

센서 네트워크를 위한 응용 플랫폼에 관한 연구

송태헌*, 양서민*, 이혁준*
*광운대학교 컴퓨터공학과
e-mail : myboss7@kw.ac.kr

A Study of the Application Platform for Sensor Networks

Taeheon Song*, Seomin Yang*, Hyukjoon Lee*
*Dept. of Computer Engineering, Kwangwoon University

요 약

물리적 환경을 관리하기 위한 센서 네트워크 응용은 그룹관리, 질의 최적화, 데이터 처리 등을 포함한 특정 응용 방식을 모두 구현해야 하므로 응용 구조가 상당히 복잡해질 수 있다. 또한, 센서 네트워크의 싱크와 센서 노드간의 경로가 실패하는 경우에는 경로를 복구하는 동안 정보의 손실이나 지연이 발생하게 된다. 본 논문에서는 이를 개선하기위해 센서 네트워크 내에 배치되어 있는 다중 싱크로부터 각 센서 노드의 정보를 수집하고 각 노드를 효율적으로 관리하기 위한 GUI 기반의 센서 네트워크 응용 플랫폼을 설계하고 구현한다.

1. 서론

최근 디지털 기술의 발달로 저가, 저전력의 다양한 기능을 가진 센서 노드가 개발되면서 센서를 이용한 응용이 활발히 연구되고 있다. 센서 네트워크의 센서 노드는 사람의 접근이 불가능한 지역이나 전쟁지역 등에서 위치를 미리 설정할 필요없이 무작위 배치되므로 구성 알고리즘과 데이터 전송 프로토콜은 자기 구성기능과 협력처리기능을 가져야 한다. 센서 노드들은 응용에 해당하는 정보를 수집하고 수집된 정보를 센서 필드의 멀티 홉(multi-hop) 경로를 통해 싱크에 응답하는 기능을 가지고 있다. 싱크는 센서 노드의 정보를 통합하여 저장하거나 인터넷을 통해 정보를 관리하는 노드에게 응답한다. 센서 네트워크를 구현하기 위해서는 무선 애드 혹(ad hoc) 기술이 필요하며 기본적으로는 기존의 무선 애드 혹 네트워크와 유사한 기능을 가지고 있지만 센서 네트워크의 특성에는 적합하지 않으며 응용 목적에 따라서 여러가지 차이점을 가지고 있다[1]. 현재, 센서 네트워크 응용을 위한 플랫폼들은 단일 싱크와 센서 노드간의 효율적인 데이터 전송을 위한 알고리즘 및 센서 노드의 에너지 소모를 줄이기 위한 연구에 초점을 두고 있다[2]. 그러나, 실제 센서 네트워크에서는 다수의 싱크가 배치될 수 있으므로 센서 네트워크 응용에서는 이들 싱크들로부터 센서 노드의 이벤트를 효율적으로 관리하고 통합하기 위한 플랫폼이 필요하다. 본 논문에서는 넓은 지역에 배치된 수많은 센서 노드들을 효율적으로

관리하기 위한 GUI 를 구현하고, 싱크와 소스 노드간의 경로 실패로 인해 정보를 수신하지 못하는 문제점을 개선하기위해 다중 싱크로부터 정보를 수집하며 유무선 네트워크의 트래픽을 줄이기 위해 내부 데이터베이스를 활용하는 응용 플랫폼을 설계하고 구현한다.

본 논문의 구성은 다음과 같다. 2 장에는 기존 유무선 네트워크 환경과는 다른 센서 네트워크의 특징들을 살펴보고, 3 장에서는 본 논문에서 제안하는 센서 네트워크 응용 플랫폼을 설명하고 각 구성 요소들을 설명한다. 4 장에는 제안된 응용 플랫폼의 구현 및 실험 결과를 그래프를 통해 살펴보고 5 장에서는 결론과 향후 과제에 대해 살펴본다.

