

GUI 기반의 센서 네트워크 관리 도구

정인욱^o, 차호정
연세대학교 컴퓨터 과학과
{inukj^o, hjcha}@cs.yonsei.ac.kr

A GUI-based Management Tool for Wireless Sensor Networks

Inuk Jung^o Hojung Cha
Dept. of Computer Science, Yonsei University

요 약

센서 네트워크는 라디오 통신으로 인해 생기는 예상치 못한 문제점을 비롯하여 제한적인 리소스 때문에 응용 개발에 어려움이 많다. 센서 네트워크가 구성된 후의 내부적인 데이터 흐름이나 네트워크에 참여하는 각 노드의 상태는 네트워크의 정보 수집을 통해 알 수 있다. 본 논문은 네트워크의 상태를 모니터링하고 실시간으로 네트워크 파라미터들을 효율적으로 설정할 수 있는 센서 네트워크 관리 시스템을 제안한다. Sensor Network Manager는 Multi-thread와 동적으로 할당 가능한 Kernel Module 기법을 지원하는 RETOS 운영체제를 기반으로 개발되었다. 간단한 플래딩 기법을 적용하여 센서 네트워크의 상태를 모니터링 한 결과 Sensor Network Manager가 개발자에게 전반적인 네트워크에 대한 통제와 모니터링이 가능하다는 것을 검증하였다.

1. 서 론

무선 센서네트워크는 라디오 통신과 센싱 장치들을 통해 사용자가 특정 환경에 대한 정보를 수집하는 방법으로 쓰인다. 본 분야의 응용과 하드웨어의 기술이 복잡해짐에 따라 무선 센서 네트워크의 역할에 대한 요구가 증가하고 있다. 신뢰성 있는 정보를 수집하기 위해 센서 네트워크는 기본적으로 오랜 시간 동안 문제 없이 작동되어야 한다. 하지만 제한적인 리소스와 근본적으로 불안한 라디오 통신으로 인한 문제로 네트워크의 트래픽 손실 등 여러 문제가 발생 할 수 있으며 표면적으로 확인하기 어렵다. 이에 따라 네트워크가 구성된 후 내부적인 상황에 대한 지속적인 정보 수집이 어렵고 네트워크 동작 상태에 대해서도 알기 쉽지 않다. 또한 센서 네트워크 분야의 표준화된 개발 환경이 없기 때문에 응용 개발에 어려움을 많이 겪는다.

본 논문은 위에서 설명한 불안 요소들을 지속적으로 모니터링하고 적절하게 해결 하며 효율적인 센서 네트워크 응용 개발 환경을 제공하는 센서 네트워크 관리 시스템을 제안한다. 무선 센서 네트워크의 동작 상태를 고려한 네트워크 모니터링 툴과 [1, 2, 3] 전반적인 네트워크 오류를 디버깅 하는 기법들이 [4] 활발히 연구되어왔으나 이들은 다른 응용과 함께 작동되었을 때의 효율성과 타당성을 입증하는 것이 부족하다. 또한 네트워크 모니터링은 단순히 노드가 사용자 쿼리 등에 반응하는 정도로 신뢰 할 수 있는 수준의 QoS를 보장할 수 없다. 이에 따

라 전반적인 네트워크가 정상적으로 작동하는 것을 알기 위해, 현재 네트워크에 대한 정보를 효율적으로 수집하고 적절하게 네트워크의 파라미터들을 실시간으로 설정할 수 있도록 하는 센서 네트워크 관리 시스템이 필요하다. 예를 들어 네트워크 내의 중요한 오류나 노드를 수동적으로 수정해야 할 문제점들을 해결하기 위해 개발자에게 알려져야 한다. 대규모 센서 네트워크 구성에서의 이러한 문제들을 수정하기 위해서는 많은 시간이 소요될 수 있기 때문에 더욱이 네트워크를 관리하기 위한 툴필과 네트워크 내의 오류들을 발견 및 수정 후 재배포 가능한 효율적인 개발 환경이 필요하다.

