

---

# RFID 시스템에서 Hybrid 방식을 이용한 효율적인 충돌 회피 기법

신성용\* · 황경호\*\*

An Efficient Hybrid Anti-collision Method in RFID systems

Song-Yong Shin\* · Gyung-Ho Hwang\*\*

---

이 논문은 2008년도 한밭대학교 교내학술연구비의 지원을 받았음.

---

## 요 약

RFID 시스템에서 다수의 태그가 동시에 자신의 ID를 리더기에 전송하게 되면 충돌로 인해 리더기의 태그 인식 시간이 지연 된다. 따라서 리더기의 태그 인식시간을 줄이기 위한 효율적인 충돌방지 기술이 필요하다. 본 논문에서는 QT(Query Tree) 방식을 기반으로 DFSA(Dynamic Framed Slotted Aloha) 방식을 추가한 hybrid 충돌 회피 방법에 대해서 제안하고 시뮬레이션을 통하여 기존의 방법과 성능을 비교한다.

## ABSTRACT

If multiple Tags in the RFID System transmit their IDs to the Reader at the same time, tag identification time is delayed due to collisions. Therefore, to reduce the reader's identification time, an efficient anti-collision technology is needed. In this paper, we propose a hybrid anti-collision method based on the QT and DFSA. Then, the performances of proposed method are compared with the existing method through extensive simulations.

## 키워드

RFID, 충돌 회피, 태그수 추정, DFSA, 쿼리 트리, 알로하

## Key word

RFID, Anti-collision, Tag Estimation, DFSA, Query Tree, Aloha

---

\* 준회원 : 한밭대학교 컴퓨터공학과

\*\* 정회원 : 한밭대학교 컴퓨터공학과(교신저자, gabriel@hanbat.ac.kr)

접수일자 : 2012. 06. 12

심사완료일자 : 2012. 07. 09

## I. 서 론

RFID(Radio Frequency Identification) 시스템은 물건에 붙어 있는 전자 태그들과 이 태그들의 정보를 읽는 리더기로 구성되어 있으며, 태그에는 ID와 물건에 대한 정보들을 가지고 있다. 태그는 그 정보들을 리더기로 송신하며, 리더기는 수신한 정보들을 처리하고, 그 결과를 다시 태그로 전송한다. 이 때 2개 이상의 태그들이 동시에 자신의 정보를 보내게 되면 충돌이 발생하여 태그 정보를 읽는데 지연이 생긴다. 이것을 막기 위해 많은 충돌회피 알고리즘이 연구 되고 있으며, 그 중에 대표적으로 FSA(Framed Slotted Aloha)기반 알고리즘과 QT(Query Tree) 기반 알고리즘이 있다.

본 논문에서는 우선 II 장에서 RFID 태그 인식 지연을 줄이기 위한 기존의 충돌 방지 알고리즘들에 대해 살펴보고 III 장에서 QT방식과 DFSA 방식을 혼합한 제안 알고리즘에 대해 설명을 한 후, IV장에서 시뮬레이션을 통해서 기존 알고리즘과 제안 알고리즘의 성능을 비교하고, V장에서 결론을 맺는다.

## II. 기존 충돌 회피 알고리즘

기존의 RFID 충돌 회피 알고리즘으로는 크게 ALOHA 기반 알고리즘과 QT 기반 알고리즘으로 나눌 수 있다. ALOHA 기반 알고리즘에서 리더기는 태그들에게 정보를 보내달라는 신호를 보내고, 이 신호를 받은 태그들은 경쟁적으로 리더기에 자신의 정보가 담긴 신호를 보낸다. 그 신호를 성공적으로 받았다면 리더기는 자신이 정보를 잘 받았다는 신호를 태그에게 보낸다. 그리고 태그는 일정 시간 동안 기다리면서 그 시간 내에 신호가 오지 않으면 충돌이 발생했다고 가정하고 다시 신호를 보낸다. 이 알고리즘은 상당히 비효율적이기 때문에 충돌을 줄이기 위해 더 효율적인 방법으로, 리더기는 한 라운드의 시간을 여러 개의 슬롯으로 나누고, 태그는 그 수를 바탕으로 자신의 아이디를 보낼 슬롯을 임의로 선택한다. 이 때 충돌이 발생하면 태그는 임의의 시간을 기다린 후에 재전송을 한다. 리더기가 슬롯의 개수를 너무 많이 설정하면, 빈 슬롯이 많아져 자원의 낭비가 커지고, 슬롯의 개수를 너

