

ALOHA 방식 RFID 시스템에서의 태그 개수 추정 방법

준회원 이 지 봉*, 김 완 진*, 종신회원 김 형 남*

Estimation of Number of Tags in ALOHA-based RFID Systems

Ji-Bong Lee*, Wan-Jin Kim* *Associate Members*, Hyoung-Nam Kim* *Lifelong Member*

요약

본 논문에서는 ALOHA 기반의 충돌 방지 알고리즘을 사용하는 RFID 시스템에서 다중 태그 인식 성능 향상을 위한 태그 개수 추정 방법을 제안한다. ALOHA 기반의 충돌 방지 알고리즘에서는 슬롯의 크기를 태그 개수에 따라 적절하게 설정하는 것이 다중 태그 인식 성능을 좌우하기 때문에, 태그 개수에 대한 정확한 추정이 필요하다. 제안된 방법은 태그 개수와 태그 인식 과정에서 발생하는 빈 슬롯 개수의 기댓값과의 관계를 이용한다. 한 ROUND를 통해 측정되는 빈 슬롯 개수를 가지고 가장 가까운 기댓값을 찾은 후, 그 기댓값에 해당하는 태그 개수를 구함으로써 태그의 개수를 추정한다. 제안된 방법은 기존 방법과 비교하여 계산양이 적으면서도 태그 개수 추정의 정확도는 높다는 장점이 있으며, 시뮬레이션 결과를 통해서 이를 보인다.

Key Words :RFID, Anti-Collision, Tag, Estimation, ALOHA

ABSTRACT

This paper proposes an estimation method of number of tags which may be necessary in improving anti-collision performance for ALOHA-based RFID systems. In ALOHA-based anti-collision algorithms, since the performance of the multi tag identification can be improved by allocating the optimal slot size corresponding to the number of tags, it is needed to exactly estimate the number of tags. The proposed method uses the a priori knowledge of the relation between the expectation of the number of empty slots and the number of tags. After measuring the number of empty slots in one ROUND, we estimate the number of tags which corresponds to the expectation of the number of empty slots nearest to the measured one. Simulation results show that the proposed method is superior to conventional methods in terms of the estimation accuracy and the computational complexity.

I. 서 론

RFID(Radio Frequency Identification)란 사물에 부착된 전자태그로부터 무선 주파수를 이용하여 정보를 송·수신하고 이와 관련된 서비스를 제공하는 기술을 일컫는다^[1]. RFID 시스템에서는 태그에 극 소형 칩을 부착해 많은 정보를 저장할 수 있으며, 상품에 태그의 부착이 용이하고 바코드에 비해 장

거리 정보 송·수신이 가능하다. 특히 유통 및 물류 분야에 사용될 900 MHz RFID 시스템은 각종 물품에 전자태그를 부착해 자동으로 물품 명세와 가격, 유통경로 및 기한 등을 파악할 수 있어 유통 및 물류에 혁신을 가져올 기술로 각광받고 있다. 또한 앞으로 센서 기술과 칩 소형화 기술 등이 더욱 발전하면 정보를 능동적으로 획득하고 처리하는 능력 까지 갖추게 되어 바코드가 하던 역할과는 비교할

* 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.

* 부산대학교 전자전기공학과 통신 및 신호처리 연구실(컴퓨터 및 정보통신 연구소) (hnkim@pusan.ac.kr)

논문번호 : KICS2006-11-513, 접수일자 : 2007년 11월 27일, 최종논문접수일자 : 2007년 6월 27일

수 없을 정도로 많은 일들을 해낼 수 있을 것으로 기대되고 있다^[2].

일반적으로 RFID 시스템에서 사용되는 태그는 수동형과 능동형이 있다. 900 MHz RFID 시스템에서는 수동형 태그가 사용되는데, 그 능력이 매우 제한적이어서 다른 태그들과 통신을 할 수 없고 단지 리더와만 통신이 가능하다. 리더는 무선 채널을 통하여 각각의 태그들과 통신을 하는데, 모든 태그들이 리더가 보낸 신호를 동시에 받게 되고 리더의 전송요구에 응답을 한다. 이 때 하나의 리더가 동시에 응답한 여러 개의 태그를 인식해야 하는 문제가 발생하는데 이를 태그 충돌이라고 한다^[1]. 여러 개의 태그를 충돌없이 인식하기 위해서는 이러한 태그 충돌 문제를 해결하는 충돌방지 알고리즘이 필요하며, 이는 900 MHz RFID 시스템에서 가장 핵심 기술로서 인식되고 있다^[3].

