

대규모 무선센서네트워크 개발·구축을 위한 센위버 플랫폼 기술 개요

김구조* · 박수영** · 박세현*** · 김희철****

1. 서 론

현대를 이른바 유비쿼터스 시대로 칭하고 있다. 이와 더불어 기존과는 다른 삶의 경험과 패러다임 또한 변하고 있다. 이처럼 유비쿼터스는 이제 전 세계적으로 최대 화두로 다뤄지고 있으며 유비쿼터스의 실현으로 실세계의 각종 사물들과 물리적 환경 전반 즉, 물리공간에 걸쳐 컴퓨터들이 편재되어 네트워크와 통합되는 정보통신 환경의 구축이 이루어지고 있다.

일반적으로, 무선 센서네트워크를 개발, 구축하는 작업은 센서네트워크 고유의 특성에 기인한 저전력 무선 네트워크 개발 측면뿐만 아니라 임베디드 시스템을 개발하는 성격을 갖는다[1].

센서네트워크의 구축을 위해서는 먼저 필요한 센서노드의 개발과 센서 애플리케이션의 개발이 필수적으로 요구된다. 나아가 구축된 센서네트워크 관리를 위한 네트워크 관리자 시스템도 요구된다.

다. 하지만, 앞에서 살펴본 바와 같이, 센서노드 상에 구동하는 센서 OS와 애플리케이션은 일반 프로그램이라기보다는 임베디드 S/W라는 점과 무선 센서네트워크의 고유 특징들로 인해 그 개발이 매우 어렵다. 특히, 신뢰성을 갖는 대규모 센서네트워크 환경에서는 이러한 작업은 더욱 어렵다. 이에 따라, 시간과 인력 측면에서 많은 자원투입이 요구된다. 따라서 대규모 센서네트워크를 개발함에 있어 센서네트워크 플랫폼이 필요한 이유를 II절에서 소개하고, III절은 국내 최초로 개발된 대표적인 센서네트워크 S/W 플랫폼인 센위버 플랫폼에 대해서 소개한다. IV절은 기존 개별 노드 단위로 이루어지던 성능 테스트를 대규모 센서네트워크 환경에서 일괄적으로 제어 및 수행할 수 있는 센위버 벤치마킹 시스템에 대해서 소개하며 마지막 결론으로 끝을 맺는다.

2. 센서네트워크 S/W 플랫폼

현재, 무선 센서네트워크(WSN, Wireless Sensor Network)는 환경 감시, 생태 조사, 교통 정보, 농업 생산, 건축물 관리, 생산물 유통 등 다양한 애플리케이션 분야에서 실용화 단계에 접어들고 있으며, 네트워크 규모도 점차 증가하는 추세에 있다.

무선 센서네트워크는 다수의 센서노드로 구성되며, 각 센서노드는 임베디드 S/W가 탑재되어

* 교신저자(Corresponding Author): 김희철, 주소: 경북 경산시 진량읍 내리리 15, 대구대학교 정보통신공학부, 전화: 053)850-4733, FAX: 053)850-4733, E-mail: hckim@daegu.ac.kr

* 대구대학교 유비쿼터스신기술연구센터 (E-mail: gjkim@utrc.re.kr)

** 대구대학교 유비쿼터스신기술연구센터 (E-mail: sypark@utrc.re.kr)

*** 대구대학교 정보통신공학부 (E-mail: sehyun@daegu.ac.kr)

**** 대구대학교 정보통신공학부

구동되는 임베디드 시스템이다. 통신 측면에서는 MAC(Media Access Control) 프로토콜과 데이터 전달 및 경로 설정을 담당하는 라우팅(Routing) 알고리즘을 포함하는 네트워크 프로토콜 스택이 탑재된 통신노드로서의 역할을 수행한다. 따라서 대규모 센서네트워크를 구축한다는 것은 저전력 임베디드 시스템으로서 저전력 무선통신 노드로서의 센서노드 기능을 수행하는 것이다. 이와 같은 대규모 센서네트워크를 개발 및 구축하기 위해서는 센서네트워크 플랫폼이 반드시 지원되어야 한다. 본 절에서는 센서네트워크의 필요성과 요구사항에 대해서 알아본다.

2.1 센서네트워크 플랫폼의 필요성

센서네트워크의 구축을 위해서는 먼저 해당 응용에 적합한 센서들을 지닌 센서노드 H/W의 개발과 해당 센서 값을 읽어 처리하는 센서 애플리케이션의 개발이 필요하다. 나아가 구축된 센서네트워크 관리를 위한 네트워크 관리자 시스템도 요구된다. 하지만, 센서노드 상에 구동하는 센서 OS와 애플리케이션은 일반 프로그램이라기보다는 임베디드 S/W라는 점과 무선 센서네트워크의 고유 특징들로 인해 그 개발이 매우 어렵다. 특히, 신뢰성을 갖는 대규모 센서네트워크 환경에서는 이러한 작업은 더욱 어렵다. 이에 따라, 시간과 인력 측면에서 많은 자원투입이 요구된다. 따라서 대규모 센서네트워크를 개발함에 있어 센서네트워크 플랫폼이 필요한 이유는 다음과 같다.

(1) 기존 자체 개발 방식의 한계

활용할 수 있는 가용 플랫폼의 부재로 인하여, 많은 경우 이러한 작업을 자체(In-house) 개발을 하고 있다. 소규모 센서네트워크 환경에서는 이러한 것이 가능하나 대규모 환경에서는 위에서 설명

한 임베디드 S/W 성격의 시스템 S/W가 필요하며, 센서네트워크의 제약사항 등으로 인하여 자체 개발 방식은 한계에 직면하게 된다. 따라서 기존 자체개발의 한계성에서 벗어나기 위해서 센서네트워크 개발 플랫폼이 필요하다.

