http://dx.doi.org/10.7840/kics.2015.40.5.871

능동형 RFID 시스템의 성능 향상을 위한 연구

김 지 태◆, 김 진 성*, 이 강 원°

Study on the Performance Improvement of Active RFID System

Ji-tae Kim◆, Jin-sung Kim*, Kang-won Lee°

요 약

본 연구에서는 우선 고속으로 태그 수집을 위한 2.4 GHz 능동형 RFID 시스템의 시뮬레이션 모델을 구축 하였다. 그리고 단순화 된 Collection 명령과 Ack 절차, 슬롯의 충돌 확률(k1)과 충돌이 발생한 슬롯의 평균 태그 수(k_2)를 이용하여 태그수를 예측하는 새로운 방법을 제안하였다 이들에 따른 능동형 RFID 시스템의 성능 추적을 구축한 시뮬레이션 모델을 통해 획득하였다. Query 명령을 사용하여 Collection 명령과 Ack를 간소화하고 k_1과 k_2를 이용하여 태그수를 예측하는 방법의 조합이 모든 성능 면에서 가장 우수한 것으로 나타났다. 이는 Query 명령을 이용한 Collection 명령과 Ack의 간소화는 태그 인식 속도를 줄이는데 기여했고 k_1과 k_2를 이용하여 태그수를 예측하는 방법은 Throughput을 향상시킬 뿐만 아니라 이로 인해 Collection 라운드 횟수를 줄여 태그 인식 속도를 줄일 수 있었다. 본 연구에서 제안한 방법은 Throughput, 평균 인식 시간, 1초 안의 평균 인식률 모두에서 이상적인 경우의 성능값과 매우 유사하게 나타나 제안한 방법의 유효성을 확인 할 수 있었다.

Key Words : Active RFID, Slotted Aloha, Tag Collection, ISO/IEC 18000-7, Query Command

ABSTRACT

The improved DFSA for 2.4GHz multi-tags active RFID is suggested in 2 different ways: 1) simplified tag collection and Ack procedure using query command and 2) modified Schoute's method to control the number of slots in the frame. To evaluate the performance of the improved system we develop the simulation model. Varying the number of tags in the system we track the performance measures such as throughput, recognition time for multi-tags and tag recognition rate during a given time. The suggested method shows the best performance over all measures. Simplification of collection and Ack commands using query commands contributes to reducing tag recognition time. And the modified Schoute's method which controls the frame size using k_1 and k_2 contributes to throughput improvement and reduces target cognition time by reducing the number of collection rounds.

Ⅰ. 서 론

RFID 시스템은 Tag 전원 공급의 유무에 따라 전원 장치가 있는 능동형(Active)과 내부나 외부로부터 직접적인 전원 공급을 리더기의 전자기장에 의해 작동하게 하는 수동형(Passive)으로 구분된다. 지난 2004

※ 본 연구는 서울과학기술대학교 교내연구비의 지원으로 수행되었습니다.
◆ First Author : Seoul National University of Science & Technology, Graduate School of Public Policy and Information Technology, jtk82@seoultech.ac.kr, 정회원
° Corresponding Author : Seoul National University of Science & Technology, Graduate School of Public Policy and Information Technology, kwlee@seoultech.ac.kr, 정회원
* Yonsei University, Department of Information and Industrial Engineering, eoee2522@yonsei.ac.kr
논문번호 : KICS2015-04-114, Received April 6, 2015; Revised April 29, 2015; Accepted April 29, 2015

년부터 정부 주도아래 RFID/USN 시장의 조기 정착과 산업 활성화를 위해 다양한 산업을 추진하여 수동형 RFID 시스템은 시장이 활성화 되어있다.

능동형 RFID 시스템은 수동형 RFID 시스템에 비해 태그의 가격, 크기, 배터리 관리 등의 측면에서는 불리하지만 인식 거리, 인식률, 안정성, 부가기능 등에서는 훨씬 우수한 성능을 나타낸다. 따라서 능동형 RFID는 건설현장과 같이 대규모 인원의 출입현황을 실시간으로 확인하는 시스템이라든지 고가의 중, 대형 물품이나 컨테이너의 추적 및 관리 분야에서 실용화가 빠르게 진행 중이다.

대표적인 능동형 RFID 시스템으로는 433MHz 대역의 주파수를 사용하는 ISO 18000-7 능동형 RFID 표준이 있는데 채널대역은 200KHz인 단일 채널 시스템으로 전송속도는 27.8Kbps 수준이다[1]. SAVI 가미 국방성 컨테이너 관리용으로 제작한 433MHz 능동형 RFID 기술이 가장 대표적이다. 그런데 433 MHz 대역의 RFID는 낮은 전송속도와 단일 채널에 따른 간섭 문제, 다른 무선 통신 대역과의 충돌 등의 문제들로 인해 최근의 다양한 시장 욕구를 충족시키는데 어려움이 있다[2-3]. 따라서 ISO 18000-7 능동형 RFID 표준이 가지는 이러한 한계를 극복하기 위해 2.4GHz ISM 대역에서 다수 채널 및 빠른 데이터 송수신이 가능한 능동형 RFID 시스템이 제안되었다[4].

국외에서 2.4GHz 능동형 RFID 시스템은 실시간 위치 확인용으로 개발되어 시장을 형성하고 있다. 2.4GHz 대역의 다수 채널 및 빠른 데이터 송수신 속도를 이용하여 다양한 응용 기술을 내놓고 있으나 아직 킬러 애플리케이션은 만들어 내지 못하고 있다. 현재 몇몇 업체들이[5] 새로운 시스템인 'Hands Free for Access Control System'(자동 출입 시스템)을 내놓고 있지만 이 시스템이 시장에서 제대로 인정받기 위해서는 저주파 Wake-up 기술과 고속의 멀티 태깅 인식 기능이 필수 불가결하다. 한편 국내에서의 2.4GHz 능동형 RFID 시스템은 단순 원거리 사물인식 및 위치 추적용으로만 개발하고 있다. 저주파 Wake-up 기술과 고속의 멀티 태깅 인식기능은 논문상으로 발표 되었을 뿐[6] 실제 구현 및 실증된 사례는 전무하다.

따라서 국내외적으로 능동형 RFID 시스템 확산을 위해 절실히 요구되는 사항은 고속의 다중 태그 인식 속도, 신뢰성 있는 태그 인식률, 배터리 교체 없이 장기간 사용 할 수 있는 태그 등이다. 이를 위해서는 다중 태그 인식률과 인식 속도를 높이고 저주파 Wake-up 기술을 사용하여 태그 배터리 동작시간을 크게 낮추는 연구와 제품개발이 필요한 시점이다.

본 연구의 선행 연구인 참고 문헌[4]에서는 능동형 RFID 시스템용 고속 태그 수집을 위한 개선된 알로하 알고리즘을 제안하였다. 이를 위해 TI 사의 CC2530 칩을 UHF부에 태그의 LF 수신부에는 Microchip 사의 MCP 2030칩을 각각 사용하여 태그와 리더기 각각의 기능을 구현하였다. 기존의 ISO 18000-7 능동형 RFID 규격에서 제시하고 있는 태그 수집 알고리즘에서는 리더기와 태그의 통신이 성공적으로 종료된 후 리더기가 각 태그들에게 별도의 슬립 명령을 전송한다. 반면에 제안된 방식에서는 리더의 명령과 Ack 정보를 통합하여 전송함으로 매 슬롯마다의 응답 여부에 따른 Ack 대기 시간이 필요하지 않아 절차상의 수신 대기 시간을 대폭 축소시켰다. 슬롯 수 제어는 충돌 슬롯의 수신 상황에 따라 전체 슬롯의 수를 변경하여 진행한다. 최소 슬롯의 수는 8 개로 설정하고 충돌 슬롯의 수가 전체 슬롯 수에 비해 1/8 미만일 경우에는 1/2 로 감소하고 1/8 이상 1/4 미만인 경우에는 동일한 슬롯 수로 하고 1/4 이상인 경우에는 2 배로 증가시킨다. 태그의 수를 30개로 고정하고 1.5m ~ 3.0m 에서 실험한 결과 성능이 우수하게 나타났다.

본 연구에서는 선행 연구[4]에서 실험적으로 구한 능동형 RFID의 성능 검증을 시뮬레이션을 통해 보다 일반화 하고 또한 성능개선을 위해 새로운 슬롯 수 제어 방법을 제안하였다. 이를 위해 다음의 측면에서 연구를 수행하였다. 첫째, 선행 연구에서 제안한 고속으로 태그 수집을 위한 2.4 GHz 능동형 RFID 시스템의 시뮬레이션 모델을 구축 하였다. 둘째, 실험에서 사용하였던 태그의 수 30 개를 증가 시켜나가면서 시뮬레이션을 수행하여 태그 수 증가에 따른 능동형 RFID 성능을 추적하였다. 셋째, 단순화 된 Collection 명령과 Ack, 그리고 충돌 슬롯수에 따른 다음 프레임의 슬롯 수 결정을 위한 새로운 방법을 제안하고 이들에 따른 능동형 RFID 시스템의 성능을 추적하였다.

이를 위해 본 연구에서는 1 장 서론에 이어 2 장에서 고속 태그 수집을 위한 능동형 RFID 시스템에 대하여 논하였다. 본 연구에서 시뮬레이션 하고자 하는 2.4 GHz 능동형 RFID 시스템의 개선된 프레임 슬롯티드(Slotted) 알로하 알고리즘, 충돌 슬롯수에 따른 다음 프레임의 슬롯 수 결정을 위한 방법, 그리고 효율적인 Sleep 기법에 대해 서술하였다. 논문에서 제안하는 2.4GHz 능동형 RFID 시스템의 인식 및 통신과정의 기본 틀은 기존 ISO 18000-7 능동형 RFID 표준을 따르도록 하였다. 3장에서는 시뮬레이션을 위한 입력 자료와 모델을 다루었으며 4장에서 결과 및 이에 대한 분석을 수행하였다. 그리고 최종 5장에 결론과

추후 연구 방향에 대하여 구술 하였다.

