

## 에러가 있는 환경에서 RFID 태그 충돌을 해결하기 위한 DFSA와 이진트리 프로토콜의 성능 비교 및 분석

임준봉, 이태진  
 성균관대학교 정보통신학부  
 {eomjb, tjlee}@ece.skku.ac.kr

### Performance Comparison and Analysis of DFSA and Binary Tree Protocol for RFID Tag Anti-Collision in Error-Prone Environment

Jun-Bong Eom, Tae-Jin Lee  
 School of Information and Communication Engineering, SungKyunKwan University

**Abstract** - RFID (Radio Frequency Identification) 시스템에서 수동 태그의 낮은 전송 파워 때문에 리더는 태그와의 통신이 많은 에러를 경험하게 된다. 이것은 리더가 태그의 정보를 빠르고 정확하게 인식하는 것을 어렵게 만든다. 따라서, 이러한 문제를 해결하기 위해 에러에 강한 알고리즘을 개발하고 적용할 필요가 있다. 리더가 태그의 정보를 인식하기 위해 사용하는 대표적인 프로토콜에는 ALOHA 기반의 DFSA (Dynamic Framed-Slotted ALOHA) 프로토콜과 트리 기반의 이진트리 프로토콜이 있다. 기존의 두 프로토콜 중 DFSA는 에러가 없을 때 이진트리 보다 성능이 우수하나, 에러가 존재할 때, 이진트리 보다 약 35% 정도 에러의 영향을 많이 받는 성질을 보인다. 본 논문에서는 에러가 있는 환경과 없는 환경에서의 DFSA와 이진트리 프로토콜의 성능을 분석하고, 시뮬레이션을 이용하여 이 논문에서의 성능 분석 방법이 타당함을 증명한다. 그리고 에러가 많은 환경에서는, 빠른 RFID 태그 인식을 위해 DFSA 프로토콜보다 이진트리 프로토콜을 사용할 것을 제안한다.

#### 1. 서 론

RFID (Radio Frequency IDentification)는 유비쿼터스 컴퓨팅의 실현을 위해 각광을 받고 있는 기술 중 하나로써, 리더기를 사용하여 떨어져 있는 물체의 정보를 무선으로 읽을 수 있는 기술이다. 이때, 많은 수의 태그가 존재하고 각각의 태그가 하나의 리더에게 그것의 ID를 전송하고자 한다면, 충돌이 발생하게 되고 이런 이유로 발생하게 되는 태그간의 충돌은 리더가 태그의 정보를 빠르고 정확하게 읽는 것을 방해하게 된다. 따라서, 태그간의 충돌을 어떻게 해결하느냐 하는 문제는 RFID 시스템에 있어 가장 중요한 문제 중 하나라고 할 수 있다[1].

하지만, 에러가 존재하는 환경이었다면, 태그 하나가 하나의 리더에게 자신의 정보를 바르게 전송하였다고 하더라도, 태그의 낮은 파워와 주위의 간섭과 잡음 등으로 인한 에러는 리더가 태그의 정보를 정확하게 읽을 수 없게 한다. 즉, 이러한 에러는 리더의 입장에서 보았을 때, 어떤 신호는 받았지만 알 수 없으므로 충돌이 일어났을 때처럼 리더로 하여금 태그에게 다시 정보를 요청하게 한다. 이것은 리더가 빠르고 정확하게 태그의 정보를 읽을 수 없게 만들고 RFID의 성능 저하는 가져온다. 이런 이유로 인해 태그간의 충돌 문제를 다룰 때, 에러를 고려해 주어야 한다.

본 논문의 구성은 다음과 같다. 2장에서는 대표적인 RFID 태그 충돌 방지 프로토콜을 설명하고 비교하였으며, 3장에서는 에러가 존재하는 환경에서 DFSA (Dynamic Framed-Slotted ALOHA)와 이진트리 프로토콜의 성능을 각각 분석한다. 그리고 그것을 바탕으로 에러에 의해 나타나는 각 프로토콜의 성능을 비교한다. 4장에서는 분석을 바탕으로 한 시뮬레이션 결과를 정리하고, 마지막으로 5장에서는 결론을 맺는다.

#### 2. 대표적인 RFID 태그 충돌 방지 프로토콜의 설명 및 비교

태그 충돌 방지 프로토콜에는 ALOHA 기반의 프로토콜과 트리 기반의 프로토콜이 있다. ALOHA 기반의 프로토콜 중 가장 인기 있는 것은 FSA (Framed-Slotted ALOHA)이다. 이것은 태그가 리더로 자신의 ID를 전송할 때 리더 프레임 내의 어떤 임의의 timeslot을 선택하여 그것의 ID를 전송함을 통해 충돌 확률을 낮추는 원리이다. FSA 중 가장 대표적인 것이 DFSA이다[2][4]. 이것은 FSA의 성공률이 태그의 수와 timeslot의 수가 같을 때 가장 높다는 것을 이용한다. 그러기 위해 DFSA는 이전의 프레임에서의 충돌률 등을 바탕으로 다음 프레임에서의 태그 수를 평가한다.