(2) 기술적 측면에서의 필요성

효과적인 센서 애플리케이션의 개발, 센서네트워크의 구축 및 유지보수를 위해서는, 센서 애플리케이션 S/W 개발 단계부터 실행코드 생성 및 이미지 적재단계, 네트워크 관리단계에 이르기까지 전 주기를 지원하는 센서네트워크 개발 플랫폼이 필요하다. 『센서네트워크 S/W 플랫폼』은 센서네트워크 구축에 공통적으로 사용할 수 있는 ① 센서 OS(OS 커널, 디바이스 드라이버, 무선 센서네트워크 스택), ② 개발도구(네트워크 관리자, 검증 및 테스트 도구 등)로 구성되며, 센서네트워크의 개발 및 구축에 업계에서 공통으로 사용할 수 있는 “센서 OS 및 개발도구”가 필요하다.

(3) 산업적 측면에서의 필요성

최근 다양한 사양의 응용분야에 센서네트워크가 구축되고 있으며 이들은 기능적으로 많은 공통 기능을 가지고 있으나 가용한 표준 플랫폼 부재로 인해 각 업체마다 독자적으로 제품개발에 필요한 모든 기술부문을 확보, 적용해야 하므로 개발기간 및 비용이 증가한다. 이는 결국 국가적으로는 중복투자 및 산업 생태계 진흥 미비 등으로 인해 산업 경쟁력 약화로 이어진다. 『센서네트워크 S/W 플랫폼』을 활용할 경우 IT융·복합 산업 육성의 핵심영역인 USN 산업이 USN H/W 산업, USN 솔루션 산업, USN 서비스 산업으로 이어지는 밸류체인 상의 전주기적 산업 생태계의 조기 구축으로 이어질 수 있다.

2.2 센서네트워크 플랫폼의 요구사항

본 소절에서는 센서네트워크 개발에 활용되고 있는 기존 자체 개발의 문제와 제한사항을 설명하고 이를 해결하기 위한 유일한 방안인 센서네트워크 플랫폼의 요구사항을 간략하게 소개한다.

현재 전 세계적으로도 플랫폼 네트워크 산업이 진행 중이나 시장 규모 증가율이 낮고 응용/실증 경험 및 노하우의 부재로 활발하게 진행되지 못하고 있다.

국내 『센서네트워크 S/W 플랫폼』 현황은 환경, 건설, 국방 등 다양한 분야에서 USN 시범사업을 수행하였으나 많은 경우 1년 미만의 단기간에 걸친 소규모 센서네트워크의 적용 가능성(Feasibility)을 제시하는 수준으로 개발되었으므로 센서네트워크 플랫폼에 대한 개념을 갖기는 어렵다.

대부분의 시범사업이 10개 이하의 노드로 구성된 소형, 단순 센서네트워크 구축으로 매우 초보적인 원격 무선 모니터링의 개념을 보여주는 수준이며, 국외의 관련 프로젝트들은 대부분 단계적 실증을 통해 실용화 수준의 성과를 도출할 수 있도록 3-7년의 중장기, 수백억 규모의 중대형 형태로 진행되고 있는 실정이다.

특히, 센서 및 H/W 기술, 센서 네트워킹, 확장형 S/W 플랫폼, 응용레벨 서비스기술 부문에 표준 플랫폼 기반이 아닌 SI 성격으로 수행하여 각 밸류체인별 전문화가 이루어지지 않고 있다. 이에 따라, 국내 일반 산업과는 달리 USN 산업은 밸류체인이 형성되지 않은 상태이며, 영세한 업체들이 H/W, S/W, 서비스 전체를 개발하고 있으며, 시장의 확대도 진행되지 않아 관련업체들의 전문성이 결핍되어 있는 상황이다.

센서네트워크 플랫폼이 신뢰성을 갖는 대규모 센서네트워크의 구축에 활용되려면 다음과 같은 요구사항을 만족시켜야 한다[2].

(1) 높은 프로그래밍 용이성(High Programmability)

센서노드 상에 구동되는 센서 애플리케이션 개발자가 쉽고 빠르게 애플리케이션을 개발할 수 있도록 친숙한 프로그램 언어와 개발환경을 제공할 수 있어야 한다. 대규모 분산 및 협업 네트워크 기반의 센서 애플리케이션으로 발전함에 따라 센서 애플리케이션 로직의 복잡도가 증가하고 있어 멀티프로그래밍과 같은 병행제어 수행능력 제공이 필수적임

(2) 높은 이식성(High Retargetability)

센서네트워크 응용분야가 점차 확대됨에 따라 향후 기업체에서 응용분야별 요구사항에 적합한 무수히 많은 종류의 센서노드를 출시할 것으로 예상되고 있어 센서노드 H/W의 이질성의 급격한 증가가 예견됨에 따라 센서 OS의 이식성과 이식 도구 제공이 필수적으로 요구된다. 동일한 센서노드 H/W 사양을 갖는다고 할지라도 센서노드의 기능, 역할, S/W 구성 등이 상이한 S/W 이질성 또한 급격히 증가할 것으로 예상되고 있어, 센서노드 H/W와 S/W 명세 기반의 S/W 커스터마징 도구 제공이 필수적이다.

(3) 높은 재구성 가능성(High Reconfigurability)

센서노드의 H/W와 S/W 명세 변경에 따라 센서노드의 기능, 역할, S/W를 정적 및 동적으로 재구성할 수 있는 능력을 제공해야 한다. 특히, 이미 구축된 센서네트워크의 유지보수를 위해 설치된 센서노드의 S/W를 무선통신을 통해 에너지 효율적으로 교체 및 수정할 수 있어야 한다.

(4) 높은 확장성(High Extensibility)

기 개발된 센서노드 S/W 뿐만 아니라 센서노드에 이미 탑재된 센서노드 S/W라 할지라도 지속적인 성능 개선이 이루어 질수 있는 환경을 제

공해야하며, 각 기업체에서 개발한 S/W 모듈들을 서로 공유할 수 있도록 S/W 모듈 개발자, 수요자 간의 공동의 혁신을 통한 플랫폼 네트워크 구축이 필수적이다.

