

# 충돌 정보와 $m$ -bit 인식을 이용한 적응형 RFID 충돌 방지 기법

## Adaptive RFID anti-collision scheme using collision information and $m$ -bit identification

이 제 율<sup>1</sup>                      신 중 민<sup>2</sup>                      양 동 민<sup>\*</sup>  
Je-yul Lee                      Jongmin Shin                      Dongmin Yang

### 요 약

RFID(Radio Frequency Identification)시스템은 하나의 RFID 리더, 다수의 RFID 태그 장치들로 이루어진 비접촉방식의 근거리 무선 인식 기술이다. RFID 태그는 자체적인 연산 수행이 가능한 능동형 태그와 이에 비해 성능은 떨어지지만 저렴한 가격으로 물류 유통에 적합한 수동형 태그로 나눌 수 있다. 데이터 처리 장치는 리더와 연결되어 리더가 전송받은 정보를 처리한다. RFID 시스템은 무선 주파수를 이용해 다수의 태그를 빠른 시간에 인식할 수 있다. RFID 시스템은 유통, 물류, 운송, 물품관리, 출입 통제, 금융 등 다양한 분야에서 응용되고 있다. 하지만 RFID 시스템을 더욱 확산시키기 위해서는 가격, 크기, 전력소모, 보안 등 해결할 문제가 많다. 그 문제들 중에서 본 논문에서는 다수의 수동형 태그를 인식할 때 발생하는 충돌 문제를 해결하기 위한 알고리즘을 제안한다. RFID 시스템에서 다수의 태그를 인식하기 위한 충돌 방지 기법에는 확률적인 방식과 결정적인 방식 그리고 이를 혼합한 하이브리드 방식이 있다.

본 논문에서는 우선 기존에 있던 확률적 방식의 충돌방지기법인 알로하 기반 프로토콜과 결정적 방식의 충돌방지기법인 트리 기반 프로토콜에 대해 소개한다. 알로하 기반 프로토콜은 시간을 슬롯 단위로 나누고 태그들이 각자 임의로 슬롯을 선택하여 자신의 ID를 전송하는 방식이다. 하지만 알로하 기반 프로토콜은 태그가 슬롯을 선택하는 것이 확률적이기 때문에 모든 태그를 인식하는 것을 보장하지 못한다. 반면, 트리 기반의 프로토콜은 리더의 전송 범위 내에 있는 모든 태그를 인식하는 것을 보장한다. 트리 기반의 프로토콜은 리더가 태그에게 질의 하면 태그가 리더에게 응답하는 방식으로 태그를 인식한다. 리더가 질의 할 때, 두 개 이상의 태그가 응답 한다면 충돌이라고 한다. 충돌이 발생하면 리더는 새로운 질의를 만들어 태그에게 전송한다. 즉, 충돌이 자주 발생하면 새로운 질의를 자주 생성해야하기 때문에 속도가 저하된다. 그렇기 때문에 다수의 태그를 빠르게 인식하기 위해서는 충돌을 줄일 수 있는 효율적인 알고리즘이 필요하다. 모든 RFID 태그는 96비트의 EPC(Electronic Product Code)의 태그ID를 가진다. 이렇게 제작된 다수의 태그들은 회사 또는 제조업체에 따라 동일한 프리픽스를 가진 유사한 태그ID를 가지게 된다. 이 경우 쿼리 트리 프로토콜을 이용하여 다수의 태그를 인식 하는 경우 충돌이 자주 일어나게 된다. 그 결과 질의-응답 수는 증가하고 유휴 노드가 발생하여 식별 효율 및 속도에 큰 영향을 미치게 된다. 이 문제를 해결하기 위해 충돌 트리 프로토콜과  $M$ -ary 쿼리 트리 프로토콜이 제안되었다. 하지만 충돌 트리 프로토콜은 쿼리 트리 프로토콜과 마찬가지로 한번에 1비트씩 밖에 인식을 못한다는 단점이 있다. 그리고 유사한 태그ID들이 다수 존재할 경우,  $M$ -ary 쿼리 트리 프로토콜을 이용해 인식 하면, 불필요한 질의-응답이 증가한다.

본 논문에서는 이러한 문제를 해결하고자  $M$ -ary 쿼리 트리 프로토콜의 매핑 함수를 이용한  $m$ -비트 인식, 맨체스터 코딩을 이용한 태그 ID의 충돌정보,  $M$ -ary 쿼리 트리의 깊이를 하나 감소시킬 수 있는 예측 기법을 이용하여 성능을 향상시킨 적응형  $M$ -ary 쿼리 트리 프로토콜을 제안한다. 본 논문에서는 기존의 트리기반의 프로토콜과 제안하는 기법을 동일한 조건으로 실험하여 비교 분석 하였다. 그 결과 제안하는 기법은 식별시간, 식별효율 등에서 다른 기법들보다 성능이 우수하다.

☞ 주제어 : 전파 식별, 충돌 방지,  $M$ -ary 질의 트리 프로토콜

1 Information & Communications Engineering, Daejeon University,  
#62 Dachak-ro, Dong-gu, Daejeon, 300-716, Korea

2 Samsung Electronics, #416, Maetan 3-dong, Yeongtong-gu,  
Suwon-si, Gyeonggi-do. Korea

\* Corresponding author (dmyang@dju.kr) Dongmin Yang

☆ 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국  
연구재단의 기초연구사업 지원을 받아 수행된 것임  
(2012R1A1A1042703)

☆ 본 논문은 2013년도 한국인터넷정보학회 춘계학술발표대회  
우수논문의 확장버전임.

[Received 24 July 2013, Reviewed 9 August 2013, Accepted 2  
October 2013]

## ABSTRACT

RFID(Radio Frequency Identification) system is non-contact identification technology. A basic RFID system consists of a reader, and a set of tags. RFID tags can be divided into active and passive tags. Active tags with power source allows their own operation execution and passive tags are small and low-cost. So passive tags are more suitable for distribution industry than active tags. A reader processes the information receiving from tags. RFID system achieves a fast identification of multiple tags using radio frequency. RFID systems has been applied into a variety of fields such as distribution, logistics, transportation, inventory management, access control, finance and etc. To encourage the introduction of RFID systems, several problems (price, size, power consumption, security) should be resolved. In this paper, we proposed an algorithm to significantly alleviate the collision problem caused by simultaneous responses of multiple tags. In the RFID systems, in anti-collision schemes, there are three methods: probabilistic, deterministic, and hybrid.

