

충돌 패턴을 고려한 RFID 태그 인식 프로토콜

A New RFID Tag Identification Protocol Utilizing Collision Patterns

박 영 재* 김 영 범**
(Young-Jae Park) (Young-Beom Kim)

요 약

RFID 시스템에서는 단일 무선 채널을 공유하는 통신으로 인하여 다수의 태그 응답에 따른 충돌(collision)이 발생한다. QT(Query Tree)와 같은 기존 프로토콜에서 이러한 충돌은 낭비되는 타임 슬롯으로 간주되어 처리된다. 본 논문에서는 다수의 태그들의 응답에 따라 발생하는 충돌의 패턴을 분석하여 성능향상에 이용할 수 있는 ASP (Adjustable splitting by patterns of collisions) 알고리즘을 제안한다.

Abstract

In RFID Systems, collisions between multiple tags frequently arise due to simultaneous responses from multiple tags using the same communication channel. Most of anti-collision protocols such as QT regard these collisions as useless cycles, thereby wasting the channel bandwidth. In this paper, we propose a new anti-collision protocol, namely ASP (Adjustable splitting by patterns of collisions) protocol that utilizes the patterns collision for noticeable performance enhancements.

Key words : RFID, Tag Identification, Collision, Query Tree

I. 서 론

RFID (Radio Frequency Identification) 시스템은 물체에 부착된 태그(tag)의 ID를 부착하고 식별하여 응용하는 기술로서 다방면에 응용이 가능하며 특히, 지능형 교통 시스템 (ITS) 구축을 위한 핵심기술로서 인식되고 있다. RFID는 리더(reader)와 태그(tag)들로 구성되어있다. 리더의 질의(interrogation)에 해당 태그가 반응하는 방식이 일반적이다. 태그는 전력 공급 유형에 따라, 능동형(active), 반수동형(semi-passive), 수동형(passive) 태그로 구분된다.

수동형 태그의 경우, 리더가 발생시키는 전자기 신호로부터 파워를 유도하여 구동되기 때문에 경제적 인 면에서 가장 적합한 형태이다. 따라서 본 논문에서는 RFID 수동형 태그에 초점을 두고 연구를 진행한다.

RFID 시스템에서 리더의 가장 중요한 역할은 자신의 인식 범위 내에 있는 모든 태그들은 최대한 빠르게 식별해 내는 것이다. 태그들을 모두 식별한 후에 리더는 다른 명령을 수행할 수 있도록 지시하는 것이다. 리더는 태그를 식별하기 위해서 질의(query) 명령을 전송하게 되는데, 태그는 정의된 동

* 주저자 : 건국대학교 전자공학부 박사과정

** 공저자 및 교신저자 : 건국대학교 전자공학부 정교수

† 논문접수일 : 2011년 6월 27일

† 논문심사일 : 2012년 1월 9일

† 게재확정일 : 2012년 1월 11일

작에 따라서 자신의 ID를 포함하는 메시지를 리더에게 전송하여 응답을 한다. 태그는 반송파감지(carrier sensing) 기능이 없기 때문에 복수의 태그들의 응답은 리더 측에서 충돌을 발생 시킬 수 있다.

따라서 RFID 리더의 질의에 응답하는 태그의 수에 따라서 다음 3가지 경우의 사이클로 생각해볼 수 있다.

Collision cycle : 복수의 태그가 동시에 응답한 경우로써 리더는 태그의 ID를 식별할 수 없다.

Idle cycle : 응답하는 태그가 없는 경우로써 낭비되는 시간이 된다.

Success cycle : 정확히 하나의 태그만이 응답함으로써 리더는 태그의 ID를 식별할 수 있다.

성공 사이클의 수는 리더가 인식해야 할 태그들의 수와 동일하기 때문에, RFID 시스템에서의 성능향상은 다른 두 경우의 사이클들을 줄이는 것으로 볼 수 있다. 특히 충돌 사이클을 줄이는 것으로 성능향상을 꾀하는 많은 논문들이 제안되었다. RFID 시스템의 태그 충돌 회피(tag anti-collision) 프로토콜은 크게 ALOHA[1] 기반과 트리 기반 프로토콜[2, 3]로 나누어진다. ALOHA 기반 프로토콜은 충돌을 줄일 수 있지만, 특정 태그가 일정 기간 이상 인식되지 않을 수도 있는 태그 기아(tag starvation) 문제를 갖고 있다. 트리 기반 프로토콜은 상대적으로 태그의 식별 시간이 더 길지만, 태그 기아 문제는 없다.

