

지능형 선반 시스템을 위한 태그 충돌 방지 알고리즘¹⁾

박재규^o 조현주 김정애 유기영

경북대학교 컴퓨터공학과 정보보호연구소

{skytosee^o, i_hjcho, sarah}@infosec.knu.ac.kr yook@knu.ac.kr

Anti-Collision Algorithm for Smart Shelves System

Jae-Gyu Park^o Hyeon-Ju Cho Jung-Ae Kim Kee-Young Yoo

Dept. of Computer Engineering, Kyungpook National University.

요 약

최근 리더기가 RF를 이용하여 사물에 부착되어있는 전자태그를 인식하는 비접촉 기술인 RFID(Radio Frequency Identification)기술을 대형 물류, 제조업 등에 확대 적용함에 따라 기업 공급망의 가시성이 높아지고 효율적인 관리가 이루어지고 있다. 하지만 RFID 태그가 널리 보급됨에 따라 태그간의 충돌문제가 대두되고 있다. 본 논문에서는 태그간의 충돌문제(Anti-collision problem)를 최소화 할 수 있는 알고리즘을 제시한다.

본장에서는 현재 많이 알려진 태그 충돌방지 알고리즘을 살펴본다.

(1) Basic 이진 탐색 알고리즘[5] : Basic 이진 탐색 알고리즘은 충돌이 발생한 부분을 반씩 줄여 나감으로써 리더기의 판독 영역 내에 있는 태그를 하나씩 인식하여 충돌을 줄여간다. 충돌이 발생한 최상위 bits가 1이면 태그 ID 전송이 지연이 되고, 충돌이 발생한 최상위 bits가 0인 태그는 ID를 전송한다. 이 과정을 순차적으로 반복하여 태그를 인식한다. 이 알고리즘의 문제점은 태그 하나를 인식하기 위해서 항상 전체 비트를 리더로 전송해야하는 문제점과 리더의 태그 인식과정에 인식된 태그가 중복된 응답을 함으로써 리더가 필터링 하도록 하는 문제점을 가지고 있다.

(2) Dynamic 이진 탐색 알고리즘[5] : Basic 이진 탐색 알고리즘을 보완하기 위한 이 알고리즘은 충돌이 발생하는 bit의 위치를 가르키는 NVB(Number of Valid Bits)를 추가하여 전송함으로써 Basic 이진 탐색 알고리즘에 비하여 전송되는 데이터 양을 줄이고 태그의 에너지 소비와 태그를 인식하기 위한 탐색 시간을 줄여준다.

(3) Basic slotted 이진 트리 알고리즘[6] : slotted 이진 탐색 알고리즘은 Time slotted를 사용하여, 태그를 인식하는 알고리즘으로 이진 탐색 알고리즘 보다 더 빠른 인식속도 향상을 위하여 제안되었다. 하지만 태그로부터의 전송이 없는 idle 슬롯이 발생하는 문제점과 전체 슬롯의 개수가 태그 수보다 작음으로 인한 슬롯의 충돌 문제가 있다.

(4) Modified slotted 이진 트리 알고리즘[6] : Basic slotted 이진 트리 알고리즘을 보완하기 위한 이 알고리즘은 ternary feedback을 이용하여 idle 슬롯을 제거함으로써 충돌 해결을 더 빠르게 할 수 있다.

(5) Bit-by-bit 이진 트리 알고리즘[7] : 판독거리 내의 모든 태그들에게 ID중 원하는 bit를 순서대로 요청하게 되면 모든 태그는 응답으로 0 또는 1을 전송하는 방식으로 Auto-ID센터에서 제안하였다. 이 알고리즘의 단점은 리더가 항상 태그의 전체 bits를 요구하게 된다.

(6) Modified bit-by-bit 이진 트리 알고리즘[8] : Modified

1. 서 론

RFID기술은 유비쿼터스 시대에 주목받는 기술이다. RFID는 태그와 리더기로 구성된다. 각 태그에는 상품의 고유한 EPC(Electronic Product Code)[1]가 들어있고 리더기는 안테나를 통해 들어온 태그정보를 읽어 개체를 인식한다. 이러한 RFID기술의 비접촉성특성과 제품관리의 편의성 때문에 많은 응용에 적용되고 있다. 예를 들어 지능형 선반 시스템의 경우 상품 하나하나를 실시간 추적 및 감시 하여 효율적으로 관리를 할 수 있어 상품의 압·출고에 드는 인력상비와 시간소모를 줄여 생산성 또한 높일 수 있다. RFID기술을 도입한 유럽에 있는 대형 유통업체인 매트로 그룹은 'Future Store'[2]를 열어 시간, 비용, 인력절감으로 커다란 이득을 보고 있다.

