



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

碩士學位論文

UHF 대역 RFID 시스템에서의
개선된 충돌방지 기법



濟州大學校 大學院

通信工學科

金志允

2006年 12月

UHF 대역 RFID 시스템에서의
개선된 충돌방지 기법

指導教授 梁斗榮

金志允

이 論文을 工學 碩士學位 論文으로 提出함

2006年 12月

金志允의 工學 碩士學位 論文을 認准함

審査委員長 左政祐 印

委 員 金興洙 印

委 員 梁斗榮 印

濟州大學校 大學院

2006 年 12 月

Improved Anti-collision Algorithm for UHF band RFID System

Ji-yoon Kim

(Supervised by professor Doo-yeong Yang)

A thesis submitted in partial fulfillment of the requirement for the degree
of Master of Engineering

2006. 12.

This thesis has been examined and approved.

Thesis director, Jeong-woo Jwa, Prof. of Telecom. Eng.

Thesis director, Heung-soo Kim, Prof. of Telecom. Eng.

Thesis director, Doo-yeong Yang, Prof. of Telecom. Eng.

(Name and signature)

2006. 12. 12

Date

Department of Telecommunication Engineering
GRADUATE SCHOOL
CHEJU NATIONAL UNIVERSITY

목 차

Abstract	1
I. 서론	3
II. UHF대역 RFID 프로토콜	5
1. ISO 18000-6 Type A/B 에어인터페이스	5
2. EPCglobal 프로토콜의 에어인터페이스	8
III. 개선된 UHF대역 RFID 충돌방지 알고리즘	15
1. 개선된 ISO 18000-6 Type A/B 충돌방지 알고리즘	15
2. 개선된 EPCglobal 프로토콜의 충돌방지 알고리즘	19
IV. 시뮬레이션 및 구현	28
1. 시뮬레이션 결과	28
2. 결과 및 고찰	37
V. 결론	43
참고 문헌	45

Abstract

RFID(Radio frequency identification) is a technology that automatically identifies objects attached on electronic tags by using radio wave. RFID is considered as a core technology in establishing ubiquitous sensor network (USN) and Telematics service. Recently, international organization for standardization(ISO) and EPC-global organization have announced their documents for RFID Air-interface which is a standard communication protocol between a reader and passive tags at the 860~960MHz.

A typical RFID system consists of a reader, tags and a middleware. To identify the tag ID, the reader communicates with tags through a radio frequency(RF) communication link. An individual tag is first identified from a group of tags in the reader's interrogation zone by using anti-collision algorithm, and then the communication occurs between the identified tags and the reader. If there are several tags in the reader's interrogation zone, a collision occurs among data received from the tags. The collision can be resolved by using anti-collision algorithms.

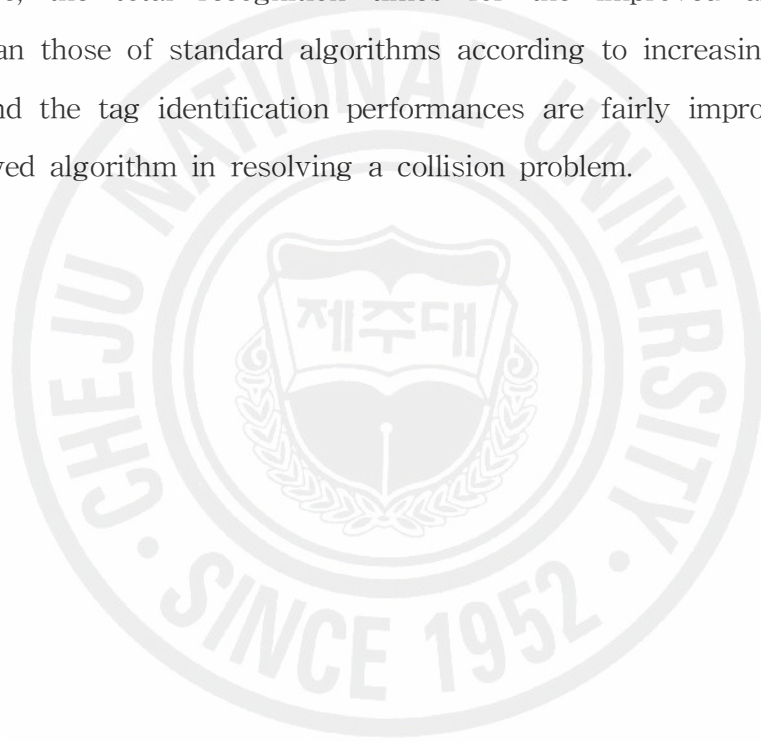
In this paper, UHF band RFID protocols, such as ISO 18000-6 Type A/B, EPC Class 0, Class 1 Gen 1, and Class 1 Gen 2, are investigated. And then the improved anti-collision algorithms using each protocol are proposed and analyzed. Also, the performance of proposed anti-collision algorithm is evaluated as comparing the standard algorithm through the simulation.

The improved ISO Type A and B algorithms use the way to reduce the timing of idle slots. The improved Class 0 algorithm utilized the collision-pointer and collision-counter, and restarts at the updated node to search the trace of tree state, but not at the root node. The improved Class 1 Gen 1 algorithm processes the bins that the collision does not occur, and then the collided bins stored in the stack are processed dynamically by searching

the collided-bit position of collided bin data. The improved Class 1 Gen 2 algorithm uses bigger Q value than the estimated number of tags when it is decided to the slot size of next slot.

As the results, in the case of improved Type A and Type B algorithms, the reduction ratios of the total recognition time are 24.3% and 6.6% for 1000 tags against those of standard algorithms, respectively. In the case of improved Class 0 , Class 1 Gen 1 and Class 1 Gen 2 algorithms, the ratios are 12.2%, 63.4% and 11.7% for the same number of tags.

Therefore, the total recognition times for the improved algorithms are shorter than those of standard algorithms according to increasing the number of tag. And the tag identification performances are fairly improved by using the improved algorithm in resolving a collision problem.



I. 서론

RFID(radio frequency identification) 시스템은 무선주파수 신호를 이용하여 태그에 부착된 IC칩에 저장되어 있는 고유 정보를 비접촉식으로 판독하는 식별장치이다. 또한 이 시스템은 리더의 인식영역 내에서 여러 개의 태그를 아주 짧은 시간 동안에 인식할 수 있고, 태그에 대용량의 데이터를 저장할 수 있으며, 반영구적으로 사용이 가능한 장점이 있다. 따라서 RFID는 기존의 바코드나 자기인식 장치의 단점을 보완하고 사용의 편리성을 향상시켜 줄 차세대의 유비쿼터스 센서 네트워크(usn)와 텔레매틱스 서비스 구현에 있어 가장 핵심적인 기술이다. 이러한 RFID 시스템은 태그, 리더, 미들웨어 및 응용 서비스 플랫폼으로 구성되고, 유무선 통신망과 연동되어 사용된다. 태그는 객체를 인식할 수 있는 정보를 가지고 공간상에 위치한다. 리더는 태그의 정보를 읽어내기 위해 태그와 송·수신하는 기기이며, 태그에서 수집된 정보를 미들웨어로 전송하는 기능을 한다. 미들웨어는 리더에서 계속적으로 발생하는 식별코드 데이터를 수집, 제어, 관리하는 기능을 하며, 객체의 정보를 활용하여 응용처리를 수행한다^{[1]-[5]}.

리더가 태그를 인식하는 과정에서 수많은 태그들이 하나의 리더 인식영역 내에 존재할 수 있다. RFID 시스템에서 리더는 인식영역 내의 태그에 요청 메시지를 전송하고, 요청 메시지를 수신한 태그는 자신의 정보를 리더에 전송한다. 그러나 인식영역 내에 다수의 태그가 존재하여 동시에 요청 메시지에 대한 응답을 하게 되면 RF 통신 채널 상에 충돌(collision)이 발생하게 되고, 결과적으로 리더는 태그의 정보를 정확하게 수신하지 못하게 된다. 이러한 문제를 해결하기 위해 충돌방지 알고리즘(anti-collision algorithm)이 사용되며, 이는 다중접속 방식 시스템의 성능을 결정짓는 중요한 요소가 된다^{[1][2][6]}. RFID 시스템의 성능은 태그를 인식하는데 필요한 시간과 태그가 소모하는 전력으로 결정되어진다. 다중접속 방식을 갖는 RFID 시스템에서 태그 데이터 간의 충돌은 인식시간과 소모 전력의 증가를 가져오는 직접적인 원인이 되므로 RFID 시스템의 효율성을 높이기 위해서는 태그 충돌을 최소화 하는 다중접속 기술이 필요하다.

현재 다양한 충돌방지 알고리즘이 제안되고 있으며, 충돌방지 알고리즘은 크게 결정적 알고리즘(deterministic algorithm)과 확률적 알고리즘(probability algorithm)으로 구분된다. 결정적 알고리즘은 이진탐색(binary search)을 기반으로 한 것이고, 확률적 알고리즘은 알로하(aloha)를 기반으로 한다. 이진탐색 방식이란 여러 개의 태그가 고유번호를 동시에 전송하는 경우, 충돌이 발생하는 비트 위치를 이용하여 태그의 고유번호를 인식하는 것을 말한다. 즉 충돌이 발생한 경우에 응답하는 태그의 수를 줄임으로써 태그 충돌을 방지한다. 이진탐색 방식은 메모리 형과 비메모리 형으로 나눌 수 있고^[7], 비메모리 형 이진트리 방식은 트리워킹, 퀴리트리, 충돌추적으로 나눌 수 있다^{[8]-[11]}. 알로하 방식은 다중 통신의 가장 단순한 형태로 데이터 패킷이 발생하자마자 태그로부터 리더로 데이터를 전송하는 방식이다. 태그들이 동시에 데이터를 리더로 전송하면 충돌이 일어나는데, 이때 리더는 충돌이 일어난 태그들에 랜덤 백오프(back-off) 방식으로 다음 재전송할 시점을 정하게 한다. 알로하 방식은 순수 알로하, 슬롯 알로하, 프레임 알로하가 있고, 프레임 알로하에는 고정프레임 알로하와, 동적프레임 알로하가 있다^{[12]-[16]}. 이러한 리더와 태그간의 통신 방식은 여러 프로토콜로 정의되어 있다. RFID에 관한 하드웨어 규정과 리더와 태그간의 통신방식 등을 제정하는 표준은 ISO 18000, EPCglobal, U-센터 등 여러 단체에서 진행되고 있다.

본 논문에서는 충돌방지 알고리즘의 성능을 개선하기 위하여 UHF 대역 프로토콜로 규정된 ISO의 18000-6 Type A와 Type B, EPCglobal의 Class 0, Class 1 Gen 1, Class 1 Gen 2의 리더와 태그사이의 통신 방식과 태그 인식 과정을 분석한다. 그리고 각 프로토콜에서 규정된 충돌방지 알고리즘인 프레임알로하, 확률적알로하(sbt), 이진트리, 빈슬롯, 슬롯알로하 알고리즘의 성능을 C언어로 구현하여 확인한다. 그리고 분석된 결과를 바탕으로 리더와 태그 간의 송수신되는 명령어를 줄이거나 태그 응답 타이밍을 줄이는 등의 방식을 사용하여 각 알고리즘의 성능을 개선하고 그 성능을 고찰한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 UHF 대역 에어인터페이스에 대해 설명하고, 3장에서는 각 프로토콜에서 규정한 충돌방지 알고리즘의 개선 방안을 제시하고, 4장에서는 시뮬레이션 결과와 리더 시스템에서의 충돌방지 알고리즘 구현에 대하여 논의하며, 5장에서는 결론을 내린다.

II. UHF 대역 RFID 프로토콜

본 장에서는 현재 UHF 대역 RFID 프로토콜로 정의된 ISO 18000-6 Type A와 Type B, EPCglobal의 EPC Class 0, EPC Class 1 Gen 1, EPC Class 1 Gen 2의 에어인터페이스에 대해 설명한다.

1. ISO 18000-6 Type A/B 에어인터페이스

1) Type A

ISO 18000-6 Type A는 충돌 방지 알고리즘으로 프레임 슬롯 알로하(frame slotted aloha) 방식을 사용한다. 변조 방법은 AM(amplitude modulation), 데이터 코딩은 리더에서 태그로 전송할 때는 PIE(pulse interval encoding), 태그에서 리더로 전송할 때는 FM0을 사용한다^{[14][17]}.

그림 1은 Type A에서 태그를 인식하는 과정을 흐름도로 나타낸 것이다. Type A에서 태그를 인식하는 과정은, 먼저 리더에서 태그로 최초의 명령을 송신하기 전에 채널이 비어있는지 확인하는 단계를 거친다. 이 시간을 T_{aq} (type A quiet time)라고 하며, 적어도 $300\mu s$ 이상을 유지해야 리더는 채널이 비어있다고 판단하고 태그에게 명령어를 전송한다. 이 시간동안 리더는 계속해서 인식영역 내에 RF 신호를 전송하고, 적절한 RF 에너지를 수신한 태그는 리더 명령을 수행할 수 있는 Ready 상태로 전환된다.

리더는 Ready 상태에 있는 태그들에게 라운드 사이즈(round size)를 포함하는 Init_round 명령어를 전송한다. 이 명령을 수신한 태그들은 태그 인식 과정을 수행하는 Round_active 상태로 전환되고, 라운드 사이즈 값과 태그 내의 RNG(random number generator)를 이용하여 자신의 ID 전송에 사용할 슬롯 넘버(slot_number)를 선택하고 슬롯카운터(slot_counter)를 0으로 초기화한다. 만약 슬롯카운터와 자신이 선택한 슬롯넘버가 일치하는 태그는 자신의 4비트 서명(signature)을 포함하는 ID를 응답으로 리더로 전송하고, 불일치하는 태그는 다음

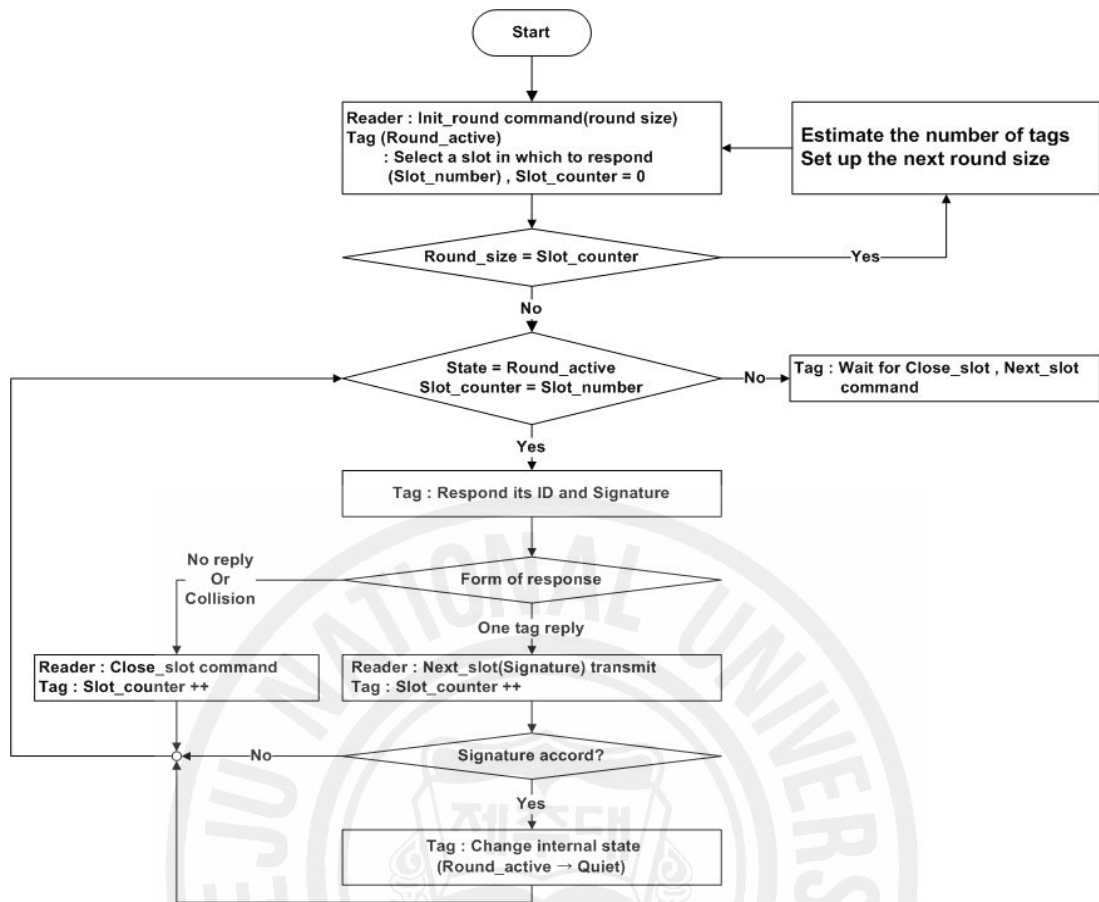


그림 1. Type A의 태그 인식 동작 흐름도

명령을 기다린다. 태그의 응답을 받은 리더에서, 그 응답유형이 하나의 태그만 응답을 하였다면, 리더는 수신한 4비트 서명을 포함하는 Next_slot 명령어를 전송한다. 이 명령어를 수신한 태그는 자신의 슬롯카운터 값을 1씩 증가하고, 명령어에 포함된 서명과 자신의 서명이 일치하는 태그는 자신의 상태를 Round_active 상태에서 Quiet 상태로 전환한다. 만약 그 응답유형이 충돌이거나 응답이 없을 경우 리더는 Close_slot 명령어를 전송하고, 이 명령어를 수신한 태그는 자신의 슬롯카운터 값을 1씩 증가한다. 그리고 나서 슬롯넘버와 슬롯카운터가 일치하는 태그는 자신의 ID와 4비트 서명을 응답한다. 한 라운드는 슬롯카운터와 라운드 사이즈가 같아질 때 까지 계속되며, 한 라운드가 끝난 뒤 인식영역 내의 태그 수를 예측하여 라운드 사이즈를 재설정한다. 리더는 Init_round 명령어를 전송한다. 태그는 전송할 새로운 슬롯넘버와 서명을 선택하여 새로운 라운

드를 시작한다. ISO 18000-6 Type A는 이러한 과정을 반복적으로 수행하여 태그를 인식한다.

2) Type B

ISO 18000-6 Type B는 충돌 방지 알고리즘으로 SBT(slotted binary tree)와 유사한 방식인 확률적(probabilistic) 알로하 방식을 사용한다. 변조 방법은 AM(amplitude modulation), 데이터 코딩은 리더에서 태그로 전송할 때는 Manchester, 태그에서 리더로 전송할 때는 FMO를 사용한다^{[17][18]}.

