DOI: 10.3745/KIPSTA.2008.15-A.3.155

# 차등식별 알고리즘을 이용한 효율적인 RFID 충돌 방지 기법

김 성 진[†] · 박 석 천[††]


## 요 약

RFID의 태그의 적용 범위는 점차 그 영역을 확대되고 있으며, 동종 및 이종 태그들이 제한된 공간에서 리더에 동시에 식별되는 과정에서 충돌이 발생할 수 있다. 따라서, 하나의 RFID 리더 영역 내에 다수의 태그 상호간에 충돌이 없이 인식하는 다중태그식별 기술은 매우 중요한 기술로 RFID 시스템의 성능과 안정성을 결정하는 핵심 기술이다. 본 논문에서는 차등식별법을 이용한 Depth-First 충돌방지 기법을 제안한다. 시뮬레이션을 통해 검증결과 기존 쿼리트리 알고리즘에 비해 17%의 식별단계 개선과 150%의 성능 개선을 확인할 수 있었다.


**키워드 : RFID, 충돌방지, 무선통신, 정보보호**

# Efficient RFID Anti-collision Scheme Using Class Identification Algorithm

Kim Sung Jin[†] · Park Seok Cheon[††]


## ABSTRACT

RFID technology has been gradually expanding its application. One of the important performance issues in RFID systems is to resolve the collision among multi-tags identification on restricted area. We consider a new anti-collision scheme based on Class Identification algorithm using Depth-First scheme. We evaluate how much performance can be improved by Class identification algorithm in the cases of Query-tree more then 17% identification rate and 150% performance.


**Keyword : RFID, Anti-Collision, Mobile, Wireless, Information Protection**

## 1. 서 론

RFID(Radio Frequency Identification) 기술의 발전은 유통, 물류, 위치확인 등에서 비약적으로 적용 범위를 넓혀가고 있다. 일반적으로 태그에 저장된 자료는 단순 식별번호로 이를 전송하는 전송구간은 무선 주파수 구간으로 무선기술의 사용은 편리한 인식이 가능하지만 무선통신과의 간섭, 다중 리더 환경에서 리더간의 충돌, 다중 태그환경에서는 태그간의 충돌과 같은 정보전송상의 문제점을 내포하고 있다. 이러한 제약조건들은 RFID 시스템의 성능과 안전성에 문제를 야기한다. 따라서, 다중 통신 객체간의 간섭을 통한 정보 손실없이 고속으로 정보를 전송할 수 있는 충돌방지 기술이 요구된다. 기존의 충돌방지 프로토콜은 ALOHA 기반 프로토콜과 트리 기반 프로토콜로 분류된다.

ALOHA 기반의 프로토콜은 확률론적 방식이라고도 하며, 각 태그가 임의의 시간을 선택하여 ID를 전송함으로써 태그 충돌 발생 확률을 감소하는 방식으로 특정 태그가 장기간 리더에게 식별되지 못하는 태그 기아 현상 문제(tag starvation problem)와 부분적인 태그비트간의 충돌로 인해, 태그 식별에 많은 시간이 소요되는 문제가 있다. 이러한 문제를 해결하기 위한 방법이 확률기반의 태그 충돌 방지 방식으로 Slotted ALOHA, Frame slotted ALOHA, Adaptive frame slotted ALOHA 방식이 있으나, 완벽하게 충돌문제를 해결할 수 없다[1].

트리기반의 프로토콜은 결정론적 방식이라고도 하며, 이러한 트리기반의 대표적인 방식으로 이진트리 방식(Binary-Tree)[2][3]과 쿼리트리(Query-Tree) 방식[4][5]이 있다. 이 방식은 동시에 ID를 전송하는 태그들을 하나의 집합으로 묶어 집합단위로 태그를 식별하는 기법으로 태그 충돌이 발생되면, 집합을 두 개의 하위집합으로 나눠 하나의 집합이 하나의 태그만을 포함할 때까지 태그 집합 분할을 수행함으로 식별 범위 내에 존재하는 모든 태그를 식별할 수 있으나, 태그 식별에 많은 시간이 소요된다는 문제는 여전히 개선해야 할 점이다.