무 적게 잡으면 충돌이 많아져 태그인식 지연이 길어진다. ALOHA 방식에서는 슬롯의 개수를 일정하게 유지하는 FSA(Framed Slotted Aloha)방식과 매 번 리더기가 슬롯 개수를 정할 수 있는 DFSA(Dynamic Framed Slotted Aloha)방식으로 나뉜다.

FSA 알고리즘은 첫 라운드에 정한 슬롯의 수를 변경하지 않기 때문에 태그의 개수를 정확히 모르는 상황에서 매우 비효율적이다. 반면 DFSA 알고리즘은 기본적으로 FSA와 같지만, 한 라운드가 끝날 때마다 다양한 방법으로 태그의 개수를 추정해서 슬롯의 수를 변경해서 태그 읽기를 시작하는 방법이다. DFSA 알고리즘은 FSA 알고리즘보다 효율적이지만 정확한 태그의 개수를 예측할 수 있는 방법이 필요하다. DFA 알고리즘은 태그 예측 방법이 제대로 이뤄지지 않는 경우 FSA 알고리즘보다 나쁜 성능을 보일 수 있다.

태그 개수 추정 방법으로는 다음과 같은 방식이 있다. 첫 번째 방법(DFSA1)은 한 라운드에서 충돌이 난 슬롯의 개수( $S_c$ )를 확인한 후  $S_c$ 에 2.3922를 곱해서 그 수보다 작거나 같은 최대 정수를 예상 태그 개수로 결정한다. 가장 효율이 좋은 슬롯의 개수는 경쟁에 참여하는 태그의 개수와 같기 때문에, 슬롯 개수를 예상 태그 개수로 결정한다.[1]

두 번째 방법(DFSA2)은 i-code[2]라는 방식으로 첫 라운드에서 충돌이 난 슬롯의 개수( $S_c$ )와 성공한 슬롯의 개수( $S_1$ )를 확인한 후 태그의 예상 개수를 다음 식(1)으로 정한다.

$$S_r = S_1 + (2 * S_c) \quad (1)$$

여기서 구한  $S_r$ 을 이용하여 태그의 개수에 따른 최적의 슬롯 개수를 표 1에서 선택하여 변화시켜 준다. 만일 슬롯 개수 32개의 High(27)와 슬롯개수 64개에서의 Low(17)와 같이 범위가 겹쳐지는 부분이 발생하면 이전 라운드의 슬롯의 개수를 기준으로 다음 라운드의 슬롯의 개수를 정한다.

예를 들어 이전 슬롯수가 64개이고, 다음 라운드의 예상 태그수가 53개인 경우, 53개라는 개수는 64개의 슬롯과 128개의 슬롯 모두에 포함된다. 이 때 이전 슬롯의 개수가 64개였으므로, 슬롯의 개수는 그대로 64개를 유지한다.

표 1. 슬롯 개수 할당  
Table. 1 Slot number allocation

N Slots	1	4	8	16	32	64	128	256
Low	-	-	-	1	10	17	51	112
High	-	-	-	9	27	56	129	∞

다음으로 QT(Query Tree)기반 알고리즘에서는 리더가 태그에게 태그의 정보를 요청할 때 프리픽스(prefix)를 같이 전송한다.

