

---

# 다중인터페이스 다중채널 2.4GHz 능동형 RFID 리더 프로토콜 구현방법에 관한 연구

김동현\* · 이채석\* · 김종덕\*\*

A Study about Implementation Method of Multi-Interface Multi-Channel 2.4GHz Active  
RFID Reader Protocol

Dong-Hyun Kim\* · Chae-Suk Lee\* · Jong-Doek Kim\*\*

---

이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된  
기초연구사업임(No. KRF-2008-D00490)

---

## 요 약

다중인터페이스, 다중채널 2.4GHz 능동형 RFID 시스템에서 리더가 태그들을 수집할 때, 특정 인터페이스로 태그가 집중화되는 문제가 발생한다. 이러한 문제를 해결하기 위해 인터페이스 간 부하 분산 프로토콜인 LP-결합과 AP-균형 프로토콜을 설계하고 시뮬레이션을 통하여 성능의 우수성을 보였다. 설계된 프로토콜들을 펌웨어레벨의 하드웨어에서 구현하기에는 세 가지 문제를 가진다. 첫째, 태그가 랜덤하게 리더의 채널을 선택하고, 리더가 태그의 채널을 변경시킬 수 있는 방법이 필요하다. 둘째, 리더와 태그 간 동기 문제이다. 셋째, 하나의 MCU가 2개의 인터페이스를 동시에 동작시키는 문제이다. 이 같은 문제를 해결하기 위해 태그 채널 변경을 위한 메시지 설계 및 태그 채널 변경 방법과 리더와 태그 간 동기를 맞추기 위한 프로토콜을 구현했다. 그리고 실험을 통해 프로토콜의 성능을 비교 분석 하였다. LP윈도우크기가 같을 경우, 부하분산을 통해 충돌 확률을 낮게 하는 LP-결합, AP-균형의 성능이 단일인터페이스보다 우수했다.

## ABSTRACT

When reader collect tags, we found that they tend to get together to specific interface in Multi-Interface Multi-Channel 2.4GHz Active RFID system. To solve this problem, we designed the LP-Combine and AP-Balanced protocol for load distribution between interfaces, then verified its superiority of the performance through the simulation. There are three problems to implement designed protocols in hardware of firmware-level. first, tag selects randomly the channel of reader and reader need the method which can change the channel of tags. second, reader has the synchronization problem between reader and tag. third, reader has problem that MCU of reader have to operate simultaneously dual interface. To solve this problems, we designed the message and implemented method for tag channel change and the protocol in order to adjust synchronization between reader and tag. Therefore, we compared and analyzed the performance of protocols by experiment. If LP windows size is same, the performance of LP-Combined protocol and AP-Balanced protocol which lower collision probability by its load distribution is more outstanding than single interface protocol performance.

## 키워드

2.4GHz 능동형 RFID, 다중인터페이스, 다중채널, 리더 프로토콜, 부하분배

## Key word

2.4GHz Active RFID, Multi-Interface, Multi-Channel, Reader Protocol, Load-Balancing

---

\* 부산대학교 컴퓨터공학과  
\*\* 부산대학교 컴퓨터공학과 부교수 (교신저자)

접수일자 : 2009. 11. 13  
심사완료일자 : 2010. 01. 02

## I. 서 론

RFID(Radio Frequency IDentification)는 무선을 사용하여 사물을 인식하는 통신기술로 다양한 분야에서 폭넓게 응용된다. RFID시스템은 정보를 수집하는 리더와 정보를 전달하는 태그로 구성된다. 태그는 전파 전파 능력에 따라, 수동형, 반수동형, 능동형으로 구분한다. 리더로부터 전파전파가 있어야만 Backscattering을 통해 자신의 ID를 리더에게 전달하는 수동형(Passive)태그와 배터리를 장착하여 내부 회로의 구동전원으로 활용하나 역시 Backscattering을 통해서만 자신의 ID를 리더에게 전달하는 반-수동형(Semi-passive)태그 그리고 배터리를 장착하여 내부 회로의 구동전원으로 활용하면서 Backscattering없이 ID뿐만 아니라 센싱정보와 같은 부가정보 전송이 가능한 것이 능동형(Active)태그이다.[1] 능동형태그는 자체 배터리를 가지고 있기 때문에 수동태그에 비해 원거리 인식이 가능한 장점을 가진다. 게다가 능동형 태그는 자신의 ID외에 센싱정보의 전송이 가능하기 때문에 컨테이너 터미널과 같은 넓은 지역에서 대량의 컨테이너에 대해 내부의 센싱정보 등을 원거리에서 인식하는 것이 가능하다. 대량의 컨테이너에 대한 선적, 하역 및 게이트 출입뿐만 아니라 컨테이너 내부의 정보를 인식하는 것이 가능하게 될 것이다. 이 같은 능동형 RFID를 항만 물류 환경에 적용하기 위한 많은 연구 개발이 이루어지고 있다.

능동형 RFID의 표준으로는 ISO/IEC 18000-7이 있는데 이 표준은 433.92MHz를 사용하는 단일채널 시스템으로 데이터의 전송속도는 27.8Kbps이다. 컨테이너 야드와 같이 많은 개수의 컨테이너에 부착된 태그를 고속으로 수집하기에는 ISO/IEC 18000-7과 같은 단일채널 시스템은 한계를 가질 수밖에 없고 컨테이너 내부의 대용량의 정보를 전송하기에 200KHz의 채널 대역폭 및 27.8Kbps의 전송속도는 문제점을 발생한다. 이러한 문제점을 해결하기 위한 하나의 방법으로 다수의 채널을 사용할 수 있는 2.4GHz대역기술이 있다. RF트랜시버에 따라 차이를 보이겠지만 하나의 채널 대역을 ISO/IEC 18000-7표준의 5배인 1Mhz로 정하더라도 100여개의 채널을 사용할 수 있고 대역폭의 증대에 따른 전송속도 또한 증가한다.