2. 센서 네트워크

센서 네트워크는 광범위하게 설치되어 있는 유무선 네트워크에 주변 환경을 감지할 수 있는 다양한 센싱 장치를 통합하여, 감지된 환경 정보들을 유무선 네트워크의 응용 서비스와 연동하는 기술이다. 센서 네트워크의 전송 전력과 라디오 범위는 블루투스나 MANET 보다 작기 때문에 이들의 MAC 프로토콜을 직접적으로 이용할 수는 없다. 그래서 센서 네트워크의 MAC 프로토콜로써 SMACS(Self-Organizing Medium Access Control for Sensor Network)[3]와 Hybrid TDMA/FDMA[4]등이 제안되었다. 또한, 센서 네트워크 프로토콜은 기존 애드 혹 라우팅 프로토콜과 유사해

서 센서 노드와 싱크 노드간에 멀티 홉 무선 애드혹 라우팅 프로토콜을 필요로 하지만 기존 애드 혹 라우팅 기법으로는 센서 네트워크에서 요구하는 모든 조건을 만족시키지 못하므로 센서 네트워크를 디자인 할 경우에는 전력 효율, 데이터 중심 질의, 정보 수집과 같은 사항들을 고려하도록 해야 한다. 센서 네트워크 응용 디자인은 센서의 특성상 전통적인 유무선 네트워크와는 다른 다음의 몇 가지 요구조건을 가지고 있다. 먼저, 전통적인 유무선 네트워크와는 달리 센서 네트워크는 노드의 수가 상당히 많으므로 전역의 식별자가 필요하지 않으며 정보의 송수신 시에는 노드 중심이 아닌 데이터 중심의 정보를 송수신한다. 다음으로, 기존의 IP 기반 네트워크는 공통된 네트워크 계층을 통해 다양한 종류의 응용을 동시에 이용하는 계층 개념을 사용하지만 센서 네트워크의 센서 노드들은 특정 응용에 따라서 디자인되며 특히 중간 센서 노드들은 데이터를 수집하거나 일시적으로 저장하고 전달하는 기능을 수행한다.

3. 센서 네트워크를 위한 응용 플랫폼

실제 넓은 범위에 배치된 센서 네트워크에서는 다양한 응용을 지원하기 위해 다중의 싱크가 존재할 수 있으며 센서 노드와 싱크간에는 다중의 라우팅 경로를 가질수있다. 이러한 환경에서는 싱크의 이동으로 인해 센서 노드가 싱크의 정보를 빈번하게 갱신해야 하므로 과도한 패킷이 발생할 수 있으며 이는 센서 노드의 전력 소모를 야기 시킨다. 이와 같은 문제점을 해결하기 위해 센서 필드를 격자 방식으로 구분하여 싱크 노드가 다른 모든 센서 노드의 정보를 갱신하지 않고 인접한 센서 노드의 정보만을 갱신하여 패킷 발생을 줄이고 전력소모를 감소시키는 TTDD(Two-Tier Data Dissemination) 방식이 제안되었다[5]. 또한, 다중 싱크 환경은 다중의 라우팅 경로가 존재하므로 정보 전달을 담당하는 중간 센서 노드의 고장으로 인해 다중의 싱크간에는 서로 다른 정보를 가질 수 있다. 이러한 경우에는 센서 네트워크의 다중 싱크로부터 정보를 수집하고 통합하여 사용자가 인식하기 쉬운 GUI 형태로 나타낼 수 있는 응용 플랫폼이 필요하다. 그림 1은 센서 네트워크의 기능 구조를 나타낸 것이다[6].

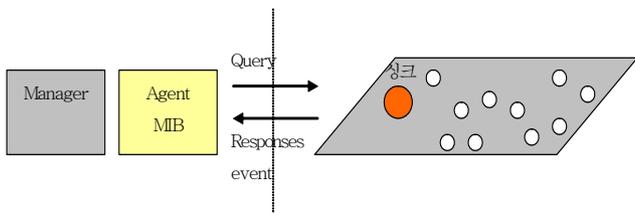


그림 1. 센서 네트워크 기능 구조

센서 네트워크 기능 구조에서 GUI 기반의 매니저는 데이터베이스 정보를 참조하여 센서 필드로 질의하고 에이전트는 싱크로부터 일치하는 이벤트 정보를 수집 및 통합하여 데이터베이스에 저장하는 계층적 관리 네트워크 형태를 가진다. 본 논문에서 제안하는

