

IP-USN을 위한 센서 네트워크 운영체제 동향

The Trends of Ubiquitous Sensor Network OS for IP-USN

임베디드 S/W 기술 동향 특집

송준근 (J.K. Song)

센서네트워크OS연구팀 연구원

마평수 (P.S. Mah)

센서네트워크OS연구팀 팀장

목 차

-
- I. 서론
 - II. 센서 네트워크 요구사항 변화
 - III. 국내외 센서 네트워크 OS
 - IV. 결론

첨단 미래도시 건설을 위한 u시티 같은 대규모 네트워크 구축 사업에 첨단 유비쿼터스 사회를 앞당기기 위한 기술로 주목 받던 센서 네트워크를 도입하기 위한 IP-USN 기술 개발이 활기를 띠고 있다. IP-USN은 기존의 IP 인프라를 기반으로 센서 노드, 게이트웨이 및 싱크 노드 등 USN 네트워크를 통합해 광범위한 확장성과 이동성을 보장, 다양한 분야에 서비스를 적용하게 해주는 기술을 의미한다. 이처럼 인터넷 망과 연계하는 대규모 센서 네트워크 응용이 많아지면서 센서 네트워크를 구성하는 핵심 소프트웨어인 센서 네트워크 운영체제와 관련 기술 역시 그 중요성이 부각되고 있다. 이에 본 고에서는 IP-USN을 위한 기술을 지원하는 국내외 센서 네트워크 운영체제에 대해 살펴본다.

I. 서론

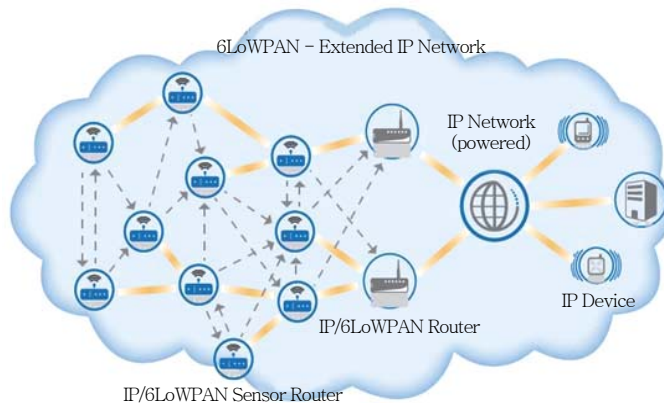
정보통신부는 2004년부터 추진해온 IT839 서비스를 도시계획 및 도시 서비스에 융·복합적으로 적용해 차세대 IT 분야의 새로운 수요창출, 성장엔진을 만들고자 하는 u시티 정책을 추진했다. 2006년 말 u시티 구축활성화 기본계획을 확정했고 이를 구체화하여 청계천, 해운대, 울산 태화강, 인천 송도 등 6개 u시티 테스트 베드 과제 지원작업을 추진하고 있다. 이에 IP 기반의 USN 통신망으로 센서노드 각각이 IP 통신이 가능하고 광대역통합망(BcN)과도 직접 연동이 가능한 IP-USN 기술이 첨단 유비쿼터스 사회를 앞당길 기술로 주목 받고 있다[1].

IP-USN은 (그림 1)에서처럼 유비쿼터스 센서 네트워크의 각 센서 노드에 IP를 부여함으로써 독립적이고 폐쇄적이던 센서 네트워크 망을 기존의 BcN, 3G, 4G, PLC 등과 같은 다양한 미디어와 통합시키는 기술이다. IP-USN은 u시티 등 대규모 네트워크에 적합한 기술로 기존 IP 인프라를 기반으로 센서

네트워크에 광범위한 확장성을 제공함으로써 다양한 서비스 제공을 가능하게 해준다[2].

IP-USN을 지원하기 위해 표준화단체인 IETF의 6LoWPAN WG이 지난 2005년부터 관련 표준화를 추진하고 있으며 최근에는 ZigBee 같은 통신기술표준화 진영에서도 센서노드에 IP를 적용하는 기술을 추진중에 있다[3],[4]. 이미 60여 년의 역사를 가져 표준이나 지재권을 선점 당한 RFID와 달리 IP-USN 기술은 비교적 새로운 기술이다. IETF에 올라가는 기술기고문 11개 중 7개가 한국산이며 특허를 가진 전세계 40여 개 업체 중 27개가 우리나라 기업 정도로 비교적 우위에 있다고 할 수 있다. <표 1>에서 볼 수 있듯이 국내 시장의 USN 성장추세는 2012년 22.46억 달러 규모가 전망되고 있으며 국가적인 이익을 위해 센서 네트워크 모든 분야, 특히 대규모 적용이 필요한 IP-USN 분야에서 국내 업체의 기술 선점이 필요하다.