Ⅱ. 2.4㎓ 능동형 RFID 시스템의 성능 개선

본 장에서는 2.4 GHz 능동형 RFID 시스템의 성능 개선을 위해 필요한 사항을 정리하였다.

2.1 개선된 프레임 슬롯티드 알로하 알고리즘

2.1.1 Collection과 Ack 절차의 간소화

기존의 능동형 RFID 표준의 프레임 슬롯티드 알로하에서는 그림 1과 같이 “Collection 명령 → 태그 응답 → 식별한 태그들로 슬립 명령 → Collection 명령” 절차가 더 이상의 태그 수신이 없을 때까지 반복된다.

위 절차에서 리더기가 성공적으로 정보를 전송한 태그에게 슬립 명령을 내릴 때 이 명령어는 하나의 태그 마다 각각 전송 된다. 이때 태그 ID 4 바이트 및 리더기 ID 2 바이트를 포함하여 최소 14 바이트가 필요하기 때문에 총 소요 시간은 정보 전송을 성공한 태그의 개수 곱하기 14 바이트 시간 이다. 본 논문에서 제안한 새로운 방식은 태그의 응답 프레임에 대한 Ack 정보를 컬렉션 명령에 포함한다. 명령과 Ack 정보를 통합하여 전송함으로 태그는 Ack 정보 수신을 위한 별도의 슬롯이 필요하지 않아 절차상의 수신 대기 시간을 대폭 축소 시켰다. Ack 정보를 표현하는 방식은 Slot-bit 와 태그에서 생성한 Random Number(Tag ID 보다 적은 데이터 량으로 이는 리더가 다음 명령의 파라미터로 충분히 표현 할 수 있어야 함)로 표현하여 Ack 정보를 함축하는 것이 가능해졌다. 이를 통해 다음 명령인 하나의 패킷에 이전 프레임에서 수신한 모든 태그들에 대한 성공적 수신 여부를 표현 할 수 있는데 이를 위해 Query/Repeat 혹은 Query/Adjust 명령을 제안하였다.

태그들은 Query/Repeat 혹은 Query/Adjust 명령을 수신하여 자신의 정보가 성공적으로 전송 되었는지를 판단하며, 만일 성공하였을 경우 슬립 모드로 들어가고 실패하였을 경우 다음 수집 라운드에 참여하게 된다. 따라서 본 알고리즘을 사용하게 되면 기존의 방식에서 하나의 태그에게 슬립 명령을 전송할 시간에 태그 수집에 참여한 모든 태그에게 Query 명령을 전송할 수 있다.

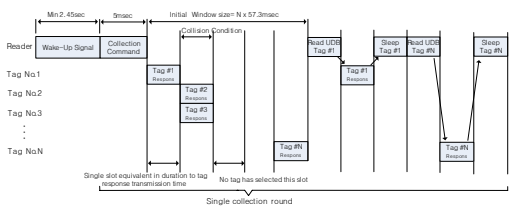

그림 1. 기존의 능동형RFID 표준의 프레임 슬롯티드 알로하
Fig. 1. Frame sloted ALOHA of active RFID standard

2.1.2 제안한 명령

리더에서 태그로 명령은 Query, Query/Repeat, 그리고 Query/Adjust 3가지다. Query 명령은 태그 정보 수집 라운드에서 사용할 타이밍 및 최초 슬롯 수를 지정하며, Query/Repeat 는 이전 슬롯의 수와 동일한 수로 라운드를 반복 할 때, Query/Adjust는 수정된 슬롯 수로 라운드를 진행할 때 사용된다. Query/Repeat 나 Query/Adjust 명령에는 다음 표 1과 표 2에서 볼 수 있듯이 이전 프레임에서 사용한 슬롯의 개수를 나타내는 비트 길이 Q를 포함한다. 표 1과 표 2의 슬롯 비트는 Q 값에 따라 정의 되는데 첫 번째 비트는 첫 번째 슬롯을 의미하며 마지막 비트는 마지막 슬롯을 의미한다. 이때 비트 값이 1 이면 첫 번째 슬롯을 사용했던 태그의 전송이 성공했음을 의미하고 0 이면 실패를 의미한다.

Random Number는 태그에서 생성한 Random Number를 나타나는데 태그는 이 정보를 수신하여 자기가 전송한 Random Number와 일치 하는지 여부를 판단하여 전송 성공 여부를 판단하는 절차를 한 번 더 갖는다. 이는 기존에 태그별로 별도의 슬립 명령으로 태그를 슬립으로 전환시키는 것과는 다르게 Query/Repeat 나 Query/Adjust 명령을 통해 리더가 성공적으로 응답 정보를 수신한 슬롯의 태그들을 슬립으로 전환시킨다. 표 2의 NewQ는 다음 프레임에서 사용될 수정된 슬롯수를 나타낸다.

리더로 부터의 수집 명령인 Query, Query/Repeat, Query/Adjust 에 대하여 태그는 수집 명령을 받은 리더 ID와 명령어 Sequence No.를 그대로 리턴하여 리

표 1. Query/Repeat 명령어 포맷
Table 1. Query/Repeat command format.

CMD	Reader ID	Seq. No.	Length of bit	Slot-bit(s)					Random Number(s)			CRC
0x66	2 bytes	1 byte	Q_{t-1}	Change according to the Q_{t-1}					listed by the number of“1”			2 bytes
				0	1	2	…	Q_{t-1}-1	RN_1	RN_2	…	

표 2. Query/Adjust 명령어 포맷
Table 2. Query/Adjust command format.

CMD	Reader ID	Seq. No.	New Q	Length of Bit	Slot-Bit(s)					Random Number(s)			CRC
0x77	2 bytes	1 byte	1 byte	Q_{t-1}	Change according to the Q_{t-1}					listed by the number of“1”			2 bytes
					0	1	2	…	Q_{t-1}-1	RN_1	RN_2	…	

표 3. 태그응답 명령어 포맷
Table 3. Tag response command format.

Reader ID	Seq. No.	Tag ID	RN-8	Battery Status	RSSI (LQI)	CRC
2 bytes	1 byte	8 bytes	1 byte	1 byte	2 bytes	2 bytes

더는 자신의 수집 명령에 대한 회신인지를 구분한다. 이외에 태그 응답에는 표 3과 같이 태그 ID, Random Number, Battery Status 등의 정보를 포함한다.

2.1.3 제안한 기법의 태그 정보 수집 절차

제안한 고속 태그 수집을 위한 개선된 프레임 슬롯티드 알로하 알고리즘에서는 태그를 효율적으로 제어하기 위해서 LF 대역과 UHF 대역 두 개의 무선 주파수를 혼합하여 운용한다[4]. 리더는 LF 대역의 Wake-up 신호 전송으로 태그와의 통신을 시작하는 Inquiry 방식으로 운영한다. 리더와 태그간의 통신에서 발생 할 수 있는 오차로 인한 시간 손실을 최소화하기 위하여 명확하게 구분되는 통신 시점과 통신시간이 절차상에 포함된다.

(1) 리더는 LF 대역으로 Wake-Up신호를 송출하여 LF 통신영역내의 모든 태그를 깨운다.

(2) 깨어난 태그는 LF 대역의 Wake-up신호에 포함된 UHF 대역의 통신채널 데이터를 수신하여 UHF 통신채널을 설정하고, LF 대역의 Wake-up 패킷 종료시점을 기준으로 일정 시간동안 UHF 대역의 Query 명령 수신을 대기한다.

(3) 이에 리더는 Wake-up 패킷의 종료 시점을 기준으로 약 2msec 지연 후 UHF 대역으로 최초 명령인 Query 명령을 태그에 전송하며, 리더와 태그는 명령 전송 혹은 수신 시작 시점의 시간을 기준으로 시간 정보를 다시 동기화 한다.

(4) 리더 및 태그는 Query 명령 패킷의 종료 시점을 기준으로 각각의 수행 지연시간(보통 2ms) 이후에서부터 Listen 구간을 시작한다. 태그는 수신한 Query 명령에서 슬롯의 수, Pre-gap 시간과 Post-gap 시간 및 슬롯 시간의 정보를 기반으로 타임 슬롯을 나누어 설정하고 랜덤으로 응답 슬롯을 선택한다. 응답 슬롯에서 송신 전에 Pre-gap 시간을 지연한 후 송신한다. 태그는 Query 명령 수신 후 전송할 슬롯을 선택한 다음 자신의 응답 슬롯까지는 Sleep 상태에 있으며, 자신의 슬롯에서 응답한 후 다시 다음 명령(Query/Repeat 등)을 수신할 시점까지 Sleep 상태로 전환된다.

(5) 리더는 태그의 Gap 시간을 포함한 시간을 슬롯 시간으로 간주하여 태그 응답을 청취한다.

(6) 리더는 전체 슬롯이 끝나는 시점에서 다음 명령(Query/Repeat 등)을 전송하는 시점 사이에 Round-gap 시간(보통 2ms)을 갖게 되며, Gap 시간동안에 슬롯 응답의 내용을 분석하여, 다음 명령을 선택하고 송신한다.

