특집

센서 네트워크 미들웨어 기술

김대영, 성종우, 송형주, 김수현 (한국정보통신대학교)

I. 서 론

유무선 통신 기술의 발전 및 모바일 정보기기의 보편화와 함께 새롭고 다양한 서비스를 제공하는 유비쿼터스 컴퓨팅의 핵심 기술의 하나로써 외부 환경의 감지와 제어 기능을 수행하는 센서 네트워크 (Sensor Network) 기술이 각광받고 있다. 센서 네트워크의 센서 노드는 다양한 실세계 정보를 센싱 할 수 있는 센서와 이를 처리할 수 있는 최소한의 프로세싱 리소스 그리고 다른 센서 노드와의 협동적인 데이터 처리를 위한 무선 네트워크 기능을 포함하고 있다. 이들 센서 네트워크는 기존의 컴퓨팅 환경과는 구별되는 제약적인 여러 특징을 가지며 이러한 특징은 센서 네트워크 응용의 개발을 더욱 어렵게 하는 원인이 되고 있다. 이러한 센서 네트워크의 하드웨어 리소스 제한을 극복하고 개발의 복잡성을 피하기 위해서 상위 응용계층에서 필요로 하는 기능을 추상화하여 제공하는 센서 네트워크 미들웨어에 대한 많은 연구가 이루어지고 있다. 센서 네트워크 미들웨어는 다양한 센서 응용 소프트웨어와 운영체제 및 네트워크 스택 사이에 존재하며, 이종 센서 노드 기반의 응용 시스템 개발, 유지보수, 설치, 수행에 필요한 제반 사항을 지원한다. 주요 미들웨어 기능으로는 무선을 통한 원격 재프로그래밍, 상황인지 미들웨어, 센서 데이터베이스 등이 있다. 본 논문에서는 센서 네트워크 미들웨어의 주요 특징과 설계 기술을 분석하고 주요 이슈가 되는 기술들을 분석해 본다. 그리고 DSWare[3], MiLAN[6], TinyLime[7], BOSS[8]와 같이 현재 활발히 연구되고 있는 대표적인 센서 네트워크 미들웨어를 살펴보기로 한다.

II. 무선 센서 네트워크 미들웨어

1. 센서 네트워크 미들웨어 소개

지난 수년간 폭발적으로 증가한 센서 네트워크에 대한 관심으로 센서 네트워크의 운영체제와 하드웨어 구성 그리고 네트워크에 대한 많은 연구가 경쟁적으로 수행되었다. 그러나 아직까지 센서 네트워크의 응용 개발은 운영체제나 센서 노드에서 제공되는 매우 저

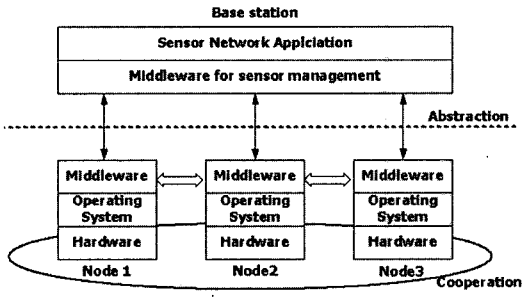
수준의 기능을 이용하여 이루어지고 있으며 센서 네트워크의 개발과정은 아직도 특정 응용을 개발 초기부터 고려하여 이에 의존적인 형태로 진행되어 왔다. 이러한 하위레벨 종속적인 센서 네트워크 응용 개발은 무선 센서 네트워크의 개발과 배포 등을 어렵게 하고 센서 네트워크 제작에 대한 전체적인 비용을 증가시켜서 다양한 응용에 센서 네트워크를 적용하려는 산업계의 요구를 충족시키지 못하고 있다.

이와 함께 센서 네트워크에서의 새로운 경향은 멀티 프로그램이 가능한 센서 네트워크를 구현하는 것이다. 과거에는 센서 하드웨어와 운영체제에 매우 의존적이며 특정 응용에 종속적인 형태로 센서 네트워크의 기능이나 기술적 적용 가능성을 보여주는 것에 초점을 맞췄다면 앞으로의 추세는 다양한 응용분야를 지원할 수 있으며 사용자와 상황의 요구에 따라 동적으로 적응할 수 있는 유연한 구조로 다양한 어플리케이션을 지원하는 센서 네트워크를 개발하는 것이 필요하다. 이러한 유연한 구조의 센서 네트워크 개발의 일환으로써 센서 네트워크 운영체제 레벨에서의 소프트웨어 진화와 동적인 소프트웨어 업로딩에 대한 연구가 수행되었으며 최근에는 센싱 데이터 추상화를 통해 다양한 응용개발을 지원할 수 있는 미들웨어에 특히 많은 연구가 수행되고 있다.

추상화는 센서 네트워크를 위해서 필수적인 데이터의 수집, 이벤트 처리 매커니즘, 전력 관리 그리고 네트워킹 등 센서 네트워크의 필수적인 기능을 포함하며 최종적으로는 센싱 결과들이 마치 하나의 블랙박스와 같이 인식될 수 있도록 해서 다양한 응용을 쉽게 개발할 수 있도록 도와준다. 이러한 미들웨어 개념의 도입은 센서 네트워크 응용이 개발되고 배포되고 운영되는 전체 흐름의 변화를 의미한다. 예를 들어, 스마트 홈과 같은 특정 환경을 위한 센서 네트워크는 이러한 추상화를 제공하는 범용 센서 네트워크를 사용해서 개발하는 경우, 응용 개발은 개개의 노드에 대한 관리와 프로그래밍이 아닌 고수준의 시스템 추상화를 통한 절차적인 쿼리 언어 등을 이용할 수 있으며 미들웨어는 센서 네트워크의 다양한 기능을 상위레벨에게 제공해서 마치 센서 네트워크 전체가 하나의 데이터베이스나 스프레드시트와 같이 인식될 수 있도록 한다.

