

논문 2007-44CI-6-7

RFID 프로토콜의 충돌방지 알고리즘의 성능 개선과 알고리즘 비교

(Performance Improvement of Anti-collision Algorithm for RFID
Protocol and Algorithm Comparison)

임 정 현*, 김 지 윤*, 좌 정 우**, 양 두 영**

(Jung-Hyun Lim, Ji-Yoon Kim, Jeong-Woo Jwa, and Doo-Yeong Yang)

요 약

본 논문에서는 RFID 시스템에 사용되는 무선 환경 표준 프로토콜 중 UHF 대역 프로토콜인 ISO 18000-6의 타입들과 EPCglobal의 클래스들에 대한 충돌방지 알고리즘을 구현하였다. 또한 표준 프로토콜의 성능을 개선한 충돌방지 알고리즘을 제안하고, 링크타이밍의 클록 주기를 2세대 클래스-1 주기인 12.5 μ s로 동일하게 한 상태에서 그 성능을 비교하였다. 그 결과, 500 개의 태그가 리더인식영역 내에 동시에 존재할 때 표준 프로토콜에 대한 태그인식 성능은 1세대 클래스-1, Type B, Type A, 클래스-0, 2세대 클래스-1의 차례로 좋아지고, 개선된 프로토콜에 대한 태그인식 성능은 Type B, Type A, 1세대 클래스-1, 클래스-0, 2세대 클래스-1의 순서로 좋아진다. 그러므로 프로토콜로 규정된 클록 주기와 리더와 태그 간의 링크타이밍에 따라 태그 인식성능이 현저히 달라짐을 알 수 있다.

Abstract

In this paper, Air-interface protocols of ISO 18000-6 Types and EPCglobal Classes applied to RFID system in UHF band are analyzed, and those anticollision algorithms are realized. Also, the each algorithm which improves the performance of standard protocol is proposed, and the performance is compared when clock period of link timing is a identical condition on 12.5 μ s. As the result, when 500 tags exist simultaneously inside reader interrogation zone, the tag recognition performance of a standard protocol is better in preceding order of Class-1 Generation-1, Type B, Type A, Class-0 and Class-1 Generation-2. And also the performance of improved protocol is better in ascending order of Type B, Type A, Class-1 Generation-1, Class-0 and Class-1 Generation-2. Therefore, performance of tag recognition remarkably depends on the regulated clock period in the protocol and link timing between a reader and a tag.

Keywords : RFID system, Anti-collision Algorithm, Clock Period, Performance of Tag Recognition

I. 서 론

RFID 시스템은 무선주파수를 이용하여 태그에 부착된 IC칩에 저장되어 있는 고유 정보를 비접촉식으로 판독하는 식별장치이다. 따라서 RFID는 기존의 바코드나 자기 인식 장치의 단점을 보완하고 사용의 편리성을 향

상시켜 줄 무선 인식 시스템 구현에 있어 가장 핵심적인 기술이다. 이러한 RFID 시스템은 태그, 리더, 미들웨어 및 응용 서비스 플랫폼으로 구성된다^[1].

RFID 시스템에서 리더는 인식영역 내의 태그에 요청 메시지를 전송하고, 요청 메시지를 수신한 태그는 자신의 정보를 리더로 전송한다. 그러나 인식영역 내에 다수의 태그가 존재하여 동시에 요청 메시지에 대한 응답을 하게 되면 RF 통신채널 상에 충돌(collision)이 발생하게 되고, 결과적으로 리더는 태그의 정보를 정확하게 수신하지 못하게 된다. 이러한 문제를 해결하기 위해 충돌방지 알고리즘(anticollision algorithm)이 사용되며,

* 학생회원, ** 정회원, 제주대학교 통신공학과
(Cheju University, Dept. of Tele-
communication Engineering)

※ 본 연구는 산업자원부와 한국산업기술재단의 지역
혁신인력양성사업으로 수행된 연구임.

접수일자: 2007년9월11일, 수정완료일: 2007년11월6일

이는 다중접속 방식 시스템의 성능을 결정짓는 중요한 요소가 된다^[2]. RFID 시스템의 성능은 태그를 인식하는데 필요한 시간과 태그가 소모하는 전력으로 결정된다. 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.

무선 인식에 관한 하드웨어 규정과 리더와 태그 간의 통신방식 등을 제정하는 표준은 ISO 18000, EPCglobal, U-센터 등 여러 단체에서 진행되고 있다.

본 논문에서는 개선된 충돌방지 알고리즘의 성능을 비교 분석하기 위하여 UHF 대역 프로토콜로 규정된 ISO의 18000-6 Type A와 Type B, EPCglobal의 클래스-0, 클래스-1의 1세대와 2세대의 알고리즘에 대한 리더와 태그 사이의 통신방식과 태그 인식과정을 분석한다. 그리고 각 프로토콜에서 규정된 충돌방지 알고리즘인 프레임 알로하, 확률적 알로하, 이진트리, 빈 슬롯, 슬롯 알로하 알고리즘의 성능을 C언어로 구현하여 확인한다. 분석된 결과를 바탕으로 리더와 태그 간의 송·수신되는 명령어를 줄이거나 태그 응답 타이밍을 줄이는 방식을 사용하여 각 알고리즘의 성능을 개선하고 그 성능을 비교 분석한다.