따라서, 본 논문에서는 트리 기반의 충돌 방지기법을 개선한 방식을 제시한다. 제안방식은 식별태그 중에서 동일

† 정 회 원 : 성남산업경제연구센터 선임연구원
†† 종신회원 : 경원대학교 소프트웨어학부 정교수
논문접수 : 2007년 9월 18일
수 정 일 : 1차 2008년 1월 25일, 2차 2008년 5월 7일
심사완료 : 2008년 5월 7일

계열의 태그를 우선 식별하고, 다른 계열의 태그 식별 방식으로 동일 계열 Depth-First기법의 차등적 식별방식으로써 유사한 태그가 군집되어 있는 환경인 물류창고 및 위치추격을 위한 목적에서 효과적으로 사용될 수 있는 방식이다. 이러한 방식을 이용할 경우, 태그 식별 소요시간과 에너지 소비량을 감소할 수 있다. 차등식별방식을 이용한 효율적인 RFID 충돌 방지 기법을 제안하고, 그 성능을 동일조건상에 정량적인 평가와 수학적 평가로 제안 기법의 성능향상을 확인하였다.

논문 구성은 1장의 서론에서 연구배경, 2장에서 기존방식에 대한 고찰, 3장에서 새로운 제안 방식 제안 및 4장에서 제안방식 시뮬레이션, 5장에서 결론을 맺도록 구성했다.

## 2. 기존 방식에 대한 고찰

2장에서는 기존에 충돌방지 프로토콜을 살펴보고, 개선점을 정리한다. 모든 방식에서 인식결과는 리더로부터 태그 ID의 파악상태에 따라, 아무런 태그도 반응하지 않는 휴면(Idle) 상태, 하나의 태그만 반응하여 인식이 성공한 식별(Success) 상태, 두 개 이상의 태그가 반응하여 충돌이 발생한 충돌(Collision) 상태의 세 가지 상태로 구분할 수 있다.

### 2.1 ALOHA 기반 프로토콜

ALOHA 기반의 프로토콜은 확률론적 방식이라고도 하며, 각 태그가 임의의 시간을 선택하여 ID를 전송함으로써 태그 충돌 발생 확률을 감소하는 방식으로, 특정 태그가 장기간 리더에게 인식되지 못하는 태그 기아 현상 문제(tag starvation problem)와 부분적인 태그비트간의 출동 문제가 있다. 이러한 문제를 해결하기 위한 방법이 확률기반의 태그 충돌 방지 방식으로 Slotted ALOHA, Frame slotted ALOHA, Adaptive frame slotted ALOHA 방식이 있다.

Slotted ALOHA은 ALOHA 방식에서 발생되는 부분충돌 문제를 해결하기 위해 태그의 데이터 전송 시작 시점을 동기화하는 방식이다. Frame slotted ALOHA 방식은 여러 슬롯을 하나의 프레임으로 묶어 한 프레임 중 하나의 슬롯에서만 전송하는 기법으로 기존에 충돌을 야기한 태그를 통한 2차 충돌을 막을 수 있는 기법이다. 이러한 방식에서 ALOHA방식은 시간, Slotted ALOHA 방식은 슬롯, Frame Slotted ALOHA 방식은 프레임크기가 각각 랜덤 스페이스로 태그가 전송시점을 결정하기 위해 사용하는 영역의 크기가 된다. 이중에서 Frame 단위의 사이즈 조정이 상대적으로 다른 방식에 비해 용이하고, 이를 최적화한 방식이 Adaptive Frame slotted ALOHA 방식으로 가장 많이 사용되는 방식으로 태그 수를 예측하고 최적의 랜덤 스페이스 값인 프레임 사이즈를 구할 수 있는 방식이다.

### 2.2 Tree 기반 프로토콜

Tree 기반 프로토콜은 결정론적 방식으로 태그가 데이터를 전송하는 시점은 리더의 메시지와 태그의 메시지가 수행한 연산의 결과로 설정하는 방식이다. 리더는 태그를 두 그룹(전송그룹과 대기 그룹)으로 나누고, 전송그룹에 대해 그룹을 구성하는 태그 수가 하나가 될 때까지 구분하는 작업을 반복 수행한다. 이 방식은 다시 이진트리 기법과 쿼리트리 기법으로 나눠진다.

