# 기계명령어-레벨의 이산-사건 시뮬레이션을 이용한 센서 네트워크 시뮬레이터 개발

정용덕[0] 김방현 김태규 김종현
연세대학교 전산학과
sofe21@gmail.com[0], legnamai@chol.com, windcry@korea.com, jhkim@dragon.yonsei.ac.kr

# Development of Sensor Network Simulator using Machine Instruction-level Discrete-Event Simulation

Yong-Doc Jung[0] Bang-Hyun Kim Tae-Kyu Kim Jong-Hyun Kim
Dept. of Computer Science, Yonsei University

요 약

유비쿼터스 컴퓨팅의 기반 설비인 센서 네트워크는 많은 수의 센서 노드들로 구성되며, 각 센서 노드의 하드웨어는 매우 작은 규모이다. 또한 최소한의 전력 소모를 위하여 센서 노드들은 동적으로 재구성되며, 노드들 간의 통신은 무선 네트워크를 통하여 이루어진다. 센서 네트워크는 구축 목적에 따라 네트워크 토폴로지 및 라우팅 방식이 결정되어야 하고, 이와 더불어 센서 노드의 하드웨어와 소프트웨어도 필요에 따라 다양하게 변경되어야 한다. 따라서 센서 네트워크가 구현되기 전에 시스템 동작과 성능을 예측할 수 있고 소프트웨어 개발 환경도 제공해주는 시뮬레이터가 사용 가능하다면, 시스템 개발 기간을 크게 단축시킬 수 있을 것이다.

기존의 센서 네트워크 시뮬레이터들은 특별한 응용을 위한 특정 기반의 하드웨어와 운영체제에 국한되어 개발되었기 때문에 다양한 센서 네트워크 환경을 지원하기에는 한계가 있으며, 센서 네트워크 설계상의 주요 요소인 전력 소모량 분석이 포함되지 않았다. 따라서 본 연구에서는 특정한 응용이나 운영체제에 제한을 받지 않으면서 다양하게 센서 네트워크 환경을 설계 및 검증할 수 있고 전력 소모량 추정도 가능한 시뮬레이터를 개발하는 것을 목표로 하였다. 본 연구에서 개발한 시뮬레이터는 기계명령어-레벨(machine instruction-level)의 이산-사건 시뮬레이션(discrete-event simulation) 기법을 이용함으로써 실제 센서 노드의 프로그램 실행 및 관련 동작들을 세부적으로 예측하는 데 사용될 수 있도록 하였다. 시뮬레이션의 작업부하(workload)인 명령어 트레이스(instruction trace)로는 ATmega128L 마이크로컨트롤러용으로 크로스 컴파일된 인텔 헥스-레코드 형식(.hex) 또는 S-레코드 형식(.srec)의 파일을 사용한다.

## 1. 서론

미국의 'Business Week'지는 미래의 기술이라는 특집기사에서 비즈니스 관점에서 주목해야 할 네 가지 기술에 센서 네트워크(sensor network)를 포함시켰고[1], MIT의 'Technology Review'지는 세상을 바꿀 10가지 미래 기술들 중의 하나로 "Brain-Wireless Sensor Networks"를 제시하기도 하였다[2]. 이와 같이 미래의 주요 기술로 주목받고 있는 센서 네트워크는, 최근 일상생활에 산재한 사물과 물리적 대상이 점차 정보의 대상으로 확대됨에 따라 인간과 컴퓨터 및 사물들이 유기적으로 연계되어 다양하고 편리한 새로운 서비스를 제공해 주는 유비쿼터스 컴퓨팅(ubiquitous computing)의 핵심 인프라가 될 것으로 예상되고 있다.

센서 네트워크는 제한된 전력량을 가진 많은 수의 작은 센서 노드들로 구성되어 있고, 센서 노드들 간의 통신은 무선으로 이루어진다. 이 네트워크는 구축 목적에 따라 네트워크 토폴로지 및 라우팅 방식이 결정되어야 하며, 그와 더불어 센서 노드의 하드웨어와 소프트웨어도 필요에 따라 다양하게 변경되어야 한다. 따라서 센서 네트워크가 구현되기 전에 시스템 동작과 성능을 예측할 수 있고 소프트웨어 개발 환경도 제공해주는 시뮬레이터가 사용 가능하다면, 시스템 개발 기간을 크게 단축시킬 수 있을 것이다.