In this paper, we introduce ALOHA-based protocol as a probabilistic method, and Tree-based protocol as a deterministic one. In Aloha-based protocols, time is divided into multiple slots. Tags randomly select their own IDs and transmit it. But Aloha-based protocol cannot guarantee that all tags are identified because they are probabilistic methods. In contrast, Tree-based protocols guarantee that a reader identifies all tags within the transmission range of the reader. In Tree-based protocols, a reader sends a query, and tags respond it with their own IDs. When a reader sends a query and two or more tags respond, a collision occurs. Then the reader makes and sends a new query. Frequent collisions make the identification performance degrade. Therefore, to identify tags quickly, it is necessary to reduce collisions efficiently. Each RFID tag has an ID of 96bit EPC(Electronic Product Code). The tags in a company or manufacturer have similar tag IDs with the same prefix. Unnecessary collisions occur while identifying multiple tags using Query Tree protocol. It results in growth of query-responses and idle time, which the identification time significantly increases. To solve this problem, Collision Tree protocol and M-ary Query Tree protocol have been proposed. However, in Collision Tree protocol and Query Tree protocol, only one bit is identified during one query-response. And, when similar tag IDs exist, M-ary Query Tree Protocol generates unnecessary query-responses.

In this paper, we propose Adaptive M-ary Query Tree protocol that improves the identification performance using  $m$ -bit recognition, collision information of tag IDs, and prediction technique. We compare our proposed scheme with other Tree-based protocols under the same conditions. We show that our proposed scheme outperforms others in terms of identification time and identification efficiency.

☞ keyword : RFID, anti-collision, M-ary Query Tree Protocol

## 1. 서 론

RFID(Radio Frequency Identification) 시스템은 전파를 이용해 객체에 부착된 태그를 인식하여 객체를 식별하거나 데이터를 기록하는 비접촉방식의 근거리 무선인식 기술이다[1]. RFID시스템은 RFID리더(Reader)와 RFID태그(Tag), 그리고 리더와 연결되어 리더가 전송받은 정보를 처리하는 데이터 처리 장치로 구성되어 있다. RFID태그는 배터리가 있어 자체적인 연산 수행이 가능한 능동형 태그와 배터리가 없는 수동형 태그로 나눌 수 있다[2]. 능동형 태그는 수동형 태그에 비해 전송 범위가 길지만 가격이 비싸고, 작은 사이즈로 생산하기 어렵다는 단점이 있다. 그렇기 때문에 본 논문에서는 저렴한 가격으로 물류 관리 분야에 사용하기 적합한 수동형 태그로 가정한다.

RFID시스템은 다양한 분야에서 사용될 것이라고 기대하고 있지만, 몇 가지 문제들 때문에 아직까지는 극히 제한된 영역에만 사용되고 있다[3,4]. 본 논문에서는 위에서 말한 몇 가지 문제 중 다수의 태그를 식별할 때 발생하는 충돌 문제를 해결하는 충돌방지 알고리즘을 제시하고자

한다. 다중태그를 식별할 때 발생하는 충돌 문제란 리더가 여러 개의 태그정보를 읽기 위해 질의(query)를 전송했을 때 이 질의를 수신한 여러 태그들이 동시에 응답(response)하여 전파 사이에 충돌이 발생하게 되는 문제이다. 이 문제로 인해 리더는 태그들의 정보를 식별하는데 실패하게 된다. 이러한 충돌 문제가 자주 발생하면 태그의 인식 효율 및 속도에 큰 영향을 미치게 되므로 효율적인 충돌 방지 알고리즘이 필요하다[2].

다수의 태그를 식별할 때 발생하는 충돌 문제를 해결하기 위한 알고리즘은 확률적인(probabilistic)방식과 결정적인(deterministic)방식 두 가지로 나눌 수 있고, 이 두 방식을 혼합한 하이브리드(hybrid) 방식이 있다[2-6]. 확률적인 방식은 시간을 슬롯 단위로 나누고 태그들이 임의로 하나의 슬롯을 선택하여 응답하는 방식으로, 하나의 슬롯에 오직 하나의 태그가 응답을 할 때만 태그를 인식한다. 이 방식은 확률을 이용하였기 때문에 모든 태그를 인식하는 것을 보장하지 못한다. 반면에 결정적인 방식은 트리(tree)를 기반으로 하기 때문에 모든 태그를 식별하는 것을 보장하고 그 과정을 예측할 수 있다. 하지만 태그를

식별하기 위해 여러 번의 질의-응답 과정이 필요하다. 그러기 때문에 만약 유사한 ID를 갖는 태그들이 다수 존재할 경우 충돌이 빈번히 발생하여 모든 태그를 식별하는데 걸리는 시간이 길어진다.

기업체에서 제품을 생산할 때 태그ID는 EPC 태그 데이터 표준을 따른다. EPC코드에도 여러 종류가 있으나 본 논문에서는 가장 널리 알려진 96비트의 GID-96 인코딩 방식의 태그를 사용한다. GID-96 인코딩 방식은 헤더, 기업체코드, 상품코드, 일련번호로 구성된다[7]. 헤더는 태그정보의 길이와 어떤 인코딩 기법으로 코딩되어 있는지를 나타내고, 기업체코드는 회사, 관리자 또는 조직을 나타내는 숫자로 EPCGlobal에 의해서 정해진다. 그리고 나머지 상품코드와 일련번호는 업체에 의해 정해지게 되는데, 이것은 어떤 회사에서 만든 동일한 제품군의 제품들을 생산했다면, 그 제품들을 보관하고 있는 창고에는 헤더, 기업체코드, 상품 코드는 동일하고, 일련번호만 순차적으로 증가하는 수많은 태그들이 존재한다는 것을 의미한다.

그렇기 때문에 본 논문에서는 RFID시스템에서 유사한 일련번호로 이루어진 다수의 태그를 인식할 때 발생할 수 있는 충돌문제를 해결하는 트리기반의 결정적 방식 알고리즘을 제안한다. 이 알고리즘은  $m$ -비트인식, 태그ID의 충돌정보, 예측기법을 이용하여 질의-응답 개수를 효과적으로 줄여 전체 태그 인식 속도와 인식 효율을 높였다.