이진트리기법은 태그 내에 난수발생기와 그룹을 구별하기 위한 카운터를 두고, 초기에 리더로부터 메시지 전송이 있을 경우, 태그는 난수발생기로 0과 1중에 하나의 난수를 생성한다. 이 난수를 카운터에 더해 카운터 값을 설정한다. 이러한 과정을 반복하며 태그는 카운터 0인 그룹과 1인 그룹으로 분리된다. 카운터 0 그룹은 메시지를 전송하고 리더는 식별여부를 태그에게 알린다. 전송중에 충돌이 발생되면 전송 태그들은 난수를 발생하여 다시 두 그룹으로 나누어지고, 전 단계에서 전송하지 않은 태그는 카운터를 1증가하고, 전송 중 충돌이 발생하지 않으면 모든 태그 카운터를 1감소시킨다. 이때, 식별에 성공한 태그는 카운터 값이 음수가 되어 식별이 완료된다. 이 방식은 태그에서 난수를 발생하여 태그 집단을 둘로 나누는 과정에서 휴면 슬롯이 여러 번 발생할 수 있으나, 확률적으로 많지 않다.

쿼리트리기법은 태그의 응답에 따라 리더가 전송하는 쿼리를 결정되는 방식으로 알고리즘은 다음과 같다.

Step1. 리더는 쿼리가 저장된 큐에서 k-비트 길이의 쿼리를 가져와 태그에게 브로드케스트 한다.

Step2. 각각의 태그들은 리더로부터 전송받은 쿼리를 자신의 태그 ID와 비트 순서대로 비교한다. 만약 자신의 태그 ID와 수신한 쿼리의 모든 비트가 일치하는 경우 자신의 태그 ID를 리더에게 전송하고, 일치하지 않으면 전송하지 않는다.

Setp3. 리더는 쿼리후 태그의 응답에 따라 세 가지 상태, 즉 휴면(Idle) 상태, 식별(Success) 상태, 충돌(Collision) 상태에 따라 상태별 동작이 이뤄진다.

- ▷ 휴면상태일 경우, 쿼리가 저장된 큐에서 다른 쿼리를 가져와 다시 태그에게 쿼리한다.
- ▷ 인식상태일 경우, 태그 ID를 메모리에 저장하고, 다시 알고리즘을 반복한다.
- ▷ 충돌상태일 경우, 리더는 이전에 태그에게 전송했던 쿼리에 0과 1을 추가하여 새로운 쿼리를 만들어 이를 큐에 저장하고 식별 프로세스를 다시 수행한다. 여기서, 태그들에서 수신되는 메시지 간에 1개라도 충돌되면 전체 태그에 대해 브로드케이트하여 전송횟수가 늘어나기 때문에 충돌시 전송횟수를 최적화하여 태그 식별 속도를 개선해야 한다. (그림 1)은 4개의 비트를 가진 4개의 태그를 식별하는 쿼리트리 알고리즘의 식별단계를 예시로 나타낸 것이다.

1. step은 전체 식별의 횟수를 의미하며, 2. reader는 리더에서 발생하는 쿼리문자열이며, 3.response는 태그의 ID와 비교결과로 c는 충돌, i는 휴면, 숫자는 성공적인 식별을 의

| step | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Reader | ε | 0 | 000 | 001 | 1 | 10 | 11 |
| Tag answer | **** | 0*1 | 1 | 1 | *00 | 00 | 00 |
| Tag1(0001) | 0001 | 0001 | 0001 | | | | |
| Tag2(0011) | 0011 | 0011 | | 0011 | | | |
| Tag3(1000) | 1000 | | | | 1000 | 1000 | |
| Tag4(1100) | 1100 | | | | 1100 | | 1100 |
| query={ε} | | | | | | | |
| | 0 | 000 | 001 | 1 | 10 | 11 | |
| | 000 | 001 | 1 | | 11 | | |
| | | 1 | | | | | |
| | | | | | | | |
| | | | | | | | |
| Memory | | | 0001 | 0001 | 0001 | 0001 | 0001 |
| | | | | 0011 | 0011 | 0011 | 0011 |
| | | | | | | 1000 | 1000 |
| | | | | | | | 1100 |

(그림 1) Query Tree 알고리즘 동작 방법

MSB LSB

| 구 분 | Header | Filter Value | Partition | Company Prefix | Item Reference | Serial Number |
|---|---|---|---|---|---|---|
| 비트 크기 | 8 | 3 | 3 | 20-24 | 24-4 | 38 |

(그림 2) SGTIN-96 코드

미한다. 4. Tagn은 태그 고유 ID이며, 5. query는 리더로부터 태그로 쿼리 문자열이 보관된 큐이며, 6. memory는 태그 아디의 식별작업 완료후에 리더 메모리에 저장되는 태그 ID 값이 된다. 기존 쿼리 큐일 경우 9회 사이클과 8회의 쿼리 생성이 요구된다.

