

RFID 시스템을 위한 개선된 하이브리드 쿼리 트리 알고리즘

김태희, 이성준, 안광선
경북대학교 전기전자컴퓨터공학부
e-mail : kth3916@hotmail.com

An Improved Hybrid Query Tree Algorithm for RFID System

Tae-Hee Kim, Seong-Joon Lee, Kwang-Seon Ahn
Dept. of Computer Engineering, Kyungpook National University

요 약

RFID 시스템에서 리더와 태그는 단일 무선 공유 채널을 갖기 때문에 RFID 수동형 태그를 위한 태그 충돌 중재가 태그 인식을 위한 중요한 이슈이다. 본 논문에서는 태그 충돌 방지를 위한 Improved Hybrid Query Tree algorithm 을 제안한다. 제안된 알고리즘은 쿼리 트리를 기반으로 태그가 리더에게 ID 를 전송하는 시점을 전송 ID 상위 3 비트 내의 '1' 값을 이용하여 결정한다. 또한 전송 받은 Tag 의 상위 3 비트의 충돌이 발생하더라도 전송 슬롯에 따라 다르므로 제안한 알고리즘에서 예측이 가능하다. 시뮬레이션을 통한 성능 평가에서 다른 트리 기반 프로토콜에 비해 제안한 알고리즘이 쿼리 횟수에서 높은 성능을 갖는다는 것을 보여준다.

1. 서론

RFID(Radio Frequency Identification)는 유비쿼터스 환경의 기술로서 물품에 부착된 전자 태그의 정보를 리더가 RF 신호를 이용하여 태그를 인식하는 자동 인식 기술이다[1].

RFID 시스템은 태그와 리더 두 가지로 구성된다. 태그는 리더의 요청에 따라 자신의 ID 를 RF 신호로 리더에게 전송하고, 리더는 태그를 인식하기 위한 작업을 수행한다. RFID 시스템 환경에서 리더와 다수의 태그가 무선 채널을 공유한다. 그러므로 다수의 태그가 응답하는 경우 충돌(collision)이 발생하여 어느 태그도 인식하지 못하는 상황이 발생한다. 그러므로 충돌을 인식하고 새로운 프리픽스를 만들어 태그에게 질의하기 위한 절차가 필요하다[2]. 이러한 절차를 충돌 방지 알고리즘 (anti-collision algorithm)이라고 부른다.

태그 충돌 방지 알고리즘은 확률적 방식과 트리 기반 방식으로 분류된다. 확률 기반의 태그 충돌 방지 기법은 확률에 근거하여 태그 아이디를 전송할 임의의 시간을 선택하고, 아이디 전송을 반복하는 방법이다. 대표적인 확률기반의 태그 충돌 방지 기법은 ALOHA, Slotted ALOHA, Frame slotted ALOHA 가 있다 [3,4,5].

트리 기반의 방법은 충돌이 발생하였을 때, 트리를 기반으로 다음 질의 프리픽스를 선택하고, 질의함으로써 영역내의 모든 태그를 인식하는 방법이다. 그러나 트리 기반의 방법은 유사 태그가 많은 경우 충돌이 많아져 인식시간이 지연된다. 대표적인 알고리즘

으로 이진 트리[6]와 Query Tree(QT) 프로토콜[7]이 있다. 그리고 두 가지 방식의 단점을 보완하는 하이브리드 알고리즘이 제안되었다.

본 논문에서는 이러한 하이브리드 알고리즘을 기반으로 하는 개선된 하이브리드 쿼리 트리 알고리즘 (IHQT, Improved Hybrid Query Tree algorithm)을 제안한다. 제안된 방법에서 태그는 아이디를 전송할 때 전송을 시작할 비트의 위치로부터 3 비트들의 합을 이용하여 전송할 타임 슬롯을 선택한다. 리더는 태그로부터 전송된 데이터 값을 맨체스터 코딩 방법을 이용하여 충돌이 발생하지 않은 부분을 찾아 전달된 코드의 값을 파악하는 것이 가능하다. 시뮬레이션에서 제안된 알고리즘은 중복성이 높을수록 좋은 성능을 보였다.

