

Ubiquitous ID 시스템에서 고속 충돌 방지 알고리즘

준회원 차재룡*, 정회원 김재현**

Fast Anti-Collision Algorithm in Ubiquitous ID System

Jae-Ryong Cha* *Associative Member*, Jae-Hyun Kim** *Regular Member*

요 약

본 논문은 Ubiquitous ID 시스템의 고속 충돌 방지 알고리즘을 제안하고 분석한다. 제안한 Ubiquitous ID 시스템에서의 고속 충돌 방지 알고리즘과 기존의 이진 탐색 알고리즘, time slot을 이용한 slotted 이진 트리 알고리즘, 그리고 Auto-ID 센터에서 제안한 bit-by-bit 이진 트리 알고리즘을 수학적 비교 및 분석하였다. 수학적 분석 결과는 OPNET 모의실험을 통하여 그 결과를 검증하였다. 분석결과에 의하면 제안한 알고리즘의 성능이 기존의 충돌 방지 알고리즘 중 가장 좋은 성능을 보이는 bit-by-bit 이진 트리 알고리즘과 비교할 때 리더의 전송요구에 응답한 순차적인 태그의 개수가 20개일 경우에는 약 5%정도의 성능이 향상되었으며 리더의 전송요구에 응답한 태그의 개수가 200개일 경우에는 100%의 성능이 향상되었다.

Key Words Ubiquitous ID system, Anti-collision algorithm, Binary Tree algorithm

ABSTRACT

This paper proposes and analyzes the anti-collision algorithm in Ubiquitous ID system. We mathematically compare the performance of the proposed algorithm with that of 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 the analytic results, comparing the proposed algorithm with bit-by-bit algorithm which is the best of existing algorithms, the performance of proposed algorithm is about 5% higher when the number of tags is 20, and 100% higher when the number of tags is 200.

I 서론

앞으로 다가올 Ubiquitous 환경에서 사람들은 인체, 이디에서나 끊임없이 필요한 정보와 서비스를 제공받을 것이다. 이러한 Ubiquitous 환경이 실현되기 위해서는 다양한 형태와 기능을 가진 개체들이 필요하다. 이러한 다양한 개체들로부터 각각 필요한 정보를 얻고, 이를 바탕으로 상황에 가장 적절한 결과를 도출하게 된다. RFID 시스템은 Ubiquitous 환

경에서 여러 개의 물리적 개체를 인식하기 위해 사용되는 센서 네트워크의 한 형태이다. 가까운 미래에 RFID 시스템은 모바일 환경 하에서 많은 정보를 필요로 하는 물류관리, 교통, 생산재고 관리 및 전자회피 보안 등의 용도로 사용될 것이다. 따라서 본 논문에서는 RFID 시스템을 Ubiquitous ID 시스템으로 명명한다. Ubiquitous ID 시스템에서 사용되는 Passive RF 태그는 그 능력이 매우 제한적이어서 다른 태그들과 통신을 할 수 없고 단지 리더의 통신할 수 있다. 리더는 무선채널을 통하여 각각의

* 아주대학교 일반대학원 전자공학과 (buidet@ajou.ac.kr), ** 아주대학교 정보통신대학 전자공학부 (jkm@ajou.ac.kr)

논문번호 040116-0311, 접수일자 2004년 3월 12일

※본 연구는 과학기술부 21세기프론티어연구개발사업의 일환으로 추진되고 있는 유비쿼터스컴퓨팅및네트워크연계기술개발사업의 지원에 의한 것임

태그들과 통신을 하는데, 모든 태그들이 리더가 보낸 신호를 동시에 듣게 되고 리더의 전송요구에 응답을 한다 이때 하나의 리더가 동시에 응답한 여러 개의 태그를 인식해야하는 문제가 발생 하는데 이를 'Anti-collision problem' 이라고 한다[1-3] Ubiquitous ID 시스템 성능의 두 가지 중요한 척도는 태그를 인식하기 위해 필요한 시간과 태그에 의해 소모되는 에너지이다 리더에 의해 전송요구가 있을 때 태그가 보낸 데이터가 적을수록 태그인식 시간이 단축되고 태그에 의해 소모된 에너지도 적다 본 논문에서는 제안한 Ubiquitous ID 시스템에서의 고속 충돌 방지 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 분석하였으며 모의실험을 통해 분석된 결과를 검증하였다

