEPROM, 수 KB의 SRAM, 수백 KB 크기의 플래시 메모리, 근거리 무선 통신 모듈로 구성된다. 센서 네트워크 기술은 센서와 무선 네트워크 기능을 이용하여 물리공간에서 측정된 아날로그 데이터를 디지털 신호로 변환하

고, 인터넷 같은 전자공간에 연결된 루트(Root) 노드로 전달하는 입력시스템의 역할을 한다. 물리적 세계와 사이버 세계를 연결할 수 있는 특징 때문에, 센서 네트워크의 개념은 새롭게 대두되고 있는 지능형 서비스들의 지능형 환경 모니터링, 위치인지 서비스, 지능형 의료시스템, 지능형 로봇 시스템 등 다양한 분야에 적용되고 있다.

센서 네트워크의 장점은 낮은 사양의 하드웨어를 이용하여 무선 애드-혹(ad-hoc) 네트워크를 구성할 수 있는 점이다. 예를 들어, 지금까지 개발된 블루투스(bluetooth), 무선랜(wireless LAN) 등의 무선 네트워크 기술들은 반드시 컴퓨터, PDA 같은 고급 컴퓨팅 장치를 필요로 하는데, 센서 네트워크 노드는 독자적으로 네트워크를 구성한다. 이런 네트워크 구성의 용이성 때문에 유비쿼터스(ubiquitous) 컴퓨팅 환경의 기반기술로 사용될 수 있을 것이다. 인텔도 기술 로드맵 중에 센서 네트워크를 기반 기술로 예측하고 있다.

근래 센서 네트워크와 관련된 칩과 시스템들이 상용화되기 시작했으나 널리 보급되고 산업화되기 위해서는 해결되어야 할 문제들이 많이 남아있다. 네트워크 프로토콜, 적절한 응용 서비스 등의 부재로 인해 빠른 시간 내에 상용화되기 힘들 것으로 예측되기도 한다.

적용되는 대부분의 환경에서, 센서 네트워크 노드는 오랜 기간동안 동작해야 하고 무선 통신 방식을 사용한다. 그래서 전기 또는 추가로 획득되는 에너지에 따라 센서 노드들의 전체적인 동작에 영향을 준다. 각 센서 네트워크 노드들은 에너지 소비를 최소화하기 위해, 라디오 등의 장치 요소들의 대부분의 동작시간에 전원을 공급하지 않는다. 물리 세계의 정보를 정확하게 입력받기 위해서는 센서 네트워크 노드들을 밀집하게 배치하는 것이 일반적인 방법

이 될 수 있다. 이러한 제한적인 하드웨어 사항, 네트워크 구성에 필요한 요소들은 지금까지의 네트워크와의 차별되는 특징이며, 네트워크 프로토콜의 기능이 더 복잡하게 된다. 이런 운영적 단점에도 불구하고 노드들을 유지보수하고 배치시키는 것은 저비용이어야만 한다. 수동적으로 구성할 수 있는 소형 장치들로 구성된 대규모의 네트워크들은 비현실적이기 때문에, 그 노드들은 자체적으로 구성되어야만 하고 개별적인 장치들을 동작시키는 것 보다 자율적으로 네트워크를 운영하고 프로그래밍 하는 수단을 제공해야만 한다. 이러한 난제점을 극복하는 것이 센서 네트워크가 가진 기술적 차별성이다.

## 2. 임베디드 네트워크 기술

### 1) 마이크로 프로세서 및 저전력 기술

센서네트워크 노드들의 하드웨어는 마이크로 프로세서, 기억장치, 센서, ADC(Analog to Digital Converter), 데이터 송수신기, 그리고 에너지원으로 구성된다. 반도체 회로가 점점 작아지면서, 주어진 시간 주파수 동작에서 작은 전력을 소비하고 다양한 응용 서비스 범위에 맞게 적용되었다. 간단한 마이크로 컨트롤러의 소형화는 기능을 추가하는 것 보다 약 10 MHz로 동작하는 동안 1mA 정도에서 동작하도록 만드는 것으로 좀 더 효율적이다. 회로들의 대부분은 항상 전원 공급이 필요한 것은 아니기 때문에 대기 모드를 동작한다면 전원은 약 1uA로 사용될 수 있다. 이러한 차이가 그때에 1% 활성화 되면, 그것의 평균 전력 소비는 단지 수 uW가 된다.

이러한 전력은 다양한 방법으로 얻을 수 있다. 태양전지는 외부에서 제공 센티미터 범위의 면적당 약 10mW, 실내에서는 10~100mW를 생성한다. 에

어떤 공기 유입출로나 창문의 진동처럼 에너지의 기계적인 소스는 약 100mW를 만들 수 있다. 전형적인 정육면체인 1cm×1cm×1cm의 건전지는 약 1000mA-hours를 저장하고, 그래서 cm 단위의 장치들은 많은 환경에서 추가적인 전원을 공급해 주지 않아도 거의 독립적으로 동작할 수 있다. 그러나 저전력의 마이크로프로세서는 PC의 것보다 10,000배 작은 저장 공간을 가지거나 프로그램 용도의 저장 공간인 ROM 100Kbyte보다 작고, 데이터를 위한 RAM은 10kbyte보다 적은 제한적인 저장 공간을 가지고 있다. 기억장소의 이러한 제한된 용량은 칩면적의 대부분과 전력예산의 많은 부분을 소비한다. 디자이너들은 통상적으로 칩과 분리된 영역에 Mbyte 정도의 대용량의 플래시 저장장치를 결합시키는 방법을 적용하고 있다.

## 2) 소형 센서

센서들은 네트워크 노드들에게 눈과 귀와 같이 물리적 신호를 제공한다. 많은 사물들은 변화하는 환경 조건이 요구될 때 그들의 전기적인 특성을 변화시킨다. 센서들은 어떤 범위를 넘는 예상된 변화를 지원할 수 있도록 제작된다. 예를 들면, 온도 감지 센서는 부드럽게 변화하는 저항체이다. ADC 컨버터는 전압의 감소를 마이크로 프로세서가 저장하거나 처리할 수 있는 이진 데이터로 변환한다. 광전지, 안개 감지 센서는 비슷하게 동작하지만 수분의 저항을 변화하기 위해 사용하거나 입사되는 광자를 사용하는 물질들에 의해 분리 되어진 미세한 회로들로 구성된다.

