

# 센서 네트워크 미들웨어 구조 및 연구현황

국민대학교 김영만\*

## 1. 서론

무선 센서 네트워크는 성숙 단계에 들어가고 있는 유비쿼터스 컴퓨팅 기술의 도움과 함께 넓은 범위에 걸쳐서 분산되어 있는 센서 디바이스들로부터 감지된 상황인지 데이터들을 무선 네트워크를 통하여 응용 서비스 서버와 연동하는 기술이다. 이러한 센서 네트워크를 통해 사람이 감시할 수 없는 지역에 대한 환경 감시, 집이나 사무실에 대한 자동화 및 방법, 건강 모니터링, 움직이는 물체에 대한 트래킹과 같은 많은 응용 서비스들을 제공하기 위하여 센서 네트워크는 경량의 하드웨어 플랫폼과 높은 수준의 동적 환경에 적응함과 동시에 일정 수준의 정확성을 확보해야 한다. 따라서 센서 네트워크용 미들웨어[1,2]는 운영체제와 응용 소프트웨어 사이에서 네트워크 연결성과 높은 수준의 시스템 인터페이스 제공과 같은 높은 수준의 추상화를 제공함으로써 응용 프로그램의 복잡성을 감소시키고, 개발자로 하여금 프로그램 구현과 디자인을 간편하게 할 수 있도록 하여야 한다.

센서 네트워크용 미들웨어는 다양한 센서 응용 소프트웨어와 운영체제 및 네트워크 스택 사이에 존재하며, 이종 센서 노드 기반의 다양한 응용 시스템 개발, 유지보수, 설치, 수행에 필요한 제반사항을 지원한다. 그러나 현재까지 센서 네트워크의 하드웨어와 통신 프로토콜 등 기반 기술에 대한 연구는 활발히 진행되어온 반면, 미들웨어 기술은 아직 초기단계에 머물러 있다.

미들웨어의 주요 기능으로는 상황인지 이벤트 발생 및 전달, 센서 분산 데이터베이스, 무선을 통한 원격 작업모듈 재설치, 문맥인지 미들웨어 등이 있다. **본 논문에서는 제한된 자원을 가진 센서 네트워크에 적합한 미들웨어의 사양과 요구 기능 및 미들웨어 구조들을 살펴보고 현재 활발히 연구되고 있는 대표적인 센서 네트워크 미들웨어들에 대하여 분석해 보기로 한다.**

본 논문의 구성은 다음과 같다. 2장에서는 센서 네트

워크 미들웨어가 가져야만 하는 사양, 기능 및 구조들에 대하여 다룬다. 3장에서는 현재 연구개발 중인 여러 미들웨어들에 대하여 2장에서 제시한 틀을 이용하여 분석 평가한다. 마지막으로 4장에서는 본 논문의 결론을 맺고 센서 네트워크 미들웨어에 대한 향후 전망에 대하여 논한다.

## 2. 센서 네트워크 미들웨어 구조

센서 네트워크가 가지는 특징으로부터 미들웨어가 제공해야 할 사양들을 도출할 수 있다. 또한 응용 소프트웨어들이 필요로 하는 기능들 중 공통적인 것들은 미들웨어에서 제공하는 것이 시스템을 효율적으로 구성하는데 필수적인데 이러한 미들웨어 사양과 기능에 적합한 센서 네트워크용 미들웨어 모델구조들이 존재한다. 이 장에서는 먼저 센서 네트워크 미들웨어에 적합한 사양에 대하여 논하고 이어서 바람직한 센서 네트워크 미들웨어 기능들을 조사해 보고 마지막으로 센서 네트워크 미들웨어 구조들에 대하여 논한다.

### 2.1 센서 네트워크 미들웨어 사양

센서 네트워크가 가지는 특징들 즉, 제한된 자원(CPU 성능, 메모리, 무선 통신의 대역 및 한계영역), 통신장애, 센서 노드의 이질성, 네트워크의 확장성, 무인 운용 등으로 인해 발생하는 문제를 효율적으로 처리하기 위해 미들웨어는 다음과 같은 사양을 만족하여야 한다.

- **지역적 알고리즘(localized algorithm)** : 가급적이면 일정 범위 내에 속한 이웃 노드들만의 통신에 의해서 달성되는, 분산 알고리즘으로 구현하여야 한다. 이는 네트워크 확장 혹은 분할이나 노드의 장애도 쉽게 극복할 수 있게 한다.
- **품질 조절 가변 알고리즘(adaptive fidelity algorithm)** : 주어진 문제에 대하여 제한 자원의 활용을 조절하여 해답을 구하는 알고리즘이 제공되어야 한다.