트리 기반의 프로토콜에서는 ALOHA 기반의 프로토콜과 다르게 태그 수를 평가할 필요가 없다. 트리 기반의 프로토콜은 충돌이 발생할 경우, 그것을 몇 개의 그룹으로 나누게 되고, 이와 같은 과정을 모든 태그가 인식될 때까지 반복하게 된다. 트리 기반의 프로토

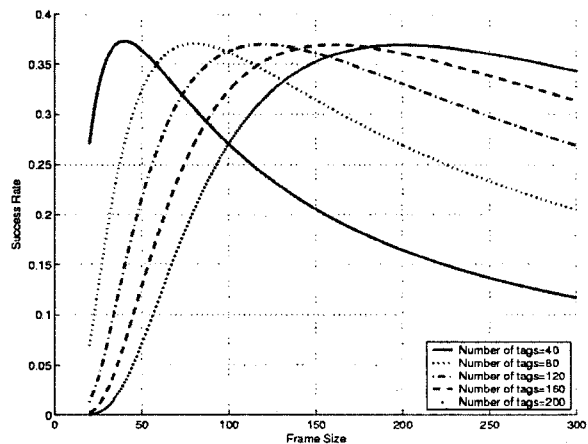
콜에서 가장 널리 사용되는 것 중 하나가 이진트리 프로토콜이다. 이 프로토콜은 충돌이 발생할 경우 모든 태그의 정보를 알 때까지 충돌을 일으킨 태그를 두 개의 그룹으로 나누게 된다. 이 프로토콜의 단점은 많은 태그가 존재할 때 초기에 너무 많은 충돌이 발생하여 timeslot의 낭비를 초래한다는 것이다[3].

#### 3. 성능 비교 분석

##### 3.1 에러를 고려할 때 DFSA의 성능 분석

DFSA에서, 리더에 의해 인식된 태그의 수는 성공할 확률과 프레임 사이즈에 의존하게 된다.  $n$ 개의 태그가 존재하고, 프레임 사이즈가  $L$ 일 때, 성공 확률은 프레임내의 하나의 timeslot에 오직 하나의 태그만이 자신의 정보가 담긴 패킷을 전송할 확률로써, 이항확률분포를 사용하여 표현할 수 있다. 이 성공률은 태그의 수와 프레임의 수가 같을 때, 최대가 된다는 것은 널리 알려져 있다[4]. 그래서 DFSA에서는 이전의 read cycle에서의 결과를 바탕으로 현재의 태그의 수를 계산하고, 다음 read cycle에서의 프레임 사이즈를 결정한다.

만약 에러가 존재하는 환경이라면, 실제 성공할 확률은 한 timeslot 안에 하나의 태그만이 정보를 전송하고 정보를 담은 패킷에는 에러가 없을 확률이 된다. 즉, 실제 성공률은 에러가 없을 경우의 성공률에 패킷 에러  $P_e$ 가 존재하지 않을 확률을 곱하여 표현할 수 있다.



〈그림 1〉 에러가 없는 환경에서 특정 태그 수에서 프레임 사이즈에 따른 성공률

그림 1은 에러가 없는 환경에서 DFSA의 성공률을 보여준다. 태그 수와 프레임 사이즈가 같고 태그의 수가 충분히 많을 때, 그림 1에서 볼 수 있듯이 이상적인 DFSA의 성공률은 약 37%가 된다. 태그 수가 적을 때는 성공률이 37%보다 높아지지만 그 오차가 적기 때문에 이상적인 DFSA의 성공률은 37%라고 할 수 있고, 패킷 에러를 고려한 성공률은  $0.37(1-P_e)$ 로 표현할 수 있다.

초기 태그의 수를 알 수 있고, 정확하게 태그의 수를 추정할 수 있는 이상적인 DFSA를 가정할 경우, 성공률을 가장 크게 하기 위해 매 read cycle에서 할당되는 timeslot의 수는 태그의 수와 같게 된다. 그래서 초기 timeslot의 수는 태그의 수  $n$ 과 같게 되며, 다음 read cycle에서 할당되는 timeslot의 수는 현재 read cycle에서 성공하지 못한 timeslot의 수와 같게 된다. 이것은 가장 크게 태그의 수에 성공하지 못할 확률 곱하여 알 수 있으며, 전체 read cycle에서 사용한 timeslot의 수  $T_{DFSA,e}$ 는 첫째 항이  $n$ 이고, 성공하지 못할 확률을 공비로 가지는 무한등비급수의 합으로써 나타낼 수 있다.