하지만 RFID태그가 널리 보급됨에 따라 태그간의 충돌문제(Anti-collision problem)[3][4]가 대두되고 있다. 이러한 태그간의 충돌문제를 최소화 할 수 있는 알고리즘의 필요성을 무엇보다 중요한 연구대상이 되고 있다. 실시간에 상품의 재고유무를 파악하기 위해서는 태그 간의 충돌을 최소화시키는 것이 필수이다. 현재 충돌방지 알고리즘의 표준으로는 Auto-ID 센터에서 제안한 bit-by-bit와 slotted aloha가 있다. 본 논문에서는 RFID 태그간의 충돌문제를 최소화 할 수 있는 Anti-Collision Algorithm을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 태그 충돌방지 알고리즘 대해 고찰하고, 3장에서 제안하는 충돌방지 알고리즘에 대해 설명하며, 4장에서 제안한 충돌방지 알고리즘 보안성 분석, 마지막으로 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

1) 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음

bit-by-bit 이진 트리 알고리즘은 충돌이 발생하였을 때, bit가 1인 태그들이 일시적으로 응답하지 않는 대기(inactive)상태로 만들어 하나의 태그가 인식이 되면 다시 active 상태로 만든다. 이 경우는 랜덤할 경우 리더가 태그를 읽어 들이는 시간이 줄어들지 않는 단점이 있다.

(7) Enhanced bit-by-bit 이진 트리 알고리즘 [8] : Modified bit-by-bit 이진 트리 알고리즘을 보완하기 위해 제안된 알고리즘이다. 이 알고리즘을 사용하면 결과적으로 태그의 전송 회수가 1만큼 증가하지만 리더의 전송 회수가 2만큼 줄고, 리더가 태그에게 보내는 전송 bit수보다 태그가 리더에게 보내는 전송 bit수가 훨씬 크다는 것을 고려함으로써 성능 향상이 가능하게 하였다. 하지만 전송비트는 줄지만 전송 횟수가 증가되는 문제점이 있다.

3. 제안하는 충돌 방지 알고리즘

본장에서는 지능형 선반 시스템을 위한 충돌방지 알고리즘을 제안한다.

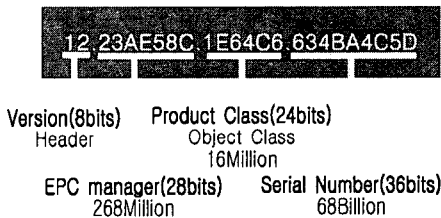


그림 1. 96bits Electronic Product Code 구성

EPC 96bits의 전체적인 구성을 그림 1에서 보여준다. 그림에서 Header, Manager, Product Class는 노출되어서는 안 될 정보이며 SN(Serial Number)은 노출되어도 보안상 문제가 되지 않는다. 태그에서 리더에게 전송할 때 보안상에서 문제가 되는 정보는 일방향 해쉬를 사용해서 보낸다. Common bits는 동일회사에서 생산되는 동일한 종류의 제품의 경우 상위 60bits가 같으며, SN값만 각 제품마다 다르다.

그림 2는 제안하는 충돌 방지 알고리즘으로 태그를 인식하는 과정이다.

REQUEST ≤	0	0	0	1	영역제한
태그 1	0	0	0	1	인식
REQUEST 3번	비트 0 송				무응답
REQUEST ≤	0	0	1	0	영역제한
태그 2	0	0	1	0	인식
REQUEST 3번	비트 0 송				무응답
REQUEST ≤	0	1	0	0	영역제한
태그 3	0	0	1	1	인식
REQUEST 3번	비트 0 송				무응답
REQUEST ≤	1	0	0	0	영역제한
태그 4	1	0	1	1	불인식
REQUEST 3번	비트 0 송				무응답

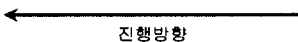


그림 2. 태그의 인식 과정

리더가 태그에게 REQUEST를 보내면 REQUEST를 만족하는 영역 내의 태그들만 응답을 한다. 응답 태그 영역을 제한하여 태그 충돌을 줄이고 태그 인식속도를 향상시킨다.

