

# RFID 시스템 인식속도 개선을 위한 충돌방지 알고리즘

김익순\* · 김천석\*\*

Anti-Collision Algorithm for High-Speed Tags in Active RFID System

Ik-Soon Kim\* · Chun-Suk Kim\*\*

요 약

RFID 시스템에서 해결하여야 할 가장 큰 문제 중 하나는 태그간의 충돌로 인해 인식 효율이 떨어진다는 것이다. 기존의 충돌 방식 중 BS 충돌 방지 알고리즘 방식은 구현이 간단하다는 장점이 있으나, 태그 수가 많아짐에 따라서 태그간 충돌에 의해 전체 태그를 읽는데 요구되는 단계별 스케줄 횟수가 증가하는 단점을 가지고 있다. 본 논문에서는 이 문제를 해결하기 위해서 충돌 예상을 최소화하기 위해 사전에 '0'과 '1'의 결정 비트를 통한 복수 응답 프레임 방식을 제안하였다.

ABSTRACT

In RFID System, one of the problem that we must solve is to devise a good anti-collision algorithms to improve the efficiency of tag identification which is usually low because of tag collision. Among of the existing RFID anti-collision algorithm, BS (Binary Search) algorithm, though simple, has a disadvantage that the stage of times used to identify the tags increase exponentially as the number of tags does. In this Paper, I propose a new anti-collision algorithm called Multi-collision reflected frame which restricts the number of stages and decided bit.

Since the proposal algorithm keep the length size of UID and density of total tag when have 100%.

키워드

Active RFID, Collision, High speed Tags  
능동형 RFID, 충돌, 고속태그

## 1. 서 론

최근 비약적인 디지털 기술과 인터넷 기술의 발달은 어느 누구에게나 손쉽게 정보를 창출하고 이용할 수 있도록 많은 변화를 초래 했다.

IP(Internet Protocol) network의 발달은 인터넷을

실시간 통신의 근간으로 삼을 만큼 충분히 이루어졌고 트래픽을 처리하는 주요수단으로 각 분야에서 활용하기에 이르렀다. 이를 바탕으로 사물의 자동 인식 관련 기술의 하나인 RFID의 보편적인 실생활 적용이 이루어지고 있다.

RFID 시스템은 태그에 전원공급을 위한 배터리의

\* 교신저자(corresponding author) : 수원지방검찰청 정보통신팀(trnftk05@naver.com)

\*\* 전남대학교 전자통신공학과 교수(kim1000s@chonnam.ac.kr)

접수일자 : 2013. 09. 23

심사(수정)일자 : 2013. 11. 25

게재확정일자 : 2013. 12. 16

장착 유무에 따라 크게 수동형 RFID 시스템과 능동형 RFID 시스템으로 구분된다. 수동형 RFID 시스템에서 태그는 자체 전원을 구비하지 않고 리더로부터 수신한 무선 주파수의 일부를 자신의 동작을 위한 전원으로 환원하여(Back-scattering) 응답을 전송하는 방식을 이용한다. 수동형 RFID 태그는 최소한의 IC 칩과 안테나만으로 구성되어 태그 단가가 매우 낮고 초박형으로 제작이 가능하다는 장점이 있으며, 이로 인해 수동형 RFID 시스템은 개별 물품 단위로 태그를 부착하고 관리하기 위한 목적을 가진 매우 다양한 응용 분야에 적용이 가능하다. 반면에 수동형 RFID 태그는 자체 전원을 사용하지 않기 때문에 리더에게 전송하는 응답 신호가 매우 미약하여서 리더가 인식할 수 있는 거리가 수 Cm에서 수 Meter 이내로 짧고 인식률이 떨어지며 전파를 흡수하는 금속 물체에 적용이 어렵다는 단점이 있다[1]

이에 반해 능동형 RFID 시스템에서의 태그는 자체적으로 배터리를 내장하고 있으며 이를 이용하여 응답을 전송함으로써 인식거리가 수십 Meter에서 수백 Meter로 매우 길고 금속 물체에 적용이 용이하며 Sensing 기능의 추가가 용이한 장점을 가지고 있으나, 배터리 구비로 인해 태그 단가가 높으며 일정 주기로 배터리의 충전 혹은 교체가 필요하다는 단점이 있다. 능동형 RFID 시스템은 그 특성상 넓은 항만 지역에서 대량의 금속 컨테이너들을 대상으로 하는 항만 물류관리 시스템에 가장 널리 적용 된다. 이러한 특수한 통신 방식 때문에 RFID 시스템에서는 다양한 문제가 발생한다. 가장 심각한 문제 중의 하나로 “충돌 문제”를 들 수 있다. “충돌 문제”는 크게 “태그 충돌 문제”, “리더-리더 충돌 문제”, “다중 리더-태그 충돌 문제” 등의 3가지로 나눌 수 있다.

첫째로 리더가 명령을 브로드캐스팅 했을 때, 만약 하나의 태그만이 응답을 전송하게 되면 리더는 그 태그를 인식하게 되지만, 2개 이상의 태그가 동시에 응답하게 되면, 무선 채널 상에서 충돌이 발생하여 리더가 이를 인식할 수 없게 된다. 이러한 문제를 “태그 충돌 문제”라고 정의 한다.

둘째로 2개 이상의 리더가 서로의 간섭 거리 내에 있고 동시에 같은 주파수를 사용하여 통신을 시도할 때 발생하는 문제를 “리더-리더 충돌 문제”라고 정의 한다. “리더-리더 충돌 문제”는 “주파수 간섭 문제”라

고도 한다.

세 번째 충돌 문제는 다수의 리더가 동시에 하나의 태그와 통신을 시도할 때 발생 하는 문제로 “다중 리더-태그 충돌 문제”라고 한다. 이러한 충돌 문제들을 해결 하는 능력은 RFID 시스템의 성능을 결정하는 중요한 요소 중의 하나이다[2][3].

본 논문에서는 현재 RFID 시스템에서 해결하여야 할 가장 큰 문제 중 하나인 태그간의 충돌로 인해 인식 효율이 떨어지는 문제점을 개선하고자 한다.

## II. 충돌방지 알고리즘

### 2.1 개요

RFID 시스템에서 판독기와 다수의 태그간의 하나의 동일 채널을 이용하기 때문에 동시에 전송되는 각각의 태그들의 신호가 서로 충돌을 일으키게 되며 이러한 충돌로 인하여 RFID 시스템의 성능을 저하시키게 된다. ISO/IEC 18000-6 Type A에서 사용되는 대표적인 충돌방지 알고리즘은 ALOHA 기법과 ALOHA 기법의 비교적 낮은 처리량을 최적화하기 위한 기법인 슬롯 ALOHA 기법이 있으며 ISO/IEC 18000-6 Type B에서는 이진검색 알고리즘과 이진트리 알고리즘이 있다. 이진검색 알고리즘에는 기본 이진검색 알고리즘과 기본 이진검색 알고리즘을 개선한 동적 이진검색 알고리즘이 있다. 이진 검색알고리즘은 충돌이 발생하면 충돌이 발생한 위치를 표시하고 충돌 최상위 비트를 0으로 나머지 비트를 1로 변환하여 그 이하의 ID를 갖는 태그를 호출하여 모든 태그를 인식하게 된다. 동적이진검색 알고리즘은 충돌이 발생한 비트 위치를 VB(Valid Bit)에 추가하고 나머지 비트를 전송하게 되어 이진검색 알고리즘에 비해 데이터 전송량을 줄일 수 있다. 이진트리 알고리즘은 충돌이 발생한 최상위 비트부터 0과 1의 두 그룹으로 선별하여 태그의 고유번호를 인식하는 기술로 가장 기본적으로 사용되는 이진트리 알고리즘에는 슬롯 이진트리 알고리즘과 비트별 이진트리 알고리즘이 있다.

