

# 센서 네트워크 기반 강의실 제어시스템 구현

빈기철<sup>†</sup>, 이종민<sup>\*\*</sup>, 권오준<sup>\*\*\*</sup>

## 요 약

최근 들어 센서 네트워크에 대한 관심이 증가됨에 따라서 다양한 분야에서 이를 활용한 연구가 이루어지고 있다. 본 논문에서는 센서 네트워크 기반 강의실 제어 시스템을 제안하고 구현한다. 강사의 강의 이력 정보를 센서 네트워크를 통하여 수집하고, 이 정보를 활용하여 강사가 강의실에 들어갈 때 강의실 내 PC, 빔 프로젝터, 전등 등을 강의에 적합한 형태로 제어해 주면 효과적인 강의 진행이 가능하다. 본 논문에서 제안한 강의실 제어 시스템이 효과적으로 동작할 수 있도록 센서 네트워크를 구축하고 성능 실험을 한다. 이를 통하여 강의실에 설치한 센서 노드의 메시지 발생 주기와, 센서 노드 간 통신이 원활하게 이루어질 수 있도록 강의실 내 배치된 센서 노드의 수를 최적화할 수 있는 값을 구하였다.

## Implementation of A Lecture Room Control System Based on a Sensor Network

Kicheol Bin<sup>†</sup>, Jong Min Lee<sup>\*\*</sup>, Oh Jun Kwon<sup>\*\*\*</sup>

## ABSTRACT

As the interest in sensor networks has been increased recently, researches have been performed in many areas using sensor networks. In this paper, we propose and implement a lecture room control system based on a sensor network. It is possible to give an efficient lecture by collecting lecturers' lecture history information through the sensor network and controlling a PC, a beam projector, lights, etc. in a lecture room appropriately using the information. We construct a sensor network and perform experiments so that the proposed lecture room control system performs efficiently. Through the experiments, we obtained the optimal message generation period of the sensor node in a lecture room and the optimal number of sensor nodes deployed in a lecture room to communicate without any trouble between sensor nodes.

**Key words:** sensor network(센서 네트워크), lecture history(강의 이력), Contiki(Contiki), uIP protocol (uIP 프로토콜)

## 1. 서 론

최근 들어 센서 네트워크에 대한 관심이 많이 증가됨에 따라서 물류, 군사, 생활, 시설물 및 농작물 관리 등 다양한 분야에서 응용과 연구가 이루어지고 있다. 센서 네트워크를 사용한 대표적인 예로 헬스

케어[1], 차량 추적 시스템[2], 생태계 감시, 가정 내 가전제품 제어를 위한 홈 네트워크 등이 있다. 센서 네트워크 시스템은 자연에서 발생하는 소리나, 빛, 온도, 움직임 등과 같은 이벤트를 감지하며 측정하고, 누적된 많은 정보들을 싱크(sink) 역할을 하는 노드에 전달을 하여 그 정보를 얻는다. 센서 네트워

※ 교신저자(Corresponding Author) : 이종민, 주소 : 부산광역시 부산진구 엄광로995(613-713), 전화 : 051)890-1736, FAX : 051)890-2629, E-mail : jongmon@deu.ac.kr  
접수일 : 2008년 10월 29일, 완료일 : 2008년 12월 26일  
<sup>†</sup> 준회원, 동의대학교 대학원 컴퓨터응용공학과 (E-mail : kchpin@deu.ac.kr)

<sup>\*\*</sup> 정회원, 동의대학교 컴퓨터소프트웨어공학과 부교수  
<sup>\*\*\*</sup> 종신회원, 동의대학교 컴퓨터소프트웨어공학과 부교수 (E-mail : ojkwon@deu.ac.kr)  
※ 이 논문은 2006학년도 동의대학교 교내 연구비에 의하여 연구되었음 (과제번호 : 2006AA170)

크를 구성하는 노드들은 데이터를 수집하는 기능과 수집된 자료를 처리하는 기능, 센서 노드 간의 데이터 전송을 위한 통신 기능 등으로 구성되어 있다.

기술의 발달과 더불어 강의실은 많은 변화를 나타내고 있다. 기존의 강의실은 책상과 판서를 할 수 있는 칠판, 야간에도 강의를 할 수 있는 전등 등이 전부였다. 그러나 현대의 강의실은 과거의 강의실과는 비교가 되지 않을 정도로 많은 변화를 이루고 있으며, 미래에는 보다 더 나은 멀티미디어 시스템으로 변화할 것이다. 강의를 효과적으로 하기 위하여 PC, 빔프로젝터, 오디오/비디오 시스템 등 다양한 멀티미디어 시스템을 강의에 잘 활용해야 한다. 그러나 이러한 멀티미디어 시스템을 강사가 강의를 진행하면서 동시에 효과적으로 제어하는 것이 쉽지 않다. 따라서 센서 네트워크를 통하여 강사의 강의 이력 정보를 수집하고, 이를 활용하여 강사가 강의실에 들어갈 때 강의실 멀티미디어 시스템을 강의에 적합한 형태로 제어해 주면 효과적인 강의 진행이 가능할 것으로 기대된다.