## 2. 관련 연구

### 2.1 알로하 기반 프로토콜

충돌 방지 알고리즘의 확률적인 방식인 알로하 기반 프로토콜은 리더가 시간을 슬롯(slot) 단위로 나누고 태그들이 각자 임의로 슬롯을 선택하여 자신의 ID를 전송하는 방식이다[2,8-10]. 이때 리더는 하나의 태그가 하나의 슬롯에 응답했을 때만 충돌 없이 태그를 인식한다.

#### 2.1.1 프레임드 슬롯티드 알로하

프레임드 슬롯티드 알로하 프로토콜(Framed Slotted ALOHA Protocol: FSA)은 하나의 프레임(frame)이 여러 개의 슬롯으로 구성되어 있다[11]. 태그는 리더로부터 전송 받은 프레임의 슬롯 중 하나를 임의로 선택하고, 자신의 ID를 전송한다. 이때 인식된 태그는 리더에게서 ACK 명령을 받고 다음 프레임에 참여하지 않는다. 한 프레임이

Downlink	Request <3>	1	2	3	ACK <1100>	Request <3>	1	2	3	ACK <0110>
Uplink		충돌	충돌	인식			충돌	인식	무응답	
Tag1.(0101)		○					○			
Tag2.(1101)		○					○			
Tag3.(1001)			○				○			
Tag4.(0110)			○					○		
Tag5.(1100)				○						

(그림 1) FSA의 태그인식 과정  
(Figure 1) Tag identification process of FSA

끝난 후, 인식하지 못한 태그를 위해 리더는 다음 프레임을 전송한다. (그림 1)은 FSA의 동작을 나타낸다.

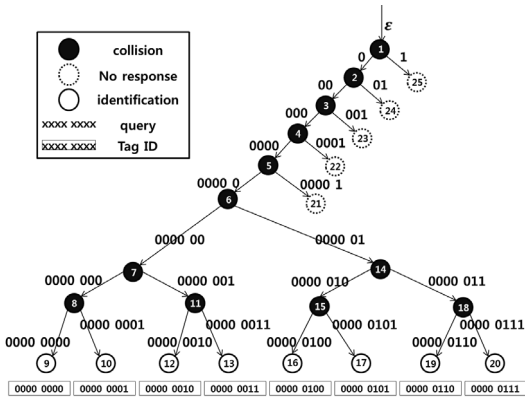
(그림 1)에서 하향링크(Downlink)와 상향링크(Uplink)는 각각 리더와 태그가 사용하는 채널이다. 하나의 프레임은 세 개의 슬롯으로 구성되어 있다. 리더는 전송범위 내에 있는 태그들을 인식하기 위해 태그ID를 요청하는 메시지를 보낸다. 이 요청 메시지를 받은 태그들은 3개의 슬롯 중 하나를 선택해 자신의 ID를 전송한다. 첫 번째 슬롯에는 Tag1과 Tag2가 동시에 응답해 충돌이 발생했다. 두 번째 슬롯에는 Tag3과 Tag4가 동시에 응답해 충돌이 발생했다. 세 번째 슬롯에는 Tag5만 유일하게 응답해 리더에 의해 인식된다. 인식된 태그는 리더로부터 ACK 명령을 받고 두 번째 프레임에서는 응답을 하지 않는다. 두 번째 프레임에서 나머지 태그를 인식하기 위해 리더는 태그ID를 요청하는 메시지를 보낸다. 첫 번째 슬롯에는 Tag1과 Tag2와 Tag3이 동시에 응답해 충돌이 발생했다. 두 번째 슬롯에는 Tag4만 유일하게 응답해 리더에 의해 인식 된다. 세 번째 슬롯에는 아무 태그도 응답을 하지 않았다. 인식된 태그는 리더로부터 ACK 명령을 받고 세 번째 프레임에서는 응답을 하지 않는다. 이 방식은 확률이 라는 불확실성을 기초로 하고 있기 때문에 리더의 전송 범위 내에 있는 태그를 모두 인식하지 못한다는 단점을 가지고 있다.

### 2.2 트리 기반 프로토콜

충돌 방지 알고리즘의 결정적인 방식인 트리 기반 프로토콜은 RFID 태그를 인식하는 과정을 트리로 나타낼 수 있다. 트리 기반의 프로토콜은 리더가 질의(query)하고 태그가 응답(response)하는 방식으로 진행된다[2,12,13]. 트리 기반의 충돌 방지 프로토콜에서 트리를 구성하는 각 노드들은 하나의 질의-응답 과정에 해당한다.

2.2.1 쿼리 트리 프로토콜

쿼리 트리 프로토콜(Query Tree Protocol : QT)은 리더의 전송 범위 내에 있는 태그에게 ID를 요청할 때 k비트의 비트 스트링(string)으로 이루어진 질의를 질의-큐에서 꺼내 태그들에게 전송한다[14]. 질의를 수신한 태그는 질의가 자신의 ID 앞부분과 같은지 확인하고 일치하면 리더에게 응답한다. 이때 리더에 응답한 태그가 둘 이상이라면 리더는 충돌이 발생한 것을 알고 이전 질의에 '0'과 '1'을 붙여 2개의 새로운 k+1비트의 질의를 만들어 질의-큐에 넣는다. 질의-큐에 넣은 질의를 꺼내 리더는 다시 태그들에게 질의-응답 과정을 반복한다. 처음 질의-큐는  $\epsilon$ 으로 초기화 되어 있고,  $\epsilon$ 을 수신한 모든 태그들은 자신의 ID를 응답한다.