인식영역 내에 태그가 더 이상 존재하지 않을 때는 비트를 자리 왼쪽으로 이동하는데 그전에 환경적인 요인으로 처음 영역에 응답 못한 태그가 있을 것을 고려하여 같은 REQUEST를 3번 보내고 응답이 없을 때 다음 REQUEST를 보낸다.

제한된 영역이지만 비트수가 왼쪽으로 이동할수록 제한 영역이 넓어져서 태그 수가 많아지고 충돌 발생 확률이 높다. 충돌이 발생되면 bit-by-bit 이진 트리 알고리즘으로 충돌된 bit가 0이면 저장하고 1이면 inactive 상태로 하여 REQUEST영역내의 모든 태그를 인식한다. 여기서 BIT-BY-BIT는 태그 마지막 BIT부터 전송하여 처음 BIT까지 전송한다. 이렇게 반복하면서 모든 태그들을 인식하여 기존 알고리즘보다 충돌횟수를 줄여 전송 비트수를 줄이므로 태그 인식속도가 증가하였다.

표 1. 사용된 태그 Serial Number

태그 1	0001
태그 2	0010
태그 3	0011
태그 4	1011

리더가 표 1에 있는 4개의 SN를 가진 태그를 인식하기 위해서 REQUEST(≤0001)를 명령을 태그에게 보낸다. 그러면 마지막 비트가 0000과 0001인 태그가 응답을 하게 된다. 하지만 여기에서 충돌은 발생되지만 2개의 태그뿐이기 때문에 바로 인식을 한다. 그리고 비트를 왼쪽으로 한자리씩 1을 이동시키는데 이렇게 함으로써 응답해야 될 태그수를 제한하고 인식속도를 빠르게 할 수 있다. 제한된 영역 내에서 태그가 충돌이 발생하면 충돌된 k번째 bit가 1인 태그들은 inactive[8] 상태로 만들고 0인 bit를 가진 태그는 저장한다. 여기에서 inactive 상태는 알고리즘 수행 중에 충돌이 발생한 bit가 1이면 리더의 REQUEST에 일시적으로 응답을 하지 않는 상태를 말하면 태그를 인식하면 다시 active 상태로 바꾸는 상태를 의미한다.

REQUEST≤0001을 수행하면 태그 1이 인식되고 더 이상 인식할 태그가 없으므로 REQUEST≤0010한다. 이후 태그 2가 바로 인식되고 이하에 있는 태그가 더 이상 없으므로 자리 이동을 한다. REQUEST≤0100은 수행하여 다음 태그를 요청한다. 같은 SN를 가진 태그가 없으므로 다음 번째 태그 3번이 인식된다. 이미 태그 1번과 태그 2번은 인식 된 상태이므로 이 요청에는 더 이상 응답을 하지 않는다. 이런 방법으로 태그 2⁰부터 2⁵까지 전부 인식한다.

표 2는 17개의 태그가 있을 때 리더의 요청에 대한 인식순서를 보여주기 위한 것이다. Collision bit-by-bit는 충돌이 발생되었을 때 bit-by-bit로 태그를 인식한다는 의미이고 충돌이 일어나지 않았을 때는 바로 next bit를 요구하여 다음 영역으로 이동한다는 의미이다.

표 2. 태그의 인식 순서

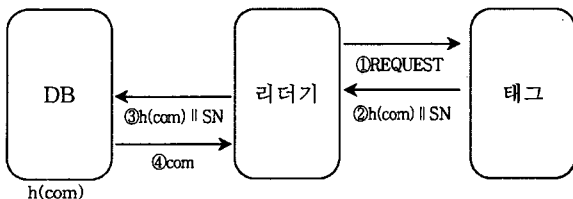
순번	BITS	REQUEST	Collision	인식순서
1	00000		Collision bit-by-bit	①
2	00001	00001	Collision bit-by-bit	②
3	00010	00010	next bit	①
4	00011		Collision bit-by-bit	②
5	00100	00100	Collision bit-by-bit	①
6	00101		Collision bit-by-bit	③
7	00110		Collision bit-by-bit	②
8	00111		Collision bit-by-bit	④
9	01000	01000	Collision bit-by-bit	①
10	01001		Collision bit-by-bit	⑤
11	01010		Collision bit-by-bit	③
12	01011		Collision bit-by-bit	⑦
13	01100		Collision bit-by-bit	②
14	01101		Collision bit-by-bit	⑥
15	01110		Collision bit-by-bit	④
16	01111		Collision bit-by-bit	⑧
17	10000	10000	Collision bit-by-bit	①

