

논문 2004-41TC-9-4

RFID 시스템에서 개선된 프레임 알로하를 이용한 고속 태그 인식 알고리즘

(High-Speed Access Technology of Tag Identification Using Advanced Framed Slotted ALOHA in an RFID System)

이 수 린*, 주 성 돈*, 이 채 우**

(Su Ryun Lee, Sung Don Joo, and Chae Woo Lee)

요 약

현재 RFID 시스템에서 해결할 가장 큰 문제 중 하나는 태그 간의 충돌로 인해 인식효율이 떨어지는 것이다. 기존의 충돌방지 알고리즘 중 프레임 알로하(Framed Slotted ALOHA)방식은 구현이 간단하다는 장점이 있으나, 태그수가 많아짐에 따라 태그 간 충돌로 인해 전체 태그를 읽는데 요구되는 슬롯의 수가 지수적으로 증가하는 단점을 갖고 있다. 본 논문에서는 이 문제를 해결하기 위해서 태그수가 많을 경우 리더의 요청 메시지에 응답하는 태그수를 제한하고, 태그수가 작을 경우 동적으로 프레임 크기를 변화시켜 고속으로 태그를 인식할 수 있는 분할 응답 프레임 알로하 알고리즘(PRFSA, Partial-Response Framed Slotted ALOHA)을 제안한다. 시뮬레이션 결과 PRFSA 알고리즘은 기존의 알고리즘보다 85~100% 성능이 향상됨을 보인다.

Abstract

In RFID system, one of the problems that we must solve is to devise a good anti-collision algorithm to improve the efficiency of tag identification which is usually low because of tag collision. Among the existing RFID anti-collision algorithms, Framed Slotted ALOHA algorithm, though simple, has a disadvantage that the number of slots used to identify the tags increases exponentially as the number of tags does. In the paper, we propose a new anti-collision algorithm called Partial-Response Framed Slotted ALOHA(PRFSA) which restricts the number of responding tags by dividing the tags into a number of groups when there are large number of tags and changes the frame size when there are small tags. Since the proposed algorithm keeps the frame size and the number of responding tags in such a way that can increase slot utilization, the algorithm shows superior performance to the existing ones. The simulation results showed that the proposed algorithm improves the slot efficiency by 85~100% compared to the existing algorithm.

Keywords : RFID, Tag Anti-Collision, Framed ALOHA, Multiple Access Technology

I. 서 론

무선 인식 기술의 중요성이 점차 커지면서 RFID (Radio Frequency Identification)는 물류, 유통 분야 및 금융서비스 등에서 현재 사용 중인 바코드(Barcode)를 대체할 기술로 주목 받고 있다^[1]. 또한 비접촉식, 대용

량 데이터의 전송가능 등의 장점을 가진 RFID 시스템은 사회 전반에 걸쳐 응용이 가능한 기술이다. 그러나 여러 가지 장점을 가진 RFID 기술은 인식한 데이터의 신뢰성 보장, RFID 기술에 대한 표준화 지연 등의 문제를 가지고 있다. 특히 현재 RFID 기술이 해결해야 할 큰 문제 중 하나는 태그 간의 충돌로 인해 태그 인식의 효율이 떨어지는 것이다^[2]. 이러한 문제를 해결하기 위하여 현재 가용 주파수 범위를 확대하여 데이터 전송속도를 높이거나, 데이터의 충돌을 최소화하여 인식률을 높이는 방법 등이 제시되고 있다. 그러나 RFID 시스템은 사용 가능한 주파수의 범위가 제한되어 가용주파수

* 학생회원, ** 정회원, 아주대학교 전자공학부
(Department of Electrical and Computer
Engineering Ajou University)

※ 본 논문은 유비쿼터스컴퓨팅프론티어사업단의 I3 과
제의 지원에 의해 수행되었습니다.

접수일자:2004년5월22일, 수정완료일:2004년9월6일

범위를 확대하는 것은 불가능하다^[3]. 따라서 태그 인식의 효율을 높이기 위해서는 데이터 충돌을 최소화하여 데이터의 재 전송 시간을 줄여야 한다.

현재 RFID 시스템은 데이터 사이의 충돌을 최소화하는 몇 가지 알고리즘(Anti-collision Algorithm)들을 사용하고 있다. 가장 널리 쓰이는 것은 시간영역에서의 다중 접속 방식인 프레임 알로하(Framed Slotted ALOHA) 계열 알고리즘과 이진 탐색(Binary search) 계열 알고리즘이다. 이중 프레임 알로하 계열의 알고리즘은 그 구현이 비교적 용이하여 많이 사용되고 있다^[2].

현재 RFID 시스템은 대부분 수동 태그(Passive Tag)를 사용하므로, 프레임 알로하 알고리즘에서 이용할 수 있는 프레임의 크기는 제한되어 있다^[4]. 따라서 리더 인식범위 내의 태그 개수가 많은 경우에도 제한된 크기의 프레임으로 태그를 인식하므로 많은 충돌이 발생하여 데이터의 빠른 인식이 불가능하다. 이 경우 응답하는 태그수를 제한할 수 있다면, 태그 충돌을 줄여 태그를 고속으로 인식할 수 있을 것이다. 본 논문에서는 태그수가 많은 경우, 리더의 요청 메시지에 응답하는 태그수를 제한하여 데이터의 충돌을 최소화하고, 효율을 극대화 하는 고속인식 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 제II장에서는 RFID 충돌방지 알고리즘 중 프레임 알로하 계열의 알고리즘을 간략히 소개하고, 각 알고리즘들이 갖는 문제점을 지적한다. 제III장에서는 제II장에서 언급한 문제점을 해결하는 알고리즘을 제안하고, 이를 분석한다. 제IV장에서는 제안한 알고리즘과 기존 알고리즘의 성능을 비교 분석하며, 마지막으로 제V장에서 본 논문을 정리한다.

