

RFID시스템에서 슬롯의 혼잡도를 이용한 DFS-ALOHA 알고리즘

이 재 구[†] · 최 승 식^{††}

요 약

RFID 리더기가 영역내의 다수의 태그를 인식할 때 태그간의 통신간섭에 의해 태그인식이 방해 받는다. 이를 피하기 위해서는 충돌방지 알고리즘이 필요하게 된다. 충돌방지 알고리즘은 크게 슬롯 알로하 기반 알고리즘과 트리기반 알고리즘으로 나뉜다. 본 논문은 ISO 18000-6 TYPE A에 정의된 알로하 기반의 Framed Slotted ALOHA(FSA) 알고리즘에 태그와 슬롯간의 혼잡도를 이용하여 성능을 개선한 Dynamic Framed Slotted ALOHA-Slot Congestion(DFSA-SC) 알고리즘을 제안한다. 시뮬레이션 결과 최초 태그 수 추정의 정확도를 높여 전체 태그 인식 시간이 줄어든 것을 확인 할 수 있었다. 나아가 태그 아이디의 중복성이 클 경우 대표적인 트리기반 알고리즘인 Query Tree 알고리즘보다 제안된 알고리즘이 우수한 것을 확인 할 수 있었다.

키워드 : RFID, 충돌회피 알고리즘, FSA 알고리즘, DFSA 알고리즘

A DFS-ALOHA Algorithm with Slot Congestion Rates in a RFID System

Jae-ku Lee[†] · Seung-sik Choi^{††}

ABSTRACT

For the implementation of a RFID system, an anti-collision algorithm is required to identify multiple tags within the range of a RFID Reader. There are two methods of anti-collision algorithms for the identification of multiple tags, conclusive algorithms based on tree and stochastic algorithms based on slotted ALOHA. In this paper, we propose a Dynamic Framed Slotted ALOHA-Slot Congestion(DFSA-SC) Algorithm. The proposed algorithm improves the efficiency of collision resolution. The performance of the proposed DFSA-SC algorithm is showed by simulation. The identification time of the proposed algorithm is shorter than that of the existing DFSA algorithm. Furthermore, when the bit duplication of the tagID is higher, the proposed algorithm is more efficient than Query Tree algorithm.

Keywords : RFID, Anit-Collision, FSA Algorithm, DFSA Algorithm

1. 서 론

RFID 기술이란 관리할 모든 개별 사물에 전자태그를 부착하여 리더기로 무선 전파를 태그에 보내면 태그가 자신의 ID정보를 리더기로 전달하는 방식으로써 비접촉 형태로 태그 정보를 인식 할 수 있는 기술이다. 기존 바코드의 문제점을 극복 할 수 있을 뿐만 아니라 모든 개별사물의 인식이 가능하여 부착된 제품의 Traceability와 Visibility를 확보 할 수 있어 유통업의 핵심기술이라 할 수 있다. 하지만 기기들 사이의 이질성, 리더기 사이의 충돌, 태그 사이의 충돌등 아직 해결해야할 문제점이 있는 상황이다. 본 논문

은 태그 사이의 충돌 문제를 해결하기 위한 것으로 태그 사이의 충돌이란 태그들 간의 통신 과정에서 태그들 간에 간섭으로 태그 인식이 불가능한 경우를 말하며 RFID 시스템의 성능과 안정성을 위해 반드시 해결해야할 문제이다. 이렇듯 다중 태그 환경에서 태그간의 충돌 현상을 회피하는 기술을 다중 태그인식 기술이라 하며 여기에 사용되는 알고리즘을 anti-collision 알고리즘이라 한다.

RFID 표준화는 현재 국제표준기구인 ISO와 국제 전기기술위원회인 IEC의 합동기술 위원회인 ISO/IEC JTC1에서 진행하고 있다. SC31은 JTC1의 31번째 위원회로 자동식별 및 데이터수집기술에 대한 표준화를 진행하고 있다. SC31은 5개의 WG으로 구성되는데 WG4에서 RFID에 대한 표준화를 진행하고 있으며 확정된 표준으로는 ISO/IEC 18000-6 Type A, Type B가 있다. 또한 산업계의 자발적인 RFID 규격단체로서 EPCglobal있다. 이곳은 MIT의 Auto-ID 센터에서 RFID 기술 보급 및 활성화를 추진하기 위한 비영리단체

※ 본 연구는 2008년 인천대학교 교내연구비 지원으로 수행되었음.

† 정 회 원 : (주)이노와이어리스 정보통신연구소 연구원

†† 종 신 회 원 : 인천대학교 컴퓨터공학과 조교수

논문접수 : 2008년 1월 16일

수정일 : 1차 2008년 10월 9일

심사완료 : 2009년 1월 16일

로써 사실상 이곳에서 표준화를 주도하고 있으며 EPCglobal Class0과 Class1 규격이 표준으로 확정되었다. 이 중에서 ISO/IEC 18000-6 Type A와 EPC Class 1에서는 Framed Slotted ALOHA(FSA) 알고리즘이 사용되었으며 ISO/IEC 18000-6 Type B와 EPC Class 0에서는 이진트리 알고리즘이 사용되었다[1,6].