II. UHF 대역 RFID 프로토콜의 분석 및 성능 개선

1. ISO 18000-6 Type A/B 에어인터페이스

(1) Type A 충돌방지 알고리즘 분석 및 성능 개선

ISO 18000-6 Type A는 충돌방지 알고리즘으로 프레임 슬롯알로하(frame slotted aloha) 방식을 사용한다^[3].

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 μ s 이상을 유지해야 리더는 채널이 비어있다고 판단하고 태그에게 명령어를 전송한다. 이 시간동안 리더는 RF 신호를 전송하고, 적절한 RF 에너지를 수신한 태그는 리더 명령어를 수행할 수 있는 대기 상태(Ready state)로 전환된다. 리더는 대기 상태에 있는 태그들에게 라운드 사이즈(round size)를 포함하는 Init_round 명령어를 전송한다. 이 명령어를 수신한 태그들은 태그 인식과정을 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 슬롯 카운터(slot_counter)를 0으로 초기화한다. 만약 슬롯 카운터와 자신이 선택한 슬롯 넘버가 일치하는 태그는 자신의 4비트 서명(signature)을 포함하는 ID를 응답으로 리더로 전송하고, 불일치하는 태그는 다음 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 슬롯 카운터 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 슬롯 카운터 값을 1씩 증가시킨다. 그리고 난후 슬롯 넘버와 슬롯 카운터가 일치하는 태그는 자신의 ID와 4비트 서명을 응답한다. 현재 상태의 라운드는 슬롯 카운터와 라운드 사이즈가 같아질 때까지 계속되며, 라운드가 끝나면 다음 상태의 라운드 사이즈를 설정한 후 리더는 Init_round 명령어를 전송한다. 태그는 전송할 새로운 슬롯 넘버와 서명을 선택하여 새로운 라운드를 시작한다. ISO 18000-6 Type A는 이러한 과정을 반복적으로 수행하여 태그를 인식한다.

본 논문에서는 리더가 대기상태에 있는 태그들에게 라운드 사이즈를 포함하는 Init_round 명령어를 전송할 때 그림 1의 점선으로 표시된 부분에서 설명된 것과 같이 인식된 슬롯과 충돌이 발생한 슬롯으로부터 인식되지 않은 태그 수를 예측하여 그에 적합한 라운드 사이즈를 설정하여 태그로 명령을 전송한다. 태그는 리더로부터 전송받은 라운드 사이즈를 기준으로 자신이 전송할 임의의 슬롯 넘버를 선택하고 이를 슬롯 카운터에 로드한다. 리더가 태그를 인식함에 있어서 슬롯 카운터와 라운드 사이즈가 같아져서 현재 상태의 라운드가 종료되면, 리더는 인식영역 내의 태그 수를 예측하여 다음 상태의 라운드 사이즈를 그림 1에서 설명되고 있는 방법에 따라 재설정 후 Init_round 명령어를 다시 태그로 전송한다. 이때 충돌이 발생한 슬롯 수를 파악하여 현재 남아 있는 태그 수를 확률이론에 따라 예측한다.

태그를 인식하는 과정에서 i 번째 라운드의 프레임 사이즈가 $N(i)$, 태그의 개수가 $n(i)$ 일 때 이항분포에 따른 확률밀도함수는 다음과 같다^[4].

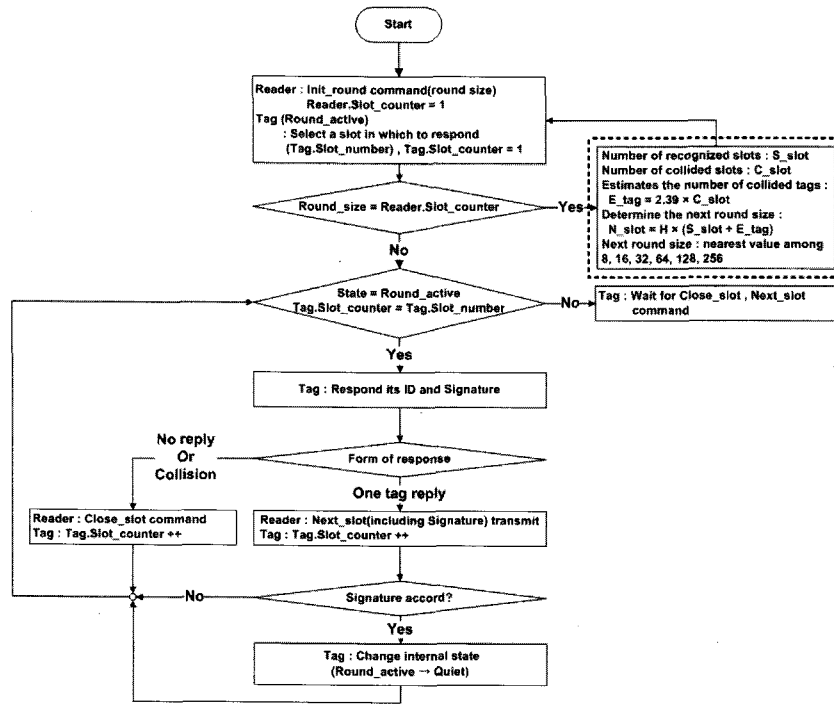


그림 1. Type A 충돌방지 알고리즘의 태그 인식과정 흐름도
 Fig. 1. Flow chart of proposed Type A anti-collision algorithm for tag recognition process.