(그림 2) QT의 태그인식 과정  
(Figure 2) Tag identification process of QT

(그림 2)는 QT의 태그ID가 '0000 0000'에서 '0000 0111'으로 순차적으로 증가하는 8개의 태그의 식별 과정을 표현한 트리이다. 식별 과정에서 트리의 노드는 각각 충돌발생, 태그식별, 무응답의 세 가지 경우로 나눌 수 있다. 충돌발생은 동시에 두 개 이상의 태그가 응답했을 때이고, 태그식별의 경우는 질의에 대하여 하나의 태그만 응답했을 경우이다. 무응답은 질의에 대하여 아무런 응답이 없을 경우로 유험 노드라고 한다. (그림 2)에서 각 노드의 숫자는 질의-응답 순서를 나타내고 가지위에 있는 숫자는 질의를 나타낸다. 이 경우 QT의 질의-응답 수는 총 25번이다

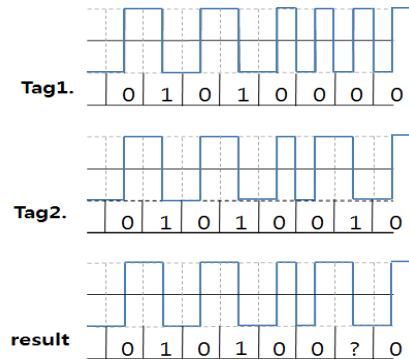
QT은 유험 노드가 발생하고, 한 번에 하나의 태그만 식별할 수 있다. 그리고 태그ID가 길고, 유사한 태그 집단으로 이루어져 있다면 트리의 깊이는 더욱 깊어지게 되

어, 인식 속도가 저하된다.

2.2.2 맨체스터 코딩

QT의 단점은 유사한 다수 태그ID들을 인식하려 할 때, 트리의 깊이가 깊어져 인식하는데 상당한 시간이 걸린다는 것이다. 이 문제점을 해결하기 위해서는 리더에 두 개 이상의 태그가 동시에 대답했을 때, 리더에서 응답한 태그들의 ID를 분석하여 충돌정보를 알아낼 필요가 있다.

본 논문에서는 충돌정보를 알아내기 위해 디지털 신호를 인코딩 방식중 하나인 맨체스터 코딩을 이용한다[1]. 맨체스터 코딩 방식은 하나의 비트가 전압의 값(value)이 아니라 전압의 전이(transition)으로 표현된다. '0' 비트와 '1' 비트는 각각 양의 전이(positive transition)와 음의 전이(negative transition)으로 표현된다. 즉, 두 개 이상의 태그들이 동시에 응답했을 때, 양의 전이와 음의 전이를 비교하면 서로 상쇄되어 맨체스터 코딩에서 발생할 수 없는 무전이(no transition) 상태가 되어 여러가 발생한 것을 알 수 있다. (그림 3)은 태그ID '0101 0000'와 '0101 0010'을 리더가 수신 했을 때, 맨체스터 코딩을 이용하여 어떤 비트에 충돌이 발생했는지를 알 수 있다.

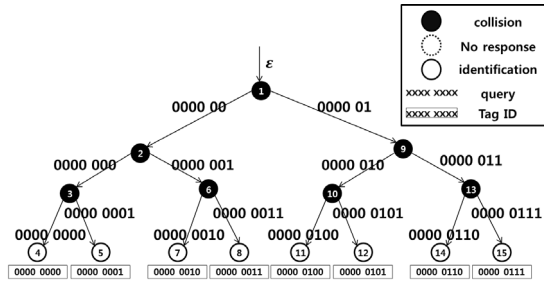


(그림 3) 맨체스터 코딩의 예  
(Figure 3) An example of manchester coding

2.2.3 충돌 트리 프로토콜

충돌 트리 프로토콜(Collision Tree Protocol : CT)은 QT를 기반으로 한다. CT는 맨체스터 코딩 기법을 이용하여 리더에서 응답한 태그ID를 분석하고, 충돌이 있다면 최초로 충돌이 발생한 비트를 알아낸다[15]. 리더는 그 정보를 가지고 다음 질의를 생성함으로써 불필요한 질의-응답의 개수를 효과적으로 줄일 수 있다.

(그림 4)는 CT의 태그ID가 '0000 0000'에서 '0000 0111'으로 순차적으로 증가하는 8개의 태그의 식별 과정을 표현한 트리이다. 위 과정에서 리더의 질의를 받은 다수의 태그들은 전송받은 질의를 제외한 자신의 태그ID를 전송하고, 리더는 맨체스터 코딩을 이용하여 충돌정보를 분석한다. 이렇게 두 개 이상의 태그로부터 리더가 수신한 응답은 '0'과 '1', 충돌을 의미하는 '\*' 이렇게 세 가지로 나타낼 수 있다. 리더는 전송받은 태그ID의 정보를 이용해 처음 비트부터 최초로 충돌이 발생한 위치에 '0'과 '1'을 질의에 추가하여 2개의 질의를 만들어 질의-큐에 넣는다. 질의-큐에 넣은 질의를 꺼내 리더는 다시 태그들에게 질의-응답 과정을 반복한다. 처음 질의-큐는  $\epsilon$ 으로 초기화 되어 있고,  $\epsilon$ 을 수신한 모든 태그들은 자신의 ID를 응답한다.



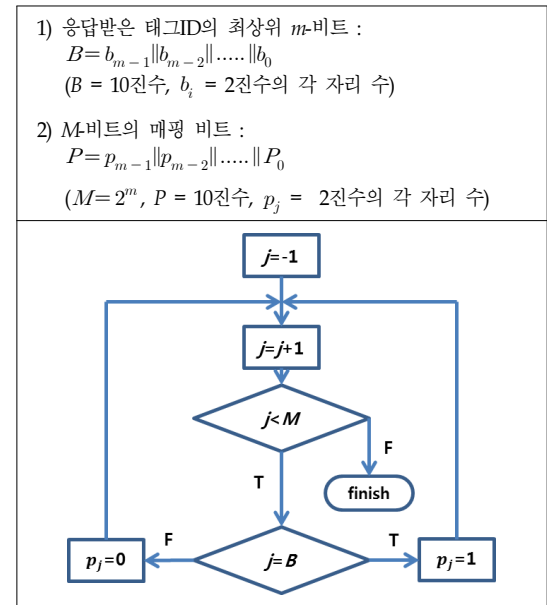
(그림 4) CT의 태그인식 과정  
(Figure 4) Tag identification process of CT

(그림 4)에서 알 수 있듯이 충돌정보를 이용해 질의를 만들었기 때문에 QT에 비해 질의-응답 수가 줄어든 것을 알 수 있다. 이 경우 CT의 질의-응답 수는 총 15번이다.

