

논문 2006-43TC-1-6

# Bin 슬롯 정보를 이용한 UHF 대역 Anti-collision 알고리즘

## ( Anti-collision algorithm using Bin slot information for UHF )

최 호 승\*, 김 재 현\*\*

(Ho-Seung Choi and Jae-Hyun Kim)

### 요 약

RFID 시스템에서의 태그인식속도와 정확성을 결정하는 Anti-collision 알고리즘은 매우 중요하다. 본 논문에서는 Bin 슬롯을 사용하는 EPC CLASS 1 UHF Anti-collision 알고리즘의 성능을 분석하고, 기존 알고리즘을 개선한 Anti-collision 알고리즘을 제안한다. 제안한 알고리즘에서는 리더가 Bin 슬롯 정보를 저장하고, 그 정보를 이용하여 불필요한 PingID 명령의 반복 회수를 줄여 태그인식시간을 줄인다. 또한 제안한 알고리즘에서 ScrollAllID 명령을 사용하면 리더는 태그 ID들의 충돌정보를 알게 되고, 그 정보를 이용하여 PingID 명령 반복회수를 줄여 태그인식시간을 단축시킬 수 있다. 제안한 알고리즘의 성능을 수학적으로 분석하고 기존의 알고리즘과 비교하였으며, 수학적 성능분석 결과 제안한 알고리즘의 성능이 월등하게 우수한 것을 확인하였으며, 시뮬레이션을 통하여 그 결과를 검증하였다. 제안한 알고리즘은 기존 알고리즘보다 랜덤한 ID를 갖는 태그를 사용했을 경우 충돌이 발생한 태그의 개수가 20개 일 때 약 70% 정도의 성능 향상이 있었으며, 태그의 개수가 200개 일 때 약 130% 정도의 성능 향상이 있었다. 또한, 기존 알고리즘은 순차적인 태그 ID를 사용하였을 경우 랜덤한 ID를 사용하였을 경우보다 성능이 저하 되었으나 제안한 알고리즘은 순차적인 태그 ID를 사용하였을 경우 랜덤한 ID를 사용하였을 경우보다 최대 약 16% 정도 성능이 향상되었다.

### Abstract

An anti-collision algorithm is very important in the RFID system, because it decides tag identification time and tag identification accuracy. We propose improved anti-collision algorithms using Bin slot in RFID system. In the proposed algorithms, if the reader memorizes the Bin slot information, it can reduce the repetition of unnecessary PingID command and the time to identify tags. If we also use ScrollAllID command in the proposed algorithm, the reader knows the sequence of collided ID bits. Using this sequence, the reader can reduce the repetition of PingID command and tag identification time. We analyze the performance of the proposed anti-collision algorithms and compare the performance of the proposed algorithms with that of the conventional algorithm. We also validate analytic results using simulation. According to the analysis, for the random tag ID, comparing the proposed algorithms with the conventional algorithm, the performance of the proposed algorithms is about 130% higher when the number of the tags is 200. And for the sequential tag ID, the performance of the conventional algorithm decreases. On the contrary, the performance of the proposed algorithm using ScrollAllID command is about 16% higher than the case of using random tag ID.

**Keywords:** Anti-collision 알고리즘, 태그 collision, EPC CLASS 1, Bin 슬롯

## I. 서 론

RFID 기술은 무선 환경에서 여러 개의 물리적 개체를 인식하기 위해 사용되는 센서 네트워크의 한 형태로 사물에 매우 작은 전자 태그를 부착하고 전파를 이용하여 사물의 정보 및 주변 환경정보를 자동으로 관리하며, 이러한 기술을 바탕으로 신정보화 시대를 이루어 향후 IT기술을 선도할 것이다. RFID 시스템에서 리더는 무선채널을 통하여 각각의 태그들과 통신을 하는데, 모든 태그들은 리더가 보낸 신호를 동시에 듣고 리더의 전송

\* 준회원, 아주대학교 (Ajou University)

\*\* 정회원, 아주대학교 (Ajou University)

※ 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크원천기반기술개발사업의 지원에 의한 것이다.

접수일자: 2005년12월15일, 수정완료일: 2005년1월19일

요구에 응답을 한다. 이때 발생하는 collision은 두 가지가 있다. 하나의 리더가 동시에 응답한 여러 개의 태그를 인식해야 하는 문제가 발생 하는데 이를 '태그 collision' 이라 한다<sup>[1][2][3][4]</sup>. 그리고 같은 주파수 대역을 사용하는 두 개 이상의 리더가 하나의 태그를 동시에 인식하려 할 때 발생하는 문제를 '리더 collision' 이라 한다<sup>[5][6]</sup>. 이 두 가지 collision을 효율적으로 해결하기 위한 Anti-collision 알고리즘의 개발은 RFID 시스템의 성능 향상을 위해 매우 중요하다. 본 논문에서는 이 두 가지 collision 중 '태그 collision'을 해결하기 위한 태그 Anti-collision 알고리즘을 제안한다. RFID 시스템 성능의 중요한 척도는 태그를 인식하기 위해 필요한 시간이다. 리더에 의한 전송요구 횟수가 적고 전송 요구 시 태그가 보낸 데이터가 적을수록 태그인식시간이 단축된다. RFID 시스템의 현재 추세는 HF보다 UHF이며, 최근의 제품들은 EPCglobal을 따르는 경향이 많다<sup>[7][8]</sup>. 따라서 본 논문에서는 EPCglobal에서 제안한 CLASS 1 UHF Anti-collision 알고리즘<sup>[8]</sup>의 성능을 분석하고 기존의 방식을 개선하여 보다 좋은 성능을 가지는 Anti-collision 알고리즘을 제안한다. 그리고 제안한 알고리즘과 기존의 CLASS 1 UHF Anti-collision 알고리즘의 성능을 수학적으로 비교 및 분석하였으며 시뮬레이션을 통해 분석된 결과를 검증하였다.

