

# RFID 시스템에서 다중 태그 인식을 위한 충돌회피 알고리즘의 성능 개선에 관한 연구

남궁호영, 민병택, 전준수\*, 김철성\*\*  
전남대학교 컴퓨터정보통신공학과, \*한국전자통신연구원  
\*\*전남대학교 전자컴퓨터정보통신공학부  
e-mail : hynamgung@naver.com

## Improvements of the Anti-collision Algorithms for Multi Tag Interrogation in RFID System

Ho-Young Namgung, Byoung-Taek Min, Jun-Soo Jeon\*, Cheol-Sung Kim\*\*  
Dept. of Computer & Information Engineering, Chonnam National University, \*ETRI,  
\*\*Dept. of Electronics Computer & Information Engineering, Chonnam National University

### Abstract

In this paper, we propose an improved anti-collision algorithm for multi tag interrogation in ubiquitous sensor network(USN) and show the result of simulation for multi tag interrogation in RFID systems. We have analyzed an EPC(Electronic Product Code) protocol which specifies the physical and logical requirements for a passive-backscatter Reader-Talk-First(RTF) RFID(Radio Frequency Identification) system operating in the 860MHz~960MHz frequency range. We have also designed and implemented the simulator of the RFID system based on the EPC protocol. Finally, we find that proposed algorithm works better than an existing algorithm.

### I. 서론

최근 RFID 국제 표준안이 확정되고 RFID 태그의 저가격 구현이 가능하게 되면서 물류 유통 분야를 중심으로 기존의 바코드를 대체하는 RFID의 다양한 산

\* 본 연구는 전남대학교 고품질 전기전자부품 및 시스템 연구센터의 연구비 지원에 의해 연구되었음.

업분야에 실용화가 가능한 상황에 접어들고 있다.

RFID는 마이크로 칩을 내장한 태그에 저장된 데이터를 무선 주파수를 이용하여 리더에서 자동 인식하는 기술로 비접촉식으로 여러 개의 태그를 동시에 인식할 수 있고, 인식시간이 짧고, 태그에 대용량의 데이터를 저장할 수 있고, 반영구적인 사용이 가능한 장점이 있다.

비접촉식, 대용량 데이터의 전송가능 등의 장점을 가진 RFID시스템이 기존의 기술을 대체하기 위해서는 보다 빠르게 많은 태그를 인식하여야 한다. 따라서 데이터의 고속 인식을 위해서는 데이터 사이의 충돌을 최소화하는 충돌 회피 알고리즘이 요구된다.

본 논문은 기존의 알고리즘의 태그인식속도를 줄여 성능을 향상시키기 위해 태그 상태와 플래그를 추가하여 EPC 규격에 부합하는 새로운 RFID 충돌회피 알고리즘을 제안한다.

### II. EPC Global protocol

EPC protocol은 860MHz~960MHz 주파수 영역에서 동작하는 RTF RFID 시스템의 물리적, 논리적 요구를 명세한다. 이 시스템은 리더와 태그가 통신 영역 안에 있으면 리더가 태그에게 통신을 하고 전력을 공급한다.

#### 2.1 태그의 구조

EPC는 타입 I부터 IV까지 4가지 타입이 있으며, 타입을 나타내는 두 비트(타입 IV의 경우 8비트)로 구성된 header와 domain manager, object class, serial number로 구성되어 있고, 총 ID의 크기는 타입 I, II, III는 64bit, 타입 IV의 경우는 96bit 이다.

태그는 ID 64bit(타입 IV의 경우 96bit)와 CRC(Cyclic Redundancy check) 16bit로 구성되어 있다. CRC는 태그 ID를 계산하여 구해지며, 리더로 전송된 태그 ID의 error check에 사용된다.

표 1 은 태그의 EPC 구조를 보여준다.