리더가 읽은 태그의 정보는 RFID 시스템 전반에서 식별 작업을 하는데 중요한 영향을 미치는 기본적인 물품 정보로써 차세대 ID 시스템에서 가장 높은 신뢰성을 가져야 될 정보이다. 하지만 무선 RF 방식으로

태그 정보가 전송되기 때문에 여러 전송 오류 상황들이 많이 생겨날 수 있다. 이러한 오류 현상들은 리더 충돌 현상(Reader Collision)에 따른 오류와 태그 충돌(Tag Collision)에 의한 오류로 분류된다.

### 2.1.1 리더 충돌 현상(Reader collision)

그림 1과 같이 동일한 전송 인터페이스를 가진 RFID 태그 칩들로 인해 리더들 사이에 간섭 현상이 일어나서, 리더가 제대로 태그를 식별하지 못하는 현상을 리더 충돌 현상이라 한다. 리더 충돌 현상은 리더와 리더 사이의 간격이 적절치 못하고, 대상 태그들이 모두 같은 무선 주파수 대역을 사용하여 식별요청을 리더에 요청하게 되는 데에서 일어난다.

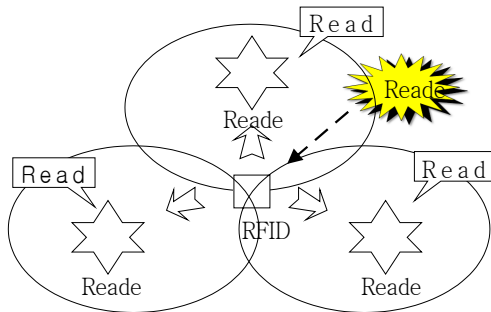


그림 1. 리더 충돌 현상  
Fig. 1 Reader collision

### 2.1.2 태그 충돌 현상(Tag collision)

일반적으로 RFID 시스템은 하나의 리더와 다수의 태그들로 이루어져 다수의 태그들이 동시에 리더에 접속하는 다중접속이 이루어진다. 그러므로 다수의 태그가 같은 처리 공간에 있을 때 태그들을 여러 없이 정확하게 인식하는 문제가 아주 중요하다. RFID 시스템에서 리더는 무선 채널을 통하여 각각의 태그들과 통신을 하는데, 그림 2와 같이 모든 태그들이 리더가 보낸 신호를 동시에 듣게 되고 리더의 명령에 응답을 한다. 이때 하나의 리더가 동시에 응답한 여러 개의 태그를 인식해야 하는 문제가 발생 하는데 이를 ‘태그 충돌’이라고 한다. 최근 설치비용 절감을 위해 다양한 태그를 인식이 가능한 멀티 리더가 도입되면서 태그 충돌 현상이 점차 늘고 있는 상태이다.

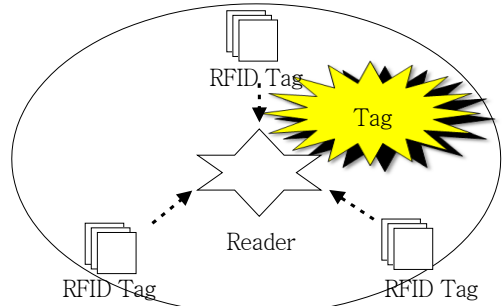


그림 2. 태그 충돌 현상  
FIG. 2 Tag collision

태그를 고속으로 인식하기 위해서 이러한 충돌문제를 해결하는 것이 충돌방지 알고리즘이며 RFID 시스템의 핵심기술 중 하나이다.

## 2.2 충돌방지 알고리즘의 종류

현재 일반적으로 사용되는 충돌방지 알고리즘은 알로하(ALOHA) 방식과 이진-트리(Binary-tree) 방식으로 나뉘어진다. 알로하 방식으로는 태그가 응답하는 시간을 고정된 몇 개의 슬롯으로 나누고 태그들이 각자 선택한 슬롯에 데이터를 전송 하도록 하는 슬롯 알로하(Slotted-ALOHA) 방식이 있다. ISO/IEC 18000-6 Type A 는 다수 태그의 구분을 위해 동적으로 슬롯의 개수를 조정하여 충돌을 조종하는 Adaptive Slotted-ALOHA 방식을 사용한다. 슬롯의 크기는 리더가 동적으로 태그들에게 전달하고, 태그는 자신의 슬롯을 랜덤으로 선택하여 응답한다[3].

이 외에 많이 사용되는 이진-트리 알고리즘은 충돌이 발생한 비트 위치를 이용하여, 동시에 응답하는 태그의 수를 적절히 제어함으로써 충돌을 방지한다. ISO/IEC 18000-6 Type B 에서는 충돌 조정을 위해 기본적으로 이진-트리 개념을 적용하여 사용하게 된다. 이 알고리즘은 충돌 발생 시 전체 태그를 세부 하위 그룹단위로 나누면서 최종적으로 하나의 태그를 찾는 과정을 반복하여 진행된다. 또한 EPC global의 Class 0, Class 1에서 이진-트리알고리즘을 채택하고 있다. 이러한 알고리즘의 성능은 태그와 리더 사이의 프로토콜과 리더 시스템에 따라 결정될 수 있다. 리더가 다수의 태그를 구분하고 특정한 하나의 태그와 데이터 교환을 하기 위해 충돌 조정을 하는 과정에서 태그는 다양한 상태 변화를 일으키며, 자신 및 다른

태그들에게 브로드캐스트 되는 명령을 각각의 현재 상태에 맞게 처리한다.

2.2.1 ALOHA 충돌방지 프로토콜

1) ALOHA 기법

ALOHA 알고리즘은 읽기 전용 태그에서 배타적으로 사용된다. 여기서는 일반적으로 ALOHA 알고리즘은 매우 적은 양의 데이터 일련번호만 전송되며 데이터는 매우 주기적인 과정에 의하여 판독기로 전송된다. 데이터 송신 시간은 반복 시간의 극히 일부만 차지하며 따라서 데이터 전송 간에는 비교적 긴 휴지기간이 존재한다. 더구나 각 태그에 대한 반복시간은 조금씩 다르다. 결과적으로 두 개의 태그가 데이터 패킷을 서로 다른 시간에 전송하여 데이터 패킷이 서로 충돌 하지 않기 위한 확률 값이 존재한다.

ALOHA 시스템에서 데이터 전송의 시간적 시퀀스는 그림 3과 같다. 개선된 부하 G는 시간 t0 (즉 0, 1, 2, 3...)에서 어떤 위치에서 전에 전송하는 트랜스폰더의 수를 의미한다. 제시된 부하(Offered Load) G는 관찰시간 T에서의 평균으로서 데이터 패킷의 전송시간 τ로부터 매우 간단하게 계산할 수 있다[3].

