

RFID 시스템에서 4 슬롯을 이용한 충돌방지 알고리즘

김용환*, 김성수**, 류명춘*, 박준호*, 정경호*

An Anti-Collision Algorithm with 4-Slot in RFID Systems

Yong-Hwan Kim*, Sung-Soo Kim**, Myung-Chun Ryoo*, Joon-Ho Park*, Kyung-Ho Chung*

요약

본 논문에서는 트리 기반의 타임 슬롯을 사용한 하이브리드 쿼리 트리 구조를 제안한다. 제안 알고리즘은 8-ary tree 구조이며, 리더의 질의에 대한 태그 ID 응답 시, 디지털 코딩 방식을 Manchester 코드를 적용하여, 충돌 위치와 충돌 비트수를 검출할 수 있다. 또한 이 알고리즘은 고정된 4개의 슬롯을 사용하여, 한 번의 질의에 다수의 태그를 인식할 수 할 수 있다. 제안 알고리즘은 리더가 응답 받는 태그 ID의 첫 번째 비트 ($[\text{prefix}+1]^{\text{th}}$, $F \in \{0 \text{ or } 1\}$)와 태그 ID의 두 번째 ~ 세 번째 비트($[\text{prefix}+2]^{\text{th}} \sim [\text{prefix}+3]^{\text{th}}$, $B_2 \in \{00 \text{ or } 11\}$, $B_1 \in \{01 \text{ or } 10\}$)에서 비트 충돌 패턴을 이용해 4개의 타임 슬롯에 응답하게 함으로써, 한 번에 8개의 태그를 인식할 수 있다. 본 논문은 알고리즘의 질의 노드(prefix) 수의 worst case를 분석하여 다중 태그를 인식하는데 걸리는 지연 시간을 추출한다. 제안 알고리즘의 인식 지연 시간은 질의-응답 수와 질의 비트 수를 기준으로 하며, 각 알고리즘의 수식을 바탕으로 산출하였다.

▶ Keywords : RFID, 하이브리드 쿼리 트리 알고리즘, 타임 슬롯

Abstract

In this paper, we propose tree-based hybrid query tree architecture utilizing time slot. 4-Bit Pattern Slot Allocation(4-SL) has a 8-ary tree structure and when tag ID responses according to query of the reader, it applies a digital coding method, the Manchester code, in order to extract the location and the number of collided bits. Also, this algorithm can recognize multiple Tags by single query using 4 fixed time slots. The architecture allows the reader to identify 8 tags at the same time by responding 4 time slots utilizing the first bit($[\text{prefix}+1]^{\text{th}}$, $F \in \{0 \text{ or } 1\}$) and bit pattern from second ~ third bits($[\text{prefix}+2]^{\text{th}} \sim [\text{prefix}+3]^{\text{th}}$, $B_2 \in \{00 \text{ or } 11\}$, $B_1 \in \{01 \text{ or } 10\}$) in tag ID. we analyze worst case of the number of query nodes(prefix) in algorithm to extract delay time for recognizing multiple tags. The identification

•제1저자 : 김용환 •교신저자 : 정경호

•투고일 : 2014. 11. 15, 심사일 : 2014. 11. 25, 게재확정일 : 2014. 12. 3.

* 경운대학교 컴퓨터공학과(Dept. of Computer Engineering, Kyungwoon University)

** 경운대학교 모바일공학과(Dept. of Mobile Engineering, Kyungwoon University)

delay time of the proposed algorithm was based on the number of query-responses and query bits, and was calculated by each algorithm.

▶ Keywords : RFID, hybrid query tree algorithm, time slots

1. 서론

RFID(Radio Frequency Identification)는 일상에 존재하는 여러 물품이나 생명체들에 부착되어 무선으로 인식할 수 있도록 하는 기술이다. RFID는 저주파수 대역의 교통카드, 도서관리, 생산라인 관리 및 고주파수 대역의 유통, 물류 시스템 등에 활용되고 있다[1]. RFID는 그림1과 같이, 무선 트랜시버인 리더와 물체에 부착된 태그로 구성되며, 태그들은 각각의 고유한 ID를 가진다[2].

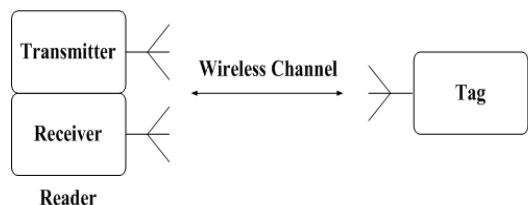


그림 1. RFID 시스템 블록도
Fig. 1. RFID system block diagram

RFID의 리더는 비 접촉 방식으로 태그의 ID를 인식하기 위해 무선 통신 방식을 채택하고 있다. 따라서 무선 통신 시 여러 가지 문제점이 발생할 수 있다. 그 중 한 가지는 리더와 태그간의 무선채널을 공유하기 때문에 발생하는 데이터 충돌이다.

RFID의 데이터 충돌은 대표적으로 리더 충돌과 태그 충돌로 나눌 수 있다[3], [4], [5]. 그림 2와 같이, 다중 리더 간의 간섭을 리더 충돌이라 하고, 하나의 리더에 다중의 태그가 동시에 응답하여 발생한 간섭을 태그 충돌이라 한다.

트리기반의 주요 알고리즘은 쿼리 트리 알고리즘, 4-ary 쿼리 트리 알고리즘이 있으며[1], [6], [7], [8], [9], [10]. 슬롯 알고리즘 기반의 알고리즘은 프레임 슬롯알로하, 동적 프레임 슬롯 알로하 등이 있다 [1], [6], [11], [12], [13]. 하이브리드 쿼리 트리 기반의 알고리즘은 하이브리드 쿼리 트

리 프로토콜, IS-ESBT, 4-BPSA 등이 있다[6], [14], [15], [16], [20].