유비쿼터스 센서 네트워크(USN)는 물류, 재해 예방, 군사, 홈네트워크, u시티 등 다양한 분야로 그



<자료>: ArchRock, 2007.

(그림 1) IP-USN 모델

<표 1> 국내시장의 USN 성장추세

(단위: 억 달러)

구분	내용	2005년	2006년	2007년	2008년	2009년	2010년	2011년	2012년
USN	기기	0.56	0.84	1.63	2.90	5.26	8.48	12.25	20.24
	서비스	0.02	0.06	0.21	0.34	0.54	0.86	1.36	2.21
	소계	0.59	0.90	1.84	3.24	5.80	9.34	13.62	22.46

<자료>: ETRI, 사업화전략연구팀, 2007.

적용범위를 넓혀가고 있다. 이러한 이유로 지금까지 센서 노드들 간의 통신에서만 센싱된 정보를 주고 받는 폐쇄적이던 환경에서 기존 인터넷 인프라 활용을 위한 서비스 영역으로 넓혀지고 있으며 그에 따른 센서 네트워크 운영체제의 요구사항 또한 높아져 가고 있다.

복잡한 응용을 처리하기 위해서는 다양한 작업을 동시에 처리하기 위한 멀티 스레드 환경과 메모리를 효율적으로 사용하기 위한 동적 메모리 관리 기능, 이미 분산되어 배치된 노드의 버그를 고칠 수 있는 원격 업그레이드 기능 등 다양한 기능을 지원하는 운영체제가 필요하다. 또한 6LoWPAN 같은 IP 기반의 망을 지원하기 위한 프로토콜을 적용하기 앞서 대규모 상황에서의 안정성 테스트 등이 선행되어야 한다.

본 고에서는 이러한 기반기술을 지원하기 위한 국내외 센서 네트워크 운영체제를 살펴보고 IP-USN을 위해 지원되는 기술에 대해 알아본다.

II. 센서 네트워크 요구사항 변화

그 동안 센서 네트워크는 임의의 환경에 무작위로 배치되고 독립적인 네트워크를 구성하여 센서 노드들 간의 통신을 통해 센싱된 정보를 수집하는 것이 주된 응용으로 간주되어 왔다. 그래서 8비트나 16비트의 컴퓨팅 파워를 가지는 저전력 MCU들을 많이 사용하였으며 노드의 크기나 가격을 최소화하기 위한 많은 노력이 있었다.

하지만 킬러 애플리케이션을 찾지 못한 상황에서 센싱한 데이터를 전송하는 서비스에서 벗어나 URC 로봇, LBS 등 지능적이고 대규모의 응용 서비스 개발이 시도되고 있다. 또한 IP-USN 등 외부 네트워크 망과의 결합을 고려하기 위해 응용의 복잡도가 증가하고, u시티, 홈 네트워크 같이 가정이나 건물 등 정해진 공간에 배치되는 응용이 증가하면서 그 요구사항 또한 변하고 있다[1],[5].

이러한 서비스 변화에 맞춰 우선 하드웨어적 플랫폼 요구사항이 변화하고 있다. 다양한 계층으로 구성된 네트워크 처리를 위해 빠른 MCU와 외부 메

모리가 필요하게 되었다. 기존의 처리속도가 느린 MCU와 수십 kB 미만의 메모리는 라우팅 테이블을 유지하거나 복잡한 처리를 하기에는 턱없이 부족한 메모리이다. 또한 센싱 정보의 복잡한 계산이나 보안, 상황 인지(context-aware) 같은 스마트한 센서 노드의 기능을 갖는 경우와 이미지 처리, 빠른 네트워크 처리 등을 위해서도 32비트 이상의 고성능 MCU가 요구되기도 한다.

하드웨어적인 요구사항 증가뿐만 아니라 센서 네트워크에 사용되는 소프트웨어 요구사항 또한 증가하고 있다. 센서 노드에서 사용되는 소프트웨어는 크게 운영체제와 미들웨어로 구분할 수 있다. 센서 노드의 기반이 되는 운영체제에서는 효과적인 동적 메모리 관리부터 복잡한 응용 처리를 위한 안정적인 멀티태스킹 환경, 복잡하고 다양한 센서를 쉽게 다루기 위한 플랫폼 및 센서 확장성 등이 대표적으로 요구되고 있다. 미들웨어 기술은 사용자에게 통합된 서비스 정보를 제공하는 역할을 한다. 센서 노드로부터 추출된 데이터를 필터링/통합/분석을 통해 상황정보를 추출하거나 싱크 노드로부터 모듈 갱신, 원격 업그레이드 같은 기능을 수행하기도 한다. 이러한 미들웨어 역시 대규모 상황에서 관리의 중요성의 증가와 유지, 보수 편의성을 위해 다양한 시도가 되고 있다[6].