본 논문에서는 이를 위하여 강사에 대한 강의 이력을 관리하고 강의실 장비를 강의시간에 맞추어 원격으로 제어할 수 있는 센서 네트워크 기반 강의실 제어 시스템을 제안하고 구현한다. 강의실 제어 시스템을 구현하기 위하여 하이버스(주) Hmote2420[3]과 Contiki 운영체제[4]를 사용하여 센서 네트워크를 구축한다. Hmote2420은 TI MSP430 마이크로 컨트롤러를 사용하며, IEEE 802.15.4 스택을 사용한 통신을 지원한다. Contiki는 이식성 높은 운영체제로써 2 킬로바이트의 램(RAM)과 40 킬로바이트의 롬(ROM)으로 운영되는 강력한 초소형 임베디드 운영체제이다. 두 개의 통신 스택(Communication Stack)인 uIP와 Rime을 지원하며, uIP는 TCP/UDP 통신을 지원하여 인터넷 연결을 쉽게 하고 Rime은 센서 네트워크를 위한 통신 모듈로서 인접한 이웃 노드와의 통신, 방송(broadcast), 플러딩(flooding) 등을 쉽게 할 수 있는 장점이 있다.

본 논문에서 제안한 강의실 제어 시스템이 효과적으로 동작할 수 있도록 센서 네트워크를 구축하고 그에 따른 성능 실험을 한다. 강의실에 설치된 센서 노드의 메시지 최적 발생 주기를 찾기 위한 실험과 강의실 내 센서 노드의 수를 최적화하기 위한 실험을 수행하여 강의실 제어 시스템 내 센서 노드 간 통신

이 원활히 잘 이루어질 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 기술한다. 3절과 4절에는 본 논문에서 제안하는 강의실 제어 시스템과 이에 대한 구현 및 성능 실험에 대하여 각각 기술한다. 마지막 5절에서는 결론을 기술한다.

## 2. 관련 연구

본 절에서는 센서 네트워크를 구성하는 센서 노드 하드웨어 플랫폼과 센서 노드 운영체제에 대하여 기술한다.

### 2.1 센서 노드 하드웨어 플랫폼

센서 노드 하드웨어 플랫폼은 미국 국방성의 지원을 받아 개발된 버클리 대학의 Mote 시리즈를 상업화한 Crossbow Mica 시리즈[5], 인텔 Mote(iMote), Telos[6] 등이 현재 많이 사용되고 있다. Mica 시리즈는 4~8 MIPS 성능의 8비트 RISC 마이크로컨트롤러인 Atmel의 ATMega 시리즈를 사용하며, 메모리는 4KB SRAM과 128KB 플래시 메모리를 사용한다 [5]. Mica는 916MHz의 무선 트랜시버를 사용하지 않, Mica2는 433/868/915MHz의 다양한 무선 밴드를 지원한다. MicaZ는 Mica2와 비슷하지만 IEEE 802.15.4 저전력 WPAN (Wireless Personal Area Network) 표준을 지원하는 RF 모듈인 Chipcon사의 Zigbee 칩 CC2420을 사용한다.

인텔 Mote (iMote)는 버클리 대학의 산업체 파트너였던 인텔사에서 개발하였으며, ARM 기반의 마이크로프로세서인 32비트 ARM7TDMI 코어를 사용한다. 12MHz의 클럭으로 기존 Mica Mote보다 빠른 성능을 보이며, 메모리 측면에서도 Mica보다 큰 512KB의 플래시 메모리와 64KB SRAM을 사용한다. RF 모듈은 Zeevo사의 2.4GHz 대역폭의 블루투스 사용하여 최대 720kbps의 전송률을 가진다.

본 논문에서 제안하는 시스템 구현시 사용하는 하이버스(주) Hmote2420은 Telos 플랫폼[6]으로, TI MSP430 마이크로 컨트롤러와 CC2420 RF칩을 사용하여 저전력으로 장기간 사용 가능하다. 그림 1은 본 논문에서 사용된 Hmote2420 센서 노드를 보여준다. 내장PCB 안테나, 외장 확장 안테나 단자 및 다양한 센서 확장을 위한 확장 포트를 제공하고 USB를 통

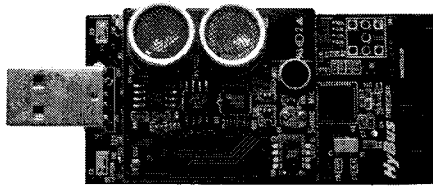


그림 1. Hmote2420 센서 노드

한 프로그래밍 및 저전력 전원 관리 기능, 온도, 조도, 습도, 마이크 등을 사용하여 다양한 종류의 데이터 측정이 가능하다.

