Journal of The Korea Society of Computer and Information
Vol. 18, No. 11, November 2013
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2013.18.11.087

RFID 시스템에서 효율적인 분리를 이용한 충돌 방지 알고리즘

김 성 수*, 윤 태 진*

An Anti Collision Algorithm Using Efficient Separation in RFID system

Sung-Soo Kim *, Tae-Jin Yun *

요 약

RFID 시스템에서, 리더의 인식영역에서 다수의 태그를 인식하는 과정에서 충돌이 발생하게 된다. 하나의 리더에게 두 개 또는 여러 태그가 응답하여 충돌이 발생하게 되며, 리더는 태그를 인식하지 못하는 일이 발생한다. 이런 충돌로 인해 리더는 인식영역 안에 있는 모든 태그를 인식하기 위한 시간이 길어지며, 이 경우 리더는 태그를 인식하지 못하는 경우가 발생된다. 리더는 인식영역 안의 모든 태그를 빠르게 인식할 수 있는 충돌 방지 알고리즘이 필요하다. 제안된 알고리즘은 태그의 응답을 효율적인 분리를 이용하여 태그 그룹으로 나누어 충돌을 회피한다. 또한, 제안된 알고리즘은 태그 아이디의 모든 비트를 알지 못해도 태그를 인식할 수 있다. 효율적인 분리를 통해 예측을 하여 리더로 부터의 응답 수를 줄인다.

▸ Keywords : RFID, 충돌 방지, 태그 충돌, 다중 태그

Abstract

In the RFID system, multiple tags respond in the process of identifying multiple tags in the reader's interrogation zone, resulting in collisions. Tag collision occurs when two or more tags respond to one reader, so that the reader cannot identify any tags. These collisions make it hard for the reader to identify all tags within the interrogation zone and delays the identifying time. In some cases, the reader cannot identify any tags. The reader needs the anti-collision algorithm which can quickly identify all the tags in the interrogation zone. The proposed algorithm efficiently divides tag groups through an efficient separation to respond, preventing collisions. Moreover, the proposed algorithm identifies tags without checking all the bits in the tags. The prediction with

•제1저자 : 김성수 •교신저자 : 윤태진
•투고일 : 2013. 9. 14, 심사일 : 2013. 10. 4, 게재확정일 : 2013. 10. 17.
* 경운대학교 모바일공학과(Dept. of Mobile Engineering, Kyungwoon University)

efficient separation reduces the number of the requests from the reader.

▸ Keywords : RFID, anti-collision, tag collision, multiple tags

I. 서 론

RFID 시스템을 이루는 IC칩 및 태그의 저가격화, 유통·물류 분야의 글로벌 서비스 확산, 무선인식 응용 유비쿼터스 시스템 기술개발 등 수많은 분야에서 무선인식 시장의 확산이 진행되고 있다[2][4]. 다양한 응용 분야에서 RFID 시스템의 성공적인 도입을 위해서는 고유한 기능인 태그의 식별에 충실하여야 한다. 특히 실시간으로 대량의 물품을 동시에 식별해야 하는 대규모 전자물류시스템에서 리더의 식별영역 내에 여러 개의 태그가 존재할 경우, 이를 충돌 없이 식별해야 하는 문제를 다중 태그 식별문제라 한다.

다중 태그 식별문제는 기존의 바코드, 교통카드, 무선결제 시스템과 같은 태그와 리더 간의 일대일 통신환경에서는 발생하지 않는다. 리더 식별영역 내에 한 개의 태그만 있을 경우 태그는 자신의 식별자에 적절한 부호화방식과 변조기법을 적용한 후 이를 리더로 전송한다. 리더에서 단일 태그 식별은 간단한 알고리즘을 통하여 구현할 수 있다. 이와 반대로 리더의 식별영역 내에 다수 개의 태그가 존재할 경우에는 여러 개의 태그가 동시에 리더의 질의에 응답하게 되므로 리더에서 충돌이 발생하게 된다. 이러한 충돌현상은 리더로 하여금 정확한 태그식별을 방해하는 원인이 된다. 특히, 대량의 물품을 실시간으로 식별해야하는 대규모 전자물류시스템과 같은 응용에 적용하기 위해서는 다중 태그 식별문제를 해결할 수 있는 충돌 방지 알고리즘이 필수적으로 요구된다.

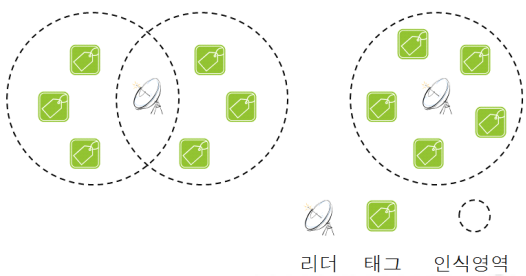


그림 1. 리더 충돌과 태그 충돌
Fig, 1. Reader Collision and Tag Collision

RFID 시스템은 여러 개의 리더와 태그들로 구성된다. 리더는 무선채널을 통해 각각의 태그들과 통신을 하며, 모든 태그들이 리더가 보낸 요청 신호를 동시에 듣게 되고 리더의 전송요청에 응답을 한다. RFID의 충돌은 대표적으로 리더 충돌과 태그 충돌로 나눌 수 있다[6]. 그림 1은 리더 충돌과 태그 충돌에 대한 그림이다. 다중 리더 환경에서 리더 간의 간섭을 리더 충돌이라 하고, 하나의 리더에 다중의 태그가 동시에 응답하여 발생한 간섭을 태그 충돌이라고 한다[3][5]. 충돌은 동시에 전송된 신호의 간섭으로 일어나게 되며 태그의 정확한 정보를 읽어 들일 수 없게 된다. 이는 정보의 손실로 이어지며 리더와 태그는 신뢰성 통신을 위하여 데이터를 재전송하게 된다. 신뢰성 통신을 위한 재전송은 충돌이 발생되지 않을 때까지 반복하게 된다. 충돌로 인해 재전송될 때 채널이 대역폭을 낭비하게 되며 인식에 따른 지연이 불가피하게 증가한다.

RFID 시스템에서 리더와 태그 간의 충돌을 방지하기 위한 충돌 방지 알고리즘은 중요한 문제이다. 일반적으로 태그의 수가 리더의 수보다 훨씬 많으므로 대부분의 RFID 시스템에서는 리더의 충돌에 대한 연구보다는 태그의 충돌에 대한 연구가 더 활발히 진행되고 있다[8][10][12][14-19]. 예를 들면 대형 할인매장 상품들에 있는 바코드를 RFID 태그로 대체하여 시스템을 구성하는 경우, 이때 태그들은 저비용, 초소형으로 설계되어 물품에 부착될 것이다. 대형 할인점을 찾은 소비자는 카트에 많은 물품들을 담아서 계산대로 올 것이다. 계산대에 있는 리더는 물품들에 부착되어 있는 태그들을 빠르게 인식해야 한다. 이런 RFID 시스템을 이용할 경우 소비자는 물품들이 담긴 카트를 계산대로 밀고 가면 계산대에 있는 리더가 물품들에 부착되어 있는 태그들을 동시에 인식하여 자동결제가 이루어져 편리하다. 이 경우 여러 물품에 부착되어 있는 태그들을 동시에 인식하는 속도는 중요한 요인이 된다[6].