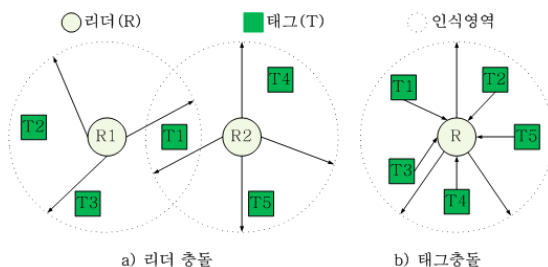


그림 2. 리더 충돌과 태그충돌
Fig. 2. Reader collision and Tag collision

본 논문의 연구는 다음과 같다. 태그 충돌 발생 시, 리더와 태그 간의 다중 태그 ID를 인식하기 위한 하이브리드 쿼리 트리 구조의 4개의 슬롯을 이용한 충돌방지 알고리즘 (Anti-Collision Algorithm with 4-Slot in RFID Systems, 4-SL)을 제안한다. 이 알고리즘은 기존의 알고리즘에 비해 리더의 타임 슬롯을 확장하고, 각 타임 슬롯에 응답하는 방식을 고안하여, 빠른 태그 인식을 할 수 있다. 본 논문은 태그 ID $[\text{prefix}+1]^{\text{th}}$ 비트와 태그 ID의 $[\text{prefix}+2]^{\text{th}}$ 비트 ~ $[\text{prefix}+3]^{\text{th}}$ 비트에서 비트 패턴을 이용해 슬롯을 선택하게 함으로써, 최대 4개의 슬롯으로 한 번에 8개의 태그를 인식 할 수 있도록 고안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 태그 ID 인식에 필요한 정보들과 기 발표된 알고리즘에 대한 연구내용을 설명한다. 3장에서는 다중 태그 인식을 위한 제안 알고리즘 기법을 나타내고, 4장에서는 제안 알고리즘의 분석 및 수식을 이용한 증명을 기술한다. 5장에서는 4장의 수식을 이용한 알고리즘별 성능을 비교하고, 6장에서는 이 논문에 대한 결론을 맺는다.

II. 관련 연구

1. 태그 ID 인식을 위한 정보

트리 기반 알고리즘에서 리더는 태그 ID의 일부분(prefix)을 질의하고, 각 태그들은 prefix가 일치하면 리더에게 일치한 prefix를 포함한 모든 태그 ID를 리더로 전송한다. 이런 태그 ID 식별 과정을 반복적으로 수행함으로써, 리더 영역 내의 모든 태그를 인식할 수 있다. 다음은 태그 인식에 필요한 세부 정보를 기술한다.

1.1 트리 방식에서의 태그 ID 상태

기본적으로 트리 기반의 알고리즘들은 계층적인 트리 구조를 가진다. 트리의 노드는 태그 ID에 대응되며, 각 비트를 나타낸다. 트리의 가지는 태그의 응답 시 충돌이 발생할 경우 분할되는 형태와 같다. 트리의 깊이(높이)는 질의-응답 수를 산출하는데 이용된다. 또한 트리의 루트에서 단말 노드까지의 경로를 탐색하면, 각 태그 ID를 알 수 있다.

트리 노드의 상태는 충돌 감지 알고리즘과 충돌 추적 알고리즘에 따라 상태에 대한 적용 방법의 차이를 가진다. 충돌 추적 알고리즘은 태그 ID에서 비트에 대한 충돌 여부와 위치, 개수를 알 수 있기 때문에, 각 노드의 상태를 다음과 같이 정의할 수 있다[2], [6].

- 인식 상태(Success State, S_S) : 태그 응답 시 유일하게 하나의 태그만이 응답한 것으로 태그 ID를 바로 인식한다.("11111")
- 무응답 상태(Idle State, S_I) : 리더가 질의한 prefix와 일치하는 태그 ID가 존재하지 않는 것으로, 어떤 태그도 응답하지 않은 상태이다.
- 충돌 상태(Collision State, S_C) : 리더가 질의한 prefix와 일치하는 다중 태그가 존재하는 경우이며, 비트 추적을 통해 충돌이 발생하지 않은 비트("111XX")는 인식된 비트를 사용함으로써, 충돌 감지 알고리즘보다 많은 정보를 가지고 새로운 prefix 생성할 수 있다. 이는 인식 성능을 높이는데 중요한 역할을 한다.

1.2 하이브리드 쿼리 트리 기반에서의 태그 ID 인식

타입 슬롯은 알로하 기반의 알고리즘에서 사용하는 방법으

로 시간 흐름에 따라 슬롯을 할당하고, 태그는 하나의 슬롯을 랜덤하게 선택하여 응답한다. 한 슬롯에 하나의 태그만 응답할 때만 태그 ID를 인식하고, 하나 이상의 태그가 응답하면 인식하지 못한다. 또한 트리 기반의 알고리즘들은 prefix를 태그들에게 전송하고, prefix에 대응되는 태그 ID를 응답받는 과정을 반복 하면서 모든 태그 ID를 인식한다. 그러나 알로하 기반의 알고리즘은 prefix를 태그들에게 전송하지 않기 때문에 최악의 경우 충돌 슬롯이 무한정 발생되거나, 모든 태그를 인식하지 못하는 경우가 발생한다.

트리 기반의 알고리즘은 한번의 prefix 전송과 한 번의 태그 응답을 반복적인 과정을 수행함으로써 태그를 인식하기 때문에 충돌 발생 시, 알로하 기반의 알고리즘보다 리더의 prefix 질의 횟수가 상대적으로 많다. 따라서 이런 문제점을 보완하고자 한 번 prefix 전송에 여러 번 태그 응답을 받을 수 있도록 타임 슬롯의 개념을 적용한 하이브리드 쿼리 트리 알고리즘이 발표되었다[6], [14], [15], [16], [20]. 따라서 제안 알고리즘은 다중 태그를 빠르고, 효율적으로 인식하기 위해 하이브리드 형태의 구조를 가진다.

2. Manchester 코드를 사용한 충돌 추적

RFID 시스템의 충돌 방지 알고리즘들은 태그 ID 비트의 충돌, 충돌의 위치, 충돌의 개수를 추적하기 위해 Manchester 디지털 코드 방식을 사용하고 있다.

Manchester 코드는 1 비트 구간에서 반드시 위상의 변화를 발생시켜 '1' 인지 '0' 인지를 판별한다. 즉 위상이 변하지 않는 상태('0' 또는 '1')로 인식 불가능한 상태를 충돌('X') 상태로 구분 할 수 있다[1], [2].

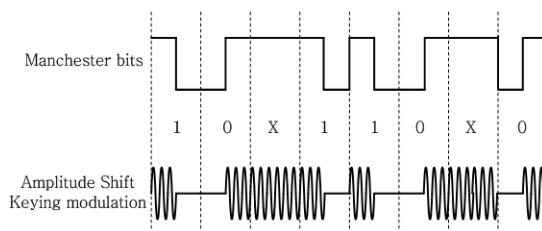


그림 3. Manchester 비트와 ASK 변조
Fig. 3. Manchester bits and ASK modulation

그림 3은 Manchester 코드 비트열과 ASK 변조 신호를 나타낸다. 이 코드는 1 비트 구간에서 위상이 변하기 때문에 비트 단위로 충돌을 추적 할 수 있음을 보여준다. 그림 3의 비트열은 "10X110X0"이고, bit3와 bit7은 비트 인식 구간의 위상의 변화가 없으므로, Manchester 코드에서는 '0'인지 '1'

인지를 구별할 수 없고, 이 비트는 오류로 판단하고, 충돌로 인식한다.