트리 기반의 충돌 회피 기법은 이진탐색(binary search) 알고리즘과 흡사하다고 볼 수 있다. 충돌이 발생 할 때마다 태그들을 두 서브셋(subset)으로 나누어 각각에 대해 다시 질의를 보내어 식별하게 된다. 트리 기반의 대표적인 프로토콜에는 BT(Binary Tree)[2]와 QT(Query Tree)[3]를 들 수 있다. BT는 각 태그들을 임의의 이진수(random binary number, 0 or 1)를 생성해 내기 위한 함수가 필요하고, 전송 순서를 결정하기 위한 카운터(메모리)도 필요하다. QT는 리더가 태그의 ID를 인식해 가면서 질의하는 방식으로 태그에 추가적인 카운터가 필요 없다는 장점이 있다. 따라서 BT는 태그 ID의 분포와 질의 횟수와는 관련이 없고, QT는 ID들이 유사한 값을 가질수록, 즉 분포가 몰려 있을수록 충돌 횟수가 많아져서 성능이 떨어지게 되는 단점이 있다. 본 논문의

제안은 충돌이 발생할 경우 해당 패턴을 분석하여 이용하는 방법인데, 다른 기법에도 적용될 수는 있지만, 태그의 ID를 인식해 가면서 수행하는 QT에서 추가적인 성능향상이 가능하기 때문에 QT에 적용하였다.

본 논문은 구성은 다음과 같다. 먼저 이해를 위하여 QT 프로토콜에 대해서 알아보고 충돌발생시 리더에서 감지하는 신호의 패턴을 알아본다. 해당 충돌 패턴들을 3가지 형태로 분석 정리하여 제시하는 ASP(Adjustable Splitting by the Patterns of collisions) 알고리즘[4]을 QT에 적용하여 성능을 향상시킨다.

II. QT(Query Tree) 프로토콜과 충돌 신호의 패턴

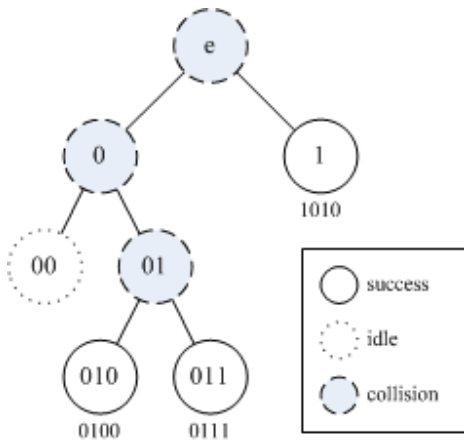
1. QT(Query Tree) 프로토콜

QT 프로토콜은 리더가 태그에 질의(query)를 포함한 메시지를 전송하고, 질의에 해당하는 태그들이 반응하여 응답메시지를 전송하는 방식이다. 이때 질의는 ε 또는 2진수의 형태로 구성되어 있고, 리더는 내부의 큐(queue)의 형태로 된 저장 메모리에 질의들을 저장하거나 사용하게 된다. 사용된 질의는 큐에서 삭제되고, 태그들의 응답에 따라 새로운 질의를 큐에 삽입하거나 계속 진행하게 된다. 처음 큐에 저장된 질의는 ε 이고 이는 모든 태그가 응답하라는 의미라고 할 수 있다. 2진수로 되어 있는 질의들은 태그의 ID가 2진수이기 때문인데, 태그는 수신된 질의와 자신의 ID가 앞자리에서부터 비교 하고 일치(prefix-matching) 하였을 경우 응답메시지를 리더에 전송하게 된다. 예를 들어 0100, 0111, 1010 이라는 ID를 갖는 태그들이 존재할 경우, 질의가 01이면 태그의 ID가 01로 시작하는 0100, 0111 태그들이 응답하게 되는 것이다.