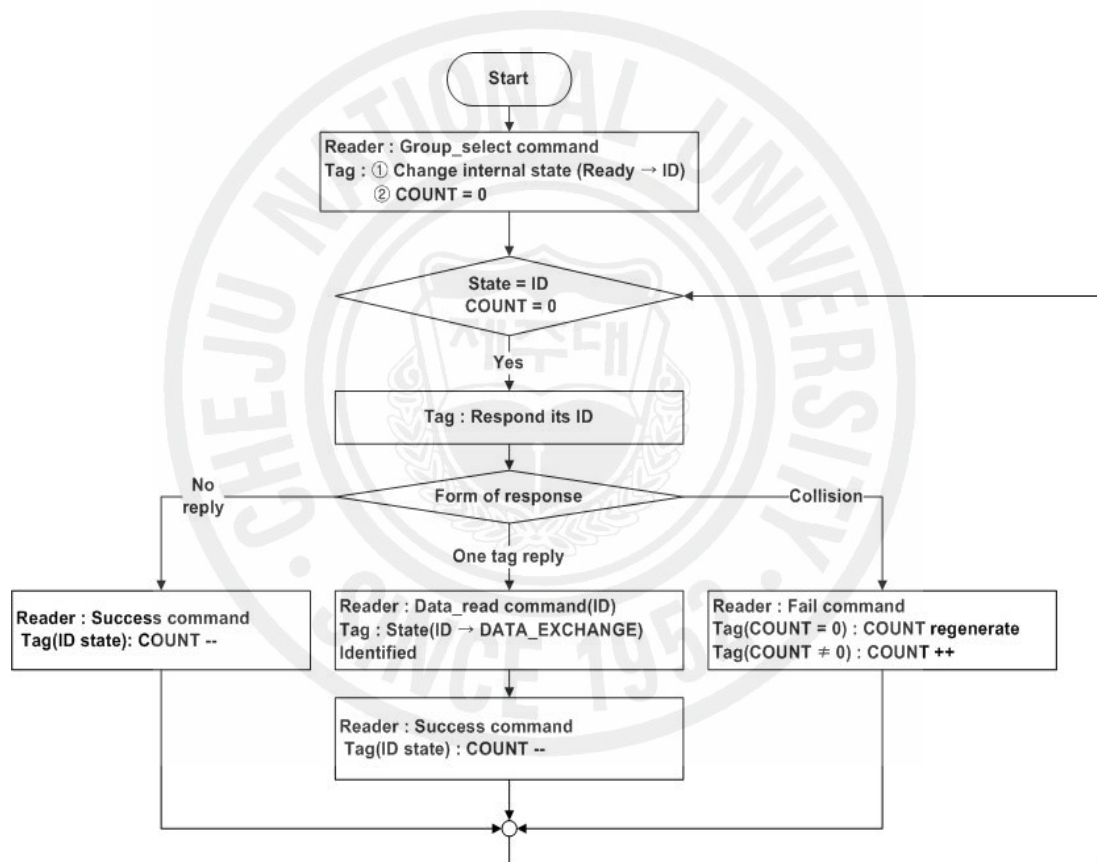


그림 2. Type B의 태그 인식 동작 흐름도

그림 2는 Type B에서 태그를 인식하는 과정을 흐름도로 나타낸 것이다. 리더의 인식 범위 내에 있는 태그가 리더로부터 동작에 필요한 에너지를 공급받으면 Power-off 상태에서 리더 명령어를 수행할 수 있는 Ready 상태로 전환된다. Ready 상태에 있는 태그들이 Group_select 명령어를 수신하면 자신의 상태를 ID

상태로 전환하고, 자신의 COUNT를 0과 1중에 0으로 설정한다. 태그의 상태가 ID 상태이고 선택한 COUNT가 0인 태그는 자신의 ID를 리더로 응답하고, 0이 아닌 태그는 다음 명령을 기다린다. 그 응답유형이 하나의 태그만 응답하였다면, 리더는 수신한 ID를 포함하는 Data_read 명령어를 전송한다. 이 명령어를 수신한 태그 중 ID가 일치하는 태그는 자신의 상태를 Data_exchange 상태로 전환시킨다. 리더는 다시 Success 명령어를 전송하고, 이 명령어를 수신한 태그는 자신의 COUNT 값을 1씩 감소시킨다. 만약 태그의 응답이 없을 경우, 리더는 Success 명령어를 전송하고, 이 명령어를 수신한 태그는 자신의 COUNT 값을 1씩 감소시킨다. 만약 충돌이 발생하였다면, 리더는 Fail 명령어를 전송한다. 이 명령어를 수신한 태그 중 COUNT가 0인 태그, 즉 이전 슬롯에서 응답을 했던 태그는 자신의 COUNT를 0과 1중에서 다시 선택한다. COUNT가 0이 아닌 태그, 즉 응답을 하지 않았던 태그는 자신의 COUNT 값을 1씩 증가시킨다. 그리고 나서 ID 상태이고, COUNT가 0인 태그는 자신의 ID를 리더로 응답한다. 이러한 과정을 반복함으로써 태그들을 인식할 수 있다.

2. EPCglobal 프로토콜의 에어인터페이스

1) EPC Class 0

Class 0은 태그를 인식하기 위한 방법으로 이진트리(binary tree) 방식을 사용한다. 이진트리 방식은 비트 단위로 리더와 태그가 신호를 송수신하며 태그 ID를 찾는 방식이다. 그 방법의 설명은 다음과 같다^{[19][20]}.

그림 3은 EPC Class 0 태그의 상태 천이도를 나타낸 것이다. Class 0 태그는 전력을 공급 받으면 Dormant 상태로 되고, 리셋 신호를 수신하면 Calibration 상태, 유효한 신호를 수신하면 Global command start 상태가 된다. 이 상태에서 태그는 비트 '0'을 수신하거나 플래그(flag)가 클리어되면 Tree traversal 상태로 들어간다. Tree traversal 상태는 태그 ID 비트열을 탐색하는 단계로 이 상태에 있는 태그는 자신의 ID 비트열을 MSB에서부터 시프트 시키면서 리더로 한 비트씩 전송한다. 리더는 태그의 응답들을 검사하고 태그로부터 수신한 비트를 다시

태그로 전송한다. 만약, 태그의 응답들이 충돌이면 리더는 프로그램 된 상태에 따라 0 또는 1로 분기하여 선택된 비트를 태그로 전송한다. 리더로부터 수신한 비트가 이전에 리더로 전송한 비트와 동일하면 태그는 한 비트를 시프트하여 다음 비트를 리더로 전송한다. 불일치한 태그는 Traversal mute 상태로 전환되어 트리과정에 참여하지 않게 된다. 이 과정은 하나의 태그가 완벽히 인식되기 위해 마지막 비트 위치까지 진행되고, 태그는 마지막 비트를 다시 한 번 더 리더로 전송한다. 리더로 전송한 비트 수가 [태그 ID 길이 + 1] 인 태그가 리더로부터 'null' 데이터를 수신하면 Singulated command 상태가 되고 태그 하나가 완벽히 인식된다. 태그 하나가 인식 되고 리더가 'null' 데이터를 전송하면 Traversal mute 상태인 태그들은 Tree start 상태로 되어 트리에 참여하고, 인식된 태그는 ID 플래그가 세팅되면 Dormant 상태가 된다.

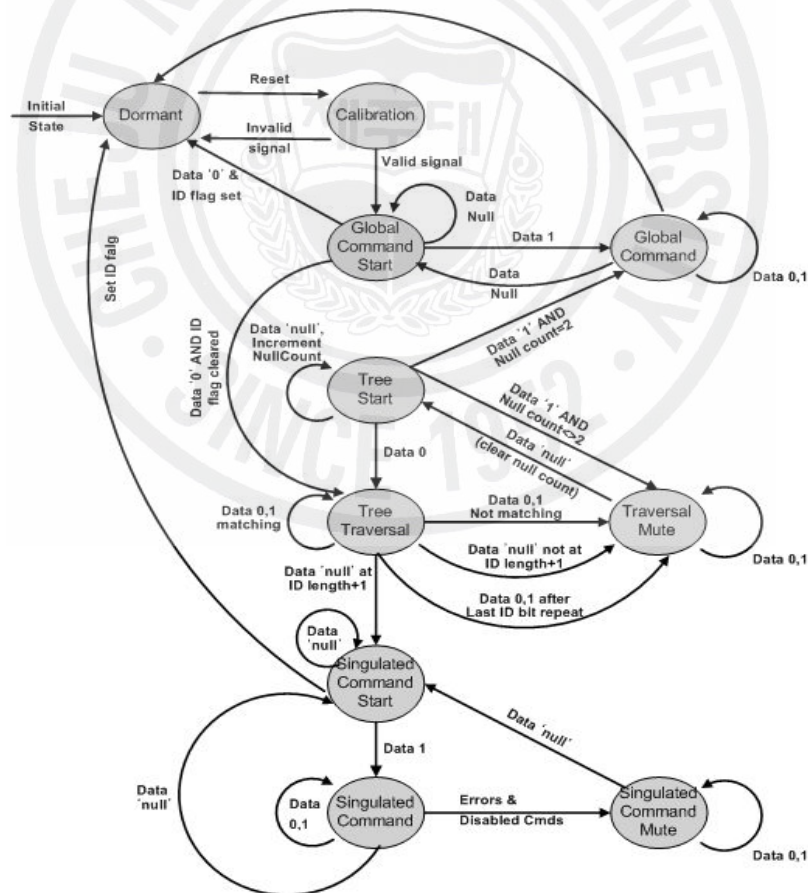


그림 3. EPC Class 0 태그의 상태천이도

2) EPC Class 1 Gen1

EPC Class 1 Gen 1 에서 리더와 태그의 통신은 패킷 단위로 이루어진다^{[21][22]}. 리더에서 태그로 보내는 패킷은 완전한 명령을, 태그에서 리더로 보내는 패킷은 완전한 응답을 포함한다. 리더가 태그로 전송하는 명령어는 8개의 필드와 5개의 패리티비트 필드로 구성되고, 그 형식은 다음과 같다.

**[PREAMBLE][CLKSYNC][SOF][CMD][P1][PTR][P2][LEN][P3][VALUE]
[P4][P5][EOF]**

태그는 리더 명령어를 수신하면 응답을 하거나, 상태를 변화 시킨다. 태그가 리더 동작 영역으로 들어가게 되어 적절한 파워의 리더 신호를 수신하면 태그는 모든 상태로 변화 가능한 Awake 상태가 된다. 태그는 ScrollID, ScrollAllID, VerifyID, PingID 명령어를 수신하면 Reply 상태가 되어 리더에게 응답 신호를 전송한다. 완전히 식별된 태그의 신호는 다른 태그를 인식하는 동안 방해가 되기 때문에 리더는 식별된 태그에 Quiet 명령어를 전송하고 수신한 태그는 잠시 응답하지 않는 Asleep 상태가 되며, 그 반대 과정은 Talk 명령어를 사용한다. 완전히 인식된 태그 중에 프라이버시 등의 문제로 제거되어야 할 때 리더는 그 태그에게 Kill 명령어를 보내고 태그는 암호와 비교하여 일치하면 Dead 상태가 된다.

태그 인식과정에서 태그가 충돌 없이 인식되었다면 리더는 그 태그에 ScrollID 명령어를 전송하고 명령어를 수신한 태그는 그 응답으로 자신의 전체 ID를 리더에 전송하고, 리더는 태그 응답을 받아 CRC체크를 하여 오류가 없다면 태그의 ID를 저장한다. PingID 명령어는 태그 인식과정에서 리더가 태그의 신호를 요구하는 명령어이다. 리더는 PingID 명령어를 인식영역 내의 임의의 태그에 전송한 후 3비트 단위의 Bin modulation 작업을 한다. PingID 명령어 패킷에서 [VALUE]에 맞는 태그는 [VALUE] 다음의 비트열과 Bin 값이 같은 Bin에서 8비트로 응답을 하게 된다. 만약 하나의 Bin에서 여러 개의 태그 신호가 응답되면 충돌이 발생한 것이고, 하나의 태그 신호가 응답되면 충돌이 일어나지 않은 것이다. 만약, 충돌이 일어나지 않았다면 그 Bin 값을 [VALUE]에 추가하여 ScrollID

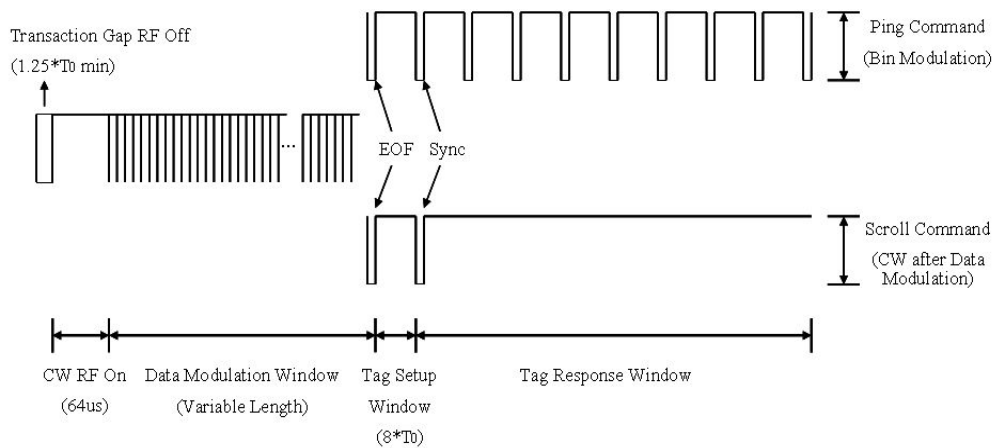


그림 4 . 리더와 태그의 변조 과정

명령어를 전송하고, 충돌이 일어났다면 그 Bin 값을 [VALUE]에 추가하여 다시 PingID 명령어를 전송한다.

그림 4는 리더와 태그의 변조 과정을 나타내었다. 리더에서 태그로 전송되는 신호는 5개의 단계로 이루어진다. 리더에서 태그로 전송되는 신호는 첫 번째 단계인 ‘Transaction Gap RF Off’에서 시작하고, 그 뒤에 $64\mu s$ 의 변조되지 않은 신호인 CW신호가 전송된다. 두 번째 단계는 리더 명령어 부분 중 ‘[PREAMBLE]’ 부분이다. 세 번째 과정은 ‘Data Modulation Window’ 부분으로 리더 명령어 패키지가 전송된다. 네 번째 단계는 짧은 ‘Tag Setup Window’로 태그에게 리더 명령어를 해석할 시간을 제공한다. 마지막 단계는 태그 응답이 수신되는 부분이다.

리더 명령어가 전송되는 부분인 ‘Data Modulation Window’에서 리더 명령어 패키지의 한 부분인 [CLKSYNC]는 태그에 동기를 제공하고, 클럭 주기 T_0 은 태그의 전송속도를 결정한다. 태그는 다음 ‘Transaction Gap’이 시작되기 전까지 리더에서 검출되어야 하고, ‘Data Modulation’의 [EOF]다음부터 다음 명령어 패키지의 ‘Transaction Gap’ 사이의 시간을 ‘Coast Interval’이라 하여 이 시간을 최대 $20ms$ 로 규정하고 있다.

Class 1 Gen 1의 충돌방지 알고리즘은 이진트리에 기반을 둔다. 이진트리는 원래 1비트 단위로 트리를 확장하는 형태이지만, Class 1 Gen 1의 빈 슬롯(bin slot) 충돌방지 알고리즘은 3비트 단위로 확장하는 형태이다. Class 1 Gen 1에

서 태그 인식은 리더가 'VALUE=0, LEN=0'을 포함하는 PingID 명령어를 인식영역 내의 임의의 태그에 전송함으로써 시작된다. PingID 명령어를 전송한 후 리더는 Bin Modulation 작업을 하고 태그의 비트열이 Bin 값과 일치하는 태그가 각 Bin에서 응답을 한다. Bin을 검사한 후 각 Bin에서 충돌 여부를 판단하고 충돌이 일어나지 않으면, 리더는 Bin 값을 [VALUE]에 포함한 후 ScrollID 명령어를 태그에 보내게 된다. [VALUE]에 맞는 태그는 자신의 전체 ID를 ScrollID 명령어에 대한 응답으로 리더에 보내게 되고, 리더는 CRC체크를 하여 오류 없이 전송 되었다면 태그 식별 코드를 저장하고 과정을 처음으로 되돌린다. 만약, Bin에서 충돌이 일어나면 리더는 Bin 값을 [VALUE]에 저장한 후 다시 PingID 명령어를 리더 인식영역 내에 전송한다. 여기서, Bin 검사는 Bin의 처음 '000'부터 시작한다.

3) EPC Class 1 Gen 2

EPC Class 1 Gen 2는 Class 1 Gen 1의 단점을 보완하고, ISO 18000-6과의 통합을 고려한 업데이트 버전이며, 현재 ISO 18000-6 Type C 버전으로 채택되었다. Class 1 Gen 2의 태그 인식은 슬롯 알로하(slotted aloha) 방식을 사용한다 [23]-[25].

그림 5는 EPC Class 1 Gen 2의 태그 인식 과정을 나타낸 것이다. 인식영역 내의 태그를 인식하기 위해서 리더는 기본적으로 SELECT, INVENTORY 그리고, ACCESS의 세 동작과정을 이용한다.

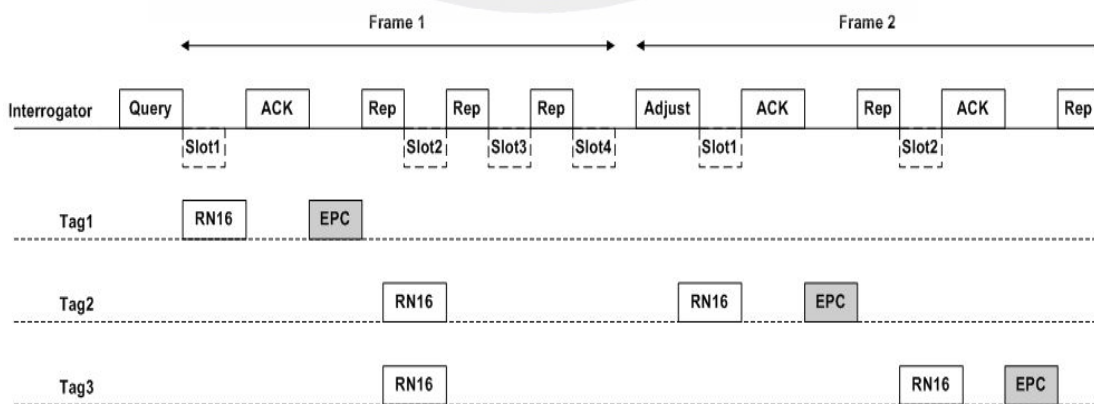


그림 5. EPC Class 1 Gen 2의 태그 인식 과정

SELECT 과정에서는 INVENTORY 과정에 앞서, 통신할 특정 태그를 리더가 선택하는 과정, INVENTORY 과정은 슬롯을 발생시켜 하나 이상의 태그가 응답하게 하고 응답한 태그 중에서 하나의 태그 인식에 필요한 PC, EPC, CRC-16을 요청하는 과정, ACCESS 과정은 INVENTORY 과정을 성공적으로 마친 태그와 리더가 일-대-일 통신을 하는 과정이다.

