

센서 미들웨어 기술

국민대학교 | 김영만 · 한재일*

1. 서 론

Ubiquitous Sensor Network(USN)은 사물의 인식정보 및 주변의 환경정보까지 탐지하여 수집된 정보를 네트워크를 통하여 실시간으로 제공하거나 관리할 수 있게 해주는 시스템으로써, USN 인프라 구축을 위한 정부의 지원계획에 따라 u-해양, u-건축, u-병원, u-농촌 등의 다양한 파일럿 프로젝트와 시범운영이 진행되고 있다[1]. 이렇듯 급속히 발전하고 있는 USN 분야에서 실세계 오프라인 정보를 효율적으로 수집하고 제공해주는 USN 미들웨어의 역할은 응용 프로그램과 하위 계층을 연결해주는 전통적인 기능 제공 외에도 다양한 인프라 서비스를 제공하기 위해 통합적이고 인간 친화적이어야 한다.

초기의 USN 미들웨어와 관련된 연구들은 특정 서비스 지원만이 가능한 제한된 기능의 미들웨어를 다루었다. 그러나 최근 USN 응용 서비스 모델들이 상호 통합되고 복잡해짐에 따라 이를 지원할 수 있는 USN 미들웨어가 필요한 실정이며[2], 이를 위해 USN 미들웨어는 다음과 같은 기능을 가져야 한다[3]. 첫째, 임의의 위치에서 발생한 데이터는 임의의 사용자가 원하는 장소까지 실시간으로 이동될 수 있는 글로벌 형태여야 한다. Anytime, Anywhere, Anything 통신이 가능한 인프라스트럭쳐가 형성되어 가는 상황에서 고부가가치 데이터의 실시간 제공은 기본적인 총족 요건이다. 둘째, 기존의 RFID, 텔레매틱스, 홈 네트워크, 센서 네트워크 등을 통합할 수 있어야 한다. 다시 말해서 통합적이고 인간 친화적인 형태의 USN 인프라스트럭쳐가 위와 같은 기반 기술들을 포용하지 못한다면 진정한 USN의 의미를 잃는 것이다.셋째, Local USN을 구성하는 각각의 센서 노드에 위치한 In-network 미들웨어와 Local USN으로부터 수집된 데이터를 관리하는 USN 데이터 서버에 위치한 Host-side 미들웨어를 구분 고안하여야 한다. 즉, USN 미

들웨어는 저용량 CPU, 밧데리 등의 제약 요건들을 가진 센서 노드들로 구성된 Local USN 망을 위한 In-network 미들웨어와 센서 노드와 비교하여 상대적으로 많은 자원을 가진 데이터 서버에 위치하는 Host-side 미들웨어로 구성된 2계층(Two-Tier) 구조를 택해야 한다.

본 논문의 구성은 다음과 같다. 2장은 센서 네트워크 미들웨어의 요구사항 및 모델에 대하여 논하고 3장에서는 센서 네트워크 미들웨어 기술개발의 최신 동향에 대해서 기술한다. 4장은 Host-side 미들웨어와 In-network 미들웨어로 구성된 Two-Tier 방식의 인간 친화형 USN 미들웨어 플랫폼을 소개하고, 5장에서 본 논문의 결론을 맺는다.

2. 센서 네트워크 미들웨어

센서 네트워크는 기존의 전통적인 네트워크와 달리 많은 센서 노드들과 센서 노드들에서 수집된 데이터를 필요로 하는 센서 네트워크 응용들로 이루어져 있다. 센서 네트워크 내에 존재하는 각각의 센서 노드들은 제한된 자원과 실세계와의 강한 결합도로 인해 발생하는 문제점들을 가지고 있고, 이러한 문제점을 해결하는데 있어서 미들웨어는 중요한 역할을 수행한다.

본 장에서는 센서 네트워크 미들웨어가 갖추어야 할 기능 요구사항들을 기술한 뒤 센서 네트워크 미들웨어 모델들에 대하여 설명한다.

2.1 센서 네트워크 미들웨어 기능 요구사항

센서 네트워크 미들웨어를 설계하기 위해서는 기본적으로 네트워크 구성 요소, 센서 노드 배치와 유지 및 관리, 그리고 응용 서비스의 수행을 원활하게 하는데 초점을 맞추어야 한다[4]. 또한 최근 다양화되고 복합적인 센서 네트워크 응용의 등장으로 인해 제한된 자원의 효율적인 이용, 동적인 환경에서의 네트워크 성능 보장, 다수의 노드 구성에 의한 확장성 등

* 종신회원

의 여러 가지 기능이 센서 네트워크 미들웨어에게 요구되고 있다. 본 절에서는 센서 네트워크 응용 계층에서 필요한 센서 네트워크 미들웨어의 기능 요구사항을 도출한다.

• 다양한 질의유형 지원 기능

센서 네트워크 응용에 따라 미들웨어는 다양한 유형의 질의를 지원할 수 있어야 한다. 센서 네트워크 미들웨어가 지원해야 할 질의 유형으로는 실시간적인 데이터 요청 질의, 특별한 조건을 처리하기 위한 질의, 연속적인 센싱 데이터를 요청하기 위한 질의, 그리고 센서 노드의 위치 추적을 위한 질의 등이 있다.

• 메타 정보 관리 기능

센서 네트워크 응용 계층이 제대로 동작하기 위해서는 자신이 이용하고자 하는 센서 네트워크에 관한 메타정보를 정확히 알고 있어야 한다. 이러한 메타정보로는 센서 네트워크 이름, 센싱되는 정보의 종류, 구동기의 종류, 노드의 위치 정보 등의 정적인 메타정보와 센서 노드의 수, 센서 노드의 잔여 전력량, 센서 네트워크의 통신 상태, 센서 노드의 오류 유무 상태 등의 동적인 메타 정보가 있을 수 있다. 센서 네트워크 미들웨어는 센서 네트워크에 대한 주기적인 모니터링 작업을 통하여 위와 같은 메타정보들을 응용 계층에게 실시간으로 지원해야 한다.

• 제한된 자원의 관리 기능

센서 노드의 에너지, 컴퓨팅 파워, 메모리 그리고 통신 대역폭과 같은 자원은 제한되어 있다. 센서 네트워크 미들웨어는 응용 계층의 질의에 따라 동적으로 작동하고 최소의 자원을 소모하도록 하는 기능과 동시에 여러 응용과 동작할 수 있는 전체적인 시스템 동작을 위한 최적의 자원 관리 기능을 가지고 있어야 한다.

• 센서 노드 간 이질성 추상화 지원 기능

센서 네트워크상의 각각의 센서 노드들을 구성하는 하드웨어는 센서 노드별로 다를 수가 있고, 이러한 센서 노드 간의 이질성으로 인해 센서 네트워크는 여러 가지 제약, 또는 다양한 문제 상황에 봉착할 수 있다. 그러므로 센서 네트워크 미들웨어는 각 노드 운영체제의 추상화를 통해 다양한 형태의 하드웨어와 네트워크의 인터페이스들을 포함하는 시스템 메커니즘을 제공해야 한다.

• 동적인 네트워크 토폴로지 구성 지원 기능

센서 네트워크 미들웨어는 각각의 센서 노드가 이동함에 따라 생기는 변화에 대해서도 기능 불량, 장

치 고장 등의 오류가 없이 동적인 네트워크 토폴로지를 구성함으로써 네트워크의 완벽한 성능을 보장할 수 있는 기능을 지원해야 한다.

• 확장성 지원 기능

각각의 센서 노드들이 점점 작아지고 가격이 저렴해짐에 따라 센서 네트워크는 수백에서 수천 개의 센서 노드들로 이루어질 수 있게 되었다. 이처럼 확장성을 지닌 센서 네트워크상의 각각의 센서 노드에 대한 설정, 유지보수 및 업그레이드는 매우 큰 문제점이다. 센서 네트워크 미들웨어는 이같은 문제점을 해결하기 위해 센서 노드 스스로 설정변경 및 유지보수를 할 수 있는 기능을 지원해야 하며 잠재적인 네트워크 확장성에 대해 유연하게 받아들일 수 있는 기능을 지니고 있어야 한다.

• 실세계와의 통합 지원 기능

각각의 센서 노드들은 실세계의 특정 현상을 감지하는데 그 목적이 있다. 만약 센서 노드들이 공통된 시간과 공간의 척도를 사용하지 않는다면 특정 현상을 감지하더라도 수집된 데이터는 정확한 정보로 활용될 수 없다[5]. 그러므로 센서 네트워크 미들웨어는 센서 노드들 간의 공통적인 시간과 공간의 척도를 설정할 수 있는 기능을 가지도록 설계되어야 하며, 다양한 센서 네트워크 응용에게 실시간 데이터 전달기능을 제공해야 한다.

• 센서 노드의 위치 인식 기능

센서 네트워크는 센서 노드에서 센싱된 데이터의 전달뿐만 아니라 센서 네트워크 응용들이 요구하는 데이터를 센싱하는 센서 노드들의 위치정보와 전체 네트워크 토폴로지를 파악할 필요가 있다. 이를 위해 센서 네트워크 미들웨어는 센서 노드들에 대한 실시간 위치 정보 인식 기능을 제공해야 한다.

• 상황 인식 기능

최근 센서 네트워크 응용들은 센싱 데이터를 수집하여 단순히 제공하는 수준이 아니라, 이들 데이터들을 수집 및 분석하여 하나의 의미 있는 상황 정보를 생성할 수 있는 기능을 필요로 하고 있다. 그러므로 센서 네트워크 미들웨어는 상황 정보를 생성하기 위하여 센싱 데이터를 통합 분석하기 위한 기능과 규칙을 정의하여 새로운 상황 정보를 생성할 수 있는 기능을 제공해야 한다.