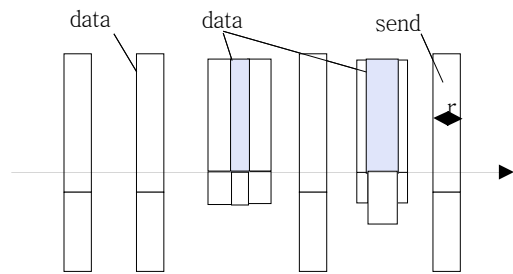


그림 3. ALOHA 시스템에서 부하 G의 정의와 처리량 S의 관계  
Fig. 3 Relation of load G and throughput

$$G = \sum_{n=1}^n \frac{r_n}{r} \cdot r_n \quad (1)$$

n = 1, 2, 3...은 시스템 내의 태그의 수이고 r<sub>n</sub> = 0, 1, 2...는 관찰 시간 동안 태그 n에서 전송되는 데이터 패킷의 수이다. 처리량(Through put) S는 오류(충돌)없

이 데이터 패킷이 전송되었을 경우 1의 값을 갖는다. 그 이외의 모든 경우에는 S가 0의 값을 갖는다.

데이터가 전송되지 않았거나 데이터 충돌로 오류가 발생하여 데이터를 읽을 수 없기 때문이다. 전송 채널의 평균처리량 S는 제시된 부하 G로부터 식(2)와 같이 산출할 수 있다[4].

$$S = G \cdot e^{(-2G)} \quad (2)$$

부하 G와 관련하여 최적의 처리량 S를 구하면 G=0.5에 G서 최대 18.4%를 구할 수 있다. 만약 제기된 부하가 작은 값을 가지면 전송 채널은 대부분 사용되지 않으며 부하 값이 증가하게 되므로 개별적인 태그간의 충돌 횟수는 즉시 급속하게 증가하게 된다. 역방향 링크에서는 ALOHA 시스템을 이용한 충돌방지 기법은 구조가 단순하기 때문에 태그의 수가 제한되어 부하가 작은 경우에 매우 유용한 충돌방지 기술이다. ALOHA 시스템에서 개별적인 패킷이 충돌 없이 전송될 수 있는 확률 즉, 성공 확률 q는 평균부하 G와 처리량 S로 S부터 다음과 같이 산출할 수 있다

$$q = \frac{S}{G} = e^{(-2G)} \quad (3)$$

식(3)으로부터 구한 데이터 시트로부터 판독영역 내에 있는 모든 태그를 읽을 수 있는 시간은 다음 표 1 같다.

표 1. 판독 영역내의 태그를 읽는 평균 소요 시간  
Table. 1 Mean read time of tag

Tag	Mean	90% Reliability	90.9% Reliability
2	150 ms	350ms	500ms
3	250 ms	550ms	800ms
4	300 ms	750ms	1000s
5	400 ms	900ms	1250s
6	500 ms	1200ms	1600s
7	600 ms	1500ms	2000s
8	800 ms	1800ms	2700s

ALOHA 기법을 이용해 RFID의 충돌 여부를 관찰하는 시간  $T$  동안  $k$ 개의 오류 없는 데이터 패킷을 전송할 수 있는 확률값  $p(k)$ 는 데이터 패킷 전송시간  $\tau$ 와 평균제시 부하  $G$ 로  $G$ 부터 계산될 수 있다. 확률값은 평균값  $G/\tau$ 를 갖는 푸아송(Poisson) 분포를 보이며 수식으로 표현하면 식 (4)와 같다.

$$p(k) = \frac{\left(G \cdot \frac{T}{\tau}\right)^k}{k!} \cdot e^{-G \frac{T}{\tau}} \quad (4)$$

## 2) 슬롯 ALOHA 기법

ALOHA 알고리즘에 비해 비교적 낮은 처리량을 최적화하기 위한 또 다른 기술이 슬롯 ALOHA 알고리즘이다. 슬롯 ALOHA 알고리즘에서 태그들은 데이터 패킷을 시간적으로 정해진 동기된 시점에서만 데이터 패킷의 전송을 시작할 수 있다. 이를 위하여 필요한 모든 태그의 동기화는 판독기에 의하여 제어될 수 있다. 따라서 이와 같은 작업은 확률적이고 판독기에 의한 TDMA (Time Division Multiple Access) 충돌방지 프로시저라고 할 수 있다. 슬롯 ALOHA 알고리즘은 충돌이 발생하는 기간(Collision Interval)은 ALOHA 알고리즘을 사용했을 경우 보다 최대 반 정도로 짧아진다[5].

데이터 패킷의 길이가 동일하여 동일한 전송시간  $\tau$ 를 갖는다고 가정하면 ALOHA 방식에서는 두 개의 태그가  $T \leq T2\tau$  내에 리더로 데이터 패킷을 전송하고자 하는 경우 충돌이 발생하게 된다. 슬롯 ALOHA 알고리즘에서는 데이터 패킷이 동기된 시각에서만 전송될 수 있으므로 충돌간격(Interval)이  $T = \tau$ 로 줄어든다. 결과적으로 슬롯 ALOHA 프로시저에서는 처리량  $S$ 는 다음과 같은 관계식을 갖는다.

$$S = G \cdot e^{-G} \quad (5)$$

식(2)와 식(5)로부터 제시된 부하에 대한 ALOHA 방식과 슬롯 ALOHA 방식의 처리량을 나타내면 그림 4와 같다.

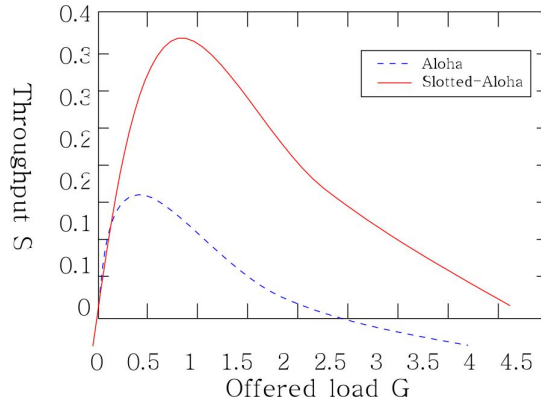


그림 4. ALOHA와 슬롯 ALOHA간의 처리량  
Fig. 4 Throughput of ALOHA and S-ALOHA

그림 4에서 보는 바와 같이 슬롯 ALOHA 알고리즘에서는 제시된 부하  $G$ 에 대하여 최대 36.8%의 최대 처리량  $S$ 를 얻을 수 있다. 그러나 몇 개의 데이터 패킷이 동시에 전송된다면 충돌이 발생하게 되며 하나의 태그가 다른 태그 보다 판독기에 더 가까이 있다면 판독기에 더 강한 신호를 전송할 수 있으므로 다른 데이터 신호들은 판독이 어렵게 된다. 이러한 현상을 캡처효과(Capture Effect)라고 하며 이와 같은 경우 처리량은 다음과 같다.

$$S = G \cdot e^{-\frac{b \cdot G}{1+b}} \quad (6)$$

여기서 결정적 역할을 하는 것이 문턱 값(Threshold)  $b$ 이다. 이 값은 데이터 패킷이 수신기에 오류 없이 도달하기 위하여 다른 데이터 보다 더 강해야 하는 양을 나타낸다.

### 2.2.2 이진 충돌방지 프로토콜

이진검색 충돌방지 알고리즘이란 각각 고유 ID를 갖고 있는 여러 개의 태그가 동시에 데이터 패킷을 전송하는 경우 충돌이 발생하는 비트 위치를 이용하여 태그의 고유번호를 인식하는 기술이다. 즉 충돌이 발생한 경우 응답하는 태그의 수를 감소시킴으로써 태그 충돌을 방지하는 기술이다. 이진검색 알고리즘에는 기본이진검색 알고리즘(Basic Binary Algorithm), 동적이진 검색 알고리즘(Dynamic Binary Algorithm)이 있다.

