

# 센서노드상의 응용모듈 동적재구성을 위한 버전동기화 도구\*

정선우<sup>o</sup>, 길아라, 정기원  
송실대학교 대학원 컴퓨터학과  
{sunoo7906<sup>o</sup>, ara, chong}@ssu.ac.kr

## A Version Synchronization Tool for Dynamic Reconfiguration of Application Modules on Sensor Nodes

Sunwoo Jung<sup>o</sup>, Ara Kil, Kiwon Chong  
Department of Computing, Graduate School, Soongsil University

### 요 약

일반적으로 수십 개에서 많게는 수백 수천 개의 센서노드로 구성된 센서네트워크 환경에서 각 노드들은 제한된 저장 공간과 효율적인 에너지자원소비 등의 제약사항을 가지고 있다. 따라서 센서노드에서 응용모듈에 대한 형상관리를 수행하기에는 어려움이 있다. 본 논문은 한국전자통신연구원(ETRI)에서 개발된 나노큐플러스 운영체제(Nano-Qplus OS)기반의 센서네트워크 환경에서 각 노드들의 저장 공간이나 에너지 자원 등의 여러 제약조건을 고려한 버전동기화 도구를 제안한다. 이것은 윈도우즈 기반의 eclipse개발환경과 버전관리도구인 CVSNT와 연동하여 각 응용모듈의 버전을 관리하고, 이를 이용하여 본 논문에서 소개하는 센서노드와 버전정보 저장소간의 버전동기화 도구인 NVSync(Nano-Qplus Version Synchronization)를 사용하여 센서노드의 응용모듈과 동기화함으로써 동적으로 노드들의 응용모듈을 재구성할 수 있게 한다.

### 1. 서 론

U-센서네트워크(Ubiquitous Sensor Network)는 향후 IT분야의 핵심 인프라 산업으로써, 현재 다양한 연구가 활발하게 진행되고 있다. 센서네트워크는 주기적으로 데이터를 수집하고, 수집된 데이터를 분석하여 시스템의 상황에 맞게 동작하는 자동화를 가능하게 하는 기술로써 여러 산업분야에 적용하기가 용이하다. 센서네트워크를 활용하면 가정 내의 원격감침이나 여러 사용자에게 적합한 환경을 제공할 뿐만 아니라, 사람이 접근하기 어렵거나 위험한 지역의 데이터를 수집할 수 있다. 하지만 현재 기술력으로 센서 노드에 활용할 수 있는 크기의 배터리로는 가용 에너지가 너무 부족하다. 따라서 최근 센서네트워크의 연구는 에너지 효율성을 고려하여 진행되고 있다. 네트워크 분야에서는 노드와 노드 사이에 정보를 송수신하는 경우에 가장 많은 에너지가 소모되기 때문에 가능한 적은 양의 데이터와 시그널을 전송하여 에너지 효율을 높이는 것이 중요한 이슈 가운데 하나이다[1].

본 논문은 센서네트워크에서 최소한의 메모리 자원을 사용하는 체계적인 노드의 응용모듈 버전관리기법과 이를 기반으로 노드의 여러 응용모듈에 대해 결함수정이나 기능 변경 및 개선 등을 더욱 효과적으로 관리하기 위한 적절한 형상관리 도구를 제안한다. 이 도구는 여러 센서네트워크운영체제 가운데 한국전자통신연구원에서 개발하고 있는 센서네트워크 운영체제인 나노큐플러스 운영체제환경에 특화하여 이를 구현한다. 본 논문에서 제안

하는 도구를 센서네트워크 노드 응용모듈의 형상관리에 적용할 경우 여러 개발자가 공동으로 센서노드의 응용모듈을 개발할 수 있어 노드를 구성하는 응용모듈의 버전변경에 대한 가시성을 향상시킬 수 있다. 또한 노드 상의 버전관리를 위한 응용모듈의 노드정보를 삽입하여 센서네트워크를 구성하는 각 노드의 버전상태를 체계적으로 관리함으로써 노드의 응용모듈을 동적재구성하여 센서노드의 응용환경의 변화에 유연히 대처할 수 있다.

