

## 속성 기반의 USN 센서 노드 동적 재구성 기법

정현준<sup>o</sup>, 김주일, 길아라, 정기원, 이우진

송실대학교 대학원 컴퓨터학과

[ignitionjhj<sup>o</sup>@hotmail.com](mailto:ignitionjhj<sup>o</sup>@hotmail.com), [sespop@empal.com](mailto:sespop@empal.com), [ara@ssu.ac.kr](mailto:ara@ssu.ac.kr), [chong@ssu.ac.kr](mailto:chong@ssu.ac.kr), [bluewj@empal.com](mailto:bluewj@empal.com)

### Attribute-based Dynamic Reconfiguration of USN Sensor Node

Hyunjune Jung<sup>o</sup>, Juil Kim, Ara Khil, Kiwon Chong, Woojin Lee  
Department of Computing, Graduate School, Soongsil University

#### 요 약

센서 노드 미들웨어는 다양한 센서 응용 소프트웨어와 운영체제 및 네트워크 기능 사이에 존재하며, 유지보수, 설치, 배포, 응용 수행에 필요한 제반 사항을 지원하며, 센서 노드에 탑재되어 센서 네트워크의 프로그램 갱신, 응용 변화에 따른 프로그래밍 조정을 수행한다. 본 논문에서는 센서 노드 미들웨어가 센서 노드의 응용 모듈을 동적으로 재구성하기 위한 방법으로 속성 값 단위의 동적 재구성 기법을 제안한다. 제안하는 기법은 센서 노드의 응용 모듈을 변경해야 할 필요가 있을 경우, 응용 모듈과 관련된 속성의 값을 변경시켜 줌으로써 응용 모듈을 재구성하는 방법이다. 논문에서는 제안하는 동적 재구성의 개념과 동적 재구성을 위한 프로세스 및 알고리즘을 설명한다. 제시하는 기법은 각처에 설치되어 동작중인 센서 노드들을 무선 통신을 이용하여 재구성함으로써 직접 센서 노드들을 떼어내거나 동작을 중단시키지 않고도 재구성함으로써 운영의 편의를 도모할 수 있는 효과가 있다. 또한 속성 값을 통한 동적 재구성방법을 통하여 용량이 큰 이미지 파일 전송으로 인한 통신 오버헤드를 줄이고 그로인해 배터리 소모를 줄일 수 있다.

#### 1. 서 론

유비쿼터스 센서 네트워크(Ubiquitous Sensor Network:USN)는 유비쿼터스 컴퓨팅 구현을 위한 기반 네트워크로 초경량, 저전력의 많은 센서들로 구성된 무선 네트워크이다 [1]. 하나의 네트워크로 연결되어 있는 수많은 센서들이 필드(Field)의 지리적, 환경적 변화를 감지하여 베이스 스테이션으로 그 정보를 전달한 후 센서 네트워크 서버를 통해 사용자에게 전달되는 방식으로 정보 수집이 이루어진다. 유비쿼터스 센서 네트워크를 통해서 언제, 어느 곳에서든 정보를 확인하고 활용할 수 있다. USN의 적용분야는 공공성이 강한 교통이나 기상, 환경, 행정정보 외에도 개인특화 서비스를 위한 각종 위치 정보나 상황정보, 사무환경 등에서 핵심기술을 응용 할 수 있는 것이다.

USN은 수 많은 작고 이질적인 센서 노드들 간의 네트워크를 의미한다[2]. 각 센서 노드들은 센싱, 컴퓨팅, 통신 기능을 포함하고 있다. USN의 특징은 제한된 리소스

와 센서노드들의 이질성, 센서 네트워크의 확장성, 무인 운용 등 이다.

센서 노드 미들웨어는 사용자에게 통합된 서비스 정보를 제공한다[3]. 센서 네트워크로부터 수집한 센싱 데이터의 필요한 자료들을 걸러내고 통합 및 분석하여 의미 있는 상황 정보를 추출하고 저장, 관리, 검색하여 응용 서비스로 전달하여 서비스 간 연계 통합 역할을 담당한다. 또한 센서 노드 미들웨어는 다양한 센서 응용 소프트웨어와 운영체제 및 네트워크 기능 사이에 존재하며, 유지보수, 설치, 배포, 응용 수행에 필요한 제반 사항을 지원한다. 센서 노드에 탑재되어 센서 네트워크의 프로그램 갱신, 응용 변화에 따른 프로그래밍 조정을 수행한다. 센서 노드에서 측정되는 센싱 데이터에 대한 데이터 베이스, 데이터 보관관리, 다양한 응용에 적합한 데이터 운영 및 처리기능, 데이터 융합 기능을 수행한다.

