

논문 2005-42TC-5-7

RFID 시스템에서의 태그 인식 알고리즘 성능분석

(Performance Analysis of Tag Identification Algorithm in RFID System)

최 호 승*, 김 재 현**

(Ho-Seung Choi and Jae-Hyun Kim)

요 약

본 논문은 RFID 시스템에서의 태그 Anti-collision 알고리즘을 제안하고 분석한다. 제안한 RFID 시스템에서의 Anti-collision 알고리즘과 기존의 이진 방식 알고리즘들(이진 탐색 알고리즘, time slot을 이용한 slotted 이진 트리 알고리즘, Auto-ID 센터에서 제안한 bit-by-bit 이진 트리 알고리즘)을 수학적으로 비교하고 분석하였다. 수학적 분석 결과는 OPNET 모의실험을 통하여 그 결과를 검증하였다. 분석 결과에 의하면 제안한 Improved bit-by-bit 이진 트리 알고리즘의 성능이 기존의 Anti-collision 알고리즘 중 가장 좋은 성능을 보이는 bit-by-bit 이진 트리 알고리즘과 비교할 때 리더의 전송요구에 응답한 태그의 개수가 20개일 경우에는 약 304% 정도의 성능향상이 있었으며 리더의 전송요구에 응답한 태그의 개수가 200개일 경우에는 839%의 성능향상이 있었다.

Abstract

This paper proposes and analyzes a Tag Anti-collision algorithm in RFID system. We mathematically compare the performance of the proposed algorithm with existing binary algorithms(binary search algorithm, slotted binary tree algorithm using time slot, and bit-by-bit binary tree algorithm proposed by Auto-ID center). We also validated analytic results using OPNET simulation. Based on analytic result, comparing the proposed Improved bit-by-bit binary tree algorithm with bit-by-bit binary tree algorithm which is the best of existing algorithms, the performance of Improved bit-by-bit binary tree algorithm is about 304% higher when the number of tags is 20, and 839% higher when the number of tags is 200.

Keywords : RFID system, Anti-collision algorithm, Binary Tree algorithm, and Improved bit-by-bit binary tree algorithm

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 기술을 기반으로

구축될 유비쿼터스 센서 네트워크 상에는 일상의 사물들, 상품들, 기업의 물류, 생산, 판매 등의 사업을 구성하는 시스템들이 모두 연결됨으로서 원가절감, 생산성 및 부가가치 향상 등의 큰 기대효과를 얻을 수 있다. 그 외에도 RFID 기술은 전장, 병원, 가정집, 사무실, 학교 등 여러 장소에서 편의를 제공하고, 보다 효율적인 업무를 할 수 있도록 도와줄 것이다. RFID 시스템에서 리더는 무선채널을 통하여 각각의 태그들과 통신을 하는데, 모든 태그들이 리더가 보낸 신호를 동시에 듣게 되고 리더의 전송요구에 응답을 한다. 이 때 발생하는 collision은 두 가지가 있다. 하나의 리더가 동시에 응답한 여러 개의 태그를 인식해야하는 문제가 발생하는데 이를 '태그 collision'이라 한다^{[1]-[3]}. 또한 같은 주파

* 준회원, ** 정회원, 아주대학교
(AJOU University)

※ "이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음."(KRF-2004-003-D00205)

접수일자: 2004년11월26일, 수정완료일: 2005년5월13일

수 대역을 사용하는 두 개 이상의 리더가 하나의 태그를 동시에 인식하려할 때 발생하는 문제를 '리더 collision' 이라 한다. 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 두 가지 중요한 척도는 태그를 인식하기 위해 필요한 시간과 태그에 의해 소모된 에너지이다. 리더에 의해 전송요구가 있을 때 태그가 보낸 데이터가 적을수록 태그인식시간이 단축되고 태그에 의해 소모된 에너지도 적게 된다. 본 논문에서는 제안한 RFID 시스템에서의 Improved bit-by-bit 이진 트리 알고리즘과 이진 탐색 알고리즘^[4], time slot을 이용한 slotted 이진 트리 알고리즘^[5], 그리고 Auto-ID 센터에서 제안한 bit-by-bit 이진 트리 알고리즘^{[6]-[8]}을 수학적으로 비교 및 분석하였으며 모의실험을 통해 분석된 결과를 검증하였다.

II장에서는 기존의 Anti-collision 알고리즘을 설명하고, III장에서는 제안한 Anti-collision 알고리즘을 설명한다. IV장에서는 제안한 알고리즘을 분석하고 성능평가를 위한 반복회수와 태그가 보낸 총 bit수를 구한다. V장에서는 수학적 분석 및 모의실험을 통해 기존의 알고리즘과 제안한 알고리즘의 성능을 비교 및 분석하며 VI장에서 결론을 맺는다.