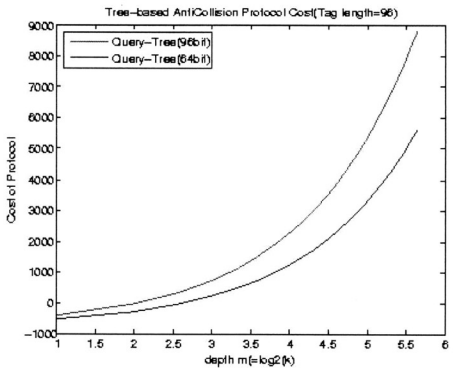
쿼리트리 알고리즘을 (그림 2)의 EPCglobal의 태그 식별 코드 구조인 SGTIN-96 구조에 적용한다면, 하나의 태그 ID를 인식하기 위해서는 최소 96바이트의 비트열을 식별해야 된다.

따라서, 내부 노드에서 쿼리트리 사이클을 일반화하면, 식 (1)만큼의 충돌 사이클이 요구된다[3].

$$n+km-1 \tag{1}$$

이때, n은 식별해야하는 태그의 수가 되고, km은 휴무 사이클의 갯수가 되며, 충돌발생시 태그를 찾는 경로수를 몇 개로 하느냐에 따라 m이 정해진다. 즉, 바이너리 트리일 경우는 m이 2가된다.

Step 4. 모든 태그 아디를 식별할 때까지, Step 1을 반복한다. 이러한 알고리즘의 수행은 큐에 있는 모든 쿼리를 수행한 후 종료하게 된다. 식별 프로세서 수행 초기에는 큐의 상태가 null로 시작되며, 리더는 null에 해당하는 ε값을 태그로 전송하여 충돌이 생길 때마다 0과 1을 추가함으로서 쿼리 깊이가 점점 길어지게 된다.




(그림 3) 쿼리트리 에너지 소비량

(그림 3)은 쿼리트리의 쿼리 깊이에 따른 에너지 소비량의 상관관계를 알 수 있다[4].

Query 트리 알고리즘 에너지소비량 쿼리 깊이가 길어질수록 쿼리트리의 에너지 소비량이 증가함을 알 수 있다. 일반적인 페시브(Passive) 태그일 경우, 리더로부터 에너지를 공급받기 때문에 에너지 소비량을 최적화하는 것은 매우 중요한 RFID 성능인자가 된다. 따라서, 에너지 소비량을 최소화하기위해서는 리더의 쿼리를 최적화하고, 식별 프로세서를 단축해야 한다.

지금까지 살펴본 쿼리트리 알고리즘에서 식별 단계 및 쿼리 깊이를 단축해야할 필요가 있음을 확인했다.

## 3. 차등식별 알고리즘 제안

본 장에서는 기존 방식에 대한 고찰을 바탕으로 개선 알고리즘을 제안하고자 한다. 제안 알고리즘의 개선점은 태그 ID 식별 단계를 최소화하고, 상황별 태그 ID 식별 범위를 차등 적용함으로써 프로토콜 구현 비용을 최소화하고자 한다.

본 논문에서는 태그 ID 식별 단계를 최적화하기 위해 식별하고자하는 태그 그룹의 성격에 따라 차등 식별 알고리즘을 제안한다. 즉, 식별 태그가 그림 5와 같이 식별할 태그가 0001, 0011, 1000, 1100이라고 했을 때, 최상위 비트를 기본으로 시작비트가 0인 비트 계열을 먼저 식별하고, 식별할 0-계열 태그가 없을 경우 1-계열 태그의 식별을 수행하는 방법이다. 즉, 동일 계열 Depth-First방식으로 식별한다. 또한, 제안 방식은 개선된 쿼리트리방식[5]과 같이 3비트 단위의 쿼리로 처리하고, 초기 쿼리전송시 초기값은 Null값이 아닌 "0"값을 전송함으로 사이클를 줄였다. 제안하는 차등식별 알고리즘의 처리 절차는 다음과 같다.