FSA 알고리즘의 문제점을 해결하기 위해 여러 가지의 Dynamic Framed Slotted ALOHA(DFSA) 알고리즘들에 대한 연구가 진행되어 왔다[2-4]. 본 논문에서는 기존 DFSA의 성능을 개선한 Dynamic Framed Slotted ALOHA-Slot Congestion (DFSA-SC) 알고리즘을 제안한다. 본 논문에서 제안하는 알고리즘들은 기존 DFSA 알고리즘에 슬롯-태그 간의 혼잡도를 이용하여 다음 단계의 슬롯 수를 변화시키는 방법을 통해 성능을 개선하는 방안이다. 시뮬레이션을 통한 검증결과 사용되는 총 슬롯수를 감소시켜 태그 인식 시간의 단축은 물론 기존 DFSA 알고리즘보다 더 높은 성능을 확인할 수 있었다.

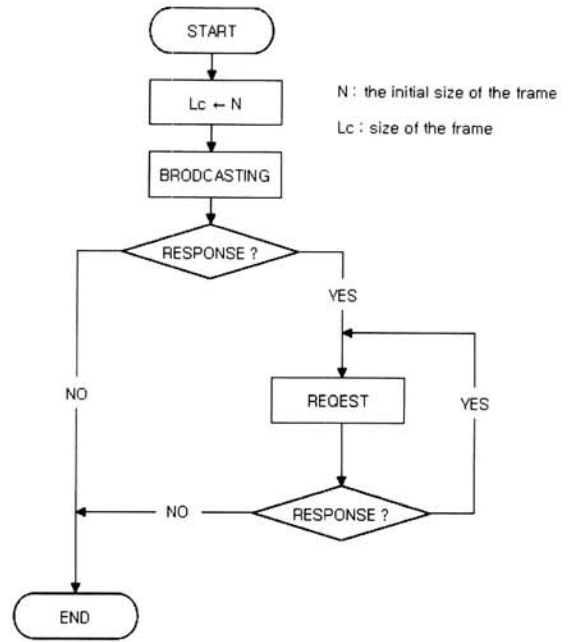
본 논문은 다음과 같이 구성되었다. 2장에서는 기존에 연구되었던 FSA 알고리즘과 여러 방법으로 변형된 DFSA 알고리즘들에 대해 설명한다. 3장에서는 제안된 DFSA-SC 알고리즘에 대해 기술하고 4장에서 제안된 알고리즘과 기존 알고리즘의 성능 평가를 수행한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

2.1 Framed Slotted ALOHA(FSA) 알고리즘

FSA 알고리즘은 ISO 18000-6 TYPE A에서 정의된 충돌방지 알고리즘이다. FSA 알고리즘은 리더와 태그의 통신에 사용되는 프레임의 크기가 고정되어 있는 알고리즘이다. 리더가 범위 안의 태그에 브로드캐스트 하는 것으로 태그인식이 시작된다. 태그는 리더의 첫 전송요청명령(REQUEST) 신호와 동시에 프레임 크기와 슬롯 선택에 대한 정보를 제공받는다. 이때 각 태그들은 전송받은 정보를 이용하여 자신이 사용할 슬롯을 선택한다. (그림 1)은 FSA 알고리즘을 순서도로 나타낸 것이며 (그림 2)은 리더가 FSA 알고리즘을 이용하여 6개의 임의의 태그를 인식하는 과정을 도식화한 것이다.

리더의 요청명령과 요청명령 사이의 간격을 리드 사이클이라고 한다. 첫 번째 리드사이클에서 태그1은 슬롯 1을 태그 2,3,4는 슬롯 2를 태그 5,6은 슬롯 3을 선택하여 자신의 태그ID를 전송하였다. 따라서 태그 1은 인식되었고 태그 2,3,4,5,6은 인식되지 않았다. 인식된 태그는 다음 단계에서 응답하지 않도록 Inactive 상태로 설정하게 된다. 두 번째 리더의 요청명령에 인식되지 않은 태그들은 다시 랜덤하게 슬롯을 선택하여 응답 하였다. 이때 첫 번째 슬롯은 어떠한 태그도 응답하지 않은 슬롯으로 Idle 슬롯이라 하며 Idle 슬롯이 발생하는 현상을 슬롯낭비현상이라 한다. 이렇듯 FSA 알고리즘은 충돌의 여부와는 상관없이 오직 고정된 프레임



(그림 1) FSA 알고리즘 순서도

	Q	B	1	2	3	B	1	2	3	B	1	2	3	B	1	2	3	B	1	2	3
R		000								101			010001					011		101	
Tag1 000		000																			
Tag2 001			001			001	001					001									
Tag3 010				010			010		010				010								
Tag4 011					011		011			011				011			011			011	
Tag5 100			100			100			100												
Tag6 101				101			101			101	101					101					101

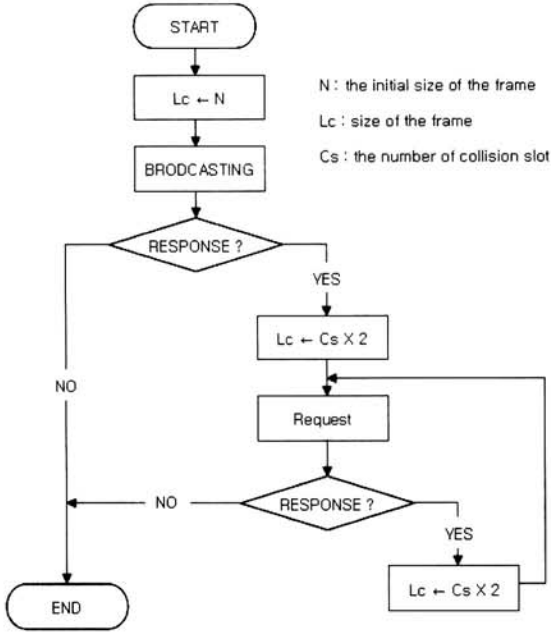
Q : Query R : Receive B : Broadcasting 1 : Slot
 Collision Idle Slot Recognition 000 : Response

(그림 2) FSA 알고리즘의 동작 예

크기만으로 동작한다. 따라서 태그인식 성능은 프레임크기에 절대적으로 영향을 받게 된다. 태그 수에 비해 너무 작은 프레임크기는 자칫 인식 자체를 불가능하게 할 수도 있으며 반대로 너무 큰 프레임 크기는 슬롯낭비현상을 일으켜 전체 성능을 저해하는 요인이 된다.

