

# 스마트 능동형 RFID 리더를 위한 RM 기반 관리 기능 구현

준회원 권 윤 근\*, 종신회원 정 상 화\*\*°, 정회원 이 윤 성\*\*

## An Implementation of Management Function based on RM for Smart Active RFID Reader

Yoon-Geun Kwon\* Associate Member, Sang-Hwa Chung\*\*° Lifelong Member,  
Yun-Sung Lee\*\* Regular Member

### 요 약

본 논문에서는 능동형 RFID 리더를 위한 EPCglobal Reader Management (RM) 기반 관리기능을 제안하고 구현하였다. RFID 리더의 관리기능을 이용하면 기존의 리더 프로토콜만으로는 불가능했던 RFID 리더의 진단을 구체적으로 할 수 있으며 RFID 리더에 이상이 발생하였을 경우 빠르게 문제를 확인할 수 있다. 또한 제안된 프로세스를 사용하여 단계적으로 진단 할 경우 RFID 리더의 상태 이상을 단계적으로 구체적으로 확인하여 상태 이상의 범위를 추정할 수 있다. 이는 리더의 빠른 복구에 도움을 주어 상태 이상으로 인한 손실을 줄이고 RFID 리더의 QoS를 높일 수 있다. 또한 구현된 RFID 리더의 관리 기능이 성능면에서 LLRP의 관리 메시지와 비교하였을 때 비슷한 크기의 메시지 크기를 보여 부하도 상대적으로 크지 않으며 관리면에서 세분화 된 정보 확인이 가능함을 확인하였다.

**Key Words** : Active RFID, Reader Management, Diagnosis, EPCglobal RM, SNMP

### ABSTRACT

In this paper, we proposed and implemented an EPCglobal Reader Management (RM) based management protocol. Using RFID Reader's management function, we could diagnose an active RFID Reader concretely which was impossible with only using Reader protocols and check the problem quickly when the RFID Reader is in failure. When diagnosing RFID Readers using the proposed RM based management protocol, we can check on their failure in stages and specifically, and can estimate range of failure. This helps RFID Reader's fast recovery, reduce loss of failure and improve QoS. We could get reader information with almost same overhead compared with the management message in LLRP. And we could get more information about RFID Reader's operation status.

### I. 서 론

능동형 RFID 기술은 라디오 주파수(RF)의 특성에

의해 인식거리도 길고 동시에 다수의 태그 인식이 가능하며 데이터 변경과 추가가 자유롭다는 장점을 가지고 있다. 또한 특정 매체가 담고 있는 정보를 자동

\* This research had been proceeded to support 'Development of Global Logistics Information Synchronization Technology for QoS', one of The Industry Original Technology Development Projects led by Ministry of Knowledge Economy.

\*\* 본 연구는 지식경제부 지방기술혁신사업(B0009720) 지원으로 수행되었음.

\* LG전자(yoongeun.kwon@lge.com), \*\* 부산대학교 컴퓨터공학과 임베디드네트워크시스템 연구실(shchung@pusan.ac.kr), (°:교신처자) 논문번호 : KICS2011-03-132, 접수일자 : 2011년 3월 3일, 최종논문접수일자 : 2011년 9월 28일

으로 식별하여 데이터 수집을 목적으로 다양한 활동이 가능하다<sup>11,12</sup>.

능동형 RFID 시스템은 물체를 식별하기 위해 부착되는 태그와 이를 인식하기 위한 능동형 RFID 리더, 다수의 리더로부터 수집된 정보를 처리하는 미들웨어로 구성되며, 다수의 태그와 리더로 구성된 복잡한 RFID 네트워크를 이루게 된다. 이러한 경우 단순한 태그의 인식정보 처리 뿐만 아니라 다수의 리더를 통합적으로 모니터링이 가능한 능동형 RFID 리더 관리 기능을 갖춘 미들웨어가 필수적이다. 예를 들어, 해외의 거점에 능동형 RFID 시스템을 구축한 경우, 국내의 IS(Information Service)로 정보가 들어오지 않을 경우 원인을 분석하고 해외에 연락을 하여 확인 후 고치는 데까지 오랜 시간이 필요하다. 하지만 리더에서 관리 프로토콜을 사용하면 RFID 리더에서 처리된 데이터의 이상을 발견하기 전에 먼저 RFID 리더의 이상 상태를 발견하고 원격으로 복구명령을 전달하여 복구 시간을 단축할 수 있다. 더 나아가 자가 복구까지 가능하도록 설정하면 QoS를 크게 향상시킬 수 있다. 이와 같이 RFID 리더에서 관리 프로토콜은 장치의 이상을 발견하고 복구하여 그 질을 향상시키기 위하여 필요한 기술이다.

ISO/IEC 표준을 제시하는 국제표준협회와 EPCglobal에서는 RF 관련 표준에서 동작 테스트, 리더 프로토콜, SSI(Software System Infrastructure) 등 다양한 계층에 대하여 표준을 제시하고 있다. 하지만 EPCglobal의 RM(Reader Management)를 제외한 대부분의 리더 프로토콜은 리더에서 태그로 명령처리나 리더의 설정 변경, 정보 조회 등의 기능이 대부분이며 리더 프로토콜과 별개로 리더의 상태를 단계적으로 진단할 수 있는 프로토콜이 없다. 따라서 관리 프로토콜로는 EPCglobal에서 제시한 RM만 존재한다고 볼 수 있다.

현재 RM은 수동형 RFID 리더를 대상으로 제시되었고, 능동형 RFID 리더를 대상으로 제시된 것은 존재하지 않는 상황이다. 또한 RFID 서비스 플랫폼을 중심으로 구현된 관리 서비스나 웹서비스를 이용한 관리 연구는 있지만 임베디드 장치 기반의 능동형 RFID 리더를 위한 관리 프로토콜의 연구는 활발히 진행되고 있지 않은 상태이다<sup>13,14</sup>.