3. 쿼리 트리 알고리즘

쿼리 트리 알고리즘(Query Tree Algorithm, QTA)은 2-ary 트리 기반의 충돌 방지 알고리즘이다. 동작 방식은 질의와 응답을 과정을 통해 리더가 하나의 태그를 인식한다. 리더는 태그 ID와 일치하는 특정 prefix $q_1q_2\dots q_k (q_k \in \{0,1\})$, $1 \leq k \leq b$, b 는 태그 ID의 비트 수)를 태그에게 질의한다. 쿼리 트리 알고리즘은 인식 트리의 향이 2이면, 쿼리 트리 알고리즘으로, 트리의 향이 4인 경우, 4-ary 쿼리 트리 알고리즘이다[2], [6], [7], [8], [9], [10], [12], [17], [18].

쿼리 트리 알고리즘은 리더의 질의에 1개 이상의 태그가 응답할 경우, 최소한 그 prefix를 가지는 태그가 2개 이상인 것을 인지하고, 질의한 prefix에 '0' 과 '1'을 추가하여 태그에게 질의를 계속한다. 이 과정에서 prefix를 가지는 유일하게 1개의 태그가 리더로 응답하면 태그를 인식한다. 즉 1 개의 태그가 응답 할 때까지 prefix를 1 비트씩 추가하면서, 인식 영역 내의 모든 태그를 인식할 때까지 반복하게 된다. 이는 트리의 깊이 우선 탐색과 유사한 구조로 진행된다.

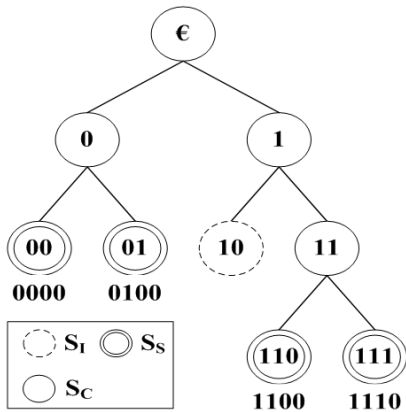


그림 4. 쿼리 트리 알고리즘
Fig. 4. Query Tree Algorithm

4-ary 쿼리 트리 알고리즘의 기본적인 동작 과정은 쿼리 트리 알고리즘과 동일하다. 리더의 인식과정에서 리더가 하나의 태그 ID를 인식하지 못할 경우, 현재의 prefix 비트열에 "00", "01", "10", "11"을 추가($q_1q_2q_3\dots q_x00$, $q_1q_2q_3\dots q_x01$, $q_1q_2q_3\dots q_x10$, $q_1q_2q_3\dots q_x11$)하여 재질의 한다[6].

4. 효율적인 분리를 통한 향상된 슬롯 이진 트리

효율적인 분리를 통한 향상된 슬롯 이진 트리(Intelligent Separation-Enhanced Slotted Binary Tree : IS-ESBT) 알고리즘은 트리 기반의 타임 슬롯 할당을 이용한 하이브리드 쿼리 트리 구조를 가진다[16]. 이 알고리즘은 Manchester 코드 방식을 적용하여, 충돌 비트 위치를 추적할 수 있다. 또한 태그가 리더의 질의에 응답할 때, prefix 다음 2비트의 정보를 패리티 메커니즘을 이용하여 슬롯을 선택한다.

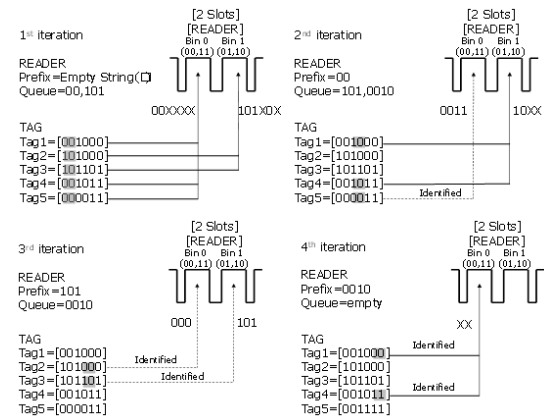


그림 5. IS-ESBT 질의-응답
Fig. 5. IS-ESBT query-response

그림 5는 IS-ESBT의 질의-응답 과정이다. 이 알고리즘은 2개의 슬롯을 사용하고, 리더의 prefix 질의에 대해 태그는 $(\text{prefix} + 1)^{\text{th}}$ 비트와 $(\text{prefix} + 2)^{\text{th}}$ 비트에 따라 슬롯0 또는 슬롯1에 응답한다. 슬롯0에는 "00" 과 "11"을 가지는 태그가 응답하고, 슬롯1에는 "01"과 "10"을 가지는 태그가 응답한다. 그림 5에서, 리더는 '€' 태그로 질의하고, 태그는 처음 2비트에 따라, "00"을 가지는 태그들은 슬롯0로, "10"을 가지는 태그들은 슬롯1로 응답한다. 태그 ID를 완전히 인식하지 못하는 경우는 큐에 새로운 질의를 넣고, 이 과정을 반복한다. 마지막 반복에서 "00" 과 "11"이 슬롯0로 응답하여 2 비트 충돌이 발생하지만, 슬롯0에는 짝수인 태그만 응답하므로, 2 비트를 인식하고, 태그 인식과정을 종료한다.

5. 4-BPSA 알고리즘

4비트 패턴에 따른 슬롯 할당 기법을 이용한 RFID 태그 충돌 방지 알고리즘(4-Bit Pattern Slot Allocation : 4-BPSA)은 4비트 패턴에 따른 슬롯 할당 기법을 이용하여 다중 태그를 식별한다[20]. 이 알고리즘은 IS-ESBT와 같이, Manchester 코드 방식을 적용하여, 충돌 비트 위치를

추적할 수 있다.

4-BPSA의 태그는 $[\text{prefix} + 4]^{\text{th}}$ 비트에 따라, 표 1과 같이, 리더에 할당된 6개의 타임 슬롯에 차례로 응답한다. 태그는 $S_i = (b_1 * 3) + (e \% 3)$ 의 결과로 나온 정수 값으로 타임 슬롯을 결정한다.