표 2에서 보여주는 것은 리더 쪽에서 응답할 대상 태그를 축소시키기 위해 대상 태그를 요청하는 질의(REQUEST)를 태그들에서 전송하게 되는데, 이때 대상 태그에 대한 범위를 2의 역승(2⁻ⁱ)으로 증가시킨다. 따라서 리더 질의에 대한 응답 대상 태그의 수 T는 (1)과 같이 되며, 전체 태그를 대상으로 하지 않기 때문에 충돌할 가능성이 있는 태그의 수가 줄어들게 되어 인식률을 높일 수 있게 된다. 그리고 충돌이 발생하는 경우는 BIT-BY-BIT BINARY 전송과 동일하게 동작한다.

$$Tag 수 T = \begin{cases} \sum_{i=0}^n 2^i & \text{for } 0 < i \leq 1 \\ \sum_{i=2}^n 2^i - \sum_{i=0}^{i-1} 2^i & \text{for } 2 \leq i \leq n \end{cases} \quad (1)$$

4. 제안한 충돌 방지 알고리즘의 보안성 분석

본장에서는 제안한 충돌 방지 알고리즘의 보안성을 분석한다. RFID시스템은 특성상 여러 가지 공격에 취약한 부분이 있다. 개인 사생활 침해로 인한 문제가 가장 크지만 대형 할인점이나 소매점의 경우 물건의 도난이 가장 큰 문제가 될 것이다. 제안한 알고리즘에서는 태그의 SN를 모두 전송할 때 생기는 도청공격을 막기 위하여 96bits 태그와 관련된 보안상 중요한 정보들은 DB에 미리 저장한 후, 고유한 36bits만 리더가 태그에게 전송 요청을 하게하여 보안이 취약한 저가 태그의 보안성을 높여줄 수 있다.



h(com): hash(Common bits)
 h(com) || SN: hash(Common bits) || Serial Number

그림 3. 리더기와 태그와의 통신 프로토콜

그림 3은 리더기와 태그와의 통신 프로토콜을 보여준다. 처리과정은 다음과 같다.

- ① 리더기가 태그에게 REQUEST를 한다.
- ② 태그는 저장되어있는 해쉬 된 common bits와 SN을 연결(concatenation)하여 리더에게 전송한다.
- ③ 리더기는 DB에 수신된 태그 값을 전송한다.
- ④ DB는 저장되어 있는 해쉬 값과 입력된 값을 비교하여 정상적인 태그인지를 판별하고 Common bits를 리더에게 전송한다.

5. 결론 및 향후 과제

본 논문에서는 지능형 선반 시스템에 적합한 충돌 방지 알고리즘을 제안하였다. 제안한 RFID 태그간의 충돌 방지 알고리즘을 지능형 선반에 적용하기 위해서 문제가 되는 태그들 간의 충돌 문제를 최소화하여 지능형 선반을 구축하는데 보다 효율적인 시스템을 가능토록 하였다.

향후 연구과제로는 리더 질의에 응답할 대상이 되는 태그의 범위를 지정해주는 REQUEST가 2의 승수로 한 비트씩 이동하는 것을 시뮬레이션을 통해 실험을 해 REQUEST 비트의 이동 간격을 최적화 시킬 수 있는 방안에 대한 연구가 필요하다.

[참고문헌]

- [1] S. A. Weis, S. E. sarma, R. L. Rivest and D. W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems,"First International conference on security in pervasive Computing., March, 2003.
- [2] D.W. Engels and S.E. Sarma, "The reader collision problem," 2002 IEEE International Conference on Systems, Man and Cybernetics, Vol.3, pp.6, Oct 6-9, 2002.
- [3] Auto-ID Center, "EPC Tag Data Specification", June, 2003
- [4] <http://www.future-store.org>
- [5] K. Finkenzeller, *RFID Handbook: Fundamentals and applications in Contactless smart Cards and Identification*, Second Edition, John Wiley & Sons Ltd, pp.212-219, 2003.
- [6] J.L. Massey, "Collision resolution algorithms and random-access communications," Univ. California, Los Angeles, Tech. Rep. UCLAENG -8016, Apr., 1980.
- [7] Auto-ID Center, Draft Protocol Specification for a Class 0 Radio Frequency Identification tag., 2003
- [8] 최호승, 김재현, "Ubiquitous ID시스템에서의 Enhanced bit-by-bit 이진 트리 알고리즘. August, 2004