GUI 기반의 센서 네트워크를 위한 응용 플랫폼은 그림 2 와 같은 구조를 가진다. 그림과 같이 센서 필드 내의 각 센서 노드는 동등한(flat) 구조로 배치되어, 소스 노드에서 발생된 이벤트 정보를 저장하고 있으며 directed diffusion[7] 방식으로 이 정보를 상호 전송하고, 다중 싱크는 전역의 ID 를 사용하여 유무선 인터넷과 연결되어 이벤트 정보를 저장한다.

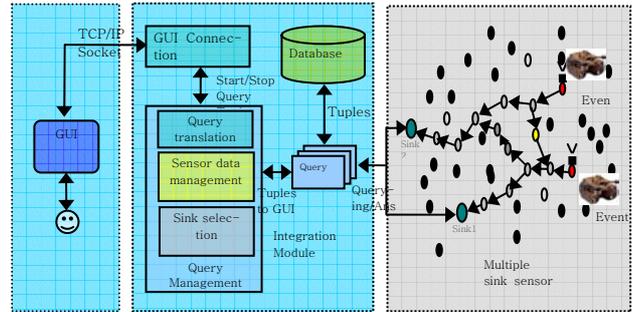


그림 2. 센서 네트워크 응용 플랫폼

센서 네트워크에서의 다중 싱크를 활용함으로써 싱크 노드와 소스 노드간의 다중 경로를 가질 수 있으며 단일 경로가 실패하는 경우에도 전송 지연이나 정보 손실등의 문제점을 개선 할 수 있고, 넓은 범위에 배치된 각 센서 노드들을 효율적으로 관리 할 수 있다. 다중 싱크를 가진 센서 네트워크를 위한 응용 플랫폼은 GUI 응용 모듈, 통합 모듈, 센서 필드로 구분하여 구현한다. 센서 네트워크의 이벤트 정보를 표시하기 위한 GUI 응용 모듈은 소켓을 통해 데이터 중심의 속성 값을 질의하고 통합 모듈에서는 질의와 일치하는 정보를 가지는 싱크 또는 내부 데이터베이스에서 응답을 받은 후 GUI 모듈에 응답한다. GUI 모듈에서는 응답 결과를 테이블 형식으로 나타내고 각 센서 노드의 위치는 지도를 사용하여 표시하며 잔여 전력이 일정한 임계치 이하인 경우에는 다른 색깔로 지도에 표시한다. 통합 모듈은 이벤트 정보를 통합하거나 질의를 해석하기 위한 역할을 하며 GUI 응용 연결 모듈, 질의 해석 모듈, 질의 관리 모듈 그리고 실제 이벤트 정보를 저장하고 있는 데이터베이스를 포함한다. GUI 응용으로부터 질의를 받은 통합 모듈은 질의를 해석하여 센서 필드에 있는 다중 싱크로 질의를 할 것인지 내부 데이터 베이스로 질의를 할 것인지를 결정한다. 통합 모듈에서 다중의 싱크로 질의를 수행함으로써 중간 센서 노드의 고장으로 인해 경로가 실패하는 경우에도 정상적인 경로의 싱크 노드로부터 지속적으로 이벤트를 받을 수 있으므로 전송 지연이나 정보 손실의 문제가 발생하지 않는다. 질의 관리모듈은 질의 내용을 일정기간 저장하고 있으며 중복된 질의인지, 잘못된 형식의 질의인지를 판단하여 불필요한 트래픽 발생을 줄인다. 통합 모듈의 데이터베이스에서는 속성 값 쌍의 정보를 가지고 있으며 다중 싱크로부터 이벤트 정보 종류, 이벤트 발생 시간, 잔여 전력, 위치 정보, ID, 이벤트 값 등의 정보를 수집하여 저장하고 주기적인 새로운 이벤트로 갱신된다. 사용자가

과거의 통계 데이터를 질의하는 경우에는 모든 싱크로 질의하므로 유무선 네트워크에 과도한 트래픽을 발생시킬 수 있으며, 이를 위해서 과거 통계 데이터를 처리하는 경우에는 내부 데이터베이스에서 처리하여 과도한 트래픽 발생을 줄일 수 있다.

