

# RFID 시스템에서의 트리 기반 메모리리스 충돌방지 알고리즘에 관한 연구

권 성 호\* · 홍 원 기\*\* · 이 용 두\*\* · 김 희 철\*\*

## 요 약

RFID(Radio Frequency Identification) 기술은 RF 신호를 사용하여 물품에 부착된 전자태그를 식별하는 비접촉 기술이다. 식별 영역 내에 다수의 태그가 존재할 경우에 요구되는 다중 태그 식별 문제는 RFID 기술 중에서도 핵심 이슈이며 이 문제는 충돌방지 알고리즘을 통하여 해결할 수 있다. 하지만 기존 알고리즘들은 구현의 복잡성과 낮은 성능이 문제로 제기되고 있다. 본 논문에서는 이러한 문제를 해결할 수 있는 트리 기반 메모리리스 충돌방지 알고리즘인 충돌 추적(Collision Tracking) 트리 알고리즘을 제안하고 성능평가 결과를 제시한다. 성능평가 결과 제안하는 알고리즘이 초당 749개의 태그를 식별하면서 기존 트리 기반 메모리리스 알고리즘인 트리-워킹 알고리즘과 쿼리 트리 알고리즘에 비해 각각 약 49배와 2.4배의 성능향상을 확인할 수 있었다.

## A Study on the Tree based Memoryless Anti-Collision Algorithm for RFID Systems

Chenghao Quan\* · Wonkee Hong\*\* · Yongdoo Lee\*\* · Hiecheol Kim\*\*

## ABSTRACT

RFID(Radio Frequency Identification) is a technology that automatically identifies objects containing the electronic tags by using radio wave. The multi-tag identification problem is the core issue in the RFID and could be resolved by the anti-collision algorithm. However, most of the existing anti-collision algorithms have a problem of heavy implementation cost and low performance. In this paper, we propose a new tree based memoryless anti-collision algorithm called a *collision tracking tree* algorithm and presents its performance evaluation results obtained by simulation. The *Collision Tracking Tree* algorithm proves itself the capability of an identification rate of 749 tags per second and the performance evaluation results also show that the proposed algorithm outperforms the other two existing tree-based memoryless algorithms, i.e., the tree-walking algorithm and the query tree algorithm about 49 and 2.4 times respectively.

**키워드** : RFID, 전자 태그(Electronic Tag), 다중 태그 식별(Multi-tag Identification), 충돌방지 알고리즘(Anti-collision Algorithm), 성능 평가(Performance Evaluation)

### 1. 서 론

최근 주목을 받고 있는 RFID(Radio Frequency Identification) 기술은 RF 신호를 사용하여 물품에 부착된 전자태그(tag)를 식별하는 비접촉식 기술로서 인터넷의 지속적인 성장과 전자태그의 저가격화, 식별코드의 표준화에 힘입어 다양한 산업분야에서 실용화가 가능한 기술로서 전망되고 있다. RFID 기술은 물리공간(atoms)의 다양한 객체들을 가상공간(bits)에 연결시켜 인터넷에 새로운 차원의 확장을 제공하는 RFID/USN(Ubiquitous Sensor Network)의 구축에 대한 가시적인 비전을 제공하며 모든 공간의 사물이 지능화

되고 언제 어디서나 제한 없는 접속이 이루어지는 유비쿼터스 환경 구현에 있어 필수불가결한 기술로서 주목받고 있다 [1-3]. 이러한 RFID 기술은 유통/물류뿐만 아니라 식품, 의료/약품, 도로/교통, 우정, 문화, 생산자동화, 소방/방재, 금융, 환경, 정보유통 등 다양한 응용분야에 적용이 가능하며 [4, 5] 2004년 7월, IDTechEx사의 Smart Label Analyst에 발표된 Frost and Sullivan사의 조사결과에 따르면 RFID 세계시장은 2003년 이후 지수 함수적으로 증가해 2005년에 50억불, 2008년에 200억불, 2015년에는 4000억불에 이를 것으로 전망하고 있다[6].

RFID 기술의 중요성에 대한 인식이 최근 전 세계적으로 급속히 확산됨에 따라 RFID 핵심/원천기술 확보 및 RFID 산업의 국가 경쟁력 확보를 위하여 선진 각국에서는 국가

\* 준 회원 : 대구대학교 대학원 정보통신공학과

\*\* 정 회원 : 대구대학교 정보통신공학부 교수

논문접수 : 2004년 8월 4일, 심사완료 : 2004년 10월 19일

적 차원의 노력이 진행되고 있다. 하지만 이러한 RFID 기술의 적용과 확산을 위해서는 태그의 저가격-저전력-초소형화 문제, 사용자의 보안 및 프라이버시 문제, 태그 식별자의 코드 표준화 문제 그리고 다중 태그 식별 문제 등의 난제들을 우선 해결하여야 한다. 특히, 다중 태그 식별 문제는 리더기(reader)의 식별영역 내에 다수의 태그가 존재할 경우 태그의 정보를 충돌 없이 전송받아 식별해야하는 태그와 리더기사이의 다대일 통신 문제로 정의되며 기존의 바코드, 교통카드, 무선 결제 시스템에서의 태그와 리더기사이의 일대일 통신 문제와는 구별된다. 이러한 다중 태그 식별 문제는 충돌방지(anti-collision) 알고리즘을 통하여 해결할 수 있다[7].

현재, 다양한 충돌방지 알고리즘들이 제시되고 있으며 이러한 알고리즘은 크게 트리 기반 알고리즘과 슬롯 알고리즘 기반 알고리즘으로 구분된다. 슬롯 알고리즘은 확률에 수학적 기초를 두고 있기 때문에 확률적(probabilistic) 알고리즘이라고 불리며 태그 식별 완전성(completeness)에 있어서 그 제한점을 갖는다. 반면, 트리 기반 알고리즘은 식별과정 자체가 예측가능 하여 확률적 알고리즘과 구분하여 결정적(deterministic) 알고리즘이라고 부른다. 이는 다시 메모리형 알고리즘과 메모리리스(memoryless) 알고리즘으로 구분된다. 메모리형 알고리즘은 태그 식별과정에서 태그의 식별 상태를 관리해야하는 부담을 안고 있다는 제한점을 가지며 이는 저가격-저전력-초소형화 문제를 해결하는데 가장 큰 문제점이 되고 있다. 이와 반대로 메모리리스 알고리즘은 다중 태그 식별 문제를 해결하는 가장 적합한 알고리즘으로 평가받고 있지만 낮은 성능을 해결해야 하는 문제점을 갖고 있다. 이와 같이 충돌방지 알고리즘은 RFID 기술 중에서도 핵심적이며 필수적인 기술 요소임에도 불구하고 알고리즘 개발과 성능에 대한 연구는 아직 초기단계에 머물러 있다.

본 논문은 기존 메모리리스 알고리즘이 갖는 성능상의 문제점을 해결하기 위하여 새로운 충돌 추적(Collision Tracking) 트리 알고리즘을 제안하고 시뮬레이션을 통한 기존 메모리리스 알고리즘과의 성능평가 결과를 제시한다. 제안하는 충돌 추적 트리 알고리즘은 다수 개의 태그가 응답할 경우 충돌이 발생한 것으로 간주하는 충돌 감지(Collision Detecting) 방식을 사용하는 기존 트리 기반 메모리리스 알고리즘과 달리 수신된 정보에서 비트 단위로 충돌이 발생한 위치를 추적하는 충돌 추적 방식을 사용하여 태그 식별을 위한 질의-응답 횟수와 전송 비트수를 줄여 성능을 향상시킨다. 시뮬레이션을 통한 성능 평가결과 제안하는 알고리즘이 초당 749개의 태그를 식별하면서 기존 알고리즘들보다 평균 48.83배와 2.35배 성능이 향상됨을 확인할 수 있었다.

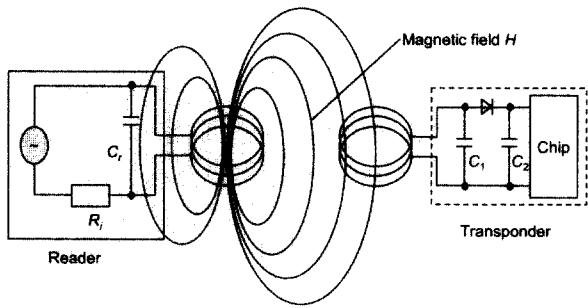
본 논문의 구성은 다음과 같다. 제2장에서는 RFID 기술에서 사용되는 결합 방식, 부호화 방식, 변조 기법을 간략히 살펴보고 충돌방지 알고리즘과 관련된 기존 연구를 소개한다. 제3장에서는 트리 기반 메모리리스 충돌방지 알고리즘인 트리-워킹 알고리즘과 퀴리 트리 알고리즘의 동작과정과 사례를 살펴본다. 제4장에서는 새로운 트리 기반의 메모리리스 충돌방지 알고리즘인 충돌 추적 트리 알고리즘을 제안하고 제5장에서는 기존 트리 기반 메모리리스 알고리즘과의 성능평가 수행 및 분석결과를 제시한다. 마지막으로 제6장에서 결론을 맺는다.

