

저비용 RFID 시스템에서의 충돌방지 알고리즘에 대한 성능평가

정회원 권성호*, 홍원기**, 이용두**, 김희철**

Performance Evaluation of Anti-collision Algorithms in the Low-cost RFID System

Cheng-hao Quan*, Won-kee Hong**, Yong-doo Lee**, Hie-cheol Kim** *Regular Members*

요약

RFID(Radio Frequency Identification) 기술은 RF 신호를 사용하여 물품에 부착된 전자태그를 비접촉식으로 식별하는 자동인식기술이다. RFID 시스템 구축에 있어 식별영역 내에 다수의 태그가 존재할 경우, 다중태그 식별(multi-tag identification)을 위한 충돌방지(anti-collision) 알고리즘이 필수적으로 요구된다. 태그 충돌방지와 관련된 기존 연구들은 각각 고유한 형태의 코드체계를 기반으로 하고 있으며 태그 식별성능에 대한 비교연구도 부족한 상태이다. 본 논문에서는 저비용(low-cost) RFID 시스템 구축을 목표로 표준화가 진행되고 있는 96-비트 EPC(Electronic Product Code) 코드를 기반으로 기존 대표적인 충돌방지 알고리즘인 트리 기반 메모리리스(tree based memoryless) 충돌방지 알고리즘들과 슬롯 알로하 기반(slot aloha based) 충돌방지 알고리즘들의 성능평가를 수행한다. 성능평가 결과 초당 평균 태그 식별개수에서 충돌 추적 트리(collusion tracking tree) 알고리즘이 다른 알고리즘들보다 최소 2배에서 최대 50배 이상의 우수한 성능을 보여준다.

Key Words RFID, EPC, multi-tag identification, anti-collision algorithm, performance evaluation

ABSTRACT

RFID(Radio Frequency Identification) is a technology that automatically identifies objects attached with electronic tags by using radio wave. For the implementation of an RFID system, an anti-collision algorithm is required to identify several tags within the RFID reader's range. Few researches report the performance trade-off among anti-collision algorithms in terms of the communications traffic between the reader and tags, the identification speed, and so on. In this paper, we analyze both tree based memoryless algorithms and slot aloha based algorithms that comprise of almost every class of existing anti-collision algorithms. To compare the performance, we evaluated each class of anti-collision algorithms with respect to low-cost RFID system with 96-bit EPC(Electronic Product Code). The results show that the collision tracking tree algorithm outperforms current tree based and aloha based algorithms by at least 2 times to 50 times.

I. 서론

유비쿼터스 환경 구현에 있어 핵심적인 기술로서 주목받고 있는 RFID(Radio Frequency Identification)

기술은 RF 신호를 사용하여 물품에 부착된 전자태그(tag)를 식별하는 비접촉식 기술로서 인터넷의 지속적인 성장과 전자태그의 저비용화, 식별코드의 표준화에 힘입어 다양한 산업분야에서 실용화가 가능한 기술로

* 대구대학교 정보통신공학과 박사과정 (chquan@daegu.ac.kr)

** 대구대학교 정보통신공학부 교수 ({wkhong, ydlee, hckim}@daegu.ac.kr)

논문번호 KICS2004-10-227, 접수일자 2004년 10월 12일

서 전망되고 있다.

RFID 시스템에서 태그식별은 리더기가 물품에 부착된 태그에 질의하면 해당 태그는 그 질의에 대하여 자신의 식별자를 리더기로 전송하는 응답과정을 통하여 수행된다 이때 리더기의 식별영역 내에 한 개의 태그만 존재할 경우 태그식별은 간단하게 처리될 수 있으나 다수 개의 태그가 존재할 경우에는 여러 개의 태그가 동시에 리더기에 응답하게 되므로 리더기에서 태그들 간의 충돌이 발생하게 된다 이러한 충돌은 리더기로 하여금 정확한 태그식별을 방해하는 원인이 되며 특히, 대량의 물품을 실시간으로 식별해야 하는 대규모 전자 물류시스템 등에 적용하기 위해서는 다중태그 식별을 효과적으로 처리하기 위한 충돌방지(anti-collision) 알고리즘^[1]이 필수적으로 요구된다

최근 들어서 RFID 시스템에 대한 연구는 저비용 RFID 시스템을 위주로 인식거리가 길고 주변 환경의 영향을 적게 받는 UHF 대역(860 ~ 960MHz)에서 주로 이루어지고 있다 이러한 연구들은 시스템 구축에 초점을 맞춰 진행되어 왔으나, 다중태그들의 충돌방지 및 고속인식에 대한 연구는 아직도 많이 부족한 상태이다 이들은 각자 고유한 태그체계를 기반으로 하고 있으며 식별자의 비트 수도 8, 16, 32-비트 등 다양한 식별자체계 하에서 연구가 수행되어 왔다

RFID 시스템의 신속한 저번확대를 위해서는 태그에 저장된 정보체계의 표준화가 필요하다 이를 위해 현재, Auto-ID 센터를 중심으로 EPCGlobal에 의한 표준화 노력으로 표준태그 정보체계인 EPC(Electronic Product Code)코드가 사실상 RFID 표준코드로 자리잡아가고 있다 EPC 코드는 제조업자와 물품명에 대한 정보를 담고 있는 기존 바코드와는 달리 개별 물품마다 고유의 코드체계를 갖추게 함으로써 물품의 위치, 상태 등 다양한 정보를 획득하여 보다 효과적으로 관리, 활용할 수 있게 한다 EPC 코드는 헤더, 제조업자 코드, 제품코드, 물품코드 등 네 개의 필드로 구성되어 96-비트와 128-비트의 크기를 가지고 있다