그 외에도 현재 표준화 과정이 진행 중인 ISO/IEC 24791, SSI(Software System Infrastructure)의 part 3, device management가 RFID 시스템에서 RFID 리더 장치관리에 대하여 언급하고 있으나 이 또한 EPCglobal의 RM의 장치관리 기법을 중심으로 추가

된 몇가지 관리 방법을 제시하는데 그치고 있다. 따라서 본 논문에서는 RM을 기반으로 능동형 RFID 리더를 위한 관리기능을 설계하였다. 또한 RM에서 사용하는 전송계층으로 XML과 SNMP가 있는데 이 중 네트워크의 부하를 줄이며 MIB의 추가정의로 확장이 용이한 SNMP를 사용하였다<sup>13,14</sup>. 또한 능동형 RFID 리더 관리 기능을 위한 리더 관리 프로토콜을 구현하였으며 SRP(Simple Reader Protocol)<sup>16,17</sup>을 구현하여 리더 프로토콜에서 생성하는 자료를 이용하여 리더의 상태를 제공할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 관련연구에서는 장치관리 프로토콜의 사용 현황과 종류를 분석하였다. 다음으로 능동형 RFID 리더를 위한 RM기반 리더 관리기능 설계 및 구현에서 장치관리기능의 설계 부분과 이를 구현한 것을 설명하며 실험 및 분석에서는 실험을 통하여 구현된 관리프로토콜에 대한 성능과 부하 분석을 하고, 마지막으로 결론 및 향후 과제에 대하여 설명하였다.

## II. 관련 연구

본 논문은 능동형 RFID 리더를 위한 리더 관리기능을 구현하고자 한다. 따라서 먼저 현재의 RFID 시스템에서 리더 관리기능 사용현황을 살펴보고, 관리 프로토콜을 사용할 때 가질 수 있는 이점을 살펴본다. 또한 리더 관리기능을 위하여 어떤 관리 프로토콜이 사용될 수 있는지 살펴본다.

### 2.1 현재 상용리더의 관리 프로토콜 사용 현황

현재 능동형 RFID에서 주요한 위치에 있는 SAVI 사나 GE Security의 경우 RFID Reader에서 RF tag나 Reader의 RF 관련 설정을 위하여 사용되는 리더 프로토콜을 사용자에게 제공한다. SAVI의 능동형 RFID 리더, SR-650의 경우 Air protocol은 ISO 18000-7을 지원한다고 되어 있으나, Reader protocol은 Savi의 자체 protocol인 UDAP(Universal Data Appliance Protocol)를 지원하고 있는 것으로 되어 있다. 또한 GE Security는 리더 프로토콜을 명시하지 않았지만, 두 회사의 경우 리더 프로토콜과는 별개의 관리 프로토콜을 사용한다고 명시하지는 않고 있다<sup>18,19</sup>.

대부분 현재의 상용 RFID 리더의 경우 외부 하드웨어 인터페이스로 시리얼 통신을 지원하고 선택적으로 Ethernet 통신을 지원하고 있다. 이는 아직은 근거리 설치의 대상으로 RFID 리더를 많이 사용하고 있으

며 원격지 RFID 리더를 대상으로 한 관리프로토콜 적용에 대해서는 크게 고려되고 있지 않은 것으로 보인다.

2.2 RFID 리더 오류의 종류 및 복구 방안

먼저 RFID 리더에서 발생할 수 있는 오류의 종류와 증상을 살펴보면 표 1과 같이 구분할 수 있다. RFID 리더에서 다양한 종류의 오류가 발생했을 때, 리더 관리 프로토콜이 존재하지 않는 경우 각각의 오류에 대해 정확하게 대처하기 보다는 프로그램 재실행 및 재부팅을 통한 복구를 하게 된다. 이는 RFID 시스템의 전체적인 성능 저해시키는 요인이 된다.

위의 네가지 상황에 대하여 리더 프로토콜만 적용된 능동형 RFID 리더와, 관리 프로토콜, 리더 프로토

콜이 함께 적용된 능동형 RFID 리더를 이용하여 획득할 수 있는 정보에 어떤 차이가 있는지를 정상동작과 비교하여 확인을 하였다. 결과는 표 2와 같이 확인할 수 있었다. 표에서 확인할 수 있듯이 리더 프로토콜만 탑재한 RFID 리더에서는 오류상황에 대하여 정확히 알 수 없었다. 특히, 리더 프로토콜 프로그램 동작 이상이나 RF 이상의 경우에는 실제 수신된 RFID 태그가 있었어도 잘못된 응답을 하거나, 수신할 수 있는 RFID 태그가 통신 범위에 있어도 이상을 알 수 없이 수신된 태그가 없다고만 알리는 등 정상동작과 구분하기 힘든 잘못된 응답을 보내는 경우도 있다. 하지만 관리 프로토콜이 함께 사용될 경우 관리 프로토콜의 정보를 이용, 이상 부분을 구별할 수 있었다.

2.3 관리 프로토콜의 종류와 비교

관리 프로토콜은 인터넷 등 각종 망을 관리하기 위한 프로토콜에서 시작되었으며, 이후 인터넷 망을 통하여 장치를 관리하는 프로토콜이 제안되었다. 대표적인 망 관리 프로토콜로 CMOT(CMIP(Common Management information Protocol; ISO 9596) Over TCP/IP)와 SNMP(Simple Network Management Protocol)가 있다.

먼저 CMOT는 TCP/IP 상에서 CMIS(Common Management Information Service; ISO 9595)를 구현하고자 한 것으로 UDP와 TCP의 접속을 위한 LPP(Lightweight Presentation Protocol; RFC 1085)를 필요로 한다. CMOT에서는 ACSE(Association Control Service Element; ISO 8649/8650)에서 관리 시스템과 관리되는 시스템 간에 협약될 수 있는 4가지 연계 방식 중 최소한 1가지 이상의 형식을 갖추도록 하고 있다. 아래의 표 3에서는 CMOT에서 사용하는 연계방식을 정리하고 있다.