센서 네트워크 시뮬레이터에 관한 연구로는 Berkeley 대학 중심의 NEST(network embedded software technology) 프로젝트[3]에서 개발된 TOSSIM(TinyOS simulator), Prowler (probabilistic wireless network simulator), Siesta(simple NEST application simulator), Rmase(routing modeling application simulation environment), 그리고 Ashut(acoustic simulator for urban terrain) 등이 있으며, 이들은 NEST 프로젝트에서 개발된 TinyOS가 탑재되어 있는 하드웨어 플랫폼(mote 또는 sensor node) 기반에서 동작하는 시뮬레이터들이다. TOSSIM은 센서 노드에 탑재된 TinyOS와 응용프로그램의 동작과 상호작용을 시뮬레이션 하는 운영체제 시뮬레이터이며. Prowler는 무선 네트워크 알고리즘을 평가할 수 있는 네트워크 시뮬레이터이고, Siesta는 NEST 응용 프로그램 및 미들웨어의 기능을 검증할 수 있는 미들웨어 시뮬레이터이다. 그리고 Rmase는 네트워크 토폴로지 및 응용 시나리오를 생성하여 센서 네트워크의 라우팅 기능을 평가하는 Prowler 기반의 라우

팅 시뮬레이터이며, Ashut는 주어진 환경에서 직접 전달되는 음향과 반사 전달되는 음향의 도착 시간을 계산할 수 있는 시뮬레이터이다[4].

그러나 이 시뮬레이터들은 TinyOS 상에서만 동작하며, 전력 소모량에 대한 부분은 고려되지 않았다. 특히 TOSSIM을 구동하기 위해서는 TinyOS에서 TOSSIM용으로 컴파일된 작업부하가 필요하고, Prowler의 경우에는 MATLAB 소프트웨어가 필요하다. 또한 Siesta는 응용 소프트웨어가 시뮬레이션을 위한 특정 API 기능을 포함하지 않을 경우에는 시뮬레이션이 수행되지 않는다.

따라서 본 연구에서는 특정 운영체제에 의존하지 않으면서 전력 소모량 추정 기능도 포함하는, 기계명령어-레벨의 센서 네트워크 시뮬레이터(machine instruction-level sensor network: 이하 MISS라 함)를 구현하였다. 특히 이산-사건 시뮬레이션 기법을 이용함으로써 센서 노드 내부 모듈간 및 노드들 간의 시간 동기화가 가능하게 하였고, 기계명령어-레벨의 시뮬레이션 작업부하를 사용하여 시뮬레이션의 정밀도를 높이는 동시에 전력 소모량 추정이 가능하게 하였다.

현재 개발된 MISS가 지원하는 센서 보드는 MICAz 형식으로서, CrossBow 사의 MPR2400 보드와 옥타컴 사의 NANO-24 보드이다. 이들 센서 보드들은 Atmel 사의 ATmega128L 마이크로컨트롤러와[5] ChipCon 사의 CC2420 RF 트랜시버가[6] 장착되어있다.

## 2. 시뮬레이터의 설계 및 구현

MISS의 구조는 그림 1에서 보는 바와 같이 센서 네트워크 시뮬레이션을 수행하는 시뮬레이션 엔진(Simulation Engine)과 인터페이스 역할을 하는 GUI로 구성된다. 시뮬레이션 엔진은 ANSI C로 구현되었고, GUI는 Java로 구현되었다. 그리고 시뮬레이션 엔진에서 이산-사건 시뮬레이션을 위한 라이브러리로는 공개 소프트웨어인 smpl[7]을 사용하였다.

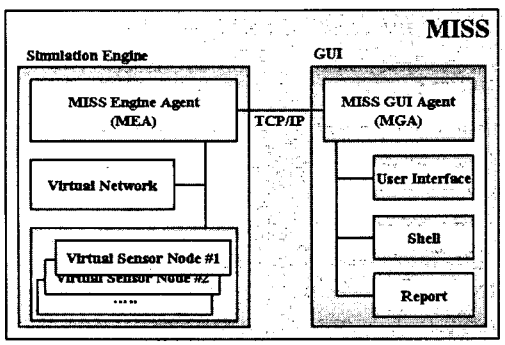


그림 1 MISS의 구조

MISS가 사용하는 시뮬레이션 작업 부하는 실제 센서 노드에 적재되는 실행 이미지(.hex 파일 혹은 .srec 파일)이다. 시뮬레이션은 이 실행 이미지를 그림 1과 같은 가상 센서 노드에 적재한 다음에, 그림 2와 같이 기계명령어 실행 동작에 따라 수행함으로써 이루어진다. 즉, 시뮬레이션은 실제 센서 노드에서와 같이 가상 플래시 메모리에서 현재 프로그램 카운터가 가리키는 위치의 기계명령어를 인출하여 실행함으로써 이루어진다.