3. 센위버 플랫폼 구성 요소

센위버 플랫폼은 궁극적으로 대규모 센서네트워크를 신속하게 개발, 구축하고 이를 효과적으로 유지보수, 관리할 수 있도록 하는데 목적을 갖는다. 이를 위해서 센위버는 다양한 구성요소들을 포함하고 있는 통합 플랫폼으로서 대규모 센서네트워크 구축에 필요한 제반 특징을 제공하며, 센서노드 S/W 아키텍처, 센서 OS, 각종 개발도구 등의 제반 구성요소를 구비하고 있다.

센위버는 센서네트워크 개발·구축·관리를 위한 통합 플랫폼으로서, 앞의 정의에서 볼 수 있듯이, 크게 4 부문, 즉 ① 센서노드 상에서 구동하는 OS 커널, 디바이스 드라이버 계층, 무선 센서네트워크 스택 등으로 구성되는 센서 OS, ② 센서 애플리케이션 프로그램을 PC에서 구동시켜

오류를 수정하고 동작의 정확성을 검증할 수 있는 PC 에뮬레이터, ③ 센서 애플리케이션 개발자를 위한 통합개발환경(IDE), ④ 센서네트워크를 효과적으로 유지·관리하는 네트워크 관리자로 구성된다(그림 1). 다음 소절부터는 각 부문의 내용을 설명한다.

3.1 센위버 센서 OS

센위버 센서 OS는 센서노드 상에서 노드 애플리케이션을 효율적으로 구동시키는데 필요한 요소들을 갖추고 있다. 이 구성요소들은 일반적인 OS의 요소뿐만 아니라 센서네트워크의 고유 성격을 만족시키기 위한 요소들을 포함한다.

(1) 센서 OS 커널

우선순위 기반 선점형 스케줄링, 다중 스레드, C 기반 API, 동적 S/W 갱신, 멀티프로그래밍을 지원하는 재구성 가능한 고성능의 센서 OS이다. 센위버 센서 OS는 다양한 센서 디바이스 드라이버, MAC 스택, 라우팅 스택 등을 제공하며, 독립된 S/W 모듈(부품) 형태로 제공된다. 기본적으로

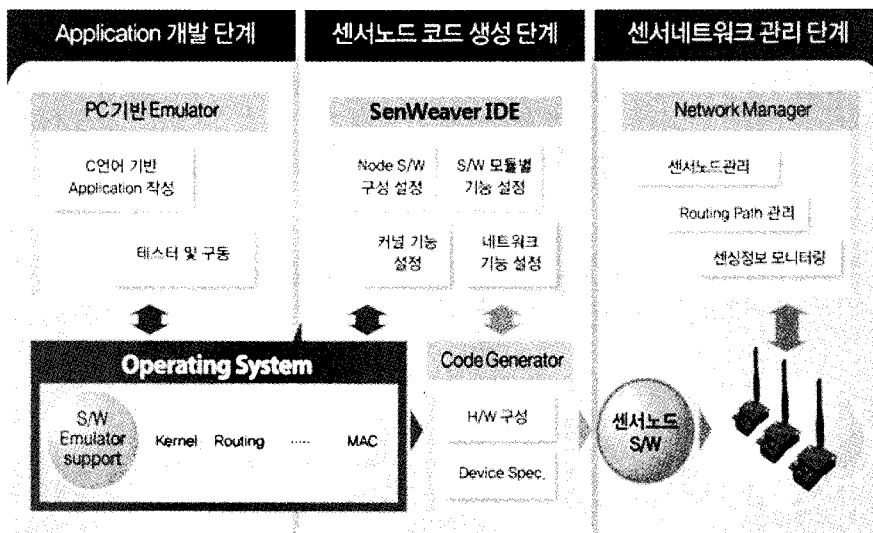


그림 1. 센위버 플랫폼 구성 요소

타 스레드 모델과 같이 우선순위 기반 선점형 스케줄링, 다중 스레드, C 기반 API 등 스레드 모델의 장점을 고루 갖추고 있다.

(2) 디바이스 드라이버

센위버 센서 OS는 다양한 센서를 지원할 수 있는 최적의 추상화된 디바이스 드라이버 모델을 제공하고 있다. 이 모델은 디바이스 드라이버에 대한 가용성 및 상용성 확보를 위해서 고도의 추상화 기술을 통해 H/W 의존성을 줄여, 재사용성, 이식성, 호환성을 높일 수 있도록 한다. 한편 디바이스 드라이버 개발자 입장에서 보면, 이 디바이스 드라이버 모델은 센서 디바이스 드라이버 개발자가 쉽고 빠르게 디바이스 드라이버를 개발할 수 있도록 하며, 센위버 IDE를 이용하여 개발된 디바이스 드라이버를 OS에 탑재할 수 있도록 하였다.

(3) 무선 네트워크 스택

센위버 플랫폼에서는 다양한 국제표준 및 사실 표준의 링크와 네트워크 프로토콜을 선택하여 사용할 수 있는 통합 무선 센서네트워크 아키텍처를 갖는다. 센위버 플랫폼에서 모듈러 S/W 아키텍처를 도입한 이유는 무선 센서네트워크에서 사용할 MAC 및 라우팅 프로토콜들이 응용요구별로 알맞은 센서네트워크 스택을 적용할 수 있도록 지원하기 위해서이다.

(4) 노드 미들웨어

센서 OS 커널과 센서노드 애플리케이션(이하 센서 애플리케이션) 사이에 위치하는 센서 서비스 기능으로 위치인식, 시간 동기화, 센서 데이터 가공 등과 같은 센서네트워크에 특화된 서비스를 제공할 수 있어야 한다.