예를 들어, 처음 "1100..." 비트열을 가지는 태그가 있다면, $b_1=1$ 이고, b_2, b_3, b_4 의 비트 '1'의 개수를 $e(3)$ 로 하고, 모듈러 연산한 결과를 슬롯을 선택한다. 식 $S_i = (1 * 3) + (1 \% 3)$ 이며, 슬롯 4를 선택하여 응답한다. 4-BPSA의 리더는 슬롯 3에 "1XXX..." 응답하였다면, Manchester 코드 검출 방식에 따라, 충돌 비트를 인지한다. 슬롯 3은 표 1에 따라 2 종류의 비트 패턴이 응답하므로, 리더는 "1000..." 과 "1111..."을 인식할 수 있다.

따라서 4-BPSA의 태그는 $[\text{prefix} + 4]^{\text{th}}$ 에 따라, 식 S_i 결과를 도출하여, 리더에 할당된 6개의 슬롯에 차례로 응답하고, 리더는 $[\text{prefix} + 4]^{\text{th}}$ 까지 발생한 충돌과 전체 태그 ID에 대해 한 개의 충돌이 발생한 경우와 충돌 비트가 없는 경우 외에는 재 질의를 통해 모든 태그를 인식할 때까지 반복한다.

표 1. 4 비트 패턴에 따른 슬롯 할당
Table 1. The decision method for each time slot.

$b_1 b_2 b_3 b_4$	b_1	$(e \% 3)$	S_i
0000	0	0	0
0111			
0001			
0010			
0100			
0011			
0101	0	2	2
0110			
1000			
1111	1	0	3
1001			
1010	1	1	4
1100			
1011			
1101	1	2	5
1110			

이 알고리즘은 리더에 매 반복마다 6개의 타임 슬롯이 고정되어 있고, 태그는 ID의 '1'의 개수와 '1'의 개수에 대한 모듈러 연산을 함께 수행해야 하므로, 태그 ID 인식과정에서 리더와 태그의 자원에 대한 부하를 초래할 수 있다.

III. 4 슬롯을 이용한 충돌방지 알고리즘

제안 알고리즘은 리더의 인식 영역에서 다중 태그를 빠르

게 인식할 수 있다. 4 슬롯을 이용한 충돌방지 알고리즘 (Anti-Collision Algorithm with 4-Slot : 4-SL)은 트리 기반의 타임 슬롯을 이용한 하이브리드 쿼리 트리 알고리즘 구조를 가진다. 그리고 최대 4개의 타임 슬롯으로 확장하여 한번에 8개의 태그를 인식 할 수 있도록 고안하였다. 또한 다른 알고리즘과의 worst case에 대한 수학적 성능 평가를 추가하였다. 아래 섹션에서는 비트 충돌 패턴과 각 슬롯에 따른 인식 방법을 사용한 4-SL 알고리즘 동작 과정을 설명한다.

1. 슬롯에 따른 태그 ID 인식

제안 알고리즘은 고정된 4 개의 슬롯을 사용한다. 아래의 각 슬롯의 응답에 따른 리더의 인식 방법이다. 표 2와 같이, 태그는 prefix 다음 첫 번째 비트($[\text{prefix} + 1]^{\text{th}}$, $F \in \{0 \text{ or } 1\}$), prefix 다음 두 번째와 세 번째 비트($[\text{prefix} + 2]^{\text{th}} \sim [\text{prefix} + 3]^{\text{th}}$, $B_2 \in \{00 \text{ or } 11\}$, $B_1 \in \{01 \text{ or } 10\}$) 패턴에 따라, 리더로 prefix를 제외한 나머지 ID를 전송한다.

표 2. 타임 슬롯의 선택 방법
Table 2. The decision method for each time slot.

Slot number	$[\text{prefix} + 1]^{\text{th}}$ bit : F	$[\text{prefix} + 2]^{\text{th}}$ bit ~ $[\text{prefix} + 3]^{\text{th}}$ bit : B
Slot 0	'0'	"00" or "11" : B_2
Slot 1	'0'	"01" or "10" : B_1
Slot 2	'1'	"00" or "11" : B_2
Slot 3	'1'	"01" or "10" : B_1

2. 비트 충돌 패턴에 따른 태그 ID 인식

제안 알고리즘은 리더가 태그로부터 전송받은 데이터를 인식하지 못할 경우, 인식하지 못한 비트 위치와 개수를 Manchester 코드 방식을 이용하여 파악할 수 있으며, 충돌 비트 패턴에 따라 인식하거나 재 질의를 결정한다. 다음은 패턴에 따른 인식 방법을 나타낸다.

- $[\text{prefix} + 3]^{\text{th}}$ 비트까지 태그 ID의 충돌이 발생한 경우: 태그로부터 전송받은 태그 ID에서 $[\text{prefix} + 3]^{\text{th}}$ 비트까지 충돌이 발생한 경우이며, 이 경우는 $[\text{prefix} + 1]^{\text{th}}$ 비트는 '0' 또는 '1'에 따라 응답하는 슬롯이 상이하므로, 충돌이 발생하지 않고, $[\text{prefix} + 2]^{\text{th}} \sim [\text{prefix} + 3]^{\text{th}}$ 는 충돌이 발생하더라도, 2 비트 충돌에 대해서 "00"과 "11"을 Manchester 코드를 이용하여 추적할 수 있고, 반대의 경우인 "01"과 "10" 패턴도 검출할 수 있다. 따라서 2 비트 충돌에 대해서 2 개의 태그 ID를 인식할 수 있다.

- $(\text{prefix}+4)^{\text{th}}$ 비트부터 태그 ID의 충돌이 1개 인 경우 : 태그로부터 전송받은 태그 ID에서 1 비트 충돌('X')만 난 경우이다. 그러나 제안 알고리즘은 충돌 추적 알고리즘으로, 비트의 위치를 알 수 있고, 그 위치에 '0' 과 '1'을 넣어, 2개의 태그 ID를 인식한다.
- $(\text{prefix}+4)^{\text{th}}$ 비트부터 태그 ID의 충돌이 2개 이상인 경우 : 태그로부터 전송받은 태그 ID에서 2 비트 충돌("XX", "X...X", "...XXX...", "X...X...X" 또는 "XX.....X..." 등)이상 발생한 경우이며, 이 패턴은 한 번에 태그 ID를 인식할 수 없으므로, 새로운 prefix를 생성한다. 이후 스택에 저장한 후 재 질의한다.

3. 리더의 질의와 태그의 응답

제안 알고리즘은 3.1 절과 3.2 절의 인식 방법을 이용하여 다중 태그들의 충돌 발생 시 태그 ID를 인식할 수 있다. 그림 6 은 리더와 태그 간의 다중 태그 인식 절차를 나타낸다. 태그 8개("000111", "110111", "100000", "010000", "001000", "111000", "011111", "101111")가 있을 경우, 태그1 과 태그 7은 F(0) 이고, F 비트 다음 2 비트 패턴이 B₂ 이므로, 슬롯0 에 응답한다. 이 때 슬롯 0에는 $(\text{prefix}+3)^{\text{th}}$ 비트까지 2개의 충돌("0XX111")만 발생하고, 리더는 Manchester 코드 검출 방식을 적용하여, 2비트 충돌을 인지하고, 충돌 비트 위치에 "00" 과 "11"을 인식할 수 있다.