### 2.2 센서 노드 운영체제

센서 노드에 사용되는 운영체제는 센서 노드의 제한된 컴퓨팅 환경을 고려하여 응용 프로그램을 개발할 수 있어야 한다. 이러한 센서 네트워크용 운영체제 중 대표적인 것으로는 버클리(Berkeley)의 TinyOS[7], FreeRTOS[8], Contiki[4] 등이 있고 국내에서는 한국전자통신연구원(ETRI)에서 개발한 Nano-Qplus가 있다

TinyOS는 버클리 대학에서 진행했던 WEBS (Wireless Embedded System) 프로젝트에서 개발된 초소형 무선 임베디드 네트워크 운영체제로, 오픈 소스이며 컴포넌트 구성 요소에 기반한 이벤트 구동(event-driven) 방식으로 동작한다. 상태 머신 프로그래밍 모델을 기반으로 하여 전체 시스템은 여러 상태 머신들로 구성되어 있다. TinyOS는 동적 메모리를 할당하기 위하여 nesC (Network Embedded System C)라는 새로운 언어를 통하여 프로그래밍을 한다. 또한 간단한 FIFO 스케줄러를 사용하기 때문에 실시간성을 고려해야 하는 프로그래밍을 하기가 매우 어렵고, 복잡한 응용 프로그램을 작성하는 데는 많은 어려움이 있다.

FreeRTOS는 소형 임베디드 시스템을 위해 설계된 확장 가능한 실시간 커널이다[8]. 많은 프로세서에서 포팅이 되어 있고 무료 개발 도구가 많이 있어 쉽게 사용 가능하며, 실행시 강력한 추적 기능을 지원하며, 스택 오버플로우 탐지 기능이 있다. 거의 모든 코드는 C로 작성되었으며 몇몇 함수만이 어셈블러로 되어 있다. 태스크의 우선순위가 높은 순서대로 실행을 한다.

본 논문에서 제안하는 시스템 구현에 사용하는 Contiki 운영체제는 거의 대부분 C언어로 기술되어

있으며, 이식성이 높고, 멀티태스킹과 효율적인 메모리 관리가 가능한 임베디드 운영체제이다[4]. 2 킬로바이트의 램(RAM)과 40 킬로바이트의 롬(ROM)으로 운영되고, 두 개의 통신 스택인 uIP와 Rime을 지원한다. uIP는 TCP/UDP 통신을 지원하여 인터넷 연결을 쉽게 한다[9]. Rime은 센서 네트워크를 위한 통신 모듈로서 저 전력 통신을 위해 구현된 멀티 홉 데이터 전송을 지원하며, 인접한 이웃 노드와의 통신, 방송(broadcast), 플러딩(flooding) 등을 쉽게 할 수 있다[10]. Contiki는 기본적으로 TinyOS와 같이 이벤트 구동 방식 모델을 따르고 있지만 Protothreads라는 기술을 통해 제한적이기는 하지만 스택 예약 없이 멀티쓰레드와 같은 코드 간의 동기화를 제공한다[11]. Contiki는 netsim 시뮬레이터를 사용하기 위한 컴파일이 가능하며 그와 별도로 COOJA라는 Contiki 운영체제를 위한 시뮬레이터를 제공한다. COOJA는 자바 기반의 크로스 레벨(cross level) 시뮬레이션이 가능한 시뮬레이터로, 응용 수준, 운영체제 수준, 하드웨어 수준 등 여러 수준에 따른 시뮬레이션이 가능하다.

## 3. 시스템 기술

### 3.1 시스템 구성

본 논문에서 제안하는 강의실 제어 시스템은 그림 2와 같이 구성된다. 강의실 제어 시스템을 구성하기 위한 센서의 종류는 크게 강의실 센서 (RS : Room Sensor), 강사 센서 (LS : Lecturer Sensor), 터미 센서(DS : Dummy Sensor)로 나누어진다. 특정 강의실에서 강사의 존재 여부를 파악하기 위하여 강의실마다 하나의 강의실 센서(RS)가 있고, 강사는 각자 자신의 강사 센서(LS)를 가지고 다닌다고 가정한다. 강의실 제어 시스템은 크게 특정 강의실 센서에서

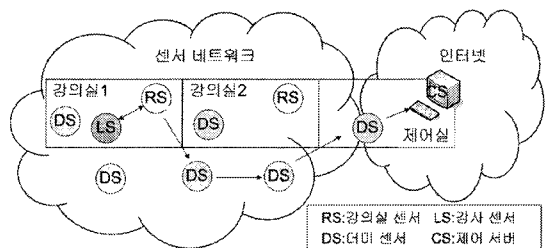


그림 2 . 강의실 제어 시스템

강사 센서의 존재 여부를 파악하는 단계와 강의실 센서에서 획득한 강사 정보를 제어 서버로 전송하여 강의 이력과 관련된 정보를 관리하는 단계가 있다.

일반적으로 강의실 센서의 무선 도달 거리가 제한적이므로 제어 서버까지의 연결 통로를 구성하기 위하여 더미 센서를 강의실 주변에 적절히 배치하여 센서 네트워크를 구성한다. 구축된 센서 네트워크는 임의의 더미 센서를 게이트웨이로 사용하여 인터넷에 연결된다. 제어 서버는 인터넷에 연결되어 있는 PC에서 수행되며, 여러 강의실 센서에서 전송되어 온 정보를 기반으로 하여 특정 강사에 대한 강의 이력 정보를 관리한다. 센서 네트워크를 구축하는 강의실 센서와 강사 센서 간에는 Contiki에서 지원하는 Rime 프로토콜[10]을 사용하여 이웃 노드 간 통신을 한다. 강의실 센서와 제어 서버 간에는 Contiki에서 지원하는 uIP 프로토콜[9]을 사용하여 UDP 통신을 한다. Contiki에서 지원하는 uIP 프로토콜을 사용함으로써 게이트웨이에서의 변환 절차 없이 센서 네트워크 외부에 존재하는 서버와 자유롭게 통신 가능한 장점이 있다.