II. 기존의 프레임 알로하 충돌방지 알고리즘

슬롯-알로하(Slotted ALOHA) 알고리즘이란 태그가

응답하는 시간을 고정된 몇 개의 슬롯으로 나누고 태그들이 각자 선택한 슬롯에 고유 번호를 전송하는 방식이다. 이 방식에서 리더는 태그가 자신의 데이터를 리더에 전송하는 시간간격인 슬롯(Slot)내에 고유 번호가 충돌 없이 완벽하게 전송될 때만 태그를 인식한다. 현재 사용 중인 RFID 시스템은 슬롯-알로하 계열의 하나인 프레임 알로하 방식을 이용한다. 프레임(Frame)이란 리더가 명령을 전송하고 다음 명령 전송까지의 시간을 의미하며 여러 개의 슬롯이 하나의 프레임을 구성한다. 이때 하나의 프레임으로 구성되는 태그 인식 과정을 리드 사이클(Read Cycle)이라고 한다. 본 장에서는 현재 RFID 시스템의 충돌 방지 알고리즘으로 적용 가능한 기본 프레임 알로하 알고리즘과 RFID 시스템의 특성을 고려한 몇 가지 지능적인 프레임 알로하 알고리즘에 대하여 살펴보고, 그 문제점을 정리한다.

1. 기본 프레임 알로하 알고리즘 (BFSA, Basic Framed Slotted ALOHA Algorithm)

BFSA 알고리즘은 리더와 태그사이의 통신에 사용되는 프레임의 크기가 고정되어 있는 알고리즘이다. BFSA 알고리즘에서 리더는 요청 메시지를 브로드캐스트(Broadcasting)할 때, 태그들에게 프레임 크기와 슬롯 선택에 대한 정보를 제공한다. 이 때 각 태그들은 요청 메시지와 함께 전송된 정보를 이용하여 프레임 내에서 자신이 사용한 슬롯을 선택한다^[5].

그림 1은 BFSA 알고리즘의 동작과정을 나타내고 있다. 첫 번째 리드 사이클에서 태그1과 태그3은 슬롯1번, 태그2와 태그5는 슬롯2에 자신의 고유번호를 전송함으로써 태그 충돌이 발생하였다. 그 결과 태그1,2,3,5는 인식되지 않았으므로 리더의 두 번째 요청메시지에 다시 랜덤하게 슬롯을 선택하여 응답하였다. 태그4는 첫 번째 리드 사이클에서 인식되었으므로 리더는 태그4에게 이후의 요청메시지에 대해 응답하지 않도록 명령한다.

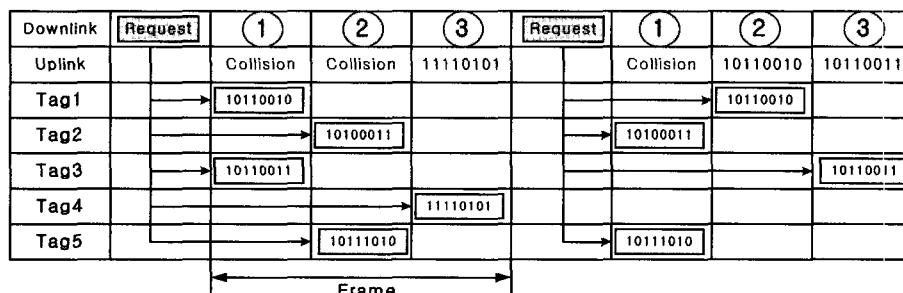


그림 1. BFSA 알고리즘의 동작과정
Fig. 1. The process of BFSA algorithm.

이때 각 리드 사이클에서 사용된 프레임의 크기는 슬롯 3개로 동일하게 유지된다.

BFSA 알고리즘은 고정된 프레임 크기를 사용하므로 그 구현이 간단하나, 태그 인식 효율이 떨어지는 단점이 있다. 예를 들면, 태그수가 많은 경우 작은 크기의 프레임을 사용한다면 리드 사이클이 여러 번 반복되더라도 하나의 태그도 인식하지 못할 수 있으며, 태그수가 작은 경우 큰 크기의 프레임을 이용한다면 불필요한 슬롯의 낭비가 발생한다.

2. 동적 프레임 알로하 알고리즘 (DFSA, Dynamic Framed Slotted ALOHA Algorithm)

DFSA 알고리즘이란 데이터의 충돌 등을 고려하여 프레임의 크기를 변화시키는 방식이다. 따라서 고정된 프레임 크기를 사용하는 BFSA 알고리즘에서 나타나는 슬롯 낭비의 문제를 해결할 수 있다. DFSA 알고리즘은 그 구현방법에 따라서 많은 변형 알고리즘이 존재한다^[2]. 본 절에서는 그 중 증가 방식(Increase)의 두 가지 구현 방법에 대해서 소개한다.

DFSA의 증가 방식 중 첫 번째 구현 방법은 전 단계의 태그 인식과정에서 얻은 슬롯의 충돌 여부를 이용하여 프레임 크기를 조정한다. 즉 충돌 발생 슬롯 수가 프레임의 일정 비율 이상인 경우에 프레임 크기를 증가시키고, 프레임 중 일정비율 이상 빈 슬롯이 발생한 경우에 프레임 크기를 감소시킨다. 이때 최소 프레임 크기를 이용하여 태그 인식을 시작하므로, 태그 개수가 작은 경우 프레임 크기를 증가 시키지 않고 효율적으로 태그를 인식 할 수 있다. 또한 태그가 많은 경우는 프레임의 크기를 증가시켜 태그를 인식하고 응답하는 태그수가 적어지면 프레임의 크기를 줄임으로써 효율적인 태그 인식이 가능하다.

DFSA의 증가 방식 중 두 번째 구현방법은 슬롯 1개나 2개로 구성된 작은 프레임으로 태그 인식을 시작하여 태그 1개가 인식될 때까지 프레임의 크기를 증가시킨다^[2]. 만일 태그 인식 과정에서 하나의 태그가 인식되면 그 프레임은 중단된다. 그리고 다시 작은 프레임으로 시작하는, 하나의 태그를 인식하기 위한 과정을 반복적으로 수행한다.