정확한 감지 기능을 구현하기 위한 감지회로 구조들은 다른 현상을 감지하기 위해 발전되어 왔다. 이러한 구조들은 수 mW 정도를 소비하고 단지 짧은 시간 동안 전원이 켜져 있다. 매우 효율적인

ADC 컨버터가 발전되어져 왔고 그 결과 센서의 하위시스템들은 프로세서와 비슷한 에너지 소모 특성을 가지고 있다.

MEMS (Microelectromechanical systems) 센서는 저비용과 고효율로 매우 넓은 부분의 물리 현상을 감지할 수 있다. 구형 공정에서 매우 얇은 기계적 구조를 만들기 위해 실리콘 위의 트랜지스터들을 세밀하게 만드는 과정을 사용한다. 이러한 공정을 이용하면 중력이나 가속도는 확대 되어지거나 디지털화 되어질 수 있는 물질 특성에서 변화의 원인이 되는 내부의 힘을 일으키는 한쪽이 고정된 물질을 편향시킬 수 있다. 최초 사용된 분야는 에어백을 자동적으로 실행시키는 회로이며 MEMS 센서와 가속도계를 사용하였다.

정확도가 높은 압전 가속도계는 큰 비용이 드는 반면에, MEMS 센서는 저가로 충분한 정확도를 제공한다. 다양한 종류의 MEMS 장치들은 다양한 힘과 화학적인 농도와 환경요인들을 감지할 수 있다.

## 3) 저전력 무선 통신

근래에 제조업자들은 센서들을 많은 가정기구나 자동차나 부속품들에 부착하였다. 그러한 놀랄만한 성장은 물리적인 데이터를 장치들이 전송하고 저장하고 처리할 수 있는 정보의 단위로 변환하는 것처럼 센서가 읽은 값을 다른 장치들에 무선으로 전송하는 것이다.

무선 라디오는 모바일 컴퓨터를 위한 무선 LAN과 핸드폰, 워키토키, 삐삐 등의 무선이 장치에 접목될 수 있도록 전형적인 CMOS 기술을 사용하여 만들어질 수 있다. 그러나 무선으로 통신하기 위해 요구된 에너지양은 거리에 따라 급속히 증가한다. 뿐만 아니라 사람이나 벽들과 간섭을 하는 방해물들은 그 신호를 감소시킨다.

무선 LAN 과 핸드폰은 수백 mW를 소비하고 성능이 좋은 내부 회로를 필요로 한다. 무선 센서네트워크 라디오는 일반적으로 약 20mW를 소비하고 그들의 범위는 수십 미터에서 측정되어 진다. 먼 거리를 커버하는 소형 무선 장치들에 대해서는, 그 네트워크는 라우터들이 인터넷에서 정보를 이동시키는 것처럼 많은 노드들을 통해 재전송 기법을 이용하여 정보를 알려주어야만 한다.

### 3. 소형 OS 구조

Unix 같은 일반적인 운영체제는 50~100Mz의 속도로 다수의 Mbit RAM과 Gbit 이상의 보조 저장 장치를 가지고 32-bit 마이크로 프로세서에서 동작한다. 오늘날 이것은 한번 충전해서 여러 시간 동작시킬 수 있는 손바닥만한 크기의 장치들에서 잘 수행되어 질 수 있다.

무선 센서 네트워크의 좀더 전형적인 운영방법은 이러한 자원들의 작은 부분을 가지고 AA 크기 배터리 두개로 일년 이상 동작한다. 더욱 이러한 응용 서비스는 복잡한 인간의 상호작용에 집중하지 않고 물리적인 세계의 구조적인 상호작용에 초점을 맞춘다.

센서 노드들은 극도의 저전력 경량 장치에서 동작할 수 있으면서 동시에 환경과 응용 프로그램의 변화에 대처하기 위해 동적인 재구성을 지원할 수 있는 특별한 구조의 운영체제가 필요하다. 또한, 센서 노드를 위한 운영체제는 네트워크와 MAC을 위해 동시에 여러 개의 비동기적인 이벤트를 다룰 수 있어야 하며 응용을 통한 분산형, 데이터 중심형 프로그래밍 모델을 지원해야 한다. 이렇게 상반된 것처럼 보이는 요구조건을 모두 만족시키기 위해서는 새로운 구조의 운영체제 및 프로그래밍 모델을 설

계하고 구현하는 것은 매우 어려운 일이다.

지난 General Purpose Operating System 및 Programming Model의 진화과정에서 살펴볼 수 있는 접근방법은 크게 세 가지로 구분할 수 있다.



- 새로운 구조의 커널 설계를 통한 접근
- 새로운 프로그래밍 모델의 컴파일러를 통한 접근
- 기존 커널의 한 모듈 또는 미들웨어 형태의 구현을 통한 접근

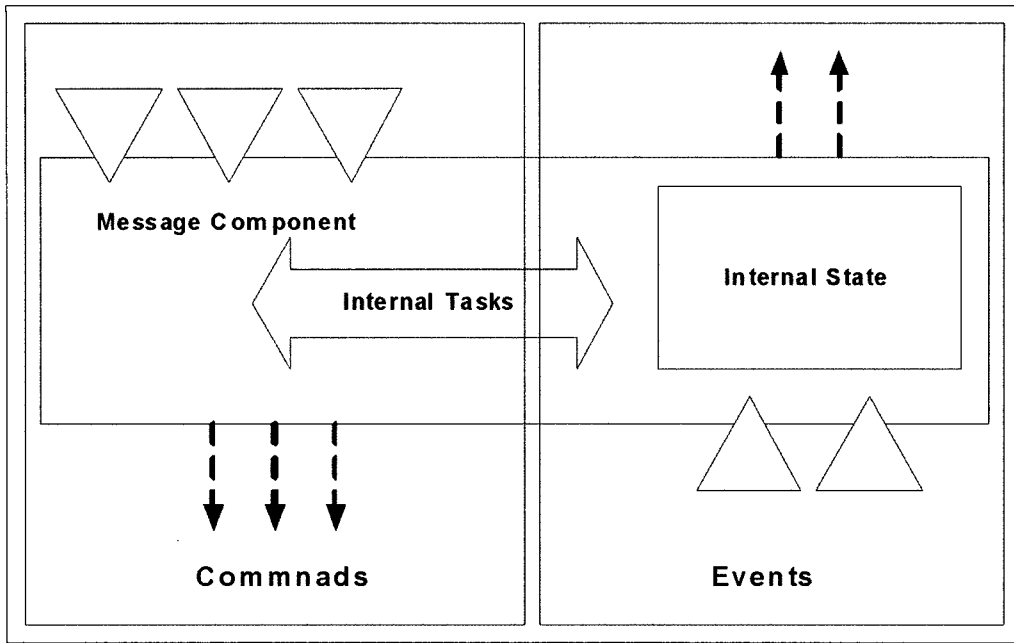
UNIX(C language), OOP(Java, C++), Real-Time OS 등과 같은 성공적인 운영체제 및 프로그래밍 모델의 등장 및 진화과정에서는 세 가지 접근 방법의 혼용을 통해 명확한 목적과 기능이 최적의 설계를 통해 구현되었다.