• 자동화된 센서 노드 모듈 갱신 기능

이동성이 있거나 광범위한 지역에 배치되어 있는 센서 노드들이 실행하는 소프트웨어 모듈을 사람이 일

일이 간신하는 것은 거의 불가능하다. 센서 네트워크 미들웨어는 이러한 센서 노드들의 모듈을 무선 네트워크를 통해 자동적으로 간신할 수 있는 방법을 제공해야 한다.

• 센싱 데이터 관리 기능

센서 노드는 센싱된 데이터를 연속적으로 센서 네트워크 응용에게 전달한다. 하지만 센서 노드의 제한된 자원 때문에 연속적인 데이터 전달은 과도한 자원 소모 문제를 발생시키게 된다. 센서 네트워크 미들웨어는 이러한 연속적인 데이터 전송을 줄이기 위해 센싱 데이터를 최적의 위치에 저장하고, 필요시 복사본을 데이터 접근이 용이한 곳에 저장하는 기능을 제공해야 한다.

• 다양한 응용 지원 기능

센서 네트워크 미들웨어는 특정 응용에 종속적인 기능을 제공할 수 있어야 하며, 동시에 다양한 응용들을 지원할 수 있는 기능을 제공해야 한다(Trade-off).

• 보안 지원 기능

센서 네트워크의 특징 중 하나는 광범위한 지역의 여러 도메인들에 관련된 침입, 화재등과 같은 민감한 정보를 수집하는 것이다. 하지만 이러한 정보를 센싱하는 각각의 노드가 열악한 환경에 배치될 수 있고, 이로 인해 센서 네트워크는 악의적인 침입이나 거부 공격(denial of service)같은 외부 공격에 노출될 위험이 크다. 이러한 문제점을 해결하기 위해 센서 네트워크 미들웨어는 센서 네트워크를 위한 포괄적이고 안전한 보안 메커니즘을 제공해야 한다.

• 서비스 품질 지원 기능

민감한 정보를 센싱하는 센서 네트워크는 수집되는 센싱 정보에 대하여 높은 신뢰도를 요구한다. 하지만 기존의 전통적인 유선 네트워크에서 사용되던 서비스 품질 메커니즘을 센서 네트워크에 적용하는 것은 제한된 자원과 동적인 토폴로지 구성으로 인해 부적절하다. 센서 네트워크 미들웨어는 데이터 처리량 및 전달 지연, 에너지 소모와 같은 특징들을 고려하여 높은 수준의 서비스 품질을 보장할 수 있는 기능을 제공해야 한다.

• 기존 네트워크와의 연동 기능

센서 네트워크는 이용 가능한 자원이 제한되어 있는 특성으로 인해 자원 집약적으로 요구되는 기능이나 대규모 정보의 저장을 인터넷과 같은 외부 네트워크에서 수행하도록 하는 것이 적합하다. 따라서 센서 네

트워크 미들웨어는 IP를 사용하지 않는 센서 네트워크와 인터넷 기반의 네트워크를 연결하기 위해서 게이트웨이 구조를 통한 외부 네트워크와의 연동 기능을 제공해야 한다[6].

2.2 센서 네트워크 미들웨어 모델

센서 네트워크 미들웨어는 제한된 자원을 가진 센서 노드와 다양한 센서 네트워크 응용 사이에 위치하며, 실제 환경에서 센서 노드가 센싱한 데이터를 응용에게 전달함에 있어서 많은 기능 요구사항들을 만족하여야 한다. 이러한 센서 네트워크 미들웨어 모델은 크게 소프트웨어 구조적인 분류와 미들웨어의 기능적인 분류로 나눌 수 있으며, 상세 내용은 다음 각 항에서 설명한다.

2.2.1 소프트웨어 구조적 분류

센서 네트워크 미들웨어 모델은 소프트웨어 구조적인 관점에서 Local USN을 구성하는 각각의 센서 노드에 위치한 In-network 미들웨어와 Local USN으로부터 수집된 데이터를 관리하는 USN 데이터 서버에 위치한 Host-side 미들웨어의 Two-Tier 구조로 구성된다. In-network 미들웨어의 예로 Agilla, Sensor Ware, MagnetOS, MATE, Impala, Mires, PADS 그리고 NEST 등이 있다. Host-side 미들웨어로는 Cougar, SINA, MiLAN 그리고 Sentire 등이 있다. 또한 센서 네트워크 미들웨어 중에서 TinyDB, DSWare 그리고 ETRI USN 미들웨어 플랫폼은 In-network와 Host-side 미들웨어를 모두 포함하고 있다(그림 1 참조).

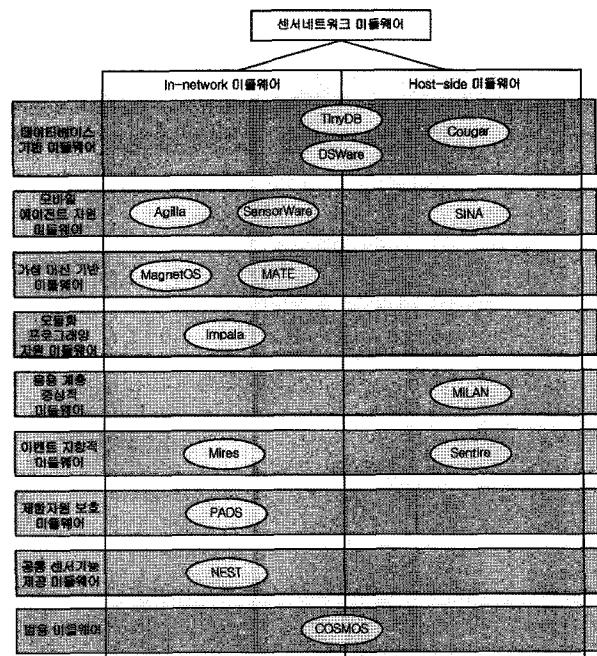


그림 1 센서네트워크 미들웨어 소프트웨어 구조적 분류

2.2.2 기능적 분류

- 데이터베이스 기반 미들웨어

데이터베이스 기반 미들웨어는 센서 네트워크를 하나의 분산된 데이터베이스로 인식하고 SQL(Structured Query Language)과 비슷한 질의어를 제공하여 센서 네트워크 응용의 사용자가 센서 네트워크 내의 센싱 데이터를 얻는데 있어서 편의성을 제공하는 미들웨어이다.

- 모바일 에이전트 지원 미들웨어

모바일 에이전트(Mobile Agent) 기반 미들웨어는 센서 노드의 실행 모듈을 갱신해야 할 때, 각 센서 노드에서 배포하고자 하는 모듈을 동적으로 다운로드하여 갱신할 수 있는 기능을 제공하는 미들웨어이다.

- 가상 머신 기반 미들웨어

가상 머신(Virtual Machine) 기반 미들웨어는 가상 머신과 인터프리터(Interpreter), 그리고 모바일 에이전트로 구성되어 있다[7]. 센서 네트워크 응용에서 필요로 하는 센서 노드의 모듈을 작성할 때 가상 머신을 통해 추상화된 코드로 모듈을 작성 할 수 있다. 이렇게 작성된 모듈은 모바일 에이전트(Mobile Agent)를 통해 네트워크 내의 각 센서 노드들에게 전달되고, 모듈을 수신한 센서 노드는 인터프리터를 통하여 모듈을 해석하고 갱신하게 된다. 프로그래머 입장에서는 최소한의 코드로 모듈을 작성할 수 있지만 각 명령어를 이해해야 한다는 단점이 있다.

- 모듈화 프로그래밍 지원 미들웨어

모듈화 프로그래밍 지원 미들웨어의 핵심은 센서 노드에 배포될 실행 모듈에 대해 최대한 모듈화 시키고 이동성 있는 코드로 작성하여 모바일 에이전트를 통해 센서 네트워크의 센서 노드들에게 배포하는 것이다. 실행 모듈을 최대한 모듈화 시켜서 작은 모듈로 만드는 이유는 모듈 전체를 센서 노드들에게 분산시킬 때 보다 적은 자원을 소비하기 때문이다. 또한 이러한 메커니즘은 센서 네트워크의 장애 관리와 동적인 토플로지 구성을 지원한다.

- 응용 중심적 미들웨어

응용 중심적인 미들웨어는 센서 네트워크 응용에서 요구하는 서비스 품질(QoS)과 센서 네트워크의 전체 자원을 비교하여 최적의 실행 조건을 도출한다. 이러한 결과로 응용 중심적인 미들웨어는 센서 네트워크의 자원의 소모를 최소화하고 서비스 품질을 최대로 만족할 수 있도록 센서 네트워크를 동적으로 구성한다.

- 이벤트 지향적 미들웨어

이벤트 지향적인 미들웨어는 비동기적 센서 이벤트 발생 및 전달에 초점을 맞추어, 센서 이벤트를 발생시키고 이벤트 통보를 원하는 사용자에게 전달하는데 필요한 상위 기능들을 제공한다.

- 제한자원 보호 지원 미들웨어

제한자원 보호 지원 미들웨어는 센서 네트워크의 문제점인 제한된 자원에 대해, 센서 네트워크의 각 모듈이 보다 효율적으로 자원을 소모하는 방법을 제공한다.