1) 기본 이진검색 알고리즘

기본 이진검색 알고리즘은 초기 판독기가 인식 가능한 영역내의 모든 태그를 호출 한다. 만약 태그의 ID 비트 간에 충돌이 발생하면 충돌이 발생한 위치를 표시하고 충돌이 발생한 최상위 비트를 0, 나머지 비트를 1로 변환하여 변환 값 이하의 ID를 갖는 태그를 호출한다. 이와 같은 과정은 모든 태그를 검색할 때까지 반복하며 이 때 하나의 태그를 인식하기 위한 반복 횟수는 다음식과 같고 태그 수에 따라 반복 횟수를 구해 도시하면 그림 4-5와 같다[61][7].

$$I(N) = \frac{\log(N)}{\log(2)} + 1 \quad \text{where, } \begin{cases} \bullet N = \text{태그수} \\ \bullet I(N) = \text{반복횟수} \end{cases} \quad (7)$$

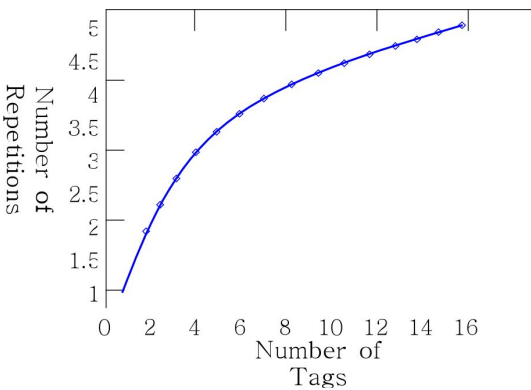


그림 5. 기본 이진검색 알고리즘을 이용하여 하나의 태그를 인식하기 위한 반복 횟수  
Fig. 5 Number of repetition using BS

그림 5에서 보는 바와 같이 기본 이진검색 알고리즘은 ID 비트가 길어질 경우 태그 인식을 위해 전송되는 데이터양이 증가하게 되고 처리 시간이 증가하는 문제점이 있다. 기본 이진검색 알고리즘에 대해 좀 더 구체적으로 고찰하기 위하여 예를 들어 설명하면 다음과 같다. 4비트의 ID를 갖는 태그 4개를 기본 이진검색 알고리즘을 이용하여 검색하는 경우 태그인식 과정은 그림 6과 같다.

기본 이진검색 알고리즘에서 검색 첫 단계에서는 판독기가 인식 영역내의 모든 태그의 ID를 호출하고 충돌 지점을 검색한다. 이 경우 모든 태그의 ID 비트를 비교하였을 때 일치하는 비트가 없으므로 판독기는 모든 비트에서 충돌(×)이 발생한 것으로 간주한다. 판독기는 충돌이 발생한 최상위 비트를 0, 나머지 비

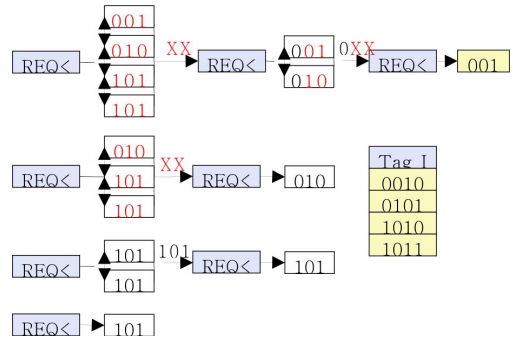


그림 6. 기본 이진검색 알고리즘을 이용한 태그 인식과정

Fig. 6 Tag recognizing process using BS

트를 1로 변환하여 REQ ≤ 0111 이하의 ID 비트를 갖는 태그를 호출하게 된다. 이 호출에 응답하는 태그는 0010, 0101 이다. 응답한 두 개의 태그를 비교하면 2, 3, 4번째 비트 모두 충돌이 발생했다. 그러므로 충돌이 발생한 최상위 비트를 0, 나머지 비트를 1로 변환하여 REQ ≤ 0011 이하의 비트를 호출한다. 이 호출에 대해서 충돌 없이 하나의 태그 0010이 응답한다. 만약에 충돌 없이 하나의 태그 ID를 식별하면 판독기는 Select 명령을 전송하여 검출된 태그를 선택한다. 선택된 태그는 간섭 없이 판독기에 의한 데이터의 읽기/쓰기가 가능하며 다른 태그들은 선택된 태그가 읽기/쓰기 명령(READ DATA)에 응답하는 동안 조용히 대기한다. 읽고 쓰기 동작을 완성한 후에 태그는 Unselect 명령에 의하여 완전히 비활성 상태로 되고 다음의 Request 명령에 더 이상 응답하지 않는다. 나머지 태그들도 동일한 방법을 반복하여 인식하게 된다.

2) 동적 이진검색 알고리즘

기본 이진검색 알고리즘은 판독기의 호출 명령에 태그의 모든 ID 비트를 전송한다. 따라서 ID의 비트가 길어지면 태그의 인식을 위해서 전송되는 데이터 양이 증가 하게 되고 그로 인해 처리시간이 길어지며 에너지 소모량이 증가하는 문제점이 발생한다. 동적 이진검색 알고리즘은 기본 이진검색 알고리즘의 문제점을 보완하기 위해 개선된 알고리즘이다. 태그 인식 과정에서 충돌이 발생하면 판독기는 Valid Bit(VB)를 추가하여 태그를 호출한다. VB는 충돌이 발생한 비트

의 위치를 나타낸다. 판독기가 VB와 함께 태그를 호출하면 태그는 판독기에 알려진 비트를 제외한 나머지 비트를 전송하며 응답을 하게 된다. 따라서 동적 이진검색 알고리즘은 기존 이진검색 알고리즘에 비해 전송 데이터양이 50% 감소하게 된다. 동적 이진검색 알고리즘을 이용하여 인식하는 과정을 앞에서 보인 기본 이진검색 알고리즘과 동일한 예를 사용하여 나타내면 그림 7과 같다.

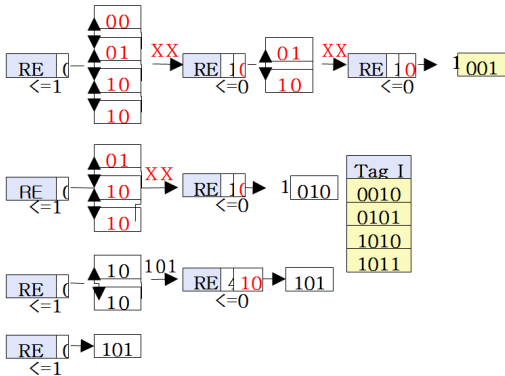


그림 7. 동적 이진검색 알고리즘을 이용한 태그 인식과정

Fig. 7 Tag recognizing process using D-BS

그림 7에서 보는 바와 같이 인식방법은 기본 이진검색 알고리즘과 동일하다. 즉 같은 알고리즘을 사용하기 때문에 태그를 인식하기 위한 반복 횟수는 동일하다. 다만 이 과정에서 VB를 추가하였기 때문에 태그를 인식하는 과정에서 전송되는 총 데이터 전송량을 감소시킬 수 있다. 동적 이진검색 알고리즘은 처음 판독기가 VB가 0인 충돌이 발생하지 않은 경우의 모든 태그를 호출하게 되면 판독기의 인식영역 내의 모든 태그는 자신의 ID를 전송하면서 응답을 하게 된다. 여 기서는 4개의 태그가 응답을 하면서 모든 비트의 충돌(x)을 판독기가 인지하게 되고 판독기는 충돌이 발생한 최상위 첫 번째 비트가 0인 태그를 호출하게 된다. 단 이 부분에서 최상위 비트를 0으로 결정하였으므로 결정된 비트의 번호를 VB=1로 전송한다. 이때 조건에 해당하는 태그(0010, 0101)는 VB와 호출에 따라 첫 번째 비트 0을 제외한 나머지 비트를 리더로 전송하면서 응답한다. 응답을 받고 나면 첫 번째 비트 0을 저장한다. 응답한 두 개의 태그에서도 세 비