II. 기존 Anti-collision 알고리즘

1. 이진 탐색 기반 알고리즘

가. 이진 탐색 알고리즘^[4]

이진 탐색 알고리즘은 collision이 발생한 부분을 점차 줄임으로써 전송 가능한 태그의 수를 줄여 collision을 해결하는 방식이다. 리더는 인식 가능한 영역에 있는 모든 태그의 ID를 수신하여 collision이 일어나는 bit의 위치를 파악한다. 그 중 collision이 발생한 최상위 bit가 1인 태그는 전송이 지연되고, 0인 태그는 ID를 전송한다. 이런 과정을 순차적으로 반복 수행함으로써 하나의 태그를 인식한다.

예를 들어 리더가 표 1의 ID가 4 bit인 4개의 태그를 인식하기 위해 REQUEST(≤ 1111) 명령을 전송하면 ID가 '1111'보다 작거나 같은 값을 가지고 있는 태그들은 모두 자신의 ID를 전송한다. 리더는 태그로부터 수신된 ID의 시퀀스를 메모리에 저장한다. 수신한 시퀀스 중

표 1. 사용된 태그의 ID

Table 1. IDs of the used tags.

태그 번호	태그 ID
1	0001
2	0010
3	1010
4	1011

첫 번째 bit가 collision이 발생하였으므로 리더는 검색 범위를 ' ≤ 0111 '로 설정하여 REQUEST 명령을 전송하게 되고 하나의 태그가 남을 때까지 이런 과정을 반복함으로써 n 개의 태그중 하나를 인식한다. n 개의 태그가 존재한다고 할 때 하나의 태그를 인식하기 위한 이진 탐색 알고리즘의 반복회수(I_{BBS})는 식 (1)과 같다.

$$I_{BBS} = \frac{\log(n)}{\log(2)} + 1 \quad (1)$$

나. 동적 이진 탐색 알고리즘^[4]

이진 탐색 알고리즘에서 태그는 항상 리더에게 모든 ID bit를 전송한다. 따라서 ID의 길이가 길어지면 하나의 태그를 인식하기 위해 전송되는 데이터의 양은 증가한다. 이는 하나의 태그를 처리하는 시간을 증가시킬 뿐만 아니라 태그의 에너지 소모량을 증가시키는 결과를 가져온다. 동적 이진 탐색 알고리즘은 이를 보완하기 위한 방법으로써, 리더는 REQUEST 명령 외에 VB(Valid Bit)를 추가하여 전송한다. VB는 collision이 발생하는 bit의 위치를 말한다. 예를 들어 리더가 받은 시퀀스가 'X0XX'이면, 리더는 collision이 발생하는 최상위 bit의 위치를 전송한다. 시퀀스 'X0XX'에서 'X'는 collision이 발생했음을 의미한다. 이때 그 위치의 값이 1인 태그는 ID의 전송이 지연되고 0인 태그들은 ID 중 VB까지의 bit를 제외한 나머지 bit를 전송한다. 이와 같은 방식으로 반복과정에서 전송되는 데이터의 양을 줄여 태그의 에너지 소비와 태그를 인식하기 위한 탐색시간을 줄인다. 동적 이진 탐색 알고리즘에서 하나의 태그를 인식하기 위한 반복회수는 이진 탐색 알고리즘과 같다. 그러나 데이터의 총 전송량은 이진 탐색 알고리즘과 비교할 때 50%까지 줄일 수 있기 때문에 하나의 태그를 인식하기 위한 탐색시간을 줄일 수 있다.

2. Slotted 이진 트리 알고리즘

가. Basic slotted 이진 트리 알고리즘^[5]

리더가 태그에게 전송요구를 하면 응답한 태그들은

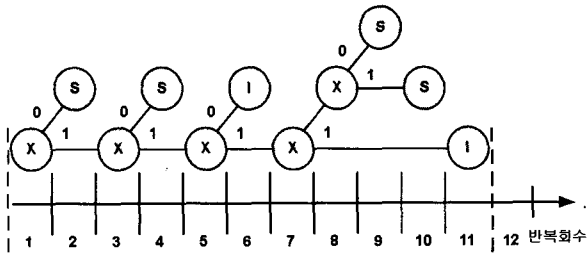


그림 1. Slotted 이진 트리 알고리즘의 예
Fig. 1. An example of Basic slotted binary tree algorithm.

