

ISO/IEC 18000-3 ASK 모드와 PJM 모드의 성능 비교

Performance Comparison of ISO / IEC 18000-3 ASK and PJM Mode

김재림* 양훈기**
(Jae-Lim Kim) (Hoon-Gee Yang)

요약

본 논문에서는 18000-3 RFID 표준인 ASK 모드와 PJM 모드 RFID 시스템을 인식속도의 관점에서 비교하였다. ASK 모드와 PJM 모드 리더/태그간 통신 과정을 표현 할 수 있는 대표 블록들의 조합을 구성하여 블록이 발생한 횟수를 이용하여 인식속도를 비교하였다. FS-Aloha 기반 프로토콜을 적용시키기 위한 Q알고리즘으로 표준 기고서에서 예시로 제안한 알고리즘, Q값 고정 알고리즘, 최소 Q 알고리즘 그리고 남은 태그 수에 연동한 Q값 설정 알고리즘을 이용하였다. 시뮬레이션 과정을 통해서 Q알고리즘에 따른 ASK 모드와 PJM 모드 속도 비교뿐만 아니라 Q알고리즘을 고정 시킨 후 ASK 모드와 PJM 모드의 인식속도를 비교하였고 속도에 영향을 미치는 파라메타들을 도출 하였다.

Abstract

This paper presents comparison of 18000-3 ASK and PJM mode RFID in terms of reading speed. This is accomplished by counting the occurrences of a few representative blocks which constitute the reader / tag interactions for each mode. To apply FS (frame slotted)-aloha protocol, four Q-algorithms, which are a Q-algorithm suggested in the standard document, a fixed Q-algorithm, a minimum Q-algorithm and the number of tag dependent Q-algorithm, are used. Through some simulations, we present the reading speed of each mode with respect to the Q-algorithm and the comparison of the reading speed between the two modes for each Q-algorithms.

Key words: RFID, ASK, PJM, reader, tag, frame slotted aloha

* 주저자 : 광운대학교 전파공학과 석사과정

** 공저자 : 광운대학교 전파공학과 교수

† 논문접수일 : 2008년 9월 11일

‡ 논문심사일 : 2008년 11월 28일

† 게재확정일 : 2008년 12월 3일

I. 서 론

RFID (radio frequency identification) 시스템은 무선 주파수를 이용하여, 마이크로 칩을 내장한 태그(tag)의 저장된 데이터를 리더를 통해 자동으로 인식, 추적, 식별 하는 기술을 말한다. 이때에 다수의 태그의 인식 효율 및 인식 속도의 향상을 위해서는 리더(reader) 및 태그 충돌(collision)을 회피하는 기술이 요구된다 [1]. RFID 시스템의 효율을 좌우하는 충돌 문제는 크게 태그 간 충돌과 리더 간 충돌이 있다. 태그 간 충돌이란 다수의 태그가 리더의 명령에 동시에 응답함으로써 태그가 전송한 데이터 사이에 충돌이 발생하는 것을 말하며, 리더 간 충돌에는 첫째로 태그 인식에 있어 리더들이 서로 간의 주파수를 간섭하여 발생하는 충돌과 둘째로 태그가 동시에 두 개 이상의 리더의 인식범위 내에 있어 간섭이 발생하여 생기는 충돌이 있다 [2, 3]. 리더간 충돌을 회피하기 위한 알고리즘은 다양하게 연구되고 있으며, 대표적으로 DCS (distributed color selection)와 VDCS (variable-maximum DCS, colorwave)가 있다 [4-6]. 태그 충돌을 회피하기 위한 방법으로는 알로하 (aloha)기반의 FS-Aloha (frame slotted aloha) 알고리즘과, 이진 (binary tree) 방식 기반의 슬롯 이진 트리 알고리즘 및 bit-by-bit 이진 트리 알고리즘이 있다 [2, 7].

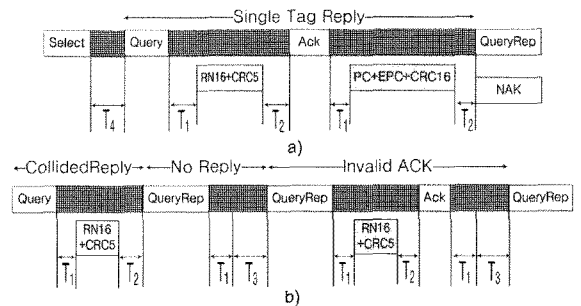
현재 짧은 인식거리 응용으로 HF (high frequency) 대역 13.56 MHz RFID 시스템은 다 방면으로 활용되고 있으며, 현재 ISO/IEC 18000-3 모드 1, 모드 2로 표준화가 완료 되었으며, 유통물류 분야에서 활용이 가능 하도록 하기위한 모드 3은 표준화가 진행 중이다 [8]. HF 대역 13.56 MHz RFID 시스템에서는 태그 충돌을 줄이기 위해서 기본적으로 FSA 기반의 태그 충돌회피 알고리즘을 사용하며 특히 모드 2의 경우 인식속도 향상을 위해 다중 채널을 사용한다 [9].

본 논문에서는 현재 표준화가 진행 중에 있는 ISO/IEC 18000-3 모드 3의 ASK 모드와 PJM 모드에 대해서 여러 가지 Q알고리즘을 적용시켜 모드별 인식 속도차이를 비교하였다.