2.2 Dynamic Framed Slotted ALOHA(DFSA) 알고리즘[3]

FSA 알고리즘은 프레임 크기가 고정되어 있다. 따라서 처음 프레임의 크기를 잘못 설정하게 되면 지나친 슬롯과 태그간의 혼잡도에 의해 인식이 불가능해질 수도 있다. 반면 태그 수에 비해 슬롯의 크기가 지나치게 크게 되면 Idle 슬롯의 수가 증가하여 슬롯낭비현상이 발생한다. 이는 알고리즘의 전체 성능을 저하시키는 중요한 요인이 된다. 이러한 FSA 알고리즘의 단점을 보완하기 위한 방법으로 DFSA 알고리즘이 연구 되었다. DFSA 알고리즘은 기존 FSA 알고리즘과는 달리 프레임의 크기를 가변 시킬 수 있는 알고리즘이다. 따라서 태그인식 과정 중간에 충돌 슬롯과 Idle 슬롯 수를 이용하여 프레임크기를 가변 시켜가며 태그를 인식



(그림 3) Vogt 알고리즘 순서도

하는 알고리즘이다. 따라서 충돌 슬롯의 수를 줄여 성능 향상을 가능하게 하며 또한 Idle 슬롯의 수를 줄여 슬롯 낭비 문제도 해결 할 수 있는 알고리즘이다. DFSA 알고리즘은 그 구현 방법에 따라 많은 변형 알고리즘이 존재하고 있으며 지금까지 연구되어온 대표적인 알고리즘은 다음과 같다.

2.2.1 Novel anti-collision algorithm[4]

Novel anti-collision algorithm은 슬롯 충돌 발생 확률을 이용하여 적절한 슬롯 수를 산출하는 방법을 사용한다. 즉, 추정된 태그 수만큼 다음 단계의 프레임 크기를 설정하여 성능을 개선하는 방법이다. P_c , P_i , 그리고 P_s 를 각각 충돌 슬롯, Idle 슬롯, 인식된 슬롯의 확률이라고 하면 충돌이 발생한 슬롯의 확률은 1에서 Idle 슬롯의 확률과 인식된 슬롯의 확률을 뺀 값이다. 이를 각각의 확률식으로 나타내면 아래 식과 같다. L_c 는 프레임의 크기를 나타낸다.

$$P_c = 1 - P_i - P_s = 1 - \left(1 - \frac{1}{L_c}\right)^n - n \cdot \frac{1}{L_c} \left(1 - \frac{1}{L_c}\right)^{n-1} \quad (1)$$

전 단계에서의 충돌 슬롯 확률을 확률식에 대입하여 응답한 태그 수(n)를 추정하는 방법이다. 따라서 다음 단계의 슬롯 수는 n 개로 설정한다. 하지만 Novel anti-collision algorithms은 충돌이 발생한 슬롯의 확률이 커지게 되면 지나치게 n 값이 커지는 문제점이 발생한다. 특히 모든 슬롯에서 충돌이 발생 할 경우 n 값은 무한대로 커지게 된다. 따라서 최대 충돌 발생 비율을 임의로 설정해줘야 하는 문제점이 존재한다.

2.2.2 Vogt 알고리즘[5]

Vogt 알고리즘은 전 단계에서의 충돌슬롯의 수를 이용하

는 방법이다. 충돌슬롯이 발생하였다는 것은 곧 2개 이상의 태그가 하나의 슬롯에 동시에 전송했다는 것을 의미한다. 따라서 다음 단계의 프레임 크기를 전단계의 충돌 슬롯 수에 2를 곱한 크기로 하여 알고리즘을 동작시키는 방법이다. 이를 정리 하면 (2)과 같다.

$$Frame\ Size = \text{전 단계 충돌 슬롯 수} \times 2 \quad (2)$$

Vogt 알고리즘은 충돌 슬롯 수를 이용하여 성능을 개선하였다. 하지만 Vogt 알고리즘은 충돌의 발생을 단순히 두 개의 태그에 의해서만 충돌이 발생하는 것으로 가정하였다. 하지만 실제로 충돌은 2개 이상의 태그에 의해서 발생함으로 완벽한 접근법이라 할 수 없다.

(그림 3)은 Vogt 알고리즘을 순서도로 나타낸 것이다. (그림 4)는 Vogt 알고리즘을 사용하여 태그 6개를 인식하는 과정을 도식화한 것이다. 첫 단계에서 2개의 슬롯에서 충돌이 발생하였기 때문에 두 번째 단계의 슬롯 개수가 4개가 되었으며 2번째 단계에서 충돌이 한 곳에서 발생 하여 3번째 단계의 슬롯 개수는 2개가 되었다.