## II. 기존의 EPC CLASS 1 UHF Anti-collision 알고리즘

EPC CLASS 1 UHF Anti-collision 알고리즘은 기본적으로 PingID 명령을 사용하여 Bin 슬롯에 태그의 응답을 받아 태그의 collision을 해결한다. 하나의 PingID 명령 응답 period는 8개의 Bin 슬롯으로 구성되어 있고, 각 Bin 슬롯은 순서대로 '000'부터 '111'까지를 나타낸다. PingID 명령을 사용하는 과정은 다음과 같다. 리더가 PingID 명령을 전송하면 리더의 인식영역 안에 있는 태그 중 active 상태이고, 리더의 명령 패킷의 [PTR] 위치에서 시작하는 [LEN] 길이에 해당하는 [VALUE]값과 일치하는 ID를 가지고 있는 태그들은 ID의 [VALUE]값 다음의 최상위 3 bit를 이용하여 8개의 Bin 슬롯에 응답한다. 따라서, ID의 [VALUE]값 다음의 최상위 3 bit가 '000'인 태그는 Bin 0에 응답을 하고, '111'인 태그는 Bin 7에 응답을 한다. 태그들의 응답을 받은 리더는 Bin

0부터 차례대로 처리한다. 리더는 Bin 슬롯의 정보를 이용하여 하나의 Bin 슬롯에 두 개 이상의 태그가 응답한 경우에는 다시 PingID 명령을 보내고, 하나의 Bin 슬롯에 하나의 태그가 응답한 경우에는 ScrollID 명령을 이용하여 응답한 태그에게 자신의 모든 ID bit를 전송하도록 한다. 따라서, ScrollID 명령에 응답하는 태그는 하나이며, ScrollID 명령을 받은 태그는 명령 패킷의 [PTR] 위치에서 시작하는 [LEN] 길이에 해당하는 [VALUE]값이 자신의 ID값과 일치하면 응답으로 자신의 모든 ID를 리더에게 전송하고, 리더는 그 태그를 인식하게 된다. 하나의 태그를 인식한 리더는 다시 PingID 명령을 전송하게 되고, 위의 과정을 반복하여 충돌이 발생한 모든 태그를 인식하게 된다. 리더가 PingID 명령과 ScrollID 명령을 사용하여 충돌이 발생한 태그를 인식하는 과정을 간단한 예로 설명하면 그림 1과 같다. 보다 자세한 사항은 [8]을 참조하면 알 수 있다.

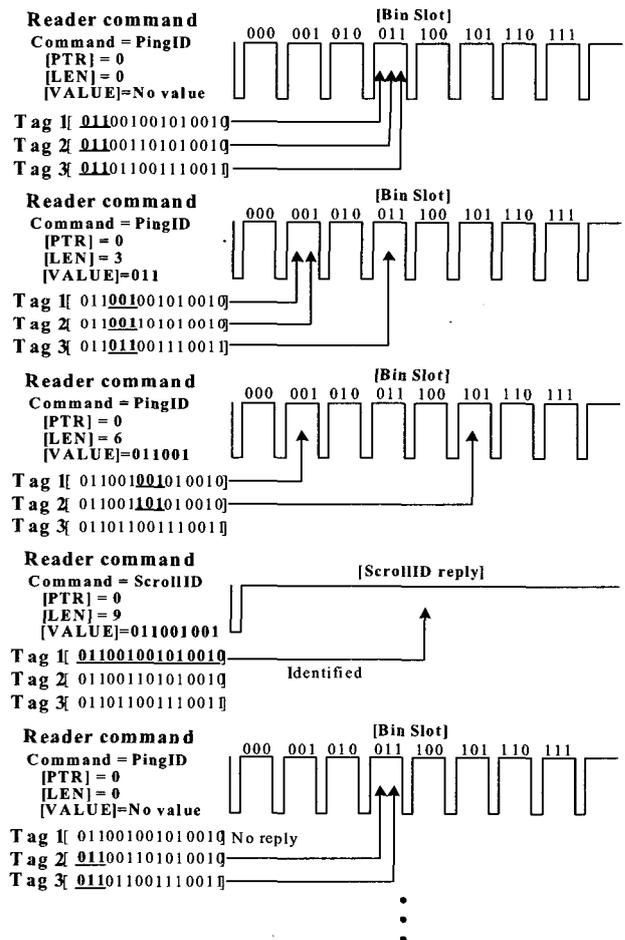


그림 1. EPC CLASS 1 UHF Anti-collision 알고리즘의 태그 인식 과정  
Fig. 1. An example of the conventional Anti-collision algorithm.

