

RFID 시스템에서 불연속 m-bits 인식을 위한 개선된 가변비트 M-ary QT 충돌해소 알고리즘

김관웅¹ · 김변곤^{2*}

Improved variable bits M-ary QT conflict resolution algorithm for discrete m-bits recognition in RFID system

Kwan-woong Kim¹ · Byun-gon Kim^{2*}

¹Thunder Technology, Director in Digital Signal Processing Team, ChonJu 55028, Korea

²Department of Electronic Engineering, Kunsan National University, Gunsan 54150, Korea

요 약

RFID 시스템에서 리더는 자신의 인식 범위 내에 있는 태그의 ID를 식별하기 위해 각각의 태그에 쿼리 메시지를 전송한다. 다중 태그가 리더의 쿼리에 동시에 응답할 수 있기 때문에 응답 메시지의 충돌이 발생할 수 있으므로, 충돌을 중재하는 절차가 필수적이다. 이러한 절차를 충돌 해소알고리즘이라 하며 RFID 시스템에서 가장 핵심적인 기술이다. 본 논문에서는 맨체스터 코딩 기법을 기반으로 충돌비트의 위치 정보를 리더와 태그에서 이용할 수 있는 가변비트 M-ary QT 알고리즘을 제안하였다. 제안된 알고리즘은 불연속의 가변 비트를 인식할 수 있기 때문에 질의-응답의 cycle 수를 줄일 수 있다. 제안된 알고리즘의 컴퓨터 시뮬레이션 결과는 기존의 M-ary QT 기법에 비하여 질의-응답의 cycle 수, 인식 효율, 통신비용 관점에서 매우 우수한 성능을 보여주고 있다.

ABSTRACT

In the RFID system, the reader transmits a query message to tags in order to identify a unique ID of tags within its detection range. The procedures for arbitrating a collision is essential because the multiple tags can response simultaneously in the same to the query of the Reader. This procedure is known as conflict resolution algorithm and it is a key technology in the RFID system. In this paper, we proposed a variable bits M-ary QT algorithm based on Manchester coding techniques. The proposed algorithm use the location information of the collision bits in the reader and tags. The proposed algorithm can reduce the number of the query-response cycle because it is capable of recognizing discontinuous bits and a variable number of bits. From computer simulation, the proposed method give better performance than the conventional M-ary QT techniques in terms of response, recognition efficiency, communication overhead.

키워드 : RFID 시스템, 충돌 방지, M-ary 쿼리 트리, 맨체스터 코딩

Key word : RFID system, anti-collision, M-ary Query Tree, Manchester Coding

Received 19 September 2016, Revised 21 September 2016, Accepted 25 September 2016

* Corresponding Author Byun-Gon Kim(E-mail: bgkim@kunsan.ac.kr, Tel: +82-63-469-4730)

Department of Electronic Engineering, Kunsan National University, Gunsan 54150, Korea

Open Access <http://dx.doi.org/10.6109/jkice.2016.20.10.1887>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

RFID(Radio Frequency IDentification)는 태그에 부착된 IC칩과 무선을 통해 식품, 동물, 사물 등 다양한 개체의 정보를 관리할 수 있는 차세대 인식기술이다. RFID/USN(Ubiquitous Sensor Network)은 유비쿼터스 사회를 구현하는 핵심 인프라로서 무한한 성장 잠재력을 내포하고 있는 산업이며 향후 인터넷에 이어 산업 전반에 커다란 변혁을 가져올 수 있는 분야이다. RFID는 각종 물품에 소형 칩인 태그(tag)를 부착하고 사물의 정보와 주변 환경 정보를 판독·해독기능이 있는 판독기(reader)를 통해 인식하는 비접촉식 인식시스템이다[1]. RFID 리더의 가장 중요한 기능은 자신의 인식 범위 내에 있는 개별 제품들에 부착된 태그의 고유한 EPC(Electronic Product Code) 코드를 식별하는 것이다[1,2]. 식별된 제품의 EPC 코드는 해당 제품의 생산 정보나 유통 이력 등의 상세 정보를 사용자에게 제공한다.

RFID의 리더는 인식 범위 내의 태그를 식별하기 위하여 태그에 쿼리 메시지(query message)를 전송하고 쿼리 메시지를 수신한 태그는 쿼리 메시지에 대응하여 응답 메시지(response message)를 리더에 전송한다. 태그로부터의 응답메시지는 다수의 태그들이 동시에 응답할 수 있기 때문에 충돌이 발생할 수 있다. 응답 메시지의 충돌이 발생할 경우 리더에서 동시에 응답한 여러 개의 태그를 식별해야하는 문제가 발생하고 이를 해결하는 기술이 충돌 해소(conflict resolution algorithm) 알고리즘이다[3]. 태그 충돌 해소 기법은 RFID 기술이 다양한 산업 분야에 적용되기 위해서 해결되어야하는 핵심적인 기술 중의 하나이다.