\* 정회원

- **데이터 중심 통신(data-centric communication)** : 센서 노드에 의해 만들어지는 데이터가 통신의 주체가 되는 새로운 영역의 통신이다. 예를 들어, 온도 센싱에 대한 응용의 경우, 기존의 통신처럼 지칭된 특정 노드들에 감지되는 온도 데이터를 단순히 요구하고 수신하는 형태가 아니라, 특정 온도가 메시지에 포함되면서 전달돼 이를 넘어서는 지역의 위치 정보를 발생시키도록 노드들에게 요구한다. 이처럼 센서 네트워크의 미들웨어에서는 기존의 미들웨어와는 달리 통신 뿐만 아니라 응용부분의 데이터를 처리하는 기능이 통합적으로 제공돼야 한다.
- **응용정보에 기반한 설계 및 운용(application knowledge in nodes)** : 기존의 네트워크에서는 응용부분에 대한 정보를 반영하지 않고 다양한 영역의 응용을 지원할 수 있도록 만능 미들웨어의 설계가 요구됐다. 하지만 센서 네트워크에서는 이러한 정보를 네트워크 기반 구조나 노드 자체에 주입할 수 있는 방법까지 미들웨어에서 제공하도록 요구한다. 예를 들어, 중간 노드에서는 응용정보에 맞춰 데이터를 캐싱하고 다수의 데이터 통합을 수행해 자원이나 에너지 효율성을 효과적으로 향상시킬 수 있도록 한다.
- **기존 네트워크와의 연계(interaction with the traditional networks)** : 센서 네트워크는 이용 가능한 자원이 제한되어 있는 특성으로 인해 자원 집약적으로 요구되는 기능이나 대규모 정보의 저장을 센서 네트워크 외부의 다른 네트워크나 장치에서 수행하도록 하는 것이 적합하다. 예를 들어, 인터넷에 내재된 이에 대응하는 가상의 기능 구조, 또는 구성요소가 이러한 기능을 대신 수행할 수 있다.

## 2.2 센서 네트워크 미들웨어 요구기능

센서 네트워크를 위한 미들웨어는 기본적으로 네트워크 요소의 개발, 센서 노드의 배포, 유지 및 관리, 감지 기반 응용의 수행을 원활하게 하는 것을 기본 목적으로 한다. 응용부분에서 요구한 문제에 적절한 답을 제공하기 위해 미들웨어는 통신의 흐름에 따라 표면적으로는 상위 레벨 태스크를 형성하고, 이들 태스크와 무선 센서 네트워크간의 통신을 실시한 후, 형성된 태스크를 노드 각각의 역량에 맞게 분할해 각 노드에 배포한다. 계속해서 감지된 센서 데이터를 조합해 상위 레벨에 적합한 정보로 만들기 위한 데이터 융합(data fusion) 및 전달을 실시한 후, 최종적으로 사용자에게 데이터를 보고하는 기능을 제공해야 한다.

한편, 센서 네트워크는 제한된 에너지의 효율적 이용,

동적 환경에서의 네트워크 건설성, 네트워크 확장성을 요구하는 특징을 갖고 있다. 따라서 네트워크의 효과적인 형성, 유지, 관리 등을 위해 미들웨어는 다음과 같은 기능이 요구된다.

- **전력 관리(power management)** : 센서 네트워크 노드들은 프로세싱 전력이 아주 제한적이므로 에너지를 효율적으로 소모하면서 기능을 수행하는 미들웨어 플랫폼으로 설계돼야 한다.
- **위치 인식(localization)** : 센서 네트워크는 센서에서 감지된 센서 데이터 전달뿐만 아니라 응용부분에서 지시하는 메시지 내의 컨텍스트 조건을 만족하는 센서 노드들을 파악하기 위해 자신의 위치를 인식해야 한다.
- **시간 동기화(time synchronization)** : 다양한 센서 노드들 간의 협력작업을 위해 시간 동기화 기능이 요구된다.
- **소프트웨어 배포 및 자동 갱신(software update)** : 이동성이 있거나 지리적으로 널리 분산되어 있는 많은 노드들에 새로운 코드를 전송하기 위한 메커니즘을 제공해야 한다.
- **센서 정보 저장(sensor database)** : 센서 노드들로부터 감지되는 센싱 정보들을 효율적으로 관리하는 기법이 필요하다.
- **데이터의 분배 및 복제(data placement & replication)** : 센싱 데이터의 접근성을 높이고, 데이터 전송에 의한 과도한 에너지 소모를 줄이기 위해, 센싱된 데이터를 최적의 위치에 저장하고, 필요시 복사본을 데이터 접근이 용이한 곳에 저장하는 기법이 필요하다.
- **보안(security)** : 센서 네트워크의 초소형 노드는 대부분 메모리 관리부가 없기 때문에, 신뢰할 수 없는 의심스러운 코드가 수행되지 않도록 미들웨어 플랫폼에 추가적인 보안기능이 요구된다.
- **센서 노드 이질성의 일반화(abstraction of sensor node's heterogeneity)** : 센서 네트워크 미들웨어는 이질성으로 파생하는 여러 가지 제약, 또는 다양한 문제 상황에 대처할 수 있도록 설계돼야 한다.
- **장애 관리(fault tolerance)** : 네트워크의 신뢰도와 가용성이 센서 네트워크가 사용하는 무선 환경보다 더 낮기 때문에 통신 실패가 훨씬 자주 발생하므로 장애 극복 기능을 가져야 한다.

