센서 노드 플랫폼에서 실시간을 적용한 DVS 기법 연구

김영만*, 김태훈*, 탁성우*
*부산대학교 컴퓨터공학과
e-mail : ssomai@gmail.com, ninkth@hotmail.com, swtak@pusan.ac.kr

A Study on Dynamic Voltage Scaling Mechanism with Real-Time on Sensor Node Platform

Youngmann Kim*, Taehoon Kim*, Sungwoo Tak*
* Dept. of Computer Science & Engineering, Pusan University

요 약

센서 노드를 위한 운영체제는 제한된 시스템 자원 하에서 동작하므로 전력 소모량을 최소화 시킬 수 있는 시스템 레벨의 저전력 기법과 함께 실시간성을 지원해야 한다. 이에 본 논문에서는 저전력 마이크로프로세서인 ATmega128L 기반의 센서 노드 하드웨어 플랫폼을 설계하고, 센서노드 플랫폼에서 동작하는 멀티스레드 기반의 실시간 운영체제인 RT-UNOS 를 개발하였다. 제안한 센서 노드 플랫폼의 동작 검증을 위하여 기존의 센서노드용 운영체제인 TinyOS 와 MANTIS, cc-EDF 와의 성능을 구현한 센서노드 상에서 실험을 진행하여 비교 분석하였다.

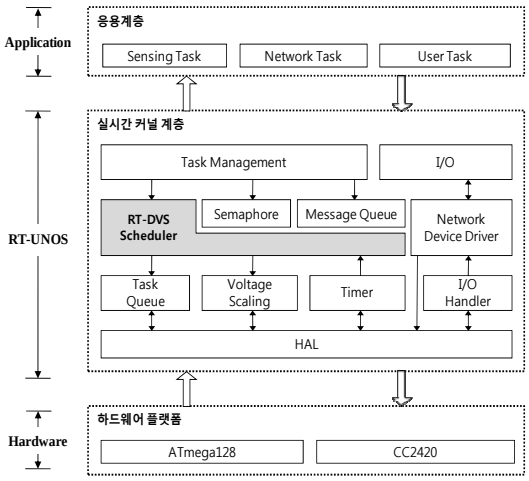
1. 서론

센서 네트워크를 구성하는 센서노드의 핵심 기술 요소로는 하드웨어 플랫폼과 센서노드에 들어가는 소프트웨어 플랫폼 기술이 있다. 센서노드의 하드웨어 플랫폼은 기술적, 경제적 제약 사항을 해결하기 위해 소형 메모리가 내장된 저전력, 초소형 프로세서를 이용하며, 소규모 데이터 전송에 적합한 지그비(Zigbee) 등의 무선 네트워크를 사용한다. 또한 필요에 따라 온도, 조도, 습도 등과 같이 다양한 종류의 센서 장비를 지원해야 한다. 소프트웨어 플랫폼에는 하드웨어 플랫폼의 효율적 관리와 실시간 처리를 지원하는 센서노드용 운영체제가 있다. 센서노드를 위한 운영체제는 범용 운영체제와 달리 특별한 요구사항들이 고려되어야 한다. 이벤트에 대한 반응이 일정한 마감시한 내에 수행되는 실시간 처리 능력이 필요하다. 또한 극도로 제한된 시스템 자원 하에서 동작하므로 전력 소모량을 최소화 시킬 수 있는 시스템 레벨의 저전력 기법도 필요하다.[1]

시스템 레벨의 저전력 설계에 관한 연구로는 현재 DPM(Dynamic Power Management) 기법과 DVS(Dynamic Voltage Scaling) 기법이 있다. DPM 기법에서는 해당 장치가 어떠한 서비스 요구도 받지 않고 있다면(즉, idle 상태일 경우) 해당 장치를 power-down 모드로 동작시키는 방식을 취한다.[2] 센서노드의 하드웨어 플랫폼에는 다양한 장치들이 있지만 최근의 연구들은 주로 프로세서의 전력 상태를 제어하는 방향으로 나아가고 있다. DVS 기법은 스케줄러가 마감시한을 만족하는 범위 내에서 프로세서의 동작 속도를 조절하여 최소한의 에너지를 소모하도록 하는 방식으로 프로세서가 자체적으로 동작속도를 조절할 수 있어야 적용 가능한 기법으로 프로세서에 의존적인 기법이다.[3]

따라서, 본 논문에서는 variable speed operating 을 지원하는 초소형 8bit 프로세서인 ATmega128(L) 기반의 센서노드 AEGIS 를 설계하고 이를 기반으로 제안한 운영체제 RT-UNOS 가 동작하는 저전력 센서노드 플랫폼을 구현하였다.

2. 실시간 센서노드 플랫폼



(그림 1) RT-UNOS 의 계층형 구조

센서 네트워크를 구성하는 각각의 센서노드는 마이크로 컨트롤러를 내장한 소형 컴퓨터 시스템으로 센

본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07 국토정보 C04)에 의해 수행되었습니다.

