

TMO모델 기반의 실시간 센서 네트워크 운영체제의 설계

이재윤⁰ 이재안 김광 허신
 한양대학교 컴퓨터공학과

{jylee⁰, jalee, kkim, shinheu}@cse.hanyang.ac.kr

Design of Real-Time Operating System for Sensor Network based on TMO Model

Jae-Yoon Lee⁰ Jae-An Lee Kwang Kim Shin Heu
 Dept. of Computer Science & Engineering, Hanyang University

요 약

무선 센서 네트워크는 최근 대두되고 있는 유비쿼터스 컴퓨팅의 연구에 힘입어, 군사용의 목적으로만 사용되던 과거와는 달리 다양한 분야에서 많은 연구가 진행되고 있다. 센서 네트워크는 제한적인 자원을 가진 노드에서 데이터 감지로 인한 이벤트, 데이터 프로세싱 그리고 노드들 간의 통신이 동시에 발생하므로 병렬처리 기능과 실시간성을 가진 운영체제가 필수적으로 요구된다. 하지만 현재 연구되고 있는 센서 네트워크를 위한 운영체제들은 이러한 요구를 만족시켜 주지 못하고 있는 실정이다. 우리는 본 논문에서 정시보장, 분산환경의 특징을 갖는 분산 실시간 객체모델인 TMO 모델을 적용하여 실시간성을 보장해주는 센서네트워크 운영체제를 제안 한다.

1. 서론

센서 네트워크는 분산되어있는 센서 노드로부터 감지하고 프로세싱된 데이터들을 네트워크를 통해 전송하여 사용자가 이를 활용할 수 있게 해주는 기술이다. 과거에는 군사용이나 과학용의 목적으로만 사용되어 왔으나, 하드웨어와 무선통신 기술의 눈부신 발전으로 인하여 다양한 분야에 활용될 수 있게 되었으며, [1] 특히 최근에 유비쿼터스 컴퓨팅의 핵심 기술로 주목 받고 있어 세계적으로 많은 연구가 진행되고 있다.

센서 네트워크는 그림 1과 같이 정밀한 데이터를 감지하기 위해 많은 센서 노드들로 구성되며 이러한 노드들은 일정 범위의 지역에 배치된다. 감지된 데이터는 센서 노드들 간의 통신을 통해 가공되거나 Sink 노드를 거쳐 외부 네트워크로 전송된다. 센서 노드는 노드의 크기, 계산 능력, 배터리, 메모리와 같은 자원에 대해서 큰 제약이 있

으며 [2], 센서 네트워크를 수 만개의 작은 자율적인 노드들로 구성된 분산 컴퓨팅 플랫폼으로 보는 데이터 중심형 프로그래밍 스타일이 일반적이다. 따라서 이러한 센서 네트워크 고유의 특성을 고려한 운영체제가 필요하다.

센서 네트워크의 운영체제는 제한적인 자원 환경을 가진 센서 노드에서 수행되어야 하므로 크기가 작아야 하고, 운영체제 자체 및 노드들간의 통신에 사용되는 전력의 소모가 적어야 하며 프로세스와 메모리의 관리가 효율적으로 설계되어야 한다. 또한 데이터 감지로 인한 이벤트, 데이터 프로세싱과 노드들간의 통신이 동시에 발생하므로 이들을 처리할 수 있는 병렬처리 기능과 실시간성이 필수적으로 요구되며 분산형, 데이터 중심형 프로그래밍 모델을 지원해야 한다.

TMO모델은 정시보장, 분산환경의 특징을 갖는 분산 실시간 객체모델이다. 우리는 앞에서 보았던 센서네트워크 운영체제의 조건들을 만족시켜 주기 위해 TMO모델을 적용시켜 설계한 센서 네트워크 운영체제를 제안하고자 한다.