제한적인 리소스 환경에서 이러한 기능들을 제공하기 위해 연구들이 중심으로 이루어지고 있다. 본 논문에서는 네트워크 모니터링과 최적화된 환경을 위한 파라미터들을 실시간으로 설정이 가능한 센서 네트워크 관리 시스템을 제안하고 있다. 센서 네트워크 관리 시스템은 크게 전반적인 센서 네트워크를 모니터링시스템과 네트워크를 설정할 수 있는 두 가지 기법으로 이루어져 있다. 첫째는 센서 네트워크의 전반적인 상태를 싱크 노드에서 수집하고 GUI를 통해 효율적으로 이러한 네트워크 정보를 관찰 하는 기법이다. 둘째는 네트워크 모니터링 기법으로 네트워크에서 수집한 정보를 토대로 네트워크의 전반적인 상황을 파악하며 적절하게 네트워크 파라미터들을 설정해 줌으로서 네트워크 상태를 최적화 시키며 현존하는 문제점들을 해결하는 기법이다. 이를 통해 사용자는 네트워크 모니터링이 가능하며 네트워크의 전반적인 통제가 가능하며 시간 효율적인 응용 개발이 가능하다. 센서 네트워크 관리 시스템은 무선 센서 네트워크를 위한 RETOS 운영체제를 기반으로 표준 C 언어로 개발되었다 [5, 6, 7].

본 논문의 구성은 다음과 같다. 다음 2 장에서는

본 연구는 한국과학재단에서 지원하는 국가지정연구실사업으로 수행하였음 (과제번호 : 2006-01546)

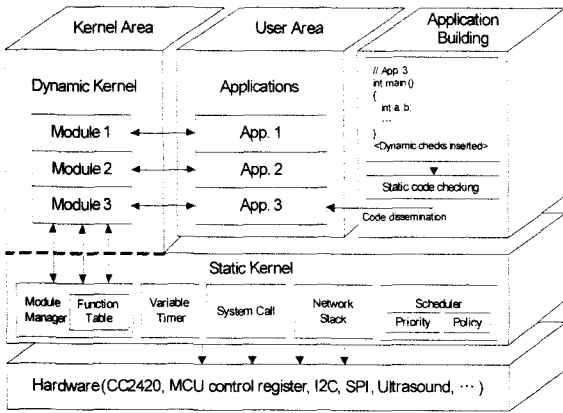


그림 1. RETOS Architecture

RETOS라는 무선 센서네트워크 운영체제에 대한 전반적인 구성을 소개 한다. 3 장에서는 네트워크 모니터링 기법에 대한 기술을 소개한다. 4 장에서는 제안한 기법의 성능을 실험으로 검증하고 5장은 결론과 향후 계획을 제시한다.

2. RETOS 개요

센서 네트워크를 구성하는 응용과 하드웨어들이 다양해 지며 복잡해짐에 따라 센서 네트워크 구성 요소와 기능들에 대한 요구사항들이 점차 증가하고 있다. 이러한 요구들에 맞추어 개발자는 응용 설계를 해야 하는 어려움이 있으며 근본적으로 지속적으로 바뀌는 네트워크 요구사항에 적합하게 대응할 수 있는 시스템이 필요하다. RETOS는 무선 센서 네트워크의 특성을 고려해서 개발되고 있는 운영체제이다. 일반 네트워크와는 달리 센서 네트워크는 여러 면에서 제한된 개발 환경을 갖추고 있으며 RETOS는 이에 따른 세 가지 원칙 아래 개발 중이다. 첫째, multi-threaded 프로그래밍 환경과, 둘째, 동적으로 할당 가능한 kernel module과, 마지막으로 3-layered 네트워크를 제공한다. RETOS는 posix 1003.1b 실시간 스케줄러를 인터페이스를 제공함으로써 스레드들이 커널 스택의 메모리를 효율적으로 접근하도록 구현되어 있다. RETOS 개발 환경은 Imote Sky [8] 무선 통신 가능한 센서 하드웨어를 기반으로 개발되고 있다. RETOS의 종합적인 구성은 그림 1에서 보여주고 있다. RETOS는 리눅스와 마찬가지로 커널 환경과 유저 환경을 분리시킨 구조로 개발되었으며, 이하 듀얼 모드 오퍼레이션으로 본 논문에서는 지칭하기로 한다. 이러한 설계의 목적은 유저가 커널 개발에 대한 부담을 줄이고 커널 개발자는 RETOS 응용과 별개로 framework에 구애 받지 않으면서 작업 할 수 있는 환경을 제공하기 위한 것이다.