DFSA 알고리즘은 리더의 인식범위 내의 태그 개수를 고려하여 프레임의 크기를 조절하므로 태그를 효율적으로 인식할 수 있다는 장점이 있다. 그러나 태그수가 지나치게 많은 경우에는 프레임 크기의 조정만으로는 태그 충돌을 줄일 수 없다. 따라서 DFSA 알고리즘

역시 태그수가 많은 경우에는 태그를 효율적으로 인식할 수 없다는 단점을 가진다.

3. 태그 수 추정 프레임 알로하 알고리즘 (AFSA, Advanced Framed Slotted ALOHA Algorithm)

AFSA 알고리즘은 기존의 BFSA 알고리즘을 개선한 방식중 하나로, 리더의 인식범위 내의 태그 수를 추정하고, 추정된 개수의 태그를 읽기 위한 최적의 프레임 크기를 결정하여 태그 인식 과정을 수행하는 충돌 방지 알고리즘이다. 이때 태그 수는 리더가 리드 사이클을 수행한 후 얻어진 충돌 발생 슬롯 수, 빈 슬롯 수, 태그 인식 슬롯수를 이용하여 추정된다^[4].

AFSA 알고리즘은 태그 수 추정을 위해서 식(1)의 태그 수 추정함수를 이용한다. 일반적으로 모든 랜덤한 실측값인 X 는 대부분 X 의 평균 기대값의 주변에 분포한다. 이를 증명하는 식이 Cherbyshev's inequality이며 태그 수 추정 함수는 이 성질을 이용하고 있다. 따라서 태그 수 추정 함수는 태그수를 변경시켜가면서 실측값과 기대값의 차이를 측정하여 차이가 가장 적게 나타나는 값을 리더의 명령에 응답한 태그수로 추정한다^{[4][6]}.

태그 수의 추정은 전단계의 리드 사이클에서 얻어진 태그를 읽은 슬롯의 수(c_1), 충돌이 발생한 슬롯의 수(c_k), 빈 슬롯의 수(c_0)와 리드 사이클에서 사용되었던 프레임 크기(N)를 이용하여 식 (1)으로 계산된다. 태그의 수는 Binomial distribution을 이용한 슬롯 상태의 평균 기대값과 실측된 슬롯 상태의 오차를 최소로 하는 값으로 정해진다^[4].

$$\epsilon_{sd}(N, c_0, c_1, c_k) = \min \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \quad (1)$$

식 (1)에서 $a_0^{N,n}$ 는 프레임 크기 N 을 이용하여 리드 사이클을 거쳤을 때 나올 수 있는 빈 슬롯의 기대값이며, $a_1^{N,n}$ 와 $a_{\geq 2}^{N,n}$ 은 각각 태그가 인식된 슬롯의 수 및 태그 충돌이 발생한 슬롯 수의 기대값을 의미한다.

AFSA 알고리즘은 태그 수를 추정하여 프레임의 크기를 결정하므로 기존의 다른 알고리즘에 비해 효율적으로 태그를 인식할 수 있다는 장점이 있다. 그러나 AFSA 알고리즘도 프레임의 크기를 조정하여 태그 인식 효율을 높이는 방법이므로 태그수가 많은 경우에는 충돌을 줄일 수 없다는 단점이 있다.

III. 분할 응답 프레임 알로하 알고리즘(PRFSA, Partial-Response Framed Slotted ALOHA)

기존의 프레임 알로하 알고리즘은 효율적인 태그 인식을 위해 프레임 크기를 변경 시키는 방법을 이용한다. 그러나 태그수가 최대 프레임 크기보다 많은 경우, 프레임 크기 변경만으로는 태그 충돌을 줄일 수 없어 태그 인식 시간이 증가한다. 본 장에서는 이러한 문제점을 해결하기 위해 기존의 알고리즘을 개선한 분할 응답 프레임 알로하 알고리즘(PRFSA, Partial-Response Framed Slotted ALOHA)을 제안한다.

1. PRFSA 알고리즘의 동작 원리

프레임 알로하 충돌방지 알고리즘은 태그 충돌이 많이 발생하는 경우 프레임 크기를 증가시켜 충돌을 줄이고 있으나, 수동 태그를 사용하는 경우에는 최대 프레임 크기가 제한된다. 따라서 태그 충돌을 줄이기 위해 태그 수에 따라 프레임의 크기를 계속 증가시킬 수 없다. 이런 경우 리더의 요청에 응답하는 태그 수를 최대 프레임 크기에 맞게 조절한다면 태그의 충돌을 줄여 효율을 높일 수 있다. 또한 태그 수가 작다면 프레임 크기를 조절함으로써 불필요한 슬롯의 사용을 줄이면서 태그 인식 효율을 높일 수 있다.

PRFSA 알고리즘은 태그 수를 추정하여 추정된 태그 수에 따라 시스템 효율이 높게 유지 될 수 있도록 응답하는 태그 수를 제한하거나(태그 응답 제한 과정), 프레임의 크기를 조절한다(프레임 크기 변경 과정). 시스템 효율은 추정된 태그 수와 사용된 프레임 크기를 이용하여 구하며, 태그 수 추정은 앞서 설명한 AFSA 알고리즘에서 사용한 추정 방법을 이용하여 행한다. 이때 시스템 효율이란 전체 프레임 중 충돌 없이 태그를 인식한 슬롯의 비율을 의미한다.

태그 응답 제한 과정은 추정을 통해 얻은 인식되지 않은 태그 수(Tag_{remain})가 일정 값($Tag_{threshold}$) 이상 일 때 동작하며, 최대 프레임 크기를 사용할 때 최적의 효율을 갖도록 응답하는 태그의 수를 제한한다. 이때 응답하는 태그 수는 인식되지 않은 전체 태그 수의 일정 비율로 나타나며, 이 비율이 결정되면 응답할 태그는 모듈로 동작(modulo operation)을 이용하여 결정된다. 모듈로 동작이란 태그의 고유 번호를 일정한 수 즉 분할 그룹(tag group)수로 나누어서 나머지가 0이 되는 태그들만 리더의 명령에 응답하는 방법이다.

