

# 수동형/반능동형 RFID 시스템의 태그 충돌 방지 알고리즘

## Part I : QueryAdjust 명령어를 이용한 AFQ 알고리즘과 Grouping에 의한 성능개선

준회원 송인찬\*, 범효\*, 종신회원 장경희\*, 정회원 신동범\*\*, 이형섭\*\*

### Tag Anti-Collision Algorithms in Passive and Semi-passive RFID Systems

#### Part I : Adjustable Framed Q Algorithm and Grouping Method by using QueryAdjust Command

InChan Song\*, Xiao Fan\* *Associate Members*, KyungHi Chang\* *Lifelong Member*,  
Dong-Beom Shin\*\*, Heyung-Sub Lee\*\* *Regular Members*

#### 요약

본 논문에서는 EPCglobal Class-1 Generation-2 (Gen2) 기반 Probabilistic Slotted 충돌방지 알고리즘에 대하여 살펴보고, 태그인식시간, 충돌 비율을 감소시키고, 데이터 처리량, 시스템 효율을 증가 시킬 수 있는 QueryAdjust 명령어를 사용한 FAFQ (fixed adjustable framed Q) 알고리즘과 AAFQ (adaptive adjustable framed Q) 알고리즘을 제안하며, 또한 Gen2 기반으로 태그 인식 효율을 향상 시킬 수 있는 Grouping 방법을 제안한다. 제안한 방법들 모두 Q 알고리즘의 성능 향상을 보이며, 제안하는 방법 중 AAFQ 알고리즘이 가장 높은 성능 향상을 나타낸다. 즉, AAFQ 알고리즘에 의하여 5% 정도의 시스템 효율 성능 향상과 4.5% 정도의 충돌 비율 감소를 얻을 수 있다. Grouping 방법은 FAFQ 알고리즘과 AAFQ 알고리즘에 대해선 Ungrouping 방법과 비슷한 성능을 보이지만, Gen2 Q 알고리즘의 경우 Ungrouping 방법과 비교 하였을 때 태그인식시간 및 충돌 비율을 감소시키고, 데이터 처리량 및 시스템 효율을 증가 시킨다.

**Key Words** : RFID, Anti-collision algorithm, Gen2 protocol, Passive, Grouping

#### ABSTRACT

In this paper, we analyze the performance of probabilistic slotted anti-collision algorithm used in EPCglobal Class-1 Generation-2 (Gen2). To increase throughput and system efficiency, and to decrease tag identification time and collision ratio, we propose new tag anti-collision algorithms, which are FAFQ (fixed adjustable framed Q) algorithm and AAFQ (adaptive adjustable framed Q) algorithm, by using QueryAdjust command. We also propose grouping method based on Gen2 to improve the efficiency of tag identification. The simulation results show that all the proposed algorithms outperform Q algorithm, and AAFQ algorithm performs the best. That is, AAFQ has an increment of 5% of system efficiency and a decrement of 4.5% of collision ratio. For FAFQ and AAFQ algorithm, the performance of grouping method is similar to that of ungrouping method. However, for Q algorithm in Gen2, grouping method can increase throughput and system efficiency, and decrease tag identification time and collision ratio compared with ungrouping method.

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 IT 신성장동력핵심기술개발사업의 일환으로 수행하였음. [2005-S-106-03,RFID/USN 용 센서 태그 및 센서 노드 기술]

\* 인하대학교 정보통신대학원 이동통신연구실 (ny10023@hanmail.net, fxismonk@gmail.com, khchang@inha.ac.kr)

\*\* 한국전자통신연구원 (sdb@etri.re.kr, leehs@etri.re.kr)

논문번호 : KICS2008-02-059, 접수일자 : 2008년 2월 1일, 최종논문접수일자 : 2008년 8월 2일

## I. 서 론

RFID (Radio Frequency Identification)란 사물에 부착된 전자태그로부터 무선 주파수를 이용하여 정보를 송수신하고 이와 관련된 서비스를 제공하는 기술을 일컫는다<sup>[1]</sup>.

RFID 시스템에서는 태그에 극소형 칩을 부착해 많은 정보를 저장할 수 있으며, 상품에 태그의 부착이 용이하고 바코드에 비해 장거리 정보 송수신이 가능하다. 특히 유통 및 물류 분야에 사용될 900 MHz RFID 시스템은 각종 물품에 전자태그를 부착해 자동으로 물품 명세와 가격, 유통경로 및 기한 등을 파악할 수 있어 유통 및 물류에 혁신을 가져올 기술로 각광받고 있다. 또한 앞으로 센서 기술과 칩 소형화 기술 등이 더욱 발전하면 정보를 능동적으로 획득하고 처리하는 능력까지 갖추게 되어 바코드가 하던 역할과는 비교할 수 없을 정도로 많은 일들을 해낼 수 있을 것으로 기대되고 있다<sup>[2]</sup>.

일반적으로 RFID 시스템에서 리더는 무선 채널을 통하여 각각의 태그들과 통신을 하는데, 모든 태그들이 리더가 보낸 신호를 동시에 받게 되고 리더의 전송요구에 응답을 한다. 이 때 하나의 리더가 동시에 응답한 여러 개의 태그를 인식해야 하는 문제가 발생하는데 이를 태그 충돌이라고 한다<sup>[1]</sup>. 여러 개의 태그를 충돌 없이 인식하기 위해서는 이러한 태그 충돌 문제를 해결하는 충돌방지 알고리즘이 필요하며, 이는 900 MHz RFID 시스템에서 가장 핵심 기술로서 인식되고 있다<sup>[3]</sup>.

본 논문에서는 900 MHz RFID 시스템에서 가장 핵심 기술로서 인식되고 있는 EPCglobal Class1 Gen2기반 Probabilistic Slotted 충돌방지 알고리즘에 대하여 살펴보았고, Gen2 기반으로 QueryAdjust 명령어를 사용한 효율적인 충돌방지 알고리즘을 제안하였다. 또한, 성능평가를 위한 성능 인덱스를 새롭게 정의하였고, 모의실험을 통하여 기존의 알고리즘과 제안한 알고리즘의 성능을 비교 및 분석의 결과를 제시하였다. 본 논문의 구성은 다음과 같다. 우선 II장에서는 기존의 Gen2기반의 Probabilistic Slotted 충돌방지 알고리즘에 대해 살펴보았고, III장에서는 Gen2 기반의 QueryAdjust 명령어를 사용한 효율적인 충돌방지 알고리즘을 제안하였으며, IV장에서는 기존의 알고리즘과 III장에서 소개된 제안된 충돌방지 알고리즘의 모의실험을 통하여 성능 비교 및 분석 하였으며, V장에서는 결론을 맺는다.