본 논문에서는 충돌방지 알고리즘 중의 하나인 ALOHA 방식에 대해 간략히 언급하고 ALOHA 방식 충돌방지 알고리즘의 성능을 결정하는 중요한 요소 중 하나인 태그 개수 추정 방법에 대한 기존의 연구들을 살펴본다. 이어서 효율적으로 태그 개수를 추정할 수 있는 새로운 방법을 제안하고 시뮬레이션을 통해 기존 방법들과의 비교, 분석 결과를 제시한다.

본 논문의 구성은 다음과 같다. II장에서는 ALOHA 방식 충돌방지 알고리즘에 대해 알아보고, III장에서는 현재 사용되고 있는 태그 개수 추정 방법에 대해 살펴본다. IV장에서는 새로운 태그 개수 추정 방법을 제안하고 V장에서는 제안하는 방법과 기존방법들의 성능 비교를 위해 시뮬레이션 결과를 제시한다. 마지막으로 VI장에서 결론을 맺는다.

II. ALOHA 방식 충돌방지 알고리즘

RFID에서 사용되는 ALOHA 방식 충돌방지 알고리즘은 슬롯 크기의 결정 방법에 따라 크게 Framed Slotted ALOHA(FSA) 방식과 Dynamic Framed Slotted ALOHA(DFSA) 방식으로 분류된다.

2.1 FSA 알고리즘

ALOHA 방식의 충돌방지 알고리즘 중 가장 기본이 되는 FSA 알고리즘은 현재 사용되고 있는 RFID 시스템 중 ISO 18000-6 TYPE A^[4]와 ISO 18000-6 TYPE C^[5]에서 사용되고 있다. FSA 알고리즘에서는 리더가 태그에게 ID 전송을 요구할 때

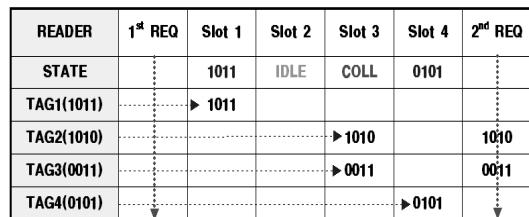


그림 1. FSA의 태그 인식 과정^[3]

태그가 자신의 슬롯 넘버를 선택할 수 있는 정보를 함께 전송한다. 태그는 리더로부터 ID 전송 요구를 수신하면 리더로부터 받은 정보를 이용하여 자신이 응답할 슬롯을 결정한 후 전송을 시도한다. 만일 두 개 이상의 태그가 같은 슬롯에서 전송을 시도하면 충돌이 발생하고 태그들이 전송한 데이터는 손실된다^[1]. 그림 1은 FSA 알고리즘을 이용하여 4개의 태그를 인식하는 과정의 한 예를 나타내고 있다.

리더가 첫 번째 ID 전송요구 (1st REQ)를 하면 태그는 자신의 슬롯 넘버를 선택하여 전송을 시도 한다. 다중 태그를 인식하는 RFID 시스템에서는 일반적으로 리더에서 한 번의 ID 전송요구가 발생하고 전체 슬롯에 대해 태그가 응답하는 과정을 한 ROUND로 정의한다. 그림 1에서 TAG 1은 Slot 1을 선택하였고 TAG 2와 TAG 3은 Slot 3을 선택하였으며 TAG 4는 Slot 4를 선택하였다. Slot 1과 Slot 4에서는 하나의 태그만 전송을 시도하였으므로 리더는 TAG 1과 TAG 4를 성공적으로 인식한다. 반면 Slot 2는 자신의 ID를 전송한 태그가 없기 때문에 빈 슬롯이 되고 Slot 3에서는 TAG 2와 TAG 3이 자신의 ID를 전송하였기 때문에 충돌이 발생한다. 리더는 충돌이 발생한 TAG 2와 TAG 3에 대해 두 번째 ID 전송요구 (2nd REQ)를 발생시키고 첫 번째 ID 전송 요구를 했을 때와 같은 방식으로 태그 인식 과정을 진행한다. 이러한 일련의 과정을 반복하여 리더는 모든 태그를 인식한다^[1].