본 논문에서는 저비용 RFID 시스템에서 96-비트 식별자를 갖는 EPC 코드를 기반으로 트리 기반 메모리리스(tree based memoryless) 충돌방지 알고리즘들과 슬롯 알로하 기반(slot aloha based) 충돌방지 알고리즘들에 대한 소개와 성능평가를 수행한다. 성능평가 결과 트리 기반 메모리리스 충돌방지 알고리즘의 하나인 충돌 추적 트리(collision tracking tree) 알고리즘이 질의-응답 횟수나 전송비트 수에서 가장 우수하였으며 초당 평균 태그 식별개수에서 다른 알고리

즘들에 비해 최소 2배에서 최대 50배 이상의 우수한 성능을 보여주었다

본 논문의 구성은 다음과 같다. 제2장에서는 영역 내 다수개의 태그 식별을 위한 다중태그 식별문제와 관련된 기존연구들을 살펴보고 제3장과 제4장에서는 각각 트리 기반 메모리리스 충돌방지 알고리즘과 슬롯 알로하 기반 충돌방지 알고리즘의 기본개념, 동작과정과 동작사례를 살펴본다 제5장에서는 기존 트리 기반 메모리리스 알고리즘과 슬롯 알로하 알고리즘에 대한 성능평가를 수행하고 분석결과를 제시한다 마지막으로 제6장에서 결론을 맺는다.

II. 관련연구

다중태그 식별을 위한 충돌방지 알고리즘과 관련하여 무선통신 환경에서의 채널 다중접근과 관련된 많은 연구가 이루어졌다 그러나 RF에서 요구되는 통신방식은 기존의 이러한 무선통신과는 전력공급, 연산능력, 태그 상호간의 존재여부와 영역내의 태그개수를 알 수 없는 등 많은 점에서 다르다^[1] 다중태그 식별을 위한 충돌방지 알고리즘은 크게 트리 기반의 결정적(deterministic) 알고리즘과 슬롯 알로하 기반의 확률적(probability) 알고리즘으로 구분할 수 있다

결정적 알고리즘은 이진비트로 표현된 태그 식별자들로 이진트리를 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 상태정보를 저장·관리해야 하는 부담을 안고 있다 대표적인 알고리즘으로 분할 트리(splitting tree) 알고리즘^[2], 비트-중재(bit-arbitration) 알고리즘^[3]이 있다 반면 메모리리스 알고리즘에서 태그의 응답은 태그에 대한 질의에 의해서만 결정된다 따라서 이러한 알고리즘을 적용할 경우 태그구현이 간단하여 저비용·저전력·초소형화 문제를 해결할 수 있는 좋은 접근방법이 될 수 있다 대표적인 알고리즘으로 트라워킹(tree-walking) 알고리즘^[4], 쿼리 트리(query tree) 알고리즘^[5]과 메모리리스형 충돌 추적 트리(collision tracking tree) 알고리즘^[6]을 들 수 있다

확률적 알고리즘은 알로하 프로토콜에 기반을 두고 있다. 리더기의 식별영역내의 태그들은 주어진 N개의 슬롯에서 태그의 정보를 전송할 슬롯을 임의로 선정하여 해당 식별자를 전송하게 되므로 슬롯간의 시간



그림 1. 충돌방지 알고리즘의 분류

차에 의해 태그충돌을 피한다. 그러나 식별영역내의 태그개수를 정확히 파악하기 어렵기 때문에 적절한 슬롯개수와 종료시점을 확률적으로 계산해야 한다. 따라서 확률에 근거한 종료시점의 결정으로 태그식별의 완전성을 지원하지 못하며 또한 충돌이 발생한 슬롯의 재전송으로 인하여 태그 식별시간에 있어서 높은 성능을 기대하기 어렵다는 단점을 갖는다. 확률적 알고리즘은 다시 ID-Slot형 알고리즘과 Bit-Slot형 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^[7], STAC(Slotted Terminating Adaptive Collection) 알고리즘^[8]이 있고 Bit-Slot형 알고리즘으로 Bit-Slot 메커니즘을 사용하는 충돌방지 알고리즘^[9]이 있다.

그림 1은 지금까지 언급한 충돌방지 알고리즘들을 분류하여 보여주고 있다.

III. 트리 기반 메모리리스 충돌방지 알고리즘

본 논문에서는 하드웨어 비용이 적게 들고 전력소모가 낮은 수동식 태그만 설명한다. 앞 절에서 언급했듯이 트리 기반 충돌방지 메모리리스 알고리즘들은 태그의 상태정보를 유지할 필요가 없기 때문에 저비용에 태그구현이 가능한 특징을 가지고 있다. 이러한 알고리즘들은 태그 식별과정 중 식별자의 일부 비트를 리더기에 저장하기 위한 메모리를 요구하며 스택이나 큐의 구조를 갖는다. 다음 소절에서는 대표적인 트리 기반 알고리즘인 트리-위킹, 쿼리 트리, 충돌 추적 트리 알고리즘을 간단히 살펴본다.

트리-위킹 알고리즘은 리더기가 k -비트 길이의 프리픽스(B_k)를 인자로 태그에 질의하는 것으로 시작한다. 영역내의 태그들은 수신한 프리픽스를 자신의 식별자

와 비교를 통해 매칭이 이루어질 경우 태그 식별자의 $(k+1)$ 번째 비트(b_{k+1})를 리더기로 전송한다. 이때, 리더기에서의 태그의 응답결과는 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})을 생성한다. 새로 생성된 프리픽스는 다음 번 질의-응답 과정에서 태그에 인자로 전달된다. 이러한 과정을 태그식별자의 비트길이만큼 수행하며 한 개의 태그가 식별될 때까지 반복한다. 한 개의 태그식별이 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 식별과정은 종료되며 이는 영역내의 모든 태그가 식별되었음을 의미한다.

쿼리 트리 알고리즘의 경우, 리더기가 보낸 k -비트 프리픽스(B_k)와 매칭된 태그들은 자신의 식별자의 $(k+1)$ 번째 비트부터 마지막 비트까지의 나머지 비트들을($ID_{k+1,i}$) 리더기로 전송한다. 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 영역내의 모든 태그가 식별될 때까지 반복된다. 큐가 비어있으면 전체 태그식별 과정은 종료되며 이는 영역내의 모든 태그가 식별되었음을 의미한다.