## 2. 관련 연구

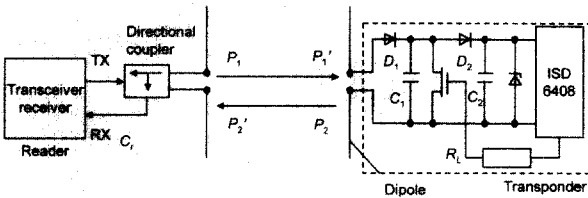
RFID 시스템은 사물에 부착된 태그와 태그의 정보를 읽을 수 있는 장치인 리더기로 구성된다. 태그의 정보를 수집하기 위해 리더기는 영역내의 모든 태그들로 질의를 보내며 해당 태그들은 응답의 형태로 자신의 식별자 정보를 리더기로 보내는 리더기와 태그사이의 통신이 이루어진다. 이때, 태그들은 에너지원의 보유 여부를 기준으로 에너지를 보유하고 있는 능동식(active) 태그와 리더기의 신호로부터 에너지를 공급받아 동작하는 수동식(passive) 태그로 구분할 수 있다. 본 논문에서는 하드웨어 비용이 적게 들고 전력 소모가 낮은 수동식 태그를 가정한다. 다음의 절들에서는 RFID 통신과 다중 태그 상에서 태그간 충돌을 방지하기 위한 알고리즘에 대해서 간략히 소개한다.

### 2.1 RFID 통신

태그와 리더기의 결합(coupling)방식은 크게 (그림 1)(a)의 상호결합(inductive coupling) 방식과 (그림 1)(b)의 전자기파 방식으로 구분된다. 그림과 같이, 상호결합 방식에서 태그는 리더기가 발생하는 자기장에서 에너지를 공급받아 안테나 회로에서 유도전압을 발생한 후 이를 정류하여 내부회로에 전원을 공급한다. 이때, 태그 안테나 회로의 부하를 조절하여 리더기 안테나에서의 전압을 변화시키는 방법으로 태그의 식별자를 리더기로 전송한다. 전자기파 방식에서는 전자기파의 반사특성을 이용하며 이 방식에서 리더기가 송출한 전력  $P_1$ 의 일부 전력  $P_1'$ 만 태그의 안테나에 전달된다. 전달된 전력  $P_1'$ 의 일부 전력은 흡수되어 정류를 거친 후 내부회로에 전원으로 공급되고 나머지 전력  $P_2$ 는 리더기로 반사된다. 반사된 전력의 일부인  $P_2'$ 만 리더기의 안테나로 전달된다. 여기서, 태그 안테나의 반사특성은 태그 안테나의 부하에 영향을 받는다. 따라서 태그는 안테나의 부하를 조절하여 리더기로 전달되는 전력을 변화시키는 방법으로 태그의 식별자를 리더기로 전송한다.

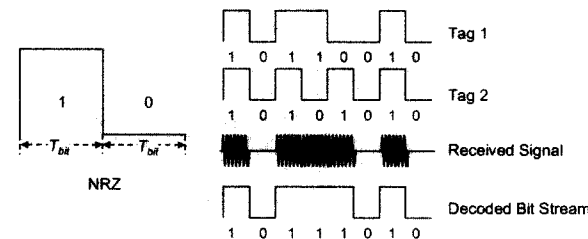


(a) 상호결합 방식

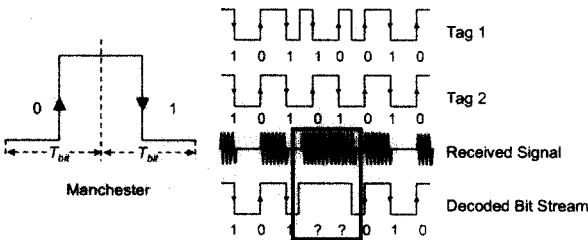


(b) 전자기과 방식

(그림 1) RFID 결합방식[8]



(a) NRZ 방식



(b) 맨체스터 방식

(그림 2) RFID 부호화 방식[8]

태그 식별자 정보는 전송을 위하여 디지털 신호로 부호화되고 이를 리더기로 전달하기 위해 아날로그 신호로 변조하는 과정이 요구된다. 대표적인 디지털 부호화 방식은 (그림 2)와 같이 NRZ 방식과 맨체스터 방식을 들 수 있다. NRZ 방식은 각 디지털 비트 구간마다 '+1' 혹은 '-1' 두 전위 상태 중 하나만 가지며 현재 전위 상태에 따라 '0' 혹은 '1'의 값이 결정된다. NRZ 방식은 디지털 신호로 부호화 시 비트 구간 내에 태그 식별자 비트 값들이 서로 다를 경우 '1'로 처리되기 때문에 태그들 간의 충돌 시점을 파악하기 어렵다.

반면에, 맨체스터 방식은 디지털 비트 구간의 중간 지점에서 항상 신호의 위상이 변화하며, 위상의 변화 형태에 따라 디지털 값이 결정되는 방식이다. 또한, 리더기에서 다중 태그들로부터 수신한 식별자 정보들을 디지털로 복호화할 때 비트 충돌이 발생했을 경우 해당 비트 구간이 하이임피던스 값을 갖게 되므로 태그들 간의 충돌을 쉽게 감지할 수 있다.

디지털 신호로 부호화된 태그 식별자 정보는 아날로그 전송로에서의 전송을 위하여 진폭 편이 변조(ASK, Amplitude Shift Keying), 주파수 편이 변조(FSK, Frequent Shift Keying) 혹은 위상 편이 변조(PSK, Phase Shift Keying) 등의 변조 기법을 이용하여 아날로그 신호로 변환한다. 진폭 편이 변조 기법은 디지털 신호의 전압변화에 따라 반송파의 진폭을 변화시키는 방식이며, 주파수 편이 변조는 두 개의 이진 값에 서로 다른 주파수를 적용하는 방식이다. 마지막으로 위상 편이 변조는 반송파의 위상을 이동시키는 방식이다. RFID 시스템에서는 비용 절감을 위해 비교적 하드웨어 구현이 용이한 진폭 편이 변조 기법이 주로 사용되고 있다.

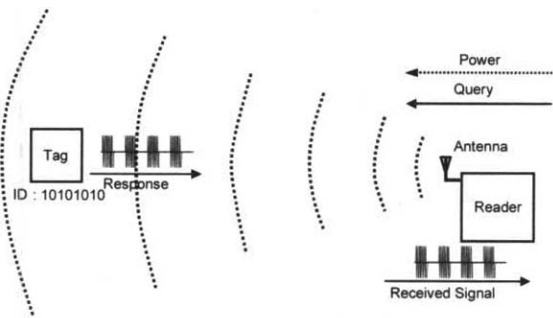
### 2.2 충돌방지 알고리즘

앞에서 언급했듯이 RFID 시스템에서 태그 식별은 (그림 3)(a)와 같이 리더기 식별영역 내에 한 개의 태그만 있을 경우 태그는 자신의 식별자에 적절한 부호화 방식과 변조 기법을 적용한 후 이를 리더기로 전송한다. 리더기에서 단일 태그 식별은 간단한 알고리즘을 통하여 구현할 수 있다. 이와 반대로 (그림 3)(b)와 같이 리더기의 식별영역 내에 다수 개의 태그가 존재할 경우에는 여러 개의 태그가 동시에 리더기의 질의에 응답하게 되므로 리더기에서 충돌이 발생하게 된다. 이러한 충돌 현상은 리더기로 하여금 정확한 태그 식별을 방해하는 원인이 된다. 특히, 대량의 물품을 실시간으로 식별해야하는 대규모 전자 물류시스템과 같은 응용에 적용하기 위해서는 다중 태그 식별 문제를 해결할 수 있는 충돌방지 알고리즘이 필수적으로 요구된다.