INVENTORY 과정은 Query, QueryAdjust, QueryRep, ACK, 그리고 NAK 등의 명령어들로 구성된다. Query는 인벤토리 프레임의 슬롯 개수를 초기화하고 태그가 프레임에 참여하게끔 한다. 프레임에 참여한 태그는 Query 명령어에 포함되어 있는 슬롯 파라미터 Q를 참조하여 랜덤넘버(random number, $0 \sim 2^Q - 1$)를 선택하고 이를 슬롯카운터(slot_counter)에 로드한다. 여기서 '0'을 선택한 태그들은 Reply 상태로 전환되고, 전환되는 즉시 리더로 응답 메시지를 전송한다. 그 밖의 '0'이 아닌 랜덤넘버를 선택한 태그들은 Arbitrate 상태로 전환되고 QueryAdjust 또는 QueryRep 명령어를 기다린다.

하나의 슬롯에 대해 단일 태그가 응답으로 RN16(random number of 16-bit)을 리더로 전송했다면, 충돌이 발생하지 않는다. 이때 리더는 수신한 RN16을 포함하는 ACK 명령어를 태그로 전송한다. 이 과정을 통해 리더에 인식된 태그는 Acknowledge 상태로 전환되고, PC, EPC, 그리고 CRC-16을 리더에 전송한다. 리더는 QueryAdjust 또는 QueryRep 명령어를 전송하여 태그를 대기상태로 전환하고 태그 데이터를 메모리에 저장한다. 만약 일정시간동안 태그가 ACK 수신에 실패하거나 잘못된 ACK를 수신하면 태그는 다시 Arbitrate 상태로 전환되어 다음 프레임까지 대기하게 된다.

다중 태그가 RN16을 동시에 리더로 전송하였다면 리더는 과형 레벨에서 충돌을 분석함으로써 충돌 태그들 중 신호레벨이 가장 큰 하나의 태그로부터 RN16을 분별해낼 수 있다. 일치하는 ACK를 받은 태그는 Acknowledge 상태로 전환되고 자신과 일치하지 않는 ACK를 받은 태그는 Arbitrate 상태로 전환되어 QueryAdjust 또는 QueryRep 명령어를 기다리게 된다. 어떤 특정 시점에서 NAK가 태그에 전송되면 그 프레임에 있는 모든 태그들은 인식메모리에 저장되지 않고 Arbitrate 상태로 전환된다.

프레임을 초기화시키는 Query가 발생한 후에, 리더는 하나 이상의

QueryAdjust 또는 QueryRep 명령어를 전송한다. QueryAdjust 명령어은 이전 Query를 반복하고 Q 값을 증감시킨다. QueryRep 명령어는 Q와 같은 파라미터들을 변화시키지 않고 Query를 반복한다. 이때 새로운 태그를 프레임에 추가시키지는 않는다. 어떤 시점에서 리더는 새로운 Query 명령어를 전송하는데, 그로 인해 새로운 프레임이 시작된다. Arbitrate나 Reply 상태에 있던 태그는 처음으로 QueryAdjust 명령어를 수신하고 Q를 조정한다. 그래서 랜덤넘버를 선택하고 랜덤넘버는 그들의 슬롯카운터로 로드된다. 여기서 Arbitrate 상태에 있는 태그들은 그들의 슬롯카운터를 QueryRep 명령어를 수신할 때마다 감소시키고 슬롯카운터가 [0000h]에 도달했을 때 Reply 상태로 전환되어 RN16을 리더에 전송한다. 이런 과정을 반복적으로 수행함으로써 태그를 인식할 수 있다. 그림 6은 하나의 태그가 응답했을 경우 INVENTORY 과정과 ACCESS 과정을 나타낸 것이다.

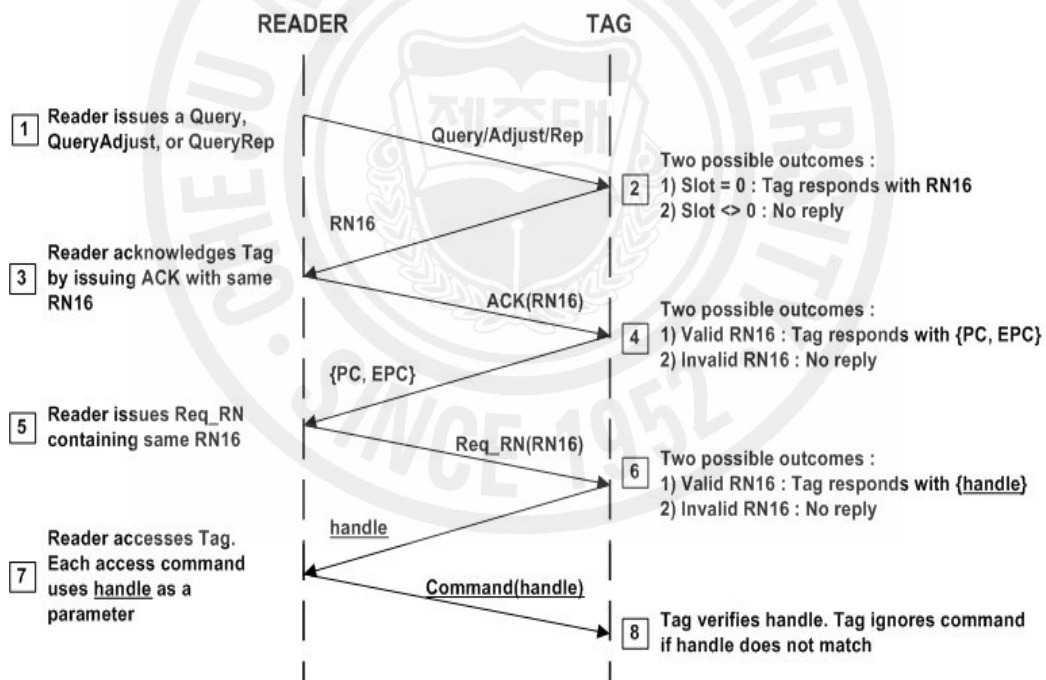


그림 6. 하나의 태그에 대한 INVENTORY, ACCESS 과정 예시

Ⅲ. 개선된 UHF 대역 RFID 충돌방지 알고리즘

본 장에서는 UHF 대역 RFID 프로토콜에서 정의된 충돌방지 알고리즘기법의 개선방안을 제안하고 이를 설명한다.

1. 개선된 ISO 18000-6 Type A/B 충돌방지 알고리즘

1) 개선된 Type A 알고리즘

그림 7은 슬롯 알로하 방식에서 성능을 개선하는 방안인 Early-end 방식이다^[26]. 이 방식은 태그의 응답이 없는 슬롯의 시간을 단축함으로써 성능을 개선하는 방식이다. 슬롯의 범위는 리더로부터의 무선신호 SOF(start of frame)와 EOF(end of frame)에 의해 결정된다. EOF 신호는 리더 명령어에 의해 슬롯 어느 구간에서나 전송이 가능하다. 태그의 응답이 있는 구간을 제외하고, 태그의 응답이 없는 구간 E-F, F-G, G-H, I-J 에서 리더가 EOF 명령어를 일찍 전송하면 슬롯 범위를 단축할 수 있다. 이를 Type A에 적용하여 성능을 개선하였다.

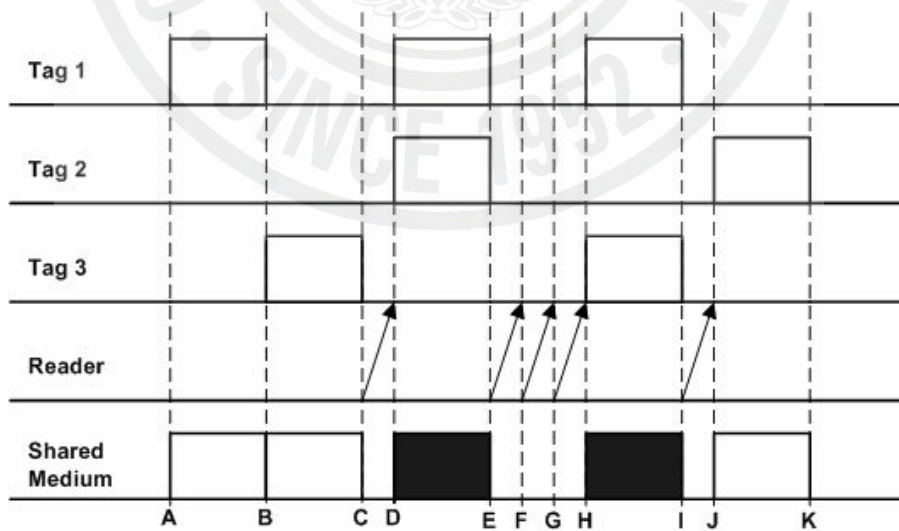


그림 7. 슬롯 알로하 성능을 개선하는 방안 : Early-end

그림 8은 Type A의 링크타이밍(link timing)을 나타낸 것이고, 그림 9는 이를 적용한 링크타이밍을 나타낸 것이다. 하나의 리더 명령어에 대하여 태그는 [Preamble][Flags][Parameters][Data][CRC-16]의 형태로 리더에 응답한다. 본 논문에서는 [Preamble][Flags][Parameters] 부분에서 응답이 있는지 없는지를 판별하여 응답이 없는 슬롯인 경우 슬롯구간을 줄여 다중태그 인식 시간의 성능을 개선하였다.

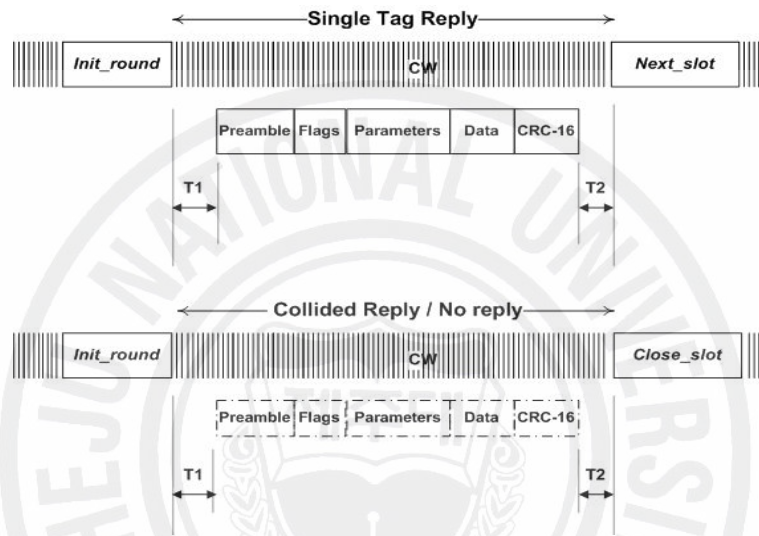


그림 8. Type A의 규정된 링크타이밍

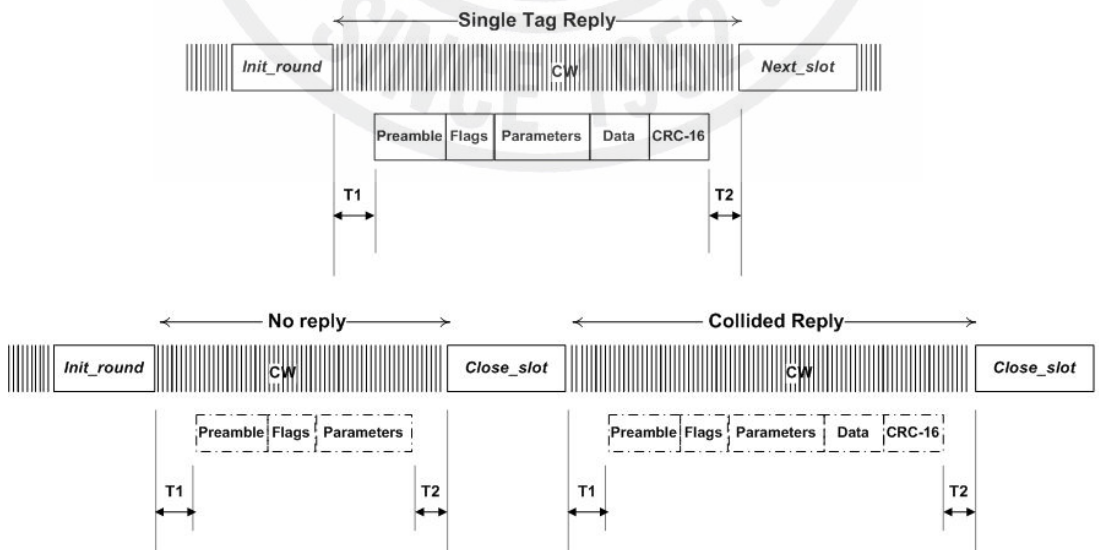


그림 9. 개선된 Type A 알고리즘의 링크타이밍

2) Type B

그림 10은 확률적 알로하(SBT) 방식을 나타낸 것이고, 그림 11은 성능을 개선한 수정된(modified) SBT 방식을 나타낸 것이다^[27]. 이 방식은 위의 Early-end 방식과 유사한 방식으로 태그의 응답이 없는 슬롯의 시간을 줄임으로써 성능을 개선하는 방법이다. 여기서 C는 충돌이 발생한 슬롯, I는 응답이 없는 슬롯, S는 하나의 태그가 응답한 슬롯이다. 충돌이 발생한 슬롯에서 태그들은 다시 '0'과 '1'로 분기하고, '0'을 선택한 태그들을 우선 인식한 뒤 '1'을 선택한 태그들을 인식하게 된다. 분기한 뒤 슬롯 2와 슬롯 4처럼 태그가 유효하게 인식되는 경우, 슬롯 3, 슬롯 5, 슬롯 8처럼 충돌이 다시 발생하는 경우, 슬롯 6처럼 태그의 응답이 없는 경우가 있다. 태그의 응답이 없는 슬롯 6의 시간을 단축하는 방법을 적용하여 Type B의 성능을 개선하였다.

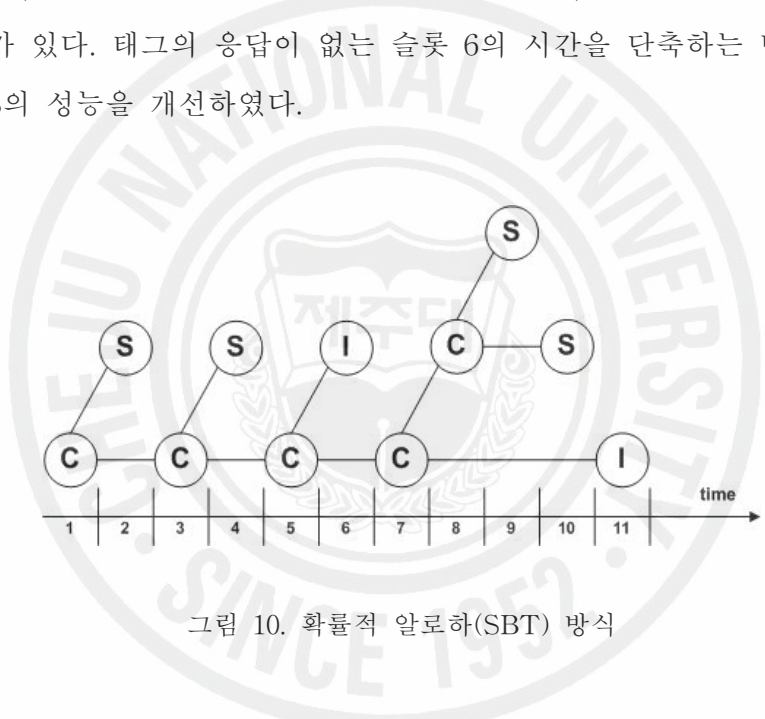


그림 10. 확률적 알로하(SBT) 방식

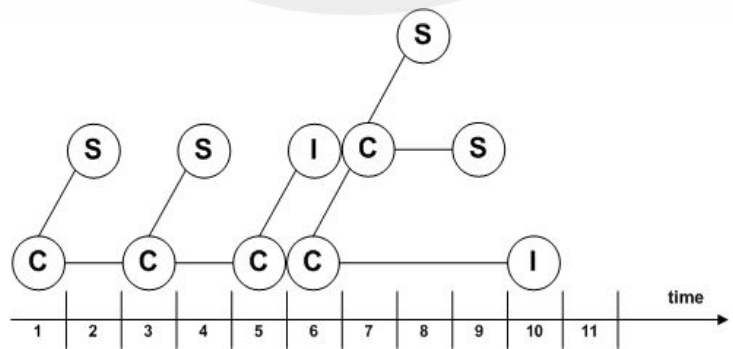


그림 11. 수정된(modified) SBT 방식

그림 12는 Type B의 링크타이밍을 나타낸 것이고, 그림 13은 이를 적용한 링크타이밍을 나타낸 것이다. 하나의 리더 명령어에 대하여 태그는 [Return Preamble][Data][CRC-16]의 형태로 리더로 응답한다. 본 논문에서는 [Return Preamble] 부분에서 응답이 있는지 없는지를 판별하여 응답이 경우 슬롯구간을 줄여 다중태그 인식 시간의 성능을 개선하였다.