표 1. 태그 Type에 따른 EPC 구조

EPC TYPE	HEADER SIZE	FIRST BITS	DOMAIN MANAGER	OBJECT CLASS	SERIAL NUMBER	TOTAL
type I	2	01	21	17	24	64
type II	2	10	15	13	34	64
type III	2	11	26	13	23	64
type IV	8	00	28	24	36	96

### 2.2 EPC Class 0 Protocol

Class 0의 태그는 리더로부터 전송된 신호에 자신의 ID 한 bit 백스캐터(backscatter)를 하면서 리더와 태그간의 통신이 이루어진다.

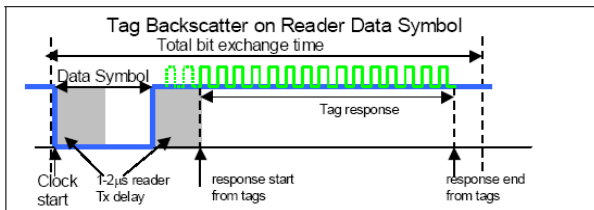


그림 1. 태그 백스캐터

태그는 리더로부터 1비트의 명령을 받게 되면 데이터 '0'과 '1'을 FSK(Frequency Shift Keying) 변조를 통해 그림1과 같이 백스캐터 하고, 리더는 데이터 '0'과 '1'을 서로 다른 안테나에서 수신한다. 이로써 리더는 태그로부터 백스캐터 신호를 인지하고, 충돌 여부를 판단하게 된다.

태그는 리더로부터 신호에 따라 자신의 상태를 변화시키면서 자신의 ID와 일치하는 신호를 받으면 태그에게 백스캐터를 통해 자신의 ID를 전송하여 리더에게 인식시킨다. 리더는 태그로부터 백스캐터 된 bit에 의해 태그에 전송할 다음 bit를 결정한다.

리더는 태그의 ID를 모두 인식할 때까지 반복 수행하고 태그는 자신의 ID와 일치하지 않는 신호를 리더로부터 받게 되면 자신의 state를 변경하여 더 이상 리더의 신호에 응답하지 않게 된다. 리더는 이런 일련의 과정을 태그의 응답이 더 이상 없을 때까지 계속 수행한다.

### 2.3 EPC Class 1 Protocol

Class 1은 리더와 태그간의 통신은 패킷으로 이루어지고 한 개의 패킷은 리더로부터는 하나의 완전한 명령을, 태그로부터는 하나의 완전한 응답을 포함한다.

리더-태그간의 통신은 단방향 통신으로 이루어진다.

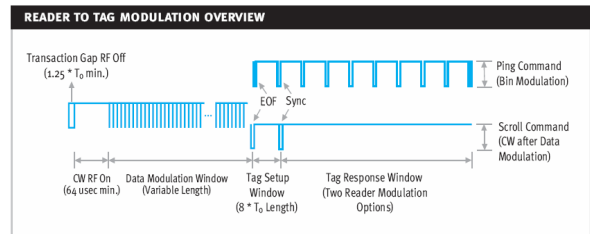


그림 2. Reader-to-Tag Logical Communication Phase

리더가 태그로 보내는 명령은 규정된 명령(required command)과 식별자를 프로그램하는 명령(identifier programming command)으로 나뉜다. 규정된 명령은 ScrollAllID, ScrollID, PingID, Quiet, Talk, Kill로 구성되어 있고, 식별자를 프로그램하는 명령어는 ProgramID, VerifyID, LockID, EraseID로 구성되어 있다. Class 1을 만족하는 태그는 자신 내부의 상태를 바꾸고 명령에 대한 응답으로 백스캐터 변조를 수행한다.

그림 2는 리더에서 태그로의 logical communication phase를 나타낸다. 리더가 태그로 통신을 하기 위해서는 우선 첫 번째 phase인  $1.25 \times T_0$ 의 transaction gap이 필요하며, 두 번째 phase는 태그에 전력을 공급하기 위하여  $64\mu s$ 의 continuous RF wave를 보낸다. 세 번째 phase는 태그에게 명령어를 보내는 Data Modulation Window이며 명령어에 포함될 VALUE나 LEN 등이 수시로 바뀌기 때문에 phase의 길이는 가변적이 된다. 4번째 phase는 태그가 응답을 하기 전의 준비 기간인 Tag Setup Window로 길이는  $8 \times T_0$ 가 되며 마지막 다섯 번째 phase는 태그가 리더로 응답을 하는 기간인 Tag Response Window이다. 북미 방식의 경우  $T_0$ 는  $14.25\mu s$ 이다.