싱 응용처리와 노드간 통신 등을 위하여 시스템 소프트웨어를 필수적으로 요구한다. 센서노드의 마이크로 컨트롤러로 많이 사용되는 ATmega128(L)을 기반으로 설계된 멀티스레드 기반의 실시간 운영체제인 RT-UNOS 는 선점형 우선순위 스케줄링 정책을 채용하였으며 동적 전압조절 기법을 적용하여 전력소모량을 줄일 수 있는 특징을 가진다. RT-UNOS 의 계층형 구조를 (그림 1)에 나타내었다. 실시간 커널 계층은 RT-UNOS 의 핵심 부분으로 태스크 관리, Timer, 동기화를 위한 세마포어와 메시지 큐, I/O 핸들러, 디바이스 드라이버를 포함한다. 네트워크 디바이스 드라이버는 센서노드의 RF 하드웨어를 제어하고, I/O 핸들러는 외부 입출력의 처리를 담당한다. 태스크 관리 모듈에 포함되어 있는 RT-UNOS 의 스케줄러는 기본적으로 우선순위 기반으로 동작하며 EDF[4] 정책을 사용하였다.

DVS 기법은 스케줄러가 시간 제약 조건을 만족하는 범위에서 프로세서의 동작 전압을 조절하여 서비스 처리 중에 최소한의 에너지를 소모하도록 하는 방식으로, 센서노드와 같은 내장형 시스템에서 수행되는 작업의 부하는 계속해서 변하기 때문에 시스템은 항상 최고의 속도로 동작할 필요가 없다는 점에 중점을 두고 있다. 최근 DVS 알고리즘을 적용한 다양한 전압 스케줄링 기법들이 제안되고 있다[5,6]. DVS 기법은 오늘날 대다수 마이크로 프로세서에서 사용하고 있는 CMOS(Complementary Metal Oxide Semiconductor) 회로의 전력소모가 구동전압에 의존하고 있음을 이용한다. CMOS 회로의 동작 주파수를 줄이면 프로세서는 더 낮은 전압에서 동작할 수 있다. CMOS 회로에서 공급 전압이 이고 회로의 커패시턴스 부하가 일 때, 전력 소모량 는 식 (1)과 같이 근사화 할 수 있다 [5].

$$P_d \approx C_{ef} \times V_{dd}^2 \times f \qquad (1)$$

식(1)에서 는 클럭 주파수를 나타내는데 프로세서의 주파수 속도는 공급 전압에 비례한다. 그러므로 공급 전압을 1/N 로 낮추면 주파수 속도는 1/N 로 감소하고, 따라서 단위 시간당 소비 전력이 로 감소하게 된다. 이때, 전체 실행 시간은 배 늘어나게 되므로 전체 작업을 완료하는데 소모되는 전력은 로 감소하게 된다[5]. 프로세서의 동작 속도를 줄인다는 것은 작업의 응답시간 면에 있어서는 좋지 않은 결과를 가져올 수 있다. 그러나 실시간 태스크는 제한된 시간 내에 실행을 완료하는 실시간성을 요구하는 작업이므로 빠른 응답시간을 요구하지는 않는다. 이에 본 장에서는 프로세서의 전력소모량을 최소화 시키는데 중점을 두고 EDF 스케줄링 정책 기반의 DVS 알고리즘인 LRT-DVS(Lazy Real-Time DVS) 를 제안하였다. 또한, 시스템에 처리할 작업이 없는 여유시간(idle-time) 동안에는 프로세서를 유휴 상태(idle mode)로 동작하게 하였다. RT-UNOS 의 스케줄러는 기본적으로 EDF 정책을 따르며, 제안한 DVS 알고리즘을 적용하여 시스템의 전력소모량을 최소화 하였다.

3. LRT-DVS

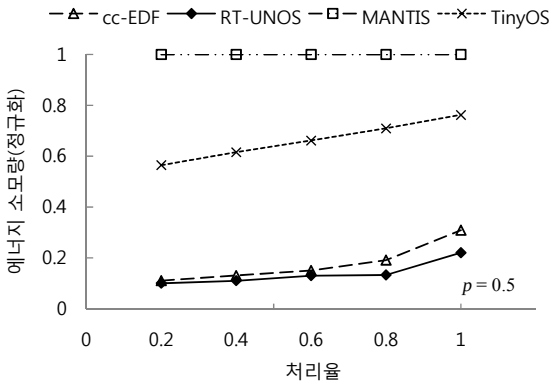
제안한 DVS 알고리즘은 2 가지 원리를 기본으로 설계되었다. 첫째, 프로세서의 전력소모량을 최소화 시키는데 중점을 두고 DPM 기법보다는 DVS 기법을 우선적으로 적용하였다. 둘째, 현재 태스크 보다 낮은 우선순위의 태스크들을 전부 최고 속도로 수행하더라도 처리 마감시한에 만족될 만큼만 느리게 수행하는 것이 유리하다. 이는 태스크들이 일반적으로 최악 실행시간 보다 빨리 완료되기 때문이다.