3.2 에뮬레이터

무선 센서 네트워크 구축에는 상당히 많은 비용과 개발 기간이 소요되기 때문에 센서 네트워크의 노드 애플리케이션 개발 혹은 모듈 테스트 단계에서 실제 센서 네트워크를 구성하여 테스트하기는 어렵다. 따라서 센서 네트워크용 에뮬레이터를 활용하여 소스코드 디버깅과 노드 및 네트워크 시뮬레이션을 수행하는 것이 바람직하다[3].

개발된 센위버 PC 에뮬레이터는 센서 애플리케이션 개발의 생산성 향상을 위해 센서 애플리케이션 코드를 센서노드에 적재하지 않고도 사전에 그 동작 및 신뢰성을 검증할 수 있는 개발도구이다. 센서 OS 플랫폼 환경 기반의 센서 애플리케이션의 시뮬레이션을 위한 에뮬레이터 S/W로서 PC 환경에서 센서 애플리케이션의 프로그램 편집, 컴파일/링킹, 디버깅이 가능하다. 또한, 가상 센서네트워크를 구성하여 노드 및 네트워크 에뮬레이터를 통해 네트워크 토폴로지 및 로그 정보까지 모니터링 하는 통합 시뮬레이션 환경을 제공한다.

에뮬레이터에서는 센서 애플리케이션 프로그램을 PC에서 구동시켜 오류를 수정하고 동작의 정확성을 검증할 수 있다. 또한 OS 동작환경, 노드 구동상태 관리, 애플리케이션 동작검증 및 리포팅, 네트워크 구성관리 등 네트워크 환경 및 통계정보를 확인할 수 있다.

센위버 PC 에뮬레이터는 노드 에뮬레이터와 네트워크 에뮬레이터로 구성되며(그림 2), 각 기능에 대한 소개는 아래에서 설명한다.

(1) 노드 에뮬레이터

노드 에뮬레이터는 기존 노드기반의 개발 환경에서 PC기반 개발 환경으로 변환하여 PC 환경에서 직접 개발하고 디버깅하는 센서 애플리케이션 개발 툴이다(그림 3). 이 에뮬레이터는 오픈소스

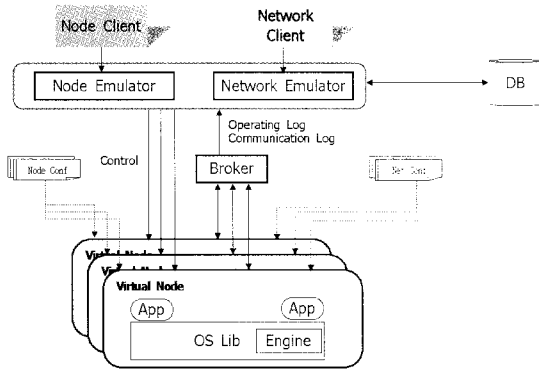


그림 2. 센서 PC 에뮬레이터 구성도

기반의 이클립스와 C언어 개발 환경을 위한 CDT 그리고 오픈소스 기반의 MinGW 컴파일러를 이용하여 개발되었다. 노드에 대한 에뮬레이션 기능을 수행하기 위하여 노드 환경 구성, 노드 구동 모니터링, 노드 구동 결과 및 분석에 대한 리포팅 기능을 제공한다.

노드 에뮬레이터는 센서 OS 계층 중 H/W 의존적인 코드를 PC 환경으로 교체하고 관련 디바이스 드라이버의 가상화를 통하여 구현할 수 있다. 그리고 무선 센서네트워크의 무선 통신 관련 부분은 소켓기반으로 가상 통신이 가능하도록 개발되었다.

노드 에뮬레이터는 임베디드 기반의 센서 OS

의 스케줄링과 태스크/스레드 매니저를 Windows 기반에서 동일하게 동작시키기 위하여 Win32 스레드와 타이머 이벤트 API를 사용하며, PC의 CPU 점유율을 낮추고 Windows 스레드의 문맥 교환(Context switching) 동작의 안정성을 보장하여 임베디드 기반 태스크/스레드는 PC 환경에서 독립적으로 동작하는 것이 가능하다. 그리고 표준 입출력 스트림(Standard I/O Stream)을 통한 시스템 동작 로그를 출력시켜 사용자는 OS 내부 동작을 모니터링 할 수 있다. 뿐만 아니라 이클립스의 디버깅 기능을 통하여 직접 변수의 값 및 함수 호출정보 등을 확인하는 것도 가능하다.

노드 에뮬레이터에서 생성된 실행파일은 하나의 가상 노드로 네트워크 에뮬레이터에서 에뮬레이션을 수행하기 위한 입력 데이터로 활용 된다.

(2) 네트워크 에뮬레이터

네트워크 에뮬레이터는 네트워크 에뮬레이션 기능을 수행하기 위하여 가상 네트워크 환경 구성, 네트워크 구동 모니터링, 네트워크 에뮬레이션 결과 및 분석에 대한 리포팅 기능을 제공한다.

노드 에뮬레이터에서 생성된 실행 이미지는 네트워크 에뮬레이터에서 개별 노드로 작동하게 된

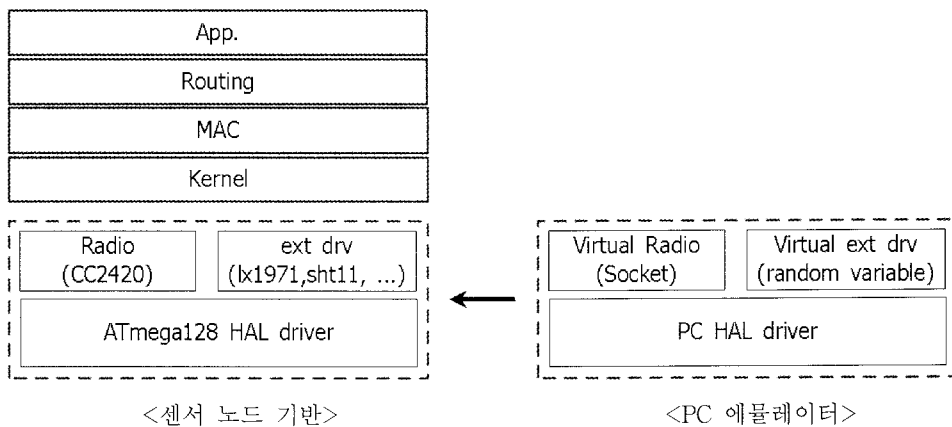


그림 3 노드 에뮬레이터.