본 논문은 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-311-D00683).

$$T_{DFSA,e} = \frac{n}{1 - (1 - (1 - P_e)0.37)} = \frac{n}{0.37(1 - P_e)} \quad (1)$$

### 3.2 에러를 고려할 때 이진트리의 성능 분석

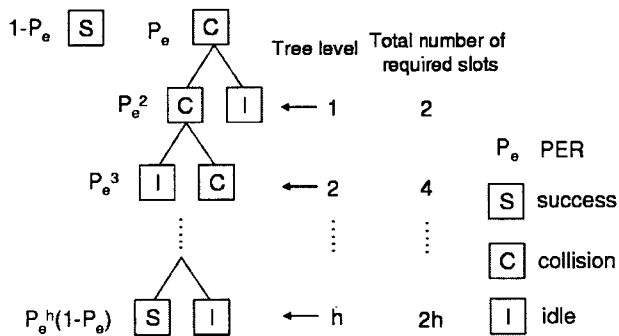
$k$ 개의 태그가 하나의 timeslot에서 충돌이 발생할 경우 이진트리는 충돌을 일으킨 태그들을 다시 2개의 그룹으로 나눈다. 이것은 태그의 수를  $n$ ,  $m$ -ary 트리를 나타내는  $m$ , 그리고 그 때 평균적으로 사용되는 프레임의 수를  $L_{m,n}$  이라고 할 때,  $n$ 개의 태그를 해결하기 위해 평균적으로 요구되는 timeslot의 수  $\alpha_n$ 은 다음과 같이 표현할 수 있다.

$$L_{m,n} = \frac{1}{1 - m^{-1-n}} \left[ 1 + \sum_{k=2}^{n-1} n C_k \frac{(m-1)^{n-k}}{m^{n-1}} L_{m,k} \right],$$

$$\alpha_n = 2 \times L_{2,n} \quad (2)$$

이 때,  $\alpha_0=2$ ,  $\alpha_1=2$  그리고  $\alpha_2=4$ 가 된다[5][6].

이진트리 프로토콜에서 하나의 태그가 자신의 정보를 하나의 timeslot에 전송하였다고 할 때, 리더는 에러가 없었다면, 성공적으로 인식을 하겠지만, 에러가 있었다면, 충돌이 일어난 것과 같이 2개의 그룹으로 다시 나누게 된다. 이렇게 2개의 그룹으로 나눈 후에도 에러의 영향을 받게 된다면, 다시 또 2개의 그룹으로 나누게 되며, 이러한 과정은 리더가 정확히 태그의 정보를 인식할 때까지 반복된다.



〈그림 2〉 이진트리 프로토콜에서 추가적인 timeslot이 사용되는 예

그림 2는 하나의 태그가 하나의 timeslot에 자신의 정보를 성공적으로 전송한 경우, 패킷 에러율  $P_e$  에 의해 이진트리가 추가적인 timeslot을 사용하는 것을 보여준다. 여기서 오직 에러에 의해서만 추가적으로 사용되는 timeslot은 그림 2에서처럼 geometric probability를 가지는 것을 알 수 있다. 그것의 평균수는 geometric probability과 트리 레벨에 따라 사용되는 총 timeslot의 수의 기대값으로 나타낼 수 있다. 태그를 인식하는 과정이 종료되려면, 모든 태그  $n$ 이 성공적으로 timeslot에 전송되는 경우가 발생해야 하므로, 전체 read cycle에서 오직 에러에 의해서 추가적으로 소비되는 timeslot의 수  $\beta_n$ 은 다음과 같이 나타낼 수 있다.

$$\beta_n = n \times \sum_{h=1}^{\infty} \left[ (1 - P_e) P_e^h \cdot 2h \right] = \frac{2nP_e}{1 - P_e} \quad (3)$$

### 3.3 에러를 고려할 때 각 프로토콜의 성능 비교

DFSA에서 오직 에러에 의해 사용되는 timeslot의 수는 패킷 에러가 존재할 때 사용한 timeslot의 수에서 패킷 에러가 존재하지 않을 때 사용한 timeslot의 수의 차를 통해 알 수 있다. 그리고 이진트리에서 오직 에러에 의해서 추가적으로 사용되는 timeslot의 수는  $\beta_n$ 이 된다. 이것을 비례적으로 나타내면, 이진트리에서 사용된 추가적인 timeslot의 수대 DFSA에서 사용된 추가적인 timeslot의 수는 약 2:2.7 이 된다. 즉, 같은 패킷 에러율이 존재할 때, 오직 에러에 의해 추가적으로 사용되는 timeslot의 수는 DFSA가 이진트리 프로토콜보다 약 35% 정도 더 많다는 것을 알 수 있다.