(7) 다음 명령 수신 시점에서 스스로 깨어난 태그는 일정 시간 리더의 명령을 수신하고, 명령에 포함된 인지 정보를 확인한다. 자신의 응답에 대한 인지가 있는 경우 바로 Sleep으로 전환되어 다음 LF 대역의 Wake-up 신호를 수신할 때까지 응답하지 않는다. 만일 인지가 없는 경우 태그는 명령(Query/Repeat 등) 수신 시점을 기준으로 다시 시간을 동기화하고 각각의 수행 지연 시간(보통 2ms) 이후에서부터 응답 슬롯을 시작하여 절차를 반복하다.

제안된 태그수집 단계의 전체 과정을 그림 2에 나타내었다. "Time Sync"로 표시된 부분은 매 시간 정보를 동기화하는 시점이고, 슬롯 시작 전 "Delay after Rx" 시간이 주어지며, 슬롯이 종료된 후 다시 명령어 전송 사이에는 Round_gap Time이 별도로 주어진다.

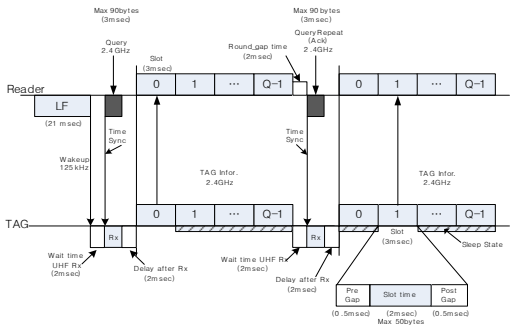

그림 2. 개선된 프레임 슬롯티드 알로하 알고리즘
Fig. 2. Improved frame sloted ALOHA algorithm

2.2 충돌 슬롯 수에 따른 다음 프레임의 슬롯 수 결정을 위한 방법

알로하 방식에서 처리율(Throughput)이 최대가 되는 한 프레임의 최적 슬롯 수는 정보를 전송하는 태그의 숫자와 같다[7]. 그런데 특정 프레임에서 실제 정보를 전송하는 태그의 수를 알 수 없기 때문에 현 프레임의 충돌 슬롯 수 등 수신 상황을 토대로 다음 프레임의 전체 슬롯수를 정해 나가는 방법을 택한다.

2.2.1 기존의 태그 개수 추정 방법

알로하 기반 태그 추정 방법은 크게 정적 추정 방법과 동적 추정 방법으로 나눌 수 있다. 먼저 정적 추

정 방법은 태그 수집 과정의 한 라운드 진행 후에 얻어진 파라미터 값과 미리 정해 놓은 상수값으로 단일 연산을 통해 다음 태그 수를 추정하는 방식이다. 다음으로 동적 추정 방법은 태그 수집 과정의 한 라운드 진행 후에 얻어진 파라미터 값으로 기댓값을 구하여 그에 따라서 태그 수를 추정하는 방식으로 여러 라운드의 파라미터값을 이용하기도 한다.

(1) 동적인 태그 수 추정 방법

태그 수집과정에서 일정한 슬롯 수에서 태그 개수를 알고 있다고 가정하면, 한 슬롯에 r 개의 태그가 응답할 확률은 이항분포를 따르기 때문에, 하나의 태그 수집 라운드 동안 발생하는 빈 슬롯(Empty Slot) 개수, 식별 슬롯 (Identified Slot) 개수, 충돌이 발생한 슬롯 (Collision Slot) 개수에 대한 기댓값을 구할 수 있다.

H. Vogt[8]는 Chebyshev 부등식에 기반을 두어 태그 추정함수와 Markov 프로세스를 이용한 태그 추정 방법과 종료시간 설정방안을 제안하였다. Vogt의 태그 추정함수는 많은 연구 결과에서 가장 태그 추정 정확도가 높은 것으로 평가되고 있다[9]. S.R.Lee에 의해서 제안된 EDFSA(Enhanced DFSA)[10]도 Vogt의 태그 추정 함수를 기반으로 그룹핑 알고리즘을 추가한 것이다. 그러나 Vogt의 태그 추정함수는 가장 최근의 관찰 정보만을 가지고 태그 수를 추정하기 때문에 기댓값과 크게 벗어난 태그 수에 대해서는 추정 정확도가 떨어지고, 프레임 크기 계산을 위한 Markov 프로세스는 인식해야 할 태그의 수가 커질수록 복잡도가 심화되기에 계산에 따른 부하가 커진다.

Park, Y. J. & Kim, Y. B.[11]는 Vogt의 방식과 다르게 가장 최근의 관찰 정보만이 아니라 동일한 크기의 프레임이 사용된 모든 관찰 정보들의 표본평균을 사용하였다. BE-PDFSA[12]는 베이시안 추정과 확률적 응답을 기반으로 DFSA알고리즘을 개선하였고 Chen[13]은 태그 수를 추정하기 위해 관측된 파라미터 값으로 사전 확률을 구하고 추가적인 관측을 통해 해당 대상의 사후 확률을 계산하여 사후 확률이 최대로 되는 태그 수로 추정한다. 하지만 이러한 기댓값을 계산하여 태그 수를 추정하는 방식은 실제 태그수가 기댓값과 크게 벗어날 경우 추정 정확도가 떨어지기에, 실제 태그 수와 고정된 상수인 초기 프레임 크기와 차이가 클 경우 태그 수 추정 정확도가 급격하게 떨어지는 단점이 있다. 최근에는 고정된 상수인 초기 프레임 크기를 가지고 태그 수집 과정을 시작하는 것이 아니라 태그수집 과정 이전에 태그 수를 추정하여 초기 프레임 크기를 결정하는 방식이 제안되었다. NEDFSA[14]는 태그 수집 과정 이전에 매우 간단한 방식으로 태그 수를 추정하고 프레임 크기를 지정해 주기 때문에 태그 수의 범위가 넓어도 높은 효율성을 가질 수 있으나 이동환경에 있는 태그를 수집하는데 적합하지 못하다.

(2) 정적인 태그 수 추정 방법

앞에서 언급한 기댓값을 이용하여 태그 개수를 추정하는 방식이 연산의 복잡도가 높은 반면에 단일 연산을 통해 태그 개수를 추정하면 RFID 시스템의 메모리에 대한 제약이 없고 추정속도 측면에서도 빠르며 알고리즘 구현도 쉽다고 볼 수 있다. 가장 간단한 방법으로 Vogt[8]의 태그 추정 방식에서 사용된 Lower Bound는 충돌이 발생한 슬롯에서 최소 2개의 태그가 점유했다는 가정을 사용한다. 하지만 최소 추정치를 이용하기 때문에 과소 추정할 가능성이 매우 높다. 다음으로 F. C. Schoute에 의해 제안된 방식[7]은 충돌이 발생한 하나의 슬롯에서 평균적으로 2.39개의 태그가 충돌에 관여된다는 통계적 관찰을 기반으로 한다. 또한 전체 태그의 개수를 추정하고 추정치를 사용하여 처리율(Throughput)이 최대가 되도록 프레임 크기를 추정된 태그 수와 동일하게 설정할 것을 제안하였다. 태그 개수를 추정하는데 비교적 높은 정확도를 가지면서 연산이 간단하고 추가적인 오버헤드 없이 구현이 쉽기에 RFID 시스템에서 가장 많이 쓰이고 있다. Zhen[15]의 알고리즘 또한 Schoute의 태그 추정함수를 사용하였고, 실험적인 방법을 통하여 도출시킨 최적프레임 크기의 설정방법을 제안하였다.

2.2.2 능동형 RFID 특성에 적합한 태그 개수 추정 방식

앞에서 언급한 동적인 태그 수 추정방식은 태그 수를 추정하기 위해 기댓값을 계산하고 메모리에 저장하여 비교하면서 여러 번의 시행착오를 거쳐야 하고 그때마다 기댓값을 구해야 하므로 추정 속도가 느려지고 구현하기 어렵다는 공통점을 가지고 있다. 또한 태그 수집 과정 이전에 태그 수를 추정하고 적절한 초기 프레임크기를 설정하는 방식은 추가적인 오버헤드가 필요하며. 태그가 고정적인 위치에 있는 경우에 적합하다. 이러한 단점을 극복할 수 있는 단일 연산을 통해 정적으로 태그 수를 추정하는 방식은 시스템의 메모리에 대한 제약이 없고 추정속도 측면에서도 빠르며 알고리즘 구현도 쉽지만, 초기 프레임 크기와 인식해야하는 태그 수의 차이가 크면 효율성이 급격히

떨어지는 단점을 가지고 있다.

다음은 능동형RFID 시스템에서 적합한 태그 개수 추정 방식은 다음의 요구 조건에 대해 정리하였다.

(1) 태그의 배터리 소모량을 최소한으로 줄이기 위해 리더의 브로드캐스팅 명령 횟수가 적은 알로하 기반 DFSA 알고리즘

(2) 태그의 이동환경에 적합하도록 추가적인 오버헤드 없이 기존의 능동형 RFID의 태그 수집과정을 따라야 한다.

(3) 리더의 인식 거리가 넓은 능동형 RFID의 특성상 인식 거리내의 이동하는 태그를 빠른 시간 안에 인식하기 위해서는 태그 수 추정의 정확도가 높아야 할 뿐만 아니라 태그 수 추정 속도가 빠르도록 태그 수 추정 과정에서 계산의 복잡도가 낮아야 한다.