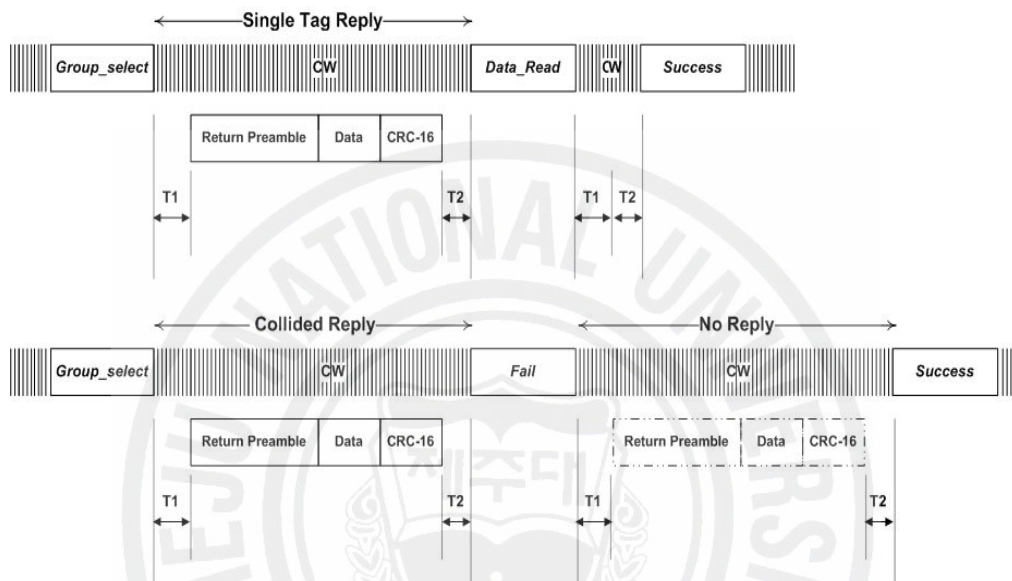


그림 12. Type B의 규정된 링크타이밍

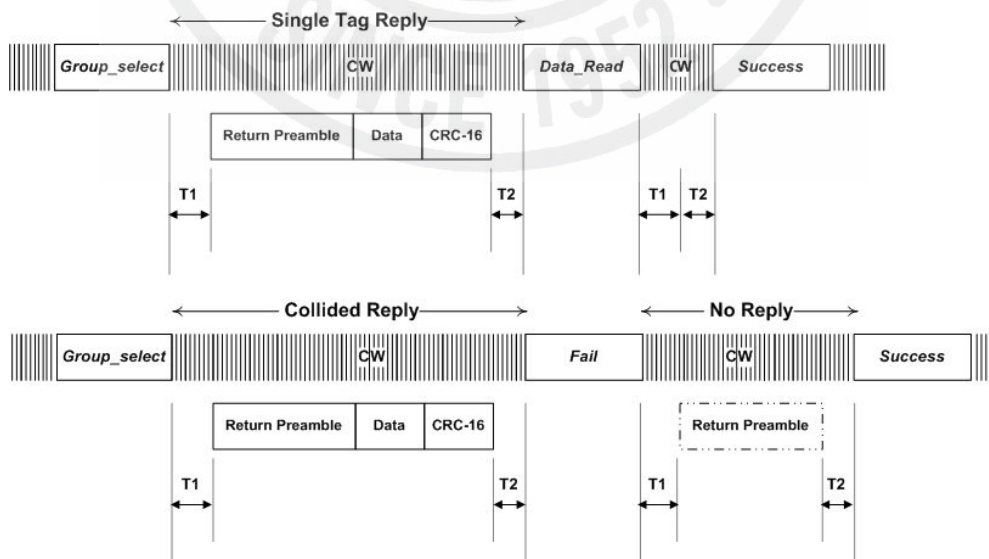


그림 13. 개선된 Type B의 링크타이밍

2. 개선된 EPCglobal 프로토콜의 충돌방지 알고리즘

1) 개선된 EPC Class 0 알고리즘

많은 수의 태그를 인식할 때, 태그들의 최상위 비트열은 동일한 패턴을 가질 수 있다. 이 점을 착안하여 개선된 Class 0 알고리즘은 매 read cycle 마다 루트(root) 노드부터 트리를 시작하지 않고 태그의 충돌이 발생한 비트 포인트(bit point)에서 시작하는 방식이다. 이를 위하여 태그에 충돌 비트 포인트(collision_bit_pointer)와 충돌 카운터(collision_counter)를 추가하여 충돌이 발생하면 충돌 비트 포인트로 충돌 위치를 기억하고 충돌 카운터를 증가시킨다. 그리고 태그 하나를 완벽히 인식 하면 충돌 카운터를 감소시켜 '0'이 되는 태그는 Tree traversal 상태로 되어 충돌 위치부터 트리를 시작한다.

그림 14와 15는 개선된 Class 0 알고리즘의 동작의 예를 설명한 것이다. 리더와 태그 간의 통신이 설정되면 태그는 Tree traversal 상태로 들어간다. 이 상태에 있는 태그는 자신의 ID 비트열을 MSB부터 시프트 시키면서 리더로 한 비트씩 전송한다. 태그의 응답들이 서로 다른 비트(0과 1)이면 충돌이고, 서로 같은 비트가 수신되면 충돌이 아니다. 만약 그 응답유형이 충돌이 아니면, 리더는 수신된 비트를 다시 태그로 전송하고, Prefix_data 열에 추가한다. 만약 그 응답유형이 충돌이면, 리더는 미리 프로그램된 상태에 따라 0과 1로 분기하는데 본 논문에서는 0으로 분기하도록 하였다. 리더로 부터 수신한 비트가 만약 이전에 전송했던 비트(j-1 번째)와 일치하는 태그는 다음 번째(j) 비트를 리더로 전송한다. 만약 불일치하는 태그인 경우 자신의 상태를 Traversal mute 상태로 전환하고, 자신의 conflict_bit_point를 현 비트로 설정하고, conflict_count를 1씩 증가한다. 이런 비트열 전송은 하나의 태그를 완벽히 인식하기 위해 마지막 비트 위치까지 진행되고, 태그는 마지막 비트를 다시 한 번 더 리더로 전송한다. 리더로 전송한 비트 수가 [태그 ID 길이 + 1] 인 태그가 리더로부터 'null' 데이터를 수신하면 Singulated command 상태가 되고 태그 하나가 완벽히 인식된다. 태그 하나가 인식되고 리더가 'null' 데이터를 전송하면 Traversal mute 상태인 태그들은 자신의 conflict_count를 1씩 감소하고, 이 값이 0이 되는 태그는 Tree start 상태로 되어 트리에 참여한다. 그리고 conflict_bit_point인 부분부터 트리를 시작한다.

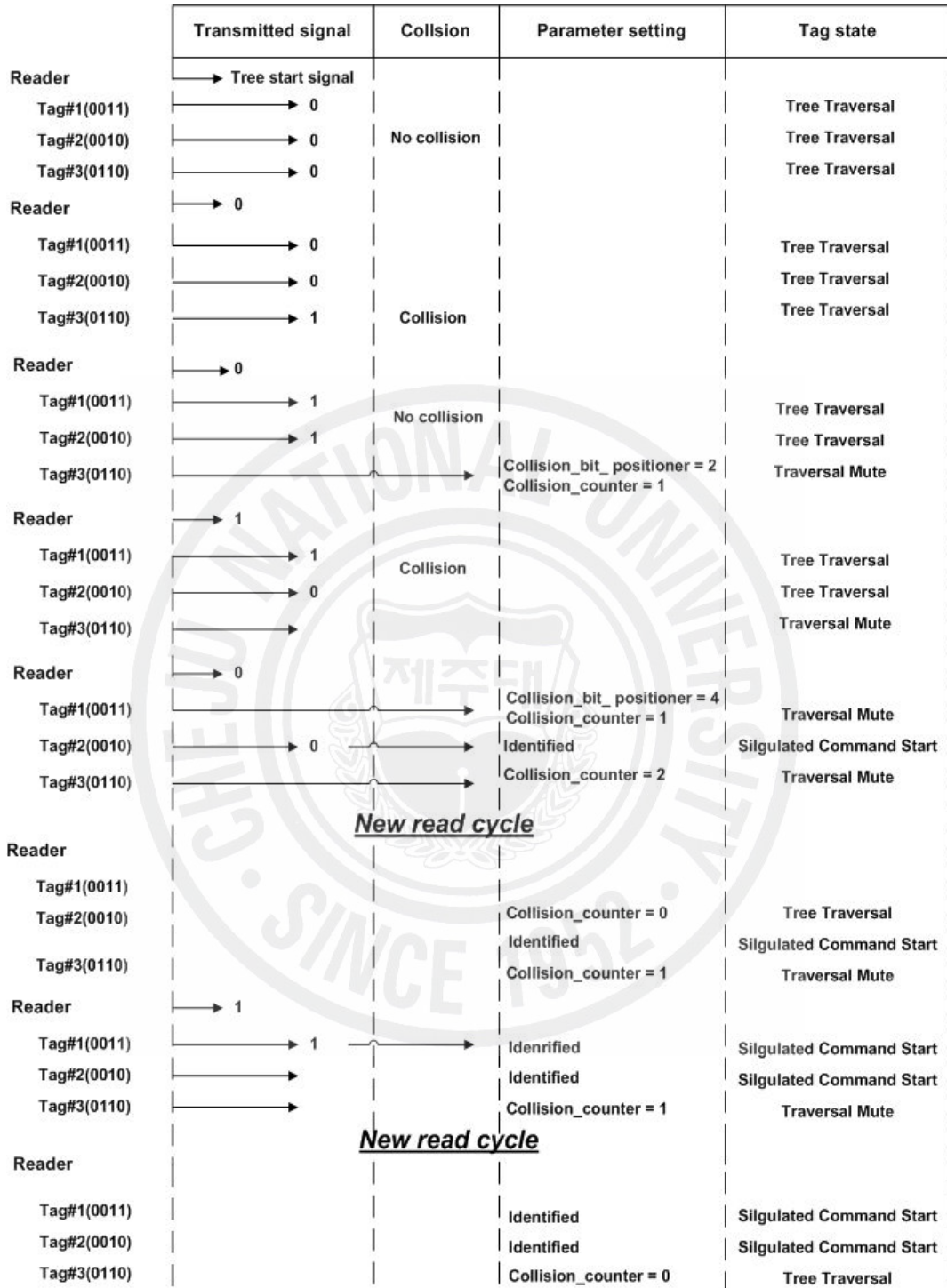


그림 14. 개선된 EPC Class 0 알고리즘의 동작 예시

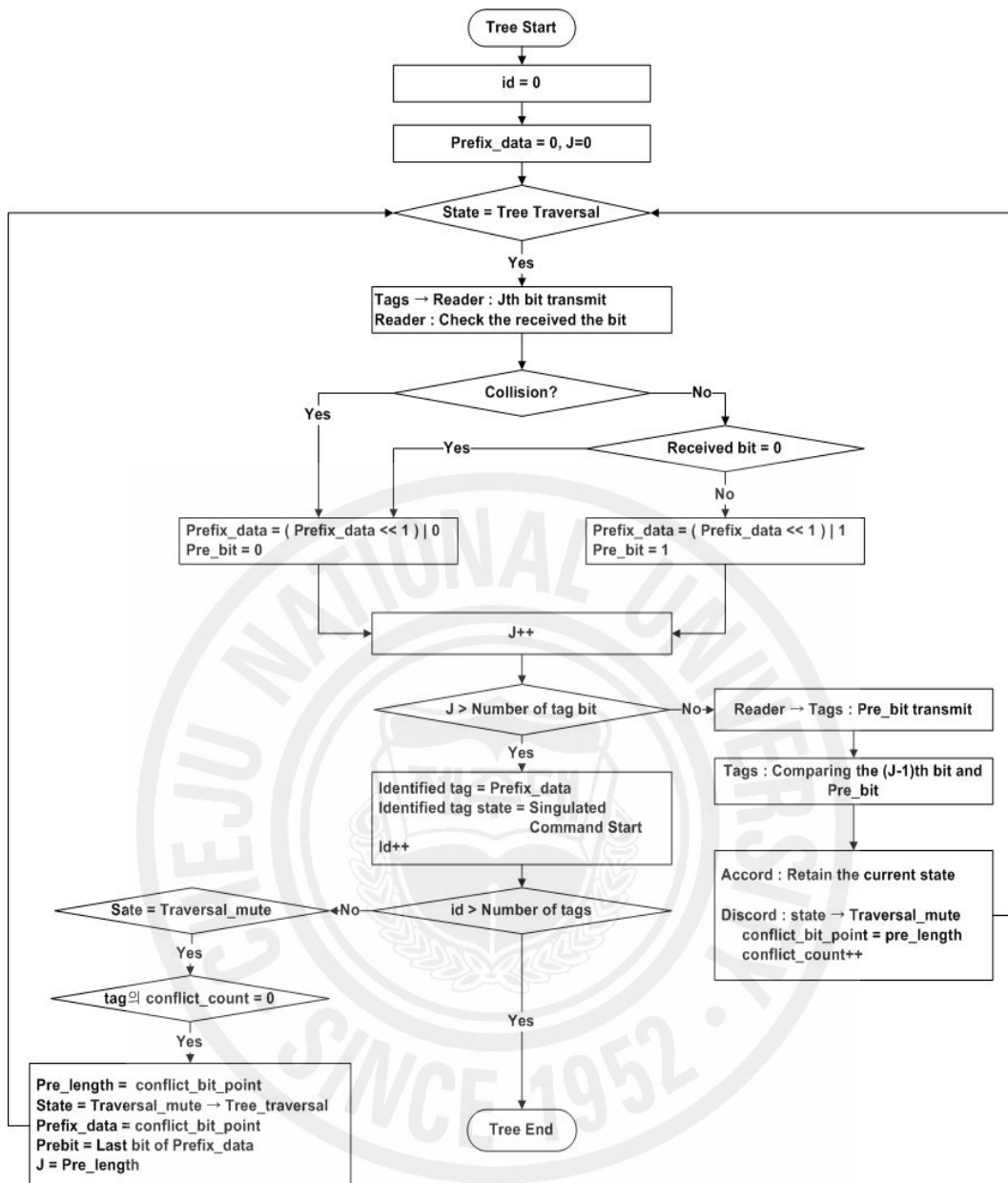


그림 15. 개선된 EPC Class 0 알고리즘의 동작 흐름도

2) 개선된 EPC Class 1 Gen 1 알고리즘

본 논문에서는 기존의 Class 1 충돌방지 알고리즘에서 스택 개념을 도입하고 PingID 명령어에 대한 태그의 응답이 8비트인 점을 착안하여, 충돌이 일어나지 않은 Bin을 우선 인식하고, 충돌이 일어난 Bin은 응답으로 수신된 8비트열들을

서로 비교하여 일치하는 비트열을 스택에 저장하는 방법이다. 하나의 태그를 완벽히 인식한 후, 다시 'VALUE=0, LEN=0'으로 하여 PingID 명령어를 보내는 것이 아니라 우선 스택을 검사하여 저장된 데이터가 있으면 그 값을 [VALUE]와 [LEN]에 저장한 후 PingID 명령어를 보내고, 스택에 저장된 데이터가 없으면 다시 'VALUE=0, LEN=0'으로 하여 PingID 명령어를 보내는 형태이다.

