

RFID 미들웨어 환경에서 센서 노드의 생존성 향상과 효율적인 프로토콜 설계를 위한 연구

최용식^o 전영준 박상현 한수 신승호
인천대학교 컴퓨터공학과

{mars^o, 0961144, tank1862, puckktan, shin0354}@incheon.ac.kr

A study on improve survivability of sensor node and design of protocol in RFID Middleware environment

Yong-Sik Choi^o Young-Jun John Sang-Hyun Park Soo Han Sung-Ho Shin
Department of Computer Science & Engineering, University of Incheon

요 약

센서 노드의 송수신 상태를 분석하여 노드의 생존성 향상과 효율적인 프로토콜 설계를 하려고 한다. 센서 노드의 분석을 위한 실험 환경은 다음과 같다. 센서 노드의 생존성- 가용 배터리, 센서 노드의 출력 - 검색 가능 영역, 센서 노드의 통신 경로 - 라우팅 테이블 생성, 센서 노드의 대역폭 - 송신 데이터의 크기이다. RFID 태그와 리더를 통한 관리 시스템과 재해방지를 위한 다양한 센서를 통한 정보 수집 시스템과 의사결정 시스템에 적용 될 수 있다. 그리고 다양한 센서 데이터로부터 수신된 데이터의 자료수집, 센서분류, 수신을 조절 시스템을 위한 프로토콜 설계 자료로 활용 가능하다.

KEYWORD: 센서 노드 , RFID, 미들웨어

1. 서 론

유비쿼터스 컴퓨팅 환경은 각 디바이스들이 생활의 곳곳에 널리 퍼져 있고 이러한 디바이스들을 통해서 어느 곳에서나 정보의 이동의 용이하다. 유비쿼터스 환경을 위한 핵심기술인 RFID(Radio Frequency Identification)를 통하여 사물의 인식정보 및 주변의 환경 정보까지 센싱하고 이를 연결하여 정보를 관리한다. 유비쿼터스 관련 서비스는 환경 정보 센싱, 자동차 분야, 환경 관리 분야, 물류 유통 분야에 사용 가능하다. 유비쿼터스 산업은 미디어 가전 산업, 홈오토메이션, 교육, 국방, 의료, 통신, 환경, 건축, 재난방제, ITS, e-커머스 산업에 이르기까지 다양한 산업이 결합되는 융합 산업의 기반이 될 것이다. 유비쿼터스 지향의 미들웨어 기술은 서비스에 따른 다양한 컴포넌트들의 동적인 변화를 가져올 것이다. 하드웨어 기술은 칩셋을 제외한 핵심 소프트웨어 기술로 차별화 될 것으로 보인다. [1], [2], [10], [11].

본 논문에서는 다양한 센서 데이터로부터 수신된 데이터를 분석하여 센서 노드의 생존성 향상과 효율적인 프로토콜 설계를 위한 자료로 사용하고자 한다.

각각의 센서 노드는 부착된 배터리에 의해 제한된 양의 전력을 사용할 수 있다. 만일 각각의 센서 노드들이 매우 낮은 주기로 통신하고 통신을 위한 경로가 적절하다면 각각의 노드들은 극히 적은 양의 전력을 소모한다. 이런 가정 하에 센서 노드가 별도의 관리 없이 수개월 이상 사용 될 수 있는 것이다. 그러나 이는 이상적인

환경만을 고려한 것이다. 가령 예측할 수 없는 극심한 방해 전파와 밀집된 센서 배치로 인한 중복 경로 설정 등의 요인으로 인해 센서 노드의 생존성은 수개월에서 단 하루로 낮아질 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 RFID/USN와 미들웨어의 주요개념과 무선 센서 네트워크에서 사용되는 운영체제인 TinyOS 개념과 설치 환경에 대하여 설명한다. 3장에서는 각 센서 노드의 생존성을 향상시키기 위해 센서 노드의 환경을 분석하였다. 센서 노드의 환경은 다음과 같다. 센서 노드의 생존성- 가용 배터리, 센서 노드의 출력 - 검색 가능 영역, 센서 노드의 통신 경로 - 라우팅 테이블 생성, 센서 노드의 대역폭 - 송신 데이터의 크기이다. 4장에서는 실험 결과를 고찰하고 마지막으로 5장에는 결론과 향후 연구방향에 대해 논의한다.