제 II장과 제 III장에서는 ASK 모드 및 PJM 모드의 통신 흐름과 프레임 구성을 설명하고, 제 IV장에서는 여러 가지 Q-알고리즘의 동작 방식에 대한 설명을 하고, 제 V장에서는 시뮬레이션 결과를 보여주고, 제 VI장에서 결론을 맺는다.

II. ASK 모드

ASK 모드에서 리더와 태그 간 통신 프로토콜에 근거한 동작과정은 아래의 <그림 1>과 같다.



<그림 1> ASK 모드 타이밍도, a)단일 태그의 응답 동작과정 b)충돌과 무응답 동작이 포함된 동작과정
<Fig. 1> ASK mode Timing diagram, a)Reply process b)Reply process with collision and no-reply

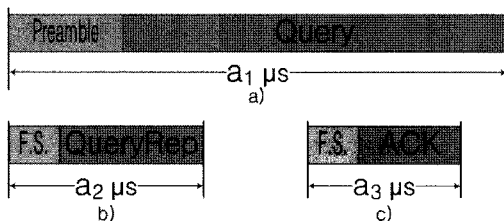
위의 <그림 1> a)는 RFID 필드에 하나의 태그가 있는 경우로서 리더는 Select 명령을 방사하여 인식 범위 내의 태그들을 인식 준비 상태로 만들게 되고, T4초 이내에 'Q' 인자가 포함된 Query 명령어를 방사하며 태그는 '0'에서 '2Q-1'의 범위를 가지는 슬롯(slot) 카운터 값을 설정하게 된다. 태그는 자신의 슬롯 카운터 값이 '0'이 되었을 때 T1초 이내에 리더에게 RN16값을 백스캐터링 (backscattering)하여 응답한 후, T2초 안에 리더로부터 ACK 명령어를 받은 태그는 다시 T1초 이내에 PC, EPC, CRC-16을 리더에게 백스캐터링한다. 리더는 태그로부터 받은 신호가 정확하다면 다른 태그의 슬롯 카운터 값을 줄이기 위해 T2초 이내에 QueryRep 명령어를 방사하게 되지만, 정확하지 않다면 역시 T2초 이내에 NAK 명령어를 전송함으로써 태그를 재인식 하게

된다. 다른 상황으로 <그림 1>의 b)에서는 QueryRep 명령어 전송 후, T1초 이내에 리더에게 응답하는 태그가 없으면 리더는 무응답 상황이라 인지하고 T3초가 지나기 전에 QueryRep 명령어를 이용하여 태그의 슬롯 카운터 값을 1씩 빼도록 하여 태그의 응답을 기다린다. 이후 슬롯카운터 값이 '0'이 되는 태그가 발생하면 그림 a)에서 설명한 것과 동일하게 태그는 RN16 값을 백스캐터링한다.

2.1절 및 2.2 절은 다중 태그 환경에서 ASK모드의 인식 속도 계산을 위해 리더-태그간의 통신 과정을 몇 개씩의 블록으로 구분한 것으로 <그림 1>의 타이밍(timing)도는 아래 블록들의 조합으로 구성됐다고 할 수 있다.

1. Reader-to-Tag(R=>T) 통신

R=>T 통신에서 리더가 사용하는 명령어들은 NAK를 제외하면 아래의 <그림 2>에서 보는 것과 같이 3가지로 구분 할 수 있다. 각각의 명령어들은 동기문제 해결과 여러 가지 인자들의 설정을 위하여 명령어 앞쪽에 헤더를 붙여 전송한다. 먼저 a)의 Query명령어 프레임은 Preamble이라는 헤더를 붙여 동기문제를 해결하고 Tari 길이 및 R=>T의 전송 속도를 설정한다. b)의 QueryRep명령어 프레임 및 c)의 ACK명령어 프레임 앞에는 FrameSync라는 헤더를 붙여 Preamble과 같은 기능을 하게 한다. 명령어 길이 a_1, a_2, a_3 값은 제 5장 시물레이션에서 제시한다.



<그림 2> R=>T의 명령어 프레임 a) Query 명령어 b) QueryRep 명령어 c) ACK 명령어
 <Fig. 2> R=>T command frame a) Query command b) QueryRep command c) ACK command

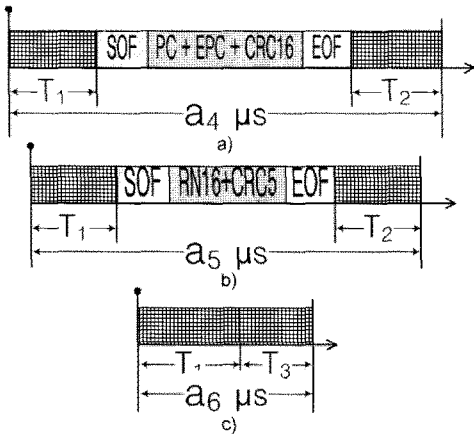
2. Tag-to-Reader(T=>R) 통신

T=>R 통신에서는 FM0, Manchester, Miller sub-carrier 등 세 가지의 부호화 방식이 있으며, 리더는 Query 명령어의 파라미터 값을 조정하여 부호화 방식 및 통신 속도를 태그에 알린다. <그림 3>은 T=>R 통신에서의 태그 응답 프레임을 나타낸 것이다. 먼저 a)는 응답 성공 프레임으로써, 태그가 리더로부터 ACK명령어를 받고 EPC코드를 백스캐터링하는 프레임이다. b)는 응답 또는 충돌 프레임으로써, 리더의 Query명령어 혹은 QueryRep명령어를 받은 태그 중 슬롯 카운터 값이 '0'된 태그가 RN16값을 리더에게 백스캐터링 하는 프레임이다. c)는 리더가 Query명령어 혹은 QueryRep명령어를 방사하는 상황에서 슬롯 카운터 값이 '0'인 태그가 하나도 없을 때, 리더가 다음 명령어인 QueryRep을 방사하기 전까지의 무응답 프레임이다.