### 2.2.4 $M$ -ary 쿼리 트리 프로토콜

$M$ -ary 쿼리 트리 프로토콜( $M$ -ary Query Tree Protocol : MQT)도 QT을 기반으로 한다. 하지만 질의-응답 과정에 하나의 비트만 식별하는 1-비트 인식 방법을 사용하는 QT과는 달리, MQT는 맨체스터 코딩 기법과 매핑 함수(mapping function)을 이용하여  $m$ -비트 인식 방법을 제공한다( $m = \log_2 M$ )[16]. MQT에서 리더는 범위 내에 있는 모든 태그에게 질의를 전송한다. 이때 태그는 리더가 전송한 질의에 해당하는 자신의 ID를 제외하고 응답한다. 태그가 자신의 ID를 응답할 때, ID의 최상위의  $m$ -비트를 매핑 함수(mapping function)를 이용하여  $M$ -비트의 매핑 비트(mapping bit)로 변환하여 전송한다. 태그들의 응답을

받은 리더는 응답 메시지의 최상위로부터  $M$ -비트를 가지고 충돌 여부와 충돌정보를 분석한다. 충돌정보를 분석한 후  $M$ -비트의 매핑 비트를  $m$ -비트의 원래 태그ID로 역변환한다. 이 과정에서 충돌이 없었다면 태그는 하나의 태그를 인식하게 되고, 그렇지 않다면 역변환된  $m$ -비트를 이전 질의에 붙여 질의-큐에  $M$ -비트 매핑 비트 내의 충돌 개수만큼 질의를 추가하게 된다. (그림 5)는 매핑 함수의 알고리즘이다.

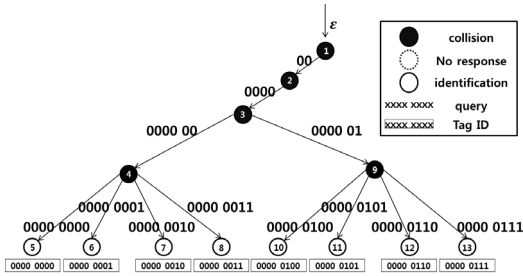


(그림 5) 매핑 함수 알고리즘  
(Figure 5) An algorithm of mapping function

MQT는  $m$ -비트 인식 방식을 사용하기 때문에 충돌 노드의 개수를 줄일 수 있고, 매핑 비트를 분석한 정보로 질의를 생성하기 때문에 유휴 노드도 제거한다.

(그림 6)는 MQT의 태그ID가 '0000 0000'에서 '0000 0111'으로 순차적으로 증가하는 8개의 태그의 식별 과정을 표현한 트리이다 ( $m=2, M=4$ ). 이 경우 MQT의 질의-응답 수는 총 13번이다.

지금까지 알로하 기반 프로토콜과 트리 기반 프로토콜에 대해 알아보았다. 알로하 기반 프로토콜은 확률을 기반으로 하기 때문에 모든 태그를 인식하는 것을 보장하지 못하지만, 트리 기반의 프로토콜은 모든 태그를 인식하는 것을 보장하지만 프로토콜의 종류에 따라 그 효율이 다르다.



(그림 6) MQT의 태그 인식 과정  
(Figure 6) Tag identification process of MQT

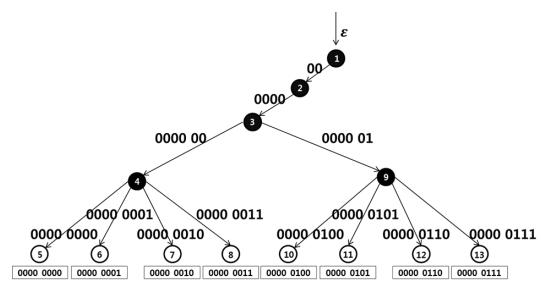
### 3. 제안하는 기법

#### 3.1 적응형 $M$ -ary 쿼리 트리 프로토콜

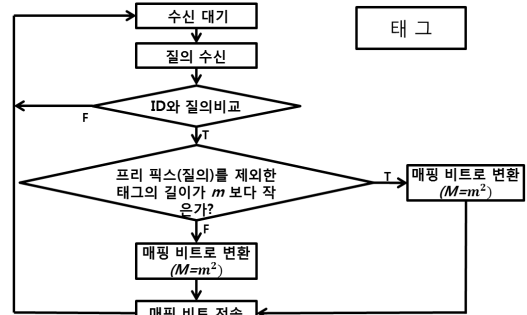
본 절에서는 트리 기반 프로토콜에서 유사한 태그ID가 모여 있는 환경에서 태그들을 식별할 때 질의의 길이가 고정된 비트씩 증가할 경우, 비효율적으로 동작하는 인식 과정을 개선하기 위한 적응형 RFID 충돌 방지 기법인 적응형  $M$ -ary 쿼리 트리 프로토콜(Adaptive  $M$ -ary Query Tree Protocol : AMQT)을 제안한다. 이 알고리즘은 리더에 응답한 태그들의 ID를 분석하여 얻은 최초의 충돌 비트 정보를 이용하여 다음 질의를 생성하고, MQT의  $m$ -비트 인식 그리고 예측 기법을 사용한다. 그렇기 때문에 이 알고리즘은 유사한 태그ID를 가진 다수의 태그를 인식할 때 식별트리(identification tree)에서 충돌노드(collision node)를 줄이고, 유향노드(idle node)와 확인노드(identification node)를 제거하여 식별시간을 크게 감소하였다. 그리고 예측 기법을 사용하여 트리의 깊이(depth)를 하나 감소시킴으로써 최대  $1/M$ 만큼의 질의-응답 개수를 추가적으로 줄였다. 예측 기법이란, 응답한 태그ID가 매핑 함수에 의해 변환되어 있을 때, 응답한 태그ID가  $M$ -비트 이하일 경우, 매핑 파트에 있는 충돌 개수를 태그의 개수로 판단하고, 매핑 파트를 분석하여 여러 개의 태그를 한 번에 인식하는 기법이다.