리더가 질의에 대하여 응답하는 태그의 수가 2개 이상인 경우 리더는 이를 충돌 사이클(collision)로 간주한다. 충돌이 발생하게 되면 충돌을 야기한 질의의 다음 비트에 각각 0, 1을 붙여(binary splitting) 큐에 저장하게 된다. 예를 들면, 01이 충돌을 야기한 질의면 큐의 마지막 부분에 010과 011이 저장되

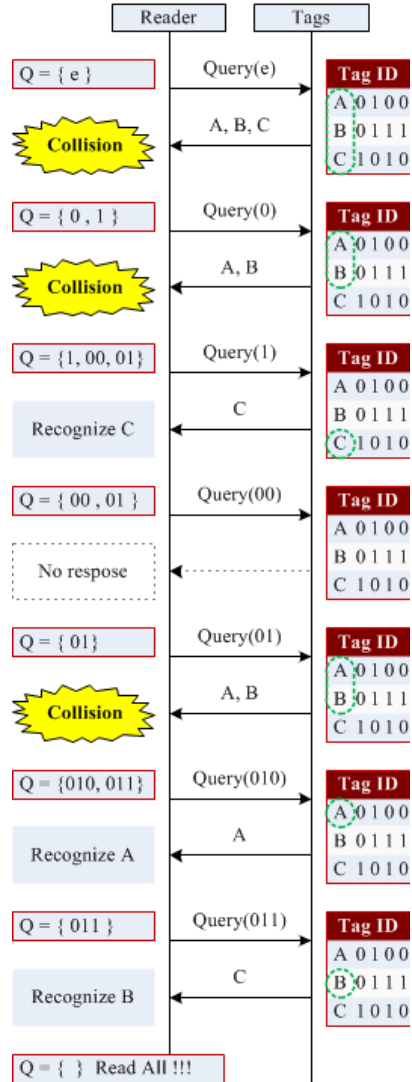
는 것이다. 리더의 질의에 대하여, 또 다른 경우로 응답이 없을 경우(idle)와 정확히 하나의 태그가 응답을 하여 인식에 성공(success) 하는 경우이다. 두 경우 모두 새로운 질의를 추가하지 않으며, 무응답의 경우도 충돌과 마찬가지로 시스템 전체의 인식 시간을 늘리게 됨으로써 될 수 있으면 발생하지 않아야 한다.

위와 같은 과정을 거치면서 리더의 큐에 저장되어 있는 질의가 존재하지 않을 때, 리더는 모든 태그를 인식하였음을 알 수 있으므로 프레임을 종료한다. 이해를 돕기 위해 ID가 0100, 0111, 1010인 3개의 태그를 QT 프로토콜을 이용하여 인식하는 과정을 보이겠다. <그림 1>은 리더의 질의 메시지에 의한 태그의 응답에 따라 태그를 2개의 하위 집합으로 나뉘는 과정을 보여주는 것이다. 질의가 0일 때 충돌이 발생하기 때문에 00과 01 두 개의 하위 집합으로 분류하고, 질의가 1의 경우처럼 충돌이 일어나지 않을 때는 하위 집합으로 나누지 않는다. 질의가 00일 경우에는 해당하는 태그가 없기 때문에 무응답 사이클이 된다.



<그림 1> Query Tree 프로토콜의 태그 인식 과정
<Fig. 1> Tag Identification in Query Tree Protocol

리더의 메모리 큐에 저장되는 질의의 변화를 확인해 가면서 QT 프로토콜의 동작과정을 살펴보면 <표 1>과 같다. <그림 2>는 QT 프로토콜의 동작을 시간 다이어그램을 통해서 알기 쉽도록 나타낸 것이다.



<그림 2> QT 프로토콜의 동작
<Fig. 2> QT Protocol Operation

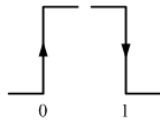
<표 1> QT 프로토콜의 동작과정
<Table 1> QT Protocol Operation

| Queue | Query | Response Tags | Result |
|-------------|-------|----------------|-----------|
| {e} | e | 0100 0111 1010 | Collision |
| {0, 1} | 0 | 0100 0111 | Collision |
| {1, 00, 01} | 1 | 1010 | Read |
| {00, 01} | 00 | | Idle |
| {01} | 01 | 0100 0111 | Collision |
| {010, 011} | 010 | 0100 | Read |
| {011} | 011 | 0111 | Read |
| { } | | | Finish |