강의실 센서는 강사의 존재 여부를 파악하는 기능과 이 정보를 제어 서버로 전송하는 기능을 가진다. 강사가 지닌 강사 센서의 존재 여부를 파악하기 위하여 강의실 센서는 주기적으로 자신의 존재 여부를 이웃한 센서 노드에 방송한다. 강의실 센서의 방송 메시지를 수신한 강사 센서에서는 이에 응답함으로써 자신의 존재 여부를 알린다. 강사 센서에 대한 정보를 획득한 강의실 센서는 현재 강의실 온도와 습도 정보 등을 같이 제어 서버로 전송한다. 강의실 센서는 강사 센서와의 통신을 위하여 Rime 통신 프로토콜을 사용하고, 제어 서버와의 통신을 위하여 uIP 통신 프로토콜을 사용한다.

강사 센서는 주기적으로 방송되는 강의실 센서의 방송 메시지를 수신하면 이에 응답함으로써 자신의 존재 여부를 알린다. 두 개 이상의 강의실 센서에서 방송되는 광고 메시지를 수신하는 경우에는 이중 수신 감도가 큰 강의실 센서에겐만 응답함으로써 특정 시간에 한 강의실에만 있음을 보장할 수 있도록 한다. 강사 센서는 강의실 센서와의 통신을 위하여 Rime 통신 프로토콜을 사용한다.

더미 센서는 강의실 센서와 제어 서버 간의 통신 경로를 보장해 줄 수 있도록 임의의 위치에 배치한

다. 센서 노드의 무선 도달 거리는 전파 특성상 한계가 있으므로 더미 센서를 적절히 배치하여 다중 홉 네트워크를 구축하여 제어 서버로 갈 수 있는 통신 경로를 만들어 주어야만 네트워크가 정상적으로 동작하게 된다.

제어 서버에서는 일정 시간 동안 특정 강의실 센서에서 특정 강사 센서에 대한 정보를 전송할 경우 해당 강의 시간에 강의가 진행되는 것으로 인식하고 그 강사에 대한 강의 진행 이력을 관리한다. 또한 제어 서버에서는 누적된 강의 이력 정보를 사용하여 특정 강사가 강의하는 시간에 맞추어 강의실에 있는 PC, 빔 프로젝터, 비디오/오디오 시스템, 냉난방기, 전등 등 강의와 관련된 장비를 원격으로 제어할 수 있도록 서비스함으로써 강의가 원활하게 진행 가능하게 한다.

### 3.2 통신 프로토콜

강의실 제어 시스템을 구축하기 위하여 사용된 통신 프로토콜은 크게 강의실 센서와 강사 센서 간 통신 프로토콜과 강의실 센서와 제어 서버 간 통신 프로토콜로 구분할 수 있다.

#### 3.2.1 강의실 센서와 강사 센서 간 통신프로토콜

강의실 센서와 강사 센서 간 통신 프로토콜 구현은 이웃 노드 간 통신을 필요로 하므로 Contiki에서 지원하는 통신 스택인 Rime 프로토콜의 abc(anonymous best-effort communication) API를 사용하여 구현한다. 강의실 센서와 강사 센서 간 통신을 위하여 사용된 자료 구조는 그림 3과 같다. 강의실 센서는 강사 센서의 이력을 관리하기 위하여 순차 번호(seqno)를 관리한다. 사용 가능한 메시지 형(type)은 강의실 센서 광고 메시지(RS ADV)와 강사 센서 응답 메시지(LS RESP) 두 종류가 있다. RS ADV 메시지는 현재 어떤 강의실에 있는지 정보를 알려주며,

```
typedef struct {
    uint8_t    type;
    rimeaddr_t from;
    rimeaddr_t to;
    uint16_t   seqno;
} rs_ls_message_t;
```

그림 3. 강의실 센서와 강사 센서 간 통신에 사용되는 자료 구조

LS RESP 메시지는 특정 강사의 정보를 알려준다.

강의실 센서(RS)에서 강사 센서 정보를 찾기 위해서 주기적으로 RS ADV 메시지를 이웃 노드로 방송하고, 강사 센서(LS)는 RS ADV 메시지를 수신하면 강사 정보를 알려주는 LS RESP 메시지를 해당 강의실 센서로 전송한다. 이를 의사 코드로 표현하면 그림 4와 같다.