다음에 초소형 저전력 무선 센서 네트워크 노드들의 시스템의 특징 및 주요 기술을 살펴본다.

#### 1) Tiny OS

UC Berkeley(David Culler)에서 WEBS (Wireless Embedded System) 프로젝트에서 센싱과 무선 통신 능력을 가진 Smart-Dust (Mote)를 위한 PHY, MAC, NET과 다양한 응용들의 동작을 위해 설계한 4kBytes 이하의 시스템 코드를 가진 소형 운영체제이다.

Tiny OS의 가장 큰 특징은 C언어에 기반한 nesC 언어를 통한 컴포넌트 기반의 시스템 및 프로그래밍의 구조와 단순 이벤트 기반의 동작이다. 이벤트 기반(Event Driven)의 시스템 동작은 저전력을 위한 효율적인 구조임과 동시에 기존의 많은 스택 메모리를 요구하는 멀티스레드 방식의 단점을 개선해 적은 메모리로 멀티태스킹을 지원할 수 있는 장점



〈그림 1〉 Tiny OS의 컴포넌트 모델

이 있다.

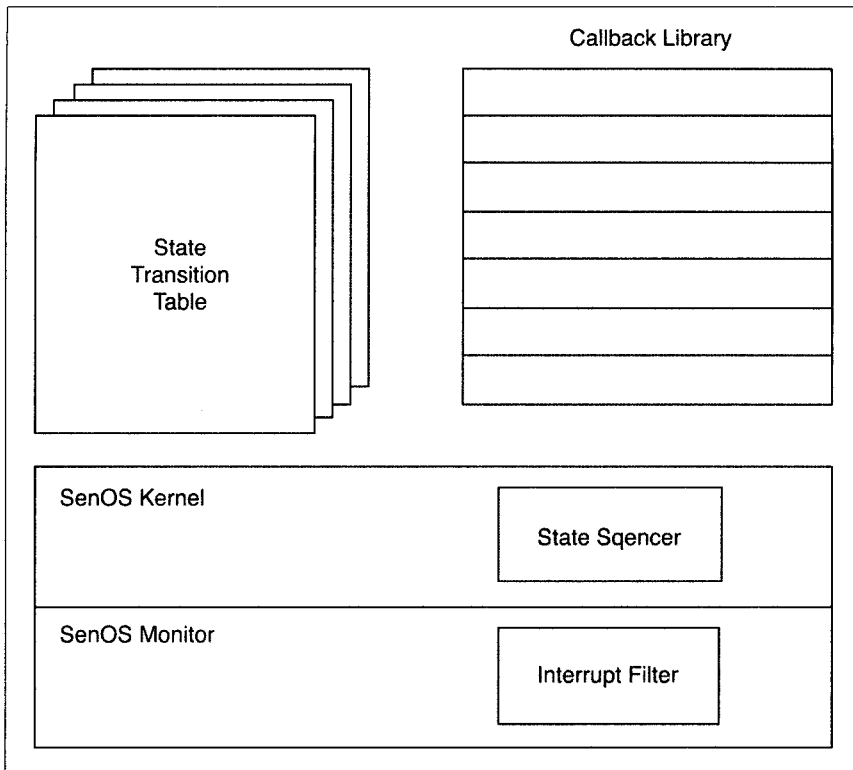
nesC에서 응용들은 잘 정의된 한 개 이상의 컴포넌트와 양방향성을 가지는 인터페이스로 만들어진다. 인터페이스는 인터페이스의 제공자가 구현할 Command 함수들의 집합과 인터페이스의 사용자가 구현할 Event의 함수들의 집합을 정의한다. 컴포넌트는 인터페이스들을 제공하고, 또한 이를 사용하는 개체이다. 컴포넌트는 실제 코드를 제공하는 Modules와 인터페이스를 통한 연결을 정의하는 Configurations의 형태로 구현된다. (〈그림 1〉)

Tiny OS는 타이머 혹은 네트워크 이벤트에 의해서 산발적(Sporadic)인 작업들로 구성된 테스크 큐에서 Non-Preemption으로 동작한다. 스케줄링은 FIFO로 진행되며, Non-Blocking I/O가 기본이다.

Tiny OS의 장점은 이벤트 기반의 작은 코드 사이즈로 다양한 시스템 리소스들을 효율적으로 제어할 수 있는 좋은 구조를 가지고 있다. 그러나, nesC를 통한 구조적인 장점은 시스템에서 제공해야 할 다양한 서비스의 측면에서 상당한 제한적인 요인으로 작용할 수 있으며, 다양한 센서용 하드웨어 플랫폼에 적용하는데 필요한 기본적인 포팅 과정 외에 nesC의 지원에 따른 상당한 작업이 필요하다. 그럼에도 불구하고 현재 여러 대학 및 연구소에서 가장 활발히 연구되고 있다.

## 2) SenOS

서울대, 한양대의 국내 대학을 중심으로 센서 네트워크 노드를 위해 유한상태 머신 (Finite State Machine) 을 기반으로 제안된 작은 운영체제의 모델이다. 센서 네트워크 노드의 응용 프로그램들은



(그림 2) FSM 기반 운영체제 구조

연속된 일련의 동작을 수행하거나 프로그램의 모드에 따라 입력 이벤트를 다르게 처리하는 경향이 있기 때문에 상태 머신(State Machine) 기반의 컴퓨팅 모델이 적합할 수 있다.[3] SenOS는 동시성과 반응성, 재구성의 요구조건을 모두 만족시키면서 동시에 코드의 크기와 전력 소모라는 제한 조건을 만족시키도록 디자인 되었다.

