

2.4GHz 다중채널 능동형 RFID시스템을 위한 대용량 태그 데이터 전송 프로토콜의 설계 및 구현

(Design and Implementation of Large Tag Data Transmission Protocol for 2.4GHz Multi-Channel Active RFID System)

이 채 석 ^{*} 김 동 현 ^{*} 김 종 덕 ^{**}
 (Chae-Suk Lee) (Dong-Hyun Kim) (Jong-Doek Kim)

요약 능동형 RFID가 많은 응용 분야에서 활용되기 위해서 대용량의 태그 데이터를 보다 빠르고 효율적으로 보낼 필요성이 대두되고 있다. ISO/IEC 18000-7 표준은 433MHz대역을 사용하고 있으며 전송 속도는 27.8kbps에 불과하다. 이러한 전송속도는 대용량 데이터를 전송하기에 무리가 있다. 우리는 빠른 대용량 태그 데이터 전송을 위하여 2.4GHz대역을 사용하였고, 보다 효율적인 데이터 전송을 위하여 데이터 전송 프로토콜을 설계하였다. 우리가 설계한 프로토콜은 Burst Read UDB라는 새로운 명령어를 사용하여 255bytes이상의 메시지 데이터를 생성하고, 이를 효과적으로 전송할 수 있도록 이루어져 있다. 프로토콜을 구현하기 위하여 Texas Instruments사에서 개발한 Smart RF04 개발보드를 사용하였으며, 통신모듈로는 CC2500트랜시버를 이용하였다. 구현을 통한 비교실험에서 63.75kbytes의 데이터를 전송할 경우, 기존 표준의 Read UDB보다 Burst Read UDB가 17.95%의 데이터 전송시간이 향상되었음을 확인하였다.

키워드 : 2.4GHz Active RFID, 대용량 태그, 태그 데이터 전송 프로토콜, ISO/IEC 18000-7

Abstract To apply active RFID technology in the various kinds of industry, it needs to quickly transmit a large amount of data. ISO/IEC 18000-7 standard uses the 433.92MHz as single channel system and its transmit rate is just 27.8kbps, that is insufficient for a large amount of data transmission. To solve this problem, we designed a new data transmission protocol using 2.4GHz band. The feature of designed protocol is not only making over 255bytes data messages using the Burst Read UDB but also efficiently transmitting it. To implement this protocol, we use Texas Instruments's SmartRF04 develop kit and CC2500 transceiver as RF module. As an evaluation of 63.75kbytes data transmission, we demonstrate that transmission time of Burst Read UDB has improved as 17.95% faster than that of Read UDB in the ISO/IEC 18000-7.

Key words : 2.4GHz, Active RFID, Large tag data, tag data transmission protocol

1. 서론

RFID(Radio Frequency Identification)기술은 RF를 이용하여 먼 거리에서 사물의 정보를 인식하는 기술을 말한다. RFID태그는 수동형(passive), 반 수동형(semi-passive), 능동형(active)으로 구분한다.

수동형 태그는 리더(reader, interrogator)로부터 전원 공급을 일체 의존하는 시스템으로 인식거리가 짧지만 단가가 저렴하며, 휴대가 간편한 장점이 있다. 반수동형 태그는 기존 수동형 태그에 자체 전원 공급을 위한 박형 전지(film battery)를 장착하여 내부 회로의 구동전원으로 활용하나 역시 리더로부터의 전원을 통해서만 전파를 송신한다. 이 같은 방법으로 수동형 태그의 최대 문제점인 인식률과 인식거리를 향상시킬 수 있다. 능동

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

[†] 학생회원 : 부산대학교 컴퓨터공학과
 quarry@pusan.ac.kr
 dhkim1106@pusan.ac.kr

^{**} 정회원 : 부산대학교 컴퓨터공학과 교수
 kimjd@pusan.ac.kr
 논문접수 : 2009년 10월 5일
 심사완료 : 2010년 2월 5일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 정보통신 제37권 제3호(2010.6)

형 태그는 태그가 전원 및 RF 트랜시버(transceiver)를 장착하고 있어 리더로부터 별도의 에너지 공급 없이 자체적으로 정보를 송신 및 수신할 수 있고, 다른 방식들에 비해 인식거리가 길고 인식률이 좋으며 여러 부가기능의 추가가 용이한 장점이 있다[1,2].

능동형 RFID 응용 분야의 확대와 함께 기본적인 식별 기능 외에 사용자 정보의 저장 및 전달, 온 습도 등의 주변 정보 센싱 및 전달, 실시간 측위(RTLS: Real Time Locating System) 등의 추가 기능 요구가 발생하고 있다. 일반적으로 RFID 시스템은 공급망 관리를 위한 물품 재고 및 이력 정보의 효율적인 관리를 목적으로 산업분야에 주로 적용되는데, 리더는 주기적으로 혹은 사용자가 원할 때 통신 범위 내의 모든 태그들로부터 정보를 빠르게 수집할 수 있어야 한다. 이러한 이유로 기존 연구들은 태그 수집(Tag Collection) 능력 향상에 중점을 두고 진행되어 왔다[3-7].

최근 발행된 차세대 능동형 RFID 표준화 로드맵에 의하면 대용량 태그 데이터를 전송하기 위한 전송 기술의 필요성을 강조하고 있다[8,9]. 대용량 메모리를 탑재한 태그를 효과적으로 관리하기 위해서는 태그의 메모리를 효과적으로 관리하는 방법과 이를 효과적으로 전송하기 위한 방법을 고려하여야 한다. 현재 능동형 태그의 메모리에 대용량으로 누적된 데이터를 전송하기 위한 연구는 이루어져 있지 않다. 본 논문에서는 대용량 메모리를 탑재한 태그의 데이터를 효과적으로 전송하기 위한 방법에 중점을 둔다.

능동형 RFID 시스템의 표준인 ISO/IEC 18000-7은 433.92MHz 대역을 사용하며 채널대역이 200KHz인 단일 채널 시스템으로 데이터 전송속도는 27.8Kbps 수준이다[10]. 대용량 태그 데이터를 효과적으로 전송하기 위해서는 ISO/IEC 18000-7의 27.8Kbps 수준의 전송속도는 부족함이 있다. ISO/IEC 18000-7은 한 번에 전송 가능한 최대 메시지의 크기를 255bytes로 허용하고 있다. 255bytes 크기의 메시지 100개를 전송할 경우 27.8 kbps의 전송속도로는 7.4 sec이라는 많은 시간이 소요되며, 이러한 시간은 실제 태그가 많은 환경에서 굉장히 큰 시간이다.