- 공통 센서기능 제공 미들웨어

공통 센서기능 제공 미들웨어는 센서 네트워크가 제공하여야 할 공통 서비스들을 모아서 제공한다. 센서 네트워크가 제공할 수 있는 공통 서비스로는 시간 동기화 서비스, 노드 위치 인식 서비스, 타임 스탬핑 등이 있다.

- 범용 미들웨어

센서 네트워크 범용 미들웨어는 공통 센서기능 제공 미들웨어보다 광의의 개념으로 센서 네트워크가 제공할 수 있는 공통 서비스 외에 수많은 기능들을 제공한다. 예를 들어, 다양한 응용에 대한 센싱 데이터 관리, 메타 정보 관리, 센서 네트워크 추상화, 서비스 품질(QoS), 보안 그리고 상황인식 등의 기능이 있다.

3. 센서 네트워크 미들웨어 기술개발 동향

본 장에서는 2장에서 다른 미들웨어 분류 기준 중 기능적인 분류에 따라 현재 연구 중에 있는 대표적인 미들웨어들을 분류하고 각각에 대하여 상세 설명 한다.

3.1 데이터베이스 기반 미들웨어

3.1.1 Cougar

Cougar[8]는 Cornell 대학의 데이터베이스 연구팀에 의해 연구되고 있는 센서 네트워크용 분산 데이터 처리 시스템이다. Cougar에서 센서 노드들의 데이터 접근과 처리는 모두 중앙 집중적인 방식이 아닌, 분산된 형태로 수행된다. Cougar는 서술적인 질의를 사용하고 네트워크 변화에 동적으로 적응하며, 유연성과 확장성과 장애 처리를 가진 시스템을 목표로 연구를 진행하고 있다. Cougar의 적응형 질의 처리(Adaptive query processing) 기술은 센서 네트워크에서의 데이터 처리에 상당히 유용하나 지역적인 정보에 의존해야 하는 문제가 발생하기도 한다.

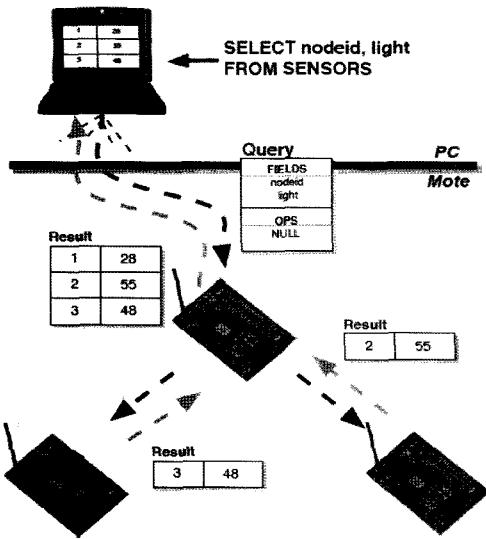


그림 2 TinyDB의 질의 처리 모델

3.1.2 TinyDB

TinyDB[9]는 Berkeley 대학에서 연구한 미들웨어로 센서 네트워크 환경에서의 데이터 기반의 응용 프로그램의 작성을 손쉽게 하는 데에 목적이 있다. TinyDB는 센서 네트워크를 가상 데이터베이스로 간주하며 SQL-like 질의언어와 Semantic Routing Tree(SRT)를 지원한다. 그림 2는 TinyDB의 질의 처리 모델의 예를 나타낸다.

또한 TinyDB는 쿼리를 통해 센서 네트워크의 데이터를 추출할 수 있는 PC응용 프로그램 작성용의 간단한 자바 API를 제공하며 GUI 형태의 질의 생성기와 결과 표시 프로그램 또한 이 API를 사용한다. TinyDB를 사용하기 위해서는 센서 네트워크의 각 센서 노드에 TinyOS 기반의 TinyDB 컴포넌트를 설치해야 한다. 즉, TinyOS가 설치된 노드에서만 이용이 가능하며, SQL-like 질의 언어를 지원하나 신규 기능 추가 시 모든 노드가 보유하고 있는 질의어 처리기 모듈을 수정해야 한다.

3.1.3 DSWare

DSWare(Data Service Middleware)[10]는 Virginia 대학에서 연구하였으며, 센서 네트워크를 위한 실시간 데이터 서비스를 유기적으로 통합하여 제공하는 데이터 서비스 미들웨어이다. 센서 네트워크의 응용 계층에서 필요한 이벤트 신청, 이벤트 탐지, 데이터 저장, 노드와 노드 그룹 관리 등의 여러 서비스들을 추상화하여 데이터 서비스로 제공하기 때문에 응용 계층은 센서 네트워크의 저수준의 복잡한 오퍼레이션과 상관없이 표준 SQL을 사용하여 센서 네트워크로부터 원하는 데이터를 얻을 수 있다. 그림 3은 DSWare의 모델 구조를 나타낸다.

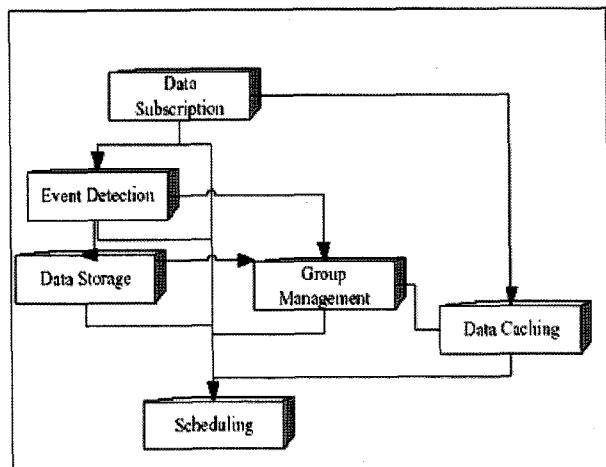


그림 3 DSWare 모델 구조

DSWare 모델의 각 구성 요소는 다음과 같다. Data Subscription은 응용 계층에서 센서 노드의 특정 이벤트를 감지할 필요가 있을 때 사용되고, Event Detection은 Subscription을 통해 등록된 이벤트가 발생하는 것을 감지하고 실시간으로 보고하는 기능을 한다. Data Storage는 데이터를 통합하고 프로세싱하며 데이터 전송 시, 손실될 가능성을 고려하여 여러 물리적인 노드에 데이터를 중복하여 저장한다. Data Caching은 가장 많이 요청되는 데이터에 대해 다수의 복사본을 만들어 노드들에게 제공하는 역할을 하며 복사된 데이터가 사용되는 것을 모니터링하여 조절한다. Group Management는 데이터를 통합하기 위해 센서 노드들을 결합하는 역할을 하며, Scheduling은 실시간 스케줄링과 에너지 인지 스케줄링의 기능을 가지고 있다.

3.2 모바일 에이전트 지원 미들웨어

3.2.1 Agilla

Agilla[11]는 Washington 대학에서 발표하였으며, 무선 센서 네트워크를 위하여 설계된 운영체제인 TinyOS에서 동작하는 모바일 에이전트 기반의 미들웨어이다. 모바일 에이전트로 구성된 Agilla는 각각의 센서 노드가 가지고 있는 상태 정보를 인접한 노드들에게 전송함으로써 무선 네트워크 환경을 구축한다. 그림 4는 Agilla의 모델을 나타낸다.

TinyOS 위에서 동작하는 Agilla 환경에서 각각의 노드는 모바일 에이전트를 가지고 있다. 모바일 에이전트는 하나의 노드에서 다른 노드로 이동할 수 있으며 한 개의 노드에서 최대 4개의 에이전트를 가질 수 있다. 한 센서 노드 내의 에이전트들은 동시에 동작 할 수 있어서 여러 응용 계층의 요청을 처리할 수 있다. 하지만 에이전트 활동에 대한 인증이나 감시 정책

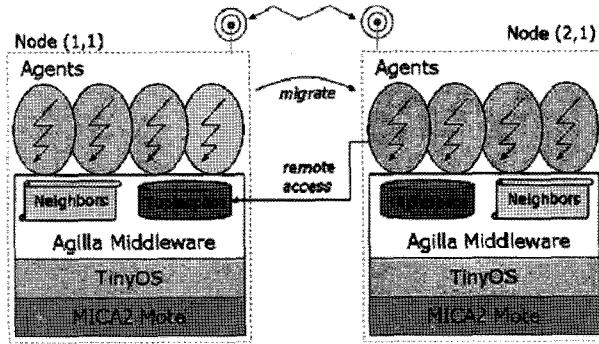


그림 4 Agilla 모델

이 없고 Assembly와 비슷한 프로그래밍 모델은 Agilla 모듈의 유지보수를 하는데 어려움이 있다.

3.2.2 SensorWare

SensorWare[12]는 UCLA 대학에서 발표한 미들웨어로써 플랫폼에 독립적인 스크립트 코드를 수행 할 수 있는 런타임 환경을 기반으로 하는 프레임워크이다. 각각의 스크립트들은 노드에서 하나의 태스크를 수행하며 자신들의 코드를 다른 노드에 복사하거나 이동시킴으로써 동적인 네트워크 토플로지를 구성할 수 있다. 스크립트를 작성하는 스크립트 언어는 Tcl을 확장한 언어로써, 개발자로부터 자원관리 같은 하드웨어와 관련된 부분을 숨겨주고, 여러 응용 계층 사이에서 센서 노드끼리의 자원을 공유하는 방식을 제공해 준다. 하지만, SensorWare는 메모리가 풍부한 환경에서 동작하는 멀티태스크나 고급 스크립트 언어를 통해 운용되므로 제한된 메모리를 가진 센서 노드에는 적합하지 않다. 그림 5는 SensorWare가 적용된 센서 노드의 구조를 보여준다.