## II. Gen2 기반의 충돌방지 알고리즘

900 MHz RFID 시스템에서 가장 핵심 기술로서 인식되고 있는 EPCglobal Class1 Gen2기반 Probabilistic Slotted 알고리즘의 기본 동작 과정은 다음과 같다.

슬롯의 송수신과정은 라운드의 시작 명령인 질의 (Query) 명령이나 재질의 (QueryRep) 명령으로 시작하며 이에 대한 응답으로 태그는 16-비트의 랜덤한 값인 RN16을 리더로 송신한다. 성공적으로 RN16을 수신한 경우, 리더는 이를 인자로 응답 (ACK) 명령을 태그로 전송하며 이를 수신한 태그 중 해당 RN16을 송신한 태그만 자신의 태그 식별자 (EPC)를 리더로 송신한다. CRC에러검사를 거쳐 성공적으로 수신하였다고 판단될 경우, 한 개의 태그가 식별된 것이다. 다음 슬롯의 송수신을 위하여 리더는 재질의 명령을 다시 태그로 전송하며 태그는 RN16으로 응답하며 상기 과정을 반복한다<sup>[4]</sup>.

그림 1은 Probabilistic Slotted 알고리즘의 기본 동작 과정을 나타내고 있다.

상기 과정 중 RN16의 성공적인 수신은 오직 하나의 태그만 응답하였을 경우이며, 응답이 없거나 다수 개의 태그 응답으로 인해 충돌이 발생한 경우는 실패로 여겨진다. 그림 2는 RN16의 성공적인 수신이 오직 하나의 태그만 응답하였을 경우이며, 그림 3은 하나의 리더에 다수 개의 태그 응답으로 인해 충돌이 발생한 경우와 태그로부터 응답이 없는 경우이다.

라운드 시작 시에 설정된 Q값에 따라 전송해야 할 슬롯개수가 결정되며 이를 모두 송수신한 경우 라운드가 종료된다.

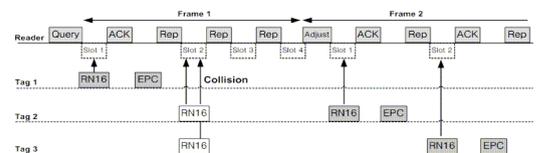


그림 1. Probabilistic Slotted 알고리즘의 기본 동작과정<sup>[5]</sup>

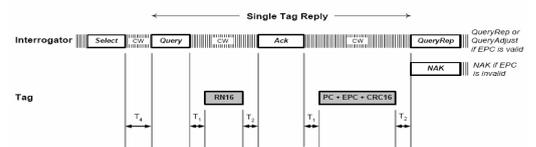


그림 2. 하나의 태그만 리더에 응답한 경우<sup>[6]</sup>

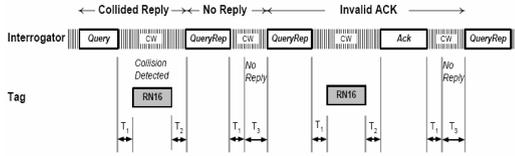


그림 3. 태그로부터 응답이 없거나 태그 충돌이 발생한 경우<sup>[6]</sup>

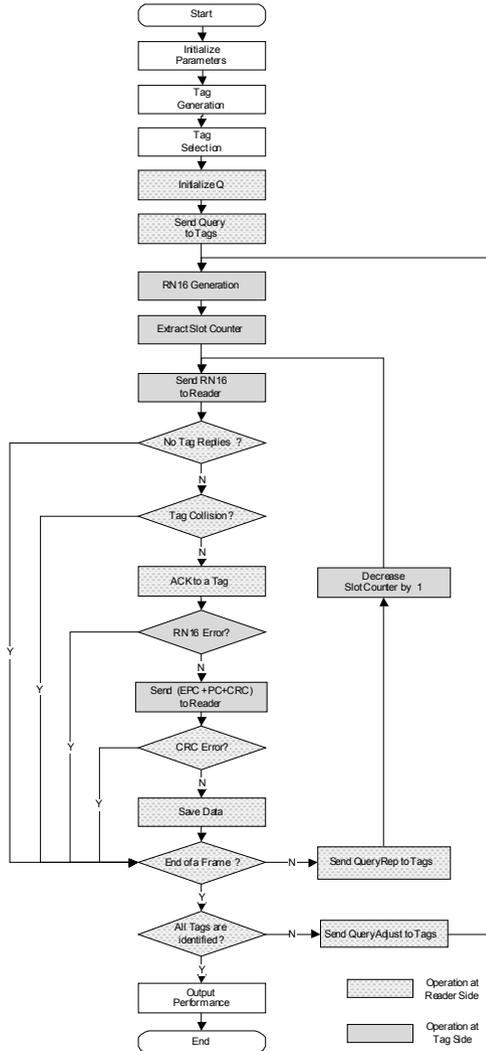


그림 4. Probabilistic Slotted 알고리즘의 흐름도

그림 4는 Probabilistic Slotted 알고리즘을 구현하기 위한 알고리즘 흐름도이다.

그림 5는 Probabilistic slotted 알고리즘을 구현하기 위한 의사코드이다.

```

Reader sends Query.
for inventory procedure
Every tag generates RN16 & slot counter.
for current frame
    If slot counter == 0
        Tag replies its RN16.
    end
    If a single tag replies
        Reader sends ACK with RN16.
        If RN16 received by tag
            == RN16 saved in tag
            Tag sends EPC to reader.
        end
        Reader sends QueryRep.
    else if multiple tags reply
        Reader sends QueryRep or QueryAdjust.
    else if no tag replies
        Reader sends QueryRep or QueryAdjust.
    end
    If tag receives QueryRep
        slot counter = slot counter - 1
    end
end
Reader sends QueryAdjust.
end
    
```

그림 5. Probabilistic slotted 알고리즘의 의사코드

### 2.1 충돌 방지 Q 알고리즘

EPCglobal Class1 Gen2에서 제시하는 충돌방지 알고리즘은 Slotted ALOHA 프로토콜을 기반으로 하는 Q 알고리즘이다.