또한 ISO/IEC 18000-7에서는 대용량 태그 데이터를 전송하기 위하여 Read UDB(Universal Data Block) 명령어를 제공하고 있다. Read UDB 명령어는 대용량의 태그 데이터를 한 번에 전송 가능한 데이터(UDB)들로 구분하여 순서대로 전송한다. 전송이 제대로 되었는지 확인하기 위하여 전송된 데이터 마다 전송확인 메시지를 받고, 해당하는 데이터가 여러개 생기거나 전송 받지 못하면 재전송을 받는다. 이와 같은 동작을 100회를 반복하여 전송한다면 이에 해당하는 전송확인 메시지를

전송 받아야 한다. 전송확인 메시지는 End-to-End 지연과 메시지 처리 및 송·수신 모드 변환시간을 발생시키며, 추가적인 에러 발생이나 손실이 발생할 경우 전체 동작시간이 더욱 증가한다.

우리는 전송속도 문제를 개선하기 위하여 2.4GHz ISM(Industrial Scientific Medical)대역을 사용하였다. 2.4GHz ISM대역은 별도의 사용권한 획득이 필요 없으며, 기존 표준에 비해 개별 채널의 대역폭이 증가하므로 전송속도가 증가하고 다수의 채널을 구성할 수 있어 주파수 중복에 따른 성능 저하를 피할 수 있다. 우리는 2.4GHz대역을 구성하기 위하여 TI사에서 개발한 CC2500 트랜시버를 선택하였고, 이것을 SmartRF04개발보드에 장착하여 개발환경을 구성하였다[11-13].

900MHz대역은 모바일 RFID, 고정형 RFID, USN을 함께 써야 하므로 주파수 배분에 대한 문제가 발생하며, 이에 따라 주파수 대역폭을 넓혀야 한다는 주장이 있다. 2.4GHz ISM 대역의 사용은 이러한 문제점을 해결할 수 있는 하나의 대안이라 할 수 있겠다[14].

그리고 또 다른 문제인 전송 프로토콜을 개선하기 위하여 태그가 리더에게 데이터를 Burst하게 전송하는 Burst Read UDB 명령어를 설계하였다. 리더가 태그에게 메시지를 보내게 되면 보내야 할 데이터를 Burst하게 연속적으로 전송한다. 그리고 Read UDB 명령어는 한 번에 전송 가능한 전체 메시지의 크기가 255bytes이지만, Burst Read UDB 명령어는 메시지의 크기를 255bytes이상 전송할 수 있도록 설계하였다. 우리는 Burst Read UDB 명령어의 구현과 더불어 표준보다 확장된 메시지의 크기를 전송할 수 있도록 구현하기 위하여 CC2500 트랜시버에서 제공하는 Variable Length Mode와 Infinite Length Mode를 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 CC2500 트랜시버와 ISO/IEC 18000-7표준의 구조를 간략히 살펴보고, 3장에서는 대용량 태그 데이터 전송을 위한 전송 프로토콜을 설계한다. 4장에서는 설계한 내용을 구현하며, 5장에서 구현된 방안들의 성능측정을 위한 실험방법을 설명하고 이를 통한 결과로 성능분석을 한다. 마지막으로 6장에서 결론 및 향후 연구계획을 언급하며 글을 맺겠다.

2. 관련 연구

2.1 CC2500 트랜시버

본 논문에서는 2.4GHz대역을 사용하기 위한 통신모듈로 Texas Instruments사의 CC2500 트랜시버(Transceiver)를 사용하였다. CC2500의 특징으로는 2400-2483.5 Mhz의 주파수 대역을 사용할 수 있으며, 최대 500kbps의 속도로 데이터를 전송할 수 있다. 데이터 전송을 위

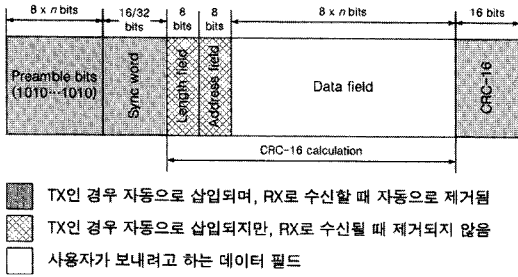


그림 1 CC2550 패킷 포맷

한 FIFO버퍼로 64bytes의 공간을 가지고 있으며 기본적으로 150m에 이르는 통신유효 범위를 가지고 있다 [15]. CC2550에서 전송되는 패킷 포맷은 그림 1과 같은 형태로 이루어진다. 전송 시, 전송 타이밍의 동기화를 위한 Preamble bits와 Sync word 필드가 칩 내부에서 자동으로 패킷의 헤더에 붙는다. 이것은 CC2550에서 이루어지는 부분이며 사용자는 해당 필드의 길이만을 수정할 수 있다. Length field 1byte는 사용자가 보내고자 하는 데이터의 전체 길이를 나타내며, Address field는 해당 기기의 주소를 나타낸다. Data field 영역은 사용자가 보내고자 하는 데이터를 나타내는 영역이며 우리가 보내고자 하는 ISO/IEC 18000-7 데이터 포맷이 여기에 기록된다. Length field가 표현할 수 있는 최대 길이는 255 bytes인데, 패킷 포맷의 길이가 255bytes 이상 늘어날 경우 Address field를 2번째 Length field로 사용한다.

기본적으로 CC2550은 모든 길이의 패킷을 전송할 수 있다. 그러나 펌웨어 레벨에서 FIFO 크기(64bytes)보다 더 큰 크기의 패킷을 전송하기 위해서는 코드의 복잡도가 증가하는 단점이 있다. CC2550은 코드의 복잡도를 줄이기 위하여 fixed, variable, 그리고 infinite라는 3가지 패킷 길이모드를 정의하고 있다.

패킷의 길이가 64bytes 보다 적은 경우일 때에는 Fixed Packet Length Mode를 사용하여 전송할 수 있으며, 펌웨어 구현 측면에서 가장 간단하게 전송 및 수신을 사용할 수 있다. Fixed Packet Length Mode에서 패킷의 길이를 10bytes로 정하였다면, 트랜시버의 FIFO에 20bytes를 넣어도 10bytes만 전송한다. 반대로, 10bytes 보다 적게 넣었다면 10bytes가 될 때까지 전송하기 않고 대기한다.

255bytes보다 적은 패킷을 전송할 경우에는 Variable Packet Length Mode를 사용한다. Variable Packet Length Mode는 FIFO의 크기 이상을 전송하거나 수신하기 위하여 정의된 모드이며, 이를 사용하기 위해서는 FIFO의 상태에 따라 주기적으로 읽거나 모두 전송한 후 다시 자동으로 채우는 부가적인 구현이 필요하다. 구현하는 방식으로는 Polling과 Interrupt Driven 구현방

식이 있다. Polling은 MCU가 CC2550의 FIFO상태를 파악하고, FIFO에 데이터를 채우거나 읽는 동작을 반복하는 방식이다. Interrupt Driven은 CC2550의 상태가 변경되었을 경우 Interrupt를 발생시키며, MCU는 Interrupt를 통하여 FIFO에 데이터를 채우거나 읽는다.