또한 CMOT 협력시스템에서 하나의 시스템은 관

표 1. 오류의 종류와 증상

| 오류 종류              | 증상   |
|--------------------|--|
| RFID 리더 운영체제 이상    | RFID 리더의 전체적 동작이 불가능한 상태, 초기 기동 불량, 운영체제 system down, kernel panic 등                   |
| 리더 프로토콜 프로그램 동작 중단 | RFID 리더의 운영체제는 가동되나, 여러 가지 이유로 리더 프로토콜 프로그램이 실행되지 않거나 중단/종료 된 상태                       |
| 리더 프로토콜 프로그램 동작 이상 | 프로그램 내부적 이상. 프로그램이 실행되나 정상동작 되지 않는 상태. RF 처리가 정상적이지만, 결과를 처리 못하거나, 잘못된 결과를 회신하는 등의 상태. |
| RFID 리더 RF 이상      | RF 통신 이상. Air Protocol에 맞지 않는 문제, 안테나 이상, 통신 설정(레지스터값) 이상 등                            |

표 2. 각 이상 상태별 복구 방안

| 오류 종류              | 방안   |
|--------------------|--|
| RFID 리더 운영체제 이상    | 초기 기동 불량 → Ramdisk 등의 update   |
| 리더 프로토콜 프로그램 동작 중단 | 단기적 → 프로그램 재실행<br>장기적 → 수정된 프로그램 전송하여 교체   |
| 리더 프로토콜 프로그램 동작 이상 | 프로그램 교체  |
| RFID 리더 RF 이상      | 통신이 불안정적 → 레지스터 값 등을 원격으로 수정<br>통신이 불가 / RSSI 확인 됨 → Air Protocol 준수 여부 확인 / 펌웨어 업데이트<br>통신 완전 불가 → 안테나 이상 여부 확인 / 관련 H/W 확인 |

표 3. COMT 연계방식

| 연계 방식              | 기능   |
|--------------------|--|
| Event              | 관리시스템에서 M-EVENT-REPORTs 전송   |
| Event / Monitor    | Event 방식 + M-GET 기능 추가   |
| Monitor / Control  | 관리시스템이 M-GET, M-SET, M-CREATE, M-DELETE, M-ACTION request 전송.<br>Event report는 허용하지 않음 |
| Full Manager Agent | 위의 모든 기능 제공 (Event, Monitor, Control)  |

리 시스템으로도 피 관리 시스템으로도 역할을 하지만, 동시에 같은 역할을 할 수는 없다. 시스템이 관리 시스템인지, 피 관리 시스템인지를 결정해야 한다.

SNMP의 경우 1990년 표준 규격이 제시되어 널리 이용되기 시작했다. SNMP는 UDP와 IP에 의해 준비되는 비연결형 서비스로 운영이 되기 때문에 지속적인 연결이 아닌 분리 처리 방식으로 정보를 주고받는다. 관리 기능에서는 GetRequest, GetNextRequest, SetRequest 등 3가지의 메시지 형식이 있으며 이 메시지는 Get-Response 형식으로 에이전트에 의해 인식된다. 또한 에이전트는 MIB이나 하부관리 리소스에 영향을 주는 비정상적인 사건을 보고하는데 Trap 메시지를 사용한다. SNMP의 관리 기능은 관리 스테이션이 에이전트에 보내게 되고, 트랩형식은 에이전트가 관리 스테이션에 보내는 방식으로 작동하게 된다. 또한 SNMP에서는 에이전트에서 호스트 MIB 자원에 private 영역을 이용하여 추가적 정보를 제공하도록 확장되었다.

SNMP가 발표된 이후에 기능이 확장 된 SNMPv2가 발표되었다. 보안기능을 강화한 S-SNMP(Secured SNMP)와 성능과 기능을 확장한 SMP(Simple Management Protocol)이 제시되었는데, 성능과 기능이 확장 된 SMP를 중심으로 합쳐져 SNMPv2가 완성되었다.

아래의 표 4는 SNMP와 CMOT를 기능적인 면과 운영적 면에서 서로 비교한 결과를 나타낸 것이다. 먼저 기능적인 면에서 CMOT는 문제의 형태와 문장식 설명 등 SNMP보다 자세한 에러 정보를 제공한다. 그리고 에러 탐지에서 SNMP는 에이전트에서 발생하

는 설정된 trap 에서만 이루어지지만, CMOT는 에이전트나 매니저 모두에서 사건에 기초한 임계나 객체 상태를 발생하는 기능을 제공한다. 장애정보 수집에 있어서 CMOT는 정보를 보내면서도 정보의 수집이 가능하지만, SNMP는 다음 요청을 기다려야 가능하다.

반면 운영적 면에서는 SNMP가 설치 및 운영 비용의 저렴함, 유지보수의 간편함 등에서 CMOT에 비해 유용하다. 특히 CMOT는 OSI의 CMISE, ACSE, ROSE, TCP/IP와 연계를 위한 LPP로 구성된다. 이러한 점은 구성에서 SNMP 대비 2배 이상 복잡하며 더 많은 메모리를 요구하고 API(Application Programming Interface)가 복잡하여 개발 단계에서의 비용 역시 증가하게 된다. 또한 CMOT는 많은 정보를 MIB에서 검색하기 때문에 운영 비용에서 SNMP보다 비효율적이다. 이것은 적은 규모의 스테이션일수록 차이가 크다.<sup>[10]</sup>

본 논문에서는 임베디드 시스템을 기반으로 리더 관리기능을 구현해야 하는 점에서 자원이 제한된 환경에서 더 유리한 SNMP 프로토콜을 이용하여 개발하였다.