### III. 제안한 Anti-collision 알고리즘

기존의 EPC CLASS 1 UHF Anti-collision 알고리즘에서는 리더가 PingID 명령과 ScrollID 명령을 이용하여 하나의 태그를 인식한 후 다음 태그를 인식하기 위해 처음부터 같은 과정을 반복한다. 그러나 만약 리더가 각 단계별로 Bin 슬롯 정보를 기억할 수 있다면 불필요한 PingID 명령의 반복횟수를 줄여 알고리즘의 성능을 향상시킬 수 있다. 제안한 알고리즘의 태그인식 과정을 살펴보면 그림 2와 같다. 그림 1에서는 하나의 태그를 인식한 후 다시 처음부터 PingID 명령을 사용해 다음 태그를 인식한다. 그러나 제안한 방식을 이용한다면 그림 2와 같이 불필요한 명령의 반복을 줄이고 보다 빠른 시간에 많은 태그를 인식할 수 있다.

또한, 기존 알고리즘은 충돌이 발생한 태그를 인식하기 위해 PingID 명령과 ScrollID 명령만을 사용한다. 그러므로 기존 알고리즘을 순차적인 ID를 갖는 태그들을

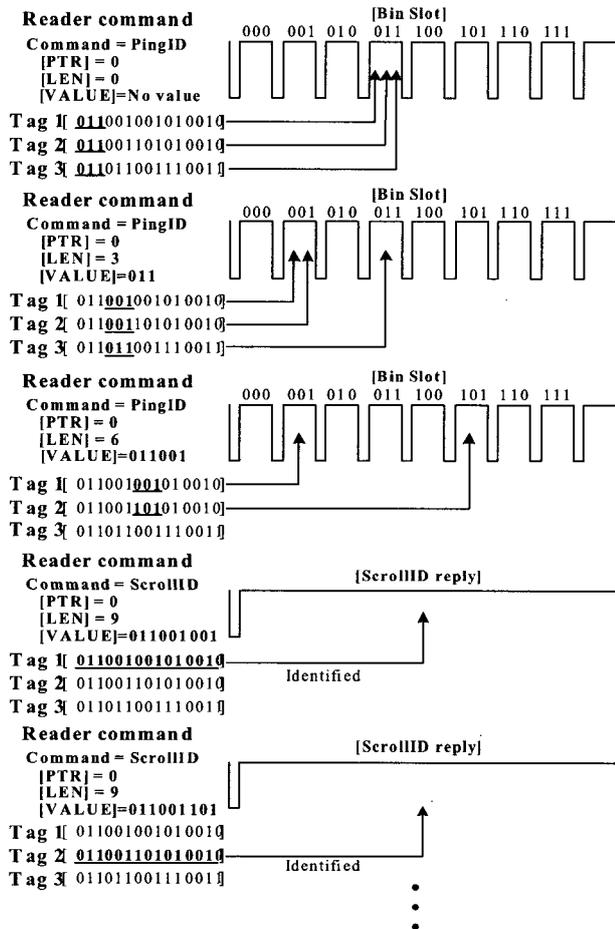


그림 2. 제안한 EPC CLASS 1 UHF용 anti-collision 알고리즘의 태그인식 과정  
Fig. 2. Tag identification procedure of the proposed algorithm for EPC CLASS 1 UHF.

인식하는데 사용한다면 태그 하나하나를 인식하기 위해 계속 PingID 명령을 반복하게 되므로 오히려 랜덤한 태그를 인식할 때보다 성능이 저하된다. 그러나 [8]에 정의되어 있는 ScrollAllID 명령을 사용한다면 태그 ID bit의 충돌 정보를 이용하여 불필요한 PingID 명령의 반복 횟수를 줄이고 알고리즘의 성능을 향상시킬 수 있다. 그 방법을 설명하면 다음과 같다. 우선 리더는 ScrollAllID 명령을 태그들에게 전송한다. ScrollAllID 명령을 받은 태그들은 자신의 모든 ID를 리더에게 전송하고, 이를 수신한 리더는 전송된 ID bit의 충돌 정보를 알 수 있게 된다. 따라서 리더는 태그들로부터 ScrollAllID에 대한 응답을 받은 후 [PTR] field를 최초 충돌이 발생한 ID bit의 위치로 설정하여 PingID 명령을 보낸다. 이 때, 모든 태그가 응답을 하고, 리더는 Bin 슬롯의 정보를 기억하면서 하나의 태그만이 응답할 때까지 PingID 명령을 반복한다. 하나의 태그가 응답을 하면 ScrollID 명령을