프레임 크기 변경 과정은 Tag_{remain} 수가 일정 태그 수($Tag_{threshold}$)보다 작을 때 동작하며, 추정된 태그 수를 최적의 효율로 인식할 수 있도록 프레임의 크기(N)를 변경한다. $Tag_{threshold}$ 와 분할 그룹수, N 의 값을 구하는 과정은 III-2에서 설명한다.

그림 2는 PRFSA 알고리즘의 동작 과정을 보이고 있다. 태그를 인식하는 최초의 리드 사이클에서 리더는 초기값으로 정해진 프레임을 이용하여 태그를 읽는다. 이 리드 사이클의 수행결과, 빈 슬롯의 수, 태그 1개가 포함된 슬롯의 수, 데이터의 충돌이 발생한 슬롯의 수(c_0, c_1, c_k)를 구할 수 있다. 이 값을 이용하여 태그수를 추정한 후 인식되지 않은 태그 수(Tag_{remain})가 일정 태그 수($Tag_{threshold}$)보다 큰 경우는 응답하는 태그 수를 제한하는 태그 응답 제한 과정으로 들어가고, 작은 경우는 프레임 크기 변경 과정으로 들어간다.

태그 응답 제한 과정에서 리더는 시스템 효율을 가장 높게 하는 태그 수로만 태그들이 응답하도록 분할 그룹

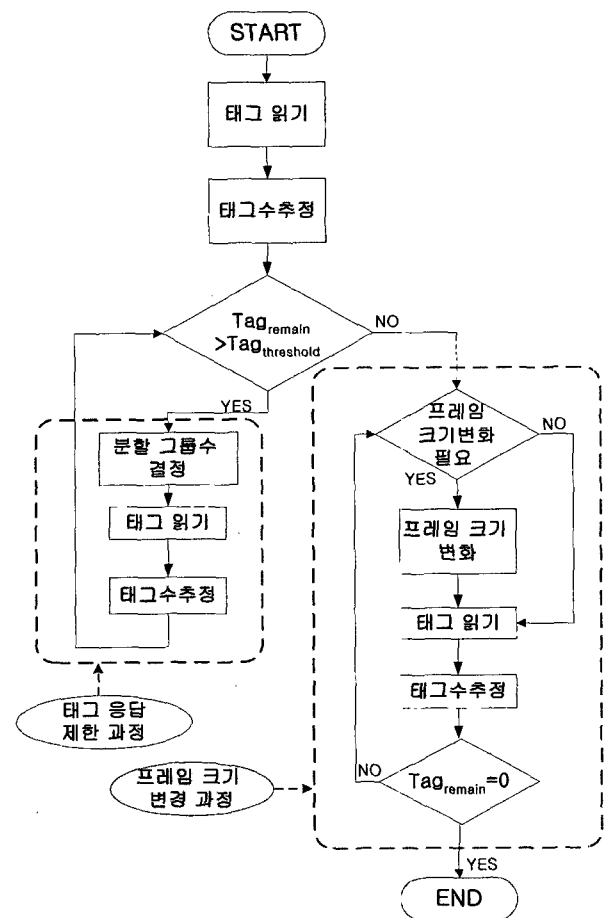


그림 2. PRFSA 알고리즘의 동작 과정
Fig. 2. The flow chart of PRFSA algorithm.

수를 결정한다. 리더는 요청 메시지를 태그에 브로드캐스트 할 때 결정된 분할 그룹수와 랜덤 변수를 함께 전송한다. 요청 메시지를 받은 태그들은 리더로부터 받은 랜덤 변수와 자신의 고유번호를 조합하여 새로운 수를 만든다. 그리고 이 수를 분할 그룹수로 나누어 나머지가 0이 되는 태그들만 응답하게 된다. 이때 전체 응답 가능한 태그 수 추정은 분할 그룹수와 한 리드 사이클에서 추정된 태그수를 곱하여 구한다. 그리고 매 리드 사이클마다 태그수를 추정하고 분할 그룹수를 새롭게 결정한다.

프레임 변경 과정은 다음의 순서로 이루어진다. 리더는 요청메시지와 랜덤 변수 그리고 최대 시스템 효율을 갖도록 하는 프레임 크기(N) 등의 파라미터를 브로드캐스트 한다. 태그들은 리더로부터 받은 파라미터와 자기의 고유번호를 이용하여 다음 프레임에서 자신이 사용할 슬롯번호를 정한다. 그리고 결정한 슬롯에 자신의 데이터를 전송한다. 매 리드 사이클 후 태그수를 추정하여 추정된 태그 수에 가장 적합한 프레임크기를 결정한다.

2. PRFSA 알고리즘의 수학적 분석

최대 프레임 크기로 동작하는 RFID 시스템은 읽어야 할 태그 수가 일정 수(Tag_{threshold})를 넘어서면 시스템 효율이 떨어진다. 따라서 PRFSA 알고리즘은 RFID 시스템의 높은 효율성을 유지하기 위해 태그 응답 제한 과정과 프레임 크기 변경 과정으로 구성된다. 본 절에서는 PRFSA 알고리즘을 위한 기본적인 수학적 배경과 Tag_{threshold}, 분할 그룹수, N등의 파라미터를 구한다.

태그는 프레임내의 한 슬롯에 임의로 응답한다. 프레임 크기가 N, 태그의 수가 n일 때, 하나의 슬롯 내에 r 개의 태그가 존재할 확률은 Binomial distribution을 따른다^{[4][7]}.

$$B_{n, \frac{1}{N}}(r) = \binom{n}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{n-r} \quad (2)$$

그러므로 프레임 크기 N인 RFID 리더가 한 리드 사이클 동안 읽을 수 있는 평균 태그 수는 다음과 같다.

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

따라서 시스템 효율은 식 (4)으로 나타낼 수 있다.