2.2 DFSA 알고리즘

일반적으로 FSA 알고리즘에서 태그의 개수가 슬롯 크기 (프레임 크기)보다 커지면 충돌 발생이 많이 이어서 태그를 인식하기 위한 시간이 증가한다. 반면 태그의 개수가 슬롯 크기보다 작을 경우에는 빈 슬롯이 많이 생겨 슬롯의 낭비가 발생한다. 따라서 다중 태그 인식 성능을 높이기 위해서는 태그의 개수를 추정하여 그 개수에 따라 슬롯의 크기를 할당하는 것이 필요하다^{[1],[3],[6]}. ALOHA 방식 충돌방지

알고리즘으로 태그 인식 과정을 진행하면서 태그의 개수를 추정하고 이렇게 추정된 태그 개수에 따라 슬롯의 크기를 변화시켜 나가는 알고리즘을 Dynamic Framed Slotted ALOHA(DFSA)라고 한다. 따라서 DFSA 알고리즘에서의 디중 태그 인식 성능은 새로운 ROUND를 시작할 때 최적의 슬롯 크기를 할당할 수 있느냐에 달려있기 때문에 이를 위해서는 태그 개수 추정의 정확성이 매우 중요함을 알 수 있다.

III. 기존 태그 개수 추정 방법

태그의 개수를 추정하는 방법은 크게 두 가지 방법으로 나눌 수 있으며, 기댓값을 이용하는 방법과 측정된 파라미터를 이용하는 방법이다.

3.1 기댓값을 이용한 태그 개수 추정 방법

일정한 슬롯 수에서 태그 개수를 알고 있다고 가정할 때, 한 ROUND 동안 발생하는 빈 슬롯(empty slot) 개수, 식별 슬롯(identified slot) 개수, 충돌이 발생한 슬롯(collision slot) 개수에 대한 기댓값을 구할 수 있다. 한 슬롯에 r 개의 태그가 응답할 확률은 이항분포를 따르기 때문에 다음 식 (1)과 같이 표현된다.

$$B_{n, \frac{1}{N}}(r) = \binom{n}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{n-r} \quad (1)$$

여기서 N 은 슬롯의 개수, n 은 태그의 개수를 나타낸다^[6].

$a_0^{N,n}$ 을 빈 슬롯 개수의 기댓값, $a_1^{N,n}$ 을 식별 슬롯 개수의 기댓값, $a_k^{N,n}$ 을 충돌이 발생한 슬롯 개수의 기댓값이라고 하자. 식 (1)을 이용하여 각각을 구하면 다음과 같다.

$$a_0^{N,n} = N \cdot B_{n, \frac{1}{N}}(0) = N \left(1 - \frac{1}{N}\right)^n \quad (2)$$

$$a_1^{N,n} = N \cdot B_{n, \frac{1}{N}}(1) = n \left(1 - \frac{1}{N}\right)^{n-1} \quad (3)$$

$$a_k^{N,n} = N - N \left(1 - \frac{1}{N}\right)^n - n \left(1 - \frac{1}{N}\right)^{n-1} \quad (4)$$