3.3 가상 머신 기반 미들웨어

3.3.1 MagnetOS

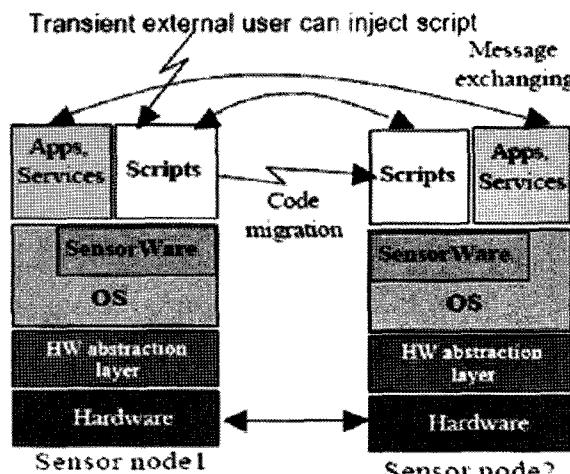


그림 5 SensorWare가 적용된 센서 노드

Cornell 대학에서 고안한 MagnetOS[13]는 분산된 무선 센서 네트워크 내에 하나의 Java 가상 머신(Virtual Machine)역할의 시스템 이미지를 제공함으로써 센서 네트워크간의 이질성을 해결한다. MagnetOS는 각각의 센서 노드에 대한 실행 모듈 모니터링, 객체 생성, 이동성 있는 모듈의 배포를 위한 서비스를 제공한다.

3.3.2 MATE

Mate[14]는 Berkeley 대학에서 연구하였으며, 센서 네트워크를 위해 개발한 가상 머신 기반의 미들웨어이다. TinyOS가 설치된 센서 노드에서 동작하며, 독자적인 바이트코드 인터프리터를 구비하고 있고 전염 모델(Infestation Model)을 통해 새로운 코드를 배포하는 메커니즘을 가지고 있다. Mate의 고수준의 인터페이스는 복잡한 프로그램을 굉장히 간결하게 만들어 주고, 센서 노드에게 새로운 실행 모듈을 전송하는데 필요한 자원 사용을 최소한으로 줄여준다. 또한 가상 머신 언어로 짜여진 24개의 단위 명령어로 구성된 ‘캡슐’이라 불리는 프로그램이 센서 노드에 탑재되어 하나의 태스크를 수행하게 되며, 네트워크를 통해 다른 센서 노드에게 전송될 수 있다. 그림 6은 Mate의 시스템 구조를 나타낸다.

3.4 모듈화 프로그래밍 지원 미들웨어

3.4.1 Impala

Impala[15]는 Princeton 대학에서 ZebraNet 프로젝트의 일환으로 시작되었다. ZebraNet 프로젝트는 센서 네트워크 기술을 이용하여 열룩말과 같은 동물들의 이동과 번식을 연구하기 위한 프로젝트다. 이러한 생태 관찰 연구의 특성상 ZebraNet에 적용되는 센서 네트워크는 에너지 소비를 최대한 효율적으로 고려해 수명을 최대화시켜야 하고, 사람이 관리하기 어려

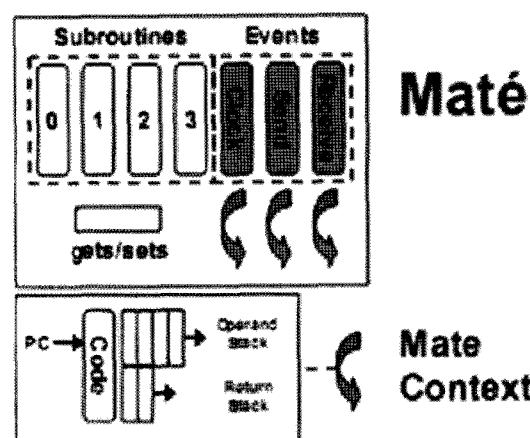


그림 6 Mate의 시스템 구조

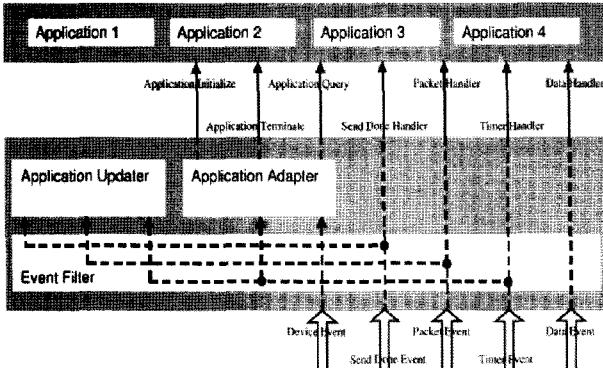


그림 7 Impala의 계층적 시스템 구조

운 환경에 배치되므로 소프트웨어의 자동 업데이트 등을 할 수 있는 기능이 지원되어야 한다. Impala는 애플리케이션의 모듈화(Modularity), 적응력(Adaptivity), 복구성(Repairability)에 초점을 맞췄다. Impala에서 개선되는 소프트웨어는 무선 네트워크를 통해 각 센서 노드에게 전달되고, 각 노드는 시스템이 동작하는 상태에서 개선을 수행할 수 있다. 또한 응용 계층에게 네트워크 관리 인터페이스를 제공함으로써 소프트웨어 시스템의 성능, 에너지 효율성, 안정성을 향상시키고, 다양한 파라미터와 디바이스 실패 등에 대한 동적 응용 적응성을 제공한다. 그림 7은 Impala의 계층적 시스템 구조를 나타낸다.

Impala에서 응용 프로토콜과 프로그램들은 상위 계층을 구성하고 있으며 하위 계층에 있는 응용 어댑터, 응용 업데이터 등의 미들웨어 애이전트가 상위계층을 지원한다. 응용 어댑터는 센서 노드들의 내부 상태에 따라 적합한 응용 프로토콜을 선택한다. 응용 업데이터는 메모리, 에너지 등의 자원에 제약을 고려한 효율적인 소프트웨어 개선 메커니즘을 제공한다. 그러나 Hewlett Packard 제품에 의존적인 미들웨어로서, 센서노드 하드웨어의 추상화가 지원되지 않는다는 단점을 가지고 있다.

3.5 응용 계층 중심적인 미들웨어

3.5.1 MiLAN

MiLAN(Middleware Linking Application and Networks) [16]은 Rochester 대학에서 개발한 스마트 메디컬 홈을 위한 센서 네트워크 미들웨어이다. MiLAN은 응용 계층에서 데이터 송신 센서노드, 라우팅 기법 그리고 서비스 품질(QoS) 등을 사전에 기술하면 응용 계층에서 요구한 서비스 품질을 현재 센서 네트워크의 자원과 비교하여 절충한 응답을 응용 계층에게 전송한다. 이는 센서 네트워크의 수명을 최대한 보장하고 응용 계층의 서비스 품질을 만족시키기 위함이다.

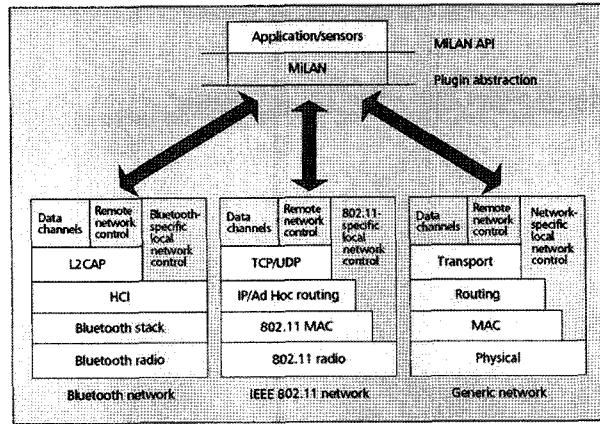


그림 8 MiLAN의 네트워크 구조

MiLAN은 응용 계층에서 필요한 기능적 요구사항을 나타내기 위해 사용하는 API와 MiLAN의 명령을 특정 네트워크 프로토콜의 명령으로 변경해서 동적인 네트워크를 구성하는 추상화 계층으로 구성된다(그림 8 참조).

3.6 이벤트 지향적인 미들웨어

3.6.1 Mires

Mires[17]는 FUP(Federal University of Pernambuco)에서 발표한 미들웨어로써 Publish/Subscribe 방식을 지원한다. 예를 들어, 특정 응용 계층에서 Subscribe를 통해 센서 노드가 배치된 곳의 환경 여부를 판정하기 위한 온도 조건을 설정해 두고 센서 노드는 자신이 센싱한 데이터가 미리 지정된 온도조건을 만족할 때 Event를 발생시키고 Publish를 통해 해당 응용 계층으로 이벤트를 전달하여 환경이 일어났음을 알리게 된다. 이러한 Publish/Subscribe 방식은 비동기적인 특성과 멀티포인트 커뮤니케이션 속성을 가지고 있다. 그림 9는 Mires의 구조를 보여준다.

Mires의 가장 중요한 컴포넌트는 Publish/Subscribe 서비스로 센서 노드와 응용 계층의 비동기적인 데이터 전송을 담당하며, 다양한 응용과의 멀티포인트 커뮤니케이션을 담당한다.