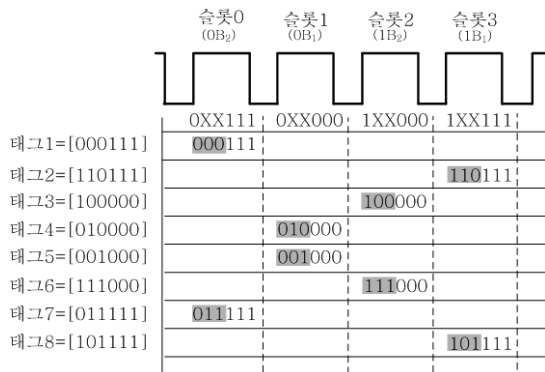


그림 6. 태그 ID의 인식과정
Fig. 6. tag IDs decision process

태그2와 태그8은 F(1) 이고, F 비트 패턴이 B₁ 이므로, 슬롯 4에 응답한다. 이 때 슬롯 4에는 슬롯1과 같이, 충돌이 발생하고, 리더는 "01" 과 "10"을 인식할 수 있다. 나머지 태그들도 위와 같은 방식으로, 슬롯을 결정하고, 리더로 ID를 전송한다.

4. 제안 알고리즘의 흐름도

그림 7은 제안 알고리즘의 흐름도이다. 리더는 초기에 'F' 을 태그로 전송한다. 이 질의 명령은 리더의 인식 영역내의 모든 태그들이 응답하게 하고, 각 태그들은 자신의 태그 ID에서 $(\text{prefix}+1)^{\text{th}}$ 비트 F와 $(\text{prefix}+2)^{\text{th}}$ 비트 ~ $(\text{prefix}+3)^{\text{th}}$ 비트가 B₁ 인지 B₂ 인지에 따라, 슬롯 0부터 순차적으로 슬롯 3 까지 선택하여 응답한다.

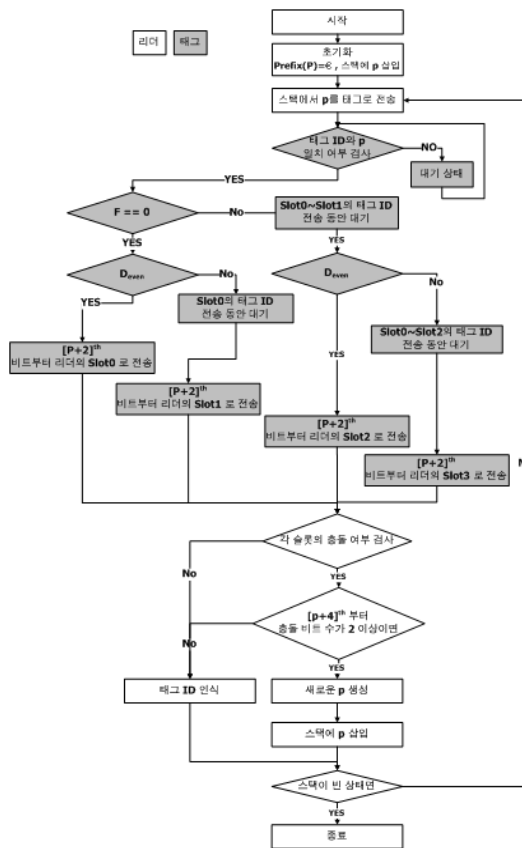


그림 7. 제안 알고리즘의 흐름도
Fig. 7. The algorithm flow chart

이 후, 리더는 슬롯0 에서 슬롯3 까지 각 타임 슬롯의 비트 패턴을 분석하여 충돌 여부와 인식여부를 검사한다. 충돌이 발생하지 않은 슬롯은 태그 ID를 인식하고, 충돌이 발생한 슬롯은 3.1 절과 3.2 절의 인식 방법에 따라, 충돌 비트가 $(\text{prefix}+3)^{\text{th}}$ 비트까지 발생한 경우와 1개 이하인 경우는 태그 ID를 인식한다. 그러나 $(\text{prefix}+4)^{\text{th}}$ 비트부터 2개 이상인 경우는 새로운 prefix를 생성한다. 이 때, 리더와 태그가 응답 슬롯에 대한 규칙에 따라, 리더는 prefix 다음 3번째 비트

까지는 인식 가능하다. 따라서 새로운 prefix는 $\text{prefix} + F('0' \text{ 또는 } '1') + B$ 비트열(B_1, B_2)로 구성된다.

제안 알고리즘의 prefix은 $[\text{prefix} + 3]$ 비트씩 증가한다. 그 이후 리더는 스택에 저장된 새로운 prefix를 차례로 pop하여 태그로 질의 명령을 전송한다. 마지막으로 스택이 빈 상태가 되면 모든 태그들을 인식한 것으로 종료한다.

IV. 알고리즘 분석

이 절에서는 제안 알고리즘을 분석한다. 제안 알고리즘은 prefix 수를 추출하여 다중 태그를 인식하는데, 지연 시간을 계산한다. 아래의 성능 분석은 리더의 prefix 수에 대한 worst case 만을 다룬다.

1. 알고리즘 정리

정리 1 : 태그 ID 인식 과정에서 리더의 질의 수(Q_N)는 prefix 수(N_N)와 같다.

$$Q_N = N_N \quad (1)$$

증명: 리더는 prefix를 태그로 전송하고, 응답받은 태그 ID는 각 타임 슬롯에서 충돌 여부를 확인한다. 타임 슬롯은 충돌상태, 인식상태, 무응답 상태로 분류되고, 인식상태는 태그 ID를 바로 인식할 수 있고, 무응답 상태는 조건에 맞는 태그가 없어 응답하지 않는 경우이며, 충돌 상태는 제안 알고리즘의 조건에 따라 새로운 prefix를 생성하거나, $[\text{prefix}+3]$ 비트까지만 충돌이 발생한 경우는 인식상태와 같이, 태그 ID를 인식한다. 따라서 새로운 prefix가 생성되는 경우는 $[\text{prefix}+4]$ 비트부터 충돌이 2개 이상 발생한 경우이며, 모든 태그 ID를 인식할 때까지 반복된다. 결론적으로 리더가 태그로 prefix를 전송하는 것은 모든 태그를 인식하지 못한 경우이므로, 질의 수는 생성되는 prefix 수와 같다.