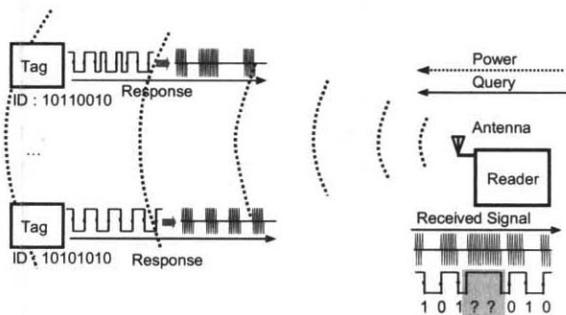
충돌방지 알고리즘과 관련된 선행연구로서 무선통신 환경에서의 채널 다중 접근에 관한 많은 연구가 이루어졌다 [9-11]. 하지만 RF 통신은 기존의 이러한 통신과는 전력공급, 연산능력, 태그 상호간의 존재 여부와 영역 내의 태그 개수를 알 수 없는 등 많은 점에서 다르다[7]. 다중 태그 식별을 위한 충돌방지 알고리즘은 (그림 4)와 같이 크게 트리 기반의 결정적 알고리즘과 슬롯 알로하 기반의 확률적 알고리즘으로 구분할 수 있다.

결정적 알고리즘은 이진 비트로 표현된 태그 식별자들로 이진 트리를 구성한 후 그 트리의 노드를 순회하며 태그 식별을 수행하는 방법으로서 태그 식별과정이 예측 가능하다는

특징을 갖고 있다. 이는 다시 메모리형 알고리즘과 메모리리스 알고리즘으로 분류할 수 있다. 메모리형 알고리즘에서 태그의 응답은 태그에 대한 질의와 태그의 현재 상태에 의하여 결정되므로 태그마다 상태 정보를 저장·관리해야 하는 부담을 안고 있다. 대표적인 알고리즘으로 분할 트리(splitting tree) 알고리즘[12]와 비트-중재(bit-arbitration) 알고리즘[13]을 들 수 있다. 반면 메모리리스 알고리즘에서 태그의 응답은 태그에 대한 질의에 의해서만 결정된다. 따라서 이러한 알고리즘을 적용할 경우 태그 구현이 간단하여 저가격-저전력-초소형화 문제를 해결할 수 있는 좋은 접근방법이 될 수 있다. 대표적인 알고리즘으로 트리-워킹(tree-walking) 알고리즘[14]과 쿼리 트리(query tree) 알고리즘[15]을 들 수 있다.



(a) 단일 태그 식별

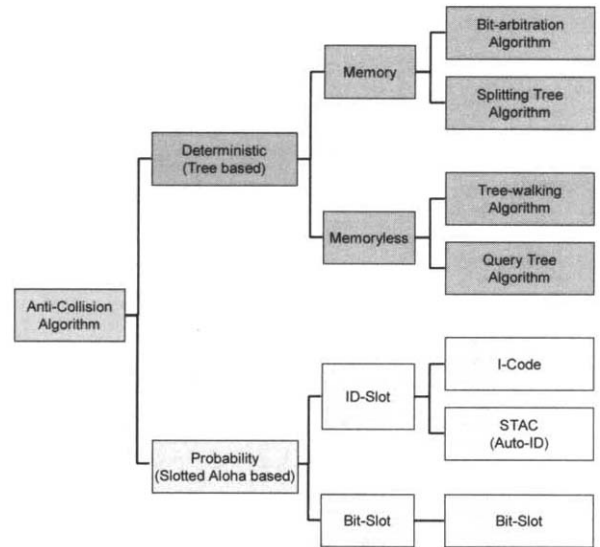


(b) 다중 태그 식별

(그림 3) 태그 식별

확률적 알고리즘은 알로하(aloha) 프로토콜 [16, 17]에 기반을 두고 있다. 리더기의 식별영역 내의 태그들은 주어진  $n$  개의 슬롯에서 태그의 정보를 전송할 슬롯을 임의로 선정하여 해당 식별자를 전송하게 되므로 슬롯간의 시간차에 의해 태그 충돌을 피한다. 그러나 식별영역 내의 태그 개수를 정확히 파악하기 어렵기 때문에 적절한 슬롯 개수와 종료 시점을 확률적으로 계산해야 한다. 따라서 확률에 근거한 종료시점의 결정으로 태그 식별의 완전성을 지원하지 못하며 또한 충돌이 발생한 슬롯의 재전송으로 인하여 태그 식별 시간에 있어서 높은 성능을 기대하기 어렵다는 단점을 갖는다. 확률적 알고리즘은 다시 ID-slot형 알고리즘과 Bit-slot

형 알고리즘으로 분류된다. ID-slot형 알고리즘은 각 슬롯에 태그의 식별자를 넣어 전송하는데 반하여 Bit-slot형 알고리즘은 각 태그마다 특수한 비트로 구성된 정보를 생성하여 슬롯에 채워 리더기로 전송한다. ID-slot형 알고리즘의 대표적인 알고리즘으로 I-Code 알고리즘[18], STAC(Slotted Terminating Adaptive Collection) 알고리즘[19]이 있고 Bit-slot형 알고리즘으로 Bit-slot 메커니즘을 사용하는 충돌방지 알고리즘[20]이 있다.



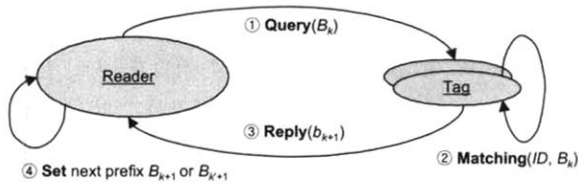
(그림 4) 충돌방지 알고리즘의 분류

### 3. 트리 기반 메모리리스 충돌방지 알고리즘

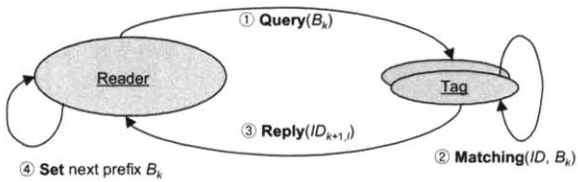
앞 장에서 언급했듯이 트리-워킹 알고리즘과 쿼리 트리 알고리즘은 태그의 상태정보를 유지할 필요가 없기 때문에 저가격에 태그 구현이 가능한 특징을 가지고 있다. (그림 5)와 같이 이 알고리즘들의 태그 식별은 반복적인 질의-응답 과정을 통하여 수행되며 리더기의 질의에 대한 태그 응답의 충돌 발생 결과에 따라 필요한 제어가 이루어진다. 단, 트리-워킹 알고리즘과 쿼리 트리 알고리즘은 리더기에 태그 식별과정 중 식별자의 일부 비트를 저장하기 위한 메모리를 요구하며 각각 스택과 큐의 구조를 갖는다.

트리-워킹 알고리즘은 (그림 5)(a)와같이 리더기가  $k$ -비트 길이의 프리픽스( $B_k$ )를 인자로 태그에 질의하는 것으로 시작한다. 영역 내의 태그들은 수신한 프리픽스를 자신의 식별자와 비교를 통해 매칭이 이루어질 경우 태그 식별자의  $(k+1)$ 번째 비트( $b_{k+1}$ )를 리더기로 전송한다. 이때, 리더기에서의 태그의 응답결과와 다음과 같은 두 가지 형태로 나타난다. 첫 번째, 모두 '0'이 수신되었거나 혹은 모두 '1'이 수신되었을 경우, 이 경우 리더기는 기존 프리픽스에 수신된 비트 값을 추가하여 새로운 프리픽스( $B_{k+1}$ )를 생성한다. 두

번째, '0'과 '1'이 동시에 수신된 경우, 이 경우는 충돌이 발생한 경우로서 충돌이 발생한 프리픽스를 스택에 저장하고, 동시에 '0'을 추가하여 새로운 프리픽스( $B_{k+1}$ )를 생성한다. 새로 생성된 프리픽스는 다음 번 질의-응답 과정에서 태그에 인자로 전달된다. 이러한 과정을 태그 식별자의 비트 길이만큼 수행하며 한 개의 태그가 식별될 때까지 반복한다. 한 개의 태그 식별이 이루어지면 스택에 저장된 프리픽스를 가져와 '1'을 추가하여 새로운 프리픽스( $B_{k+1}$ )를 생성하며, 이를 인자로 하여 새로운 질의-응답 과정을 수행한다. 스택이 비어있으면 전체 태그 식별 과정은 종료가 되며 이는 영역 내의 모든 태그가 식별되었음을 의미한다.