II장에서는 기존의 충돌 방지 알고리즘을 설명하고, III장에서는 제안한 고속 충돌 방지 알고리즘을 설명한다 IV장에서는 제안한 알고리즘의 성능분석으로써 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 충돌 방지 알고리즘

1 이진 탐색 알고리즘

1) Basic 이진 탐색 알고리즘 (4)

Basic 이진 탐색 알고리즘은 충돌이 발생한 부분을 점차 줄임으로써 전송 가능한 태그의 수를 줄여 충돌을 해결하는 방식이다 리더는 인식 가능한 영역에 있는 모든 태그의 ID를 수신하여 충돌이 일어나는 bit의 위치를 파악한다. 그 중 충돌이 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 bits인 4개의 태그를 인식하기 위해 REQUEST(≤ 1111) 명령을 전송하면 ID가 '1111'보다 작거나 길은 값을 가지고 있는 태그들은 모두 자신의 ID를 전송한다 리더는 태그로부터 수신된 ID의 시퀀스를 메모리에 저장한다 받은 시퀀스 중 첫 번째 bit가 충돌이 발생하였으므로 임

표 1 사용된 태그의 ID

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

의로 검색범위를 ' ≤ 0111 '로 해서 REQUEST 명령을 전송하게 되고 하나의 태그가 남을 때까지 이런 과정을 반복함으로써 n 개의 태그 중 하나를 인식한다 n 개의 태그가 존재한다고 할 때 하나의 태그를 인식하기 위한 Basic 이진 탐색 알고리즘의 반복회수(I_{BBS})는 (1)과 같다

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

이 때, 반복회수는 리더의 요구에 태그가 응답한 회수의 총합을 나타낸다

2) Dynamic 이진 탐색 알고리즘 (4)

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

2 Slotted 이진 트리 알고리즘

1) Basic slotted 이진 트리 알고리즘 (5)

리더가 태그에게 전송요구를 하면 태그들은 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 재 전송을 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 충돌이 발생하게 되면 다시 랜덤하게 0 또는 1을 선택해서 또 다른 두개의 하위 그룹으로 나누어진다. 이런 과정을 반복적으로 수행함으로써 모든 충돌을 해결할 수 있게 된다. 그림 1은 표 1의 4개의 태그를 인식하기 위한 Basic slotted 이진 트리 알고리즘의 진행순서를 나타낸다. 그림 1에서, 'X'는 리더의 요구에 응답한 태그의 개수가 두 개 이상이어서 충돌이 발생했음을 의미하고, 'I'는 idle 슬롯을 의미하며, 'S'는 리더의 요구에 응답한 태그의 개수가 한 개이어서 성공적인 전송이 이루어졌음을 의미한다. 자세한 내용은 [5]를 참고하기 바란다. Slotted 이진 트리 알고리즘에서 n 개의 태그를 인식하기 위한 반복회수(I_{BSBT})는 (2)와 같다.

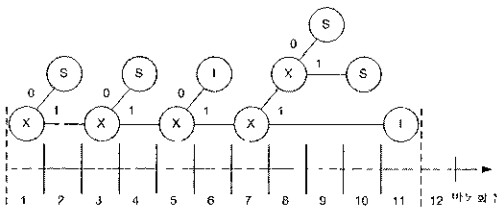


그림 1 Slotted 이진 트리 알고리즘의 예

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

이 때, q 는 태그가 0 또는 1을 선택할 확률로 0.5이다.

2) Modified slotted 이진 트리 알고리즘 (5)

Basic slotted 이진 트리 알고리즘에서 각 태그들이 0 또는 1을 선택할 때 모두 0을 선택하거나 1을 선택하게 되면 idle 상태가 존재하게 되는데, Modified slotted 이진 트리 알고리즘은 ternary feedback을 이용하여 idle 슬롯을 제거함으로써 충돌 해결 과정을 좀 더 빠르게 할 수 있다. 자세한 내용은 [5]를 참고하기 바란다. Modified slotted 이진 트리 알고리즘에서 n 개의 태그를 인식하기 위한 반복회수(I_{MSBT})는 (3)과 같다.