충돌해소 알고리즘은 태그의 응답을 중재하는 방법에 따라 크게 확률적(probability) 알고리즘과 결정적(deterministic) 알고리즘으로 분류된다. 확률적 알고리즘은 알로하(aloha) 프로토콜에 기반을 두며 슬롯 간의 시간차를 이용하여 태그 식별을 수행한다[4]. 하지만 응답 슬롯을 태그가 임의(random)로 선택하기 때문에 태그 수가 많아지면 성능이 저하될 수 있으며 소위 태그 기아문제(tag starvation problem)로 인해 일정한 시간 안에 모든 태그를 인식하지 못 할 수도 있다. 이러한 확률적인 방법은 본질적으로 확률의 불확실성을 가지고 있기 때문에 리더의 전송범위 내에 있는 모든 태그들을 식별하는 것을 보장하지 못하는 단점이 있다.

결정적 알고리즘은 쿼리 트리(query tree) 프로토콜에 기반을 두고 있으며, 태그의 고유한 ID를 이용하여 응답 그룹을 점진적으로 나누어 가며 충돌을 해소한다. 이 과정에서 질의 트리를 형성하고 그 트리의 노드를 순회하며 태그 식별을 수행한다. 대표적인 트리 기반의 태그 충돌 해소 프로토콜로는 쿼리 트리 기법, 충돌 트리(collision tree), M-ary QT(Query Tree) 등이 있다[5].

본 논문에서는 M-ary QT 기법을 기반으로 맨체스터 코딩 기법을 이용하여 인식 비트 수(m)를 가변 할 수 있는 기법을 제안하였다. M-ary QT 기법은 연속적인 인식 비트 수의 인식 방식을 제공하지만 제안된 기법에서는 불연속적인 인식 비트 수의 인식 방법을 제공한다. 또한 제안된 기법에서는 맨체스터 코딩기법을 이용하여 충돌비트 위치 정보를 리더에서 응답 메시지를 분석할 때 뿐 만아니라 태그에서 쿼리 메시지를 분석할 때 도 이용하였다. 태그에서 쿼리 메시지의 충돌 비트 위치 정보를 이용하여 인식 비트 수를 가변 할 수 있으며 응답 메시지의 길이에 따라 결정하였다. 인식 비트 수가 크면 하나의 쿼리 메시지에 많은 수의 비트를 인식할 수 있지만 확장 비트 수 $M(2^m)$ 가 증가하여 응답 메시지의 크기가 커지게 된다. 응답 메시지의 크기가 크면 비트 에러 발생 확률이 증가하기 때문에 응답 메시지의 크기를 제한해야 한다. 따라서 제안된 알고리즘에서는 응답 메시지의 크기가 태그 ID의 전체 비트 수의 1.1배를 넘지 않는 범위 내에서 인식 비트 수를 가변하였다. 또한 다수의 쿼리 메시지를 하나의 쿼리 메시지로 통합할 수 있는 기법을 제안하여 질의-응답의 cycle 수를 줄여 효율을 향상 시켰으며, 컴퓨터 시뮬레이션을 이용하여 기존의 M-ary QT 기법보다 태그 식별 시간, 인식 효율성 및 통신비용 관점에서 우수한 성능을 보임을 확인할 수 있었다.

II. 관련 연구

QT(Query Tree) 알고리즘은 트리 기반의 대표적인 프로토콜로서 리더의 질의에 대응하여 특정 태그가 응답한다. 리더의 쿼리 메시지는 태그 ID의 일부인 prefix를 포함하고 있다. 쿼리 메시지를 수신한 태그는 수신한 prefix를 태그 ID의 prefix와 비교하여 일치할 경우에 prefix 비트를 제외한 태그의 나머지 비트로 구성된 응

답 메시지를 전송한다. 그러나, 다수의 태그가 동시에 응답하여 메시지 충돌이 발생할 경우 서버에서 태그를 인식할 수 없기 때문에 리더는 prefix의 길이를 1비트씩 늘려가면서 충돌을 일으킨 태그들을 두 부분 집합(subsets)으로 나누고 각각의 부분 집합에 대해 다시 쿼리 메시지를 보내서 식별하게 된다[6,7]. 그림 1은 QT 알고리즘의 태그 인식과정을 보여주는 예이다.