2. 관련연구

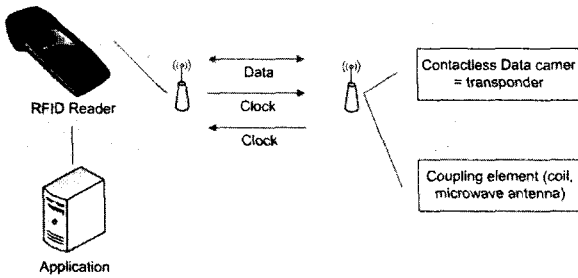
2.1 RFID (Radio Frequency Identification)

자동인식(Auto-ID)은 서비스 산업, 구매 유통, 재고관리 산업분야, 제조사 및 자재 유통 등 다양한 분야에서 보편화되고 있다. 자동인식은 사람, 동물, 상품 및 운송 중에 있는 제품에 관한 정보를 제공하기 위해 존재하고

있다. 과거에는 바코드 라벨이 자동인식 시스템의 혁명을 일으키는 촉매가 되었으나 부적합하게 되는 경우의 수가 증가하고 있다. [연결이 부자연스러움 수정필요] 전력전달과 데이터 전송에 사용되는 절차 때문에 비접촉 ID 시스템인 RFID(Radio Frequency Identification Systems)이다. 최근 몇 년 동안 비접촉 인식은 기존의 정확하게 분리되는 학문이 아닌, 다른 학문과 연관관계가 있는 분야로 발전해 오고 있다. 이 분야는 지극히 다른 분야: HF 기술과 EMC, 반도체 기술, 데이터 보호 및 암호화, 통신, 제조 기술 및 많은 관련 분야로부터의 요소들을 종합하고 있다. [4], [8], [11].

RFID 시스템은 언제나 두 개의 구성 요소로 되어 있다. [그림 1]

- 인식되어야 할 대상인 트랜스폰더
- 설계 및 기술방법에 따라 읽기 또는 쓰기/읽기 장치로 구분되는 송신부 또는 리더



[그림 1] RFID 시스템의 주요 구성

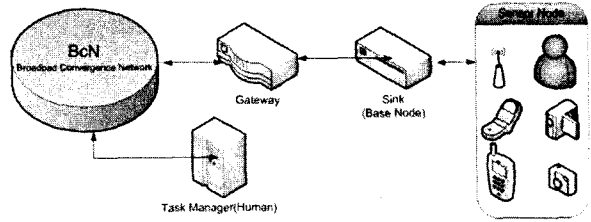
일반적으로 리더는 제어 기능과 트랜스폰더와 연결 기능을 하는 무선 주파수 모듈(송신기와 수신기)을 가지고 있다. 또한 대부분의 리더는 추가적 인터페이스(RS 232, RS 248 등)가 있어서 수신된 데이터를 다른 시스템으로 송신한다(PC, 로봇 제어장치 등). [10], [12].

2.2 USN(Ubiquitous Sensor Network)

USN(Ubiquitous Sensor Network)은 모든 사물에 전자 태그(RFID)를 부착하여 사물 정보 및 환경정보까지 감지하여 유무선 네트워크에 실시간으로 연결하여 수집, 가공하여 유비쿼터스 사회에서 필요한 정보를 제공하고 센서를 관리하는 시스템을 의미한다.

이것은 기존의 RFID 기술에 네트워크 기능이 추가되어 사물의 정보화하고 센서 기술 및 저가의 저전력 무선 통신 기술의 발달로 인하여 센서 자체가 지능화 되고 센서 노드의 상호 네트워킹을 구성할 수 있다. [그림 3]은 센서 네트워크를 구성하는 센서 노드의 블록도와 센서

네트워크의 구조를 나타낸다. [5].