무선 기술의 사용은 편리한 인식 과정 및 넓은 인식 거리 등 다양한 장점을 가지지만 이기종 무선통신과의 간섭, 다중 리더 환경에서의 리더 간 충돌, 그리고 다중 태그 환경에서의 태그 간 충돌 등 시스템 성능에 제약을 주는 여러 가지 문제를 발생시킨다. 특히 태그들 간에 충돌은 가장 보편적으로 발생하는 문제로 RFID 시스템의 성능과 안정성에 결정적인 영

향을 미친다. RFID 다중 태그 인식 기술은 이러한 충돌 현상을 최소화하여 다수의 태그를 빠르게 인식하는 기술이다. 다중 태그 인식 기술은 충돌 대처가 핵심으로 충돌 방지 기술로도 불린다.

본 논문에서는 하나의 리더가 다수의 태그를 식별하는 다중 태그 환경에서 효율적인 분리를 이용하여 태그를 예측하는 충돌 방지 알고리즘을 제안한다. 또한 분석과 실험을 통해 성능을 검증한다. 기존의 알고리즘은 태그들 간 충돌이 발생하지 않도록 중재하여 특정시점에 하나의 태그만이 응답하도록 하여 모든 태그를 식별하는 반면, 효율적인 분리를 이용한 충돌 방지 알고리즘은 식별 과정에서 해당 슬롯의 응답 정보와 충돌 비트의 개수를 이용하여 실제 존재하는 태그 아이디를 예측하여 태그 아이디를 식별한다. 태그 아이디의 모든 비트 값을 확인하지 않고도 태그를 식별 할 수 있다.

II. 관련 연구

2.1 맨체스터 코드를 사용한 충돌 추적

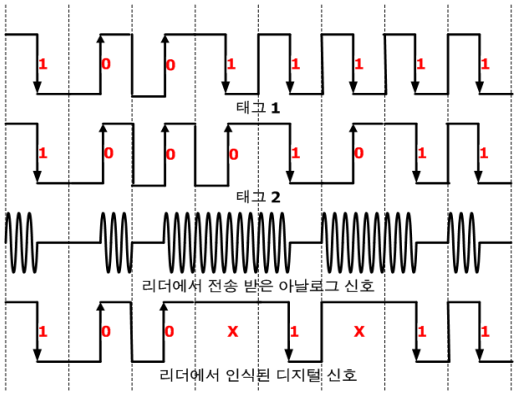


그림 2.맨체스터 부호화를 통한 개별 비트의 충돌 검출
Fig. 2. Collision behaviour for manchester code

RFID 시스템의 충돌 방지 알고리즘들은 태그 아이디 비트의 충돌, 충돌의 위치, 충돌의 개수를 추적하기 위해 맨체스터 디지털 코드 방식을 사용하고 있다. 맨체스터 코드는 1 비트 구간에서 반드시 위상의 변화를 발생시켜 '1' 인지 '0' 인지를 판별한다. 즉 위상이 변하지 않는 상태('0' 또는 '1'로 인식 불가능한 상태)를 충돌('X') 상태로 구분 할 수 있다 [6][12-13].

그림 2는 맨체스터 코드를 적용한 것으로, 비트 단위로 충돌을 추적할 수 있음을 보여준다. 태그 1은 "10011111"이고, 태그 2는 "10001011" 의 태그 아이디를 가진다고 가정한다. 태그 1과 태그 2가 리더로 자신의 아이디를 전송하면, 리더는 태그로부터 전송받은 각 태그 아이디를 디지털 신호로 변환한다. 이 변환된 신호에서 비트 4와 비트 6은 양의 변화와 음의 변화가 서로 중첩되어 위상이 변하지 않았다. 따라서 맨체스터 코드는 이를 허용되지 않으므로 오류로 판단하고 충돌로 인식한다.

2.2 쿼리 트리 알고리즘(Query Tree Algorithm)

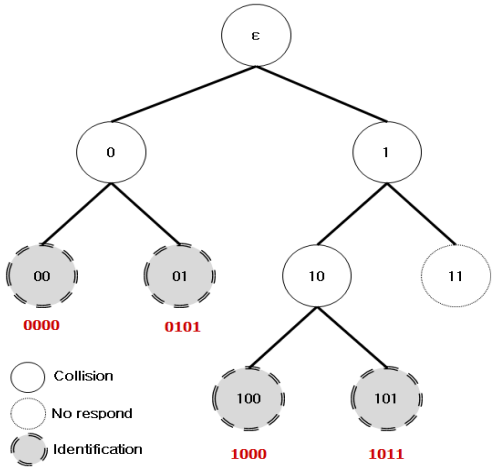


그림 3. 쿼리 트리 알고리즘에서 리더의 요청과 태그 응답 방법
Fig. 3. Reader's request and tag's response in query tree algorithm

쿼리 트리 알고리즘[1][8-9]은 2-ary 트리 기반의 충돌 방지 알고리즘으로 동작 방식이 간단하고, 구현이 쉽다. 쿼리 트리 알고리즘은 질의와 응답을 한 라운드로 구성된다. 각 라운드에서, 리더는 태그 아이디를 포함하는 특정 프리픽스 $q_1q_2q_3 \ldots q_k$ ($q_k \in \{0,1\}$, $1 \leq k \leq b$, b는 태그 아이디의 비트 수)를 태그에게 질의한다. 리더의 질의에 1개 이상의 태그가 응답할 경우, 최소한 그 프리픽스를 가지는 태그가 2개 이상인 것을 인지하고, 질의한 프리픽스에 '0' 과 '1'을 추가하여 태그에게 질의를 계속한다. 이 과정에서 프리픽스를 가지는 유일하게 1개의 태그가 리더로 응답하면 태그를 인식한다. 즉 1 개의 태그가 응답 할 때까지 프리픽스를 1 비트씩 추가하면서, 인식 영역 내의 모든 태그를 인식할 때까지 반복하게 된다.