충돌 추적 트리 알고리즘의 경우, 리더기는 k -비트 길이의 프리픽스(B_k)를 인자로 태그에 질의하며 영역내의 태그들은 수신한 프리픽스를 자신의 식별자와 비교를 통해 매칭이 이루어질 경우 태그식별자의 $(k+1)$ 번째 비트부터 식별자의 마지막 비트까지($ID_{k+1,i}$)를 순서대로 리더기로 전송한다. 리더기는 태그가 전송한 식별자정보를 수신함과 동시에 수신된 비트의 충돌발생 여부를 검사한다. 이때, '0'과 '1'이 동시에 수신되어 충돌이 발생하면 나머지 비트의 수신을 중지하고 태그로 식별자 전송중지 명령을 보내고 모두

'0' 혹은 '1'이 수신되면 나머지 비트의 수신을 계속 한다 모든 비트에서 충돌 없이 식별자정보의 마지막 비트까지 수신되면 한 개의 태그가 식별된다 충돌이 발생했을 경우, 기존 프리픽스에 단순히 '0' 혹은 '1'을 추가하는 트리-워킹 알고리즘이나 쿼리 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 태그식별자 일부 비트를 리더기에 저장하는 메모리 구조는 스택이나 큐를 사용할 수 있으며 사용하는 메모리구조에 따라 깊이 우선탐색이나 너비우선탐색이 된다 한 개의 태그가 식별되면 영역내의 나머지 태그를 식별하기 위하여 스택이나 큐에서 새로운 프리픽스를 가져오게 된다. 이러한 과정은 영역내의 모든 태그가 식별될 때까지 반복된다 스택이나 큐가 비어있으면 전체 태그식별 과정은 종료되며 이는 영역내의 모든 태그가 식별되었음을 의미한다.

IV. 슬롯 알로하 기반 충돌방지 알고리즘

슬롯 알로하 기반 충돌방지 알고리즘은 알로하 프로토콜에 기반을 두고 있으며 I-Code 알고리즘, STAC 알고리즘과 Bit-Slot 알고리즘을 들 수 있다

I-Code 알고리즘에서 하나의 리더기 사이클 즉, 질의-응답 과정은 여러 개의 슬롯으로 구성되는 하나의 프레임으로 정의된다 태그식별을 위해 리더기는 먼저 $\langle I, rnd, N \rangle$ 정보를 인자로 하여 태그들로 전송한다 (I : 태그 식별자 범위, rnd : 랜덤값 생성에 필요한 종자(seed)값, N 한 프레임 내의 슬롯 개수) 이때, 태그들은 프레임에서 임의로 하나의 슬롯을 선택하고 그 슬롯에 식별자정보를 적재하여 리더기로 전송한다. 리더기는 프레임의 각 슬롯에 적재된 식별자 값을 사용하여 태그들을 식별하게 된다. 이러한 과정은 영역내의 모든 태그들이 식별되어졌다고 판단될 때까지 반복된다 위의 식별 과정에는 프레임의 크기 즉, 한 프레임 내의 슬롯 개수(N)와 식별 종료시점에 따른 두 가지 문제점이 제기된다

첫째, N 의 결정시 지나치게 큰 N 은 타임 슬롯의 낭비를 초래하게 되고 지나치게 작은 N 은 태그들 사이의 충돌 발생률을 증가시킨다. 따라서 적합한 N 값의 결정은 최적의 성능을 위해서 필요하며 I-Code 알고리즘에서는 다음과 같은 방식을 사용한다 매 리더기 사이클마다 태그로부터 응답받은 프레임 내의 각 슬롯들은 1) 비는 경우, 2) 한 개의 태그 식별자만이

표 1 프레임크기 테이블^[7]

N	1	4	8	16	32	64	128	256
n_{Low}	-	-	-	1	10	17	51	112
n_{High}	-	-	-	9	27	56	129	Inf

적재되어 있는 경우, 3) 여러 개 태그들의 식별자가 적재되어 있는 경우와 같은 세 가지로 구분할 수 있다. 따라서 이는 다음과 같이 표현할 수 있다 리더기가 수신한 프레임은 위의 각 경우에 준하여 $\langle c_0, c_1, c_k \rangle$ 로 표현되는 슬롯개수의 분포를 갖는다 여기서, c_0 는 비어 있는 슬롯의 개수, c_1 은 한 개의 태그 식별자만 전송되어진 슬롯의 개수, 그리고 c_k 는 여러 개의 태그 식별자정보들이 전송되어진 슬롯의 개수를 나타낸다. 한 리더기 사이클에서 리더기는 슬롯개수의 분포를 이용하여 식별영역 내 태그개수의 최소한계치 (minimum bound)를 n 이라고 할 때, $n = c_1 + 2c_k$ 관계를 이용하여 산출한다 산출된 n 은 프레임크기 테이블^[7]인 표 1의 n_{Low} 와 n_{High} 값들과의 비교를 통하여 다음번 질의-응답과정에 사용될 새로운 프레임의 크기 즉, 프레임을 구성하는 슬롯 개수 N 을 결정한다. 예를 들어 산출된 n 이 15이면 표에 의하여 N 은 32로 설정된다. 이 경우, n 은 표에서 N 이 32일 때 n_{Low} 와 n_{High} 의 범위에 속한다. 즉 $(n_{Low} \ 10) < (n: \ 15) < (n_{High} \ 27)$ 이므로 N 을 32로 결정한다.