4. 구현 및 실험 결과

본 논문에서 GUI 응용 모듈과 통합 모듈의 질의는 소켓을 통해 통합 모듈이나 센서 필드에 전달되며 질의 해석 모듈에서는 GUI 의 질의를 해석한다. GUI 응용 모듈에서는 질의 결과를 테이블 형식으로 표시하고 각 센서 노드의 위치를 지도를 통해 표시하며 특정 시간대 질의를 위한 TimeTo, TimeFrom, ValueTo, ValueFrom 과 실시간 질의를 위한 Duration, Frequency 으로 구성되고 환경 정보를 추출하기 위해 온도, 습도, 강수량의 Type 을 구성하여 실험한다. 그림 3 은 질의 결과를 나타낸 것으로 사용자는 60 초 동안 10 초 간격으로 각 싱크에 온도 정보를 요구하는 interest 를 멀티캐스트로 전송하고 질의를 받은 싱크 노드는 interest 에 일치하는 정보를 가지고 있는 경우에는 10 초 간격으로 응답한다.

Node ID	Type	Value	Location	Time	Power
40	온도	25	200,300	16:00	70
45	온도	27	250,300	13:30	80
70	온도	26	250,400	14:00	70
250	온도	30	550,820	17:00	50
260	온도	25	200,300	15:10	70
263	온도	27	250,300	16:30	80
267	온도	26	250,440	19:00	70
270	온도	30	560,700	16:20	50

그림 3. 실시간 질의 결과 테이블

그림 4는 실시간 데이터 질의 결과를 지도를 통해 나타낸 것으로 질의 결과 테이블의 정보를 바탕으로 각 센서 노드의 위치 정보를 나타내었고 잔여 전력의 정도를 색깔별로 구분하여 표시하였다.

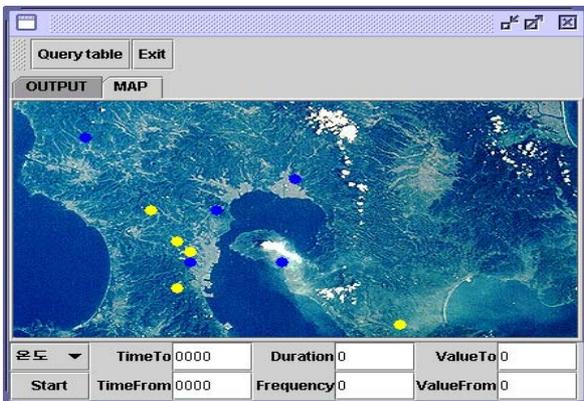


그림 4. 질의 결과 맵

그림 5 는 경로가 실패한 경우와 그렇지 않은 경우의 전송 지연을 비교하기 위해 네트워크 시뮬레이션의 interest 를 설정한 것이다. 싱크에서는 setSubscription()을 수행하여 interest 를 전송하고 interest 를 받은 소스 노드에서는 일치하는 interest 인 경우에는 setupPublication()을 수행하여 정보를 전송하고 정보를 받은 싱크는 경로를 강화하게 된다. interest 를 전송할 setupSubscription()의 속성 값은 Temp 의 Type 을 가지며 최고 온도가 30 도 이상이고 최저 온도가 10 도 이하의 값을 설정하여 브로드캐스트하고 소스 노드에서는 싱크의 setupSubscription()과 같은 속성 값을 저장하고 있는 경우에는 subPublication()의 속성 값인 최저 온도 8 도, 최고 온도 35 도, Temp Type 의 정보를 싱크로 응답한다.

```

handle SenderApp::setupPublication ()
{
    attrs.push_back(NRClassAttr.make(NRAttribute::IS,
        NRAttribute::DATA_CLASS));
    attrs.push_back(LowTemperatureAttr.make(NRAttribute::IS,
        8.00));
    attrs.push_back(HighTemperatureAttr.make(NRAttribute::IS,
        35.00));
    attrs.push_back(TargetAttr.make(NRAttribute::IS, "Temp"));
    handle h = dr_>publish(&attrs);
}
    
```

```

handle ReceiverApp::setupSubscription ()
{
    attrs.push_back(NRClassAttr.make(NRAttribute::
        NRAttribute::INTEREST_CLASS));
    attrs.push_back(HighTemperatureAttr.make(NRAttribute::
        GT,30.00));
    attrs.push_back(HighTemperatureAttr.make(NRAttribute::
        LE,10.00));
    attrs.push_back(TargetAttr.make(NRAttribute::IS,
        "Temp"));
}
    
```