Step 1. 초기 큐를 "0"상태로 초기화하고 모든 태그에 브로드캐스트한다. 회신되는 태그신호를 받아 초기 리더가 식별할 태그군을 선정한다. 만약, "0" 쿼리에 대해 휴면상태이면 "1"을 쿼리한다.

Step 2. "0" 쿼리에 대해 충돌이 있으면, 0-계열 태그를 2개 생성하여 큐에 저장한다. 반복 쿼리를 통해 0-계열 태그를 모두 인식하고, 충돌이 생기면 다시 2개의 쿼리를 생성한다.

Step 3. 0-계열 쿼리의 충돌횟수만큼 식별이 완료되면 "1" 쿼리를 전송하여 충돌횟수 파악을 통해 1-계열 태그수를 확인하고, 0-계열 쿼리와 동일방법으로 모두 식별 루틴을 반복하며 질의 문자열이 보관된 큐에 더 이상이 쿼리가 없으면 식별을 종료한다.

(그림 4)는 차등 식별 알고리즘의 동작 방법을 나타내고 있다.

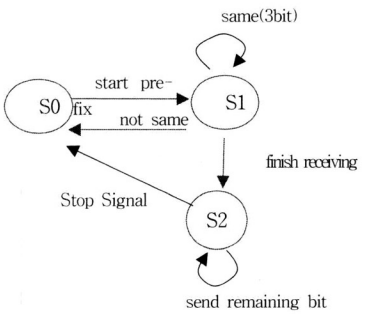
| 1.step | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 2.reader | 0 | 000 | 001 | 100 | 100 | 110 |
| 3.response | * | 1 | 1 | * | 1 | 1 |
| 4.Tag1(0001) | 0 | 0001 | | | | |
| Tag2(0011) | 0 | | 0011 | | | |
| Tag3(1000) | | | | 1000 | 1000 | |
| Tag4(1100) | | | | 1100 | | 1100 |
| 5. query={0} | | | | | | |
| | 000 | 001 | 1 | 100 | 110 | |
| | 001 | | | 110 | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| Memory | | 0001 | 0001 | 0001 | 0001 | 0001 |
| | | | 0011 | 0011 | 0011 | 0011 |
| | | | | 1000 | 1000 | 1000 |
| | | | | | | 1100 |

(그림 4) 차등 식별 알고리즘 동작 방법

제안 알고리즘은 특히, 동종계열 태그를 식별할 경우 효과적이며, 이종계열 태그를 식별할 경우에 비해 소요시간이 최대 50%로 효율적이다.

차등 식별 알고리즘의 상태 다이어그램은 (그림 5)와 같다.

기존 제안 기법[5]과 다른 점은 S0단계에서 prefix를 검사할 때, 최초 식별시 "0"을 쿼리하고 S1단계에서 충돌이 없으면 "1"을 쿼리하여 충돌이 발생하는 그룹부터 최종노드까지 3비트 단위로 비교하여 식별한다.




S0 - 초기상태 S1 - 수신상태 S2 - 송신상태 S4-휴면단계

(그림 5) 제안 알고리즘 상태 다이어그램

송신상태 S2에서 충돌이 발생하여 초기상태 S0로 전이할 때는, 제안된 개선 방식과 같이 리더에서 태그로 stop신호를 발생하여 충돌이후의 비트는 전송하지 않도록 차단하여 추가 충돌을 막은 다음 3비트 단위의 쿼리를 재전송하도록 한다.

## 4. 성능 평가

본 장에서는 제안한 차등식별 알고리즘을 이용한 충돌방기 기법의 성능 평가를 동일 조건상에 정량적 평가와 수학적 해석법으로 수행하였다.

정량적 방법은 (그림 2)와 (그림 3)에서 설명한 4개의 3비트 사이즈 태그를 통해 전체 알고리즘 동작수, 충돌횟수를 통해 성능을 비교하였다.

수학적인 성능평가는 프로토콜에 소요되는 사이클 수를 이용하여 쿼리트리 프로토콜에 대한 제안 프로토콜과 성능을 비교하였다[5].

### 4.1 정량적 평가

기존 쿼리트리 방식과 제안방식의 성능 비교는 전송 태그는 Tag1에서 4까지이며, 이종 태그 식별시 적용하는 태그 아디는 0001, 0011, 1000, 1100 인 동일 조건에서 수행한다.