(그림 7)은 위에 (그림 5)의 MQT 태그 인식 과정 트리에 예측기법을 적용한 예이다. 예측기법을 적용하여 마지막에 1-비트씩 인식하는 확인 노드들을 제거하고 4-비트씩 인식하도록 하였다. (그림 8)과 (그림 9)은 각각 AMQT의 태그와 리더의 알고리즘이다. 여기서 사용되는 매핑 함수는 MQT의 매핑 함수를 사용한다.

(그림 10)은 AMQT의 태그ID가 '0000 0000'에서 '0000 0111'으로 순차적으로 증가하는 8개의 태그의 식별 과정

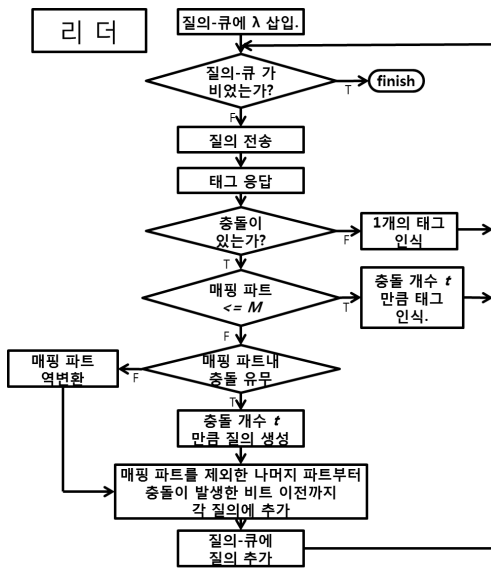


(그림 7) 예측기법을 적용한 MQT의 예  
(Figure 7) An example of MQT applied to expectation method



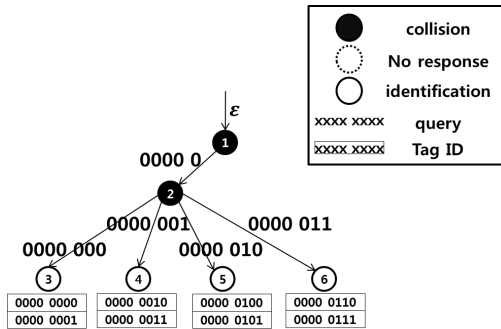
(그림 8) AMQT 태그 알고리즘  
(Figure 8) An algorithm of AMQT tag

을 표현한 트리이다( $m=2, M=4$ ). 태그들이 식별되는 과정을 살펴보면, 처음 질의-큐에는  $\epsilon$ 로 초기화된 질의가 있다. 리더는 질의-큐에서  $\epsilon$  질의를 꺼내어 범위 내에 있는 모든 태그들 Tag1(0000 0000), Tag2(0000 0001), Tag3(0000 0010), Tag4(0000 0011), Tag5(0000 0100), Tag6(0000 0101), Tag7(0000 0110), Tag8(0000 0111)에게 전송한다.  $\epsilon$ 를 수신한 여덟 개의 태그들은 최상위비트로부터 2-비트를 매핑 함수를 이용하여 매핑 비트로 변환하여 리더에게 전송한다. 매핑 비트로 변환한 메시지는 다음과 같고



(그림 9) AMQT 리더 알고리즘

(Figure 9) An algorithm of AMQT reader



(그림 10) AMQT의 태그 인식 과정

(Figure 10) Tag identification process of AMQT

' '는 매핑 파트를 의미한다. Tag1('0001'00 0000), Tag2('0001'00 0001), Tag3('0001'00 0010), Tag4('0001'00 0011), Tag5('0001'00 0100), Tag6('0001'00 0101), Tag7('0001'00 0110), Tag8('0001'00 0111). 리더는 여덟 개의 태그로부터 '0001'00 0\*\*\*'의 응답 메시지를 받게 된다. 이때 '\*'은 충돌을 의미한다. 리더는 매핑 파트 내에 충돌이 없는 것을 확인하고 매핑 파트를 원래의 2-비트인 00으로 변환하여 질의를 만든다. 그리고 매핑 파트를 제외한 나머지 부분의 최상위비트부터 최초로 충돌이 발생한 이전까지 비트(000)를 질의에 붙인다. 이렇게 만들어진

0000 0이라는 질의를 질의 큐에 넣고, 리더는 다시 질의를 태그들에게 전송한다. 질의 0000 0을 수신한 태그들은 자신의 ID 앞부분과 일치하는지 비교한 후 일치한다면, 자신의 ID 앞부분에 0000 0을 제외한 나머지 부분에서 최상위 비트로부터 2-비트를 매핑 비트로 변환하여 리더에게 전송한다. 매핑 비트로 변환한 메시지는 다음과 같다. Tag1('0001'0), Tag2('0001'1), Tag3('0010'0), Tag4('0010'1), Tag5('0100'0), Tag6('0100'1), Tag7('1000'0), Tag8('1000'1). 리더는 여덟 개의 태그로부터 '\*\*\*\*'\*의 응답 메시지를 받게 된다. 리더는 매핑 파트 내에 충돌이 있는 것을 확인하고 충돌 위치에 해당하는 0001, 0010, 0100, 1000을 원래의 2-비트인 00, 01, 10, 11으로 변환하여 질의를 만든다. 그리고 매핑 파트를 제외한 나머지 부분의 최상위비트에서 충돌이 발생했으므로 더 이상의 비트를 추가하지 않고 이전 질의에 00, 01, 10, 11을 붙여 4개의 질의 0000 000, 0000 001, 0000 010, 0000 011을 만들어 질의-큐에 넣고, 리더는 다시 질의를 태그들에게 전송한다. 질의 0000 000을 수신한 태그들은 자신의 ID 앞부분과 일치하는지 비교한다. ID 앞부분이 일치하는 Tag1, Tag2은 0000 000을 제외한 나머지 부분에서 최상위비트로부터 1-비트를 매핑 비트로 변환하여 리더에게 전송한다. 매핑 비트로 변환한 메시지는 Tag1('01')와 Tag2('10')이다. 리더는 두 개의 태그로부터 '\*\*'의 응답 메시지를 받게 된다. 리더는 응답 메시지의 길이가 4-비트 이하인 것을 확인하고 충돌이 발생한 개수인 두 개의 태그를 인식하게 된다. 이 때 인식할 태그ID는 매핑 파트의 충돌 위치에 해당하는 01, 10을 원래의 1-비트인 0, 1로 변환해 질의에 붙여 인식하게 된다. 두 개의 태그를 인식한 리더는 질의-큐의 상태를 확인한 후 질의가 남아 있다면 다시 질의를 태그들에게 전송한다. 질의 0000 001을 수신한 태그들은 자신의 ID 앞부분과 일치하는지 비교한다. ID 앞부분이 일치하는 Tag3, Tag4은 0000 001을 제외한 나머지 부분에서 최상위 비트로부터 1-비트를 매핑 비트로 변환하여 리더에게 전송한다. 매핑 비트로 변환한 메시지는 Tag1('01')와 Tag2('10')이다. 리더는 두 개의 태그로부터 '\*\*'의 응답 메시지를 받게 된다. 리더는 응답 메시지의 길이가 4-비트 이하인 것을 확인하고 충돌이 발생한 개수인 두 개의 태그를 인식하게 된다. 이 때 인식할 태그ID는 매핑 파트의 충돌 위치에 해당하는 01, 10을 원래의 1-비트인 0, 1로 변환해 질의에 붙여 인식하게 된다. 두 개의 태그를 인식한 리더는 질의-큐의 상태를 확인한 후 질의가 남아 있다면 다시 질의를 태그들에게 전송한다. 질의 0000 010을 수신한 태그들은 자신의 ID 앞부분과 일치하는지 비교한