그림 5. interest 설정

네트워크 시뮬레이션에서의 실험을 위해 그림 6 과 같은 토폴로지를 사용하였으며 50 개의 센서 노드가 고정된 상태로 배치되고 센서 노드 49 번과 1 번을 싱크로 설정하여 interest 를 전송하도록 하였으며, 0 번 노드를 소스 노드로 설정하여 싱크 노드의 interest 에 응답하도록 하였다. 15 번 노드와 17 번 노드는 정보를 전달하는 노드로서 경로상의 센서 노드가 전력 소모나 고장으로 인해 경로가 실패하는 경우와 정상적인 경우의 데이터 전송 지연과 데이터 처리율을 비교 측정하기 위해 실험 시작 35 초에 소스 노드 0 번과 싱크 노드 49 번 사이의 15 번 노드를 강제로 이동시켜 경로가 실패하도록 하였다. 토폴로지의 모든 센서 노드는 802.11 MAC 을 사용하여 500x500m² 의 지형을 사용하고 각 센서 노드의 링크 레벨 지연은 25us 로 설정하였으며 정보의 전송은 애드 혹의 directed diffusion 방식을 사용한다. 각 센서 노드의 위치지정은 미리 정해진 시나리오를 통해 임의로 지정하고 100 초동안 실험을 진행하였다. 그림 7 은 경로가 실패한 49 번 싱크에서의 데

이더 처리율과 경로가 정상적인 1 번 싱크에서의 데이터 처리율을 비교한다. 35 초 이후부터는 49 번 싱크와 소스 노드간의 경로가 실패하므로 소스 노드로부터의 데이터 중에서 8~11 번까지의 데이터는 수신할 수 없는 것을 확인 할 수 있으며 12 번 이후의 데이터를 수신하는 경우에도 지연시간은 정상적인 경로를 가지는 1 번 싱크보다 높은 지연 시간을 가지는 것을 확인 할 수 있다.

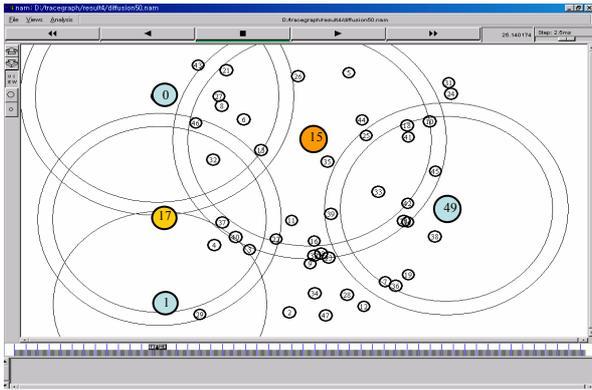


그림 6. 실험에 사용한 토폴로지

초기 센서 필드에서는 경로를 설정하기 위해 directed diffusion 방식으로 interest 를 브로드캐스트하며 싱크가 첫 번째 이벤트를 수신하기 위해서는 6sec 의 지연이 발생하게 되고 경로가 설정된 후에는 경로를 강화하여 최적의 경로로 이벤트를 전송하므로 평균 4.8ms 의 전송지연이 발생한다.