랜덤하게 0과 1을 선택함으로써 두개의 그룹으로 나누어진다. 만일 리더가 i 번째 슬롯에서 태그에게 전송요구를 하면 모든 태그들은 i 번째 슬롯에서 자신의 ID를 전송한다. 리더의 전송 요구에 응답한 태그 중 0을 선택한 그룹의 태그들은 $i+1$ 번째 슬롯에서 전송을 시도하고 1을 선택한 그룹의 태그들은 0을 선택한 그룹의 태그들이 모두 성공적으로 ID를 전송할 때까지 기다리게 된다. $i+1$ 번째 슬롯이 idle 슬롯이거나 성공적으로 전송을 하게 되면 1을 선택한 두 번째 그룹의 태그들은 $i+2$ 번째 슬롯에서 재전송을 하게 된다. 여기서 idle 슬롯이란 태그로부터의 전송이 없는 슬롯을 의미하며, 리더의 전송요구에 응답한 태그가 모두 0 또는 1을 선택할 때 발생할 수 있다. 그러나 $i+1$ 번째 슬롯에서 또다시 collision이 발생하게 되면 다시 랜덤하게 0 또는 1을 선택해서 또 다른 두개의 하위 그룹으로 나누어진다. 이런 과정을 반복적으로 수행함으로써 모든 collision을 해결할 수 있게 된다. 그림 1은 표 1에 있는 4개의 태그를 인식하기 위한 Basic slotted 이진 트리 알고리즘의 진행순서를 나타낸다.

그림 1에서, 'X'는 리더의 요구에 응답한 태그의 개수가 2개 이상이어서 collision이 발생했음을 의미하고, 'I'는 idle 슬롯을 의미하며, 'S'는 리더의 요구에 응답한 태그의 개수가 1개이어서 성공적인 전송이 이루어졌음을 의미한다. Slotted 이진 트리 알고리즘에서 n 개의 태그를 해결하기 위한 반복회수(I_{BSBT})는 식 (2)와 같다.

$$I_{BSBT} = 1 + \sum_{k=2}^n \binom{n}{k} \frac{2(k-1)(-1)^k}{[1-p^k - (1-p)^k]} \quad (2)$$

이 때, p 는 태그가 0 또는 1을 선택할 확률로 0.5이다.

나. Modified slotted 이진 트리 알고리즘^[5]

Basic slotted 이진 트리 알고리즘에서 각 태그들이 0 또는 1을 선택할 때 모두 0을 선택하거나 1을 선택하게

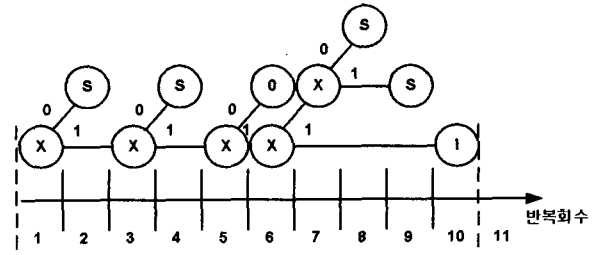


그림 2. Slotted 이진 트리 알고리즘의 예
Fig. 2. An example of Modified slotted binary tree algorithm.

되면 idle상태가 존재하게 되는데, Modified slotted 이진 트리 알고리즘은 ternary feedback을 이용하여 idle 슬롯을 제거함으로써 collision 해결 과정을 좀 더 빠르게 할 수 있다^[5]. 그림 2는 표 1에 있는 4개의 태그를 인식하기 위한 Modified slotted 이진 트리 알고리즘의 진행순서를 나타낸다.

그림 2에서는 그림 1에서의 idle 슬롯이 제거되어 반복회수가 1회 줄어드는 것을 확인할 수 있다. Modified slotted 이진 트리 알고리즘에서 n 개의 태그를 해결하기 위한 반복회수(I_{MSBT})는 식 (3)과 같다.

$$I_{MSBT} = 1 + \sum_{k=2}^n \binom{n}{k} \frac{(-1)^k [k(1+p) - 1 - p^k]}{[1-p^k - (1-p)^k]} \quad (3), n \geq 2$$

이때, p 는 Basic slotted 이진 트리 알고리즘과 동일하다.

3. Bit-by-bit 이진 트리 알고리즘^{[6]-[8]}

Auto-ID 센터에서 제안한 bit-by-bit 이진 트리 알고리즘에서, 리더가 인식할 수 있는 영역내의 모든 태그들에게 ID 중 원하는 bit를 순서대로 요청하게 되면 모든 태그들은 리더의 요구에 대한 응답으로 0 또는 1을 전송한다. collision이 발생 하지 않으면 리더는 태그로부터 받은 bit를 메모리에 저장한 후 다음 bit를 요청하게 된다. 그러나 collision이 발생하면 리더는 0 또는 1을 가진 그룹 중에서 알고리즘에 의해 하나의 그룹을 선택하고 다음 bit를 요청하게 된다. 예를 들면, 리더가 인식할 수 있는 범위 내의 모든 태그에게 ID의 k 번째 bit 전송요구(k 의 초기값은 1)를 하게 되면 모든 태그들은 리더의 요구에 대한 응답으로 k 번째 bit를 전송하게 된다. 태그로부터 받은 k 번째 bit가 collision이 발생하지 않으면 k 번째 bit를 메모리에 저장한 후 다음 bit의 전송을 요구한다. 그러나 collision이 발생하면 메모리에

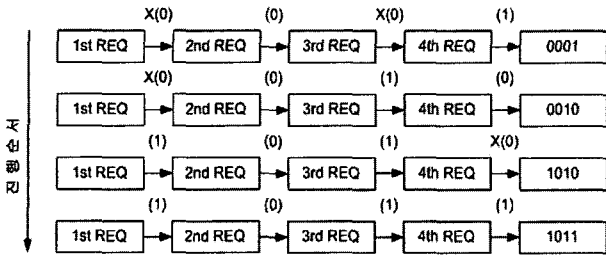


그림 3. bit-by-bit 이진 트리 알고리즘의 예
Fig. 3. An example of bit-by-bit binary tree algorithm.