다. ID 앞부분이 일치하는 Tag5, Tag6은 0000 010을 제외한 나머지 부분에서 최상위비트로부터 1-비트를 매핑 비트로 변환하여 리더에게 전송한다. 매핑 비트로 변환한 메시지는 Tag1('01')와 Tag2('10')이다. 리더는 두 개의 태그로부터 '\*\*'의 응답 메시지를 받게 된다. 리더는 응답 메시지의 길이가 4-비트 이하인 것을 확인하고 충돌이 발생한 개수인 두 개의 태그를 인식하게 된다. 이때 인식할 태그ID는 매핑 파트의 충돌 위치에 해당하는 01, 10을 원래의 1-비트인 0, 1로 변환해 질의에 붙여 인식하게 된다. 두 개의 태그를 인식한 리더는 질의-큐의 상태를 확인한 후 질의가 남아 있다면 다시 질의를 태그들에게 전송한다. 질의 0000 011을 수신한 태그들은 자신의 ID 앞부분과 일치하는지 비교한다. ID 앞부분이 일치하는 Tag7, Tag8은 0000 011을 제외한 나머지 부분에서 최상위비트로부터 1-비트를 매핑 비트로 변환하여 리더에게 전송한다. 매핑 비트로 변환한 메시지는 Tag1('01')와 Tag2('10')이다. 리더는 두 개의 태그로부터 '\*\*'의 응답 메시지를 받게 된다. 리더는 응답 메시지의 길이가 4-비트 이하인 것을 확인하고 충돌이 발생한 개수인 두 개의 태그를 인식하게 된다. 이때 인식할 태그ID는 매핑 파트의 충돌 위치에 해당하는 01, 10을 원래의 1-비트인 0, 1로 변환해 질의에 붙여 인식하게 된다. 두 개의 태그를 인식한 리더는 질의-큐의 상태를 확인한 후 질의가 남아있지 않은 것을 확인하고 종료한다. 이 경우 AMQT의 질의-응답 수는 총 6번이다.

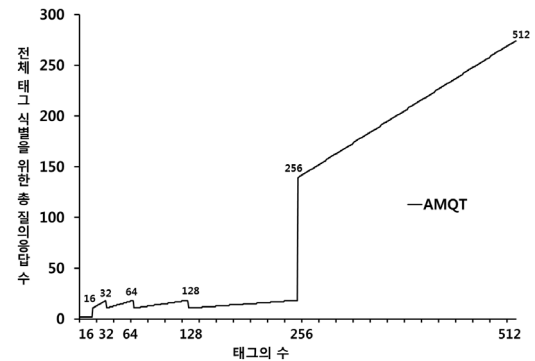
#### 4. 실험 및 고찰

본 논문에서 제안하는 AMQT 성능 평가를 위해 유사한 태그ID 집단을 인식하는 실험을 통해 QT, CT, MQT와 성능 비교를 하였다. 실험은 컴퓨터 시뮬레이션을 이용하였으며, 에러율을 적용하지 않은 이상적인 통신 채널 환경을 고려하여 수행 하였다. (표 1)은 태그ID가 '0000 0000'에서 '0000 0111'으로 순차적으로 증가하는 8개의 태그를 식별하기 위한 총 질의-응답 수이다. 트리 기반의 프로토콜을 이용하여 8개의 태그를 인식했을 때, QT는 25번, CT는 15번, MQT는 13번, AMQT는 6번으로 AMQT의 성능이 QT에 비해 4배 이상 CT, MQT에 비해 2배 이상 증가했다.

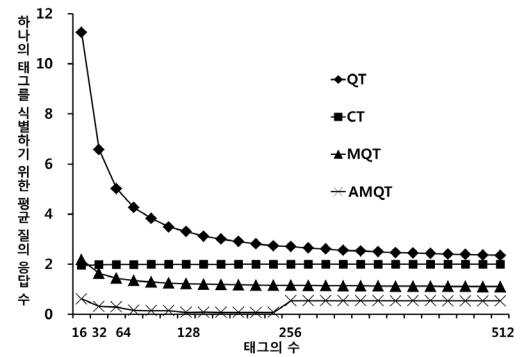
더 자세한 성능 평가를 위해 96-비트의 태그ID를 '0000...0000'부터 순차적으로 증가시켜 태그가 2개일 때부터 512개일 때 까지 태그를 인식하는 실험을 하였다. 여기서 실험의 공정성을 더하기 위해 QT는 기존 방식이

(표 1) 8개의 8비트 태그 인식 결과  
(Table 1) Eight 8-bit tag identify results

트리기반 프로토콜	총 질의-응답 수	인식 효율
AMQT	6	× 4.16
QT	25	× 1
CT	15	× 1.66
MQT	13	× 1.92



(그림 11) AMQT의 태그 식별을 위한 총 질의-응답 수  
(Figure 11) Total number of query and response for AMQT tag identification



(그림 12) QT기반 프로토콜의 성능 비교  
(Figure 12) performance comparison of protocol based on QT