태그는 자기 아이디의 앞부분과 프리픽스가 일치하는 경우 자신의 정보를 전송한다. 이 때 리더기는 응답 태그가 없으면 해당 태그가 없는 것으로 판단하고, 여러 개의 태그가 동시에 응답하면 충돌로, 한 태그만 응답하면 성공으로 인식한다. 이 때 충돌이 발생했다면 리더기는 자신이 보낸 프리픽스와 일치하는 ID를 가진 태그가 여러 개가 있음을 알게 된다. 리더기는 충돌이 발생할 때마다 방금 전송한 프리픽스 뒤에 '0'과 '1'을 계속해서 붙여 나가면서 큐(queue)에 저장해 놓고, 이것들을 순차적으로 질의함으로써 태그의 ID를 가지고 트리를 만든다. 큐의 초기 값은 '0'과 '1'이다.

그림 1은 식별 ID가 각각 '010', '011', '100'인 3개의 태그가 있다고 가정하고 쿼리 트리 프로토콜을 실행한 예제이다. 초기 큐에는 프리픽스 {'0', '1'}이 설정되어 있으며 리더기는 큐에서 프리픽스를 꺼내 태그에게 질의한다. 먼저 '0'을 질의했을 때 태그 '010'과 '011'이 자신의 ID와 프리픽스가 일치하므로 동시에 응답하게 되어 충돌이 발생한다. 이것을 통해 리더기는 '0'으로 시작하는 태그가 2개 이상 있다고 보고, '0'뒤에 '0'과 '1'을 붙인 '00'과 '01'을 큐에 넣는다.

다음에는 큐에서 프리픽스 '1'을 꺼내와 질의하게 된다. 여기에 맞는 태그는 '100' 1개 이므로 이것만 응답하게 되고 리더기에게 정상적으로 인식된다. 이렇게 해서 1라운드가 끝나고 전에 큐에 넣은 '00'과 '01'을 대상으로 하여 다시 라운드를 시작하게 된다.

다음 2라운드에서 리더기는 '00'과 '01'을 프리픽스로 해서 질의하고 '00'으로 시작하는 태그는 없으므로 무응답이 된다. 후에 '01'을 질의하고 충돌이 나서 '010'과 '011'을 큐에 넣어 진행한다. 이런 방식으로 진행해 나가다 큐에 남은 프리픽스가 없을 때 태그의 ID 트리가 완성된 것으로 보고 종료한다.[4]

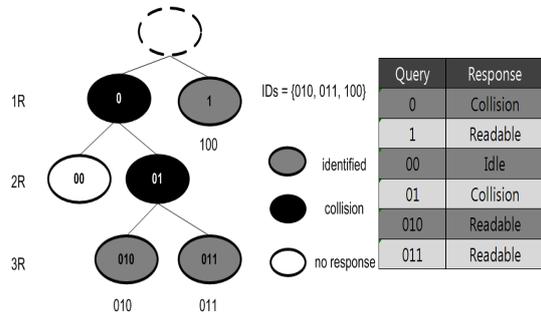


그림 1. QT 프로토콜 동작  
Fig. 1 QT protocol example

반면 QTR(Query Tree Algorithm with Reversed Tag IDs) 알고리즘은 태그 ID의 특성상 생기는 QT의 단점을 해결한다. 즉, 태그 ID의 앞부분에는 생산 국가나 지역 등의 정보가 있어 중복되는 경우가 많고, 뒷부분에는 그 물건의 고유 번호가 있어서 중복되는 경우가 거의 없다. 그래서 QTR은 QT에서 태그 ID의 앞부터 뒤로 순차적으로 인식하는 것과는 반대로 뒤에서 앞으로 순차적으로 인식을 한다.[4]