다. 개별 노드는 윈도우 소켓을 이용하여 서로 통신을 하게 되고, 이때의 통신 상태 및 네트워크 상태가 모니터링 된다. 네트워크 에뮬레이터는 이런 일련의 작업들을 제어, 관리하고 사용자들에게 네트워크 상태를 쉽게 확인 할 수 있게 도와준다.

사용자들이 확인할 수 있는 네트워크 상태는 노드 간 통신 프로토콜과 토폴로지 그리고 네트워크 트래픽 등이 있으며 실사용 환경에 적용하기 전 네트워크의 문제점 등을 사전에 확인하고 수정이 가능하다.

노드 에뮬레이터에서 노드 간 가상 통신은 윈도우의 소켓이 사용되었다. 그리고 이들 통신을 중개하고 네트워크 에뮬레이터에게 필요한 정보를 제공하기위하여 네트워크 브로커(Network broker)라는 중개자가 존재 하게 된다(그림 4).

3.3 통합개발도구(IDE)

센위버 IDE(Integrated Development Environment)는 최대한 사용자 편의성을 높일 수 있도록 여러 가지 개발도구들이 상호 유기적으로 하나의 개발환경에 통합한 통합개발환경이다.

(1) 센위버 IDE의 기능

센위버 IDE는 S/W 컴포넌트 조합 기능을 위한

모듈 조합(Module composition) 기능, S/W 커스터마이징(Customization) 기능, S/W 코드 자동생성(Automatic code generation) 기능, S/W 코드 퓨징(Fusing) 기능을 제공하는 개발도구들이 유기적으로 구성되어 있다. 이와 같은 4가지 핵심기능을 가지는 통합개발환경을 제공함으로써 개발자는 윈스톱으로 센서 애플리케이션을 개발 할 수 있다.

① 모듈 조합

센위버는 일종의 “S/W 컴포넌트”인 “S/W 모듈”을 기반으로 하는 컴포넌트 S/W 아키텍처를 갖는다. 따라서 센서네트워크 개발자는 지정된 센서노드에 대하여 가용한 모듈들을 조합하여 필요한 노드 S/W를 만들 수 있다. 센위버 IDE는 플랫폼에서 제공하는 모든 모듈들과 애플리케이션 S/W들 중에서 센서노드에 탑재할 모듈들을 선택한 후 필요한 모듈들만 센서노드에 탑재시킬 수 있다.

② S/W 커스터마이징

센위버 IDE는 구축할 센서네트워크 응용 분야의 요구사항과 센서네트워크를 구성하는 센서노드 H/W 자원의 크기에 최적화된 코드를 생성하기 위해서 커스터마이징 기능을 지원한다.

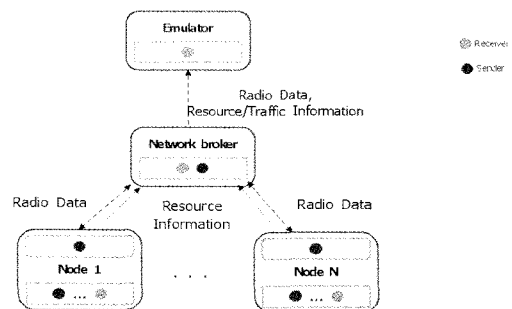
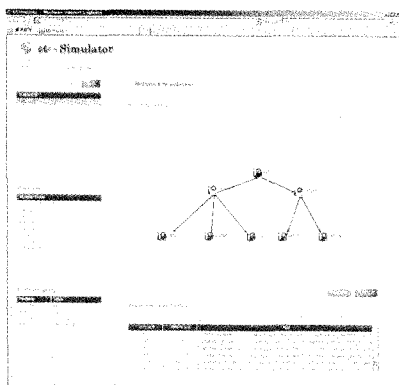


그림 4. 네트워크 에뮬레이터

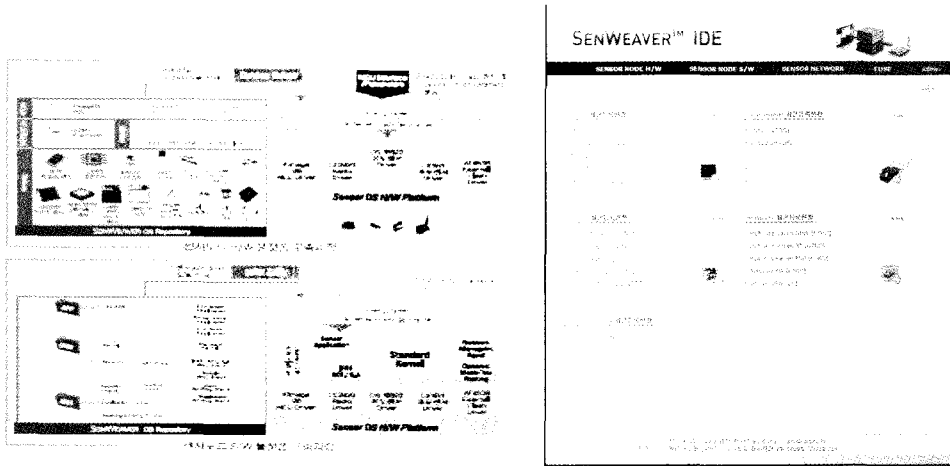


그림 5. 센위버 IDE

③ 코드 자동생성

센위버 IDE는 센서 애플리케이션 프로그램의 작성 후에 센서네트워크를 구성하는 센서노드에 대하여 해당 노드의 H/W 사양에 맞는 실행코드 이미지를 자동으로 생성시켜준다. 이는 시스템 구축자가 디바이스 드라이버나 시스템 S/W의 소스를 변경할 필요가 없이 노드 애플리케이션 프로그램에만 전념할 수 있도록 한다.