2. 충돌 신호 분석

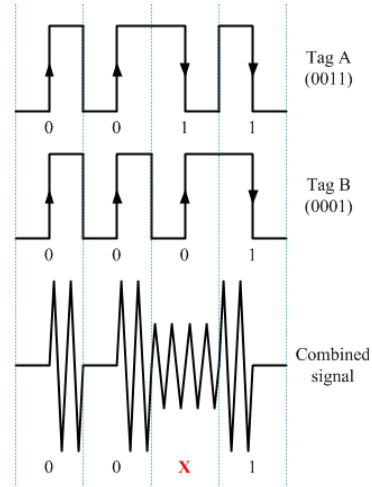
RFID 수동 태그(passive tags)는 리더로부터 메시지와 전력을 공급받고 후방산란(back-scattering) 기법을 통하여 응답메시지를 리더로 전송한다.

기존의 연구에서는 다수의 태그들이 동시에 응답하는 경우 해당 사이클은 충돌 사이클로 규정하고 사용 불가능한 것으로 간주하였다. RFID는 여러 가지 NRZ, Manchester, Miller 코딩 등의 코딩 기법과 ASK, FSK, PSK 와 같은 변조 기법을 사용하고 있다. 본 논문에서는 맨체스터(Manchester) 인코딩 및 ASK 변조를 사용하는 통신에서의 충돌 신호를 확인하고자 한다. 맨체스터 코드는 신호레벨의 변이(transition)에 따라서 비트의 값이 정의되고, <그림 3>은 맨체스터를 사용한 비트 코딩 방법의 예이다.

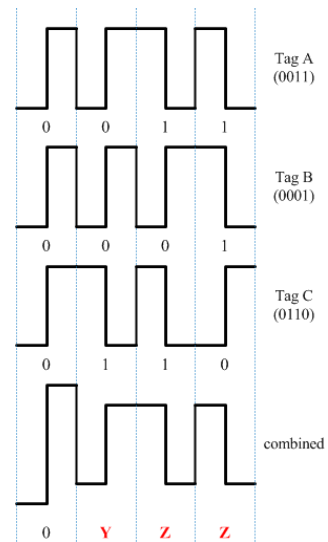


<그림 3> 맨체스터를 사용한 비트 코딩
<Fig. 3> Bit-coding by Manchester

예를 들어 0011과 0001이라는 ID를 가지는 두 개의 태그가 응답한 경우를 고려하자. <그림 4>는 이 경우에서 리더에서의 수신한 신호의 예이다. 각각의 태그의 ID에서 비트가 틀린 구간만 신호가 합성이 되어, 신호레벨의 변이가 없어졌다. 따라서 태그 A와 태그 B는 각각 00X1의 ID를 가지고 있다고 판단할 수 있다. 위의 예에서의 X와 같은 구간처럼 신호레벨의 변이가 없고 균등한 경우를 본 논문에서는 임의로 “충돌패턴 X” 라고 정의하겠다. 충돌패턴 X는 전체 ID에 해당하는 구간에서 “01XX01X1...”과 같이 여러 차례 나올 수 있고, 충돌되는 곳이 최소한 2개 이상의 다른 비트 ID를 갖는 태그가 응답하였음을 알 수 있다. 단, 여기서 2개의 태그가 각각의 ID가 상이할 수 있기 때문에, 실제로 오직 2개의 태그만이 존재하더라도 충돌패턴 X는 여러 차례 나올 수 있고, 충돌 위치에서는 응답한 태그들의 0 또는 1의 수가 동일하다는 점을 알 수 있다.



<그림 4> 2개의 태그가 응답한 경우의 합성신호의 예
<Fig. 4> Example of Combined-signal in case of 2 Tags



<그림 5> 3개의 태그가 응답한 경우의 합성신호의 예
<Fig. 5> Example of Combined-signal in case of 3 Tags

<그림 5>는 3개의 태그가 응답한 경우의 예로써 충돌 부분이 충돌패턴 X와는 모양이 상이하다. 이것들도 충돌패턴 Y, 충돌패턴 Z로 각각 정의 하겠다. 충돌패턴 Y는 해당 비트의 자리에서 0의 개수가 1의 개수보다 많은 경우에 발생하는데, 리더에서는 신호레벨의 변이가 있는 0의 값과 약간 유사하지만, 0은 아님을 알 수 있다. 마찬가지로 충돌패

턴 Z도 정의할 수 있다. 충돌패턴 Y와 충돌패턴 Z는 태그가 약간의 신호레벨의 차이가 존재하려면 비트의 수가 서로 달라야 함으로, 응답한 태그의 개수가 최소한 3개 이상임을 알 수 있다.