인터넷과 같은 대형 분산 시스템과 개인용 컴퓨터를 통틀어서 사용되는 광범위한 미들웨어의 종류를 생각해 볼 때 센서 네트워크 미들웨어를 명확하게 정의하는 것은 쉽지 않다. 하지만 기존의 여러 연구를 통해서 다양하게 정립된 미들웨어 개념을 고려하면 미들웨어란 커널 레벨이 아닌 응용 영역에서 사용자 또는 상위 응용에게 인터페이스 형태의 다양한 기능과 하위 레벨의 추상화를 제공해주는 소프트웨어라고 요약될 수 있다. 센서 네트워크 미들웨어는 기존의 미들웨어 개념과 마찬가지로 기본적으로 다양한 센서 노드로 구성된 하드웨어 계층과 운영체제 상위에서 존재하며, 응용 소프트웨어에 추상화된 인터페이스를 제공하는 역할을 한다. 하지만 이러한 기본적인 센서 네트워크 미들웨어의 정의에도 불구하고 기존의 미들웨어를 센서 네트워크에 그대로 적용시킬 수는 없다. 센서 네트워크는 일반적인 데스크탑 환경이나 모바일, 분산 환경과는 또 다른 특징을 가지

며 센서 네트워크의 특성상 미들웨어, 센서 노드 운영체제, 네트워크 또는 상위 응용이 더욱 밀접하게 연관되어 각 기능을 명확하게 분리하기가 쉽지 않기 때문이다. 따라서 센서 네트워크 미들웨어는 이러한 센서 네트워크의 특징을 만족시킬 수 있도록 설계되어야 한다. 그림1은 센서 네트워크에서 미들웨어의 역할과 위치를 나타낸다.



〈그림 1〉 미들웨어 관점에서의 센서 네트워크 구조

2. 센서 네트워크 미들웨어 특징

센서 네트워크는 제한된 컴퓨팅 능력과 에너지 그리고 낮은 통신 대역폭뿐만 아니라, 그 특성상 설치되는 환경의 영향을 많이 받으며, 노드 위치의 변화, 센서 네트워크 일부의 유실등 센서 네트워크의 전체 혹은 부분이 동적인 변화를 겪기 쉽다. 심지어는 전력의 완전 소모나 환경의 영향으로 노드가 동작하지 못하더라도 해당 노드를 수리할 수가 없는 경우가 생길 수 있기 때문에 센서 네트워크 미들웨어는 이러한 점을 최대한 고려해서 설계 되어야 한다.

센서 네트워크 미들웨어에 있어서 가장 큰 영향을 주게 되는 센서 네트워크 서비스의 토폴로지는 중앙 집중적인 방법이 주로 쓰였다. 중앙 집중적인 서비스 토폴로지를 기반으로 하는 네트워크는 센싱된 정보를 중앙의 베이스 스테이션으로 집중시키고 베이스 스테이션에서 복잡하고 지능적인 태스크를 수행하도록 한다. 이러한 중앙 집중 방식은 단일 베이스 스테이션의 결함 등에 대처할 수 없고 효율적인 에너지 사용이 어려우며 대형 네트워크 구조에 적합하지 않다. 이와는 반대로 각각의 센서 노드들이 근접한 센서 노드들과 협력을 하는 분산된 방식을 사용한다면, 센서 네트워크 규모가 커짐에 따라 네트워크 파티션(network partition)과 특정 노드의 에러(failure)등이 발생함에도 효과적으로 대처할 수 있게 된다.

아래에 센서 네트워크 미들웨어 개발에서 고려되어할 주요 특징들을 논하였다.

●자동 설정

일반적으로 센서 네트워크가 이용되는 분야는 위험하거나 사용자가 쉽게 접근할 수 없는 지역인 경우가 많다. 또한 경우에 따라서 노드의 수가 수백 또는 수천 개가 될 수 있으므로 사용자가 이러한 노드들을 일일이 관리하고 업데이트하며 설정하는 것은 불가능하다. 따라서 개개의 센서 노드들은 사용자의 개입 없이 자동으로 동작할 수 있어야 하며 이것은 네트워크 설정과 전력 관리, 데이터 수집 등이 주위 상황에 맞도록 자동으로 설정이 되어야 한다는 것을 의미한다.

●지역적인 알고리즘

네트워크 전체에 대한 응용 목적이 일정 이웃에 속하는 노드들만의 통신에 의해서 달성되는 분산된 알고리즘을 바탕으로 한다.

이는 증가하는 네트워크 크기에 따른 확장과 네트워크의 분할이나 노드의 장애도 쉽게 극복할 수 있게 한다.

●협동 작업의 가능

각각의 노드들이 비록 개별적으로 설정되고 동작할 수 있다고 하더라도 개별 노드들의 제한된 성능으로 인해서 실제 센서 네트워크 응용에서는 노드들이 협동하여 전체적으로 지정된 임무를 수행하거나 범용적인 기능을 수행할 수 있다.

●품질 조절이 가능한 알고리즘

센서 네트워크에서 센싱된 정보를 처리하는 센서 노드들의 신뢰성은 제한된 기능을 가지는 센서와 네트워크의 불확실성등으로 인해 개개의 노드들에 의해서 제공되는 센싱 값들을 전적으로 신뢰할 수 없는 문제가 있다. 따라서 주어진 문제에 대한 답의 품질이 제한된 자원의 활용을 조절하여 제공될 수 있도록 대표 변수들이 지정되거나 조절을 가능하게 하는 알고리즘이 광범위하게 제공되어야 한다.

●데이터 중심 통신

센서 노드들에 의하여 만들어지는 데이터가 통신의 주체가 되는 새로운 방식의 통신이다. 예를 들어 온도 센싱에 대한 응용의 경우, 기존의 통신에서와 같이 주소에 의해 명시된 특정 노드들에 의해 감지되는 온도 데이터를 단순히 요구하고 수신하는 통신이 아니라, 특정 온도 값이 메시지에 포함되어 전달되어 이 값을 넘어서는 지역의 위치 정보를 발생시키도록 노드들에게 요구할 수 있다. 이처럼 속성 기반의 통신이 기존 통신의 요구-응답 방식보다 센서 네트워크의 특성에 적합하기 때문에 데이터 중심으로 처리하는 기능이 센서 네트워크의 미들웨어에서 통합적으로 제공되어야 한다.

●이벤트 중심 설계

센서 네트워크의 데이터 중심 통신과 센서 노드들의 제한된 리소스 문제를 고려할 때 기존의 클라이언트/서버 기반의 polling이나 request/response 모델이 효율적으로 사용되기 어렵다. 센서 네트워크에서는 각각의 센싱 값들이 읽혀지는 경우 적절한 형태의 이벤트가 발생되게 되고 베이스 스테이션은 필요로 하는 이벤트에 대해서 미리 등록하여서 이벤트를 중심으로 데이터를 처리하게 되는 publish/subscribe가 널리 사용된다.