[그림 3] USN 구조도

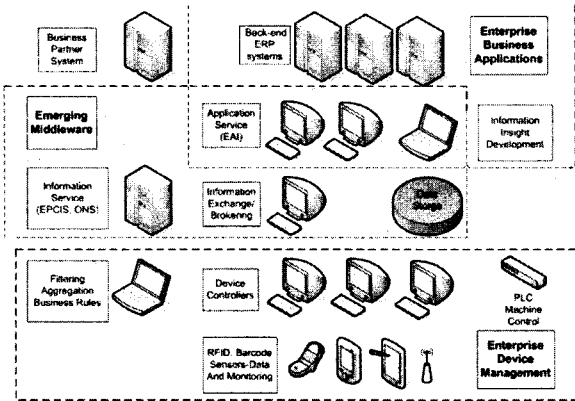
센서는 저가, 고기능 무선통신, 저전력의 강점을 지닌 RFID 태그가 모든 사물에 부착되어 위치정보, 온도, 습도, 압력 등의 환경 정보를 제공하며 USN은 이 정보를 가공하여 조도, 온도 등 환경을 조절하는데 이용한다. 이와 같은 것은 MEMS (Micro-Electro-Mechanical System) 기술을 이용한 속도 감지기, CCTV, 마이크, 히터, 쿨러 등 사람이 보거나 느끼지 못할 정도의 작은 센서가 무수히 연결된다. 즉 각각의 네트워크의 노드가 된다. 이것은 정보 제공 역할, 전달, 수행 역할 모두를 수행한다. [6], [9].

2.3 RFID 미들웨어

유비쿼터스 서비스는 다수의 분산된 센서 데이터의 스트림 처리 기술과 분산 객체 정보 통합 관리 기술이 필요하다. 이러한 문제점을 해결하는데 필요한 것은 RFID 미들웨어이다. RFID 미들웨어는 [그림 4]와 같이 하드웨어 영역에 속하는 RFID 리더와 통신하여 얻어진 데이터를 백엔드 애플리케이션이 사용할 수 있도록 연결시켜 주는 역할을 한다. 일반적으로 하드웨어 영역과 미들웨어, 애플리케이션을 포함하는 소프트웨어 영역으로 나누어 볼 수 있다. [4], [14].

RFID 미들웨어는 다음과 같은 요구사항을 가진다.

- 1) RFID 미들웨어는 다양한 형태의 리더 인터페이스, 다양한 코드 및 망 연동, 여러 가지 응용 플랫폼에 대해서도 상호 운영성을 보장할 수 있어야 한다.
- 2) 상호 운영성을 보장하는 미들웨어를 개방형 미들웨어라 하며, 이를 위하여 표준화된 코드, 정보표현, 교환규약을 준수하고 정보교환을 위한 메시징 기술을 적용해야 한다.
- 3) 요구사항을 준수하기 위해서는 EPC Specification, ISO 표준, 웹서비스 등과 같은 참조 표현을 준용하고 멀티리더 프로토콜(EPC, ISO 15961, Alien)을 지원하는 통합 인터페이스의 설계 및 구현이 요구된다.



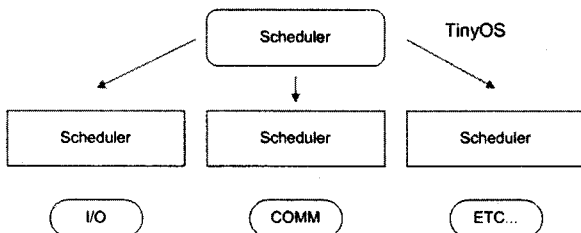
[그림 4] RFID 시스템과 미들웨어

RFID 미들웨어의 주요 기능은 데이터 모니터 및 관리 (데이터 수집, 필터링, 라우팅, 실시간 데이터 처리), 디바이스 관리 및 모니터, 어플리케이션 지원, 정보서비스, 공통 어플리케이션 레이어 지원이 있다.

2.4 TinyOS

기존의 UNIX와 같은 운영체제가 32비트 마이크로 프로세서 위에서 수메가 바이트 메모리를 필요로 한다. 센서 네트워크에서는 센서노드는 8-16비트 마이크로 프로세서의 수십 kbyte 정도의 메모리를 가진다. 사용 가능한 운영체제로는 TinyOS, MicroC/OS, Nucleus, Nano-X등이 있다. TinyOS는 하드웨어로 직접 구현되었으며 하나의 프로세서로서 하나의 물리적인 주소 공간만 필요하다. 메모리는 컴파일시 동적으로 할당하며 소프트웨어 신호나 예외처리 대신 Function Call을 사용하므로 센서네트워크에 적합한 운영체제이다. [3], [15].