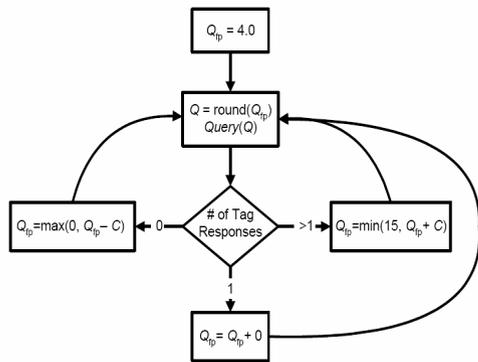


그림 6. 충돌방지를 위한 Q 알고리즘<sup>[6]</sup>

그림 6은 EPCglobal Class1 Gen2에서 제시하는 Q 알고리즘이다.

Q 알고리즘은 Q 파라미터를 이용해서 시간을 2<sup>Q</sup> 개의 슬롯이란 단위로 나누고 각각의 시간 슬롯에

태그들이 랜덤하게 응답하게 함으로써 태그의 충돌을 줄이는 방법이다. 따라서 Q에 따라 충돌방지 알고리즘의 성능이 결정된다. Q는 0~15까지 정수로 이에 따라 생성될 수 있는 슬롯의 개수는  $2^0 \sim 2^{15}$ 개이다. Q값은  $Q_{fp}$ 값에 따라 변하게 되는데  $Q_{fp}$ 는 유리수 값을 가지는 파라미터이며, Q값은 정수 값을 가지는 파라미터이다. Q 알고리즘은  $Q_{fp}$ 의 초기 값을 4로 두고 태그 인식 과정을 시작한다.

리더의 질의(Query)에 대해  $2^4$ 개의 슬롯에서 태그들이 랜덤하게 응답할 경우, 다음과 같은 3가지 경우가 발생한다<sup>7)</sup>.

Case 1. If a single tag replies,

$$Q_{fp} = Q_{fp}$$

Case 2. If no tags reply,

$$Q_{fp} = Q_{fp} - C, \text{ due to the empty slot.}$$

Case 3. If multiple tags reply,

$$Q_{fp} = Q_{fp} + C, \text{ due to the collided slot.}$$

여기서, C는  $Q_{fp}$ 에 더해지거나 빼지는 값으로  $0.1 < C < 0.5$ 까지의 범위를 가진다<sup>6)</sup>.

## 2.2 Q 알고리즘의 성능 인덱스

본 논문에서는 Gen2 알고리즘의 성능을 검증하기 위해 다음과 같은 네 개의 성능 인덱스를 사용한다.

### 2.2.1 태그인식시간 (Identification Time)

기존의 연구에선 태그인식시간을 모든 태그들이 리더에 인식될 때까지 시간으로 정의하였다.

$$N_{Total} = \sum_i L_i \quad (1)$$

여기서,  $L_i$ 는  $i$ 번째 라운드의 슬롯 개수를 의미한다.

식 (1)은 태그를 인식하기 위해 소요된 슬롯 개수를 의미한다. 상기에 식(1)에 따라 태그인식시간은 슬롯 전송에 소요된 시간과 관계없이 모든 슬롯 전송에 동일한 시간을 부여하여 계산하였다. 하지만, Gen2 알고리즘의 경우 그림 2와 그림 3에서 확인할 수 있는 것처럼 식별 슬롯, 빈 슬롯, 충돌발생 슬롯에 소요된 전송시간이 고정되어 있는 것이 아니라 리더와 태그 통신 중 명령어 전송 횟수,  $T_{ari}$  (데이터 전송시간)에 따라 슬롯에 소요된 전송시간은 서로 다르다. 따라서 본 논문에서는 태그인식시간을 식 (2)와 같이 정의한다.

### Identification Time

$$= T_{Select} + T_4 + T_{Query} + T_1 + T_{RN16} + T_2 + T_{ACK} + T_1 + T_{EPC} + T_2 + \dots \quad (2)$$

### 2.2.2 데이터 처리량 (Throughput)

본 논문에서는 Gen2의 데이터 처리량을 다음 식 (3)과 같이 정의 한다.

$$\begin{aligned} \text{Throughput} &= \frac{\text{Total Transmission Bits}}{\text{Identification Time}} \text{ (bps)} \\ &= \frac{\sum_i (\text{Transmission Bits in the } i^{\text{th}} \text{ Successful Slot})}{\text{Identification Time}} \\ &= \frac{\text{Data Rate} \times \sum_i (\text{ } i^{\text{th}} \text{ Successful Slot Time})}{\text{Identification Time}} \\ &= \text{Data Rate} \times \frac{\text{Total Successful Slot Time (only considering Traffic)}}{\text{Identification Time}} \end{aligned} \quad (3)$$

### 2.2.3 시스템 효율 (System Efficiency)

본 논문에서는 기존의 연구들에서 제시된 시스템 효율에 대해 살펴보고, 슬롯전송에 소요된 시간 개념을 부여한 새로운 시스템 효율을 정의한다.

기존 대부분의 연구에선 식 (1)을 이용하여 식 (4)와 같이 나타내었다.

$$\text{System Efficiency} = \frac{\sum_i c_{1,i}}{N_{Total}} \quad (4)$$

여기서,  $c_{1,i}$ 은  $i$ 번째 라운드의 식별 슬롯개수이다.

기존 논문에서는 식 (1), (4)을 이용하여 시스템 효율을 다음과 같이 정의 하였다.

$$\text{System Efficiency} = \frac{\text{Number of Successful Slots}}{\text{Total Number of Slots}} \quad (5)$$

식 (5)는 슬롯 전송에 소요된 시간과 관계없이 식별슬롯의 개수만 판별하여 시스템 효율을 정의 하였다. 하지만 앞서 언급했던 바와 같이 Gen2 알고리즘에선 식별 슬롯, 빈 슬롯, 충돌발생 슬롯에 소요된 전송시간이 고정되어 있는 것이 아니라 서로 다르기 때문에 식별슬롯의 개수를 판별 하는 것 보다 식별 슬롯에 소요된 전송시간을 판별 하는 것이 더 바람직하다<sup>4)</sup>. 따라서 본 논문에서는 식 (2)와 (5)를 이용하여 시스템 효율을 다음 식 (6)과 같이 정의한다.

$$\text{System Efficiency} = \frac{\text{Total Successful Slot Time}}{\text{Identification Time}} \quad (6)$$