●정보에 기반한 노드 설계 및 그 운용

기존의 네트워크에서는 응용에 대한 정보를 반영하지 않고 다양한 영역의 응용을 지원할 수 있도록 만능인 미들웨어의 설계가 요구되었지만, 센서 네트워크에서는 이러한 정보를 네트워크 기반 구조나 노드 자체에 주입할 수 있는 방법까지 미들웨어에서 제공하도록 요구한다. 예를 들어, 중간 노드에서는 응용에 맞추어 데이터를 캐싱하고 다수의 데이터 통합을 수행하여 자원이나 에너지 효율성을 효과적으로 향상시킬 수 있도록 한다.

●기존 네트워크나 장치들과의 연계된 동작

센서 네트워크는 자원이 제한된 특성으로 인하여 자원 집약적인 기능이나 크기가 큰 정보의 저장은 센서 네트워크 외부의 다른

네트워크나 장치에서 수행되는 것이 적합하다. 예로서, 인터넷에 내재하여 이에 대응하는 가상의 기능 구조 또는 구성요소가 이러한 기능을 대신 수행할 수 있다. 기존 네트워크의 메커니즘이나 기반 구조가 센서 네트워크와의 연계에는 적합하지는 않아서 비록 이를 지원하는 구조 설계와 구현이 어려운 것이 사실이지만, 센서 네트워크의 미들웨어는 기존 네트워크의 역량을 센서 네트워크에서 연계하여 활용, 접목시킬 수 있는 역할을 수행하여야 할 것이다.

3. 센서 네트워크 미들웨어 기능별 분류

센서 네트워크를 위한 미들웨어는 기본적으로 네트워크 요소의 개발, 센서 노드의 배포, 유지 및 관리, 감지 기반 응용의 수행을 원활하게 함을 목적으로 한다. 응용이 요구한 문제에 적절한 답을 제공하기 위하여 미들웨어는 통신의 흐름에 따라 표면적으로는 상위 레벨 감지 태스크의 형성, 태스크와 무선 센서 네트워크간의 통신, 형성된 태스크를 노드 각각의 역량에 맞게 분할하고 각 노드에 배포하기 위한 상호 협력, 감지된 데이터를 조합하여 상위 레벨에 적합한 정보로 만들기 위한 데이터 융합(data fusion), 최종적으로 태스크 요청자에게 결정된 데이터의 보고 등과 같은 기능을 제공해야 한다.

한편, 센서 네트워크는 특징적으로 에너지 소모의 효율성, 동적인 환경에서 네트워크의 건실성, 다수의 노드 구성에 의한 확장성을 요구한다. 따라서, 네트워크의 효과적인 형성, 유지, 관리 등에 대한 미들웨어의 기능도 다음과 같이 요구된다.

●실시간성 지원

자동차의 센서 시스템과 같은 실시간 응용에서는 센싱 데이터의 획득과 프로세싱 그리고 적절한 액션의 제어가 실시간으로 이루어져야 하는 않은 경우가 많다. 그러나 수백 또는 수천의 센서 네트워크가 단일한 목적을 위해서 함께 네트워크상으로 연결되어서 동작할 때 실시간성을 보장하는 것은 매우 어려운 문제이며 이를 해결하기 위한 많은 연구가 수행되고 있다

●센서 네트워크 네이밍

대규모 센서 네트워크에서는 센서 각각의 노드들보다는 전체 센서 네트워크에서의 데이터가 중요하며 개개의 센싱 값들은 관심밖에 있거나 실제로 개개의 센서 노드를 관리하기가 어려운 경우가 많다. 이러한 경우 개개의 센서 노드들에 대한 네이밍을 제공하는 대신 센서의 속성과 기능에 대해서 센서 네트워크 상의 노드들을 네이밍하는 방법이 사용될 수 있다.

●서비스 디스커버리

인터넷 기반의 웹 서비스를 가능하게 한 것이 구글과 야후와 같은 웹 검색 엔진이었다면 유비쿼터스 센서 네트워크에서는 센서 네트워크가 제공하는 서비스에 대한 디스커버리가 필수적이다. 사용자가 센서 네트워크에 연결되고 제어하기 위해서는 특정 센서 네트워크를 검색할 수 있어야 하며 센서 네트워크에서의 제어를 위해서는 특정 액츄에이터와 센서등을 이용할 수 있어야 한다. 이 과정에서 사용자의 컨텍스트와 찾으려는 센서 네트워크의 속성 등이 고려되어야 한다.

●기존 네트워크와의 연동

센서 네트워크를 실제 적용하기 위해서는 기존 IP 기반의 인터넷과의 연동이 필수적이다. 하지만 센서 노드들간의 통신이 비IP 기반 네트워크를 이용하므로 센서 네트워크와 인터넷 기반의 네트워크를 연결하기 위해서는 게이트웨이 구조를 이용하거나 오버레이 기반의 네트워크 지원이 필요하다.

●전력 관리

센서 네트워크의 노드들은 주로 배터리로부터 전원을 공급받기 때문에 에너지가 매우 제한적이다. 에너지를 효율적으로 소모하고, 네트워크의 생존 시간을 최대화 시킬 수 있는 전력 관리 미들웨어가 필요하다.

●위치 인식

배치된 센서 노드들에 의한 메시지 전달의 최적 경로를 결정하는 라우팅을 위해서도 노드의 위치인식기능이 필요하며, 센서에서 감지된 센서 데이터의 단순전달뿐만 아니라 응용에서 필요로 하는 컨텍스트를 위해서도 위치 정보는 매우 유용하게 이용된다. 단순히 하위 계층에서 위치 인식 알고리즘의 구현뿐만 아니라, 다양한 환경과 위치 인식 센서들의 퓨전을 위한 위치 인식 미들웨어의 기능이 요구된다.

●소프트웨어의 배포 및 자동 갱신

이동성이 있거나 지리적으로 널리 분산되어 있는 많은 노드 에게 코드를 전송하기 위한 메커니즘을 제공하여야 한다. 이는 다수의 센서 노드로 구성되는 센서 네트워크의 특성 또는 그 배포 방법 때문이며 최신 소프트웨어로의 자동 갱신이 고려되어야 한다.

●센서 정보 관리

센서 네트워크를 구성하는 다수의 센서 노드들로부터 감지되는 센싱 정보들을 효율적으로 관리하는 기법이 필요하다.

