

논문 2008-45CI-4-7

# RFID 시스템에서의 적응형 리더 충돌 방지 알고리즘

## (An Adaptive Anti-collision Algorithm for RFID Systems)

옥 치 영\*, 권 성 호\*\*, 최 진 철\*, 최 길 영\*\*, 모 희 숙\*\*, 이 채 우\*\*\*

(Chi-Young Ok, Cheng-Hao Quan, Jin-Chul Choi, Gil-Young Choi, Hee-Sook Mo, and Chae-Woo Lee)

### 요 약

인접한 RFID 리더가 동시에 같은 채널을 사용할 경우 리더간에 충돌이 발생한다. RFID가 밀집된 상황에서는 채널수가 많다고 하더라도 RFID 리더간에 채널접근을 적절히 제어하지 않을 경우 RFID 리더간의 잦은 충돌로 인해 성능을 보장할 수 없다. 기존에 사용되는 리더 충돌 방지 기법인 FH(Frequency Hopping) 및 LBT(Listen Before Talk) 방식의 경우 채널수가 많은 환경에서 효율적으로 채널을 할당하는 메커니즘을 갖고 있지 않기 때문에 동시에 다채널, 밀집리더 환경에서는 성능을 보장할 수 없는 문제가 있다. 본 논문에서는 밀집 리더 환경에서 리더 충돌 문제를 개선할 수 있는 확률적 채널 호핑 알고리즘을 제안한다. 확률적 채널 호핑 알고리즘은 채널을 점유하기 전 LBT 방식을 기반으로 랜덤 백오프를 사용함으로써 여러 리더가 경쟁할 경우 발생하는 충돌을 줄여준다. 또한, LBT 과정에서 타 채널로 호핑하는 확률을 리더의 밀집도 및 채널의 사용률을 반영하여 적응적으로 할당함으로써 채널 이용률을 높인다. 시뮬레이션을 통해 제안하는 알고리즘이 기존의 방법에 비해 우수함을 보인다.

### Abstract

Reader collision may occur when neighboring RFID readers use the same channel at the same time. Especially when the readers are operated in dense mode, even though many channels are available, because of frequent reader collisions we can not guarantee the performance of RFID readers. Conventional solutions such as FH(Frequency Hopping) or LBT(Listen Before Talk) are not effective in this situation because they can not schedule RFID readers effectively when RFID readers are operated in multi-channel, dense reader mode. In this paper, we propose a new RFID reader anti-collision algorithm which employs LBT, random backoff before channel access, and probabilistic channel hopping at the same time. While LBT and Random backoff before channel access reduces collisions between competing readers, probabilistic channel hopping increases channel utilization by adaptively changing the hopping probability by reflecting the reader density and utilization. Simulation results shows that our algorithm outperforms conventional methods.

**Keywords :** RFID, anti-collision, Reader Anti-collision, LBT

### I. 서 론

최근 전 세계적으로 유통, 물류, 의료, 국방 등 여러

분야에 RFID 기술의 적용하기 위한 많은 연구가 진행되고 있다. 특히 2005년 2세대 RFID 기술 (Gen 2)의 규격이 ISO 1800-6 Type C로 규격화됨에 따라 전 세계적으로 RFID 시스템의 도입이 더욱 확산되고 있다<sup>[1]</sup>. 국내에서는 모바일 단말기에 RFID 칩을 내장하여 활용성을 향상시키기 위한 모바일 RFID 시스템의 도입이 시도되고 있다. 하지만 아직은 모바일 RFID에 대한 전 세계적인 선례가 많지 않으므로 모바일 RFID에서 큰 이슈가 되고 있는 간섭 문제에 대한 체계적인 연구가 필요한 실정이다. 특히 채널수가 제한된 환경에서 많은 리더들의 동작을 요구할 수밖에 없는 RFID의 특성상 다중 리더간의 간섭 현상 회피 기술의 개발이 시급하

\* 학생회원, \*\*\* 정회원, 아주대학교 전자공학과  
(School of Electrical and Computer Engineering, Ajou University)

\*\* 정회원, 한국전자통신연구원  
(Electronics and Telecommunications Research Institute)

\* This work was supported by the MKE of Korea[2008-S-023-1, Development of Next Generation RFID Technology for Item Level Applications].

접수일자: 2008년3월26일, 수정완료일: 2008년7월10일

다. 따라서 밀집 리더 환경에서 발생하는 리더간의 간섭 현상을 회피 하는 기존 연구<sup>[2-6]</sup>의 분석을 통하여 새로운 간섭 회피 기술을 개발하는 것이 필수적이다. 그리고 이를 평가할 수 있는 시뮬레이터를 개발하여 향후 발생할 다중 리더 간섭 및 충돌 문제를 방지할 수 있도록 체계적으로 대응할 수 있는 기반이 필요하다. RFID 시스템에서 간섭과 충돌이란 채널에서 정상적인 태그 인식이 방해받는 현상을 의미하는 것으로, 간섭은 신호 레벨에서, 충돌은 프로토콜 레벨에서 다루어지는 개념으로 정의할 수 있다<sup>[7]</sup>.