3.2.2 강의실 센서와 제어 서버 간 통신 프로토콜

제어 서버는 인터넷에 연결되어 있으므로 센서 네트워크 내에 있는 강의실 센서에서 외부 네트워크인 인터넷에 연결될 수 있는 방법이 필요하다. Contiki에서 인터넷 프로토콜을 지원하는 uIP 통신 스택을 사용하여 본 논문에서 구현하는 강의실 제어 시스템을 사설망(private network)으로 구성하면 쉽게 인터넷에 연결 가능하다. 즉, 센서 네트워크에서 외부 네트워크로 연결하기 위하여 통신 프로토콜을 변환할 필요가 없으므로 보다 효율적인 통신이 가능하다. 강의실 센서와 제어 서버 간 통신을 위하여 사용된

```

RS_Advertise_Protocol()
seqno = 0;
set a timer to a given period  $t_{rs}$ ;
while (1) {
    wait an event;
    if (the timer is expired) {
        broadcast an RS ADV message
        with seqno;
        increment seqno;
    } else if (LS RESP message is received) {
        send an LS INFO message
        to the control server;
    }
}

LS_Response_Protocol()
while (1) {
    wait an event;
    if (RS ADV message is received) {
        get seqno from the received RS ADV
        message;
        send an LS RESP message with seqno
        to the RS which sent the RS ADV
        message;
    }
}
    
```

그림 4. 강의실 센서와 강사 센서 간 통신 프로토콜

```

typedef struct {
    rimeaddr_t rs;
    rimeaddr_t ls;
    uint16_t seqno;
    uint16_t temp;
    uint16_t humidity;
} ls_info_t;
    
```

그림 5. 강의실 센서와 제어 서버 간 통신에 사용되는 자료 구조

LS INFO 메시지의 자료 구조는 그림 5와 같다. 임의의 강의실에서 특정 강사가 강의 중임을 알려 주기 위하여 해당 강의실 센서의 주소와 강의중인 강사가 지닌 강사 센서의 주소 정보를 포함하고 있다. 그리고 강사가 진행 중인 강의의 이력을 관리하기 위하여 강사 센서에서 보내온 LS RESP 메시지의 seqno 정보를 같이 제어 서버로 송신한다. 그 외에 강의실 온도와 습도 정보도 같이 송신하여 강의실 제어에 활용할 수 있도록 한다.

강의실 센서는 그림 4에서 기술한 것처럼 이웃한 강사 센서로부터 LS RESP 메시지를 수신하자마자 이를 이용하여 LS INFO 메시지를 만들고 이를 즉시 제어 서버로 전송한다. 제어 서버(CS)는 그림 6과 같이 여러 강의실 센서와의 통신을 위하여 UDP 서버로 동작하며, 수신한 LS INFO 메시지 정보를 사용하여 특정 강사의 강의 이력 정보를 관리한다. 다음은 제어 서버의 강의 이력 관리 알고리즘에 사용되는 약어에 대한 설명이다.

- $t_{i,j}$  : i번째 요일의 j번째 타임 슬롯.  $i(1 \leq i \leq 5)$  와  $j(1 \leq j \leq 9)$ 는 각각 월~금요일과 1~9교시를 표현한다.
- $t_{cs}$  : 타이머 주기. 대학 강의의 경우 1시간 주기를 가진다.
- $S_i$  : 임시 강의 이력 집합. 타임 슬롯  $t_{i,j}$  동안 여러 강의실로부터 수신한 LS INFO 메시지로부터 (강의실, 강사 정보, 순차번호)를 저장하여 강사의 강의 이력 정보를 모은다.
- $S_{rs,ls}$  : 임시 강의 이력 집합의 부분집합. 타임 슬롯  $t_{i,j}$  동안 강의실 rs에 있는 강사 ls의 강의 이력 정보를 나타낸다.
- S : 예약된 강의 집합. 이 정보를 사용하여 추후 관리자가 강의실 제어 정보를 설정하여 강의 진행에 도움을 줄 수 있다.

```

CS_History_Management()
Lecture history set  $S_i = \emptyset$ ;
Reserved lecture history set  $S = \emptyset$ ;
Lecture room set  $R = \emptyset$ ;
Lecturer set  $L = \emptyset$ ;
set a timer to a given period  $t_{cs}$ ;
while (1) {
    wait an event;
    if (LS INFO message is arrived) {
        get (rs, ls, seqno) from the most recent LS
        INFO message;
         $S_i = S_i \cup \{(rs, ls, seqno)\}$ ;
         $R = R \cup \{rs\}$ ;
         $L = L \cup \{ls\}$ ;
    }
    if (the timer is expired) {
        for all  $rs \in R$  {
             $S_{rs,ls} = \emptyset$ ;
            for all  $ls \in L$  {
                 $S_{rs,ls} = \{(r,l,s) \in S_i | (r=rs) \wedge (l=ls)\}$ ;
                if (  $|S_{rs,ls}| > T_{cnt}$  )
                     $S = S \cup \{(rs, ls, t_{i,j})\}$ ;
            }
             $S_i = \emptyset$ ;
        }
    }
}

```

그림 6. 제어 서버의 강의 이력 관리 알고리즘

- $R$ : 강의실 집합. 제어 서버로 LS INFO 메시지를 전송하는 강의실 정보를 관리한다.
- $L$ : 강사 집합. 특정 강의실에 있는 강사 정보를 관리한다.
- $T_{cnt}$ : 강의 진행 여부를 판단하기 위한 임계값.