●데이터의 분배 및 복제

센싱 데이터의 접근성을 높이고, 데이터 전송에 의한 과도한 에너지 소모를 줄이기 위해서, 센싱된 데이터를 최적의 위치에 저장하고, 필요시 복사본을 데이터 접근이 용이한 곳에 저장하는 기법등이 필요하다.

●센서 네트워크의 보안

초소형 노드는 대부분 메모리 관리부가 없기 마련이므로 신뢰할 수 없는 의심스러운 코드가 수행되지 않도록 미들웨어 플랫폼에는 추가적인 보안기능이 요구된다.

●장애 관리

네트워크의 신뢰도와 가용성이 센서 네트워크가 사용하는 무선의 환경보다 더 낮기 때문에 통신의 실패는 훨씬 자주 발생한다. 더구나, 전력소모에 대한 제약 때문에 자원이 가용적이지 않을 수도 있다.

III. 센서 네트워크 미들웨어 연구 동향

대표적인 센서네트워크 미들웨어로는 Cornell대학의 Cougar[5], Delaware 대학의 SINA[1], Rochester 대학의 MiLAN[6], Virginia 대학의 DSWare[3], UCLA의 SensorWare[9], 프린스턴 대학의 Impala[4], UCB의 Mate[2] 등이

있으며, 그 외에 Sentire[10], TinyLime[7], ICU의 BOSS[8] 등이 있다.

Cougar[5]와 SINA[1]는 센서 네트워크를 하나의 분산된 데이터베이스로 정의하고, 쿼리 형식을 사용하여 센서 네트워크의 정보에 접근한다. Cougar[5]에서는 데이터를 수집하고 쿼리 결과의 계산에 소모되는 에너지를 작게 하도록 쿼리를 센서 노드에 분산시킴으로써 전력 소모를 관리한다. SINA[1]에서는 지리적으로 근접하게 분포하는 센서 노드들로부터 유사한 정보의 재전송을 제한하는 방법뿐만 아니라 효율적인 데이터 융합을 위하여 센서들의 계층적인 그룹화를 수행하는 하위 레벨의 메카니즘도 포함한다. 또한 MiLAN[6] 미들웨어는 운영체제와 어플리케이션 사이에 존재하는 기존의 미들웨어와는 달리 네트워크 프로토콜 스택으로 확장될 수 있는 구조를 특징으로 한다.

Bombilla[2]는 TinyOS 위에 수행되는 초경량의 통신 중심 가상 머신 형태의 미들웨어로 스스로의 bytecode 해독기를 구비하며, 에너지 효율적인 방식으로 센서 네트워크의 재프로그래밍을 제공하고자 구현되었다. Bombilla[2]는 전염모델(infection model)을 통하여 새로운 코드를 배포시키는 메카니즘을 가지고 있다. 따라서, 프로그래머가 각 센서에 대한 새로운 코딩을 염려할 필요 없이 한 노드로 전달된 코드가 바이러스와 같은 형식으로 네트워크로 퍼져나가는 특징을 지닌다.

SensorWare[9]는 개별적인 노드들의 동작 제어를 위한 태스크 프로그램이 이동성을 가지며 프로시저에 기반하는 프로그래밍 언어인 Tcl로 정의된다. SensorWare에서는 임의적인 분산 알고리즘이 구현 가능하므로, 특정 감지용 태스크를 구현하기 위하여 run-time환경을 바꿀 필요가 없다는 장점을 가진다.

DSWare[3]는 UCI의 AutoSec 과 유사하게 일정 패턴으로 정의되는 복합된 이벤트를 기본 프로그래밍의 Abstraction으로 제공함으로써 데이터 서비스의 Abstraction을 수행하는데, 이벤트의 보고에 대한 데드라인과 확실한 이벤트에 대한 구간 등의 정의와 같이 실시간성을 다양하게 제공하는 특징을 지닌다.

Impala[4]는 프린스턴 대학에서 동물들의 생태를 센서 네트워크 기술을 이용하여 연구하기 위한 ZebraNet 프로젝트의 일환으로 시작되었다. 생태 관찰 연구의 특성상 센서 네트워크의 에너지 소비를 최대한 효율적으로 고려하여 수명을 최대화시켜야 하고, 노드들이 사람이 관리하기 어려운 생태 환경에 배치되므로 자동적으로 소프트웨어를 최신 것으로 바꿀 수 있는 기능이 요구된다. Impala는 이미 동작 중인 소프트웨어의 간단한 오류 수정으로부터 완전한 응용 전체의 교환에 이르기까지 광범한 영역의 소프트웨어 최신화 기능을 노드의 운행 중에도 가능하도록 시도하였다.

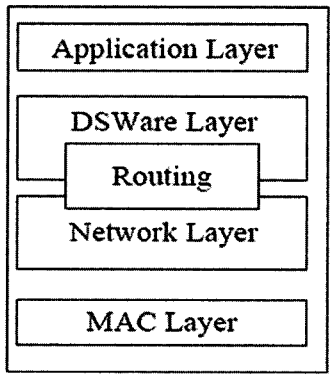
Sentire[10]는 확장 가능한 컴포넌트 기반의 센서 네트워크 프레임워크로 각각의 컴포넌트는 상호 통신이 가능하며 플러그인 형태의 인트라스트럭처를 제공한다. Sentire[10]는 크게 매니저(manager)와 메시지(message)로 구성되며 인터페이스 매니저, 데이터 매니저, 리소스 매니저, 센서/액추에이터 매니저를 포함하고 이들끼리의 통신을 위한 메시지를 정의하고 있다.

끝으로 센서 네트워크를 이질적인 기존 네트워크 환경에 통합하기 위한 연구도 진행되고 있는데, 예를 들어 센서 네트워크 서비스를 UPnP(Universal Plug and Play)[11]를 통해 제공하는 것이 그 예이다. ICU에서는 BOSS[8] 구조를 통해서 베이스 노드에서 UPnP[11]의 모든 기능 (addressing, discovery, description, control, eventing 그리고 presentation)을 이용할 수 있도록 서비스, 컨트롤 그리고 이벤트 매니저를 구현한 BOSS[8] 프레임워크를 구현하였다. 다음에 대표적인 센서 네트워크 미들웨어를 소개한다.