우리는 다수의 채널과 2개의 인터페이스를 사용하는 2.4GHz 능동형 RFID리더를 사용하여 성능을 향상시키고자 한다. 다중 채널 2.4GHz 능동형 RFID 리더에서 만약 인터페이스가 하나라면 특정시점에 하나의 채널만을 사용하여 태그를 수집해야 한다. 그러면 다중 채널 사용에 대한 성능향상은 보이기 어렵다. 리더가 다수의 인터페이스를 사용한다 하더라도 만약 태그들이 고정된 채널을 사용한다면 특정 채널을 가지는 태그의 분포에 따라 리더가 태그들을 수집하는 시간은 결정된다. 태그가 랜덤하게 인터페이스의 채널을 선택할 수 있다하더라도 만약 특정 인터페이스로 집중화되면 다수의 인터페이스를 사용하는 것에 대한 성능기대는 어렵다. 본 연구진은 이 같은 문제를 해결하기 위해 다중인터페이스, 다중채널 2.4GHz 능동형 RFID 리더 프로토콜을 설계하고 시뮬레이션을 통하여 성능의 우수성을 보였다[2].

우리가 사용하는 다중인터페이스 다중채널 2.4GHz 능동형 RFID 리더는 하나의 ARM 7 MCU에 ChipCon사의 CC2500 2.4GHz RF트랜시버 2개를 사용한다. 펌웨어 레벨의 장비에서 설계된 프로토콜을 구현하기 위해서는 세 가지 문제를 해결해야 한다. 첫째, 태그들이 스스로 채널을 선택하고, 리더가 태그의 채널을 변경시킬 수 있어야 한다. 둘째, 리더와 태그 간 동기화가 이루어져야 한다. 셋째, 하나의 MCU에 의해 두 개의 인터페이스가 동시에 동작해야 한다. 이 같은 문제를 해결하기 위해 태그의 채널을 변경시키기 위한 메시지 포맷을 설계하고 구현했다.

리더와 태그 간 동기화를 맞추기 위한 동기화 지연을 고려하여 리더 프로토콜을 구현했다. 하나의 MCU가 2개의 인터페이스를 동시에 동작시키게 하기위해 각 인터페이스에서 수신완료에 대한 응답으로 변경되는 Flag값을 Polling방식으로 확인한다. 슬롯의 길이는 전송지연과 데이터인식시간 및 인식률을 확인하여 결정했다. 슬롯의 길이와 윈도우의 크기가 전체 인식시간에 영향을 미치기 때문에 최적의 슬롯크기와 윈도우크기가 필요하다. 최적의 윈도우크기를 결정하기 위해서는 태그 개수 추정이 선행되어야 한다. 태그 개수 추정을 하기 위해서는 인식이 안 된 슬롯이 빈 것인지 충돌에 의한 것인지를 알아야 하는데 현재는 확인하기 어렵다. 동적 윈도우크기 설정은 다음 연구과제로 남기겠다.

본 논문의 구성은 다음과 같다. II장에서는 관련연구로 ISO/IEC 18000-7 능동형 RFID 표준에서의 태그 인식과정과 설계된 LP-결합프로토콜 및 AP-균형프로토콜에서의 태그 인식과정을 설명하고, III장에서는 2.4GHz 능동형 RFID 리더 프로토콜 구현하는데 필요한 환경 및 구현방법에 대하여 설명하고 IV장에서는 구현된 2.4GHz 능동형 RFID 리더 프로토콜의 성능측정을 위한 실험환경 및 실험방법에 대한 설명과, V장에서는 실험을 통한 결과를 바탕으로 성능평가 후, VI장에서는 결론 및 향후연구계획을 언급하며 끝을 맺겠다.

## II. 관련연구

### 1. ISO/IEC 18000-7인식과정

그림1은 ISO/IEC 18000-7 능동형 RFID 표준의 인식과정을 나타낸 것이다. 리더는 sleep모드인 태그를 깨우기 위해 WP(Wake up Period)에 wake up 신호를 브로드캐스트 한다. 이 wake up 신호를 전송 받은 태그는 인식될 준비를 한다. WP가 끝나면 CP(Collection Period)가 반복되는데 이 CP는 세부적으로 SP(Synchronization), LP(Listen Period), AP(Acknowledge Period)로 구성되며 전체 태그들이 모두 인식될 때까지 이 과정을 반복한다. SP는 "Collection Command"를 브로드캐스트하는 구간이다. 이 명령에는 리더가 인식해야하는 태그들을 분류하는 과정도 포함한다.