임의로 k 번째 bit를 0으로 저장한 후 다음 bit의 전송요구와 함께 k 번째 bit가 1인 모든 태그를 inactive 상태로 만든다. (inactive 상태 : 알고리즘 수행 중 collision이 발생했을 때 collision이 발생한 bit가 1인 태그들이 일시적으로 리더의 다음 bit 전송 요구에 응답하지 않는 대기상태, 하나의 태그가 인식되면 리더는 inactive 상태의 태그들을 리더의 전송요구에 응답할 수 있는 active 상태로 만든다). 이런 과정을 태그의 ID길이만큼 반복함으로써 하나의 태그를 인식하게 된다.

그림 3은 bit-by-bit 이진 트리 알고리즘을 사용하여 표 1에 있는 4개의 태그를 인식하기 위한 과정을 나타낸다. bit-by-bit 이진 트리 알고리즘을 사용하여 표 1에 있는 4개의 태그를 인식하기 위한 반복회수는 $16(4 \times 4 \text{ bit})$ 이다. 만일 n 개의 태그가 존재하고 각 태그의 ID가 j bit이라면 모든 태그를 인식하기 위해 필요한 반복회수(I_{BBT})는 식 (4)와 같다.

$$I_{BBT} = n \times j \quad (4)$$

III. 제안한 Anti-collision 알고리즘

Bit-by-bit 이진 트리 알고리즘^{[6]-[8]}의 단점은 리더가 항상 태그 ID의 모든 bit를 요구한다는 것이다. 본 논문에서 제안한 RFID 시스템에서 Anti-collision 알고리즘은 이러한 문제점을 해결하고 개선함으로써 태그인식과정을 단축시켰다. 리더는 우선 인식 가능한 모든 태그에게 모든 ID bit의 전송을 요구한다. 태그들이 리더에게 ID bit을 전송할 때 전송한 bit가 모두 1(0)이면 리더는 메모리에 1(0)을 저장하고, collision이 발생할 경우 메모리에 'X'를 저장한다. 따라서, 리더는 태그들에게 모든 ID bit를 전송받고 그 시퀀스를 메모리에 저장한다. 초기에 모든 ID bit의 전송을 마친 후 리더는 태그에게 bit-by-bit 방식으로 collision이 발생한 bit만을 요구하여 태그를 인식하게 된다. 리더는 태그 인식

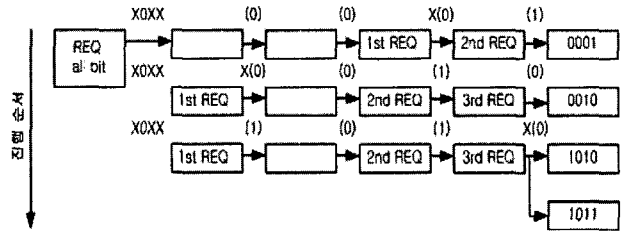


그림 4. Improved bit-by-bit 이진 트리 알고리즘의 예
Fig. 4. An example of Improved bit-by-bit binary tree algorithm.

과정 중 inactive 상태인 태그가 없는 경우에 '0(1)'을 전송받게 되면 인식되지 않은 모든 태그들의 해당 시퀀스의 ID bit가 '0(1)'인 것을 알 수 있으므로 해당 시퀀스를 'X'에서 '0(1)'로 바꾸게 된다. 그러므로 다음에 인식되는 태그는 보다 적은 반복회수로 인식이 가능하게 되고, 같은 원리로 그 다음에 인식되는 태그들은 점차 적은 반복회수를 갖게 된다. 만일 리더가 태그에게 마지막 bit 전송요구를 했을 때 collision이 발생하면 ID 중 마지막 bit만 다른 두 개의 태그가 존재한다는 것을 알 수 있으므로 알고리즘에 의해 동시에 두 개의 태그를 인식할 수 있다. 그림 4는 표 1에 있는 4개의 태그를 인식하기 위한 제안한 Improved bit-by-bit 이진 트리 알고리즘의 진행순서를 나타낸다.