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

여기서, $q=0$ 은 태그의 응답이 없는 빈 슬롯 E_{slot} 의 확률, $q=1$ 은 성공적으로 태그의 응답 데이터가 리더로 전송될 확률, 나머지($q \geq 2$)는 충돌이 발생할 확률이다. 이전 프레임에서 인식된 슬롯은 하나의 태그가 응답한 슬롯이고 충돌 슬롯은 최소 두 개의 태그가 응답한 슬롯이다. 즉 태그 수의 최소값은 하나의 태그가 응답한 인식된 슬롯의 수와 최소 두 개 이상의 태그가 응답한 충돌이 발생한 슬롯 개수의 배수의 합으로서 다음과 같이 나타낼 수 있다.

$$n_{est-min} = S_{slot} + 2 \times C_{slot} \quad (2)$$

실제 충돌 슬롯에서 구성하는 태그의 수는 2개보다 많은 경우도 있게 된다. 이 경우 태그의 수를 무한대라고 가정하고 태그 수의 기대 값을 구하면 식 (3)과 같이 나타낼 수 있고 현재 i 번째 라운드에서 예측된 태그 수는 충돌이 발생한 슬롯 수 C_{slot} 과 성공한 슬롯 수 S_{slot} 을 사용하여 식 (4)와 같이 쓸 수 있다.

$$K = \lim_{N \rightarrow \infty} \sum_{q=2}^N q P_q(i) = 2.39 \quad (3)$$

$$n_{est} = (P_1(i) \times N(i)) + \left(\sum_{q=2}^N q P_q(i)\right) \times C_{slot} = S_{slot} + K \times C_{slot} \quad (4)$$

따라서 다음 라운드에 참여하는 태그의 수는 충돌이 발생한 슬롯의 개수로부터 예측할 수 있으며, 예측된 프레임 사이즈는 다음과 같다.

$$N(i+1) = \left(1 - \frac{1}{K}\right) \times n_{est} \quad (5)$$

ISO 18000-6 Type A에서는 라운드 크기를 지정하기 위하여 3비트의 2진수를 사용할 수 있으며, 000에서부터 110까지의 각 3비트의 2진수에는 1, 8, 16, 32, 64, 128, 256이라는 라운드 크기가 할당되어 있고 111은 향후의 사용을 위한 부분으로 할당되어 있다.

그림 1은 Type A 표준 프로토콜에 논문에서 제안한 라운드 크기 추정 알고리즘을 포함시켜 나타낸 충돌방지 알고리즘의 흐름도이다. 이 알고리즘은 확률이론을 도입하여 예측한 태그 수로부터 다음 라운드의 크기를 설정하는 처리과정이 포함된다. 또한 식 (4)로 추정된 라운드 크기에 따라 설정된 동적프레임 슬롯 할로하 방식과 더불어 리더와 태그 간의 응답에 따른 링크 형식에 Early-end 방식^[5]을 적용하여 전체 프레임에서 태그의 응답이 없는 슬롯의 점유 시간을 단축하여 성능을 개선한다. 이 때 슬롯의 범위는 리더로부터의 무선신호

SOF(start of frame)와 EOF(end of frame)에 의해 결정된다. EOF 신호는 리더 명령에 의해 슬롯의 어느 구간에서나 전송이 가능하므로 태그의 응답이 있는 구간을 제외하고, 태그의 응답이 없는 구간에서 리더가 EOF 명령어를 일찍 전송하면 슬롯의 점유 시간이 감소된다. 하나의 리더 명령에 대한 태그 응답의 링크 형식은 [Preamble][Flags][Parameters][Data][CRC-16]으로 구성되지만, [Data][CRC-16]을 제외한 [Preamble] [Flags] [Parameters]를 우선 파악하여 태그 응답의 유무를 판별하고, 응답이 없는 슬롯인 경우 Close_slot 명령을 전송함으로써 슬롯의 점유 시간을 줄임으로써 태그 인식 시간을 감소시켜 인식성능을 향상시킨다.

(2) Type B 충돌방지 알고리즘 분석 및 성능 개선

ISO 18000-6 Type B는 충돌방지 알고리즘으로 이진 탐색(binary search) 방식의 일종인 SBT(slotted binary tree) 방식을 사용한다^[6].

리더의 인식 범위 내에 있는 태그가 리더로부터 동작에 필요한 에너지를 공급받으면 Power-off 상태에서 리더 명령어를 수행할 수 있는 대기 상태로 전환된다. 대기 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 난후 ID 상태이고, COUNT가 0인 태그는 자신의 ID를 리더로 응답한다. 이러한 과정을 반복함으로써 리더는 인식영역 내에 있는 다중 태그들을 충돌 없이 인식한다.

본 논문에서는 응답의 형태(Form of response)에 따라 판단기준을 충돌이 발생한 슬롯, 응답이 없는 슬롯, 하나의 태그가 응답한 슬롯으로 분석한다. 충돌이 발생한 슬롯에서 태그들은 다시 '0'과 '1'로 분기하고 '0'을 선택한 태그들을 우선 인식한 뒤 '1'을 선택한 태그들을 인식하게 된다. 분기가 이루어진 후에도 태그가 유효하게 인식되는 경우, 충돌이 다시 발생하는 경우, 태그의 응답이 없는 경우가 있다. 태그의 응답에 따라 카운터의 값을 증감시키고, 명령어를 보내는 시간을 단축하는 방법을 적용하여 Type B의 성능을 개선한다. 그리고 하나의 리더 명령어에 대한 태그 응답의 링크형식은 [Data][CRC-16]에 앞서 [Return Preamble]를 먼저 파악하여 태그의 응답 유무를 판별하고, 응답이 없는 슬롯인 경우 Success 명령어를 전송하여 전체 프레임에서 하나의 슬롯이 점유하는 시간을 감소시킴으로써 태그 인식 성능을 향상시킨다.