RETOS 커널은 크게 정적 커널과 동적 커널 범위로 나뉜다. 정적 코드는 리소스 치명적일 수 있는 코드와 하드웨어 리소스를 관리하며, 동적 커널은 실시간으로 커널 모듈을 탑재시키는 LKM (Loadable Kernel Module)

기능을 제공 하고 있다. LKM은 효율적인 메모리 관리를 위해 불필요한 모듈들을 커널에서 제거하고 센서 네트워크의 요구 변경사항에 적절하게 필요한 모듈을 탑재시키는 역할을 한다. 3-layered 네트워크는 네트워크 프로토콜들의 효율적인 재사용을 위해 구현되어있다. 센서네트워크는 상황에 따라 네트워크 프로토콜이 바뀌는 일이 빈번함으로 적절하게 이에 맞춰서 여러 가지 프로토콜을 쓸 수 있도록 계층 구조로 설계되었다. 각 계층은 아래와 같이 각기 다른 기능들을 제공하고 있다. 첫째 레이어 MLL (MAC and Data Link Layer)은 물리적인 네트워크 연결과 데이터 전송을 하며, 둘째 레이어 NSL (Network Supporting Layer)은 이웃 노드 관리와 그들과의 데이터 전송을 하며, 마지막 레이어 DNL (Dynamic Network Layer)은 다양한 네트워크 프로토콜을 구현 또한 사용자를 위한 API를 개발하는 환경으로 쓰인다.

3. 센서 네트워크 관리 도구

다음은 본 논문에서 제안하는 센서 네트워크 관리 시스템에 대한 기능과 구현을 소개한다. 이 시스템은 사용자에게 전반적인 센서 네트워크의 모니터링과 통제를 목적으로 한다. 센서 네트워크 관리 시스템은 크게 센서 네트워크의 상태를 모니터링 하는 것과 수집한 네트워크 정보를 분석함으로 네트워크 상태를 파악하는 기법으로 구성 되어 있다.

3.1 센서 네트워크 모니터링

센서 네트워크 모니터링 기법은 하드웨어와 네트워크 리소스를 최소한으로 사용하도록 설계되었으며. 이 기법은 우선적으로 노드에서 시행 되는 타 응용 프로그램의 리소스 확보에 영향을 미치지 않은 한도 내에서 개발되었으며 커널 내에서 포함되기 때문에 센서 네트워크 응용과는 독립적으로 동작되도록 설계하였다. 이를 위해 전체적인 구현이 간단해야하며 이러한 기능들을 사용자에게 GUI를 통해 효율적으로 제공해야 한다.

우선적으로 센서 네트워크 모니터링 기법은 메모리 효율적으로 구현하기 위해 두 가지 방법이 있다. 첫째는, 하드웨어 리소스를 최소한으로 쓰도록 구현하는 것과, 둘째는 LKM을 통해 불필요한 모듈들을 커널에서 삭제함으로 전체적인 네트워크 리소스의 사용을 줄이는 것이다. RETOS는 커널 모듈의 구현과 응용단의 구현을 모듈화된 시스템으로 구현 가능하기 때문에 응용도 LKM을 통해 삭제 또는 탑재 할 수 있기 때문에 더욱 최적화된 리소스 사용이 가능하다. 센서 네트워크 관리 시스템은 이러한 모듈 기능을 최대한 효율적으로 사용하며 모듈 코드를 네트워크를 통해 온라인으로 전송함으로써 필요한 노드에 탑재 한다. 센서 네트워크 모니터링 기법은 개별적으로 사용자를 위해 프로그래밍 인터페이스를 정적 커널과 동적 커널에서 제공한다. 여기서 인터페이스는 사용자가 관심 있는 네트워크 정보를 선택하여 수집할 수 있는 것을 말하며 탑재 또는 삭제하고자 하는 모듈을 선택 할 수 있는 기능을 제공한다. 패킷 손실 또는 특정 노드의 배터리 잔량 같은 네트워크 속성들은 NSL

표 1. 네트워크 구성 요소

속성	기술	속성	기술
노드 ID	노드 고유의 ID	패킷 전송 비율	성공적인 패킷 전송 비율
위치	노드의 위치 정보	패킷 전송 시간	패킷 전송 시간
LQI	Link quality indication	ACK/NACK	ACK/NACK 사용여부
RSSI	라디오 전파 세기	배터리 레벨	배터리 잔량
CCA	Channel loading status	이웃 노드 정보	이웃 노드 테이블