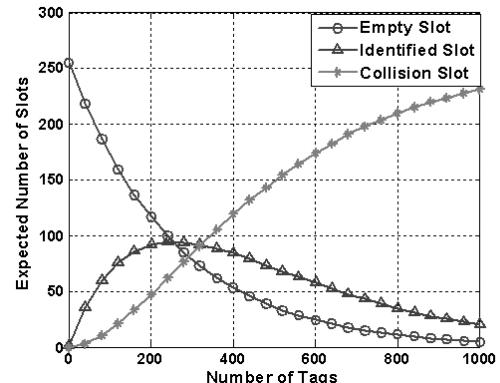


그림 2. 슬롯 개수가 256일 때 태그 개수에 따른 각 파라미터의 기댓값

그림 2는 슬롯 개수가 256일 때의 각각의 기댓값을 나타낸 것이다. 기댓값을 이용하여 태그 개수를 추정하는 방법은 체비세프 부등식(Chebyshev's inequality)에 기반을 두고 있다. 즉, 일반적으로 랜덤한 값은 대부분 기댓값 주변에 분포되므로 한 ROUND의 태그 인식 과정을 거친 후 얻어지는 빈 슬롯의 개수, 인식 슬롯의 개수, 충돌이 발생한 슬롯 개수는 실제 태그 개수를 알고 있을 때 얻을 수 있는 각각의 파라미터에 대한 기댓값 주변의 값을 가진다^[6]. 따라서 실제 한 ROUND의 태그 인식과정을 통해 측정된 파라미터를 이용하여 그에 가장 가까운 기댓값을 갖는 태그 개수를 찾음으로써 태그 개수에 대한 추정이 가능하다. 이때 사용하는 파라미터는 사용자에 따라 다양한 조합으로 적용할 수 있지만 그림 2에서 보듯이 식별 슬롯 개수의 기댓값은 비단조(non-monotonic) 함수의 형태를 가지고 단독으로 사용하기는 힘들다.

측정된 파라미터 값에 가장 가까운 기댓값을 가지는 태그 개수를 찾는 방법은 크게 두 가지로 나뉜다. 첫 번째 방법은 태그 개수에 따른 파라미터들의 기댓값을 테이블로 만들어 태그 개수를 찾는 방법이다^[7]. 이 방법은 테이블에서 측정값에 가장 가까운 기댓값을 가지는 태그 개수를 직접 사상(direct mapping)을 통해 찾아 주면 되므로 추가적인 연산이 필요 없어 빠른 추정이 가능하다. 그러나 이 방법을 시스템에 적용할 경우 테이블을 저장할 추가적인 메모리가 필요하게 된다. 따라서 메모리의 제약이 있는 시스템에서는 사용하기가 어려운 단점이 있다.

두 번째 방법은 다양한 탐색 알고리즘을 통해 측

정값에 가장 가까운 기댓값을 가지는 태그 개수를 찾는 방법이다^[6]. 즉, 임의의 태그 개수를 통해 기댓값을 구하여 측정된 값과 비교하는 과정을 수차례 반복하여 가장 작은 오차를 가지는 태그 개수를 구하게 된다. 이 방법은 테이블을 저장할 메모리는 필요하지 않지만 태그 개수를 추정하기 위해 여러 번의 시행착오를 거쳐야 하고 그때마다 기댓값을 구해야 하므로 추정 속도가 느려지게 된다.

현재 사용되는 기댓값을 이용한 태그 개수 추정 방법의 한 예로 참고문헌 [6]에서 Vogt는 다음 식 (5)를 이용하여 태그 개수를 추정하는 방법을 제안했다.

$$e_{vd}(N, c_0, c_1, c_k) = \min_n \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ \vdots \\ a_k^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_k \end{pmatrix} \right| \quad (5)$$

여기서 e_{vd} 는 추정된 태그 개수이며 c_0, c_1, c_k 는 한 ROUND 동안 측정되는 빈 슬롯 개수, 인식 슬롯 개수, 충돌이 발생한 슬롯 개수를 나타낸다. 식 (5)에서 보듯이 세 파라미터에 대한 기댓값과 측정값과의 오차를 최소화 하는 태그 개수를 찾아 그 값을 추정된 태그 개수로 정한다. 이 방법은 세 파라미터를 모두 사용하므로 매우 정확한 추정치를 얻을 수 있는 반면에 오차를 최소화 하는 태그 개수를 찾기 위해 세 가지 파라미터에 대한 기댓값을 모두 구해야 하므로 필요한 계산양이 많은 단점이 있다.