[그림 5]에서 제시된 TinyOS의 특성은 아래와 같다.



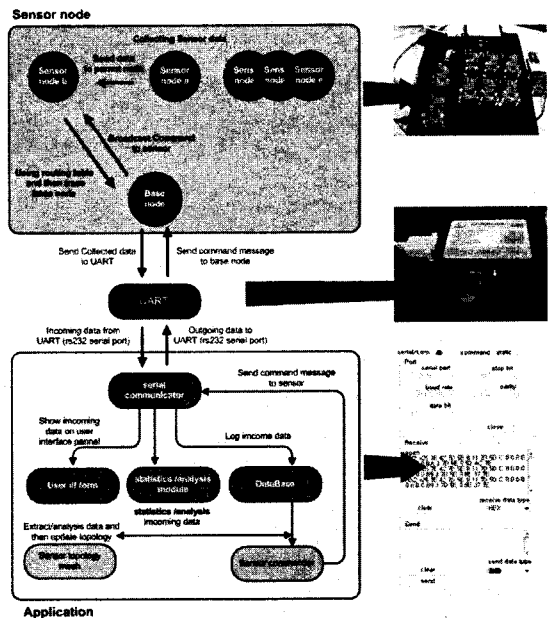
[그림 5] TinyOS 구조

- 1) 작은 크기의 OS (4KB 미만의 실행 이미지)
- 2) NesC 언어로 구현
- 3) Task & Event 기반의 동시성 모델

- 4) 모든 하드웨어 자원은 컴포넌트 형태로 추상화
- 5) 플랫폼이 변경되더라도 컴포넌트만 교체하면 됨
- 6) AM(Active Message)에 의한 패킷 추상화

3. 센서 노드 환경의 설계 및 구현

각 센서 노드의 구성은 [그림 6]과 같다. 센서 노드의 환경은 센서 노드의 생존성- 가용 배터리, 센서 노드의 출력 - 검색 가능 영역, 센서 노드의 통신 경로 - 라우팅 테이블 생성, 센서 노드의 대역폭 -송신 데이터의 크기를 말한다. [11].



[그림 6] 센서 환경 구성도

3.1 센서 노드 데이터 덤프 모듈 설계

3.1.1 커맨드 설계

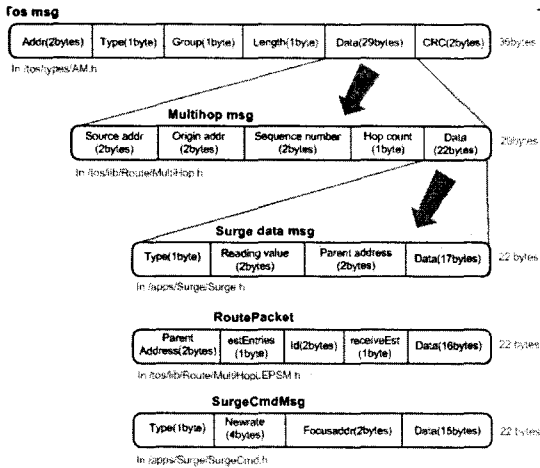
어플리케이션에서 연결된 각 센서노드를 대상으로 명령을 내리기 위함이며 이를 위한 필요한 사항은 다음과 같다.

- 1) 커맨드는 노드가 특정한 명령을 수행하게 한다.
- 2) 화면구성에서 커맨드 설계를 위한 부분은 다음과 같은 사용 패턴을 갖는다.
 - (1) 사용자가 어떠한 타겟을 지정하냐에 따라 이후 사용 가능 행동이 정해진다.
 - (2) 사용되는 타겟에 따라 사전에 사용가능 행동을 정의

한다.