4-SL은 Manchester 코드 방식을 사용하므로, 충돌을 감지만 하는 QTA와 달리, 충돌 비트의 개수와 위치를 추출 할 수 있다. 따라서 생성되는 prefix 수는 비트 길이에 영향을 받지 않고, 태그 수에 영향을 받는다.

2. 제안 알고리즘의 worst case

제안 알고리즘은 IS-ESBT와 달리, 8-ary 구조의 충돌 검출 방식이다. 제안 알고리즘은 4개의 타임 슬롯을 사용하며, 각 타임 슬롯에서 $[\text{prefix}+4]$ 비트부터 충돌이 2개 이상 발생한 경우, prefix가 생성된다. 그리고 한 번의 prefix 생성

시, 인식 할 수 있는 최대 태그 수는 8개이며, 제안 알고리즘의 생성 prefix 수에 대한 worst case(N_{\max})는 (2)와 같다. 이 조건은 $8^2 < n \leq 8^{b-3}$ 의 조건에서 만족한다.

$$N_{\max} \leq \frac{8}{7}n - \frac{8}{7} \quad (2)$$

증명: 제안 알고리즘은 생성 되는 prefix를 QTA와 같이, 트리 구조로 표현할 수 있다. 이 트리는 인식 트리로 표현하며, 트리의 노드는 생성되는 prefix로 대응시킨다. 태그의 ID는 유일한 비트열 $R_1R_2R_3...R_b$ ($R_b \in \{0, 1\}$, b 는 태그 ID의 비트 길이)로 나타내고, 트리의 단말 노드는 모든 비트가 인식된 태그 ID이다. 인식 트리의 높이는 h ($0 < h \leq \lfloor \log_8 n + 1 \rfloor$)로 구성하고, 트리의 높이가 한 단계씩 증가하면, prefix 비트 길이는 3^h 로 증가한다.

4-SL은 인식 트리의 각 높이에서, prefix가 생성되는 수는 상이하다. 태그의 수가 $\lfloor \log_8 n - 1 \rfloor$ 를 기준으로, 슬롯에서 인식되는 트리의 노드는 완전 트리를 형성하고, 이 때 prefix 수(N_N)는 8^h 개다. 이후, 태그 수가 $\lfloor \log_8 n - 1 \rfloor < h \leq \lfloor \log_8 n + 1 \rfloor$ 를 기준으로, 최소 태그 수가 2개이면, 충돌 비트 수는 2비트 이상 발생 할 수 있다. 모든 태그를 인식하기 위해서 $n/2$ 개의 prefix를 생성해야 한다. 높이 h 에서 최대 $N_{\max}(n, h)$ 은 (3)과 같다.

$$N_{\max}(n, h) = \begin{cases} 8^h, & 0 < h \leq \lfloor \log_8 n - 1 \rfloor \\ \frac{n}{2}, & \lfloor \log_8 n - 1 \rfloor < h \leq \lfloor \log_8 n + 1 \rfloor \end{cases} \quad (3)$$

식 (3)을 이용하여, 최대 $N_{\max}(n, h)$ 는 (4)와 같이 도출한다.

$$\begin{aligned} N_{\max} &\leq \sum_{h=1}^{\lfloor \log_8 n + 1 \rfloor} N_{\max} \\ &\leq \sum_{h=1}^{\lfloor \log_8 n - 1 \rfloor} 8^h + \sum_{h=\lfloor \log_8 n \rfloor}^{\lfloor \log_8 n + 1 \rfloor} \frac{n}{2} \\ &\leq \frac{8}{7}n - \frac{8}{7} \end{aligned} \quad (4)$$

IS-ESBT의 worst case를 구하기 위해, IS-ESBT의 prefix의 길이가 2비트씩 증가한다고 가정한다. 인식 트리의

높이는 $\lfloor \log_4 n + 1 \rfloor$ 까지 $n/2$ 개의 prefix를 생성해야 된다. 따라서 높이 h 에서 최대 $N_{\max}(n, h)$ 은 다음과 같다.

$$N_{\max}(n, h) = \begin{cases} 4^h, & 0 < h \leq \lfloor \log_4 n - 1 \rfloor \\ \frac{n}{2}, & \lfloor \log_4 n \rfloor \leq h \leq \lfloor \log_4 n + 1 \rfloor \end{cases} \quad (5)$$

V. 성능평가

본 장에서는 이 논문에서 제안 알고리즘의 성능을 평가한다. 성능 평가는 4장의 성능 분석을 바탕으로 성능 평가를 진행하였고, 실제 무선 통신에서 존재하는 여러 제약이나 환경적 문제를 고려하지 않았다. 알고리즘의 인식 지연 시간은 리더의 prefix 비트 수와 리더와 태그간의 질의 수와 응답 수를 기준으로 하며, 각 알고리즘의 수식을 바탕으로 산출하였다. 리더와 태그 간의 질의-응답 시, 응답 방식이 알고리즘 별로 상이하기 때문에 질의 수를 기준으로, IS-ESBT는 질의 수 + (질의 수 * 타임 슬롯 수(2)), 4-BPSA는 질의 수 + (질의 수 * 타임 슬롯 수(6)), 제안 알고리즘은 질의 수 + (질의 수 * 타임 슬롯 수(4))으로 산출하였다.

본 성능 평가에 사용된 태그 ID는 EPC Tag Data Standards Version 1.6을 따른다[19]. 태그 ID의 길이는 96비트이며, 태그 ID의 할당은 랜덤으로 할당하였다.

그림 8은 각 알고리즘의 질의 수와 응답 수를 나타낸다. 성능 평가 결과는 제안 알고리즘이 질의 수와 응답 수에서 IS-ESBT에 비해 약 29% 성능 저하를 보였고, 4-BPSA에 비해 약 30% 개선되었다. worst case 분석에서, IS-ESBT의 수치가 4-SL과 4-BPSA에 비해, 상대적으로 적은 수의 슬롯(2개)을 사용하기 때문에 질의-응답 수에서도 좋은 성능을 보인다. 4-BPSA는 태그 인식을 위해 6개의 타임 슬롯을 사용한다. 이 슬롯은 태그 인식과정에서 유동적으로 슬롯의 수를 변동할 수 없고, 고정되어 있기 때문에, 4-SL과 IS-ESBT에 질의-응답 수에는 낮은 성능을 보인다. 이는 할당된 모든 타임 슬롯에 태그 ID가 응답 할 수는 없다. 이로 인한 시간 지연이 발생한다.