## III. 기존의 충돌회피 알고리즘

### 3.1 EPC Class 0

EPC Class 0의 충돌회피 알고리즘은 bit-by-bit로 동작하는 binary tree 방식을 사용한다.

리더-태그 간 통신은 1bit씩 수행하게 되는데 리더에서 명령어의 한 비트를 보내면 태그는 수신한 비트를 리더에게 백스캐터 한다. 리더가 태그의 응답을 수신했을 때 충돌이 감지되면 리더는 임의의 가지

(branch)로 분기하게 되어 임의의 비트를 보냄으로써 해당하는 ID를 갖는 태그들만 응답하게 하여 태그를 구별하게 된다. 태그의 ID 전체를 수신했을 때 리더는 CRC를 검사하여 수신된 ID의 유효성을 판단한다.

이 방식은 충돌이 발생하여 인식되지 않았던 태그를 인식하기 위해 태그 ID의 첫 번째 비트부터 인식하는 과정을 반복하므로  $N$ 개의 태그를 인식하기 위해서는  $N \times 82$ (태그의 ID 64bit일 경우)개의 명령을 실행하므로 다수의 태그를 인식하는데 오랜 시간이 걸리는 단점이 있다.

### 3.2 EPC Class 1

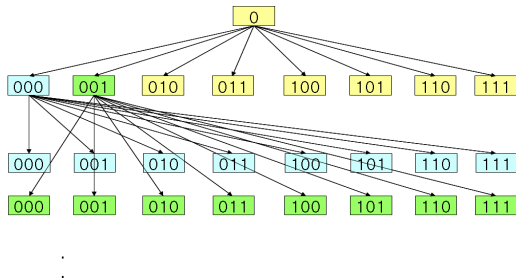


그림 3. Class 1의 충돌회피 알고리즘

Class 1의 충돌회피 알고리즘은 그림 3에서와 같이 리더는 태그의 인식을 위해 PingID 명령어를 보낸 후 8개의 Bin을 보내게 된다. 이때 태그는 자신의 첫 번째 3비트와 일치하는 Bin 기간 동안 응답을 하게 되며 리더는 하나의 Bin slot을 통하여 태그 응답의 충돌 여부를 검사하게 된다. 충돌 여부는 ScrollID 명령어를 보낸 후 오는 응답의 CRC를 검사하여 일치하지 않으면 충돌이 일어난 것으로 간주한다. table을 이용하여 충돌이 없는 Bin slot의 3비트를 저장하여 충돌이 나지 않을 때까지 PingID 명령어로 태그를 구분한 후 ScrollID 명령어를 보내어 태그를 인식하고 바로 후에 PingID를 보내는 것이 아니라 table에 저장되어 있는 태그의 ID에 대해 ScrollID 명령어를 보내게 된다.

## IV. 제안된 충돌회피 알고리즘

### 4.1 EPC Class 0

제안된 알고리즘은 기존의 알고리즘에 태그의 상태와 플래그를 추가시켜 충돌이 일어나기 전까지의 ID를 저장하여 다음 태그 인식 시에 리더의 명령 횟수를 줄여 태그의 인식시간을 단축시킨다.

제안된 알고리즘은 그림 4에서 태그1과 태그2의 백스캐터에 의해  $i+1$ 번째 비트에서 충돌 발생을 리더에서 인지하게 되면 리더는  $i$ 번째까지의 태그 ID를 저

장하고 태그 1의 ID 인식을 계속 수행한다.