마지막으로 무선 네트워크의 요구사항이 변화하였다. 무선 센서 네트워크에서 무선 통신 프로토콜은 반드시 필요한 필수사항으로 효율적인 라우팅, 신뢰성 있는 고장복구 및 회피방안이 요구된다. 대부분의 플랫폼의 경우 IEEE 802.15.4 호환 프로토콜을 자체 개발하여 사용하고 있으며, 국내 대부분의 WSN 업체들은 IEEE 802.15.4/ZigBee 프로토콜을 채택하여 제공하였으나, 현재는 차세대 무선 프로토콜로 IPv6를 근간으로 하는 IP-USN 프로토콜을 집중적으로 개발하는 실정이다[5].

(그림 2)는 IP-USN에서 사용되는 네트워크 스택을 보여주고 있다. 기존 TCP/IP 프로토콜을 간소화하여 대학이나 기업체에서 IPv4와 센서 네트워크와의 연동을 위한 시도가 있었으며, ZigBee 네트워

Tiny TCP/IP		6LoWPAN
USN Applications		USN Applications
Application Protocols (data transfer, management, etc.)		Application Protocols
Transport Layer (TCP/UDP)		Transport Layer (TCP/UDP)
Network Layer (IPv4 or IPv6)		Network Layer (IPv6)
		Adaptation Layer (6LoWPAN)
Logical Link Interface (IEEE 802.2 LLC)		Logical Link Interface (IEEE 802.2 LLC)
Wire-line PHY/MAC (RS-232, 422, 423, 485, CANopen, MicroLAN)	Wireless PHY/MAC (IEEE 802.15.4, Bluetooth, WLAN)	Wireless PHY/MAC (IEEE 802.15.4)

(그림 2) IP-USN 네트워크 솔루션

크와 TCP/IP를 위한 게이트웨이 또한 개발되었다 [4],[7]. 현재 IETF에서 진행중인 6LoWPAN을 통한 표준화 시도가 진행중이며 ArchRock, 아이비트 같은 회사에서 상용화를 진행하고 있다[8],[9].

Ⅲ. 국내외 센서 네트워크 OS

본 장에서는 IP 망을 지원하는 국내외 센서 네트워크 운영체제의 특징을 살펴보고, 앞서 IP-USN에 적용하기 위해 변화하는 센서 네트워크의 요구사항을 얼마나 잘 반영할 수 있는지에 대해 살펴본다.

현재 개발되어 있는 IP-USN을 지원하는 센서 노드를 위한 초소형 운영체제는 크게 두 종류로 구분할 수 있다. 우선 이벤트 기반 모델로 TinyOS와 Contiki를 들 수 있다[10],[11]. 이벤트 기반의 모델은 하나의 프로그램 흐름을 가지며 각각 이벤트에 따라 별도의 핸들러를 가지고 처리하게 된다. 또 다른 종류는 멀티 스레드 환경을 제공하는 운영체제로 NanoQplus, Nano-RT, FreeRTOS 등이 여기에 속한다고 할 수 있다[12]-[14].

전통적으로 성능상의 이점과 제어 흐름 상태 관리가 쉽다는 이유로 이벤트 기반의 프로그래밍 모델이 선호되어 왔다. 하지만 IP-USN 서비스에서처럼

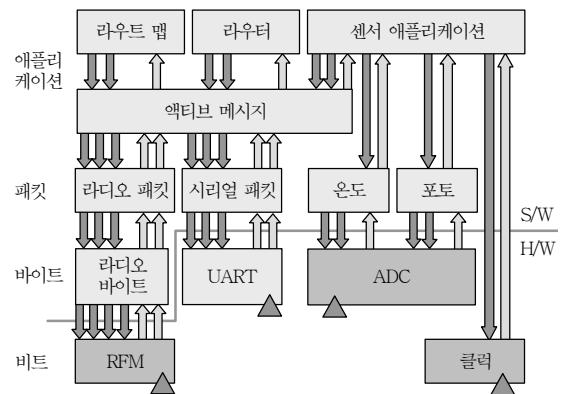
응용의 복잡도가 증가하고 다양한 작업처리를 위해서는 여러 일을 효과적으로 처리하고 개발의 복잡도를 낮출 수 있는 멀티 스레드 기반의 운영체제가 선호되고 있다.