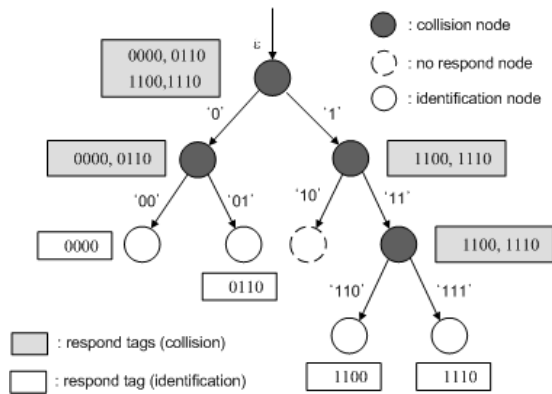


Fig. 1 Query tree algorithm

이와 같은 QT 기법은 리더의 전송범위 내에 존재하는 모든 태그의 식별을 보장한다. 그러나, 한 번의 쿼리 전송에 하나의 비트만 식별할 수 있고, 충돌 비트의 위치 정보를 알 수 없기 때문에 무응답 쿼리 메시지가 존재하여 식별하는 시간이 상당히 길어질 수 있다.

태그 응답에서 충돌 감지 기법은 충돌 여부만을 판단하지만 충돌 추적 기법은 비트 단위로 충돌 위치를 추적할 수 있다. 충돌 추적 기법(collision tracking)은 맨체스터 코딩을 사용하여 구현이 가능하다[1].

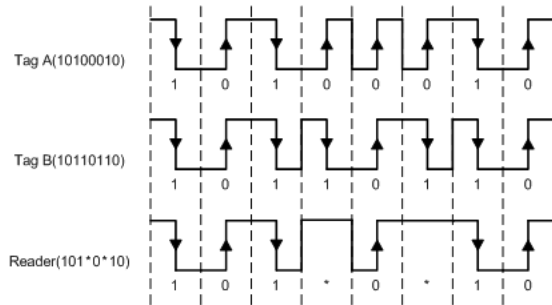


Fig. 2 Example of Manchester Coding

맨체스터 코딩에서 하나의 비트는 전압의 크기가 아니라 전압의 전이로 표현되고, 비트의 충돌이 발생하면 무전이 상태가 된다. 그림 2는 bit 4와 bit 6 구간에서 충돌이 발생하여 무전이 상태가 된 경우를 보여주고 있다. CT(Collision Tree) 알고리즘은 충돌 추적 기법을 사용하여 QT 알고리즘의 성능을 개선하였다[8]. 리더는 태그의 응답 메시지에서 충돌 비트의 위치를 인식할 수 있기 때문에 충돌이 발생하지 않은 prefix 비트를 다음 쿼리 메시지의 prefix에 포함시킬 수 있을 뿐만 아니라 무응답 노드가 발생하지 않기 때문에 QT 알고리즘의 성능을 획기적으로 개선시킬 수 있다.

M-ary QT 기법에서는 m 비트를 인식할 수 있는 알고리즘을 제안하였다. 매핑함수는 m-bits 스트링을 $M(2^m)$ 비트 스트링으로 변환하는 함수로서 표 1과 같은 변환 규칙을 이용하여 변환한다[9,10,11,12].

Table. 1 Mapping Table of 2-bits

2 bits	$4(2^2)$ bits
00	0001
01	0010
10	0100
11	1000

그림 3은 4개의 태그(00001010, 00110111, 10111001, 11001101)를 식별하는 2-ary QT 알고리즘의 예를 보여 준다. 초기에 리더가 'ε'(empty string)을 질의하면 4개의 모든 태그들이 응답하는데, 태그 ID prefix의 2비트를 매핑함수를 이용하여 변환하고, 태그 ID의 나머지 비트들을 추가하여 응답 메시지를 생성하여 맨체스터 코딩 기법으로 변환하여 전송한다. 모든 태그들이 응답하여 수신 메시지는 '**0* ****'이다. 수신 메시지의 처음 4비트 '**0*'는 확장 비트 영역(2^2)의 메시지이고 태그 ID의 나머지 부분에서의 충돌 정보는 '****'이다. 확장 비트 영역에서의 3개의 충돌 비트 각각은 표 1과 같은 역 매핑 함수를 이용하여 첫 번째 충돌 비트 '*00'는 '11', 두 번째 충돌 비트 '*0*0'는 '10', 세 번째 충돌 비트 '*00*'는 '00'으로 인식되어 다음 쿼리 메시지의 prefix에 확장된다. 두 번째 쿼리 메시지 '00'의 응답 메시지는 '**0* **1*'이다. 응답 메시지의 4비트 확장 영역 메시지 '*00*'를 분석하여 두 개의 인식 비트 '11'과 '00'를 구하여, 다음 쿼리 메시지의 prefix를

‘0011’과 ‘0000’으로 확장한다. 이와 같은 방식으로 모든 태그를 인식할 때까지 진행하며, M-ary QT 기법은 m 비트의 연속 비트를 인식할 수 있을 뿐만 아니라 유희노드를 없앨 수 있기 때문에 태그 인식 효율을 향상시킬 수 있다.