타임 슬롯 수에 의한 시간 지연을 해결하기 위해서는, 리더의 질의에 대해, 응답 하는 태그 ID의 효율적인 분배가 이루어져야 한다. 또한 데이터 송·수신시, 무선 상태가 좋지 않다면, 한 번의 질의-응답에 많은 정보를 받을 수 있는 타임 슬롯

의 수가 많은 알고리즘이 적절할 것이고, 반대의 경우라면, 다수의 질의-응답을 통해, 태그 ID를 인식하는 것이 적절할 것으로 보인다.

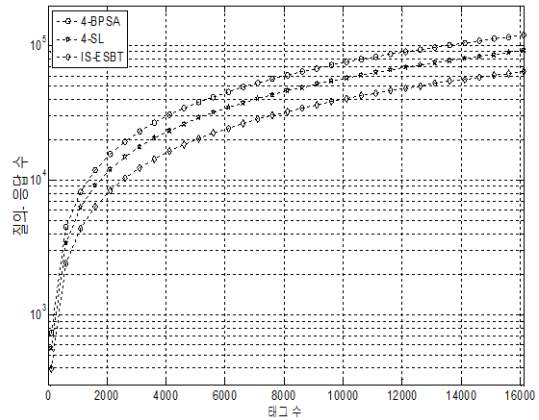


그림 8. 질의 응답 수
Fig. 8. The number of query-responses

그림 9는 알고리즘별 질의 비트 수에 대한 결과를 나타낸다. 질의 비트 수에서는 제안 알고리즘이 다른 알고리즘에 비해 좋은 성능을 보인다. 전송 비트 수는 IS-ESBT에 비해 약 2%, 4-BPSA에 비해 약 1% 개선되었다.

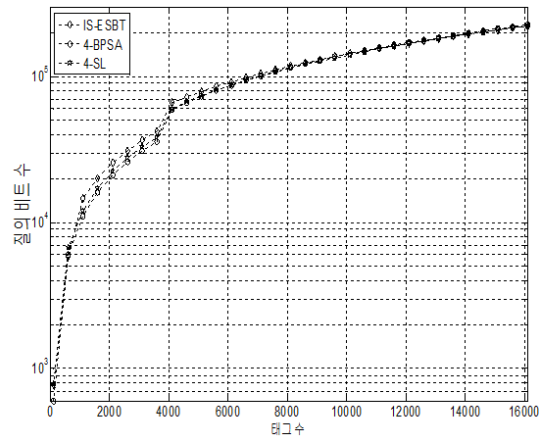


그림 9. 질의 비트 수
Fig. 9. The number of query bits

알고리즘들은 태그 수에 영향을 받고, 태그 수에 따라 질의 prefix를 생성하므로, IS-ESBT, 4-BPSA에 비해 질의-응답 하는 과정에서 태그 ID의 충돌 발생시, $[\text{prefix} + 1]^{\text{th}}$ 비트 (F)와 $[\text{prefix} + 2]^{\text{th}}$ 비트 ~ $[\text{prefix} + 3]^{\text{th}}$ 비트 (B)에 따라

슬롯을 결정하기 때문에 태그 ID의 분배가 향상된다.

표 3은 그림 9의 성능 결과에 대한 수치를 나타낸다. 각 알고리즘별 태그 수에 대한 질의 비트 수이며, 태그 수에 따라 알고리즘별 질의 비트 수의 차이를 나타낸다. 4-SL은 태그 수가 6,100개부터 4-BPSA, IS-ESBT보다 좋은 결과를 나타낸다.

표 3. 알고리즘별 질의 비트 수

Table 3. The number of query bits by each algorithm.

태그 수	100	600	1,100	1,800	2,100	2,600	3,100	3,600	4,100	4,600	5,100
알고리즘											
4-SL	774	6,708	11,958	17,208	22,458	27,708	32,958	38,208	60,368	67,118	73,868
4-BPSA	800	6,084	11,064	16,064	21,064	26,064	31,064	36,064	59,512	68,512	73,512
IS-ESBT	772	5,856	14,604	20,104	25,604	31,104	36,604	42,104	66,044	72,544	78,044

태그 수	5,600	6,100	6,600	7,100	7,600	8,100	8,600	9,100	9,600	10,100	10,600
알고리즘											
4-SL	80,618	87,368	94,118	100,868	107,618	114,368	121,118	127,868	134,618	141,368	148,118
4-BPSA	80,512	87,512	94,512	101,512	108,512	115,512	122,512	129,512	136,512	143,512	150,512
IS-ESBT	85,544	92,044	98,544	105,044	111,544	118,044	124,544	131,044	137,544	144,044	150,544

태그 수	11,100	11,600	12,100	12,600	13,100	13,600	14,100	14,600	15,100	15,600	16,100
알고리즘											
4-SL	154,868	161,618	168,368	175,118	181,868	188,618	195,368	202,118	208,868	215,618	222,368
4-BPSA	157,512	164,512	171,512	178,512	185,512	192,512	199,512	206,512	213,512	220,512	227,512
IS-ESBT	157,044	163,544	170,044	176,544	183,044	189,544	196,044	202,544	209,044	215,544	222,044

VI. 결론

RFID는 무선 주파수를 이용하여, 리더와 태그간의 데이터를 전송하기 때문에, 리더의 위치와 태그의 수에 따라 인식 방법을 달리한다. 리더와 리더간의 간섭에 따른 통신 방식과 하나의 리더와 다수의 태그간의 통신 방식에 따라 적절한 설계가 필요하다. 본 논문은 하나의 리더와 다중 태그인식을 위해 리더와 태그간의 응답 방식에 대한 알고리즘을 제안하였다.

제안한 알고리즘은 리더는 슬롯 0부터 슬롯 3까지의 타임 슬롯을 할당하고, 태그는 자신의 ID에서 $(\text{prefix}+1)^{\text{th}}$ 비트 F 와 $(\text{prefix}+2)^{\text{th}}$ 비트 $\sim (\text{prefix}+3)^{\text{th}}$ 비트가 B_1 인지 B_2 인지에 따라, 슬롯 0 부터 순차적으로 슬롯 3 까지 선택하여 응답한다.

이 방식은 첫 번째 비트(F)는 '0' 또는 '1' 밖에 존재하지 않기 때문에, 해당 슬롯에 응답할 때 충돌이 발생하지 않는다. 나머지 2 비트는 B_1 인지 B_2 에 따라 응답하므로, 충돌이 발생하더라도, Manchester 코드를 이용하여 판별할 수 있다. 결론적으로 본 논문은 간단하면서도, 확정적인 방법을 제안하여 한 번의 인식과정에서 최대 8개의 태그 ID를 인식할 수 있는 방법을 제안하였다.