트에서 충돌이 발생한다. 판독기는 첫 번째 비트가 0, 두 번째 비트가 0인 태그를 호출하게 되고 두 비트를 결정하였으므로 VB=2로 전송한다. 이와 같은 호출에 대해 해당되는 태그는VB에 따라 상위 두 비트를 제외한 ID 비트를 전송 한다. 하나의 태그가 충돌 없이 검출되면 판독기는 SELECT 명령을 전송하여 검출된 태그를 선택하고 태그의 정보를 저장한다. 이 후 태그는 Unselect 명령에 의하여 완전히 비활성 상태가 되고 다음의 Request 명령에 더 이상 응답하지 않는다. 같은 방법으로 나머지 모든 비트가 검색될 때까지 반복하게 된다. 동적 이진검색 알고리즘에서 하나의 태그를 인식하기 위한 반복 횟수는 기본 이진검색 알고리즘과 동일하다. 그러나 데이터의 총 전송량은 이진검색 알고리즘과 비교할 때 50%까지 감소한다.

### 2.2.3 이진트리 충돌방지 프로토콜

이진트리 충돌방지 알고리즘이란 각각 고유 ID를 갖고 있는 여러 개의 태그가 동시에 데이터 패킷을 전송하는 경우 충돌이 발생하는 비트를 파악하여 충돌이 발생한 최상위 비트부터 '0'과 '1'의 두 그룹으로 선별하여 태그의 고유번호를 인식하는 기술이다. 이진트리 알고리즘은 충돌이 발생한 경우 그룹을 구분하여 검색횟수 및 전송데이터를 효율적으로 감소시키는 기술이다. 가장 기본적으로 사용되는 이진트리 알고리즘에는 슬롯 이진트리 알고리즘(Slotted Binary Tree Algorithm)과 비트별 이진트리 알고리즘(Bit-by-Bit Binary Tree Algorithm) 이 있다[8].

#### 1) 슬롯 이진트리 알고리즘

슬롯 이진트리 알고리즘의 기본적인 개념은 판독기의 인식 영역내에 존재하는 태그의 ID를 호출하여 충돌이 발생한 부분부터 '0'과 '1'의 두 그룹으로 나누어 검색하는 방식이다. 판독기가 '0'의 그룹을 선택할 경우 '1'의 그룹의 태그들은 비활성 상태로 대기하게 된다. 한 그룹을 선택하여 하위 ID 비트들에 대해서도 그룹을 분할하여 태그 ID를 인식한다. 태그 인식 과정을 반복하여 하나의 태그가 검출되면 전 단계의 분기된 슬롯에서비활성 상태로 대기 하던 그룹을 활성화시켜 태그를 검출하게 된다. 이와 같은 방식을 반복하여 모든 태그를 인식하게 된다. 이때 각 단계의 인식과정에서 호출에 응답하는 전송 데이터는 최상위

비트부터 하위비트까지 차례로 한 비트씩 전송하게 되므로 이진검색 알고리즘보다 데이터 전송량을 감소시킬 수 있다. 태그 인식을 위한 평균 반복 횟수는 다음식과 같다[8]

$$I_{BEST} = 1 + \sum_{k=2}^n \binom{n}{k} \frac{2(k-1)(-1)^k}{[1-p^k - 1(1-p)^k]} \quad n \geq 2 \quad (8)$$

여기서  $n$ 은 태그 수이고  $p$ 는 '0' 또는 '1'을 선택할 확률이다. 이때 확률  $p$ 는 0.5 이다. 슬롯 이진트리 알고리즘을 이용하여 4비트의 4개 태그를 인식하는 과정을 나타내면 그림 6과 같다. 그림 8에서 보는 바와 같이 최상위 비트를 기준으로 '0'과 '1' 그룹으로 나누어 검색을 하게 된다. '0'의 그룹을 선택하게 되면 '1'의 그룹 태그들은 비활성 상태로 대기 하게 된다.

그룹을 선택하고 나면 최상위 비트가 '0'인 그룹에 해당하는 태그 ID의 두 번째 비트를 호출한다. 하나의 태그가 응답하면 마지막 하위 비트까지 차례로 호출하여 하나의 태그 ID를 인식한다. 그러나 두 번째 비트 호출에 대해 두 개 이상의 태그가 응답한 경우 다시 '0'과 '1'그룹으로 구분하여 '0' 그룹을 호출하고 '1' 그룹은 비활성화 시킨다. 이와 같은 방식을 반복하여 하나의 태그(0010)가 검출되면 바로 전 단계의 분기된 슬롯 두 번째 슬롯에서 비활성 상태의 그룹을 활성화시켜 태그를 검출하게 된다. 슬롯 이진트리 알고리즘은 충돌 발생지점에서 임의의 그룹을 선택함으로써 검색범위를 좁혀가는 방식으로 호출 횟수를 줄일 수 있다. 또한 각 호출에 응답하는 태그는 자신의 ID 비트 중 해당하는 한 비트만 전송함으로써 총 전송 데이터양도 감소시킬 수 있다.

## 2) 비트별 이진트리 알고리즘

비트별 이진트리 알고리즘은 판독기의 요청메시지에 따라 태그가 자신의 일련번호를 한 비트 씩 전송하는 방식이다. 태그들이 보내는 비트 사이에 충돌이 발생하지 않으면 판독기는 태그로부터 받은 비트를 메모리에 저장한 후 다음 비트를 요청하게 된다. 그러나 충돌이 발생하면 판독기는 '0' 또는 '1' 중에 하나를 선택하여 이에 해당하는 과정을 태그의 일련번호 길이만큼 반복함으로써 하나의 태그를 인식하게 된다.

만일  $n$ 개의 태그가 존재하고 각 태그의 고유번호가  $j$  비트라면 모든 태그를 인식하기 위해 필요한 총 반복 횟수는 식(9)과 같다[9].

$$I_{BBT} = n \times j \quad (9)$$

판독기의 요청에 의해서 모든 태그는 자신의 첫 번째 비트부터 마지막 비트까지 순차적으로 하나씩 전송한다. 충돌이 발생하지 않은 비트의 값은 메모리에 저장하고 다음 비트 값을 요구하게 된다. 판독기는 임의로 '0'의 값을 가지는 태그만을 선택하고 '1'의 값을 가지는 태그는 일시적으로 비활성화 시킨 후 최상위 비트가 '0'인 태그들의 두 번째 비트를 호출한다. 또 다시 충돌이 발생하면 리더는 '0'을 선택하고 '1'을 비활성화 시킨다. 이와 같은 과정을 태그 ID 비트의 길이만큼 반복하면서 태그 ID 하나를 검출한다. 검출된 태그는 인식 신호를 전송하고 태그의 데이터를 저장한 후 비활성화시키고 나머지 태그들에 의해 처음부터 같은 방식을 반복하면서 인식 영역내의 모든 태그를 검출하게 된다. 즉 비트별 이진트리 알고리즘은 간단하게 구현 할 수 있으나 하나의 태그를 검출하기 위해서 태그의 ID 길이만큼 검색 과정을 반복하여야 하며 또한 인식 영역내의 태그가 여러 개 존재하는 경우 검색 횟수가 비례하여 증가하게 된다[9].