에서 제공 해주기 때문에 사용자가 직접 정의 하지 않아도 되며 추가적인 정보가 필요할 시에는 사용자가 수동적으로 수집 할 수 있도록 인터페이스가 제공 되어있다. 현재 NSL에서 지원되는 네트워크 구성 요소는 표 1에서 보여주고 있다. 네트워크 분석 기법은 이러한 정보들을 수집을 하면서도 이웃 노드에 대한 연결 상태와 각 이웃 노드에 대한 정보도 관리 한다. 이에 따라 사용자는 네트워크의 토폴로지가 어떻게 구성되어있는지를 알 수 있다. 최신정보를 제공하기 위해 이러한 네트워크 정보들은 지속적으로 갱신 된다. NSL은 네트워크 또한 각 노드의 어떤 응용 프로그램과 커널 모듈들이 탑재되어 있는지에 대한 정보도 포함하고 있으므로 전반적인 네트워크가 어떠한 기능을 수행하고 있는지에 대해 쉽게 알 수 있다. 네트워크 정보를 효율적으로 관찰 할 수 있기 위해 구현한 유저 인터페이스를 그림 2에서 보여준다. 그림 3에서는 사용자가 노드에 대한 더 많은 정보가 필요할 때 총괄적으로 보여 주는 인터페이스도 제공 되어 짐으로 전체적인 정보 보다는 각 노드에 대한 개별적인 정보를 쉽게 관찰할 수 있다.

3.2 센서 네트워크 파라미터 설정

네트워크 파라미터 설정 기법은 모니터링의 정보 수집을 어느 정도의 주기로 할 것이며 또는 노드의 라우팅 경로 등을 네트워크의 현 상태에 적절하게 조절 할 수 있으며 각 노드에 응용 프로그램이나 모듈을 탑재 또는 삭제 할 수 있는 환경을 제공한다. 이에 따라 노드가 수행하는 기능과 수행 방법을 통제함으로써 네트워크의 전반적인 관리를 할 수 있으며 네트워크의 문제점을 발견 했을 당시 적절하게 문제되는 점을 중점적으로 분석하고 해결 할 수 있는 수단으로 사용 될 수 있다. 특정 노드의 응용 프로그램에 문제가 있다면 싱크 노드에서 응용 프로그램을 디버깅하여 응용 코드를 재배포하여 패치를 할 수도 있다. 이러한 통제를 위해 셸 명령어 창을 구현 하였으며 이는 노드와 직접적으로 통신을 할 수 있는 인터페이스를 제공하기 위한 목적을 가지고 있다. 예를 들어 셸 명령어 창으로 ID를 5번으로 하는 노드의 평균 RSSI 값을 1분 주기로 10분 동안 수집하라는 명령어를 아래와 같이 내릴 수 있다.

> query neighbortable 5, 10, 1

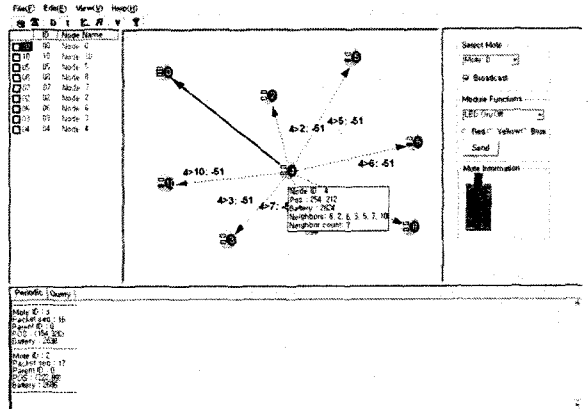


그림 2. 네트워크 모니터링을 위한 GUI

Node ID	: 8
Parent ID	: 0
Position X	: 528
Position Y	: 234
Battery	: 2962
LQI	: 231
LAST_UPDATE:	
	2006/08/31 11:33:31

그림 3. 노드 정보 표시

4 분석

다음은 센서 네트워크 관리 시스템에 대한 성능을 분석한다. 센서 네트워크 응용은 간단한 플래딩 기법을 사용 하며, 네트워크 정보 수집 시간과 네트워크 내부 오류에 대한 반응 시간을 측정하는 것을 목적으로 한다. 이것을 두 가지 실험 셋으로 나누었는데 이들은 네트워크 유저가 시행한 쿼리에 대한 반응 시간과 특정 노드의 동작 오류에 대한 반응 시간을 측정 한 것이다. 여기서 사용되는 플래딩 기법은 매 20초 마다 HELLO 패킷을 네트워크에 브로드 캐스팅을 한다. 노드들은 센서 네트워크 관리 시스템을 탑재한 총 10 홉 카운트로 10개가 배치되었다. 사용자는 10초 마다 가장 먼 노드에게 이웃 노드 테이블에 대한 쿼리 요청을 한다.