SenOS는 다음의 구성요소가 필요하다. (<그림 2>)



- 이벤트 큐로부터 입력을 받아들이는 상태 정렬기(State Sequencer)
- FIFO방식으로 동작하며 입력을 저장하는 이벤트 큐

- 출력 함수들을 담고 있는 콜백 함수 라이브러리
- 각 유효한 상태전이와 연관된 콜백 함수들을 정의하고 있는 상태 전이 테이블

커널은 입력 이벤트가 큐에 들어와 있다면 큐의 입력 이벤트에 따라 현재 상태를 전이시키고 관련된 출력함수를 호출한다. 콜백 라이브러리는 미리 정의된 시스템 함수들의 집합으로, 상태 전이 테이블이 런타임에 리로드 되거나 수정될 수 있는 반면에 커널과 콜백 함수 라이브러리는 정적으로 빌드되어 센서 노드의 플래쉬 롬에 저장된다. SenOS는 여러 개의 응용프로그램을 지원할 수 있으며, 상태 전이 테이블을 교체하는방법

으로 여러 개의 응용 프로그램을 동시에 수행시킬 수 있다.

SenOS는 FSM에 기반한 새로운 프로그래밍 모델 뿐만 아니라 센서 노드에 적합할 수 있는 명확한 제어 구조를 가지고 있다. 그러나, 센서 노드에서 일반적으로 사용하는 데이터와 텍스트가 분리된 Harvard 구조의 8bit MCU에서는 적합하지 않다. 텍스트 영역인 플래쉬 메모리로의 다운로드가 매우 제한적이기 때문에, FSM기반 커널 구조의 장점을 충분히 구현하기 위해서는 별도의 외부 하드웨어 장치의 도움을 받아야 한다. 이러한 몇 가지 제약사항에도 불구하고 FSM을 통한 운영체제의 구조는 매우 효율적이다.

### 3) Bertha

MIT를 중심으로 진행된 Pushpin 플랫폼을 위해 디자인된 작은 운영체제이다. Pushpin 플랫폼의 개념은 MEMS 기술의 발전으로 인해 초소형 모듈의 개발이 가능해졌으며 이를 페인트 칠을 하는 방식으로 작은 센서장치들을 건물의 벽이나 천장에 장착하는 개념에서 출발하였다.

Pushpin은 20MIPS (8051-core)의 마이크로 컨트롤러를 장착한 모듈로서 사전에 어떤 위치정보 없이 임의적으로 설치된 Pushpin들 간에 알고리즘적 자동 집결(Algorithmic self-assembly)의 개념을 구현하고, 운영체제를 통해 네트워크를 포함한 시스템 자원을 관리하고, 응용을 위한 IDE (Integrated Development Environment) 툴을 포함하는 프로그래밍 모델을 제공한다.

Bertha는 기능별로 다음과 같은 네 가지로 구성된다.



- Process Fragment Management Subsystem : 정형화된 Task(IPfrag)들의 관리
- Bulletin Board System Subsystem : 시스템 콜과 IPC를 위한 구조
- Neighborhood Watch Subsystem : 플랫폼간의 Pfrag의 이동을 관리
- Network SubSystem : 통신 프로토콜의 구현

Bertha OS의 가장 큰 특징은 2kBytes의 정형화된 태스크 추상화를 통해 센서 네트워크 상에서 동작 가능한 분산 컴퓨팅 프로그래밍 모델을 제공한다. 플랫폼간의 태스크의 이동은 이동상태 정보를 갱신하고 유지하는 과정을 통해서 이루어진다. 따라서, 네트워크와 매우 밀접하게 연동되어 동작하는 시스템이다. 그러나, 비교적 큰 코드사이즈와 8051-core의 Pushpin 플랫폼에 상당히 의존적 이어서 다양한 센서 노드 플랫폼에 적용하기에는 어려움이 있다.

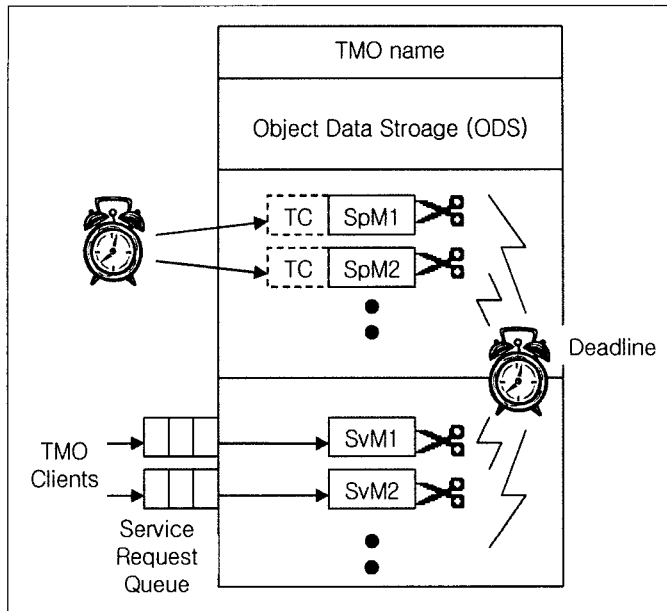
### 4) TMO

실시간 분산 객체 모델인 TMO(Time Triggered Message-Triggered Object)는 미국 UCI Dream Lab.에서 개발되었으며 실시간 태스크들의 적시 수행 보장(Timeliness guaranteed computing)을 목적으로 하고 있다. TMO는 시스템 설계 시 적시 수행 보장(Design time guarantee)을 제공하고 객체 모델이기 때문에 시스템 설계가 매우 용이해진다.

TMO는 실시간성이 요구되지 않는 병렬 컴퓨팅에도 적합한 구조를 가지고 있으며 IPC를 지원하여 분산된 TMO간의 통신도 가능하다.

TMO의 구조는 <그림 3>과 같다.