다른 예로 참고문헌 [3]에서는 충돌이 발생한 슬롯 개수만을 이용하는 방법을 제안하고 있다. 이 방법은 전체 슬롯 개수에 대한 충돌이 발생한 슬롯 개수의 비율인 Collision Ratio를 구하고 이 값은 태그 개수에 따라 달라진다는 것을 이용한다. 즉, 한 ROUND 동안 발생한 충돌 슬롯 개수를 가지고 Collision Ratio를 구한 후, 그 ROUND에서 사용된 전체 슬롯 개수는 알고 있으므로 이에 해당하는 그림 2에 제시된 것과 같은 충돌 슬롯의 기댓값으로 Collision Ratio를 구해서 가장 근접한 값에 해당하는 태그 개수를 찾음으로써 태그 개수를 추정한다. 이 방법은 Collision Ratio가 하나의 슬롯에서 충돌이 발생할 확률을 의미한다는 점에서 기준의 태그 개수에 따른 충돌 슬롯 개수의 기댓값을 사용하는 방법과는 차별성이 있기는 하지만 결국은 충돌 슬롯 개수의 기댓값에 전체 슬롯의 개수를 나누어준 값과 같으므로 기댓값을 이용하는 방법과 동일하다.

고 할 수 있다. 이 방법은 하나의 파라미터만을 사용하기 때문에 계산양은 Vogt가 제안한 방법에 비해 적지만 추정치의 정확도는 떨어진다.

3.2 단일 연산을 통한 태그 개수 추정 방법

3.1절에서 언급하였듯이 테이블을 만들어 태그 개수를 추정하는 방법은 실제 시스템에 적용 시 추가적인 메모리를 필요로 하고 탐색 알고리즘을 이용하는 방법은 추정속도가 느리다는 단점이 있다. 이를 극복하기 위한 방법으로 단일 연산을 통해 태그 개수를 추정하는 방법들이 있다. 이러한 방법은 한 번의 연산으로 태그 개수를 추정하므로 시스템의 메모리에 대한 제약이 없고 추정속도 측면에서도 빠르다고 할 수 있다.

그 첫 번째 방법으로 태그 개수의 최소치를 추정하는 방법이 있다. 태그 개수의 최소치 e_{\min} 은 다음 식 (6)과 같이 계산된다^[6].

$$e_{\min}(N, c_0, c_1, c_k) = c_1 + 2c_k \quad (6)$$

충돌이 발생한 슬롯은 최소한 두 개의 태그가 동시에 응답했기 때문에 나타나는 것인데, 식 (6)은 충돌 슬롯에는 태그가 항상 두 개가 있다는 가정 하에 계산된 것으로 추정된 태그 개수는 최소치를 의미한다^{[6],[18]}. 한편, $c_k = N - c_0 - c_1$ 을 식 (6)에 대입하여 정리하면 다음 식 (7)을 얻을 수 있다.

$$\begin{aligned} e_{\min}(N, c_0, c_1, c_k) &= c_1 + 2c_k \\ &= 2N - 2c_0 - c_1 \leq 2N \end{aligned} \quad (7)$$

최소치 추정 기법은 식 (7)에서 보는 바와 같이 추정할 수 있는 값의 최대치가 전체 슬롯 개수의 두 배가 되므로, 태그 개수가 그보다 클 경우에 상당히 큰 오차가 발생한다. 따라서, 이 방법은 태그 개수가 설정된 전체 슬롯 개수의 두 배보다 작을 때에 유용하다고 할 수 있다^[9].

두 번째 방법은 충돌이 발생한 슬롯에 응답하는 평균 태그 개수가 2.3922개임을 이용해 실제 태그 개수를 추정하는 것으로, 추정되는 태그 수는 다음 식 (8)과 같이 된다^[3].

$$n = c_1 + 2.3922 \times c_k \quad (8)$$

이 방법 역시 첫 번째 방법과 마찬가지로 c_k 의 값이 전체 슬롯 개수 이상의 값을 가질 수 없으므

로 태그 개수가 슬롯 개수의 2.3922 배 이상일 때는 태그 개수에 대한 추정의 정확도가 떨어지게 된다.

세 번째 방법은 3.1절에서 언급된 태그 개수에 따른 빈 슬롯 개수, 식별 슬롯 개수의 기댓값을 나타내는 식 (2), (3)을 이용한 방법이다. 식 (2)를 식 (3)으로 나누어 정리하면 다음 식 (9)를 얻을 수 있다.

$$n = (N-1)(a_0^{N,n}/a_1^{N,n}) \quad (9)$$

여기서 3.1절에서 언급한 체비세프 부등식에 근거하여 기댓값인 $a_1^{N,n}, a_0^{N,n}$ 대신 측정값 c_1, c_0 를 넣음으로써 태그 개수를 추정하기 위한 방법인 식 (10)을 얻을 수 있다^[9].

$$n = (N-1)(c_0/c_1) \quad (10)$$

즉, 한 ROUND를 통해 얻어지는 파라미터의 측정값인 c_0, c_1 을 식 (10)에 대입하여 태그 개수를 추정한다. 이 방법은 앞선 두 가지 방법에 비해 적용 범위가 넓다는 장점이 있다^[9].