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

3 bit-by-bit 이진 트리 알고리즘 (6-8)

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. 충돌이 발생하지 않으면 리더는 태그로부터 받은 bit를 메모리에 저장한 후 다음 bit를 요청하게 된다. 그러나 충돌이 발생하면 리더는 0 또는 1을 가진 그룹 중에서 알고리즘에 의해 하나의 그룹을 선택하고 다음 bit를 요청하게 된다. 이런 과정을 태그의 ID길이만큼 반복함으로써 하나의 태그를 인식하게 된다. 그림 2는 bit-by-bit 이진 트리 알고리즘을 사용하여 표 1의 4개의 태그를 인식하기 위한 과정을 나타낸다. 이 경우 반복회수는 16(4 × 4 bits)이다. 만일 n 개의 태그가 존재하고 각 태그의 ID가

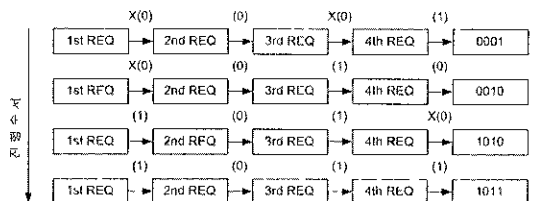


그림 2 Bit-by-bit 이진 트리 알고리즘의 예

j bit라면 모든 태그를 인식하기 위해 필요한 반복 회수(I_{BBT})는 (4)와 같다

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

III. 제안한 고속 충돌 방지 알고리즘

Bit-by-bit 이진 트리 알고리즘[6-8]의 단점은 리더가 항상 태그 ID의 모든 bit를 요구한다는 것이다. 본 논문에서 제안한 Ubiquitous ID 시스템에서 고속 충돌 방지 알고리즘은 이러한 문제점을 해결하고 개선함으로써 태그인식과정을 단축시켰다. 리더가 인식할 수 있는 범위 내의 모든 태그에게 ID의 l 번째 bit 전송요구(l 의 초기값은 1)를 하게 되면 모든 태그들은 리더의 요구에 대한 응답으로 l 번째 bit를 전송하게 된다. 태그로부터 수신된 l 번째 bit가 충돌이 아니면 l 번째 bit를 메모리에 저장한 후 다음 bit의 전송을 요구한다. 그러나 충돌이 발생하면 메모리에 임의로 l 번째 bit를 0으로 저장한 후 다음 bit의 전송요구와 함께 l 번째 bit가 1인 모든 태그를 inactive 상태로 만든다 (inactive 상태 알고리즘 수행 중 충돌이 발생했을 때 충돌이 발생한 bit가 1인 태그들이 일시적으로 리더의 다음 bit 전송요구에 응답하지 않는 대기상태이며, 하나의 태그가 인식되면 리더는 inactive 상태의 태그들을 리더의 전송요구에 응답할 수 있는 active 상태로 만든다) 만일 리더가 태그에게 마지막 bit 전송요구를 했을 때 충돌이 발생하면 ID중 마지막 bit만 다른 두 개의 태그가 존재한다는 것을 알 수 있으므로 알고리즘에 의해 동시에 두 개의 태그를 인식할 수 있다. 그림 3에서 ID가 '1010'인 태그와 '1011'인 태그는 ID중 마지막 bit만 다르기 때문에 리더가 마지막 bit 전송요구를 했을 때 태그로부터 수신된 데이터가 충돌이 발생하면 리더는 재전송 요구 없이 두 개의 태그를 동시에 인식한다. 따라서 기존의

bit-by-bit 이진 트리 알고리즘에 비하여 모든 태그를 인식하기 위한 시간을 줄일 수 있고 태그가 전송한 bit의 수도 적기 때문에 태그에 의해 소모된 에너지도 적다. 하나(또는 두 개)의 태그 인식 과정이 끝나면 알고리즘 진행 동안 inactive 상태로 되어있는 태그를 active 상태로 만들고 다시 위의 과정을 반복함으로써 모든 태그를 인식할 수 있게 된다. 제안한 고속 충돌 방지 알고리즘을 사용하여 표 1의 4개의 태그를 인식하기 위한 반복회수는 $12(3 \times 4 \text{ bits})$ 이다.

IV 성능 분석

본 장에서는 제안한 고속 충돌 방지 알고리즘의 성능을 분석하기 위하여 반복회수와 태그가 전송한 총 bit수를 계산한다. 그리고 총 태그의 개수는 $2n$ 개이고 $2n$ 개의 태그는 모두 순차적인 ID를 갖고며 태그 ID의 길이는 36 bits[6] 라고 가정한다.