표 1과 같은 4개의 태그를 인식하기 위해서 그림 4와 같이 우선 리더는 사용된 모든 태그들에게 모든 ID bit의 전송을 요청한다. 리더는 태그들에게 k 번째 bit을 전송받을 때 그 bit들이 모두 0(1)이면 그 위치의 bit를 0(1)으로 저장하고, collision이 발생할 경우 구분하기 위해 X로 표시한다. 따라서 리더에 저장된 bit정보는 X0XX가 된다. 첫 번째, 세 번째, 네 번째 bit은 collision이 발생한 것이고, 2번째 bit은 0이다. 리더는 모든 태그들의 ID bit을 전송 받아 collision 상태를 파악한 후 첫 번째 collision이 발생한 bit부터 마지막 collision이 발생한 부분까지 bit-by-bit 방식으로 태그로부터 ID bit를 받고 bit를 받는 중 inactive 상태인 태그가 없는 경우에 collision이 발생하지 않으면 그 위치의 상태를 X상태에서 전송 받은 bit로 바꾸어 저장한 후 태그들에게 다음으로 collision이 발생한 bit를 요구한다. 따라서 첫 번째로 인식되는 태그(0001)는 리더로부터 bit 전송요구를 2번 받는다. 이 때, 첫 번째 collision이 발생한 bit는 리더에서 0으로 인식하여 태그에게 bit 전송 요구를 하지 않는다. 두 번째와 세 번째로 인식되는 태그(0010, 1010)는 리더로부터 bit 전송요구를 3번 받는다. 여기서, ID가 '1010'인 태그와 '1011'

인 태그는 ID 중 마지막 bit만 다르기 때문에 리더가 마지막 bit 전송요구를 했을 때 태그로부터 수신된 데이터가 collision이 발생하면 리더는 재전송 요구 없이 두 개의 태그를 동시에 인식한다. 그러므로 모든 태그를 인식하기 위한 리더의 전송 회수는 9이고, 태그의 전송 회수는 12이다. 태그의 총 전송 회수가 리더의 전송 회수보다 3이 큰 이유는 초기에 리더가 모든 태그에게 자신의 모든 ID bit을 전송을 요구하기 때문에 태그는 자신의 (ID bit수 - 1)만큼 리더보다 더 많은 전송 회수를 갖는다. 그림 3의 Bit-by-bit 이진 트리 알고리즘과 그림 4의 Improved bit-by-bit 이진 트리 알고리즘을 비교할 때 Improved bit-by-bit 이진 트리 알고리즘에서 태그의 전송 회수는 4만큼 감소했으며 리더의 전송 회수가 7만큼 감소했고, 리더가 태그에게 보내는 전송 bit수가 태그가 리더에게 보내는 전송 bit수보다 훨씬 크다는 점을 고려하면 Bit-by-bit 이진 트리 알고리즘보다 확실히 성능이 향상된 것을 알 수 있다. 또한, 사용된 태그의 ID가 순차적인 경우에는 collision이 발생하는 bit의 수가 줄어들기 때문에 성능이 더욱 향상된다. 자세한 성능분석은 IV장에서 기술하였다.

IV. 성능 분석

본장에서는 제안한 Improved bit-by-bit 이진 트리 알고리즘의 성능을 분석하기 위하여 반복회수와 태그가 전송한 총 bit수를 계산한다. 태그 ID의 길이는 36 bit^[6]라고 가정한다.

1. 제안한 알고리즘의 반복회수 분석

순차적인 n 개의 ID를 갖는 태그들 중 임의의 m 개의 태그들이 사용된다고 가정하고, 총 태그 ID는 36 bit로 가정할 때, n 의 범위는 $2^r < n \leq 2^{(r+1)}$ (단, $0 \leq r \leq 35$, r 은 정수)이고, Φ 를 마지막 bit만 collision이 발생한 태그들을 고려하지 않았을 경우의 반복회수라 하면 Φ 는 식 (5)와 같다.

$$\begin{aligned} \Phi = & \sum_{k=1}^{\log_2 m \left(1 - \frac{2^r}{n}\right)} \frac{m \left(1 - \frac{2^r}{n}\right)}{2^k} (r+1 - k) \\ & + \left(\frac{2^r}{n} \times m + 1\right)(r+1) \\ & + \left(\frac{m \left(1 - \frac{2^r}{n}\right)}{2^{k_{\max}}} - 1\right)(r - k_{\max}) \end{aligned} \quad (5)$$

이 때, k_{\max} 는 $\log_2 m \left(1 - \frac{2^r}{n}\right)$ 보다 작거나 같은 최댓값이 된다. 따라서, 마지막 bit만 collision이 발생한 태그들을 고려했을 경우의 반복회수는 Φ 에 마지막 bit만 collision이 발생한 태그들을 고려한 틈을 곱하여 나타낼 수 있다.

제안한 알고리즘의 반복회수는 태그의 개수가 짝수일 때와 홀수일 때로 구분하여 분석하였다. 그리고 태그의 개수가 짝수일 때와 홀수일 때 각각에 대해 태그의 개수가 총 태그개수의 50%이하일 때와 초과일 때로 구분하여 분석하였다.