## 2.3 센서 네트워크 미들웨어 모델구조

센서 네트워크 미들웨어는 제한된 자원을 가진 노드

상에서 장시간에 걸쳐 고장 및 요구사항의 변경에 적응하여야 하며 물리적인 세계에서 발생하는 센서 데이터를 데이터 사용자에게 높은 신뢰도로 전달해 주는데 필요한 공통 서비스를 제공하여야 한다. 본 절에서는 이러한 상황에 적합한 미들웨어 모델구조들에 대하여 제안한다.

### 2.3.1 데이터 중심 구조

센서 네트워크는 사용자에게 센서 데이터를 제공하는 서비스를 기본으로 한다. 따라서 센서 네트워크 미들웨어는 사용자가 원하는 센서 데이터를 수집하고 전달하는 과정을 효율적이고 신뢰성 있게 진행하는데 있어서 필요한 공통 기능들을 서비스로 제공하는 것이 필요하다. 다음 장에서 소개하는 미들웨어들 중 일부는 이러한 데이터 중심 서비스를 위한 미들웨어 구조를 구현하고 있다. 수집된 센서 데이터는 그 성격상 단독으로 존재하는 독립 데이터가 있으며 또한 여러 장소에서 수집된 측정 데이터들을 처리하여 얻어지는 협동 데이터가 있다. 요구되는 신뢰도를 가진 협동 데이터를 생성하는데 있어서 적절한 노드 수 및 위치들을 결정하고 작업 내용을 전달한 후 측정된 데이터들을 모아서 처리하여 협동 데이터를 생성하는데 있어서의 일련의 과정을 지원하는 기능이 미들웨어에서 제공되어야 한다.

### 2.3.2 이벤트 중심 구조

어떤 센서 데이터는 사용자가 원하는 순간에 존재하지 않고 임의의 순간에 스스로 발생하여 사용자에게 비동기적으로 전달되는 경우가 있다. 예를 들면, 센서 네트워크가 산림 안에 분산 설치되어 있는 경우, 산림내의 온도 데이터는 임의의 순간에 측정되어질 수 있어 사용자가 원하면 즉시 수집 전달이 가능하므로 사용자와 센서네트워크 사이의 관계는 앞의 데이터 중심 구조로 자연스럽게 설명되나, 산림에서 발생하는 화재는 언제 발생하는지에 대한 예측을 허용하지 않으므로 사용자는 화재 데이터를 원하더라도 그 데이터는 당장 전해지지 못하고 시간이 흘러서 화재가 실제로 발생하는 순간 화재 발생이 이벤트 형태로 사용자에게 전달되므로 이러한 서비스는 위의 데이터 중심 구조에 적합하지 못하다. 따라서 이러한 비동기적 센서 이벤트 발생 및 전달에 초점을 맞추어, 센서 이벤트를 발생시키고 이벤트 통보를 원하는 사용자에게 전달하는데 필요한 상위 기능들을 제공하는 이벤트 중심 구조의 미들웨어가 필요하다.

### 2.3.3 이동 에이전트 지원 구조

센서 네트워크는 한번 설치되면 관리자의 손길이 닿지 못하는 상황 하에서 오랜 기간 자율적으로 동작하여야 하므로 사용기간 중에 네트워크 용도 변경 등으로 인하여 응용 소프트웨어나 프로토콜들이 바뀌어야 할 때

소프트웨어 모듈을 동적으로 다운로드하고 기존의 응용 프로그램을 대체하도록 지원하는 플랫폼이 필요하다. 이 개념을 확장시킨 것이 이동 에이전트 모델로서 응용 프로그램 뿐만 아니라 미들웨어나 운영체제를 구성하는 모듈들을 자동적으로 업데이트할 수 있도록 하는 구조의 미들웨어가 필요하다.

### 2.3.4 제한자원 보호 구조

센서 네트워크는 제한된 에너지를 가진 배터리를 사용하여 장기간 동작하는 것을 전제로 하므로 에너지를 최대한 절약하는 방안이 모든 소프트웨어 모듈들에 적용되어야 한다. 따라서 에너지 절감을 포함하여 부족한 자원(메모리, 계산 기능 등)을 효율적으로 사용하는데 필요한 각종 기능들과 에너지 절감형 공통 서비스 모듈들을 제공하는 자원보호 미들웨어가 필요하다.

### 2.3.5 공통 센서기능 제공 구조

센서 네트워크가 제공하여야 할 공통 서비스들을 모아서 미들웨어에서 제공하는 것은 모든 미들웨어의 기본적인 사양에 해당된다. 특히 센서 네트워크가 제공할 수 있는 공통 서비스로 시간 동기화 서비스, 노드 위치 인식 서비스 등이 있다.

## 3. 센서 네트워크 미들웨어 연구동향

이 장에서는 앞에서 제안한 미들웨어 모델구조에 따라 현재 연구 중에 있는 대표적인 미들웨어들을 분석한다.