2.2.4 충돌 비율 (Collision Ratio)

충돌 발생 비, 충돌 비율을 식 (1), (4)을 이용하여 식 (7)과 같이 나타낼 수 있다.

$$C_{ratio} = \frac{\sum c_{k,i}}{N_{Total}} \quad (7)$$

여기서,  $c_{k,i}$ 은  $i$ 번째 라운드 의 충돌 슬롯개수이다. 따라서 본 논문에서는 충돌 비율을 식 (8)과 같이 정의한다.

$$C_{ratio} = \frac{\text{Total Number of Collided Slots}}{\text{Total Number of Slots}} \quad (8)$$

2.3 Q 알고리즘의 이론적 성능 분석

Probabilistic slotted 알고리즘의 Q 알고리즘은 DFSA (Dynamic Framed Slotted ALOHA) 알고리즘을 기반으로 하고 있다. 따라서  $n$ 개의 태그가  $N$ 개의 슬롯을 사용하여 리더와 통신할 경우 하나의 슬롯 내에  $r$ 개의 태그가 존재할 확률은 이항분포 (binomial distribution)를 따르며 다음의 식 (9)로 나타낼 수 있다<sup>8)</sup>.

$$B_{n, \frac{1}{N}}(r) = \binom{n}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{n-r} \quad (9)$$

하나의 슬롯 내에 태그가 존재하지 않을 빈 슬롯개수의 평균 기댓값,  $a_0^{N,n}$ 은 다음과 같이 식 (10)로 나타낼 수 있다.

$$a_0^{N,n} = NB_{n, \frac{1}{N}}(0) = N \binom{n}{0} \left(\frac{1}{N}\right)^0 \left(1 - \frac{1}{N}\right)^n = N \left(1 - \frac{1}{N}\right)^n \quad (10)$$

하나의 슬롯 내에 하나의 태그만 존재 할 식별 슬롯개수의 평균 기댓값,  $a_1^{N,n}$ 는 다음과 같이 식 (11)로 나타낼 수 있다.

$$a_1^{N,n} = NB_{n, \frac{1}{N}}(1) = N \binom{n}{1} \left(\frac{1}{N}\right)^1 \left(1 - \frac{1}{N}\right)^{n-1} = n \left(1 - \frac{1}{N}\right)^{n-1} \quad (11)$$

식 (10), (11)로부터 충돌슬롯개수의 평균 기댓값,  $a_k^{N,n}$ 을 식 (12)과 같이 나타낼 수 있다.

$$a_k^{N,n} = NB_{n, \frac{1}{N}}(k) = N - a_0^{N,n} - a_1^{N,n}, \quad k \geq 2. \quad (12)$$

Gen2 알고리즘의  $i$ 번째 라운드 슬롯개수,  $L_i$ 는 Q 알고리즘의 QueryAdjust 명령어에 의해  $2 \cdot L_i, L_i/2, L_{i-1}$ 과 같이 증가, 감소, 또는 유지된다.

$$L_i = \begin{cases} 2 \cdot L_{i-1} & \text{if } I \geq 1 \\ L_{i-1} & \text{if } I = 0 \\ L_{i-1} / 2 & \text{if } I \leq -1 \end{cases} \quad (13)$$

라운드 슬롯개수 증가 감소, 유지를 결정하기 위한 변수  $I$ 는 충돌슬롯개수의 평균 기댓값과 빈 슬롯개수의 평균 기댓값, 식 (10)과 식 (12)을 이용해 다음 식 (14)와 같이 나타낼 수 있다.

$$I = \text{round}[(a_k^{N,n} - a_0^{N,n}) \times C]. \quad (14)$$

식 (13)와 식 (14)을 이용하여 식 (1)의 시스템에서 사용되는 전체 슬롯개수의 평균 기댓값을 구할 수 있다.

상기의 식 (14)에 실제 슬롯 상태정보인  $c_{k,i}$ 와  $c_{0,i}$ 를 대입하여 시스템에서 사용되는 변수  $I$ 의 실측값을 구할 수 있다.

$$I = \text{round}[(c_{k,i} - c_{0,i}) \times C]. \quad (15)$$

또한, 식 (13)와 식 (15)을 이용하여 식 (1)의 시스템에서 사용되는 전체 슬롯개수의 실측값을 구할 수 있다.

본 논문에서는 기댓값 식 (14)와 실측값 식 (15)을 통해 Gen2 알고리즘에서 사용된 전체 슬롯개수의 기댓값과 실측값을 비교해본다.

그림 7은 Gen2 알고리즘에서 사용된 전체 슬롯개수의 기댓값과 실측값을 비교한 그림으로 태그개수가 증가함에 따라 사용된 슬롯개수가 증가함을 볼 수 있다.

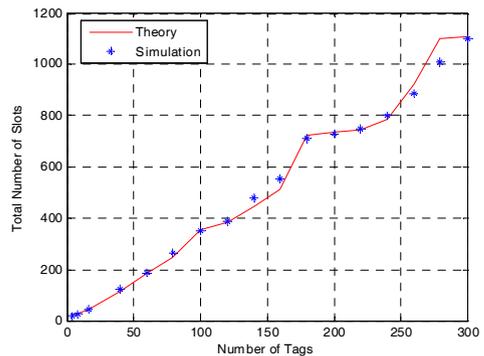


그림 7. Gen2 알고리즘에서 사용된 전체 슬롯개수(기댓값 vs 실측값)

식 (4)를 이용해 시스템 효율의 실측값과 식 (16)의 시스템 효율의 기댓값을 비교 할 수 있다.

$$\text{System Efficiency} = \frac{\sum_i a_{i,i}}{N_{\text{Total}}} \quad (16)$$

여기서,  $a_{i,i}$ 는 식 (11)로부터 구할 수 있다.

그림 8은 시스템 효율의 기댓값과 실측값을 비교한 그림이다.

또한, 식 (7)을 이용해 충돌 비율의 실측값과 식 (17)의 충돌 비율의 기댓값을 비교 할 수 있다.

$$C_{\text{ratio}} = \frac{\sum_i a_{k,i}}{N_{\text{Total}}} \quad (17)$$

여기서,  $a_{k,i}$ 는 식 (12)으로부터 구할 수 있다.

그림 9는 충돌 비율의 기댓값과 실측값을 비교한 그림이다.

그림 7~9로부터 수학적 근거로 계산된 기댓값과 실제 슬롯 상태정보로 계산된 실측값은 상당히 일치함을 볼 수 있다.