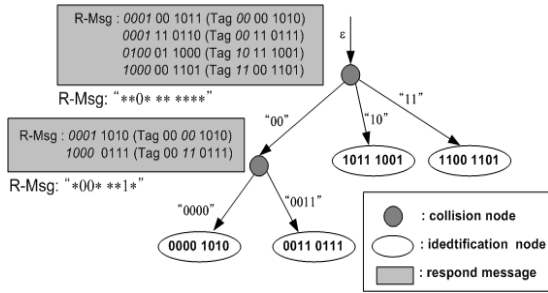


Fig. 3 M-ary Query tree algorithm

III. 제안된 M-ary QT 기법

본 논문에서는 M-ary QT 알고리즘을 기반으로 맨체스터 코딩 기법을 이용하여 불연속적인 위치의 비트를 인식할 수 있는 알고리즘을 제안하였다. 또한 인식 비트의 수를 가변 할 수 있을 뿐만 아니라 다수의 쿼리 메시지를 하나의 쿼리 메시지로 통합할 수 있는 알고리즘을 제안하여 M-ary QT 알고리즘의 성능을 효율적으로 향상시켰다.

제안된 알고리즘의 태그를 인식하는 과정은 그림 4와 같이 리더에서 쿼리 메시지를 태그에 전송하고, 응답 메시지를 분석하여 새로운 태그 ID를 인식하거나 새로운 쿼리 메시지를 생성하여 쿼리 스택에 저장 한다. 이러한 과정을 질의-응답 cycle이라 하며 모든 태그를 인식할 때까지 반복한다. 쿼리 메시지의 구성은 이미 인식하고 있는 태그 ID의 prefix와 새로운 태그 ID의 인식을 위한 인식비트 ‘*’로 구성되어 있다. 인식비트 ‘*’는 쿼리 메시지를 맨체스터 코딩 기법을 이용하여 변환할 때 무전이 비트, 즉, 충돌 비트를 추가하여 구현하였다. 태그에서의 응답 메시지 생성과정은 그림 5와 같이 쿼리 메시지의 prefix와 충돌비트 ‘*’를 제외한 태그 ID의 prefix가 일치하면, 충돌 비트 위치의 m 비트를 표 1과 같은 규칙으로 매핑하여 확장 비트 $M(2^m)$ 을 생성하고 나머지 부분의 태그 ID를 추가하여 생성한다.

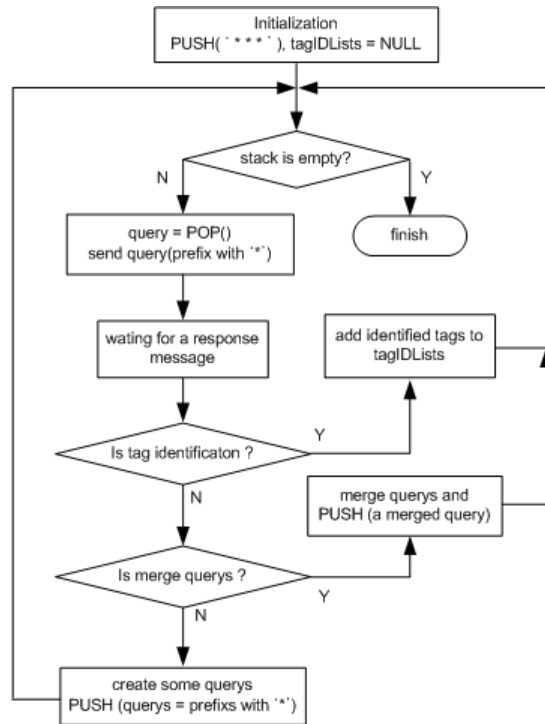


Fig. 4 Proposed M-ary QT algorithm for reader

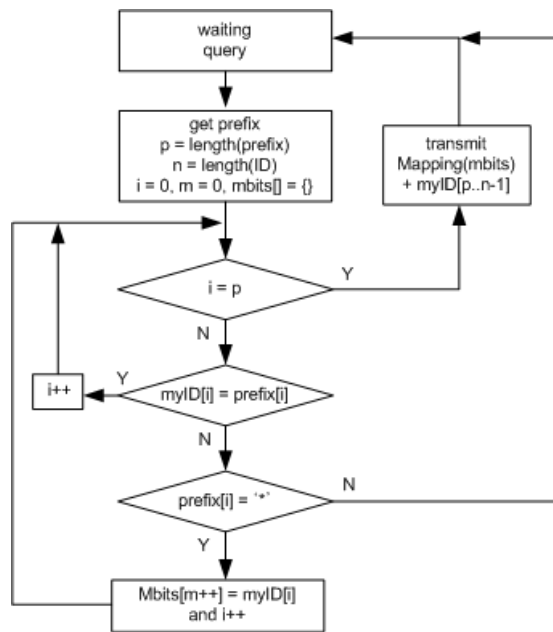


Fig. 5 Proposed M-ary QT algorithm for tag

리더에서 수신한 응답 메시지 구성은 그림 6과 같이 $4(2^2)$ 비트의 확장영역과 태그 ID의 나머지 부분 “myID [p..n-1]” 으로 구성되어 있고, 태그 ID의 나머지 부분은 앞에서부터 충돌이 발생하지 않은 rIDnc (no collision) 부분과 충돌이 포함된 rIDwc (with collision) 부분으로 구성되어 있다. 리더에서 응답 메시지 처리과정은 다음과 같다.