### 3.1 데이터 중심 미들웨어

#### 3.1.1 Cougar

Cougar[3]는 코넬 대학의 데이터베이스 연구팀에 의해서 연구되고 있는 센서 네트워크용 분산 데이터 처리 시스템이다. 현재 연구되고 있는 많은 센서 네트워크 응용은 베이스 노드가 모든 데이터를 수집하고 중앙에서 데이터를 처리하는 방식을 취하고 있는 반면, Cougar에서는 데이터 접근과 처리가 모두 분산된 형태로 수행된다. Cougar는 서술적인(declarative) 쿼리를 사용하므로 사용자가 네트워크의 물리적 특성으로부터 가려지는 특성도 지닌다. Cougar는 네트워크의 변화에 동적으로 적응할 수 있고, 유연성과 확장성(scalability)이 높으며, 고장 감내성(fault tolerance)을 가진 시스템을 목표로 하고 있는데, 이 목표들에 대해 기본적으로 아래와 같은 방향으로 연구를 진행하고 있다.

- 센서 네트워크는 분산된 형태의 데이터베이스다. 사용자의 쿼리는 네트워크의 상황과 노드들이 가지는 잔류 전력, 처리능력 등을 고려해 최적으로 처리

된다.

- 최적의 동작을 위해서 데이터 관리계층 (data management layer)과 라우팅 계층 (routing layer)은 설계 단계에서부터 상호간의 최적화가 고려되어야 한다.

이러한 연구방향 하에서 Cougar의 연구팀은 합산 (aggregation), 다양한 응용분야를 지원하기 위한 쿼리 언어 (query language), 쿼리 최적화기 (query optimizer), 다중 쿼리 최적화 (multi-query optimization) 등에 대한 연구를 진행하고 있다.

Cougar의 적응형 쿼리 처리 (adaptive query processing) 기술은 센서 네트워크에서의 데이터 처리에 상당히 유용할 것으로 예상되나 센서 노드 자체가 가지는 여러 가지 제약사항이나 지역적인 정보에 의존해야 한다는 점에서 앞으로 더 많은 연구를 필요로 한다.

그림 1는 Cougar의 홈페이지에서 직접 실행해 볼 수 있는 온라인 데모로, 목표물의 위치 추적에 대한 쿼리를 만들고 실제로 추적되는 데모를 보여준다.

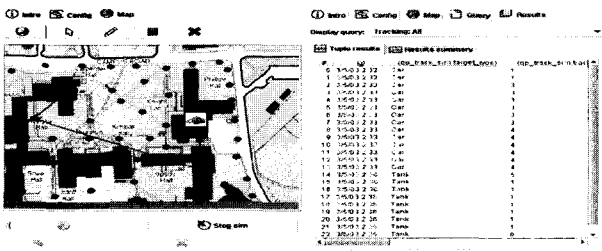


그림 1 목표물 위치추적에 대한 쿼리 생성 및 실행 데모

### 3.1.2 SAMANTA

SAMANTA (Swarm oriented Adaptive Middleware Architecture for Networked Time critical Applications)는 버지니아 대학에서 신뢰성 있는 데이터 관리를 위해서 설계된 미들웨어로서 주변 상황의 변화에 맞추어 데이터 보관 및 전달의 신뢰성을 높이기 위하여 데이터 복제 위치들을 동적으로 변화시키는데 이때 복제 위치는 메시지 전달과 복제에 필요한 전체 에너지를 최소화 하도록 선택된다. 그림 2는 SAMANTA의 미들웨어 내부구조를 나타낸다.

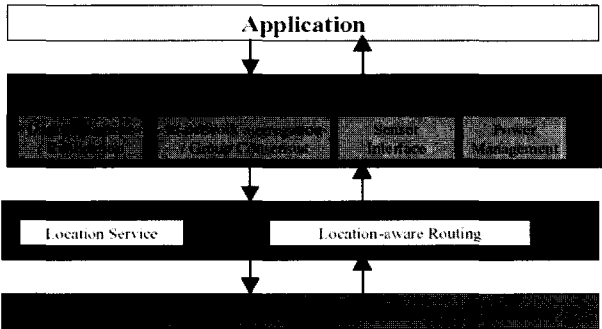


그림 2 SAMANTA 미들웨어 구조

### 3.1.3 SINA

SINA[4]는 센서 네트워크를 하나의 분산 데이터베이스로 간주하여 쿼리 형식을 사용해 센서 네트워크의 정보에 접근한다. SINA에서는 지리적으로 가까운 곳에 분포하는 센서 노드들로부터의 유사한 정보의 재전송을 제한하는 방법뿐만 아니라 효율적인 데이터 융합을 위해 센서들의 계층적인 그룹화를 수행하는 하위 레벨의 메커니즘도 포함한다.

## 3.2 이벤트 중심 미들웨어

### 3.2.1 DSWare

DSWare[5]는 일정 패턴으로 정의되는 복합된 이벤트를 기본 프로그래밍 기능으로 제공함으로써 데이터 서비스 기능을 수행하는데, 이벤트의 보고에 대한 데드라인과 확실한 이벤트에 대한 구간 등의 정의처럼 실시간적인 부분도 다양하게 제공하는 특징을 지닌다. 그림 3은 DSWare 미들웨어를 구성하는 모듈들 간의 관계를 나타낸다.