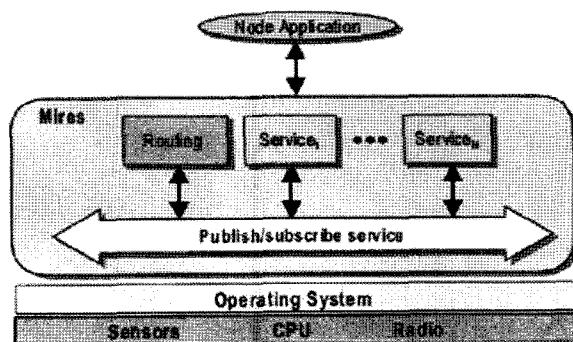


그림 9 Mires의 구조

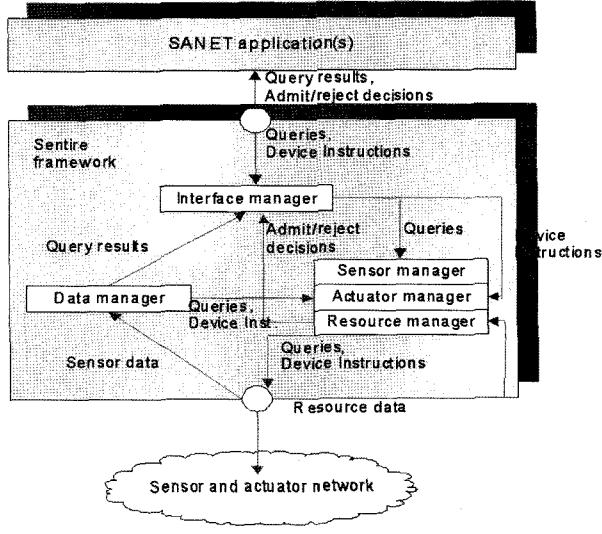


그림 10 Sentire의 데이터 흐름

3.6.2 Sentire

Sentire[18]는 Rensselaer Polytechnic Institute와 IBM에서 고안하였으며, 센서와 Actuator로 구성된 SANET(sensor and actuator networks)을 위한 미들웨어이다. Sentire는 응용의 질의를 통해 SANET에서 수집한 데이터를 정해진 수준의 품질로 변환하여 응용 계층에게 전달해 주기 위해 크게 두 가지 특징을 가진다. 첫 번째는 인터페이스, 데이터, 자원 그리고 센서와 Actuator를 관리하는 것이고, 두 번째로 Priority, Timestamp, Sending Manager에 따라서 메시지를 효과적으로 처리할 수 있게 구성된 Sentire Message를 사용한다. 그림 10은 Sentire의 데이터 흐름을 나타낸다.

Resource manager는 센서와 Actuator의 자원을 체크하고, Interface manager는 센서를 가동시킨다. SANET에서 발생한 센싱 데이터는 Data manager로 전송되어 패턴 분석 및 정제가 이루어진다. 그리고 가공된 데이터를 Interface manager에게 전송한 후 마지막으로 응용계층에게 결과를 보고한다.

3.7 제한자원 보호 지원 미들웨어

3.7.1 PADS

PADS(Power Aware Distributed Systems)[19]는 UCLA/USC 대학에서 연구한 센서 네트워크 미들웨어로써 에너지 절감을 위하여 CPU, RF모듈, 센서 모듈간의 에너지 관리 기능들을 유기적으로 관리하는 기능을 최적화하는데 목적이 있다. 이를 위하여 RTOS(Real-Time Operating System) 스케줄링 기법을 사용하여 정확한 시간에 동적으로 CPU 전압과 RF 모듈레이션 스케일 등을 조절하여 최소한의 에너지로 프로그램 실행, 센서 측정 및 메시지 송수신이 이루어지도록 한다. 그림 11은 PADS 미들웨어의 구조를 보여준다.

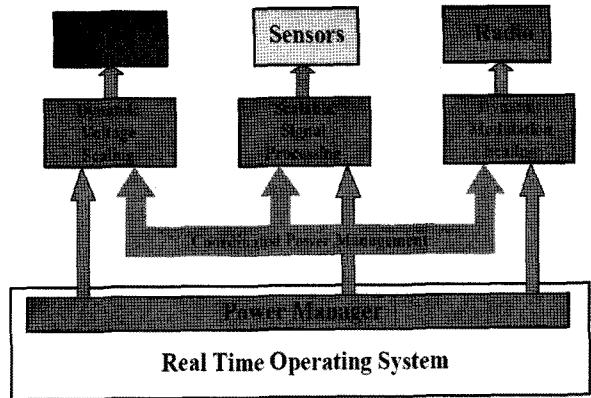


그림 11 PADS의 구조

3.8 공통 센서기능 제공 미들웨어

3.8.1 NEST

DARPA NEST 프로젝트는 센서 네트워크에 적합한 미들웨어에 대하여 연구하는 과제로서, 센서 네트워크가 가져야 할 공통 기능들을 묶어서 미들웨어를 구성하였다. 예를 들면, Vanderbilt 대학의 미들웨어는 몇몇의 독립 기능들을 하나의 미들웨어 안에 묶어서 구현하고 있다. 대표적인 기능 중 하나로 위치인식 시스템인 RIPS(Radio Interferometric Positioning System)[20]는 그림 12와 같이 동작한다.

두 센서 노드가 동시에 각각 다른 두 개의 센서 노드로 무선 메시지를 전송하게 되면 충돌이 생기게 되고 이러한 라디오 주파수 충돌은 센서 노드가 자신의 위치를 결정하는데 사용된다. 그림 12에서 볼 수 있듯이 A와 B가 동시에 각기 다른 신호를 보내는데 이것은 C와 D노드에서 각각 조합된 신호로 나타내어진다. C와 D는 조합된 두 개의 신호를 비교하여 그 차이로 노드간 거리를 계산한다. RIPS는 최악의 경우 10 센티미터 이하의 오차를 가진다.

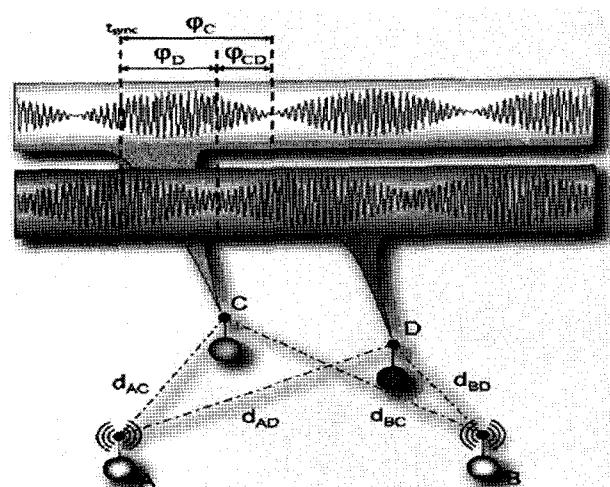


그림 12 RIPS의 원리

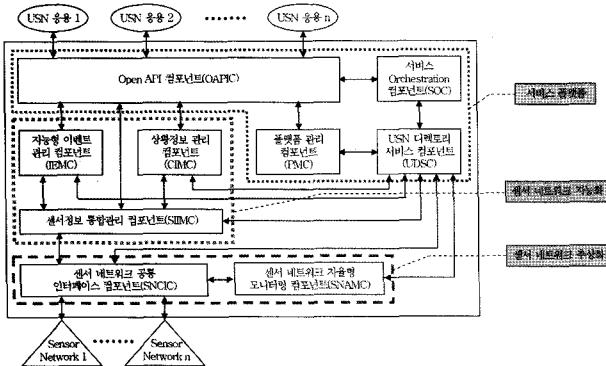


그림 13 ETRI COSMOS 구조

또 다른 기능으로는 시간동기화 프로토콜인 FTSP (Flooding Time Synchronization Protocol)[21]를 들 수 있다. FTSP는 센서 노드들 중에 하나의 리더를 선출한 후, 브로드캐스트하여 시간 동기화를 이루고 있으며 MICA2 플랫폼 상에서 백만분의 2초 미만의 시간 오차를 달성하고 있다. 이 밖에도 스택 모니터, 타임 스탬핑, 시스템 분석 메시징 서비스 등의 기능이 미들웨어에 포함되어 있다.

3.9 범용 미들웨어

3.9.1 ETRI USN 미들웨어 플랫폼

ETRI에서 연구하고 있는 USN 미들웨어 플랫폼인 COSMOS[22]는 센서 네트워크 미들웨어를 위한 기능 요구사항을 대부분 수용하는 구조를 가지고 있으며, In-network 미들웨어와 Server-side 미들웨어(본 논문의 Host-side와 동일한 의미임)를 포함하며, 그림 13과 같은 구조를 가진다.

ETRI의 USN 미들웨어 플랫폼은 센서 네트워크 추상화, 센서 네트워크 지능화, 서비스 플랫폼의 세 단계로 나누어 연구를 진행하고 있다. 첫째, 센서 네트워크 추상화는 센서 네트워크 인프라와 직접 연계되는 부분으로, 다수의 이기종 센서 네트워크들을 용이하게 통합할 수 있게 하고 센서 네트워크의 상태를 실시간 모니터링 할 수 있는 기능을 한다. 둘째, 센

서 네트워크 지능화는 센서 네트워크 공통 인터페이스로부터 연속적으로 입력되는 센싱정보에 대하여 의미를 부여하는 단계로 센싱정보를 효율적으로 관리하고 이를 센싱정보와 기구축된 비즈니스 정보를 분석하여 새로운 상황정보를 생성하는 역할을 수행한다. 마지막으로 서비스 플랫폼은 USN 응용 서비스의 개발을 효율적으로 지원하기 위하여 openAPI를 제공하고 다수의 서비스 사용자들에 대한 관리 기능 등을 제공한다.