그림 3은 쿼리 트리 알고리즘의 인식 과정을 나타낸다. 초기에 리더는 'ε' (empty string)을 태그들에게 질의하고, 태그는 특정 프리픽스에 상관없이 모든 태그들이 리더로 응답한

다. 이 때 리더의 인식 영역 내에 1개의 태그만 존재할 경우 더 이상의 질의 없이 태그를 인식한다. 아래의 경우는 4 개의 태그(0000, 0101, 1000, 1011)가 존재 하므로 리더는 '0'과 '1'을 큐(Queue)에 추가하고, 큐에서 '0'을 pop하여 태그에게 질의한다.

이런 과정을 반복한 후 "00", "01", "100", "101"의 프리픽스를 질의 했을 때 리더는 각각의 태그를 인식한다.

2.3 충돌 추적 트리 알고리즘(Collision Tracking Tree Algorithm)

충돌 추적 트리 알고리즘(CTTA)은 충돌 추적을 이용하는 것을 제외하고는 대표적인 쿼리 트리 알고리즘 기반의 알고리즘이다[7]. 기본동작은 쿼리 트리 알고리즘과 비슷하지만 가장 큰 차이점인 충돌 추적을 이용하여 충돌이 발생된 비트의 전 비트까지를 프리픽스로 생성을 하게 된다. 리더가 k-비트 길이의 프리픽스를 인자로 태그에 질의하는 것으로 시작한다. 영역 내의 태그들은 수신한 프리픽스를 자신의 식별자와 비교를 통해 매칭(matching)이 이루어질 경우 태그 식별자의 (k+1)번째 비트에서 마지막 비트까지를 리더로 전송한다. 이때, 리더는 수신된 정보에서 충돌이 발생한 위치를 검사하게 되며 충돌이 발생되는 지점인 n번째 비트를 수신하게 되면 태그의 전송을 중단하는 ACK 신호를 보내어 태그의 전송을 중단시킨다. 리더는 이때 새로운 2개의 프리픽스를 생성하게 된다. 기존에 보낸 'k-비트의 프리픽스' + '정상적으로 수신된 n-1 비트' + ('0' 또는 '1') 인 2개의 프리픽스를 생성한다. 이후 생성된 2개의 프리픽스를 재전송하여 충돌이 발생되지 않을 때까지 반복하게 된다.

2.4 효율적인 슬롯 할당을 통한 예측 알고리즘(ESA-IA)

슬롯 할당을 통한 예측이 가능한 ESA-IA(Efficient Slot Allocation-Inference Algorithm)에 대해 설명한다[13]. ESA-IA에서 태그는 리더의 요청 프리픽스와 일치하는 태그들은 해당 슬롯에 응답을 한다. 전체 태그 아이디의 '1'의 개수가 짝수인 태그 그룹은 슬롯 0에 응답하고, 전체 태그 아이디의 '1'의 개수가 홀수인 태그 그룹은 슬롯 1에 응답하게 하여 태그를 예측한다. 슬롯 0에 충돌된 비트가 2개인 경우는 충돌된 비트에 "00"과 "11"을 예측하며, 슬롯 1에 충돌된 비트가 2개인 경우는 충돌된 비트에 "01"과 "10"을 예측한다. 이때 충돌된 비트가 연속 충돌이든 비연속 충돌이든 예측이 가능하다. 즉, 전체적인 질의 횟수를 줄여 태그 인식을 위한 시간 단축을 한다. 충돌된 비트가 연속 충돌이든 비연속 충돌에 대해 예측을 위해 맨체스터 부호화를 사용해야 한다.

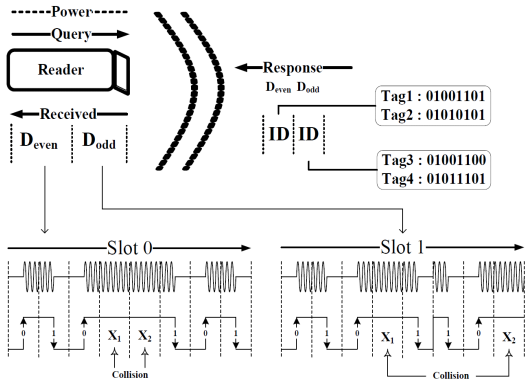


그림 4. ESA-IA에서 리더의 요청과 태그 응답 방법
Fig. 4. Reader's request and tag's response in ESA-IA

그림 4는 ESA-IA에서 리더의 요청과 태그 응답 방법을 나타낸다. 태그 1, 태그 2는 슬롯 0에 응답하며, $D_{even} = 0$ 인 경우이다. 이 경우는 태그 아이디 비트의 '1' 의 개수가 짝수 개인 경우이다. 남아있는 1의 개수의 값을 구하면 1 이 된다. 남아있는 1의 개수의 값이 1 인 경우는 2 개의 충돌된 비트에 '0' 과 '1' 인 태그(01001101) 와 2 개의 충돌된 비트에 '1' 과 '0' 인 태그(01010101)를 예측할 수 있다. 태그 3, 태그 4는 슬롯 1에 응답하며, D_{odd}가 1 인 경우이며, 이 경우는 태그 아이디 비트의 '1' 의 개수가 홀수 개인 경우이다. 남아있는 1의 개수의 값을 구하면 0이 된다. 태그 아이디 비트에 남아있는 '1' 의 개수가 0인 경우는 2 개의 충돌된 비트에 '0' 과 '0' 인 태그(01001100)와 2 개의 충돌된 비트에 '1' 과 '1' 인 태그(01011101)를 예측할 수 있다. 이처럼 향상된 슬롯 할당을 이용하여 2개의 태그를 예측할 수 있다.

III. 효율적인 분리를 통한 충돌 방지 알고리즘(Anti Collision Algorithm Using Efficient Separation)

ESACA(Anti Collision Algorithm Using Efficient Separation) 알고리즘은 ESA-IA의 문제점을 보완한다. ESA-IA의 문제점은 '1'의 개수가 항상 짝수 개의 경우는 해당 슬롯인 슬롯 0에만 응답을 하게 되어 한곳으로 치우치는 문제점이 있다. 가령 모든 태그가 태그 아이디의 '1'의 개수가 짝수일 경우 슬롯 1은 항상 빈 슬롯이 발생하는 문제를 가지

고 있다. 이 문제를 ESACA 알고리즘에서는 프리픽스 다음 비트의 '0' 또는 '1'을 보고 슬롯을 할당을 하여 향상된 분리를 통해 한쪽으로 치우치는 문제를 해결할 수 있다.

3.1 예측 알고리즘 검증을 위한 정리

제안한 알고리즘의 질의 노드 수를 분석하여 다중 태그를 인식할 때 걸리는 지연 시간을 분석한다. 응답에 대한 지연은 각 알고리즘의 방식이 상이하므로, 질의·응답 횟수는 질의노드 수 + 응답 수로 계산한다.