④ 퓨징

센위버 IDE는 생성된 실행코드를 해당 센서노드에 다운로드 하는 퓨징 기능을 제공한다.

3.4 네트워크 관리자

센위버 네트워크 관리자(Network Manager)는 센서노드 배치 후 물리적 센서네트워크 공간을 그래픽 기반의 정보로 제공하고 센서네트워크를 효과적으로 유지 및 보수하는 관리도구로서 센서노드 관리 및 제어, 네트워크 모니터링, 데이터 수집 등의 실시간 처리기능을 제공한다.

(1) 노드 네트워크 관리 에이전트

센서네트워크의 각 노드는 다양한 H/W와 S/

W의 결합으로 구동이 되며, 센서네트워크 기술을 기반으로 자가 구성 네트워크를 형성한다. 형성된 네트워크를 기반으로 응용서비스를 제공하기 위해서는 센서네트워크와 호스트 간의 정보교환을 위한 프로토콜이 존재하며, 규정된 프로토콜의 기반으로 응용서비스 시스템은 송/수신된 정보를 활용하여 네트워크 모니터링 및 관리를 수행한다. 이와 같은 관리 프로토콜을 구현한 노드 네트워크 관리 에이전트는 노드/네트워크 정보 등의 형상관리 및 제어, 환경정보의 송/수신을 통한 환경정보 모니터링, 전송 정보를 통한 트래픽 모니터링 및 제어, 센서노드의 S/W 관리, 전력 관리 등의 기능을 수행한다.

(2) 센위버 네트워크 관리자

센위버 네트워크 관리자는 센서노드 배치 후 센서네트워크를 효과적으로 유지, 보수, 관리하는 도구로서 두 가지 기능, 즉 노드 레벨 및 네트워크 레벨 관리기능을 제공 한다. 첫 번째, 노드 레벨의 기능으로는 노드 S/W의 구성 관리 기능, 노드의 에너지 및 로그 조회 기능, 노드의 구동환경 변경 기능, 노드의 S/W 모듈 교체(Reprogramming) 기능, 노드의 특정 S/W의 구동환경 변경 기능 등

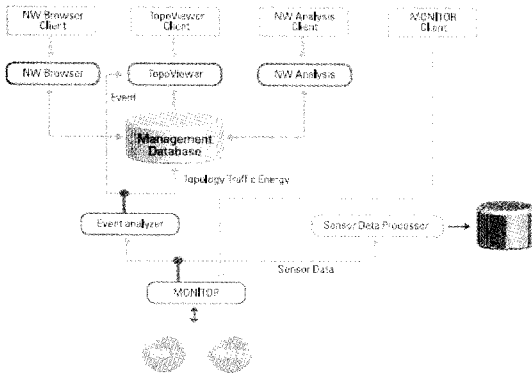


그림 6. 관리 에이전트

을 제공한다. 두 번째, 네트워크 레벨 관리 기능은 에너지 맵, 이웃노드(Neighbor node) 리스트, 토폴로지 맵, 트래픽 표시 기능을 제공한다.

4. 벤치마킹 시스템

센위버 벤치마킹 시스템은 기존의 개발노드단위로 이루어지던 성능 테스트를 대규모 센서네트워크 환경에서 벤치마킹 에이전트와 매니저를 통해 일괄적으로 제어 및 벤치마킹을 수행할 수 있으며, OS를 구성하는 각 모듈(Kernel, Network,

Application)의 동작 및 테스트 결과를 모니터링할 수 있는 시스템이다[4-5]. 벤치마킹 매니저를 통해 네트워크를 구성하는 노드들의 실행 이미지를 일괄적으로 업데이트 할 수 있어 대규모 네트워크 환경에서 각 모듈의 내재적 결함을 보다 쉽고 빠르게 발견할 수 있다. 또한, 사용자가 원하는 대규모 네트워크의 테스트 및 성능 검증을 신속하게 수행할 수 있고, 검증결과에 대한 높은 신뢰성을 확보 할 수 있다.

벤치마킹 시스템의 특징은 아래와 같다.

① 노드 단위 제어 및 모니터링: 각 개별 노드 단위로 제어가 가능하며 테스트 결과 및 각 모듈의 동작상태를 원격으로 모니터링 할 수 있다.

② 유/무선 인터페이스: 벤치마킹 시스템의 설치 환경에 따라 WiFi(802.11b)를 이용한 무선환경과 TCP/IP 모듈을 이용한 유선환경을 지원한다.

③ OS를 구성하는 각 모듈의 특정 동작 모니터링 : Kernel, Network, Application들의 동작을 각 Matrix를 수집하여 모니터링 할 수 있으며, 이는 네트워크를 구성하는 각 노드를 디버깅하는 효과