1) 그림 6과 같이 확장 영역 ‘0*0*’의 모든 비트 ‘1’과 충돌비트 ‘*’를 역 매핑하여 인식비트 ‘00’과 ‘10’을 구하고, 전송한 쿼리 메시지 ‘**’의 충돌비트 위치에 순차적으로 대입하여 prefix list를 구한다.

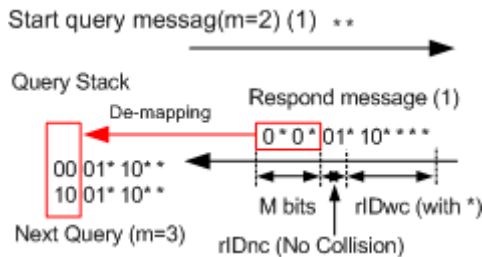


Fig. 6 Configuration of the response message

2) prefix list의 모든 prefix ‘00’과 ‘10’에 충돌이 발생하지 않은 ‘01’을 추가하여 ‘0001’과 ‘1001’을 구한다.

3) 다음 쿼리에 포함될 인식비트 수(m)을 다음과 같이 계산한다.

$$QM.size = prefix.size + m \quad (1)$$

$$RM.size = M(2^m) + TagID.size - QM.size \quad (2)$$

위 식에서, QM은 다음 쿼리 메시지, prefix는 인식된 비트열, m은 인식비트 수, RM은 응답 메시지, TagID는 태그 ID이다. 인식비트 수가 크면 보다 많은 비트를 인식할 수 있지만, 응답 메시지의 비트 수가 증가하여 비트에러 발생확률이 증가 하는 등의 단점이 있기 때문에 응답 메시지의 비트 수가 태그 ID의 전체 비트 수의 1.1 배 이내로 하였다.

4) 다음 쿼리 메시지는 그림 6과 같이 prefix list의 모든 prefix ‘0001’과 ‘1001’에 태그 ID의 나머지 부분 중 충돌이 포함된 부분 ‘*10* ***’에서 m개의 충돌비트를 추가하여 생성하고 쿼리 큐에 push한다. 그림 6의

예에서는 ‘0001*10**’과 ‘1001*10**’이 생성되어 쿼리 스택에 저장되어 있다.

5) 새로운 태그 인식은 태그 ID의 나머지 부분에서 충돌이 발생하지 않은 경우에 그림 7과 같이 구할 수 있다. 확장영역 ‘0**0*000’에서 인식비트 ‘011’, ‘101’, ‘110’을 구하여 쿼리 메시지 ‘0001* 10**’의 충돌비트 위치에 순차적으로 대입하여 구할 수 있다.

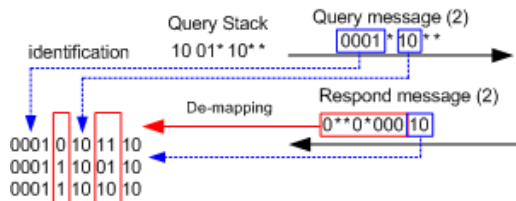


Fig. 7 New Tag Identification

6) 쿼리 메시지의 통합과정은 처리과정 (1), (2), (3), (4)에서 쿼리 메시지 list를 구하여 모든 쿼리 메시지의 비트를 비교하여 충돌이 발생한 충돌비트 수를 계산한다. 충돌비트 수가 처리과정 (3)에서 계산한 인식비트 수보다 작거나 같다면 그림 8과 같이 하나의 쿼리 메시지로 통합할 수 있다.

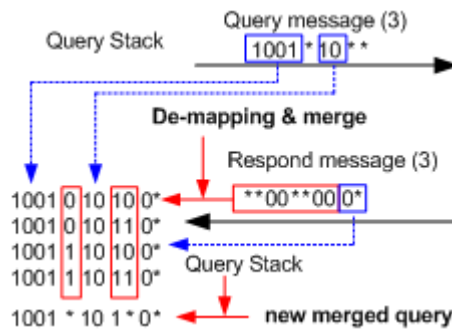


Fig. 8 New Query merge

7) 위의 과정을 모든 태그를 인식할 때까지 반복한다.

그림 9는 제안된 가변비트 M-ary QT 충돌해소 알고리즘의 예를 보여주고 있다. 제안된 알고리즘의 예에서는 ‘00010101110’, ‘00011100110’, ‘00011101010’, ‘10010101000’, ‘10010101101’, ‘10011101000’, ‘10011101001’, ‘10011101100’의 8개의 Tag ID를 인식하는 과정을 보여주고 있다.

