

# 로그 기반 무선 센서 네트워크 시뮬레이터

조민희\* · 손철수\* · 김원중\*

\*국립순천대학교

Log-based Wireless Sensor Network Simulator

Min-hee Cho\* · Cheol-su Son\* · Won-jung Kim\*

\*Sunchon National University

E-mail : kwj@sunchon.ac.kr

## 요 약

무선 센서 네트워크를 이용한 어플리케이션을 테스트할 경우 무선 센서 노드의 전원 공급, 설치 및 배포, 유지 보수 및 디버깅에 어려움이 있다. 따라서 무선 센서 네트워크 기반의 어플리케이션의 효율적인 개발과 유지 보수를 위해서는 시뮬레이터가 필수적으로 필요하다. 그러나 기존의 무선 센서 네트워크 시뮬레이터들은 무선 네트워크 구성과 유지에 관련된 MAC 주소 배분, 라우팅, 전원 관리 측면을 주요한 테스트 요소로 보았기 때문에 싱크 노드를 통하여 자료를 수집하고 처리하는 호스트 어플리케이션의 기능을 테스트하기에 부적합하다. 본 연구에서는 싱크 노드와 연결된 호스트에서 동작하는 어플리케이션을 테스트할 수 있도록 무선 센서 네트워크 상에서 전달되는 전문을 로깅하고 시뮬레이터에서 전문을 조회하여 테스트 할 수 있는 로그 기반의 무선 센서 네트워크 시뮬레이터 제안하고 설계 및 구현하였다.

## ABSTRACT

In case of application test in wireless sensor network, there are many difficulties in power supplying to wireless sensor node, installing and deploying of sensor nodes, maintaining and debugging. For efficient development and maintenance of wireless sensor network-based application, a simulator is essentially needed. However, the existing wireless sensor network simulators are focused to distribution of MAC address, routing, power management, it is not suitable to test the function of application in host which processes message through sink node. In this paper, we designed and implemented a log-based simulator for application running in host connected sink node.

## 키워드

Wireless Sensor Network, Simulator, Logging, Performance

## I. 서 론

무선 센서 네트워크는 다수의 저성능, 저비용, 저전력의 센서 노드로 구성된다. 그런데 무선 센서 네트워크를 이용한 어플리케이션을 테스트할 경우 다음과 같은 다양한 문제점이 있다.

무선 센서 네트워크의 노드들의 위치는 동적으로 배치될 수 있어야 하기 때문에 전지를 통하여 전원을 공급받도록 되어 있다. 따라서 테스트시 많은 전지의 교체 또는 충전이 필요하다. 또한 수

십 개에서 수 천 개의 무선 센서 노드를 배포하여 설치하고 다시 수거하는 데 많은 비용과 시간이 소요되는 된다. 배포된 센서 노드에 대한 성능 업그레이드를 위한 유지 보수에도 많은 노력과 시간이 필요하다. 센서 노드들은 저성능, 저비용으로 제작되어 있기 때문에 개별적으로 디버깅하는 것은 쉽지 않다. 이와 같이 개별적으로 동작하는 다수의 노드에 대한 테스트시 전력, 제어, 설치, 배포, 유지보수 및 디버깅의 문제로 시뮬레이터가 반드시 필요하다[1,2,3].

그러나 기존의 무선 센서 네트워크 시뮬레이터들은 무선 네트워크 구성과 유지에 관련된 MAC 주소 배분, 라우팅, 전원 관리 측면을 주요한 테스트 요소로 보았기 때문에 싱크 노드를 통하여 자료를 수집하고 처리하는 호스트 어플리케이션의 기능을 테스트하기에 부적합하다. 또한 간단한 테스트하기 위하여 시뮬레이터를 이해하고 학습하는 데 많은 시간이 소요되고, 테스트를 위한 시뮬레이터와의 인터페이스 및 모듈 구현이 어렵다. 무선 센서 노드들에는 다양한 어플리케이션이 배포되어 질 수 있으나 시뮬레이터에서는 단일 어플리케이션 위주로 동작하게 되어 있어 종합적인 테스트에 한계가 있다.

무선 센서 네트워크 환경에서 발생하는 전문을 이용한 호스트의 어플리케이션의 동작 성능 평가와 빠른 시뮬레이션 환경 구축 및 편리한 디버깅을 위한 새로운 개념의 무선 센서 네트워크 시뮬레이터가 필요하다. 본 연구에서는 싱크 노드와 연결된 호스트에서 동작하는 어플리케이션을 테스트할 수 있도록 무선 센서 네트워크 상에서 전달되는 전문을 로그 형태로 저장하고 저장된 로그를 분류하여 효율적으로 테스트 할 수 있도록 로그 기반의 무선 센서 네트워크 시뮬레이터 제안하고 설계 및 구현하였다.