1. TinyOS

TinyOS는 미국 UC 버클리 대학에서 개발된 센서 네트워크를 위한 운영체제로, 전 세계적으로 가장 큰 센서 네트워크 커뮤니티를 형성하고 있다. 개발자들의 활발한 참여로 빠른 기능 구현 및 업그레이드가 진행되고 있으며 다양한 플랫폼에 구현이 되어 있고 각종 하드웨어 장치 관련 컴포넌트, MAC 프로토콜, 네트워크 프로토콜, 센서 인터페이스 등을 소스 레벨에서 완전 공개하고 있다.

TinyOS는 이벤트 발생 중심의 상태 천이 방식을 채택한 센서 네트워크용 운영체제로 동시적인 프로세싱 및 제한된 하드웨어 메모리 공간에서의 효율적인 성능을 지원해주는 운영체제이다. 상태 머신 기반의 구조를 가지며 응용 프로그램은 각 독립적인 컴포넌트를 연결하는 방식으로 이루어진다(그림 3) 참조). 이 명령을 처리하는 이벤트 처리기는 그 명령에 따른 상태변화를 일으켜 해당 작업을 처리하게 된다. 간단한 FIFO 스케줄러를 사용하기 때문에 실시간성 등을 고려하기 어렵고 복잡한 응용을 작성하는 데 어려움이 있다.

동적 메모리를 할당하지 않지만 nesC라는 언어



(그림 3) 컴포넌트 기반 TinyOS 응용 예

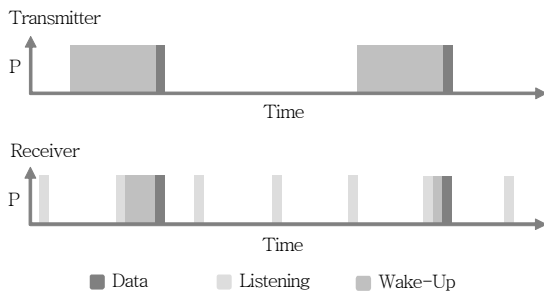
를 통하여 컴파일 시간에 컴포넌트가 요구하는 메모리 크기를 정적으로 할당하여 센서 네트워크용 응용 프로그램을 작성할 수 있다. nesC는 안정성을 위해 전체 프로그램에 대한 분석을 통해 최적화를 수행하는 컴포넌트 기반 언어로 TinyOS의 이벤트 기반 동시성 모델을 가능하게 해주지만 프로그래머가 새로 익혀야 하기 때문에 접근하기 어렵다는 단점이 있다.

TinyOS의 큰 강점으로 센싱한 데이터를 효과적으로 관리하기 위한 TinyDB나 센서 네트워크에서의 보안을 위한 TinySec 등의 모듈을 지원함으로써 사용자가 손쉽게 다양한 기능을 가진 센서 네트워크를 구현할 수 있게 지원하고 있다.

TinyOS는 2006년 11월 기존 1.1버전의 많은 부분을 개선시킨 2.0버전을 발표하였으며 현재 2.0.2 버전까지 발표되어 있다. 2.0 이후 버전의 경우 인터페이스와 추상화 과정에 있어 많은 변화가 있어 이전 버전과의 호환성은 지원하지 않지만, 코드를 업그레이드 하기가 더 쉬워졌으며 안정성이 향상되었다.

버클리대학 교수로 재직중인 David E. Culler 교수는 IP-USN에 적용하기 위해 ArchRock을 설립, CTO로 재직하면서 TinyOS에 6LoWPAN을 구현하고 RFC 4944 등의 수정에 적극 참여하는 등 관련 기술 선점에 힘쓰고 있다[8],[15].

ArchRock의 IPv6 네트워크 스택에는 IEEE 802.15.4 라디오 위에 스케줄링과 동기화가 필요 없는 (그림 4)와 같은 sampled listening의 링크 레이어 프로토콜을 사용한다. 이를 통해 노드의 에너지를 절약하고 빠른 응답성을 가지며 이동성을 가지는 응용을 지원할 수 있다.



(그림 4) Sampled Listening

2. Contiki

Contiki의 가장 큰 특징은 동적 모듈 재할당 기능을 제공한다는 것이다. Contiki는 기본적으로 TinyOS 같이 이벤트 기반 모델을 따르고 있지만, proto-thread라는 기술을 통해 제한적이기는 하지만 별도의 스택 예약 없이 멀티 스레드 같은 코드 간의 동기화를 제공한다[11]. 하지만 지역 변수를 사용할 수 없고 지정된 구간에서만 블로킹(blocking)이 가능하다는 제약이 있어 완전한 멀티 스레드 센서 운영체제라고는 할 수는 없다. 역시 센서 노드용 운영체제를 타깃으로 만들어졌기 때문에 2kB의 RAM과 40kB 정도의 ROM을 가지고 구동될 수 있다.