보조 정리 1 : 태그 아이디의 각 비트는 트리의 비 단말 노드에 대응되며, 완전한 태그 아이디는 단말 노드와 같다.

정리 1 : 태그 아이디 인식 과정은 트리의 루트(ε)에서 단말 노드까지의 경로를 탐색 하는 것이다.

증명 : 리더는 트리의 각 노드에 해당하는 프리픽스 비트 $P_1P_2P_3 \ldots P_{b-2}$($P_k \in \{0,1\}$, b는 태그 아이디의 비트길이)를 태그로 전송하고, 태그는 $R_1R_2R_3 \ldots R_b$($R_k \in \{0,1\}$, b는 태그 아이디의 비트길이)의 비트를 리더로 전송한다. 따라서 단말 노드에서 인식된 아이디는 $R_1R_2R_3 \ldots R_b$와 같다.

정리 2 : 트리의 노드를 프리픽스 노드(N)라 하고, 프리픽스 노드는 4개의 타임 슬롯을 이용하여 태그의 응답을 받으며, 각 타임 슬롯은 인식 상태(SS), 충돌 상태(SC), 무응답 상태(SI) 가진다.

그림 5는 제안된 알고리즘에 대해 프리픽스 노드 생성을 보여주며, 두 비트씩 증가하면서 프리픽스가 생성된다.

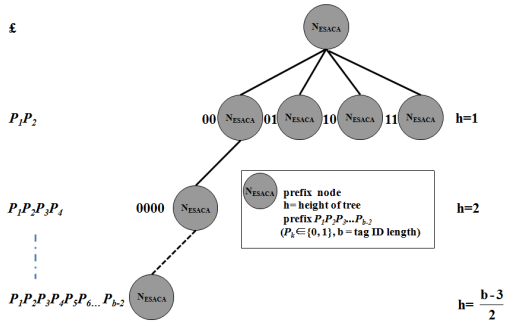


그림 5. 프리픽스 노드 생성
Fig. 5. generation of prefix node

증명 : 리더는 태그들로부터 프리픽스가 동일하며, 프리픽스 다음 한 비트와 태그 아이디의 '1'의 개수가 같은 태그는 각 타임 슬롯에 동시에 응답받고, 각 슬롯은 충돌이 없거나 2개 이상의 충돌이 발생한다. 타임 슬롯에 하나의 태그만 응답하거나 비트 충돌이 2개 이하로 발생한 경우는 태그 인식 상태(SS) 가 되고, 더 이상의 질의는 생성하지 않는다. 비트 충돌이 3개 이상 발생한 경우는 충돌 상태(SC) 가 되고, 새로운 질의를 생성한다. 마지막으로 타임 슬롯의 조건에 맞는 아이디가 존재하지 않는 경우는 무응답 상태(SI) 가 된다.

정의 1 : 프리픽스 노드의 비트열은 충돌 비트가 3개 이상인 경우 항상 1 비트씩 증가한다. 프리픽스 노드의 비트열이 충돌인 경우 항상 2 비트씩 증가한다.

정리 3 : 태그 아이디 인식 과정에서 질의 횟수(프리픽스 노드는 충돌수와 같다.

$$N_{ESACA} = C_{ESACA} \tag{1}$$

증명 : 충돌 발생 여부를 검출 방식을 사용하므로, 각 타임 슬롯의 상태(SC , SI , SS)에 따라 비트 충돌 패턴을 분석하여 존재할 수 없는 프리픽스 노드(N_{ESACA})는 생성하지 않는다. 존재하지 않는 프리픽스 노드를 생성하지 않기 때문에 충돌을 줄일 수 있다. 따라서 새로운 프리픽스 노드를 생성하는 경우는 충돌이 3개 이상 발생한 경우이며, 전송 받은 비트열 ($R_1R_20XX\cdots XX\cdots R_b$)을 분석하여, R_1R_2000, R_1R_2001, R_1R_2010, R_1R_2011를 생성하고, 프리픽스의 비트 길이가 b-3까지 증가하면 모든 태그 인식 과정은 종료된다.

ESACA는 트리로 이루어지며 트리의 특성상 충돌이 발생하게 되면 4개의 서브트리로 분리하게 된다. 이 서브 트리는 프리픽스 노드에 해당된다. 충돌이 발생하게 되면 두 개의 프리픽스 노드 노드가 된다. 생성된 프리픽스 노드에는 항상 태그가 존재한다. 프리픽스 노드가 충돌이 된다. 따라서 태그 아이디 인식 과정에서 총 질의 횟수는 수식(1)과 같다.

3.2 비트 충돌 패턴에 따른 태그 아이디 인식

태그 아이디를 인식하기 위해서 태그들의 응답 신호를 분석한다. 리더는 응답 신호에 대하여 다음의 인식 정보를 유지한다.

N_{id} = 아이디 비트의 '1' 의 개수
$D_{(even\ or\ odd)}$ = 짝수 또는 홀수 정보
N_1 = 아이디 비트에서 확인된 '1' 의 수
N_c = 아이디 비트에서 충돌된 비트의 개수
N_{1r} = Nc에 포함되어야 하는 '1' 의 개수
리더는 위의 5가지 인식 정보를 가지고 태그를 예측한다.

Nc가 2인 경우는 식(2)의 $D_{(even\ or\ odd)}$ 를 이용하여 충돌된 두 개의 태그를 인식한다. 태그를 인식 할 수 있는 경우는 다음과 같다.

Nc = 0 : 충돌(X)이 전체 태그 아이디 중에서 0개인 경우 충돌이 없는 경우로써, 리더의 요청에 하나의 태그가 응답한 경우이므로 리더는 하나의 태그를 인식할 수 있다.

Nc = 1 : 충돌(X)이 전체 태그 아이디 중에서 1개 인 경우 태그로부터 전송받은 태그 아이디에서 1 비트 충돌(X)만 난 경우이다. 그러나 제안 알고리즘에서는 나머지 태그 아이디에서 '1'의 개수가 짝수인 경우와 홀수인 경우를 나눠서 전송받기 때문에 이런 경우는 발생하지 않는다.