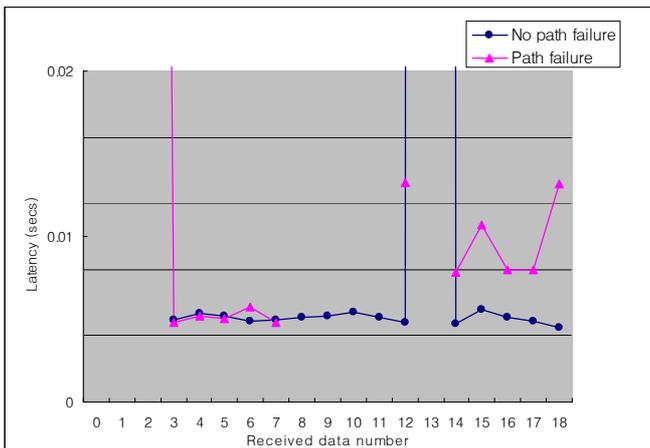


그림 7. 이벤트 전송 지연 비교

이벤트 전송 지연 실험에서는 데이터 전송 중에 센서 노드의 이동이나 센서 노드의 전력 소모로 인해 싱크와 소스 노드간의 경로가 실패하는 경우에는 이벤트 손실이 발생하며 경로를 재 설정하기 위한 전송 지연이 발생하는 것을 확인할 수 있었고 이를 해결하기 위해 다중의 싱크로 부터 이벤트를 통합하는 GUI 기반의 응용 플랫폼을 구현하여 정보에 대한 신뢰성을 높이고 전송 지연을 줄일 수 있다.

5. 결론

본 논문에서는 다중 싱크 노드가 존재하는 넓은 범위의 센서 네트워크 환경에서 하나의 싱크에서만 이벤트를 수집하는 경우에 중간 센서 노드의 고장으로 인해 싱크와 소스 노드간의 경로가 실패하여 경로를 복구하는 동안에는 정보를 수집하지 못하거나 전송 지연이 발생하는 문제점을 논의하였고, 이를 해결하기 위해 여러 싱크를 사용하여 여러 경로로 이벤트를 수집하고 통합하는 응용 플랫폼을 설계하고 구현하였다. 이를 통해서 정보에 대한 신뢰성을 높일 수 있고 전송 지연을 감소시켰으며 응용과 네트워크 시뮬레이션 실험을 통해 결과를 확인하였다. GUI 모듈에서는 응용 플랫폼을 통해 수집하고 통합한 주변의 환경 이벤트 정보를 사용자가 보기 쉽게 테이블 형식으로 나타내었고, 센서 노드의 위치와 소모된 전력을 지도를 통해 나타내었다. 향후, 경로 실패로 인한 정보 손실 및 전송 지연의 문제는 MAC 계층이나 네트워크 계층에서 계속 다루어져야 할 것이다.

참고문헌

- [1] I.F. Akyildiz, W.Su, Y. Sankarasubramaniam, E. Cayirci. Wireless Sensor networks: a survey, IEEE Communications Magazine, August 2002, pp, 102-114
- [2] C. Shen, C. Srisathapornphat, C Jaikaeo, Sensor Information Networking Architecture and Applications, IEEE Personal Communication August 2001
- [3] Kay Romer, Oliver Kasten, Fredemann Mattern. Middleware Challenges for Wireless Sensor Networks. ACM SIGMOBILE Mobile Computing and Communications Review, October 2002
- [4] S.H.Cho and A.P. Chandrakasan, Energy Efficient Protocols for Low Duty Cycle Wireless Microsensor Networks, ICASSP 2001
- [5] S Tilak, N. Abu-Ghazaleh, W Heinzelman, A Taxonomy of Wireless Micro-Sensor Network Models, Mobile Computing and Communications Review, vol.6, no.2, April 2002, pp. 28-36
- [6] Linnyer Beatrys Ruiz, Joes Marcos Nogueira and Antonio A. F. Loureiro. MANNA: A Management Architecture for Wireless Sensor Networks, IEEE Communications Magazine, February 2003
- [7] D. Estrin, R. Govindan, J. Heidemann, S. Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks, ACM MobiCom, August 1999