아닌, 리더의 질의에 일치하는 부분을 제외한 자신의 ID를 전송하는 방식으로 수정하여 실험 하였다. (그림 11)은 AMQT의 전체 태그를 식별하는데 걸리는 총 질의-응답 개수이다. 여기서 MQT와 제안하는 시스템의 m은 4로 설정하였다. AMQT의 질의-응답 개수는 계단형으로 증가



하는 것을 볼 수 있다. 그 이유는 한 번에 인식할 수 있는 태그의 수가  $m^2$ 이기 때문이다. 즉,  $m=4$ 일 때, 질의-응답 개수는 태그가 0부터  $2^4$ 개일 때 까지는 질의-응답 수는 1번으로 일정하고,  $2^{n-1}+1$  부터  $2^n$ 개일 때까지는 일정하게 증가한다( $n \geq m$ ). 하지만 (그림 12)처럼 QT기반의 다른 태그들과의 성능을 비교해 보면 여전히 다른 기법에 비해 우수한 성능을 보이는 것을 알 수 있다.

## 5. 결 론

본 논문은 RFID 시스템에서 효율적으로 태그를 인식하기 위한 AMQT를 제안하였다. 이 알고리즘은 맨체스터 코딩을 이용하여 질의에 응답한 태그ID를 분석하여 최초의 충돌 비트 위치인 충돌 정보를 이용하여 불필요한 질의-응답 수를 효과적으로 줄이고, 맨체스터 코딩과 매핑 함수를 이용한  $m$ -비트인식을 사용하여 충돌 노드의 수를 줄였다. 그리고 예측기법을 이용하여 확인 노드를 제거하여 유사한 태그ID들의 집합에서 기존의 트리기반의 다른 기법들 보다 빠르게 인식한다.

## 참 고 문 헌(Reference)

[1] K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification", John Wiley & Sons, pp.206-219, 2003.

[2] 김수철, 여상수, 김성권, "수동형 RFID 태그 충돌 방지 알고리즘에 관한 비교 연구", 한국정보기술 융합학회논문지Vol.3 No2 [2010], pp.163-177.

[3] Daniel W. Engels, "The Reader Collision Problem", Auto-ID Center, November 2001.

[4] 김태훈, 여상수, 신재동, 광진, 방효창, 조성연, "RFID 태그 충돌 방지 기법들에 대한 성능 비교", Journal of the Korean data analysis society, 9 권, 6 호, pp. 3139-3149, 2007.12

[5] 김정환, 김영태, 박용수, 안광선, "RFID 시스템에서 비트변화감지를 이용한 하이브리드 충돌 방지 프로토콜", 인터넷정보학회논문지 제10권 제2호, 133-141, 2009.4

[6] 정승민, 조정식, 김성권, "FQTR: RFID 시스템을

위한 새로운 하이브리드 태그 충돌 방지 프로토콜", 정보과학회 논문지, vol. 36, no. 7, pp. 560-570, 2009.7

[7] "EPC Tag Data Standard 1.6", EPCglobal, 2011.

[8] H. Vogt, "Efficient object identification with passive RFID tags", in Proc. Int. Conf. Pervasive Comput., pp. 98-113, Zurich, Switzerland, Aug. 2002.

[9] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Instant collision resolution for tag identification in RFID networks", Ad Hoc Networks, vol. 5, no. 8, pp. 1220-1232, Nov. 2007.

[10] G. Bagnato, G. Maselli, C. Etrioli, and C. Vicari, "Performance analysis of anti-collision protocols for RFID systems", in Proc. IEEE 69th Veh. Technol. Conf., pp.1-5, Barcelona, Spain, Apr. 2009.

[11] Jae-Ryong Cha and Jae-Hyun Kim, "Dynamic Framed Slotted ALOHA Algorithms using Fast Tag Estimation Method for RFID System", Consumer C-ommunications and Networking Conference, IEEE, vol. 2, January 2006, pp. 768-772.

[12] J. Myung, W. Lee, J. Srivastava, and T. K. Shih, "Tag-splitting: adaptive collision arbitration protocols for RFID tag identification", IEEE Trans. Parallel Distributed Syst., vol. 18, no. 6, pp. 763-775, June 2007.

[13] J. H. Choi, D. Lee, and H. Lee, "Query tree-based reservation for efficient RFID tag anti-collision", IEEE Commun. Lett., vol. 11, no. 1, pp. 85-87, Jan. 2007.

[14] C. Law, K. Lee, and K. Y. Siu, "Efficient Memoryless Protocol for Tag identification", Proc. ACM DIALM '00, 2000.

[15] Xiaolin Jia, Quanyuan Feng, Chengzhen Ma. "An Efficient Anti-Collision Protocol for RFID Tag Identification", IEEE Communications Letters, Vol. 14 Issue. 11 Nov 2010, pp.1014-1016.

[16] Jongmin Shin, Dongmin Yang, Checha Kim, "Multiple RFID tags Identification with M-ary Query tree search scheme", IEEE Communications Letters, Vol. 17 NO. 3 Mar 2013, pp.604-607.

● 저 자 소개 ●



**이 제 율(Je-yul Lee)**

2013년 대전대학교 정보통신공학과(공학사)  
2013년~현재 대전대학교 대학원 정보통신공학과 석사 과정  
관심분야 : 무선이동통신  
E-mail : jylee@pine.dju.ac.kr



**신 종 민(Jongmin Shin)**

2003년 홍익대학교 컴퓨터공학과(공학사)  
2005년 POSTECH 컴퓨터공학과(공학석사)  
2011년 POSTECH 컴퓨터공학과(공학박사)  
2011년~현재 삼성전자 VD사업부 책임연구원  
관심분야 : 무선이동통신, 유비쿼터스 컴퓨팅  
E-mail : jm0621.shin@samsung.com



**양 동 민(Dongmin Yang)**

2000년 POSTECH 컴퓨터공학과(공학사)  
2003년 POSTECH 컴퓨터공학과(공학석사)  
2011년 POSTECH 컴퓨터공학과(공학박사)  
2011년~현재 대전대학교 정보통신공학과 교수  
관심분야 : 무선이동통신, 인지 라디오 네트워크  
E-mail : dmyang@dju.kr