1. DSWare (Data Service Middleware)

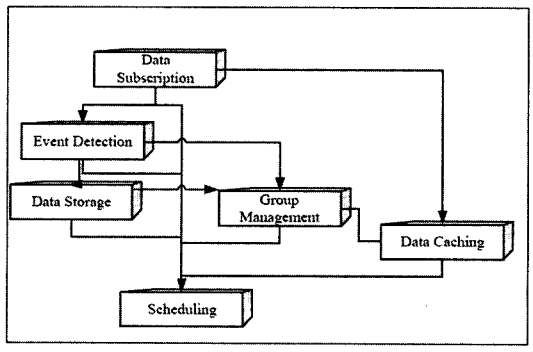
Virginia 대학의 DSWare는 다양한 센서 네트워크에 실시간 데이터 서비스를 통합하여 제공하기 위한 데이터 서비스 미들웨어이다. 일반적으로 센서 네트워크 응용은 이벤트 서브스크립션 작성, 이벤트 탐지, 데이터저장, 노드와 노드그룹 관리, 스케줄링등의 여러 서비스를 지원해야 하지만 기존의 센서 네트워크는 응용에 의존적인 방법으로 제한적인 데이터 서비스만을 제공하는 문제점이 있었다. DSWare는 여러 센서 응용에서 공통적으로 필요로 하는 서비스를 추상화하여 데이터 서비스로 제공하므로 응용은 복잡한 저수준의 센서 네트워크 오퍼레이션과 상관없이 표준 SQL을 사용하여 센서 네트워크로부터 원하는 데이터를 얻을 수 있다.

그림 2의 센서 네트워크 소프트웨어 구조에서 DSWare layer는 Network Layer와 Application Layer사이에 위치하며 DSWare Layer를 통해서 어플리케이션이 필요로 하는 공통적인 데이터서비스를 단일화하여 제공한다. 센서 네트워크 기능 요소 중 Routing은 DSWare의 그룹관리 컴포넌트와 스케줄링 컴포넌트를 통하여 배터리인지와 실시간 스케줄링을 사용하기 때문에 DSWare Layer와 Network Layer사이에서 공통적으로 존재한다.



〈그림 2〉 DSWare 계층을 포함하는 소프트웨어 구조

그림 3과 같이 DSWare 프레임워크는 데이터 서브스크립션 컴포넌트, 이벤트탐지 컴포넌트, 데이터 스토리지 컴포넌트, 데이터 캐싱 컴포넌트, 그룹관리 컴포넌트, 스케줄링 컴포넌트로 구성되어 있다. 다음은 각각의 컴포넌트에 대한 설명이다.



〈그림 3〉 DSWare 프레임워크

●데이터 서브스크립션(Data Subscription) 컴포넌트는 데이터를 획득하는 경로와 안정된 트래픽을 지원하는 정보를 제공하는 쿼리를 생성한다. 예를 들면, 경찰의 PDA(베이스 스테이션)는 센서네트워크에게 "지금부터 2시간동안 3분마다 Ivy도로와 Alderman도로를 가로지르는 트래픽 상태를 보여줘"라고 정보를 요청하면, 이에 따라 베이스 스테이션은 "노드 A, 지속시간 D(duration), 주기 R(1 per 3 min.)"같은 서브스크립션을 생성한다.

●이벤트 탐지(Event Detection) 컴포넌트는 표준 SQL형식을 사용하여 이벤트를 등록하고 취소하며 애플리케이션의 코드를 그대로 사용하여 이벤트를 데이터베이스나 센서네트워크에 등록한다.

●데이터 스토리지(Data Storage) 컴포넌트는 데이터를 통합하고 로컬 프로세싱을 수행한다. 단일 식별자를 가진 논리적인 스토리지 노드는 여러 개의 물리적인 노드로 매핑하여 서브스크립션 쿼리를 분배하고 이중에서 한 노드가 데이터 리포트를 베이스 스테이션으로 전송하게 된다. 각 노드에 저장되어 있는 데이터는 손실될 수 있기 때문에, 여러 물리적인 노드에 데이터를 중복하여 저장한다.

●데이터 캐싱(Data Caching) 컴포넌트는 가장 많이 요청되는 데이터를 여러 개의 데이터 복사본으로 만들어 노드들에게 제공한다. 또한 자주 쿼리를 생성하는 노드에게 데이터 복사본을 전달할지를 결정한다. 이웃 노드들과 통신하면서 데이터 복사본의 사용을 모니터링하고 데이터 복사본의 수를 조절한다.

●그룹관리(Group Management) 컴포넌트는 데이터를 통합하기 위해 노드들을 결합한다. 신뢰성 있는 데이터를 획득하고 에러가 포함된 데이터의 전달을 미연에 방지하며, 노드의 에너지관리를 담당한다.

●스케줄링(Scheduling) 컴포넌트는 모든 컴포넌트를 위한 실시간 스케줄링 서비스와 에너지 인지 스케줄링을 제공한다.

일반적으로 센서 네트워크 응용을 개발하기 위해서는 공통적인 서비스들에 대한 고려없이 각 응용마다 전체 소프트웨어를 제각기 구현해야 하지만 DSWare는 컴포넌트를 사용해서 센서 네트워크의 데이터를 추상화하여 응용에게 제공한다. 따라서 DSWare는 응용에서 저수준의 태스크에 상관없이 데이터베이스의 형태로 센서 노드의 데이터를 이용할 수 있게 한다.

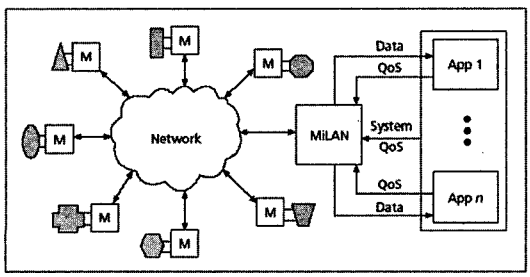
2. MiLAN (Middleware Linking Applications and Networks)

Rochester 대학의 MiLAN은 센서 네트워크 미들웨어 레벨에서 제공하는 기능과 응용의 다양한 기능적 요구사항을 연결하기 위해서 디자인 되었다. 센서네트워크 응용은 다양한 용도에 따라서 항상 다른 기능적 요구 사항을 가진다. 예를 들면 특정 환경을 감시하는 응용의 경우, 각 센서의 커버리지가 중복되

는 것을 최소화함으로써 불필요한 전력 소모를 줄이고, 특정 이벤트가 발생했을 경우, 더 많은 센서들에게 데이터를 보내게 함으로써 좀더 정확한 측정을 할 수 있어야 한다. 또한 헬스 모니터링 응용의 경우 환자의 상태에 따라 센싱 빈도를 조절함에 따라서 센서 노드의 에너지를 효율적으로 관리해야 한다.