1 제안한 알고리즘의 반복회수 분석

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

1) 태그의 개수가 짝수($2m$)일 경우

ID가 순차적인 $2n$ 개의 태그 중 사용된 태그의 개수($2m$)가 총 태그 개수의 50%이하일 때 ($0 < 2m \leq n$), 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 확률 p_k 를 구하면 (5)와 같다.

$$p_k = \frac{\binom{n}{k} \binom{n-k}{2(m-k)} 2^{2(m-k)}}{\binom{2n}{2m}}, \quad 0 \leq k \leq m \quad (5)$$

그리고 ID중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k 쌍일 때 전송회수는 $(2m - k)$ 번이다. 따라서 사용된 태그의 개수가 짝수이고 총 태그개수의 50%이하 일 때, $2m$ 개의 태그를 인식하기 위한 반복회수(I_{MBBT})는 (6)과 같다.

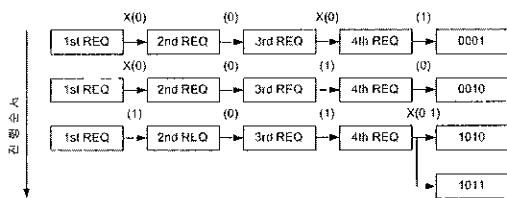


그림 3 제안한 고속 충돌 방지 알고리즘의 예

$$\begin{aligned}
 I_{MBBT} &= \sum_{k=0}^m p_k (2m - k) \\
 &= \sum_{k=0}^m \frac{\binom{n}{k} \binom{n-k}{2(m-k)} 2^{2(m-k)}}{\binom{2n}{2m}} \cdot (2m - k)
 \end{aligned} \tag{6}$$

ID가 순차적인 2n개의 태그 중 사용된 태그의 개수(2m)가 총 태그 개수의 50%를 초과할 때 (n < 2m ≤ 2n), 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k쌍일 확률 p_k를 계산하면 (7)과 같다

$$p_k = \frac{\binom{n}{k} \binom{n-k}{k-2m+n} 2^{2(m-k)}}{\binom{2n}{2m}} \quad 2m - n \leq k \leq m \tag{7}$$

ID중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k쌍일 때 전송회수는 (2m - k)번이다 따라서 사용된 태그의 개수가 짝수이고 총 태그 개수의 50%를 초과할 때 2m개의 태그를 인식하기 위한 반복회수(I_{MBBT})는 (8)과 같다

$$\begin{aligned}
 I_{MBBT} &= \sum_{k=2m-n}^m p_k (2m - k) \\
 &= \sum_{k=2m-n}^m \frac{\binom{n}{k} \binom{n-k}{k-2m+n} 2^{2(m-k)}}{\binom{2n}{2m}} \cdot (2m - k)
 \end{aligned} \tag{8}$$

2) 태그의 개수가 홀수(2m-1)일 경우

ID가 순차적인 2n개의 태그 중 사용된 태그의 개수(2m-1)가 총 태그 개수의 50%미만일 때 (0 < 2m - 1 < n), 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k쌍일 확률 p_k를 구하면 (9)와 같다

$$p_k = \frac{\binom{n}{k} \binom{n-k}{2m-2k-1} 2^{2(m-2k-1)}}{\binom{2n}{2m-1}} \quad , 0 \leq k \leq m-1 \tag{9}$$

ID중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k쌍일 때 전송회수는 (2m-1-k)번이다 따라서 사용된 태그의 개수가 홀수이고 총 태그 개수의 50%미만일 때 (2m-1)개의 태그를 인식하기 위한 반복회수(I_{MBBT})는 (10)과 같다

$$\begin{aligned}
 I_{MBBT} &= \sum_{k=0}^{m-1} p_k (2m - 1 - k) \\
 &= \sum_{k=0}^{m-1} \frac{\binom{n}{k} \binom{n-k}{2m-2k-1} 2^{2(m-2k-1)}}{\binom{2n}{2m-1}} \cdot (2m - 1 - k)
 \end{aligned} \tag{10}$$

ID가 순차적인 2n개의 태그 중 사용된 태그의 개수(2m-1)가 총 태그 개수의 50%를 초과 할 때 (n < 2m - 1 < 2n), 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k쌍일 확률 p_k를 계산하면 (11)과 같다

$$p_k = \frac{\binom{n}{k} \binom{n-k}{k-2m+n+1} 2^{2(m-2k-1)}}{\binom{2n}{2m-1}} \quad , 2m - n - 1 \leq k \leq m - 1 \tag{11}$$