네트워크 스택으로는 uIP와 Rime를 지원하는데, uIP는 Contiki 개발자인 Adam Dunker이 제안한 8비트 MCU를 위한 TCP/IP 스택으로 임베디드 환경에서 인터넷을 통해 통신을 할 수 있도록 해준다[16]. uIP는 현재 1.0버전까지 나와 있으며 IP, ICMP, UDP와 TCP 프로토콜을 지원한다. 일부 기능을 제거하고 느린 MCU 등을 고려하여 디자인되었기 때문에 delayed acknowledgment 기능을 제외하면 25,000kB/s 정도의 최대전송률을 가진다. BSD socket 인터페이스와 유사한 protosocket 라이브러리를 제공하여 사용자가 쉽게 TCP 연결을 사용할 수 있도록 하였으며, DNS resolver, SMTP e-mail sender, telnet server 등 다양한 애플리케이션을 함께 제공하고 있다. Rime은 저전력 통신을 위해 구현된 멀티 홉 데이터 전송 프로토콜로 플러딩(flooding)을 통해 best-effort 브로드캐스팅을 지원하며, 컴파일된 코드가 600byte 미만의 간단한 프로토콜이다[17].

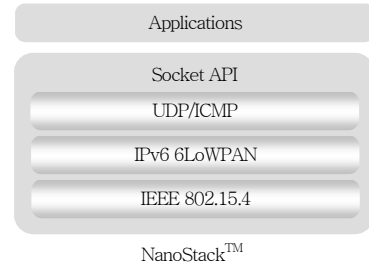
Contiki 응용을 테스트해 볼 수 있는 Cooja라는 시뮬레이션 툴을 제공하며[18], 최근 릴리즈된 2.1 버전에서는 노드의 에너지 상황을 실시간으로 모니터링 할 수 있는 기능을 지원한다[19].

3. FreeRTOS

FreeRTOS는 오픈 소스의 소형 리얼타임 커널이

다. ARM7, MSP430, AVR 계열, PIC 계열 등 14개의 상당히 많은 플랫폼을 지원하며 빠른 업데이트를 지원한다. 또한 상업적 이용에 있어서도 로열티가 없는 장점을 가지고 있다. FreeRTOS의 특징으로는 우선순위 기반 라운드 로빈 방식의 태스크 선점형(preemptive) 스케줄링을 기본으로 태스크 보다 가볍게 동작하는 양보를 통해 실행이 전환되는 협조루틴(co-routines) 기능을 부가적으로 제공한다. 협조루틴은 일반 태스크와 다르게 모두 하나의 스택을 사용하며, RAM 사용에 제약이 있다. 그 외에 메시지 큐, 세마포어 등의 IPC 기능을 제공하며 많은 예제를 통해 쉽게 프로그램에 접할 수 있다. 꾸준한 업데이트로 현재 버전 4.7까지 나와 있으며, freeRTOS를 기반으로 GPL 라이선스를 따르지 않는 OpenRTOS와 안전 계기 시스템에 적용을 위한 표준인 IEC 61508의 인증을 받은 SafeRTOS가 존재한다. 또한 플랫폼을 선정하기 좋게 MCU에 따른 비교 분석 자료를 제공하고 있으며 로열티가 없는 강점으로 다양한 확장 모듈들이 개발되고 있다[14].

FreeRTOS는 자체 네트워크 기능이 부족하여 사실상 센서 네트워크용 운영체제라고 말하기에는 무리가 있지만 사용자들이나 기업의 참여로 많은 네트워크 관련 모듈을 가지고 있다. Contiki에 사용되는 Adam Dunkels의 uIP가 포팅되어 있으며, 핀란드의 Sensinode사에서는 FreeRTOS를 바탕으로 6LoWPAN을 지원하는 NanoStack을 개발하였다((그림 5) 참조)[20].

