

RFID 태그 충돌 방지 프로토콜의 성능 비교를 위한 시뮬레이션 시스템*

신재동^o, 여상수, 김성권

중앙대학교 컴퓨터공학부, 단국대학교 정보컴퓨터학부, 중앙대학교 컴퓨터공학부
mulliz^o@alg.cse.cau.ac.kr, ssyeo@dankook.ac.kr, skkim@cau.ac.kr

Simulation System for Anti-Collision Protocols in RFID

Jae-Dong Shin^o, Sang-Soo Yeo, Sung Kwon Kim
School of Computer Science and Engineering, Chung-Ang University
School of Information and Computer Science, Dankook University
School of Computer Science and Engineering, Chung-Ang University

요 약

RFID(Radio Frequency Identification) 기술은 RF 신호를 사용하여 물품에 부착된 전자 태그(tag)를 식별하는 비접촉 자동인식 기술이다. 이런 RFID 기술의 확산을 위해서는 리더가 다수의 태그를 빠른 시간 안에 인식하는 다중 태그 식별 문제를 해결 해야만 한다. 지금까지 이 문제를 해결하기 위한 충돌 방지(anti-collision) 알고리즘이 많이 개발되었다. 본 논문에서는 이런 충돌 방지 알고리즘 중에서 18000-6 Type A, Type B와 Type C로 새롭게 채택된 EPCglobal Class1 Generation2, 그리고 Query Tree Protocol에 대한 시뮬레이션 시스템의 구조를 소개하고 실제 시뮬레이션 했을 때의 각 알고리즘에 대한 간단한 성능 평가를 한다.

1. 서 론

RFID(Radio Frequency IDentification) 기술은 RF 신호를 사용하여 물품에 부착된 전자 태그(tag)를 식별하는 비접촉 기술이다. 이러한 RFID 기술의 확산을 위해서는 태그의 저가격, 저전력, 초소형화 문제, 보안 및 프라이버시 문제, 태그 식별자의 코드 표준화 문제 그리고 다중 태그 식별 문제 등의 난제들을 우선 해결해야 한다. 이 중에서 다중 태그 식별 문제는 리더기(reader)의 식별 영역 내에 다수의 태그가 존재할 경우 태그의 정보를 충돌 없이 전송받아 식별해야하는 리더기와 태그 사이의 일대다 통신 문제로 정의된다. 이러한 태그 식별 문제는 충돌 방지(anti-collision) 알고리즘을 통하여 해결할 수 있고 이는 RFID 시스템에서 핵심이 되는 기술이다.[1]

이러한 충돌 방지 프로토콜은 크게 2가지, 확률적 충돌 방지(Stochastic Collision Resolution) 방법과 결정적 충돌 방지(Deterministic Collision Resolution) 방법으로 나눌 수 있다. 먼저 확률적 충돌 방지 방법은 보통 알로하(ALOHA) 알고리즘을 기반으로 하고 있다. 알로하 알고리즘은 시간을 슬롯 단위로 나누어 태그의 응답이 되도록 한 슬롯에는 하나의 태그만 응답하게 하는 알고리즘이다. 하지만 이 방법은 확률이라는 불확실성에 기초를 두고 있기 때문에 리더기 식별 영역 내의 모든 태그를 인식하지 못할 수도 있다. 한편 결정적 충돌 방지 방법은 트리(tree) 생성을 기반으로 하고 있다. 이 방법의 알고리즘은 트리를 만들 때 태그의 고유한 식별 ID를 사용하여 트리를 만들기 때문에 모든 태그를 인식할 수 있고

과정을 예측할 수 있는 장점이 있다.

본 논문에서는 시뮬레이션 할 알고리즘인 18000-6 Type A, Type B, Type C, Query Tree Protocol에 관하여 간단한 개념을 살펴보고 시뮬레이션 프로그램의 동작 방법과 환경, 그리고 실행 결과에 대하여 이야기한다.