Nc = 2 : 충돌의 개수가 전체 태그 아이디 중에서 2개인 경우 태그로부터 전송받은 태그 아이디에서 2 비트 충돌 (X_1X_2 또는 $X_1 \ldots X_2$)만 발생한 경우이며, 이 패턴은 태그 아이디에서 '1'의 개수를 짝수인 경우와 홀수인 경우로 분리하여, 각각 따로 전송 받기 때문에 2 비트 충돌에 대해서 짝수인 경우는 "00" 과 "11" 이 되고, 홀수인 경우는 "01" 과 "10" 이 된다. 따라서 2 비트 충돌에 대해서 두 개의 태그 아이디를 인식할 수 있다.

$$(N_1 + N_{1r}) \% 2 = D_{(even\ or\ odd)} \quad (N_{1r} \leq 2) \qquad (2)$$

Nc = 3 : 충돌의 개수가 전체 태그 아이디 중에서 3개 이상인 경우 태그로부터 전송받은 태그 아이디에서 3 비트 충돌 ($\ldots X_1X_2X_3 \ldots$ 또는 $X_1 \ldots X_2 \ldots X_3$ 또는 $X_1X_2 \ldots X_3 \ldots$등) 이상인 경우이며, 이 패턴은 한 번에 태그 아이디를 인식할 수 없으므로, 새로운 프리픽스를 생성한다. 이후 큐에 저장한 후 재 질의한다.

3.3 효율적인 분리

F=0, D_{even} : 리더로 전송할 태그 아이디의 첫 비트([프리픽스 +1]번째 비트, F)가 0이고, 전체 태그 아이디의 '1'의 개수가 짝수 일 때, 슬롯 0에 전송한다.

F=0, D_{odd} : 리더로 전송할 태그 아이디의 첫 비트([프리픽스 +1]번째 비트, F)가 0이고, 전체 태그 아이디의 '1'의 개수가 홀수 일 때, 슬롯 1에 전송한다.

F=1, D_{even} : 리더로 전송할 태그 아이디의 첫 비트([프리픽스 +1]번째 비트, F)가 1이고, 전체 태그 아이디의 '1'의 개수가 짝수 일 때, 슬롯 2에 전송한다.

F=1, D_{odd} : 리더로 전송할 태그 아이디의 첫 비트([프리픽스 +1]번째 비트, F)가 1이고, 전체 태그 아이디의 '1'의 개수가 홀수 일 때, 슬롯 3에 전송한다.

3.4 제안 알고리즘 동작 예

그림 6은 제안된 알고리즘에서 리더와 태그 사이의 응답 방법에 대해 나타낸다. $D_{(even\ or\ odd)}$을 이용하여 태그 아이디 비트의 충돌된 비트의 값을 구하는 방법을 나타낸다. 태그 1, 태그 2는 슬롯 0에 응답하며, $F=0, D_{even}=0$인 경우이다. 이 경우는 태그 아이디 비트의 '1' 의 개수가 짝수 개인 경우이다.

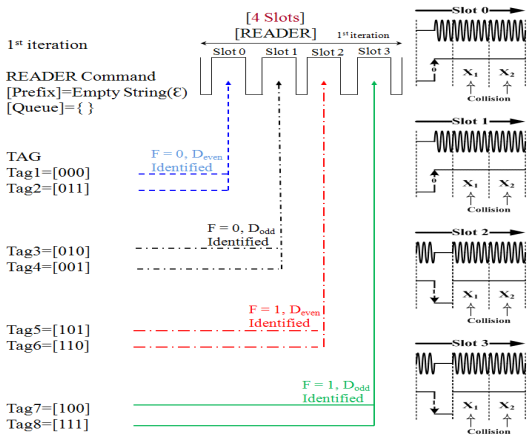


그림 6. 제안된 알고리즘에서 리더의 요청과 태그 응답 방법
Fig. 6. Reader's request and tag's response in proposed algorithm

$D_{even}, N_1 = 0, N_c = 2$인 경우로써, 식(2)을 이용하여 N1r의 값을 구하면 0 이 된다. N1r 이 0 인 경우는 2 개의 충돌된 비트에 '0' 과 '0' 인 태그(000) 와 2 개의 충돌된 비트에 '1' 과 '1' 인 태그(011)를 예측할 수 있다. 태그 7, 태그 8은 슬롯 3에 응답하며, D_{odd}가 1 인 경우이며, 이 경우는 태그 아이디 비트의 '1' 의 개수가 홀수 개인 경우이다. $F=0, D_{odd}, N_1 = 1, N_c = 2$인 경우로써, 식(2)을 이용하여 N1r의 값을 구하면 0이 된다. 태그 아이디 비트에 남아있는 '1' 의 개수가 0인 경우는 2 개의 충돌된 비트에 '0' 과 '0' 인 태그(100)와 2 개의 충돌된 비트에 '1' 과 '1' 인 태그(111)을 예측할 수 있다. 이처럼 향상된 슬롯 할당을 이용하여 2 개의 태그를 예측할 수 있다.

3.5 질의 횟수 분석

ESACA는 4-ary 구조의 충돌 검출 방식이다. 알고리즘의 충돌 검출 방식은 4개의 타임 슬롯을 사용하며, 각 타임 슬롯

에서 3개 이상의 충돌 발생 시 프리픽스 노드가 생성된다. 따라서 $N_{ESACA}(n)$의 최악의 경우는 다음과 같이 나타낼 수 있다.

$$N_{ESACA}(n) \le \frac{4}{3}(n-1) \quad (3)$$

증명: 태그의 아이디는 유일한 비트 길이(b)를 가지며, 그림 4에서 트리의 높이 $h(0 < h \le \lfloor \frac{b-1}{2} \rfloor)$에서, N_{ESACA}의 비트 길이는 3.1의 정의 1에 의해 $2h$이다. 각 높이에서 최악의 경우는 최대 태그 개수가 존재하는 경우이며, 이 때 N_{ESACA}은 $4^{\log_4 n-1}$ 개이다. 그리고 최대 태그 수보다 작은 경우에 최악의 경우는 $h = \lceil \log_4 n-1 \rceil$ 개의 N_{ESACA}을 생성 후 2비트씩 증가하면서 N_{ESACA}의 비트열이 b-2까지 생성한다. 충돌 수(C_{ESACA})는 트리 높이가 $\lfloor \frac{b-3}{2} \rfloor$ 까지 발생한다. 이 경우 모든 태그를 인식하기 위한 N_{ESACA}는 트리의 최대 높이가 $\lfloor \frac{b-1}{2} \rfloor$ 까지 $\frac{n}{2}$개씩 생성해야 한다. 따라서 태그의 개수(n)가 $4^2 \le n < 4^{b-3}$ 이며, n개의 태그를 위한 높이가 h에서 최대 충돌 수 $C_{ESACA}(n,h)$은 다음과 같다.

$$C_{ESACA}(n,h) = \begin{cases} 4^h & \text{if } 0 < h \le \lfloor \log_4 n-1 \rfloor \\ \frac{n}{2} & \text{if } \lfloor \log_4 n-1 \rfloor < h \le \lfloor \log_4 n+1 \rfloor \end{cases} \quad (4)$$

식 (4)를 이용하여 최대 $C_{ESACA}(n)$을 계산하면 아래와 같다.

$$C_{ESACA}(n) \le \sum_{h=1}^{\log_4 n+1} C_{ESACA}(n,h)$$

$$= \sum_{h=1}^{\lfloor \log_4 n-1 \rfloor} 4^h + \sum_{h=\lfloor \log_4 n \rfloor}^{\lfloor \log_4 n+1 \rfloor} \frac{n}{2}$$

$$\le \frac{4}{3}(n-1) \quad (5)$$

IV. 성능평가

4.1 실험 환경과 성능평가 방법

본 논문에서 성능평가를 위한 태그는 EPCTM Tag Data Standards Version 1.4를 따르는 태그에서 시리얼 넘버에 해당되는 부분만을 사용하였다[11]. 표 1은 General Identifier(GID-96)의 헤더 필드를 보여준다. Header, General Manager Number, Object Class의 경우는 유사성이 많은 비트들로 구성되어 있기 때문에 본 실험에서는 실제 각 상품에 대한 고유 식별번호를 나타내는 시리얼 넘버(Serial Number)만을 생성하여 실험을 하였다.