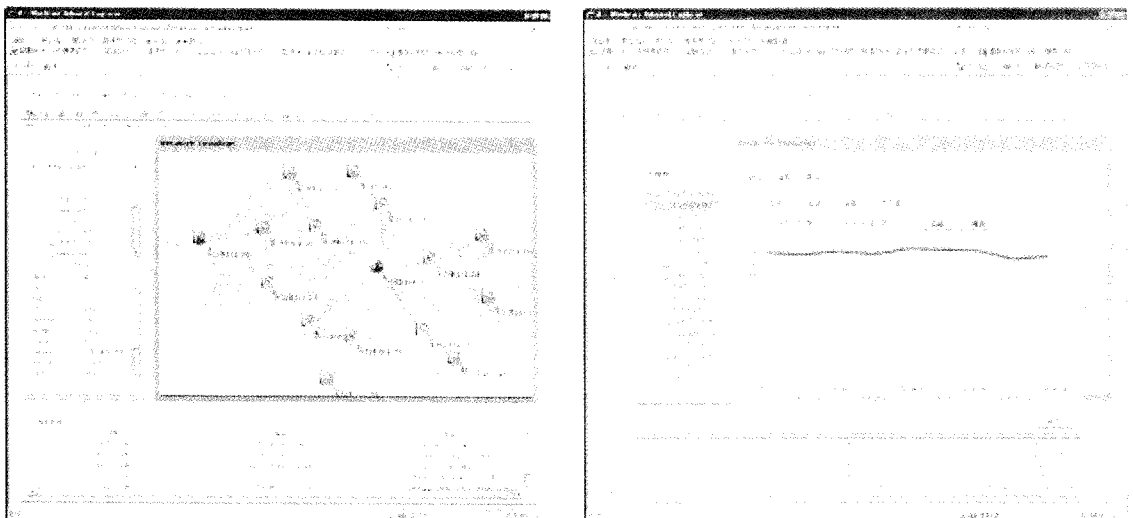


그림 7. 네트워크 관리자

를 가진다.

④ 테스트 편의성 제공 : 최대 200개 노드에 동시 리프로그래밍이 가능하며 테스트 시나리오를 작성하여 자동 테스트를 진행할 수 있으므로 사용자가 원하는 테스트를 보다 쉽고 정확하게 수행할 수 있다.

4.1 벤치마킹 H/W 시스템 아키텍처

센서버 벤치마킹 시스템은 전체 시스템 관리 및 테스트 결과 로깅을 위하여 무선 센서네트워크 외에 무선 WiFi와 유선 TCP/IP 네트워크를 사용한다. 무선 WiFi의 경우 설치와 이동성의 장점을 가지는 반면, 센서 네트워크 무선 채널과 중복되는 부분이 존재한다. 따라서 신뢰성 있는 실험 결과를 획득하기 위해서는 채널 출동 여부를 확인하여야 한다. 개발된 벤치마킹 시스템의 기본 설정은 아래와 같다.

IEEE 802.11b		IEEE 802.15.4	
ch.	Freq.(GHz)	ch.	Freq.(GHz)
1	2.401-2.423	17	2.423-2.483

WiFi 무선 채널이 1번으로 설정됨에 따라 WSN의 11~14번 채널(2.402~2.423 GHz)은 사용이 불가능하다. 외부의 WiFi 사용에 의하여 벤치마킹 설치 환경의 전파환경이 좋지 않은 경우 TCP/IP모듈을 사용하여 유선환경으로 변경 가능하다. 유선의 경우 설치 시 이동성 측면에서는 단점을 가지고 있으나 실험 결과는 WiFi에 비해 신뢰성을 가지고 있다.

4.2 벤치마킹 에이전트

벤치마킹 시스템에서 각 노드에 탑재되는 에이전트는 벤치마킹 애플리케이션으로 매니저와의

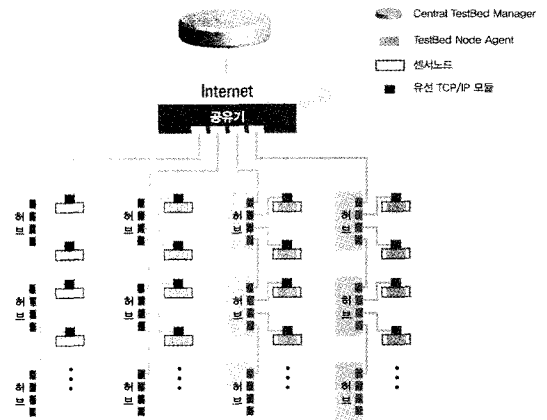


그림 8. 유선 벤치마킹 시스템 블럭도

통신을 통해 제어 및 테스트 명령을 수행한다. 센서 노드 에이전트의 주요 수행 사항은 아래와 같다.

- (1) 센서 노드 및 벤치마킹 환경 설정
 - 벤치마킹 매니저로부터 명령을 통하여 센서 네트워크 동작에 필요한 노드 주소, 토폴로지 등의 센서 네트워크 환경 설정 변경
 - 벤치마킹에 관련된 레포팅 주기, 파라미터 등의 설정 변경
- (2) 벤치마킹 수행
 - 수행 할 벤치마킹 종류에 따라 동작 스프레드 생성과 시스템 자원 할당
- (3) 벤치마킹 결과 레포팅
 - 벤치마킹 동작 설정에 따라 주기적인 결과 레포팅 또는 벤치마킹 수행 완료 후 통계 결과 레포팅 수행

4.3 벤치마킹 매니저

벤치마킹 시스템을 제어 및 모니터링 하기 위한 PC 기반의 S/W로, 센서 OS모듈(Kernel, Network, Application)의 동작 상태를 각 모듈의