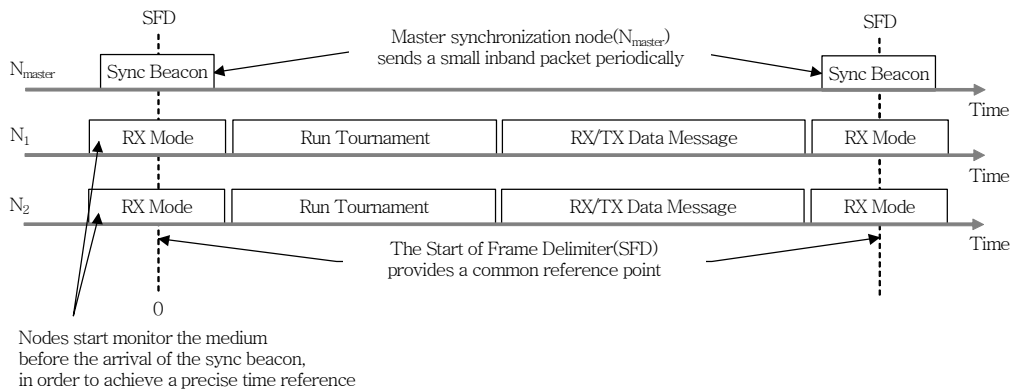


(그림 5) NanoStack Architecture

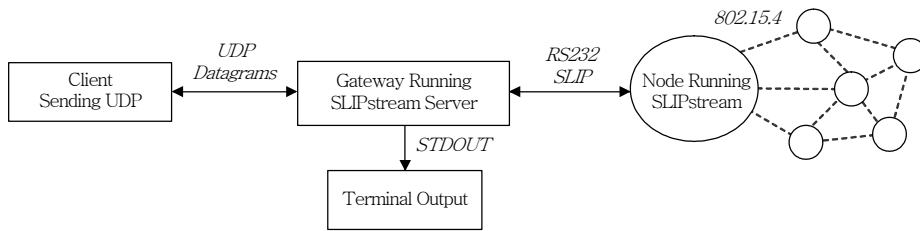
Sensinode사의 NanoStack은 TI의 CC2430 기반의 Nano 시리즈 플랫폼과 MSP430과 CC2420으로 구현된 Micro 시리즈 플랫폼에 올라가는 네트워크 스택으로 FreeRTOS 위에 구현되어 있다. NanoStack은 2007년 10월 말에 GPL 라이선스로 상업적으로 로열티 없이 공개되어 6LoWPAN을 도입하는 데 앞장서고 있다.

4. Nano-Resource Kernel

Nano-RK는 Carnegie Mellon 대학에서 개발된 자원 예약 방식(resource reservation-based)의 실시간 운영체제이다. Nano-RK는 2kB 미만의 RAM과 16kB 정도의 ROM의 작은 크기로 다양한 기능을 제공하는 RTOS로 고정된 우선순위를 가지고 선점형 멀티태스킹을 지원하며, 가상 에너지 예약을 통해 태스크 레벨의 에너지 관리가 가능하다. 또한 전원이 낮아지거나 노드가 예고없이 재시작되는 등의 예상치 않은 에러를 감지할 수 있다. 현재 2007



(그림 6) WiDom Protocol



(그림 7) SLIPstream Server

년 10월에 공개된 pre-beta 버전을 다운받을 수 있다[21].

Nano-RK는 RT-Link, WiDom 그리고 b-MAC의 다양한 링크 레이어 프로토콜을 지원한다. 충돌이 없는 TDMA 프로토콜인 RT-Link는 노드 간의 동기화를 위해 외부의 별도 하드웨어를 이용하여 out-of-band 시간 동기화와 소프트웨어적으로 in-band 시간 동기화를 지원한다[22]. WiDom은 Polytechnic Institute of Porto에서 개발된 dominance protocol로 CAN bus에서 사용되는 것과 같은 방식을 (그림 6)과 같이 무선 네트워크에 도입하였다. 또한 low-power listen CSMA 프로토콜인 b-MAC 프로토콜이 구현되어 있다[23].

라우팅 프로토콜로는 flooding, DSR을 지원하며, (그림 7)과 같은 IP 네트워크와의 통신을 위해 SLIPstream이라는 게이트웨이 프로토콜을 지원한다. SLIPstream을 이용하여 클라이언트에서 UDP 프로토콜을 이용해 센서 네트워크와의 통신이 가능하다[24].

5. NanoQplus

NanoQplus는 한국전자통신연구원에서 개발된 센서 네트워크용 초소형 운영체제이다. NanoQplus는 멀티 스레드 운영체제로 친숙한 C언어를 사용하여 사용자들이 손쉽게 멀티 스레드 응용을 작성할 수 있게 디자인되었다. 또한 완전 선점형 스케줄러를 제공하여 높은 응답성을 제공한다. 낮은 커널 이미지를 제공하기 위해 사용자 응용에 최적화된 커널 설정 기능을 통해 적은 메모리 제약을 극복하였다. 하드웨어 의존적 코드와 비의존적 코드를 분리하여

센서 노드의 포팅이 용이하며, MSP430, Atmega 128 외에도 ARM7, 8051 extended, Star 12 등 다양한 플랫폼을 지원한다. 수많은 샘플 프로그램과 많은 대학 기관, 업체의 검증 및 피드백을 통해 안정성을 확보하였다. 또한 NanoEsto라는 개발환경을 제공하며 무선 환경에서 원격으로 소프트웨어를 업그레이드 하는 원격 업그레이더와 센서 네트워크 모니터링 소프트웨어 등의 관리 소프트웨어를 지원한다[12].