4. USN 미들웨어 플랫폼

USN 미들웨어는 USN 응용에서 USN 계층의 자원과 데이터를 쉽게 활용할 수 있도록 다양한 기본 서비스를 제공하는 시스템 소프트웨어로 정의할 수 있다. 최근의 USN 응용 서비스는 여러 USN 서비스 모델이 융합되는 추세로써 이와 같은 USN 응용 서비스의 용이한 개발을 지원할 수 있는 범용 USN 미들웨어가 요구된다. 본 장에서는 USN 미들웨어 플랫폼의 기본 개념과 시스템 구조에 대하여 설명하고, 기본적인 서비스 시나리오를 살펴본 후, Two-Tier 형식의 계층적 USN 미들웨어의 소프트웨어 구조에 대하여 소개한다.

4.1 USN 미들웨어 플랫폼 기반의 시스템 구조

USN 기반의 광역서비스를 클라이언트에게 제공하는데 적합한 USN 미들웨어 플랫폼의 시스템 구조는 그림 14에서 보는 것처럼 Data 서버와 USN Sink 노드의 연결 방법에 따라 Tightly-coupled, Loosely-coupled, Separately-coupled USN의 세 가지 형태로 구현될 수 있다. Tightly-coupled USN의 경우 Data 서버와 USN Sink 노드와의 연결이 Serial 또는 USB를 통한 직접적인 일대일 연결로 이루어지며 외부로부터의 접근은 Data 서버를 필히 경유하여야 한다. Loosely-coupled USN은 외부 통신을 Data 서버가 전달하며 LAN 또는 WLAN과 같은 내부 지역망을 통해 Local USN에서 발생하는 데이터를 USN Sink 노드가 수집하여 Data 서

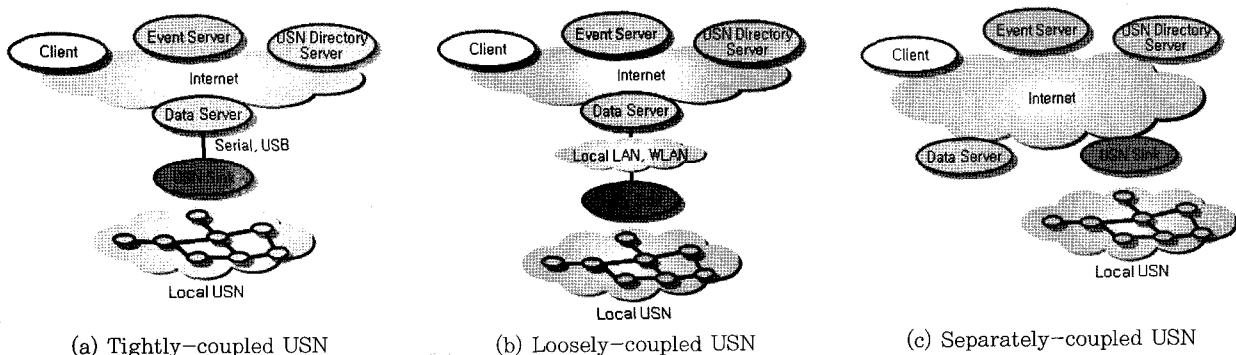


그림 14 USN 미들웨어 플랫폼의 시스템 구조

버로 중계하는 형식이다. 마지막으로 Separately-coupled USN은 USN Sink 노드가 인터넷에 직접 연결된 형태로써 클라이언트가 Sink 노드로부터 직접 데이터를 전송 받을 수 있다.

USN 미들웨어 플랫폼은 기본적으로 클라이언트, Data 서버, Event 서버, USN Directory 서버 그리고 Local USN으로 구성된다. 클라이언트는 USN Client Application이 USN Host-side 미들웨어를 통하여 사용자가 원하는 USN 인프라 서비스를 각종 USN 서버에게 요구하는 주체이다. Local USN은 클라이언트가 원하는 정보를 실시간으로 생성하며, Data 서버는 Local USN에서 생성된 정보를 저장/관리한다. Event 서버는 수집된 데이터가 특정 조건에 부합될 때 비동기적으로 발생하는 이벤트를 관리한다. USN Directory 서버는 USN 자원에 대한 정보를 관리하며 USN 자원에 대한 정보 검색을 실시간으로 제공한다. 일반적으로 클라이언트는 USN 자원에 대한 정보를 알지 못하기 때문에 USN Directory 서버에게 질의하여 각종 USN 자원에 대한 정보를 제공 받는다.

개별 USN에서 발생하는 데이터가 클라이언트에 제공되는 방식으로는 임의의 순간 클라이언트로부터의 데이터 요구를 실시간으로 제공하는 Request/Response 풀링 방식, 클라이언트가 원하는 데이터를 주기적으로 클라이언트에게 전송하는 모니터링 방식과 “온도가 50도 이상일 때 데이터를 전송하라”와 같이 클라이언트가 지정한 조건이 만족되었을 때 발생하는 이벤트를 실시간으로 통보하는 방식이 있다. 풀링 및 모니터링 방식의 경우 Data 서버를 통해 데이터를 전송받으며 이벤트 방식의 경우 USN에서 발생한 이벤트는 Event 서버를 통하여 이벤트 통보를 신청한 클라이언트에게 제공된다. 이러한 Directory 서버 및 Event 서버는 클라이언트와 마찬가지로 상위 고급

서비스 처리를 지원할 수 있는 USN Host-side 미들웨어 상에서 동작한다.

Data 서버는 USN Service Application, USN 자원에 손쉽게 접근하기 위한 공통기능들을 제공하는 USN Host-side 미들웨어, USN Sink 노드와의 통신을 통해 Local USN망과 정보교환 및 상황설정 그리고 명령 전송을 지원하는 Data Protocol로 구성되어 Local USN과 인터넷을 연결하는 역할을 한다.

Local USN 망에서의 각 노드는 USN Node Application과 USN In-network 미들웨어로 이루어진다. USN In-network 미들웨어는 센서 노드에서 실행되는 응용 프로그램들이 필요로 하는 공통 기능 외에도 USN Node Application이 동적으로 탑재되고 실행되는 로딩기능 및 노드 태스크 관리 등의 다양한 기능을 제공하는 컴포넌트들을 포함하고 있다.

4.2 USN 서비스 시나리오

본 절은 위에서 설명한 USN 미들웨어 플랫폼의 시스템 구조를 바탕으로 클라이언트에게 USN 망에서 생성된 데이터를 어떤 과정을 거쳐 제공하는지에 대한 서비스 진행과정을 살펴본다. 위에서 언급하였듯이 클라이언트는 USN Directory 서버에게 질의하여 각종 USN 자원에 대한 정보를 제공 받는다. 따라서 USN Directory 서버는 모든 USN 자원에 대한 정보(metadata)를 저장하고 있어야 한다. 각 USN 자원에 대한 정보는 시스템이 초기화 될 때 또는 USN 자원이 시스템에 추가될 때 USN Directory 서버에 등록될 수 있다. USN Directory 서비스는 중앙집중식, 계층적, Peer-to-Peer 방식으로 구현될 수 있으나 본 절은 지면상의 제한으로 간결한 설명을 위해 중앙집중식 USN Directory 서비스를 가정한다. 또한 데이터 제공 방식 중 풀링 및 모니터링 방식에 대한 서비스 시나리오만을 기술한다.

일반적으로 대부분의 USN 서비스는 사용자가 USN Directory 서비스에 대한 질의로 시작하여 최종적으로 Local USN 망에서 생성된 데이터가 Data 서버를 통하여 클라이언트에게 전달되는 과정으로 이루어져 있다. 센서 데이터 풀링/모니터링 서비스는 그림 15와 같은 과정으로 이루어진다. 풀링 서비스는 구체적으로 다음과 같은 과정을 거쳐 진행되며, 모니터링 서비스의 경우 ⑥-⑧번 과정이 주기적으로 발생하는 것을 제외하면 풀링 서비스와 진행과정이 같다.

- ① 클라이언트는 “한국대학교 과학관 202호 온도”에 관한 센서 정보를 얻기 위해 USN Directory 서버에게 한국대학교 과학관의 Local USN을 담

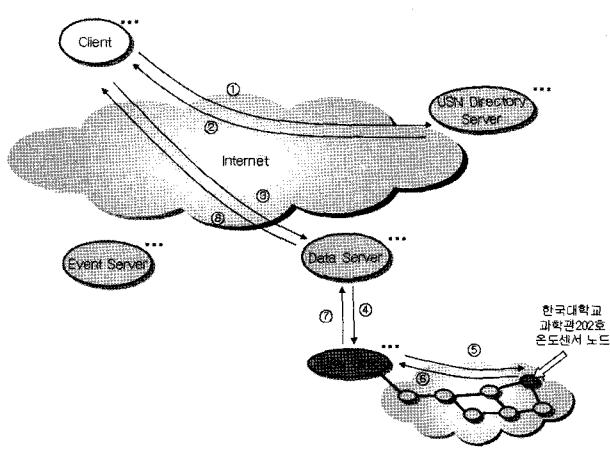


그림 15 센서 데이터 풀링/모니터링 서비스

- 당하는 Data 서버의 주소를 요구하는 메시지를 보낸다.
- ② USN Directory 서버는 클라이언트가 보낸 요청 메시지에 대한 응답으로 한국대학교 과학관을 관리하는 Data 서버의 주소를 넘겨준다.
 - ③ Data 서버의 주소를 전송받은 클라이언트는 Data 서버에 접속하여 과학관 202호의 온도 정보를 요청한다.
 - ④ Data 서버는 클라이언트가 원하는 온도 정보를 USN Sink 노드를 통해 202호 담당 USN 망에 요청한다.
 - ⑤ USN Sink 노드는 USN 망 내의 과학관 202호에 설치된 USN 노드에게 온도 데이터 전송을 요구 한다.
 - ⑥ 202호 USN 노드는 온도 정보를 USN Sink 노드로 전송한다.
 - ⑦ USN Sink 노드는 과학관 202호의 온도 정보를 Data 서버로 중계한다.
 - ⑧ Data 서버는 Local USN 망에서 생성된 과학관 202 호의 온도 정보를 클라이언트 측에 전달한다.