지금까지 살펴본 대부분의 태그 개수 추정 방법은 측정 파라미터를 이용하기 때문에 실제 측정되는 파라미터의 값이 이론적인 기댓값과 얼마나 차이가 있느냐에 따라 추정 성능이 큰 차이를 보이게 된다. 즉, 태그 개수에 대한 기댓값과 측정값의 차이가 크면 클수록 추정치의 오차가 커지게 된다. 특히 세 번째 방법은 식별 슬롯 개수와 빈 슬롯 개수의 비를 이용하여 태그 개수를 추정하므로 추정치가 실측값 오차에 매우 민감하게 되는 문제가 있다. 이는 식별 슬롯 개수와 빈 슬롯 개수가 서로 독립적인 관계가 있는 것이 아니라 식별 슬롯 개수가 늘어나면 빈 슬롯의 개수가 줄어드는 반비례 관계가 있어서 둘 사이의 비를 이용하게 되면 추정치의 정확도가 실측값 오차에 크게 영향을 받게 되기 때문이다.

IV. 제안하는 태그 개수 추정 방법

기존의 태그 개수 추정 방법들은 시스템의 메모리에 대한 제약이 있거나 추정 속도가 느리다는 단점이 있었고, 이를 극복하기 위한 방법인 단일 연산을 통한 태그 개수 추정 방법들은 추정의 정확도가 떨어지는 문제가 있었다. 따라서 본 논문에서는 이러한 문제를 개선하기 위해 단일 연산을 통해 태그

개수를 추정하면서도 정확도가 높은 태그 개수 추정 방법을 제안한다.

제안하는 태그 개수 추정 방법은 한 ROUND의 태그 인식 과정을 거쳐 얻을 수 있는 파라미터 중 빈 슬롯 개수를 이용한다. 3.1절에서 한 슬롯에 태그가 응답할 확률은 이항분포를 가짐을 이용해 빈 슬롯 개수의 기댓값을 구했다. 빈 슬롯 개수의 기댓값에 대한 식 (2)를 보면 간단한 지수 함수의 형태로 나타남을 알 수 있다. 따라서 식 (2)를 정리하여 n (태그 개수)에 대한 식으로 나타내면 다음과 같다.

$$n = \frac{\log(a_0^{N,n}) - \log(N)}{\log(N-1) - \log(N)} \quad (11)$$

여기서 3.1절에서 언급한 체비세프 부등식에 근거하여 기댓값인 $a_0^{N,n}$ 대신 측정값 c_0 를 넣음으로써 제안하는 태그 개수 추정 방법인 식 (12)를 얻을 수 있다.

$$n = \frac{\log(c_0) - \log(N)}{\log(N-1) - \log(N)} \quad (12)$$

제안된 방법은 기댓값을 이용하는 방법이지만, 테이블을 만들지 않아도 되고 여러 번의 시행착오를 겪을 필요 없이 한 번의 연산을 통해 측정값과 가장 가까운 기댓값을 가지는 태그 개수를 추정하므로 추정 속도가 빠르다. 또한 3.2절에서 언급된 방법들과 달리 적용 범위에 제한이 없고, 두 파라미터의 비를 이용하지 않고 하나의 파라미터만을 사용하므로 측정값의 오차에 대한 영향을 작게 받아서 추정의 정확도가 높다는 장점이 있다.

V. 시뮬레이션 결과 및 분석

제안된 방법의 성능 검증을 위해 제안된 방법과 기존의 방법을 이용하여 태그 개수를 추정하는 시뮬레이션을 수행하여 그 결과를 비교, 분석하였다. 시뮬레이션에서 사용된 슬롯 개수는 ISO 18000-6 Type C 프로토콜에 근거하여 2^Q 개로 하였다^[5]. 여기서 Q는 1~15까지의 값을 가진다. 태그의 개수는 0개에서 무한 개까지 설정할 수 있으나 본 논문에서는 현실적인 상황을 고려하여 1000개까지로 한정하였다.