$t_{i,j}$ 는 현재 시각 정보를 사용하면 쉽게 계산할 수 있다. 예를 들어, 월요일 1교시, 화요일 3교시는 각각  $t_{1,1}$ ,  $t_{2,3}$ 으로 표현한다. 임의의 강의실 센서에서 전송하는 LS INFO 메시지를 제어 서버에서 수신하자마자 메시지의 정보를 사용하여 (강의실, 강사, 순차번호)를 강의 이력 집합  $S_i$ 에 추가하고, 해당 강의실과 강사를 각각 강의실 집합  $R$ 과 강사 집합  $L$ 에 추가한다. 타이머가 종료되면 매 타임 슬롯의 끝에서 특정 강의실에서 특정 강사 센서를 임계값  $T_{cnt}$ 보다 더 많이 탐지하였는지 판단해서 그 타임 슬롯에 강의가 진행되었는지 여부를 판단한다. 이때 특정 강의실에서 특정 강사가 강의를 진행한 것으로 판단되면 추후

관리자가 해당 강의실 내 각종 장비를 제어하여 강의 진행을 도울 수 있도록 할 수 있다.

## 4. 구현 및 성능 실험

### 4.1 시스템 구현

강의실 제어 시스템은 동의대학교 정보공학과 9층에 시스템을 설치하여 성능 실험을 하였다. 강의실 센서, 강사 센서, 더미 센서와 같은 센서 모듈에는 Contiki 운영체제[4]를 사용하였고 제어 서버용 프로그램은 Qt 라이브러리를 사용하여 개발하였다. 제어 서버용 프로그램은 크게 각 강의실의 상황을 볼 수 있는 모니터링 화면과 임의의 강사에 대한 강의 이력을 관리하고 강의실을 제어하는 관리 화면으로 구성된다.

그림 7의 강의실 모니터링 화면에서는 각 강의실에서 전송되는 강사 정보를 실시간으로 보여준다. 강의실에 강사가 있을 경우에는 해당 강사의 정보와 그 강의실의 온도, 습도 정보를 실시간으로 모니터링할 수 있다. 화면 하단부에서는 강의실 센서에서 전송되는 LS INFO 메시지의 정보를 텍스트 형태로 보여준다.

그림 8은 강의실 제어 시스템의 제어 서버용 프로그램 중 관리 화면을 보여준다. 특정 강의실을 선택하면 시간대 별로 어느 강사에 의하여 강의가 진행되었는지 여부를 알 수 있으며, 일정 시간 이상 강의를 진행하는 경우 해당 강사를 인식하여 강의실 내에 있는 각종 장비(PC, 빔 프로젝터, 전등 등)를 제어하여 강의가 원활하게 이루어질 수 있도록 할 수 있다.

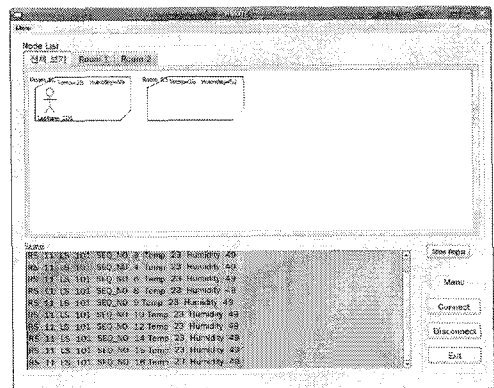


그림 7. 강의실 모니터링 화면

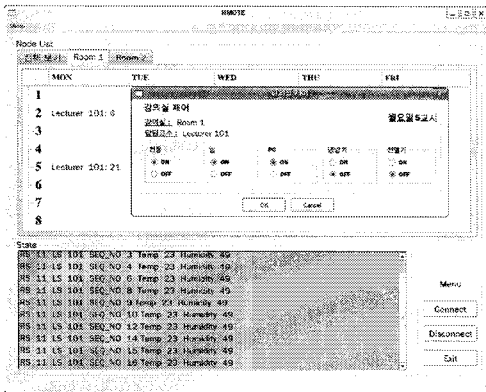


그림 8. 강의실 제어 관리 화면

### 4.2 성능 실험

성능 실험을 위한 테스트 환경은 그림 2와 같이 두 개의 강의실에 강의실 센서를 설치하고 다른 강의실에 제어 서버를 설치하여 구축한다. 본 논문에서는 그림 4의 RS ADV 메시지의 발생 주기( $t_{rs}$ )를 최적으로 설정하기 위한 실험을 수행하였으며, 두 개의 강의실에 각각 한 개의 강의실 센서와 여러 개의 터미센서를 설치하여 획득한 강사 센서 정보를 강의실 센서로부터 제어 서버로 잘 전달해 줄 수 있는 여부를 실험하였다.