TMO는 개인 자료 저장소 (private ODS)와 메소



〈그림 3〉 TMO 구조

드들로 구성된다. 메소드들은 특성에 따라 SpM (Spontaneous Method)과 SvM(Service Method) 두 가지로 구분된다. SpM은 시간 조건에 따라 주기적으로 구동되는 메소드이며 SvM은 외부 메시지에 의해 구동되는 메소드이다. 시스템 설계시 SpM의 실행 주기를 지정하고 SpM과 SvM의 수행시간 데드라인(deadline)이 주어진다. 데드라인을 초과했을 경우 예외 처리를 할 수 있다. SpM 및 SvM은 다른 SvM에게 메시지를 송신하여 서비스를 요구할 수 있다. SpM과 SvM이 객체내의 공유 데이터에 동시에 접근했을 때의 우선 순위는 SpM에게 있다. 설계 시 적시 수행 보장을 위해 정한 규칙이며 BCC(Basic Concurrency Constraint s)라 한다. BCC가 필요없는 연성 실시간 (Soft Realtime) 시스템일 경우 공유 자원의 병행 접근 제어를 위해 CREW(Concurrent Read Exclusive Write) 모니터를 제공한다.

TMO는 센서 노드를 위한 모델은 아니다. 그러

나, 상당히 단순하면서 명확한 시스템 인터페이스를 통해 실시간성과 시스템의 예측가능성을 제공하는 좋은 구조를 가지고 있다.

### 5) Contiki

스웨덴의 SICS에서 작은 무선 네트워크 노드를 위한 8bit 마이크로 컨트롤러 용 General Purpose OS이다. Tiny OS와 마찬가지로 이벤트 기반으로 동작하는 시스템이며 주요한 특징은 런 타임에서 프로그램 코드를 다운로드하여 실행할 수 있는 구조를 가지고 있다. 이 기능은 무선 센서 네트워크의 스케일이 큰 시스템에서 관리적인 측면에서 큰 장점이 될 수 있다.

이외에도 preemptive multi-threading, TCP/IP Networking, Windowing System 과 같은 기존 운영체제에서의 서비스도 지원한다. 또한, Contiki의 장점은 플랫폼이 다양해짐에 따라 Common



Software Infrastructure를 제공함으로써 이식성을 높이는 중요한 이슈에 대해서 만족한다.

#### 4) 센서 네트워크 플랫폼

버클리 모트와 TinyOS는 시스템들의 이슈를 연구하고 초기에 실제 테스트 응용 애플리케이션을 배치하기 위해 널리 사용되고 있다. 마이크로컨트롤러는 적당한 양의 RAM과 프로그램을 위한 저장 공간을 제공하고 내부에 ADC를 포함한다. 넓은 주파수 범위의 모뎀을 가지고 있는 간단하면서 주파수에 민감한 무선 송수신 모듈은 개발자가 네트워크를 구성하기 위해 사용하는 연결성을 제공한다. 칩과 분리된 플래쉬 메모리는 네트워크를 통해서 그것이 전송하는 동안의 프로그램과 칩에 붙어있는 RAM을 초과하여 임시 저장하는 데이터 모두를 저장하기 위한 메모리를 제공한다. 다수의 센서 보드들은 이러한 플랫폼을 위해 고안되어 왔다. TinyOS를 지원하는 하드웨어 플랫폼인 MOTE는 1999년부터 다양한 형태로 연구 개발되어 왔다.

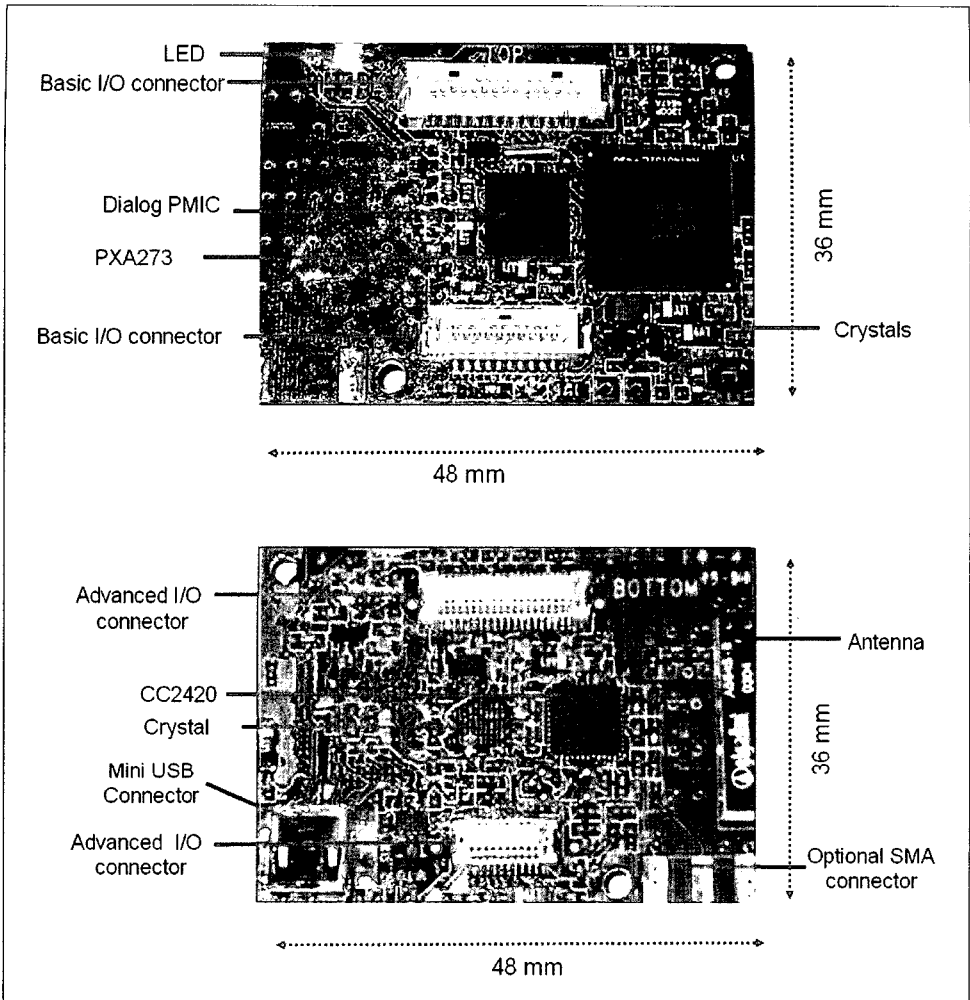
인텔 아이모트는 기능이 높은 ARM 마이크로 프로세서와 저장공간이 하나의 패키지로 결합되어진 무선 송수신기를 함께 탑재하고 있는 상용화된 칩을 사용하여 최근에 완성된 플랫폼이다. 무선 송수신기