<그림 3>의 a)와 b)에서 SOF와 EOF가 프레임 구성에 포함 되었는데, 먼저 SOF는 Start of Frame으로서, R=>T의 Preamble과 같이 명령어의 동기를 맞추기 위해 존재 하며, EOF는 End of Frame으로써, 태그 응답의 끝을 알려주는 신호이다. 각각의 프레임 시간 값 a_4, a_5, a_6 는 부호화 방식에 따라 가변적이며 제5장 시물레이션에서 구체적인 값을 제시한다.

III. PJM 모드

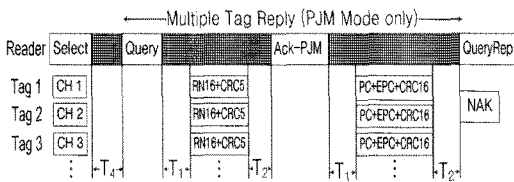
<그림 4>는 PJM 모드에서 리더와 태그 간 통신 프로토콜에 근거한 동작과정을 나타낸 것이다. 기본적인 인벤토리 라운드(inventory round) 과정에서는 ASK 모드와 크게 다른 점이 없으며, 리더 명령어들의 기능 및 태그의 응답 역시 동일하다. PJM 모드는 8개까지의 다중 채널을 이용하여 병렬적으로 8개의 태그를 동시에 인식 할 수 있다는 점이 ASK모드와의 가장 큰 차이점이다. 다시 말해 R=>T 통신에서는 리더가 하나의 주파수를 이용하여 명령어를 태그들에게 방사하고, T=>R 통신에서는 태그가 8개의 응답 주파수 중 하나를 선택하여 리더에 응답하거나 태그가 응답 주파수를 랜덤하게 선택하



<그림 3> T=>R 백스캐터링 프레임 a)성공 프레임 b)충돌 또는 응답 프레임 c)무응답 프레임

<Fig. 3> T=>R backscattering frame a)Success frame b)Collision or reply frame c)No-reply frame

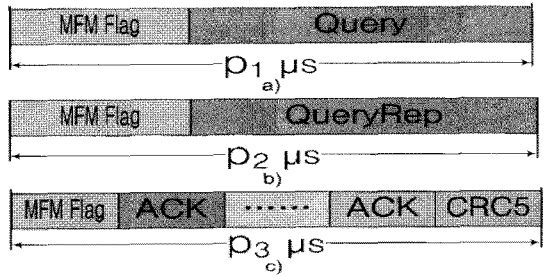
게 하여, 병렬적으로 동시에 응답하는 것이다. 또 다른 차이점은 R=>T 통신에서 리더 명령어와 태그 응답 프레임 앞에 붙어 있는 헤더프레임의 구조가 다르다는 것과, <그림 4>에서도 보듯이 태그들이 각각의 채널에서 동시에 리더의 명령어를 수행함에 따라 리더의 ACK 명령어의 프레임 구성이 ASK 모드와 다르게 구성된다는 것이다. 3.1절 및 3.2절에서는 PJM 모드의 리더·태그 통신과정을 구성하는 블록들을 추출하였으며 <그림 4>의 통신과정은 이들 블록들의 조합에 의해 구성된다고 할 수 있다.



<그림 4> PJM 모드 타이밍도
<Fig. 4> PJM-mode timing diagram

1. Reader-to-Tag(R=>T) 통신

<그림 5>는 R=>T 통신에서 리더 명령의 프레임



<그림 5> R=>T 명령어 프레임 a) Query 명령어 b) QueryRep 명령어 c) PJM-ACK 명령어

<Fig. 5> R=>T command frame a) Query command b) QueryRep command c) PJM-ACK command

구조를 나타낸다.

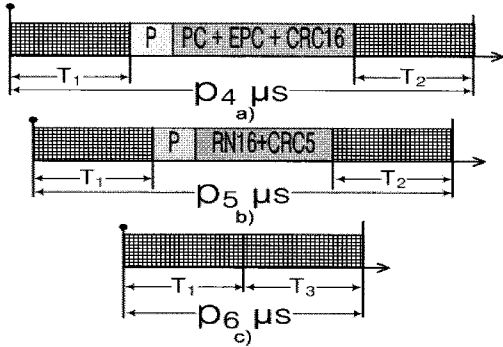
PJM모드에서 R=>T 명령어 프레임 구성 역시 ASK모드와 비슷하지만 각각의 명령어 앞의 헤더프레임이 ASK모드와 다르다. ASK모드의 경우 Preamble 과 Frame-Sync를 사용했으나, PJM의 경우 하나의 헤더프레임, MFM Flag만을 사용하여 명령어의 시작 및 비트 간격 타이밍을 정의한다.