표 1. General Identifier(GID-96)의 헤더 필드
Table. 1. The general identifier (GID-96) includes three fields in addition to the header

	Header	General Manager Number	Object Class	Serial Number
GID-96	8	28	24	36
	00110101 이진 값	268,435,455 최대10진 값	16,777,215 최대10진 값	68,719,476,735 최대10진 값

본 논문에서 실험하기 위한 프로그램은 C# 으로 작성하였다. 실험에서 사용한 태그 아이디 할당 방법은 랜덤 할당 방법과 순차 할당 방법을 사용하였다. 랜덤 할당 방법은 태그의 아이디를 랜덤 값을 생성하여 할당하는 방법이다. 랜덤 할당 방법을 사용하면 태그 아이디가 가질 수 있는 전체 범위에 아이디가 고르게 분포된다. 이는 태그들의 아이디 간의 비트 값의 유사성이 적다는 것을 의미한다. 랜덤 할당 방법에서 태그 아이디가 가지는 값의 범위는 균일분포(uniform distribution)를 갖는 난수발생 함수를 사용하였다. 순차 할당 방법은 1씩 시퀀스하게 증가하면서 태그 아이디를 생성하였다.

본 논문에서는 태그들이 가지는 패턴의 유사성이 없는 상태를 표현하기 위해 시간변수를 종자(seed)값으로 하여 아이디 비트 크기의 균일분포를 가지는 랜덤함수를 사용한다. 순차 할당 방법은 태그의 아이디를 특정 범위에 집중시키며, 인식 영역에 아이디 값이 유사한 태그로 구성되는 환경을 만들 수 있다. 비트 패턴은 36비트일 경우, 최대 236-1의 값을 가질 수 있지만 실험에서는 태그의 개수를 10^1~10^3개까지, 10^3~10^5개까지의 태그를 리더의 인식영역에 동시에 있다고 간주하고 5번의 실험을 수행하여 평균값을 각각 구하여 실험

을 진행하였다. 순차 할당 방법과 랜덤 할당 방법으로 태그 아이디를 생성하여 태그 개수를 변화시키면서 리더와 태그 간의 질의 수를 측정하였다. 분석을 이용한 비교분석은 태그 개수를 변화시키면서 매트랩(MATLAB)을 이용하여 리더와 태그 간의 질의 수를 비교분석을 하였다.

4.2 실험 결과 및 고찰

리더와 태그 간의 질의 횟수를 비교 분석하였다. 제안된 알고리즘의 경우는 연속충돌 또는 비연속 충돌에 대해 태그를 예측하기 때문에 질의 횟수가 많이 줄어든다. 태그를 순차 할당의 경우에는 트리의 한쪽 방향으로 트리가 생성되기 때문에 충돌을 추적하는 알고리즘보다 좋은 성능을 보여준다.

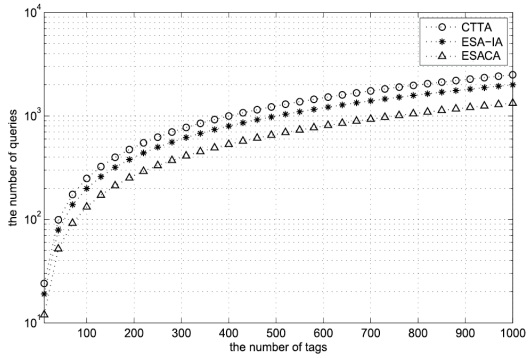
질의 횟수는 리더와 태그 간의 질의 횟수(요청-응답)이다. 리더가 태그를 빠르게 인식하기 위해서는 질의 횟수를 줄이는 것이다. 제안된 ESACA 알고리즘은 효율적인 분리를 통해 태그를 빠르게 인식한다. 이는 전체 질의 횟수를 줄이며, 전체 시간을 줄이는 결과를 가져온다. 실험에서는 태그 아이디를 순차적인 할당 방법과 랜덤 할당 방법을 사용하여 생성하고 태그 수의 개수를 변경하면서 실험을 하였다.

그림 7은 태그의 개수를 $10^1 \sim 10^3$개까지, $10^3 \sim 10^5$개까지 증가하면서 CTTA, ESA-IA 그리고 ESACA 알고리즘을 리더와 태그 간의 질의 횟수를 비교하였다. 태그의 개수를 변화시키면서 각 알고리즘에 대해 최악의 경우(worst case)에 대해 매트랩을 이용하여 그래프로 작성하였다. CTTA, ESA-IA 그리고 ESACA 알고리즘에 대해 비교하였다. 3개의 알고리즘은 충돌이 발생한 비트의 위치와 태그의 충돌 개수를 알기 위하여 맨체스터 부호화를 사용한다.

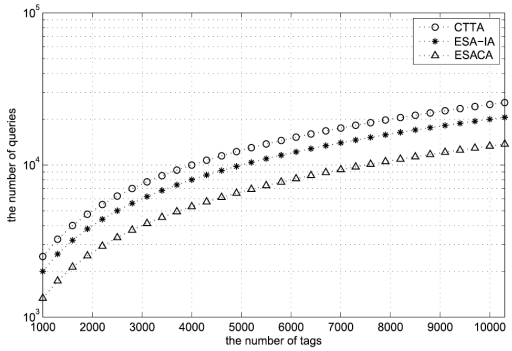
표 2. 분석에 따른 평균 질의 횟수
Table. 2. The average of request-response for one tag identification in accordance with analysis

알고리즘	질의 횟수의 비례상수($a=y/x$)					횟수
	10	500	1000	5000	10000	비교
CTTA	2.40	2.50	2.50	2.50	2.50	1.00
ESA-IA	1.90	2.00	2.00	2.00	2.00	0.80
ESACA	1.20	1.33	1.33	1.33	1.33	0.53

표 2에서, x축을 태그의 개수, y축을 질의 횟수라고 할 때, 비례상수 $a=\Delta y/\Delta x$를 조사하면, 분석에 따른 평균 질의 횟수는 태그의 개수가 증가하여도 3개의 알고리즘은 태그의 개수에 상관없이 일정하게 증가되는 것을 볼 수 있다. CTTA에 비해 ESA-IA는 약 0.2, ESACA는 약 0.5 정도의 평균 질의 횟수를 줄였다.