본 논문은 무선 센서 네트워크 시뮬레이터에 관련된 연구, 새로운 시뮬레이터의 설계, 구현 및 시뮬레이션 그리고 결론으로 구성된다.

## II. 관련 연구

무선 센서 네트워크의 라우팅, 전력 소모, 노드의 분배 등을 테스트하기 위한 대표적인 시뮬레이터에는 TOSSIM, NS2, EMSTAR 등이 있다.

### 가. TOSSIM

TOSSIM[4]은 무선 센서 네트워크 운영체제인 TinyOS[7]를 위한 이산적 이벤트 시뮬레이터이다. TOSSIM과 TinyOS는 UC 버클리 대학에서 개발되었다. TOSSIM은 TinyOS 컴퍼넌트, 이벤트 기반의 실행 모듈, 라디오 모델과 ADC 모델, 하드웨어 추상화 컴퍼넌트, 통신 서비스로 구성되어 있다. 자바로 구현된 TinyViz는 TOSSIM에 접속하여 시뮬레이션을 시각화한다. NesC로 구현된 센서 모듈의 소스를 시뮬레이션 용으로 다시 컴파일하여 사용할 수 있다. 그러나 단일한 어플리케이션의 이미지만을 가질 수 있어 다양한 어플리케이션을 동시에 시뮬레이션을 하기가 힘들고, 외부에서 제어기가 용이하지 않다.

### 나. NS2

NS2[8]는 패킷 레벨의 IP 네트워크에 초점을 둔 이산적 네트워크 시뮬레이터로 UC 버클리 대학에서 개발되었다. 오픈 소스로 유무선 네트워크에서

TCP와 UDP 같은 프로토콜, RED와 ECN과 같은 라우터 큐잉 정책, 멀티 캐스팅, 웹 서버 등의 성능 평가 및 개발 등에 많이 사용되고 있다. NS2의 구조는 이벤트 스케줄러, 네트워크 컴퍼넌트, Tcl, OTcl 라이브러리, Tcl 스크립트 언어로 구성된다. OCL, TCL 와 C++로 객체를 생성하여 시뮬레이션이 가능하다. 또한 시뮬레이션의 결과를 시각화하여 볼 수 있다. 그러나 실제 센서 네트워크에서 사용하는 소스를 그대로 사용할 수 없고, 무선 센서 네트워크 관련된 프로토콜은 거의 공개가 되지 않아 다시 작성하여야 하므로 문제가 많고 또한 배우기가 쉽지 않다.

### 다. EMSTAR

EMSTAR[5]는 리눅스 기반의 무선 센서 네트워크 개발과 배포를 위한 소프트웨어 시스템이다. EMSTAR의 구조는 IPC 기반의 메시지 전송을 구현하기 위한 라이브러리, 시뮬레이션과 시스템의 시각화를 위한 도구, 네트워크킹과 센싱을 지원하기 위한 서비스로 구성된다. EMSTAR의 플랫폼인 스타게이트 마이크로서버는 400MHz의 Intel Xscale 프로세서와 리눅스 개발 환경으로 구성된다.

## III. 설계

무선 센서 네트워크의 싱크 노드에 연결된 호스트에서 동작하는 어플리케이션의 성능을 테스트하기 위한 로그 기반의 시뮬레이터를 구현하기 위하여 다음과 같이 설계를 하였다.

### 가. 설계 목표

하드웨어, 센서 네트워크의 구조, 운영체제 등에 제약을 받지 않고 또한 반복된 재현을 위하여 로그 레벨의 시뮬레이터를 구현하는 것이 설계 목표이다. 이러한 로그 레벨의 시뮬레이터는 실제 네트워크 상에서 발생한 전문을 로그 형태로 분류하여 저장하고 재전송함으로써 하드웨어와 네트워크에 독립적으로 동작하고, 또한 로깅된 전문을 복사, 수정, 삭제를 함으로써 다양한 시나리오로 시뮬레이션을 수행할 수 있다.

호스트 및 센서 네트워크의 성능 평가를 위하여 로그를 저장할 메시지 큐에는 메시지가 입력된 시간, 큐에 있는 메시지의 처리가 시작된 시간, 메시지 처리가 완료된 시간을 기록한다. 큐에 입력된 시간을 계산하여 네트워크에서 발생하는 전문의 트래픽을 분석할 수 있고, 처리 시작 시간과 완료 시간을 계산하여 메시지 별로 처리시간을 분석하여 호스트에서 전문의 처리율을 분석할 수 있다.