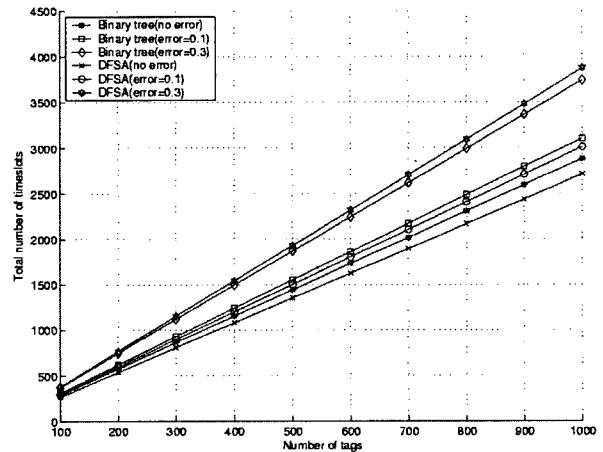
### 4. 시뮬레이션 결과

시뮬레이션 환경은 패킷 에러율을 0, 0.1, 그리고 0.3으로 설정하였으며, DFSA는 이상적인 DFSA의 경우를 가정하여 이전의 read cycle을 통해 항상 정확히 태그의 수를 추정할 수 있게 하였다. 그리고 시뮬레이션 반복 횟수는 10000번이다.

그림 3은 에러가 없는 환경과 패킷 에러율이 10%일 때는 이상적인 DFSA가 이진트리 프로토콜보다 더 좋은 성능을 보인다. 하지만 이상적인 DFSA가 아닌 실제 DFSA인 경우, 에러로 인해 태그 수

추정이 어려울 것이므로 DFSA의 성능이 뛰어나다고 말하기 힘들 것이다. 그리고 패킷 에러율이 30% 일 때는 이진트리 프로토콜이 DFSA 보다 더 적은 timeslot을 사용하는 것을 알 수 있다. 이 결과는 DFSA가 이진트리 프로토콜보다 에러가 있는 환경에서 더 많은 추가적인 timeslot을 사용한다는 것을 보여준다.

그리고 태그의 수가 1000일 때, 패킷 에러율이 0%과 30%인 상황에서 이진트리 프로토콜은 약 2881, 3738 개의 timeslot을, 그리고 DFSA는 약 2710, 3869개의 timeslot을 사용한다. 그 차이는 각각 857, 1159개로서 3장에서 분석한 것처럼 약 2:2.7 정도의 비율이 된다. 이것을 통해, 우리가 분석한 방법이 타당하다는 것을 알 수 있다.



〈그림 3〉 태그의 수에 따라 태그들 모두 인식하기 위해 사용되는 timeslot의 수

### 5. 결론

본 논문에서는 RFID 태그 충돌을 해결하기 위해 사용되어 지는 대표적인 프로토콜인 DFSA와 이진트리에 대해, 채널 에러가 존재하는 환경이 각 프로토콜의 성능에 어떠한 영향을 미치는 지에 대해 알아보았다. 태그의 정보 중 하나의 비트만 오류가 발생하여도 되지 않기 때문에 패킷 에러의 개념을 적용하였으며, 패킷 에러로 인해 추가적으로 사용되는 timeslot의 수가 프로토콜마다 달라진다는 것을 분석하였다. 분석한 결과와 시뮬레이션 결과를 통해, 에러의 영향으로 DFSA는 이진트리 보다 약 35%의 timeslot을 더 소모한다는 것을 알 수 있었다. 그래서 이 논문에서는, 에러와 충돌을 동시에 고려하였을 때, 우리는 에러가 많은 환경에서 이진트리 프로토콜을 사용하는 것이 효율적인 RFID 태그 인식을 위해 더 바람직하다는 것을 제안하였다.

### 〈참고 문헌〉

- [1] R. Want, "An Introduction to RFID Technology," IEEE Pervasive Computing, vol. 5, no. 1, pp. 25-33, Jan. 2006.
- [2] "EPCTM Radio-frequency Identification Protocols Class-1 Generation-2 UHF RFID Protocol For Communications at 860MHz-960MHz Version 1.0.9," EPCglobal, Jan. 2005.
- [3] J. Myung, W. Lee and J. Srivastava, "Adaptive Binary Splitting for Efficient RFID Tag Anti-Collision," IEEE Comm. Letters, vol. 10, no.3, pp. 144-146, Mar. 2006.
- [4] S. Lee, S. Joo and C. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," in Proc. of MobiQuitous, pp. 166-172, Jul. 2005.
- [5] A. J. E. M. Janssen and M. J. M. de Jong, "Analysis of Contention Tree Algorithms," IEEE Trans. on Information Theory, vol. 46, no. 6 pp. 2163-2172, Sep. 2000.
- [6] J.-C. Huang and T. Berger, Delay analysis of interval searching contention resolution algorithms, IEEE Trans. on Information Theory, vol. IT-31, pp. 264-273, Mar. 1985