쿼리 메시지 (1) ‘***’, (2) ‘0001* 10* *’, (3) ‘1001* 10* *’, (4) ‘1001* 101* 0* *’에서 볼 수 있는 것과 같이 제안된 알고리즘은 불연속적인 m 비트를 인식할 수 있을 뿐만 아니라 가변적인 ($m=2, m=3$) 비트를 인식할 수 있는 장점을 가지고 있다. 특히, 쿼리 메시지 (3) ‘1001* 10* *’의 응답 메시지 (3) ‘**00* *00 0*’의 경우에 기존의 방법으로 다음의 쿼리 메시지를 생성하면 ‘1001010100*’, ‘1001010110*’, ‘1001110100*’, ‘1001110110*’와 같은 4개의 쿼리 메시지가 생성되어 4번의 질의-응답 과정을 거쳐야 하지만, 제안된 알고리즘의 쿼리 메시지 통합기법을 적용하면 쿼리 메시지 (4) ‘1001*101*0*’와 같은 하나의 쿼리 메시지로 통합할 수 있기 때문에 1번의 질의-응답과정으로 새로운 태그를 인식할 수 있다.

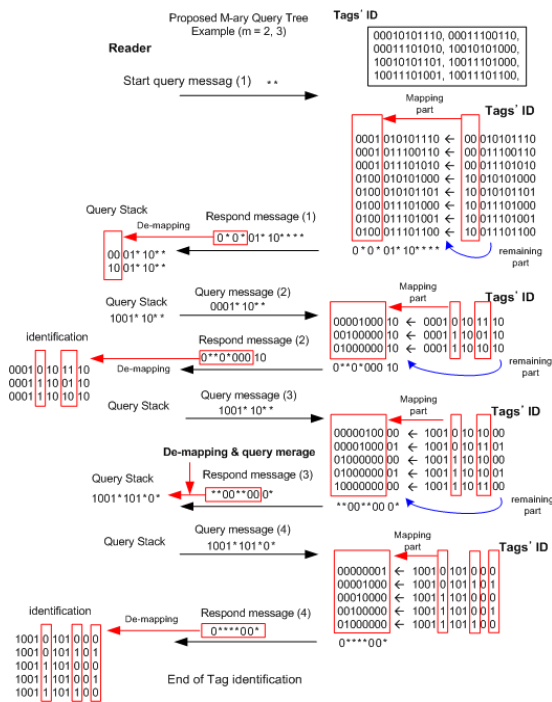


Fig. 9 Example of proposed Tag identification

IV. 시뮬레이션 결과

시뮬레이션 환경은 하나의 리더와 다수의 태그들로 구성되어 있고, 모든 태그는 리더의 인식 범위 내에 있

는 경우를 가정하였다. 태그 ID는 랜덤하게 부여된 64 bits 길이의 균일한 분포를 가지고 있으며, 태그의 개수는 16에서 1024개까지 증가시키면서 성능을 비교하였다. 시뮬레이션은 M-ary 쿼리 트리 알고리즘과 제안된 알고리즘의 핵심동작을 구현하여 다음의 세 가지 성능 지표를 가지고 비교 분석하였다. 첫 번째는 모든 태그를 인식하기 위한 질의-응답의 cycle 수이다. 질의-응답의 cycle 수가 작을수록 태그 인식 시간이 줄고 인식 효율이 향상된다. 두 번째로 전체 태그 수를 질의-응답의 cycle 수로 나누어 인식효율을 계산하였다. 인식효율 평가요소는 하나의 질의-응답 과정에서 평균 몇 개의 태그를 인식할 수 있는지를 평가할 수 있다. 세 번째는 모든 질의-응답 메시지의 총 비트 수를 태그 수로 나누어 통신비용을 계산하였다. 통신비용 평가요소는 하나의 태그를 인식하는데 전송되는 평균 비트수를 계산하였다.

그림 10은 4-ary QT 알고리즘과 제안된 가변비트 M-ary QT 알고리즘의 태그 수에 따른 질의-응답의 cycle 수를 보여주고 있다. 제안된 알고리즘의 인식비트 수 m 은 3, 4, 5, 6 으로 가변 하였으며, 최대 응답 비트수는 70bits 이하가 되도록 인식비트 수 m 을 결정하였다.