2.2.3 DFSA-2.39 알고리즘[2,7]

DFSA-2.39 알고리즘은 Vogt 알고리즘의 문제점 이었던 셋 이상의 태그가 하나의 슬롯에 전송하였을 경우까지 감안한 방법으로 확률적으로 하나의 충돌 슬롯에 몇 개의 태그가 전송을 하는가를 계산하여 이를 충돌 슬롯 수에 곱하여 나온 값을 다음 프레임 크기로 하는 방법이다. 이를 수식으로 정리하면 다음과 같다.

$$Frame\ Size = \text{전 단계 충돌 슬롯 수} \times 2.3992 \quad (3)$$

Vogt 알고리즘에서는 단순히 2를 곱하여 다음 단계의 슬롯 수를 결정하였지만 DFSA 알고리즘에서는 하나의 슬롯에 응답할 태그 수를 확률적으로 계산한 값을 곱하여 다음 단계의 슬롯 수를 결정하므로 좀 더 좋은 성능을 보인다.

DFSA-2.39 알고리즘과 Vogt 알고리즘 모두 최초 처음단계에서의 충돌빈도를 전혀 고려하지 못하는 문제가 존재한다. (그림 5)는 DFSA-2.39 알고리즘의 동작 과정을 순서도

Q	B	1	2	3	B	1	2	3	4	B	1	2	3	4	B	1	2
R	000	000	000	000	100	000	000	000	000	101	101	000	000	000	000	000	000
Tag1	000	000															
Tag2	001		001			001									001		
Tag3	010			010			010		010								
Tag4	011				011			011		011						011	
Tag5	100			100	100												
Tag6	101				101					101	101						

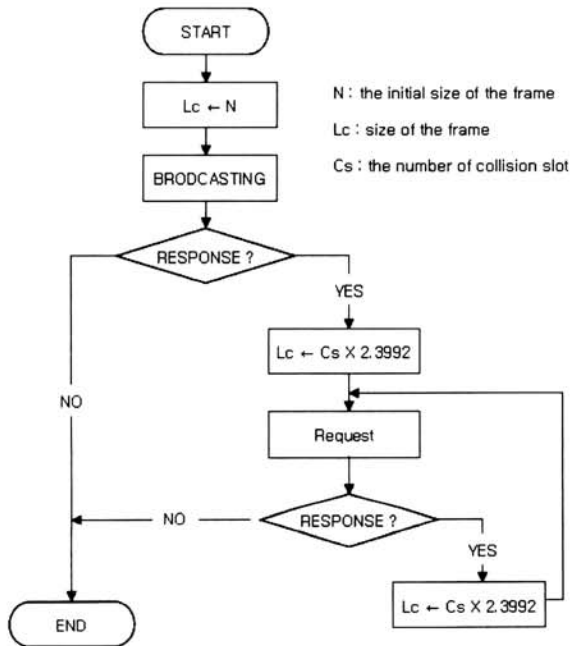
Q : Query R : Receive B : Broadcasting 1 : Slot
 Collision Idle Slot Recognition 000 : Response

(그림 4) Vogt 알고리즘의 동작 예

로 나타낸 것이며 (그림 6)은 DFSA-2.39 알고리즘을 사용하여 태그 6개를 인식하는 과정을 도식화 한 것이다. 앞에서 살펴본 (그림 4)와는 달리 두 번째 단계의 프레임 크기가 5로 커지게 되어 좀 더 빠른 태그 인식을 기대 할 수 있다.

3. 제안된 DFSA-SC 알고리즘

위에서 언급한 기존의 DFSA 알고리즘들은 충돌 슬롯수를 이용하여 적정 슬롯 수를 추정하는 방법을 통해 성능개선을 하였다. 하지만 이들 알고리즘은 최초 단계에서의 슬롯과 태그사이의 혼잡도를 전혀 고려 못하는 문제점을 갖고 있다. 예를 들면 최초 프레임 크기가 10일 때 모든 슬롯에서 충돌이 발생 하였다면 10개 보다 훨씬 더 많은 태그가 존재하게 된다. 하지만 기존 알고리즘들은 이를 전혀 고려하지 못한다. 이에 본 논문은 기존에 연구되었던 DFSA 알고리즘에 최초 단계에서의 슬롯과 태그사이의 혼잡도를 감안하여 슬롯 혼잡도를 이용한 DFSA-SC 알고리즘을 제안한다.



(그림 5) DFSA-2.39 순서도

Q	B	1	2	3	B	1	2	3	4	B	1	2	3
R	000				000					000			
Tag1 000		000											
Tag2 001			001				001						
Tag3 010				010				010				010	
Tag4 011					011				011				011
Tag5 100			100			100							
Tag6 101				101					101				

Q : Query R : Receive B : Broadcasting 1 : Slot
 Collision Idle Slot Recognition Response

(그림 6) DFSA-2.39 알고리즘의 동작 예

3.1 최적 프레임 크기

본 절에서는 리더범위 내의 태그 수에 따라 최적의 프레임 크기의 관계를 구하도록 한다. DFSA 알고리즘에서 가변적인 프레임 크기를 설정함에 있어 매우 중요한 관계라 할 수 있다. DFSA 알고리즘에서 프레임 크기를 L_c 이라고 할 때 하나의 프레임에서 태그가 전송을 시도할 확률 P 는 (4)과 같다.

$$p = \frac{1}{L_c} \tag{4}$$

또한 P_c , P_i , 그리고 P_s 를 각각 충돌슬롯, Idle 슬롯, 인식된 슬롯의 확률이라고 하면 충돌이 발생한 슬롯의 확률은 1에서 Idle 슬롯의 확률과 인식된 슬롯의 확률을 뺀 값이다. 이를 각각의 확률식으로 나타내면 (5)와 같다.

$$P_c = 1 - P_i - P_s = 1 - (1-p)^n - np(1-p)^{n-1} \tag{5}$$

전체 시스템에서 처리율 E 는 (6)과 같이 정의 할 수 있다.

$$E = \frac{P_s}{P_s + P_c + P_i} = np(1-p)^{n-1} \tag{6}$$

S 가 최대일 때 프레임크기와 태그수의 관계를 구하기 위하여 E 를 미분하면 식 (7)과 같다.