2. EPCglobal 프로토콜의 에어인터페이스

(1) EPC 클래스-0의 분석 및 성능 개선

클래스-0은 태그를 인식하기 위한 방법으로 이진트리(binary tree)방식^[7]을 사용한다.

클래스-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 클리어 되면 Tree traversal 상태로 들어간다. Tree traversal 상태는 태그 ID 비트열을 탐색하는 단계로 이 상태에 있는 태그는 자신의 ID 비트열을 MSB(most significant bit)에서부터 시프트 시키면서 리더로 한 비트 씩 전송한다. 리더는 태그의 응답들을 검사하고 태그로부터 수신한 비트를 다시 태그로 전송한다. 만약, 태그의 응답들이 충돌이면 리더는 프로그래밍 된 상태에 따라 0 또는 1로 분기하여 선택된 비트를 태그로 전송한다. 리더로부터 수신한 비트가 이전에 리더로 전송한 비트와 동일하면 태그는 1비트를 시프트 시켜 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

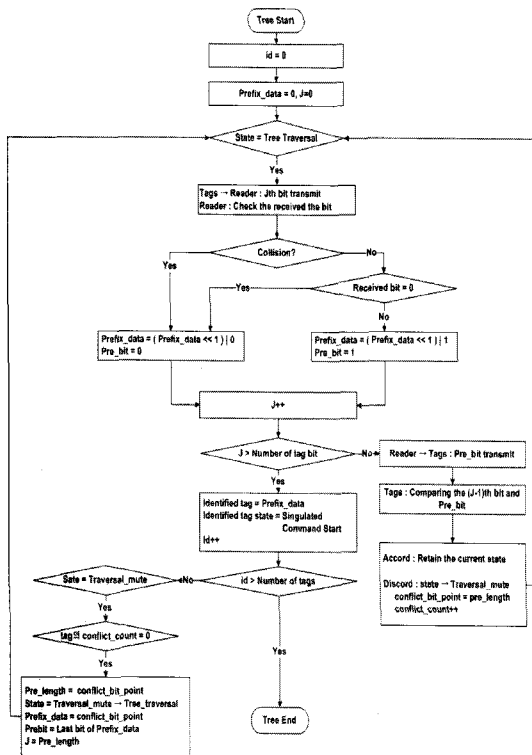


그림 2. 개선된 클래스-0 충돌방지 알고리즘의 흐름도
 Fig. 2. Flow chart of improved Class-0 anticollision algorithm.

되고 태그 하나가 완벽히 인식된다. 태그 하나가 인식되고 리더가 'null' 데이터를 전송하면 Traversal mute 상태인 태그들은 Tree start 상태로 되어 트리에 참여하고, 인식된 태그는 ID 플래그가 세팅되고 Dormant 상태가 된다.

EPC 클래스-0의 충돌방지 알고리즘을 개선하기 위해 리더가 많은 수의 태그를 인식할 때, 태그들의 최상위 비트열은 동일한 패턴을 가진다는 점을 이용한다. 매 read cycle마다 루트노드로부터 트리를 시작하지 않고 태그의 충돌이 발생한 비트 포인트(bit point)에서 시작하는 방식이다. 이를 위하여 태그에 충돌 비트 포인터(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 인식하면 충돌 카운터를 감소시켜 '0'이 되는 태그는 Tree traversal 상태로 되어 충돌 위치부터 트리를 시작한다.

그림 2는 개선된 클래스-0 알고리즘의 흐름도를 나타낸 것이다. 리더와 태그 간의 통신이 설정되면 태그는 Tree traversal 상태로 들어간다. 이 상태에 있는 태그는 자신의 ID 비트열을 MSB부터 시프트 시키면서 리더로 한 비트 씩 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_data열에 추가한다. 만약 그 응답유형이 충돌이면, 리더는 미리 프로그래밍 된 상태에 따라 0과 1로 분기하는데 본 논문에서는 0으로 분기하도록 하였다. 리더로부터 수신한 비트가 만약 이전에 전송했던 비트(j-1번째)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 ID길이+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.

(2) 1세대 클래스-1 알고리즘의 분석 및 성능 개선

EPC 1세대 클래스-1(Class-1 Generation-1) 알고리즘에서 리더와 태그의 통신은 패킷 단위로 이루어진다 [8].

태그는 충분한 에너지를 받게 되면 PowerOFF State에서 Quiet State로 이동한다. Quiet State에 있는 태그들은 리더로부터 Talk 명령을 수신하고, Talk 명령 내의 [PTR]이 가리키는 위치의 [VALUE] 값과 일치하는 태그는 Active State로 이동한다. 반대로 Active State에서 Quiet 명령을 받게 되면 Quiet 명령 내의 [PTR]이 가리키는 위치의 [VALUE] 값과 일치하는 태그는 Quiet State로 이동한다. Active State의 태그는 PingID 명령을 사용하여 태그를 인식하고 PingID를 받은 태그들은 자신의 태그 ID 중 최상위 3비트와 일치하는 빈 슬롯을 할당 받는다. 태그를 인식한 후 ScrollID 명령을 사용하여 인식된 태그를 Quiet State로 이동시킨다. 만약, Active State에서 리더의 Kill 명령을 받은 태그가 있다면, 그 태그는 Killed State로 이동하고 Dead 상태가 된다.