둘째, 확률적 기법에 기반을 둔 I-Code 알고리즘은 태그식별이 완료된 시점을 정확히 파악하기 어렵다 I-Code 알고리즘에서는 태그 식별과정을 동질 마코브 과정(homogeneous markov process)을 이용한 모델을 도입하여 그 해결책을 제시하고 있다 즉, s 사이클 후에 영역내의 모든 태그들이 식별될 확률을 $Q^s_q(0)[n]$ 으로 정의한다^[7]. 여기서 Q 는 변환행렬 (transformation matrix)이고 Q^s 는 s 사이클 후의 변환행렬을 의미한다 $q(0)$ 는 초기 상태이고 n 은 영역내의 태그개수의 예측 값이다. 예를 들어 N 이 64, n 이 40일 경우, 9번의 사이클을 종료한 후 40 개의 태그 모두를 식별할 수 있는 확률은 96.45%이다. 반복적으로 이루어지는 태그 식별과정은 아래와 같은 조건을 만족하면 종료된다

$$Q^s_q(0)[n] > a \quad (a \geq 0.99) \quad (1)$$

STAC 알고리즘의 경우 리더기는 빈 슬롯이나 충돌발생 슬롯을 감지하면 해당 슬롯 전송을 중지하고 새로운 슬롯 전송을 시작하게 하는 전송중지 명령 (close slot sequence)을 태그로 전송함으로써 불필요

한 전송 오버헤드를 줄여 성능을 향상시킨다

I-Code 알고리즘과 STAC 알고리즘에서 프레임은 N 개의 슬롯으로 구성되고 빈 슬롯을 제외한 각 슬롯에는 태그 식별자들이 적재된다 즉, 슬롯의 비트크기는 태그식별자의 비트크기와 같다 따라서 이러한 알고리즘들을 ID-Slot 알고리즘이라고 부른다

Bit-Slot 알고리즘의 경우 프레임은 특별한 비트들로 구성되어 알고리즘의 동작과정은 다음과 같다 리더기의 질의에 영역내의 태그들은 태그식별자 크기를 갖는 임의의 랜덤 값을 생성하여 리더기로 전송한다. 생성된 값은 오직 한 비트만 '1'을 가지고 나머지 비트는 모두 '0'인 특별한 값을 갖는다 리더기는 수신한 프레임을 비트 순서대로 검사한다 검사결과 해당 비트 위치에서 '1'이 없는 경우는 응답이 없는 경우이고, 두개 이상의 '1'이 전송된 경우는 충돌이 발생한 경우이다 오직 한 개의 '1'만 전송된 경우는 태그를 식별할 수 있는 경우로서 수신한 랜덤값을 영역내의 태그로 다시 전송한다 해당 랜덤값을 전송한 태그만 자기의 태그식별자를 리더기로 전송하면서 태그식별이 이루어진다 태그식별 과정에서 영역 내 다수개의 태그 중 한 개의 태그를 선택하는 과정을 태그 단일화(singulation)과정이라고 한다 태그 단일화과정이 완료되면 선택된 태그는 식별자를 리더기로 전송한다

Bit-Slot 알고리즘은 ID-Slot 알고리즘과 달리 태그 식별과정을 태그 단일화과정과 태그 식별자전송과정으로 구분하며 태그 단일화과정에서 프레임의 크기는 태그식별자의 비트길이에 같으며 프레임은 비트단위로 의미를 갖는 특별한 값으로 구성된다 따라서 이를 ID-Slot 알고리즘과 구분하여 Bit-Slot 알고리즘이라고 부르며 프레임의 크기가 작기 때문에 태그식별 속도가 빠르다

V. 성능평가

본 절에서는 위에서 살펴본 트리 기반 충돌방지 알고리즘과 슬롯 알로하 기반 충돌방지 알고리즘에 대한 성능평가를 실시한다 성능평가를 위한 시뮬레이션 환경은 C언어를 기반으로 설계하였다

슬롯 알로하 기반 충돌방지 알고리즘들은 태그식별자의 값과 무관하게 영역내의 태그개수와 설정된 프레임의 크기에 따라 성능이 결정된다. 식별자의 비트길이의 증가는 태그식별 시간의 증가를 수반하지만 알고리즘들의 특성과는 무관하다. 이와 반대로 트리 기반 알고리즘들은 태그식별자의 값에 따라 태그 식

별과정이 결정되며 태그식별자의 비트길이에 따라 성능의 변화가 크다 따라서 슬롯 알로하 충돌방지 알고리즘들은 식별자 비트길이가 96일 경우만 고려하였다 또한 기존 8, 16, 32-비트의 식별자들과 달리 모든 사물에 식별자를 부여하기 위하여 현재 EPCGlobal에서는 96-비트의 EPC에 대하여 국제 표준화를 추진 중에 있으며 본 절에서 알고리즘에 대한 성능분석은 이를 기준으로 96-비트 태그 식별자를 사용하였다.

본 절에서는 우선 8, 16, 32-비트 식별자를 가질 경우 트리 기반 충돌방지 알고리즘들의 성능을 비교하였으며 식별자 비트길이가 성능에 미치는 영향을 알아보았다. 다음으로, 트리 기반 충돌방지 알고리즘과 슬롯 알로하 기반 충돌 방지 알고리즘에 대하여 성능분석을 수행하였으며 그 방법은 다음과 같다 트리 기반 충돌방지 알고리즘의 경우 태그개수는 최대 65,536(2^{16})개까지 고려하였다. 이는 영역내의 태그개수가 일반적으로 최대 4,000~8,000($2^{12} \sim 2^{13}$)개 인 것을 고려하여 그 범위를 좀 더 확장한 것이다 태그개수를 2에서부터 매번 2배씩 65,536(2^{16})개 까지 증가시키면서 질의-응답 횟수, 전송비트 수와 초당 태그 식별개수를 산출하여 성능분석을 수행하였다. 슬롯 알로하 기반 충돌방지 알고리즘의 경우 태그개수는 최대 2,048(2^{11})개까지 고려하였다 슬롯 알로하 기반 충돌방지 알고리즘에서 태그개수와 프레임의 크기는 서로 연관되어 있고 대역폭의 제한으로 최대 프레임의 크기를 512개 슬롯으로 설정하였다. 태그개수를 2에서부터 매번 2배씩 2,048(2^{11})개까지 증가시키면서 전송프레임 수, 전송비트 수와 초당 태그 식별개수를 산출하여 성능분석을 수행하였다. 마지막으로 충돌방지 알고리즘들의 초당 태그 식별개수를 비교하였다 초당 태그 식별개수 산출 시 사용된 비트 전송율은 80Kbps이고 질의명령은 8비트, 무응답 및 충돌 발생 시, 검출과 처리에 필요한 시간은 각각 3비트 시간으로 계산하였다^[8,10,11] 성능평가에 사용된 수치는 10회 반복적인 실험을 통해 얻은 값들을 평균한 값이다