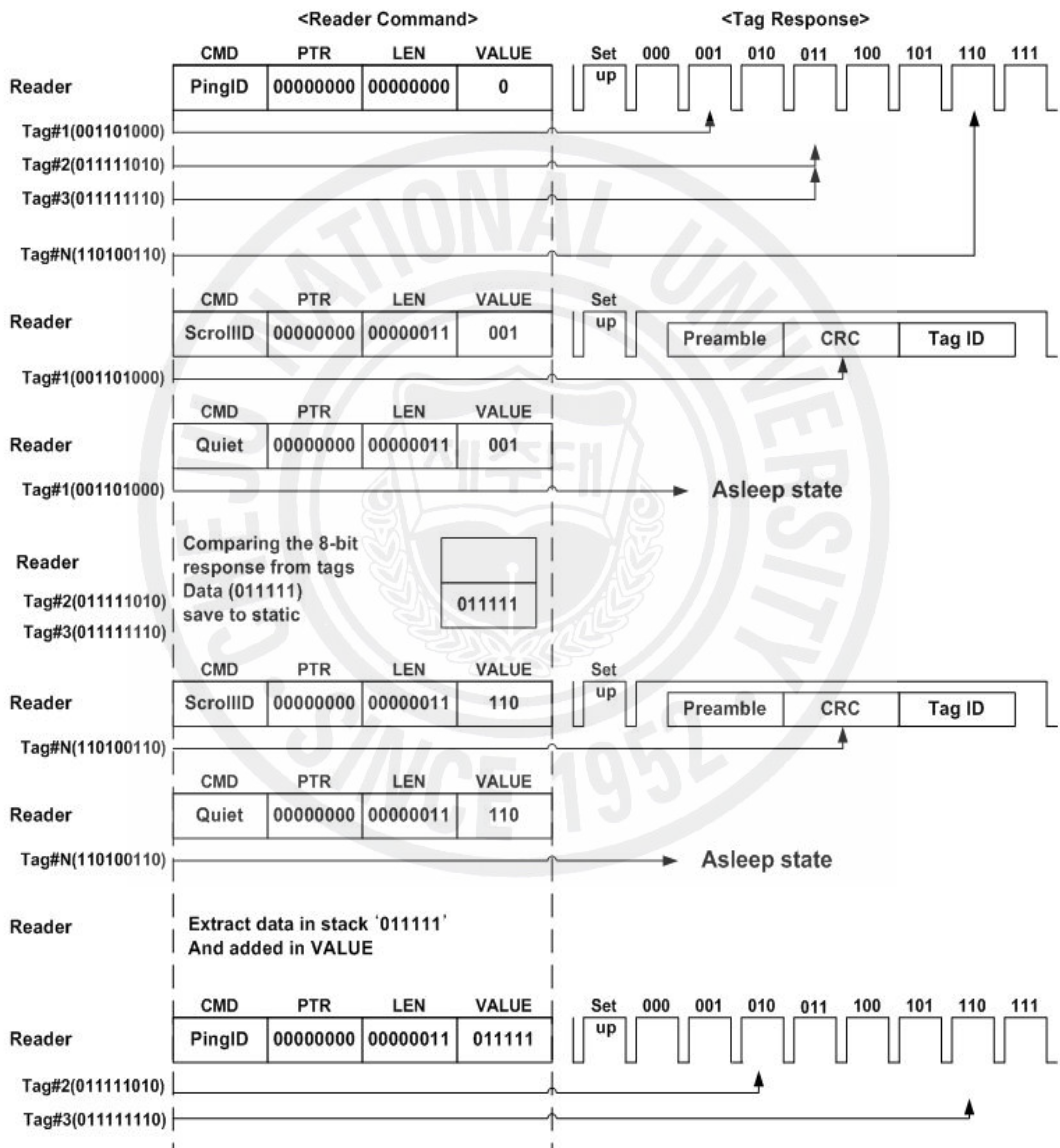


그림 16. 개선된 EPC Class 1 Gen 1 알고리즘의 태그 인식 과정 예시

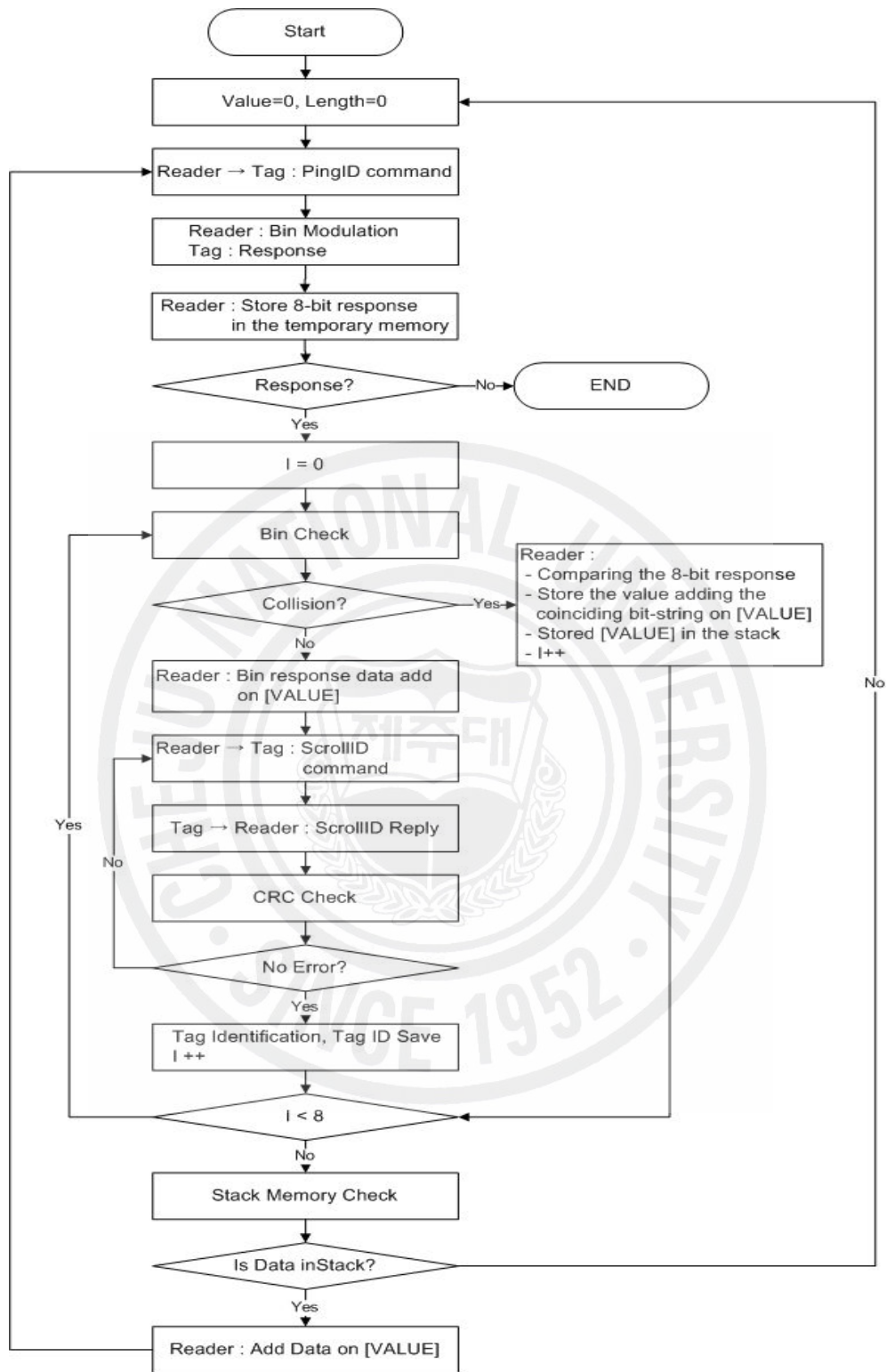


그림 17. 개선된 EPC Class 1 Gen 1 알고리즘의 동작 흐름도

그림 16과 그림 17은 개선된 EPC Class 1 Gen 1 알고리즘의 태그 인식 과정을 나타낸 것이다. Class 1 Gen 1 에서 태그 인식은 리더가 'VALUE=0, LEN=0'을 포함하는 PingID 명령어를 인식영역 내의 임의의 태그에 전송함으로써 시작된다. PingID 명령어를 전송한 후 리더는 Bin Modulation 작업을 하고 태그는 자신의 비트열과 Bin이 일치하는 각 Bin에서 자신의 ID중 8비트를 응답으로 전송한다. 이때 리더는 각 Bin에서의 태그의 응답을 임시 메모리에 저장한다. 리더는 첫 Bin(000)부터 마지막 Bin(111) 까지 순서대로 Bin 검사를 한다. 프로토콜에서는 앞의 Bin에서 태그의 응답이 있으면 그 이후의 Bin에 대해서는 처리를 하지 않는 반면, 개선된 알고리즘에서는 임시메모리를 사용하여 각 Bin에서의 응답을 저장했기 때문에 8개의 Bin에서의 태그 응답을 처리할 수 있다. 각 Bin에서 충돌 여부를 판단하고 충돌이 일어나지 않으면, 리더는 Bin 값을 [VALUE]에 포함한 후 ScrollID 명령어를 태그에 보내게 된다. [VALUE] 값과 자신의 비트열이 일치하는 태그는 자신의 전체 ID를 ScrollID 명령어에 대한 응답으로 리더에 보내게 되고, 리더는 CRC체크를 하여 오류 없이 전송 되었다면 태그 식별 코드를 저장한다. 충돌이 발생하면, 리더는 응답으로 들어온 8비트열들을 비교하고, 일치하는 비트열까지 스택에 저장한다. 그리고 난 후 다음 충돌이 없는 Bin에서의 태그를 먼저 인식한다. 8개의 Bin 검사가 끝난 후, 리더는 스택을 검사하고, 스택에 저장된 데이터가 있으면 그 데이터를 [VALUE]에 포함한 PingID 명령어를 전송한다. 만약 스택에 저장된 데이터가 없다면, 리더는 'VALUE=0, LEN=0' 인 PingID 명령어를 전송한다.

3) 개선된 EPC Class 1 Gen2 알고리즘

기존의 Class 1 Gen 2 충돌방지 알고리즘에서는 태그의 수가 슬롯 수보다 많아 충돌이 많이 발생하거나, 태그의 수가 슬롯 수보다 적어 슬롯의 낭비가 있을 경우 리더는 Query_Adjust 명령어를 사용하여 Q 값을 1씩 증감하여 슬롯의 개수를 조절한다. 본 논문에서는 Q 값을 1씩 증감시키지 않고, 태그 수를 추정하여 그에 맞는 Q 값을 사용한다. 그리고 Class 1 Gen 2에서 충돌이 발생한 슬롯의 시간이 응답이 없는 idle 슬롯의 시간보다 길기 때문에 충돌이 발생하는 슬롯을 줄이기 위하여 예측한 태그 값보다 큰 Q 값을 사용하였다. 예를 들어 예측한 태

그 수가 72일 경우 Q 값을 6과 7중에 7을 선택 하여 슬롯 개수를 128개로 하여 충돌을 최소화하는 방법을 사용하였다.

그림 18은 Class 1 Gen 2의 동작 과정과 본 논문에서 사용된 Q-선택 과정을 나타낸 그림이다. Query나 Query_Adjust 명령어에 대한 한 라운드가 종료된 후에 다음 라운드의 사이즈를 결정하게 된다. 이때 충돌이 발생한 슬롯 수를 세어 현재 남아 있는 태그 수를 예측 한다. i 번째 라운드에서 프레임 사이즈가 $N(i)$, 태그의 개수가 $n(i)$, 충돌이 발생한 슬롯 수 C_{slot} , 성공한 슬롯 수 S_{slot} , 빈 슬롯 수는 E_{slot} 라고 한다. 충돌 슬롯이 발생할 확률, 성공 슬롯이 발생할 확률, 빈 슬롯이 발생할 확률은 모두 이항분포(binominal distribution)를 따른다^[28].

$$P_q(i) = \binom{n(i)}{q} \left(\frac{1}{N(i)} \right)^q \left(\frac{N(i)-1}{N(i)} \right)^{n(i)-q} \quad (1)$$

여기서, $q=0$ 일 때 $P_0(i)$ 는 태그의 응답이 없는 빈 슬롯의 확률, $q=1$ 일 때 $P_1(i)$ 은 성공적으로 전송될 확률, 나머지($q \geq 2$)는 충돌이 발생할 확률이다. 따라서 충돌이 발생할 확률은 다음과 같다.

$$\sum_{q=2}^N P_q(i) = 1 - P_0(i) - P_1(i) \quad (2)$$

현재 i 번째 라운드에서 예측된 태그 수는 성공한 슬롯 수와 충돌이 발생한 슬롯 수를 사용하여 다음과 같이 쓸 수 있다.

$$n_{est} = (P_1(i) \times N(i)) + \left(\sum_{q=2}^N q P_q(i) \right) \times C_{slot} = S_{slot} + K \times C_{slot} \quad (3)$$

$$K = \lim_{N \rightarrow \infty} \sum_{q=2}^N q P_q(i) = 2.39 \quad (4)$$

따라서 다음 라운드에 참여하는 태그 수는 충돌이 발생한 태그이므로, 다음 라운드 크기는 다음과 같다.

$$N(i+1) = H \times (K \times C_{slot}) \quad , \quad \begin{cases} H = 1 + \frac{1}{K}, & \text{Static frame ALOHA} \\ H = 1 - \frac{1}{K}, & \text{Dynamic frame ALOHA} \end{cases} \quad (5)$$

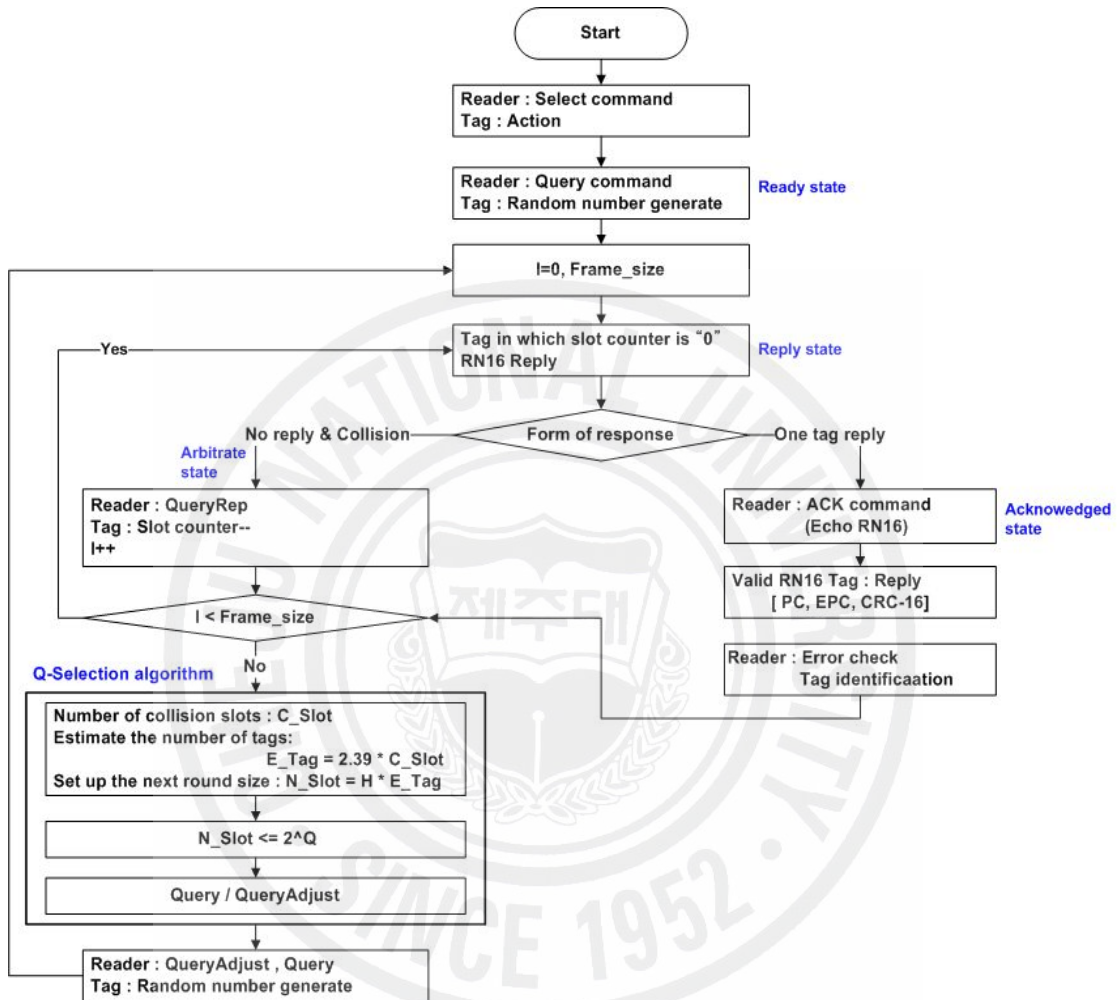


그림 18. 개선된 EPC Class 1 Gen 2 알고리즘의 흐름도

위의 동작과정을 자세히 설명하면 다음과 같다. 태그 ID를 인식하는 INVENTORY 과정에 앞서 통신할 특정 태그를 선택하는 Select 명령어를 전송한다. 리더는 프레임의 슬롯 개수를 초기화하고, 태그가 프레임에 참여하게 하는 Query 명령어를 전송한다. 이 명령어를 수신한 태그는 자신의 상태를 Ready 상태로 전환하고, Query 명령어에 포함된 Q-파라미터를 사용하여 $(0 \sim 2^Q - 1)$ 범

위에서 자신의 슬롯번호(RN16)를 선택하고 이를 슬롯카운터로 로드한다. 여기서 '0'을 선택한 태그들은 Reply 상태로 전환되고, 전환되는 즉시 리더로 자신의 RN16을 응답 메시지를 전송한다. 그 밖의 '0'이 아닌 랜덤번호를 선택한 태그들은 Arbitrate 상태로 전환되고 QueryAdjust 또는 QueryRep 명령어를 기다린다. 만약 그 응답유형이 하나의 태그만 응답 하였다면, 리더는 수신한 RN16을 포함하는 ACK 명령어를 태그로 전송한다. ACK 명령어에 포함된 RN16과 자신의 RN16이 일치하는 태그는 Acknowledge 상태로 전환되고, [PC, EPC, CRC]를 응답으로 리더에 전송한다. 만약 그 응답유형이 충돌이거나 응답이 없다면, 리더는 QueryRep 명령어를 전송하고, 이 명령어를 수신한 태그는 자신의 슬롯카운터 값을 1씩 감소하고, 다시 이 값이 '0'인 태그는 Reply 상태로 전환되어 자신의 RN16을 응답으로 전송한다. 이 과정을 프레임 사이즈만큼 반복하게 되고, 만약 하나의 프레임이 종료되면, 리더는 인식영역 내에 인식되지 않고 남아 있는 태그의 개수를 추정한다. 식 (3)과 식 (5)과 같이 충돌이 발생한 슬롯 수를 사용하여 태그의 개수를 추정하고, 충돌을 최소화하기 위하여 예측한 태그 값보다 큰 Q 값을 설정한다. 재설정된 Q 값을 포함하는 QueryAdjust 명령어를 전송하여 위의 과정을 반복한다.

IV. 시뮬레이션 및 구현

1. 시뮬레이션 결과

본 논문은 UHF 대역 RFID 프로토콜인 ISO 18000-6 Type A/B, EPC Class 0, EPC Class1 Gen 1, EPC Class1 Gen 2의 충돌방지 알고리즘의 성능을 개선하고 분석하였다. 충돌방지 알고리즘은 C언어를 이용하여 구현하였고, 태그 비트 수는 64비트로 가정하며, 태그 수는 최소 10개에서 최대 1000개에 대해 시뮬레이션을 수행하였다.

1) ISO 18000-6 Type A와 B의 시뮬레이션 결과

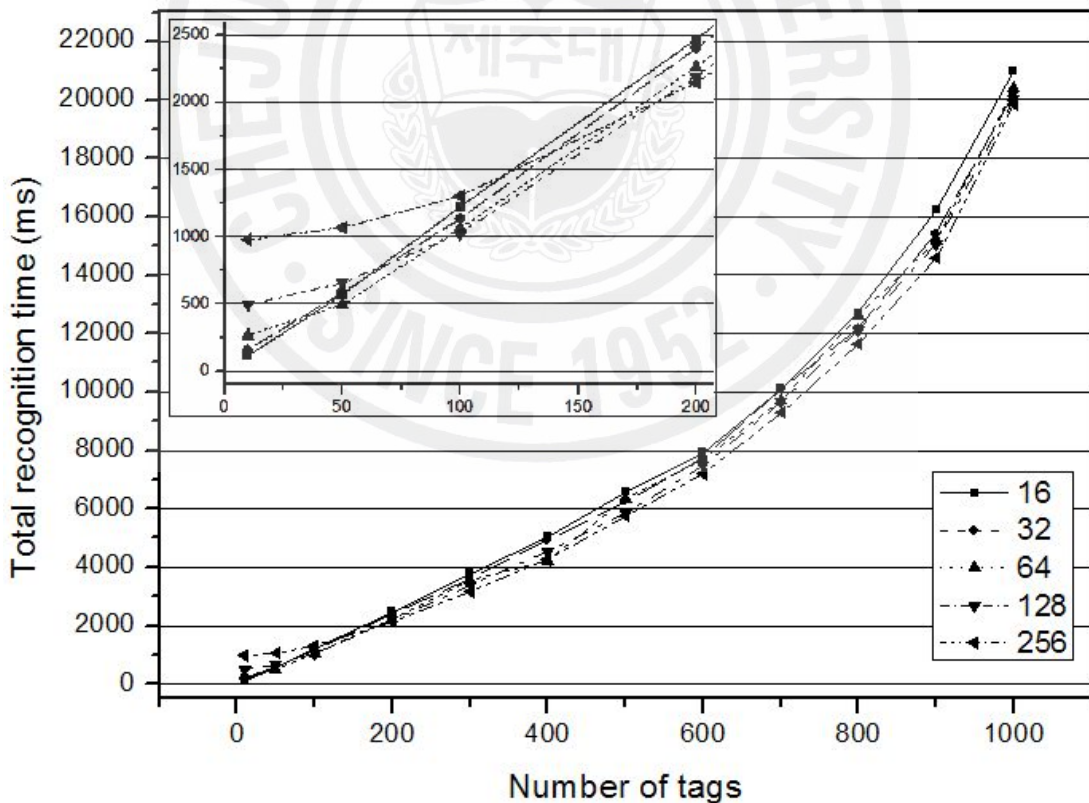


그림 19. ISO 18000-6 Type A 표준 충돌방지 알고리즘의 초기 슬롯 수에 따른 총 태그 인식시간

표 1. ISO 18000-6 Type A 표준 충돌방지 알고리즘의 초기 슬롯 수에 따른 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)				
	초기 슬롯 수				
	16	32	64	128	256
10	116.9	156.1	267.6	498.6	981.0
50	572.9	587.8	496.2	658.3	1069.4
100	1223.3	1135.1	1058.8	1025.2	1305.1
200	2468.7	2404.4	2262.0	2194.5	2146.2
300	3787.1	3612.6	3545.4	3381.7	3150.9
400	5032.9	4917.3	4189.4	4565.5	4256.5
500	6578.3	6247.1	6346.7	5864.7	5756.2
600	7928.2	7739.9	7723.5	7460.2	7200.2
700	10104.9	10086.3	9728.3	9673.2	9303.4
800	12711.3	12166.2	12619.5	12098.1	11632.9
900	16238.4	15419.3	15152.8	15085.9	14579.3
1000	20986.5	20223.3	20428.6	20007.6	19858.7

ISO 18000-6 Type A/B의 링크타이밍을 나타낸 그림 8과 12의 시간 규정을 적용하여 태그 인식 시간을 계산하였다. Type A인 경우 초기 슬롯 수는 사용자가 지정할 수 있기 때문에 본 논문에서는 슬롯 수로 지정할 수 있는 8, 16, 32, 64, 128, 256을 초기 슬롯 개수로 지정하여 태그 인식 시간을 계산하였다. 표 1과 그림 19는 ISO 18000-6 Type A 충돌방지 알고리즘의 초기 슬롯 수에 따른 총 태그 인식 시간을 나타낸 것이다. 시뮬레이션 결과 태그의 개수와 슬롯 수가 같을 때 가장 좋은 성능을 나타낸다.