(a) 태그의 수 - $10^1 \sim 10^3$
(a) The number of tags - $10^1 \sim 10^3$



(b) 태그의 수 - $10^3 \sim 10^5$
(b) The number of tags - $10^3 \sim 10^5$

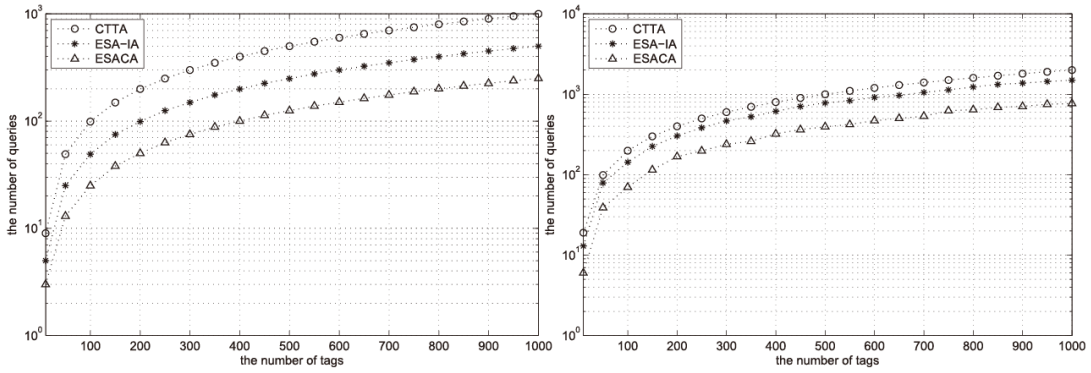
그림 7. 분석에 따른 리더와 태그 간의 질의 횟수
Fig. 7. the number of query between reader and tag in accordance with analysis

그림 8은 실험을 통해 순차 할당 방법과 랜덤 할당 방법으로 36비트의 태그 아이디를 가지고 태그의 개수를 증가시키면서 리더와 태그 간의 질의 횟수를 비교하였다. 태그의 개수와 순차랜덤 할당 방법에 상관없이 예측 알고리즘에서 알 수 있는 연속충돌 또는 비연속 충돌된 비트의 수가 2 개인 경우의 수가 증가하여 태그 인식의 증가로 인해 우수한 성능을 가진다.

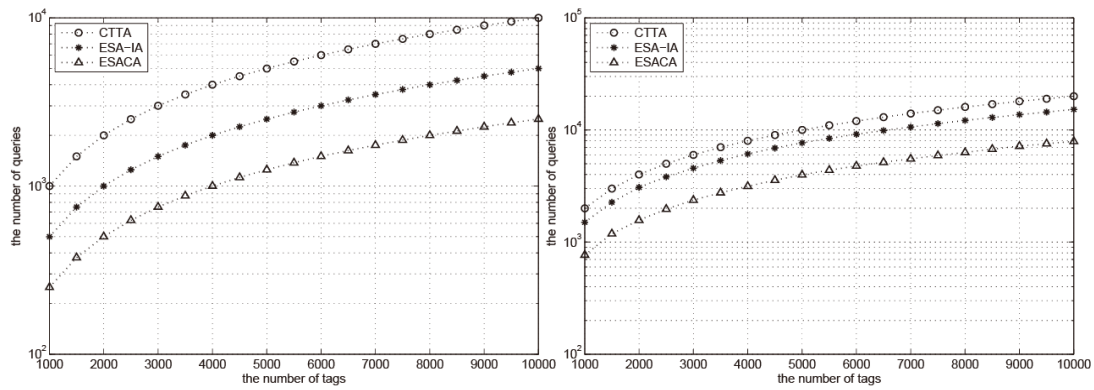
표 3. 실험(순차·랜덤 할당 방법)에 따른 평균 질의 횟수
Table. 3. The average of request-response for one tag identification in accordance with experiment (random and sequential assignment)

할당 방법	알고리즘	질의 횟수의 비례상수($a=y/x$)					횟수
		10	500	1000	5000	10000	비교
순차 할당	CTTA	0.90	1.00	1.00	1.00	1.00	1.00
	ESA-IA	0.50	0.50	0.50	0.50	0.50	0.50
	ESACA	0.30	0.25	0.25	0.25	0.25	0.26
랜덤 할당	CTTA	1.90	2.00	2.00	2.00	2.00	2.02
	ESA-IA	1.30	1.56	1.50	1.53	1.52	1.51
	ESACA	0.60	0.79	0.77	0.80	0.79	0.76

표 3은 실험에 따른 평균 질의 횟수를 비교한 것이다. 평



(a) 태그의 수 - 10^1~10^3 (순차 할당) (b) 태그의 수 - 10^1~10^3 (랜덤할당)
(a) The number of tags - 10^1~10^3 (sequential assignment) (b) The number of tags - 10^1~10^3 (random assignment)



(c) 태그의 수 - 10^3~10^5 (순차 할당) (d) 태그의 수 - 10^3~10^5 (랜덤 할당)
(c) The number of tags - 10^3~10^5 (sequential assignment) (d) The number of tags - 10^3~10^5 (random assignment)

그림 8. 실험에 따른 리더와 태그 간의 질의 횟수
Fig. 8. The number of query between reader and tag in accordance with experiment

균 질의 횟수는 CTTA에 비해 순차 할당의 경우 ESA-IA는 약 0.5, ESACA는 약 0.26 정도의 평균 질의 횟수를 줄이며, 랜덤 할당의 경우는 ESA-IA의 경우는 약 0.76, ESACA의 경우는 0.52정도 평균 질의 횟수를 줄이는 것을 볼 수 있다.