가. 태그의 개수가 짝수일 경우

ID가 순차적인 n 개의 태그 중 사용된 태그의 개수 (m)가 총 태그 개수의 50%이하($0 < m \leq \frac{n}{2}$)일 때, 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 확률 p_k 를 구하면 식 (6)과 같다.

$$p_k = \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2} - k}{m - 2k} 2^{(m-2k)}}{\binom{n}{m}}, 0 \leq k \leq \frac{m}{2} \quad (6)$$

그리고 ID 중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 때 전송 회수는 $(m-k)$ 번이다. 따라서 사용된 태그의 개수가 짝수이고 총 태그개수의 50%이하 일 때 m 개의 태그를 인식하기 위한 반복회수 (I_{IBBT})는 식 (7)과 같다.

$$\begin{aligned} I_{IBBT} = & \Phi \times \frac{1}{m} \sum_{k=0}^{\frac{m}{2}} p_k (m - k) \\ = & \Phi \times \frac{1}{m} \sum_{k=0}^{\frac{m}{2}} \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2} - k}{m - 2k} 2^{(m-2k)}}{\binom{n}{m}} \\ & \cdot (m - k) \end{aligned} \quad (7)$$

ID가 순차적인 n 의 태그 중 사용된 태그의 개수 (m)가 총 태그 개수의 50%를 초과($\frac{n}{2} < m \leq n$)할 때, 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 확률 p_k 를 계산하면 식 (8)과 같다.

$$p_k = \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2}-k}{k-m+\frac{n}{2}} 2^{(m-2k)}}{\binom{n}{m}}, m - \frac{n}{2} \leq k \leq \frac{m}{2} \quad (8)$$

ID 중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 때 전송 회수는 $(m-k)$ 번이다. 따라서 사용된 태그의 개수가 짝수이고 총 태그 개수의 50%를 초과할 때 m 개의 태그를 인식하기 위한 반복회수 (I_{IBBT})는 식 (9)와 같다.

$$\begin{aligned} I_{IBBT} &= \Phi \times \frac{1}{m} \sum_{k=m-\frac{n}{2}}^{\frac{m}{2}} p_k (m-k) \\ &= \Phi \times \frac{1}{m} \sum_{k=m-\frac{n}{2}}^{\frac{m}{2}} \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2}-k}{k-m+\frac{n}{2}} 2^{(m-2k)}}{\binom{n}{m}} \cdot (m-k) \end{aligned} \quad (9)$$

나. 태그의 개수가 홀수일 경우

ID가 순차적인 n 개의 태그 중 사용된 태그의 개수 (m)가 총 태그 개수의 50%미만($0 < m < \frac{n}{2}$)일 때, 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 확률 p_k 를 구하면 식 (10)과 같다.

$$p_k = \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2}-k}{m-2k} 2^{(m-2k)}}{\binom{n}{m}}, 0 \leq k \leq \frac{m}{2} \quad (10)$$

ID 중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 때 전송 회수는 $(m-k)$ 번이다. 따라서 사용된 태그의 개수가 홀수이고 총 태그개수의 50%미만 일 때 m 개의 태그를 인식하기 위한 반복회수(I_{IBBT})는 식 (11)과 같다.

$$\begin{aligned} I_{IBBT} &= \Phi \times \frac{1}{m} \sum_{k=0}^{\frac{m-1}{2}} p_k (m-k) \\ &= \Phi \times \frac{1}{m} \sum_{k=0}^{\frac{m-1}{2}} \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2}-k}{m-2k} 2^{(m-2k)}}{\binom{n}{m}} \cdot (m-k) \end{aligned} \quad (11)$$

ID가 순차적인 n 개의 태그 중 사용된 태그의 개수 (m)가 총 태그 개수의 50%를 초과 ($\frac{n}{2} < m < n$)할 때, 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 확률 p_k 를 계산하면 식 (12)와 같다.

$$\begin{aligned} p_k &= \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2}-k}{k-m+\frac{n}{2}} 2^{(m-2k)}}{\binom{n}{m}} \\ &, m - \frac{n}{2} \leq k \leq \frac{m-1}{2} \end{aligned} \quad (12)$$

ID 중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 때 전송회수는 $(m-k)$ 번이다. 따라서 사용된 태그의 개수가 홀수이고 총 태그개수의 50%를 초과 할 때 m 개의 태그를 인식하기 위한 반복회수(I_{IBBT})는 식 (12)와 같다.

$$\begin{aligned} I_{MBBT} &= \Phi \times \frac{1}{m} \sum_{k=m-\frac{n}{2}}^{\frac{m-1}{2}} p_k (m-k) \\ &= \Phi \times \frac{1}{m} \sum_{k=m-\frac{n}{2}}^{\frac{m-1}{2}} \frac{\binom{\frac{n}{2}}{k} \binom{\frac{n}{2}-k}{k-m+\frac{n}{2}} 2^{(m-2k)}}{\binom{n}{m}} \cdot (m-k) \end{aligned} \quad (13)$$

최종적으로, 사용된 태그를 모두 인식하는데 필요한 반복회수는 식 (7), (9), (11) 및 (13)을 사용하여 구할 수 있다.