ID중 마지막 bit를 제외한 모든 bit가 동일한 두개의 태그가 k쌍일 때 전송회수는 (2m-1-k)번이다 따라서 사용된 태그의 개수가 홀수이고 총 태그 개수의 50%를 초과할 때 (2m-1)개의 태그를 인식하기 위한 반복회수(I_{MBBT})는 (12)와 같다

$$\begin{aligned}
 I_{MBBT} &= \sum_{k=2m-n-1}^{m-1} p_k (2m - 1 - k) \\
 &= \sum_{k=2m-n-1}^{m-1} \frac{\binom{n}{k} \binom{n-k}{k-2m+n+1} 2^{2(m-2k-1)}}{\binom{2n}{2m-1}} \cdot (2m - 1 - k)
 \end{aligned} \tag{12}$$

최종적으로, 사용된 태그를 모두 인식하는데 필요한 반복회수는 식 (6), (8), (10) 및 (12)를 사용하여 구할 수 있다 따라서 제안한 고속 충돌 방지 알고리즘의 반복회수(I_{MBBT})를 정리하면 (13), (14)

와 같다

- 태그의 개수가 짝수일 때

$$I_{MBBT} = \begin{cases} \sum_{k=0}^m \frac{\binom{n}{k} \binom{n-k}{2(m-k)}}{\binom{2n}{2m}} 2^{2(m-k)} \\ \cdot (2m-k), 0 \leq k \leq m \end{cases} \quad (13)$$

$$I_{MBBT} = \begin{cases} \sum_{k=2m-n}^m \frac{\binom{n}{k} \binom{n-k}{k-2m+n}}{\binom{2n}{2m}} 2^{2(m-k)} \\ \cdot (2m-k), 2m-n \leq k \leq m \end{cases}$$

- 태그의 개수가 홀수일 때

$$I_{MBBT} = \begin{cases} \sum_{k=0}^{m-1} \frac{\binom{n}{k} \binom{n-k}{2m-2k-1}}{\binom{2n}{2m-1}} 2^{2(m-2k-1)} \\ (2m-1-k), 0 \leq k \leq m-1 \end{cases} \quad (14)$$

$$I_{MBBT} = \begin{cases} \sum_{k=2m-n-1}^{m-1} \frac{\binom{n}{k} \binom{n-k}{k-2m+n+1}}{\binom{2n}{2m-1}} 2^{2(m-2k-1)} \\ (2m-1-k), 2m-n-1 \leq k \leq m-1 \end{cases}$$

2 태그가 보낸 총 bit 수

Ubiquitous ID 시스템 성능의 두 가지 중요한 척도는 태그를 인식하기 위해 필요한 시간과 태그에 의해 소모된 에너지이다 리더의 전송요구가 있을 때 태그가 보낸 데이터가 적을수록 태그인식시간이 단축되고 태그에 의해 소모된 에너지도 적게 된다 따라서 본 논문에서는 반복회수와 태그가 보낸 총 bit수를 구한다 각 알고리즘의 반복회수를 I 라 하고 매 반복마다 태그가 보낸 bit 수를 B_T 라 하면 태그가 보낸 총 bit 수 B_T 은 (15)와 같다

$$B_T = I \cdot B_T \quad (15)$$

V. 분석 결과 및 고찰

그림 4는 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 1bit만을 전송한다 그러나 이진 탐색 알고리즘과 Basic

slotted 이진 트리 알고리즘은 리더로부터의 전송요구가 있을 때마다 태그 ID의 모든 bit 또는 2 bit 이상을 전송한다(단, Modified 이진 트리 알고리즘은 리더로부터의 전송요구가 있을 때마다 태그가 전송하는 bit의 수가 점점 줄어든다) 그림 4에서 100개의 태그를 인식하기 위해 이진 탐색 알고리즘은 22,491bits를 전송하고 modified slotted 이진 트리 알고리즘은 9,482bits를 전송하며 bit-by-bit 이진 트리 알고리즘은 3,600bits를 전송한다. 그러나 본 논문에서 제안한 충돌 방지 알고리즘은 100개의 태그를 인식하기 위해 2,704bits만을 전송하므로 기존의 충돌 방지 알고리즘과 비교할 때 많은 성능향상이 있음을 알 수 있다 또한 그 중 성능이 가장 우수한 bit-by-bit 이진 트리 알고리즘과 비교할 때 태그의 개수가 200개인 경우에는 100%의 성능향상이 있음을 알 수 있었다. 따라서 제안한 알고리즘은 n 개의 태그를 인식하기 위해 태그가 전송한 bit 수가 기존의 알고리즘에 비해 적기 때문에 태그를 인식하기 위한 시간이 단축되고, 태그에 의해 소모된 에너지도 적다