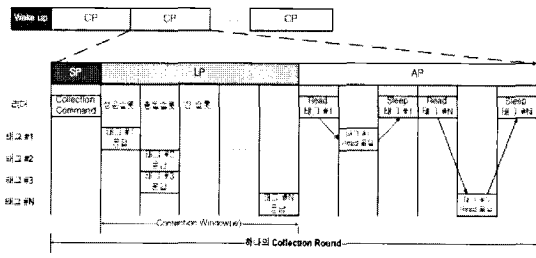


그림 1. ISO/IEC 18000-7 능동형 RFID 표준의 인식과정

Fig. 1. Collection sequence of ISO/IEC 18000-7 Active RFID standard

특정태그들만을 인식해야 될 때 이 명령어에 그 내용을 포함하여 수집명령을 전송함으로써 특정 태그만을 인식할 수 있다. 또한 이어지는 LP에서 사용할 윈도우 크기  $\omega$ 를 포함한다. LP구간은 slotted aloha 방식으로 태그 간 충돌을 회피하면서 태그가 리더에게 자신의 ID를 전송하는 구간이다. AP구간은 LP구간에서 성공적으로 자신의 ID를 전송한 태그에 대해 태그가 가지고 있는 부가정보, 즉 센싱정보나 사용자 정보 등을 인식시키고 태그를 다시 절전모드로 변경시키는 구간이다. LP구간에서 자신의 ID를 전달하지 못한 태그들이나 AP구간에서 사용자 정보나 센싱정보를 전달하지 못한 태그들은 다음 라운드에 다시 참여하여 인식이 성공할 때까지 이 과정을 반복한다[3].

현재 능동형 RFID에 대한 표준은 ISO/IEC 18000-7이 유일하다. 그래서 다중인터페이스, 다중채널 2.4GHz 능동형 RFID의 리더와 태그 간 인식과정은 ISO/IEC 18000-7의 인식과정을 따르도록 하겠다.

### 2. 다중인터페이스, 다중채널 2.4GHz 능동형 RFID 리더 프로토콜

#### 2.1 LP-결합 프로토콜

그림2는 각 인터페이스에 균등하게 태그들이 분포하게 설계된 LP-결합 프로토콜에 대한 동작과정을 설명한다. 먼저 각 인터페이스의 SP, LP, AP구간은 동기 방식으로 동작한다. 리더는 SP구간에 전송하는 "Collection Command"에 두 인터페이스의 정보를 모두 전송한다. 그것을 인식한 태그는 인터페이스 중에 하나를 선택한다. 인터페이스의 개수를 N이라고 하면 확률적으로  $1/N$ 로 태그들은 각 인터페이스에 인식된다. 인터페이스의 개수에 반비례하여 태그들이 각각의 인터페이스를 선택한다. 각 인터페이스에 균등하게 인식되는 전체 태그가 N개의 인터페이스로 분산되기 때문에 그만큼 인식시간도 단축시킨다. SP구간의 수집명령에 의해 인터페이스를 선택한 태그들은 LP구간에서 Slotted Aloha 방식으로 자신의 ID를 전달한다. LP구간에서 인식된 태그들은 자신의 사용자 정보 또는 센싱정보를 전달하기 위해 AP구간에 순차적으로 응답한다. 리더의 Read명령에 따라 태그는 응답하고 응답이 완료된 태그는 리더의 절전 명령에 따라 태그는 다시 절전모드로 들어간다.

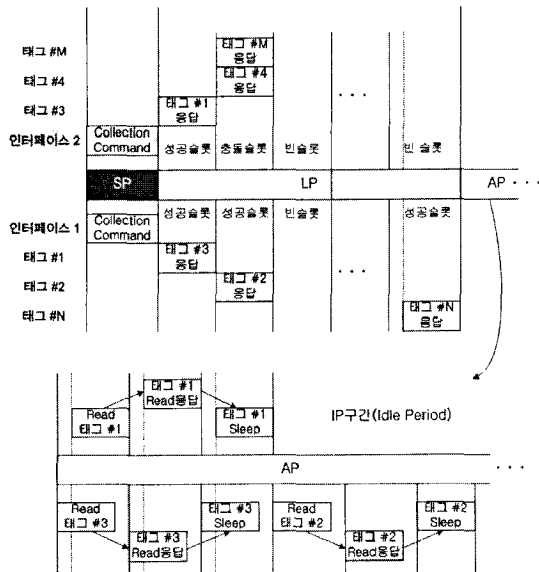


그림 2. LP-결합 프로토콜의 동작과정  
Fig. 2. Collection sequence of LP-Combine protocol

그러나 각 인터페이스의 LP구간에 확률적으로 1/N개로 인식되지 않고 태그가 특정 인터페이스로 물리게 되면 그림2에서와 같이 특정인터페이스에 IP(Idle Period)가 발생한다. 그리고 그 차이가 크면 클수록, AP구간에 전송해야하는 데이터의 크기가 크면 클수록 이 IP에 따른 전체적인 인식시간은 증가한다. 이 IP를 줄이기 위해 AP-균형 프로토콜을 제안한다.

### 2.2 AP-균형 프로토콜

그림3은 AP-균형 프로토콜의 동작 과정을 나타낸다. LP구간에서 Slotted Aloha 방식으로 태그의 ID를 수집하는 것은 LP-결합 프로토콜과 동일하다. 각 인터페이스의 수집명령을 동시에 인식한 태그들이 선택 가능한 인터페이스 중 하나를 선택한다 할지라도 이것은 확률적인 문제이기 때문에 어느 특정 인터페이스로 태그들이 물리는 현상은 언제든지 발생할 수 있다. 이로 인해 AP구간에서 IP가 발생할 수 있다고 설명했다. 이 IP구간의 차이를 줄이고자 제안된 것이 AP-균형 프로토콜이다. 리더는 LP구간에 인식된 태그의 개수를 비교한다. 만약 AP에서 인식해야하는 태그 개수의 차이가 2개 이상일 경우 CSR(Channel Switch Read)명령을 이용하여 다른 인터페이스로 스위칭 시켜 인식한다. 이 같

은 경우 AP구간에서 인식되어야 되는 태그 개수의 차이는 하나이상 발생할 수 없다. 이렇게 함으로써 전체적인 인식시간을 단축시키고 AP구간에서 발생하는 IP를 단축시킨다.