제안 알고리즘과 비교 알고리즘의 worst case를 분석하고,

IS-ESBT, 4-BPSA, 제안 알고리즘의 리더와 태그 간의 질의 수와 응답 수, 리더의 질의 비트 수를 도출하였다. 성능 평가 결과에서, 제안된 알고리즘은 질의 수와 응답 수의 경우는 IS-ESBT에 비해 약 29% 성능 저하를 나타냈고, 4-BPSA에 비해 약 30% 성능 개선을 나타냈다. 질의 비트 수는 IS-ESBT에 비해 약 2%, 4-BPSA에 비해 약 1%의 성능이 개선되었다.

참고문헌

- [1] K. Finkenzeller, "RFID Hand Book : Fundamentals and Applications in Contactless Smart Card and Identification, 2nd Ed.," Chichester, Sussex, U. K., WILEY, pp. 13-21, 2003.
- [2] Y. Kim, "N-ary Query Tree Algorithm for Anti-collision in RFID System," Doctoral dissertation, Graduate School of Kyungpook National University, Feb, 2013.
- [3] D. W. Engels and S. E. Sarma, "The Reader Collision Problem," In Proceedings of IEEE International Conference on System, Man and Cybernetics, Hammamet, Vol. 3, Oct, 2002.
- [4] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An Anticollision Algorithm for the Reader Collision," In proceedings of IEEE International Conference on Communications, Vol. 2, pp. 1206-1210, May, 2003.
- [5] 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/Dec, 2001.
- [6] Y. Kim, M. Ryoo, J. Park, "Anti-Collision Algorithm for Improvement of Multiple Tag Identification in RFID System," Journal of Korea Society of IT Services, Vol. 12, No. 3, pp. 331-343, Sep, 2013.
- [7] Y. Kim, S. Kim, S. Lee, and K. Ahn, "Improved 4-ary Query Tree Algorithm for Anti-Collision in RFID System," International Conference on Advanced Information Networking and Applications, pp. 699-704, May, 2009.
- [8] C. Law, K. Lee, and K. Y. Siu, "Efficient

- Memoryless protocol for Tag Identification," Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, ACM, pp. 75-84, Aug, 2000.
- [9] J. Myung, W. Lee, "Adaptive Binary Splitting: A RFID Tag Collision Arbitration Protocol for Tag Identification," IEEE International Conference on Broadband Networks, Vol. 1, pp. 347-355, Oct, 2005.
- [10] D. Baek, S. Kim, K. 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, Sep, 2008.
- [11] J. Cha, J. Kim, "Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system," Consumer Communications and Networking Conference, Vol. 2, pp. 768-772, Jan, 2006.
- [12] J. Eom, T. Lee, R. Rietman, A. Yener, "An efficient framed-slotted ALOHA algorithm with pilot frame and binary selection for anti-collision of RFID tags," IEEE Communications Letters, vol. 12, no. 11, pp. 861-863, Nov, 2008.
- [13] L. Zhu, T. -S. P. Yum, "A Critical Survey and Analysis of RFID Anti-Collision Mechanisms," IEEE Communications Magazine, vol. 49, no. 5, pp. 214-221, May, 2011.
- [14] S. Kim, T. Yun, "An Anti Collision Algorithm Using Efficient Separation in RFID system," Journal of the Korea Society of Computer and Information, Vol. 18, No. 11, pp. 87-97, Nov, 2013.
- [15] J. Ryu, H. Lee, Y. Seok, T. Kwon, Y. Choi, "A Hybrid Approach to Arbitrate Tag Collisions in RFID systems," Journal of KIISE, Vol. 34, No. 6, pp. 483-492, Dec, 2007.
- [16] S. Kim, Y. Kim, K. Ahn, "An Enhanced Slotted Binary Tree Algorithm with Intelligent Separation in RFID Systems," the Fourteenth IEEE Symposium on Computers and Communications, pp. 237-242, July, 2009
- [17] J. Myung, 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.
- [18] D. Shih, P. Sun, D. C. Y, S. M. Huang, "Taxonomy and Survey of RFID Anti-collision protocols," Computer Communications, vol. 29, no. 11, pp. 2150-2166, July, 2006.
- [19] GS1: EPCglobal Tag Data Standards Version 1.6, http://www.gs1.org/gsm/kc/epcglobal/tds/tds_1_6-RatifiedStd-20110922.pdf
- [20] Y. Kim, S. Kim, K. Chung, K. Ahn, "A RFID Tag Anti-Collision Algorithm Using 4-Bit Pattern Slot Allocation Method," Journal of the Korean Society for Internet Information, Vol. 14, No. 4, pp. 25-33, Aug, 2013.

저 자 소 개



김 용 환
 2004: 경운대학교
 컴퓨터공학과 공학사.
 2006: 경북대학교
 컴퓨터공학과 공학석사.
 2013: 경북대학교
 컴퓨터공학과 공학박사
 현 재: 경운대학교
 컴퓨터공학과 조교수
 관심분야: 임베디드 시스템, RFID
 Email : hypnus@ikw.ac.kr



김 성 수
 2002: 금오공과대학교
 컴퓨터공학과 공학사.
 2005: 경북대학교
 컴퓨터공학과 공학석사.
 2012: 경북대학교
 컴퓨터공학과 공학박사
 현 재: 경운대학교
 모바일공학과 조교수
 관심분야: 임베디드 시스템, RFID
 Email : ninny@ikw.ac.kr



류 명 춘
 1989: 영남대학교
 컴퓨터공학과 공학사.
 1991: 영남대학교
 컴퓨터공학과 공학석사.
 2009: 영남대학교
 컴퓨터공학과 공학박사
 현 재: 경운대학교
 컴퓨터공학과 교수
 관심분야: 지능정보시스템,
 데이터베이스
 Email : mcryoo@ikw.ac.kr



박 준 호
 1990 : 영남대학교
 컴퓨터공학과 공학사.
 1994 : 영남대학교
 컴퓨터공학과 공학석사.
 2002 : 영남대학교
 컴퓨터공학과 공학박사
 현 재: 경운대학교
 컴퓨터공학과 교수
 관심분야: 컴퓨터시스템설계, 영상처리
 Email : jhpark@ikw.ac.kr



정 경 호
 2000 : 대구대학교
 컴퓨터공학과 공학사.
 2002 : 경북대학교
 컴퓨터공학과 공학석사.
 2011 : 경북대학교
 컴퓨터공학과 공학박사
 현 재: 경운대학교
 컴퓨터공학과 조교수
 관심분야: 임베디드 시스템, RFID,
 정보보호
 Email : mccart@ikw.ac.kr