본 논문의 구성은 다음과 같다. 먼저 2 장에서는 관련 연구로 확률적 방식과 트리 방식의 충돌 방지 알고리즘을 설명한다. 3 장에서는 개선된 하이브리드 쿼리 트리 알고리즘을 제안하고 동작 방식을 설명한다. 4 장에서는 제안 알고리즘의 성능 평가를 하고 마지막으로 5 장에서 결론을 기술한다.

2. 관련연구

트리 기반 알고리즘은 일반적으로 2 진 트리를 기반으로 태그의 질의 프리픽스를 생성한다. 어떤 질의가 주어졌을 때 다중의 태그들이 응답함으로써 충돌이 발생하면 프리픽스를 확장해 새로운 ID 를 만들게 된다. 리더가 프리픽스를 전송한 후 태그의 아이디를 받는 상태에서 상황에 따라 3 가지의 상태를 가지게

된다. 유힬 사이클(Idle cycle)은 일치하는 프리픽스를 가진 태그가 존재하지 않아 어떠한 태그도 응답하지 않는 상태이고, 충돌 사이클(Collision cycle)은 2 개 이상의 태그가 동시에 응답하여 tag 의 존재를 인식하지 못하는 상태이다. 마지막으로, 단 하나의 태그만 응답함으로써 태그의 아이디를 인식하는 상태를 인식 사이클(Success cycle)이라고 부른다.

쿼리 트리 알고리즘의 동작과정[8]은 리더가 태그에게 k 비트의 프리픽스를 모든 태그에게 전파하면, 태그는 전달된 프리픽스와 자신이 소유한 아이디가 일치하는지 검사한다. 만일 일치한다면 태그는 k+1 번째부터 나머지 아이디를 리더에게 응답한다. 충돌 사이클이 발생했다면 같은 프리픽스를 가진 태그가 두 개 이상 존재한다는 것을 의미하므로 리더는 큐에 현재 프리픽스에 '0'과 '1'을 접목한 새로운 두 개의 프리픽스를 큐에 입력한다. QT 알고리즘은 구현이 쉽고, 적은 비용으로 생산이 가능하다는 장점을 갖는다. 그러나, 태그 아이디의 분포 상황을 고려하지 않은 상태에서 태그 아이디를 확장하는 것은 유힬 사이클의 원인이 된다. 또한 이러한 환경으로 인해 태그 인식에 많은 질의와 전송 비트 수를 필요로 한다.

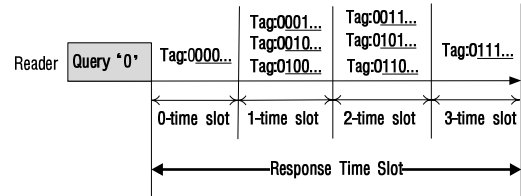
하이브리드 쿼리 트리 알고리즘(HQT, hybrid query tree algorithm) [9]은 4-ary 검색 트리 기법을 적용하여 프리픽스가 1 비트에서 2 비트씩 증가하도록 확장하였다. 이로 인해 태그 들 사이의 충돌 사이클을 줄이고 있으나 유힬 사이클의 증가를 가져왔다. 이러한 문제점을 해결하기 위해 HQT 는 슬롯드 지연 기법을 적용하였다. 슬롯드 지연 기법이란 전달된 프리픽스와 일치하는 태그들이 즉시 응답하는 것이 아니라 특정 시간 대기한 후 데이터를 전송하는 기법이다. HQT 는 슬롯 지연 기법을 이용하여 유힬 사이클의 수를 줄이는 것을 시도하였다. 즉, 시작 위치와 완료 위치를 점검하여 불필요한 프리픽스의 확장을 줄이는 것이다. 그러나, 제안된 방법에서는 비지 상태만을 점검하기 때문에 중간에 보내지지 않은 슬롯의 경우 탐지가 불가능하다. 또한, 슬롯의 개념을 변형하여 한 슬롯 당 시간을 줄이고 있으나 질의 횟수를 늘려야 하는 문제를 발생시킨다.