1 식별자 길이의 변화에 따른 트리 기반 충돌방지 알고리즘들의 성능 비교

그림 2는 식별자 비트길이가 8, 16, 32일 경우 트리 기반 충돌방지 알고리즘들의 태그 당 질의-응답 횟수와 태그 당 전송비트 수의 변화를 나타내었다. 그림의 태그 당 질의-응답 횟수의 변화에서 볼 수 있는 바와 같이 다음과 같은 두 가지를 확인할 수 있었다 첫째, 식별자 비트길이의 증가에 따라 쿼리 트리

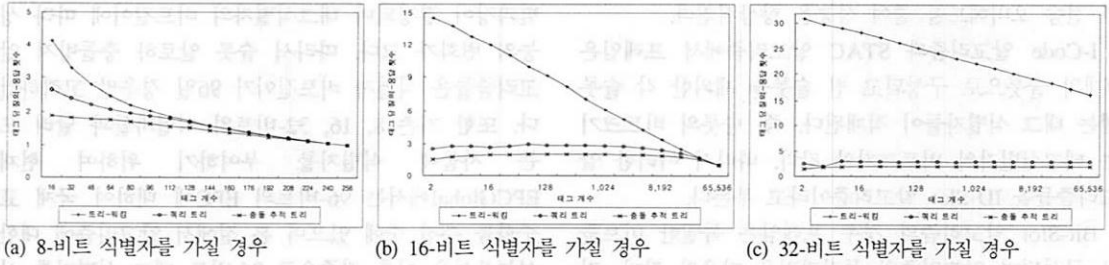


그림 2. 트리 기반 메모리래스 충돌방지 알고리즘의 태그 당 질의-응답 횟수의 비교

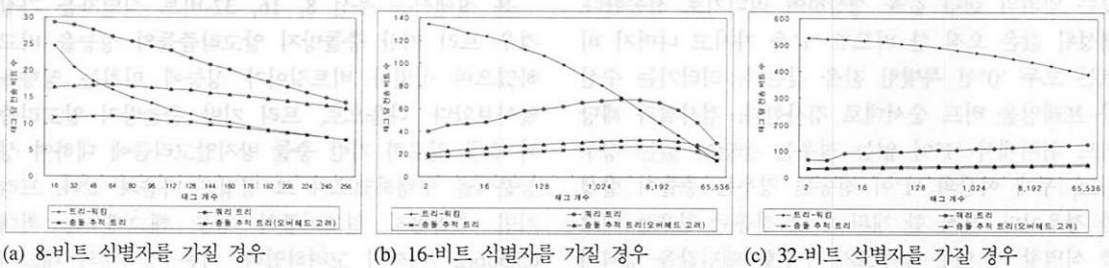


그림 3. 트리 기반 메모리래스 충돌방지 알고리즘의 태그 당 전송비트 수 비교

(QT) 알고리즘, 충돌 추적 트리(CT) 알고리즘과 트리-워킹(TW) 알고리즘의 태그 당 질의-응답 횟수의 차이가 점점 커짐을 볼 수 있다. 따라서 태그 식별자의 비트길이가 클 경우 트리-워킹 알고리즘은 비효율적이란 것을 알 수 있다. 둘째, 태그개수가 증가함에 따라 쿼리 트리 알고리즘, 충돌 추적 트리 알고리즘에서의 태그 당 질의-응답 횟수는 큰 변화가 없지만 트리-워킹 알고리즘에서는 변화가 큰 것을 볼 수 있고 이러한 변화는 태그개수가 증가함에 따라 감소하는 것으로 나타났다. 따라서 트리-워킹 알고리즘은 태그개수가 많을수록 더 효율적이란 것을 알 수 있다.

그림 3은 태그 당 전송비트 수의 변화를 나타내었다. 그림에서 충돌 추적 트리 알고리즘이 모든 경우에서 가장 우수함을 볼 수 있는데 이는 충돌 추적에 따른 오버헤드를 고려하지 않았기 때문이다. 오버헤드를 고려한 충돌 추적 트리(CTo) 알고리즘은 충돌 발생시 감지와 처리에 각각 3비트의 시간을 소요한다고 가정하였다. 8-비트의 태그식별자를 가질 경우 태그개수가 64개 이하 일 때에는 CTo(가장 우수함) > TW > QT 순으로 나타났다가 64개 이상일 때에는 TW > CTo > QT 순으로 나타났다. 16-비트의 태그 식별자를 가질 경우 태그개수가 2,048개 이하 일 때에는 CTo > QT > TW 순으로 나타났다가 2,048개 이상일 때에는 CTo > TW > QT 순으로 나타났다가 32,768개 이상일 때에는 TW > CTo > QT 순으로 트리-워킹 알고리즘이 가장 우수하게 나타났다. 32-비트의 태그식별자를 가질 경우

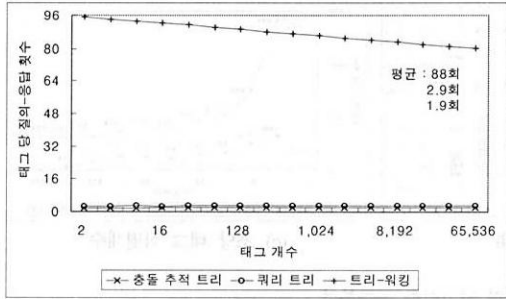
태그개수에 무관하게 CTo(가장 우수함) > QT > TW 순으로 충돌 추적 트리 알고리즘이 가장 우수한 것으로 볼 수 있다. 따라서 태그식별자의 비트길이가와 영역내의 태그 개수에 따라 적절한 알고리즘을 선택하는 것이 바람직하나 태그식별자의 비트길이가 32를 초과할 경우 충돌 추적 트리 알고리즘을 우선 고려해야 함을 알 수 있다.