255bytes보다 큰 패킷을 전송하기 위해서는 Infinite Packet Length Mode가 사용된다. Infinite Packet Length Mode와 Variable Packet Length Mode와의 차이점은 Variable Mode는 Tx모드로 놓게 되면 255bytes까지 패킷 카운팅을 하여 자동적으로 Tx모드가 정지되는 반면에, Infinite Mode는 칩에서 255bytes 이상을 셀 수 없으므로 프로그램 적으로 Tx모드를 정지할 때까지 FIFO에 있는 데이터들을 계속해서 전송한다. 255bytes이상의 패킷을 전송 할 경우, 칩에서 계산 가능한 패킷 길이를 초과하므로 칩 레벨에서의 CRC의 계산 및 다른 상태 레지스터들을 사용할 수 없다. 그러므로 Infinite Packet Length Mode를 사용할 때에는 CRC의 계산 및 추가를 하기 위하여 Fixed Packet Length Mode와 함께 조합하여 사용하며, 이에 따른 코드의 구현이 필요하다.

본 논문에서는 Infinite Packet Mode를 사용하여 255bytes이상의 대용량 패킷을 전송할 수 있도록 하였다.

2.2 ISO/IEC 18000-7

ISO/IEC 18000-7의 동작 과정은 태그를 수면모드에서 동작모드로 전환시키는 Wakeup구간이 먼저 시작되며, 모든 태그를 처리할 때까지 수집구간(Collection Period)이 반복되어 수행한다. 수집구간은 SP(Synchronization Period)과 LP(Listen Period), 그리고 태그가 가지고 있는 정보를 읽거나 다른 옵션을 설정할 수 있는 AP(Acknowledge Period)로 구성된다.

그림 2는 ISO/IEC 18000-7 표준에서 사용되는 메시지 포맷 및 사용되는 명령어들을 정리한 것이다. ISO/IEC 18000-7은 리더와 태그의 통신을 위하여 3가지의 전송포맷을 정의하고 있다. Broadcast포맷 형식을 사용하는 명령어는 SP구간에서 태그를 수집하기 위하여 사용하는 Collection with UDB 명령어가 유일하며, 다른 명령어들은 AP구간에서 사용되며 모두 P2P포맷 형식을 따르고 있다. 명령어 동작 시, Command Code와 Command Arguments는 명령어 코드의 종류에 따라 값과 길이가 틀려진다. CC2550 Packet Length는 표준에서 정의된 필드는 아니지만, CC2550트랜시버는 가장 앞에 길이필드가 있어야 동작하므로 본 논문에서 추가된 것이다.

대용량 태그 데이터 전송 위한 명령어는 AP에서 수행되는 Read UDB 명령어가 유일하며, 자세한 동작과정은 다음 3장에서 자세하게 설명하도록 하겠다. 현재

Interrogator-to-tag message format (point-to-point)

CC2500 Packet Length	Protocol ID	Packet Options	Packet Length	Tag Manufacturer	Tag Serial Number	Interrogator ID	Command Code	Command Arguments	CRC (CCITT)
1 Byte	0x40	1 Byte	1 Byte	2 Byte	4 Byte	2 Byte	1 Byte	N Byte	2 Byte

Interrogator-to-tag message format (broadcast)

CC2500 Packet Length	Protocol ID	Packet Options	Packet Length	Interrogator ID	Command Code	Command Arguments	CRC(CCITT)
1 Byte	0x40	1 Byte	1 Byte	2 Byte	1 Byte	N Byte	2 Byte

Tag-to-interrogator response message format (point-to-point)

CC2500 Packet Length	Protocol ID	Tag Status	Packet Length	Interrogator ID	Tag Serial Number	Command Code	Response Data	CRC (CCITT)
1 Byte	0x40	2 Byte	1 Byte	2 Byte	4 Byte	1 Byte	N Byte	2 Byte

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
0x60	Command_read_memory();	Read tag memory	P2P	Etc
0x70	Command_read_UDB();	Read UDB	P2P	AP

그림 2 ISO/IEC 18000-7 메시지 포맷 및 명령어 목록

2.4GHz 기반 능동형 RFID시스템에 대한 표준은 정의되어 있지 않으므로, 본 논문에서는 통신과정의 기본 틀을 ISO/IEC 18000-7표준을 따른다.

3. 대용량 태그 데이터 전송 프로토콜의 설계

3.1 메시지 구조 정의

그림 3은 데이터 포맷들의 명칭 정의 및 패킷 생성 과정을 나타낸 것이다. 본 논문에서는 전송하고자 하는 순수 값을 데이터라고 정의하며, 이 데이터에 ISO/IEC 18000-7의 헤더와 CRC가 추가된 형태를 메시지라고 하겠다. 하나의 전송할 메시지가 FIFO보다 큰 경우, 여러 횟수로 나누어서 전송해야 하는데, 이때 전송되는 것을 하나의 패킷이라고 정의한다.

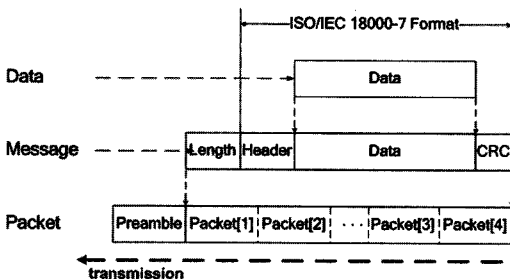


그림 3 명칭 정의 및 패킷 생성 과정

3.2 대용량 태그 데이터 전송속도 증대를 위한 전송 방법의 설계

ISO/IEC 18000-7 표준에서 대용량 전송을 위하여 사용되는 명령어는 Read UDB 명령어가 유일하다. Read UDB명령어는 리더가 태그의 데이터를 블록 단위로 읽는 명령어이다.

리더는 태그에게 그림 4의 (a)와 같이 각 1byte씩 구성되어있는 명령어필드를 P2P메시지 포맷에 추가하여 전송한다. Sequence ID는 여러 개의 메시지가 전송되는 동안 에러가 발생하면 메시지의 식별을 위한 것으로서, 초기에 0으로 설정되어 있다. 이후 태그로부터 메시지 전달을 성공하면 1씩 증가시켜 전송한다.