### III. 능동형 RFID 리더를 위한 RM 기반 리더 관리기능 설계 및 구현

능동형 RFID 리더를 위한 RM 기반 리더 관리기능을 개발하기 위해 EPCglobal RM 표준에 정의되어 있는 리더 동작 검증 부분과 리더 정보 조회 기능을 구현하였다. EPCglobal RM에서는 객체모델을 이용하여 모니터링 할 수 있는 단위부위를 객체로 표현하고 있으며, Reader Device를 포함하여 몇 개의 객체를 기준으로 아주 다양한 정보를 제공할 수 있도록 하였으나, 본 논문에서는 RFID 리더의 동작 상태에 대하여 정상 동작하고 있는지를 확인하기 위한 기능만 사용하였다.

그림 1은 본 논문에서 사용한 RFID 리더의 구조와 RFID 미들웨어와 연동되는 구조를 표현한 그림이다. RFID 리더 위에는 SNMP와 SRP 모듈이 탑재되어 동작하고 이는 Ethernet을 통하여 미들웨어와 통신하게 된다. RFID 미들웨어에서 SNMP와 SRP 모듈을 이용하여 RFID 리더의 각 모듈에 해당하는 명령을 만들어 Ethernet으로 전달하게 된다. RFID 리더는 Ethernet을 통하여 RFID 미들웨어로부터 전달되는 각 모듈에 해당하는 명령을 수신하게 된다. 특히, SNMP에서는 SNMP에서 필요한 정보를 구조로 유지/제공하기 위하여 MIB(management information base)를

표 4. SNMP와 CMOT의 비교<sup>[10]</sup>

| Management need                   | SNMP      | CMOT |
|-----------------------------------|-----------|------|
| Functional Management             |           |      |
| Fault Management                  | Fair      | Fair |
| Configuration Management          | Fair      | Good |
| Performance Management            | Poor      | Fair |
| Access Control                    | Poor      | Fair |
| Accounting                        | Poor      | Poor |
| Network Planning                  | Fair      | Fair |
| Operational                       |           |      |
| Efficiency and Cost-Effectiveness | Fair      | Poor |
| Personal Related Costs            | Fair      | Poor |
| Implementation                    |           |      |
| Memory Cost                       | Very Good | Poor |
| Processing Cost                   | Good      | Poor |
| Application Cost                  | Good      | Poor |

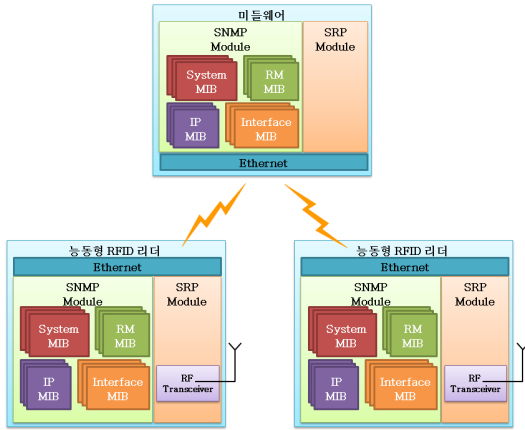


그림 1. SNMP와 SRP가 탑재된 미들웨어와 능동형 RFID 리더의 연동구조

사용한다. 일반적으로 하나의 MIB에는 하나의 객체 아래에 그 객체에 해당하는 세부정보를 포함하고 있는 여러개의 하위 객체가 연결되어 있다. 이와 같은 방법으로 한 분류의 MIB에서는 해당 분류에 대한 다양한 정보(MIB)을 포함하게 된다. SNMP에서는 기본적으로 System MIB, Interface MIB, IP MIB 등 여러 가지 MIB을 제공하며 추가적으로 사용자가 private MIB을 사용할 수 있다. 본 구현에서는 RFID 리더의 정보를 제공하기 위하여 EPCglobal에서 제정한 RM MIB을 사용하였다.

RFID 리더의 동작 상태에 대한 모니터링은 EPCglobal의 RM MIB을 구현하여 get 명령을 통하여 해당 정보를 조회할 수 있도록 제공하였으며 SNMP Trap을 발생하여 이상상태에 대하여 보고할 수 있도록 하였다. 본 논문에서는 SNMP MIB 내용 중 RF 동작의 결과를 확인하기 위하여 RM MIB 중 에서 태그 수집과 태그 쓰기의 실패를 확인할 수 있는 epcgAntRdPntTagsNotIdentified (OID:1.1.1.3.1.1.2)와 epcgAntRdPntWriteFailures (OID:1.1.1.3.1.1.7)를 관찰하였다. 명령이 정상적으로 동작하였을 경우에는 리더 프로토콜을 통하여 성공을 알 수 있으므로 명령이 실패하였을 경우 그 횟수를 더하도록 하여 실패를 확인할 수 있도록 하였다.

태그 명령 모니터링의 경우 epcgAntennaReadPoints (OID: 1.1.1.3) 부분에서 확인 가능하다. 이 부분은 안

표 5. 사용한 RM MIB의 종류

| 모니터링 대상 | 동작 종류 | 해당명령의 RM OID  |
|---------|-------|---------------|
| 태그 명령   | 태그 수집 | 1.1.1.3.1.1.2 |
|         | 태그 쓰기 | 1.1.1.3.1.1.7 |

테나에서 일어나는 동작에 관한 정보를 제공하는 MIB이다. 수동형 RFID 리더의 경우에는 안테나의 인식 거리가 짧고, smoothing이나 filtering 등을 통하여 여러 번 감지된 정보를 가공하여 제공하게 된다. 따라서 개별의 안테나를 하나의 장치로 보고 각 안테나에서 각기 다른 명령을 수행할 수 있으며 각 안테나에서 일어난 각 동작에 대하여 정보 제공이 가능하다. 하지만 능동형 RFID 리더에서는 수신거리가 충분하므로 한 개의 안테나로도 충분한 통신이 가능하다. 따라서 명령 수행 장치를 안테나로 보지 않고 리더 장치로 보게 된다. 따라서 RM MIB의 안테나에서 제공해주는 명령수행 결과의 정보 부분을 능동형 RFID 리더에서 수행되는 명령수행의 결과로 사용하였다.