먼저 강의실 센서의 적절한 RS ADV 메시지 발생 주기  $t_{rs}$ 를 설정하기 위하여 그림 9와 같이  $t_{rs}$ 를 변화시켜 LS INFO 메시지 도착률을 측정하였다. LS INFO 메시지 도착률은 강의실 센서에서 일정 시간 동안 발생한 LS INFO 메시지와 제어 서버에 도착한 LS INFO 메시지의 비율로 정의한다. 강의실 센서에서 제어 서버로 전송하는 LS INFO 메시지가 센서 네트워크 내에서 전달되어지는 도중 유실되지 않도록 강의실 내에서 직접적으로 강의실 센서와 제어 서버가 통신 가능하도록 테스트 환경을 구축하였다. 그림 10은  $t_{rs}$ 에 따른 LS INFO 메시지 평균 도착률을 보여준다. 강의실 센서 광고 메시지 발생 주기가 아주 짧은 경우(2초)를 제외하고는 90% 이상 메시지가 제어 서버에 도착함을 알 수 있다. 특히 강의실 센서 광고 메시지 발생 주기가 5초, 120초인 경우에는 97% 이상 메시지가 도착하여 좋은 성능을 보이고 있다.  $t_{rs}$ 가 길수록 센서 노드의 에너지 효율 면에서 좋은 성능을 보일 것으로 예상되나 주기가 너무 길어지면 강사의 강의 여부를 파악하기 어려워지므로 그

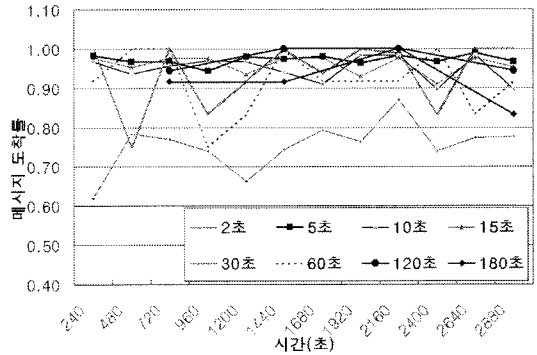


그림 9. RS ADV 메시지 발생 주기( $t_{rs}$ ) 변화에 따른 LS INFO 메시지 도착률

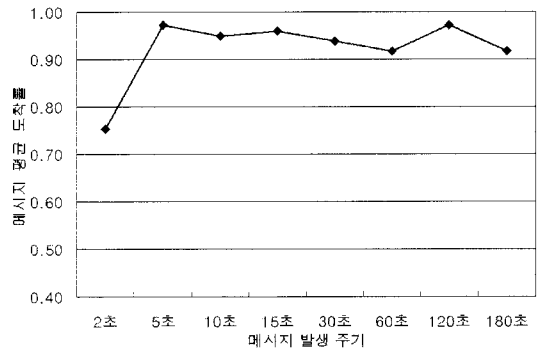


그림 10. RS ADV 메시지 발생 주기( $t_{rs}$ )에 따른 LS INFO 메시지 평균 도착률

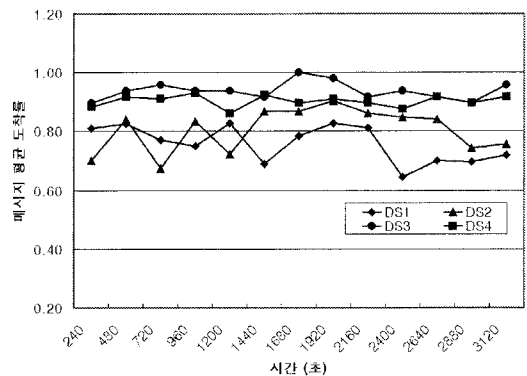


그림 11. 강의실 당 터미 센서 추가에 따른 LS INFO 메시지 평균 도착률

주기를 적절하게 설정해야 한다. 따라서 대학 강의의 경우 1시간 주기로 진행되므로  $t_{rs}$ 를 120초로 설정하면 시간당 30회 강사 센서 유무를 확인할 수 있어 통계를 구하기에 적절할 것으로 예상된다.

다음으로는 그림 2와 같이 두 개의 강의실에 강의

실 센서를 설치하고 다른 강의실에 제어 서버를 설치하여 테스트 환경을 구축하였다. 강의실 센서의 RS ADV 메시지 발생 주기는 5초로 설정하였다. 각 강의실마다 더미 센서를 추가적으로 배치하면서 강의실 센서에서 제어 서버로 전송되는 LS INFO 메시지 평균 도착률을 측정하였다.  $DS_i$  ( $i=1\sim 4$ )는 각 강의실에 배치된 더미 센서의 수를 의미한다. 각 강의실에 배치된 더미 센서가 1개일 경우 약 76% 정도의 LS INFO 메시지가 제어 서버로 전달되었으며, 강의실 별 더미 센서의 수가 1개씩 추가될 때마다 LS INFO 메시지 도착률은 증가함을 알 수 있다. 강의실 별 더미 센서의 수가 3개 이상일 경우 평균 90% 이상의 LS INFO 메시지 도착률을 보이고 있어 강사의 강의 진행 여부를 판단하는데 무리가 없을 것으로 판단된다.