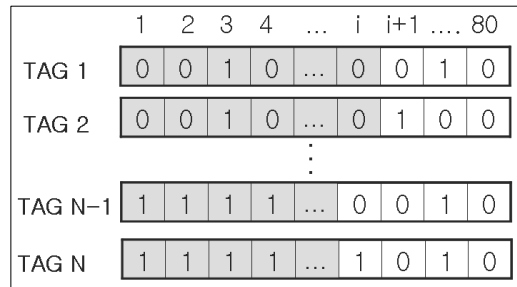


그림 4. 제안된 충돌회피 알고리즘

이때 태그 2는 추가된 플래그를 1로 변경시켜 다음 태그 인식 시 다른 태그와 구분지어 지게 된다. 태그1의 인식이 완료되면 리더는 태그 1의 ID를 저장하고 두 번째 태그(TAG2)를 인식하기 위해 첫 ID 비트부터 인식하는 기존 알고리즘과 달리 리더는 태그1의 ID 인식 시에  $i+1$  번째부터 인식하고,  $i$  번째까지 ID는 저장된 값을 통해 태그 ID를 인식한다. 이로써 리더는 태그의  $i$  번째 ID까지 인식하는 과정을 반복하지 않고 태그의 ID를 인식하여 인식 시간을 줄인다.

### 4.2 EPC Class 1

기존의 충돌회피 알고리즘은 VALUE의 길이가 길어져 충돌이 많이 일어났을 경우 리더가 전송해야 할 비트의 수가 늘어나게 된다. 따라서 본 논문에서는 기존 방식이 VALUE + Bin 값을 이용하여 PingID 명령어를 보내는 것에 반하여, VALUE 값을 뺀 Bin 값만을 보내어 데이터의 전송 시간을 줄임과 동시에 태그의 인식 시간을 줄이는 방법을 제안한다.

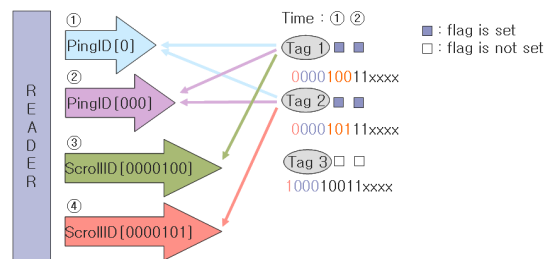


그림 5. 제안하는 충돌회피 알고리즘

우선 그림 5에서와 같이 태그에 하나의 flag를 추가하여 이전의 PingID 명령에 응답을 했을 경우 flag를 설정하여 이전 ID가 일치하지 않는 태그의 응답을 방지한다. 알고리즘은 ①의 PingID 명령으로 시작된다. 태그의 첫 비트가 0인 것들만 리더에게 응답을 할 것이다. 이때, 응답을 하는 경우, 태그의 flag를 set하게 된다. 태그의 그 다음 3비트를 구분하기 위해 ②의 PingID 명령을 보내게 되는데, 세 가지 태그 모두의

다음 3비트가 '000'으로 같지만, 3번 태그는 이전 명령에 응답하지 않았으므로 flag가 set되지 않았고, 이번 명령에도 응답하지 않는다. 따라서 태그 1과 2만 응답을 하게 된다. PingID 명령으로 구분이 끝난 태그는 ScrollID 명령을 통해 최종적으로 인식이 된다. flag의 상태는 리더에게 ScrollID 명령을 받게 되면 모두 reset되어 다음 PingID 명령에 응답할 수 있도록 한다. ScrollID 명령은 flag 상태와는 상관없이 동작한다.

제안된 알고리즘을 사용하면 리더가 PingID 명령어를 보낼 때 VALUE의 값이 3비트 이내로 고정되어 기존의 방식에 비해 리더가 보내야 할 데이터의 양이 줄게 되어 태그의 인식 시간이 줄게 된다.

### V. Simulation

본 장에서는 제안된 알고리즘의 simulation 결과를 제시하고 기존의 EPC Protocol 알고리즘과 성능을 비교한다.