리더의 동작 상태를 확인하기 위하여 리더의 각 명령에 대한 과정은 그림 2와 같이 정리 하였다. 각 동작을 수행하며 리더의 동작 상태와 결과를 생성하고 이 정보를 SNMP의 MIB 구조를 이용하여 제공한다. 또한 이와 같이 생성된 정보는 SNMP의 기본 MIB과 Reader를 위한 RM MIB 정보를 이용하여 그림 3과 같이 상태 이상을 추정하고 어디 상태가 문제가 있는지 확인할 수 있는 자료가 될 수 있다. 이를 위하여 EPCglobal SMI(Structure of Management Information) MIB(RM MIB)을 작성하였으며 SNMP 개발 도구의 MIB2C를 이용하여 이를 C코드로 변환하여 동작을 작성하였다.

리더의 상태 이상은 네트워크 상태 이상, 리더 동작 이상, 리더의 RF 부분 이상으로 3가지로 구분하였으며 SNMP가 네트워크를 기반으로 작동하므로 네트워크 상태 이상은 Ping 명령 등 네트워크를 확인하는 명

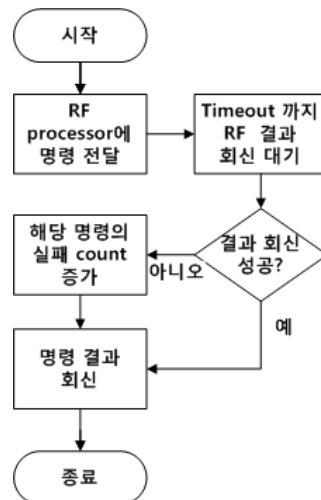


그림 2. 리더의 동작 절차

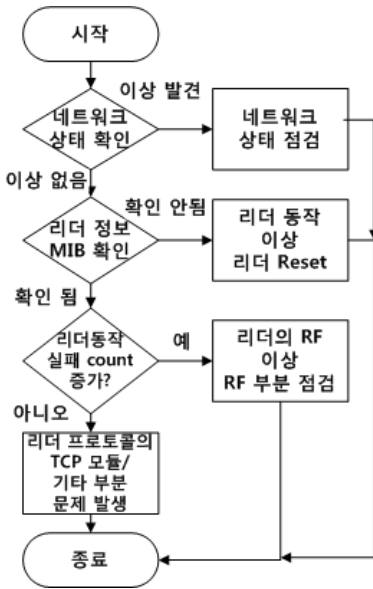


그림 3. 리더 이상 점검 절차

령을 통하여 먼저 수행되어야 한다. 그 이후 SNMP를 통하여 System Group MIB이나 RM MIB을 조회하는 방법으로 각 단계에 대하여 상태 이상을 점검할 수 있다.

#### IV. 실험 및 분석

##### 4.1 개발환경

본 논문에서는 부산대학교 임베디드 네트워크 시스템 연구실에서 개발한 433MHz 능동형 RFID 고정형 리더를 사용하였다. 그림 4는 능동형 RFID 고정형 리더의 시스템 구조도를 나타낸 것으로 능동형 RFID 리더는 ARM9 프로세서를 탑재한 임베디드 시스템 보드를 기반으로 개발되었으며 임베디드 리눅스 커널이 탑재되어 있다. 능동형 RFID 리더는 Main Processor 보드와 433MHz RF 모듈을 제어하는 RF Processor 보드로 구성되어 있으며 RM 기반 리더 관리 기능은 임베디드 리눅스 커널을 통해 구동된다. 태그의 경우

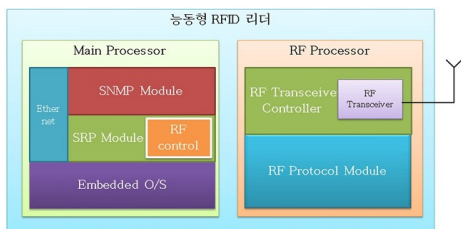


그림 4. 능동형 RFID 리더의 시스템 구조도

ATmega128 MCU를 탑재한 433MHz RFID 태그를 사용하였다. S/W 개발환경은 Ubuntu 10.04 LTS 버전이 설치된 Desktop에 Arm-linux-gcc 2.95.3을 이용하여 개발하였다. 또한 SNMP는 NET-SNMP 5.6 버전을 이용하였고, SNMP에서 개발 과정에서 MIB2C라는 MIB 구조체를 C 코드로 변환해주는 모듈에서 Perl이 필요하여 Perl(버전 5.8.8)과 Perl 라이브러리(libperl-dev)를 설치하였다.

실험에는 433MHz 능동형 RFID 고정형 리더와 총 10개의 433MHz 능동형 RFID 태그가 사용되었다. 호스트 시스템으로는 Windows 7 운영체제가 설치된 데스크톱을 이용하였으며, 리더는 UART 인터페이스를 통하여 모니터링하고, 호스트 프로그램과 리더는 TCP/IP 소켓통신을 통하여 호스트 프로그램으로부터 명령을 전송하였다. SRP 명령을 전달하는 호스트 프로그램은 MFC를 이용하여 구현하였다. SNMP 메시지를 확인하는 프로그램으로 iREASONING의 MIB Browser를 사용하였다.

실험은 크게 두가지로 먼저 SNMP를 이용하여 RFID 리더의 정보를 조회하여 리더의 상태를 판단할 수 있음을 보였다. 또한 SNMP 메시지의 크기를 packet sniffer를 이용하여 측정하여 LLRP 리더 프로토콜의 Reader 정보와 상태를 조회할 수 있는 메시지 와 비교하였다.