1세대 클래스-1의 충돌방지 알고리즘은 이진트리에

기반을 둔다. 이진트리는 원래 1비트 단위로 트리를 확장하는 형태이지만, 1세대 클래스-1의 빈 슬롯(bin slot) 충돌방지 알고리즘은 3비트 단위로 확장하는 형태이다. 1세대 클래스-1에서 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.

본 논문에서는 기존의 1세대 클래스-1 충돌방지 알고리즘에서 스택 개념을 도입하고 충돌이 일어나지 않은 Bin을 우선 인식하고, 충돌이 일어난 Bin은 응답으로 수신된 8비트열들을 서로 비교하여 일치하는 비트열을 스택에 저장하여 알고리즘을 개선하였다. 하나의 태그를 완벽히 인식한 후, 다시 'VALUE=0, LEN=0'으로 하여 PingID 명령어를 보내는 방법이 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 보낸다.

그림 3은 개선된 EPC 1세대 클래스-1 알고리즘의 흐름도를 나타낸 것이다.

개선된 1세대 클래스-1 알고리즘에서 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 임시

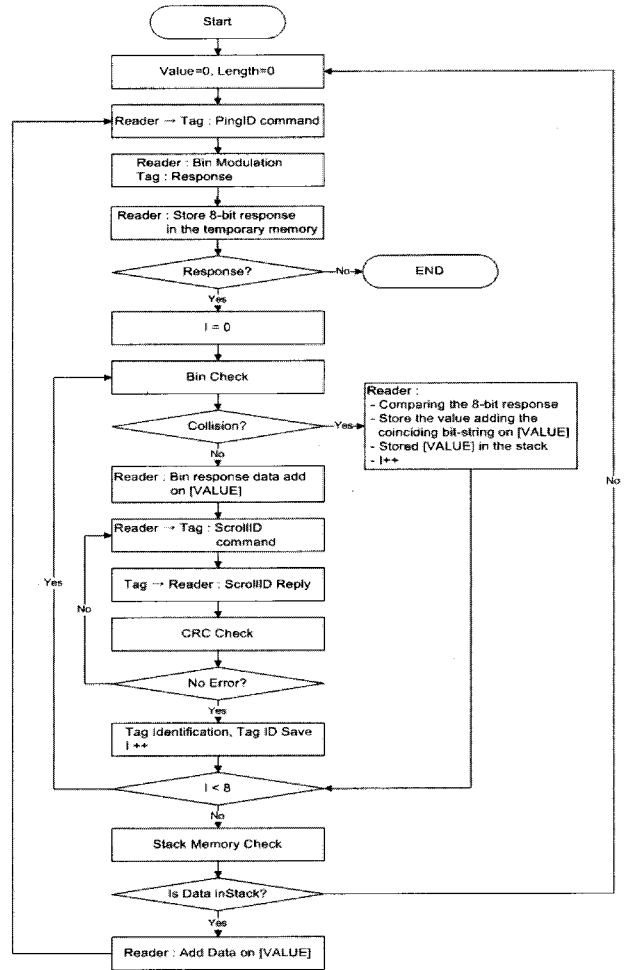


그림 3. 개선된 1세대 클래스-1 충돌방지 알고리즘의 흐름도

Fig. 3. Flow chart of improved Class-1 Generation-1 anticollision algorithm.

메모리를 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) 2세대 클래스-1 알고리즘의 분석 및 성능 개선

EPC 2세대 클래스-1(Class-1 Generation-2)의 태그 인식은 슬롯 알로하(slotted aloha) 방식을 사용한다^[9]. 인식영역 내의 태그를 인식하기 위해서 리더는 기본적으로 SELECT, INVENTORY 그리고, ACCESS의 세 동작과정을 이용한다.

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. 먼저 리더는 Select 명령을 태그에 내리며 Select 명령을 받은 태그들은 자신이 속할 INVENTORY를 선택받고, 동작을 준비하게 된다. 다시 리더는 태그에 INVENTORY 명령들을 내려 보내 태그를 인식하게 되고 ACCESS 명령들을 통해 태그의 데이터를 읽기 또는 쓰기를 수행하게 된다. INVENTORY 과정은 Query, QueryAdjust, QueryRep, ACK, 그리고 NAK 등의 명령어들로 구성된다. Query는 INVENTORY 프레임의 슬롯 개수를 초기화하고 태그가 프레임에 참여하게끔 한다. 프레임에 참여한 태그는 Query 명령어에 포함되어 있는 슬롯 파라미터 Q를 참조하여 랜덤 넘버(random number)를 선택하고 이를 슬롯 카운터(slot_counter)에 로드한다. 여기서 '0'을 선택한 태그들은 Reply 상태로 전환되고, 전환되는 즉시 리더로 응답 메시지를 전송한다. 그 밖의 '0'이 아닌 랜덤 넘버를 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. 프레임을 초기화시키는 Query가 발생한 후에, 리더는 하나 이상의 QueryAdjust 또는 QueryRep 명령어를 전송한다. QueryAdjust 명령어는 이전 Query를 반복하고 Q 값을