(a) 트리-워킹 알고리즘



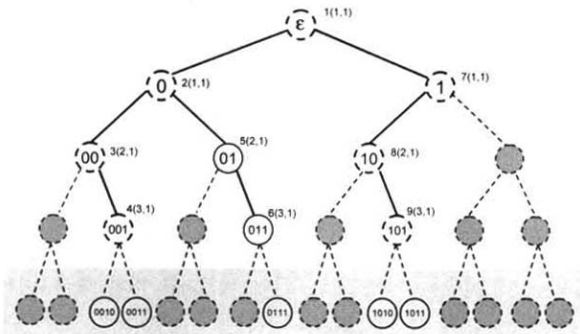
(b) 쿼리 트리 알고리즘

(그림 5) 트리 기반 메모리래스 충돌방지 알고리즘의 동작과정.

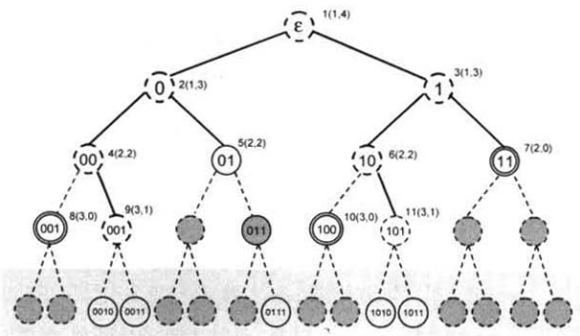
쿼리 트리 알고리즘의 경우, 리더기가 보낸  $k$ -비트 프리픽스( $B_k$ )와 매칭된 태그들은 자신의 식별자의  $(k+1)$ 번째 비트부터 마지막 비트까지의 나머지 비트들을 ( $ID_{k+1}$ ) 리더기로 전송한다(그림 7)(b). 이때 태그의 응답결과는 다음과 같이 두 가지 경우가 존재한다. 첫째, 오직 하나의 태그만 응답하였을 경우와 태그로부터의 응답이 없을 경우이다 이 경우는 큐에 저장된 프리픽스를 가져와 새로운 프리픽스( $B_k$ )를 생성한다. 둘째, 다수 개의 태그가 동시에 응답하면서 충돌이 발생할 경우로 이때는 기존 프리픽스에 '0'과 '1'을 추가하여 각각 큐에 저장한 후 큐에서 새로운 프리픽스( $B_k$ )를 가져와 다음 질의-응답 시 태그에 인자로 전달한다. 이러한 과정은 영역 내의 모든 태그가 식별될 때까지 반복된다. 큐가 비어있으면 전체 태그 식별 과정은 종료되며 이는 영역 내의 모든 태그가 식별되었음을 의미한다.

(그림 6)은 트리 기반 메모리래스 충돌방지 알고리즘의 사례를 보여준다. 그림에서 사선으로 표시된 노드는 실제 존재하지 않는 노드이다. 음영으로 표시된 부분에서 단선원 노드 안에 표시된 이진수치는 리더기의 영역 내에 존재하는

태그들의 식별자를 나타낸다. 즉, 리더기 영역 내에 '0010', '0011', '0111', '1010', '1011'의 식별자를 갖는 5개의 4비트 태그가 존재한다고 가정한다. 그 외의 각 노드들의 원 안에 표시된 이진 값들은 질의-응답과정에서 태그로 전달되는 인자, 즉, 프리픽스를 의미하며 ' $\epsilon$ '은 영역내의 모든 태그가 매칭되는 특별한 프리픽스 값이다. 각 노드의 프리픽스를 인자로 하여 태그로 질의/응답 시 리더기에서 충돌이 발생하면 점선 노드, 충돌이 발생하지 않으면 실선 노드, 응답이 없으면 이중선 노드로 표시하였다. 각 노드들의 오른쪽 상단에  $S(Q, R)$ 형태로 표시된 수치에서  $S$ 는 태그 식별이 이루어지는 순서를 나타내고  $Q$ 는 질의 비트 수로서 리더기로부터 태그로 전송된 프리픽스의 비트 길이를,  $R$ 은 응답 비트 수로서 태그로부터 리더기로 전송된 태그 식별자의 비트 길이를 의미한다. 그림에서 보듯이 트리-워킹 알고리즘은 깊이 우선 탐색의 순서로 쿼리 트리 알고리즘은 너비 우선 탐색으로 태그 식별이 이루어진다.



(a) 트리-워킹 알고리즘



(b) 쿼리 트리 알고리즘

(그림 6) 트리 기반 메모리래스 충돌방지 알고리즘의 사례

트리-워킹 알고리즘의 사례를 보여주는 (그림 6)(a)의 두 번째 질의-응답과정에서 리더기가 프리픽스 '0'을 인자로 태그에 질의하면 영역 내의 '0010', '0011', '0111'의 식별자를 가진 3개의 태그가 매칭이 되면서 각자 식별자의 프리픽스 다

음 비트인 '0', '0', '1'을 리더기로 전송한다. 이 과정에서  $Q$ 와  $R$ 은 각각 1이 되며 '0'과 '1'이 동시에 수신되므로 리더기에서는 충돌이 발생하게 된다. 충돌이 발생하면 리더기는 기존 프리픽스('0')를 스택에 저장하고 동시에 '0'을 추가하여 새로운 프리픽스('00')를 생성한 후 새로운 질의-응답과정(3번째)을 시작하게 된다. 3번째 질의-응답과정에서 영역 내의 '0010', '0011'의 식별자를 가진 두개의 태그가 매칭이 되면서 각자 식별자의 프리픽스 다음 비트인 '1'을 리더기로 전송한다. 이 경우 리더기는 모두 '1'을 수신하므로 충돌이 발생하지 않고 기존 프리픽스에 '1'을 추가하여 새로운 프리픽스('001')를 생성한다. 4번째 질의-응답 과정에서 프리픽스 '001'에 매칭되는 두개의 식별자 '0010', '0011'이 리더기로 응답한다. 이때 '0'과 '1'이 동시에 수신되면서 충돌이 발생하나 프리픽스의 길이가 태그 식별자 비트 길이 보다 하나가 작은 3이므로 리더기는 단순히 영역 내에 '0010'과 '0011'의 식별자를 가진 두 개의 태그가 존재한다고 판단하게 된다. 따라서 이 경우는 기존 프리픽스를 스택에 저장할 필요가 없으며 다만 새로운 프리픽스를 생성하기 위하여 스택에 저장된 프리픽스를 가져온다. 이때, 프리픽스는 두 번째 과정에서 충돌이 발생하여 스택에 저장된 '0'이며 여기에 '1'을 추가하여 새로운 프리픽스 '01'을 생성하고 5번째 질의-응답과정을 시작한다. 이와 같이 영역 내의 모든 태그를 식별하기 위하여 9번의 질의-응답과정이 수행되며 이 과정에서 리더기와 태그 간에 전송된 비트 수는 질의 비트 수 18과 응답 비트 수 9의 총 합인 27임을 알 수 있다.

(그림 6)(b)는 쿼리 트리 알고리즘의 사례를 보여주고 있다. 질의-응답과정 중 4번째 단계에서 리더기가 프리픽스 '00'값을 인자로 태그에 질의하면 영역 내의 '0010', '0011'의 식별자를 가진 두개의 태그가 매칭이 되면서 각자 식별자의 프리픽스 다음 비트부터 식별자의 마지막 비트까지의 '10', '11'을 리더기로 전송한다. 이 과정에서  $Q$ 는 2이고  $R$ 은 2가 된다. 이때 리더기는 충돌을 인지하게 되고 기존 프리픽스('00')에 '0'과 '1'을 추가하여 각각 큐에 저장한 후 큐에서 새로운 프리픽스를 가져온다. 본 사례에서 큐에서 가져온 프리픽스는 '01'로서 이 프리픽스를 인자로 새로운 질의-응답과정(5번째)을 시작하게 된다. 이때, 영역 내의 '0111'의 식별자를 가진 태그가 매칭이 되면서 식별자의 프리픽스 다음 비트부터 식별자의 마지막 비트까지의 '11'을 리더기로 전송한다. 따라서, 충돌은 발생하지 않으며 리더기는 '0111'의 식별자를 가진 태그를 식별하게 된다. 영역 내의 나머지 태그들을 식별하기 위해 리더기는 큐에 저장된 프리픽스를 가져와 새로운 질의-응답 과정을 반복한다. 쿼리 트리 알고리즘의 경우 모든 태그를 식별하기 위하여 그림에서 보듯이 총 11번

의 질의-응답과정이 수행되었으며 이 과정에서 질의 비트 수는 23이고 응답 비트 수는 18로서 모두 41비트가 전송되었음을 알 수 있다.