표 2와 그림 20은 Type A와 B의 표준 충돌방지 알고리즘과 성능을 개선한 충돌방지 알고리즘의 총 태그 인식 시간을 나타낸 것이다. Type A에서 초기 슬롯 수가 16개인 경우 태그 수가 100개 일 때 개선된 충돌방지 알고리즘은 표준 알고리즘보다 태그 인식 시간이 35.2%, 태그 수가 1000개 일 때 24.3% 감소하였다. Type B에서 개선된 충돌방지 알고리즘은 표준 알고리즘보다 태그 인식 시간이 태그 수 100개 일 때 6.0%, 태그 수 1000개 일 때 6.6% 감소하였다.

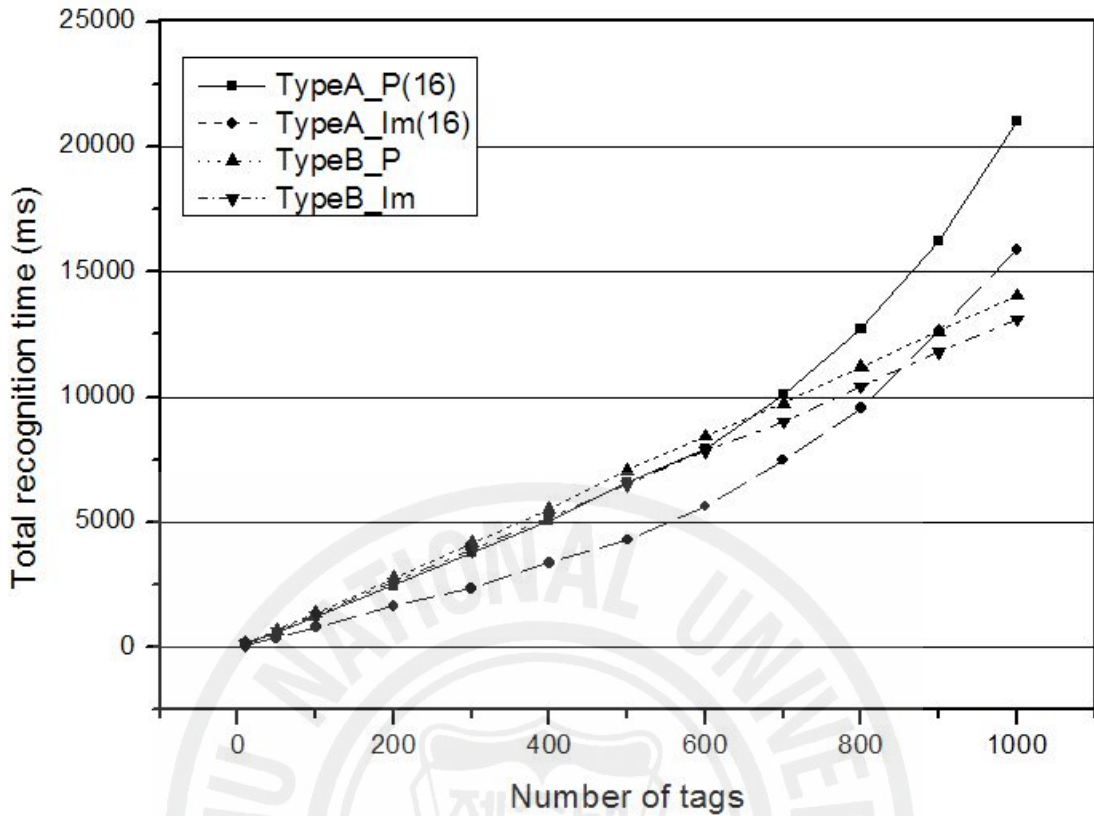


그림 20. ISO 18000-6 Type A/B 충돌방지 알고리즘의 총 태그 인식 시간

표 2. ISO 18000-6 Type A/B 충돌방지 알고리즘의 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)			
	초기 슬롯 수(16)		Type B 표준	Type B 개선
	Type A 표준	Type A 개선		
10	116.9	54.9	139.6	119.0
50	572.9	371.6	688.6	642.9
100	1223.3	793.4	1363.5	1281.5
200	2468.7	1651.3	2773.0	2629.1
300	3787.1	2360.8	4155.0	3887.8
400	5032.9	3374.0	5508.5	5204.2
500	6578.3	4290.0	7070.7	6528.4
600	7928.2	5621.0	8422.9	7856.2
700	10104.9	7487.4	9755.8	9027.1
800	12711.3	9542.8	11207.6	10431.5
900	16238.4	12629.3	12628.9	11804.1
1000	20986.5	15877.5	14044.1	13111.3

2) EPC Class 0, Class 1 Gen 1, Class 1 Gen 2의 시뮬레이션 결과

표 3은 Class 0의 충돌방지 알고리즘에서 태그 당 리더로 전송되는 평균 비트 수를 나타낸 것이다. Class 0의 충돌방지 알고리즘은 이진트리 방식으로 리더와 태그 간에 통신 설정이 이루어진 후 태그 ID 인식은 비트 단위로 이루어진다. 리더와 태그 간 1비트 전송시간은 $12.5\mu s$ 로 규정되어 있다. Class 0 알고리즘의 시뮬레이션 결과는 다음과 같다.

표 3. EPC Class 0 충돌방지 알고리즘의 태그 당 전송 평균 비트 수

태그 수	태그 당 전송 평균 비트수	
	Class 0 표준	Class 0 개선
10		77.7
50		75.5
100		74.5
200		73.5
300		72.9
400		72.5
500	81	72.2
600		71.9
700		71.7
800		71.5
900		71.3
1000		71.1

Class 0의 총 인식시간은 마스터 리셋 시간(T_M), 오실레이터 켈리브레이션 시간(T_{OC}), 데이터 켈리브레이션 시간(T_{DC}), 태그 ID 인식 시간(T_{ID})을 포함한다. 태그의 총 인식시간(T_R)을 구하는 식은 다음과 같다.

$$\begin{aligned}
 T_R &= T_M + T_{OC} + T_{DC} + T_{ID} \\
 &= 800 + 116 + 38.5 + (12.5 \times T_B \times T_{Num})
 \end{aligned}
 \tag{6}$$

여기서, T_B 는 태그와 리더 간 전송되는 비트 수이고, T_{Num} 는 총 태그 수이다.

$$T_B = EPC + CRC + Last\ bit = 64 + 16 + 1 = 81
 \tag{7}$$

표 3에서 개선된 Class 0 충돌방지 알고리즘은 태그 수가 증가 할수록 태그 당 전송되는 평균 비트수가 감소함을 알 수 있다.

표 4는 Class 1 Gen 1 충돌방지 알고리즘에서 리더가 태그로 전송하는 총 명령어 수와 명령어 포맷에 포함되는 [VALUE]필드의 길이를 나타낸 것이다. 이를 이용하여 총 인식 시간은 식 (8)과 (9)를 이용하여 구한다.

표 4. EPC Class 1 Gen 1 충돌방지 알고리즘의 총 명령어 수와 [VALUE]필드의 길이

태그 수	Class 1 Gen 1 표준			Class 1 Gen 1 개선		
	Total command	VLP	VLS	Total command	VLP	VLS
10	37.0	1.41	5.10	24.8	3.50	6.90
50	223.0	2.52	7.38	121.2	6.08	9.06
100	482.8	3.07	8.48	242.4	6.94	10.03
200	1034.6	3.58	9.52	487.6	8.14	11.18
300	1611.0	3.87	10.11	729.4	8.83	11.77
400	2197.0	4.04	10.48	977.8	9.25	12.26
500	2807.6	4.21	10.85	1222.0	9.54	12.58
600	3424.0	4.27	11.12	1427.2	9.77	12.84
700	4011.0	4.51	11.40	1715.2	10.00	13.11
800	4676.0	4.54	11.54	1956.4	10.14	13.24
900	4322.4	4.64	11.73	2198.4	10.34	13.43
1000	5950.2	4.68	11.85	2437.2	10.45	13.52

$$C_{Total} = N_P + N_S + N_Q = N_P + 2 \times N_{Tags} \quad (8)$$

여기서 N_P , N_S , N_Q 는 태그 인식에 필요한 PingID, ScrollID, Quiet 명령어 수이다. 만약 PingID 명령어, ScrollID 명령어, Quiet 명령어 1개당 평균 [VALUE]길이를 각각 VLP , VLS , $VLQ(=VLS)$, 각 명령어 1개당 전송 시간을 T_{P1} , T_{S1} , T_{Q1} 이라하면, 태그의 총 인식시간 T_{total} 은 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 T_{P1} &= 1.25T_0 + 64\mu s + 51T_0 + [VLP]T_0 + 8T_0 + 64T_0 + 2.5T_0 \\
 T_{S1} &= 1.25T_0 + 64\mu s + 51T_0 + [VLS]T_0 + 8T_0 + [tagbit]T_0 + 2.5T_0 \\
 T_{Q1} &= 1.25T_0 + 64\mu s + 51T_0 + [VLS]T_0 + 8T_0 + 2.5T_0 \\
 T_{Total} &= N_P T_{P1} + N_S T_{S1} + N_Q T_{Q1} \quad (9)
 \end{aligned}$$

태그의 총 인식 시간은 그림 4의 시간 규정을 적용하여 계산하였고, T_0 는 복미 방식에서 규정된 $14.25\mu s$ 을 기준으로 계산하였다. 리더에서 태그로 전송하는 신호는 첫 번째 단계 transaction gap $1.25T_0$, 두 번째 단계 CW RF신호는 $64\mu s$, 세 번째 단계 data modulation 부분, 네 번째 단계 TAG Setup $8T_0$, 마지막 단계인 태그 응답 전송부분은 $64T_0$, 다음 transaction gap과의 인터벌 $2.5T_0$ 이다. 세 번째 단계인 data modulation부분은 리더 명령어 포맷에서 [VALUE] 길이를 제외한 51비트에 표 4에 주어진 VLP, VLS을 더하여 계산하였다.

그림 21과 표 5는 Class 1 Gen 2의 충돌방지 알고리즘의 태그 인식 시간 계산을 위한 각 명령어와 태그 응답에 대한 타이밍을 규정한 것이다. 이 링크타이밍을 적용하여 Class 1 Gen 2 알고리즘의 태그 인식 시간을 계산하였다.

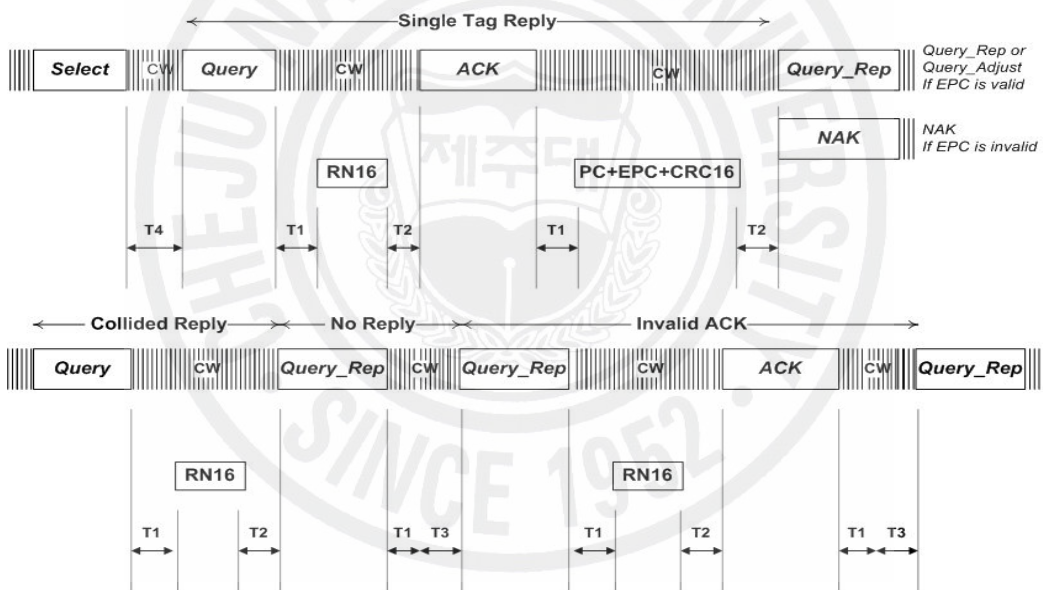


그림 21. EPC Class 1 Gen 2의 규정된 링크타이밍

표 5. EPC Class 1 Gen 2의 규정된 링크타이밍

	Minimum	Typical	Maximum
T_1	$MAX(RT_{cal}, 10T_{pri}) * (1 - FT) - 2\mu s$	$MAX(RT_{cal}, 10T_{pri})$	$MAX(RT_{cal}, 10T_{pri}) * (1 - FT) + 2\mu s$
T_2	$3.0T_{pri}$		$20.0T_{pri}$
T_3	$0.0T_{pri}$		
T_4	$2.0T_{pri}$		

Class 1 Gen 2 프로토콜에서 가능한 슬롯 수($2^Q, 1 \leq Q \leq 15$)가 정해져 있기 때문에 예측한 태그 수와 정확하게 동일한 슬롯 수를 다음 라운드에 사용할 수 없다. 따라서 본 논문에서는 Class 1 Gen 2에서 태그 수를 추정하여 다음 라운드의 슬롯 크기를 할당할 때 어떤 방식을 사용하는 지에 따른 태그 인식시간을 비교하였다. 표 6과 그림 22는 그 결과는 나타낸 것이다. Est 1은 표준 프로토콜에서 규정된 방식으로 태그 수가 슬롯 수보다 많아서 충돌이 많이 발생한 경우, 또는 태그 수가 슬롯 수보다 많아서 빈 슬롯 수가 많이 발생한 경우 Q의 변환을 1씩 증감하는 방식이다. Est 2는 예측한 태그 수와 가장 가까운 값으로 슬롯 수를 변환하는 방법이고, Est 3은 작은 값으로, Est 4는 본 논문에서 제안하는 방식으로 예측한 태그 수보다 큰 값으로 변환하는 방법이다. 개선된 Est 4 방식은 표준의 Est 1 알고리즘보다 태그 인식 시간이 태그 수 100개 일 때 7.9%, 태그 수가 1000개 일 때 11.7% 감소하였다.

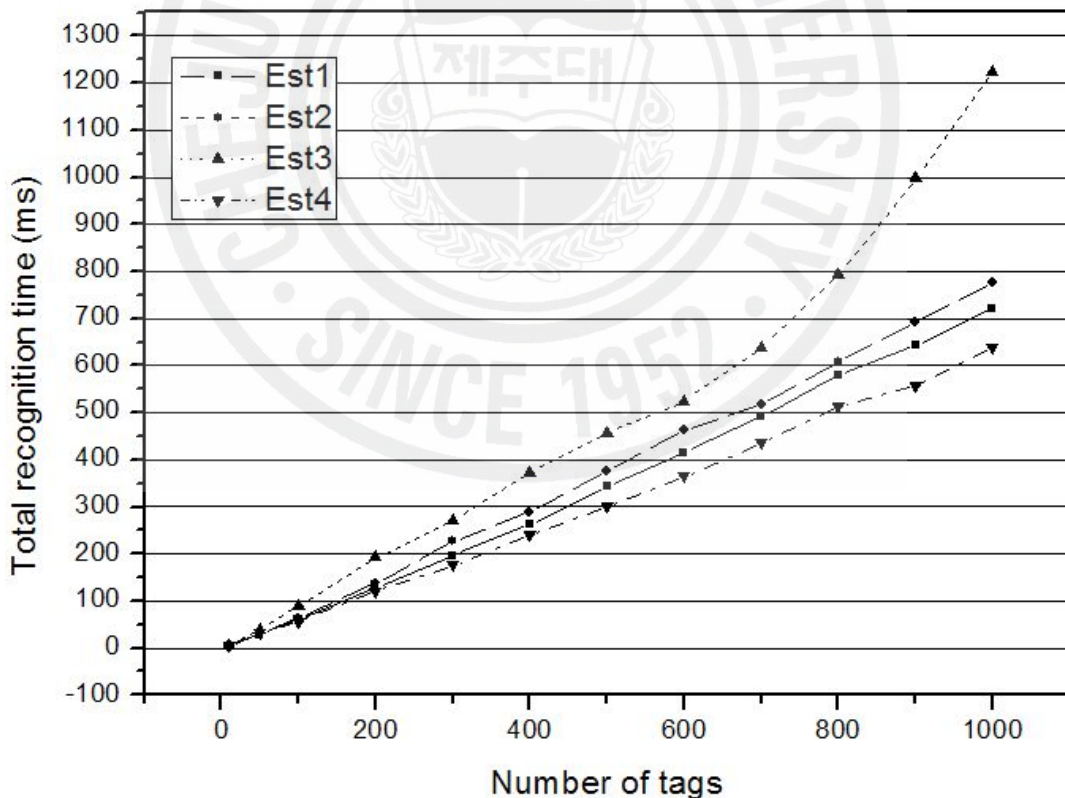


그림 22. EPC Class 1 Gen 2 충돌방지 알고리즘의 Q-확장 방법에 따른 총 태그 인식 시간

표 6. EPC Class 1 Gen 2 충돌방지 알고리즘의 Q-확장 방법에 따른 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)			
	Est 1(표준)	Est 2(실험)	Est 3(실험)	Est 4(개선)
10	5.5	5.6	5.6	5.6
50	28.8	30.3	40.3	31.4
100	61.7	64.8	89.0	56.8
200	129.4	138.2	192.5	121.3
300	197.1	227.1	271.2	175.8
400	262.5	289.5	372.8	240.7
500	342.2	374.5	556.4	301.4
600	415.6	463.0	524.4	364.1
700	491.2	517.2	637.8	436.7
800	549.9	607.0	793.7	514.1
900	542.7	692.6	997.4	557.7
1000	722.2	776.7	1222.7	637.5

표 7과 그림 23은 EPC 프로토콜의 각 알고리즘 별 총 태그 인식시간을 나타낸 것이다. 시뮬레이션 결과 태그 수 100개 일 때 개선된 Class 0 알고리즘은 표준 알고리즘보다 태그 인식 시간이 8%, 태그 수 1000개 일 때 12.2%가 감소하였다. 개선된 Class 1 Gen 1 알고리즘은 표준 알고리즘보다 태그 인식 시간이 태그 수 100개 일 때 54.5%, 태그 수 1000개 일 때 63.4% 감소하였다. 개선된 Class 1 Gen 2 알고리즘은 표준 알고리즘보다 태그 인식 시간이 태그 수 100개 일 때 7.9%, 태그 수 1000개 일 때 11.7% 감소하였다. Class 0 나 Class 1 Gen 2 보다 Class 1 Gen 1 충돌방지 알고리즘의 인식 시간이 큰 이유는 Class 1 Gen 1 알고리즘은 리더와 태그 간에 패킷 단위로 통신을 하기 때문이다.