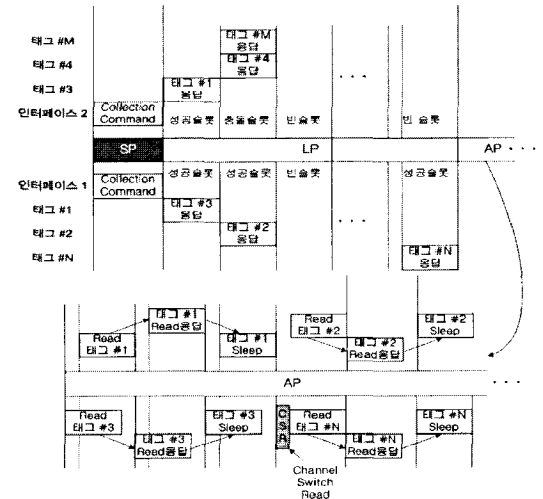


그림 3. AP-균형 프로토콜의 동작 과정  
Fig. 3. Collection sequence of AP-Balanced protocol

## III. 다중인터페이스 다중채널 2.4GHz 능동형 리더 프로토콜 구현

### 1. ISO/IEC 18000-7의 메시지 포맷

태그가 리더로부터 전송받게 되는 메시지는 크게 Broadcast와 Point-to-Point로 구분한다. 이 두 가지 메시지를 Command Code에 따라 사용용도가 다르다[3].

표1은 인식구간별로 사용되는 데이터 포맷과 기능을 함수별로 Command code에 따라 분류한 것이다.

표1. Command list  
Table 1. Command list

Command code	Function	Description	Format	Using
0x25	Wakeup()	Tag wake up	Broadcast	SP
0x1F	Command_collection_with_UDB()	Collection with UDB	Broadcast	SP LP
0x15	Command_sleep()	Tag Sleep	P2P	AP
0xE0	Command_write_memory()	Write tag memory	P2P	Etc
0xB0	Command_read_memory()	Read tag memory	P2P	Etc
0x70	Command_read_UDB()	Read UDB	P2P	AP

2. 태그 채널 선택 및 변경 방법

그림4는 ISO/IEC 18000-7의 SP에서 리더가 태그들에게 전송하는 Collection with Universal Data Block 메시지에 다중인터페이스의 정보를 추가한 메시지 포맷을 나타내고 있다.

Command Code	Windows Size	Max Packet Length	Type
1Byte	2Byte	1Byte	1Byte

Command Code	Windows Size	Max Packet Length	Type	Channel number	Channel Info 1	Channel Info 2
1Byte	2Byte	1Byte	1Byte	1Byte	1Byte	1Byte

그림4. 변경된 Collection with Universal Data Block  
Fig. 4. Changed Collection with Universal Data Block

다중인터페이스를 사용하는 시스템은 인터페이스 2개의 정보를 전송하기 때문에 그림4와 같이 채널의 개수와 각 인터페이스의 채널 정보를 추가하여 SP구간에 태그들에게 전송한다.

다른 채널을 사용하는 인터페이스에 태그가 균등하게 인식되게 하기 위해서는 태그 채널 변경이 필요하다. 태그의 채널을 변경하기 위해 Point-to-Point packet에 명령어를 추가했다.

그림5는 Point-to-Point packet의 Command Code에 채널변경을 할 수 있는 Command Code와 채널 값을 나타낸다. 리더로부터 그림5와 같은 프레임 전송받은 태그는 명령어를 확인한 후 자신의 채널을 Channel Value에 따라 변경한다.

CC2500 Packet Length	Protocol ID	Packet Control	Packet Length	Tag Manufacturer	Tag Serial Number	Interrogator ID	Command Code	Command Arguments	CRC
1 Byte	0x42	1 Byte	16 Byte	2 Byte	4 Byte	2 Byte	1 Byte	N Byte	2 Byte

0x77    Channel Value

그림5. 태그 채널 변경을 위한 명령어 추가  
Fig. 5. Addition command for tag channel change

3. 리더 태그 간 동기화 방법

리더의 수집명령에 태그가 응답을 하고 태그의 응답을 리더가 인식하기 위해서는 리더와 태그 간 동기를 맞추는 과정이 필요하다.

그림6은 리더의 수집명령에 태그가 응답을 하고 그 응답에 리더가 동작할 때 발생하는 동기화 지연에 대한 그림이다. 동기화지연을 고려하지 않고 SP다음에 바로

LP의 윈도우슬롯이 시작하면 동기화지연시간만큼의 시간 후의 LP슬롯에 태그들의 응답이 인식된다. 리더의 수집명령과 태그의 응답은 각각 16byte 데이터를 사용한다. 250kbps의 전송속도를 가지는 RF트랜시버를 사용하기 때문에 이론적인 전송지연은 1ms정도이다. RF칩의 최고속도에 대한 값이며, 무선 환경을 고려하지 않은 수치이다. 그리고 태그가 수집명령을 받아 메시지를 확인한 후 응답을 하기까지의 시간을 고려하지 않은 값이다. 실험을 통해 확인한 결과 4ms의 동기화 지연이 필요했다.