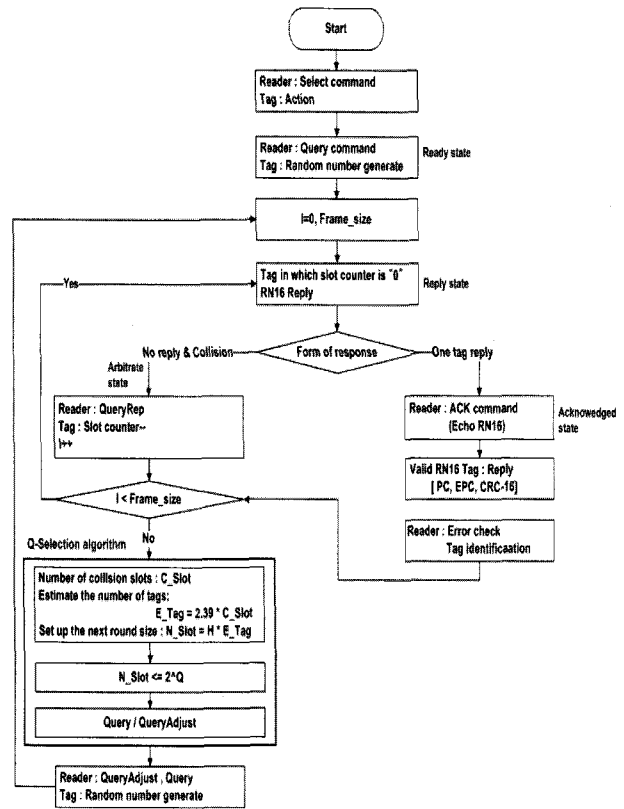


그림 4. 개선된 2세대 클래스-1 충돌방지 알고리즘의 흐름도

Fig. 4. Flow chart of improved Class-1 Generation-2 anticollision algorithm.

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 랜덤 넘버를 선택하고 랜덤 넘버는 그들의 슬롯 카운터로 로드된다. 여기서 Arbitrate 상태에 있는 태그들은 그들의 슬롯 카운터를 QueryRep 명령어를 수신할 때마다 감소시키고 슬롯 카운터가 [0000h]에 도달했을 때 Reply 상태로 전환되어 RN16을 리더에 전송한다. 이런 과정을 반복적으로 수행함으로써 태그를 인식할 수 있다.

기존의 2세대 클래스-1 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 확률이론으로 추정하여 그에 맞는 Q 값을 사용한다.

그리고 개선된 알고리즘에서는 충돌이 발생한 슬롯의 시간이 응답이 없는 idle 슬롯의 시간보다 길기 때문에 충돌이 발생하는 슬롯을 줄이기 위하여 예측한 태그 값보다 가장 근접하면서 큰 Q 값을 사용하였다. 예를 들어 예측한 태그 수가 72일 경우 Q 값을 6과 7중에 7을 선택 하여 슬롯 개수를 128개로 하여 충돌을 최소화하는 방법을 사용하였다.

그림 4는 개선된 2세대 클래스-1 알고리즘의 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 그리고 빈 슬롯이 발생할 확률은 모두 앞의 식 (1)의 이항분포를 따른다. 현재 i 번째 라운드에서 예측된 태그 수는 성공한 슬롯 수와 충돌이 발생한 슬롯 수를 사용하면 식 (2), (3)과 같이 쓸 수 있다. 또한 다음 라운드에 참여하는 태그 수는 충돌이 발생한 태그이므로, 라운드 크기는 식 (4)와 같다.

위의 동작과정을 설명하면, 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 대기 상태로 전환하고, Query 명령어에 포함된 Q-파라미터를 사용하여 $(0 \sim 2^Q - 1)$ 범위에서 자신의 슬롯 넘버(RN16)를 선택하고 이를 슬롯 카운터로 로드한다. 여기서 '0'을 선택한 태그들은 Arbitrate 상태로 전환되고, 전환되는 즉시 리더로 자신의 RN16을 응답 메시지로 전송한다. 그 밖의 '0'이 아닌 랜덤 넘버를 선택한 태그들은 Arbitrate 상태로 전환되고 QueryAdjust 또는 QueryRep 명령어를 기다린다. 만약 그 응답유형이 하나의 태그만 응답하였다면, 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

슬롯 카운터 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 (2)와 식 (4)와 같이 충돌이 발생한 슬롯 수를 사용하여 태그의 개수를 추정하고, 충돌을 최소화하기 위하여 예측한 태그 값보다 가장 근접하면서 큰 Q 값을 설정한다. 재설정된 Q 값을 포함하는 QueryAdjust 명령어를 전송하여 위의 과정을 반복한다.

III. 시뮬레이션 및 성능 비교

본 논문은 UHF 대역 RFID 프로토콜인 ISO 18000-6 Type A/B, EPC 클래스-0, 1세대 클래스-1, 2세대 클래스-1의 충돌방지 알고리즘의 성능을 개선하고 분석하였다. 충돌방지 알고리즘은 C언어를 이용하여 구현하고, 프로토콜로 규정된 클록 주기를 동일하게 하여 통신특성에 따라 태그 인식시간 성능이 어떻게 달라지는지를 분석하였다.

그림 5는 Type A, Type B, 클래스-0, 1세대 클래스-1, 2세대 클래스-1의 프로토콜로 규정된 표준 충돌방지 알고리즘의 태그 인식시간을 비교한 것이다.