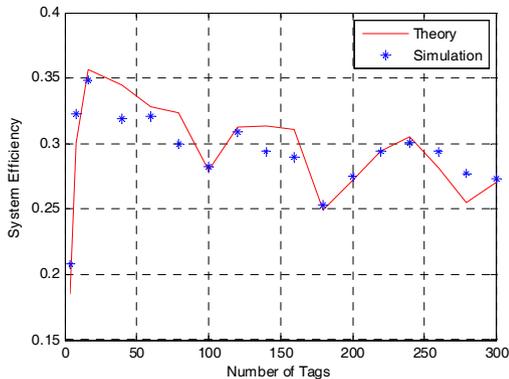


그림 8. Gen2 알고리즘의 시스템 효율 (기댓값 vs 실측값)

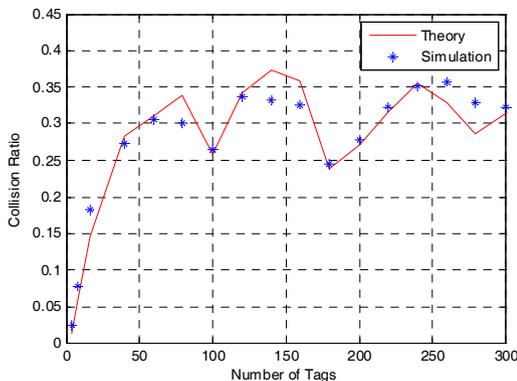


그림 9. Gen2 알고리즘의 충돌 비율 (기댓값 vs 실측값)

### III. 제안된 QueryAdjust 기반의 Gen2 충돌방지 알고리즘

기존의 Gen2의 Q알고리즘의 경우 QueryAdjust 명령어를 라운드 종료 시 슬롯 개수를 조절하기 위해 사용한다. 본 논문에서는 슬롯 개수를 조절하는 QueryAdjust 명령어를 이용하여 태그인식시간, 데이터 처리량, 시스템 효율, 충돌 비율을 개선 할 수 있는 새로운 충돌방지 알고리즘을 제시한다.

#### 3.1 AFQ 알고리즘

AFQ (Adjustable Framed Q) 알고리즘은 현재의 라운드 종료 전 연속적인 충돌 슬롯이 발생하거나 연속적인 빈 슬롯이 발생 할 경우, 현재 라운드 내에서 QueryAdjust 명령어를 사용하여 라운드를 종료한 후, 다음 라운드의 슬롯 개수를 조절하는 방법이다. 연속적인 충돌 슬롯이 발생한다는 것은 현재 시스템 내에 슬롯개수보다 태그개수가 많다는 것을 의미한다. 따라서 슬롯개수를 증가시켜 새로운 라운드 시작함으로써 시스템 효율을 증가 시켜야 한다.

그림 10은 제안하는 QueryAdjust 기반의 Gen2 충돌방지 알고리즘 흐름도이다.

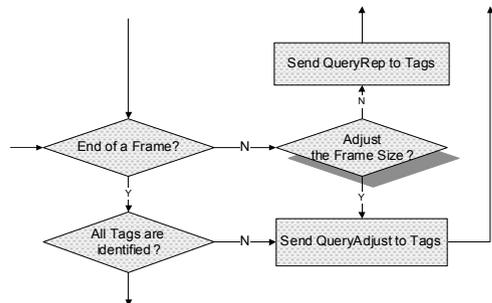


그림 10. 제안하는 QueryAdjust 기반의 Gen2 충돌방지 알고리즘 흐름도

#### 3.1.1 FAFQ 알고리즘

FAFQ (Fixed Adjustable Framed Q) 알고리즘은 현재 라운드 내에서 연속적인 충돌 슬롯이 threshold 값 이상 발생 할 경우 QueryAdjust 명령어를 사용하여 다음 라운드 슬롯 개수를 증가시키고 연속적인 빈 슬롯이 threshold 값 이상 발생 할 경우 QueryAdjust 명령어를 사용하여 다음 라운드 슬롯 개수를 감소시킨다. 본 논문에서는 FAFQ 알고리즘을 사용하지 않는 경우와 FAFQ 알고리즘의 threshold 값을 서로 다르게 설정하여 사용하는 경

```

If number of continuous collided slots > Thcoll
    Reader sends QueryAdjust.
    Q = Q + 1
else if number of continuous empty slots > Themp
    Reader sends QueryAdjust.
    Q = Q - 1
else
    Reader sends QueryRep.
end
    
```

그림 11. FAFQ 알고리즘의 의사코드

우의 성능을 모의실험을 통하여 비교한다. 4장의 모의실험 결과로부터 FAFQ 알고리즘의 threshold 값이 5인 경우, 가장 높은 성능 향상을 보인다. 따라서 FAFQ 알고리즘의 threshold 값을 5로 고정한다.

그림 11은 FAFQ 알고리즘을 구현하기위한 의사코드이다.

3.1.2 AAFQ 알고리즘

FAFQ 알고리즘에서 연속적인 충돌 발생 슬롯개수, 빈 슬롯개수의 threshold 값을 5로 고정하였다. AAFQ (Adaptive Adjustable Framed Q) 알고리즘은 threshold 값을 표 1과 같이 Q값에 따라 threshold 값을 적절하게 조절하여, 다음 라운드의 슬롯 개수를 동적으로 조절한다. AAFQ 알고리즘은 QueryAdjust 명령어의 사용 빈도수를 줄여 태그인식시간, 충돌 비율을 최소화 할 수 있고, 데이터 처리량, 시스템 효율을 최대화 할 수 있다.

그림 12는 AAFQ 알고리즘을 구현하기위한 의사코드이다.