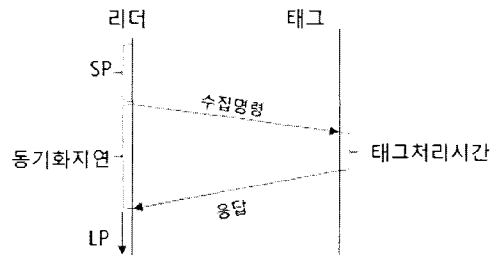


그림6. 리더와 태그 동기화방법  
Fig. 6. Synchronization method between reader and tag

4. LP에서의 태그 ID인식 방법

그림7은 LP의 각 슬롯에서 태그들이 인식되는 과정을 그림으로 나타낸 것이다. ISO/IEC 18000-7과는 다르게 LP에서의 한 슬롯의 시간은 3ms, 슬롯의 개수는 임의로 조절할 수 있도록 구현하였다.

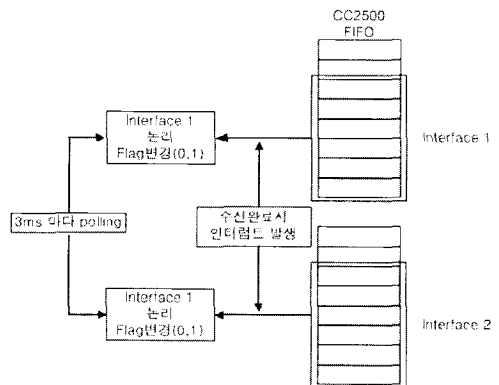


그림7. LP에서 태그 ID인식 방법  
Fig. 7 Collection method of tag ID in LP

슬롯의 크기는 3ms으로 이루어지는데, 데이터 수신 구간 2ms과 CC2500의 FIFO값의 초기화와 전송지연을 고려한 슬롯가드 1ms으로 구성된다.

CC2500의 FIFO값의 크기는 64Byte인데 태그로부터 메시지 수신이 완료되면 인터럽트를 발생시켜 해당 인터페이스의 논리적 Flag값을 변경한다. MCU는 Polling 기법을 이용하여 슬롯 주기인 3ms마다 각 인터페이스의 Flag값을 확인한다. Flag값이 변경된 것을 확인한 후 FIFO의 태그 메시지 값을 MCU의 Rx\_Buffer로 복사한다. 이와 같은 과정을 각 인터페이스 LP구간의 각 슬롯에서 반복하며, Rx\_Buffer에 복사된 메시지를 통해 태그가 슬롯에 성공적으로 인식되었다고 판단한다.

5. 태그 개수 균등화 방법

그림8은 AP에서 각 인터페이스에 인식된 태그의 개수를 확인하여 각 인터페이스에 인식될 태그의 개수를 균등화하는 것을 나타낸다.

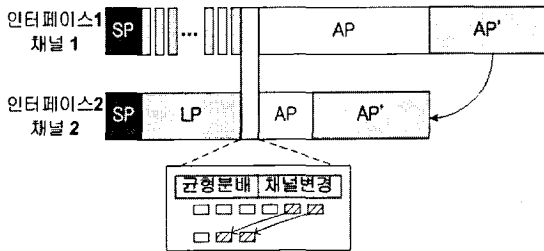


그림8. 태그 개수 균등화 방법

Fig. 8. Balancing method of the number of tags

LP가 끝나게 되면 리더는 각 인터페이스의 LP에 인식된 태그의 개수를 확인한다. 이 때 각 인터페이스에 인식된 태그의 개수의 차이가 2개 이상이 되면 다수가 인식된 인터페이스의 태그를 소수가 인식된 인터페이스로 채널을 변경시킨 후 인식한다. 먼저 각 인터페이스에 인식된 태그의 개수를 파악한다. 인터페이스1의 LP에서 인식된 태그의 개수를 m 이라고하고, 인터페이스2의 LP에서 인식된 태그의 개수는 n 이며, 인터페이스의 개수는 l 이라고 한다.

$$CSt = \left\lfloor \frac{m-n}{l} \right\rfloor \quad (1)$$

그러면, AP구간에서 채널 스위칭 되어야 되는 태그의 개수 CSt는 식(1)과 같이 표시할 수 있다. LP구간이 끝나면 식(1)에 의해 채널 스위칭이 필요한 태그의 개수를 계산한 후 채널 스위칭 명령에 의해 태그의 채널을 변경한 후 인식한다. 만약 인터페이스의 개수가 2개이고, 각 인터페이스에 인식된 태그의 개수가 같거나 차이가 1일 경우는 채널 스위칭이 되어야 되는 태그의 개수가 최소 0 일 것이고, 어느 특정 인터페이스로만 태그가 집중될 경우, 최대  $\frac{m}{2}, \frac{n}{2}$  만큼의 태그를 채널 스위칭 시켜 인식해야 할 것이다.

IV. 실험

1. 실험 환경

그림9는 실험을 위한 리더와 태그의 모습을 보여주고 있다. 리더는 ARM7 MCU에 CC2500 RF트랜시버가 2개 장착된 리더이고 태그는 TI사의 CC2510트랜시버를 사용하여 동작하는 태그이다[4][5]. 실험환경의 구성은 2개의 인터페이스를 사용하는 리더와 총 70개의 태그로 구성하였다. 리더와 호스트 컴퓨터는 SPI(Serial Peripheral Interface)를 통해 연결되어 있으며, 리더와 태그사이의 거리는 1m를 유지하였다. 태그는 1m x 1m 에 70개의 태그를 무작위로 분포시켰다.