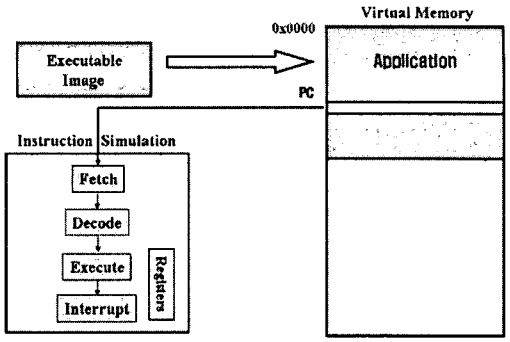


그림 2 기계어-레벨의 시뮬레이션

명령어의 인출과 실행은 이산-사건 시뮬레이션 기법을 이용하여 사건(event) 단위로 처리된다. 그림 3은 ATmega128L 마이크로컨트롤러의 CPU 코어에 의한 명령어 실행, 타이머/카운터 및 I/O 포트 동작, 그리고 인터럽트 등을 시뮬레이션 하기 위한 사건 루틴들 사이의 스케줄링 관계와 시간을 보여주고 있다. 이 사건 루틴들은 동작의 순차성에 따라 적절한 사건 시간(event time) 간격으로 관련 사건을 스케줄링 하게 된다. 스케줄링 된 사건들은 시간 스탬프(time stamp)와 함께 사건 리스트(event list)에 저장되며, 순차적으로 호출되어 수행된다.

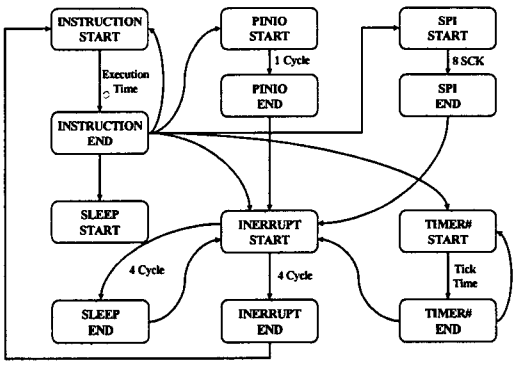


그림 3 사건(event)들 간의 스케줄링 관계와 시간

## 3. 시뮬레이터의 사용 예

MISS를 실행하는 과정은 먼저 센서 노드를 생성한 후, 응용 프로그램을 적재하게 된다. 센서 노드는 사용자가 원하는 수만큼의 센서 노드들을 동시에 여러 개 생성할 수 있으며, 나중에 필요 없는 노드는 제거 할 수도 있다. 작업부하로는 ATmega128L용으로 크로스 컴파일 된 '.hex' 형식 또는 '.srec' 형식의 실행 이미지를 노드의 플래시 메모리에 적재하며, 각 노드에는 서로 다른 실행 이미지를 적재할 수 있다. 시뮬레이션의 시간 정밀도는 ms 단위로 하였다.

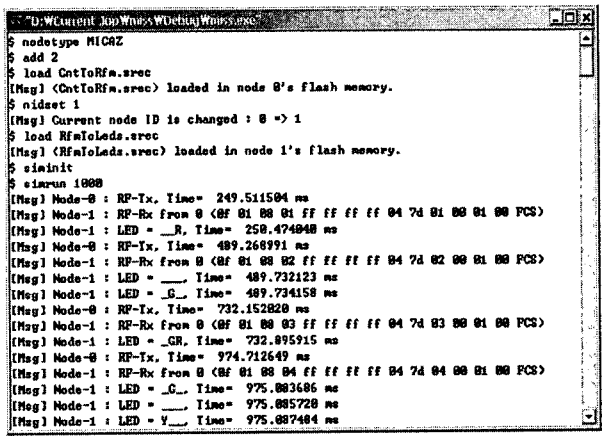


그림 4 MISS 실행 예

그림 4은 MISS의 예제 실행화면으로서, 두 개의 센서 노드들을 MICAz 타입으로 생성한 후에, 0번 노드에는 카운트 값을 RF 통신을 이용하여 주변 노드로 전송하는 프로그램인 "CntToRfm.srec"를 적재하였다. 그리고 1번 센서 노드에는 RF 통신으로 수신된 카운터 값의 하위 세 비트를 LED로 디스플레이 하는 프로그램인 "RfmToLeds.srec"를 적재한 다음에, 두 센서 노드의 동작을 1000ms 동안 시뮬레이션 하였다. 그림 4의 시뮬레이션 결과에서 보면, "Node-0 : RF-Tx, Time= 489.268991 ms"은 1번 센서 노드에서 489.268991ms때 RF 전송을 실행되었다는 것을 가리키며, "Node-1 : RF-Rx, from0 <0f 01 08 02 ff ff ff ff ~>"은 1번 노드에서 0번 노드로부터 데이터를 수신하였다는 것을 나타낸다. 수신된 데이터에서 상위 네 번째 값은 0번 노드로부터 수신된 카운터 값이다. "Node-1 : LED = _G_, Time=489.734158"은 1번 노드에서 0번 노드로부터 수신한 카운터 값의 하위 세 비트에 해당하는 값을 LED에 표시하였다는 것을 나타낸다. 이 예의 경우, 카운터 값은 02(즉, 0000 0010)이므로 green이 켜졌다는 것을 확인할 수 있다.