III. ASP(Adjustable Splitting by the Patterns of collisions) 프로토콜

기존의 연구에서는 다수의 태그들이 동시에 응답하는 경우 해당 사이클은 충돌 사이클로 규정하고 사용 불가능한 것으로 간주하였다. 특히 QT 같은 프로토콜은 태그들의 ID의 분포에 따라 성능의 차이가 심하다는 단점을 가지고 있다.

충돌패턴에 따른 특징을 X인 경우와 Y or Z의 경우로 구분하여 정리할 수 있다.

충돌패턴 X : 신호의 단차가 없으며 다음과 같이 판단할 수 있다. 충돌되는 부분에서 1을 가지는 태그의 개수와 0을 가지는 태그의 개수가 동일하다. 전체에서 충돌패턴 X만 존재할 경우 태그는 최소한 2개 이상이 존재하고, 충돌패턴 X가 한곳에서만 존재하는 경우 해당비트가 각각 0과 1인 2개의 태그가 존재함을 판별할 수 있다.

충돌패턴 Y or 충돌패턴 Z : 신호의 단차가 정상적인 0 또는 1의 경우와 상이하며 다음과 같이 판단할 수 있다. 태그의 ID는 중복되지 않기 때문에, Y와 Z 패턴은 두 부분 이상에서 존재하게 된다. 충돌되는 부분에서 1을 가지는 태그의 개수와 0을 가지는 태그의 숫자 중에서 어느 한쪽이 많기 때문에 응답하는 태그는 최소한 3개 이상이 존재한다.

충돌 패턴을 QT에 적용할 수 있는 경우를 살펴보면 다음과 같고 <표 2>에 정리하였다.

충돌패턴 X만 존재 : 최소한 태그가 2개 이상으로 존재함을 알 수 있으므로, 이진분할(binary splitting)을 적용한다. 단, 한곳에서만 충돌패턴 X가 존재하면 바로 2개의 태그를 인식 가능하다.

충돌패턴 (Y or Z)만 존재 : 최소한 태그가 3개 이상으로 존재함을 알 수 있다. 이진분할 또는 4개로 분할(4-ary splitting) 할 수 있는데, 태그의 개수가 정확히 3개인 경우는 이진분할을 사용할 경우 충돌 사이클이 1회 발생하고, 4-ary는 무응답 사이클이 1회 발생하여 성능이 동일하다. 그러나 4개 이상의

태그가 존재한다면 4-ary가 유리하다. 그러므로 전체적으로 4-ary를 사용하는 것이 유리하다.

충돌패턴 (Y or Z)과 X가 모두 존재 : 최소한 태그가 4개 이상으로 존재함을 알 수 있다. 따라서 4-ary로 분할한다.

<표 2> 충돌 패턴의 개수에 따른 동작
<Table 2> Operation by Numbers of Collision Patterns

| # of Pattern | | Operation of Reader | |
|--------------|------|---------------------|----------------------|
| X | Y/Z | Algorithm | Remark |
| 1 | 0 | Read the 2 tags | # of resp. tags == 2 |
| > 1 | 0 | Binary splitting | # of resp. tags >= 2 |
| 0 | >= 1 | 4-ary splitting | # of resp. tags >= 3 |
| >= 1 | >= 1 | 4-ary splitting | # of resp. tags >= 4 |

ASP 프로토콜은 QT 프로토콜에서 태그의 동작은 동일하게 유지하였고, 리더에서 태그 충돌발생시 충돌 패턴에 따른 동작을 추가하였다. <그림 6>은 이것을 의사 코드로 표현하였다.

<그림 6> ASP 프로토콜 리더의 충돌발생시 동작
<Fig. 6> Reader Operation by Collisions in ASP

```

/* After detecting collision due to response by
multiple tags */
if # of (Y or Z) > 0 then
    4-ary splitting
else
    if # of X > 1 then
        binary splitting
    else /* # of X == 1 */
        read the 2 tags
    end if
end if
    
```

이해를 돕기 위하여 <표 3>과 <표 4>는 다른 ID를 갖는 태그들 0011, 0100, 0110, 1010, 1011 이 인식되는 과정을 각각 QT 와 ASP를 적용한 예를 나타내었다. QT는 총 13회, ASP는 총 5회의 사이클이 소요되었다.