(a) Interrogator-to-tag Read UDB command field

command code	command arguments		
	Sequence ID	Max Packet Length	Type
0x70	1 byte	1 byte	1 byte

(b) Tag-to-Interrogator Read UDB command field

command code	command arguments	
	Sequence ID Countdown	Universal Data Block
0x70	1 byte	N bytes

그림 4 Read UDB 명령어 필드

표 1 분석을 위해 사용할 상수 정의

상수	의미
$T_{transmit}$	리더가 태그에게 데이터 수신을 위한 명령한 시간부터 태그에게 받은 데이터를 처리하기 까지 걸리는 시간
$T_{Intt-Tag}$	리더로부터 태그까지 명령어가 전송되는 전송지연 시간
$T_{Tag-process}$	태그가 리더로부터 전송된 명령어를 처리하여 UDB를 구성하고 수신모드를 송신모드로 변경하는 시간
$T_{Tag-Intt}$	태그로부터 리더까지 데이터가 전송되는 전송지연 시간
$T_{Intt-process}$	리더가 태그로부터 전송된 데이터를 처리하여 UDB를 저장하며 수신모드를 송신모드로 변경하는 시간
$T_{total-readUDB}$	Read UDB 명령어를 수행하는 전체 시간
$T_{total-burstUDB}$	Burst Read UDB 명령어를 수행하는 전체 시간

이것을 태그의 메시지 전송에 대한 Ack메시지라고 정의한다. Max Packet Length는 리더가 태그로부터 한 번에 전송 받을 메시지의 크기를 이야기한다. RF트랜시버의 성능에 따라 최대 255bytes까지 지정할 수 있으며, 태그는 이 값을 이용하여 데이터를 몇 개의 UDB로 만들 것인지 결정한다. Type은 데이터의 종류를 결정하는 것으로써 사용자의 요구에 따라 값이 결정된다.

태그는 명령어를 수신한 뒤 리더에게 그림 4의 (b)와 같은 응답 필드를 전송한다. 1byte로 구성된 Sequence ID Countdown은 리더로부터 전송된 Max Packet Length와 자신이 전송할 메시지를 이용하여 초기값을 결정하며, 성공적으로 전송될 때마다 1씩 감소한다. 최종적으로 0을 전송한 뒤 리더로부터 명령어를 받으면 모두 전송한 것으로 판단한다.

표 1은 명령어들의 분석을 위한 상수들을 정의한 것이다. 본 논문에서는 태그 메모리의 처리 및 읽는 시간은 고려하지 않는다. 그림 5는 표 1을 통하여 Read UDB의 동작과정을 분석한 것이다. 이를 통하여 리더로부터 태그까지 1개의 메시지가 전송되는 과정을 계산하면 식 (1)을 구할 수 있다. N 개의 데이터를 전송하였다면 식 (2)와 같이 표현 할 수 있다. E 는 전송도중 발생한 에러의 개수를 나타낸다.

대용량 데이터 전송은 많은 수의 메시지가 전송되며,

정지대기방식은 메시지 하나하나를 확인한 다음 전송하므로 통신의 신뢰성은 우수하다. 하지만 데이터마다 Ack전송을 해야 하므로 End-to-End delay가 발생하고 잦은 송·수신 모드 변환과정으로 인하여 최종적으로 시간지연 문제가 발생한다.

본 논문에서는 Read UDB 동작에서 Ack전송시간과 송·수신 모드 변환시간의 단점을 개선시키기 위하여 새로운 전송방식을 설계한다.

그림 6은 우리가 설계한 Burst Read UDB의 동작과정을 나타낸 것이다. Ack전송을 줄이기 위하여 리더는 최초 한 번의 명령만을 태그에게 전송한다. 태그는 명령을 처리하여 UDB의 크기와 개수를 계산하여 한 번에 연속적(Burst)으로 리더에게 전송한다. 리더는 데이터를 수신되는 동안 Sequence ID Countdown 값을 보고, 수신이 되지 않거나 에러가 난 패킷을 파악한다. 태그로부터의 전송이 완료되면 태그에게 에러나 수신이 되지 않은 Sequence 목록을 정리하여 재전송 요구 목록을 태그에게 전송한다. 태그는 Sequence 목록을 보고 목록 순서대로 연속적으로 에러회복을 위한 전송을 시작한다.

식 (3)은 Burst Read UDB의 동작 중 에러를 포함하여 모든 전송을 완료 하였을 경우를 수식으로 나타낸 것이다. 식 (2)를 이용하여 Read UDB의 전체수행시간과 Burst Read UDB의 수행시간을 빼면 식 (4)를 얻을

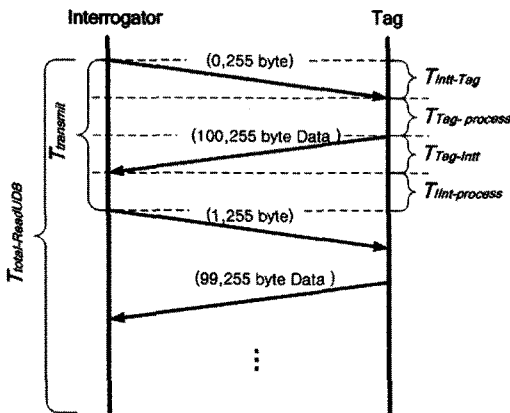


그림 5 Read UDB 명령어의 동작과정

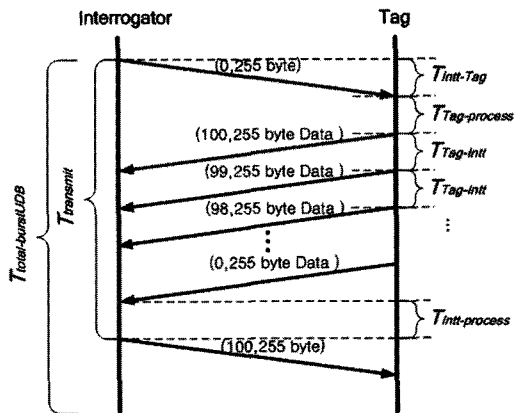


그림 6 Burst Read UDB 명령어의 동작과정

$$T_{transmit} = T_{Intt-Tag} + T_{Tag-process} + T_{Tag-Intt} + T_{Intt-process} \tag{1}$$

$$T_{total-readUDB} = N T_{transmit} + E T_{transmit} \tag{2}$$

$$T_{total-burstUDB} = (T_{Intt-Tag} + T_{Tag-process} + N T_{Tag-Intt} + T_{Intt-process}) + T_{Intt-Tag} + E(T_{Tag-process} + T_{Tag-Intt}) \tag{3}$$

$$T_{total-readUDB} - T_{total-burstUDB} \approx N(T_{Intt-Tag} + T_{Tag-process} + T_{Intt-process}) + E(T_{Intt-Tag} + T_{Intt-process}) \tag{4}$$

수 있다. 식 (4)는 각각의 수행시간 중 중복된 시간을 빼 나머지 수행되는 시간을 나타낸 것으로써, Read UDB가 Burst Read UDB에 비하여 리더-태그전송에 관한 End-to-End delay, 태그 송·수신 모드 전환시간 및 리더 송·수신 모드 전환시간이 전체전송횟수의 N 번 더 수행된다는 것을 알 수 있다. 그리고 에러가 발생한다면 Burst Read UDB보다 E 가 발생한 만큼 리더-태그에 관한 End-to-End delay 및 리더 송·수신 모드 전환시간이 추가된다. 즉, 전체전송횟수가 늘어나고 이에 따른 에러 발생횟수에 따라 Read UDB와 Burst Read UDB의 차이는 선형적으로 차이가 벌어짐을 예측할 수 있다.