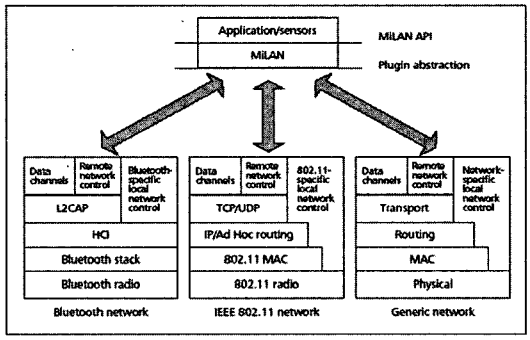
MiLAN에서는 어느 센서들이 센싱된 데이터를 보낼 것인지, 각각의 센서들은 어떤 라우팅 기법으로 통신을 할 것인지 등의 내용을 사전에 응용 에서 기술하면 응용이 원하는 서비스 품질(Quality of Service, QoS)을 만족시키기 위해, 동적으로 네트워크를 구성한다.

이를 위해서, MiLAN은 응용, 시스템 그리고 센서 네트워크로부터 발생하는 3가지의 정보를 사용한다. 우선은 응용으로부터 원하는 QoS를 기술한 정보와 해당 요구 사항을 만족시키기 위해 어느 센서들을 이용해야 할지에 대한 정보를 얻는다. 또한 미들웨어에서 동작하는 응용들의 중요도 및 상호 작용을 기술한 정보를 시스템으로부터 얻는다. 마지막으로 센서네트워크로부터 센서들의 에너지 및 채널 대역폭등의 자원(Resource)에 관련된 정보를 얻는다.



〈그림 4〉 MiLAN 상위 레벨 다이어그램

그림 4는 MiLAN을 사용하는 네트워크 구조로서, MiLAN은 응용이 보낸 요구 사항을 받은 다음, 각각의 센서네트워크의 상황을 모니터링 하면서, 요구 사항을 만족시키기 위해 동적으로 센서네트워크의 구성을 최적화 한다. MiLAN은 응용과 운영체제 사이에 위치하는 기존 미들웨어와 달리, 네트워크 프로토콜 스택을 확장한 구조를 가진다.



〈그림 5〉 MiLAN 컴포넌트

그림 5에서 볼 수 있듯이 MiLAN은 응용 어플리케이션이 필요한 기능적 요구 사항을 나타내기 위해 사용하는 API와 MiLAN의 명령을 특정 네트워크 프로토콜의 명령으로 변경해서 여러 가지의 물리 네트워크를 연결해줄 수 있는 추상화 계층으로 구성된다.

3. TinyLime (Bridging Mobile and Sensor Networks through Middleware)

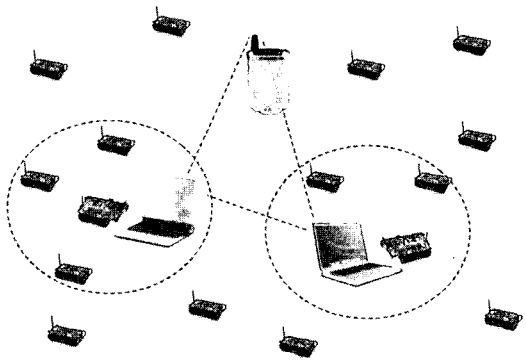
University of Lugano에서는 Lime(모바일 환경에서 tuplespace를 지원하기 위한 확장된 형태의 Linda)의 기능을 센서 네트워크에 적용시킨 TinyLime을 연구하였다. TinyLime은 모바일 환경의 응용과 센서 사이의 데이터 공유를 지원하기 위해 tuplespace라는 공

유 메모리를 이용한다. TinyLime은 모바일 환경의 사용자가 PDA 등의 단말을 이용해서, 주위 센서로부터 필요한 정보를 받는 응용 프로그램의 경우, 기존 센서 네트워크에서의 중앙 집중적인 센서 데이터 수집 방식을 이용하는 것은 비효율적이라는 점에 주목하였다.

이를 위해서는 사용자와 센서의 위치를 알아야 하고 해당 위치 정보를 통해, 사용자 근처의 센서로부터 데이터를 직접 받는 것이 아닌, 센서 데이터가 모이는 일종의 베이스 노드를 통해서 받아야 하기 때문이다. 이는 사용자 주위의 센서들이 자신의 센싱 데이터를 불필요하게 베이스 노드로 전달하는 것을 필요로 한다는 것이다.

따라서 TinyLime 에서는 기존의 중앙 집중적인 데이터 수집 방식이 아닌, 그림 6과 같은 모바일 환경의 클라이언트 및 베이스 스테이션(Base Station)과 센서 네트워크를 연결하는 새로운 센서 네트워크의 구성을 제안하였다. 여기서 클라이언트(PDA)와 베이스 스테이션(노트북)들은 서로 애드혹 방식으로 연결되고 각각의 베이스 스테이션들은 자신이 직접 연결되어 있는 센서로부터 값을 얻을 수 있으며 상호 데이터를 주고 받을 수 있다. TineLime에서는 클라이언트(PDA)가 이동하며 데이터를 수집하기 때문에 정보 수집을 위해서 센서들 사이에서의 멀티홉 라우팅에 의존하지 않는 특징을 가진다.

TinyLime은 이러한 구조를 위해, Lime (Linda in a Mobile Environment)이라는 모바일 환경에서의 데이터 공유를 지원하는 미들웨어를 확장시켰다. Lime은 Linda라는 tuple space를 통한 공유 메모리 모델을 모바

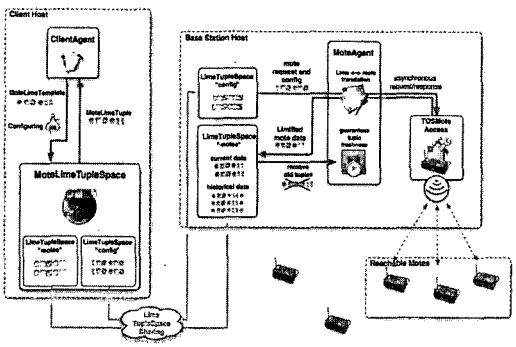


〈그림 6〉 센서 네트워크 구성

일 환경에 적용하기 위해 확장한 것이다. Lime에서는 호스트들의 동적 구성이 빈번한 모바일 환경에서 데이터 공유를 위해, 각각의 모바일 호스트들이 가지고 있는 tuple space를 서로 공유한다. 따라서 각각의 모바일 호스트들은 공유된 tuple space를 통해 서로 데이터를 교환하거나, 쿼리를 통해 원하는 데이터를 얻어올 수 있다. 또한 호스트는 원하는 데이터를 등록하여, 공유된 tuple space에 해당 데이터가 발생했을 때, 이벤트를 받을 수 있다.