#### 5.1 EPC Class 0

그림 6은 기존의 알고리즘과 제안된 알고리즘의 태그 수에 따른 리더에서 태그로 전송된 비트수를 제시한다. 실험 결과, 제안된 알고리즘은 기존의 EPC Class0 알고리즘에 비해 태그 수가 500개 일 경우, 리더에서 태그로 보내는 비트 수를 10% 줄여 성능 향상을 보였고, 태그 수가 2000개 일 경우에는 리더에서 태그로 보내는 비트수가 20%이상 줄어들었다.

#### 5.2 EPC Class 1

그림 7은 PingID 명령어의 태그의 수에 따른 총 전송 데이터양을 기존 방식과 본 논문에서 제안하는 방식으로 시뮬레이션한 결과이다. 태그의 수가 적을 때는 기존 방식과 제안한 방식의 성능 차가 크지 않으나 태그의 수가 많아질수록 기존의 방식보다 제안한 방식의 전송 데이터양이 현저히 줄어들음을 알 수 있다.

### VI. 결론

본 논문에서는 EPC Protocol에 부합하는 새로운 충돌방지 알고리즘을 제안하였다.

EPC Class 0의 경우, 제한된 알고리즘은 기존의 알고리즘에 태그 상태와 플래그를 추가시켜 리더에서 태그로 보내는 비트를 줄이고 태그의 인식속도를 단축시켜 성능을 향상시킨다.

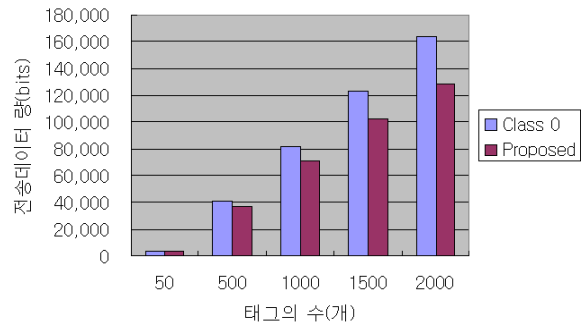


그림 6. 태그 수에 따른 전송 비트 수(Class0)

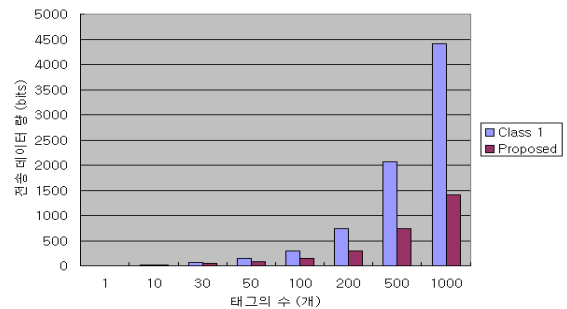


그림 7. 태그 수에 따른 전송 비트 수(Class1)

EPC Class 1은 태그의 모든 비트를 읽지 않고도 충돌회피가 가능하다. 그러나 태그의 수가 많을수록 태그의 인식 시간이 급격히 증가하는 단점이 있다. 제안된 알고리즘은 PingID 명령에서 VALUE의 비트 수를 줄임으로써 성능을 개선하였다.

제안된 알고리즘은 기존의 EPC 알고리즘에 비해 태그의 수가 적을 경우 성능 개선이 미약하나 태그의 수가 많아질수록 성능 개선이 커짐을 확인할 수 있다.

제안한 알고리즘을 적용한 시뮬레이션 결과를 토대로 최적의 성능을 갖는 RFID Reader 개발 비용을 감소시킬 수 있을 것이다.

### 참고문헌

- [1] Klaus Finkenzeller, "RFID handbook-Second Edition", Jone Wiley & Sons, 2003.
- [2] "Draft protocol specification for a 900MHz Class0 Radio Frequency Identification Tag", Auto-ID Center, 2003.
- [3] "860MHz-960MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical Communication Interface Specification Recommended Standard-Version 1.0.0", Auto-ID Center, 2002.