위에서 살펴보았듯이 트리-워킹 알고리즘에서는 리더기의 질의에 대하여 영역 내의 태그들은 항상 한 비트의 태그 식별자 정보만 전송한다. 따라서 영역 내에 오직 한 개의 태그만 존재함에도 불구하고 태그의 식별자 마지막 비트까지 질의-응답과정을 반복 수행하게 되며 이는 태그의 식별자 비트 길이가 늘어남에 따라 성능이 저하되는 요인이 된다. 특히, 현재 표준화가 진행 중에 있는 96비트, 128비트, 256비트의 태그 식별자 [21]를 고려하고 영역 내의 태그의 개수가 일반적으로 최대 4,000~8,000(12~13비트 길이)개 인 것을 감안하면 트리-워킹 알고리즘의 성능 저하는 심각한 문제가 된다. 쿼리 트리 알고리즘에서는 태그의 식별자 정보 전체를 전송하는 방법을 통해 이러한 문제를 해결하고 있지만 무응답 노드의 발생과 트리-워킹 알고리즘에 비해 많은 충돌 노드의 개수가 문제가 된다. 쿼리 트리 알고리즘에서 이러한 문제가 발생하는 근본적인 이유는 태그가 리더기로 전체 식별자 정보를 전송함에도 불구하고 리더기가 수신된 식별자 정보를 전혀 이용하지 않고 단순히 충돌 발생여부만 결정하기 때문이다.

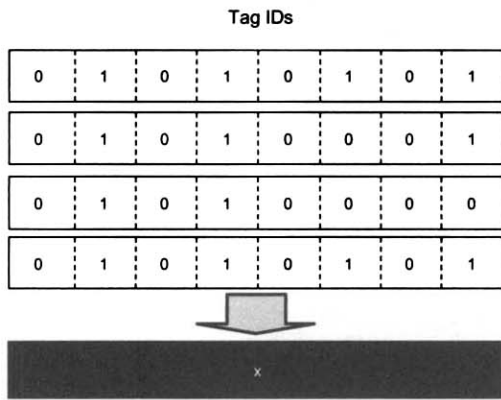
#### 4. 충돌 추적 트리 알고리즘

본 논문에서는 기존 트리-워킹 알고리즘과 쿼리 트리 알고리즘의 성능저하 문제를 개선하기 위해 태그로부터 전송받은 식별자 정보들로부터 충돌 지점을 파악해 태그 식별 속도를 개선하기 위한 충돌 추적 트리 알고리즘을 제안한다.

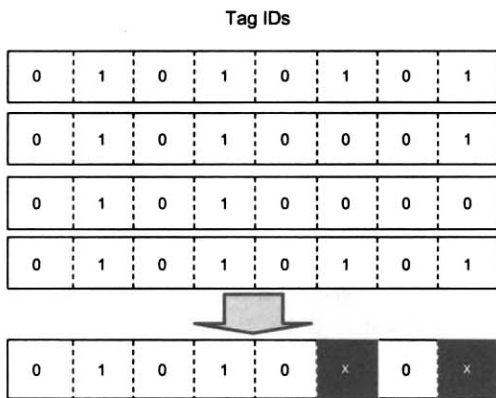
본 논문에서 제안하는 충돌 추적(Collision Tracking) 방식과 대비하여 트리-워킹 알고리즘과 쿼리 트리 알고리즘은 충돌 감지(Collision Detecting) 방식이라 볼 수 있는데, 이는 질의-응답 과정에서 태그로부터 받은 정보에서 단지 충돌 발생 여부만을 인지하기 때문이다. 반면에 충돌 추적 방식은 태그 정보들로부터 충돌 지점을 찾아내 질의-응답 횟수와 전송 비트 수를 획기적으로 줄여 줄 수 있는 방법이다. 이는 리더기가 수신한 태그 식별자 정보에서 충돌이 발생한 비트의 위치를 정확히 추적한 후 충돌 발생 이전의 비트 내 용을 태그 식별에 사용하는 원리를 바탕으로 한다.

(그림 7)은 리더기가 수신한 정보에 대한 처리 방식을 충돌 감지 방식과 충돌 추적 방식으로 비교하여 보여준다. 그림에서 4개의 태그들은 8비트 길이의 식별자를 가지며 리더기의 질의에 의하여 각자의 태그 식별자를 리더기로 전송한다. (그림 7)(a)는 다수개의 태그가 동시에 응답할 경우 단순

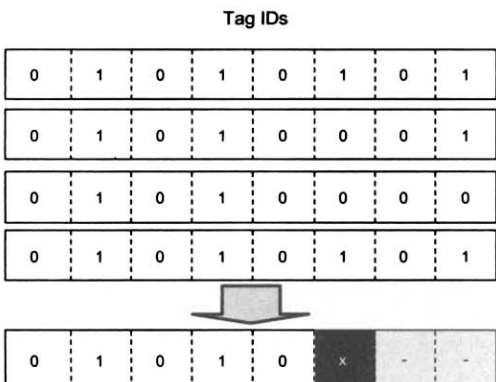
히 충돌이 발생한 것으로 간주하는 충돌 감지 방식이다. 반면 충돌 추적 방식은 (그림 7)(b)와 같이 충돌이 발생한 6번째 비트와 8번째 비트의 위치를 모두 정확히 추적하는 완전 충돌 추적(CCT, Complete Collision Tracking) 방식과 (그림 7)(c)와 같이 처음 충돌이 발생한 6번째 비트의 위치만 추적하는 처음 위치 충돌 추적(FCT, First Collision Tracking) 방식으로 구분된다. 본 논문에서는 구현 비용에 비해 성능이 우수한 FCT 방식을 위주로 설명을 한다.



(a) Collision Detecting



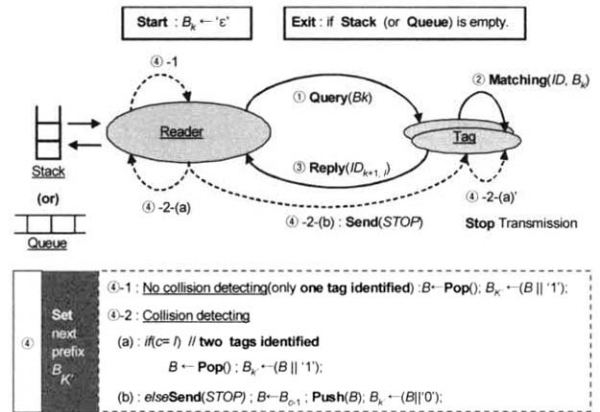
(b) Complete Collision Tracking



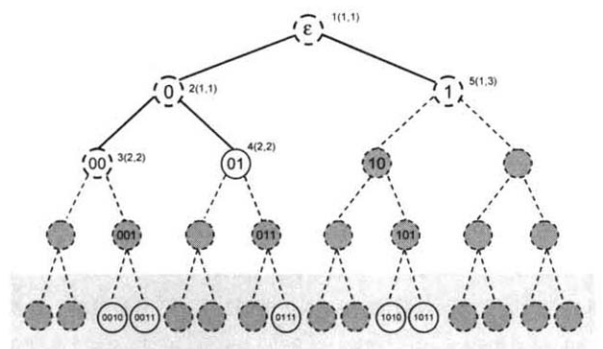
(c) First Collision Tracking

(그림 7) 충돌 감지 방식과 충돌 추적 방식

(그림 8)은 제안하는 충돌 추적 트리 알고리즘의 동작과 정과 사례를 보여준다.