TinyLime은 이러한 Lime을 센서처럼 매우 제한된 리소스를 가진 장치와 효율적인 전원 관리를 지원하게끔 확장하였으며, 버클리 센서 플랫폼인 MOTE를 사용하였다. 그림 7은 TinyLime 구조를 보여준다. 센서 노드 Mote는 Lime에서는 tuple space를 가진 모바일 호스트였지만 TinyLime에서는 베이스 스테이션 호스트의 MoteAgent와 상호 작용하는 역할을 수행한다.

각각의 컴포넌트를 살펴보면, 클라이언트 호스트는 MoteLimeTupleSpace를 통해 Motes와 상호 작용한다. 여기서 MoteLime TupleSpace는 클라이언트의 request 메시지



〈그림 7〉 TinyLime 구조

를 전달하기 위한 config tuple space와 실제 센싱 된 데이터가 저장되는 motes tuple space로 구성된다. MoteAgent는 베이스 스테이션에서 동작하며, 센싱된 데이터를 관리하거나, 클라이언트의 request 메시지 처리와 이벤트 등록을 담당한다. TOSMoteAccess는 MoteAgent와 Mote들 사이의 통신을 제공한다. 즉 MoteAgent에서 보낸 request 메시지를 센서 네트워크 메시지로 변환하여 Mote로 전달하고, 그에 따른 결과 메시지를 다시 MoteAgent로 전달하는 역할을 한다.

전체적인 동작 과정은 클라이언트가 request 메시지를 config tuple space를 통해 Base Station Host에 전달하면, MoteAgent는 이러한 request 메시지를 TOSMOTAccess 컴포넌트에 전달하여, Mote로부터 센싱된 데이터를 받는다. 이 데이터는 motes tuple space에 저장되고 클라이언트는 원하는 센서의 데이터를 motes tuple space를 통해 얻을 수 있다.

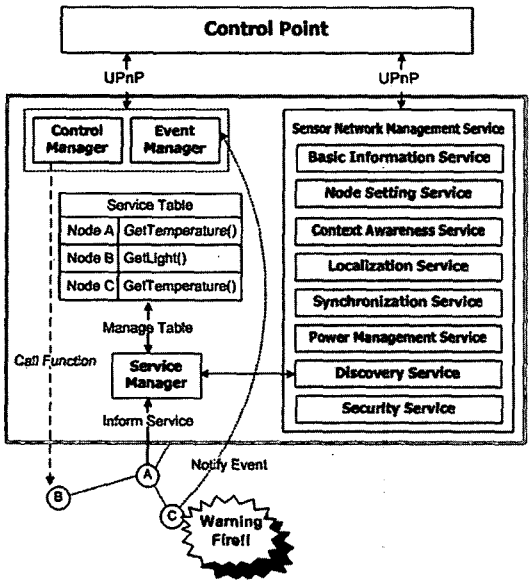
4. BOSS (Bridge of Sensors)

센서 네트워크는 그 활용 영역을 스마트 홈, 백화점, 물류 등으로 넓혀가고 있으며 센서 네트워크의 활용 범위가 넓어질수록 센서 노드들의 효율적인 설정과 관리가 필요하게 된다. ICU에서는 센서 네트워크를 IP 기반의 이 기종 네트워크와 연동하여 제어하기 위해 그림 8과 같은 BOSS 구조를 제안하였다. BOSS 구조는 센서 네트워크가 제공하는 여러 가지 서비스를 IP 기반의 UPnP[11]를 이용하여 발견하고 제어하며 이용할 수 있게 한다.

센서 네트워크에서 사용하는 비IP 기반의 네트워크와 인터넷의 IP 기반 네트워크를 연결하기 위해서는 양쪽의 기능을 연결해 줄 수 있는 게이트웨이 기능이 필요하며 센서 네트워크의 발견과 공통된 인터페이스를 통한 제어 매커니즘이 필요하다. BOSS에서의 게이트웨이 구조와 UPnP[11] 서비스는 이러한 문제를 해결한다.

UPnP[11]는 UPnP[11]를 지원하는 장치가 네트워크에 연결된 후, 특별한 설정 없이 사용자가 해당 장치로부터 제공하는 서비스를 이용할 수 있게 해준다. 이러한 UPnP[11]를 센서 네트워크에 적용하면 센서 네트워크 이용자는 수많은 센서들을 수동적인 설치 없이 바로 이용 및 관리 할 수 있게 된다. 그러나 UPnP[11]는 TCP/IP기반의 프로토콜 기반 위에서 동작하기 때문에 낮은 성능의 CPU와 제한된 메모리를 사용하는 초소형의 센서 노드에는 적합하지 않다. 따라서 BOSS에서는 UPnP[11]에 적합하지 않은 센서 노드들을 위해, BOSS(Bridge of the SensorS)라는 브리지 역할의 센서 네트워크 베이스 노드를 사용하였다. BOSS는 센서 네트워크 환경에 UPnP를 적용하기 위한 핵심 구조로서, UPnP[11]가 적합하지 않은 센서 노드들을 대

신해서, UPnP[11] 프로토콜을 처리해줌과 동시에, 센서 네트워크 환경에서 사용될 수 있는 다양한 UPnP[11] 기반의 서비스 및 관리 환경을 제공한다.



〈그림 8〉 센서 네트워크와 UPnP 네트워크의 연결

또한 센서 네트워크에서 하나의 노드에서 얻은 센싱 데이터를 통해 얻을 수 있는 정보는 지극히 미약하며, 상황 인식 (context awareness) 기능을 활용해서 좀더 효율적으로 센서 네트워크를 활용할 수 있다. 이런 효율적인 상황 인식 기능을 활용하기 위해, BOSS에서는 UPnP[11] 기반의 상황 인지 프레임워크를 제공한다. 이것은 상황을 정의하기 위한 XML 기반의 언어, 컴파일러 그리고 센서 네트워크로부터 받은 상황 인지 정보를 저장하기 위한 데이터 베이스로 구성되어 있다. 사용자는 제어 장치 등을 이용해서 원하는 상황 인지를 정의할 수 있으며, 그러한 상황이 발생했을 때, 적합한 처리를 할 수 있게 된다.