2. 태그가 보낸 총 bit 수

RFID 시스템 성능의 두 가지 중요한 척도는 태그를 인식하기 위해 필요한 시간과 태그에 의해 소모된 에너지이다. 리더에 의해 전송요구가 있을 때 태그가 보낸 데이터가 적을수록 태그인식시간이 단축되고 태그에 의해 소모되는 에너지도 적게 된다. 따라서 본 논문에서는 리더의 요구에 태그가 응답한 회수의 총 합인 반복회수와 태그가 보낸 총 bit수를 구한다. Improved bit-by-bit 이진 트리 알고리즘을 제외한 각 알고리즘의 반복회수를 I 라 하고 매 반복마다 태그가 보낸 bit수를 B_I 라 하면 태그가 보낸 총 bit 수 B_{total} 은 식 (14)와 같다.

$$B_{total} = I \cdot B_I \quad (14)$$

Improved bit-by-bit 이진 트리 알고리즘의 경우 리더가 최초 태그에게 모든 ID bit 전송을 요구할 때 태그는 리더에게 36 bit을 전송을 하게 되므로, 태그가 보낸 총 bit 수는 반복회수에 35를 더하면 된다.

V. 분석 결과 및 고찰

그림 5는 각 알고리즘에서 태그의 개수에 따른 태그가 전송한 총 bit수를 나타낸 것이며 이것은 수학적 분석에 의한 결과이다. 제안한 알고리즘은 리더로부터 전송요구가 있을 때 마다 태그 ID 중 1 bit 만을 전송한다. 그러나 이진 탐색 알고리즘과 Basic slotted 이진 트리 알고리즘은 리더로부터의 전송요구가 있을 때마다 태그 ID의 모든 bit 또는 2 bit 이상을 전송한다.(단, 동적 이진 탐색 알고리즘은 리더로부터의 전송요구가 있을 때마다 태그가 전송하는 bit의 수가 점점 줄어든다.)

그림 5에서 100개의 태그를 인식하기 위해 이진 탐색 알고리즘은 22491 bit를 전송하고 modified slotted 이진 트리 알고리즘은 9482 bit를 전송하며, Auto-ID 센터에서 제안한 bit-by-bit 이진 트리 알고리즘은 3600 bit를 전송한다. 그러나 본 논문에서 제안한 Improved bit-by-bit 이진 트리 알고리즘은 100개의 태그를 인식하기 위해 578 bit만을 전송하므로 기존의 Anti-collision 알고리즘과 비교할 때 많은 성능향상이 있음을 알 수 있다. 또한 제안한 Improved bit-by-bit 이진 트리 알고리즘은 태그의 개수가 200개인 경우 기존의 알고리즘 중 성능이 가장 뛰어난 bit-by-bit 이진 트리 알고리즘과 비교할 때 839%의 성능향상이 있음을 알 수 있었다.

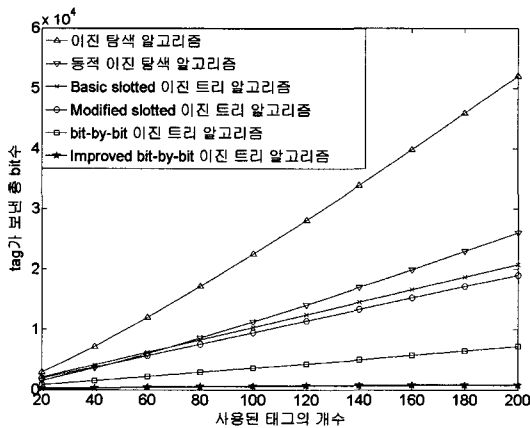


그림 5. 태그개수에 대한 태그가 보낸 총 bit 수
Fig. 5. The number of total transferred bits from the tag versus the number of used tags.

그림 6은 bit-by-bit 이진 트리 알고리즘(BBT)과 제안한 Improved bit-by-bit 이진 트리 알고리즘(IBBT)에서 태그의 ID bit수^[9]에 따른 태그의 개수에 대한 반복회수를 나타낸다.