2.2.3 본 연구의 제안방법(k1과 k2를 이용한 방법)

능동형 RFID 시스템에서 사용 가능한 태그 수 추정 방법은 위에서 언급한 요구 조건을 만족 시키는 추정 속도가 빠르며 알고리즘 구현도 간단한 단일연산을 통한 정적인 방법이 유력해 보인다. 이를 위해 본 연구의 선행 연구[4]에서는 현 프레임의 충돌 슬롯의 수가 전체 슬롯 수에 비해 1/8 이상, 1/4 미만인 경우에는 다음 프레임의 슬롯 수를 동일하게 설정하였고, 1/8 미만인 경우에는 1/2 로 감소시키고 1/4 이상인 경우에는 2 배로 증가시켰다. 최초 전체 슬롯의 수는 32 개로 시작하며 최소 슬롯의 수는 8 개로 제한하였다. 그러나 제안한 파라미터 값들은 이론적 근거 없이 직관적 경험에 바탕을 두고 있기 때문에 최적의 성능을 보장해 줄 수 없다.

실제 태그의 수를 n, 한 프레임 내 슬롯의 수를 N, 그리고 하나의 슬롯에 충돌이 발생할 경우 충돌에 관여된 태그의 수를 k라 하자. Schoute[7]는 통계적 관찰을 기반으로 처리율(Throughput)이 최대가 되도록 N과 n이 동일하다고 가정 하에 k의 평균값을 2.3922로 추정하였다. 따라서 Schoute 의 태그 수 추정 식을 이용 할 경우 다음 프레임의 프레임 크기는 2.3922*k가 된다. 그런데 Schoute의 2.3922라는 값은 태그의 수 n과 프레임의 크기 N 이 동일하다는 가정 하에 유도된 값이기 때문에 프레임의 크기 N에 비해 태그의 수 n이 작을 경우에는 태그 수를 과대 추정하고 n이 클 경우에는 태그 수를 과소 추정하는 단점이 있다.

인식영역 내의 실제 태그 수 n, 프레임 크기 N, 그리고 랜덤변수 X 를 t 번째 태그 수집 라운드를 수행한 후에 한 슬롯에 존재하는 태그 수라고 가정하자. t 번째 태그 수집라운드가 시작하면, n개의 태그들은 N개의 전체 슬롯 중에서 하나를 선택하여 전송을 시도한다. 이러한 태그의 슬롯 선택 과정은 이항분포 $B(n, 1/N)$를 따르기 때문에, 하나의 슬롯이 빈 슬롯, 성공 슬롯, 충돌 슬롯이 될 확률(k_1)은 다음과 같이 나타낼 수 있다.

$$P[X=0]=(1-\frac{1}{N})^n \tag{1}$$

$$P[X=1]=\frac{n}{N}(1-\frac{1}{N})^{n-1} \tag{2}$$

$$\begin{aligned} k_1 &= P[X\geq 2]=1-P[X=0]-P[X=1] \\ &=1-(1-\frac{1}{N})^n-\frac{n}{N}(1-\frac{1}{N})^{n-1} \end{aligned} \tag{3}$$

또한 한 슬롯이 충돌슬롯으로 관찰되고, 그 충돌슬롯을 실제 k개의 태그가 점유하였을 조건부 확률 $P[X=k|X\geq 2]=\frac{P[X=k]}{P[X\geq 2]}, k=2,3...$의 기댓값($k_2$)은 다음 식을 통해 구할 수 있다.

$$\begin{aligned} k_2 &= E[X|X\geq 2]=\frac{\sum_{k=2}^{oo} kP[X=k]}{P[X\geq 2]} \\ &=\frac{\frac{n}{N}-\frac{n}{N}(1-\frac{1}{N})^{n-1}}{1-(1-\frac{1}{N})^n-\frac{n}{N}(1-\frac{1}{N})^{n-1}} \end{aligned} \tag{4}$$

본 연구에서는 인식해야 하는 태그 수와 프레임크기의 비율인 $\lambda(=n/N)$을 통하여 프레임 크기 N보다 많거나 적은 수의 태그를 인식해야 되는 경우를 알아보았다. 다음은 앞서 구한 식(3)과 식(4)를 $n=\lambda N$로 치환하여 나타낸 것이다.

$$\begin{aligned} k_1 &= P[X\geq 2] \\ &=1-(1-\frac{1}{N})^{\lambda N}-\lambda(1-\frac{1}{N})^{\lambda N-1} \end{aligned} \tag{5}$$

$$\begin{aligned} k_2 &= E[X|X\geq 2] \\ &=\frac{\lambda-\lambda(1-\frac{1}{N})^{\lambda N-1}}{1-(1-\frac{1}{N})^{\lambda N}-\lambda(1-\frac{1}{N})^{\lambda N-1}} \end{aligned} \tag{6}$$

식(5)와 식(6)에서 프레임크기 N이 무한이 크다고

가정하면 자연로그 정의 $\lim_{N\to\infty}(1+\frac{1}{N})^N = e$ 에 따라서 다음과 같은 근사식을 얻을 수 있다.

$$k_1 = P[X \geq 2] \approx 1 - e^{-\lambda} - \lambda e^{-\lambda} \quad (7)$$

$$k_2 = E[X|X \geq 2] \approx \frac{\lambda - \lambda e^{-\lambda}}{1 - e^{-\lambda} - \lambda e^{-\lambda}} \quad (8)$$

다음 표 4 는 λ(= n/N) 에 따라서 식(7)과 식(8)을 통해 k_1과 k_2를 구하였다. 표에서 볼 수 있듯이 N 이 무한히 크다고 가정하고 λ가 1일 때 Schoute의 값인 k_2 = 2.3922을 구할 수 있지만, λ가 1보다 작으면 k_2는 2.3922 보다 작은 값을 갖고 1 보다 크면 2.3922 보다 큰 값을 갖는다.

따라서 본 연구에서는 현 프레임의 어떠한 충돌 슬롯 수신 상황에서도 동일하게 2.3922를 사용하는 대신에 λ값에 따라 k_2값을 적절하게 변화 시키는 방법을 제안하고자 한다. 그런데 실제 λ값을 구하는 것은 태그 수인 n을 예측해야 하기 때문에 현실적으로 불가능 하다. 따라서 본 연구에서는 k_1 값에 따라 k_2값을 적절하게 조정해 사용하는 방법을 제안하였다. 특정 슬롯의 충돌 확률인 k_1은 "(평균 충돌 슬롯 수/전체 슬롯 수)" 로 볼 수 있는데 이는 시스템 운영과정에서 이전 프레임의 슬롯 수와 충돌 슬롯 수로부터 예측할 수 있다. 본 연구에서는 k_1값을 3 구간으로 나누어 다음과 같이 k_2값을 조정해 나갔다.

표 4. λ에 따른 k_1과 k_2
Table 4. k_1 and k_2 according to λ.

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
k_1	0.0047	0.0175	0.0369	0.0616	0.0902	0.1219	0.1558	0.1912	0.2275	0.2642
k_2	2.0339	2.0689	2.1051	2.1425	2.1810	2.2208	2.2617	2.3040	2.3475	2.3922
λ	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
k_1	0.3010	0.3374	0.3732	0.4082	0.4422	0.4751	0.5068	0.5372	0.5663	0.5940
k_2	2.4382	2.4856	2.5342	2.5841	2.6354	2.6880	2.7418	2.7970	2.8535	2.9114
λ	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3
k_1	0.6204	0.6454	0.6691	0.6916	0.7127	0.7326	0.7513	0.7689	0.7854	0.8009
k_2	2.9705	3.0309	3.0926	3.1556	3.2198	3.2853	3.3521	3.4200	3.4892	3.5595
λ	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4
k_1	0.8153	0.8288	0.8414	0.8532	0.8641	0.8743	0.8838	0.8926	0.9008	0.9084
k_2	3.6310	3.7036	3.7774	3.8522	3.9281	4.0050	4.0830	4.1619	4.2418	4.3226
λ	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	5
k_1	0.9155	0.9220	0.9281	0.9337	0.9389	0.9437	0.9482	0.9523	0.9561	0.9596
k_2	4.4043	4.4869	4.5703	4.6546	4.7396	4.8254	4.9119	4.9991	5.0870	5.1755
λ	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
k_1	0.9734	0.9826	0.9887	0.9927	0.9953	0.9970	0.9981	0.9988	0.9992	0.9995
k_2	5.6270	6.0908	6.5642	7.0450	7.5313	8.0215	8.5147	9.0100	9.5068	10.0045

- 첫 번째 구간은 k_1가 0.2642보다 작은 구간으로 (λ 가 0과 1사이의 값을 갖는 구간) 이 때 k_2값은 이 구간의 평균인 2.23를 사용한다.
- 두 번째 구간은 k_1가 0.2642 보다 크고 0.5940보다 작은 구간으로 (λ 가 1과 3사이의 값을 갖는 구간) 이때 k_2값은 이 구간의 평균인 2.53을 사용한다.
- 세 번째 구간은 k_1가 0.5940 보다 크고 1 보다 작은 구간으로 (λ 가 3과 10사이의 값을 갖는 구간) 이때 k_2은 이 구간의 평균인 6.8을 사용한다.

본 연구에서 제안한 방법에서 구간을 결정해주는 것은 k_1값인데 이는 "충돌 슬롯 수/전체 슬롯 수"의 평균값이다. 그런데 이 방법을 실제로 사용 할 경우에는 평균값 대신 현 프레임의 관측치 인 "충돌 슬롯 수/전체 슬롯 수" 만을 사용하기 때문에 약간의 오차가 있을 수 있다.

Ⅲ. 시뮬레이션 모델

본 연구에서 제안한 방법의 성능을 검증하기 위해서 Java언어를 사용하여 능동형 RFID 시뮬레이션 모델을 개발하였다. RFID 시뮬레이션 환경은 다음과 같다. Wake up 시간은 고려하지 않고 RFID 리더의 인식 범위 안에서 RFID 리더와 인식해야 하는 태그간의 통신은 단일채널을 이용하며 에러 없이 전송된다고 가정하였다.