는 노트북과 휴대폰에 널리 적용되고 있는 블루투스 표준을 적용시켰다. 무선 통신은 높은 범위에서 동작하고 정교한 주파수 호핑 프로토콜을 가지고 있다. ARM 프로세서는 블루투스 라디오를 운영하고 무선 통신 모듈부터 시리얼 포트까지 패킷을 전달하는 기능을 한다. 아이모트에서 동작하는 TinyOS는 다양한 센서들과 라우팅 기능들을 제공하고 상위레벨의 정보를 처리하고 전력소비를 관리하는 하나의 통합된 시스템을 제공하면서 ARM 프로세서를 직접 구동한다. 낮은 수준의 TinyOS 컴포넌트들의 대부분은 하드웨어에 직접 연동된다. 특징은 스마트 테스트 프로젝트에서 개발된 저전력의 ADC와 효율적인 라디오 모듈을 포함한다. 전체적인 디자인은 단지 5 제곱 밀리미터의 영역만을 차지한다. 그것은 1% 활성화 비율로 AA 배터리 두개를 사용하면 100년 동안 동작할 수 있다고 평가된다.

센서 네트워크를 인터넷과 같은 다른 네트워크로 전달해 주기 위해 다른 한쪽 끝단에서의 노드들은 IEEE 802.11같은 넓은 범위의 라디오나 휴대폰의 모뎀을 가지고 있고 리눅스 같은 운영체제에서 동작하는 32-bit 프로세서 기반의 게이트웨이 장치들과 통합된다. 이러한 노드 그룹은 대부분의 분산 시스템에서 매우 중요한 역할을 할 것이다. 이러한

(표) Mote 하드웨어 플랫폼 종류

Mote	WeC	rene	dot	mica	mica2	mica2dot	iMote	btNode
년도	1999	2000	2001	2002	2003	2003	2003	2003
프로세서 속도	4 MHz				7 MHz	4 MHz	12 MHz	7 MHz
플래쉬(KB)	8	8	16	128	128	128	512	128
램(KB)	0.5	0.5	1	4	4	4	64	4
라디오(kBaud)	10	10	10	40	40	40	460	460
라디오종류	RFM				ChipCon	ChipCon	ZeevoBT	EricsonBT
마이크로컨트롤러	Atmel						ARM	Atmel
확장	불가능	가능	불가능	가능	가능	가능	가능	가능



〈그림 4〉 Intel의 iMote2 플랫폼

노드들은 그 시스템을 수행하고 형성하고 감시하기 위한 응용 서비스에서 네트워크 밖의 데이터를 검색하기 위한 게이트웨이로 동작할 것이다. 좀 더 정교한 이질적인 시스템에서는 이러한 게이트웨이 노드들은 좀 더 넓게 분포되어 지고 데이터를 모으고 저장하고 융합하고 정밀 센서들을 위한 호스트처럼 사용된다. 게이트웨이 들은 많은 에너지를 사용하기 때문에 태양판 처럼 재충전되는 형태들과 대용

량의 배터리를 이용하여 동작하게 된다. 공장 같은 경우에는 시설물 또는 건축물의 전원을 이용할 수 있을 경우 안정적인 전원을 끌어올 수 있다.

### 5. 자율 구성 네트워크

하나의 네트워크는 다른 노드들에 연결하는 다중의 링크들을 가지고 각각 많은 노드들로 구성된다.

정보는 생성지점부터 사용되는 지점까지 하나의 라우트를 통해 흡에서 흡으로 이동한다.

인터넷 같은 서로 연결된 네트워크에서 각각의 라우터는 라우팅 그래프를 형성하면서 다른 라우터들의 하나의 구체적인 집합에 연결한다. 무선 센서 네트워크에서 각 노드들은 노드 근체에 통신링크들의 한 집합을 제공하는 무선 송수신기를 가지고 있다. 정보를 교환함으로써, 노드들은 그들의 이웃 노드를 발견하고 응용 애플리케이션의 요구에 따라 데이터를 라우팅하는 방법을 결정하는 분배 알고리즘을 수행한다. 비록 물리적인 위치가 연결성을 주로 결정하지만 장애물이나 간섭이나 환경적인 요소, 안테나 방향 그리고 이동성 같은 변수들은 연결의 우선순위를 결정하기 어렵게 만든다.

**1) 무선 네트워크 연결성**

무선 센서네트워크의 네트워크 기능은 레이어를 나누어 처리하여 성능을 향상시킬 수 있다. 가장 낮은 레이어는 물리적인 라디오 장치를 제어한다. 라디오 들은 전송개체 제어가 자연에 의해 이루어지는데, 하나의 노드가 전송할 때 다른 노드들의 집합은 신호를 수신한다. 그렇지 않으면 동시에 다른 노드가 전송하는 신호에 의해 잘못 선택되어질 것이다. 라디오 전송 채널에서 충돌을 피하기 위해 링크 레이어는 채널을 청취하고 전송 채널에 다른 무선 전송이 없을 때만 전송한다. 그것은 라디오 신호에서 부호화된 패킷으로부터 구조적인 비트들의 배열로 전송한다.

전송하지 않을 때, 노드들은 채널을 샘플링하고 수신기가 송신자의 시간을 가지고 노드 자체를 배열하는 것을 허용하는 패킷의 시작부분에서 특정한 데이터 수신을 위해 항상 수신을 수행한다. 해당 패킷 레이어는 버퍼들을 관리하고, 라디오 장치에서 패킷들의 일정을 정하고 오류를 검출하거나 정정하

고, 패킷 손실을 조절하고 시스템이나 응용 애플리케이션에 패킷들을 신속히 처리한다.