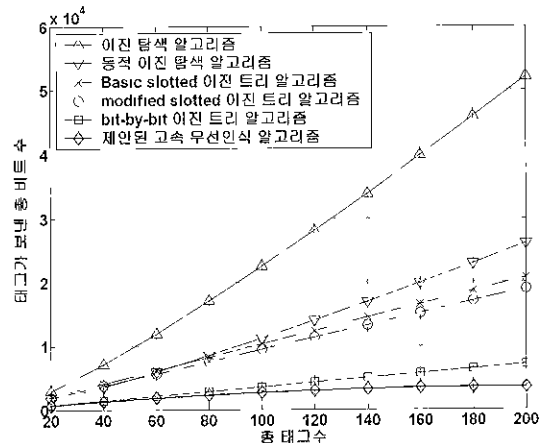


그림 4 태그 개수에 대한 태그가 보낸 총 bit 수

그림 5는 bit-by-bit 이진 트리 알고리즘과 제안한 충돌 방지 알고리즘에서 태그의 개수에 대한 반복회수를 나타낸다 그림 5에서 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 제안한 알고리즘은 bit-by-bit 이진 트리 알고리즘과 비교할 때 순차적

인 200개의 태그 중 사용된 태그가 임의의 20개일

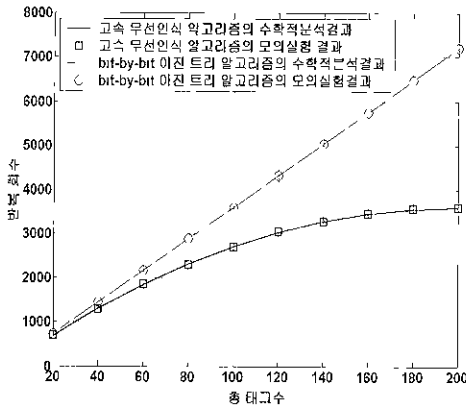


그림 5 Bit-by-bit 방식과 제안한 알고리즘에서 태그의 개수에 대한 반복회수

경우에는 약 5% 정도의 성능향상이 있고, 사용된 태그가 200개일 경우에는 100%의 성능향상이 있음을 알 수 있었다 따라서 태그의 개수가 증가 할 수록 제안한 알고리즘이 더욱 더 좋은 성능을 보임을 알 수 있었다

VI. 결론

본 논문에서는 Ubiquitous ID 시스템에서의 고속 충돌 방지 알고리즘을 제안하였고, 기존의 충돌 방지 알고리즘과 성능을 수학적으로 비교 분석하였으며, 시뮬레이션을 통하여 분석 결과를 검증하였다 제안한 알고리즘은 리더의 요구가 있을 때마다 1bit 만을 전송하는 bit-by-bit 방식으로서 태그의 수가 많을수록 bit-by-bit 이진 트리 알고리즘보다 더 좋은 성능을 보이고, 대그로부터 전송된 bit수가 다른 방식에 비하여 가장 적기 때문에 태그에 의해 소모된 에너지도 가장 적다 분석 결과에 의하면 사용된 태그의 개수가 총 태그 개수의 10%일 경우에는 5%의 성능향상이 있었고 태그의 개수가 총 태그 개수의 100%일 경우에는 100%의 성능향상이 있음을 확인했다 또한 제안한 알고리즘은 태그의 개수가 많을수록 성능향상이 증가함을 알 수 있었다 따라서 제안한 고속 충돌 방지 알고리즘을 Ubiquitous ID 시스템에 적용한다면 적은양의 에너지로 같은 시간 내에 훨씬 더 많은 태그를 인식할 수 있으므로 Ubiquitous ID 시스템의 성능 향상에 크게 기여할 수 있을 것이다

참고문헌

- [1] H Vogt, "Efficient Object Identification with Passive RFID tags," International Conference on Pervasive Computing, Zurich, 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, Mar, 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, Jul 4-8, Athens, 1999
- [8] 차재룡, 최호성, 김재현, "Ubiquitous ID 시스템에서 고속 무선인식 알고리즘," JCCI2004, pp 317, Apr 28-30, 2004.