### 2.2.4 충돌방지 프로토콜의 장·단점 분석

슬롯 이진트리 충돌방지 알고리즘과 비트별 이진트리 충돌방지 알고리즘은 임의의 '0'과 '1'의 그룹을 나누어 반복적으로 태그를 구별함으로써 불필요한 검색 과정을 거친다. 태그를 호출하는 반복 횟수의 증가는 태그의 정보전송 횟수를 증가시킴으로써 제한된 에너지를 사용하여야 하는 태그의 성능을 저하시키는 원인이 된다. 또한 태그에서 전송되는 정보비트의 증가는 판독기의 정보처리 능력에 큰 부담을 준다. 이는 결국 고가의 판독기 연산장치를 요구하게 됨으로써 RFID 전체 시스템의 비용을 증대 시키게 된다. 기존의 이진검색 알고리즘에서 태그는 모든 태그ID 비트를 판독기에 보낸다. 즉 태그 충돌이 일어나게 되면 일어난 시점으로부터 다음 비트를 검색하는 것이 아니라 처음부터 다시 모든 비트를 검색하게 된다.

따라서 태그 ID의 충돌이 일어난 비트수가  $S$ 이고



태그의 ID 비트 수가  $n$ 일 때 태그 ID의 길이가 보내고자 하는 ID의 길이보다  $5^{n-1}$  만큼 더 길어지게 된다. 그러므로 태그 ID의 길이가 길어지면 하나의 태그를 인식하기 위해 전송해야 하는 데이터양이 증가할 뿐만 아니라 하나의 태그를 처리하는 시간 증가로 인해 에너지 소모량을 증가시키는 결과를 가져온다.

### III. 고속 태그 충돌 방지 알고리즘 제안

제안 한 충돌 방지 알고리즘의 주요한 목표로는 기존 바이너리 서치(Binary Search) 계열의 BER (Bit Error Rate)를 개선하기 위하여 두 개 이상의 태그로부터 복수의 사용자 ID 충돌 전송 프레임의 반복 전송 횟수 및 충돌 프레임 수를 줄이는 데 있다고 하겠다. 충돌 전송 프레임의 반복 전송 횟수를 줄이기 위해서는 태그의 전송 절차에 따라 사용자 ID간섭을 최소화 하기위한 스케줄이 제안되어야 한다. 기존 BS 계열의 충돌 방지 알고리즘은 많은 반복 횟수와 큰 지연 시간이 소요되어지는 단점을 가지고 있다. BS 계열은 기본적으로 Tree-Based 충돌 회피 방식을 적용하고 있어 이를 기반으로 충돌 회피 알고리즘이 제안되어진다. 본고에서는 이용 가능한 태그의 인식 번호는 단순성을 위하여 4비트로 인식되어지는 것으로 가정한다[9][10][11].

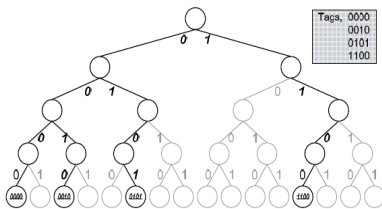


그림 8. 일반적인 BS계열 Tree-Based 알고리즘 리더와 태그 사이는 4비트로 가정  
Fig. 8 TB Algorithm with normal BS

그림 8은 기본 2진 BS기반 Tree-Based 검색 충돌 회피 알고리즘은 리더의 요청 메시지에 따라 태그가 자신의 고유번호를 한 비트 씩 전송하는 방식이다.[4] 리더가 태그의 인식 번호를 구별하기 위해 하나의 비트를 보내면 그때 각 태그들은 이용이 가능한 자신의 인식 번호를 가진 비트 정보와 비교한다. 이 구성 방

식에 의하여 비트의 위치와 이용도는 비교되어지며 태그들의 메모리에 저장되어지게 된다. 태그들이 보내는 비트 사이에 충돌이 발생하지 않으면 리더는 태그로부터 받은 비트를 메모리에 저장한 후 다음 비트를 요청하게 된다. 그러나 충돌이 발생하면 리더는 0 또는 1 중에서 하나를 선택하여 이에 해당되는 태그들만 다시 응답하도록 하고 다음 비트를 요청하게 된다. 이런 과정을 태그의 고유번호 길이만큼 반복함으로써 하나의 태그를 인식하게 된다[8][9]

interrogator	Responses from transponders
t0 : broad cast	
t1 : send 0	receive 0000/0010/0101/1100 (XXXX)
t2 : send 0	receive 0000/0010/0101 (0XXX)
t3 : send 0	receive 0000/0010 (00XX)
t4 : broad cast	receive 0000
t5 : send 0	receive 0010/0101/1100 (XXXX)
t6 : send 0	receive 0010/0101
t7 : broad cast	receive 0010
t8 : send 0	receive 0101/1100 (X10X)
t9 : broad cast	receive 0101
t10	receive 1100

그림 9. 기본적인 Tree-Based 검색 구조  
Fig. 9 Search structure of basic TB

그림 9은 매 단계마다 비트 송수신 추적에 대한 내용을 보여주고 있다. 리더기는 t0에서 브로드캐스트 프레임을 전송한 후 모든 태그들의 인식 번호들은 t1에 응답되어지게 된다. t1에서 리더기는 '0'을 보내고 태그들은 첫 번째 위치의 비트가 '0'과 동일한 태그를 받게 된다. 세 번째 태그의 식별 번호를 받은 후에 리더기는 t2에서 또다시 '0'을 보내면 t2에 '0'과 동일한 두 번째 비트를 받게 된다. t3에 다시 '0'을 보낸 후, t4에서 리더기는 처음으로 완전한 태그 인식 번호인 '0000'을 얻게 된다. 나머지 태그의 인식 번호를 얻기 위해서도 t0, t1, t2, t3와 같은 과정을 반복하여 t0~t10까지 11단계를 반복하여 최종 4비트 이용 식별을 모두 얻게 된다. 위와 같은 기본 Tree-Based 충돌 회피 검색 구조의 검색 성능을 좀 더 높이기 위해 Query Tree 검색 알고리즘이 제안되어졌다

그림 10은 Query Tree 알고리즘 송수신 절차를 보여준다.

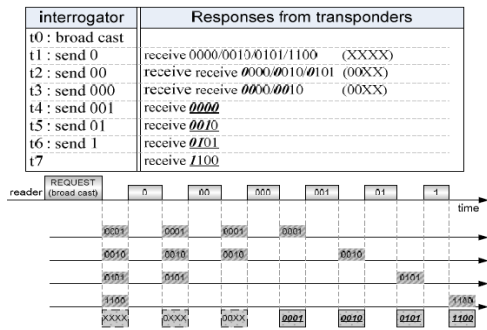


그림 10. Query Tree 검색 구조  
Fig. 10 Search structure of Query Tree

Query Tree 검색 알고리즘도 이미 그림 9와 같은 송수신 절차를 가져간다. t0시간에 리더기는 브로드캐스트 프레임에 보냄으로써 충돌 회피 절차를 시작한다.