이와 같이 Burst Read UDB 동작은 Ack 처리를 한번에 하므로 식 (4)와 같은 시간을 줄여 많은 데이터를 빠르게 전송할 수 있어 대용량 태그 데이터 전송방법으로 더 적합하다.

3.3 데이터양 증대를 위한 메시지 구조 설계

앞서 설명한 그림 2의 ISO/IEC 18000-7 메시지 포맷을 보면 Packet Length 필드가 1byte인 것을 확인할 수 있다. 이는 표준의 Read UDB 프로토콜에서 사용 가능한 메시지 포맷의 전체 길이가 255bytes까지 허용함을 의미한다. 이 중 데이터를 제외한 태그에서 리더로 전송하는 메시지 헤더의 길이는 15bytes를 차지하고 있다. 예를 들어 차세대 RFID 태그는 최소 32kbytes의 저장 공간을 가지고 있는데, 저장된 데이터의 크기가 25kbytes라고 할 경우 길이가 255bytes인 메시지가 100번의 전송을 하여야 하며 이중 1.5kbytes를 헤더가 차지한다. 전체용량 중 5.88%의 용량을 헤더가 차지하는 오버헤드가 발생한다. 기존 메시지 포맷의 길이를 512 bytes로 증가하면 전송횟수가 50번으로 줄어들게 된다. 전송횟수가 줄어들면 50번에 해당하는 헤더의 길이를 줄일 수 있으며, 이에 해당하는 길이만큼 데이터를 더 전송할 수 있다.

그림 7은 메시지 생성 방법에 대한 예를 보이고 있다. 그림 7의 (a)는 하드웨어에 의존적으로, 즉 트랜시버의 FIFO크기만큼 메시지를 전송할 경우를 이야기한다.

(a)와 같이 생성된 메시지에 FIFO버퍼의 크기만큼 패킷을 나누어 4회에 걸쳐서 255bytes 메시지를 전송할 경우, 헤더와 CRC가 차지하는 양에 의하여 실제 전송되는 데이터의 양이 줄어든다. 그림 7의 (b)는 표준에 근거하여 메시지를 생성한 경우를 보이고 있다. 우리는 (a)에서 전송되는 각각의 메시지를 묶어 표준에서 허용

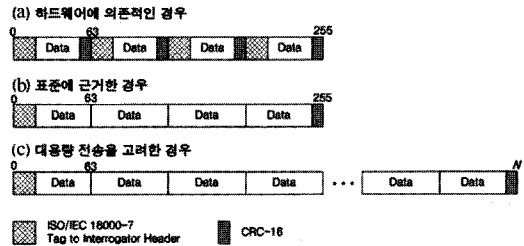


그림 7 메시지 생성 구조 비교

하는 하나의 255bytes 메시지로 만들었다. 하나의 메시지를 4개의 패킷으로 나누어 전송하며, 첫 번째 패킷에 메시지의 헤더가 차지하고 마지막 패킷에 전체 메시지에 대한 CRC가 붙는다. 255bytes를 전송하는 동안 (a)에 비해 헤더와 CRC는 한번만 사용하므로 1/4에 해당하는 오버헤드를 줄일 수 있다. 그림 7의 (c)는 대용량 전송을 위하여 (b)를 개선한 구조이다. 우리는 2.4GHz 대역을 사용하므로 433MHz대역을 기반으로 설계된 기존표준의 255bytes길이보다 그 이상을 전송하기에 무리가 없다. 또한, 메시지가 더 길어졌기 때문에, (b)보다 헤더 및 CRC 오버헤드를 줄여 한 번에 더 많은 데이터를 전송할 수 있다.

그러나 (a)는 메시지의 길이가 작은 대신 에러가 발생하면 해당 메시지만 재전송 받을 수 있지만 (c)의 경우 패킷 하나라도 에러가 발생하거나 소실되면 해당 메시지는 에러로 판별되므로 메시지 전체를 재전송 받아야 한다. 메시지의 전체길이를 증대하는 것은 헤더와 CRC오버헤드를 줄일 수 있지만, 에러 발생확률이 증가하여 원하는 성능이 나오지 않을 수 있다.

개별 에러 패킷 문제를 해결하기 위해서는 전송되는 RF환경을 파악하여 환경에 맞는 적절한 길이를 동적으로 결정하는 방법이 필요하다.

4. 대용량 태그 데이터 전송 프로토콜의 구현

4.1 구현환경

그림 8은 대용량 태그 데이터 전송의 테스트베드를 구현하기 위한 장비를 보여주고 있다. 본 논문에서 사용한 구현장비는 TI사에서 제작한 SmartRF04 Evaluation Board를 사용하였다. MCU는 8051코어를 사용하며, CC2500를 비롯하여 CCxx00계열의 트랜시버 시리즈를 탈 부착할 수 있다.



그림 8 구현환경

4.2 대용량 태그 데이터 전송 프로토콜의 메시지 포맷 및 전송구조 구현

우리는 255bytes이상의 메시지를 전송하기 위하여 CC2500의 Infinite Length Mode를 사용하여 전송하고자 한다. 그러나 Infinite Length Mode는 길이 필드로 2bytes가 필요하므로 표준 헤더메시지 포맷의 길이 부분을 수정하여야 한다. 그림 9는 태그에서 리더로 전송되는 P2P메시지 포맷의 전송구조를 보여주고 있다. 수정된 응답메시지는 Packet Length 필드를 2byte로 지정하여 255bytes이상의 메시지의 길이를 표현할 수 있도록 하였다.

그러나 모든 경우가 255bytes이상이지는 않기 때문에 전송하고자 하는 데이터의 길이에 따라 Packet Length 필드를 조절 할 수 있도록 하였다. 하나의 메시지가 완성이 되면 준비된 전송버퍼에 값을 저장한다. 이후 Packet Length의 크기에 따라 전송모드를 설정하게 되며, 설정에 따라 CC2500의 FIFO에 패킷을 넣으면서 전송을 수행한다.

4.3 Read UDB 및 Burst Read UDB 명령어의 구현

그림 10은 ISO/IEC 18000-7의 AP에서 리더가 태그에게 전송하는 Read UDB 및 Burst Read UDB 메시지 포맷을 나타내고 있다.

Read UDB의 Sequence ID와 Max Packet Length, Type필드는 앞서 3.2절에서 설명한 바 있다. Read UDB는 Sequence ID와 Sequence ID Countdown를 통하여 데이터 전송마다 Ack를 사용하여 에러가 난 UDB를 재

- Interrogator-to-tag message format

command code	command arguments		
	Sequence ID	Max Packet Length	Type
0x70	1 byte	1 byte	1 byte

- Tag-to-Interrogator message format

command code	command arguments	
	Sequence ID Countdown	Universal Data Block
0x70	1 byte	N bytes

그림 10 Read UDB의 메시지 포맷

전송할 수 있다. 만약 해당 Sequence ID에 해당하는 Sequence ID Countdown이 도착하지 않는다면, 리더는 계속 대기해야 한다.