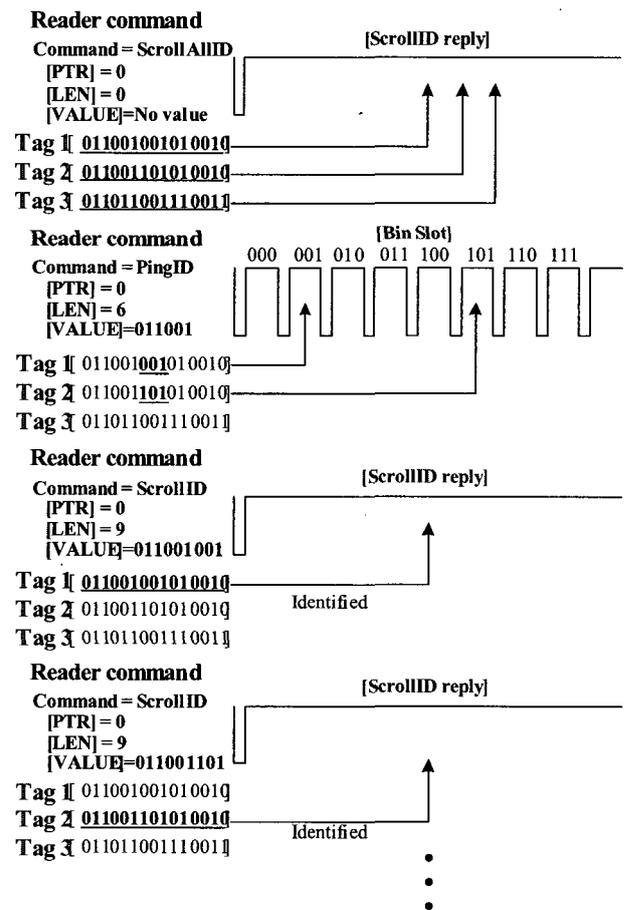


그림 3. ScrollAllID 명령을 사용하였을 경우 제안한 EPC CLASS 1 UHF용 anti-collision 알고리즘의 태그 인식 과정  
Fig. 3. Tag identification procedure of the proposed algorithm for EPC CLASS 1 UHF using ScrollAllID command.

사용하여 응답한 태그를 인식하고, Bin 슬롯 정보를 이용하여 처음 단계로 돌아가 PingID 명령을 반복할 필요 없이 최초 하나의 태그가 인식되기 바로 전 단계로 돌아가 태그 인식과정을 거쳐 다음 태그를 인식하게 되고, 충돌이 발생한 모든 태그를 인식할 때까지 이와 같은 과정을 반복한다. 그림 3은 ScrollAllID 명령을 사용하였을 경우 제안한 알고리즘의 태그 인식과정을 나타낸다.

#### IV. 성능 분석

본 장에서는 기존의 EPC CLASS 1 UHF Anti-collision 알고리즘과 제안한 Anti-collision 알고리즘의 성능을 수학적으로 분석한다. 분석결과로 리더가 전송하는 PingID 명령의 반복횟수와 태그 인식 시간을 도출한다.

##### 1. 기존의 EPC CLASS 1 UHF Anti-collision 알고리즘

태그의 ID는 랜덤하고, 충돌이 발생한 태그의 개수는  $m$ , Bin 슬롯의 개수를  $r$ 이라 할 때, PingID 명령 전송 시 Bin 0에서 Bin  $n-1$ 까지 응답한 태그가 없고 Bin  $n$ 에서 2개 이상의 태그가 응답할 확률을  $p_{1n}$ 이라 정의하면  $p_{1n}$ 은 식 (1)과 같으며

$$P_{1n} = \left(\frac{r-n}{r}\right)^m \left(1 - \left(\frac{r-1-n}{r-n}\right)^m - m \left(\frac{r-1-n}{r-n}\right)^{m-1} \cdot \frac{1}{r-n}\right) \quad (1)$$

이때,  $r$ 은 8이고  $n$ 은  $0 \leq n \leq 7$ 인 정수이다.

따라서, 첫 단계에서 다시 리더가 PingID 명령을 전송할 확률  $p_1$ 은 (2)와 같다.

$$P_1 = \sum_{n=0}^7 \left(\frac{r-n}{r}\right)^m \left(1 - \left(\frac{r-1-n}{r-n}\right)^m - m \left(\frac{r-1-n}{r-n}\right)^{m-1} \cdot \frac{1}{r-n}\right) \quad (2)$$

리더가 두 번째 PingID 명령 전송 시 Bin 0에서 Bin  $n-1$ 까지 응답한 태그가 없고 Bin  $n$ 에서 2개 이상의 태그가 응답할 확률을  $p_{2n}$ 이라 정의하면  $p_{2n}$ 은 식 (3)과 같이 계산된다.

$$P_{2n} = \left(\frac{r-n}{r}\right)^m \left(1 - \left(\frac{r-1-n}{r-n}\right)^m - \frac{m}{r} \left(\frac{r-1-n}{r-n}\right)^{m-1} \cdot \frac{1}{r-n}\right) \quad (3)$$

따라서 두 번째 단계에서 다시 리더가 PingID 명령을 전송할 확률  $p_2$ 는 (4)와 같다.

$$P_2 = \sum_{n=0}^7 \left(\frac{r-n}{r}\right)^m \left(1 - \left(\frac{r-1-n}{r-n}\right)^m - \frac{m}{r} \left(\frac{r-1-n}{r-n}\right)^{m-1} \cdot \frac{1}{r-n}\right) \quad (4)$$

그러므로 최초 하나의 태그를 인식하기 위한 리더의 PingID 명령 전송횟수의 기대값  $I_1$ 은 식 (5)와 같이 계산된다.