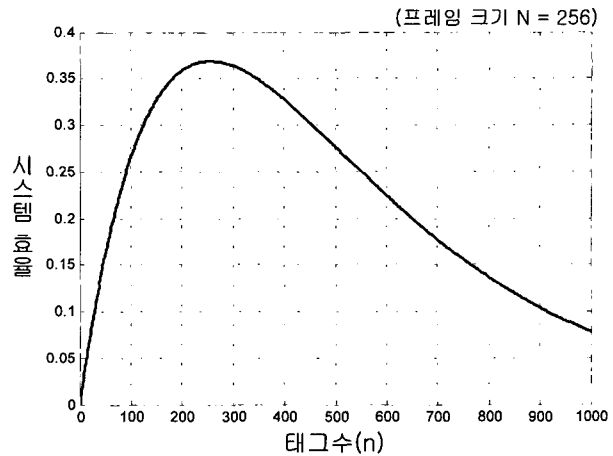


그림 3. 시스템 효율
Fig. 3. System efficiency.

$$\text{시스템 효율} = \frac{\text{태그를 인식한 슬롯수}}{\text{프레임의 전체 슬롯수}} = a_1^{N,n}/N \quad (4)$$

그림 3은 프레임 크기 N이 256일 때, 시스템 효율을 태그 수에 따라 나타내고 있다.

시스템의 효율이 가장 높을 때의 태그 수는 식 (3)을 미분하여 그 미분결과를 0으로 하는 태그수를 구함으로써 얻는다.

$$\begin{aligned} \frac{d(a_1^{N,n})}{dn} &= \left(1 - \frac{1}{N}\right)^{n-1} + n \left(1 - \frac{1}{N}\right)^{n-1} \ln \left(1 - \frac{1}{N}\right) \quad (5) \\ &= \left(1 - \frac{1}{N}\right)^{n-1} \left\{1 + n \ln \left(1 - \frac{1}{N}\right)\right\} \\ &= 0 \end{aligned}$$

식 (5)의 결과로부터 RFID 시스템의 프레임 크기에 따른 최적의 응답 태그 수는 다음과 같다.

$$n = - \frac{1}{\ln \left(1 - \frac{1}{N}\right)} \quad (6)$$

식 (6)을 이용하여 태그 수가 n인 경우, 최적의 프레임의 크기는 식(7)을 이용하여 구할 수 있다.

$$N = \frac{1}{1 - e^{-\frac{1}{n}}} = \frac{e^{\frac{1}{n}}}{e^{\frac{1}{n}} - 1} \quad (7)$$

식 (7)을 태그수가 일정 규모(단, n ≫ 1) 이상일 때 Taylor series를 이용하여 전개하면 다음과 같다.

$$N? \frac{1 + \frac{1}{N}}{1 + \frac{1}{N} - 1} = n + 1 \tag{8}$$

식 (8)에 의해 응답 태그수가 63,127,255 일 때 최적의 프레임 크기는 64, 128, 256 이다. 그러므로 최적 응답 태그 수 n 은 프레임의 크기와 근사한 값을 갖는다.

그림 4와 식(8)으로부터 RFID 시스템이 높은 효율성을 갖기 위해 태그 수에 따라 프레임 크기(N)를 조절하거나 최대 프레임 크기에 맞게 분할 그룹수를 이용하여 응답하는 태그수를 제한하여야 한다. 다음은 본 알고리즘의 동작을 위해 태그 응답 제한 과정과 프레임 크기 변경 과정을 구분하는 $Tag_{threshold}$, 응답하는 태그수를 제한하기 위한 분할 그룹수, 태그 수에 따른 최적의 프레임 크기(N)을 구한다.

본 논문의 PRFSA 알고리즘은 분할 그룹수로 2의 지수승(2,4,8,...)을 이용한다. 2의 지수승을 이용하는 경우, RFID 태그는 쉬프트 레지스터(Shift Register)를 이용하여 자신의 그룹을 간단하게 결정할 수 있기 때문이다. 태그 응답 제한 과정에서 2의 지수승으로 구성된 분할 그룹수를 적용하면 기존에 응답하는 태그수를 1/2, 1/4, 1/8로 줄일 수 있다. 이때 분할 그룹수를 사용하여 태그응답을 제한한 경우의 시스템 효율은 태그응답을 제한하지 않은 경우보다 낮아지면 안 된다. 만약 시스템의 효율이 더 낮아진다면 태그응답 제한을 통해 얻을 수 있는 이득이 없기 때문이다. 이 사실을 바탕으로 (4)를 이용해 응답 가능한 최적의 태그수를 얻을 수 있다. (단, 최대 프레임의 크기는 256이라 가정한다.)

$$\frac{a_1^{256, n/2}}{256} \geq \frac{a_1^{256, n}}{256} \tag{9}$$

따라서 식(9)을 풀이하면 다음과 같다.

$$\frac{n}{2} \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{\frac{n}{2} - 1} \geq n \cdot \left(\frac{1}{256} \right) \left(1 - \frac{1}{256} \right)^{n - 1} \tag{10}$$

그러므로

$$n \geq 354 \tag{11}$$

식 (11)으로부터 $Tag_{threshold}$ 값은 354임을 알 수 있다. 따라서 추정을 통해 얻은 인식되지 않은 태그의 수 (Tag_{remain})가 354개 이상일 때는 분할 그룹수를 이용하여 태그응답을 제한한다. 추정된 Tag_{remain} 에 따른

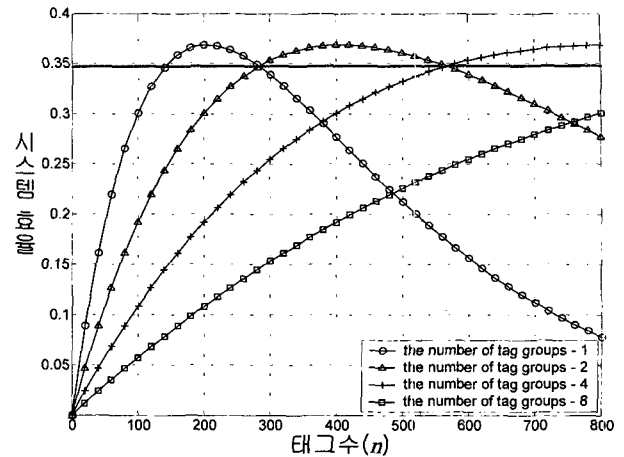


그림 4. 분할 그룹 수에 따른 시스템 효율
Fig. 4. The system efficiency vs. the number of tag groups.