알로하 기반의 프로토콜은 보통 슬롯 알로하(Slotted ALOHA) 프로토콜[5]을 말하며 이는 태그의 응답 시간을 고정된 몇 개의 슬롯(Slot)으로 나누고 각 태그들은 자신이 사용할 슬롯을 선택하여 자신의 태그 ID 를 전송하는 방식이다. 여기서 슬롯이란 시간 간격을 의미한다. 이 방식에서 리더는 나누어진 각 타임 슬롯에 태그 ID 가 두 개 이상 응답하면 충돌 발생으로 인식한다.

3. Improved Hybrid Query Tree protocol

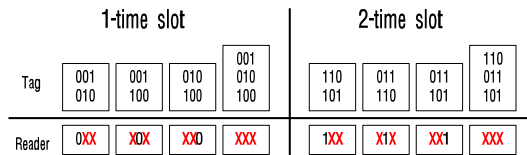
본 논문에서 제안하는 Improved Hybrid Query Tree(IHQT)는 태그의 ID 비트의 1 의 개수를 가지고 응답슬롯 선택하여 태그 ID 를 응답한다. IHQT 는 4-ary 와 QT 알고리즘과는 달리 idle cycle 을 생성하지 않고 collision 과 쿼리의 횟수를 줄여 인식 시간을 줄인다.

IHQT 알고리즘은 리더로부터 k 개의 비트를 가진 prefix_k 를 전송 받는다. prefix_k 와 일치하는 태그는 ID_{k+1} 번째의 비트부터 3 개의 비트의 '1' 의 개수를 카운트한다. 이 카운트 값은 Counter(C)라하고 태그는 C 값에 따라 응답 슬롯을 선택한다. 응답 슬롯을 선택한 태그들은 ID_{k+1} 부터 전체의 ID 를 리더에게 응답하게 된다. 그림 1 은 태그가 C 값에 따라 응답 슬롯을 선택하는 것을 나타내고 있다.



(그림 1) IHQT Tag 응답 슬롯 선택 알고리즘

그림 2 는 응답 슬롯에 따른 태그 충돌의 경우를 나타내고 있다. 1-time slot 에서는 프리픽스 다음의 상위 3 비트의 C 값이 1 인 태그만이 응답하기 때문에 0 의 위치와 충돌 비트의 위치를 알면 응답한 태그들의 3 비트 예측이 가능하다. 첫 번째 경우를 살펴보면 '001', '010' 태그가 응답하여 리더에서는 '0XX'로 인식하게 된다 이때 응답 슬롯이 1-time 이기 때문에 C 값이 1 이 되게 된다. 그러면 '0XX'가 나올 수 있는 태그 응답의 경우는 '001'과 '010'만이 되는 것이다. 또한 C 값이 1 인 태그가 모두 응답하였을 경우는 'XXX'로 인식이 되며 이 경우에도 모든 응답 태그를 예측할 수 있다. 2-time slot 에서도 마찬가지로 '1'의 위치와 충돌 비트만 알면 나머지 충돌 비트내의 '1'과 '0'의 위치를 예측 가능하다. 이렇게 예측한 비트들은 이전에 보내진 프리픽스와 연결하여 새로운 프리픽스를 생성하고 큐에 push 한다.