이를 해결하기 위하여 태그로부터 전송 받는 시간을 측정하여 메시지 재전송을 위한 Timeout기능을 구현하였다. 그러나 만약 리더가 명령 메시지를 전송하였지만, 태그에 문제가 발생하여 영원히 응답을 하지 않는 경우에는 Timeout에 따라 명령메시지를 무한 반복하여 전송하므로 이를 보완하기 위하여 Timeout횟수를 5회로 제한하였다.

그림 11은 Burst Read UDB의 메시지 포맷을 나타낸 것이다. Burst Read UDB는 UDB를 모두 전송 받고 한번에 Ack처리를 한다. Max Packet Length필드는 리더에서 태그로 메시지를 전송할 때 태그로부터의 전송 받을 수 있는 패킷의 크기를 지정하고, Control 필드를 통하여 태그의 데이터 전송을 제어한다. Control필드는 Control Command의 종류에 따라 3가지로 나눈다.

Burst Read UDB명령의 가장 처음단계에 전송시작명령을 내리기 위하여 Control Command필드를 0x00으로 설정하여 태그에게 전송한다. 태그로부터 전송 받은 UDB들 중, 에러가 발생하거나 수신을 받지 못한 경우 리더는 Control Command를 0xEE로 설정하여 에러가 난 개수와 에러를 회복하기 위하여 재전송 받을 Sequence ID들을 기록하여 태그에게 전송한다. 에러회복

- Tag-to-Interrogator response message format(Point-to-Point)

Protocol ID	Tag Status	Packet Length	interrogator ID	Tag Manufacturer ID	Tag Serial Number	command code	Response Data	CRC-CCITT (XModem)
1 byte	2 bytes	1 byte	2 bytes	2 bytes	4 byte	1 byte	N bytes	2 bytes

- 수정된 Tag-to-Interrogator response message format(Point-to-Point)

Protocol ID	Tag Status	Packet Length	interrogator ID	Tag Manufacturer ID	Tag Serial Number	command code	Response Data	CRC-CCITT (XModem)
1 byte	2 bytes	2 byte	2 bytes	2 bytes	4 byte	1 byte	N bytes	2 bytes

그림 9 255bytes이상 표현을 위한 Packet Length Field 수정

- Interrogator-to-tag message format

command code	command arguments		
	Max Packet Length	Type	Control
0x78	2 byte	1 byte	N bytes

Control command	Error Count	Error Sequence ID
0xEE	1 byte	N byte

Control command
0x00

Control command
0xFF

- Tag-to-interrogator message format

command code	command arguments		
	Total Sequence ID Count	Sequence ID Countdown	Universal Data Block
0x70	1 byte	1 byte	N bytes

그림 11 Burst Read UDB의 메시지 포맷

이 모두 끝나고, UDB를 모두 전송 받았다면 0xFF를 전송하여 동작이 모두 끝이 났음을 태그에게 알린다.

태그가 리더에게 UDB를 전송할 때, 총 몇 개의 UDB를 전송할 것인지 기록하는 Total Sequence ID Count가 있다. 이 필드가 없다면 리더가 순서대로 Sequence ID를 수신하였을 때, 가장 처음부터 에러가 나거나 전송되지 않을 경우 리더는 태그가 몇 개의 전송을 완료하였는지 알 수 없다. 그러나 Total Sequence ID Count 필드를 사용하면 전송 하고자 하는 메시지들을 안전하게 전송할 수 있다.

5. 실험 및 성능평가

5.1 실험 환경 및 방법

본 논문에서의 실험환경은 그림 12와 같다. Smart-RF04 개발보드를 이용하여 리더와 태그를 구성하였으며, 각각의 거리는 50m로 고정하였다. 데이터 전송과정을 파악하기 위하여 2대의 노트북으로 모니터링 하였다. 리더와 태그의 전송속도는 250kbps로 설정하였다.

실험은 야외에서 진행되었으며 다음과 같이 수행되었다. 첫째, 대용량 전송을 위하여 메시지 생성 크기에 따른 성능을 측정하였다. 둘째, 효과적인 대용량 전송을 위하여 기존의 메시지 전송방법과 제안한 방법의 성능을 각각 측정하였다.

5.2 메시지 생성구조에 따른 대용량전송 성능 비교

그림 13은 전송할 메시지길이에 따른 초당 데이터 전송률을 보여주고 있다. 15bytes헤더를 포함하고 메시지

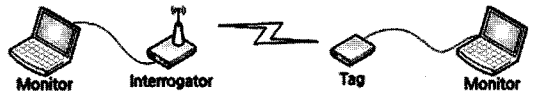


그림 12 실험 환경

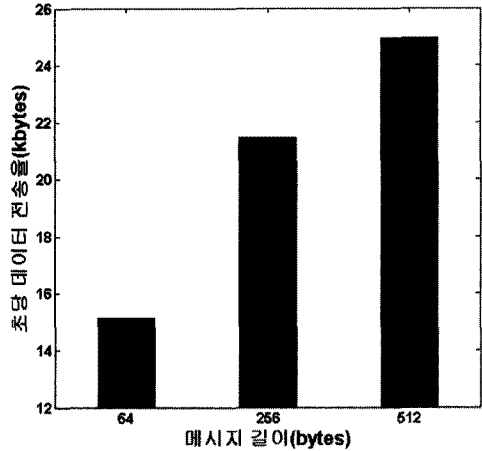


그림 13 메시지 생성 구조에 따른 초당 데이터 전송률

의 길이를 다르게 하여 102,400bytes의 데이터를 수신하는 데 걸리는 시간을 측정하였고, 이를 통해 초당 데이터 전송률을 구하였다.

실험결과에서 보는 바와 같이 512bytes의 메시지 길이로 전송하는 것이 64bytes일 때보다 9,811bytes를 더 전송할 수 있는 것으로 파악되었다. 64bytes가 512bytes에 비하여 상대적으로 전송률이 떨어지는 이유는 메시지 전송 시 메시지마다 포함되는 헤더로 인하여 헤더 오버헤드가 발생하기 때문이다. 64bytes를 사용하여 데이터를 전송할 경우 전체 패킷이 102,400bytes동안 전송되면서 실제 전송되는 데이터는 78,400bytes로 헤더가 23.4%에 해당하는 오버헤드를 차지하고 있다. 반면에 512bytes를 사용하여 전송할 경우 실제 전송되는 데이터는 99,400bytes이며, 헤더가 2.9%에 해당한다. 이와 같이 대용량 전송을 하기 위해서는, 메시지 길이를 크게 하는 것이 전송률을 높일 수 있다는 것을 알 수 있다.