그림 3은 3.1절에서 언급한 기댓값을 이용한 추

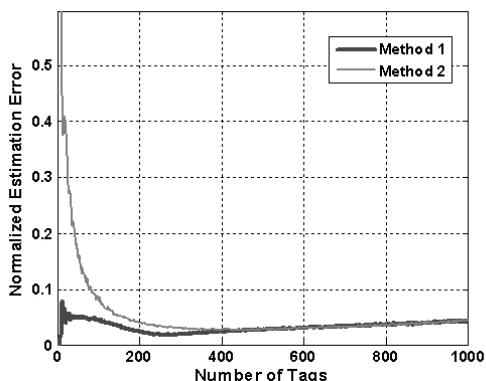


그림 3. 기댓값을 이용한 태그 개수 추정 방법의 추정 오차

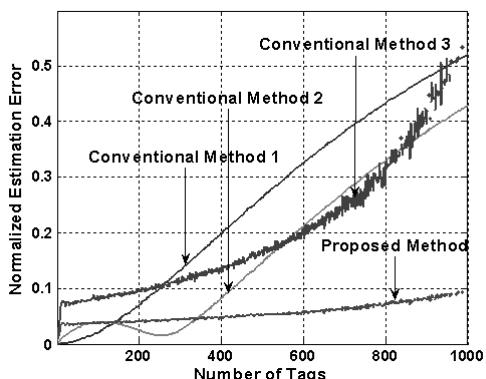


그림 4. 단일 연산으로 태그 개수를 추정하는 방법 및 제안된 방법의 추정오차

정방법으로 추정된 태그 개수의 실제 태그 개수와의 오차를 태그 개수에 따라 보여주고 있다. 결과는 1000번 반복하여 평균한 값이고 실제 태그 개수로 나누어 정규화 하였다. 추정 오차는 실제 태그 개수에서 추정치를 뺀 값의 크기로 정의하였다. 사용된 Q값은 8 (슬롯 개수 256)이다. 여기서 Method 1은 세 가지 파라미터 모두를 이용하는 방법이고 Method 2는 충돌이 발생한 슬롯 개수만을 이용하는 방법이다. 그림 3에서 보듯이 태그 개수가 많을 때는 두 가지 방법이 거의 비슷한 추정 오차를 가지지만 개수가 적어질수록 세 가지 파라미터를 모두 이용한 방법의 추정 오차가 작음을 알 수 있다.

그림 4는 3.2절에서 언급된 태그 개수 추정 방법인 식 (6)을 이용한 추정 방법으로 구해진 태그 개수 (Conventional Method 1), 식 (8)을 이용한 추정 방법으로 구해진 태그 개수 (Conventional Method

2) 및 식 (10)을 이용한 추정 방법으로 구해진 태그 개수 (Conventional Method 3), 그리고 제안된 방법으로 추정된 태그 개수 (Proposed Method)의 실제 태그 개수와의 오차를 태그 개수에 따라 보여주고 있다. 시뮬레이션에 사용된 슬롯 크기 및 오차를 구하는 방법은 그림 3에서 사용한 방법과 동일하다. 그림 4에서 보듯이 태그 개수가 적을 때는 기존 방법과 제안된 방법의 추정 오차가 거의 비슷하지만 태그 개수가 많아질수록 제안된 방법으로 추정된 태그 개수의 추정 오차가 작아짐을 알 수 있다. 그리고 그림 3과 그림 4를 비교해 보면 일반적으로 기댓값을 이용한 태그 개수 추정 방법이 좀 더 정확한 추정을 하였음을 알 수 있다. 또한 제안된 방법의 추정 오차는 기댓값을 이용한 태그 개수 추정 방법의 추정 오차에 거의 근접함을 알 수 있다. 즉, 제안하는 태그 개수 추정 방법은 기댓값을 이용한 추정 방법에 필적하는 정확성을 보이면서도 단 한번의 연산으로 태그 개수를 추정하므로 빠르고 정확한 태그 개수 추정 방법이라 할 수 있다.