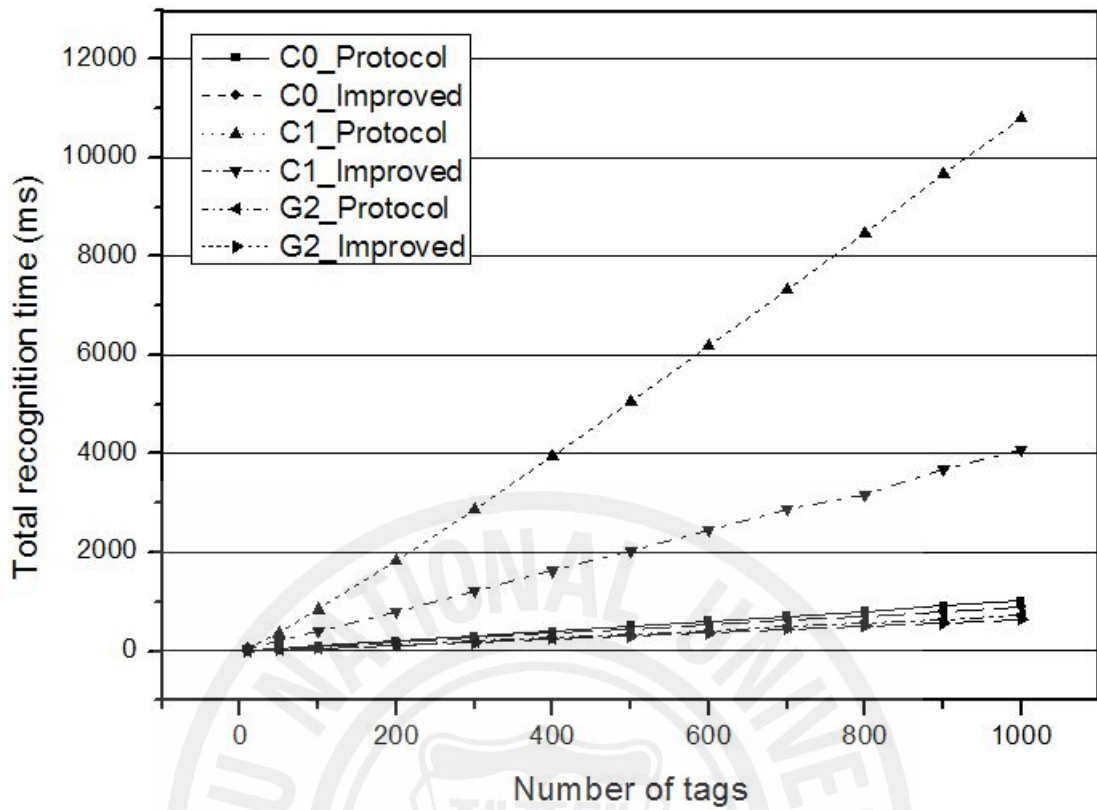


그림 23. EPC 프로토콜 충돌방지 알고리즘의 총 태그 인식 시간

표 7. EPC 프로토콜 충돌방지 알고리즘의 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)					
	Class 0 표준	Class 0 개선	Class1 Gen1 표준	Class1 Gen1 개선	Class1 Gen2 표준	Class1 Gen2 개선
10	11.1	10.7	61.9	39.5	5.5	5.6
50	51.6	48.1	386.4	195.8	28.8	31.4
100	102.2	94.0	848.3	394.9	61.7	56.8
200	203.5	184.7	1839.1	803.4	129.4	121.3
300	304.7	274.4	2881.5	1207.4	197.1	175.8
400	406.0	363.3	3943.9	1627.1	262.5	240.7
500	507.2	452.0	5057.7	2038.8	342.2	301.4
600	608.5	540.2	6181.8	2459.5	415.6	364.1
700	709.7	628.1	7316.2	2875.8	491.2	436.7
800	811.0	715.6	8477.5	3182.6	579.9	514.1
900	912.2	802.6	9666.9	3693.8	542.7	557.7
1000	1013.5	890.0	10817.2	4096.5	722.2	637.5

2. 결과 및 고찰

본 논문에서는 UHF 대역 RFID 프로토콜인 ISO의 18000-6 Type A와 B, EPCglobal의 Class 0, Class 1 Gen 1, Class 1 Gen 2의 에어인터페이스를 분석하고, 각 프로토콜에서 규정된 충돌방지 알고리즘을 C언어로 구현하였다. 이를 바탕으로 다중태그 인식에 사용하는 충돌방지 알고리즘의 성능 개선 방안을 제시하고, C언어로 구현하여 그 성능을 표준 충돌방지 알고리즘과 비교하여 분석하였다. 이 장에서는 프로토콜로 규정된 충돌방지 알고리즘의 태그 인식 시간 성능을 서로 비교한다. 그리고 프로토콜로 규정된 클럭 주기를 동일하게 하여 통신 특성에 따라 성능이 어떻게 달라지는지를 분석한다.

표 8과 그림 24는 Type A, Type B, Class 0, Class 1 Gen 2, Class 1 Gen 2의 프로토콜로 규정된 충돌방지 알고리즘의 태그 인식 시간을 비교한 것이다. 시뮬레이션 결과 동일한 슬롯알로하 방식을 사용하는 Class 1 Gen 2와 Type A를 비교해 보았을 때 Class 1 Gen 2의 성능이 Type A 보다 좋음을 알 수 있다. 이는 프로토콜에 규정된 클럭 주기가 Class 1 Gen 2인 경우에는 $12.5\mu s$, Type A인 경우에는 $20\mu s$ 로 더 길고 리더 명령어에 대한 태그의 응답 시에 Class 1 Gen 2는 16비트의 RN16을 전송하고 Type A는 ID와 CRC 전체를 보내 전송비트 수가 많기 때문에 태그 인식 시간에 더 오랜 시간이 소요된다. 그리고 Type A인 경우 태그 인식 시간이 태그 수 400개 이상에서 급격히 증가하는데, 이는 Type A의 경우 확장 가능한 슬롯의 최대 범위가 256개이기 때문이다. 이진트리 방식의 계열을 사용하는 Type B, Class 0, Class 1 Gen 1을 비교해 보았을 때 태그 인식 시간이 Class 0가 가장 짧고 다음에는 Class 1 Gen 1, Type B 순이다. 이는 Class 0는 리더와 태그의 통신이 설정된 후에 1비트 단위로 리더와 태그 간 통신을 하면서 태그 ID를 찾아내는 반면, Class 1 Gen 1과 Type B는 패킷 단위로 통신이 이루어지기 때문에 Class 0의 성능이 좋다. Class 1 Gen 1은 태그 ID를 찾을 때 그 응답으로 ID의 일부인 8비트를 전송하는 반면, Type B는 전체 ID를 리더로 전송하고 Class 1 Gen1의 클럭 주기는 $14.25\mu s$ 이고, Type B는 $20\mu s$ 이기 때문에 Type B의 인식 시간이 가장 길다.

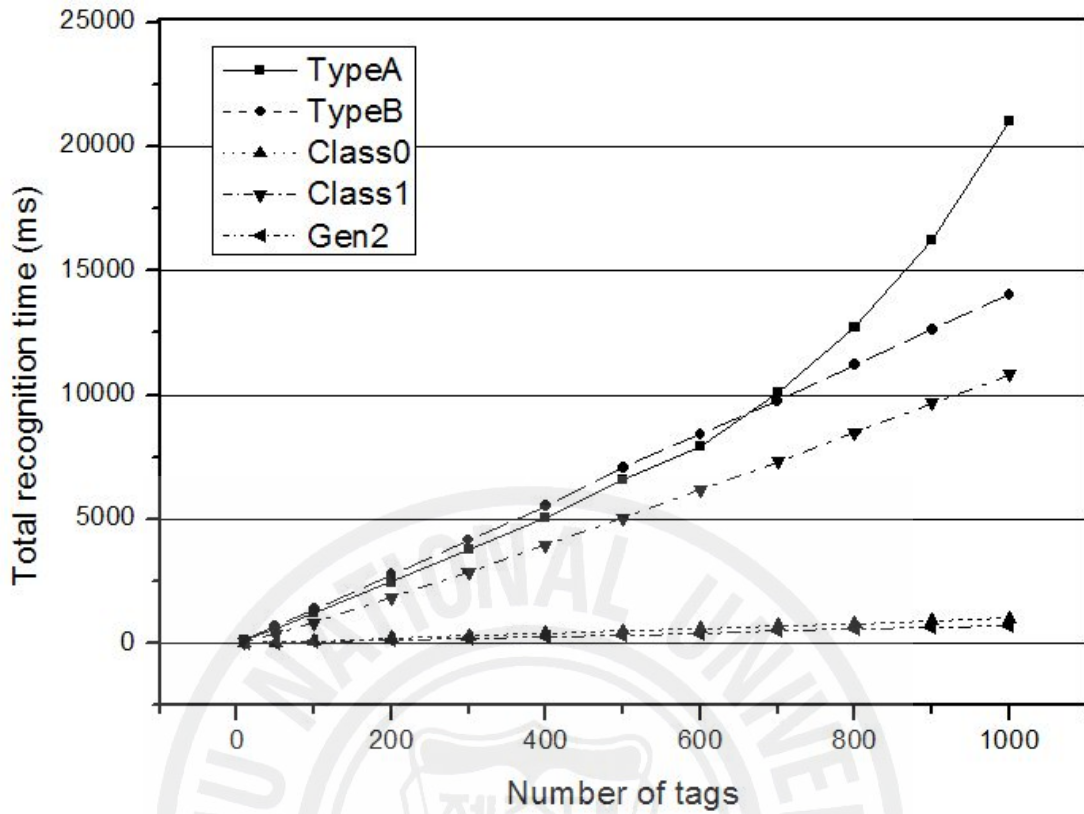


그림 24. 표준 충돌방지 알고리즘의 총 태그 인식 시간

표 8. 표준 충돌방지 알고리즘의 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)				
	Type A	Type B	Class 0	Class1 Gen1	Class1 Gen2
10	116.9	139.6	11.1	91.6	5.5
50	572.9	688.6	51.6	386.4	28.8
100	1223.3	1363.5	102.2	848.3	61.7
200	2468.7	2773.0	203.5	1839.1	129.4
300	3787.1	4155.0	304.7	2881.5	197.1
400	5032.9	5508.5	406.0	3943.9	262.5
500	6578.3	7070.7	507.2	5057.7	342.2
600	7928.2	8422.9	608.5	6181.8	415.6
700	10104.9	9755.8	709.7	7316.2	491.2
800	12711.3	11207.6	811.0	8477.5	579.9
900	16238.4	12628.9	912.2	9666.9	642.7
1000	20986.5	14044.1	1013.5	10817.2	722.2

표 9 와 그림 25는 Type A, Type B, Class 0, Class 1 Gen 1, Class 1 Gen 2 의 개선된 충돌방지 알고리즘의 총 태그 인식 시간을 비교한 것이다.

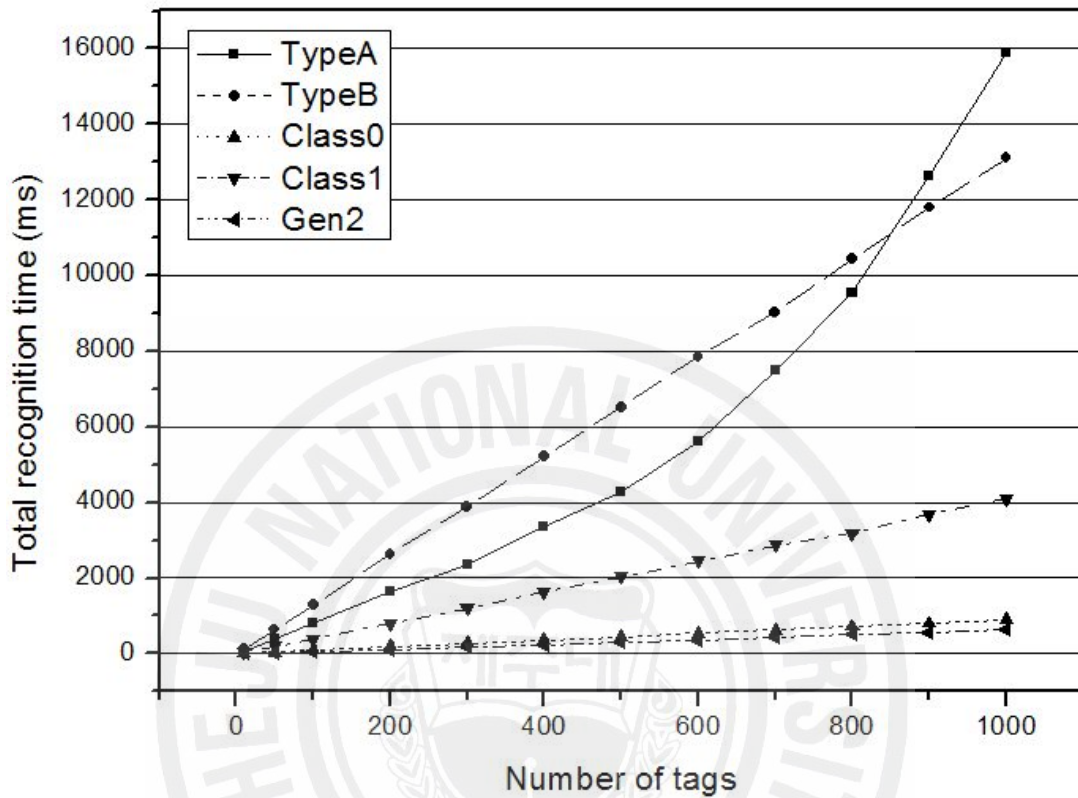


그림 25. 개선된 충돌방지 알고리즘의 총 태그 인식 시간

표 9. 개선된 충돌방지 알고리즘의 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)				
	Type A	Type B	Class 0	Class1 Gen1	Class1 Gen2
10	54.9	119.0	10.7	39.5	5.6
50	371.6	642.9	48.1	195.8	31.4
100	793.4	1281.5	94.0	394.9	56.8
200	1651.3	2629.1	184.7	803.4	121.3
300	2360.8	3887.8	274.4	1207.4	175.8
400	3374.0	5204.2	363.3	1627.1	240.7
500	4290.0	6528.4	452.0	2038.8	301.4
600	5621.0	7856.2	540.2	2459.5	364.1
700	7487.4	9027.1	628.1	2875.8	436.7
800	9542.8	10431.5	715.6	3182.6	514.1
900	12629.3	11804.1	802.6	3693.8	557.7
1000	15877.5	13111.3	890.0	4096.5	637.5

앞의 프로토콜로 규정된 표준 충돌방지 알고리즘의 시뮬레이션과 비교해 보았을 때 태그 수 800개 이하에서 Type A와 Type B의 태그 인식 시간 성능이 표준보다 차이가 크을 볼 수 있다. 이는 빈 슬롯 수가 800개 이하에서는 Type B보다 Type A가 더 많아 그 시간을 더 단축할 수 있었기 때문이다.

표 10과 그림 26은 리더와 태그 간 클럭 주기(T_0)를 $12.5\mu s$ 로 동일하게 하여 표준 충돌방지 알고리즘의 총 태그 인식 시간을 비교한 것이고, 표 11과 그림 27은 리더와 태그 간 클럭 주기를 $12.5\mu s$ 로 동일하게 하여 개선된 충돌방지 알고리즘의 총 태그 인식 시간을 비교한 것이다. 그림 24와 비교해보면, Class 0과 Class 1 Gen 2의 태그 인식 시간 성능 순위는 앞의 시뮬레이션 결과와 차이가 없지만, Type A와 B, Class 1 Gen 1의 성능이 차이를 보였다. 이는 Type A와 B의 프로토콜로 규정된 클럭 주기가 $20\mu s$ 인데 반해 Class 1 Gen 1은 $14.25\mu s$ 이기 때문에 Type A와 B의 태그인식 시간이 더 감소되었기 때문이다. 이로써 프로토콜로 규정된 클럭주기에 따라 태그인식 성능이 좌우됨을 알 수 있다.