예를 들어 사용자는 특정 온도, 습도 또는 특정 지역의 사람의 존재 유무 등과 같은 상황을 정의했을 때, 해당하는 센서들을 통해서 올라온 데이터와 그것들을 융합하여 그러한 상황을 만족하는 조건을 탐지하고, 그에 따른 처리를 할 수 있다. 이것은 센싱된 데이터를 무조건 베이스 노드에서 전달해서 처리하는 것이 아닌, 노드 레벨의 처리를 요구한다. 따라서 굳이 베이스 노드까지 전달할 필요가 없는 상황 정보는 노드 레벨에서 걸러지게 되므로, 노드간의 커뮤니케이션 횟수를 줄일 수 있게 되고, 따라서 효율적인 에너지 절약이 가능하다.

이를 위해 상황 메시지를 전달하기 위한 오버레이 네트워크를 구성하고, 이벤트가 발생하는 지역에서의 효율적인 정보 전달과 노드 레벨의 결정을 위한 부가 정보 등을 이용한다. 끝으로 UPnP[11] 상황 인지 프레임워크는 전통적인 중앙 집중적인 구조에 비해서 사용자의 설정이 필요 없는 자동 구성, 조건에 따른 빠른 대응 등을 제공한다.

IV. 결론

무선 센서 네트워크 기술은 다양한 산업 응용 및 유비쿼터스 컴퓨팅 서비스를 실현시키는 핵심 기술로써, 대부분의 세부 기술들이 제한된 자원을 가지는 컴퓨팅 환경에서 동작하며 저전력 요구 사항 등의 특수한 특징을 만족 시켜야 하는 임베디드 소프트웨어의 형태로 구현이 된다. 센서 네트워크 미들웨어는 이러한 하위 시스템의 복잡도와 하드웨어에 따라 발생하는 여러 가지 제한을 추

상화하며 실제 센서 네트워크 응용의 개발과 배포 그리고 관리에 필수적인 많은 기능들을 포함한다. 지금까지의 많은 센서 네트워크 연구는 단편적이고 응용에 종속적인 기술을 사용하여 센서 네트워크 자체의 가능성을 보여주었지만 실제 센서 네트워크를 개발하고 적용하기 위해서는 훨씬 다양한 기술들이 고려되어야 하며 이를 지원하며 통합할 수 있는 미들웨어의 개발이 필수적이다. 이러한 센서 네트워크 미들웨어 개발을 통해서 RFID/USN IT 인프라의 중심 기술인 센서 네트워크를 통한 유비쿼터스 시대를 앞당기게 될 것이다.

참고 문헌

[1] C.Shen, C. Srisathapornphat, C. Jaikeo, "Sensor Information Networking Architecture and Applications," IEEE Personal Communications, Vol.8, No.4, pp. 52-59, Aug. 2001.

[2] 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.

[3] S. Li, S. Son, and J. Stankovic, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks," Int'l Workshop on Information Processing in Sensor Networks(IPSN'03), Palo Alto, CA, Apr. 2003.

[4] T. Liu and M.Martonosi, "Impala: A MiddlewareSystem for Managing Autonomic, Parallel Sensor Systems," Proc. of ACM SIGPLAN symposium on Principles and practice of parallel programming, pp.107-118, 2003.

[5] Y. Yao and J. E. Gehrke, "The Cougar Approach to In-Network Query Processing in Sensor Networks," Sigmod Record, Vol.31, No.3, Sep. 2002.

[6] A. Murphy and W. Heinzelman, "MiLAN: Middleware Linking Applications and Networks," TR-795, University of Rochester, Computer Science, Nov. 2002

[7] Carlo Curino, Matteo Giani, Marco Giorgetta, Alessandro Giusti, "Tiny Lime: Bridging Mobile and Sensor Networks through Middleware". in Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), March 8-12, 2005

[8] Hyungjoo Song,Daeyoung Kim, Kangwoo Lee, Jongwoo Sung, "UPnP-based Sensor Network Managemetn Architecture and Implementation", Second International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2005), April 2005, Osaka University, Osaka, JAPAN

[9] A. Boulis, C.C. Han, and M. B. Srivastava, "Design and Implementation of a Framework for Programmable and Efficient Sensor Networks," MobiSys 2003, San Franscisco, USA, May 2003.

[10] J.W. Branch, J.S. Davis, D.M. Sow, C. Bisdikian, "Sentire: A Framework for Building Middleware for Sensor and Actuator

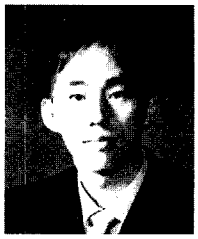
Networks"", Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on 2005 Page(s):396 - 400

[11] UPnP forum, http://upnp.org

저자소개

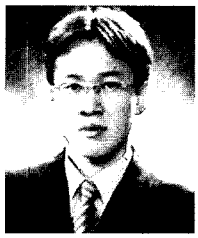
김 대 영

1992년 1월 – 1997년 8월 한국전자통신연구원 연구원
1999년 5월 – 1999년 8월 AlliedSignal Aerospace 연구소 방문연구원
2001년 9월 – 2002년 1월 Arizona State University 컴퓨터공학과 연구 조교수
2002년 2월 – 현재 한국정보통신대학교 조교수
주관심 분야 Sensor Networks, Real-Time and Embedded Systems, Ad-Hoc Networks



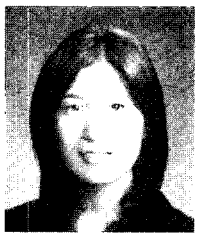
성 종 우

2002년 8월 성균관대학교 정보통신공학부 학사
2004년 8월 한국정보통신대학교 컴퓨터공학 석사
2004년 9월 – 현재 한국정보통신대학교 박사과정
주관심 분야 Home Network, RFID middleware, Sensor Network



송 형 주

2004년 2월 아주대학교 정보및컴퓨터공학부 학사
2004년 2월 – 현재 한국정보통신대학교 석사과정
주관심 분야 RFID, Sensor Network Middleware, Real-Time and Embedded Systems



김 수 현

2003년 8월 계명대학교 정보통신대학 정보통신학부 학사
2004년 9월 – 현재 한국정보통신대학교 석사과정
주관심 분야 RFID, Sensor Network Middleware