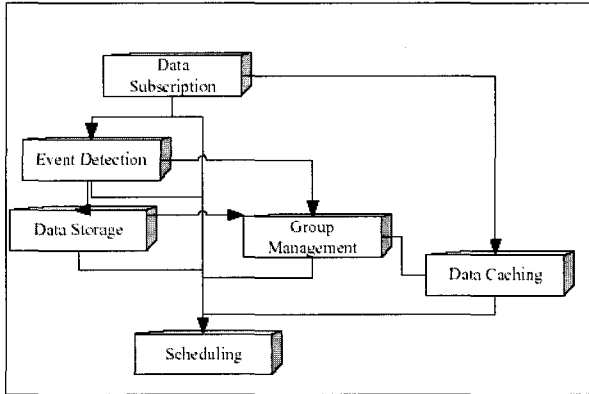


그림 3 DSWare 미들웨어 내부구조

## 3.3 이동 에이전트 지원 미들웨어

### 3.3.1 SensorWare

SensorWare[6]는 센서 네트워크의 효율적인 활용과 프로그래밍을 위한 제안된 프레임워크다. 스크립트화 할 수 있는 경량의 런타임 환경을 기반으로 하며, 더욱 간단하고 치밀하면서도 플랫폼에 무관하게 센서 노드를 제어할 수 있는 스크립트의 수행을 목적으로 한다.

SensorWare에서는 경량의 모바일 스크립트가 서로 다른 노드에 복사되거나 이동될 수 있으므로 사용자는 센서 네트워크의 제어를 위해 모든 센서와의 일대일 통신을 통한 정보처리 부담을 줄일 수 있다. 예를 들어, 사용자가 특정 사물의 위치를 파악하고자 한다고 했을 때, 사용자는 센서 네트워크로 스크립트나 간단한 상태 천이 머신 형태의 쿼리를 사용자 근처에 있는 하나 이상의 센서 노드에 보낸다. 이때 해당 센서 노드는 사용자가 원하

는 사물의 위치 추적과 연관 있는 센서 노드에 관련된 모바일 스크립트를 보내고, 이 스크립트는 이벤트나 상태에 따라 다른 영역으로 계속 복사되거나 이동하게 된다. 사물의 움직임이 발생했을 때, 센서 노드들은 서로 정보를 교환해 사물의 위치에 대한 정보를 파악한 다음, 사용자에게 사물의 위치를 알려주게 된다. 이 과정에서 SensorWare는 필요 없는 센서들 간의 통신을 줄여주므로 센서 노드의 전력 손실을 덜어줄 뿐만 아니라, 센싱된 정보의 신속한 전달을 가능하게 한다.

또한, SensorWare는 개발자로부터 자원관리 같은 하드웨어와 관련된 부분을 숨겨주고, 여러 응용 프로그램 사이에서 센서 노드끼리의 자원을 공유하는 방식을 제공해준다. 뿐만 아니라 고급 스크립트 언어를 사용하기 때문에 센서 네트워크 환경에서의 편리한 개발환경을 제공해 준다. 하지만, SensorWare는 메모리가 풍부한 환경에서 동작하는 멀티태스킹나 고급 스크립트 언어를 통해 운용되므로 Berkeley Mote처럼 제한된 메모리를 가진 센서 노드에는 적합하지 않다.

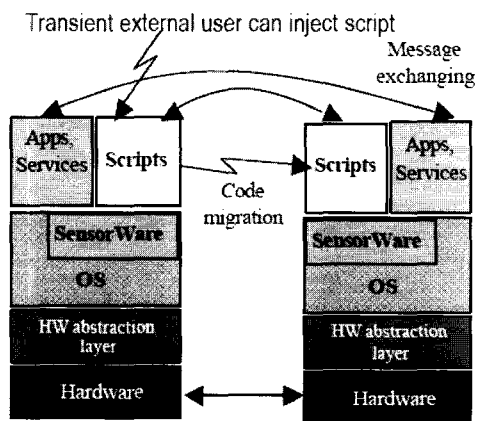


그림 4 센서 노드 구조

SensorWare는 그림 4와 같이 센서 노드 운영체제의 일반적인 기능과 멀티쓰레드 환경을 제공받으며, 제어 스크립트를 위한 런타임 환경을 제공한다. 이로 인해 센서 노드를 제어하는 스크립트가 네트워크로 복사, 또는 이동할 때 전적으로 SensorWare 계층에 의존하게 된다.

### 3.3.2 Maté

Maté(7)는 TinyOS 상에서 수행되는 초경량의 통신 중심 가상 머신 형태의 미들웨어로, 독자적인 Bytecode 해독기를 구비하고 있으며, 효율적인 에너지 방식을 통해 센서 네트워크의 재프로그래밍을 제공하고자 구현됐다. Maté는 전염모델(infection model)을 통해 새로운 코드를 배포하는 메커니즘을 가지고 있다. 따라서 프로그래머가 각 센서에 대한 새로운 코딩을 염려할 필요 없이 하나의 노드로 전달된 코드가 바이러스와 같은 형식으로

네트워크로 퍼져나가는 특징을 지닌다. Bombilla는 Maté 개념에 의하여 구현된 미들웨어의 명칭으로 TinyOS 상에서 동작하는 스택 기반의 작은 가상머신 (virtual machine)이다. 다양한 센싱 기능과 임무를 가지며 극도로 자원사용이 제한되어 있는 Mote (Berkeley 센서 노드)들로 구성된 센서 네트워크에, 동적 프로그래밍이 효율적으로 수행되도록 하기 위해 설계됐다. Maté에서는 수백 수천 개의 센서 노드마다 목적에 맞도록 응용 프로그램의 코드를 수정해 인스톨하는 것은 불가능하므로 Mote가 임의의 프로그램을 수행할 수 있도록 동적으로 지시할 수 있고, 프로그램도 간결한 방식으로 표현하도록 했다.