2. 트리 기반 충돌방지 알고리즘의 성능분석

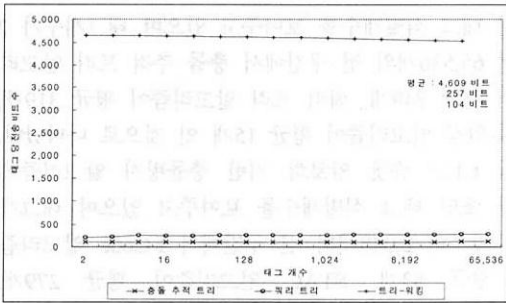
트리 기반 충돌방지 알고리즘에서 질의-응답 횟수와 전송비트 수는 알고리즘의 성능을 결정하는 중요한 요소이다. 본 소절에서는 96-비트 태그식별자를 가진 경우 태그개수의 증가에 따른 질의-응답 횟수와 전송비트 수의 변화를 차례로 살펴본다.

질의-응답 횟수는 영역내의 태그식별을 위한 리더기와 태그들 간의 통신횟수로서 한번의 질의-응답 과정을 1회로 정의한다. 단, 무응답 질의가 존재하여 질의 횟수가 응답 횟수보다 많은 쿼리 트리 알고리즘의 경우 질의 회수를 질의-응답 횟수로 간주한다.

그림 4.a는 각 알고리즘별 태그 당 질의-응답 횟수를 보여주고 있으며 영역내의 96-비트 태그를 식별하는데 충돌 추적 트리 알고리즘이 1.9회, 쿼리 트리 알고리즘이 2.9회 인 것으로 나타났다. 트리-워킹 알고리즘의 경우에는 88회가 소요됨으로써 다른 알고리즘에 비해 훨씬 높게 나타나고 있다. 이는 트리-워킹 알고리즘의 질의-응답 과정이 태그 식별자의 비트별로



(a) 태그 당 질의-응답 횟수



(b) 태그 당 전송비트 수

그림 4. 96-비트 식별자를 가질 경우 트리 기반 메모리리스 충돌방지 알고리즘의 성능분석

반복 수행되면서 질의-응답 횟수가 태그의 식별자 비트길이에 비례하여 증가하기 때문이다. 표 2는 태그의 응답결과에 따른 태그 당 질의-응답 횟수의 점유율을 비교하여 보여주고 있다. 그림과 같이 트리-워킹 알고리즘에서는 충돌이 발생하지 않은 경우가 98.92%로서 대부분을 차지하는 것을 볼 수 있으며 이는 전체 비트 공간에 비해 영역 내 존재하는 태그의 개수가 적고 태그식별이 이루어질 때까지 비트단위로 질의-응답 과정이 이루어지는 알고리즘의 특성상 대부분의 질의-응답은 충돌이 발생하지 않기 때문이다. 이러한 문제를 해결하기 위하여 쿼리 트리 알고리즘은 태그식별자 전체를 리더기로 전송하고 있지만 무응답과 잦은 충돌이 발생하는 문제가 있다. 그림과 같이 무응답이 16%, 충돌발생이 49%를 차지하는 것을 볼 수 있다. 이에 비해 충돌 추적 트리 알고리즘은 충돌 추적 기법을 적용함으로써 쿼리 트리 알고리즘에 비해 충돌이 발생한 질의-응답 횟수를 1.54배 ($1.4/0.91=1.54$) 줄이고 무응답이 발생한 질의-응답 횟수를 제거함으로써 전체 질의-응답 횟수에서 쿼리 트리 알고리즘에 비해 우수하게 나타나고 있다.

전송비트 수는 앞 절에서 언급한 질의비트 수(Q)와 응답비트 수(R)의 합으로서 질의 시 인자로 사용한 프리픽스의 비트길이와 응답 시 리더기로 전송한 식

표 2. 태그 당 질의-응답 횟수 점유율

알고리즘		트리-워킹 알고리즘	쿼리 트리 알고리즘	충돌 추적 트리 알고리즘
충돌이 발생한 경우	횟수	0.95	1.40	0.91
	백분율	1.08%	49%	48.10%
충돌이 발생하지 않은 경우	횟수	87.05	1.00	1.00
	백분율	98.92%	35%	51.90%
응답이 발생하지 않은 경우	횟수	-	0.45	-
	백분율	-	16%	-

표 3. 태그 당 전송비트 수 점유율

알고리즘		트리-워킹 알고리즘	쿼리 트리 알고리즘	충돌 추적 트리 알고리즘
질 의	비트 수	4,521	26	17
	백분율	98.10%	10.11%	16.14%
응 답	비트 수	88	231	87
	백분율	1.90%	89.89%	83.86%