### 2. 관련연구

#### 2.1 센서네트워크

센서네트워크는 유비쿼터스 컴퓨팅분야에서 핵심기술로써 작은 범위에서 동작하는 RF(Radio Frequency) 통신 매체를 통해 연결된 무선 지능 센서들이 유비쿼터스 컴퓨팅 환경에서 사람과 컴퓨터 사이에 중개인으로써의 역할을 수행한다. 기존의 컴퓨팅 플랫폼과 달리 센서노드들은 다음과 같은 세 가지 특징을 가진다[2]. 1) 센싱정보에 대한 처리 능력과 저장메모리, 배터리 등의 모든 자원이 매우 제한적이다. 2) 센서네트워크를 여러 작은 자율적인 노드들로 구성된 분산컴퓨팅 플랫폼으로 보는 데이터 중심형(data-centric) 프로그래밍 스타일이 일반적이다. 3) 재사용을 고려하지 않는 일회용 컴퓨팅 플랫폼이다.

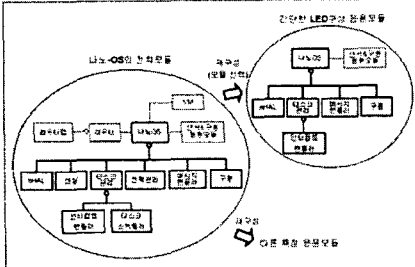
센서노드는 이러한 특징 때문에 에너지소비와 저장메모리가 적은 장치에서 동작할 수 있으면서 동시에 환경과 응용 환경의 변화에 대처하기 위해 동적재구성을 지원할 수 있는 독특한 구조의 운영환경이 필요하다. 최근 이러한 문제점을 해결하기 위하여 구축된 노드의 비정상

\* 본 연구는 송실대학교 교내연구비 지원으로 이루어졌음

적인 고장이거나 에너지 고갈 등의 문제점이 발생하였을 경우 노드 스스로의 판단에 의해 가장 적절한 노드의 연결을 구성하는 자기구성(Self-organization), 노드의 근형적인 에너지 소비, 사용하지 않은 노드에 대해 휴면상태로 설정하여 에너지 소비를 줄이는 방법 등 여러 라우팅 기술 등을 통해 각 노드의 라이프 타임을 늘리는데 초점을 맞춰서 연구들이 진행되고 있다. 이와 함께 각 노드의 적은 저장 공간에서 효과적인 역할을 수행하고 데이터(패킷)중심의 센싱정보를 보다 안전하고 신뢰할 수 있는 전송 방법들이 연구되고 있다.

### 2.2 나노큐플러스스(Nano-Qplus)

나노큐플러스스[5]는 한국전자통신연구원(ETRI)에서 개발한 국산 초소형 운영체제 개발솔루션으로, 소형 센서네트워크 시스템 개발을 위한 운영체제기술, 무선통신 기술, 목표 시스템 기능설정 및 설치 기술, 소스편집, 모니터링 등의 기능을 수행하는 소프트웨어 기술들이 적용된 것이다. 이 운영체제는 국방, 디지털헬스, 의료, 환경, 건설분야 등 산업 전반에 걸쳐 적용 가능하다. 또한 이는 여러 응용분야에 적합한 초소형(10KB이만 커널), 분산/실시간, 스마트(상황인식) 운영체제이며, 유비쿼터스 센서네트워크(USN)를 구축하기 위한 나노운영체제 플랫폼 제공한다.