그러나 256bytes와 512bytes를 비교하였을 때, 2배의 길이를 전송함에도 불구하고 초당 데이터 전송률이 2배에 가까이 차이가 나지 않는다. 이는 64bytes의 메시지를 전송할 경우에는 에러가 발생하면 해당 메시지를 재전송하면 되지만, 512bytes 메시지의 경우는 패킷을 전송하는 횟수가 더 많기 때문에 전송도중 패킷이 하나라도 소실되거나 에러가 발생할 확률이 높다. 특히, 패킷이 하나라도 문제가 생기면 해당하는 메시지는 에러로 판별되므로 메시지 전체를 다시 재전송한다. 이와 같이

에러가 많이 발생하는 RF환경에서 메시지의 길이를 크게 늘려 전송하면 기대했던 성능이 나오지 않을 수 있다. 반면 RF환경이 좋다면 메시지의 길이를 더 늘리는 것이 유효한 방법이다.

차후 RF환경을 분석하여 전송 메시지 크기를 동적으로 결정하는 방법과, 부분 패킷의 에러 발생 시 해당 패킷만 복구하는 방법이 추가된다면 더 좋은 성능을 낼 수 있을 것이다.

5.3 Read UDB vs. Burst Read UDB 명령어의 수행시간 비교

그림 14는 태그 데이터 전송방법에서 태그 데이터 양에 따른 수행시간을 보여준다. 명령어들의 비교를 위하여 메시지의 크기는 헤더 포함 255bytes로 고정하였다. 그리고 전송되는 횟수를 10번부터 시작하여 10회씩 증가시켰고, 최고 250번까지 늘렸다. 각각의 전송횟수마다 100회 반복하여 측정하였다. 결과값은 100회 반복한 측정값의 평균을 뜻한다. 그림 14에서 보여주는 수행시간은 에러 회복을 위한 재전송 시간을 포함하여 데이터를 모두 전송받는 시간을 말한다.

실험결과에서 보는 바와 같이 Burst Read UDB가 Read UDB보다 전송되는 데이터의 양이 커질 때 성능의 향상 정도가 선형적으로 증가하는 것을 볼 수 있다. 2.55kbytes를 전송하는데 있어 Read UDB는 125.2ms이고, Burst Read UDB는 111ms가 걸렸다. 전송되는 데이터의 양이 적을 경우에는 14ms정도의 차이밖에 보이지 않지만, 63.75kbytes를 전송할 경우에는 차이가 555.8ms로 Read UDB수행시간에 비해 17.95%의 성능이 향상되었다.

전송하는 태그 데이터의 양이 증가할수록 선형적으로 차이가 벌어지는 이유는 전송횟수가 많아질수록 에러가 발생하기 때문인데 Read UDB의 경우에는 에러 당 Ack메시지와 메시지 재전송시간이 발생하기 때문에 일정한 기울기를 유지하며 증가한다. 그러나 Burst Read UDB는 여러 개의 에러가 발생하였을 경우 한번의 Ack 메시지만 전송하면 전체 에러메시지를 재전송 받을 수 있다. 이러한 차이로 인해 Read UDB보다 차이를 보이며 증가한다.

그림 15는 그림 14의 내용을 전송속도로 표현한 것이다. 그림에서 보는 바와 같이 Burst Read UDB가 전체적인 전송속도가 좋음을 볼 수 있다. Read UDB의 경우에는 태그 데이터양의 차이에 따라 전송속도의 변화가 170kbps~179kbps사이를 유지하지만, Burst Read UDB는 처음 186kbps전송속도에서 63.75kbytes를 전송할 때에는 201kbps까지 점차 증가하여 유지한다. 이는 Read UDB의 경우 에러 당 Ack메시지를 수신하기 때문에 정해진 시간 내에 전송되는 비트의 수가 일정하기

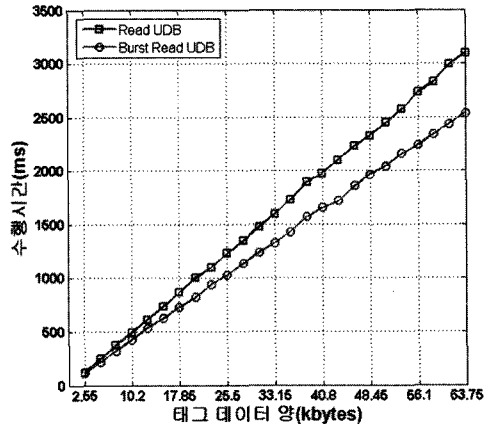


그림 14 태그 데이터 양에 따른 수행시간 비교

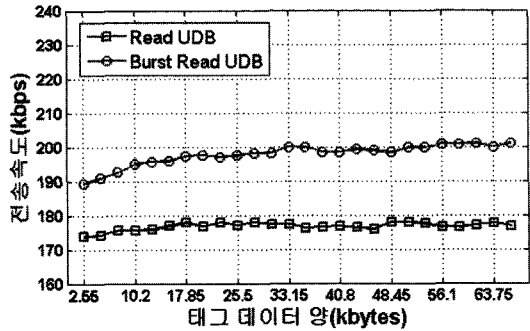


그림 15 태그 데이터 양에 따른 전송속도 비교

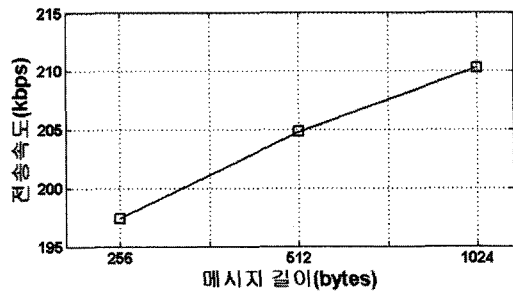


그림 16 Burst Read UDB 명령어의 전송 용량 증대를 통한 전송속도 측정

때문이다. 그러나 Burst Read UDB는 에러의 개수에 상관없이 Ack메시지를 한번만 전송하기 때문에 송수신 변화에 따른 오버헤드가 적게 발생하며 이는 동일한 시간에 재전송 메시지만 전송하므로 결과적으로 전송속도의 증가로 이어진다.

그림 16은 Burst Read UDB명령어에서 40kbytes를 전송할 때, 메시지의 전송용량을 변화시켜 각각의 전송속도를 측정할 것이다. Read UDB는 메시지 전송용량

이 256bytes이상 사용할 수 없으므로 이 경우 측정할 수 없다. 측정 결과를 보면 메시지의 크기가 증가하여 전송할수록 전송속도가 증가하는 것을 확인할 수 있다. 즉, Burst Read UDB명령어를 사용하여 40kbytes의 데이터를 전송할 때 그림 15는 255bytes의 메시지의 크기로 200 kbps정도의 속도를 가지지만 메시지의 크기가 1024bytes로 증가시킨 경우 전송속도가 10kbps의 전송속도가 향상된다.