기존의 무선 센서 네트워크에서 각 무선 센서 노드들에 대한 재구성에서는 유선 통신을 이용한 소프트웨어 다운로드 방식을 사용하였다. 즉, 각처에 설치되어 동작

중인 무선 센서 노드들을 중단시키고 시리얼 라인 혹은 패러럴 라인의 유선 통신을 이용하여 각 무선 센서 노드로 프로그램을 다운로드하는 방식이었다. 이는 센서 노드들의 재구성을 위해 기능을 일시적으로 중단 시켜야 한다는 한계점을 가지고 있다.

이러한 문제를 해결하기 위해 동적으로 재구성하기 위한 여러 가지 기법이 연구되었다. 그런데 이러한 기법 [4,5]은 센서 노드 응용을 재구성하기 위하여 응용프로그램 전체를 변경하기 때문에 통신량이 늘어나고 전력 소모가 많아진다. 이에 따라 본 논문에서는 프로그램 전체가 아닌 필요한 부분만을 동적으로 재구성하는 기법을 제안한다.

## 2. 기존의 동적재구성 기법

현재 미들웨어에 대한 여러 가지 연구가 진행되고 있다. 동적 재구성 미들웨어에 관한 연구 중에서 SwapCIAO[4]의 동적 컴포넌트 재구성은 시스템의 정지 시간 없이 효과적으로 오퍼레이션 동작 아래에서 업데이트 동작이 이루어진다. 기존의 클라이언트의 컴포넌트를 갱신된 컴포넌트로 바꾸는 과정에서 시스템다운 시간이 없이 동적으로 컴포넌트를 바꿀 수 있는 구조를 가지고 있다. 이러한 구조는 통신 중 손실 없는 컴포넌트의 업데이트와 오버헤드를 처리해야하는 것을 요구한다.

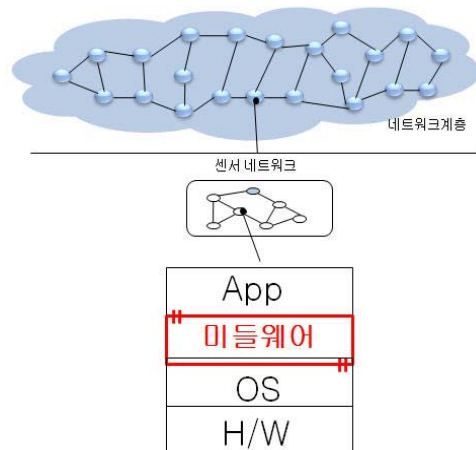
GREEN[5]의 동적인 재구성은 무선 네트워크 기술과 디바이스 계산능력의 진보를 통하여 퍼베이션 컴퓨팅을 위해 고안되었다. GREEN은 고정된 네트워크와 무선 네트워크 사이를 지원한다. 즉 GREEN 어플리케이션은 네트워크 타입(WAN, MANET)과 디바이스 타입에 등의 이질적인 환경을 움직인다. 그러나 많은 미들웨어는, 특히 고정 기반베이스 네트워크를 목표로 만들어 졌다. 그와는 극단적으로 무선 네트워크만을 위해 설계되었다. 하지만 하나의 동일한 네트워크 환경에 따라 통신하고, 하나의 디바이스에 타입에 따라 미들웨어가 만들어지는 것은 환경 제약과 필요조건의 다양성을 만족할 수 없다. GREEN은 배치와 실행 시 재구성이 가능한 미들웨어이다.

이러한 기법들은 컴포넌트 단위의 동적인 재구성을 지원함으로써 동적인 재구성이 가능하였으나 통신량에 따른 오버헤드가 높아지는 한계점을 가지고 있다. 이를 위해 본 논문에서는 동적 재구성하기 위해 모듈의 일부분을 전송함으로써 통신량을 줄이고 그에 따라서 배터리 소모를 줄이고 센서 노드의 중단 없이 재구성하는 방법을 제시한다.

## 3. 센서 노드 미들웨어 개념

센서 네트워크 미들웨어를 설계하기 위해서는 기본적으로 네트워크 구성 요소, 센서 노드 배치와 유지 및 관리, 그리고 응용 서비스의 수행을 원활하게 하는데 초점을 맞추어야 한다. 또한 최근 다양화되고 복합적인 센서 네트워크 응용의 등장으로 인해 제한된 자원의 효율적인 이용, 동적인 환경에서의 네트워크 성능 보장, 다수의 노드 구성에 의한 확장성 등의 여러 가지 기능이 센서 네트워크 미들웨어에게 요구되고 있다.