2. 관련 연구

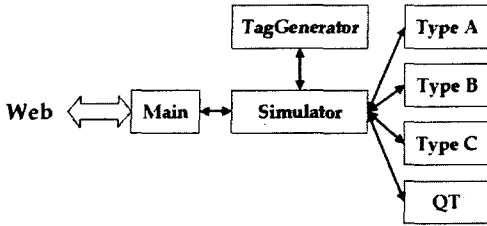
2.1 18000-6 Type A

18000-6 Type A[2]는 Framed Slotted ALOHA 알고리즘을 기반으로 하고 있다. 리더는 태그에게 질의를 하면서 프레임의 크기를 알려준다. 태그는 프레임의 크기 안에서 랜덤하게 슬롯(slot)을 결정하고 8비트의 슬롯 카운트 메모리에 기억한다. 그리고 태그는 자신이 선택한 슬롯 순서까지 기다리다가 대답을 하게 된다. 슬롯의 진행은 리더가 슬롯이 끝날 때마다 보내는 명령(Close_Slot, Next_Slot)을 보고 알 수 있다. 태그는 이 슬롯이 끝날 때 마다 보내는 명령을 보고 자신이 먼저 랜덤하게 선택한 슬롯과 비교해 차례를 알 수 있게 되는 것이다. 또한 전 프레임에서의 충돌수와 무응답수를 이용하여 프레임의 크기를 조정한다.

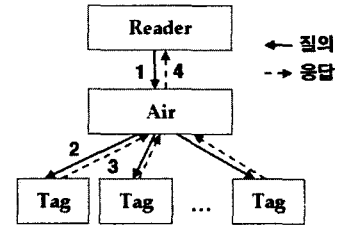
2.2 18000-6 Type B

18000-6 Type B[2]는 Binary-Tree Protocol을 기반으로 하고 있다. 이 프로토콜에서는 각 태그들이 카운트 메모리를 가져야한다. 리더는 태그에게 질의를 하면 태그는 랜덤하게 '0'과 '1'사이의 값을 생성하고 카운트에 저장한다. 이 때 카운트 값이 '0'인 태그들만 응답을 하는 게 원칙이다. 충돌이 나면 방금 응답했던 태그들을 상대로 다시 랜덤하게 '0'과 '1'사이의 값을 생성하게 하고 또 이 중에서 '0'인 태그들만 응답을 하게 한다. 여기

* 본 연구는 한국과학재단 특정기초연구(R01-2005-000-10568-0) 지원으로 수행되었음



[그림1] 시뮬레이션 시스템 구조



[그림2] 충돌 방지 알고리즘 구조

서 카운트 값이 '0'이 아니었던 태그들은 카운트 값을 1 증가해서 기다리게 된다. 이렇게 카운트 값이 '0'인 것 위주로 먼저 트리를 만들다 보면 한 개의 태그만 응답하게 되고 후에 카운트가 '0'이 아니었던 태그들의 카운트를 줄여 기다리던 태그들에게도 기회를 준다.

2.3 18000-6 Type C

18000-6 Type C 표준은 EPCglobal Class1. Gen2.[3]가 2005년 6월 확정되어 현재 표준화 진행 중이다. Type C 알고리즘은 Slotted Random Protocol이라고도 불리며 Type A의 변형으로 기본은 Type A와 같이 FS-ALOHA이다. Type A와 차이점은 크게 2가지가 있다. 먼저 슬롯 카운트 메모리가 15비트로 커져서 프레임의 크기가 크게 늘어났고 리더는 슬롯이 끝날 때 마다 명령(Query_Rep)을 보내고 이것을 받은 태그는 랜덤하게 선택했던 슬롯 카운트를 '1'씩 감소해서 '0'이 되었을 때 전송하는 방식을 택한다.

2.4 Query Tree Protocol

QT 프로토콜[4]은 결정적 충돌 방지 방법 중 대표적인 기술이다. QT 프로토콜은 모든 태그의 식별 ID가 유일하다는 데 착안한다. 그래서 이 ID를 사용하여 트리를 만드는 것이다. 리더는 태그에게 질의할 때 k 비트 프리픽스(P_k)를 함께 전송한다. 태그는 자신의 ID의 앞부분과 프리픽스가 같으면 자신의 ID를 리더에게 응답한다.

이 때 리더의 질의에 대하여 둘 이상의 태그가 동시에 응답하면 충돌이 발생한다. 리더는 충돌이 일어난 걸 알고, 결국 같은 프리픽스의 태그가 여러 개 있다는 것을 알게 된다. 그런 경우 방금 전송한 프리픽스 뒤에 '0'과 '1'을 붙인 새로운 $k+1$ 비트의 프리픽스(P_{k+1})를 만들어 큐(Queue)에 넣는다. 큐에 넣은 프리픽스는 나중에 다시 질의하게 된다. 그럼으로써 태그의 식별 ID를 가지고 트리를 만드는 것이다.