이때 0000/0010/0101/1100의 4비트 태그 식별 번호를 모두 받게 된다. 기본적인 Tree-Based 검색 구조에서는 매번 1비트 정보를 보내지만 Query Tree 검색에서는 브로드캐스트 후 패턴을 exclusive-OR 동작과 비교 하며 충돌 회피 중재 비트로 처음에는 '0' 그다음에는 '00' 3단계에서는 '000' 4단계에서는 역순으로 '001' 5단계 '01' 6단계 '1'을 송수신하여 t0~T7단계로 8단계로 성능이 개선될 수 있도록 하는 방식이다. 최근에 연구되어 좀더 향상된 Query Tree 검색 알고리즘으로는 위치 비트와 충돌 회피를 위한 패턴 정보로 구성된 특별한 프레임 형식을 정의하여 성능을 개선시킨 알고리즘이다. 이 그림 11 알고리즘은 그림 10와 리더기의 송수신 절차 방식을 제외하고는 나머지는 같은 구조를 가지고 있다. 이 그림 11 알고리즘은 그림 9와 그림 10의 장점을 혼합한 방식이다.

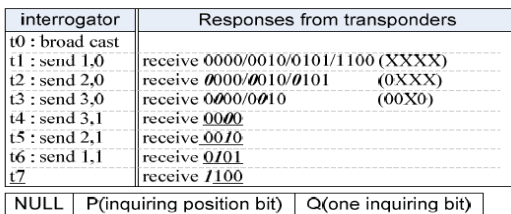


그림 11. 향상된 Query Tree 검색 구조  
Fig. 11 Search structure of advanced Query Tree

t1에서 보내어진 '1,0'은 첫 번째 위치에서 '0'을 수

신하는 리더 중재기 역할을 수행하는 것으로 리더기의 전송 프레임 길이는 짧으며 각 단계별로 수신 되어진 정보를 저장할 필요가 없게 되어 보다 성능이 개선되었음을 알 수 있다. 그러나 이러한 많은 장점이 있음에도 불구하고 리더기의 전송 스케줄이 그림 11은 그림 10과 마찬가지로 리더기 송수신 단계가 8단계 (t0~t7)단계로 스케줄이 되어진다[10][11].

본 논문에서는 태그 인식 향상을 개선하기 위해 제안하는 알고리즘 개선 목표로는 위에서 이미 정리되어진 기본적인 Tree-Based, Query Tree 검색등의 단점을 보완한 방식으로 두 개 이상의 태그들로부터 복수의 인식 번호를 수신 받음으로써 리더 중재기의 프레임 전송 단계를 줄이는데 있도록 한다. 즉 제안한 구성 및 스케줄 방식에서는 리더 중재기의 특별한 인식 번호 요청 프레임에 따라, 각 태그들의 전송 절차는 인식 번호간의 간섭 현상을 최소화 하도록 스케줄 되어지게 구성하여 리더기의 중재 전송 스케줄도 줄이도록 함으로써 전체 태그 인식 성능을 향상 시키는데 있다고 하겠다. 이와 같은 구성을 이용하기 위해서는, 두 개의 충돌 회피 프레임이 활성화 프레임과 복수 응답 프레임으로 재구성 되어지도록 한다[9].

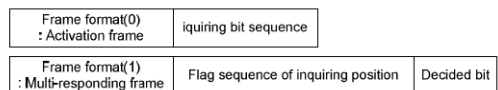


그림 12. 새롭게 제안된 검색 프레임 구조  
Fig. 12 Search frame structure of new suggested

활성화 프레임(AF)은 제공 되어진 그림 [9]와 같은 Query Tree 알고리즘과 거의 같이 동일하게 행하여지고 있다. 그림 12는 충돌회피 프레임을 새롭게 제안한다. 태그 활성화 프레임(AF)는 어떠한 비트 패턴과 어떤 플래그 프레임 포맷을 포함한다. 또한 복수 응답 프레임 (MF)은 프레임 형식, 위치 문의 플래그 절차 (FSIP), 그리고 결정 비트 정보를 포함한다. FSIP 스트리밍 비트 패턴에서, 첫 번째 위치가 '1'의 값이면 리더기 '1'과 '0'에 의해 결정된 첫 번째로 충돌된 위치를 나타내며 두 번째 '1'은 충돌 회피 프레임 질의에 대한 마지막 위치를 나타낸다. 만약 어떤 태그가 리더기로부터 이용 가능하다는 활성화 프레임을 받게 되면 이는 다중 응답 프레임이 대응할 수 있는 활성

화 상태로 복수의 태그가 동작하게 되었다고 할 수 있다. 반면 태그가 비활성화 상태로 가면 태그는 활성화 (AF) 수신 때 까지는 대응하지 않는다. 올바른 충돌 회피 성능 개선을 위해서는 태그 식별 인식 정보 구조에서 수신되어진 첫 번째 충돌 정보는 두 개의 다른 비트가 함께 수신되어진 것으로 여겨질 수 있다. 그러므로 첫 번째로 충돌 되어진 비트 위치에서 태그 식별 인식 정보로 '0' 과 '1' 값 양쪽이 이용되어 졌다고 쉽게 결론지어질 수 있다. 그러나 두 번째 충돌 비트 값은 첫 번째 충돌 비트처럼 손쉽게 결론 내어질 수 없다. 이러한 충돌 회피를 최소화하기 위해서 리더기와 태그사이에는 스케줄이 정리되어야 한다.

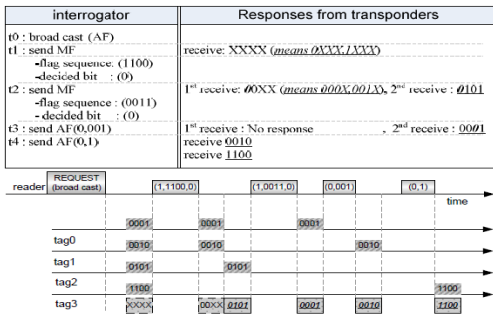


그림 13. 제안된 검색 엔지의 스케줄 구조  
Fig 13. Schedule structure of new suggested

t0에 리더기는 사용자 식별을 위해 모든 태그를 활성화 상태로 만든 브로드캐스트 프레임 전송한다

t1시간에 'XXXX'의 수신으로부터 '0XXX'와 '1XXX' 인식 번호를 전달한다. 그때 리더기는 '1XXX'를 저장하고 '0XXX'를 사용함으로써 복수 응답 프레임( MF)를 송부한다. 첫 번째 비트 위치에서 결정 되어진 비트가 '0'임을 알려주기 위해, 리더기는 위치지정을 위한 플래그 절차를 결정 비트가 '0'을 가지고서 '1100'과 동일한 위치를 가진 플래그 순서를 가지도록 다중 응답 프레임(MF)를 송부한다. t2에서 리더기는 분리된 첫 번째와 두 번째 타임 슬롯에서 '00XX'와 '0101'을 받게 된다. '0101'스트림은 확정된 인증 번호이므로 리더기는 '0101'스트림을 저장하고 '00XX'를 사용하여 새로운 다중 응답 프레임( MF)를 만든다. 더 이상 충돌이 없는 태그 인식 번호판이 없으면 리더기는 새로운 능동적 활성화 프레임(AF)을 만들게 된다. 본 구성 스케줄링을 통하여 t0~t4의 5단계로 인식

식별 및 충돌 회피 방식을 최소화하여 개선할 수 있음을 알 수 있다.

#### IV. 성능 비교 분석

제안된 충돌 회피 알고리즘의 성능을 구현하기 위해서는, 태그 검색 시뮬레이션이 Tree-Based 방식과 제안된 M/F 방식으로 검증하였다. 이때 시뮬레이션을 통한 검증 요소로는 개별 태그 인식 번호의 길이와 태그의 밀도로 기준을 수립 하였다. 태그 밀도는 리더기 운영 범위에서 최대한 존재할 수 있는 태그의 최대 밀도 수 1024 노드를 100%로 수립하였다.