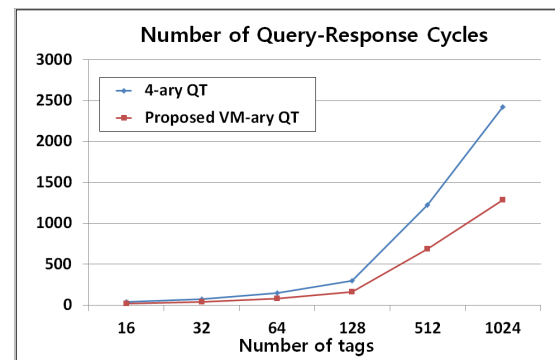


Fig. 10 Number of query-resopnse cycles

그림 10의 결과에서 전체 태그의 수가 1024인 경우에 4-ary QT 알고리즘의 질의-응답의 cycle 수는 2,423 번 이었고, 제안된 가변비트 M-ary QT 알고리즘의 질의-응답 cycle 수는 1,289번 이었다. 이러한 결과는 제안된 알고리즘의 효율이 기존의 4-ary QT 알고리즘에 비하여 약 46% 이상 향상된 결과를 보여주고 있다. 이러한 결과는 제안된 가변비트 M-ary QT 알고리즘이 맨

체스터 코딩을 이용하여 불연속의 가변 비트를 인식할 수 있을 뿐만 아니라, 조건에 따라 다수의 쿼리 메시지를 하나의 쿼리 메시지로 통합하여 매우 효율적인 태그 인식 기법을 적용하고 있기 때문에 판단된다.

그림 11은 태그 인식 효율을 보여주고 있다. 제안된 가변비트 M-ary QT 알고리즘의 인식 효율은 평균 74%인 결과를 보여주고 있고, 기존의 4-ary QT 알고리즘의 인식 효율은 평균 42%인 결과를 보여주고 있다. 그림 11의 결과에서 보여주듯이 제안된 가변비트 M-ary QT 알고리즘의 인식효율은 기존의 4-ary QT 알고리즘에 비하여 약 30% 이상 향상된 결과를 보여주고 있다. 즉, 기존의 4-ary QT 알고리즘에서 하나의 쿼리-응답의 cycle에 평균 0.4개의 태그를 인식할 수 있지만, 제안된 가변비트 M-ary QT 알고리즘에서는 평균 0.7개의 태그를 인식할 수 있다.

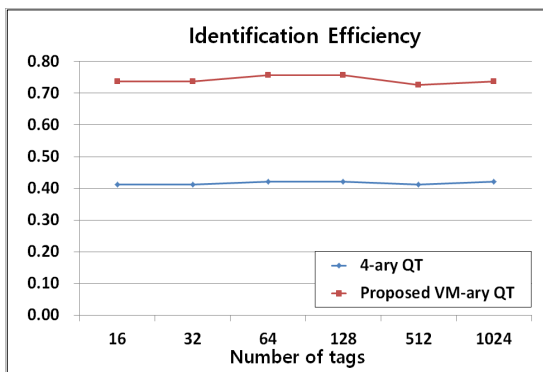


Fig. 11 Identification Efficiency

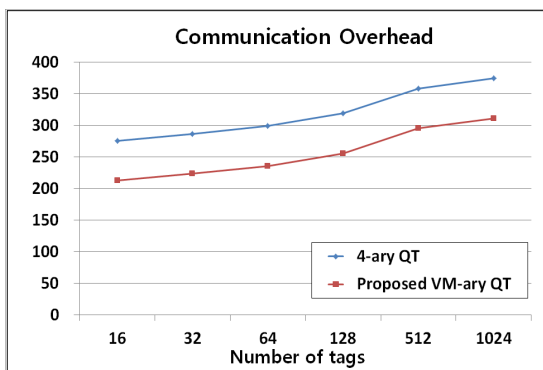


Fig. 12 Communication Overhead

그림 12는 통신비용의 결과를 보여주고 있으며, 제안된 가변비트 M-ary QT 알고리즘에서 하나의 태그를 인식하기 위해 평균 256.04 비트의 송수신이 필요했으며, 기존의 4-ary QT 알고리즘에서는 평균 318.93 비트의 송수신이 필요하였다. 시뮬레이션 결과 제안된 알고리즘은 기존의 4-ary QT 알고리즘에 비하여 통신비용은 약 0.8배 낮은 결과를 보여주고 있다.

V. 결론

RFID 리더의 가장 중요한 기능은 자신의 인식 범위 내에 있는 개별 제품들에 부착된 태그의 고유한 EPC (Electronic Product Code) 코드를 식별하는 것이다. 모든 태그를 인식할 수 있는 결정적인 방법은 리더의 질의에 태그에서 응답하는 방식으로 태그를 인식한다. 다수의 태그에서 응답하여 충돌이 발생할 경우 이를 해소하기 위한 많은 기법이 제안되었으며, M-ary QT 기법은 맨체스터 코딩 기법을 기반으로 충돌 비트의 위치정보를 이용하여 매우 효율적으로 태그를 인식하는 기법이다.