2. 관련연구

2.1 센서 네트워크 운영체제

지난 몇 년 동안, 세계적으로 무선 센서 네트워크를 위한 시스템 소프트웨어 개발에 많은 관심이 있었다. 무선 센서 네트워크를 위해 디자인된 가장 유명한 운영체제는 "NEST" 라는 DARPA 프로젝트의 수행 결과물로 개발된 버클리 대학의 TinyOS [3]이다. 그것은 이벤트 발생에

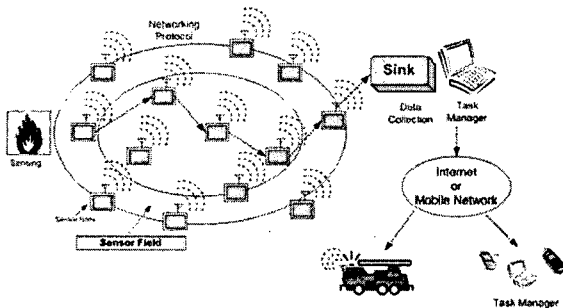


그림 1. 무선센서 네트워크 구조

의한 상태 천이 방식을 채택한 상태 머신 기반의 프로그래밍 개념을 사용한 운영체제로 제한된 메모리 공간의 효율적인 이용 및 프로세싱의 동시성 등을 지원한다. TinyOS는 재사용 가능한 소프트웨어 컴포넌트 기반의 운영체제이고, 상태 머신 기반의 구조를 가지는 운영체제로 각각의 상태는 TinyOS의 컴포넌트가 해당된다. 또한 작은 크기 RAM의 필요, 빠른 문맥전환, 그리고 power aware operation의 특징을 가지고 있다. 그러나 TinyOS는 작업의 실시간성에 대한 고려 없이 설계되었기 때문에 실시간성을 보장해 주지 못한다는 단점이 있다. TinyOS 이외에 연구되고 있는 센서네트워크용 운영체제는 초소형 스트레드에 기반한 멀티쓰레드 구조를 채택한 미국 콜로라도 대학의 MENTIS OS[4]와 센서 노드간 협력작업을 위한 메시지 통신을 강조한 EU의 EYES 프로젝트[5]에 관련된 PEEROS[6]가 있으며, 최근 우리나라의 ETRI에서 개발된 Nano Qplus[7]가 있다.

2.2 TMO

TMO모델은 Time-triggered Message-triggered Object의 약자로서 Kane Kim등에 의해서 개발된 Object Structuring Scheme이다. 이 모델은 정시보장, 객체지향, 분산환경 등의 특징을 갖는 분산 실시간 객체모델이며 1990년대 초반부터 세계적으로 주목 받고 있는 새로운 프로그래밍 패러다임이다. 일반적으로 복수개의 TMO 네트워크로 설계된 시스템은 분산 환경에서 시간 조건에 의해 수행된다. TMO모델은 실시간 시스템의 엔지니어가 디자인 단계에서부터 실시간성을 제공할 수 있도록 해주며, 실시간 시스템의 추상화에 대한 단순성을 제공해준다. 그리고 실시간 시스템의 시간적인 분석을 쉽고 간단하게 해준다. 즉 TMO는 정시보장 컴퓨팅(Timeless guaranteed computing)의 개념을 위해 제안된 분산 실시간 객체로 다음과 같은 특징을 갖는다.[8,9,10,11]

- (1) 일반적인 객체의 멤버 형태 이외에 새로운 두 가지 형태의 실시간 멤버 스트레드를 가질 수 있는 동적 객체이다.
- (2) TMO 객체 내의 자료 멤버를 공유하는 멤버 스트레드는 주어진 시간 조건에 의해 동작하는 SpM(Time-triggered Method : Spontaneous Method)과, 분산 및 국부 IPC 메시지의 수신에 의해 동작하는 SvM(Message-triggered Method : Service Method)의 두 가지 형태가 있다.
- (3) SpM은 주기에 의해 구동되고 구동 시작 후에는 주어진 finish deadline에 의해 scheduling된다. SvM은 IPC event 메시지의 수신에 의해 구동되고 구동이 시작되면 주어진 deadline에 의해 scheduling 된다.
- (4) BCC(Basic Concurrency Control) : SpM과 SvM이 동시에 내부데이터를 접근하였을 때에는 SpM이 우선 접근한다.TMO는 경성 또는 연성 실시간 응용뿐만 아니라 일반적인 분산 병행 프로그램 응용에도 사용할 수 있는 객체 모델이다.
- (5) 경성 실시간 응용의 경우, TMO는 SpM에 의한 설계 시 시간 보장 개념을 제공한다.