이와 같이 큐에 저장된 전문들의 다양한 조건에 의하여 조회할 수 있고, 필요에 따라 분류하고, 메시지를 편집하고 파싱이 용이하도록 사용자 인터페이스를 구성한다.

나. 유즈 케이스

무선 센서 네트워크의 시뮬레이터의 유즈 케이스는 그림 1과 같다. 액터는 시뮬레이션을 준비하는 사람과 실제로 시뮬레이션을 하는 사람이다. 시뮬레이션을 사전에 준비하는 사람은 시뮬레이터를 싱크 노드에 연결하고 필요한 전문을 로깅한다. 시뮬레이션을 하는 사람은 로깅된 메시지를 조회하여 분류하고 시뮬레이션 시나리오에 맞게 삽입, 수정, 삭제, 편집하고 시뮬레이션을 시작하고 종료한다.

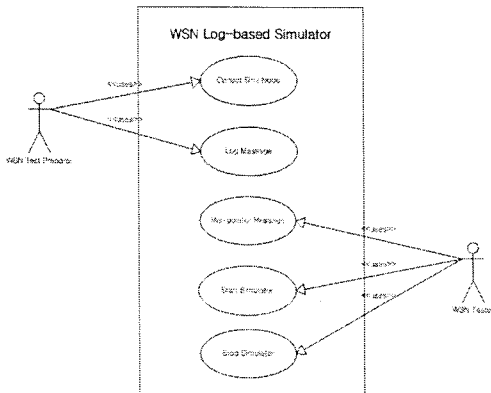


그림 1. 유즈 케이스 다이어그램

다. 시스템 구조도

무선 센서 네트워크는 원래 싱크 노드를 통하여 호스트의 어플리케이션과 연결되어야 한다. 그러나 그림 2와 같이 무선 센서 네트워크의 시뮬레이터를 위하여 가상 싱크 노드가 무선 센서 네트워크와 호스트 사이에 존재한다. 가상 싱크 노드의 메시지 로거는 싱크 노드에 송수신하는 전문을 로깅하고 메시지 플레이어는 응용프로그램에 메시지를 송수신한다.

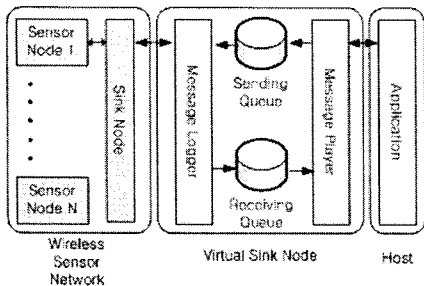


그림 2. 시스템 구조도

라. 시퀀스 다이어그램

메시지 로거 개체의 중요한 처리 흐름인 메시지를 로깅하는 처리 순서는 그림 3의 시퀀스 다이어그램으로 기술된다. 메시지 로거는 싱크 노드

와 통신하기 위하여 통신 포트를 오픈하고 싱크 노드로부터 메시지를 수신하기 위하여 기다린다. 싱크 노드로부터 메시지를 수신하면 큐에 메시지를 저장하도록 호출한다. 메시지 플레이어는 주기적 또는 이벤트에 의하여 큐에게 새로 처리할 메시지를 요청한다. 만약 메시지가 있으면 어플리케이션에 전달하여 처리토록하고 그 결과를 큐에 전달하여 처리가 완료되었음을 기록하도록 한다.

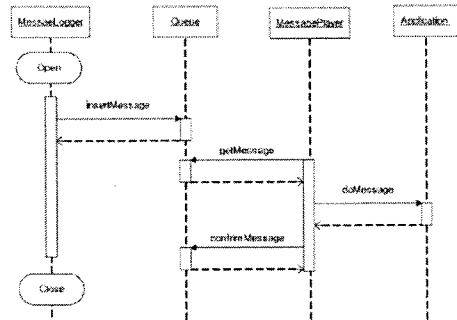


그림 3. 시퀀스 다이어그램

마. 큐 파일 구조

무선 센서 네트워크와 어플리케이션에서 수신한 전문은 시뮬레이터에서 재생하여야 하기 때문에 큐 파일에 로깅하여야 한다. 큐 파일은 전체 구조는 그림 4와 같이 큐 파일의 정보가 저장되는 헤더부와 각각의 메시지가 로깅되는 메시지부로 구성된다.

Header	Queue Information
Message	Message 1
	Message 2
	Message 3
	Message 4
	.....
	Message Max

그림 4. 큐 파일 구조

큐 파일의 헤더 구조는 그림 5와 같이 각각의 메시지가 차지할 수 있는 최대 길이, 큐 파일에 저장될 수 있는 최대 메시지 갯수, 처리된 메시지 포인터, 새로 저장될 메시지 포인터로 구성된다.