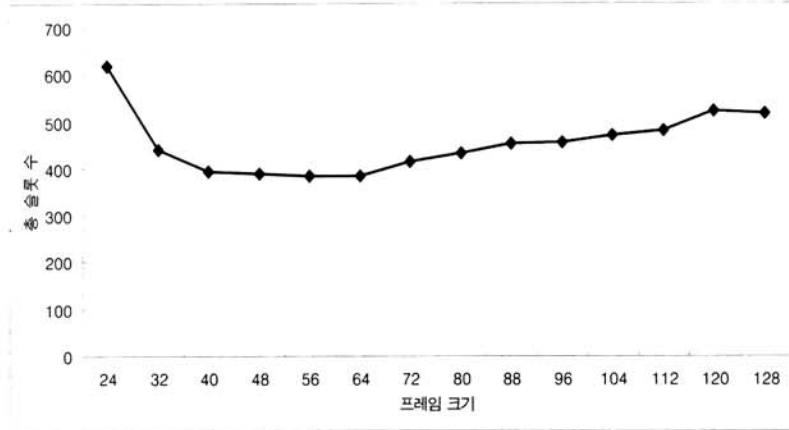
$$\frac{d}{dp} S = n(1-p)^{n-1} - n(n-1)p(1-p)^{n-2} = 0 \tag{7}$$

식 (7)을 p 와 n 으로 풀면 (8)와 같은 식을 구할 수 있다.

$$p = \frac{1}{n} \tag{8}$$

p 는 $1/L_c$ 이므로 식 (8)에 의해 최대의 성능을 기대할 수 있는 최적의 프레임 크기(L_c)는 태그 수(n)와 같은 경우라는 것을 알 수 있다. 따라서 리더 범위내의 태그의 개수를 추정하여 그 크기만큼을 프레임 크기로 설정할 때 가장 큰 성능을 기대 할 수 있다. 아래 (그림 7)은 FSA 알고리즘을 사용한 경우로 리더 범위 내 태그가 100일 때 슬롯수의 변화에 따른 성능을 나타낸다. 100번 반복 후 평균값으로 그래프를 그렸다.

(그림 7)을 보면 FSA 알고리즘을 사용하여 태그 100개를 인식 할 때 프레임 크기가 약60일 때 가장 좋은 성능을 나타내는 것을 확인 할 수 있다. 그 이유는 전체 태그 100개



(그림 7) 태그100일 때 최초 프레임 크기에 따른 성능

모두 인식하는 동안 REQUEST 단계가 반복 되면서 인식되지 않은 태그의 수가 줄어들기 때문이다. 이는 결국 FSA 알고리즘은 리더 범위 내 태그 수를 알고 있어도 좋은 성능을 기대 할 수 없다는 것을 나타낸다.

3.2 리더 범위 내 태그 수 찾기

3.1절에서 살펴보았듯이 FSA 알고리즘은 DFSA 알고리즘에 비해 더 좋은 성능을 나타 낼 수 없으며 DFSA 알고리즘에서 가장 높은 성능을 기대할 수 있는 프레임의 크기는 리더 범위 내 인식할 태그 수와 일치할 때이다. 따라서 본 논문에서 제안하는 DFSA-SC 알고리즘은 리더범위 내 태그 수를 빠르게 찾기 위한 방법이다. 리더가 태그를 인식할 때 최초 단계는 리더범위 내의 태그 수를 전혀 알 수 없는 단계이다. 따라서 실제 태그 수와 추정된 태그 수와의 차이가 가장 큰 단계이다. 이에 본 논문은 최초 단계에서의 슬롯과 태그사이의 혼잡도를 감안하여 보다 빠르게 리더 범위 내 태그 수를 추정하는 방법을 제안한다. 첫 단계에서의 슬롯과 태그사이의 혼잡도가 높아질 수록 실제 태그의 수가 기하급수적으로 증가하는 것을 식(1)에서 알 수 있다. 하지만 식(1)은 슬롯과 태그사이의 혼잡도가 1이 되었을 때 태그수가 무한대로 증가 하게 된다. 본 논문은 이를 방지하기 위해 슬롯과 태그사이의 혼잡도 τ 는 다음과 같이 정의하도록 한다.

$$\tau = \text{충돌 슬롯 수} / \text{전체 슬롯 수} \quad (9)$$

혼잡도가 클 경우는 태그 수를 보다 빠르게 찾고 혼잡도가 작을 경우는 슬롯낭비현상을 방지하기 위하여 상수 C를 더한 값을 곱해주도록 한다. 상수 C가 크게 되면 초기 태그 수 추정이 빨라 질 수 있으나 지나치게 커지게 되면 슬롯 낭비현상이 발생하여 전체 성능을 저해시키는 요인이 된다. 리더 범위 내 태그가 100개일 때 시뮬레이션 결과 가장 좋은 성능을 보인 C의 크기는 2이었다. 하지만 C의 값은 환경에 따라 가변적으로 변화 시킬 수 있다. 이를 종합하여 본 논문에서 제안하는 알고리즘은 다음과 같다. 첫 단계에서