CT(Collision Tracking : 충돌 추적 알고리즘) 알고리즘은 최근에 제안된 태그 충돌방지 알고리즘으로 리더기가 n-비트 길이의 프리픽스 Pn 를 인자로 리더기의 영역 내의 태그에게 질의를 하게 되면, 태그들은 수신한 프리픽스를 자신의 식별자와 비교한다. 비교 후 매칭이 이루어질 경우 그 태그의 식별자의 n+1 비트부터 마지막 비트까지 리더기에게 순서대로 전송한다. 리더기는 태그가 전송한 식별자 정보를 수신함과 동시에 수신된 비트의 충돌 발생 여부를 조사하게 된다. 만약 '0'과 '1'이 동시에 수신되어 리더기에서 충돌이 발생하면 나머지 비트들의 송신을 중지하도록 태그로 식별자 중지 명령을 보낸다. 이 때 단순히 프리픽스에 '0'이나 '1'을 추가하여 다시 태그들에게 전송하는 쿼리 트리 알고리즘과는 달리 식별자 중지 명령을 받은 태그는 그 즉시 식별자 비트 전송을 중지하고 새로운 프리픽스 Pn\* 을 받을 준비를 한다. 리더기는 충돌이 없이 정상적으로 수신된 비트들을 프리픽스 Pn 에 추가하여 새로운 프리픽스 Pn\* 을 생성하여 다시 태그에게 전송한다. 그리고 충돌 없이 '0'이나 '1'이 계속 수신되어 마지막 비트까지 정상적으로 수신을 마치면 한 개의 태그가 식별되게 된다.[5]

CT에서 개선된 RCT(Revised Collision Tracking)방법에서는 첫 비트 충돌 추적을 사용함으로써 리더기에서 충돌이 발생하면 나머지 비트들의 송신을 중지하는 CT 방식과는 달리 모든 비트 충돌 추적을 사용함으로써 프리픽스와 태그에서 생성한 난수를 끝까지 비교해서 충돌이 난 비트들의 정보를 사용하여 리더기에서 질의수를 줄인다. 만약 충돌이 난 비트가 있으면 충돌이 난 위치에 ‘0’과 ‘1’을 대입해서 바로 인식하고 메모리에 저장함으로써 질의수를 줄인다.[5]

### III. 제안 알고리즘

본 논문에서 제안하는 충돌 회피 알고리즘은 앞에서 설명한 QT 방식과, DFSA1 방식 및 모든 전송 비트의 충돌 추적 알고리즘을 혼합한 하이브리드 방식이다. 제안하는 방법의 진행과정은 그림 2와 같다.

제안 알고리즘의 시작은 QT 기반 알고리즘과 같지만, 태그 ID에서 충돌이 났을 때 앞에서 설명한 모든 비트 충돌추적 알고리즘을 이용하여 몇 개의 비트에서 충돌이 났는지 충돌 비트수(ColbitNum)를 알 수 있다. QT 알고리즘의 매 회 질의에서 전송된 태그들의 ID중 충돌이 난 비트 수를 고려하여 이 값이 90보다 작은 경우에는 태그의 개수가 작다고 판단하여 DFSA 방식으로 전환하고, 충돌 비트 수가 90보다 큰 경우에는 QT 알고리즘 방식을 계속 수행한다.

충 태그의 개수가 N개 이고 태그 ID의 비트 수가 96개 인 경우, k개의 비트가 충돌이 날 확률은 식 (2)로 나타낼 수 있다.

$$\binom{96}{k} \cdot \left(1 - \left(\frac{1}{2}\right)^N \cdot 2\right)^k \cdot \left(\left(\frac{1}{2}\right)^N \cdot 2\right)^{96-k} \quad (2)$$

식 (2)는 96개의 비트 중에서 k개의 비트를 선택하는 경우의 수, k개의 비트에서는 충돌이 발생해야 하므로 모두 ‘0’인 경우와 모두 ‘1’인 경우를 제외한 확률, (96-k)개의 비트는 충돌이 발생하지 않기 때문에 모두 ‘0’ 또는 ‘1’이 되는 확률의 곱으로 구할 수 있다.