VI. 결 론

본 논문에서는 RFID 시스템에서 ALOHA 기반 충돌방지 알고리즘의 성능을 향상시킬 수 있는 새로운 태그 개수 추정 방법을 제안하였다. 제안된 방법은 태그 개수에 따른 빈 슬롯 개수의 기댓값을 이용해 실제 태그 개수를 추정하는 방법이다. 시뮬레이션을 통해 제안된 방법을 기존 방법과 비교, 분석한 결과 제안된 방법이 기존 방법에 비해 작은 추정 오차를 가짐을 확인할 수 있었다. 또한 이 방법은 태그 개수를 추정하기 위해 따로 테이블을 만들지 않아 추가적인 메모리가 필요 없고 한 번의 연산으로 태그 개수를 추정하므로 추정 시간 또한 짧다는 장점이 있다. 따라서 이 방법을 ALOHA 기반 충돌 방지 알고리즘에 적용할 경우 충돌 방지 알고리즘의 성능 향상에 크게 기여할 수 있을 것으로 기대된다.

참 고 문 헌

- [1] K. Finkenzeller, RFID Handbook ; Fundamentals and applications in Contactless Smart Cards and Identification, Second Edition, John Wiley & Sons Ltd, pp. 195-219, 2003.
- [2] S. Sarma, D. Brock, and D. Engels, "Radio

- frequency identification and electronic product code,"IEEE MICRO, 2001.
- [3] 차재룡, 김재현 "RFID 시스템에서의 태그 수를 추정하는 ALOHA방식Anti-collision 알고리즘" 한국통신학회논문지, '05-9, Vol.30, No.9A, 2005.
- [4] ISO/IEC 18000-6:2003(E), Part 6: Parameters for air interface communications at 860-960 MHz, Nov. 26, 2003.
- [5] EPCglobal. "EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz - 960MHz version 1.1.0 Draft 1," EPCglobal, July 2005.
- [6] H. Vogt, "Efficient Object Identification with Passive RFID Tags," Proc. Int. Confer. on Pervasive Computing, LNCS.2414, pp. 98-113, Springer-Verlag, August 2002.
- [7] Jia Zhai, Gi-Nam, "An Anti-collision Algorithm Using Two-Functioned Estimation for RFID Tags," Computational Science and Its Applications - ICCSA 2005, LNCS.3483, pp. 702-711, May 2005.
- [8] H. Vogt, "Multiple Object Identification with Passive RFID Tags," 2002 IEEE International Conference on Systems, Man and Cybernetics, October 2002.
- [9] 권성호, 모희숙, 최길영 "슬롯 알로하 기반 RFID 시스템에서의 태그개수 추정기법," 한국통신학회 추계종합학술발표회, Vol.32, p.283, 2005.

이 지 봉(Ji-Bong Lee)



준회원

2006년 2월 부산대학교 전자전
기통신공학부 공학사
2006년 3월~현재 부산대학교 전
자공학과 석사과정
<관심분야> 적응신호처리, 디지
털 TV, 디지털 통신, RFID,
OFDM

김 완 진(Wan-Jin Kim)



준회원

2005년 2월 부산대학교 전자전
기통신공학부 공학사
2007년 2월 부산대학교 전자공
학과 공학석사
2007년 3월~현재 부산대학교 전
자공학과 박사과정
<관심분야> 적응신호처리, 디지
털 통신, RFID, OFDM

김 혁 남(Hyoung-Nam Kim)



종신회원

1993년 2월 포항공과대학교 전
자전기공학과 공학사
1995년 2월 포항공과대학교 전
자전기공학과 공학석사
2000년 2월 포항공과대학교 전
자전기공학과 공학박사
2000년 4월 포항공과대학교 전
자컴퓨터공학부 박사후 연구원
2003년 2월 한국전자통신연구원 무선방송연구소 선임
연구원
2007년 2월 부산대학교 전자공학과 조교수
2007년 3월~현재 부산대학교 전자공학과 부교수
<관심분야> 적응신호처리, 디지털 TV, 디지털 통신,
RFID, OFDM 시스템, 멀티미디어 시스템