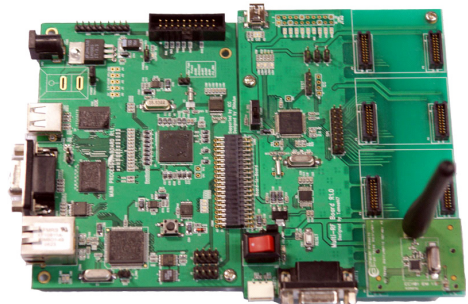


그림 5. 실험에 사용된 RFID 리더 사진

##### 4.2 능동형 RFID 리더의 관리기능 동작 실험

첫번째 실험은 SRP 호스트 프로그램에서 능동형 RFID 리더에 SRP 형식의 명령을 전달하며 그 동작을 확인하였다. 그림 3의 절차에 이용할 수 있도록 시험 대상은 리더의 정보 MIB, 태그명령 결과 MIB, 리더 동작 상태 MIB을 여러 상태에서 시험하였다.

먼저 리더의 정보 MIB을 호출하였다. SNMP에는 정보를 가져오는 방법이 get과 walk의 두가지로 분류할 수 있다. 그 중 get은 단일항목에 대한 정보를 가져

오는 명령이며, walk는 해당 종류의 하위 종류에 대한 정보를 모두 가져오는 명령이다. 본 리더의 정보를 가져오는 시험에서는 walk 명령을 이용하여 RM MIB의 ReaderDevice 정보를 RFID 리더에 조회를 하여 그림 6과 같이 정보를 수집하는 것을 확인할 수 있었다.

관리기능을 확인하기 위한 두번째 실험으로서 RFID 리더의 동작을 고의적으로 실패시켜 timeout 을 유발하여 개별 동작에 대한 실패 횟수가 정확하게 기록이 되는지 확인 하였다. 태그 수집과 태그 쓰기에 대하여 실험을 하였으며, 태그 수집에는 RFID 태그의 수집과정에서 sleep 과정에 들어가지 않게 조작한 태그를 이용하여 리더가 태그 수집 완료할 수 없도록 조작하였다. 태그 쓰기에 대해서는 태그의 코드를 수정하여 태그의 데이터를 회신하기 전 2초의 대기시간이 발생하도록 태그를 수정하였다. 태그가 명령을 받고 회신하기 전에 태그의 전원을 차단하여 태그의 회신을 받는 과정에서 timeout을 유발하여 쓰기 과정에서 실패가 일어나는 상황을 만들어 정상적으로 실패 횟수를 기록하는지를 확인 하였다.

그림 7과 그림 8은 각각 태그수집 실패 횟수와 태그쓰기 실패 횟수를 조회한 것이다. 각각 10회에 걸쳐 실험을 진행하였는데 실패한 10회 모두 정상적으로 값을 기록한 것을 확인할 수 있다. 이와 같이 SNMP와 RM MIB을 RFID 리더에 구현하여 리더 관리 프로토콜로 적용을 하였으며 리더의 동작 관련 정보에 대하여 외부에서 확인이 가능함을 확인 하였다. 또한 리더 장치의 정보와 동작 실패 횟수를 각각 조회하여 리더의 정보를 단계적으로 조회함으로써, 기존의 RFID 리더에서 리더의 상태 확인을 리더 프로토콜에

| Name/OID                        | Value                            |
|---------------------------------|----------------------------------|
| epcgRdrDevDescription.0         | support ISO24791 standard reader |
| epcgRdrDevRole.0                | Active RFID reader               |
| epcgRdrDevEpc.0                 | urn:epc:id:gid:1.222.3           |
| epcgRdrDevSerialNumber.0        | ens001                           |
| epcgRdrDevTimeUtc.0             | 1296622178                       |
| epcgRdrDevReboot.0              | 0                                |
| epcgRdrDevResetStatistics.0     | 0                                |
| epcgRdrDevResetTimestamp.0      |                                  |
| epcgRdrDevNormalizePowerLevel.0 | 0                                |
| epcgRdrDevNormalizeNoiseLevel.0 | 0                                |
| epcgRdrDevOperStatus.0          | up                               |
| epcgRdrDevOperStateEnable.0     | 0                                |

그림 6. SNMP에 의해 수집된 RFID 리더의 정보

| Name/OID                        | Value |
|---------------------------------|-------|
| epcgAntRdPntTagsIdentified.0    | 0     |
| epcgAntRdPntTagsNotIdentified.0 | 0     |

| Name/OID                        | Value |
|---------------------------------|-------|
| epcgAntRdPntTagsIdentified.0    | 0     |
| epcgAntRdPntTagsNotIdentified.0 | 10    |

그림 7. 태그수집 실패 확인 실험 결과

| Name/OID                               | Value |
|--|-------|
| epcgAntRdPntWriteOperations.0          | 0     |
| epcgAntRdPntWriteFailures.0            | 0     |
| epcgAntRdPntWriteFailuresNotifEnable.0 | 0     |
| epcgAntRdPntWriteFailuresNotifLevel.0  | error |

| Name/OID                               | Value |
|--|-------|
| epcgAntRdPntWriteOperations.0          | 0     |
| epcgAntRdPntWriteFailures.0            | 10    |
| epcgAntRdPntWriteFailuresNotifEnable.0 | 0     |
| epcgAntRdPntWriteFailuresNotifLevel.0  | error |

그림 8. 태그 쓰기 실패 확인 실험 결과

의존하던 것에서 벗어나 리더의 동작 상태를 단계적으로 점검할 수 있음을 확인하였다.