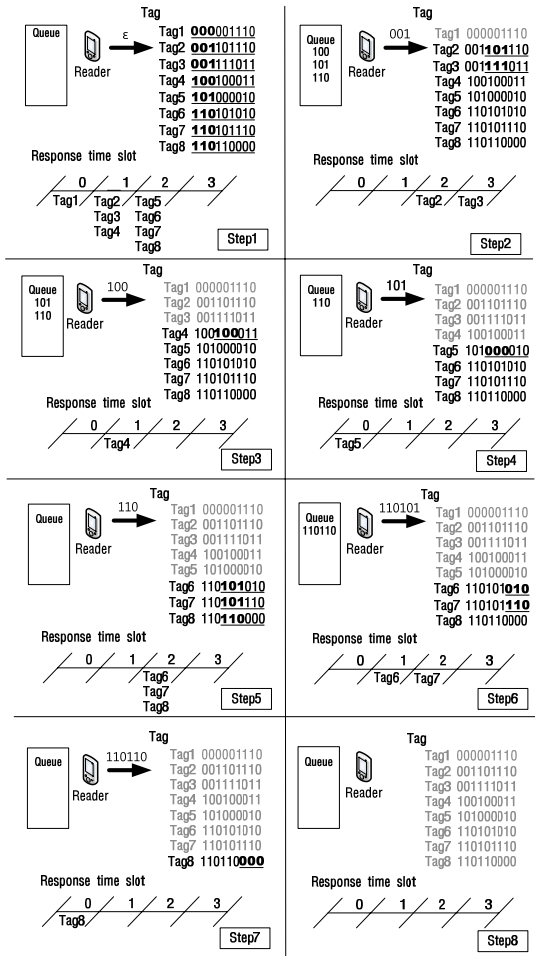
(그림 2) IHQT 타임 슬롯에 따른 Tag 충돌

그림 3 은 태그 응답 시 상위 3 비트내의 충돌이 존재하지 않을 경우에 ID 를 인식하는 범위를 나타내고 있다. Tag 1, Tag 2 가 1-time slot 에서 ID 전송을 시작하고 리더에서는 태 6 번째 비트까지 충돌 없이 전송 받았다. 하지만 7 번째 비트부터 충돌이 발생해 태그 ID 를 인식할 수 없는 상태가 되었다. IHQT 에서 상위 3 비트 내의 충돌은 예측할 수 있지만 나머지 비트에서 충돌이 일어나게 되면 예측이 불가능하게 된다. 그러므로 리더는 충돌이 일어나기 이전까지 지점 '001010'를 인식하며 인식한 ID 를 가지고 새로운 프리픽스를 생성한다.

Tag1 : 0 0 1 0 1 0	1 0 0 000
Tag2 : 0 0 1 0 1 0	0 1 0 000
Reader: 0 0 1 0 1 0	
	X X 0 000

(그림 3) 태그 충돌시 ID 인식 범위

그림 4 는 임의적으로 생성된 9-bit 태그 8 개를 인식하는 과정을 나타내고 있다.



(그림 4) IHQT 동작 예제

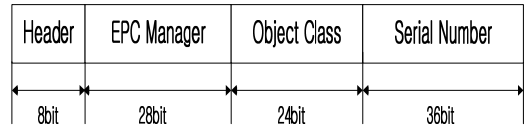
리더가 태그에게 ‘ε’ 문자열을 전파하면 태그들은 자신의 상위 3 비트의 값에 따라 응답 슬롯을 선택하고 ID 를 전송 한다. 스텝 1)에서는 0-time 에 응답하는 태그는 하나이기 때문에 태그가 인식이 된다. 1 타임과 2 타임 에서는 응답한 태그가 2 개 이상이므로 충돌 비트를 확인해 상위 3 비트의 값을 예측하고 새로운 쿼리를 생성한다. 이러한 과정을 반복하여 9-bit 의 태그 8 개를 7 번의 쿼리를 통해 모두 인식하였다.

예제에서 제시한 태그를 인식하기 위해 필요한 질의 횟수는 QT 의 경우 25 회 HQT 의 경우는 21 회가 요구된다. 그러나 제안된 논문의 경우 충돌 이후 ID 의 비트정보를 파악 할 수 없기 때문에 빈 슬롯의 수가 증가 할 수 있다는 단점을 가지고 있다. 그러나 제안된 알고리즘은 충돌 사이클을 줄임으로써 트리의 노드 수를 줄이고 특정 그룹으로 분류되어 있는 경우 좋은 성능을 발휘하는 것으로 성능평가에서 나타났다.