3. 시뮬레이션 시스템

3.1 시뮬레이션 시스템 구조

시뮬레이션 시스템의 프로그래밍 언어는 Python 2.4이며 CGI로 개발 되서 웹(<http://alg.cse.cau.ac.kr/sim/>)에서 접근 가능하다. 전체 구조는 [그림1]과 같다. 웹에서 Main 클래스에 접근하여 시뮬레이션을 요청한다. 이 때 Main 클래스는 시뮬레이션에 필요한 인자를 받아 Simulator 클래스에게 전달해준다. Simulator 클래스는 인자로 받은 태그 수만큼 TagGenerator 클래스를 사용

하여 모두 다른 태그 식별 ID를 만든다. 후에 요청받은 알고리즘을 각각의 알고리즘 클래스에게 수행토록 한다.

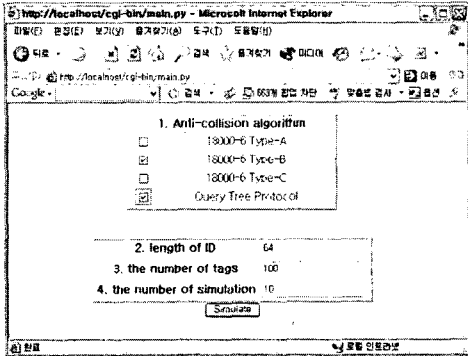
시뮬레이션 시스템의 모든 충돌 방지 알고리즘은 [그림2]와 같은 구조를 가진다. 먼저 실제 시뮬레이션에 들어가기 전에 각각의 Tag 클래스는 TagGenerator가 만든 고유 식별 ID를 가지도록 한다. 그리고 Air 클래스는 이런 Tag 클래스들을 가지고 있게 하고 Reader 클래스는 Air 클래스를 가지게 한다. 결국 Reader클래스는 Tag 클래스와 직접 메시지를 송수신하는 게 아니라 Air 클래스를 통해서만 Tag와 질의를 보내거나 응답을 받을 수 있다. 이렇게 Air 클래스가 Reader 클래스와 Tag 클래스 사이에 들어가는 이유는 Reader 클래스가 Tag 클래스들을 전혀 모르게 하기 위해서다. 시뮬레이션이 아닌 실제 환경에서 리더기가 태그가 몇 개인지도 모르고 충돌 방지 알고리즘을 하기 때문에 이와 유사하게 한 것이다.

시뮬레이션에 들어가면 Reader 클래스는 자신의 알고리즘을 수행하면서 태그에게 질의를 하고 싶으면 Air클래스에게 질의한다. 그러면 Air 클래스는 Reader에게 받은 질의의 그대로 자신이 가지고 있는 모든 Tag 클래스에게 똑같이 한다. Tag 클래스들은 각각 자신이 응답할 것이 있으면 유효한 응답을 하고 응답할 것이 없으면 무효한 응답을 한다. Air 클래스는 이 Tag 클래스들의 응답 중 유효한 응답들만 모으고 이를 다시 Reader 클래스에게 응답한다. 이러한 일련의 과정은 리더기와 태그 사이의 한 번의 질의-응답으로 간주된다.

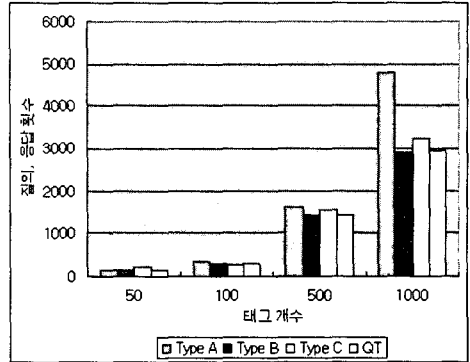
한편 Air 클래스에서 Reader 클래스로 보내는 Tag 클래스의 응답의 개수는 0개이면 무응답(no response), 1개이면 인식(identified), 2개 이상이면 충돌(collision)로 본다. 본 시뮬레이션 프로그램에서는 전파 방해는 없다고 가정한다.