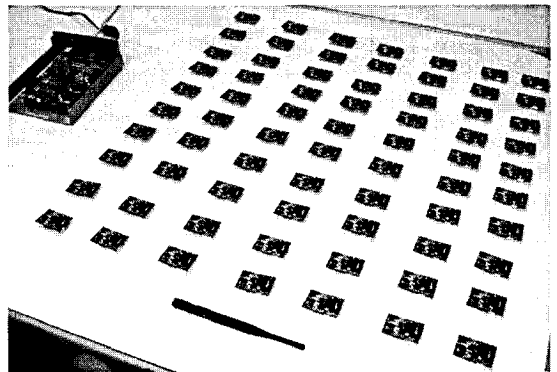


그림9. 실험 환경

Fig. 9. Experiment environment

2. 실험 방법

단일인터페이스와 LP-결합 프로토콜, AP-균형 프로토콜에 대해서 인터페이스를 2개 장착된 리더를 이용하여 실험하였다. 실험은 크게 세 가지로 실시하였다. 첫째, LP의 윈도우 사이즈가 8, 16, 32일 때 태그의 개수를 10개씩 증가시키며 전체 태그 인식시간을 측정하였다.

둘째, 각 윈도우 사이즈에 대해 태그의 개수를 10개씩 증가시키며 전체태그를 인식하는데 소요되는 라운드의 횟수를 측정하였다. 마지막으로 각 윈도우 사이즈에 대해 태그 70개를 인식하는데 각 라운드에서 인식되는 태그의 개수를 측정하였다. 각 실험은 100회씩 반복하여 실시하였다.

그림10은 단일 인터페이스, LP-결합 프로토콜, AP-균형 프로토콜의 동작과정과 성능을 나타내기 위해 개발한 프로그램의 GUI화면이다. 그림의 ①은 각 프로토콜의 태그수집시간을 현재와 평균으로 구분하여 막대그래프로 나타내었다. ②는 프로토콜들을 동작시키기 위한 기능버튼들과 텍스트형식의 결과 값이다. ③은 각 프로토콜들에서 동작되는 과정을 나타내고 있는데, 라운드의 개수와 각 라운드에서 인식된 태그의 개수를 나타내고 있다. AP-균형 프로토콜의 경우 LP에서 인식된 태그 개수 차이에 의해 AP에서 채널 변경된 태그를 별도로 표시했다.

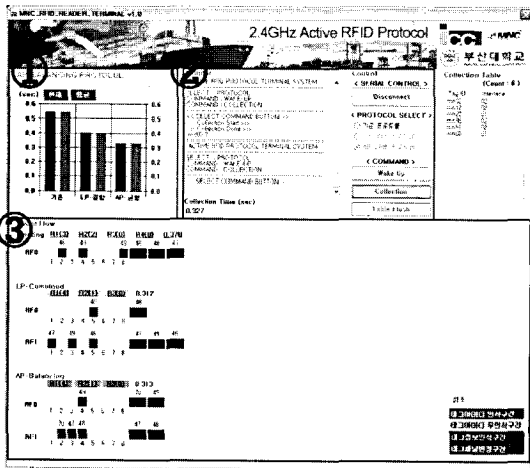


그림10. 태그인식을 위해 개발한 프로그램의 GUI화면  
Fig. 10. GUI screen for tag collection protocol

V. 성능평가

1. 프로토콜별 전체 태그 인식시간

그림11은 LP의 윈도우 개수가 16일 때 각 프로토콜별 전체 태그 인식시간을 태그의 개수를 10개씩 증가시키며 측정한 결과이다. 태그의 개수가 증가할수록 단일 인터페이스와 다중 인터페이스 프로토콜과의 성능 차이는 커졌다. 70개의 태그에 대하여, LP-결합 프로토콜은 단일인터페이스보다 697ms의 수집시간을 단축시켰고, AP-균형 프로토콜은 LP-결합 프로토콜보다 23ms의 수집시간을 단축시켰다. 이 같은 결과는 윈도우 크기를 8 또는 32로 하였을 때도 비슷한 성능분포를 보였다.

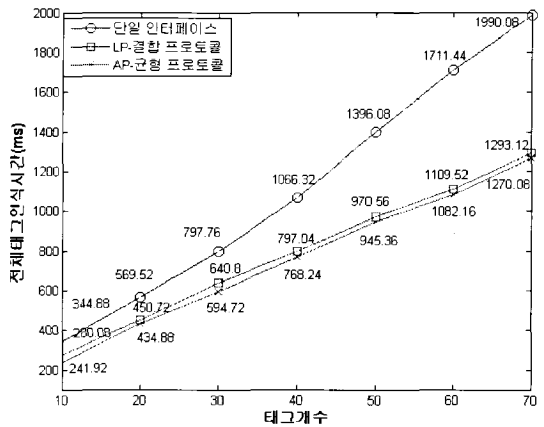


그림11. 프로토콜별 전체 태그 인식시간  
Fig. 11. Total tag collection time about each protocol

2. 태그개수별 전체태그인식 라운드 수

그림12는 윈도우 크기가 16일 때 프로토콜별 전체 태그를 인식하는데 사용된 라운드 수를 태그 수를 10개씩 증가시키며 나타내었다. 인식라운드 수에 따라 인식시간이 결정되기 때문에 인식시간과 비슷한 성능분포를 보였다.