- (3) clear 는 사용자의 모든 작업을 초기화한다.
- (4) 사용자는 일련의 선택순서(타켓-명령-횟수)의 일련 작업을 마치게 되는 시점에서 전송 가능한 문자열이 생성된다.
- (5) 사용자의 명령행위는 통상적인 행위가 아니라 별도의 작업이다. 사용자에게 전송 후 작업을 보존하지 않고 초기화 한다.

3.1.2 메시지 매니저의 설계



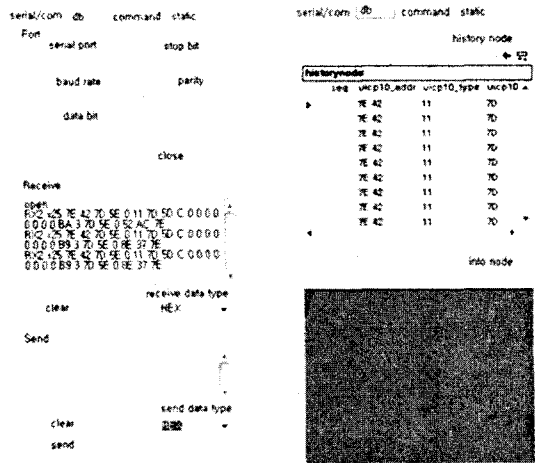
[그림 7] Tos msg

메시지 매니저는 시리얼 통신으로부터의 데이터를 UIIC 메인폼을 통해 건넬받아 이를 몇 개의 배열형태로 구성한다. 메시지 매니저는 다음과 같은 기능을 갖는다. [그림 7].

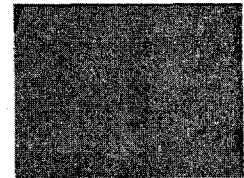
- (1) log 서비스 -> 데이터 원본 저장.
- (2) 토폴로지 서비스 -> 화면에 토폴로지를 그린다.
- (3) 노드 데이터 이력 서비스 -> 노드당 데이터 이력을 보여준다.

3.2 모듈별 기능

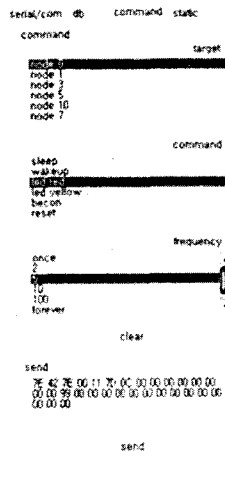
UART는 센서노드의 데이터를 받아 상위 애플리케이션에 전달하게 된다. 이를 위해 UART와 시리얼 통신을 수립하여야 하며 이를 위한 접속 속성값 설정이 필요하다. DB 시리얼 통신을 통해 전달받은 데이터를 DB 에 입력하여 로그를 남긴다. [그림 8](a), (b).



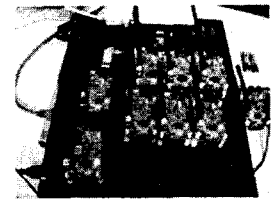
(a) serial communication display



(b) log data details, filter



(a) Command



(b) sensor nodes

[그림 8] 센서 데이터 덤프 모듈 기능

커맨드 부분은 토폴로지를 구성하고 있는 센서 노드에 지정된 동작을 수행하도록 명령을 내린다.

- (1) target : 토폴로지를 구성하고 있는 센서.
- (2) command : 해당 센서가 수행할 동작
- (3) frequency : 동작에 대한 수행 주기

사용자는 토폴로지를 구성하고 있는 센서에 특정한 명령을 지정된 주기만큼 수행시키려 한다. 애플리케이션에서는 선택된 명령을 1차적으로 Tos msg 형태로 변환하며 2차적으로 이를 hex 형태로 변환시키고 이후 전송하게 된다. [그림 8] (c).

[그림 8] (e)은 센서 데이터를 수집하기위해 해당 센싱 칩을 on-board 한 상태의 센서노드이다.