이를 위해 Mote들의 운영체제 위에 매우 간단하면서 어셈블리와 유사한 가상 머신을 동일하게 올려 전혀 다른 센서 노드 플랫폼 상에 동일한 응용 프로그램을 실행시키는 방법을 취하고 있다. VM 언어로 짜여진 최대 24개의 단위 명령어로 구성된 '캡슐'이라 불리는 프로그램이 노드에 올려져 하나의 태스크로 수행하게 된다. 캡슐도 SensorWare의 모바일 스크립트처럼 스스로 이동할 수 있다.

Maté는 극도로 제한적인 메모리의 사용 때문에 프로그래밍을 더 쉽고 효율적으로 만들 수 있는 몇 가지 특징이 제한돼 있다. 가상머신의 메모리 Footprint를 제어하기 위해 프로그래머는 8개의 기능 즉, 캡슐만을 사용할 수 있으므로, 이들 각각의 명령어가 강력하더라도 실현될 수 있는 응용 범위에는 큰 제한이 따른다. 그림 5는 Maté 시스템 구조를 보여준다.

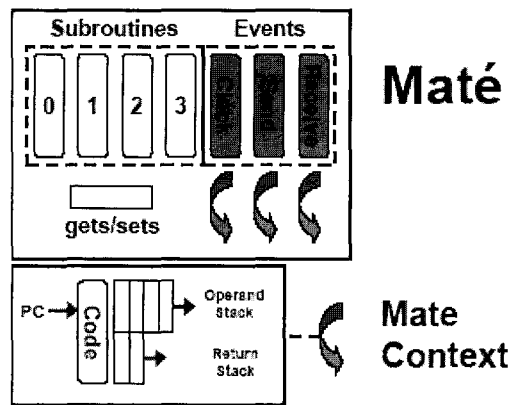


그림 5 Maté 시스템 구조

## 3.4 제한자원 보호 미들웨어

### 3.4.1 Impala

Impala(8)는 프린스턴 대학에서 ZebraNet 프로젝트의 일환으로 시작되었다. ZebraNet 프로젝트는 생물학과와 전자공학과의 협력해서 센서 네트워크 기술을 이용해 일

룩말과 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 어려운 환경에 배치되므로 소프트웨어의 자동 업데이트 등을 할 수 있는 기능이 지원돼야 한다. 이러한 ZebraNet의 특수한 요구사항을 미들웨어 레벨에서 해결하기 위한 시도가 Impala다.

Impala는 애플리케이션의 모듈화(modularity), 적응력(adaptivity), 복구성(repairability)에 초점을 맞췄다. Impala에서 소프트웨어 업데이트는 무선 전송을 통해 각 노드에 전달되고, 각 노드는 시스템이 동작하는 상태에서 업데이트를 수행할 수 있다. 또한, Impala는 실시간 애플리케이션의 적응을 위한 인터페이스를 제공함으로써 소프트웨어 시스템의 성능, 에너지 효율성, 안정성을 향상시키고, 다양한 파라미터와 디바이스 실패 등에 대한 동적 응용 적응성을 제공한다. 프린스턴 대학의 연구팀은 GPS를 가진 HP/Compaq iPAQ 포켓 PC에 Impala의 프로토타입 시스템을 구현했다. Impala는 계층적 시스템 구조를 가지고 있는데, 그림 6에서와 같이 응용 프로그램과 프로그램들은 상위 계층에 올라가고 그 밑의 계층에서 응용 어댑터, 응용 업데이트 등의 미들웨어 에이전트가 상위 계층을 지원한다.

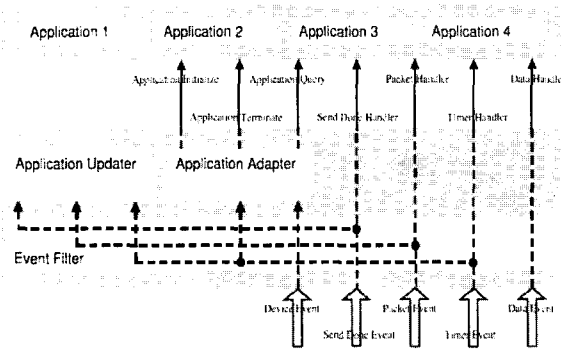


그림 6 Impala 기능구조

응용 어댑터는 센서 노드들의 현 내부 상태에 따라, 수행되어야 하는 응용 프로토콜을 결정하는 역할을 한다. 예를 들어, 프로토콜 1은 하나의 이웃 노드에게 데이터를 보내기 위한 단거리 라디오를 사용하고, 프로토콜 2는 여러 개의 노드에게 데이터를 보내기 위해 장거리 라디오를 사용할 수 있다. 이때 응용 어댑터가 각 노드의 현재 내부 상태를 파악해 적절한 응용 프로토콜을 결정할 수 있다.