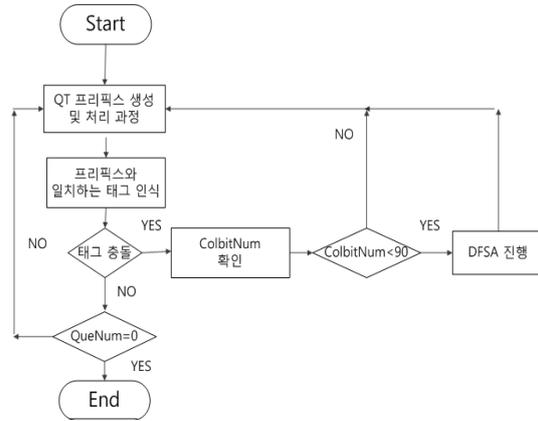


그림 2. 제안하는 Hybrid 충돌 회피 방안  
Fig. 2 Proposed hybrid anti-collision method

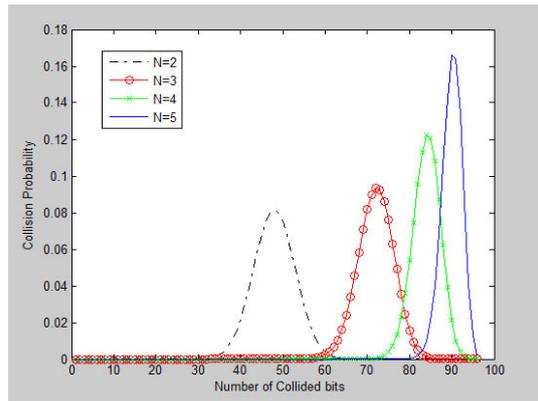


그림 3. 태그 개수 N에 따른 충돌 비트수 분포도  
Fig. 3 collision probability for number of collided bits with N Tags

그림 3은 전체 태그의 개수 N이 주어졌을 때, 태그의 전체 ID에서 충돌이 발생한 비트 개수별 발생 확률을 나타낸다. 전체 태그 개수 N이 2개일 때는 96개의 태그 ID 비트 중에서 50개 정도 비트의 충돌이 발생할 확률이 가장 크며, N이 3개 일 때는 70개의 비트가 충돌 날 확률이 가장 크다. N이 4개와 5개 일 때는 각각 80개의 비트와 90개의 비트 정도가 충돌 날 확률이 가장 높게 된다. 그리고 6개 이상일 때는 거의 전 비트가 충돌이 나게 된다. 제안하는 방식에서는 그림 3의 결과를 바탕으로 충돌 비트 수가 90개 미만일 때는 DFSA 방식으로 처리를 한다.

DFSA의 과정은 그림 4와 같다. 전체 태그 ID 중에서 충돌이 발생한 비트수가 50개 이하일 때는 슬롯 개수를 2개, 51개에서 70개 사이일 때는 슬롯 개수를 3개, 71개에서 80개 사이일 때는 슬롯 개수를 4개, 81개에서 89개 사이일 때는 슬롯 개수를 5개로 설정한 후, DFSA 알고리즘의 태그 인식을 시작한다. 그 이후에는 충돌이 난 슬롯 개수에 2.3922를 곱해서 그 수보다 작거나 같은 최대 정수를 예상 태그 개수로 정하는 DFSA1 방식을 사용하였다. QT 알고리즘의 프리픽스와 일치하는 모든 태그들을 인식한 후 새로운 프리픽스를 사용하여 QT과정을 재시작한다. 추가적으로 DFSA의 과정에서 성능을 높이기 위하여 슬롯 중에서 태그들이 전송을 하지 않는 빈 슬롯일 때는 슬롯의 시간을 일찍 끝내는 Slotted Aloha with Early End[3]방식을 적용하였다.