4.3 USN 미들웨어 소프트웨어 구조

위에서 언급하였듯이 USN 미들웨어는 제한된 자원의 USN Node로 구성된 Local USN 망을 위한 In-network 미들웨어와 상대적으로 많은 자원을 가진 Data

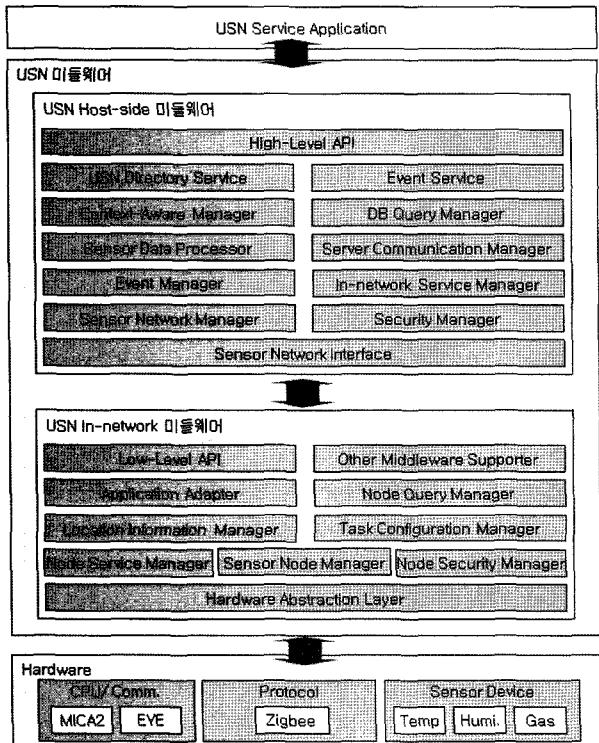


그림 16 미들웨어 소프트웨어 구조

서버에 위치하는 Host-side 미들웨어로 구성된 Two-Tier 구조를 택해야 한다. ETRI의 COSMOS, TinyDB 그리고 DSWare는 Two-Tier 구조를 가지고 있는 미들웨어의 예이다. 본 장은 USN 미들웨어 플랫폼의 기능요구사항을 바탕으로 설계한 USN 미들웨어의 소프트웨어 구조를 제시한다. 그럼 16은 USN 미들웨어의 기본적인 구성을 보이고 있다. Host-side 미들웨어는 High-Level API(Application Interface), USN Directory Service, Event Service, Context-Aware Manager, DB Query Manager, Sensor Data Processor, Server Communication Manager, Event Manager, In-network Service Manager, Sensor Network Manager, Security Manager 그리고 Sensor Network Interface로 구성된다. 각 구성요소에 관한 자세한 사항은 다음과 같다.

- High-Level API

High-Level API는 상위 USN Service Application과의 직접적인 통신을 하기 위한 컴포넌트이며 Host Service Listener와 Host Service Binder로 구성된다. Host Service란 USN Service Application에게 제공하기 위한 USN Host-side 미들웨어가 제공하는 서비스를 지칭한다. 반면 Node Service는 USN In-network 미들웨어가 제공하는 서비스를 말한다. 이는 USN In-network 미들웨어를 통하여 센서 노드에서 측정된 온도, 습도, 가스 등의 센서 정보를 제공하는 서비스이다. 이렇게 Host Service와 Node Service를 바탕으로 USN Service Application은 실시간 고부가가치 데이터 서비스를 클라이언트에게 제공한다.

High-Level API 컴포넌트의 구성 요소 중 Host Service Listener는 USN Service Application이 요구한 서비스 데이터를 수신하기 위한 모듈이며 Host Service Binder는 USN Service Application과 Host Service를 연결시키는 역할을 하는 모듈이다.

- USN Directory Service

USN Directory Service는 사용자가 요구하는 USN 데이터를 담당하는 Data 서버 또는 Event 서버를 검색하여 그 결과를 응답하는 역할을 하며 Event-Data (ED) Server Binder와 ED Server Finder로 구성된다. ED Server Binder는 주어진 요구 데이터를 Event 서버 또는 Data 서버 주소와 맵핑하기 위한 모듈이고, ED Server Finder는 서비스를 제공하는 목적지 Event 서버 또는 Data 서버를 검색하기 위한 모듈이다.

- Event Service

Event Service는 USN 노드에서 생성되는 이벤트 관련 데이터를 분석하여 이벤트 발생시 이벤트 정보를

제공하기 위한 것으로 IPv6 스택을 기반으로 모든 USN 노드에 IP 주소를 할당하여 해당 주소의 USN 노드로부터 발생되는 센싱 데이터에 관계되는 이벤트를 제공한다. 이벤트란 사용자가 요구한 지역의 “50도 이상 고온 이벤트”와 같이 조건문으로 표현될 수 있으며 발생 시점의 예측이 불가능하나 조건문에 부합되는 상황 발생시 실시간으로 통보 받아야 하는 사건을 일컫는다.

- Context-Aware Manager

Context-Aware Manager는 센서 데이터로부터 실시간 상황정보(규칙, 패턴)를 마이닝하여 응용 서비스에 전달하고, 지능적으로 상황을 판단하여 자율적 의사 결정 및 서비스 실행이 가능한 에이전트 기능을 제공하며 USN 망에서 발생하는 다양한 데이터들을 토대로 Context를 보급하는 Context Disseminator, 상황 인지 서비스를 발견하기 위한 Service Discovery, 지능적 상황판단을 위한 추론 엔진인 Reasoning Engine, 모듈 관리자인 Context Manager로 구성된다.

- DB Query Manager

DB Query Manager는 사용자가 DB 형식의 Query 문을 통해 센서 데이터에 대한 요청을 할 때 DB Query 문을 USN에 적합한 명령으로 해석하여 각 USN 노드에게 DB 응답에 필요한 센서 데이터를 수집하기 위한 명령을 전달해주는 컴포넌트이다.

- Sensor Data Processor

USN에서 생성된 데이터들의 종류로는 비동기적 이벤트를 기록하는 것과 데이터 쿼리를 통한 기록, 실시간 모니터링, 작업 명령, 에이전트를 통한 데이터 수집 등이 있다. 이러한 데이터들을 처리하기 위하여 Sensor Data Processor는 Asynchronous Event Reporting, Data Query Model Reporting, Real-Time Monitoring, Work Commander, Agent로 구성된다.

- Server Communication Manager (SCM)

USN 인프라에는 여러 서버(Directory 서버, Event 서버, Data 서버)들이 인터넷에 분산되어 존재하는데 서버 간 원활한 통신을 지원하기 위하여 Server to Server Protocol을 기반으로 한 SCM이 필요하다.

- Event Manager

Event Manager는 Event Service를 제공하기 위해 USN 망에서 발생하는 이벤트들을 관리하는 컴포넌트이며 이벤트의 흐름을 제어하기 위한 Event Flow Control 모듈과 이벤트를 수신하기 위한 Event Listener

모듈로 구성된다.

- In-network Service Manager

In-network Service Manager는 USN Node들이 제공하는 온도, 습도, 가스 등의 센싱 정보들을 관리하기 위한 컴포넌트로 Host Service와 Node Service 연결을 위한 Node Service Binder 모듈과 Host Service가 어떠한 Node Service를 요구하는지 수신하기 위한 Node Service Listener 모듈로 구성된다.

- Sensor Network Manager

Sensor Network Manager는 Local USN Node들을 관리하기 위한 컴포넌트로 USN 노드를 찾기 위한 Node Discovery 모듈과 배터리의 잔량 등 USN 노드의 상태를 모니터링 하기 위한 Sensor Monitor 모듈로 구성된다.

- Security Manager

Security Manager는 악의적인 침입이나 거부 공격(denial of service)과 같은 외부 공격 및 USN Host-side 미들웨어에서 수집된 데이터를 검증하기 위한 기능을 가진다. 또한 USN In-network 미들웨어 센서 데이터를 보호하기 위한 정책을 제공한다.

- Sensor Network Interface

Sensor Network Interface는 USN Host-side 미들웨어와 USN In-network 미들웨어간의 통신을 위한 컴포넌트로 USN 망에서 발생한 이벤트를 인지·생성하는 Event Generator, USN Node 내의 센서 설정 정보를 담고 있는 Sensor Profiler, USN In-network 미들웨어와 통신을 하기 위한 Protocol Stack/Relay, 명령을 전달하고 처리하기 위한 Command Processor, 센서 데이터를 송수신하기 위한 Data Communication 모듈로 구성된다.

USN In-network 미들웨어는 Low-Level API, Other Middleware Supporter, Application Adapter, Node Query Manager, Location Information Manager, Task Configuration Manager, Node Service Manager, Sensor Node Manager, Node Security Manager 그리고 Hardware Abstraction Layer로 구성된다. 각 구성요소에 대한 설명은 다음과 같다.