응용 업데이트의 목적은 메모리, 에너지 등의 자원에 제약을 받는 센서 네트워크를 위한 효율적인 소프트웨어 갱신 메커니즘을 제공하는 것으로, Impala에서는 다음과

같은 메커니즘을 사용한다. 먼저 완전 업데이트와 불완전 업데이트를 모두 코드 메모리에 저장한다. 완전 업데이트는 실행을 위해 로딩되고, 불완전 업데이트는 지난 실행부터 이어서 실행할 수 있도록 로딩된다. Impala는 소프트웨어 전환을 위해 on-demand 전환방식을 사용한다. 센서 노드들은 실제 코드를 교환하기 전에 소프트웨어 버전 정보를 주기적으로 먼저 교환한다. 버전 정보의 교환주기는 모든 센서 노드가 최신의 업데이트를 가지고 있는가에 따라 자동으로 결정된다. 소프트웨어의 최신화는 이웃하는 노드의 소프트웨어 버전 번호를 확인해 새로운 버전으로 확인되면 해당 소프트웨어를 요구함으로써 peer-to-peer로 전파된다. 또한, 장치에 대한 응용 적응력을 제공하기 위해 응용 소프트웨어의 연혁도 사용한다.

### 3.4.2 MiLAN

MiLAN(9)은 로체스터 대학에서 건강 관리를 위한 스마트 메디컬 홈을 위한 센서 네트워크용 미들웨어로서 개발한 것으로 사용자가 요구하는 각종 의료용 데이터(ECG, 맥박수, 호흡횟수, 혈압, 혈류량, 산소량, 환자위치, 동작상태)에서 요구되는 신뢰도 요구를 만족시키면서 전체 소비에너지를 최소화하는 측정 계획을 계산하고, 그 계획에 따라서 해당 센서들을 활성화하며 측정된 데이터를 수집하고 사용자에게 전달하는 서비스를 제공한다. 그림 7은 센서 측정치들과 의료용 데이터 간의 신뢰도 관계를 보여준다.

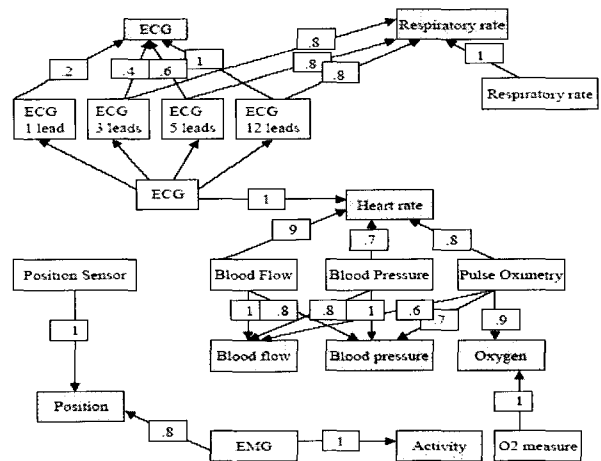


그림 7 센서 데이터와 의료용 데이터 간 신뢰도 관계

### 3.4.3 PADS

PADS (Power Aware Distributed Systems)는 UCLA/USC 대학에서 연구하고 있는 센서 네트워크 미들웨어로서 에너지 절감을 위하여 CPU, RF 모듈, 센서 모듈들 간의 에너지 관리 기능들을 유기적으로 관리하는 기능을 최적화하여 구현하고 있다. 이를 위하여 RTOS 스케줄 기법을 동원하여 정확한 시간에 동적으로

CPU전압과 RF 모듈레이션 스케일 등을 조절하여 최소한의 에너지로 프로그램 실행, 센서 측정 및 메시지 송수신이 이루어지도록 한다. 그림 8은 PADS 미들웨어 구조를 보여준다.