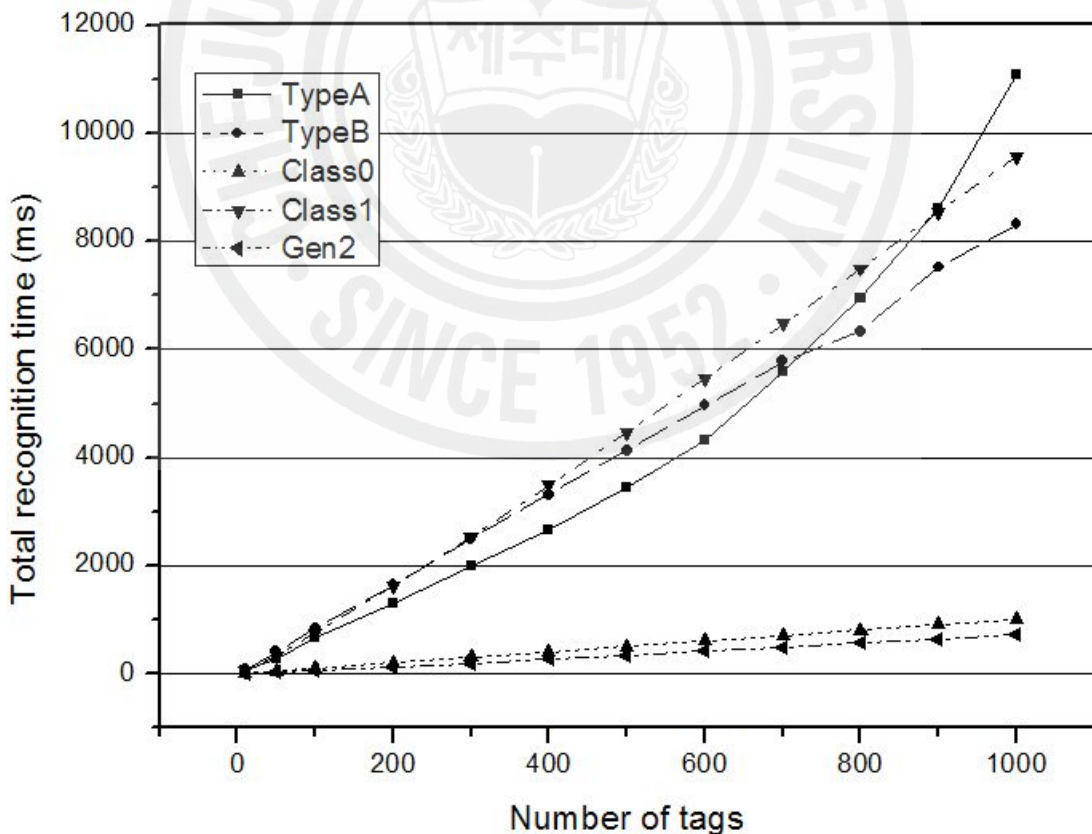


그림 26. T_0 을 $12.5\mu s$ 로 동일하게 한 경우 표준 충돌방지 알고리즘의 총 태그 인식 시간

표 10. T_0 을 $12.5\mu s$ 로 동일하게 한 경우 표준 충돌방지 알고리즘의 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)				
	Type A	Type B	Class 0	Class1 Gen1	Class1 Gen2
10	60.3	78.1	11.1	54.6	5.5
50	290.5	407.2	51.6	341.2	28.8
100	662.4	836.9	102.2	749.4	61.7
200	1315.5	1643.8	203.5	1625.4	129.4
300	1987.7	2501.6	304.7	2547.1	197.1
400	2650.8	3311.4	406.0	3486.7	262.5
500	3458.6	4135.7	507.2	4471.9	342.2
600	4326.1	4970.4	608.5	5466.2	415.6
700	5580.4	5779.3	709.7	6470.4	491.2
800	6940.4	6336.0	811.0	7497.6	579.9
900	8592.7	7518.5	912.2	8550.2	642.7
1000	11085.4	8321.1	1013.5	9567.9	722.2

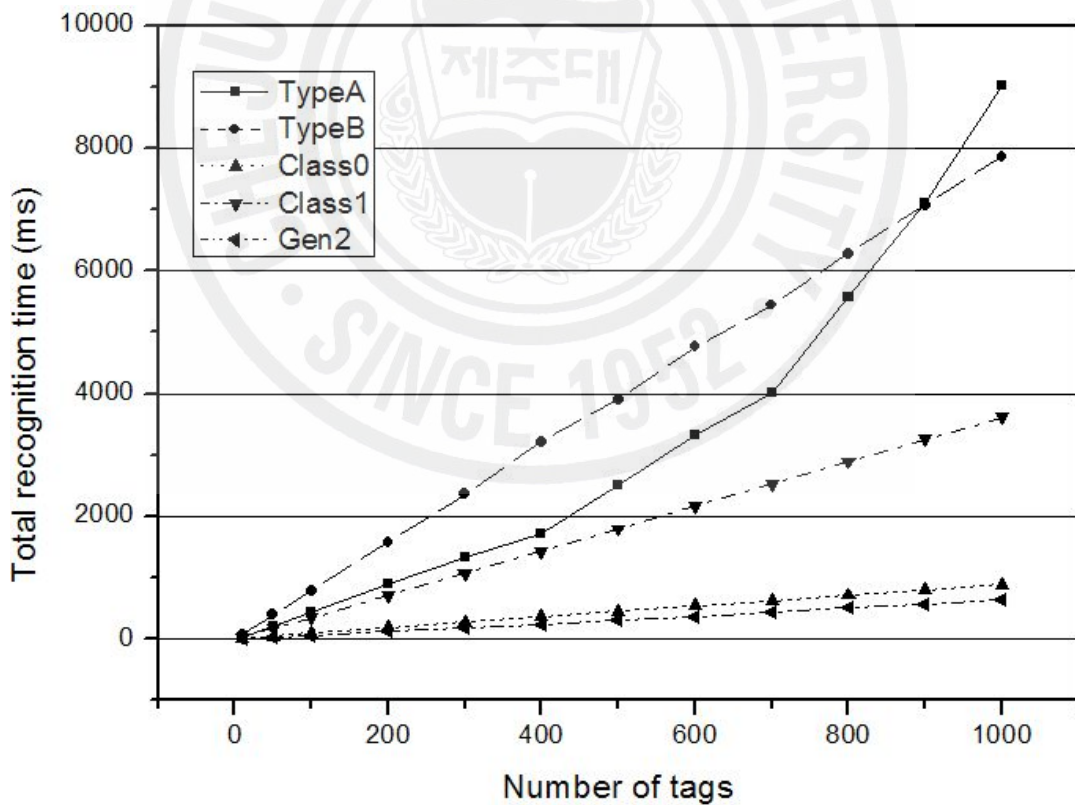


그림 27. T_0 을 $12.5\mu s$ 로 동일하게 한 경우 개선된 충돌방지 알고리즘의 총 태그 인식 시간

표 11. T_0 을 $12.5\mu s$ 로 동일하게 한 경우 개선된 충돌방지 알고리즘의 총 태그 인식 시간

태그 수	총 태그 인식 시간 (ms)				
	Type A	Type B	Class 0	Class1 Gen1	Class1Gen2
10	33.2	74.0	10.7	34.8	5.6
50	204.2	392.2	48.1	172.7	31.4
100	432.8	785.4	94.1	348.3	56.8
200	896.9	1581.7	184.7	708.6	121.3
300	1335.3	2361.7	274.4	1064.9	175.8
400	1771.0	3207.8	363.3	1434.9	240.7
500	2503.0	3906.9	452.2	1798.0	301.4
600	3324.2	4759.3	540.2	2169.0	364.1
700	4018.5	5441.3	628.1	2536.1	436.7
800	5566.2	6281.3	715.6	2894.9	514.1
900	7112.2	7065.6	802.6	3257.5	557.7
1000	9020.8	7866.4	889.9	3612.5	637.5



V. 결 론

본 논문은 유비쿼터스 센서 네트워크와 텔레매틱스 서비스에 사용되는 RFID 표준 중 UHF 대역 프로토콜인 ISO의 18000-6 Type A와 B, EPCglobal의 Class 0, Class 1 Gen 1, Class 1 Gen 2의 에어인터페이스를 분석하고 각 프로토콜에서 규정된 충돌방지 알고리즘을 구현하였다. 그리고 각 프로토콜에서 규정하는 충돌방지 알고리즘의 성능을 개선한 알고리즘을 제안하고 성능을 비교하였다.

프레임 슬롯 알로하 방식을 사용하는 개선된 Type A 충돌방지 알고리즘은 태그의 응답 필드 중 태그 응답 지연시간과 [Preamble][Flags][Parameters] 부분에서 응답이 있는지 없는지를 판별하여 응답이 경우 슬롯의 시간을 줄여 다중태그 인식 시간의 성능을 개선하였다.

SBT와 유사한 확률적 알로하 방식을 사용하는 개선된 Type B 충돌방지 알고리즘은 Type A와 유사한 방식을 사용하여 태그의 응답 필드 중 태그 응답 지연시간과 [Return Preamble] 부분에서 응답이 있는지 없는지를 판별하여 응답이 경우 슬롯의 시간을 줄여 다중태그 인식 시간의 성능을 개선하였다.

이진트리 방식을 사용하는 개선된 Class 0 알고리즘은 태그에 충돌포인트와 충돌카운터를 추가하여 충돌이 발생할 때마다 충돌포인트를 저장하고, 충돌카운터를 증가한다. 그리고 태그 하나를 완벽히 인식한 뒤 루트노드부터 트리를 시작하지 않고, 충돌카운터를 감소하고 이 값이 0이 되는 태그의 충돌포인트에서 트리를 시작하는 방법이다.

이진트리에 기반을 둔 빈 슬롯 방식을 사용하는 개선된 Class 1 Gen 1 충돌방지 알고리즘은 표준으로 정해진 충돌방지 알고리즘에 스택의 개념을 도입하고, 트리를 고정된 3비트로 확장하는 것이 아니라 PingID 명령어에 대한 태그의 8비트 응답신호를 비교하여 트리확장에서 비트 길이를 최대 8비트까지 동적으로 확장하는 방법이다. PingID 명령어를 전송한 후 리더는 Bin Modulation 작업을 하고, 태그 응답을 검사한다. 만약 Bin에서 충돌이 발생하면, 리더는 응답으로 들어온 8비트열들을 비교하고, 일치하는 비트열까지 스택에 저장한다. 그리고 난 후

다음 충돌이 없는 Bin에서의 태그를 먼저 인식한다. 8개의 Bin 검사가 끝난 후, 리더는 스택을 검사하고, 스택에 저장된 데이터가 있으면 그 데이터를 포함한 PingID 명령어를 전송하는 방식이다.

슬롯 알로하 방식을 사용하는 개선된 Class 1 Gen 2 충돌방지 알고리즘은 한 라운드가 끝난 뒤, 충돌이 발생한 슬롯 수를 사용하여 리더의 인식영역 내에 인식이 되지 않은 태그 수를 예측한다. 그리고 다음 라운드 슬롯 수 결정시에 Q 값을 1씩 변화시키지 않고, 추정된 태그 수에 맞게 Q를 변환하는 방법이다. Class 1 Gen 2에서 충돌이 발생한 슬롯의 시간이 응답이 없는 idle 슬롯의 시간보다 길기 때문에 충돌이 발생하는 슬롯을 줄이기 위하여 예측한 태그 값보다 큰 Q 값을 사용하였다.

프로토콜로 규정된 충돌방지 알고리즘의 태그 인식 시간에 대한 성능을 서로 비교한 결과, 같은 종류의 알로하나 이진트리를 사용하는 알고리즘일지라도 프로토콜로 규정된 클럭 주기와 리더 명령어 포맷이나 태그의 응답 형태와 같은 리더와 태그간의 통신방식에 따라 그 성능이 달라짐을 확인하였다.

개선된 알고리즘과 표준 알고리즘의 성능 시뮬레이션 결과 개선된 Type A 알고리즘은 표준 알고리즘보다 태그 인식 시간이 태그 수 100개 일 때 35.2%, 태그 수 1000개 일 때 14.3%가 감소하였다. 개선된 Type B 알고리즘은 태그 수 100개 일 때 6.0%, 태그 수 1000개 일 때 6.6%가 감소하였다. 개선된 Class 0 알고리즘은 태그 수 100개 일 때 8%, 태그 수 1000개 일 때 12.2%가 감소하였다. 개선된 Class 1 Gen 1 알고리즘은 태그 수 100개 일 때 54.5%, 태그 수 1000개 일 때 63.4% 감소하였다. 개선된 Class 1 Gen 2 알고리즘은 태그 수 100개 일 때 7.9%, 태그 수 1000개 일 때 11.7% 감소하였다. 개선된 충돌방지 알고리즘은 표준 알고리즘보다 태그 인식 성능이 상당히 개선됨을 알 수 있다. 따라서 개선된 충돌방지 알고리즘은 유비쿼터스 센서 네트워크와 텔레메틱스 서비스 등에 사용되는 RFID 시스템의 성능 개선 방안으로 적용될 수 있다.

본 논문에서는 충돌방지 알고리즘을 C언어 구현하여 컴퓨터 시뮬레이션을 수행함으로써 그 성능을 확인하였다. 향후에는 프로그램된 충돌방지 알고리즘을 직접 리더시스템에 포팅하여 태그와 리더 간 통신에 적용할 계획이다.

참 고 문 헌

- [1] K. Finkenzeller, *RFID Handbook*, Wiley & Sons, 2003.
- [2] 유승화, *유비쿼터스 사회의 RFID*, 전자신문사, pp. 22-295, 2005.
- [3] 황태욱, 김영수, 박경환, “900MHz 대역 RFID 시스템의 무선인터페이스 표준화 동향”, *한국전자과학회지*, 6(3), July 2005.
- [4] 양희중, 임준민, “RFID 현황 분석 및 발전 방향에 관한 연구”, *한국산업경영시스템학회*, 28(4), Dec. 2005.
- [5] 정병주, “유비쿼터스 사회로의 발전과 과제”, *한국지방정부학회*, pp. 355-366, Nov. 2005.
- [6] P. Hawkes, “Anti-collision and transponder selection methods for grouped vicinity” Cards and PFID tags”, *RFID Technology (Ref. No. 1999/123)*, *IEE Colloquium on Publication*, Oct. 1999.
- [7] 이현지, 김종덕, “충돌 비트 위치를 활용한 RFID 다중 태그 인식 알고리즘”, *한국통신학회*, 31권 제4A호, April 2006.
- [8] L. Liu, Z. Xie, J. Xi, S. Lai, “An improved anti-collision algorithm in RFID system”, *Mobile Technology Applications and Systems 2005 2nd International Conference*, Nov. 2005.
- [9] J. Myung, W. Lee, “An adaptive memoryless protocol for RFID tag collision arbitration”, *Multimedia IEEE Transactions*, 8(5), pp. 1096-1101, Oct. 2006.
- [10] T. P. Wang, “Enhanced binary search with cut-through operation for anti-collision in RFID systems”, *Communications Letters. IEEE*, 10(4), pp. 236-238, April 2006.
- [11] L. Bolotnyy, G. Robins, “Randomized pseudo-random function tree walking algorithm for secure radio-frequency identification”, *Automatic Identification Advanced Technologies*, pp. 43-48, Oct. 2005.

- [12] S. R. Lee, S. D. Joo, C. W. Lee, "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification", *MobiQuitous 2005*, pp. 166-172, July 2005.
- [13] C. Wille, M. Floerkemeier, "Comparison of transmission schemes for framed ALOHA based RFID protocols", *SAINT Workshops 2006*, Jan. 2006.
- [14] J. R. Cha, J. H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system", *Parallel and Distributed Systems*, Vol. 2, pp. 63-67, July 2005.
- [15] M. A. Bonuccelli, F. Lonetti, F. Martelli, "Tree slotted aloha: a new protocol for tag identification in RFID Networks", *WoWMoM 2006*, pp. 603-608, June 2006.
- [16] H. Vogt, "Efficient object identification with passive RFID tags", *Proc. of Pervasive 2002*, pp. 98-113. 2002.
- [17] ISO/ICE 18000-6 : 20039(E), "Part 6 : Parameters for air interface communications at 860-960 MHz", Nov. 2003.
- [18] 권대근, 김완진, 김형남, "900MHz 표준 프로토콜(ISO/IEC 18000-6 Type B)에서의 충돌방지 성능 개선", *한국통신학회논문지*, 6(5), May 2006.
- [19] Technical Report, "Draft protocol specification for a 900 MHz Class 0 radio frequency identification tag", *Auto-ID Center Massachusetts of Technology*, Feb. 2003.
- [20] M. Nanjundiahm, V. Chaudhary, "Improvement to the anti-collision protocol specification for 900MHz Class 0 radio frequency identification tag", *AINA'05*, 2005.
- [21] Auto-ID Center Massachusetts of Technology, Technical Report, "860~930MHz Class 1 radio frequency identification tag radio frequency & logical communication interface specification candidate recommendation, version1.0.1", Nov. 2002.
- [22] J. Y. Kim, B. S. Kang, D. Y. Yang, "An Improvement of

- bin-slotted anti-collision algorithm for ubiquitous ID system”, *International Journal of Contents*, 2(1), pp. 34-38, March 2006.
- [23] EPCglobal, “EPC™ radio-frequency identity protocols Class-1 Generation-2 UHF RFID protocol for communications at 860MHz~960MHz Version 1.0.9”, 2004.
- [24] 권성호, 모희숙 외, “Gen 2기반 RFID 시스템에서의 충돌방지 알고리즘에 관한 연구”, *한국통신학회논문지*, 6(6), May 2006.
- [25] Y. Kawakita, J. Mitsugi, “anti-collision performance of Gen 2 air protocol in random error communication link”, *SAINTW'06*, 2005.
- [26] E. Zurich, “RFID multiple access methods”, *Summer semester 2004. Luc André Burdet*, 2004.
- [27] R. Rom, M. Sidi, *Multiple Access Protocols Performance and analysis*, Springer-Verlag, pp. 107-142, June 1989.
- [28] B. Zhen, M. Kobayashi, “Framed ALOHA for multiple RFID objects identification”, *IEICE Trans. Communication*, Vol. E88-B, pp. 991-999, March 2005.

감사의 글

돌이켜 보면 짧은 석사과정이지만 본 논문이 결실을 맺을 수 있도록 물심양면으로 격려해 주시고 아낌없는 충고와 지도를 해주신 저의 지도교수님이신 양두영 교수님께 깊이 감사를 드립니다. 그리고 바쁘신 가운데도 항상 곁에서 보다는 논문이 되도록 많은 조언과 충고를 해주신 김홍수 교수님과 좌정우 교수님께 깊은 감사를 드립니다. 또한 대학 시절부터 항상 인자한 모습으로 학문의 길을 이끌어 주신 이용학 교수님, 문건 교수님, 임재운 교수님, 강진식 교수님께 깊이 또 깊이 감사드립니다.

대학원 생활을 하면서 항상 좋은 말씀과 지도를 해주신 강부식 선생님, 이권익 선생님, 홍성욱 선생님, 강봉수 선생님 정말 감사드립니다. 통신공학과 조교 선생님이신 진호오빠와 윤희언니에게도 감사의 마음을 전합니다. 그리고 같은 연구실에서 벗으로서 항상 힘이 되어준 서방신기 멤버인 정현 오빠, 창남 오빠, 도형 오빠 그리고 은주에게도 고마운 마음을 전하고 싶습니다. 우리 연구실 식구였던 성익 오빠, 기원 오빠, 태성 오빠, 준호 오빠, 윤길, 종석, 보람이와 프로그램을 코딩할 때 많은 도움을 주었던 이동통신연구실의 경보와 동현 오빠에게도 고마운 마음을 전하고 싶습니다. 그리고 항상 좋은 정보를 주는 김동현 오빠, 보길 오빠도 참 감사합니다.

대학원 생활동안 연락도 자주 못한 내 친구들 재미, 미숙, 민희, 연화야~ 고맙고 미안~ 앞으로 연락 자주 할게. 그리고 내 짜증을 잘 받아 주었던 지현 오빠도 참 고마워.

끝으로 오늘이 있기까지 말없이 지켜봐 주신 부모님과 오빠에게도 감사의 말을 전합니다.