이동성이 있거나 광범위한 지역에 배치되어 있는 센서 노드들이 실행하는 소프트웨어 모듈을 사람이 일일이 갱신하는 것은 거의 불가능하기 때문에 센서 네트워크 미들웨어는 이러한 센서 노드의 모듈을 무선 네트워크를 통해 자동적으로 갱신할 수 있는 방법을 제공하여야 한다. 본 논문에서는 소프트웨어 모듈 중 일부분의 정보를 송출함으로써 효율적으로 센서노드 속성을 재구성 하고, 센서 노드의 중단 없이 재구성할 수 있는 방법을 제시한다. <그림 1>은 센서 노드 미들웨어의 개념도를 보여주고 미들웨어는 운영체제와 응용 소프트웨어 사이에 존재하며 센서 네트워크와 응용 소프트웨어 개발과 유지보수를 편리하게 하는 역할을 한다.



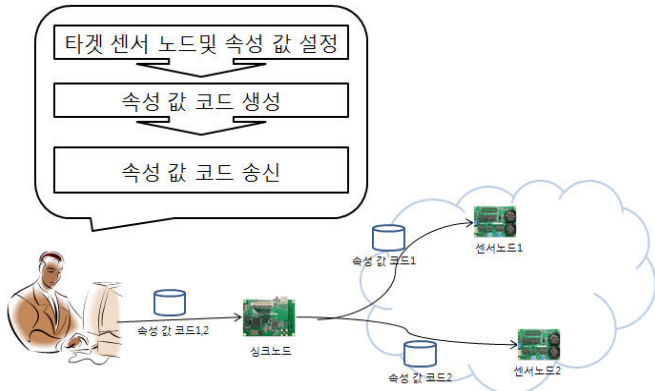
<그림 1> 센서 노드 미들웨어의 개념도

## 4. 센서 노드 미들웨어의 동적 재구성 기법

이 장에서는 센서 노드 미들웨어 동적 재구성의 개념에 대해 설명하고, 재구성 프로세스에 대해 설명한다. 그리고 속성 값을 이용한 동적 재구성 알고리즘의 의사와

드를 보여준다.

4.1 센서 노드 미들웨어 동적 재구성 개념



<그림 2> 센서 노드 미들웨어 동적 재구성 기법

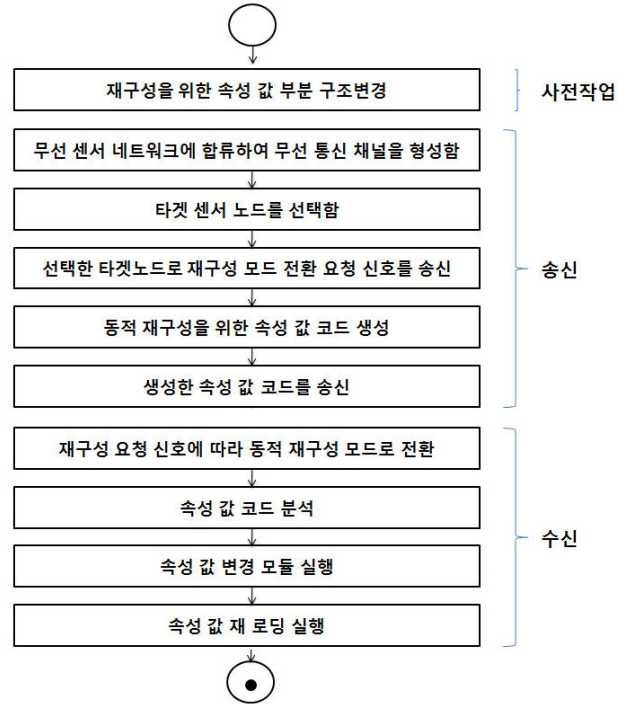
<그림 2>는 본 논문에서 제시하는 센서 노드 미들웨어 동적 재구성 기법을 나타낸다. 특정지역의 데이터를 수집하는 센서노드의 응용환경 변화로 인해 응용모듈의 설정을 변경하고자 할 경우 각 노드의 이미지 전체를 전송하지 않고 속성 값만을 전송함으로써 동적 재구성하는 것이다. 센서 네트워크를 구성하는 센서 노드들이 에너지를 가장 많이 소비하는 경우가 센서 노드가 해당 지역의 데이터를 감시할 경우이다. 때문에 시스템을 구현하여 적용할 경우 특정지역의 데이터를 수집하고자 하는 정보가 변경될 경우가 있다. 이런 경우 구축된 노드들을 다시 수거하여 해당 노드에 프로그램을 적재하는 비효율적인 방식에서 벗어날 수 있다.