$$I_1 = 1 + \sum_{k=0}^{\infty} \sum_{n=0}^7 \left(\frac{r-n}{r}\right)^k \left(1 - \left(\frac{r-1-n}{r-n}\right)^k - \frac{m}{r^k} \left(\frac{r-1-n}{r-n}\right)^{k-1} \cdot \frac{1}{r-n}\right) \quad (5)$$

단,  $\frac{m}{r^k} > 1$

리더가 하나의 태그를 인식하면 남은 태그의 개수는  $m-1$ 이 되고 두 번째 태그를 인식하기 위해서 위의 과정을 처음부터 반복하므로 리더가  $m$ 개의 태그를 모두 인식할 때까지 전송한 PingID 명령의 평균 반복횟수  $I_{total}$ 은 식 (6)과 같이 계산된다.

$$I_{total} = m + \sum_{L=2}^{\infty} \sum_{n=0}^7 \left(\frac{r-n}{r}\right)^L \left(1 - \left(\frac{r-1-n}{r-n}\right)^L - \frac{L}{r^L} \left(\frac{r-1-n}{r-n}\right)^{L-1} \cdot \frac{1}{r-n}\right) \quad (6)$$

단,  $\frac{m}{r^L} > 1$

그리고 리더는  $m$ 개의 태그를 인식하기 위하여  $m$ 번의 ScrollID 명령을 반복하므로 리더가  $m$ 개의 태그를 인식하는데 걸리는 총 시간은 리더가 PingID 명령과 ScrollID 명령을 전송하는데 걸리는 시간과 그에 따른 태그의 응답 시간, 리더와 태그의 지연 시간을 모두 합하여 구할 수 있다. 리더가 태그에게 보내는 패킷의 크기를  $RL\_C$ 라 하고, 리더의 data rate을  $DR_{reader}$ 라 하면, 리더가  $m$ 개의 태그를 인식하는 동안 리더가 태그에게 명령을 전송하는데 걸리는 시간  $t_{reader}$ 는 (7)과 같으며,

$$t_{reader} = \frac{RL\_C(I_{total} + m)}{DR_{reader}} \quad (7)$$

$RL\_C$ 는 147 bit이며,  $DR_{reader}$ 는 70,180 bps 이다.

또한, 리더의 PingID 명령에 의한 태그의 응답 패킷의 크기를  $TL\_P$ 라 하고, ScrollID 명령에 의한 태그의 응답 패킷의 크기를  $TL\_S$ 로 정의하며, 태그의 data rate을  $DR_{tag}$ 라 할 때, 리더가  $m$ 개의 태그를 인식하는 동안 태그가 리더에게 응답을 전송하는데 걸리는 시간  $t_{tag}$ 는 식 (8)과 같으며,

$$t_{tag} = \frac{TL\_P \times I_{total} + TL\_S \times m}{DR_{tag}} \quad (8)$$

$TL_P$ 는 8 bit,  $TL_S$ 는 120 bit 이고,  $DR_{tag}$ 는 140,350 bps이다.

리더의 전송 지연을  $DE_{reader}$ 라 하고, 태그의 전송 지연을  $DE_{tag}$ 라 할 때, 리더가  $m$ 개의 태그를 인식하는 동안의 총 지연시간  $t_{delay}$ 는 식 (9)와 같고,

$$t_{delay} = (I_{total} + m)(DE_{reader} + DE_{tag}) \quad (9)$$

$DE_{reader}$ 는 17.81  $\mu s$ 이고  $DE_{tag}$ 는 57  $\mu s$ 이다.

따라서 리더가  $m$ 개의 태그를 인식하는데 걸리는 총 시간  $t_{total}$ 은 식 (10)과 같다.

$$t_{total} = t_{reader} + t_{tag} + t_{delay} \quad (10)$$

### 2. 제안한 Anti-collision 알고리즘

태그의 ID는 랜덤하고, 충돌이 발생한 태그의 개수를  $m$ 이라 하며, Bin 슬롯의 개수를  $r$ 이라 할 때 최초 리더가 PingID 명령 전송 시 하나의 Bin 슬롯에 2개 이상의 태그가 응답할 확률  $P_{col}$ 은 (11)과 같다.

$$P_{col} = \left( 1 - \left( \frac{r-1}{r} \right)^m - m \left( \frac{r-1}{r} \right)^{m-1} \cdot \frac{1}{r} \right) \quad (11)$$

Bin 슬롯은 모두  $r$ 개이므로 첫 단계에서 리더가 다시 전송할 PingID 명령 개수의 기대값  $I_1$ 은 식 (12)와 같이 계산된다.

$$I_1 = r \times \left( 1 - \left( \frac{r-1}{r} \right)^m - m \left( \frac{r-1}{r} \right)^{m-1} \cdot \frac{1}{r} \right) \quad (12)$$

제안한 알고리즘의 경우 리더가 Bin 슬롯의 정보를 기억하므로 Bin 슬롯들은 트리 구조를 이루게 된다. 따라서 두 번째 단계에서 리더가 다시 전송할 PingID 명령 개수의 기대값  $I_2$ 는 식 (13)과 같다.

$$I_2 = r^2 \times \left( 1 - \left( \frac{r-1}{r} \right)^{\frac{m}{r}} - \frac{m}{r} \left( \frac{r-1}{r} \right)^{\frac{m}{r}-1} \cdot \frac{1}{r} \right) \quad (13)$$