4. 실험 결과 및 고찰

Tos msg 데이터 구조를 통해 UART에 전달되는 데이터 덤프중 일부를 분석한 결과 시리얼 통신을 위해 주고 받는 raw 한 데이터와 센서노드간의 통신을 위한 데이터 처리에 차이가 있음을 알 수 있었다. 이와 같은 메시지 구조를 상위 어플리케이션에서 활용하기 위해서는 메시지 구조에 맞는 프로토콜 설계 구조와 데이터 분석이 필요하다. UART 를 통해 덤프된 데이터 중 일부이다.

```
7E 42 7E 00 11 7D 0C 00 00 05 00 60 01 00 00
3B 02 00 00 11 EC 7E
7E 42 7E 00 11 7D 0C 00 00 00 00 00 00 00 00
D8 00 7E 00 B8 41 7E
```

우선 7D 5D => 7D로 7D 5E => 7E로 변환한 값이다.

데이터를 보면 앞에 7E 42 라는 데이터가 들어온다. 7E는 Zigbee에서 데이터를 수신하면 항상 첫 데이터로 들어오는 값이며, 42라는 값은 패킷에 타입을 정의 하는데 이 부분에는 총 5가지에 값이 들어 갈 수 있다. 그 4가지는 [표 1]과 같다.

[표 1] 수신된 데이터

P_PACKET_NO_ACK(0x42)	유저 패킷을 보냈을 때 ACK신호를 기다리지 않음
P_PACKET_ACK(0x41)	유저 패킷을 보냈을 때 ACK신호를 수신. Receiver는 반드시 P_ACK타입의 데이터를 수신함
P_ACK(0x40)	베이스에 송신하는 ACK신호임
P_UNKNOWN(0xFF)	Unknown Packet Type임

TOS 위의 덤프된 데이터는 [표 2]와같이 분석 할 수 있다. 데이터 구조는 [그림 8] (a)와 같다.

[표 2] 덤프된 데이터

Message Address	Broadcast Address (0xFFFF) UART Address (0x007e)
Message Type	AMTYPE_XUART = 0x00 AMTYPE_MHOP_DEBUG = 0x03 AMTYPE_SURGE_MSG = 0x11 AMTYPE_XSENSOR = 0x32 AMTYPE_XMULTIHOP = 0x33 AMTYPE_MHOP_MSG = 0xFA

Group ID	TinyOS Surge에서는 일반적으로 125번의 그룹아이디를 사용함. (0x7D)
Data Length	이 값이 들어 간 시점부터 몇 바이트의 데이터를 받을지를 확인해주는 데이터임. Data Length는 CRC데이터 숫자까지 포함 함

수신된 덤프 데이터들 중 둘을 선택하여 [그림 32]의 구조에 맞게 분해한 결과는 [표 3], [표 4]와 같다.

[표 3] Tos MSG

ADDR	TYPE	GROUP	LENGTH	DATA	CRC
7E 00	11	7D	0C	00 00 05 00 60 01 00 00 3B 02 00 00	11 E
7E 00	11	7D	0C	00 00 00 00 00 00 00 00 D8 00 7E 00	B8 41

[표 4] Multihop Message

DATA				
Source ADDR	Origin ADDR	SN	HopCounter	DATA (SurMSG)
00 00	05 00	60 01	00	00 3B 02 00 00
00 00	00 00	00 00	00	00 D8 00 7E 00

DATA(SurMSG)에서 가장 처음에 있는 00은 데이터 타입을 말한다. 여기에 들어 갈 수 있는 몇 가지 데이터 타입은 아래와 같다.

- SURGE_TYPE_SENSORREADING (0x00)
- SURGE_TYPE_ROOTBEACON (0x01)
- SURGE_TYPE_SETRATE (0x02)
- SURGE_TYPE_SLEEP (0x03)
- SURGE_TYPE_WAKEUP (0x04)
- SURGE_TYPE_FOCUS (0x05)
- SURGE_TYPE_UNFOCUS (0x06)

0x00을 제외하고는 모두 송신된 데이터임을 확인 할 수 있다.

```
7E 42 7E 00 11 7D 0C 00 00 00 00 00 00 00 00
99 00 00 00 00 00 00 00 00 00 00 00 00 00
```