## 5. 결 론

본 논문에서는 강사에 대한 강의 이력 정보를 관리하고 이 정보를 활용하여 강의실 장비를 강의 시간에 맞추어 원격으로 제어할 수 있는 센서 네트워크 기반 강의실 제어 시스템을 제안하고 구현하였다. 강의실 제어 시스템을 구현하기 위하여 센서 노드는 하이버스(주) Hmote2420을 사용하였으며, 센서 노드 운영체제는 Contiki를 사용하였다. Contiki 운영체제는 초소형 임베디드 운영체제로서 센서 네트워크 통신을 위한 Rime 프로토콜과 TCP/UDP 통신을 위한 uIP 프로토콜을 지원하며, 이를 각각 강의실 센서와 강사 센서 간 통신, 강의실 센서와 제어 서버 간 통신 구현에 사용하였다. 제어 서버용 프로그램은 Qt를 사용하여 리눅스, 윈도우즈와 같은 다른 운영체제에서도 사용할 수 있도록 개발하였다. 강의실 제어 시스템이 효과적으로 동작할 수 있도록 테스트 환경을 구축하여 실험함으로써 최적의 강의실 센서 메시지 발생 주기를 구하였으며, 또한 강의실 센서와 제어 서버 간 통신이 원활하게 이루어지도록 강의실 내 더미 센서의 적절한 수를 실험을 통하여 구하였다.

본 논문에서 제안한 강의실 제어 시스템을 활용하면 강사의 강의 이력 정보를 수집하여 강의실 내 PC, 빔 프로젝터, 전등 등의 장비를 강의에 맞게 효과적으로 운용할 수 있으며, 이는 강사가 장비의 ON, OFF에 작동에 구애 받지 않고 강의에만 전념하여 효과적인 강의가 이루어질 수 있다. 또한, 강의 후 강의 관련

장비를 OFF하지 않고 강의실을 퇴실하는 일이 종종 일어난다. 이는 강의실에 강의가 없는 상황에서도 빔 프로젝터, 전등 냉난방기가 계속 가동되고 있음을 의미하며, 전기 등 불필요한 경비지출이 많이 소요된다. 본 논문에서 구현한 강의실 제어 시스템은 장비의 자동 제어로 관리되어 이러한 문제점을 해결하여 경비절감 효과가 기대되고, 장비의 수명연장이 기대된다. 강의실 제어 시스템을 구현하기 위하여 사용된 센서 네트워크는 위치 추적과 같은 다른 응용 분야로도 연구를 확장할 수 있을 것으로 기대된다.

## 참 고 문 헌

- [1] Healthcare and Life Sciences, <http://www.ibm.com/healthcare>.
- [2] <http://robotics.eecs.berkeley.edu/~pister/29Palms0103>.
- [3] 하이버스(주), <http://www.hybus.net>.
- [4] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," In *the 29th Annual IEEE Int'l Conference on Local Computer Networks*, pp. 455-462, 2004.
- [5] Crossbow Inc., <http://www.xbow.com>.
- [6] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," In *The Fourth International Conference on Information Processing in Sensor Networks*, Apr. 2005.
- [7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System Architecture Directions for Networked Sensors," In *the Proceedings of Architectural Support for Programming Languages and Operating Systems*, Apr. 27, 2000.
- [8] FreeRTOS: <http://www.freertos.org>.
- [9] A. Dunkels, "Full TCP/IP for 8 Bit Architectures," In *Proceedings of the First International Conference on Mobile Systems, Applications and Services*, May 2003.
- [10] A. Dunkels, "Rime - A Lightweight Layered Communication Stack for Sensor Networks,"



In the 4th European Conference on Wireless Sensor Networks, Apr. 2007.

[11] A. Dunkels, "Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded System," In the 4th ACM Conference on Embedded Networked Sensor Systems, Nov. 2006.



**빈 기 철**

1990년 동아대학교 물리학과 이학사  
 2000년 부경대학교 전산정보학과 이학석사  
 2009년 동의대학교 컴퓨터응용공과 공학박사  
 1990년~1986년 (주)케이씨비 대표이사

2001년~현재 동의대학교 컴퓨터소프트웨어공학과 겸임 부교수  
 1998년~현재 (주)우인테크 전무이사  
 관심분야 : 모바일컴퓨팅, 라우팅, 센서네트워크



**이 종 민**

1992년 경북대학교 컴퓨터공학과 공학사  
 1994년 한국과학기술원 전산학과 공학석사  
 2000년 한국과학기술원 전자전산학과 공학박사  
 1997년~2002년 삼성전자 무선사업부 책임연구원

2005년 Research associate, University of California, Santa Cruz  
 2002년~현재 동의대학교 컴퓨터소프트웨어공학과 부교수  
 관심분야 : 모바일컴퓨팅, 라우팅, 센서네트워크



**권 오 준**

1986년 경북대학교 전자공학과 공학사  
 1992년 충남대학교 전산학과 이학석사  
 1998년 포항공대 전자계산학과 공학박사  
 1986년~2002년 한국전자통신연구원 선임연구원

2000년~현재 동의대학교 컴퓨터소프트웨어공학과 부교수  
 2007년~현재 한국전자통신연구원 이동통신연구본부 초빙연구원  
 관심분야 : 컴퓨터네트워크, 정보보호, 무선 인터넷, 패턴인식, 인공지능