3.2 시뮬레이션 시스템 실행 및 결과

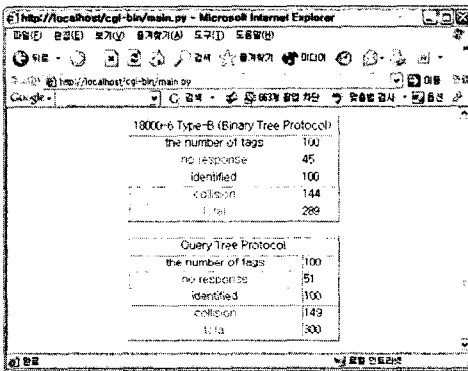
실제 시뮬레이션에 들어가는 인자는 [그림3]에서 보듯이 총 4가지이다. 첫째는 충돌 방지 알고리즘으로 어떤 알고리즘을 시뮬레이션 할 것인가를 선택한다. 이 때 복수 선택도 가능하게 이렇게 하면 같은 태그 데이터 집합을 가지고 시뮬레이션하게 되서 알고리즘 간 비교가 가능하다. 두 번째는 태그 식별 ID의 길이이다. 세 번째는 태그의 수로 실제 시뮬레이션에 들어가는 태그의 수이다. 마지막 인자는 시뮬레이션 횟수로 확률적 충돌 방지 방법인 경우에 매번 할 때 마다 결과가 다르게 나올 수 있기 때문에 여러 회의 평균치로 성능을 측정해야 한다. 마지막 인자는 이것을 위한 인자이다. 예를 들어 10회의 시뮬레이션을 하고 결과로는 10회의 평균이 나온다.



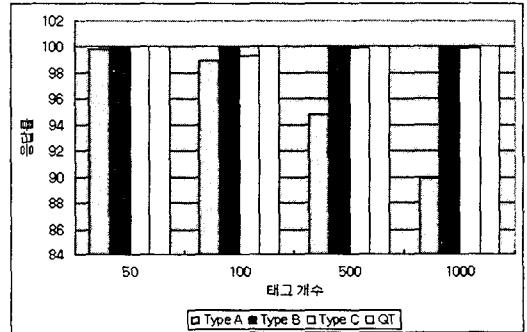
[그림3] 시뮬레이션 실행 화면



[그림5] 질의-응답 횟수



[그림4] 시뮬레이션 결과 화면



[그림6] 태그 인식률

[그림4]는 시뮬레이션 결과를 보여주는 웹페이지이다. 시뮬레이션 한 알고리즘의 이름과 시뮬레이션에 들어간 태그 수, 그리고 그 결과가 나오게 된다.

시뮬레이션 결과로는 무응답과 인식, 충돌의 횟수가 차례로 나온다. 이 때 인식 수가 시뮬레이션에 들어간 태그 수 보다 작을 수 있다. 이런 경우는 모두 인식이 안 된 경우이다. 마지막으로 총합(total)은 앞 결과 3가지의 합으로 총 질의-응답 개념으로 볼 수 있다. 즉, 총합이 클수록 리더가 태그들을 모두 인식하는 데 오래 걸린다고 볼 수 있다.

4. 시뮬레이션 결과

태그 식별 ID의 길이는 18000-6에서 가정하고 있는 64비트로 하고 태그 수를 늘려가며 각 알고리즘 간 비교를 하겠다. 각 알고리즘 당 시뮬레이션 횟수는 100회로 하였다.

시뮬레이션 결과는 [그림5]에서 보듯이 Type B와 QT 알고리즘이 태그가 많아질수록 가장 나은 성능을 보여주고 있다. 그리고 Type A는 프레임 크기가 최대 256슬롯 밖에 안 되서 태그 수가 많아지면 속도가 상당히 느려진다. 한편 확률적 충돌 방지 방법의 알고리즘은 태그 수가 많아지면 [그림6]과 같이 모든 태그를 인식하지 못하는 것을 알 수 있다.

5. 결론

기존에 나온 충돌 방지 프로토콜 중 18000-6의 Type A, Type B, Type C와 QT 프로토콜을 시뮬레이션할 수 있는 시스템을 살피고 실행 결과를 보았다. 이 시뮬레이션 시스템의 장점은 웹으로 공개되어 있기 때문에 누구나 접근하여 원하는 조건을 입력하여 시뮬레이션을 할 수 있다는 점이다. 또한 이 시스템을 향후 개발되는 충돌 방지 알고리즘을 기존의 알고리즘들과 쉽게 비교 분석할 수 있도록 설계되었다.

참고 문헌

- [1] 권성호, 홍원기, 이용두, 김희철, "RFID 시스템에서의 트리 기반 메모리리스 충돌방지 알고리즘에 관한 연구", 정보처리학회논문지 C 제 11-C권 wp6호, 2004.12
- [2] "KS X ISO/IEC 18000-6", ISO standard document, <http://www.ks.or.kr>
- [3] EPCglobal, "Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960MHz", Version 1.0.9, 2005.
- [4] Ching Law, Kayi Lee and Kai-Yeung Siu, "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.