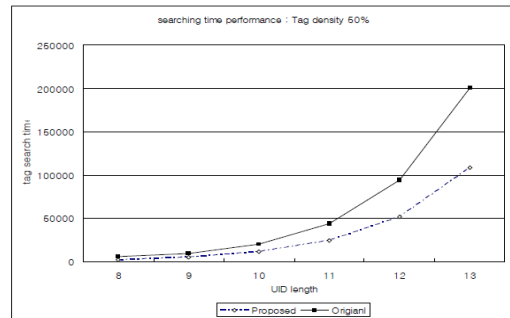


그림 14. 기본 Tree-Base 충돌 회피 방식과 고정형 태그 밀도를 가진 제안형 구조 사이의 검색 시간  
Fig. 14 Searching time of TB algorithm with collision avoidance an suggested TB

그림 14 검색 시간이 50%로 줄어들고 있는 태그 밀도 시간을 보여준다.

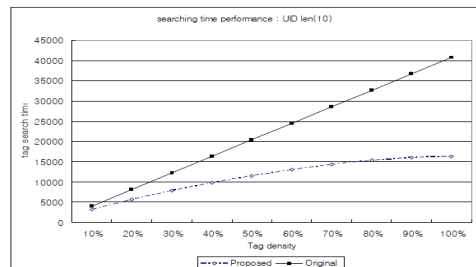


그림 15. 고정 태그 밀도를 가진 기본 Tree-based 알고리즘과 제안 MF방식의 검색 시간 비교  
Fig. 15 Searching time comparison normal TB algorithm and suggested MF

그림 15는 식별 인식표 길이가 10으로 고정되어 있을 때 태그 밀도의 변수로서 검색 시간을 보여준다. 기본 충돌 회피 Tree-Based 태그 검색 시간은 거의 선형으로 증가하나 제안된 MF방식은 태그 밀도를 따라서 증가율이 점점 더 줄어들게 된다는 것을 알 수 있었다. 이 결과는 제안 되어진 MF 방식의 검색 알고리즘이 기본 Tree-based 알고리즘보다도 50%이상의 성능 효과를 보여준다는 것을 알 수 있었다.

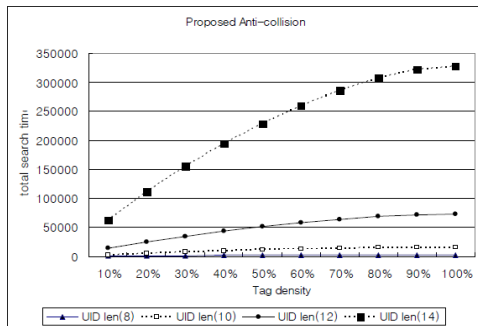


그림 16. 고정 인식 정보 길이 (10)을 가진 기본 Tree-based 알고리즘과 제안 MF방식의 검색 시간 비교

Fig. 16 Searching time comparison TB algorithm and suggested MF

그림 16은 식별 인식표 길이와 태그 밀도가 변수로서 함께 사용되었을 때를 보여준다. 100% 태그 밀도를 가지고서 14로 인식 번호의 길이가 고정되어 있을 때 총 태그의 검색 규격은 600000 ETU 검색 시간을 보여 준다. 예를 들어 13.56MHz RF ID경우에 가장 작은 ETU 기간은 1.2us이다. 대략 0.7초가 모든 Tag를 증명하는데 적용되어짐을 보여준다는 것을 알 수 있었다.

## V. 결론

본 논문을 통하여, 복수의 충돌을 회피하기 위하여 사용되어진 효율적인 Tree-based 충돌 회피 프레임 구조와 복수 응답 태그 스케줄을 사용한 방법이 제안 되어졌다. 결론적으로 개선된 알고리즘 구조는 같은 태그 수를 가지고서도 검색 및 인식 성능 향상이 30% 이상이 개선되어졌음을 알 수 있었다. 또한 같은

길이에서는 50%의 검색 시간이 절감되어졌음을 알 수 있었다.

연구과정에서 드러난 문제점으로는 태그들의 응답이 거리에 따라서 약간의 시간차를 보일 수 있다는 단점을 지적할 수 있다. 또한 RFID 리더의 경우에는 큰 문제가 보이지 않지만, 태그의 경우 구현에 추가적인 오버 헤드가 따르는 것으로 판단된다.

## 참고 문헌

- [1] Lee Su Ryun, "Multi-Sensing current status of RFID system", Journal of electromagnetic engineering an science, Vol. 15, No. 2, pp. 44-45, 4. 2004.
- [2] Lee Gun Ho, "Technology of RFID", TTA journal, pp. 124-120, 10. 2003.
- [3] Klaus Finkenzeller, RFID handbook, Second Edition, Jone Wiley & Sons, 2003.
- [4] R. Raphael, S. Moshe, Multiple Access Protocols, Springer-Verlag, 1990.
- [5] Biddle, P. England, P. Peinado, M. and Wilman, B., "The Darknet and the future of content protection", In Digital Rights Management-Technological, Economic, Legal and Political Aspects. LNCS 2770, Springer, pp. 344-365.
- [6] H. Vogt, "Efficient Object Identification with Passive RFID tags," IEEE ICPC, Zurich, 2002, pp. 98-113.
- [7] S. Sarma, D. Brock, and D. Engels, "Radio frequency identification and electronic product code," IEEE MICRO, 2001, pp. 50-54.
- [8] Klaus Finkenzeller, RFID handbook, Second Edition, Jone Wiley & Sons, 2003
- [9] Young-Jun Lim, "Massive RFID Tag Write Technique using Parallel Deployment of Readers", The journal of The Korea Institute of Electronic Communication Sciences, Vol. 7, No. 6, pp. 1493-1498, 2012.
- [10] Myeong-sook Shin, "Method to Reduce the Time when Identifying RFID Tag by using Computational Grid", The journal of The Korea Institute of Electronic Communication Sciences, Vol. 5, No. 5, pp. 547-554, 2010.
- [11] Han Young Lee, "High Tag anti-collision

algorithm to improve the efficiency of tag identification in active RFID system", The journal of The Korea Institute of Electronic Communication Sciences, Vol. 7, No. 2, pp. 235-242, 2012.

## 저자 소개



### 김익순(Ik-Soon Kim)

2005년 학점인정-정보통신과(공학사)

2009년 한양대학교 전자통신과 졸업(공학석사)

2013년 전남대학교 전자통신과 졸업(공학박사)

현재 수원지방검찰청 정보통신팀장

서울특별시 교육청 자재공법선정위원

조달청 기술평가위원

경기도 공동주택 관리조사단 전문위원

한국방송통신전파진흥원 시험문제 출제위원

※ 자격 : 정보통신기술사

※ 관심분야 : u-City 및 u-Campus IT 구축, 차세대 검찰 통신망 및 법무통신망 융합, BIS, 통합관제 운영센터 제어, 지능형 CCTV



### 김천석(Chun-Suk Kim)

1980년 9월 광운대학교 전자공학과 졸업(공학사)

1982년 9월 건국대학교 전자공학과 졸업(공학석사)

1998년 경남대학교 전자공학과 졸업(공학박사)

1982년~2006년 여수대학교 전자통신공학과 교수

2006년~현재 전남대학교 전자통신공학과 교수

※ 관심분야 : 수중통신, 정보통신분야