표 1. AAFQ 알고리즘의 Threshold 선택 방법

Q	0	1	2	3	...	14	15
Continuous collided slots / empty slots threshold	0	1	2	3	...	14	15
Total # of slots	2 <sup>0</sup>	2 <sup>1</sup>	2 <sup>2</sup>	2 <sup>3</sup>	...	2 <sup>14</sup>	2 <sup>15</sup>

```

Reader sends QueryAdjust.
If Q == Q + 1
    Themp = Themp + 1
    Thcoll = Thcoll + 1
else if Q == Q - 1
    Themp = Themp - 1
    Thcoll = Thcoll - 1
end
    
```

그림 12. AAFQ 알고리즘의 의사코드

3.2 Grouping 방법

현재 RFID 리더는 RF 전력을 2<sup>1</sup> ~ 2<sup>4</sup>의 4단계로 나누어 송출할 수 있으며, 리더와 태그간 거리에 따라 리더의 RF 송출 전력 4단계에 반응하는 태그들은 각기 다르다. 따라서 리더 인식범위 안에 있는 모든 태그들은 리더의 전력 송출에 따라 네 개의 그룹으로 나누어질 수 있고, 나누어진 그룹마다 각각의 충돌방지 알고리즘을 수행함으로써, 태그 인식 시간, 데이터 처리량, 시스템 효율, 충돌 비율 등의 시스템의 성능을 개선할 수 있다.

IV. 모의실험을 통한 성능분석

본 논문에서는 EPCglobal Class1 Gen2기반 Probabilistic Slotted 충돌방지 알고리즘과 제안된 Gen2 기반으로 QueryAdjust 명령어를 사용한 효율적인 충돌방지 알고리즘과 Grouping 방법의 성능을 비교 및 분석한다.

4.1 모의실험 환경 및 파라미터

태그, 리더를 MatLab를 이용 하여 모델링 하였으며 모의실험에서 설정된 파라미터들은 표 2와 같다. 태그의 개수는 0개에서 무한개까지 설정할 수

표 2. Gen2 알고리즘의 모의실험 파라미터<sup>[9]</sup>

Parameters	Descriptions	Values in Specification	Values in Simulation
Tari	Reference time interval for a data-0 in Interrogator-to-Tag signaling	6.25 μs, 12.5 μs, or 25 μs	12.5 μs
DR	Divide Ratio	64/3 or 8	8
RTcal	Interrogator-to-Tag calibration	2.5 Tari ≤ RTcal ≤ 3.0 Tari	3 Tari = 37.5 μs
TRcal	Tag-to-Interrogator calibration	17.2 μs ≤ TRcal ≤ 200 μs, if DR = 8 RTcal ≤ TRcal ≤ 3 RTcal	2 RTcal = 75 μs
LF	Link frequency	LF = DR/TRcal	107 kHz
Tpri	Link pulse-repetition interval	Tpri = 1/LF	9.375 μs
T1	Time from Interrogator transmission to Tag response	MAX (RTcal, 10Tpri)	10 Tpri = 93.75 μs
T2	Time from Tag response to Interrogator transmission	3.0 Tpri ≤ T2 ≤ 20.0 Tpri	10 Tpri = 93.75 μs
T3	Time an Interrogator waits, after T1, before it issues another command	0.0 Tpri	0 μs
T4	Minimum time between Interrogator commands	2.0 RTcal	75 μs
T=>R Data Rate	Tag-to-Interrogator link data rate	LF, if FM0 modulation	LF = 107 kbps

있으나 본 논문에서는 현실적인 상황을 고려하여 1000개까지 한정하였다.

#### 4.2 AFQ 알고리즘 모의실험결과

그림 13은 FAFQ 알고리즘을 사용하지 않는 경우와 FAFQ 알고리즘의 threshold 값을 서로 다르게 설정하여 사용하는 경우의 성능을 비교한 결과이다. 이 중 FAFQ 알고리즘의 threshold 값이 5인 경우, 가장 높은 성능 향상을 보인다.

그림 14는 태그 개수를 0개에서부터 1000개까지 증가 시킬 경우, 이 때 측정된 각각의 태그인식시간을 보여주고 있다. 그림 14와 같이 태그의 개수가 증가함에 따라 태그인식시간이 증가하는 것을 볼 수 있다. 제안한 알고리즘을 사용했을 경우, 태그 개수가 800개 일 때, 기존의 충돌방지 Q 알고리즘보다 약 0.2초의 태그인식시간이 향상되는 것을 볼 수 있다.

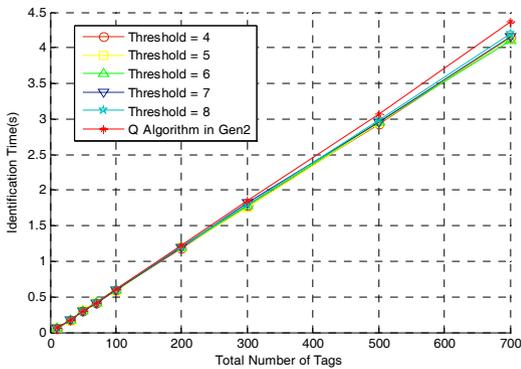


그림 13. threshold 값에 따른 FAFQ 알고리즘의 성능 비교

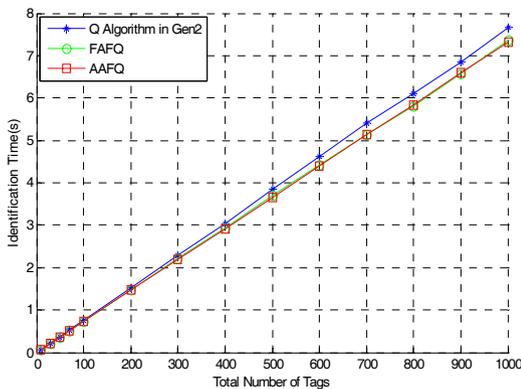


그림 14. 기존의 Q 알고리즘과 제안된 충돌방지 알고리즘의 태그인식시간 비교

그림 15는 태그 개수를 0개에서부터 1000개까지 증가 시킬 경우, 이 때 측정된 각각의 데이터 처리량을 보여주고 있다. 그림 15와 같이 기존의 충돌방지 Q 알고리즘을 사용했을 경우, 데이터 처리량이 74kbps를 기준으로 나타나는 것을 볼 수 있다. 제안한 알고리즘을 사용 했을 경우, 기존의 충돌방지 Q 알고리즘보다 3kbps~4kbps가량 데이터 처리량이 향상되는 것을 볼 수 있다.

그림 16은 태그 개수를 0개에서부터 1000개까지 증가 시킬 경우, 이 때 측정된 각각의 시스템 효율을 보여주고 있다. 그림 16과 같이 기존의 충돌방지 Q 알고리즘을 사용했을 경우, 시스템 효율이 69.5%를 기준으로 나타나는 것을 볼 수 있다. 제안한 알고리즘을 사용 했을 경우, 기존의 충돌방지 Q 알고리즘보다 5%가량 시스템 효율이 향상되는 것을 볼 수 있다.