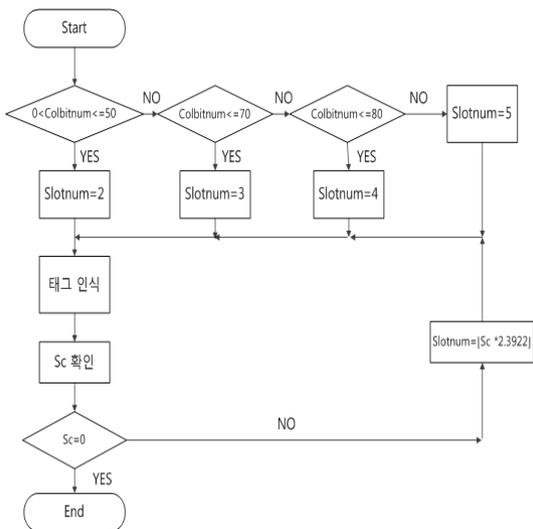


그림 4. 제안방안에서 사용한 DFSA 방식  
Fig. 4 DFSA scheme in proposed method

#### IV. 성능 평가

본 논문에서 제안하는 hybrid 방식의 충돌 회피 알고리즘은 시뮬레이션을 통해서 QT 알고리즘과 성능을 비교하였다. 시뮬레이션에서는 리더기가 1개만 존재하고, 모든 태그들은 리더기와 통신할 수 있는 범위에 있으며, 외부에서의 전송과 같은 간섭은 발생하지 않는다고

가정하였다. 질의를 통해 태그 ID를 인식하는 QT 알고리즘의 특성상 성능의 지표로 시간이 아닌 질의 횟수로 고려했으며, 전체 태그 개수는 10개부터 100개까지 10개씩 증가시켰다. DFSA 알고리즘을 적용할 때는 하나의 슬롯을 한 번의 질의로 정했으며, 빈 슬롯은 Slotted Aloha with Early End[3] 방식을 적용해서 0.5로 정했다. 그림 5에서 보듯이 제안한 방식은 QT 알고리즘에 비해서 태그의 개수가 증가할수록 질의 횟수가 더 작아지는 것을 알 수 있다.

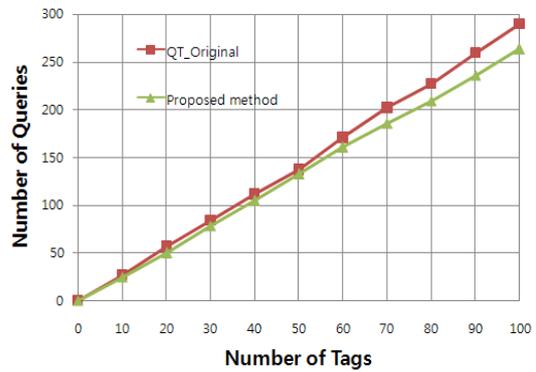


그림 5. 태그 개수에 따른 평균 질의 횟수  
Fig. 5 Average number of queries vs. number of tags

그림 6은 DFSA1과 QT 알고리즘, 그리고 제안 알고리즘의 효율성을 비교한 한 것이다. 효율성은 식 (3)과 같다.

$$(성공\ 질의\ 개수 / 전체\ 질의\ 개수) * 100 \quad (3)$$

DFSA1 알고리즘의 경우 초기 슬롯의 개수에 따라 효율성 결과가 달라지기 때문에 초기 슬롯의 개수가 3개, 5개, 10개일 때의 값을 구했다. 빈 슬롯을 0.5번의 질의로, 기타 슬롯을 1번의 질의로 정해서 측정 하였다. 제안하는 알고리즘의 효율성이 QT 알고리즘의 효율성 보다 2~3%정도 더 좋은 것을 알 수 있다. DFSA1 방식은 초기 슬롯의 개수와 태그의 개수가 비슷한 경우 좋은 성능을 보여주나, 초기 슬롯의 개수와 전체 태그 개수의 차이가 많이 날수록 효율이 떨어져 제안하는 알고리즘보다 효율성이 더 나빠진다. DFSA 알고리즘을 사용하는 RFID 시스템에서 초기 태그의 개수를 정확히 예측하는 방법

이 없기 때문에, 제안한 방법은 DFSA 방식에 비해 안정적인 성능을 보여주는 장점이 있다.