시뮬레이션 결과 동일한 슬롯 알로카 방식을 사용하는 2세대 클래스-1과 Type A를 비교해 보았을 때 2세대 클래스-1의 성능이 Type A보다 좋음을 알 수 있다. 이는 프로토콜에서 규정된 클록 주기가 2세대 클래스-1인 경우에는 $12.5 \mu s$, Type A인 경우에는 $20 \mu s$ 로 더 길고 리더 명령어에 대한 태그의 응답 시에 2세대 클래스-1은 16비트의 RN16을 전송하고 Type A는 ID와 CRC 전체를 보내 전송비트 수가 많기 때문에 태그 인식시간이 더 오래 걸렸다. 그리고 Type A인 경우 태그 인식시간이 태그 수 400개 이상에서 급격히 증가하는데, 이는 Type A의 경우 확장 가능한 슬롯의 최대 범위가 256개이기 때문이다. 이진트리 방식의 계열을 사용하는 Type B, 클래스-0, 1세대 클래스-1을 비교해 보았을 때 태그 인식시간이 클래스-0이 가장 짧고 다음에는 1세대 클래스-1, Type B 순이다. 이는 클래스-0과 리더와 태그의 통신이 설정된 후에 1비트 단위로 리더와 태그 간 통신을 하면서 태그 ID를 찾아내는 반면, 1세대 클래스-1과 Type B는 패킷 단위로 통신이 이루어지기 때문에 클래스-0의 성능이 좋다. 1세대 클래스-1은 태그 ID를 찾을 때 그 응답으로 ID의 일부인 8비

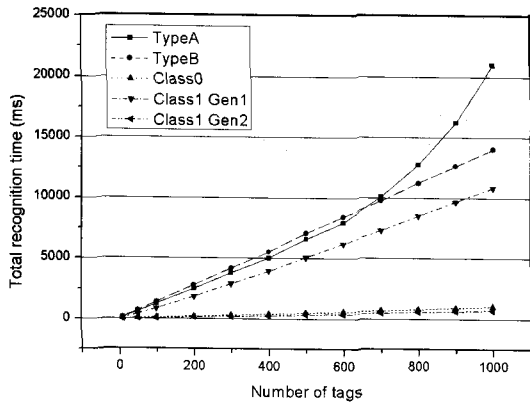


그림 5. 표준 충돌방지 알고리즘의 총 태그 인식시간
Fig. 5. Total tag recognition time of standard anticollision algorithm.

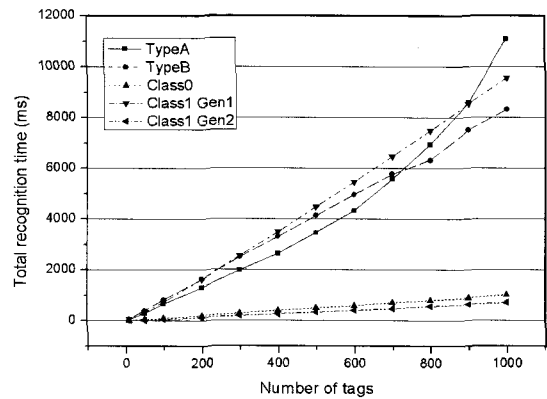


그림 7. $T_0=12.5 \mu s$ 일 때, 표준 충돌방지 알고리즘의 총 태그 인식시간
Fig. 7. $T_0=12.5 \mu s$, Total tag recognition time of standard anticollision algorithm.

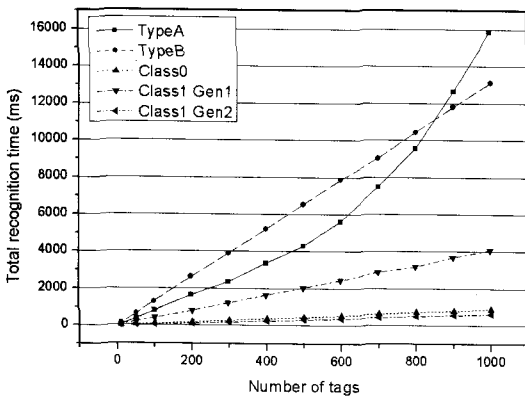


그림 6. 개선된 충돌방지 알고리즘의 총 태그 인식시간
Fig. 6. Total tag recognition time of improved anticollision algorithm.

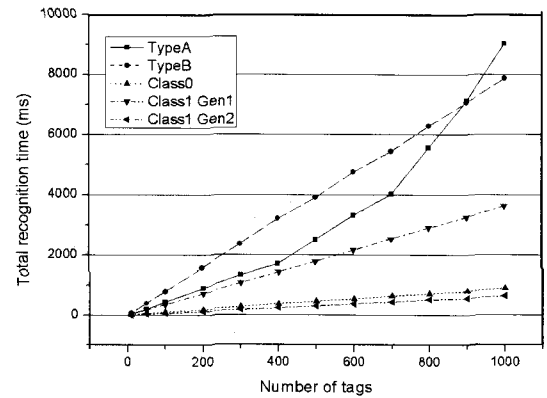


그림 8. $T_0=12.5 \mu s$ 일 때, 개선된 충돌방지 알고리즘의 총 태그 인식시간
Fig. 8. $T_0=12.5 \mu s$, Total tag recognition time of improved anticollision algorithm.

트를 전송하는 반면, Type B는 전체 ID를 리더로 전송하고 1세대 클래스-1의 클럭 주기는 $14.25 \mu s$ 이고, Type B는 $20 \mu s$ 이기 때문에 Type B의 인식시간이 가장 길다.

그림 6은 Type A, Type B, 클래스-0, 1세대 클래스-1, 2세대 클래스-1의 개선된 충돌방지 알고리즘의 총 태그 인식시간을 비교한 것이다.