4. 성능평가

성능평가는 시뮬레이션 프로그램을 사용하여 진행하였다. 실험에서 사용한 ID 는 EPC global 의 표준 중에 하나인 96bit 태그를 사용하였다.

96bit 태그는 그림 5 와 같이 Header, EPC Manager, Object class, Serial number 로 구성되어있다.



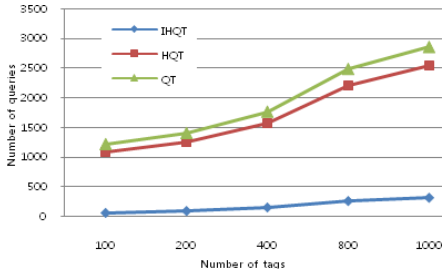
(그림 5) EPC 96 bit Tag Structure

Header 는 데이터 유형 및 길이를 정의하고, EPC Manager 는 상품에 대한 분류와 일련번호를 관리하는 기관이나 기업을 표시한다. Object Class 는 바코드의 상품 품목코드에 해당하는 것으로 각 품목 또는 단위를 표시하고 Serial Number 는 각 상품들에 대하여 부여되는 고유한 식별번호를 나타낸다

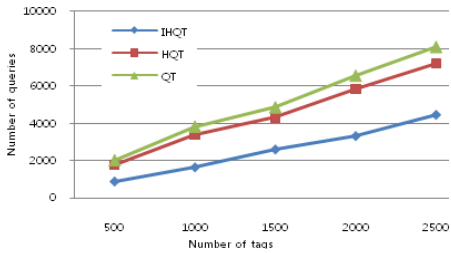
실험에서 사용한 태그 ID 의 구성은 Header, EPC Manager, Object class 필드를 무작위로 10 개를 생성하였다. Serial Number 필드는 0 부터 시작하여 1 씩 증가시키면서 태그를 생성하거나 Serial Number 필드 전체를 랜덤하게 생성하였다. Serial Number 필드를 제외한 나머지의 비트가 같다는 것은 동일 제품 코드를 나타내는 것이며 Serial Number 가 순차적으로 있는 것은 물류 창고에 동일 제품이 여러 개 쌓여있는 것을 의미한다.

태그의 개수는 순차적일 경우 각각 10 의 군에 대해 10 개부터 100 개까지 생성하여 전체 태그개수가 100 개부터 1000 개까지 비교하였다. Serial number 필드가 랜덤한 경우에는 5 개 군을 형성해 각 군에 대해 태그 100 개부터 500 개까지 생성하여 전체 태그의 개수를 1000 개부터 2500 개까지 실험하였다.

실험에서는 제안한 알고리즘과 HQT 알고리즘과 QT 알고리즘으로 쿼리 횟수와 1 개 태그를 인식하기 위한 평균 쿼리 수를 비교하였다. 그림 6 의 (a)에서의 결과를 보면 IHQT 에서 Serial Number 필드가 순차적인 100 개의 태그를 인식하기 위해 필요한 쿼리의 수는 66 이다. 이는 HQT 알고리즘의 1173 개에 비해 성능이 10 배 이상 향상되었다. 그림 6 의 (b)는 Serial Number 필드가 랜덤하게 생성된 태그의 실험 결과이다. 태그가 랜덤한 ID 를 가지더라도 인식 성능은 QT 와 HQT 에 비해 향상되었다.