센서 노드의 동적 재구성을 위해 타겟 센서 노드와 속성 값 설정을 하고 그 설정 값을 이용하여 속성 값 코드를 생성한다. 속성 값 코드를 싱크노드에 송신하고 싱크노드는 속성 값 코드를 통해 센서 노드에 송신한다.

4.2 동적 재구성 프로세스

속성 값을 이용한 동적 재구성을 위해서는 속성 값 부분의 구조 변경이 필요하다. 센서 네트워크에서는 수많은 노드들이 존재하고 그들 간의 통신이 빈번히 발생한다. 그러므로 동적 재구성을 하기위한 신호를 송신해 센서 노드를 제어하여야 한다. 타겟 노드는 신호를 수신 후에 재구성을 위한 동작을 한다.

<그림 3>은 동적 재구성 프로세스를 보여준다.



<그림 3> 동적 재구성 프로세스

재구성을 위한 속성 값 부분 구조변경 단계에서는 사전작업으로써 속성 값을 통한 재구성을 위해서 기존의 속성 값 처리 부분을 모듈화 함으로써 속성 값 변경을 가능하도록 구조를 변경 한다.

송신측의 프로세스인 무선 센서 네트워크에 합류하여 무선 통신 채널을 형성하는 단계에서는 싱크 노드와 통신을 하기 위해 센서 네트워크에 합류 하여 센서 노드와의 통신을 준비한다. 타겟 센서 노드를 선택하는 단계에서는 동적 재구성을 하고자 하는 센서 노드를 선택한다. 값으로는 고유한 아이디 등이 될 수 있다. 선택한 타겟 노드로 재구성 모드 전환 요청 신호를 송신하는 단계에서는 센서 노드의 기존에 수행하는 통신을 계속 유지하면서 속성 값 변경이 불가능하기 때문에 준비를 위해 재구성 모드 전환 요청 신호를 송신한다. 동적 재구성을 위한 속성 값 정보를 포함하는 속성 값 코드를 생성한다. 생성한 속성 값 코드를 송신 단계에서는 타겟 노드에 속성 값 코드를 송신한다.

수신측의 프로세스인 재구성 요청 신호에 따라 동적 재구성 모드로 전환단계에서는 센서 노드가 행하던 수집 행위를 멈추고 재구성을 위하여 대기하는 것을 말한다. 속성 값 코드 분석 단계에서는 수신한 속성 값 코드를 분석하여 자신의 노드가 맞는지 그리고 제대로 된 속성 값이 설정되어 있는지에 대해서 분석한다. 속성 값이 유효한 것이라면 속성 값의 변경을 위해 재구성하는 모듈

을 실행하는 부분이다. 속성 값 재 로딩 실행 단계에서는 속성 값의 재구성을 센서 노드에 적용하기 위해 속성 값 재 로딩을 실행한다.

4.3 센서 노드 속성 값의 동적 재구성 방법

속성 값에 의한 동적 재구성을 위해서는 센서 네트워크를 구성하는 각 노드들의 속성 값을 정의하고, 역할에 따라 필요한 속성 값을 설정해야 한다. <표 1>은 동적 재구성을 위한 속성 값 코드를 생성하기 위해 설정해 주어야 하는 노드의 속성을 종류별로 분류해 놓은 것이다.

<표 1> 노드의 속성

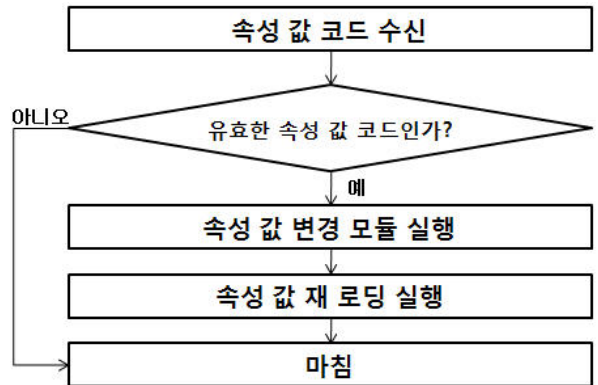
속성 분류	설명
네트워크	센서 네트워크 통신을 수행하는 데 필요한 속성이 포함된다. 예) nodeID, adjacentActuatorNodeID, PAN_Cordinator_Node_Enable, defaultMACAddr, associationPermiStartNodeID, associationPermitEndNodeID, nextHopRoutingFirstNodeID, nextHopRoutingSecondNodeID, rfChannel, scanChannel, Zigbee RF
스케줄러	스케줄러의 종류를 설정하기 위한 속성이 포함된다.
디바이스 드라이버	각 센서 노드의 디바이스 드라이버를 생성하기 위해 필요한 속성이 포함된다. 예) EEPROM_Enable, UART, Flash_Memory_Enable, RSSI_Enable
센서 타입	센서의 종류를 결정하기 위한 속성이 포함된다. 예) Sensor_Temperature_Enable, Sensor_Light_Enable, Sensor_Gas_Enable, Sensor_Humidity_Enable