5. 결론 및 기대효과

유비쿼터스 컴퓨팅 환경에서는 각 디바이스들이 생활의 곳곳에 널리 퍼져 있고 이러한 디바이스들을 통해서 어느 곳에서나 정보의 이동의 용이하다. 유비쿼터스 환경을 위한 핵심기술인 RFID를 통하여 사물의 인식정보 및 주변의 환경 정보까지 센싱하고 이를 연결하여 정보를 관리한다. 유비쿼터스 관련 서비스는 환경 정보 센싱, 자동차 분야, 환경 관리 분야, 물류 유통 분야에 사용 가능하다. RFID 태그와 리더 등으로부터 수신된 정보들에 대한 유용한 서비스를 제공해줄 RFID 미들웨어 기술의 수요가 크게 늘 것이며, 다양한 플랫폼에서 개발이 시도 될 것이다. 다양한 센서 데이터의 특징을 반영한 임베디드(Embedded) 기술과 센서 데이터의 정보 추출이 가능한 미들웨어 시스템은 기존의 미들웨어와 연동이 가능하며 다양한 RFID/USN 환경에서 응용될 수 있다. [1], [2], [10], [11].

각각의 센서 노드는 부착된 배터리에 의해 제한된 양의 전력을 사용할 수 있다. 만일 각각의 센서 노드들이 매우 낮은 주기로 통신하고 통신을 위한 경로가 적절하다면 각각의 노드들은 극히 적은 양의 전력을 소모한다. 이런 가정 하에 센서 노드가 별도의 관리 없이 수 개월 이상 사용 될 수 있는 것이다. 그러나 이는 이상적인 환경만을 고려한 것이다. 가령 예측할 수 없는 극심한 방해 전파와 밀집된 센서 배치로 인한 중복 경로 설정 등의 요인으로 인해 센서 노드의 생존성은 수개월에서 단 하루로 낮아질 수 있다.

센서 노드의 송수신 상태를 분석하여 노드의 생존성 향상과 효율적인 프로토콜 설계를 할 수 있는 센서 데이터를 수집 하였다.

수집된 센서 노드 데이터는 RFID 태그와 리더를 통한 관리 시스템과 재해방지를 위한 다양한 센서를 통한 정보 수집 시스템과 의사결정 시스템에 사용되는 메시지 구조 설계에 사용될 수 있으며 데이터의 자료수집, 센서 분류, 수신율 조절 시스템을 위한 프로토콜 설계 자료로 활용 가능하다.

참고문헌

[1] 김성진 외, 'RFID/USN 산업동향 및 발전보상', 전자통신동향분석 제20권 제3호, 2005.
 [2] 김정훈 외, '센서 네트워크 및 애플리케이션 기술 동향', 전자공학회지, 제32권 7호, 2005.
 [3] 남상엽, 'Mote-Kit를 이용한 무선 센서 네트워크 활용', 상학당, 2005.
 [4] 동북아전자물류센터, 'RFID 기술 교육 세미나', 2005.
 [5] 정보통신부, 'u-센서 네트워크 구축 기본계획', 2004.
 [6] 조위덕 외, '센서 네트워크 기술 개요', KETI, 2003.
 [7] 박승창, '유비쿼터스 모바일 컴퓨팅', 진한도서, 2003.
 [8] 이은곤, 'RFID 확산 추진현황 및 전망', 정보통신정책 제16권 통권 344호, 2004.
 [9] 이재용, '유비쿼터스 센서 네트워킹 기술' TTA 저널 제95호, 2004.
 [10] 한국RFID협회, 'RFID/USN 국내외 적용사례 세미나', 2004.
 [11] Arati Manjeshwar et al, 'TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks', Proc. Second Int'l Workshop', IEEE Proc. Of the Int'l. Parallel and Distributed Processing Symposium, 2002.
 [12] Klaus Finkenzeller, 'RFID HANDBOOK', 영진.com, 2004.
 [13] Ken Sakamura, 'Ubiquitous Computing from Vision to Realit', TRON Project Forum Workshop, 2003.
 [14] Sun microsystems, 'The Sun Global RFID Network Vision', 2004.
 [15] TinyOS Tutorial, <http://www.tinyos.net/>
 [16] <http://www.ukoreaforum.or.kr>