결과적으로 리더가  $m$ 개의 태그를 모두 인식할 때까지 전송한 PingID 명령의 반복횟수  $I_{total}$ 은 식 (14)와 같이 계산되며,

$$I_{total} = \sum_{k=0}^{\infty} I_k = I_0 + \sum_{k=1}^{\infty} r^k \times \left( 1 - \left( \frac{r-1}{r} \right)^{\frac{m}{r^k}} - \frac{m}{r^{k-1}} \left( \frac{r-1}{r} \right)^{\frac{m}{r^k}-1} \cdot \frac{1}{r} \right)$$

단,  $\frac{m}{r^{k-1}} > 1$  (14)

$I_0$ 는 최초 리더의 PingID 명령 전송횟수로 1이다.

제안한 알고리즘의 경우 리더가  $m$ 개의 태그를 인식하는데 걸리는 총 시간  $t_{total}$ 은 (7), (8), (9)와 같은 과정을 거쳐 (10)과 같이 구할 수 있다. ScrollAllID 명령을 사용하는 경우 ScrollID 명령의 수만 1만큼 증가하므로 리더가  $m$ 개의 태그를 인식하는데 걸리는 총 시간  $t_{total}$ 은 (7), (8), (9)의 과정에  $m$ 대신  $m+1$ 을 대입하여 (10)과 같이 구할 수 있다.

### V. 수학적 분석 및 시뮬레이션 결과

본 장에서는 기존의 EPC CLASS 1 UHF Anti-collision 알고리즘과 제안한 알고리즘의 성능을 수학적으로 비교 및 분석하였고, 시뮬레이션을 통해 그 결과를 검증하였다. 태그의 ID는 96 bit이고, 그 중 실제로 태그를 인식하기 위한 부분은 36 bit로 가정하였다. 실제와 동일한 환경의 시뮬레이터 구축을 위해서 시뮬레이터에서 사용된 리더와 태그의 전송 패킷의 크기는 실제 패킷 크기와 동일하게 하였으며, 리더와 태그의 data rate도 실제와 동일하게 하였고 리더와 태그의 패킷 전송 지연을 모두 고려하였다<sup>[8]</sup>.

그림 4는 기존의 EPC CLASS 1 UHF Anti-collision 알고리즘과 제안한 알고리즘, ScrollAllID 명령을 사용하여 제안한 알고리즘에 대한 충돌이 발생한 태그의 개수에 따른 리더의 PingID 명령 전송 반복횟수를 나타낸다. 태그의 ID는 랜덤한 ID를 사용하였으며, 선은 수학적 분석결과를 나타내고, 도형은 시뮬레이션 결과를 나타낸다. 그림 4를 보면 기존의 알고리즘에서 수학적 분석결과와 시뮬레이션 결과가 약간의 차이를 나타낸다.

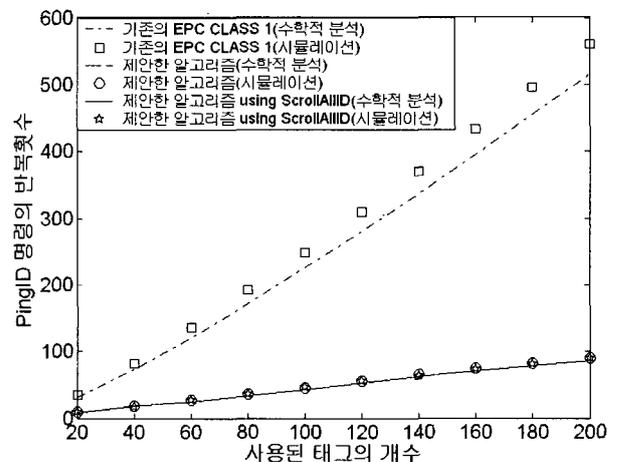


그림 4. 충돌이 발생한 태그 개수에 따른 리더의 PingID 명령의 반복횟수 (랜덤한 태그ID 사용)

Fig. 4. The number of PingID command for the number of used tags (random tag ID).

그 이유는 PingID 명령의 반복횟수를 계산하는 과정에서  $m/r^k \leq 1$  이후에 발생할 수 있는 추가적인 리더의 PingID 명령을 고려하지 않았기 때문이다. 그림 4에서 알 수 있듯이 제안한 알고리즘이 기존의 알고리즘보다 리더의 PingID 전송 반복횟수가 현저히 적다.

그림 5는 위의 세 알고리즘에 대한 충돌이 발생한 태그의 개수에 따른 리더의 태그인식시간을 나타낸다. 태그의 ID는 랜덤한 ID를 사용하였으며, 선은 수학적 분석결과를 나타내고, 도형은 시뮬레이션 결과를 나타낸다. 그림 5에서 알 수 있듯이 제안한 알고리즘이 기존의 알고리즘보다 태그인식시간에 있어 태그의 개수가 20개 일 때 약 70% 정도의 성능 향상이 있었으며, 태그의 개수가 200개 일 때 약 130% 정도의 성능 향상이 있었다. 초당 태그인식속도로 계산하면 기존의 알고리즘의 경우 초당 약 117개의 태그 인식이 가능하며, 제안한 알고리즘의 경우 초당 약 252개의 태그 인식이 가능하다.