IV. 결 론

RFID 시스템에서 리더의 식별영역 내에 다수 개의 태그가 존재할 경우에는 여러 개의 태그가 동시에 리더의 질의에 응답하게 되므로 리더에서 충돌이 발생하게 된다. 이러한 충돌현상은 리더로 하여금 정확한 태그식별을 방해하는 원인이 된다. 리더는 인식영역 내의 모든 태그들을 빠르게 인식하는 충돌 방지 알고리즘이 필요하다. 제안한 알고리즘은 효율적인 분리를 통해 슬롯을 할당하여 리더가 태그의 아이디를 예측하여 모든 태그를 인식하는 알고리즘이다. 기존 알고리즘인 ESA-IA의 문제점은 프리픽스가 증가하여도 항상 짝수 개의 경우는 해당 슬롯인 슬롯 0에만 응답을 하게 되어 좀 더 향상된 분리를 하지 못하는 문제점이 있다. 가령 모든 태그가 태그 아이디의 '1'의 개수가 짝수일 경우 슬롯 1은 항상 빈 슬롯이 발생하는 문제를 가지고 있다. 이 문제는 ESACA을 이용하여 프리픽스 다음 비트의 '0' 또는 '1'을 보고 슬롯을 할당을 하여 향상된 분리를 통해 한쪽으로 치우치는 문제를 해결할 수 있다. 실험결과에서 제안된 알고리즘은 ESA-IA의 경우 약 0.76의 평균 질의 횟수를 0.52까지 줄이 줄인다. 질의 횟수를 감소는 전체 태그를 인식하는 시간을 줄이는 효과를 가져 온다. 이는 효율적인 분리를 통해 태그를 예측하여 전체 질의 수를 줄임으로써 성능에 미치는 영향이 크기 때문이다.

각 상품에 대한 고유 식별번호인 시리얼 넘버를 순차적인 방법으로 할당한 제품들의 경우에 제안한 알고리즘을 사용하

면 빠르게 태그 인식이 가능하다. 향후 연구과제로는 제안한 알고리즘을 하드웨어에 적용하여 실제 환경에서의 성능을 검증해야 하는 연구가 필요하다.

참고문헌

〔1〕 C. Law, K. Lee, and K. Y. Siu, "Efficient Memoryless protocol for Tag Identification," in Proc. 4th international workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM'00), ACM, pp. 75-84, 2000.

〔2〕 R. Want, D. M. Russell, "Ubiquitous Electronic Tagging," IEEE Distributed Systems Online, vol. 1, no. 1, Sep. 2000.

〔3〕 S. Sarma, D. Brock, and D. Engels, "Radio Frequency Identification and the Electronic Product Code," IEEE Micro, vol. 21, no. 6, pp. 50-54, Nov. 2001.

〔4〕 K. Romer, T. Schoch, "Infrastructure Concepts for Tag-Based Ubiquitous Computing Applications," Workshop on Concepts and Models for Ubiquitous Computing at Ubicomp 2002, Goteborg, Sweden, Sep. 2002.

〔5〕 D. W. Engels, S. E. Sarma, "The Reader Collision Problem," in Proc. IEEE International Conference on System, Man and Cybernetics, Hammamet, Tunisie, Oct. 2002.

〔6〕 K. Finkenzeller, RFID Hand Book : Fundamentals and Applications in Contactless Smart Card and Identification, 2nd ed. Chicester, Sussex, U.K.: Wiley, 2003.

〔7〕 F. Zhou, D. Jin, C. Huang, and M. Hao, ""Optimize The Power Consumption of Passive Electronic Tags for Anti-collision Schemes," in Proc. The 5th International Conference on ASIC, vol. 2, pp. 1213- 1217, Oct. 2003.

〔8〕 J. Myung, and W. Lee, "Adaptive Binary Splitting: A RFID Tag Collision Arbitration Protocol for Tag Identification," ACM/Springer Mobile networks and Applications, vol. 11, no. 5, pp. 711-722, Oct. 2006.

〔9〕 D. Shih, P.L. Sun, D.C.Yen and S. M.Huang, "Taxonomy and Survey of RFID Anti-collision protocols," Computer and Communicatios, vol. 29, pp. 2150-2166, 2006.

〔10〕 S. Kim and P. Park, "An efficient tree-based Tag Anti-collision protocol for RFID systems," IEEE Communications Letters, vol. 11, pp. 449-451, May 2007.

〔11〕 EPCglobal, EPCglobal Tag Data Standards Version 1.4, 〔Online〕. Available: http://www.epcglobalinc.org/standar-ds/tds/tds_1_4-standard-20080611.pdf

〔12〕 D.H Baek, S.S Kim, Y.H. Kim, K.S. Ahn, "A Fast Tag Prediction Algorithm using Extra Bit in RFID System," Journal of The Korea Society of Computer and Information, Vol. 13, No. 5, pp. 255-261, 2008.

〔13〕 S.S Kim, Y.H. Kim, K.S. Ahn, "An Inference Algorithm with Efficient Slot Allocation for RFID Tag Identification," IEICE Trans. on Communications, Vol. E93-B, No.1, pp.170-173, 2010.

〔14〕 P.S. Jeong, W.S. Jung, C.Y. Yun, Y.H. Oh, "A Study on the Performance Improvement of Anti-Collision Algorithm for RFID," Journal of Korea Information and Communications Society, Vol. 34, No. 6, pp.149-155, 2009.

〔15〕 X. Jia, Q. Feng, C Ma, "An Efficient Anti-Collision Protocol for RFID Tag Identification," IEEE Communications Letters, Vol. 14, No. 11, pp.1014-1016, 2010.

〔16〕 S.W. Choi, J.H. Choi, J Yoo, "An Efficient Anti-Collision Protocol for Coping with the Capture Effect in RFID Tag Identification," Journal of Korean Institute of Information Scientists and Engineers, Vol. 38, No. 6, pp.476-482, 2011.

〔17〕 Y.H Cho, J.G Kook, "Energy Effective Tag Anti-collision Protocol for Mobile RFID System," Journal of the Korea Society of Computer and Information, Vol. 17, No. 2, pp. 207-214, 2012.

[18] J. Shin, B. Jeon and D. Yang, "Multiple RFID Tags Identification with M-ary Query Tree Scheme," IEEE Communications Letters, Vol. 17, No. 3, pp.604-607, 2013.
[19] D. Yang, J. Shin, "EMQT : A study on Enhanced M-ary Query Tree Algorithm for Sequential Tag IDs," Journal of Korea Information and Communications Society, Vol. 38B, No. 6, pp.435-445, 2013.

저 자 소 개

김 성 수

2002: 금오공과대학교
컴퓨터공학과 공학사
2005: 경북대학교
컴퓨터공학과 공학석사
2012: 경북대학교
컴퓨터공학과 공학박사
현 재: 경운대학교
모바일공학 조교수
관심분야: 임베디드 시스템, 정보보호, RFID , 센서 네트워크
Email : ninny@ikw.ac.kr

윤 태 진

1994: 경북대학교
컴퓨터공학과 공학사
1996: 경북대학교
컴퓨터공학과 공학석사
2012: 경북대학교
컴퓨터공학과 공학박사
현 재: 경운대학교
모바일공학 교수
관심분야: 임베디드 시스템, 정보보안, 센서 네트워크
Email : tjyun@ikw.ac.kr