(a) 동작과정



(b) 사례

(그림 8) 충돌 추적 트리 알고리즘

(그림 8)(a)와 같이 리더기는  $k$ -비트 길이의 프리픽스( $B_k$ )를 인자로 태그에 질의하며 영역 내의 태그들은 수신한 프리픽스를 자신의 식별자와 비교를 통해 매칭이 이루어질 경우 태그 식별자의  $(k+1)$ 번째 비트부터 식별자의 마지막 비트까지( $ID_{k+1,l}$ )를 순서대로 리더기로 전송한다. 리더기는 태그가 전송한 식별자 정보를 수신함과 동시에 수신된 비트의 충돌 발생 여부를 검사한다. 이때, '0'과 '1'이 동시에 수신되어 충돌이 발생하면 나머지 비트의 수신을 중지하고 태그로 식별자 전송 중지 명령을 보내고 모두 '0' 혹은 '1'이 수신되면 나머지 비트의 수신을 계속한다. 모든 비트에서 충돌 없이 식별자 정보의 마지막 비트까지 수신되면 한 개의 태그가 식별된다. 충돌이 발생했을 경우, 기존 프리픽스에 단순히 '0' 혹은 '1'을 추가하는 트리-워킹 알고리즘이나 퀴리 트리 알고리즘과 달리 충돌 없이 수신된 비트들을 모두 추가하여 새로운 프리픽스를 생성하고 이를 저장하여 다음 번 질의-응답과정에 인자로 사용한다. 알고리즘의 구현에 따라 질의-응답 과정에서 태그 식별자 일부 비트를 리더기에 저

〈표 1〉 사례를 이용한 트리 기반 메모리리스 알고리즘의 비교.

알 고 리 즘	질의-응답 횟수	전송 비트 수			노드 수		
		질의 비트 수(Q)	응답 비트 수(R)	전체 비트 수	충돌 발생 노드 수	충돌 미발생 노드 수	무응답 노드 수
트리-워킹 알고리즘	9	18	9	27	4	5	-
쿼리 트리 알고리즘	11	23	18	41	7	1	3
제안하는 충돌 추적 트리 알고리즘	5	7	9	16	4	1	-

장하는 메모리 구조는 스택이나 큐를 사용할 수 있으며 사용하는 메모리 구조에 따라 깊이 우선 탐색이나 너비 우선 탐색이 된다. 한 개의 태그가 식별되면 영역 내의 나머지 태그를 식별하기 위하여 스택이나 큐에서 새로운 프리픽스를 가져오게 된다. 이러한 과정은 영역 내의 모든 태그가 식별될 때까지 반복된다. 스택이나 큐가 비어있으면 전체 태그 식별 과정은 종료되며 이는 영역내의 모든 태그가 식별되었음을 의미한다. 알고리즘의 구현에 있어서 식별자 전송 중지 명령은 하드웨어 혹은 소프트웨어 측면에서 여러 가지로 고려할 수 있으며 상황에 따라 전송이 되지 않을 수도 있다. 이와 관련된 내용은 본 논문의 범위를 벗어나므로 여기서는 생략한다. 이 외에도 (그림 8)(a)는 스택을 사용한 경우 프리픽스를 생성하는 과정도 보여주고 있다.

(그림 8)(b)는 스택을 사용한 충돌 추적 트리 알고리즘의 사례를 보여주고 있다. 응답 비트 수  $R$ 의 계산에 있어서 식별자 전송 중지 명령에 따른 오버헤드는 무시하였다. 쿼리 트리 알고리즘과 비교했을 경우 1, 2번째 질의-응답 과정은 유사하며 네 번째 과정은 동일하다. 세 번째 질의-응답과정에서 리더기가 프리픽스 '00'을 인자로 태그에 질의하면 영역 내의 '0010', '0011'의 식별자를 가진 2개의 태그가 매칭이 되면서 각자 식별자의 프리픽스 다음 비트부터 식별자의 마지막 비트까지의 비트들을 순서대로 리더기로 전송한다. 리더기에 첫 번째 수신된 비트는 모두 '1'로서 충돌이 발생하지 않으므로 나머지의 비트의 수신을 계속한다. 두 번째 수신된 비트에서 '0'과 '1'이 동시에 수신되어 충돌이 발생하였으므로 리더기는 태그로 식별자 전송 중지 명령을 보내야 되지만 이 경우 충돌이 발생한 비트의 위치가 태그 식별자의 마지막 비트로서 단순히 영역 내에 '0010'과 '0011'의 식별자를 가진 두 개의 태그가 존재한다고 판단하게 되면서 실제로는 전송하지 않는다. 이 과정에서  $Q$ 와  $R$ 은 각각 2이다. 두 개의 태그가 동시에 식별되고 리더기는 스택에서 '01'을 가져와 새로운 프리픽스를 생성하고 4번째 질의-응답 과정을 시작하게 된다. 한편, 다섯 번째 질의-응답 과정에서는 리더기가 프리픽스 '1'값을 인자로 태그에 질의하면 영역 내의 '1010', '0011'의 식별자를 가진 2개의 태그가 매칭이 되면서 각자 식별자의 프리픽스 다음 비트부터 식별자의 마지막 비트까지의 '010', '011'을 비트 순서대

로 리더기로 전송한다. 리더기에 첫 번째와 두 번째 수신된 비트는 모두 '0'과 '1'로서 충돌이 발생하지 않으므로 나머지 비트의 수신을 계속한다. 세 번째 수신된 비트에서 충돌이 발생하였으므로 리더기는 태그로 식별자 전송 중지 명령을 보내야 한다. 그러나 충돌이 발생한 비트의 위치가 태그 식별자의 마지막 비트로서 단순히 영역 내에 '1010'과 '1011'의 식별자를 가진 두 개의 태그가 존재한다고 판단하게 되면서 실제로는 전송하지 않는다. 이와 같이 영역 내의 모든 태그를 식별하기 위하여 그림에서 보듯이 총 5번의 질의-응답과정이 수행되었으며 이 과정에서 모두 16비트가 전송되었음을 알 수 있다.

〈표 1〉은 본 논문에서 설명한 사례를 이용하여 세 가지 트리 기반 메모리리스 알고리즘을 비교하여 보여주고 있다. 제안하는 충돌 추적 트리 알고리즘은 질의-응답 횟수, 전송 비트 수, 방문 노드 수 등 모든 성능요소에서 효과적임을 알 수 있다.

## 5. 성능평가

본 절에서는 제안하는 충돌 추적 트리 알고리즘의 성능 향상을 검증하기 위하여 시뮬레이션을 통한 성능평가를 실시하였다. 시뮬레이션 환경은 C언어를 기반으로 설계하였으며 사용된 태그 식별자의 길이( $L$ )는 EPCGlobal [21]에서 표준화가 진행 중인 96비트로 가정하였다. 위에서 살펴본 트리 기반 메모리리스 알고리즘인 트리-워킹 알고리즘, 쿼리 트리 알고리즘과의 성능 비교를 수행하였으며 각각에 대하여 영역 내의 태그 식별자의 수( $n$ )를 2에서부터 매번 2배씩  $65,536(2^{16})$ 개까지 증가시키면서 충돌 발생 노드 수, 충돌 미발생 노드 수, 무응답 노드 수, 질의 횟수, 응답 횟수, 질의 비트 수( $Q$ ), 응답 비트 수( $R$ )등을 산출하였다. 성능평가의 정확도를 높이기 위하여 성능평가에 사용된 수치는 10회 반복을 통해 얻은 값들을 평균한 값으로 하였다. 초당 식별 태그 개수 산출 시 사용된 비트 전송율은 80Kbps이고 질의 명령은 8비트, 무응답 및 충돌 발생 시, 검출과 처리에 필요한 시간은 각각 3비트 시간으로 계산하였다[19, 22, 23].

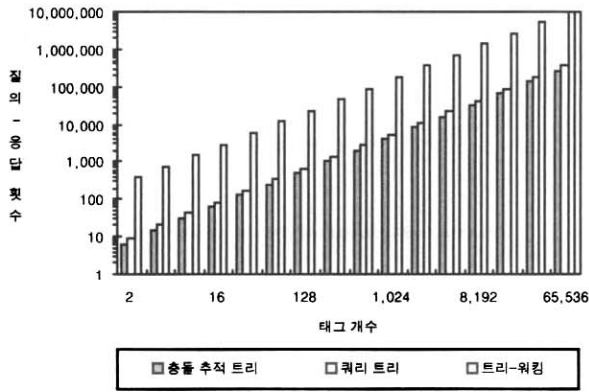
본 장에서는 우선, 질의-응답 횟수와 전송 비트 수 측면에서 제안하는 알고리즘과 기존 트리-워킹 알고리즘, 쿼리 트



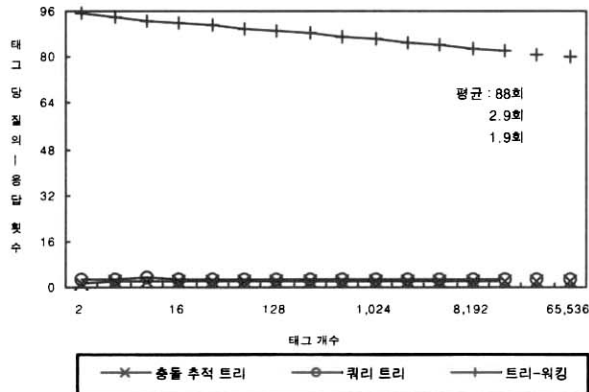
리 알고리즘을 비교·분석하고 마지막으로 초당 태그 식별 개수 비교를 통해 제안하는 알고리즘의 성능향상을 확인한다.