Message Max Size	Message Max Count	Read Pointer	Write Pointer
Integer	Integer	Integer	Integer

그림 5. 큐 파일 헤더 구조

큐 파일에 저장되는 레코드는 그림 6과 같이 큐가 메시지를 저장한 시각, 어플리케이션이 메시지를 읽어간 시각, 처리 완료한 시각, 처리 결과, 메시지로 구성된다.

Field	Data Type
Input	Datetime
Start Time	Datetime
End Time	Datetime
Result	Integer
Length	Integer
Frame	Sting(MessageMaxSize)

그림 6. 큐 파일 레코드 구조

#### IV. 구현 및 시뮬레이션

TinyOS 기반의 모드로 구성된 무선 센서 네트워크를 대상으로 시뮬레이터를 구현한다. 시뮬레이터의 개발 및 운영 환경은 표 1과 같다. 기술되어 있는 하드웨어의 사양은 권장 사양이 아니고 운영될 수 있는 최소 사양이다.

<표 1 개발 및 운영 환경>

하드웨어	CPU : Intel Pentium 4 1GHz RAM : 500 MB HDD : 20GB PORT: USB to Serial
소프트웨어	O S: Windows XP TOOL: Visual Studio 2008 C# .NET Framework 3.5

위에서 기술한 하드웨어 최소 사양에서 시뮬레이터는 최대 256KB의 전문을 처리할 수 있고, 최대 50개의 무선 센서 노드에 대한 전문을 분류하여 개별적으로 어플리케이션에 동시에 전달할 수 있다. 또한 로깅된 데이터를 복사, 수정, 삭제에 의한 편집으로 다양한 시나리오를 만들어 시뮬레이션을 수행할 수 있다. 그림 7은 시뮬레이션이 수행되고 있는 메인 화면으로 센서로부터 수신한 전문을 무선 센서 노드 별로 분류하고 항목별로 파싱하여 그 값을 실시간으로 표시하여 모니터링 하고 있다.

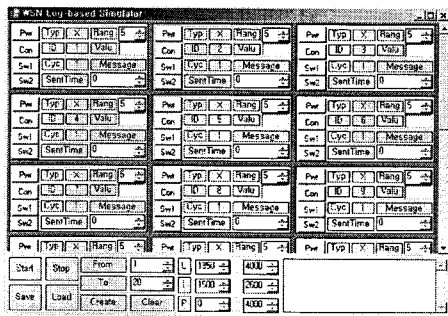


그림 7. 시뮬레이터 메인 화면

#### V. 결론

본 연구에서는 싱크 노드와 연결된 호스트에서 동작하는 어플리케이션을 테스트할 수 있도록 무선 센서 네트워크 상에서 전달되는 전문을 로그 형태로 저장하고 저장된 로그를 분류하여 효율적으로 테스트 할 수 있도록 로그 기반의 무선 센서 네트워크 시뮬레이터 제안하고 설계 및 구현 하였다.

로그 기반의 시뮬레이터는 로깅된 전문을 재생하는 기법을 사용하여 호스트 어플리케이션의 성능 평가하고 테스트 하는 데는 유리하지만, 무선 센서 노드와의 상호 작용이 부족하기 때문에 무선 센서 노드를 시뮬레이터에 컴퍼넌트 형태로 포함하는 연구가 지속적으로 필요하다.

#### 참고문헌

- [1] E. Egea-Lopez et al., "Simulation scalability issues in wireless sensor networks," Communications Magazine, IEEE, vol. 44, no. 7, pp. 64-73, Jul. 2006.
- [2] A. Martinez-Sala et al., "An Accurate Radio Channel Model for Wireless Sensor Networks Simulation," Journal of Communications and Networks, vol. 7, no. 4, pp. 401-407, Dec. 2005.
- [3] Wen, Y., Wolski, R., and Moore, G., "DiSenS: Scalable Distributed Sensor Network Simulation," Proceedings of ACM Principles and Practices of Parallel Programming (PPoPP) 2007, pp. 24-34, Mar. 2007.
- [4] Philip Levis, Nelson Lee, Matt Welsh, and David Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," SenSys'03, pp.126-137, Nov. 2003.
- [5] L. Girod, et al., "EmStar: a Software Environment for Developing and Deploying Wireless Sensor Networks," in the Proceedings of USENIX General Track, 2004.
- [6] <http://www.cs.berkeley.edu/~pal/research/tosssim.html>.
- [7] <http://www.tinyos.net/>
- [8] <http://www.isi.edu/nsnam/ns/>