분할 그룹수의 결정은 식(9)을 이용하여 구할 수 있다. (단, 분할 그룹수 1은 태그응답을 제한하지 않는 경우이다.) 즉, 분할 그룹수로 4가 적용되는 시점은 식(9)의 $n/2$ 대신 $n/4$ 을 대입하면 구할 수 있으며, 분할 그룹수로 8이 적용되는 시점은 $n/2$ 대신 $n/8$ 을 대입하면 구할 수 있다. 그 이상의 분할 그룹수도 유사한 방식을 이용하여 얻을 수 있다. 식 (12)은 분할 그룹수가 1,2,4 그리고 8인 경우, 태그 개수의 적용범위를 나타내고 있다.

$$= \begin{cases} 1, & \text{if } Tag_{remain} \leq 354 \\ 2, & \text{if } 355 \leq Tag_{remain} < 708 \\ 4, & \text{if } 708 \leq Tag_{remain} < 1417 \\ 8, & \text{if } 1417 \leq Tag_{remain} < 2832 \end{cases} \tag{12}$$

분할 그룹수로 1,2,4 그리고 8을 이용한 경우 분할 그룹수에 따른 시스템 효율을 그림 4에 나타내었다.

식 (12)에 따라 태그응답을 제한하는 PRFSA 알고리즘은 한번에 응답하는 태그수가 354개 이하이므로 리더의 효율은 그림 4에 표시된 직선의 윗부분(34.6%)에서 항상 유지된다. 그러므로 기존의 방법에서는 태그수가 많아짐에 따라 급격히 시스템의 효율이 떨어지지만, PRFSA 알고리즘의 방법은 높은 효율을 계속 유지할 수 있다.

태그 수 추정을 통해 얻은 Tag_{remain} 값이 $Tag_{threshold}$ (354)보다 작은 경우, Tag_{remain} 값에 대해 최적의 프레임 크기를 찾아야 한다. 식(4)을 이용하여 프레임 크기를 달리하면서 태그 수에 따른 시스템의 효율을 그림 5에 나타내었다. 최적 프레임 크기는 그림 5에 나타난 프레임의 크기에 따른 시스템의 효율을 이용하여 구한다.

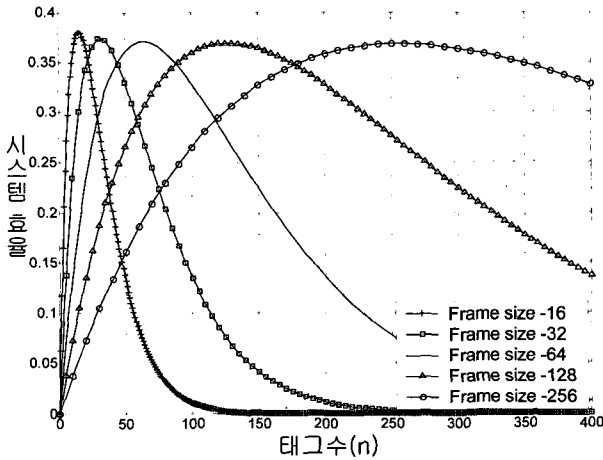


그림 5. 프레임 크기에 따른 시스템 효율
Fig. 5. The system efficiency vs. frame size.

프레임 크기에 따른 시스템 효율을 나타낸 그림 6에서 남은(인식해야 할) 태그 수에 따라 효율이 가장 높은 프레임의 크기를 선택한다. 그러므로 그림 6의 결과를 바탕으로 k 번째 리드 사이클의 프레임 크기 N_k 는 다음과 같이 결정된다.

$$\text{Frame size} = \begin{cases} N_k = 256, & \text{Tag}_{\text{remain}} \geq 177 \\ N_k = 128, & \text{Tag}_{\text{remain}} \geq 82 \\ N_k = 64, & \text{Tag}_{\text{remain}} \geq 41 \\ N_k = 32, & \text{Tag}_{\text{remain}} \geq 20 \\ N_k = 16, & \text{else} \end{cases} \quad (13)$$

프레임의 크기가 16보다 작은 경우는 오버헤드가 너무 크기 때문에 더 이상 줄이는 것은 무의미하다. 그러므로 PRFSA 알고리즘은 프레임 크기를 16까지만 변화도록 하였다.

IV. PRFSA 알고리즘의 성능 분석

본 논문에서 제안한 PRFSA 알고리즘은 시스템 효율을 사용하여, 태그수가 많은 경우 모듈로 방법을 이용하여 응답하는 태그수를 제한하고, 태그수가 작은 경우 프레임 크기를 변화시킴으로써 태그 인식의 효율을 높이고, 고속 인식이 가능케 하고 있다. 따라서 본장에서는 태그 수의 변화에 따른 PRFSA 알고리즘의 성능을 분석한다. 태그수가 최대 프레임 크기(256개)보다 큰 경우 태그 응답 제한 과정이 사용되므로 태그수를 1000개까지 증가시켜 기존 알고리즘과 성능을 비교한다. 특히 태그 수가 200개 이하인 경우는 그 수가 최대 프레임 크기보다 작으므로 PRFSA 알고리즘은 프레임 크기 변경 과정을 사용한다. 따라서 프레임 크기 변경 과정을

사용하는 경우의 PRFSA 알고리즘과 기존의 알고리즘의 성능을 비교, 분석하기 위하여 전체 태그 인식 과정에서 이 부분을 확대하여 살펴본다.