<표 2>는 <표 1>에서 분류해 놓은 속성을 바탕으로 싱크 역할을 수행하는 노드의 속성 값을 설정한 예를 보여준다. 이러한 속성 값 설정을 통하여 각 노드의 역할에 따라 스케줄러 타입, 노드의 타입 등을 결정할 수 있다.

<표 2> 노드의 속성값 설정 예

속성(Attribute)	값(Value)
Adjacent Actuator nodeID	"0"
Association Permit End Node ID	"99"
Association Permit Start Node ID	"1"
Default MAC Addr	"1111"
EEPROM Enable	"false"
Flash Memory Enable	"false"
Next Hop Routing First Node ID	"5"
Next Hop Routing Second Node ID	"1"
Node ID	"0"
Node Type	"SINK"
Rf Channel	"25"
RSSI Enable	"false"
...	...

<그림 4>는 센서 노드 속성 값의 동적 재구성 기법을 구현하기 위해 작성한 알고리즘이다. 타겟 센서 노드가 속성 값 코드를 수신함에 따라 디바이스의 속성 값을 설정해주시고 하고 받아오기도 한다. 재구성 하고자 하는 센서 노드의 속성 값에 따라 관련 모듈을 실행한다.



<그림 4> 센서 노드 속성 값 재구성 알고리즘

5. 결론 및 향후연구

기존의 동적 재구성에 대한 연구에서는 컴포넌트를 교체하는 방식으로 재구성을 하였다. 그러나 이러한 방법은 통신량에 따른 오버헤드가 높아지고 전력 소모가 많은 문제점을 가지고 있다.

이러한 문제점을 보완하기 위하여 본 논문에서는 속성 값 단위의 동적 재구성 기법을 제안한다. 제안하는 기법은 센서 노드의 응용 모듈을 변경해야 할 필요가 있을

경우, 응용 모듈과 관련된 속성의 값을 변경시켜 줌으로써 응용 모듈을 재구성하는 방법이다.

이 기법은 각처에 설치되어 동작중인 센서 노드들을 무선 통신을 이용하여 재구성함으로써 운영상의 편의성을 향상시킬 수 있다. 즉 직접 센서 노드들을 떼어내거나 동작을 중단시키지 않고도 재구성함으로써 운영의 편의를 도모할 수 있는 효과가 있다. 또한 속성 값을 통한 동적 재구성방법을 통하여 용량이 큰 이미지 파일 전송으로 인한 통신 오버헤드를 줄이고 그로인해 배터리 소모를 줄일 수 있다.

향후에는 본 논문에서 제시한 기법을 바탕으로 속성 값을 이용한 동적 재구성 미들웨어를 구현하는 연구를 진행할 것이다.

## 6. 참고문헌

- [1] 정기원, 김주일, 이우진, “유비쿼터스 센서 네트워크에서 응용 프로그램 개발을 위한 모델 기반 통합 개발 도구,” 한국정보과학회 논문지 C - 컴퓨팅의 실제, VOL.13 NO.07 pp.0442 ~ 0453, 2007.12
- [2] 유남철, 유비쿼터스 센서 네트워크 기술개발 동향, 전자정보센터, 2007.2
- [3] 김영만, 한재일, 센서 미들웨어 기술, 한국정보과학회 학회지, VOL.25 NO.12 pp.0035 ~ 0048, 2007.12
- [4] Balasubramanian, J., Natarajan, B., Schmidt, D. C.,: “Middleware Support for Dynamic Component Updating,” On the move to meaningful internet systems 2005 : CoopIS, DOA, and ODBASE, pp.978-996, 2005
- [5] Sivaharan, T., Blair, G., Coulson, G., “GREEN: a configurable and re-configurable publish-subscribe middleware for pervasive computing,” On the Move to Meaningful Internet Systems 2005, Proceedings (Lecture Notes in Computer Science Vol.3760), 2005, pp.732-49