본 연구에서 제안한 방법은 크게 Query 명령을 사용한 Collection 명령과 Ack 간소화와 슬롯의 충돌 확률(k1)과 충돌이 발생한 슬롯의 평균 태그 수(k2)를 이용한 다음 프레임의 슬롯 수 결정 두 가지다. 제안한 방법과 기존 방법의 성능을 비교하기 위하여 표5와 같이 다음 6가지 시뮬레이션 모델들(A1, A2, A3, B1, B2, B3)을 구축하였다. 실험에서는 태그 개수를 1개에서 100개까지 1개 단위로 변화시키면서 각 실험

표 5. 시뮬레이션 모델의 조합
Table 5. Combination of simulation model.

Method to estimate number of tags	Collection command and ack process	
	ISO 18000-7	Proposed
Method used in experimental test[4]	A1	B1
Schoute method[7]	A2	B2
method using k_1 and k_2	A3	B3

당 10,000번의 시뮬레이션을 수행하여 평균값을 계산하였다.

본 연구에서 제안한 k_1과 k_2를 이용한 방법을 본 연구의 선행 연구에서 제안한 실측에 사용한 방법과 단일 연산 방법의 가장 대표적인 Schoute 방법과 비교해 보았다.

또한 본 연구에서 제안한 k_1과 k_2를 이용하여 다음 프레임의 슬롯 수를 결정하는 방법의 성능은 k_1 값을 이용하여 구간을 어떻게 나누느냐에 따라 달라 질 수도 있다. 본 연구에서는 이를 조사하기 위하여 서로 다른 구간을 사용할 경우 성능을 비교 분석해 보았다.

3.1 시뮬레이션을 위한 입력 자료

본 시뮬레이션을 위하여 사용한 입력 자료는 다음과 같다.

3.1.1 ISO/IEC 18000-7

표 6은 시뮬레이션 실험에서 사용된 시간 파라미터의 값들을 보여준다. 이 값들은 ISO/IEC 18000-7 표준을 따라서 설정하였다. 하지만 제안하는 방식의 효율적인 태그 수집 절차를 통한 개선 정도를 좀 더 정확하게 알아보고자 표준의 433MHz 대역의 전송속도가 아닌 제안하는 방식의 2.4Ghz 대역의 전송속도로 파라미터 값을 설정하였다.

Time Slot의 크기는 태그가 추가적인 데이터 없이 태그-ID 만을 전송한다고 가정하였다.

표 6. 시간 파라미터
Table 6. Time parameter.

Time	Parameter Value	=preamble+(command size*period for one byte)+end period+Slot Guard Time
Collection Command	1776μs	1776μs=1341μs+(12byte*32μs)+51μs
Sleep Command	1840μs	1840μs=1341μs+(14byte*32μs)+51μs
Time Slot	4ms	3963μs=1323μs+(20byte*32μs)+51μs+2ms

3.1.2 2.4GHz대역에서 Query 명령을 사용한 Collection 명령의 간소화

제안하는 프로토콜에서 기본적인 파라미터 값은 표 7과 같지만 가변적인 파라미터 값을 갖는 Query/Repeat와 Query/Adjust는 상황에 따른 추가적인 데이터의 전송이 필요하다. 태그 수집과정에서 t번째 인식 라운드의 프레임 크기를 Qt, (t=1,2,..) 라 할 때 t+1번째 인식 라운드의 Query/Repeat나 Query/Adjust의 전송 데이터에 Qt*1bit가 더해져야한다. Query/Adjust의 경우에는 새로운 슬롯 수를 나타내는 1 byte의 NewQ 가 추가로 더해진다.

표 7. Query 명령과 타임 슬롯 파라미터
Table 7. Query command and time slot parameter.

Time	Parameter Value	=preamble+(command size*period for one byte)+end period+Gap Time
Query	1712μs	1712μs=1341μs+(10byte*32μs)+51μs
Query Repeat	1616μs	1616μs=1341μs+(7byte(+v)*32μs)+51μs
Query Adjust	1649μs	1649μs=1341μs+(7byte(+v)*32μs)+51μs
Time Slot	3ms	2982μs=1323μs+(19byte*32μs)+51μs+1ms

3.2 능동형 고속 RFID 성능측정 지표

3.2.1 처리율(Throughput)

알로하 기반의 충돌 방지 알고리즘이 사용하고 있는 RFID 시스템의 처리율은 태그 수집 과정에서 각 라운드의 프레임크기의 합인 총 슬롯 수와 그 중에서 인식이 성공한 슬롯의 비로 정의할 수 있다. 따라서 태그 수집 라운드 후에 관찰된 빈 슬롯의 수를 I, 성공 슬롯을 S, 충돌슬롯을 C라고 하면 RFID 시스템의 처리율 SE 는 다음 식(9)와 같이 계산할 수 있다.

$$SE = \frac{S}{I+S+C} \tag{9}$$

3.2.2 태그 인식 시간

앞서 설명한 처리율은 RFID시스템의 효율을 평가하는 중요한 척도 이지만 브로드캐스팅으로 이루어지는 태그 수집 명령시간을 고려하지 않는다. 일반적으로 트리기반 충돌방지 프로토콜이 알로하 기반 충돌방지 프로토콜보다 효율성이 높지만, 잦은 브로드캐스팅으로 인하여 전체 태그 인식 시간은 오히려 증가하는 결과를 보여준다. 따라서 RFID 시스템별 특성을 반영할 수 있는 태그 인식 시간을 측정하는 것이 필요하다. 태그 인식 시간은 리더의 최초 Collection 명령으로부터 리더의 인식범위 안에 있는 모든 태그를 인식하는데 걸리는 시간의 평균값으로 정의한다.

3.2.3 인식률

반복적인 실험을 통해 얻은 태그 인식 시간의 평균값은 편차를 반영하지 못한다. 특히 리더의 위치는 고정되어 있고 이동하는 태그를 인식해야 되는 RFID시스템의 경우에는 태그가 리더의 인식범위를 벗어나기 전에 인식에 성공해야 한다. 따라서 태그 인식 시간의 평균값뿐만 아니라 주어진 시간 내에 모든 태그의 인

식 여부를 반영할 수 있는 성능 측정 지표가 필요하다. 본 연구에서는 리더가 인식해야 하는 태그 수가 주어졌을 때 10,000회 반복실험을 통하여 그 중 1초 안에 모든 태그가 인식된 실험의 수를 인식률로 정의한다.

Ⅳ. 결과 및 분석

시뮬레이션 수행 결과를 다음의 네 가지 측면에서 검토 하였다.

- Collection명령과 Ack간소화에 따른 성능 개선
- 다음 프레임의 슬롯 수를 결정하는 방법에 따른 성능 변화
- 위 두 가지 방법의 조합(6 가지 방법)에 따른 성능 분석
- k_1값을 이용한 구간결정 방법에 따른 성능변화

4.1 Collection 명령과 Ack 간소화 에 따른 성능 개선

이를 조사하기 위해 시뮬레이션 A1 과 B1, A2 와 B2 그리고 A3 와 B3의 결과를 비교 분석하였다. 시뮬레이션 결과는 다음과 같다.

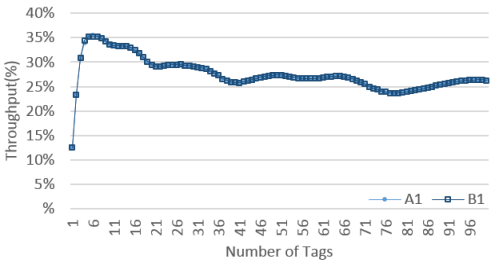

그림 3. A1 과 B1의 Throughput 비교
Fig. 3. Throughput comparison of A1 and B1

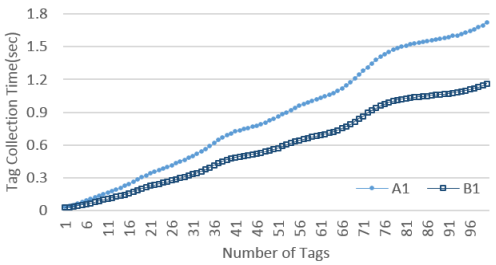

그림 4. A1 과 B1의 평균 태그 인식 시간 비교
Fig. 4. Comparison of average tag recognition time of A1 and B1

4.1.1 A1 과 B1

기존 실측에 사용된 충돌방지 알고리즘 하에서 ISO 18000-7과 제안된 방법인 Collection 명령과 Ack 간소화의 성능을 Throughput, 태그인식시간, 그리고 인식률 세 가지 측면에서 비교하였다.

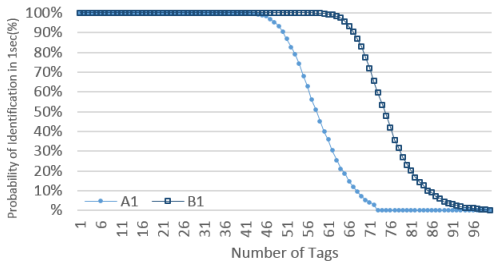

그림 5. A1 과 B1의 태그 수에 따른 1초안에 인식률
Fig. 5. recognition rate within 1 sec according to the number of tags of A1 and B1

4.1.2 A2 와 B2

Schoute의 DFSA 충돌방지 알고리즘 하에서 ISO 18000-7과 제안된 방법인 Collection 명령과 Ack 간소화의 성능을 Throughput, 태그인식시간, 그리고 인식률 세 가지 측면에서 비교하였다.