PRFSA 알고리즘과 비교하는 알고리즘으로 BFS 알고리즘과 DFSA 알고리즘의 증가방식 중 첫 번째 구현 방법을 사용한다.(이하 '증가 방식'이라고 한다.) 각 알고리즘에서 사용하는 최대 프레임 크기는 256개의 슬롯이다. BFS 알고리즘은 256개의 슬롯으로 구성된 프레임을 이용하며, DFSA 알고리즘 중 증가방식은 프레임 크기가 16슬롯인 경우부터 시작하여 최대 증가 프레임까지 증가 시키도록 하였다. 이때 전체 프레임의 70% 이상 충돌이 발생한 경우 프레임의 크기를 현재 프레임 크기의 2배로 증가시키고, 전체 프레임의 30% 이상 빈 슬롯이 존재하는 경우에 현재 프레임의 절반으로 프레임 크기를 줄인다. PRFSA 알고리즘에서 초기 프레임 크기는 128개의 슬롯을 이용하는 것으로 설정한다. 모든 결과는 평균적인 결과를 사용하였으며, 태그 인식의 동작은 리더가 요청 메시지를 브로드 캐스트 했을 때 응답하는 태그들이 없는 경우에 끝낸다고 설정한다.

1. 태그 수가 최대 프레임 크기보다 작은 경우

본 절에서는 태그수가 최대 프레임 크기(256개)보다 작은 경우의 각 알고리즘의 성능을 분석한다. 따라서 그림 6에서는 태그 수의 증가가 200개 미만인 경우의 각 알고리즘의 성능을 나타내고 있다. 슬롯 256개인 프레임을 사용하는 BFS 알고리즘은 최소한 2번의 리드 사이클이 필요하므로 기본적으로 512개의 슬롯을 사용하게 된다. 따라서 다른 알고리즘에 비하여 태그수가 적은 경우 태그를 인식하기 위하여 필요한 슬롯수가 많다. 또한 읽어야 하는 태그수가 적은 경우에도 큰 프레임을 이용하므로 슬롯의 낭비가 심함을 보여준다.

DFSA 알고리즘 중 증가방식은 태그수의 증가에 비례하여 태그를 모두 인식하기 위해 필요한 슬롯수가 선형적으로 증가하고 있다. 이것은 증가방식이 작은 프레임 크기인 16슬롯 프레임부터 태그를 인식하기 시작하여 태그수의 증가에 따라 프레임의 크기를 증가시켜 충돌을 줄이는 방식을 사용하기 때문이다. 태그수의 증가에 따른 사용 슬롯의 증가폭은 BFS 알고리즘과 유사한 형태를 취한다. 그 이유는 증가 방식 알고리즘의 프레임 크기가 태그수의 증가에 따라서 빠르게 증가하여 최대 증가 프레임까지 증가하기 때문이다.

PRFSA 알고리즘의 경우, 초기 프레임의 크기를 128 슬롯인 프레임을 이용하였으므로 초기 사용 슬롯의 수

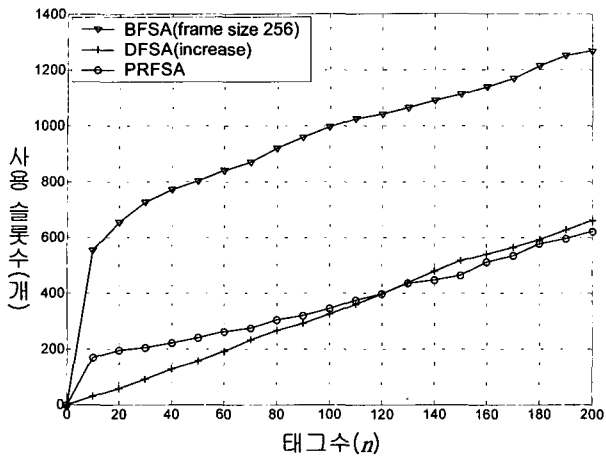


그림 6. 태그 수가 200개 미만인 경우의 태그 인식에 사용되는 총 슬롯 수
 Fig. 6. The total number of slots used to identify under 200 tags.

는 증가방식보다는 많다. 그러나 태그수가 120개 이상인 경우, PRFSA 알고리즘이 더 우수한 성능을 보이고 있다. 이처럼 PRFSA는 초기 추정과정을 마치면, 리더는 최적의 시스템 효율을 갖기 위해 동적으로 프레임의 크기를 변화시키기 때문에 일정 태그 수(120개) 이상이 되면 사용되는 슬롯 수가 다른 알고리즘에 비해 빠르게 줄어들음을 알 수 있다.

2. 태그수가 최대 프레임 크기보다 큰 경우

본 절에서는 태그 수가 최대 프레임 크기(256개)보다 큰 경우의 각 알고리즘의 성능을 비교분석한다. 따라서 그림 7은 태그수를 1000개까지 증가시킨 경우의 각 알고리즘의 성능을 나타내고 있다. BFSA 알고리즘과 DFSA 알고리즘 중 증가방식은 태그수의 증가에 따라 사용 슬롯의 수가 지수적으로 증가하고 있음을 볼 수 있다. 이것은 최대 증가 프레임에 비해 태그수가 지나치게 많아 태그를 인식하지 못하는 리더 사이클의 수가 증가하기 때문이다. 다만 증가방식의 경우, 읽어야 하는 태그수가 감소함에 따라 프레임 크기를 줄여 슬롯의 낭비를 줄이고 있으므로 BFSA 알고리즘보다 사용한 슬롯의 수가 적게 나타난다.

PRFSA 알고리즘은 태그수가 증가함에 따라 사용 슬롯의 수가 선형적으로 증가하고 있다. 이는 리더의 요청 메시지에 응답하는 태그의 수를 최대 프레임 크기에 맞추므로써 태그 인식률을 계속 높은 상태로 유지하기 때문이다. PRFSA 알고리즘의 성능을 다른 알고리즘과 비교하면, BFSA 알고리즘에 비하여 100%정도, DFSA 알고리즘과는 85%정도 성능이 향상됨을 보이고 있다.