300byte 미만의 RAM을 사용하는 NanoMAC과 100개 이상의 대규모 환경에서도 사용 가능한 on-demand 방식의 RENO 라우팅 프로토콜을 제공하여 센서 네트워크 상황에서 통신을 위한 에너지 소비를 최소화하고 있다.

이미 IPv4 네트워크와 통합을 위한 게이트웨이가 개발되어 있으며, 현재 자체 개발한 IEEE 802.15.4 프로토콜 위에 IPv6를 지원하는 표준 프로토콜인 6LoWPAN 기술을 개발하였다.

IV. 결론

본 고를 통해 IP-USN을 위한 요구사항과 센서 네트워크 운영체제 기술에 대해 살펴보았다. <표 2>에서처럼 많은 센서 노드 운영체제가 uIP, 6LoWPAN 등의 기술을 적용하며 다가올 IP-USN 시대를 준비하고 있다.

2004년부터 IETF에서 진행되어온 6LoWPAN은 국내에서 삼성전자, 아주대학교, ETRI 등이 적극 참여하면서 다양한 기술을 제안하고 있다. 하지만 TinyOS 진영에서 ArchRock을 통해 6LoWPAN

〈표 2〉 센서 네트워크 운영체제 비교

	동적 모듈	실행 모델	지원 플랫폼	IP-USN 관련 모듈	네트워크 지원	관련 도구/툴
TinyOS	미지원	컴포넌트 베이스 이벤트 드리븐	Atmega128, 3 ARM7, MSP430, PXA271 등	6LoWPAN (ArchRock)	B-mac, Flooding	TOSSIM, TOSVIS, TinyDB, TinySEC
Contiki	지원	Protothread 기반 이벤트 드리븐	MSP430 AT91SAM7s	uIP protosocket	uIP, Rime	COOJA
Nano-RK	미지원	Reservation 베이스 멀티 스레드	Atmega32 Atmega128	SLIPstream	RT-Link, WiDom b-MAC, DSR	-
FreeRTOS	미지원	멀티 스레드	MSP430, AVR CC2430, ARM7, PIC 등	uIP, NanoStack (Sensinode)	IEEE 802.15.4, 6LoWPAN	
NanoQplus	미지원	멀티 스레드	MSP430 Atmega128 CC2430, Star12, ARM7	IPv4 Gateway, 6LoWPAN	IEEE 802.15.4, ZigBee, 6LoWPAN, NanoMAC RENO routing	NanoESTO, NanoMON, 원격업그레이더

RFC 4944 기술을 적용하였고 그에 따라 RFC 4944 표준은 상당 부분 수정되었다[13]. 또한 ArchRock과 CISCO의 적극적인 표준 참여로 IETF 내에 센서 네트워크를 위한 라우팅 워크 그룹을 추진중에 있어 미국 기업의 주도적인 표준화 추진이 우려되고 있다 [2],[13].

한국정보사회진흥원(NIA)은 2005년부터 다양한 비 IT 분야에서 센서를 활용한 미래 u-IT 신기술 서비스에 대해 현장시험 및 시범사업을 추진했다. USN 기반 지하수 모니터링 시스템과 U-올림픽·독도 재난/재해 조기예보 시스템 등 IP-USN 기술의 상용화를 촉진하고 있다. 또한 IP-USN 포럼을 중심으로 한국형 IP-USN 제품 개발에 힘을 모으고 있다. 카서의 B-CDMA(Binary CDMA)를 이용해

모듈을 제작하고 IP-USN OS인 NanoQplus를 중심으로 아이비트의 IPv6 프로토콜 기술을 적용함으로써 실질적인 IP-USN 핵심 기술 확보를 눈앞에 두고 있다.

센서 노드에 IP를 활용하는 시장은 전 세계적으로 커지고 있다[1]. 다양한 시범사업을 통해 IP-USN 기술력을 확보하고 있는 시점에서 IP-USN 표준을 다루는 6LoWPAN WG는 상대적으로 많은 한국기관들이 참여하고 있다. 앞으로 국내 기술이 세계 IP-USN 시장을 주도하기 위해 센서 네트워크나 메시 라우팅 등 관련 연구를 하는 국내 그룹들의 보다 적극적인 연구와 표준화 참여가 필요하다.