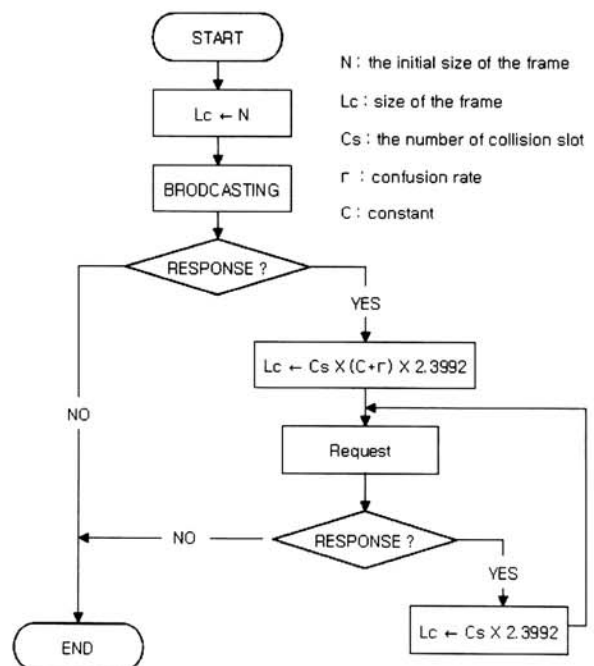
충돌이 발생한 슬롯 수에 [2]논문에서 제안한 2.3992를 곱하고 혼잡도에 상수 C를 더한 값에 곱하여 나온 값을 두 번째 단계에서의 프레임 크기로 한다. 그 후 다음단계부터는 기존의 DFSA 방법을 따르도록 한다. 이를 수식으로 정리 하면 (10), (11)과 같다.

$$\text{두 번째 Frame Size} = \text{첫 단계에서의 충돌 슬롯 수} \times 2.3992 \times (\tau + C) \quad (10)$$

$$\text{Frame Size} = \text{전 단계 충돌 슬롯 수} \times 2.3992 \quad (11)$$

따라서 처음 단계에서의 태그 수 추정이 좀 더 정확해져 이후의 전체 과정에 영향을 주어 전체적인 성능 향상을 기대할 수 있다.

(그림 8)는 제안된 알고리즘을 사용하여 태그 6개를 인식



(그림 8) 제안된 DFS-ALOHA 알고리즘 순서도

Q	B	1	2	3	B	1	2	3	4	5	6	7	8
R	000	000			100001	011010						101	
Tag1	000												
Tag2	001					001		001					
Tag3	010					010				010			
Tag4	011					011				011			
Tag5	100					100							
Tag6	101					101							101

Q : Query
 : Collision

R : Receive
 : Idle Slot

B : Broadcasting
 : Recognition

1 : Slot
000 : Response

(그림 9) 제안된 DFSA-CS 알고리즘의 동작 예

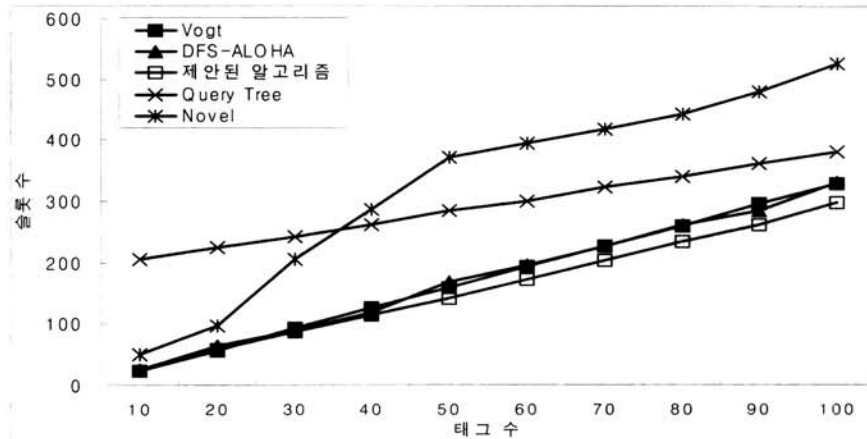
하는 과정을 도식한 그림이다. Tag2, Tag3, Tag4는 2번 슬롯을 Tag5, Tag6은 3번 슬롯을 선택하여 충돌이 발생 하였다. 첫 번째 과정에서 혼잡도는 2/3이다. 상수 C의 값은 2로 설정하였다. 따라서 다음단계에서의 슬롯의 크기는 (5)식에 의해 8이 되었다. 두 번째 단계에서의 프레임의 크기는 8로 설정되었다. 기존의 알고리즘에 비해 본 논문에서 제안한 DFSA-CS 알고리즘이 보다 빠른 태그 수를 추정하여 성능이 개선된 것을 확인 할 수 있다. 제안된 알고리즘의 특성상

태그 수에 비해 슬롯의 수가 적을 경우 보다 큰 성능을 기대할 수 있으며 태그 수에 비해 프레임의 크기가 큰 경우는 혼잡도 값이 작아지기 때문에 슬롯 낭비 현상을 피할 수 있다.