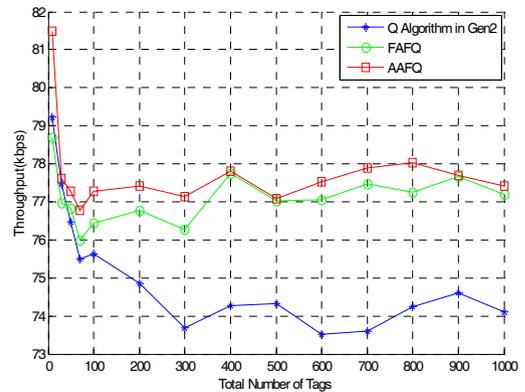


그림 15. 기존의 Q 알고리즘과 제안된 충돌방지 알고리즘의 데이터 처리량 비교

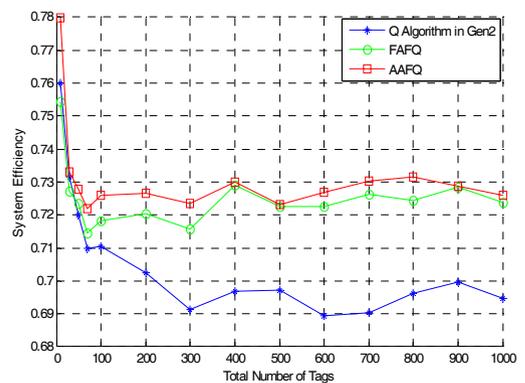


그림 16. 기존의 Q 알고리즘과 제안된 충돌방지 알고리즘의 시스템 효율 비교

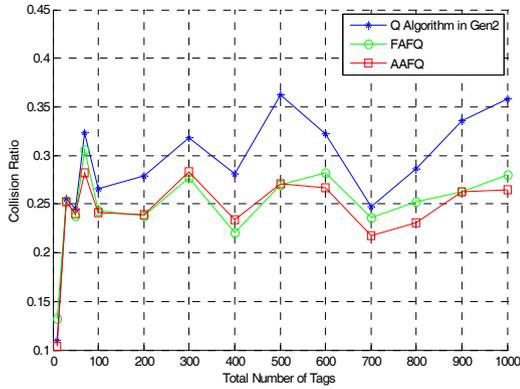


그림 17. 기존의 Q 알고리즘과 제안된 충돌방지 알고리즘의 충돌 비율 비교

그림 17은 태그 개수를 0개에서부터 1000개까지 증가시킬 경우, 이 때 측정된 각각의 충돌 비율을 보여주고 있다. 그림 17과 같이 기존의 충돌방지 Q 알고리즘을 사용했을 경우, 충돌 비율이 30%를 기준으로 나타나는 것을 볼 수 있다. 제안한 알고리즘을 사용했을 경우, 기존의 충돌방지 Q 알고리즘보다 4%~5%가량 충돌 비율이 감소하는 것을 볼 수 있다.

### 4.3 Grouping 방법 모의실험결과

Grouping 방법도 충돌방지 Q 알고리즘과 마찬가지로 태그, 리더를 MatLab를 이용 하여 모델링 하였으며 모의실험에서 설정된 파라미터들 및 태그 개수 설정 또한 충돌방지 Q 알고리즘과 동일하다.

리더의 인식거리에 있는 태그들은 2차원 상에 균일하게 위치하고, 또한 이동이 없다고 가정한다.

그림 18은 기존의 충돌방지 Q 알고리즘과 Grouping 방법의 태그인식시간을 비교 했을 경우, Grouping 방법의 태그인식시간은 Ungrouping 방법

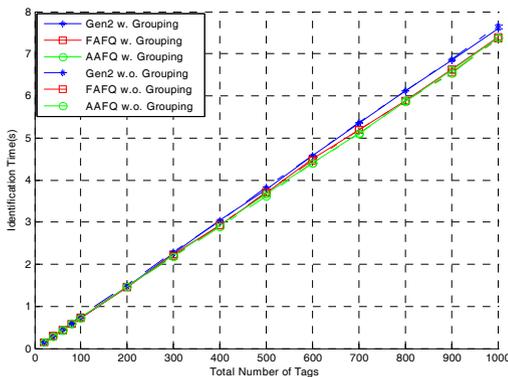


그림 18. 기존의 Q 알고리즘과 Grouping 방법의 태그인식 시간 비교

의 태그인식시간과 비슷하게 나타난다.

그림 19~21은 기존의 충돌방지 Q 알고리즘과 Grouping 방법의 데이터 처리량, 시스템 효율, 충돌 비율을 나타낸 것이다. Grouping 방법을 사용했을 경우, 기존의 충돌방지 Q 알고리즘은 높은 성능 개

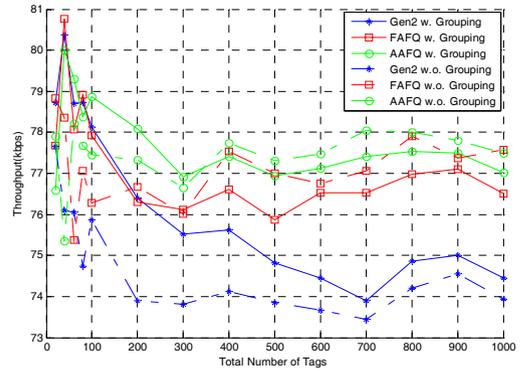


그림 19. 기존의 Q 알고리즘과 Grouping 방법의 데이터 처리율 비교

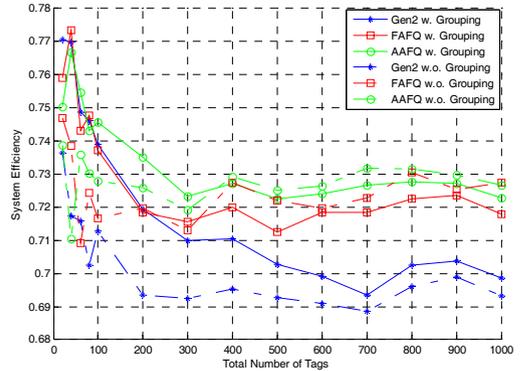


그림 20. 기존의 Q 알고리즘과 Grouping 방법의 시스템 효율 비교

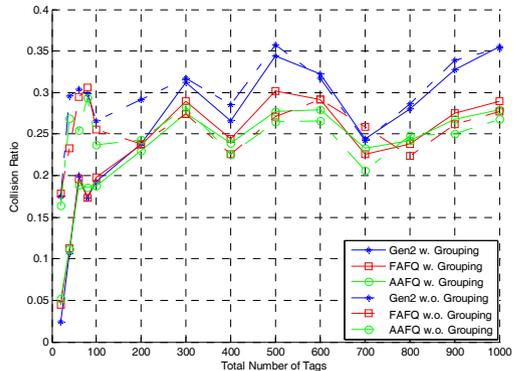


그림 21. 기존의 Q 알고리즘과 Grouping 방법의 충돌 비율 비교

선 효과를 볼 수 있지만, AFQ 알고리즘에 대해선 Grouping으로 인해 태그개수가 줄어들어 연속적인 충돌 슬롯이나 빈 슬롯이 threshold 보다 발생 확률이 낮아 Ungrouping 방법의 성능과 비슷하다.