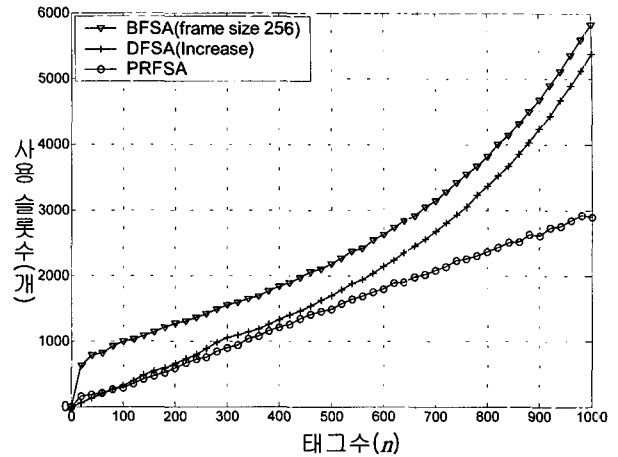


그림 7. 태그수가 1000개까지 증가한 경우의 태그 인식에 사용되는 총 슬롯 수
 Fig. 7. The total number of slots used to identify upto 1000 tags.

그림 6과 7에서는 사용한 슬롯수를 이용하여 알고리즘의 성능을 나타냈다. 그러나 실제로는 슬롯수와 인식 시간이 100% 비례하지는 않는다. 그 이유는 매 리더 사이클 당 한번씩 발생하는 리더의 요청메시지 전송시간이 고려되지 않았기 때문이다. 그러나 리더의 요청메시지는 매 리더 사이클마다 1번만 발생하며, 프레임의 크기가 큰 경우에는 한 리더 사이클에서 요청메시지가 차지하는 비중이 작을 것이므로 인식시간은 위의 곡선과 매우 유사하게 나타날 것이다^{[5][6]}.

V. 결 론

RFID 시스템이 기존의 기술을 대체하기 위해서는 많은 태그로부터의 보다 빠른 데이터의 인식이 요구된다. 그러나 기존의 알고리즘은 태그수가 증가함에 따라 필요한 슬롯의 수가 지수적으로 증가함으로써 고속의 인식이 불가능하였다. 본 논문에서는 이 문제를 해결하기 위하여 PRFSA 알고리즘을 제안하였다. PRFSA 알고리즘은 시스템 효율의 개념을 이용한 충돌방지 알고리즘으로 시스템 효율을 34.6% 이상의 높은 값으로 유지시키기 위해 프레임의 크기를 변화 시키거나 모듈로 방법을 이용하여 응답하는 태그수를 제한한다. 따라서 리더가 지원하는 최대 프레임크기에 비해 태그수가 많더라도 사용 슬롯수가 선형적으로 증가하게 된다. 시뮬레이션 결과, 태그 1000개를 인식하는 경우 PRFSA 알고리즘은 BFSA 알고리즘에 비하여 100%의 성능 향상을 보이고 있으며 DFSA 알고리즘 중 증가방식에 비해서는 85%의 성능 향상을 나타내고 있다. 그러므로 PRF

SA 알고리즘은 기존의 알고리즘에 비하여 월등한 성능을 보임을 알 수 있다.

본 논문에서 제안한 PRFSA 알고리즘은 시스템 구현의 단순화를 위하여 분할 그룹수로 2의 지수승만을 사용하고 있다. 그러나 RFID 태그의 성능이 더 향상된다면 분할 그룹수로 2의 지수승만이 아니라 정수 값을 사용할 수 있을 것이다. 이 경우 응답하는 태그의 개수를 프레임의 크기에 더욱 근접시킬 수 있으므로 제안한 알고리즘의 태그 인식 효율은 더 향상될 것으로 기대된다.

[5] PHILIPS Semiconductor, "I-CODE1 System Design Guide", *Technical Report*, May 2002.
 [6] Harald Vogt, "Multiple Object Identification with Passive RFID Tags", 2002 IEEE International Conference on Systems, Man and Cybernetics, Oct. 2002.
 [7] Walter A Shewhart, Samuel S Wilks, *An Introduction to Probability Theory and Its Application-Second Edition*, Wiley publications, 1960.

참 고 문 헌

[1] 김상태, "RFID 기술개요 및 국내외 동향분석", 전자부품연구원 전자정보처리센터, Aug. 2003.
 [2] Klaus Finkenzeller, *RFID handbook - Second Edition*, John Wiley & Sons, 2003, pp.195-219
 [3] 장동원, 조평동, "RFID 기술기준 도입을 위한 기술 분석", 전자통신동향분석, Dec. 2003.
 [4] Harald Vogt, "Efficient Object Identification with Passive RFID Tags", in *Proc. Pervasive2002*, pp.98-113, 2002.

저 자 소 개



이 수 련(학생회원)
 2004년 2월 아주대학교
 전자공학과 학사 졸업
 2004년 3월~현재 아주대학교
 대학원 전자공학과
 석사 과정
 <주관심분야: 유비쿼터스 네트워크, Internet Qos, RFID 시스템>



주 성 돈(학생회원)
 2003년 2월 아주대학교
 전자공학과 졸업
 2003년 3월~현재 아주대학교
 대학원 전자공학과 석사과정
 <주관심분야: Traffic engineering, Wireless Communication>



이 채 우(정회원)
 1985년 서울대학교 제어계측
 학사 졸업.
 1988년 한국과학기술원
 전자공학과 석사 졸업.
 1995년 University of Iowa
 박사 졸업.
 1985년 1월~1985년 12월 (주)금성통신 연구원.
 1988년 9월~1999년 3월 한국통신 선임연구원.
 1999년 3월~2001년 9월 Lucent Technologies
 Korea 이사.
 2001년 9월~2002년 2월 한양대학교 겸임교수.
 2002년 3월~현재 아주대학교 전자공학과 조교수
 <주관심분야: 광대역 통신망, Ubiquitous networking, Traffic Engineering>