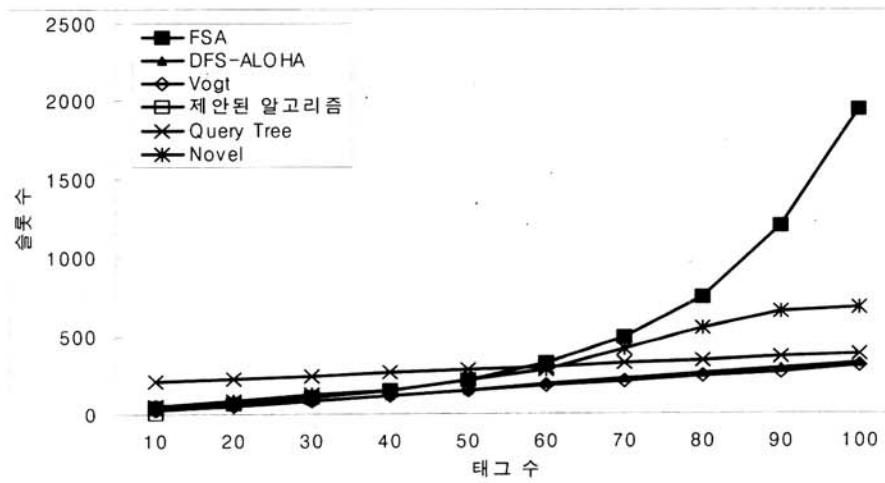
4. 실험 및 평가

본 절에서는 지금까지 살펴본 알고리즘들과 제안된 알고리즘에 대한 성능평가를 실시한다. 성능 평가를 위한 시뮬레이션 환경은 C언어를 기반으로 설계하였다. 성능 측정 기준은 모든 태그를 인식하기까지 사용된 총 슬롯의 수로 하였다. 실험값은 최초 슬롯수를 8, 16, 32, 64, 128의 각각의 경우에 태그 수를 10개부터 100개까지 10개씩 증가 시키며 100번 반복 후 평균값으로 하였다. 식(10)에서 정의한 상수 C는 2로 설정하였다. Query Tree algorithm[6]의 경우 태그 ID 길이는 96bit로 태그ID가 순차적인 경우로 가정하였으며 질의-응답의 한 사이클의 시간은 하나의 슬롯 타임과 같다고 가정하였다.

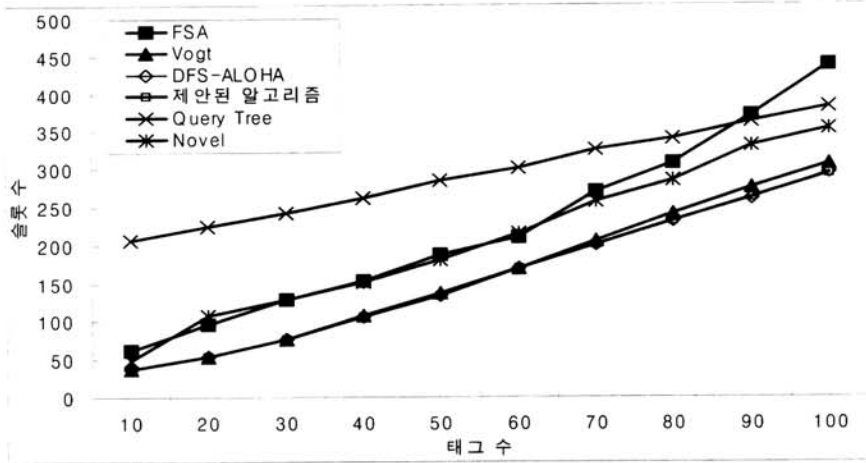
(그림 10)는 최초 프레임 크기가 8인 경우로 FSA 알고리즘으로는 실험값을 얻지 못하였다. 이는 너무 큰 혼잡도로 인해 태그 인식이 불가능한 경우이다. (그림 10)~(그림 14)를