별자정보의 비트길이의 합을 의미한다. 그림 4.b는 각 알고리즘별 태그 당 전송비트 수를 보여주고 있으며 영역내의 96-비트 태그를 식별하는데 충돌 추적 트리 알고리즘이 104비트, 쿼리 트리 알고리즘이 257비트가 전송되는 것으로 나타났다. 트리-워킹 알고리즘의 경우에는 4,609비트가 전송됨으로써 다른 알고리즘에 비해 훨씬 높게 나타나고 있다. 이는 트리-워킹 알고리즘의 높은 질의-응답 횟수와 식별과정이 비트 단위로 이루어지는 알고리즘의 특성상 태그로 전송하는 프리픽스가 식별이 이루어질 때까지 한 비트씩 증가하기 때문이다. 표 3은 각 알고리즘별 태그 당 질의/응답 비트 수 및 백분율을 비교하여 보여준다. 앞에서 설명하였듯이 트리-워킹 알고리즘은 태그식별이 이루어질 때까지 프리픽스 정보를 매번 한 비트씩 증가하면서 질의-응답 과정을 반복하기 때문에 한 비트씩 전송하는 응답에 비해 질의에 의해 전송되는 비트 수가 많다. 그림과 같이 태그 당 응답비트 수가 평균 88비트 인 것에 비해 질의비트 수가 4,521비트로서 전체의 98.10%를 차지하는 것을 볼 수 있다. 쿼리 트리 알고리즘의 경우 태그 당 응답비트 수가 평균 231비트인데 반해 충돌 추적 트리 알고리즘은 87비트로서 낮은 전송량을 갖는데 이는 태그 식별자정보 전체를 전송하여 태그를 식별하는 쿼리 트리 알고리즘에 비해 충돌 추적 트리 알고리즘은 충돌이 발생하면 전송을 중지하기 때문이다.

3. 슬롯 알로하 기반 충돌방지 알고리즘의 성능분석

그림 5는 슬롯 알로하 기반 충돌방지 알고리즘에 대한 성능분석의 결과를 나타내고 있다. 그림 5.a는 태그개수 증가에 따른 전송프레임 수의 변화를 나타

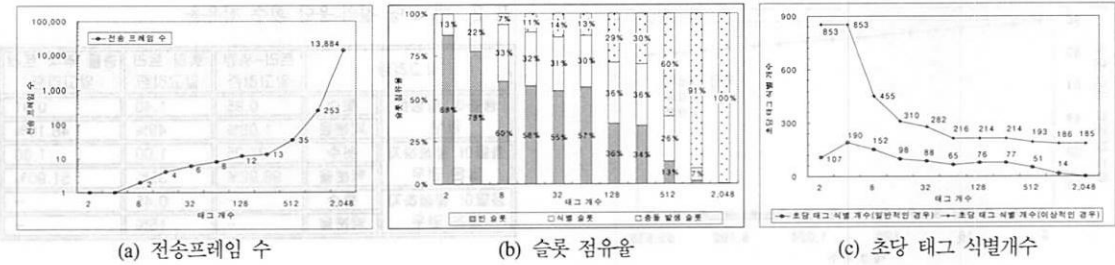


그림 5. 슬롯 할로하 기반 충돌방지 알고리즘 성능분석

내고 있다. 그림과 같이 태그개수가 512개 이상 일 경우 전송프레임 수가 급격히 증가함을 볼 수 있다. 이는 그림 5.b에서 볼 수 있듯이 태그개수가 512일 경우 전송된 프레임에서 충돌이 발생한 슬롯이 전체의 60%, 1,024개 일 경우 91%, 2,048일 경우 거의 100%에 달하는 것으로 나타나고 있기 때문이다. 그리고 태그개수가 64개 이하 일 경우 빈 슬롯이 전체 슬롯의 55% 이상을 차지하고 태그개수 전 구간에서 식별슬롯 최대 개수가 전체의 36%이하로서 대부분의 슬롯을 낭비하고 있음을 알 수 있다. 그림 5.c는 일반적인 경우와 이상적인 경우 초당 태그 식별개수를 보여주고 있다. 여기서 일반적인 경우라고 함은 빈 슬롯이나 충돌발생 슬롯에 관계없이 전체 프레임을 모두 전송함을 의미하고 이상적인 경우는 프레임 중 오직 한 개의 태그 식별자정보만 적재되어 식별이 이루어지는 슬롯만 전송할 경우를 의미한다. 초당 태그 식별개수는 전송비트 수를 비트 전송율로 나눔으로서 계산되었다. 충돌발생 슬롯이 대부분(60% 이상)을 차지하는 경우를 제외하면 그림과 같이 일반적인 경우 초당 평균 태그 식별개수는 최고 190에서 최저 77개 인 것으로 나타났다.

4. 초당 태그 식별개수

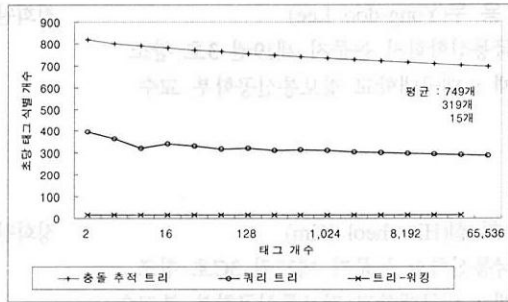
초당 태그 식별개수는 충돌 방지 알고리즘의 성능을 비교할 수 있는 궁극적인 척도로서 영역내의 태그들을 식별하기 위해 필요한 리더기와 태그들 간의 통신량에 의하여 계산된다. 이러한 통신량은 시뮬레이션에서 얻어낸 질의-응답 횟수(전송 프레임 수), 전송비트 수와 통신오버헤드에 의해 결정된다. 통신 오버헤드를 포함한 실제 전송된 총 비트 수를 계산하여 이를 비트 전송율로 나눔으로서 초당 태그 식별개수를 계산하였다.