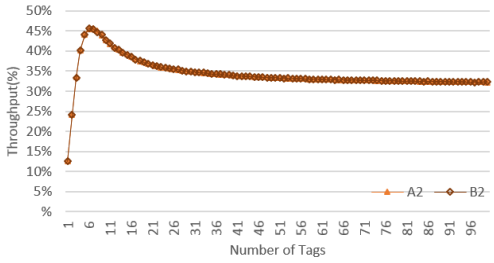

그림 6. A2 와 B2의 Throughput 비교
Fig. 6. Throughput comparison of A2 and B2

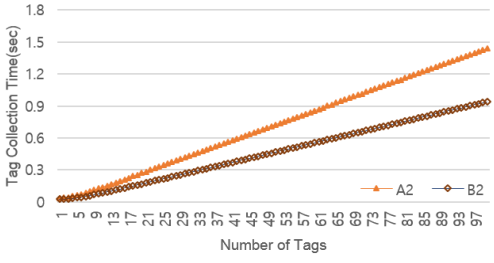

그림 7. A2 와 B2의 평균 태그 인식 시간 비교
Fig. 7. Comparison of average tag recognition time of A2 and B2

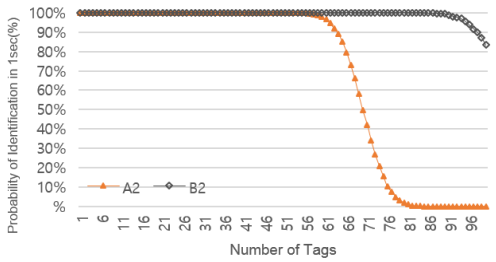

그림 8. A2 와 B2의 태그 수에 따른 1초안에 인식률
Fig. 8. Recognition rate within 1 sec according to the number of tags of A2 and B2

4.1.3 A3 와 B3

제안하는 k_1과 k_2를 이용한 DFSA 충돌방지 알고리즘 하에서 ISO 18000-7과 제안된 방법인 Collection 명령과 Ack 간소화의 성능을 Throughput, 태그인식 시간, 그리고 인식률 세 가지 측면에서 비교하였다.

4.1.4 종 합

어떠한 태그 수 추정 방법 하에서도 기존 ISO 18000-7과 Collection과 Ack 명령 간소화의 Throughput은 거의 차이가 없는 것으로 나타났다. Collection과 Ack 명령 간소화 방법은 Collection과

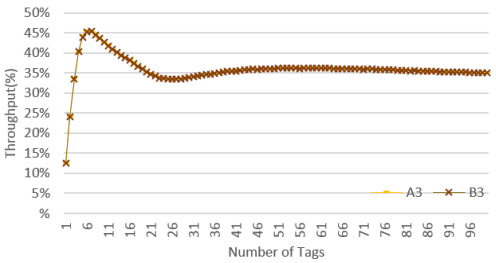

그림 9. A3 와 B3의 Throughput 비교
Fig. 9. Throughput comparison of A3 and B3

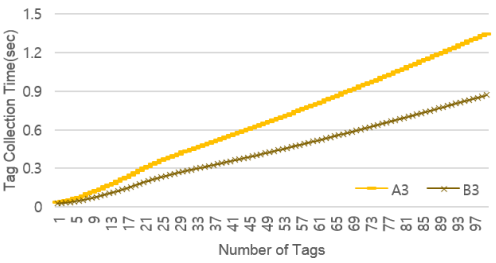

그림 10. A3 와 B3의 평균 태그 인식 시간 비교
Fig. 10. Comparison of average tag recognition time of A3 and B3

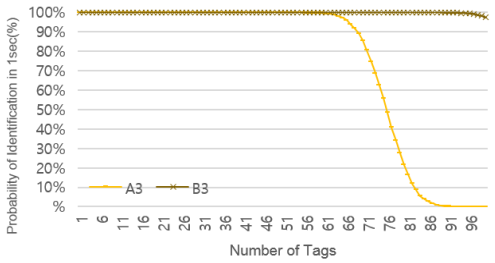

그림 11. A3 와 B3의 태그 수에 따른 1초안에 인식 확률
Fig. 11. Recognition rate within 1 sec according to the number of tags of A3 and B3

Ack 절차를 간소화함으로써 태그 인식시간을 줄이는 데 목적이 있다. 따라서 기존 ISO 18000-7 방법과 비교하여 Throughput에 차이가 없는 것은 당연해 보인다. 반면에 평균 태그 인식 시간과 인식률은 모든 태그 수 추정 방법 하에서 Collection과 Ack 명령 간소화 방법이 기존 ISO 18000-7 방법에 비해 훨씬 우수한 것으로 나타났다. 실제로 본 연구의 k_1과 k_2를 이용한 태그 수 추정 방법을 사용 할 경우 평균 태그 인식 시간은 태그 수가 100개 일 때도 1초 아래에 머물고 있으며 태그 인식률도 태그 수가 약 90개 까지는 1의 값을 갖는다. 반면에 ISO 18000-7의 경우는 태그 수가 75개를 초과하면 평균 태그 인식 시간이 1초를 넘게 되며 인식률도 태그 수가 55개를 넘어서면 1 아래로 떨어짐을 볼 수 있다.

위의 표 8은 1~100개의 태그 수에 따른 평균 인식 시간과 1초안에 인식 가능한 인식률이 0.999 이상을 만족하는 최대 태그 수에 대한 Collection 명령과 Ack의 간소화 방법의 개선율을 보여준다.

표 8. Collection 명령과 Ack의 간소화 방법의 평균 개선율 (%)
Table 8. Average improvement rate of the simplified collection and ack command(%).

Improvement rate	A1 vs B1	A2 vs B2	A3 vs B3
Identification Time[Ave.]	-67.26%	-64.66%	-64.15%
Identification tag[Max.]	+41.46%	+59.26%	+60.34%

4.2 k_1과 k_2를 이용하여 다음 프레임의 슬롯 수를 결정하는 방법의 성능 개선

이를 조사하기 위해 시뮬레이션 A1 과 A2 그리고 A3의 결과를 비교분석하였다. 또한 시뮬레이션 B1와 B2 그리고 B3 결과도 비교분석하였다. 시뮬레이션

결과는 다음 그림과 같다.

4.2.1 A1 과 A2 그리고 A3

2.4 GHz 대역에서 ISO 18000-7 방법과 절차를 사용해 기존 실측에 사용된 DFSA 충돌방지 알고리즘, Schoute의 DFSA 충돌방지 알고리즘, 본 연구에서 제안하는 k_1과 k_2를 이용한 DFSA 충돌방지 알고리즘의 성능을 비교하였다.

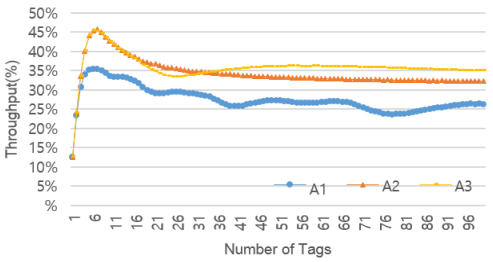

그림 12. A1 과 A2 그리고 A3의 Throughput 비교
Fig. 12. Throughput comparison of A1, A2 and A3

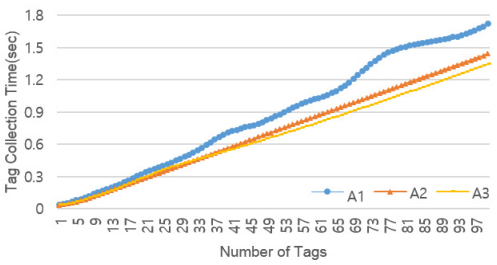

그림 13. A1과 A2 그리고 A3의 평균 태그 인식 시간 비교
Fig. 13. Comparison of average tag recognition time of A1, A2 and A3

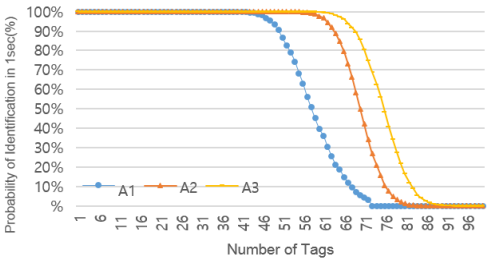

그림 14. A1 과 A2 그리고 A3의 태그 수에 따른 1초안 인식확률
Fig. 14. Recognition rate within 1 sec according to the number of tags of A1, A2 and A3

4.2.2 B1 와 B2 그리고 B3

2.4GHz대역에서 Query 명령을 사용한 Collection 명령과 Ack의 간소화 방법을 사용해 기존 실측에 사용된 DFSA 충돌방지 알고리즘, Schoute의 DFSA 충돌방지 알고리즘, 본 연구에서 제안하는 k_1과 k_2를 이용한 DFSA 충돌방지 알고리즘의 성능을 비교하였다.