3. TMO모델 기반의 실시간 센서 네트워크 운영체제

센서네트워크는 군사용 목적뿐 아니라 과학적, 의학적, 상업적 용도 등 그 응용분야가 다양하게 확장되고 있다. 센서네트워크 시스템은 정해진 시간 이내에 사용자 또는 관련된 센서 네트워크 응용 소프트웨어가 탑재된 장치에 정보를 제공해줄 수 있어야 그 의미가 있다. 그렇기 때문에 시간 제약조건을 만족 시켜주는 것은 센서네트워크를 위한 운영체제의 중요한 요소라고 할 수 있다.

예를 들면, 원자력 발전소에서 방사능 유출을 감시하는 센서 네트워크 시스템에서 방사능 유출을 감지했다면 되도록 빠른 시간 안에 제어 시스템에서 그에 따른 조치가 취해져야만 하고 늦으면 늦을수록 방사능 유출에 따른 피해는 커지게 된다. 그러므로 이러한 시스템에서 사용되는 센서네트워크 시스템은 실시간성을 만족시켜 줄 수 있어야 한다.

센서 네트워크 시스템에 이용되는 하나의 노드에서는 데이터의 감지, 프로세싱, 전송 작업과 다른 노드들로부터 다음 노드로의 메시지의 전달 작업이 동시에 일어나므로, 센서네트워크의 운영체제에서 필요한 실시간성은 (1) 하나의 노드에서 동시에 일어나는 작업들의 우선순위에 따른 실시간 스케줄링과, (2)분산 환경에서 다른 노드로부터 받은 메시지를 유효한 시간 내에 처리할 수 있는 능력이 필요하다. 하지만 그러한 실시간성을 만족시켜주는 센서네트워크를 위한 운영체제는 2.2에서 보았던 것과 같이 아직까지 개발되지 않았다. 본 논문에서 우리는 (1), (2)의 요구사항을 만족시켜 주기 위하여 TMO모델을 기반으로 한 센서 네트워크 운영체제를 제안 한다.

3.1 운영체제 구조

본 논문에서 제안 하는 TMO 모델 기반의 센서 네트워크

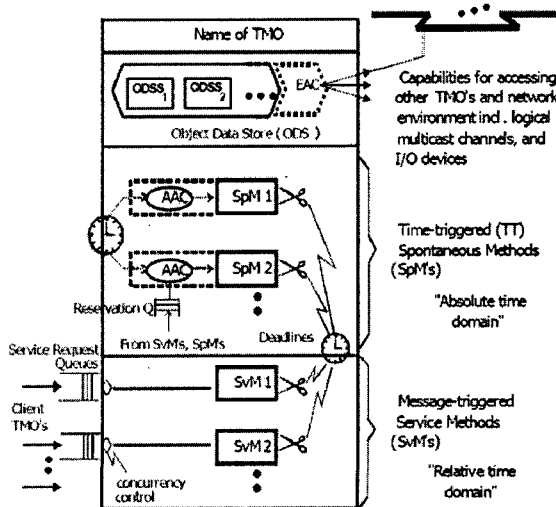


그림2. TMO 모델의 기본구조

를 위한 운영체제의 구조는 그림 3과 같이 실시간 스케줄러를 포함한 커널과 TMO 어플리케이션을 지원할 TMO System API 그리고 모듈형식의 디바이스 드라이버들로 구성되며 다음과 같은 특징이 있다.