각 윈도우크기에 대해 태그의 개수가 증가할수록 인식라운드 수는 절반 또는 절반 이상의 성능개선을 보이는 반면 전체태그 인식시간은 절반 이하의 성능이 개선되었다. 왜냐하면 단일인터페이스에 비해 LP-결합 프로토콜, AP-균형 프로토콜은 동기화문제, 태그 채널 스위칭 시간 등의 유희구간이 생기기 때문에 하나의 라운드

를 동작시키는 시간이 단일인터페이스보다 크기 때문이다.

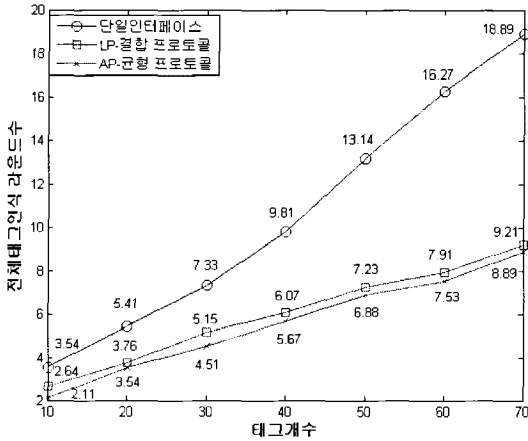


그림 12. 윈도우 크기가 16일 때 프로토콜별 전체태그인식 라운드 수

Fig. 12. The number of total tag collection round about each protocol in windows size(16)

3. LP윈도우크기별 전체 태그 인식시간

그림 13은 LP-결합 프로토콜에서 LP의 윈도우크기가 8, 16, 32일 때, 태그 수를 10개씩 증가하면서 전체 태그를 수집하는데 소요되는 시간을 나타내고 있다.

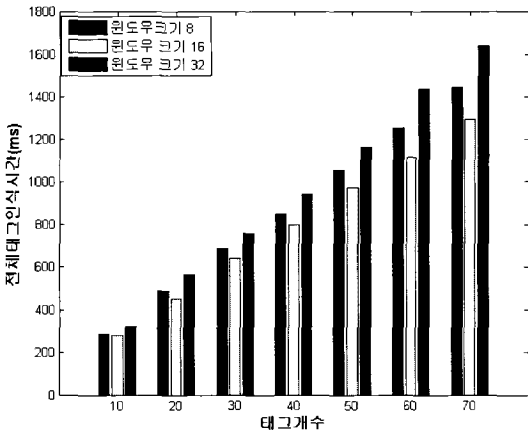


그림 13. LP-결합 프로토콜에서 LP윈도우크기별 전체 태그 인식 시간

Fig. 13. Total tag collection time about each LP windows size in LP-Combined protocol

LP의 윈도우크기를 동적으로 조절할 수 없는 환경에서 윈도우의 크기가 전체 태그 수집시간에 미치는 영향을 알아보기 위한 것이다. 윈도우 크기가 16인 경우가 대체로 빠른 태그 수집 시간을 나타내고 있다. 왜냐하면, 윈도우 크기가 8인 경우는 하나의 라운드에 소요되는 시간은 적으나 충돌에 의한 라운드 수가 증가하고, 윈도우 크기가 32인 경우는 인식 라운드 수는 적으나 한 라운드에 소요되는 시간이 크기 때문이다. 만약 Vogt가 제안한 것과 같은 태그 개수 추정방법 사용이 가능하다면 LP의 윈도우크기를 동적으로 하여 좀 더 빠른 태그 수집이 가능할 것이다[6].

4. 70개의 태그에 대해 각 라운드별 인식태그 수

그림 14는 LP의 윈도우크기를 8로 하여 70개의 태그를 인식할 때 각 라운드에서 인식되는 태그의 개수에 대한 평균을 각 프로토콜별로 나타내고 있다.

$$Slotw = Slot1 + Slot0 + Slotc \quad (2)$$

인식과정의 LP슬롯은 식(2)와 같이 구성된다. Slotw은 전체 윈도우슬롯의 개수이고, Slot1은 인식 슬롯의 개수, Slot0은 빈 슬롯의 개수, Slotc은 충돌 슬롯의 개수이다.

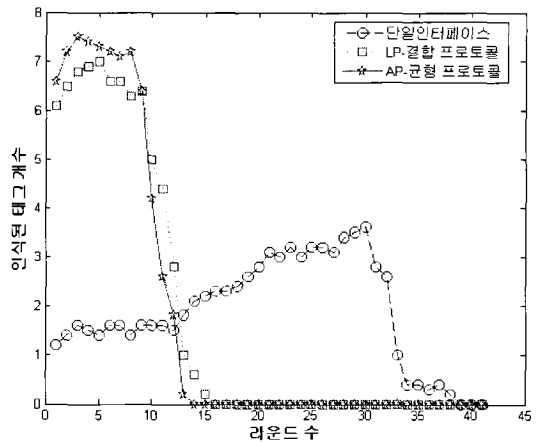


그림 14. LP의 윈도우크기가 8일 때 태그70개에 대한 각 라운드별 태그인식개수

Fig. 14. The number of tag of each round about 70tags in windows size(8) of LP



단일인터페이스인 경우 인식된 태그로 인한 충돌의 감소에 의해 라운드가 증가할수록 인식 개수가 증가하는 것을 확인했다. LP-결합 프로토콜과 AP-균형 프로토콜은 전체 70개의 태그를 35개씩 나누어 각 인터페이스에 인식되는 효과가 있다. Slotc의 감소에 의한 Slotl의 증가 때문에 단일 인터페이스보다 6배의 인식성능을 나타내었다.