<그림 5>의 a)와 b)는 Query명령어 프레임과 QueryRep명령어 프레임을 나타내는데 그 기능은 ASK 모드와 동일하다. c)는 PJM-ACK 명령어 프레임으로써, ASK 모드와 다르게 여러 개의 ACK가 가능하다. 이는 태그들이 각기 다른 채널을 통해서 RN16값을 보낼 수 있으므로 두개 이상의 태그 응답을 수신한 리더는 하나의 주파수만을 사용하여 명령어를 방사하므로 여러 개의 ACK를 직렬, 즉 시퀀셜(sequential)하게 보낸다.

2. Tag-to-Reader(T=>R) 통신

아래의 <그림 6>은 PJM모드의 T=>R 백스캐터링 신호들의 프레임 구성을 나타낸다. ASK 모드와의 가장 큰 차이점은 ASK 모드에서는 SOF, EOF의 헤더, 테일 프레임이 존재 하였는데, PJM모드에서는 a), b) 프레임 구성에서 보듯이 R=>T 통신에서와 마찬가지로 MFM Flag라는 헤더 프레임을 가진다. R=>T 통신의 MFM Flag와는 Flag구성 및 기능은 동일하지만 비트 간격이 9.44μs로 두 배이다. a), b), c)

프레임의 기능은 ASK 모드와 동일하다. <그림 5>와 <그림 6>의 프레임 길이 p1~p6값은 제 5장 시뮬레이션에서 제시하겠다.



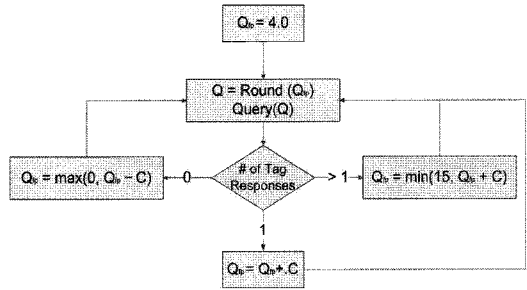
<그림 6> T=>R 백스캐터링 프레임 구성 a) 성공 프레임 b) 충돌 및 응답프레임 c) 무응답 프레임
<Fig. 6> T=>R backscattering frame a) Success frame b) Collision and reply frame c) No-reply frame

IV. Q-알고리즘

FSA를 사용하는 경우 슬롯카운터 값을 정하는 방법은 태그 인식 속도에 영향을 준다. 본 절에서는 다양한 Q 알고리즘을 적용시켜 ASK 및 PJM 모드의 인식속도를 비교하고자 한다.

1. EPC Global에서 제시한 Q알고리즘 (Q-A)

아래 <그림 7>은 EPC Global에서 예시로 제시한 Q알고리즘으로 리더가 QueryRep을 보냈을 때 태그로부터의 응답 상황에 따라 Q값을 업데이트 하는 알고리즘이다. 즉 QueryRep후 응답이 없으면 프레임내 슬롯의 개수가 많다고 가정하고 Q값을 C만큼 줄이고 충돌이 발생하면 Q값을 C만큼 늘리게 되고 하나의 태그가 응답하면 Q값을 그대로 두는 것이다. 이 과정을 거쳐 업데이트된 Q값은 다음 Query 명령에서 인자로 사용하게 된다. 여기에 사용하는 C의 값 범위는 $0.1 < C < 0.5$ 범위를 가지며 Q값이 클 경우 C의 값은 작은 값이 사용되고 Q의 값이 작을 경우에는 C값은 큰 값이 사용된다.



<그림 7> EPC Global에서 제시한 Q알고리즘
<Fig. 7> Q Algorithm of EPC Global

2. 최소 Q 알고리즘 (Q-B)

EPC Global에서 제시한 Q알고리즘으로 시뮬레이션 하였을 때 충돌이 많이 일어나서 Q값이 커졌다가 무응답이 많이 생겨 0까지 떨어지는 현상이 발생하는 경우 다시 충돌이 많아지므로 최소 Q값을 정하여 그 이하로 떨어짐을 방지 하여 좀 더 효율적으로 슬롯 낭비를 줄일 수 있다.

3. 고정 Q 알고리즘 (Q-C)

처음에 정의된 Q의 값을 모든 인벤토리가 끝날 때까지 사용하는 방법이다. 이 알고리즘은 인벤토리가 진행함에 따라 태그수가 줄어들게 되나 이에 적응적으로 Q값이 감소되지 않으므로 성능은 점점 나빠지게 된다.

4. 남은 태그 수에 연동하는 Q 알고리즘 (Q-D)

Slotted Aloha방식에서 최대 throughput은 슬롯에 한 개의 입력이 오는 경우로서 0.37정도 값을 갖는다. 그러므로 남은 태그 수를 추정할 수 있다면 남은 태그 수가 2^Q 와 동일한 Q값을 설정함으로써 평균적으로 한 슬롯당 한 개의 태그응답이 계속 오는 효과가 있어 최대 효율을 얻을 수 있다[10].