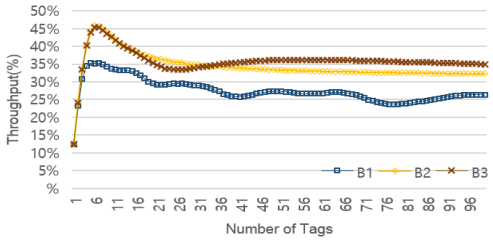

그림 15. B1과 B2 그리고 B3의 Throughput 비교
Fig. 15. Throughput comparison of B1, B2 and B3

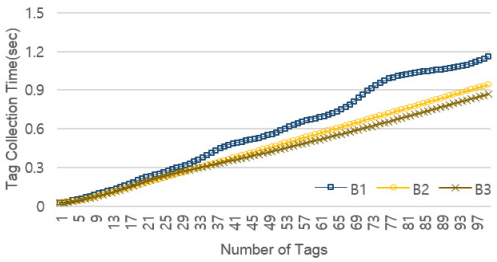

그림 16. B1과 B2 그리고 B3의 평균 태그 인식 시간 비교
Fig. 16. Comparison of average tag recognition time of B1, B2 and B3

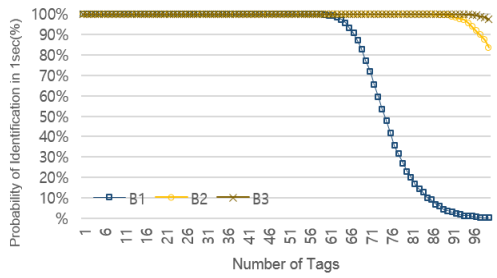

그림 17. B1과 B2 그리고 B3의 태그 수에 따른 1초안에 인식률
Fig. 17. Recognition rate within 1 sec according to the number of tags of B1, B2 and B3

4.2.3 종합

본 연구에서 제안한 k_1과 k_2를 이용한 방법이 실측에 사용했던 방법, 그리고 Schoute의 DFSA 충돌방지

알고리즘에 비해 Throughput, 평균 태그 인식 시간, 그리고 1초 안에 인식률 모든 면에서 우수한 것으로 나타났다. 실제로 Query 명령을 사용한 Collection 명령과 Ack의 간소화 방법 하에서 태그 수가 100개일 때 Throughput은 약 35.89%로 Schoute의 34.36%, 기존 실측에 사용한 방법의 27.5%에 비해 가장 높았다. 평균 태그 인식 시간도 태그 수가 100개일 때 0.86초로 Schoute의 0.94초, 그리고 기존 실측에서 사용했던 방법의 1.17초에 비해 가장 낮았다. 1초안에 인식률 0.999를 만족하는 최대 태그 수도 93개로 다른 두 방법의 57개, 86개보다 가장 많았다.

다음 표 9는 태그 수 1~100 개에서 k_1과 k_2를 이용하여 다음 프레임의 슬롯 수를 결정하는 방법의 성능 향상 개선율을 보기위해 ISO 18000-7과 Collection 명령과 Ack 간소화 방식 각각 하에서 다른 태그 수 추정 알고리즘과 비교한 결과의 평균 개선율을 나타낸다.

표 9. k_1과 k_2를 이용한 방법의 평균 개선율(%)
Table 9. Average improvement rate of the method using k1 and k_2(%).

Improvement rate(%)	A3 vs A1	A3 vs A2	B3 vs B1	B3 vs B2
Throughput	+30.41%	+4.46%	+30.44%	+4.47%
Ave. Identification Time	-29.72%	-6.23%	-36.01%	-7.08%
Max. Identification tag	+41.46%	+7.41%	+60.34%	+8.14%

4.3 위 두 가지 방법의 조합(6 가지 방법)에 따른 성능 분석

이를 조사하기 위해 시뮬레이션 A1,A2,A3,B1,B2 그리고 B3의 결과를 비교 분석하였다. 시뮬레이션 결과를 다음 그림에 나타냈다.

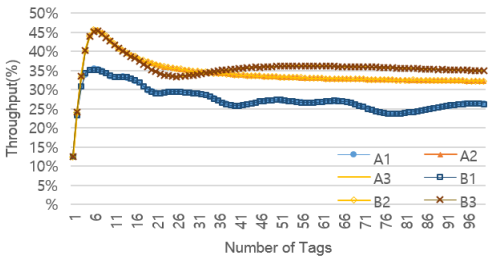

그림 18. A1,A2,A3,B1,B2,B3의 Throughput 비교
Fig. 18. Throughput comparison of A1,A2,A3,B1,B2,B3

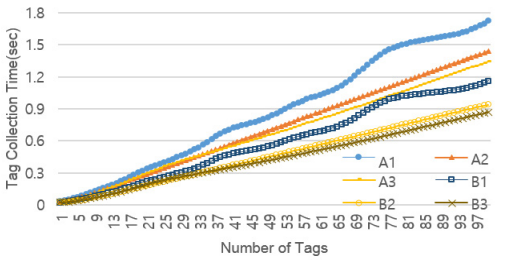

그림 19. A1,A2,A3,B1,B2,B3의 평균 태그 인식 시간 비교
Fig. 19. Comparison of average tag recognition time of A1,A2,A3,B1,B2,B3

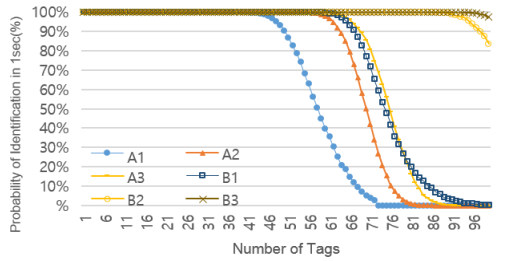

그림 20. A1,A2,A3,B1,B2,B3의 태그 수에 따른 1초안에 인식률
Fig. 20. Recognition rate within 1 sec according to the number of tags of A1,A2,A3,B1,B2,B3

위의 그림들에서 볼 수 있듯이 B3조합, 즉 Query 명령을 사용하여 Collection 명령과 Ack의 간소화하고 k_1과 k_2를 이용하여 태그수를 예측하는 방법이 모든 성능 면에서 가장 우수한 것으로 나타났다. Query 명령을 이용한 Collection 명령과 Ack의 간소화는 태그 인식 속도를 줄이는데 기여했고 k_1과 k_2를 이용하여 태그수를 예측하는 방법은 Throughput을 향상시킬 뿐만 아니라 이로 인해 Collection 라운드 횟수를 줄여 태그 인식 속도를 줄일 수 있었다.

다음 표 10에 1~100 개의 태그 수에 따라서

표 10. 1~100개의 태그 수에 따른 성능 측정 지표 결과 값들의 평균값
Table 10. Average value of the performance measure according to the number of the tags 1 to 100.

	A1	A2	A3	B1	B2	B3
Throughput(%)	26.3%	33.8%	35.4%	27.7%	33.9%	35.89%
Ave. Identification time(sec)	0.88	0.72	0.68	0.59	0.46	0.43
Max. Identification tag	41	54	58	58	86	93

Throughput, 평균 인식 시간, 1초안에 인식률까지 3가지의 성능 측정 지표에 대한 결과 값들의 평균을 나타냈다.

4.4 k_1값을 이용한 구간 결정 방법에 따른 성능 변화

2장 2절 에서는 k_1값에 따라 세 구간으로 나누어 k_2 값을 예측하는 방법을 제시하였다. 구간 결정 방법에 따른 성능 변화를 조사하기 위해 다음과 같이 4가지 경우를 고려하였고 시뮬레이션 B3를 이용하였다. 각 구간을 나누는 구간 값은 표 4에서 나뉘는 구간 크기가 동일하도록 설정했다.

- B3-1: 전체를 하나의 구간으로 본 경우로 본 연구의 Schoute 방법에 해당 한다
- B3-2: 전체를 2개의 구간으로 나눈다.
- B3-3: 전체를 3개의 구간으로 나누었는데 본 연구에서 제안한 방법이다.
- B3-4: 전체를 4개의 구간으로 나눈다.

각 경우의 시뮬레이션 결과들을 다음 그림들에 나타냈다.

위 그림들에서 볼 수 있듯이 k_1의 값을 이용하여 구간을 몇 개로 나누느냐에 따른 성능 면에서 차이가 있다. 1개나 2개 보다는 3개나 4개가 모든 성능 면에서 우수하게 나타났고 3개와 4개의 차이는 매우 미미 하게 나타났다. 물론 k_1을 이용한 최적의 구간 설정 방법과 구간 값 결정 방법이 존재 할 텐데 이에 대한 구체적인 연구는 추후로 미룬다.

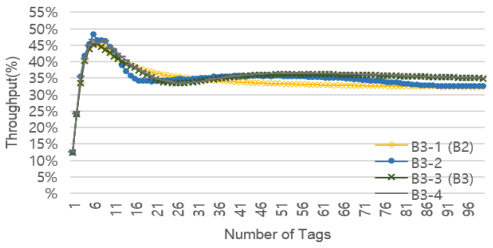

그림 21. B3-1, B3-2, B3-3, B3-4의 Throughput 비교
Fig. 21. Throughput comparison of B3-1, B3-2, B3-3, B3-4

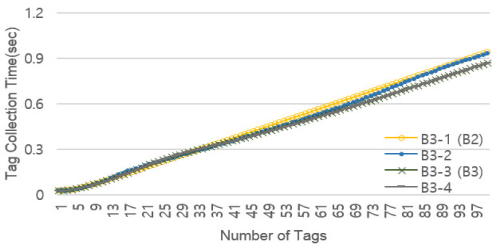

그림 22. B3-1, B3-2, B3-3, B3-4 평균 태그 인식시간 비교
Fig. 22. Comparison of average tag recognition time of B3-1, B3-2, B3-3, B3-4

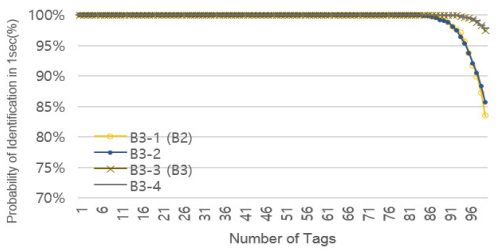

그림 23. B3-1, B3-2, B3-3, B3-4의 태그 수에 따른 1초안에 인식률
Fig. 23. Recognition rate within 1 sec according to the number of tags of B3-1, B3-2, B3-3, B3-4