성능 평가 인자는 식별 단계수, 충돌횟수로 하며, 이들 인자는 곧 총 식별시간이 된다. 성능확인은 다음 <표 1>과 같다. 이때 식별단계 소요시간 인자는 Ts, 충동횟수에 따른 소요시간인자를 Tc로 했을 경우 각각의 총 식별시간을 구할 수 있었다.

조건에서 소요시간 인자(Ts, Tc)값을 1을 설정할 경우 식별시 약 17%의 식별단계를 개선하였고, 충동횟수는 150%의 향상을 확인하였다. 그리고, 초기 쿼리 전송시 태그 ID가 균등한 배치일 경우 쿼리트리방식이 차등식별방식보다 충돌수가 2배 이상임을 확인할 수 있다.

〈표 1〉 프로토콜 성능 비교

| 성능인자 | 쿼리트리 | 차등식별 |
|---|---|---|
| 식별단계 | 7회 | 6회 |
| 충돌횟수 | 5회 | 2회 |
| 초기충돌횟수 | 4회 | 2회 |
| 총식별 시간 | 9Ts+4Tc | 8Ts+2Tc |

### 4.2 수학적 평가

제안 알고리즘의 성능평가를 위해 [4]에서 제안하는 식(2)와 같은 성능 평가 인자로 태그 아디식별에 소요되는 전체 사이클의 횟수로 한다.

$$Cyc_{query} = \sum_{i=0}^{m-1}(n+3)2^i + (n+5)2^m \qquad (2)$$
$$= (k(2n+8) - (n+3))$$

식(2)에서 개별 태그에 식별을 위해서는 최종 태그 식별 노드에 따라 타입-Ⅰ,Ⅱ로 나눈다. 타입-Ⅰ은 최종 노드 식별 이전단계에 소요되는 사이클 수는 한 개 비트씩 전송함

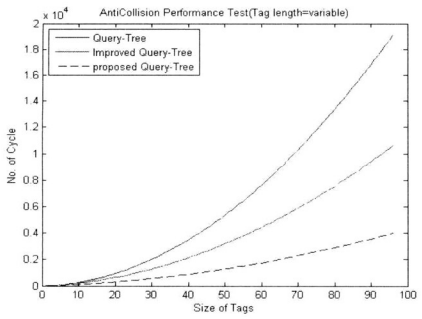
으로 최소 태그 크기(n)에 송수신 사이클이 필요하고, 태그의 시작 비트 및 종료비트 전후에 NULL 비트를 추가(2사이클)에 다음 태그 인식을 위한 wait 사이클(1사이클)을 합해 n+3사이클이 소요되어 이진트리의 depth 크기만큼 확장하고, 타입-Ⅱ는 최종 노드 인식에 필요한 사이클로 타입-Ⅰ에 2개의 태그 저장 사이틀(2사이클)을 추가한 n+5개의 사이클이 필요하다.

개선된 쿼리트리 알고리즘[5]에서는 타입-Ⅰ에서 소요 사이클을 i + 5로 제안하였다. 이때, i는 수신신호수이자 노드 레벨수를 의미한다. 5는 2개의 Null 사이클과 3개의 회신 사이클 수를 합을 의미한다. 이를 정리하면, 식(3)과 같이 사이클 수를 개선을 제안하였다. 또한, 송수신 필요한 신호 비트수를 3비트로 제안하고, 충돌이 발생하면, 리더는 충돌을 확인하기 위한 1클럭과 태그에게 stop신호를 발생하기 위해 각각 1개 클럭, 2개 클럭을 발생하고, 이로인해 태그는 충돌 비트 이후의 2비트를 수신한다. 이러한 방법으로 타입-Ⅰ에서는 (i+5) 사이클을 타입-Ⅱ에서는 쿼리트리 알고리즘과 동일 수의 클럭이 필요하다.

$$Cyc_{improved} = \sum_{i=0}^{m-1}(i+5)2^i + (n+5)2^m = k(n+m+8)-3 \quad (3)$$

본 논문에서 제안하는 차등식별 알고리즘은 타입-Ⅰ에서는 초기 null값 대신 "0"을 전송하여 충돌횟수를 줄이면서 시작태그 비트에 따라 0또는 1계열로 구분한다. 시작비트가 0인 태그들이 있어 충돌이 생기면 태그로 stop신호를 전송하고, 충돌 확인을 위한 추가쿼리를 전송하지 않아 매 노드마다 사이클을 1회 감소한다. 또한, 쿼리 전송단위를 3비트 단위로 전송하면 쿼리트리 알고리즘에 대해서는 n/3 사이클이 소요되고, 개선된 쿼리트리 알고리즘에 대해서는 i/3개의 노드수 만큼의 수신 사이클이 발생한다.