V. 시뮬레이션

1. 시뮬레이션 인자 설정

<표 1> 링크 타이밍 인자
<Table 1> Link timing parameters

Parameter	T1	T2	T3	T4
Value	73.1 μ s	151 μ s	26.9 μ s	100 μ s

<표 1>의 T1은 리더의 전송과 태그의 응답 사이에 발생하는 시간이고, T2는 리더가 백스캐터링 된 태그의 신호에 응답하기 위해 복조하는데 필요한 시간이다. T3는 T1시간 후 QueryRep 명령어를 전송하기 전에 리더가 기다리는 시간이며, 마지막으로 T4는 T1+T3로서 리더의 명령어 사이에 필요한 최소 시간이다.

ASK 모드의 경우 1 Tari값은 8 μ s ~ 25 μ s 사이의 값을 갖도록 표준에 규정되어 있으므로 본 시뮬레이션에서는 10 μ s로 가정 하였고 태그가 리더에 전송 할 때는 FM0 및 Miller 인코딩을 하여 전송 하는 경우로 가정 하여 각각의 경우 비트율을 424kbit/s, 53kbit/s로 하였다. Pulse Interval Encoding에서

<표 2> ASK 모드에서의 프레임 시간
<Table 2> Frame time in ASK mode

	Encoding	
	FM0 (μ s)	Miller (μ s)
Preamble	139.44	139.44
Frame-Sync	49.44	49.44
SOF	23.6	188.7
EOF	4.72	37.74
그림 2의 a ₁	469.44	469.44
그림 2의 a ₂	109.44	109.44
그림 2의 a ₃	319.44	319.44
그림 3의 a ₄	478.98	2261.1
그림 3의 a ₅	301.98	846.6
그림 3의 a ₆	100	100

Data-0 = 1×Tari, Data-1 = 2×Tari이지만 비트수에 따라서 모두 1.5Tari라 가정하였다. 그리고 한 슬롯의 정의는 리더가 태그에게 명령어를 보냈을 때를 한 슬롯으로 정의를 하고 슬롯의 수를 계산 하였다.

<표 2>는 ASK모드의 Preamble, Frame-Sync, EOF, EOF, <그림 2>의 a₁, a₂, a₃와 <그림 3>의 a₄, a₅, a₆값을 나타낸 표이다.

우선 R=>T에게 보내는 명령어의 타이밍을 정리 하면 명령어를 인식하는 Preamble와 Frame-Sync시간을 구하는 방법은 아래와 같다.

$$\text{Preamble} = \text{Delimiter} + \text{Data-0} + \text{RTcal} + \text{TRcal}$$

$$\text{Frame-Sync} = \text{Delimiter} + \text{Data-0} + \text{RTcal}$$

<표 2>의 Preamble와 Frame-Sync값은 1 Tari를 10 μ s로 가정 하고 계산 하였을 때 FM0와 Miller 의 두 경우 모두 139.44 μ s, 49.44 μ s가 나왔다.

<그림 2>의 a₁(Query), a₂(QueryRep), a₃(Ack) 명령어 프레임 시간은 아래와 같이 계산된다.

$$\text{Query}(a_1) = \text{Preamble} + 22 \times (1.5) \text{Tari}$$

$$\text{QueryRep}(a_2) = \text{Frame-Sync} + 4 \times (1.5) \text{Tari}$$

$$\text{Ack}(a_3) = \text{Frame-Sync} + 18 \times (1.5) \text{Tari}$$

FM0는 1bit를 전송 하는데 2.36 μ s, Miller는 18.87 μ s가 소요된다고 가정 하고 계산하였다.

T=>R로 명령어를 보낼 때 태그 명령어를 인식하는 SOF(Start of frame)와 EOF(End of frame)가 붙게 되며 SOF와 EOF는 SOF= 10bit, EOF= 2bit 값을 갖는다.

<그림 3>의 a₄, a₅, a₆을 구하는 방법은 다음 과 같다.

$$a_4 = T_1 + \text{SOF} + (\text{PC} + \text{EPC} + \text{CRC5}) + \text{EOF} + T_2$$

$$a_5 = T_1 + \text{SOF} + (\text{RN16} + \text{CRC5}) + \text{EOF} + T_2$$

$$a_6 = T_1 + T_3$$

다음 PJM 모드의 경우 EPC 표준에 근거하여 리

더가 태그로 전송 할 때는 212kbit/s, 태그에서 리더로 전송 될 때는 106kbit/s로 권고하였다. 따라서 시간을 계산 할 때 R=>T로 명령어를 전송 할 때는 4.72μs, T=>R로 명령어 전송 할 때는 9.44μs의 시간으로 계산 하였다. 아래의 <표 3>은 <그림 5와 6>에 나타난 명령어 프레임의 시간을 계산한 값이다.

<표 3> PJM 모드 프레임 시간
<Table 3> Frame timing of PJM

R=>T MFM Flag	75.52μs
T=>R MFM Flag	151.04μs
그림 5의 P ₁	176.36μs
그림 5의 P ₂	94.40μs
그림 5의 P ₃	표 4 참고
그림 6의 P ₄	1281.38μs
그림 6의 P ₅	573.38μs
그림 6의 P ₆	100μs

PJM모드에서 시그널 인식을 위한 R=>T의 MFM Flag, T=>R MFM Flag 의 타이밍은 다음과 같다.

$$R \Rightarrow T \text{ MFM Flag} = 16\text{bit} \times 4.72\mu\text{s} = 75.52\mu\text{s}$$

$$T \Rightarrow R \text{ MFM Flag} = 16\text{bit} \times 9.44\mu\text{s} = 151.04\mu\text{s}$$