**2) 분산 네트워크 및 데이터 수집**

센서 네트워크 사용자들은 노드들의 집합이 노드들의 동작을 통합하고 데이터 정보를 전송, 처리하는 것을 허용하는 프로토콜을 수행하기 위해 이러한 기초적인 통신 능력을 사용한다. 센서 네트워크에서 기초적인 무선 전송 능력은 노드들 사이에서 확산되는 데이터 특성과 연관된다. 이것은 하나의 기본 노드가 약간의 구별되는 정보를 가진 패킷을 전송하는 위치에서 유동적인 프로토콜에 의해 수행되어진다. 수신하는 노드들은 패킷을 재전송하고 그 결과 좀 더 멀리 떨어진 노드들은 그것을 수신할 수 있다. 그러나 한 노드는 다수의 주변 노드들로부터 같은 메시지의 다른 버전을 받을 수 있고, 그래서 그 네트워크는 중복된 것을 알아내고 금지시키기 위한 구별되는 제어 정보를 사용한다. 프로토콜이 동작하는 것은 충돌을 피하고 불필요한 전송을 없애는 다양한 기술들을 사용한다.

네트워크는 명령어를 전송하고, 경로를 통해 네트워크 제어를 수행하고 형성하기 위해 분배과정을 사용한다. 또한 라우트를 형성하기 위해서도 분배과정을 사용한다. 각각의 패킷은 기초되는 노드로부터 그것의 거리와 송신기를 구분한다. 분산되어진 라우트 트리(route tree)를 형성하기 위해 노드들은 그 루트 근처의 노드 하나를 정의해서 기록한다. 네트워크는 원래의 노드로 거슬러 올라가거나 그 트리 각각의 레벨에서 데이터를 처리함으로써 데이터를 모으기 위하여 이러한 역방향의 통신을 사용할 수 있다.

그 기본 노드는 일부의 고 레벨 작업에 의해 결정되어지는 것처럼 좀 더 강력한 네트워크나 센서네트워크 내의 모이는 접점에서 게이트웨이가 된다.

종종 트리구조와 데이터 수집은 연관되어 있다. 데이터는 부모 노드가 발견되면 그 트리를 따라서 동작을 시작할 수 있다. 노드들은 데이터 메시지를 수신하면서 부모 노드가 될 노드들을 정하게 된다. 네트워크는 최선의 라우팅을 강화하기 위해 통계수치를 계속적으로 모은다.

이러한 통신 패턴은 많은 클라이언트 컴퓨터들이 연결을 열어놓고 많은 양의 데이터를 주고받는 인터넷에서 사용되는 방식들과는 다르다. 센서 네트워크에서는 통신은 전체로서 수행되고 참가자들은 물리적 위치나 센서 값들의 범위 같은 특성에 의해 구별되어 정의된다. 라우팅은 노드들이 특성에 의해 데이터에 관심을 표현하는 과정인 유도 확산 과정처럼 형성된다. 그 노드들은 라우팅 도표를 형성하기 위해 관심있는 데이터 종류를 외부방향으로 전송하고, 그것들은 결합되어진 하부구조에서 관심도를 기반으로 기울기 데이터를 연산하고 이것을 증가시키는 방향으로 데이터를 수집한다.

신뢰도는 다양한 패턴에 따라 나타나는데, 센서 네트워크는 신뢰성을 결정하는 데이터 근원지와 전달 목적지 사이의 패킷 매칭이나 바이트를 사용하는 두 지점간의 연결을 설정하는 인터넷과 대조적으로 그것들이 확실하게 홉에서 홉으로 데이터를 전송하는 위치에서 효율적인 네트워크 접근법을 배치해야 한다.

### 3) 무선 통신 에너지 절약

보통 하나의 노드를 수행하기 위해 가장 많은 에너지를 소모하는 동작인 통신은 반드시 여러 노드들 간의 제한된 통신영역의 공유를 위해 경쟁해야만 한다. 네트워크 스택은 통신을 제거함에 의하거나 통신의 필요성이 없을 때 라디오의 전원을 꺼놓고 에너지 사용을 최소화하기 위한 시도를 한다.

다수의 접근법은 가능하다. 예를 들면 노드들은

데이터를 그 위치에서 처리할 수 있고 그것들이 원하는 이벤트를 발견할 때만 통신을 수행한다. 이런 접근법은 데이터 수집을 시간이나 원하는 영역에 집중하는 환경 감시 시스템이나 지능 경보 시스템에 사용된다.

대부분의 경우 카메라처럼 고 전력의 센서 장치들을 동작시킨다. 네트워크 내에서 집합을 형성하는 것은 통신의 기능을 낮출 수 있다. 예를 들면 응용 애플리케이션은 어떤 지리적 영역에서 공유된 노드들에서 평균온도를 결정하기 위한 데이터가 필요할 것이다. 데이터를 모으는 라우팅처럼 관심대상의 감지 데이터들의 수집을 위해 하위집합을 모으는 것은 라우팅 트리의 종단에서 수행되어질 수 있고, 그 결과 각각의 노드들은 그 하부구조의 통계적인 요약을 제공하기 위해 기껏해야 하나의 패킷만을 전송한다. 좀 더 정교한 집합은 관심대상의 분포영역에서 감지 데이터와 연관된다.

압축과정은 낮은 레벨에서 에너지를 보존할 수 있다. 약간의 프로토콜에서의 부담은 신뢰성을 증가시키고 충돌을 관리하고 라우팅 구조를 유지하기 위한 데이터 통신과 연관된다. 센서 네트워크는 데이터 메시지의 제어정보를 편송하거나 다른 노드들을 위해 할당된 패킷들을 수신하면서 프로토콜 메시지를 해석할 수 있다. 그것들은 경쟁을 제거하기 위해 미리 예정된 시간을 사용할 수 있고, 그때 라디오는 계속 동작을 한다. 이것은 주기적으로 낮은 비율의 샘플링에 의해서 높은 레벨의 응용 애플리케이션을 가지고 통합되어 질 수 있다. 일반적으로 시분할 방식에서 네트워크는 낮은 레이어 내에서 에너지 소모 감소 모듈을 구현한다.

공간차원에서 네트워크는 재전송이나 그룹 정의 같은 특정 노드에 대해 구체적인 의무를 할당할 수 있다. 결국 네트워크는 단지 한 부분 동안만 수신한

후에 라디오 전원을 차단함으로써 원하지 않는 패킷 수신을 거부할 수 있다. 그러나 이러한 많은 최적화가 상호 충돌할 수 있기 때문에 프로토콜의 많은 기능 추가는 응용 애플리케이션과 플랫폼의 특징을 기반으로 고려되어야 한다.