4.5 최적해와 비교

마지막으로 Collection 명령과 Ack 절차를 간소화하고 k_1과 k_2를 이용하여 태그수를 예측하는 방법의 성능을 가장 이상적인 최적인 경우와 비교해 보았다. 실제 운용 환경 하에서는 태그의 수를 알 수 없다. 그러나 시뮬레이션을 수행 중 우리는 실제 태그의 수를 알 수 있기 때문에 이를 실제 태그 수로 사용하고 성능값을 구해 이를 비교해 보았다. 이 성능 값들은 실제적으로 도달 하기는 불가능하지만 예측 알고리즘들이 도달 할 수 있는 상한 값이기 때문에 이들과 비교를 통해 본 연구에서 제안한 방법의 우수성을 객관적으로 평가 할 수 있다. 이상적인 경우에도 Query를 통해 Collection과 Ack 명령의 단순화를 가정 하였다. 다음 그림 24, 그림 25, 그리고 그림 26에 각각 Throughput, 평균 인식 시간, 평균 인식률을 이상적인 최적(Optimal)값과 비교하였는데 세 가지 성능 측정

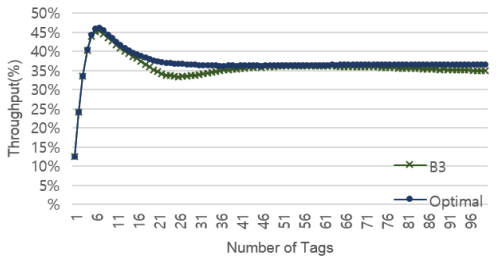

그림 24. B3 와 (Optimal)의 Throughput 비교
Fig. 24. Throughput comparison of B3 and (Optimal)

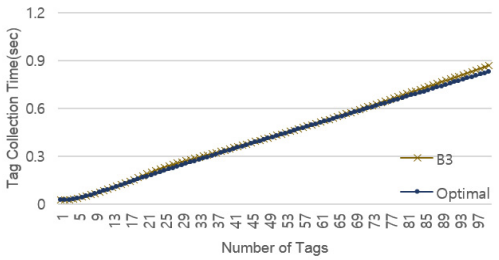

그림 25. B3 와 (Optimal) 평균 태그 인식 시간 비교
Fig. 25. Comparison of average tag recognition time of B3 and (Optimal)

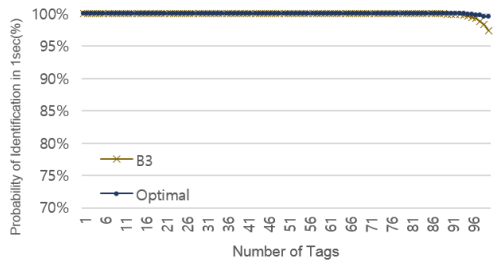

그림 26. B3와 (Optimal)의 태그 수에 따른 1초안에 인식률
Fig. 26. Recognition rate within 1 sec according to the number of tags of B3 and (Optimal)

치 모두에서 매우 근사한 결과를 보여 주어 본 연구에서 제안한 방법의 우수성을 확인 할 수 있었다.

다음 표 11에 이상적인 최적방법과의 평균 차이를 나타냈다. Throughput은 0.9%, 평균인식시간은 0.1초, 최대 인식가능 태그 수는 3개로 그 차이가 매우 미미하게 나타났다.

표 11. 1~100개의 태그 수에 따른 성능 측정 지표 결과 값들의 평균값
Table 11. Average value of the performance measure according to the number of the tags 1 to 100

	B3	Optimal	Difference
Throughput(%)	35.89	36.79	0.9%
Ave. Identification time(sec)	0.43	0.42	0.01
Max. Identification tag	93	96	3

V. 결 론

Query 명령을 사용하여 Collection 명령과 Ack를 간소화하고 슬롯의 충돌 확률(k1)과 충돌이 발생한 슬롯의 평균 태그 수(k2)를 이용하여 태그수를 예측하는 방법의 조합이 모든 성능 면에서 가장 우수한 것으로 나타났다. 이는 Query 명령을 이용한 Collection 명령과 Ack의 간소화는 태그 인식 속도를 줄이는데 기여했고 k_1과 k_2를 이용하여 태그수를 예측하는 방법은 Throughput을 향상시킬 뿐만 아니라 이로 인해 Collection 라운드 횟수를 줄여 태그 인식 속도를 줄일 수 있었다.

Collection 명령과 Ack를 간소화하고 k_1과 k_2를 이용하여 태그수를 예측하는 방법의 성능을 태그 수를 안다고 가정한 이상적인 경우의 성능과 비교해 보았다. 이 성능 값 들은 실제적으로 도달 하기는 불가능하지만 예측 알고리즘들이 도달 할 수 있는 상한 값이다. 본 연구에서 제안한 방법은 Throughput, 평균 인식 시간, 1초 안의 평균 인식률 모두에서 이상적인 경우의 성능값과 매우 유사하게 나타나 제안한 방법의 유효성을 확인 할 수 있었다.

본 연구에서 제안한 k_1과 k_2를 이용하여 태그수를 예측하는 방법의 성능은 구간을 몇 개로 나누느냐와 구간 값 들을 어떻게 설정하느냐에 따라 차이가 있을 수 있다. 본문에서 간단하게 구간에 따른 성능 추이를 살펴보았지만 최적의 구간수와 그에 따른 최적의 구간 값 결정은 추후과제로 남긴다.

References

[1] ISO/IEC 18000-7, *Information Technology-Radio Frequency Identification for Item Management - Part 7: Parameters for Active Air Interface Communications at 433 MHz*, pp. 3-54, 2009.

[2] J. H. Joo and S. H. Chung, "Implementation of an efficient slotted CSMA/CA anti-collision protocol for active RFID system," *J. KICS*, vol. 37A, no. 12, pp. 1013-1022, Dec. 2012.

[3] S. R. Lee, Y. W. Lee, and Y. I. Joo, "An RFID tag anti-collision protocol for port logistics system," *J. KICS*, vol. 38C, no. 2, pp. 202-207, Feb. 2013.

[4] J.-T. Kim, B.-G. Kang, and K.-W. Lee, "An Implementation of modified frame slotted ALOHA algorithm for fast tag collection in an active RFID system," *J. KICS,* vol. 39B, No. 09, pp. 598-605, Sept. 2014.

[5] I.-S. Kim and C.-S. Kim, "Anti-collision

algorithm for high-speed tags in active RFID system," *J. KIECS*, vol. 8, no. 12, pp. 1891-1904, Dec. 2013.

[6] H.-Y. Lee, "High-tag anti-collision algorithm to improve the efficiency of tag identification in active RFID system," *J. KIECS*, vol. 7, no. 2, pp. 235-242, Apr. 2012.

[7] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 565-568, Apr. 1983.

[8] H. Vogt, "Multiple object identification with passive RFID tags," in *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, vol. 3, pp. 6, 2002.

[9] D. K. Klair, K. W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Commun. Surveys & Tutorials*, vol. 12, no. 3, pp. 400-421, 2010.

[10] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," *The 2nd Int. Annu. Conf. Mob. and Ubiquitous Systems: Networking and Services*, pp. 166-172, 2005.

[11] Y. J. Park and Y. B. Kim, "On the accuracy of RFID tag estimation functions," *J. ICCE*, vol. 10, no. 1, pp. 33-39, 2012.

[12] C. Wang, M. LI, J. Qiao, W. Wang, and X. Li, "An advanced dynamic framed-slotted aloha algorithm based on bayesian estimation and probability response," *Int. J. Antennas and Propagation,* vol. 2013, no. 743468, p. 8, 2013.

[13] W. T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," *IEEE Trans. Automation Sci. Eng.*, vol. 6, no. 1, pp. 9-15, 2009.

[14] R. P. B. Mota and D. M. Batista, "A dynamic frame slotted ALOHA anti-collision algorithm for the internet of things," in *Proc. the 29th Annu. ACM Symp. Applied Computing*, pp. 686-691, Mar. 2014.

[15] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," *IEICE Trans. Commun.*, vol. E88-B, no. 3, pp. 991-999, Mar. 2005.

김 지 태 (Ji-tae Kim)

1983년 2월 : 경상대학교 해양과학대 전자통신학과(공학사)
2007년 2월 : 서울과학기술대학교 전자공학과(공학석사)
2007년 3월~현재 : 서울과학기술대학교 IT정책전문대학원 박사과정
2000년 8월~2007년 5월 (주)크레디패스 연구소장
2007년 6월~2011년12월 (주)엔디에스 연구소장
2012년 1월~현재 : (주)한맥이엔지 상무이사
<관심분야> 위치추적시스템, RFID, 무선센서네트워

김 진 성 (Jin-sung Kim)

2015년 2월 : 서울과학기술대학교 글로벌융합산업공학과(공학사)
2015년 3월~현재 : 연세대학교 정보산업공학과(석사과정)
<관심분야> RFID, IoT, Smart Factory, Optimization of Production Process, Sustainable SCM

이 강 원 (Kang-won Lee)

1980년 2월 : 서울대학교 산업공학과(공학사)
1982년 2월 : 서울대학교 산업공학과(공학석사)
1985년 2월 : Kansas State Univ. 산업공학과(공학박사)
1982년 12월~1985년 12월 : 미국 캔사스 주립대학교 연구조교
1985년 12월~1989년 12월 : 한국전자통신연구원 선임연구원
1989년 1월~현재 : 서울과학기술대학교 글로벌융합산업공학과 교수
<관심분야> 정보통신, 품질 및 신뢰성, O.R., 차세대 이동통신, RFID