R=>T의 명령어 프레임에서의 <그림 5>의 P1 (Query), P2(QueryRep), P3(Ack-PJM) (그림 5 참조)이 있는데 이에 대한 값은 다음과 같이 계산된다.

$$\begin{aligned} \text{Query} &= \text{MFMFlag} + 22\text{bit} = 179.36\mu\text{s} \\ \text{QueryRep} &= \text{MFMFlag} + 4\text{bit} = 94.40\mu\text{s} \\ \text{Ack-PJM} &= \text{MFMFlag} + 18\text{bit} \times (\text{응답 채널 수}) + \text{CRC}-5 \\ &= 75.52\mu\text{s} + 84.96\mu\text{s} \times (\text{응답 채널 수}) + 23.60\mu\text{s} \end{aligned}$$

<표 4>는 응답한 채널수에 따른 Ack-PJM프레임 시간 길이다.

<표 4> PJM-ACK의 시간 간격
<Table 4> Duration of PJM-ACK

응답한 채널 개수	Ack-PJM 길이(C)
1	184.08μs
2	269.04μs
3	354μs
4	438.96μs
5	523.92μs
6	608.88μs
7	693.84μs
8	778.8μs

다음은 <그림 6>의 P₄, P₅, P₆ 프레임 시간 계산과 정의이다.

$$P_4 = T_1 + \text{MFMFlag} + (\text{RN16} + \text{CRC5}) + T_2 = 573.38\mu\text{s}$$

$$\begin{aligned} P_5 &= T_1 + \text{MFMFlag} + (\text{PC} + \text{EPC} + \text{CRC5}) + T_2 \\ &= 1281.38\mu\text{s} \end{aligned}$$

$$P_6 = T_1 + T_3 = 100\mu\text{s}$$

2. 시뮬레이션 결과

5.1에서 제시한 프레임 시간에 근거해서 IV절에서 제시한 Q알고리즘을 적용시키면서 ASK 모드와 PJM모드의 인식 속도 및 효율관점에서 비교하였다. 태그의 개수를 2⁸ = 256개로 하고 초기 Q값을 태그 개수에 최적인 8로 정하였다. PJM 모드는 8개의 채널을 사용할 수 있다고 가정해서 태그가 랜덤하게 채널을 선택 하는 경우 채널당 평균 256/8 = 32개의 태그가 있게 되므로 2⁵ = 32에 의해 Q값을 8의에도 5인 경우에 대해서도 시뮬레이션 하였다.

아래 시뮬레이션 결과 분석 과정에서 슬롯 효율 ρ는 다음과 같이 정의 하였다.

$$\rho = \frac{\text{EPCReply}}{\text{총슬롯수}}$$

<표 5> Q-A 시뮬레이션 결과 값
<Table 5> Simulation results by Q-A

동작	슬롯	시간(ms)
Select	19	1.80
Collision	179	151.54
No Reply	1155	115.50
RN16 Reply	256	192.63
EPC Reply	256	579.09
Query	19	8.92
QueryRep	1590	174.36
Ack	256	81.78
합계	1865	1303.82
슬롯 효율	13.7%	

<표 6> Q-B 시뮬레이션 결과 값 (최소 Qmin=2)
<Table 6> Simulation results by Q-B (Qmin is 2)

동작	슬롯	시간(ms)
Select	21	0.20
Collision	212	179.48
No Reply	448	44.80
RN16 Reply	256	192.63
EPC Reply	256	579.09
Query	21	9.86
QueryRep	895	98.15
Ack	256	81.78
합계	1172	1185.79
슬롯 효율	21.8%	

<표 7> Q-C 시뮬레이션 결과 값 (Q=8)
<Table 7> Simulation results by Q-C (Q is 8)

동작	슬롯	시간(ms)
Select	5	0.40
Collision	110	82.77
No Reply	914	91.40
RN16 Reply	256	192.63
EPC Reply	256	579.09
Query	5	2.34
QueryRep	1275	139.54
Ack	256	81.78
합계	1536	1169.95
슬롯 효율	17%	

<표 8> Q-D 알고리즘 시뮬레이션 결과
<Table 8> simulation results by Q-D

동작	슬롯	시간(ms)
Select	9	0.80
Collision	152	114.37
No Reply	236	23.60
RN16 Reply	256	192.63
EPC Reply	256	579.09
Query	9	4.22
QueryRep	635	69.50
Ack	256	81.78
합계	900	1065.00
슬롯 효율	28.4%	

<표 5-8>은 ASK 모드 시뮬레이션 결과다. <표 5>는 EPC Global에서 제시한 Q알고리즘, <표 6>은 최소 Q 알고리즘(최소값 Qmin = 2), <표 7>은 고정 Q 알고리즘 그리고 <표 8>은 남은 태그 수에 연동하는 Q알고리즘이다.

<표 5 ~ 8>의 슬롯 효율 및 전체 읽는데 걸린 시간을 비교 해 보면 ASK의 경우 Q-D 알고리즘은 Q-A, Q-B, Q-C 알고리즘 보다 슬롯 효율 면에서는 14.7%, 6.6%, 11.4% 좋아 졌고 그리고 인식 속도 면에서는 1.22배, 1.11배, 1.10배 빨리 인식했음을 알 수 있다.