[그림 1] 나노큐플러스스 운영체제와 응용모듈 재구성

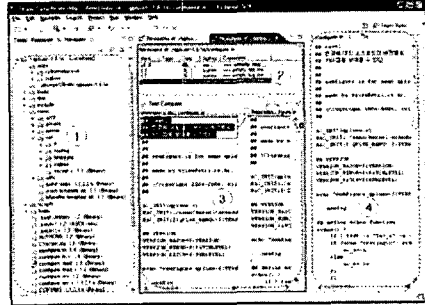
응용환경에 적합한 운영체제의 요구사항을 만족하는 모델링을 바탕으로 [그림 1]과 같은 재구성 방법을 사용하고 있다. 전체 나노 운영체제의 모듈에서 적용하고자 하는 응용에 필요로 하는 모듈만을 설정하여 그에 최적화된 응용모듈을 구성한다. 재구성 방법에 의해 최적화된 코드들은 실제 설치 전이라면 프로그램 인터페이스를 통해 프로그램 할 수 있다. 그러나 센서 노드들이 설치되어 동작하는 중에 발생하는 버그나 새로운 요구사항에 대응하기 위해서는 동적인 재구성 방법이 지원되어야 한다[3].

### 2.3 버전관리도구[4]

버전관리도구가 필요한 가장 큰 이유는 소프트웨어를 개발 시에 한 사람이 큰 프로젝트를 단일로 진행할 수 없기 때문에 공동 작업을 하기 위함이다. 버전관리도구의 가장 큰 장점은 잘못 수정한 소스를 쉽게 되돌릴 수 있다는 점이다. 프로젝트가 커지다 보면 응용모듈이 복잡해지고 이에 대한 버전정보의 변경에 대한 가시성이 떨어지게 된다. 이러한 문제점을 해결하기 위해서는 변경사항이 모두 기록되고 무엇을 변경 했는지 쉽게 볼 수 있는 도구가 필요하다.

### 3. 노드 응용모듈의 버전관리

각 노드 응용모듈은 여러 하위모듈들로 구성되어 있으며 그 하위모듈은 더 세부적인 하위모듈 또는 그 하위모듈을 구성하는 여러 형상항목으로 분류하여 계층적으로 구성된다. 일반적으로 형상항목은 소스코드, 문서, 테스트 집합 등을 의미하지만, 본 논문에서는 형상항목을 소스코드 단위로 한정하며 소스코드의 변경이 노드의 기존의 기능을 향상시키거나 새로운 기능이 추가할 경우 버전정보를 넘버링을 하여 저장소에 저장한다[5].



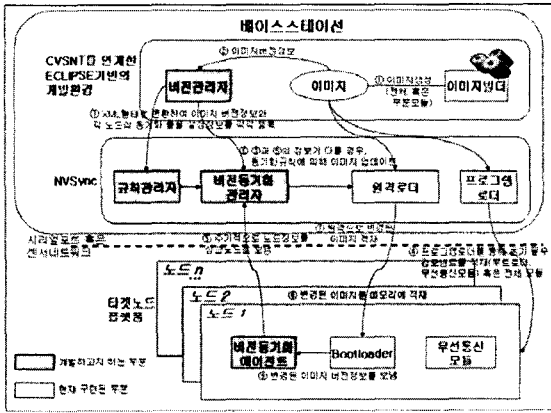
[그림 2] eclipse와 CVSNT를 연동한 공동개발환경

[그림 2]는 eclipse 기반의 개발환경에서 버전관리도구인 CVSNT를 연동하여 형상항목에 대한 버전관리를 나타낸다. 개발팀 혹은 개발자들이 신규로 생성되는 소스코드와 그 소스코드의 버전정보를 저장소에 저장한다. 또한 모듈의 변경이 필요할 경우 해당모듈을 다시 불러와서 수정한 후에 리버전(reversion) 과정을 통해 소스코드를 수정하고 버전정보 변경을 수락(commit)한다. 이 버전관리도구는 각 모듈을 CVSNT 서버에 저장하여 개발자들이 클라이언트로 접속하여 모듈의 변경에 대한 공동 작업이 가능하고, 별도로 모듈을 개발하기 위한 가지치기를 지원함으로써 수정에 대한 복구를 용이하게 한다. 노드의 응용모듈에 대한 변경상태를 추적하기 위해 각 노드의 노드정보만을 이용하여 개발자환경의 버전정보 저장소에 저장되어 있는 버전상태를 자동적으로 화면에 보여줌으로써 개발자가 한눈에 변경상태를 알 수 있으며 노드의 버전상태를 체크하여 항상 최신의 버전을 유지할 수 있다.