본 논문에서는 충돌비트의 위치를 정보를 리더에서 뿐만 아니라 태그에서도 이용할 수 있는 개선된 가변비트 M-ary QT 알고리즘을 제안하였다. 리더에서 태그로 전송하는 쿼리 메시지에 충돌비트를 추가함으로써 비연속적인 m bits 인식이 가능할 뿐만 아니라 가변 비트를 인식할 수 있는 기법을 제안하였다. 또한 응답메시지를 분석하여 다수의 질의 메시지를 하나의 질의 메시지로 통합할 수 있는 기법을 제안하여 M-ary QT 기법의 효율을 향상시킬 수 있었다.

시뮬레이션 결과 제안된 가변비트 M-ary QT 알고리즘은 기존의 M-ary QT 기법에 비하여 질의-응답의 cycle 수는 46% 이상, 인식 효율은 32%, 통신비용은 0.8배 향상된 결과를 보여주고 있다. 시뮬레이션 결과에서 보여주듯이 제안된 가변비트 M-ary QT 알고리즘은 충돌비트 위치 정보를 효율적으로 이용하여 태그 인식과정의 시간을 효과적으로 줄일 수 있음을 알 수 있다. 추후 연구과제로 태그 인식 과정의 효율을 향상시키기 위해 확률적(probability) 알고리즘에서 충돌 비트 위치 정보를 이용할 수 있는 방안을 연구하고자 한다.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 3th ed. John Wiley & Sons, 2003.

[2] K. NamGung, Y. Choi, Y. Kwon, C. Jun, S. Park and J. K. Han, "A Study on Applying RFID Systems of Korean Pharmaceutical Industry," in *Proceeding of the 5th International Conference on Convergence and Hybrid Information Technology*, Daejeon, pp. 235-243, Sep. 2011.

[3] C. H. Quan, W. K. Hong, Y. D. Lee and H. C. kim, "Performance Evaluation of Anti-collision Algorithms in the Low-cost RFID System," *Journal of KICS*, vol. 30, no. 1B, pp.17-26, Jan. 2005.

[4] X. Jia, Q. Feng and C Ma, "An Efficient Anti-Collision Protocol for RFID Tag Identification," *IEEE Communication Letter*, vol. 14, no. 11, pp. 1014-1016, Sep. 2010.

[5] Dongmin Yang and Jongmin Shin, "EMQT : A Study on Enhanced M-ary Query Tree Algorithm for Sequential Tag IDs," *The Journal of the KICS*, vol. 38, no. 6B, pp. 435-445, June 2013.

[6] C. N. Yang, L. J. Hu and J. B. Lai, "Query Tree Algorithm for RFID Tag with Binary-Coded Decimal EPC," *Journal of Information Science and Engineering*, vol. 16, no. 10, pp. 1616-1619, Oct. 2012.

[7] X. Q. Yan, Y. Liu, B. Li and X. M. Liu, "A memoryless binary query tree based Successive Scheme for passive RFID tag collision resolution," *Information Fusion*, vol. 22, no. 1, pp. 26-38, Mar. 2015.

[8] X. P. Zong, R. K. Lu, P. G. Wang, T. Zhang and S. Liu, "A Novel Anti-Collision Algorithm of Tags in RFID System," *Applied Mechanics and Materials*, vol. 721, no. 1, pp. 649-652, Nov. 2014.

[9] D. M. Yang, J. M. Shin and T. K. Park, "Adaptive M-ary Query Tree Protocol for the Fast RFID Tag Arbitration," *Information*, vol. 17, no. 3, pp. 1031-1036, Mar. 2014.

[10] Ching-Nung Yang, Song-Ruei Cai and Li-Zhe Sun, "Parallel Response Ternary Query Tree for RFID Tag Anti-Collision," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 72-79, Mar. 2015.

[11] Y. Kim, S. Kim, S. Lee, and K Ahn, "Improved 4-ary query tree algorithm for anti-collision in RFID system," in *Proc. IEEE AINA 2009, Bradford, U.K.*, pp. 699-704, May 2009.

[12] H. Gou and Y. Yoo, "Bit collision detection based query tree protocol for anti-collision in an RFID system," *Int. J. Innovative Computer Information Control*, vol. 8, no. 5, pp. 3081-3102, May 2012.



김관웅(Kwan-Woong Kim)

1996년 전북대학교 전자공학과 공학사
 1998년 전북대학교 대학원 전자공학과 공학석사
 2002년 전북대학교 대학원 전자공학과 공학박사
 2010년~현재 (주)썬더테크놀로지 기술이사
 ※관심분야 : 무선통신시스템, DSP 신호처리



김변곤(Byun-Gon Kim)

1990년 한국항공대학교 항공전자공학과 공학사
 1997년 전북대학교 대학원 전자공학과 공학석사
 2003년 전북대학교 대학원 전자공학과 공학박사
 2005년~현재 군산대학교 전자공학과 교수
 ※관심분야 : RFID 통신시스템, 디지털신호처리