<표 9> Q-A 시뮬레이션 결과 값
<Table 9> Simulation results by Q-A

동작	슬롯	시간(ms)
Select	11	1.00
Collision	9	5.16
No Reply	89	8.90
RN16 Reply	168	96.32
EPC Reply	168	215.27
Query	11	1.94
QueryRep	255	24.07
Ack	168	36.79
합계	434	388.45
슬롯 효율	38.7%	

<표 10> Q-B 시뮬레이션 결과 값 (최소 Qmin=2)
<Table 10> Simulation results by Q-B (Qmin is 2)

동작	슬롯	시간(ms)
Select	3	0.20
Collision	6	3.44
No Reply	93	9.30
RN16 Reply	165	94.60
EPC Reply	165	211.43
Query	3	0.53
QueryRep	264	24.64
Ack	165	36.43
합계	432	380.37
슬롯 효율	38.2%	

<표 11> Q-C 시뮬레이션 결과 값 (Q=5)
<Table 11> Simulation results by Q-C (Q is 5)

동작	슬롯	시간(ms)
Select	4	0.30
Collision	4	2.30
No Reply	18	1.80
RN16 Reply	106	60.78
EPC Reply	106	135.83
Query	4	0.72
QueryRep	128	11.70
Ack	106	30.84
합계	238	243.96
슬롯 효율	44.5%	

<표 12> Q-C 알고리즘 시뮬레이션 결과
<Table 12> Simulation results by Q-C (Q is 8)

동작	슬롯	시간(ms)
Select	2	0.10
Collision	7	4.01
No Reply	328	32.80
RN16 Reply	177	101.49
EPC Reply	177	226.80
Query	2	0.36
QueryRep	512	48.14
Ack	177	38.45
합계	691	452.16
슬롯 효율	25.6%	

<표 13> Q-D 알고리즘 시뮬레이션 결과
<Table 13> Simulation results by Q-D

동작	슬롯	시간(ms)
Select	6	0.50
Collision	3	1.72
No Reply	0	0.00
RN16 Reply	81	46.44
EPC Reply	81	103.80
Query	6	1.08
QueryRep	84	7.36
Ack	81	27.67
합계	171	188.57
슬롯 효율	49.1%	

<표 9~13>은 PJM 모드를 이용한 시뮬레이션 이다. <표 9>는 EPC Global에서 제시한 Q알고리즘, 표 10은 최소 Q알고리즘(최소값 Qmin = 2), <표 11, 12>는 고정 Q 알고리즘을 이용 하였다. 그리고 <표 13>의 경우는 남은 태그 수에 연동하는 Q알고리즘 이다. <표 10>의 경우 최소 Q값을 2로 가정하였고, <표 11>의 경우는 Q의 값을 5로, <표 12>는 Q의 값을 8로 고정 하였다.

<표 9~13>의 슬롯 효율 및 전체 읽는데 걸린 시간을 비교 해 보면 PJM의 경우 Q-D 알고리즘은 Q-A, Q-B, Q-C 알고리즘 보다 슬롯 효율 면에서는 8.7%, 9.2%, 2.9%, 21.8% 좋아 졌고 그리고 인식 속도 면에서는 2.05배, 2.02배, 1.29배, 2.40배 빨리 인식했음을 알 수 있다.

<표 14> ASK 와 PJM의 시뮬레이션 조건
<Table 14> Simulation parameters of ASK and PJM

	ASK	ASK	PJM 4 Channel	PJM 8 Channel
1 Tari	10 μ s	10 μ s	4.72 μ s	4.72 μ s
R=>T Data rate	100kbit/s		212kbit/s	
T=>R Data rate	424kbit/s	53kbit/s	106kbit/s	
R=>T Data encoding	PIE Encoding		MFM Encoding	
T=>R Data encoding	FM0	Miller	MFM Encoding	

위 <표 14>는 ASK 모드와 PJM 모드를 비교하기 위한 시뮬레이션 조건을 표로 나타낸 것이다. 태그 수를 200개부터 500개까지 20개씩 증가 하였고 충돌 방지 알고리즘은 Q-D 알고리즘을 이용하였다. 리더에서 태그로 Data rate는 ASK의 경우 100kbit/s, PJM의 경우는 212kbit/s로 하였다. T=>R로 백스캐터링 할 때 ASK의 경우 FM0과 Miller 인코딩 하였고 PJM의 경우는 MFM 인코딩하고 채널수는 4개와 8개로 구분 하여 시뮬레이션 하였다. 시뮬레이션 결과를 보면 아래 <그림 8>과 같이 ASK 모드에서 Miller와 FM0 두 개를 비교 하였을 때 FM0이 Miller 방식 보다 2.6배 정도 인식 속도가 빠름을 알 수 있고 그리고 PJM의 경우 8개의 채널을 이용하는 것이 4개의 채널을 이용하는 것 보다 1.7배 정도 빠름을 알 수 있다. 그리고 8채널을 이용한 PJM 방식이 ASK의 FM0 인코딩을 이용한 시스템 보다 2.3배 빨리 태그를 인식하였다.