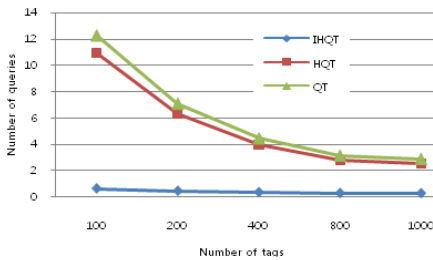
(a) Sequential assignment in serial number field



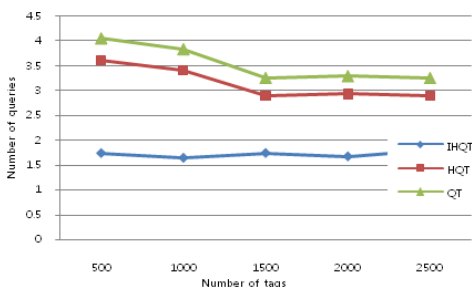
(a) Random assignment in serial number field

(그림 6) 96bit Tag 인식을 위한 쿼리 수

그림 7 은 한 개의 태그를 인식하기 위한 평균 쿼리의 수를 나타내고 있다. Serial number 필드가 순차적일 때 랜덤 할 경우보다 다른 알고리즘에 비해 성능이 더 높게 나타났다.



(a) Sequential assignment in serial number field



(b) Random assignment in serial number field

(그림 7) 하나의 태그를 인식하기 위한 평균 쿼리 수

5. 결론

RFID 시스템은 다수의 태그를 리더가 빠르게 인식할 수 있는 알고리즘이 필요하다. QT 에서는 한 비트씩 확장하면서 태그를 인식하고 HQT 에서는 2 비트씩 확장하면서 인식과정을 거친다. 하지만 확장과정을 거치면서 idle cycle 이 늘어나게 되어 불필요한 질의 응답과정이 생긴다. 본 논문에서는 이러한 문제점들을 개선하여 태그확장을 빠르게 할 수 있는 새로운 IHQT 를 제안하였다.

그러나 IHQT 는 태그 비트 확장 속도 면에서는 빠르나 태그가 랜덤하게 분포되어있을 때 응답과정에서 빈 슬롯이 발생하게 된다. 슬롯의 응답타임을 어떻게 설정하느냐에 따라 시스템의 전체 인식시간이 결정될 것이고 만약 응답 슬롯이 타임이 길어진다면 태그 인식 시간 또한 늘어날 것이다. 응답슬롯이 길어짐에 따라 빈 슬롯이 발생한다면 낭비되는 시간 또한 늘어나게 되는 것이다. 향후 낭비되는 빈 슬롯을 줄일 수 있도록 응답 슬롯을 동적으로 할당하는 연구가 필요하다.

참고문헌

- [1] F. Mattern, "The Vision and Foundations of Ubiquitous Computing," Upgrade, Vol.2, No.5, pp2-6, October, 2001.
- [2] J. Myung, and W. Lee, "Adaptive Binary Splitting: A RFID Tag Collision Arbitration Protocol for Tag Identification," ACM/Springer Mobile networks and Applications (ACM MoNET), vol.11, no.5, pp.711-722, October 2006.
- [3] EPC™ Radio-Frequency Identification Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz Version 1.0.8, EPCglobal, Dec. 2004.
- [4] Information technology automatic identification and data capture techniques-Radio frequency identification for item management Air interface. Part 6. Parameters for Air interface communications at 860-960 MHz, Final Draft International Standard ISO 18000-6, Nov. 2003.
- [5] K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. John Wiley & Sons, 2003.
- [6] T.Wang, "Enhanced binary search with cut-through operation for anti-collision in RFID systems," IEEE Commun. Lett., vol. 10, no.4, pp.236-238, April 2006.
- [7] J. H. Chio, D. Lee, H. Lee, "Query tree-based reservation for efficient RFID tag anti-collision," IEEE Commun. Lett., vol. 11, no. 1, pp. 85-87, April 2006.
- [8] Ching Law, Kayi Lee, and Kai-Yeung Sju, "Efficient Memoryless Protocol for Tag Identification", In Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication, ACM, pp. 75-84. August 2000.
- [9] J. Ryu, H. Lee, Y. Seok, T. Kwon and Y.Choi, "A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID system," IEEE ICC, pp 5981-5986, Jun. 2007