5.1 질의-응답 횟수 및 전송 비트 수

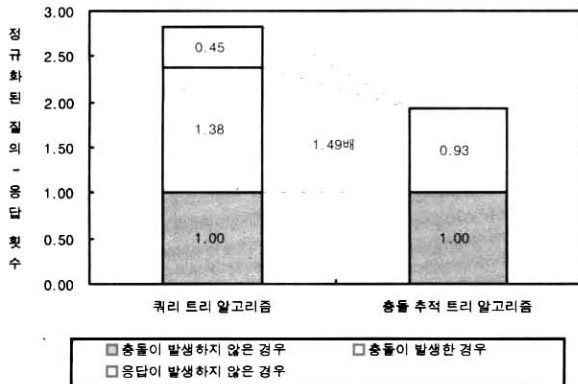
질의-응답 횟수는 영역 내의 태그 식별을 위한 리더기와 태그들 간의 통신 횟수로서 한번의 질의-응답 과정을 1회로



(a) 질의-응답 횟수



(b) 태그 당 질의-응답 횟수



(c) 정규화한 질의-응답 횟수 점유율 비교

(그림 9) 트리 기반 메모리래스 충돌방지 알고리즘의 질의-응답 횟수 비교

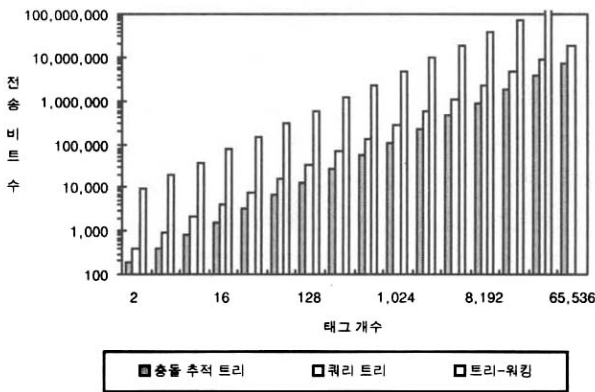
정의한다. 단, 무응답 질의가 존재하여 질의 횟수가 응답 횟수보다 많은 쿼리 트리 알고리즘의 경우 질의 회수를 질의-응답 횟수로 간주한다.

(그림 9)(a)는 식별 영역 내 태그 개수의 증가에 따른 각 알고리즘들의 질의-응답 횟수를 보여주고 있으며 그림과 같이 제안하는 충돌 추적 트리 알고리즘이 가장 우수한 것으로 나타나고 있으며 제안하는 알고리즘이 트리-워킹 알고리즘과 쿼리 트리 알고리즘에 비해 각각 평균 45.59배, 1.36배 성능이 향상됨을 확인할 수 있다. (그림 9)(b)는 각 알고리즘별 태그 당 질의-응답 횟수를 보여주고 있다. 트리-워킹 알고리즘의 경우 영역 내의 96비트 태그를 식별하는데 태그 당 질의-응답 횟수가 평균 88회가 소요됨으로써 다른 알고리즘에 비해 훨씬 높게 나타나고 있다. 이는 앞에서 설명하였듯이 트리-워킹 알고리즘은 질의-응답 과정이 태그 식별자의 비트별로 반복 수행되기 때문이다. 따라서, 태그 당 질의-응답 횟수는 태그의 식별자 비트 길이에 비례하여 증가하게 된다. 이와 같이 과도한 질의-응답 횟수는 트리-워킹 알고리즘의 성능을 낮추는 요인이 된다. (그림 9)(c)는 쿼리 트리 알고리즘과 충돌 추적 트리 알고리즘에서 충돌이 발생하지 않은 질의-응답 횟수를 1로 정규화하여 각각의 특성 분포를 나타낸 그래프이다. 그림과 같이 충돌 추적 트리 알고리즘이 쿼리 트리 알고리즘에 비해 충돌이 발생한 질의-응답 횟수를 1.49배 줄이고 무응답이 발생한 질의-응답 횟수를 제거함으로써 전체적인 질의-응답 횟수에서 쿼리 트리 알고리즘보다 우수하게 나타나고 있다.

시물레이션에서 산출한 전송 비트 수는 앞 절에서 언급한 질의 비트 수(Q)와 응답 비트 수(R)의 합으로서 질의 시 인자로 사용한 프리픽스의 비트 길이와 응답 시 리더기로 전송한 식별자 정보의 비트 길이의 합을 의미한다.

(그림 10)(a)는 식별 영역 내 태그 개수의 증가에 따른 각 알고리즘들의 전송 비트 수를 비교하여 보여주고 있으며 그림과 같이 제안하는 충돌 추적 트리 알고리즘이 가장 우수함을 볼 수 있으며 제안하는 알고리즘이 트리-워킹 알고리즘과 쿼리 트리 알고리즘에 비해 각각 평균 44.62배, 2.47배 낮은 전송량을 보여주고 있다. (그림 10)(b)는 각 알고리즘별 태그 당 질의, 응답 비트 수 및 백분율을 비교하여 보여줌과 동시에 트리-워킹 알고리즘과 쿼리 트리 알고리즘 대비 제안하는 알고리즘의 성능향상도 보여주고 있다. 앞에서 설명하였듯이 트리-워킹 알고리즘은 태그 식별자의 마지막 비트까지 프리픽스 정보를 매번 한 비트씩 증가하여 질의-응답 과정을 반복하기 때문에 한 비트씩 전송하는 응답에 비해 질의에 의해 전송되는 비트 수가 많다. 이 또한 트리-워킹 알고리즘의 성능이 낮은 주요한 원인임을 알 수 있으며 그림과 같이 태그 당 응답 비트 수가 평균 88비트인 것

에 비해 질의 비트 수가 4,521비트로서 전체의 약 98%를 차지하는 것을 볼 수 있다. 쿼리 트리 알고리즘의 경우 태그 당 응답 비트 수가 평균 231비트인데 반해 제안하는 알고리즘은 87비트로서 이는 태그 식별자 정보 전체를 전송하여 태그를 식별하는 쿼리 트리 알고리즘에 비해 제안하는 알고리즘은 충돌이 발생하면 전송을 중지하기 때문이다. 또한 제안하는 알고리즘은 태그 당 질의 비트 수, 응답 비트 수 측면에서 모두 기존 알고리즘에 비해 우수한 것을 확인할 수 있었으며 질의-응답 횟수의 시간 복잡도는  $(n-1)$ 이고 전송 비트 수 시간 복잡도는 실험구간 내에서 평균  $(1.08 \times n \times k)$ 인 것으로 나타났다.



(a) 전송 비트 수

알고리즘	트리-워킹 알고리즘	쿼리 트리 알고리즘	충돌 추적 트리 알고리즘	충돌 추적 트리 알고리즘의 성능향상	
				트리-워킹 알고리즘 대비	쿼리 트리 알고리즘 대비
질의	비트 수	4,521	26	17	269
	백분율	98.10%	10.11%	16.14%	
응답	비트 수	88	231	87	1.00
	백분율	1.90%	89.89%	83.86%	

(b) 질의, 응답 비트 수 비교

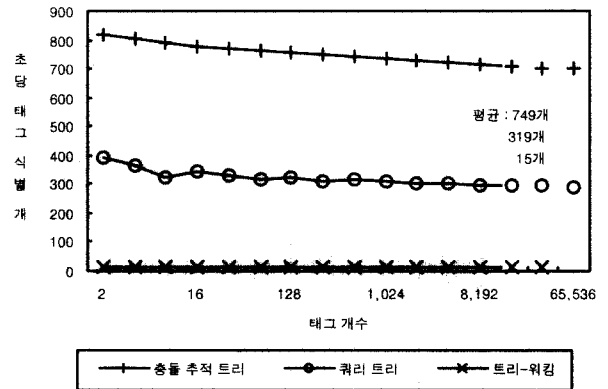
(그림 10) 트리 기반 메모리리스 충돌방지 알고리즘의 전송 비트 수 비교

### 5.2 초당 태그 식별 개수

초당 태그 식별 개수는 충돌 방지 알고리즘의 성능을 비교할 수 있는 궁극적인 척도로서 영역 내의 태그들을 식별하기 위해 필요한 리더기와 태그들 간의 통신 오버헤드를 포함한 실제 전송된 총 비트 수를 계산하여 이를 전송율로 나눔으로써 계산하였다.