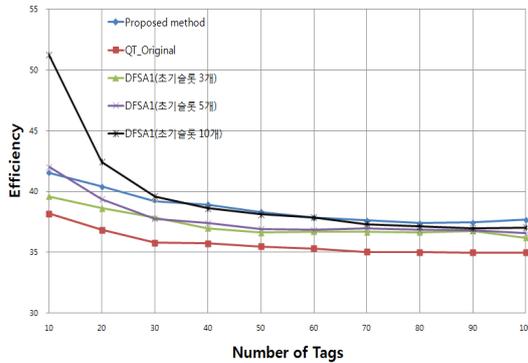


그림 6. 알고리즘의 효율성  
Fig. 6. Efficiency of Algorithms

### V. 결 론

RFID 시스템은 무선을 통한 ID 관리의 편리함으로 인해 세계적으로 시장의 규모가 점점 커지고 있으며, 효율적으로 RFID 태그들의 ID를 인식하기 위한 핵심 알고리즘에 대한 연구가 활발하다. 본 논문에서는 QT 방식과 DFSA 방식을 혼합한 hybrid 방식의 충돌 회피 알고리즘을 제안하였으며, 시뮬레이션을 통해서 기존의 QT 알고리즘 보다 질의 횟수나 효율성면에서 더 좋은 성능을 보여주는 것을 알 수 있었다. 또한 DFSA 알고리즘의 효율성이 초기에 설정된 슬롯 개수에 따라서 달라지는 것과 다르게 제안한 알고리즘은 태그 개수에 따라 비교적 일정한 성능을 보여준다.

### 참고문헌

[1] 차재룡, 김재현, “RFID 시스템에서의 태그 수를 추정하는 ALOHA 방식 Anti-collision 알고리즘”, 한국통신학회논문지, 제30권, 9A호, pp.814- 821, 2005.  
[2] Harald Vogt, “Efficient Object Identification with Passive RFID Tags,” In *International Conference on Pervasive Computing*, pp.98- 113, April, 2002.

[3] Dheeraj K. Klair, Kwan-Wu Chin and Raad Raad, “A Survey and Tutorial of RFID Anti-Collision Protocols,” *IEEE Communications and Surveys & Tutorials*, Vol. 12, No. 3, 2010.  
[4] 정승민, 조정식, 김성권 “FQTR : RFID 시스템을 위한 새로운 하이브리드 태그 충돌 방지 프로토콜”, 정보과학회논문지 : 소프트웨어 및 응용, 제36권, 제7호, pp. 560-570, 2009.  
[5] 최승진, 신재동, 김성권, “RFID 시스템에서 개선된 충돌 추적 방식을 이용한 태그 인식 알고리즘”, 한국정보과학회 한국컴퓨터종합학술대회 논문집, 제34권 제1호(D), pp.473-476, 2007.

저자소개



신성용(Song-Yong Shin)

2010년~ 현재 한밭대학교  
정보통신전문대학원  
컴퓨터공학과 석사과정

※관심분야: RFID 시스템, 이동통신 프로토콜



황경호 (Gyung-Ho Hwang)

1998년 KAIST 전기및전자공학과  
공학사  
2000년 KAIST 전자전산학과  
공학석사

2005년 KAIST 전자전산학과공학박사  
2005년 ~ 2007년 삼성전자 무선사업부 책임 연구원  
2007년 ~ 현재 한밭대학교 컴퓨터공학과 조교수  
※관심분야: 이동통신 MAC 프로토콜, 무선 mesh 네트워크, RFID 시스템