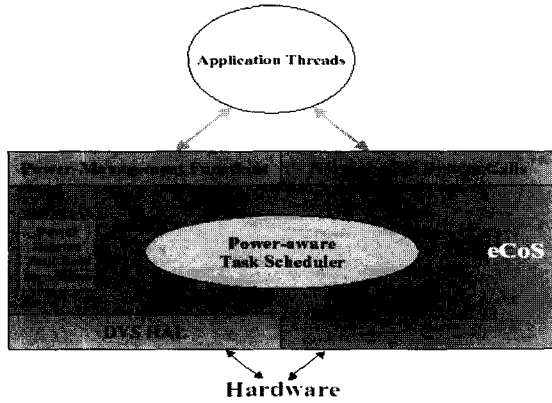


그림 8 PADS 미들웨어 구조

### 3.5 공통 센서기능 제공 미들웨어

#### 3.5.1 DARPA NEST 미들웨어

최근 수행되고 있는 DARPA NEST 프로젝트는 센서 네트워크에 적합한 미들웨어를 대상으로 한 연구를 수행하고 있는데 센서 네트워크가 가져야 할 공통 기능들을 묶어서 미들웨어를 구성해가고 있다. 예를 들면 벤더빌트 대학의 미들웨어는 현재 다음과 같은 9개 독립 기능들을 미들웨어 안에 구현하고 있다. 시간 동기화 프로토콜인 FTSP (Flooding Time Synchronization Protocol)은 리더를 선출하여 브로드캐스트를 하여 시간 동기화를 이루고 있으며 MICA2 플랫폼 상에서 백만분의 2초 미만의 시간 오차를 달성하고 있다고 한다. 한편 소리를 이용한 자동 노드 위치 인식 서비스를 제공하여 클랩 발견 서비스와 함께 저가 위치 추적 시스템의 구현을 지원하고 있고 그 외에도 스택 모니터, 타임 스탬핑, 시스템 분석 메시징 서비스 등이 미들웨어에 들어 있다.

#### 3.5.2 SCADDS

SCADDS(Scalable Coordination Architectures for Deeply Distributed Systems)는 남가주 대학에서 설계된 센서 네트워크 미들웨어로서 확장성을 가진 협동 구조로서 설계되었다. 현재 제공되고 있는 미들웨어 서비스로는 Direct Diffusion 라우팅, 적응적 신뢰도 유지기능, 노드 위치인식, 시간 동기, 에너지 절감 MAC, 통신 프로토콜 스택 등이 구현되어 있다.

## 4. 결론 및 향후 전망

본 논문에서는 센서 네트워크가 가지는 특징들에서

발생하는 미들웨어 요구사항 및 기능들을 제시하였고 이러한 사양 및 기능들을 구현하는데 적합한 미들웨어 모델 구조들을 제안하였다. 둘째로 제안된 미들웨어 모델 구조 프레임워크를 이용하여 현재 연구개발 중에 있는 대표적인 미들웨어들을 분류하였다.

본 논문에서 제안한 미들웨어 모델 구조들은 상호보완적인 관계를 가지고 있어서 다양한 미들웨어 기능들을 제공하여야 하는 경우 복수개의 모델 구조들을 구현한 복합적인 형태의 미들웨어가 필요하리라고 여겨지나 한편 저기능 센서 네트워크용 단말 노드들은 제한된 자원으로 인하여 복잡한 미들웨어를 설치하는데 어려움이 따른다. 따라서 향후 센서 네트워크 미들웨어는 고기능형과 저기능형으로 나뉘어 진행되리라 예상된다. 고기능형 복합구조 미들웨어는 주로 게이트웨이나 지역 관리담당 노드에 설치되어 각종 응용 소프트웨어에 다양한 서비스를 제공하고, 저기능형 단순구조 미들웨어는 간단한 응용분야에 적합한 저가형 단말 노드들에 설치되리라 여겨진다.

## 참고문헌

- [1] D. Y. Kim, "Sensor Networks(V) Sensor Network Middleware," FA Journal, Jul. 2004.
- [2] K. Romer, O. Kasten, and F. Mattern, "Middleware Challenges for Wireless Sensor Networks," ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 6, No. 4, October 2002.
- [3] Y. Yao and J. Gehrke, "The Cougar Approach to In Network Query Processing in Sensor Networks," SIGMOD Record, Vol. 31, No. 3, Sept. 2002.
- [4] C. Shen, C. Srisathapornphat, C. Jaikeo, "Sensor Information Networking Architecture and Applications," IEEE Personal Communications, Vol.8, No.4, pp. 52-59, Aug. 2001.
- [5] S. Li, S. Son, and J. Stankovic, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks," Intl Workshop on Information Processing in Sensor Networks (IPSN'03), Palo Alto, CA, Apr. 2003.
- [6] A. Boulis, C.C. Han, and M. B. Srivastava, "Design and Implementation of a Framework for Programmable and Efficient Sensor Networks,"

- In The First International Conference on Mobile Systems, Applications, and Services (MobiSys), San Francisco, CA, 2003.
- [7] P. Levis and D. Culler, "Mate : A Virtual Machine for Tiny Networked Sensors," Proc. of ACM Conf. Architectural Support for Programming Languages and Operating Systems, San Jose, CA, Oct. 2002.
- [8] T. Liu and M. Martonosi, "Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems," Proc. of ACM SIGPLAN symposium on Principles and practice of parallel programming, pp.107-118, 2003.
- [9] W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, and M. A. Perillo, "Middleware to Support Sensor Network Applications," IEEE Network, Vol.18, No.1, pp.6-14, Jan. 2004.

---

김 영 만



1988 Ohio State University, 전산과  
학과 석사  
1992 Ohio State University, 전산과  
학과 박사  
현재 국민대학교 컴퓨터학부 교수  
관심분야 : Ad Hoc 네트워크, 분산 미들  
웨어, 센서 네트워크, 소프트웨어  
스트리밍  
E-mail : ymkim@kookmin.ac.kr

---

• 제17회 영상처리 및 이해에 관한 워크샵 •

- 일 자 : 2005년 1월 19일
- 장 소 : 제주도 라마다플라자 호텔
- 주 최 : 컴퓨터비전및패턴인식연구회
- 내 용 : 논문발표 등
- 상세안내 : <http://ipis.cau.ac.kr//ipiu2005/>