### V. 결 론

본 논문에서는 Gen2 충돌방지 Q 알고리즘을 이용하여 현재 라운드 내에서 QueryAdjust 명령어를 사용하는 두 개의 충돌방지 알고리즘을 제안하고, Grouping 방법을 소개하였다. 제안한 알고리즘의 성능 검증을 위하여 태그인식시간, 데이터 처리량, 시스템 효율, 충돌 비율과 같은 성능 인덱스를 새롭게 정의하였으며, 새롭게 정의한 성능 인덱스로 모의실험을 통하여 기존의 Gen2 기반의 알고리즘과 제안한 QueryAdjust 기반의 Gen2 충돌방지 알고리즘의 성능을 비교 및 분석 하였다. 제안한 QueryAdjust 기반의 Gen2 충돌방지 알고리즘을 사용하는 경우, 기존의 충돌방지 Q 알고리즘과 성능을 비교 및 분석 하였을 때 태그인식시간, 데이터 처리량, 시스템 효율, 충돌 비율과 같은 성능이 월등히 향상됨을 확인하였다. 또한 Grouping 방법은 Gen2 충돌방지 Q 알고리즘의 태그인식시간 및 충돌 비율을 감소시키고, 데이터 처리량 및 시스템 효율을 증가 시킬 수 있지만, FAFQ 알고리즘과 AAFQ 알고리즘에 대해선 모의실험 결과에서처럼 Ungrouping 방법과 비슷한 성능을 보인다.

### 참 고 문 헌

[1] K. Finkenzeller, *RFID Handbook ; Fundamentals and applications in contactless Smart Cards and Identification*. 2<sup>nd</sup> Edition, John Wiley & Sons Ltd, pp.195-219, 2003.

[2] S. Sarma, D. Brock, and D. Engels, "Radio frequency identification and electronic product code," *IEEE MICRO*, 2001.

[3] 차재룡, 김재현, "RFID 시스템에서의 태그수를 추정하는 ALOHA 방식 Anti-collision 알고리즘," *한국통신학회논문지*, Vol.30, No.9, pp.814-821, 2005, 9.

[4] 권성호, 모희숙, 최길영, 표철식, 채중석, "Gen2기반 RFID 시스템에서의 충돌방지 알고리즘에 관한 연구," *한국통신학회논문지*, Vol.31, No.6, pp.561-571, 2006. 6.

[5] Kawakita Y., and Mitsugi J., "Anti-collision performance of Gen2 air protocol in random error communication link," in *Proc. The International Symposium on Applications and the Internet Workshops*, Jan. 2006, pp.68-71.

[6] EPCglobal, *Radio-Frequency Identity Protocols Class 1 Generation-2 UHF Protocol for Communications at 860 MHz-960 MHz Version 1.1.0*, Dec. 17, 2005.

[7] 김정곤, 이제경, "UHF 대역 RFID 시스템의 충돌 방지 기술 동향," *한국통신학회지*, 23권 제12호, pp.93-106, 2006. 12.

[8] Harald Vogt, "Efficient object identification with passive RFID tags," in *Proc. Conf. on Pervasive Computing*, Vol.2414, 2002, pp.98-113.

[9] 윤희석, 마나르 모하이센, 장경희, 배지훈, 최길영, "EPCglobal Gen-2 RFID 시스템 태그와 리더간의 무선 전송 방식 성능 분석," *한국전자과학회논문지*, 18권 제9호, pp.1047-1056, 2007. 9.

송 인 찬 (InChan Song)

준회원

2007년 2월 인천대학교 전자공학과 (공학사)  
 2007년 2월~현재 인하대학교 정보통신 대학원 석사과정  
 <관심분야> RFID / USN Systems, MAC Protocol, IEEE 802.11/15



범 호 (Xiao Fan)

준회원

2003년 2월 중국의과대학교 임상의학과 (공학사)  
 2005년 2월 청화대학교 소프트웨어공학과 (공학사)  
 2007년 2월~현재 인하대학교 정보통신 대학원 석사과정  
 <관심분야> RFID / USN Systems, MAC Protocol, IEEE 802.11/15



장 경 희 (KyungHi Chang)

중신회원



1985년 2월 연세대학교 전자공학과 (공학사)

1987년 2월 연세대학교 전자공학과 (공학석사)

1992년 8월 Texas A & M Univ., EE Dept. (Ph.D.)

1989년~1990년 삼성종합기술

원 주임연구원

1992년~2003년 한국전자통신연구원, 이동통신연구소 무선전송방식연구팀장 (책임연구원)

2003년~현재 인하대학교 정보통신대학원 부교수

<관심분야> 4세대 이동통신 및 3GPP LTE 무선 전송 방식, WMAN 및 DMB 시스템 무선전송기술, Cognitive Radio, Cross-layer Design, Cooperative Relaying System, RFID / USN Systems

신 등 범 (Dong-Beom Shin)

정회원



1991년 2월 충남대학교 전자공학과 (공학사)

1993년 2월 충남대학교 전자공학과 (공학석사)

2000년~현재 한국전자통신연구원 선임연구원

<관심분야> RFID System,

RTLS System, Network System

이 형 섭 (Heyung-Sub Lee)

정회원



1985년 2월 충남대학교 전자공학과 (공학사)

1994년 8월 충남대학교 전자공학과 (공학석사)

2002년 8월 충남대학교 전자공학과 (공학박사)

1991년~현재 한국전자통신연구

원 책임연구원/팀장

<관심분야> RFID System, RTLS System, Network System