### 4. 노드의 버전동기화 도구

이 도구는 특정지역의 데이터를 수집하는 센서노드의 응용환경 변화로 인해 응용모듈이 변경되어야 경우나 노드의 성능개선을 위해 설치된 노드의 응용모듈을 수정하고 할 경우 각 노드의 노드정보맵에 저장된 노드의 버전정보를 수정함과 동시에 무선통신을 통해 타겟노드의 노드정보를 전송받아 타겟노드의 응용모듈을 동적으로 재구성하는 도구이다. 센서네트워크를 구성하는 센서노드들의 가장 많이 에너지를 소비하게 하는 부분은 센서노드가 해당 지역의 데이터를 수집(sensing)할 때이기 때문에 본 논문에서 제안하는 시스템을 구현하여 적용할 경우 특정지역의 데이터를 수집하고자 하는 정보가 변경

될 경우 해당노드에 필요한 센싱정보만을 수집할 수 있도록 응용모듈을 변경할 수 있다. 또한 네트워크를 구성하는 노드들의 응용모듈이 변경될 경우에도 구축된 노드들을 다시 수거하여 해당노드에 각각에 프로그램을 적재하는 비효율적인 방식에서 벗어나 수 있다. 이는 각각의 노드에 대한 노드정보를 NVSync의 노드정보맵에 저장하여 설치된 노드의 응용모듈의 버전정보를 확인할 수 있으며, 해당노드의 버전정보를 알아보기 위한 별도의 이벤트가 필요하지 않다. 또한 응용모듈의 변경이 필요한 노드들의 응용모듈을 NVSync를 통해 동적으로 수정할 수 있게 된다. [그림 3]은 버전동기화 도구인 NVSync의 전체 시스템 아키텍처를 나타내며, [표 1]은 각 구성요소의 기능에 대한 설명이다.



[그림 3] NVSync의 시스템 아키텍처

[표 1] NVSync 시스템의 기능

구성요소	설명
NVSync UI	• NVSync 시스템의 전체 사용자인터페이스
버전동기화 관리자	• 센서네트워크를 구성하는 노드의 노드정보맵을 등록/수정하는 기능 • 구축된 타겟노드의 노드정보와 노드정보맵에 저장된 값을 맵핑/해제하는 기능 • 노드정보맵과 노드모니터의 버전정보를 비교하여 버전동기화를 결정하는 기능 • 각 노드의 버전동기화 사전규칙을 결정하는 기능
버전동기화 엔진	• 응용모듈 저장소에서 변경된 노드정보를 XML 형태로 저장된 노드정보맵의 데이터베이스 기능
규칙관리자	• 즉시모드, 주기모드, 사용자모드로 노드의 동기화규칙을 설정
프로그램 번더	• 초기 타겟보드에 응용환경에 맞는 프로그램 혹은 초기 필수 컴포넌트를 적재하는 기능
원격유니	• 무선통신을 통해 타겟노드의 변경 이미지를 재구성하기 위해 부트로더로 보내는 기능
버전동기화 에이전트	• 업데이트된 노드정보를 싱크노드로 보내는 기능 • 부트로더에 의해 타겟노드의 응용모듈에 대한 버전동기화를 수행후 노드모니터에 업데이트된 노드정보를 넘겨주는 기능