그림 5는 그림 4의 예제와 동일한 응용 프로그램을 실행 시킨 경우에 시뮬레이션에 걸린 시간과 전력 소모량을 예측한 결과를 보여주는 리포트 화면이다. 그림 5에서 "Simulated time=60000ms"란 센서 노드 H/W가 응용 프로그램을 60000ms 동안 실행했다는 것을 나타내고, "Simulation time = 52970.0000"이란 이 시뮬레이션을 위하여 실제 PC의 CPU가 소요한 시간이 52.97sec 였다는 것을 가리킨다. 전력소모량을 가리키는 "Power Consumption" 부분에서는 MCU와 RF 통신 모듈이 활성화(active) 상태 혹은 비활성화(idle) 상태인 경우에 각각 소모하는 전력량과 자연적으로 소실되는 전력량으로 나누어 보여주고 있다. 가상 센서 노드의 초기 전력량은 2000mA-hr로 설정하였으며, 소모 전력량은 CrossBow사에서 제공하는 전력 소모량을 이용하여 계산하였다.[8] 잔유 배터리 수명의 계산은 현재 소모된 전력량과 시간을 이용하여 몇 일 후에 전력량이 완전히 소모될지를 계산한 것이다. 이 결과는 센서 네트워크의 수명을 예측하는 데 사용될 수 있다.

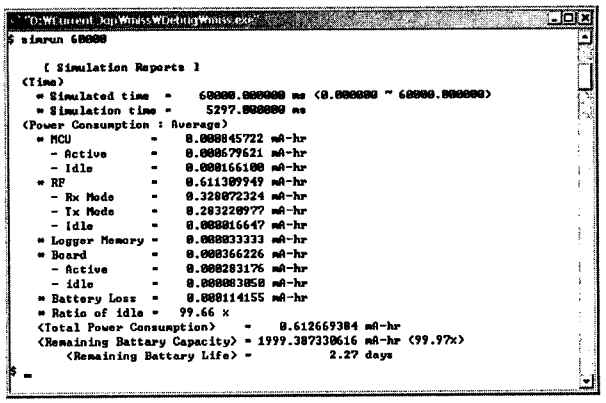


그림 5 MISS 실행 예 2

## 3. 결론 및 향후 연구 과제

본 연구에서 개발한 센서 네트워크 시뮬레이터(MISS)는 센서 노드 하드웨어나 소프트웨어 개발 이전에 센서 네트워크 환경을 설계하고 검증할 수 있는 유용한 도구이다. 특히 MISS의 구현에서는 기계명령어-레벨의 이산-사건 시뮬레이션 기법을 이용하였기 때문에, 기존의 센서 네트워크 시뮬레이터들에 비하여 더 높은 정밀도의 시뮬레이션이 가능해졌다. 즉, 프로그램 실행을 포함한 시스템 동작 시간을 실제 H/W에서의 시간에 근접하게 추정할 수 있다. 그리고 MISS는 전력 소모량 추정 기능을 제공하기 때문에 센서 네트워크의 수명도 예측할 수 있게 해준다. 또한 MISS는 실제 센서 노드에 적재되는 실행이미지를 작업부하로 사용하기 때문에, 운영체제나 라이브러리와 같은 소프트웨어에 독립적인 범용 시뮬레이터이다. 이것은 기존의 센서 네트워크 시뮬레이터들이 가지고 있는 가장 큰 문제점인 소프트웨어 의존성 문제를 해결하였다는 것을 의미한다.

현재 MISS는 MICAz 형식의 센서 보드만을 지원하며, 사용자 인터페이스는 텍스트만을 제공한다. 다양한 다른 형식의 센서 보드들을 지원할 수 있게 하고, 사용자 편의성을 위해 MISS의 사용자 인터페이스를 GUI로 구현하는 것이 앞으로 수행되어야 할 과제이다.

## 5. 참고 문헌

[1] Business week, Aug. 2003
[2] MIT Enterprise Technical Review, Feb. 2003
[3] http://webs.cs.berkeley.edu
[4] http://www.tinyos.net
[5] Atmel, ATmega128(L) Complete, May 2004
[6] Chipcon, SmartRF CC2420 Preliminary Datasheet 1.2, June 2004
[7] M. H. MacDougall, Simulating Computer Systems, MIT Press, July 1987
[8] Crossbow, "MPR/MIB Users Manua," April 2005