약어 정리

● 용어해설 ●

IP-USN(Internet Protocol-Ubiquitous Sensor Network): 센서 네트워크와 인터넷망 간의 연동을 통해 인터넷 인프라 기반의 광범위한 확장성을 제공하고 센서노드, 게이트웨이 및 싱크 노드의 이동성을 보장하는 USN 기술 및 서비스

6LoWPAN(IPv6 over Low power WPAN): IEEE 802.15.4를 PHY/MAC으로 하는 저전력 WPAN에 IPv6를 탑재하여 네트워크를 직접 연동하기 위한 기술

6LoWPAN	IPv6 over Low power WPAN
IP	Internet Protocol
LBS	Location-Based Service
MAC	Media Access Control
MCU	Micro Control Unit
nesC	Network Embedded System C
RTOS	Real-Time Operating System
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

URC Ubiquitous Robotic Companion
 USN Ubiquitous Sensor Network
 WG Working Group

참고 문헌

- [1] 오세근, “최근 IP-USN Trend와 발전 전망,” 주간기술동향, 통권 1300호, 2007년 6월, pp.12-20.
- [2] 김은숙, 김용운 “6LoWPAN 기반의 IP-USN 기술표준화 동향,” 전자통신동향분석, 제22권 제6호, 2007년 12월, pp.24-32.
- [3] “IPv6 over Low power WPAN WG(6lowpan),” <http://www.ietf.org/html.charters/6lowpan-charter.html>
- [4] Alexandra Dmitrienko, “ZIGBEE-TO-TCP/IP GATEWAY: NEW OPPORTUNITIES FOR ZIGBEE- BASED SENSOR NETWORKS,” IWES 2007, Sep. 2007.
- [5] 김석우, “센서 네트워크 연구개발 및 상용화 사례,” 주간 기술동향, 통권 1325호, 2007년 12월 5일, pp.1-14.
- [6] David E. Culler, “Building Wireless Sensor Applications as Wireless Web Services,” The Industrial Wireless Book, July 2007.
- [7] Shu Lei et al., “Connecting Sensor Networks with TCP/IP Network,” International Workshop on Sensor Networks(IWSN 2006), Jan. 2006.
- [8] Archrock corporation, “A Sensor Network Architecture for the IP Enterprise,” IPSN 07, Apr. 2007.
- [9] “아이비트, IP-USN 모듈 및 라우터 장비 개발,” 전자신문, 2006년 11월 21일.
- [10] P Levis et al., “TinyOS: An Operating System for Sensor Networks,” Ambient Intelligence, Springer 2005, pp.115-148.
- [11] Adam Dunkels, “Protothreads: Simplifying Event-Driven Programming of Memory-Constrained Embedded System,” *ACM Sensys 2006*, Boulder, Colorado, Nov. 2006.
- [12] “nanoQplus,” <http://www.qplus.or.kr>
- [13] A. Eswaran, A. Rowe, and R. Rajkumar, “Nano-RK: An Energy-Aware Resource-Centric Operating System for Sensor Networks,” *IEEE Real-Time Systems Symp.*, Dec. 2005.
- [14] “freeRTOS,” <http://www.freertos.org>
- [15] 박수홍, “미국 주도의 센서 네트워크 라우팅 기술 표준화에 대한 국내 대처 방안 시급,” 한국정보통신기술협회, 2007. 11.
- [16] Adam Dunkels, Juan Alonso, and Thiemo Voigt, “Making TCP/IP Vialbe for Wireless Sensor Networks,” EWSN, 2004.
- [17] Adam Dunkels, “RIME - A lightweight Layered Communication Stack for Sensor Networks,” *EWSN 2007*, Netherlands, Jan. 2007.
- [18] Fredrik Osterlind et al., “Cross-Level Sensor Network Simulation with COOJA,” *Sense App 2006*, Florida, USA, Nov. 2006.
- [19] Adam Dunkels et al., “Software-based On-line Energy Estimation for Sensor Nodes,” *In Proc. of the Fourth Workshop on Embedded Networked Sensors(Emnets IV)*, Cork, Ireland, June 2007.
- [20] “SensiNode,” <http://www.sensinode.com>
- [21] A. Eswaran, A. Rowe, and R. Rajkumar, “Nano-RK: An Energy-Aware Resource-Centric Operating System for Sensor Networks,” *IEEE Real-Time Systems Symp.*, Dec. 2005.
- [22] A. Rowe, R. Mangharam, and R. Rajkumar, “RT-Link: A Time-Synchronized Link Protocol for Energy Constrained Multi-hop Wireless Networks,” *SECON 2006*, Sep. 2006.
- [23] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Wireless Sensor Networks,” *Sensys 04*, Nov. 2004.
- [24] “SLIPstream,” <http://www.nanork.org/nano-RK/wiki/SLIPstream>