간단한 시나리오를 예로 들면, 가정 내의 가스누출을 검침하는 센서노드가 특정부분에 있었다가 가스보관소가 변경된다면 최초로 설치한 노드에는 가스에 대한 수정정보는 더 이상 필요하지 않게 되고 새로운 가스보관소에는 가스센싱이 가능한 노드가 필요하다. 이때 최초 설정된 노드는 불필요한 정보를 계속 수집하게 되고 그 정보를 싱크노드로 보내려고 하면 그때마다 불필요한 정보를 싱크노드로 전송하게 된다. 불필요한 정보를 전송하는

것은 노드의 에너지를 낭비하는 요소가 된다. 이런 경우에 NVSync를 사용하면 기존의 가스보관소에 위치한 노드들에 대해 가스 센싱기능을 제거하고 새로운 가스보관소에 위치한 노드들에 가스누출 감지기능을 추가함으로써 센서노드들이 올바르게 동작하게 할 수 있다.

[표 2] 기존 센서네트워크운영체제와의 비교

기능	TinyOS	SOS	Maté	Nano-Opus (NVSync)
운영환경	Event-driven	Event-driven	Event-driven	Thread-driven
재프로그래밍 용이	Yes (NesC, C)	Yes (C used)	Yes (VM)	Yes (autoconf, automake)
재구성 방식	전체	모듈	파라미터	전체
유연성과 최적화	Yes (강함)	Yes (약함)	No	Yes (강함)
업데이트 비용	높음	중간	낮음	높음
원격 업데이트 유무	Yes	Yes	Yes	Yes
버전관리도구 지원	Yes (CVS)	Yes (CVS)	No	Yes (CVS)
자기적응성	No (사용자)	No (사용자)	No (사용자)	Yes (사전규칙)
버전동기화	No	No	No	Yes

5. 결론 및 향후연구

본 논문에서는 센서네트워크를 구성하는 노드들의 응용모듈의 변경에 대한 가시성을 향상시키기 위한 버전관리 기법과 기구축된 센서네트워크의 동적재구성을 지원하는 도구를 제안하였다. 노드정보만을 이용하므로 형상관리를 위한 노드의 제약된 저장공간을 최소화할 수 있다. 또한 본 논문에서 제안한 버전동기화를 통해 노드의 응용모듈을 동적으로 재구성이 가능하여 특정 목적으로 구축된 센서네트워크를 최선의 상태로 유지할 수 있다.

향후 연구에서는 본 논문에서 제시한 설계를 바탕으로 노드의 응용모듈의 변경정보를 관리하기 위한 CVS기반의 형상관리 도구를 구현하고, MATLAB을 사용하여 에너지소비와 메모리 사용에 대한 시뮬레이션을 수행함으로써 효과성을 검증한다. 또한 각 노드의 전체응용모듈을 더 작은 단위로 동적으로 재구성하는 버전동기화에 관한 연구를 진행할 것이다.

6. 참고문헌

- [1] 이재용, "유비쿼터스 센서 네트워킹 기술," TTA저널 제95호, pp.78-83, 2004. 10
- [2] 홍성수, 김태형, 함승현, 박선희, "SenOS: 동적 센서 노드 재구성을 위한 상태 기반 운영체제 구조," 한국정보과학회, 제30회 추계학술대회, pp.349-351, 2003. 10
- [3] 신용삼, 최희석, 이광용, 박승민, "동적 재구성 가능한 센서 네트워크 시스템 구현을 위한 하드웨어 및 소프트웨어 플랫폼 설계 및 구현," 제8회 차세대통신소프트웨어학술대회, 2004. 12
- [4] 형상관리도구, <http://www.daveeaton.com/acm/CMTTools.html>
- [5] Hagen Volzer, Anthony MacDonald, Brenton Atchison, Andrew Hanlon, Peter Kindsay, Paul Strooper, "SubCM: A Tool for Improving Visibility of Software Change in an Industrial Setting," IEEE Computer Society, Oct.2004