## 6. 데이터 보호

고밀도의 네트워크, 실시간 처리, 네트워크 내의 데이터 처리는 대규모의 물리 구조 전체에서 무엇이 일어나고 있는지를 감지할 수 있는 능력에서 기존의 환경과는 많은 차이점을 갖고 있다. 환경 감시와 상황에 근거한 모니터링에서는 데이터 수집의 목적과 데이터를 사용하기 위한 모임들과 분배범위는 명확하다. 집, 근무지, 수송 터미널, 쇼핑 같은 좀더 일반적인 인간 행위가 나타나는 곳에서 우연한 설정에서 모호하게 된다. 이러한 경우에 많은 잠재적으로 관심을 가지고 있는 데이터 그룹은 데이터를 위해 다양한 사용을 할 수 있다.

점유나 움직임, 심지어 물리적인 상태 같은 세밀한 센싱은 감지와 분배 뿐 아니라 데이터 내용에 대한 관심을 더욱 증폭시킨다. 오늘날 많이 사용되는 감시카메라에 의해 수집된 영상 데이터는 매우 강력한 센서처럼 보일 수 있지만, 네트워크 접근과 자동화되어진 분석은 제한된다. 이러한 사회적 요인

들은 센서 네트워크 기술과 연관되어 왔다. 이러한 영역은 초기 상태부터 활발히 연구 되었다.

## III. 결론

센서 네트워크 기술은, 작고 단순한 하드웨어를 기반으로 소형 소프트웨어 개발을 지향하고 있기 때문에 고급 기술 창조를 통한 신산업의 창출이라는 기술적 기대를 실현하기에는 구현 난이도에 대한 회의적 견해가 많은 것 같다. 그러나 기술적 난이도와는 상관없이 새로운 컴퓨팅 개념의 도출과 실제 구현을 지향하는 기술 개발 방향은 매우 긍정적으로 평가받을 수 있는 추진 방향이다.

센서 네트워크 같은 저전력 무선 네트워크가 실제 산업과 생활에 적용되고 상용화되기 위해서는 안정되고 상호운용이 가능한 네트워크 프로토콜의 실제 구현이 요구되는데, 이런 요구사항을 해결할 수 있는 방법은 지그비 규격의 제품이 개발되거나 소형 OS를 기반으로 한 기술구현 등으로 예상된다.

센서 네트워크의 다양한 산업에의 적용은 관련 산업화를 가속화 할 것으로 기대된다. 단기적으로는 무선 네트워크 시장의 확대를 가져올 것이며, 장기적으로는 유비쿼터스 컴퓨팅에서의 끊임없는 네트워크를 구성할 수 있는 기반 기술로 사용될 것이다.

### 참고 문헌

- [1] 권수갑, "Ubiquitous Computing 개념과 동향", pp. 8~11, 전자부품연구원 전자정보센터, 2003년 3월.
- [2] TinyOS, <http://www.tinyos.net/>
- [3] S. Hong and T.-H. Kim, SenOS: State-driven Operating System Architecture for Dynamic Sensor Node Reconfigurability. International Conference on Ubiquitous Computing (ICUC). pp. 201-203, Seoul, Korea, October, 2003.
- [4] Contiki, <http://www.sics.se/~adam/contiki/index.html>

- [5] Pushpin, [http://web.media.mit.edu/~lifton/Push pin/](http://web.media.mit.edu/~lifton/Push%20pin/)
- [6] TMO, <http://dream.eng.uci.edu>
- [7] M.Angels Moncusi and Alex Arenas, Jesus Labarta, "A Modified Dual-Priority Scheduling Algorithm for hard real-time systems to improve energy savings." Compilers and Operating System for low power, Kluwer Academic Publishers. pp.17-36, Boston, USA, 2003.
- [8] <http://www.intel.com/research/exploratory>
- [9] <http://www.chipcon.com>
- [10] <http://www.moteiv.com>
- [11] J. Hill, R. Szweczyk, A. Woo, S. Hollar, D. E. Culler and K. S. J. Pister, System architecture directions for networked sensors, Proceedings of ASPLOS, pages 93-104, Boston, MA, USA, Nov. 2000.
- [12] [http://www.intel.com/research/vert\\_manuf\\_condmaint.htm](http://www.intel.com/research/vert_manuf_condmaint.htm)
- [13] David Culler, Deborah Estrin, Mani Srivastava, Overview of Sensor Networks, IEEE Computer, pages 41-49, Aug 2004.
- [14] [http://today.cs.berkeley.edu/retreat-6-04/buonadonna-NEST\\_06\\_04.ppt](http://today.cs.berkeley.edu/retreat-6-04/buonadonna-NEST_06_04.ppt)
- [15] Robert Szweczyk, Joe Polastre, Alan Mainwaring, John Anderson, David Culler, An Analysis of a Large Scale Habitat Monitoring Application, ACM Conference on Embedded Networked Sensor Systems, Nov 2004.

## 필자 소개



### 강 정 훈

- 1997년 2월 : 단국대학교 전자공학과 (학사)
- 1999년 2월 : 단국대학교 전자공학과 (석사)
- 1999년 ~ 현재 : 전자부품연구원 선임연구원
- 주관심분야 : 센서 네트워크, 네트워크 임베디드 시스템, 스마트 스페이스 미들웨어, 상황인지 미디어 서비스



### 황 태 호

- 1998년 2월 : 한국외국어대학교 컴퓨터공학과 (학사)
- 2000년 2월 : 한국외국어대학교 컴퓨터공학과 (석사)
- 2000년 ~ 현재 : 전자부품연구원 선임연구원
- 주관심분야 : 센서 네트워크, 네트워크 임베디드 시스템, 스마트 스페이스 미들웨어



### 송 병 철

- 1994년 2월 : 명지대학교 전자공학과 (공학사)
- 1996년 2월 : 명지대학교 전자공학과 (공학석사)
- 1996년 3월 - 현재 : 전자부품연구원 책임연구원
- 주관심분야 : 디지털통신네트워크, 센서 네트워크, 유비쿼터스 시스템