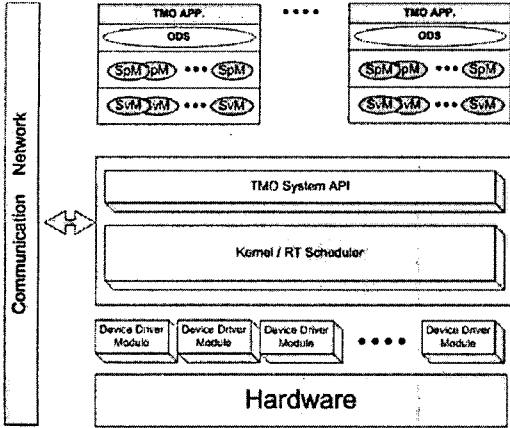


그림3. TMO기반의 OS 구조

커널은 전통적인 운영체제에서 사용되는 모든 기능들을 포함하지 않고 센서 네트워크를 위한 운영체제에 필요한 최소한의 핵심 기능들만을 수행하며, RT 스케줄러는 clock interrupt handler에 의해 주기적으로 호출되며, TMO 응용프로그램의 실행중인 SpM 및 SvM들을 데드라인 기반의 우선순위로 실시간 스케줄링 한다. TMO System API 에서는 메시지들의 처리를 위한 IPC 및 동기화 등을 위한 시스템 콜이 제공되며, device driver는 다양한 하드웨어 환경에의 이식성을 높이기 위해 모듈형식으로 구성된다.

4. 결론

센서 네트워크를 위한 운영체제에 실시간성은 대단히 중요한 요소임에도 불구하고 TinyOS 등 기존에 연구되어진 운영체제들에서는 고려되지 않고 있다.

본 논문에서 우리는 센서 네트워크 시스템의 운영체제에 실시간성이 중요한 이유를 설명하였고 실시간성을 보장해 주기 위해 TMO 모델을 적용한 센서 네트워크 운영체제를 제안하였다. 이 운영체제를 사용하면 노드에서 동시에 일어나는 작업들의 우선순위에 따른 실시간 스케줄링이 가능하며, 분산 환경에서 다른 노드로부터 받은 메시지를 유효한 시간 내에 처리할 수 있게 해줌으로써 시스템의 실시간성을 만족 시켜 줄 수 있다.

5. 참고문헌

[1] C. Chong and S. P. Kumar. "Sensor Networks: Evolution, Opportunities, and Challenges." *Proceedings of the IEEE*, Vol 91, No. 8, August 2003.
 [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "A Survey On Sensor Networks", *IEEE*

Communications, August 2002, pp. 102-114.
 [3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, *ASPLOS 2000*.
 [4] H. Abrach, S. Bhatti, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, R. Han, "MANTIS: System Support for Multimodal Networks of In-situ Sensors", *2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA) 2003*, pp. 50-59.
 [5] EYES: (webpage) <http://eyes.eu.org>
 [6] Mulder, J.: "PEEROS - Preemptive EYES Real-Time Operating System." *Master's thesis, Faculty of EEMCS, University of Twente (2003)*
 [7] Nano Qplus: (webpage) <http://www.etri.re.kr/>
 [8] Kim, K.H. (Kane) and Kopetz, H., "A Real-Time Object Model RTO.k and an Experimental Investigation of Its Potentials", *Proc. COMPSAC '94 (IEEE Computer Society's 1994 Int'l Computer Software & Applications Conf.)*, Nov.1994, Taipei, pp.392-402.
 [9] Kim, K.H.(Kane), "Towards New-Generation Real-Time Object-Oriented Computing" , *Proc. FTDCS '95 (IEEE Computer Society's 5th Workshop on Future Trends of Distributed Computing Systems)*, Aug 1995, Cheju Island, pp.520-529
 [10] Kim, K.H., "A Utopian View of Future Object-Oriented Real-Time Dependable Computer Systems" , *(Invited Paper) Proc. 1st Int'l Workshop on Real-Time Computing Systems and Applications*, Dec 1994, Seoul, pp.59-69
 [11] Kim K.H. et al., "Distinguishing Features and Potential Roles of the RTO.k Object Model" , *Proc. 1994 IEEE CS Workshop on Object-Oriented Real-Time Dependable Systems (WORDS)*, Oct 1994, Dana Point, (Proceedings to be published in June, 1995, A draft version in the preliminary workshop proceedings)