(그림 6)은 제안 알고리즘과 기존 알고리즘을 비교한 그래프로 태그수의 변화에 따른 사이클 수를 나타낸 것이며, 성능평가의 전제조건으로 태그 내 별도의 충돌방지용 플러그비트를 두고, 플러그비트는 3의 배수 비트로 구성한다. 제안 기법의 사이클 수를 정리하여 새로운 사이클 수를 도출하는 식(4)와 같이 도출하였다. 따라서, 제안기법은 유사한 태그가 군집되어 있는 환경인 물류창고 및 위치추격을 위한 목적에 용이하다.




(그림 6) 제안 알고리즘과 기존방식의 성능 비교

$$Cyc_{propsed} = \sum_{i=0}^{m-1}\{(i/3+4)2^i + (n+4)2^m\} - 1 = \frac{k(m+3n+22)-13}{3} \quad (4)$$

## 5. 결론 및 향후 과제

RFID태그의 사용범위가 확대되고 있는 상황에서 다중태그간의 충돌로 인한 태그 전력소비 문제와 식별 시간문제의 개선을 위해 많은 연구가 있었다. 본 논문에서는 리더를 통해 다중 태그를 식별하기 위해 효과적인 방법으로 동일개열의 태그를 우선 식별하고, 이종 개열의 태그를 식별하는 차등식별 충동방지 기법을 제안하였다.

제안한 충돌방지 기법은 동일 실험조건에서 기존 쿼리트리 알고리즘에 비해 17%의 식별단계 개선과 150%의 충돌 개선을 확인할 수 있었다. 제한 기법을 통해, 식별단계 및 충돌 회수를 최적화하여 RFID 시스템 전체의 식별 성능 및 충돌 문제를 개선을 기대하며, 향후, RFID 태그 정보보호를 위한 태그 ID 보안과 충돌방지 기법을 결합한 RFID 정보보호 시스템으로 발전시키고자 한다.

## 참 고 문 헌

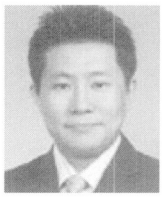
[1] K. Finkenzeller, RFID Handbook:' "Radio-Frequency Identification Fundamentals and Applications in Contactless Smart Cards and Identification," Second Edition, John Wiley, New York, 2003.

[2] D. R. Hush and C. Wood, "Analysis for Tree Algorithms for RFID Arbitration," ISIT, Cambridge, MA, USA, pp.107, 1998.

[3] C. Law, K. Lee, K. Y. Siu, "Efficient Memory-less Protocol for Tag Identification," In Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, Massachusetts, ISA, pp.75-84, 2000.

[4] F. Zhou, D. Jin, C. Huang and H. Min, "Optimizing the Power Consumption of Passive Electronic Tags For Anti-collision Schemes," In Proceedings of the 5th ASICON, Beijing, China, pp.1213-1217, 2003.

[5] F. Zhou, C. Chen, D. Jin, C. Huang and H. Min, "Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID Systems," ISLPED'04, Newport Brach, California, USA, pp.357-362, 2004.



## 김 성 진

e-mail : sjnetk@hotmail.com
1996년 서경대학교 컴퓨터과학과(학사)
1998년 경원대학교 대학원 전자계산학과 (공학석사)
2008년 경원대학교 대학원 전자계산학과 (공학박사)
1999년~2007년 서울현대전문학교 컴퓨터정보학과 교수
2007년~현 재 성남산업경제연구센터 선임연구원
관심분야 : 유비쿼터스 컴퓨팅, RFID, 정보보호

## 박 석 천

e-mail : scpark@kyungwon.ac.kr
1977년 고려대학교전자공학과(학사)
1982년 고려대학교 대학원 컴퓨터공학과 (공학석사)
1989년 고려대학교 대학원 컴퓨터공학 (공학박사)
1979년~1985년 금성통신연구소
1991년~1992년 University of California, Irvine Post Doc.
1988년~현 재 경원대학교 소프트웨어학부 정교수
관심분야 : 모바일 통신, 유비쿼터스 컴퓨팅, 정보보안, USN