시뮬레이션 결과 PJM 모드와 Miller 사용한 ASK 모드를 비교해 보았을 때 EPC Global에서 제시한 Q 알고리즘을 사용하였을 경우 25% 슬롯 효율이 좋아졌고 3.36배 정도의 빠른 인식 속도 갖게 되었다. 최소 Q 알고리즘을 이용 했을 때는 16.4%의 슬롯 효율이 좋아졌고, 태그 인식 속도는 3.12배 빨랐다. Q값을 고정 하였을 경우에는 슬롯 효율은 8.6% 좋아졌고, 태그 인식 속도는 2.59배의 인식 속도가

빨랐다. 그리고 남은 태그 수에 연동한 Q 알고리즘에서는 PJM 모드가 ASK 모드 보다 슬롯 효율이 20.7% 좋아졌고 인식 속도는 5.65배의 인식 속도를 얻었다.

VI. 결 론

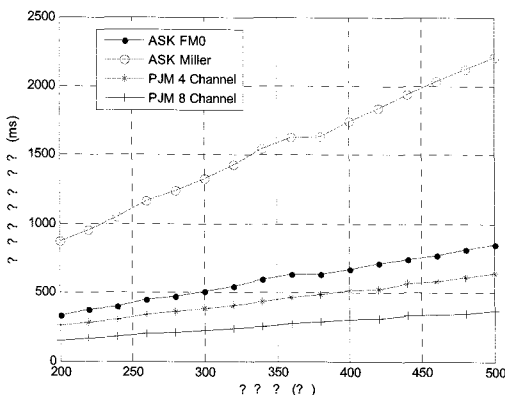
본 논문에서는 18000-3 RFID 표준인 ASK 모드와 PJM 모드 RFID 시스템을 인식속도와 슬롯의 효율 관점에서 비교하였다. 인식속도 비교를 위하여 ASK 모드와 PJM 모드 프로토콜을 리더/태그간 통신 부분을 블록 별로 나누어 블록이 발생한 횟수를 이용하였다. FSA 기반 프로토콜을 적용시키기 위해서 Q 알고리즘으로 EPC Global에서 예시로 제안한 알고리즘, Q값 고정 알고리즘, 최소 Q 알고리즘 그리고 남은 수에 근거한 Q값 설정 알고리즘을 이용하였다.

시뮬레이션 결과 남은 태그 수에 근거한 Q 알고리즘을 이용한 PJM 모드와 Miller 인코딩을 사용한 ASK 모드의 슬롯 효율과 인식 속도를 비교해 보았을 때 PJM 모드가 ASK 모드 보다 각각 20.7% 슬롯 효율과 5.65배의 인식 속도를 얻을 수 있었다.

본 논문에서의 ASK 모드와 PJM 모드의 비교는 시뮬레이션에 의존해서 수행되었으며 추후 연구내용으로 이론적인 접근에 의해 두 방식의 인식속도를 비교하는 것이 필요하다.

참고문헌

- [1] S. Sarma, D. Brock, and D. Engels, "Radio frequency identification and electronic product code," *IEEE Microwave* / vol. 21 no. 6, pp. 50-54, Nov. 2001.
- [2] K. Finkenzeller, *RFID Handbook; Fundamentals and applications in Contactless Smart Cards and Identification*, 2nd Edition, John Wiley & Sons, pp. 195-219, 2003.
- [3] D. W. Engels, *The reader collision problem*, Technical report, epcglobal.org, 2002.



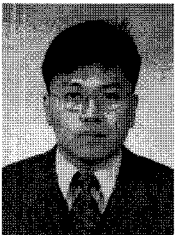
<그림 8> ASK와 PJM 비교 시뮬레이션 결과 값
<Fig. 8> Contrast Simulation results of ASK and PJM

- [4] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: a MAC for RFID reader networks," *Proc. IEEE Wireless Communications and Networking*, vol. 3, pp. 1701 - 1704, Mar. 2003.
- [5] S. Sarma, J. Waldrop, and D. Engels, "Colorwave : An anti-collision algorithm for the reader collision problem," *Proc. IEEE Int. Conf. Communications*, vol. 2, pp. 1206-1210, May 2003.
- [6] S. M. Birari, *Mitigating the reader collision problem in RFID networks with mobile readers*, Master's Thesis, Kanwai Rekhi School of Information Technology, 2005.
- [7] 차재룡, 김재현, "RFID 시스템에서의 태그 수를 추정하는 ALOHA 방식 Anti-collision 알고리즘," *한국통신학회논문지*, 제30권, 제9호, pp. 814~821, 2005. 9.
- [8] 강유성, *UHF 기술 및 HF 기술의 국제 표준화 논쟁*, 한국전자통신연구원 RFID/USN보안연구팀, IT Standard Weekly, 2008. 6.
- [9] EPCglobal, *EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 HF RFID Protocol for Communications at 13.56 MHz Version 1.0.1*.
- [10] J. E. Wieselthier, A. Ephremides, and L. A. Michaels, "An exact analysis and performance evaluation of framed ALOHA with capture," *IEEE Trans. Communications*, vol. 37, no. 2, pp. 125-137, Feb. 1989.

저자소개



김 재 림 (Kim, Jae-Lim)
 2007년 3월 ~ 현재 : 광운대학교 전파공학과 (석사)
 2007년 2월 : 배재대학교 정보통신공학과 (학사)



양 훈 기 (Yang, Hoon-Gee)
 1993년 3월 ~ 현재 : 광운대학교 전파공학과 교수
 1992 : SUNY at Buffalo ECE (석사)
 1987 : SUNY at Buffalo ECE (석사)
 1985 : 연세대학교 전자공학과 (학사)