6. 결론

능동형 RFID가 많은 응용 분야에서 활용되기 위해서는 대용량의 태그 데이터를 보다 빨리 그리고 효율적으로 보낼 필요가 있다. 본 연구진은 ISO/IEC 18000-7 표준보다 빠른 대용량 태그 데이터 전송을 위하여 2.4GHz대역을 사용하였다. 그리고 보다 효율적인 데이터 전송을 위한 프로토콜을 설계하였다. 프로토콜은 Burst Read UDB명령어를 이용하여 255bytes이상의 메시지 데이터를 전송할 수 있도록 설계되었으며, TI사에서 개발한 Smart RF04 개발보드와 CC2500트랜시버를 이용하여 실제 구현하였다.

이를 통해 리더와 태그를 구성하였으며, 구현된 시스템을 이용하여 메시지 길이에 따른 초당 데이터 전송을 측정을 위한 실험을 하였다. 실험결과, 한 번에 전송할 수 있는 메시지의 용량이 커질수록 데이터 전송율이 증가함을 확인 하였다. 그러나 무리한 메시지 용량의 증가는 에러율 발생확률을 증가 시킨다. 이를 해결하기 위하여 RF 환경에 따른 적절한 메시지의 크기를 결정하여야 하며, 향후 RF환경을 분석하여 메시지 크기를 동적으로 결정할 수 있다면 더욱 좋은 성능을 보일 것이라 예상된다.

Read UDB와 Burst Read UDB 명령어의 수행시간 비교실험에서는 64.75kbytes의 데이터를 전송할 경우, 기존의 Read UDB의 전송방법보다 Burst Read UDB가 17.95%의 전송시간 성능이 향상되었음을 확인하였다. 또한 Burst Read UDB는 데이터 전송 횟수가 증가하여 에러가 발생할수록 Ack메시지에 의한 오버헤드가 감소하기 때문에 Read UDB에 비해 전송속도 측면에서 20kbps 정도 더 우수했다.

우리는 향후 대용량을 전송할 수 있는 실제 태그를 구현하고, 여러 개의 대용량 태그를 준비한 환경에서 제안하는 대용량 태그 데이터 프로토콜에 대한 가능성과 우수성을 보이도록 할 것이다.

참고 문헌

[1] Klaus Finkenzeller, "RFID Handbook: fundamentals and applications in contactless smart cards and identification," Wiley press, 2003.

- [2] 오세원, 표철식, 채종석, "RFID 표준화 및 기술 동향", 전자통신동향분석, 제20권 제3호, pp.56-66, 2005. 6.
- [3] H. Vogt, "Efficient Object Identification with Passive RFID Tags," *Proceedings of International Conference on Pervasive Computing*, Zurich, Swiss, 2002.
- [4] 권성호, 홍원기, 이용두, 김희철, "저비용 RFID 시스템에서의 충돌방지 알고리즘에 대한 성능평가", 한국통신학회논문지, pp.17-26, 제30권, 1B호, 2005.
- [5] ISO/IEC 24730-2, "Real-time locating systems (RTLS) -- Part 2: 2.4 GHz air interface protocol," 2006.
- [6] D. Engels, "The Reader Collision Problem," White Paper of Auto-ID Center, Nov. 2001, available at <http://www.autoidlabs.org/single-view/dir/article/6/98/page.html>
- [7] K.S. Leong, M.L. Ng and P.H. Cole, "Synchronization of RFID readers for dense RFID reader environments," *International Symposium on Applications and Internet Workshops*, Phoenix, AZ., 2006.
- [8] 표준화연구보고서, 한국정보통신기술협회, 2009. 3., <http://www.itfind.or.kr/itfind/getFile.htm?identifier=02-004-090311-000014>
- [9] Active IHF Tag i-Q310 ATR ISO 18000-7 - Container Security Tag, SPEC, <http://rfid.spec.com/index.php/ordering-catalog/active-rfid-transponders-/container-tag>
- [10] 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 433 MHz," 2008.
- [11] 2.4Ghz Long Range Active RFID, SynoMetrix, <http://www.synometrix.com/>
- [12] Open Beacon Project - a free active 2.4Ghz beacon design, <http://www.openbeacon.org>
- [13] CC2500/CC2550DK Development Kit, <http://focus.ti.com/docs/toolsw/folders/print/cc2500-cc2550dk.html>
- [14] 박승창, "900MHz 대역 차세대 RFID의 통신 기술 동향", 정보통신연구진흥원, 2008. 4. 15. <http://www.iita.re.kr>
- [15] CC2500 Transceiver, Texas Instruments, <http://focus.ti.com/docs/prod/folders/print/cc2500.html>
- [16] H. Cho, Y. Baek, "Design and implementation of an active RFID system platform," *Applications and the Internet Workshops, 2006. SAINT Workshops 2006. International Symposium on*, p.4, 23-27, Jan. 2006.
- [17] Won-Ju Yoon, Sang-Hwa Chung, Hyun-Pil Kim, Seong-Joon Lee, Implementation of a 433 MHz Active RFID System for U-Port," *Advanced Communication Technology, The 9th International Conference on*, vol.1, pp.106-109, 12-14 Feb. 2007.
- [18] 박현성, 김동현, 정상화, 백윤주, 김종덕, "다중 채널 다중 인터페이스 능동형 RFID 리더 및 프로토콜", 한국정보과학회, 정보과학회논문지: 정보통신 제36권 제2호, pp.118-129, 2009. 4.

- [19] Won-Ju Yoon, Sang-Hwa Chung, Seong-Joon Lee, "Implementation and performance evaluation of an active RFID system for fast tag collection," *Computer Communications*, 31(17), Nov. 2008.
- [20] Jihoon Myung, Wonjun Lee, Jaideep Srivastava, Timothy K. Shih, "Tag-Splitting: Adaptive Collision Arbitration Protocols for RFID Tag Identification," *IEEE Transaction on Parallel and Distributed Systems*, 18(6), pp.763-775, Jun. 2007.



이 채 석

2008년 경성대학교 컴퓨터공학과 학사
현재 부산대학교 컴퓨터공학과 석사과정
관심분야는 무선통신, RFID/USN



김 동 현

1998년 동의대학교 정보통신공학과 학사
2004년 동의대학교 정보통신공학과 석사
현재 부산대학교 컴퓨터공학과 박사과정
관심분야는 이동통신망, RFID/USN



김 종 덕

1994년 서울대학교 계산통계학과 학사. 1996
년 서울대학교 전산학과 석사. 2003년
서울대학교 컴퓨터공학과 박사. 2004년~
현재 부산대학교 컴퓨터공학과 조교수,
컴퓨터 및 정보통신연구소 연구원. 관심
분야는 무선통신, 이동통신망, RFID/USN