LRT-DVS 알고리즘에서는 현재 태스크 T_i를 제외한 모든 태스크는 최고 속도로 실행한다고 가정하고 의 주파수 조정계수 α 를 구한다. 다시 말해 이전 태스크의 이른 완료(early-completion)로 인한 여유시간을 전체 태스크에 균등하게 배분하여 활용하는 알고리즘 cc-EDF[6]와는 달리 LRT-DVS 알고리즘은 시스템의 여유시간을 현재 태스크에 모두 할당하여 현재 태스크의 주파수 조정계수를 구하고 현재 태스크가 종료되면 그 때 남은 총 여유시간을 다시 그 다음 태스크에 할당하는 방식이다. 이는 나중에 실행되는 태스크 역시 최악실행시간 보다 일찍 완료되어서 실제로는 α=1 보다 낮은 조정계수로 실행할 수 있기 때문이다. 주파수 조정계수 는 매 문맥전환 시점에 계산된다.

태스크 T_i 가 생성 될 때 이전에 수행한 태스크 T_j 가 없다면 새로운 태스크 T_i 를 큐의 제일 앞에 삽입 후 바로 실행한다. 그렇지 않으면 T_j 의 남은 마감시한과 T_i 의 남은 마감시한을 비교하여 문맥전환 실행여부를 결정한다. 만약 시스템에 실행할 태스크가 없다면 프로세서를 저전력 상태(Sleep mode)로 변경한다.

4. 실험

제안한 시스템 소프트웨어 플랫폼 RT-UNOS 의 성능 분석을 위해 이벤트 기반 운영체제와 멀티 스레드 운영체제를 각각 대표하는 MANTIS 0.9.5 와 TinyOS 1.0 을 비교대상으로 하여 실험을 진행하였다. 또한 제안한 동적 전압조절 기법인 LRT-DVS 기법의 성능을 평가하기 위해 RT-UNOS 에 cc-EDF[6] 기법을 탑재하여 비교하였다.



(그림 2) 에너지 소모량 비교

그림 2 는 항상 1 의 속도로 동작하는 MANTIS 의 경우, 처리율(U)과 실제 실행시간 비율(p)에 상관없이 항상 1 로 나타난다. 시스템 전체의 처리율이 증가할수록 전체적인 에너지소모량은 증가하고 실제 실행시

간 비율이 감소할수록 RT-UNOS 와 TinyOS, MANTIS 와의 차이가 커짐을 확인할 수 있다. 이는 MANTIS 와 TinyOS 가 p 가 감소할수록 증가하게 되는 여유시간을 활용하지 못하기 때문이다.

5. 결론

본 논문에서는 저전력 마이크로프로세서인 ATmega128L 기반의 센서 노드 하드웨어 플랫폼을 설계하고, 센서노드 플랫폼에서 동작하는 멀티스레드 기반의 실시간 운영체제인 RT-UNOS 를 개발하였다. 또한 온라인 시점에서 태스크의 남은 여유시간을 계산하여 동작 주파수를 동적으로 결정하는 저전력 태스크 스케줄링 기법인 LRT-DVS 를 제안하여 RT-UNOS 에 탑재하였다.

제안한 센서 노드 플랫폼의 동작 검증을 위하여 기존의 센서노드용 운영체제인 TinyOS 와 MANTIS, cc-EDF 와의 성능을 구현한 센서노드 상에서 실험을 진행하여 비교 분석하였다. 성능 분석 결과, LRT-DVS 를 탑재한 RT-UNOS 는 TinyOS 와 MANTIS 의 1/4 의 에너지소모, cc-EDF 보다 최대 15% 향상된 에너지효율성을 보였다.

참고문헌

[1] 김대영, 김재언, 유성은, 성종우, “USN 센서 네트워크 기술,” OSIA Standards & Technology Review, 제 25 권, 제 1 호, pp.67-77, 2005 년 12 월.

[2] Luca Benin, Alessandro Bogliolo, and Giovanni De Micheli, "A Survey of Design Techniques for System-level Dynamic Power Management," IEEE Transactions on Very Large Scale Integration Systems, Vol. 8, No. 3, pp. 299-316, June 2000.

[3] Thmoas Burd, Trevor Pering, Anthony Stratakos, and Robert Brodersen, "A Dynamic Voltage Scaled Microprocessor System," IEEE International Solid-State Circuits Conference Digest of Technical Papers, pp.294-295, Feb 2000.

[4] Cravin Mani Krishna and Kang Gun Shin, Real-Time Systems, McGraw-Hill, 1997.

[5] Mark Weise, Brent Welch, Alan Demers, and Scott Shenker, "Schduling for Reduced CPU Energy," Proc. of the 1st Symposium on Operating Systems Design and Implementation, Monterey, Canada, November 1994.

[6] Padmanabhan Pillai and Kang Gun Shin, "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems," Proc. of the eighteenth ACM symposium on Operating Systems Principles, Alberta, Canada, October 2001.