그림 3은 각 쿼리에 대한 반응 시간과 노드의 오류를 발견 하는데 걸린 시간을 측정 한 것이다. 응답 시간의 비례는 비슷하지만, 오류 반응 시간의 기울기가 홉 카운트가 늘어남에 따라 점차 완만해 지는 것을 볼 수 있다. 이러한 이유는 오류 동작을 범한 노드를 포함하는 이웃 노드들이 많다는 것과 그로 인해 싱크 까지 보내지는 각 이웃 노드들의 네트워크 정보가 쿼리 시나리오 보다 많고 좋은 라우팅 경로를 택할 수 있는 확률이 크기 때문이다. 이로 인한 네트워크 오버헤드가 증가하는 것이 단점이 될 수 있지만, 네트워크를 분석 하며 얻는 이득이 더 크다.

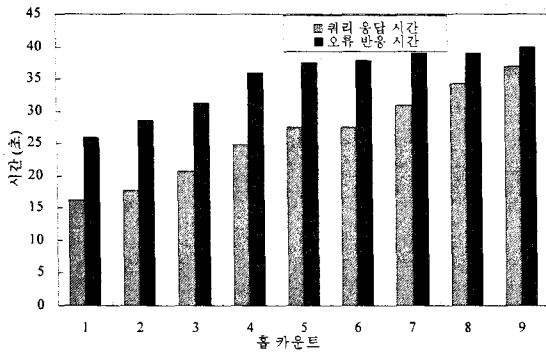


그림 3. 홉 카운트에 따른 사용자 쿼리와 노드 동작 오류에 대한 반응시간

5 결론

본 논문은 센서 네트워크 개발을 위한 효율적인 센서 네트워크 응용 개발 환경과 네트워크의 전반적인 분석을 할 수 있는 모니터링 기법을 제안하고 있으며 성능분석을 통해 활용성을 입증하였다. 제안 하는 기법은 총 2 가지 구현으로 구성 된다. 먼저 네트워크 모니터링을 통한 효율적인 센서 네트워크의 상태를 관찰할 수 있는 기법과 전반적인 센서 네트워크의 상태를 파악하는 기법을 제공 해준다. 총괄적으로 두 기법을 통해 현 네트워크 상태를 분석하고 네트워크 이벤트나 오류 동작에 대한 정보를 수집하여 그에 대한 대응책을 싱크에서 적절히 네트워크 파라미터 설정을 통해 해결 할 수 있는 환경을 제공 해준다. 이는 개발자에게 더욱 효율적인 개발 환경을 제공 해주며 네트워크의 전반적인 이해를 돕기 위한 목적을 가지고 구현 되었다.

향후 계획으로는 네트워크상에서 상주하는 사용자 응용의 상태를 파악할 수 있는 모니터링 기법을 고려한다. 예를 들어 모바일 유제트 트래킹이나 응용 라우팅 알고리즘의 동향을 사용자가 직접 관찰할 수 있는 시스템을 생각해 볼 수 있다.

6 참고 문헌

[1] G. Tolle and D. Culler, "Design of an Application-Cooperative Management System for Wireless Sensor Networks," In Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN), Istanbul, Turkey, January, 2005.

[2] J. Zhao, R. Govindan, and D. Estrin, "Sensor Network Tomography: Wireless Sensor Networks," Student Research Poster, ACM SIGCOMM 2001, San Diego, CA, August 2001.

[3] R. Szcwcyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), 2004.

[4] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E.

Kohler, and D. Estrin, "Sympathy for the Sensor Network Debugger," In Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys '05), San Diego, CA, 2005.

[5] H. Kim and H. Cha, "Towards a Resilient Operating System for Wireless Sensor Networks," In Proceedings of the 2006 USENIX Annual Technical Conference, MA, May 2006.

[6] H. Shin and H. Cha, "Supporting Application-Oriented Kernel Functionality for Resource Constrained Wireless Sensor Nodes," Technical report, April 2006.

[7] S. Choi and H. Cha, "Application-Centric Networking Framework for Wireless Sensor Nodes" In Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS), July, 2006.

[8] Tmote Sky, <http://www.moteiv.com>

[9] J. Hui and D. Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), 2004.