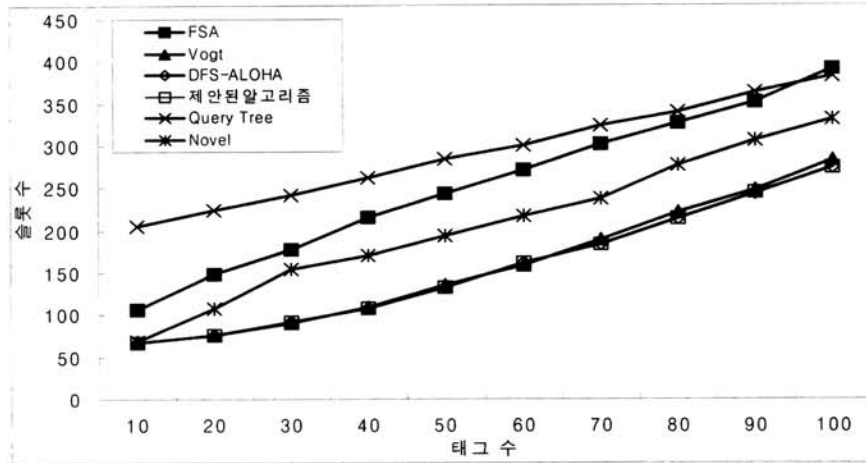
(그림 10) 최초 프레임 크기가 8인 경우



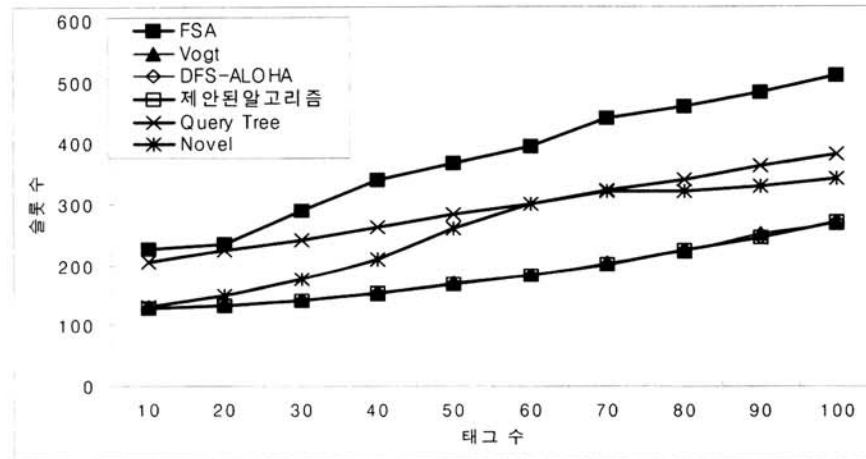
(그림 11) 최초 프레임 크기가 16인 경우



(그림 12) 최초 프레임 크기가 32인 경우



(그림 13) 최초 프레임 크기가 64인 경우



(그림 14) 최초 프레임 크기가 128인 경우

살펴보면 제안된 알고리즘이 기존 알고리즘들에 비해 성능 개선된 것을 볼 수 있다. 특히 프레임 크기가 작을수록 그 성능개선이 뚜렷한 것을 확인 할 수 있다. 즉 최초 프레임 크기가 8인 경우에 성능이 가장 좋게 나타났다. 이 경우는 상대적으로 가장 혼잡도가 높은 경우이다. 태그 수에 비해 프레임의 크기가 가장 작아 최초 첫 단계에서 충돌이 가장 많이 발생하는 실험 환경이기 때문이다. 또한 최초 프레임

크기가 커지게 되면 최초 단계에서의 충돌이 상대적으로 적어지게 된다. 하지만 제안된 알고리즘의 혼잡도는 최초 단계에서의 충돌이 많이 발생하면 커지고 충돌이 적으면 적어지는 구조이기 때문에 최초 단계에서의 충돌이 적다하더라도 슬롯이 낭비되는 현상을 방지 할 수 있다. 또한 Query Tree는 대표적인 트리 기반 알고리즘으로 태그ID가 순차적인 경우 제안된 알고리즘이 우수한 성능을 나타내는 것을

확인 할 수 있다. 특히 태그수가 적을수록 더 좋은 성능을 나타낸다. 이는 태그ID의 중복도가 높고 태그 수가 적을 경우 트리기반 알고리즘 보다 ALOHA 기반 알고리즘이 좀 더 좋은 성능을 기대할 수 있다.

5. 결 론

본 논문은 ISO 18000-6 TYPE A에 정의된 FSA 알고리즘을 기반으로 성능을 개선한 DFSA-SC 알고리즘을 제안하였다. 본 논문에서는 기존 DFSA 알고리즘의 태그수 추정 복잡도를 개선하고 충돌 슬롯의 수와 최초 태그 인식 과정에서의 혼잡도를 이용하여 성능을 개선하였다. 시뮬레이션 결과 제안된 DFSA-SC 알고리즘이 다른 알고리즘에 비해 보다 우수한 성능을 보이고 있음을 확인 할 수 있었다. 또한 대표적인 트리기반의 알고리즘인 Query Tree 알고리즘과의 비교실험을 통해 알로하 기반의 알고리즘이 태그수가 적고 ID의 중복성이 클 경우 트리기반 알고리즘보다 높은 성능을 기대 할 수 있다는 것을 확인 할 수 있었다.

참 고 문 헌

- [1] ISO/IEC 18000-6 : 2003(E), Part 6 : Parameters for air interface communications at 860-960 MHz, Nov., 26, 2003.
- [2] 차재룡, 김재현, "RFID 시스템에서의 태그 수를 추정하는 ALOHA 방식 Anti-collision 알고리즘," 한국통학회논문지, Vol.30, No.9A, pp.814-821. March, 2005.
- [3] Jae-Ryong Cha, Jae-Hyun Kim, "Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system," Consumer Communications and Networking Conference, 2006. CCNC 2006. 2006 3rd.
- [4] J. Cha and J. Kim, "Novel anti-collision algorithm for fast object identification in RFID system," in Proc. Parallel and Distributed System 2005, pp.177-172.
- [5] H. Vogt, "Efficient Object Identification with Passive RFID tags," In International Conference on Pervasive Computing, Zurich, 2002, pp.98-113.
- [6] 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, pp.75-84. ACM. August, 2000.
- [7] F.C. Schoute, "Dynamic Length ALOHA," IEEE Transactions on Communication, Apr., 1983, Vol.31, pp.565-568.



이 재 구

e-mail : whistler29@hanmail.net
2008년 인천대학교 컴퓨터공학과(학사)
2008년~현 재 (주)이노와이어리스 정보통신연구소 연구원
관심분야 : RFID, Sensor Network



최 승 식

e-mail : sschoi@incheon.ac.kr
1988년 연세대학교 전자공학과(학사)
1990년 KAIST 전기 및 전자공학과(석사)
2002년 KAIST 전기 및 전자공학과(박사)
1990년~1998년 KT 통신망연구소 전임 연구원
2002년~2004년 KT 서비스 개발 연구소 책임연구원
2002년~현 재 인천대학교 컴퓨터공학과 조교수
관심분야 : 무선 MAC, 무선자원관리, 무선인터넷 프로토콜