〈표 3〉 QT에서의 태그 인식과정
 〈Table 3〉 Tag Identification in QT

| Queue | Qry | Response Tags | Result |
|----------------------|------|-----------------------------|-----------|
| {e} | e | 0011 0100 0110 1010 1011 | Collision |
| {0,1} | 0 | 0011 0100 0110 | Collision |
| {1,00,01} | 1 | 1010 11011 | Collision |
| {00,01,10,11} | 00 | 0011 | Read |
| {01,10,11} | 01 | 0100 0110 | Collision |
| {10,11,010,011} | 10 | 1010 1011 | Collision |
| {11,010,011,100,101} | 11 | | Idle |
| {010,011,100,101} | 010 | 0100 | Read |
| {011,100,101} | 011 | 0110 | Read |
| {100,101} | 100 | | Idle |
| {101} | 101 | 1010 1011 | Collision |
| {1010,1011} | 1010 | 1010 | Read |
| {1011} | 1011 | 1011 | Read |
| {} | | | Finish |

〈표 4〉 ASP에서의 태그 인식과정
 〈Table 4〉 Tag Identification in ASP

| Queue | Qry | Response Tags | Result / operation |
|---------------|-----|-----------------------------|--------------------|
| {e} | e | 0011 0100 0110 1010 1011 | YYZY / 4-ary |
| {00,01,10,11} | 00 | 0011 | Read |
| {01,10,11} | 01 | 0100 0110 | 01X0 / Read 2 |
| {10,11} | 10 | 1010 1011 | 101X / Read 2 |
| {11} | 11 | | Idle |
| {} | | | Finish |

충돌 부분에서 QT는 이진분할로만 진행하고, ASP는 충돌 패턴에 따라 2 혹은 4개의 그룹으로 분할하는 것을 알 수 있다. 충돌 패턴 X가 1회 나온

경우는 인식 사이클이 필요 없이 2개의 태그를 동시에 인식하여 성능이 획기적으로 향상되었다.

IV. 결 론

RFID 통신에서 리더의 다중 태그 인식과정에서 기존의 연구들은 충돌 사이클을 무의미한 사이클로 간주하였고, 다만 다수의 태그가 응답하였는지의 여부만 판단하였다. 태그의 숫자가 많아질수록 충돌 횟수는 늘어나게 되는데, QT와 같은 프로토콜은 항상 일정한 2개의 그룹으로만 분할하였다.

ASP 프로토콜은 충돌 사이클을 2 또는 4개의 그룹으로 유연하게 분할 할 수 있는 알고리즘과 2개의 태그를 동시에 인식하는 알고리즘을 제시하여 전체적인 성능 개선을 도출하였다.

참 고 문 헌

- [1] J. R. Cha and J. H. Kim, "Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system," in *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2006.
- [2] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," in *IEEE Trans. Inform. Theory*, vol. 25, no. 4, pp.505-515, September 1979.
- [3] C. Law, K. Lee and K.-Y. Siu, "Efficient Memoryless Protocol for Tag Identification," in *International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, August 2000.
- [4] J. Myung and W. LEE, "Adaptive Splitting Protocols for RFID Tag Collision Arbitration," in *ACM Mobihoc*, May 2006.

저자소개



박 영 재 (Park, Young-Jae)

2010년 : 건국대학교 박사과정 수료(전자정보통신전공)
2000년 3월 ~ 현재 : 건국대학교 공학석사(전자정보통신전공)
2004년 9월 ~ 2006년 7월 : (주)위트콤 SW개발원
2001년 1월 ~ 2001년 7월 : (주)머큐리 연구개발원



김 영 범 (Kim, Young-Beom)

1984년 : 서울대학교 공과대학 전자공학과 학사
1996년 : 미국 매릴랜드 주립대 전자공학 박사
1986년 3월 ~ 1988년 4월 한국통신 전임연구원
1997년 9월 ~ 현재 : 건국대학교 정보통신대학 전자공학부 교수
2003년 9월 ~ 2004년 8월 : Univ. of British Columbia, Vancouver, Canada 방문교수