(그림 11)은 식별 영역 내 태그 개수의 증가에 따른 각 알고리즘들의 초당 태그 식별 개수를 비교하여 보여주고 있으며 그림과 같이 제안하는 충돌 추적 트리 알고리즘이 초당

749개의 태그를 식별함으로써 쿼리 트리 알고리즘의 319개, 트리-워킹 알고리즘의 15개보다 훨씬 많은 태그를 식별할 수 있으며 제안하는 알고리즘이 트리-워킹 알고리즘과 쿼리 트리 알고리즘에 비해 각각 평균 48.83배, 2.35배 성능이 향상됨을 확인할 수 있다.



(그림 11) 트리 기반 메모리리스 충돌방지 알고리즘의 초당 태그 식별 개수 비교

## 6. 결 론

본 논문에서는 기존 트리 기반 메모리리스 충돌방지 알고리즘들의 성능저하 문제를 개선한 충돌 추적 트리 알고리즘을 제안하고 시뮬레이션을 통하여 얻은 성능평가 결과를 제시하였다. 제안하는 트리 기반 메모리리스 알고리즘은 수신된 정보에서 비트 단위로 충돌이 발생한 위치를 추적하여 충돌 발생 이전의 비트 내용을 태그 식별에 사용함으로써 질의-응답 횟수, 전송 비트 수를 줄여 성능을 향상시켰다. 성능평가를 통해 제안하는 알고리즘이 초당 749개의 태그를 식별할 수 있으며 기존 트리-워킹 알고리즘과 쿼리 트리 알고리즘에 비해 각각 48.83배와 2.35배 성능이 향상된 것으로 나타났다.

향후, 충돌 추적을 효과적으로 지원하기 위한 태그 및 리더기의 신호 전달 체계, 하드웨어 구현에 대한 연구와 충돌방지 알고리즘을 적용한 RFID 시스템 프로토콜 설계에 대한 연구를 진행할 계획이다. 또한 트리 기반 메모리리스 알고리즘뿐만 아니라 슬롯 알로하 기반 확률적 알고리즘을 포함한 충돌방지 알고리즘에 대한 성능평가와 함께 능동형 태그를 위한 트리 기반 메모리형 충돌방지 알고리즘에 대한 연구도 요구된다.

## 참 고 문 헌

[1] M. Weiser, "The Computer of the 21st Century," *Scientific*

- America*, Vol.265, No.3, pp.94-104, September, 1991.
- [2] R. Want, K. Fishik, A. Gujar and B. Harrison, "Bridging Physical and Virtual Worlds with Electronic Tags," *In ACM Conference on Human Factors in Computing Systems (CHI99)*, Pittsburgh, PA., pp.370-377, May, 1999.
- [3] Want, R., Russell, D. M., "Ubiquitous Electronic Tagging," *Distributed Systems Online*, Vol.1, No.1, September, 2000.
- [4] F. Mattern, "The Vision and Foundations of Ubiquitous Computing," *Upgrade*, Vol.2, No.5, pp.2-6, October, 2001.
- [5] K. Romer, T., "Schoch. Infrastructure Concepts for Tag-Based Ubiquitous Computing Applications," *Workshop on Concepts and Models for Ubiquitous Computing at Ubi-comp 2002*, Goteborg, Sweden, September, 2002.
- [6] IDTechEx Ltd., <http://www.idtechex.com>.
- [7] 권성호, 김희철, "EPC 네트워크를 위한 다중 RFID 태그 식별 알고리즘의 분석", *한국인터넷정보학회*, Vol.4, No.4, pp. 27-37, December, 2003.
- [8] Klaus Finkenzeller, "Fundamentals and Applications in Contactless Smart Cards and Identification, Second Edition," *John Wiley & Sons Inc.*, April, 2003.
- [9] J. L. Massey, "Collision Resolution Algorithms and Random-Access Communications," In G. Longa, editor, *Multi-User Communication Systems*, Springer-Verlag, New York, pp.73-173, 1981.
- [10] P. Mathys and P. Flajolet, "Q-ary Collision Resolution Algorithms in Random Access Systems with Free or Blocked Channel Access," *IEEE Transactions on Information Theory*, IT, Vol.31, No.2, pp.217-243, March, 1985.
- [11] A. J. E. M. Janssen, M. J. de Jong, "Analysis of Contention Tree Algorithms," *IEEE Trans. Inform. Theory*, Vol.46, No.9, pp.2163-2172, September, 2000.
- [12] Hush, Don R. and Wood, Cliff, "Analysis of Tree Algorithms for RFID Arbitration," *In IEEE International Symposium on Information Theory*, IEEE, pp.107-, 1998.
- [13] Jacomet M., Ehsam A., Gehrig U., "Contactless identification device with anti-collision algorithm," *IEEE Computer Society, CCCC'99, Conference on Circuits, Systems, Computers and Communications*, Athens., July, 1999.
- [14] A. Juels, R. Rivest and M. Szydlo, "The Blocker Tag : Selective Blocking of RFID Tags for Consumer Privacy," *Proceedings of the 10th ACM conference on Computer and communication security*, ISBN : 1-58113-738-9, pp. 103-111, 2003.
- [15] Law, Ching, Lee, Kayi and Siu, Kai-Yeung, "Efficient Memoryless protocol for Tag Identification," *In Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM, pp.75-84, August, 2000.
- [16] Frits C. Schoute, "Control of ALOHA signalling in a Mobile Radio Trunking System," *In International Conference on Radio Spectrum Conservation Techniques*, IEEE, pp.38-42, 1980.
- [17] Jeffrey E. Wieselthier, Anthony Ephremides and Larry A. Michaels, "An Exact Analysis and Performance Evaluation of Framed ALOHA with Capture," *IEEE Transactions on Communications*, COM, Vol.37, No.2, pp.125-137, 1989.
- [18] Vogt, H., "Efficient Object Identification with Passive RFID Tags," *In International Conference on Pervasive Computing, LNCS*. Springer-Verlag, 2002.
- [19] Auto-ID Center, "13.56MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification : Candidate Recommendation, Version 1.0.0," *Auto-ID Center*, May, 2003.
- [20] Changsoon Kim, Kyunglang Park, Hiecheol Kim, Shindug Kim, "An Efficient Stochastic Anti-collision Algorithm using Bit-Slot Mechanism," *PDP '2004*, July, 2004.
- [21] EPCGlobal Inc, <http://www.epcglobalinc.org>.
- [22] Auto-ID Center, "Protocol Specification for 900MHz RFID Tag," *Auto-ID Center*, February, 2003.
- [23] Auto-ID Center, "860MHz~930MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical Communication Interface Specification Candidate Recommendation, Version 1.0.1," *Auto-ID Center*, November, 2002.

### 권 성 호



e-mail : chquan@daegu.ac.kr

1992년 (중)연변대학교 전자공학과 졸업  
2001년 대구대학교 정보통신공학과 공학석사

2001년~현재 대구대학교 정보통신공학과 박사과정

관심분야 : RFID, EPC 네트워크, 지능형 컴퓨팅

### 홍 원 기



e-mail : wkhong@daegu.ac.kr

1995년 연세대학교 컴퓨터과학과(공학사)  
1997년 연세대학교 컴퓨터과학과 공학석사  
2001년 연세대학교 컴퓨터과학과 공학박사  
2004년~현재 대구대학교 정보통신공학부 전임강사

관심분야 : RFID, 컴퓨터 구조, 임베디드 시스템, 분산 컴퓨팅

## 이 용 두



e-mail : ydlee@daegu.ac.kr

1975년 한국항공대학교 통신학과(공학사)

1982년 영남대학교 전자공학과 공학석사

1995년 한국항공대학교 전자공학과 공학  
박사

1982년~현재 대구대학교 정보통신공학부  
교수

관심분야 : RFID, 임베디드 시스템, 지능형 컴퓨팅

## 김 희 철



e-mail : hckim@daegu.ac.kr

1983년 연세대학교 전자공학과(공학사)

1991년 University of Southern California  
(Computer Eng. M.S.)

1996년 University of Southern California  
(Computer Eng. Ph.D.)

1997년~현재 대구대학교 정보통신공학부 부교수

관심분야 : RFID, 병렬처리, 지능형 컴퓨팅