### 4.3 능동형 RFID 리더의 관리기능 성능 비교

구현된 관리기능의 성능을 확인하기 위하여 기존의 리더 프로토콜의 관리 기능과 작동 부하를 비교하였다. 실험은 두가지로 먼저 관리 프로토콜의 성능을 확인하기 위하여 본 논문에서 구현한 관리 기능을 탑재한 RFID 리더와 관리기능이 없이 리더 프로토콜을 이용하여 상태를 알 수 있는 RFID 리더를 비교하였다. 그리고 관리기능의 부하의 정도를 확인하기 위하여 본 논문에서 구현한 RFID 리더와 기존의 리더 프로토콜을 이용한 RFID 리더의 데이터를 수집하고 분석하여 크기를 비교하였다.

먼저, 성능을 비교하기 위하여 다양한 리더의 오동작 환경을 설정하고, 리더 프로토콜을 이용하는 리더와 관리프로토콜을 적용한 리더를 비교하였다. 오동작 상황은 이상이 발생할 수 있는 계층을 고려하여 네트워크 이상, 리더 운영체제 이상, 리더 프로토콜 프로세서 이상, 리더 프로토콜의 오동작, 리더 프로토콜의 RF 이상 등의 단계로 나눌 수 있다. 이 중, 리더 프로토콜이나 관리 프로토콜이 동작할 수 있는 상태에서

표 6. 각 이상 상태별 RP / SNMP 확인 방법

| 오류 종류              | 리더 프로토콜      | 관리 프로토콜 + 리더 프로토콜  |
|--------------------|--------------|--|
| RFID 리더 운영체제 이상    | No response  | No response  |
| 리더 프로토콜 프로그램 동작 중단 | No response  | RP no Response<br>SNMP Response                          |
| 리더 프로토콜 프로그램 동작 이상 | bad response | RF 수신 count 등 확인하여 이상 확인 (동작 성공 count는 증가, RP 결과는 식별 없음) |
| RFID 리더 RF 이상      | bad response | RF 수신관련 결과 없고, RSSI 값 아주 좋지 않음 확인                        |

비교하기 위하여 네트워크 이상을 제외한 과정에 대하여 확인하였다.

다음으로 관리 프로토콜의 부하를 비교하였다. 실험 방법은 WALK 명령을 이용하여 다수의 SNMP 데이터를 수집하고 분석하여 패킷 오버헤드를 제외한 실제 데이터를 위한 패킷의 크기를 분석하여 LLRP의 Get Reader Capability Response 메시지와 크기를 비교하였다.

또한 수신된 데이터 중 그림 9와 같이 실제 SNMP 개별 데이터를 확인 하였다. sysUpTime 항목에 대하여 실제 데이터를 확인, 분석 하였는데, 우선 그림 10과 같이 sysUpTime의 OID와 그 값이 전달되는 것을 확인할 수 있었다. 또한 전체 Packet 88 Byte 중 41Byte가 SNMP의 실제 내용을 포함하고 있는 데이터 크기인 것을 확인할 수 있었다.

그림 11의 상단은 LLRP의 Binary 형식의 Get Reader Capability Response 메시지 형태이다. 그림에서 표시된 것 중 (0-1)로 표기된 것은 선택적으로 사용 가능한 부분이다. 리더의 상태를 알기 위하여 General Device Capability Parameter까지 조회하게 되면 UTF-8 String을 제외한 총 데이터의 길이가 40Byte로서, LLRP로 리더의 정보를 조회하는데 최소 41Byte 이상이 필요하게 된다. 이를 통하여 SNMP와 LLRP 메시지 데이터를 비교분석 해본 결과 LLRP 등 대부분 리더 프로토콜의 메시지는 한번에 그 내용 전체를 보내지만 SNMP의 메시지는 필요한 부분만 선택적으로 개별로 전송되어 나머지 필요하지 않은 데

그림 9. SNMP walk 실행 결과 packet

그림 10. SNMP 데이터(sysUpTime)

|   |   |   |   |   |   |   |   |   |   |    |                        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11                     | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| Message Length [15:0]                       |   |   |   |   |   |   |   |   |   |    | Message Length [31:16] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Message ID[15:0]                            |   |   |   |   |   |   |   |   |   |    | Message ID[31:16]      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LLRPStatus Parameter                        |   |   |   |   |   |   |   |   |   |    |                        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| GeneralDeviceCapabilities Parameter (0-1)   |   |   |   |   |   |   |   |   |   |    |                        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| LLRPCapabilities Parameter (0-1)            |   |   |   |   |   |   |   |   |   |    |                        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| RegulatoryCapabilities Parameter (0-1)      |   |   |   |   |   |   |   |   |   |    |                        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| AirProtocolLLRPCapabilities Parameter (0-1) |   |   |   |   |   |   |   |   |   |    |                        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Custom Parameter (0-n)                      |   |   |   |   |   |   |   |   |   |    |                        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

|   |   |   |   |   |   |   |   |   |   |    |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11  | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| Message Length                            |   |   |   |   |   |   |   |   |   |    | Message Length                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Message ID                                |   |   |   |   |   |   |   |   |   |    | Message ID                                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Message Length                            |   |   |   |   |   |   |   |   |   |    | Message Length                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Error Description Byte Count              |   |   |   |   |   |   |   |   |   |    | Error Description (UTF-8 String)          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Error Description (UTF-8 String)          |   |   |   |   |   |   |   |   |   |    | Error Description (UTF-8 String)          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Message Length                            |   |   |   |   |   |   |   |   |   |    | Message Length                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Rsvd                                      |   |   |   |   |   |   |   |   |   |    | Rsvd                                      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Device Manufacturer name                  |   |   |   |   |   |   |   |   |   |    | Device Manufacturer name                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Model Name                                |   |   |   |   |   |   |   |   |   |    | Model Name                                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Firmware Version (UTF-8 String)           |   |   |   |   |   |   |   |   |   |    | Firmware Version (UTF-8 String)           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Receive Sensitivity Table Entry Parameter |   |   |   |   |   |   |   |   |   |    | Receive Sensitivity Table Entry Parameter |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

그림 11. LLRP Get Reader Capability Response 메시지 형식(binary)

이터를 제외하고 선택적으로 데이터를 요청할 수 있는 이점이 있다. 데이터 전송량을 비교하기 위해서 논문에서 구현한 RM 기반 관리 메시지를 EPCglobal의 LLRP 메시지와 비교를 하였다. LLRP에서는 XML 형식과 Binary 형식이 있는데 둘 중 메시지 크기가 작은 Binary 형태와 비교하여도 큰 차이가 없는 데이터 크기로 데이터를 조회하여 읽어 올 수 있음을 확인할 수 있었다.