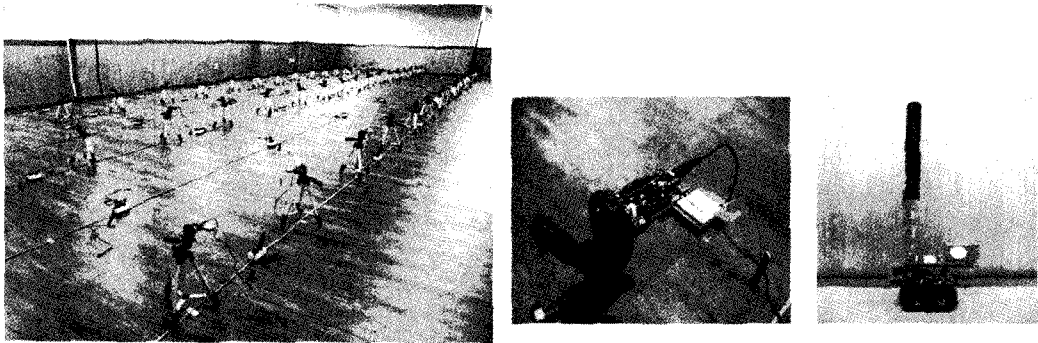


그림 9. (a)벤치마킹 시스템 설치 전경, (b)WiFi를 이용한 무선 벤치마킹 모듈, (c)TCP/IP를 이용한 유선 벤치마킹 모듈

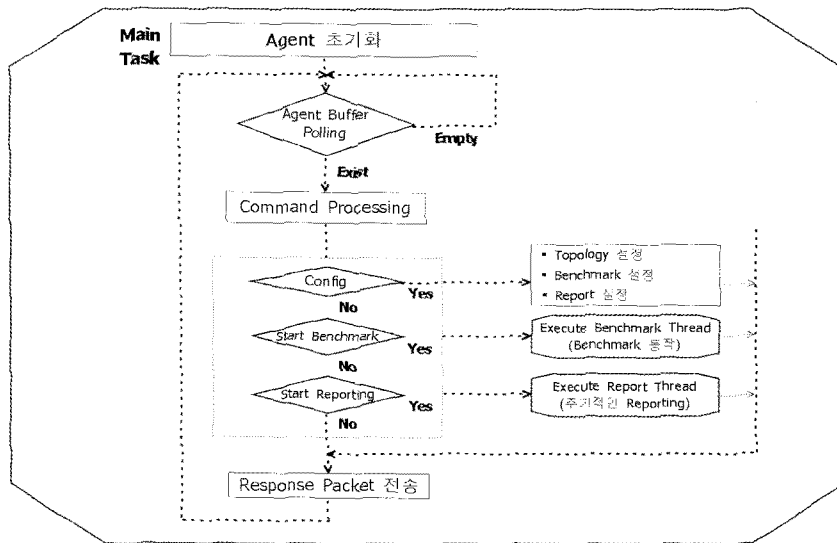


그림 10. 벤치마킹 에이전트

Metric을 통해 모니터링 할 수 있으며, WiFi 및 TCP/IP 모듈을 통해 네트워크를 구성하는 각 노드를 개별적으로 제어 할 수 있다. 또한 네트워크를 구성하는 노드의 실행 이미지를 일괄적으로 업데이트 할 수 있으며, 노드의 네트워크 토폴로지를 원경으로 설정 및 운용할 수 있다.

벤치마킹 매니저는 벤치마킹 할 테스트 OS 및 애플리케이션의 Metric을 설정과 업데이트를 수행하는 벤치마킹 Configurator, 실험 결과를 각 노드로부터 전송받아 UI를 통해 사용자에게 제공하

는 벤치마킹 Monitor 그리고 각 노드로부터 전송 받은 테스트 결과를 분석하여 보다 직관적이고 객관적인 정보로 가공 및 저장하는 벤치마킹 Analyzer로 구성된다.

5. 향후 전망

센서네트워크는 다수의 센서들이 유·무선으로 연결되어 각종 정보를 수집, 통합, 가공하고 이를 다양한 분야에 활용하는 네트워크로 미래 지능

서네트워크 플랫폼 기반의 공통기술 활용에 대한 중요성을 간과한다면 미래 신성장산업으로서 대외적 경쟁력과 산업화의 기회를 놓칠 수도 있을 것이다.

참 고 문 헌

[1] Yick, J. Mukherjee, and Ghosal, D., "Wireless sensor network survey", Computer Networks, Vol, 52, Issue. 12, pp. 2290-2330, Aug. 2008.
 [2] Mueller, R.; Rellermeier, J.S., "A Generic

Platform for Sensor Network Applications", Mobile Adhoc and Sensor System, pp.1-3, 2007.
 [3] Misun, Y., Junkeun, S., "A Flexible Sensor Network Monitoring Software", ACT, Vol. 2, pp. 1423-1426, 2007.
 [4] Kansei, Sensor Testbed for At-Scale Experiments in the Ohio State University <http://ceti.cse.ohio-state.edu/kansei/>.
 [5] Sensornets, Soda Hall wireless and sensor network testbeds in the Berkeley university. <https://www.millennium.berkeley.edu/wiki/sensornets>.



김 구 조

- 2002년 2월 대구대학교 전자공학과 공학사
- 2007년 2월 대구대학교 컴퓨터 및 통신 공학석사
- 2007년 9월 ~ 현재 대구대학교 UTRC 주임 연구원
- 2002년 9월 ~ 2004년 8월 (주) GT Telecom 연구원
- 관심분야: 임베디드 시스템, 실시간 OS



박 세 현

- 1995년 2월 경북대학교 전자공학과(학사)
- 1997년 2월 경북대학교 컴퓨터공학과 석사
- 2000년 2월 경북대학교 컴퓨터공학과 박사
- 2001년 ~ 2004년 조선대학교 컴퓨터공학부 조교수
- 2004년 ~ 현재 대구대학교 컴퓨터공학부 부교수
- 관심분야: 컴퓨터비전, 이미지해해, 패턴인식, 영상분할, 교
통감시



박 수 영

- 1991년 2월 대구가톨릭대학교 이학사
- 1996년 2월 대구가톨릭대학교 이학석사
- 2002년 2월 대구가톨릭대학교 이학박사
- 2009년 3월 ~ 현재 대구대학교 UTRC 책임연구원
- 2007년 3월 ~ 2009년 2월 경북대학교 Post Doc.
- 2001년 3월 ~ 2007년 2월 위덕대학교 조교수
- 관심분야: 무선센서네트워크, 임베디드 시스템



김 희 철

- 1983년 2월 연세대학교 전자공학과 학사
- 1990년 7월 Univ. of Southern California 컴퓨터공학 석사
- 1995년 8월 Univ. of Southern California 컴퓨터공학 박사
- 1997년 ~ 현재 대구대학교 정보통신공학부 교수
- 관심분야: : 운영체제, 무선센서네트워크, 인공지능