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

그림 7은 리더와 태그 간 클럭 주기(T_0)를 $12.5 \mu s$ 로 동일하게 하여 표준 충돌방지 알고리즘의 총 태그 인식시간을 비교한 것이고, 그림 8은 리더와 태그 간 클럭 주기를 $12.5 \mu s$ 로 동일하게 하여 개선된 충돌방지 알고

리즘의 총 태그 인식시간을 비교한 것이다.

앞의 그림 5와 그림 6을 클럭 주기(T_0)를 $12.5 \mu s$ 로 동일하게 한 경우인 그림 7, 8과 비교해보면 클래스-0과 2세대 클래스-1의 태그 인식시간 성능 순위는 앞의 시뮬레이션 결과와 차이가 없지만, Type A와 B, 1세대 클래스-1의 성능은 차이를 보였다. 이는 Type A와 B의 프로토콜로 규정된 클럭 주기가 $20 \mu s$ 인데 반해 1세대 클래스-1은 $14.25 \mu s$ 이기 때문에 Type A와 B의 태그 인식시간이 더 감소되었기 때문이다. 이로써 프로토콜로 규정된 클럭 주기에 따라 태그 인식성능이 좌우됨을 알 수 있다.

IV. 결론

본 논문에서는 RFID 시스템에서 사용하는 UHF 대

역 프로토콜인 ISO의 18000-6 Type A와 Type B, EPCglobal의 클래스-0, 1세대 클래스-1, 2세대 클래스-1의 에어인터페이스를 분석하고 규정된 충돌방지 알고리즘을 구현하였다. 그리고 프로토콜에서 규정하는 충돌방지 알고리즘의 성능을 개선한 알고리즘을 제안하고 성능을 비교하였다.

프로토콜에서 규정된 충돌방지 알고리즘의 태그 인식시간에 대한 성능을 서로 비교한 결과, 같은 종류의 알고리즘이나 이진트리를 사용하는 알고리즘일지라도 프로토콜로 규정된 클록 주기와 리더 명령어 포맷이나 태그의 응답 형태와 같은 리더와 태그 간의 통신방식에 따라 그 성능이 달라짐을 확인하였다. 따라서 본 논문에서 제안하는 송수신 되는 명령어를 줄이거나 링크 타이밍을 줄이는 방식을 사용하면 리더와 태그 간 인식시간을 줄일 수 있다. 또한 태그 데이터 간의 충돌을 최소화하여 RFID 시스템의 성능향상에 기여할 수 있다.

참 고 문 헌

- candidate recommendation version 1.0.1," 2002.
- [9] EPCglobal, "EPCTM radio-frequency identity protocol Class-1 Generation-2 UHF RFID protocol for communication at 860~960 MHz version 1.0.9," 2004.
- [1] D. W. Engels and S. E. Sarma, "The reader collision problem," IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, pp. 6, October 2002.
- [2] K. Finkenzeller, "RFID Handbook" Wiley & Sons, 2003.
- [3] ISO/IEC "Information technology - radio frequency identification for item management part7: parameters for active air interface communications at 860 MHz to 960 MHz," May 2004.
- [4] B. Zhen and M. Kobayashi, "Framed ALOHA for multiple RFID objects identification," IEICE Trans. Communication, Vol. E88-B, pp. 991-999, 2005.
- [5] E. Zurich, "RFID multiple access methods," Summer semester 2004, Luc Andre Burdet, 2004.
- [6] H. S. Choi, J. R. Cha and J. H. Kim, "Fast wireless anti-collision algorithm in ubiquitous ID system," Vehicular Technology Conference 2004, Vol. 6, pp. 4589-4592, September 2004.
- [7] Auto-ID Center Massachusetts of Technology, Technical Report, "Draft protocol specification for a 900 MHz Class 0 radio frequency identification tag," 2003.
- [8] Auto-ID Center Massachusetts of Technology, Technical Report, "860~930 MHz Class-1 radio frequency identification tag radio frequency & logical communication interface specification

저 자 소 개



임 정 현(학생회원)
2005년 제주대학교
통신컴퓨터공학부
학사졸업.
2006년~현재 제주대학교 대학원
통신공학과 석사과정.

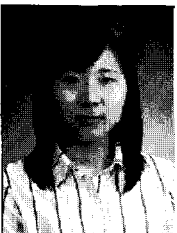
<주관심분야 : RF 회로 및 RFID, 안테나, 마이
크로파 능동회로>



좌 정 우(정회원)
1985년 한양대학교
전자공학과 학사졸업.
1987년 KAIST 전기 및
전자공학과 석사졸업.
2001년 KAIST 정보 및
통신공학과 박사졸업.

2002년~현재 제주대학교 통신컴퓨터공학부
통신공학전공 조교수

<주관심분야 : 무선인터넷 BM, 이동통신 프로토
콜, 이동통신 시스템>



김 지 윤(학생회원)
2005년 제주대학교
통신컴퓨터공학부
학사졸업.
2007년 제주대학교 통신공학과
석사졸업.

<주관심분야: RF 회로 및 RFID, RFID 프로토
콜>



양 두 영(정회원)-교신저자
1984년 제주대학교 통신공학과
학사졸업.
1989년 한양대학교 전자통신
공학과 석사졸업.
1992년 한양대학교 전자통신
공학과 박사졸업.

1992년~현재 제주대학교 통신컴퓨터공학부
교수.

<주관심분야 : RF 회로 및 RFID, 이동통신, 위성
통신>