V. 결론

본 논문에서는 리더 프로토콜에 의존한 리더 관리에서 리더 장치에 문제가 발생하였을 경우 그 관리와 원인 분석이 되지 않는 문제를 지적하였다. 해결방안으로 관리프로토콜을 적용함으로써 인하여 리더 프로토콜이 분석할 수 없었던 RFID 리더 동작 실패에 대한 원인을 단계적으로 확인할 수 있음을 제시하고 리더 장치의 상태 확인과 동작 실패에 대한 리더 정보를 관리프로토콜로 구현, 검증하였다. 관리 프로토콜을 이용하여 리더의 이상상태 정보를 구체적으로 확인할 수 있음을 확인하였다. 또한 Binary 형식의 리더프로토콜의 정보조회 명령과 크기를 비교하였을 때 비슷한 크기를 가져옴으로서 개별 정보를 조회함에 있어 통신상에 성능차이가 없음을 보였다.

하지만 관리프로토콜이 네트워크기반으로 장치 관리를 하고 있으므로 네트워크가 단절된 상태에서 관리나 복구의 한계가 여전히 존재한다. 또한 리더프로토콜로 단일 조회를 전달할 때는 Binary 형태의 다른 프로토콜과 유사한 크기지만, WALK 명령을 이용하



여 다수의 정보에 대한 조회를 할 때는 전송 데이터의 양이 훨씬 많은 것을 알 수 있다. 관리 프로토콜을 사용할 경우 사용하지 않았을 경우에는 구체적으로 알 수 없었던 RFID 리더의 상태 정보에 대하여 정확하게 파악할 수 있었다. 이는 RFID 리더의 문제가 있는 부분의 정보를 구체화 할 수 있어 QoS를 높이는데 기여할 수 있음을 확인할 수 있었다.

향후 연구로는 네트워크가 단절된 상태에서 장치의 자가진단과 장치의 복구를 하는 프로세스에 대한 연구와, 본 논문에서 사용된 리더 관리 프로토콜보다 더 적은 부하로 장치의 정보를 확인할 수 있는 방법에 대한 연구가 필요하다.

### 참 고 문 헌

- [1] 박정현, “RFID 기술 수준과 도입 사례”, 전자통신동향분석, 제21권, 제3호, pp.137-146, June 2006.
- [2] 오세원, 표철식, 채종석, “RFID 표준화 및 기술 동향”, 전자통신동향분석, 제20권, 제3호, pp.56-66, June 2005.
- [3] ISO/IEC 24791, <http://iso.org>
- [4] V Prabakar, Dr. BV Kumar, SV Subrahmanya “Management of RFID-centric business networks using Web Services”, Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), p.133, Feb 2006.
- [5] Wang Minli, Wang Gang, He Dajun, “Device Management in RFID Public Service Platform”, Third 2008 International Conference on Convergence and Hybrid Information Technology, pp. 1133-1136, Nov 2008.
- [6] Se Won Oh, Hyochan Bang, Jae Gak Hwang, “Light-weight RFID Device Interface for Controlling RFID Tag Memory Access”, The 12th International Conference on Advanced Communication Technology (ICACT 2010), pp. 1516-1521, Feb 2010.
- [7] Simple RFID Reader Protocol for Controlling RFID Tag Memory, TTA.KO-06.0220, TTA, 2009.
- [8] Savi, <http://www.savi.com/products/rfid/>
- [9] GE security, <http://www.gesecurity.com/csd>

- [10] 우동진, 정병남, 오상수, 남명희, “망 관리 기술 동향”, 전자통신동향분석, 제9권, 제4호, pp. 99-112, Jan 1995.
- [11] EPCglobal, <http://www.epcglobalinc.org/standards>

권 윤 근 (Yoon-Geun Kwon)

준회원



2009년 2월 부산대학교 컴퓨터 공학과 졸업  
 2011년 2월 부산대학교 컴퓨터 공학과 석사  
 2011년 3월~현재 LG전자 연구원  
 <관심분야> Android, Multimedia Framework, RFID System, System Diagnosis

정 상 화 (Sang-Hwa Chung)

종신회원



1985년 2월 서울대학교 전기 공학과 학사  
 1988년 5월 Iowa State Univ. 컴퓨터공학과 석사  
 1993년 8월 Univ. of Southern California 컴퓨터공학과 박사  
 1993년~1994년 Univ. of Central Florida 컴퓨터공학과 조교수  
 1994년~현재 부산대학교 컴퓨터공학과 교수, 컴퓨터 및 정보통신연구소 연구원  
 2002년~2003년 Oregon State Univ. 컴퓨터공학과 초빙교수  
 <관심분야> 클러스터 시스템, TOE, RDMA, RFID, 무선랜 메쉬 네트워크

이 윤 성 (Yun-Sung Lee)

정회원



2006년 2월 부산대학교 컴퓨터 공학과 졸업  
 2008년 2월 부산대학교 컴퓨터 공학과 석사  
 2008년 3월~현재 부산대학교 컴퓨터공학과 박사과정  
 <관심분야> RFID, 무선랜 메쉬네트워크, 센서네트워크