그림 6은 식별영역 내 태그개수의 증가에 따른 각 알고리즘들의 초당 태그 식별개수를 나타내고 있다. 그림 6.a는 트리 기반 충돌방지 알고리즘들에 대한

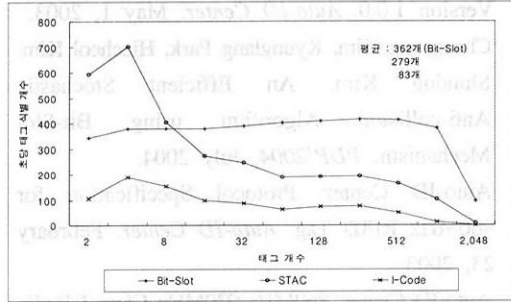
초당 태그 식별개수를 보여주고 있으며 태그개수가 2부터 65,536개의 전 구간에서 충돌 추적 트리 알고리즘이 평균 749개, 쿼리 트리 알고리즘이 평균 319개, 트리-워킹 알고리즘이 평균 15개 인 것으로 나타났다. 그림 6.b는 슬롯 할로하 기반 충돌방지 알고리즘에 대한 초당 태그 식별개수를 보여주고 있으며 태그개수가 2부터 2,048개의 전 구간에서 I-Code 알고리즘이 평균 83개, STAC 알고리즘이 평균 279개, Bit-Slot 알고리즘이 평균 362개 인 것으로 나타났다. 따라서 위에서 살펴본 충돌방지 알고리즘들 중 충돌 추적 트리 알고리즘이 초당 749개의 태그를 식별하면서 가장 우수한 것으로 나타났다.

VI. 결론

본 논문에서는 저비용 RFID 시스템에서 96-비트 식별자를 갖는 EPC 코드를 기반으로 트리 기반 메모리리스 충돌방지 알고리즘들과 슬롯 할로하 기반 충돌방지 알고리즘들을 소개하고 성능평가 결과를 제시하였다. 성능평가 결과 트리-워킹 알고리즘은 태그 식별자 비트길이가 증가할수록 질의-응답 횟수와 전송비트 수가 과도하게 늘어나면서 성능이 저하되고 쿼리 트리 알고리즘은 태그 식별자정보 전체를 전송함에도 불구하고 충돌 감지 기법을 사용하기 때문에 높은 성능을 기대하기 어렵다. 프레임크기의 설정에 따라 태그식별 성능이 좌우되는 I-Code 알고리즘, STAC 알고리즘은 일반적으로 영역내의 태그개수가 적은 경우에 사용가능하나 충돌발생 슬롯이나 빈 슬롯에 따른 통신오버헤드 때문에 역시 높은 성능은 기대하기 어렵다. Bit-Slot 알고리즘은 프레임의 크기가 작기 때문에 I-Code 알고리즘이나 STAC 알고리즘보다 성능이 우수한 반면 태그 식별과정을 태그 단일화과정과 태그 식별자전송과정으로 구분하기 때문에 역시 높은 성능을 기대하기 어렵다. 마지막으로, 트리 기반 메모리리스 알고리즘인 충돌 추적 트리 알고리즘은 충돌



(a) 트리 기반 메모리리스 충돌방지 알고리즘



(b) 슬롯 알고리즘 기반 충돌방지 알고리즘

그림 6. 충돌방지 알고리즘의 초당 태그 식별개수 비교

이 발생한 위치를 정확히 추적하여 질의-응답을 수행함으로써 질의-응답 횟수나 전송비트 수에서 다른 알고리즘보다 훨씬 우수한 성능을 보여주었다. 시뮬레이션 결과 충돌 추적 트리 알고리즘이 최대 65,536개의 96-비트 태그를 식별함에 있어서 평균 초당 749개의 태그를 식별할 수 있었으며 이는 다른 알고리즘들에 비해 최소 2배에서 최대 50배 이상의 높은 성능이다.

RF 통신은 전력공급, 연산능력 등 많은 면에서 기존의 무선통신과 차별되며 이는 기존 무선통신에서 연구 개발된 다양한 하드웨어 구현 기법들의 적용을 어렵게 한다. 따라서, 고성능 충돌방지 알고리즘들을 효과적으로 지원하기 위한 연구가 요구된다. 현재 전 세계적으로 RFID용 주파수 할당, 통신방식 등 관련 국가간의 기술규격은 많은 차이를 보이고 있다. 특히, RFID용 주파수대역에서 국가마다 서로 다른 대역 (ISO : 860~960MHz대역, 미국 : 902~928MHz대역, 유럽 : 865~870MHz대역, 국내 : 908~914MHz대역)을 UHF대역으로 할당하면서 RFID 시스템은 다양한 동작주파수대역을 지원할 수 있어야 한다. 또한 RFID 시장은 향후 4~5년에 걸쳐 표준화가 이루어질 것으로 예상되면서 기존에 개발된 다양한 충돌방지 알고리즘을 지원하는 프로토콜들이 공존하게 되며 이는 RFID 시스템 구현에 있어서 다중 프로토콜 지원 기능도 포함하여야 함을 의미한다. 따라서 향후 태그 및 리더기의 신호전달 체계 및 하드웨어 구현에 대한 연구와 함께 다중 밴드/프로토콜 지원 RFID 시스템 개발을 진행할 계획이다.

참 고 문 헌

- [1] 권성호, 김희철. EPC 네트워크를 위한 다중 RFID 태그 식별 알고리즘의 분석. *한국인터넷정보학회*, Vol. 4, No. 4, Pages 27-37. December 2003.
- [2] Hush, Don R. and Wood, Cliff. Analysis of Tree Algorithms for RFID Arbitration. *In IEEE International Symposium on Information Theory*, Pages 107-. IEEE, 1998.
- [3] Jacomet M, Ehram A, Gehrig U. Contactless identification device with anti-collision algorithm. *IEEE Computer Society, CSCC '99, Conference on Circuits, Systems, Computers and Communications*, Athens. 4-8 July 1999.
- [4] 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, Pages 103-111. 2003.
- [5] 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*, Pages 75-84. ACM. August 2000.
- [6] 권성호, 박경량, 김희철. 충돌 추적 기법을 적용한 다중 태그 식별 알고리즘의 설계 및 구현. *대한전자공학회계중합학술대회*, Vol. 27, No. 1, Pages 679-682. June 2004.
- [7] Vogt, H. Efficient Object Identification with Passive RFID Tags. *In International Conference on Pervasive Computing, LNCS*. Springer-Verlag. 2002.
- [8] Auto-ID Center. 13.56MHz ISM Band Class 1 Radio Frequency Identification Tag Interface Specification: Candidate Recommendation,