그림 6에서 선으로 표시된 부분은 수학적 분석의 결과를 나타내고 기호로 표시된 부분은 OPNET을 이용한 모의실험 결과를 나타낸다. 수학적 분석에 의해 얻은 값과 모의실험의 결과 값이 매우 유사함을 알 수 있었다. 그림에서 알 수 있듯이 bit-by-bit 이진 트리 알고리즘은 태그의 ID bit수가 증가할수록 태그를 인식하기 위한 반복회수가 증가하는 것을 알 수 있다. 그러나 제안한 Improved bit-by-bit 이진 트리 알고리즘은 태그의 ID bit수가 증가하여도 반복회수에는 변함이 없는 것을 알 수 있다. 태그의 ID bit수가 많을수록 제안한 알고리즘이 더욱 우수한 성능을 보이는 것을 알 수 있다. 제안한 Improved bit-by-bit 이진 트리 알고리즘은 bit-by-bit 이진 트리 알고리즘과 비교할 때 태그의 ID가 36bit인 경우 순차적인 200개의 태그 중 사용된 태그가 임의의 20개일 경우에는 약 304%정도의 성능향상이 있고 사용된 태그가 200개일 경우에는 839%의 성능향상이 있음을 알 수 있었다. 따라서 태그의 개수가 증가할수록 제안한 Improved bit-by-bit 이진 트리 알고리즘이 더욱더 좋은 성능을 보임을 알 수 있었다. 따라서 분석결과에 의하면 제안한 알고리즘은 n 개의 태그를 인식하기 위해 태그가 보낸 bit수가 기존의 알고리즘에 비해 적기 때문에 태그를 인식하기 위한 시간이 단축되고, 태그에 의해 소모되는 에너지도 적다.

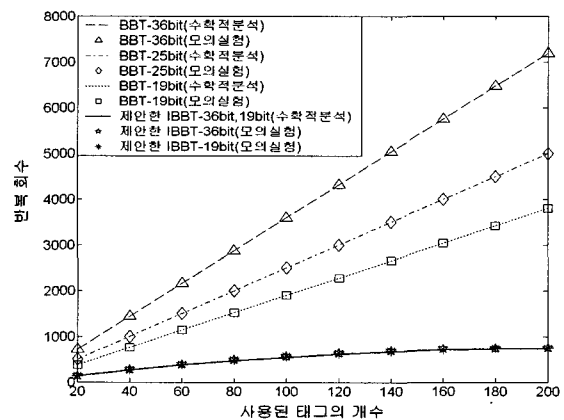


그림 6. bit-by-bit 방식과 제안한 알고리즘에서 태그의 개수에 대한 반복회수
Fig. 6. The number of iterations versus the number of used tags in bit-by-bit binary tree algorithm and the proposed algorithm.

VI. 결 론

본 논문에서는 RFID 시스템에서의 Anti-collision 알고리즘으로 Improved bit-by-bit 이진 트리 알고리즘을 제안하였고, 기존의 Anti-collision 알고리즘과 성능을 수학적으로 비교 및 분석하였으며 모의실험을 통하여 분석 결과를 검증하였다. 제안한 알고리즘은 사용된 태그의 수가 많아질수록 기존의 bit-by-bit 이진 트리 알고리즘보다 더 좋은 성능을 보이고 태그로부터 전송된 bit수가 다른 방식에 비하여 가장 적다. 분석 결과에 의하면 제안한 Improved bit-by-bit 이진 트리 알고리즘의 경우 bit-by-bit 이진 트리 알고리즘과 비교할 때 사용된 태그의 개수가 총 태그 개수의 100%일 경우에는 839%의 성능향상이 있음을 확인했다. 따라서 제안한 Anti-collision 알고리즘을 RFID 시스템에 적용한다면 적은양의 에너지로 같은 시간 내에 훨씬 더 많은 태그를 인식할 수 있으므로 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, Zürich, 2002.

[2] S. A. Weis, S. E. Sarma, R. L. Rivest and D. W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," First International Conference on Security in Pervasive Computing., March, 2003.

[3] 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.

[4] K. Finkenzeller, *RFID Handbook; Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition*, John Wiley & Sons Ltd, pp.195-219, 2003.

[5] J. L. Massey, "Collision resolution algorithms and random-access communications," Univ. California, Los Angeles, Tech. Rep. UCLAENG-8016, Apr., 1980.

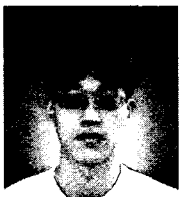
[6] Auto-ID Center, *Draft Protocol Specification for a Class 0 Radio Frequency Identification tag*, 2003.

[7] M. Jacomet, A. Ehram, and U. Gehrig, "Contactless Identification Device With Anticollision Algorithm," IEEE Computer Society CSCC'99, pp.269-273, Jul. 4-8, Athens 1999.

[8] 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.

[9] EPC Global, *EPC™ Tag Data Standards Version 1.1 Rev.1.24*, Apr. 2004.

저 자 소 개



최 호 승(준회원)
 2004년 2월 아주대학교 정보통신
 정보통신대학 전자공학부
 학사 졸업.
 2004년 3월~현재 아주대학교
 전자공학부 석사 재학 중.

<주관심분야 : 통신, RFID, MAC protocol>



김 재 현(정회원)
 1991년 2월 한양대학교
 전자계산학과 졸업
 1993년 2월 한양대학교
 전자계산학과 공학석사
 1996년 8월 한양대학교
 전자계산학과 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>