- Low-Level API

USN Node에 탑재된 응용프로그램이 미들웨어 서비스를 사용하고자 할 때 연결 기능을 제공하는 컴포넌트이다.

- Other Middleware Supporter

Other Middleware Supporter는 기존의 RFID, 텔레매틱스, 홈 네트워크, 센서 네트워크 분야의 미들웨어인 OSGi, Jini, Havi, EPC 등을 지원하여 통합적인 USN 인프라를 이룰 수 있다.

- Application Adapter

Application Adapter는 필요에 따라 응용프로그램을 동적으로 설치 및 실행함으로서 보다 다양한 서비스를 제공한다. 예를 들어 온도를 센싱하는 응용프로그램이 실행중에 있는 USN Node에게 습도를 측정하는 새로운 작업을 부여하고자 할 때 데이터 서버는 습도 센싱을 위한 응용 프로그램을 다운로드 · 설치 · 실행하도록 하여 USN Node의 교체 없이 즉각적인 정보수집이 가능하다.

- Node Query Manager

센서 노드에서 쿼리 연산을 수행한다. 쿼리에 대한 오류검증 및 최적화는 Data 서버에서 수행되고, 여기서는 쿼리처리를 위한 기본적인 연산만을 수행한다.

- Location Information Manager

USN Node는 자신의 위치 정보를 내부적으로 생성 및 기록할 필요가 있으며 Location Information Manager가 이 작업을 담당한다. USN Node의 위치정보는 USN Host-side 미들웨어를 통해 Data 서버에게 알려져서 DB Query 및 이벤트 인지를 위한 데이터 처리 · 수집 등에 사용된다.

- Task Configuration Manager

여러 상황에 기민한 반응 할 수 있도록 Task 구성 을 지원하는 컴포넌트이다.

- Node Service Manager

USN In-network 미들웨어가 제공하는 Node Service(온도, 습도, 가스 등 센서 데이터 수집 서비스)는 USN Host-side 미들웨어를 거쳐 Host Service를 통해 집약되며 USN 데이터 서비스로서 클라이언트에게 전달되는데 Node Service Manager는 USN Host-side 미들웨어에게 USN 망의 Node Service를 제공하는 역할을 한다.

- Sensor Node Manager

USN Node를 관리하기 위한 공통기능 컴포넌트로 시간 동기화를 위한 Clock Synchronizer와 Task Scheduler로 구성된다.

- Node Security Manager

USN Node들 간의 무선 통신은 외부 공격에 의하여

센싱 정보를 도청 당하거나, 또는 센싱 정보가 조작될 수 있는 가능성이 매우 높다. Node Security Manager는 In-network 미들웨어에서 센서 데이터를 보호하기 위한 기능을 제공한다.

- Hardware Abstraction Layer

하드웨어 디바이스를 추상화하는 기능을 담당하는 컴포넌트로 CPU/Communication Controller, Protocol Stack, Sensor Controller로 구성된다. CPU/ Communication Controller는 노드를 구성하는 하위 디바이스를 관리하며 Protocol Stack은 USN Node간 통신을 위한 프로토콜 스택이며, Sensor Controller는 온도, 습도, 가스 등의 센서 디바이스들을 제어하기 위한 모듈이다.

5. 결 론

본 논문에서는 센서 네트워크 미들웨어의 기능 요구사항을 기술하였으며, 현재까지 연구된 다양한 센서 네트워크 미들웨어들을 위치 및 기능적으로 분류한 뒤 각 미들웨어의 기술개발 동향에 대해 살펴보았다. 그리고 이러한 다양한 분야의 미들웨어들을 바탕으로 통합 설계 · 제작된 USN 미들웨어 플랫폼을 소개하였다. 특히 Two-Tier 구조의 USN 미들웨어는 인간, 컴퓨터, 사물이 유기적으로 연결된 USN 인프라 상에서 Anytime, Anywhere, Anything 형태의 고부가 가치 인프라 서비스를 효과적으로 제공할 수 있을 것으로 기대된다.

참고문헌

- [1] 정보통신부, “u-센서 네트워크 구축 기본계획”, 2004.
- [2] 김민수, 김광수, 이용준, “USN 미들웨어의 특징 및 기술개발 동향”, 주간기술동향, 제1284호, 2007.
- [3] 김영만, “USN 인프라 서비스 지원을 위한 계층적 미들웨어 플랫폼 구조”, 정보처리학회지, 2005.
- [4] 김영만, “센서 네트워크 미들웨어 구조 및 연구 현황”, 정보과학회지, 제22권, 제12호, 2004.
- [5] Mohammad M. Molla and Sheikh Iqbal Ahamed, “A Survey of Middleware for Sensor Network and Challenges,” 2006 International Conference on Parallel Processing Workshops(ICPPW'06), pp. 223–228, 2006.
- [6] 김대영, 성종우, 송형주, 김수현, “센서 네트워크 미들웨어 기술”, 전자공학회지, 제32권, 제7호, pp. 800–814, 2005.
- [7] Salem Hadim and Nader Mohamed, “Middleware Challenges and Approaches for Wireless Sensor Networks,” IEEE Distributed System Online, Vol. 7,

- No. 3, 2006.
- [8] Y. Yao and J. Gehrke, "The Cougar Approach to In Network Query Processing in Sensor Networks," SIGMOD Record, Vol. 31, No. 3, 2002.
- [9] S. R. Madden, M. J. Franklin, and J. M. Hellerstein, "TinyDB: An Acquisitional Query Processing System for Sensor System Networks," ACM Trans. Database Systems, Vol. 30, No. 1, pp. 122–173, 2005.
- [10] S. Li, S. Son, and J. Stankovic, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks," Proc. 2nd Int'l Workshop Information Processing in Sensor Networks(ISPN 03), LNCS 2634, pp. 502–517, 2003.
- [11] Chien-Liang Fok, Gruia-Catalin Roman, and Chen-yang Lu. "Agilla: A Mobile Agent Middleware for Sensor Networks," Washington University in St. Louis, Technical Report, 2006.
- [12] A. Boulis, C. C. Han, and M. B Srivastava, "Design and Implementation of a Framework for Efficient and Programmable Sensor Networks," In MobiSys, San Francisco, USA, 2003.
- [13] R. Barr , et al., "On the Need for System-Level Support for Ad hoc and Sensor Networks," Operating Systems Rev., Vol. 36, No. 2, pp. 1–5, 2002.
- [14] P. Levis and D. Culler, "Mate: A Tiny Virtual Machine for Sensor Networks," Proc. 10th Int'l Conf. Architectural Support for Programming Languages and Operating System(ASPLOS-X), ACM Press, pp. 85–95, 2002.
- [15] T. Liu and M. Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," Proc. ACM SIGPLAN Symp. Principles and Practice of Parallel Programming(PPoPP 03), pp. 107–118, 2003.
- [16] A. Murphy and W. Heinzelman, "MiLAN: Middleware Linking Applications and Networks," TR-795, University of Rochester, Computer Science, 2002.
- [17] E. Souto et al., "A Message-Oriented Middleware for Sensor Networks," Proc. 2nd Int'l Workshop Middleware for Pervasive and Ad-Hoc Computing (MPAC 04), ACM Press, pp. 127–134, 2004.
- [18] J. W. Branch, J. S. Davis, D. M. Sow, and C. Bisdikian, "Sentire: a framework for building middleware for sensor and actuator networks," in Proc. of the 3rd IEEE PERCOM Workshop, pp. 396–400, 2005.
- [19] Brian Schott, Bob Parker, Mani Srivastava, and Charles Chien, "PADS: Power Aware Distributed Systems," USC/ISI, 2000(<http://pads.easet.isi.edu/>).
- [20] Maroti M., Kusy B., Balogh G., Volgyesi P., Molnar K., Nadas A., "Radio Interferometric Geolocation," SenSys 05, San Diego, 2005.
- [21] Maroti M., Kusy B., Simon G., Ledeczi A., "The Flooding Time Synchronization Protocol," SenSys 04, Baltimore, 2004.
- [22] 김민수, 이용준, 박종현, "USN 미들웨어 기술개발 동향", 전자통신동향분석 제22권 제3호 2007년 6월.



김영만

1976~1980 서울대학교 기계공학과
1980~1982 한국과학기술원 기계공학과
1982~1983 중소기업진흥공단
1983~1986 LG전자 연구원(Robot, NC M/C, Automation, Compiler/OS, 광통신)
1987~1988 Ohio State Univ. 전산학 석사
1989~1992 Ohio State Univ. 전산학 박사
1992~1996 Mitsubishi Materials Corp., Central Research Lab. 초빙연구원(공장자동화용 펄드네트워크 설계/개발)
2001~2004 Ohio State Univ. 초빙 연구원(DARPA NEST 센서네트워크 미들웨어 프로젝트 수행)
1996~현재 국민대학교 컴퓨터공학부 교수
관심분야 : USN(미들웨어, 실시간 라우팅 알고리즘, 디지털 신호처리, 무선 프로토콜(WLAN, WPAN, Mobile), 소프트웨어 스트리밍, TPM 기반 신뢰컴퓨팅
E-mail : ymkim@kookmin.ac.kr



한재일

1976~1980 연세대학교 수학과
1984~1986 Syracuse University, 전산학 석사
1986~1992 Syracuse University 전산학 박사
1993~1995 한국전자통신연구소 선임연구원
1995~1995 안양대학교 전임강사
1995~현재 국민대학교 컴퓨터공학부 교수
관심분야 : 미들웨어, 분산처리 시스템, 객체지향 시스템, 컴퓨터 네트워크 보안
E-mail : jhan@kookmin.ac.kr