그림 6은 위의 세 알고리즘에 대해 순차적인 태그 ID를 사용하였을 경우 충돌이 발생한 태그의 개수에 따른 리더의 PingID 명령 전송 반복횟수를 나타내며 이는 시뮬레이션 결과이다. 그림 6에서 기존의 알고리즘은 순차적인 ID를 갖는 태그에 대해서 오히려 리더의 PingID 명령 전송 반복횟수가 증가한 것을 알 수 있다. 순차적인 태그를 사용할 경우 리더가 PingID 명령 전송 시 같은 Bin 슬롯으로 응답하는 태그가 많기 때문에 하나의 태그 인식 후 다시 처음부터 PingID 명령 전송을 반복하는 기존의 알고리즘의 경우 PingID 명령 전송 반복횟수가 현저히 증가하였다. 반면에 제안한 알고리즘의 경우에는 PingID 명령 전송 반복횟수가 줄어든 것을 알

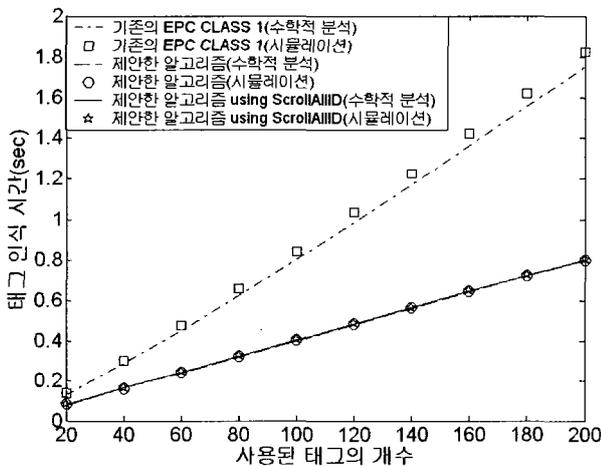


그림 5. 충돌이 발생한 태그 개수에 따른 리더의 태그인식시간 (랜덤한 태그 ID 사용)  
Fig. 5. Tag identification time for the number of used tags (random tag ID).

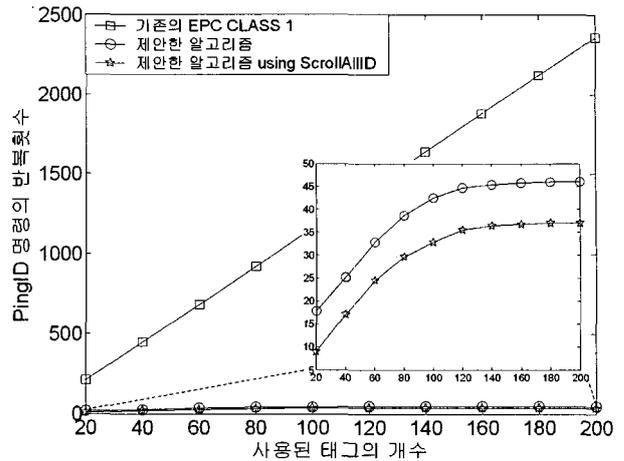


그림 6. 충돌이 발생한 태그 개수에 따른 리더의 PingID 명령의 반복횟수 (순차적인 태그 ID 사용)  
Fig. 6. The number of PingID command for the number of used tags (sequential tag ID).

수 있다. 제안한 알고리즘의 경우 리더가 Bin 슬롯 정보를 기억하기 때문에 불필요한 PingID 명령의 반복을 줄일 수 있다. 그리고 ScrollAIIID 명령을 사용할 경우 리더는 충돌이 발생한 태그들의 ID bit의 충돌 정보를 알 수 있으므로, 최초 충돌이 발생한 위치부터 PingID 명령 전송을 시작하므로 순차적인 태그 ID를 사용할 경우 PingID 명령 전송 반복횟수를 더 줄일 수 있다.

그림 7은 앞에서 비교한 세 알고리즘에 대해 순차적인 태그 ID를 사용하였을 경우 충돌이 발생한 태그의 개수에 따른 리더의 태그인식시간을 나타내며 이는 시뮬레이션 결과이다. 그림 7에서 나타난 바와 같이 기존 알고리즘은 순차적인 태그 ID를 사용하였을 경우 랜덤

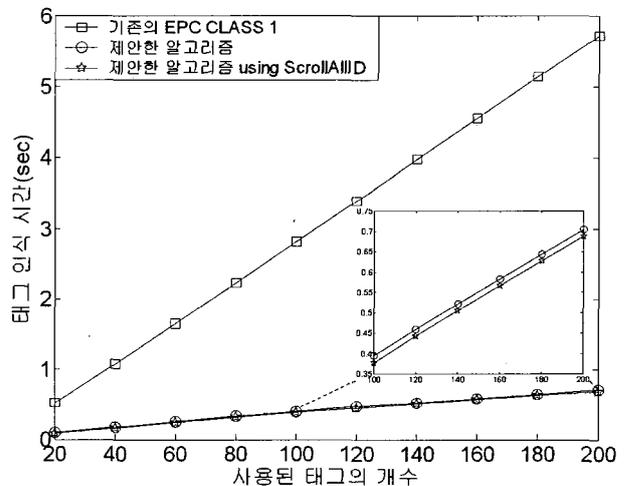


그림 7. 충돌이 발생한 태그 개수에 따른 리더의 태그인식시간 (순차적인 태그 ID 사용)  
Fig. 7. Tag identification time for the number of used tags (sequential tag ID).