그림15는 LP의 윈도우크기를 16으로 하여 70개의 태그를 인식할 때 각 라운드에서 인식되는 태그의 개수에 대한 평균을 각 프로토콜별로 나타낸 것이다. 단일인터페이스의 경우 라운드가 증가할수록 인식된 태그에 의한 충돌의 감소에도 불구하고 인식되는 태그의 개수가 줄어드는 것을 확인했다. 이것은 슬롯에 충돌이 일어날 확률보다 빈 슬롯이 될 확률이 더 커지기 때문이다.

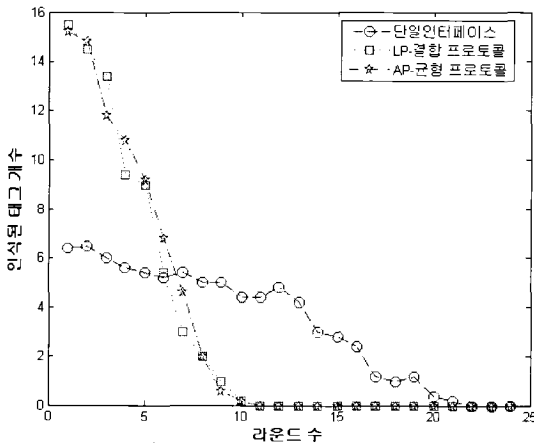


그림 15. LP의 윈도우크기가 16일 때 태그70개에 대한 각 라운드별 태그인식개수

Fig. 15. The number of tag of each round about 70tags in windows size(16) of LP

## VI. 결론 및 향후 연구계획

다중인터페이스, 다중채널 2.4GHz 능동형 RFID 프로토콜인 LP-결합프로토콜과 AP-균형 프로토콜을 펌웨어 레벨의 하드웨어에서 구현하고 성능을 평가했다. 구현된 프로토콜을 단일인터페이스리더와 비교하였을 때, 태그의 개수가 많아질수록 인식라운드 수, 인식시간의

성능 차이가 크게 나타나는 것을 확인했다. 70개의 태그를 인식하는 인식시간을 비교해보니 윈도우크기가 16인 단일인터페이스리더보다 윈도우크기가 8인 LP-결합, AP-균형 프로토콜의 인식시간이 짧았다. 왜냐하면 멀티인터페이스를 이용하여 각 인터페이스에 인식되어야 되는 태그들을 분산시킴으로써 태그 충돌확률이 낮기 때문이다. LP-결합 프로토콜에서 윈도우크기를 8, 16, 32로 변화하면서 전체 태그 인식시간을 측정 한 결과 윈도우크기가 16인 경우가 가장 우수했다. 윈도우수가 8인 경우는 충돌에 의한 인식라운드수가 많아지고, 윈도우수가 32인 경우는 인식라운드 수는 적지만 하나의 라운드가 소모하는 시간이 많기 때문에 전체적인 수집시간이 커졌다. LP의 윈도우의 크기에 따라 전체인식시간의 성능이 좌우되는 것을 실험을 통해 확인 하였다. 본 연구진은 윈도우의 크기를 동적으로 할당하기위해 필요한 무선인지방법을 연구하여 태그의 개수를 추정 한 후 동적으로 윈도우크기를 설정하여 고속의 태그 수집을 할 수 있는 연구를 계속 수행하도록 하겠다.

### 감사의 글

이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. KRF-2008-D00490)

### 참고문헌

- [1] Klaus Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Card Identification*, 2nd edition, John Wiley & Sons, 2003.
- [2] 박현성, 김동현, 정상화, 백운주, 김종덕, "다중채널 다중 인터페이스 능동형 RFID 리더 및 프로토콜," 한국정보과학회논문지 제36권 제2호, pp. 118-129, 2009년 4월.
- [3] ISO/IEC 18000.7.2, "Information technology AIDC techniques - Radio frequency identification for item management - Air interface, Part 7: Parameters for active air interface communications at 433MHz," 2008.

- [4] 2.4GHz Active RFID Reader, 부산대학교 컴퓨터공학과 임베디드 네트워크시스템연구실,  
<http://miranda.ce.pusan.ac.kr>
- [5] 박세영, 김택현, 최 훈, 조현태, 백윤주, “저전력 2.4GHz 능동형 RFID 태그의 설계 및 구현,” UCT 2008, p134~136, 2008년 7월
- [6] H. Vogt, “Efficient Object Identification with Passive RFID Tag,” Proceedings of International Conference on Pervasive Computing, Zurich, Swiss, 2002.

### 저자소개



김동현(Dong-Hyun Kim)

1998년 동의대학교  
정보통신공학과 학사  
2004년 동의대학교  
정보통신공학과 석사

현재 부산대학교 컴퓨터공학과 박사과정  
※ 관심분야: 이동통신망, RFID/USN



이채석(Chae-Suk Lee)

2008년 경성대학교 컴퓨터공학과  
학사  
현재 부산대학교 컴퓨터공학과  
석사과정

※ 관심분야: 무선통신, 이동통신망, RFID/USN



김종덕(Jong-Doek Kim)

1994년 서울대학교 계산통계학과  
학사  
1996년 서울대학교 전산학과  
석사

2003년 서울대학교 컴퓨터공학과 박사  
2004년 ~ 현재 부산대학교 컴퓨터공학과 부교수  
※ 관심분야: 무선통신, 이동통신망, RFID/USN