한 태그 ID를 사용하였을 경우보다 태그 인식시간이 증가하였다. 제안한 알고리즘은 순차적인 태그 ID를 사용하였을 경우 랜덤한 ID를 사용하였을 경우보다 200개의 충돌이 발생한 태그에 대해서 태그인식시간에 있어 약 13% 정도 성능이 향상되었으며, ScrollAllID 명령을 사용한 경우에는 약 16% 정도 성능이 향상되었다.

## VI. 결 론

본 논문에서는 EPC CLASS 1 UHF Anti-collision 알고리즘의 성능을 분석하였고, 향상된 성능을 보이는 개선된 Anti-collision 알고리즘을 제안하였다. 제안한 알고리즘을 수학적으로 성능을 분석하였으며, 시뮬레이션을 통해 그 수학적 분석의 결과를 검증하였다. 제안한 알고리즘은 기존 알고리즘보다 랜덤한 ID를 갖는 태그를 사용했을 경우 충돌이 발생한 태그의 개수가 20개 일 때 약 70% 정도의 성능 향상이 있었으며, 태그의 개수가 200개 일 때 약 130% 정도의 성능 향상이 있었다. 또한, 현재 출시되고 있는 하나의 안테나와 리더에 의해서는 기존의 알고리즘의 경우 초당 약 117개의 태그 인식이 가능하나 제안한 알고리즘의 경우 초당 약 252개의 태그 인식이 가능하므로 알고리즘의 성능이 월등히 향상된 것을 알 수 있다. 그리고 RFID 시스템에서는 순차적인 ID를 갖는 태그들을 사용할 경우가 빈번히 발생한다. 그러한 경우에 기존 알고리즘을 사용하면

PingID 명령의 많은 반복으로 인하여 알고리즘의 성능이 저하되나, 제안한 알고리즘을 사용할 경우 랜덤한 ID를 사용할 경우보다 약 13% 정도 성능이 향상되었으며, ScrollAllID 명령을 사용한 경우에는 약 16% 정도 성능이 향상되었다. 따라서, 제안한 알고리즘을 RFID 시스템에 적용한다면 같은 시간 내에 더 많은 태그를 인식할 수 있으므로 RFID 시스템 성능 향상에 크게 기여할 수 있을 것이다.

## 참 고 문 헌

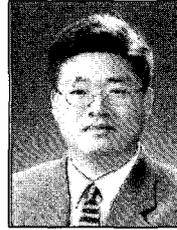
- [1] H. Vogt, "Efficient Object Identification with Passive RFID tags," *International Conference on Pervasive Computing*, pp.98-113, Zurich, 2002.
- [2] M. Jacomet, A. Ehram, and U. Gehrig, "Contactless Identification Device With Anticollision Algorithm," *IEEE Computer Society CCCC99*, pp.269-273, Jul. 4-8, Athens 1999.
- [3] H. S. Choi, J. R. Cha and J. H. Kim, "Fast Wireless Anti-collision Algorithm in Ubiquitous ID System," in *Proc. IEEE VTC 2004*, L.A., USA, Sep., 26-29, 2004.
- [4] 최호승, 김재현, "RFID 시스템에서의 태그 인식 알고리즘 성능분석," *전자공학회논문지*, 제 42권, 제 5호, pp.47-54, 2005년 5월.
- [5] D.W. Engels and S.E. Sarma, "The reader collision problem," 2002 IEEE International Conference on Systems, Man and Cybernetics, Vol.3, pp.6, Oct. 6-9, 2002.
- [6] S. Sarma, J. Waldrop, and D. Engels, "Colorwave : An Anti-collision Algorithm for the Reader Collision Problem," IEEE International Conference on Communications, ICC '03, vol. 2, pp.1206-1210, May 2003.
- [7] Auto-ID Center, *Draft Protocol Specification for a Class 0 Radio Frequency Identification tag.*, 2003.
- [8] EPCglobal, *EPC™ Tag Data Standards Version 1.1 Rev.1.24*, Apr. 2004.

## 저 자 소 개



최 호 승 (준회원)  
 2004년 아주대학교 정보통신대학  
 전자공학부 학사  
 2004년 3월~현재 아주대학교  
 전자공학부 석사 재학 중

<주관심분야 : 통신, RFID, MAC protocol>



김 재 현 (정회원)  
 1991년 한양대학교 전자계산학과  
 학사  
 1993년 한양대학교 전자계산학과  
 공학석사  
 1996년 8월 한양대학교 전자계산  
 학과 공학박사

1997년 7월~1998년 6월 UCLA 전기과 Postdoc  
 연구원

1997년 7월~1998년 9월 IRI Corp. CA, USA

1998년 11월~2003년 2월 Bell Labs, Lucent Tech.

2003년 3월~현재 아주대학교 정보통신대학  
 전자공학부 조교수

<주관심분야 : 무선 인터넷 QoS, MAC 프로토콜,  
 IEEE 802.11/15/16/20>