RFID에서 리더 충돌이 발생하는 경우는 크게 두 가지이다. 첫째, 인접한 리더가 동시에 같은 주파수를 사용하여 태그와 통신하려 하거나 다른 주파수를 사용하는 리더가 같은 태그와 통신하려 할 때 리더 충돌이 발생할 수 있다. 두 리더의 전송 범위 내에 있는 태그를 동시에 인식하려 할 때 문제가 발생된다. 둘째, 처리 공간이 같은 리더들이 동시에 같은 주파수를 사용하여 각각 태그와 통신을 할 경우 리더 충돌이 발생할 수 있다. 이는 리더와 태그간 링크의 비대칭성에 의해 발생한다.

현재 우리나라에서는 RFID 시스템에서의 매체 접근 방식을 기술기준에서 정의하고 있다<sup>[8]</sup>. 기술기준에서는 간섭을 최소화하기 위해 송신전력을 제한하고 채널화를 통해 규제를 하고 있다. 또한, 매체 접근 방식의 주류를 이루고 있는 주파수 도약(Frequency Hopping)과 LBT(Listen Before Talk) 방식에 대한 기본적인 기준만을 정의하고 있다. 예를 들어 LBT 방식에서는 최소 5ms 동안 채널을 감지하도록 되어있고, 최대 4초 동안 하나의 채널을 사용하는 것이 가능하다. 하나의 채널을 사용한 후에는 형평성 있는 채널 분배를 위해 0.1초 동안 해당 채널을 사용할 수 없도록 규정하고 있다. 이와 같이 기술기준에서는 매우 기본적인 틀에서만 매체 접근 방식에 대해 정의하고 있으며 리더 간의 간섭 문제를 해결하기 위한 채널 선택 방법, 채널 감지 방법, 채널 감지 후 동작 방식 등 구체적인 방식은 정의하고 있지 않다. 따라서 각 리더들의 요구 특성을 고려한 효율적인 매체 접근 방법이 필요하다.

현재 RFID 시스템이 광범위하게 응용되면서 점차적으로 많은 리더가 동시에 동작할 수 있도록 시스템 설계가 필요하다. RFID 시스템은 초창기에 하나의 리더가 채널을 사용하는 단일 리더(Single Reader) 환경에서 동작하였고, 여러 리더가 존재하는 다중 리더(Multiple Reader) 환경에서 동작할 수 있도록 발전하였다. 최근에는 동작을 원하는 리더가 사용가능한 채널수

보다 많은 밀집 리더(Dense Reader) 환경에서의 동작이 요구되고 있다. 이에 따라 리더 간에 주파수 간섭을 고려하지 않고 설계되었던 기존의 RFID 시스템 설계 방식에 비해 동작환경 내에 많은 리더가 존재하는 경우에도 RFID의 성능이 떨어지지 않도록 하는 것이 앞으로 RFID를 확산하기 위한 가장 큰 선결 과제이다. 이를 위해 본 논문에서는 밀집 리더 환경에서 특히 뛰어난 성능을 보이는 리더 충돌 방지 알고리즘을 제안하고자 한다. 본 논문에서 제안하는 알고리즘은 LBT 방식을 기반으로 하여 채널의 효율적인 사용을 위해 타 채널로의 호핑 확률을 계산하여 적용한다. 본 연구 결과에 의하면, 4개의 채널과 4개의 리더가 존재하는 환경에서 기존의 LBT 방식을 사용하는 경우 최대 처리율이 45%를 보이는 반면, 제안하는 알고리즘을 적용하는 경우 80%까지 증가하는 것으로 나타났다. 4개의 채널과 8개의 리더가 존재하는 경우에는 각각 45%와 85%로 나타나 처리율 면에서 성능 향상을 검증할 수 있었다. 또한, 리더의 전송 요구가 발생한 후 채널을 점유하여 사용하기 시작할 때까지의 대기시간을 비교해본 결과 마찬가지로 큰 성능 향상을 보였다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 RFID 시스템 매체 접근 방식에 대해서 알아보고, 기존 방식의 문제점을 제시한다. III장에서는 제안하는 알고리즘을 동작 단계별로 설명한다. 그리고 IV장에서는 시뮬레이션을 통해 성능을 분석하고 마지막으로 V장에서 결론을 맺는다.

## II. 관련 연구

RFID 시스템에서의 매체 접근 방식은 크게 공간 분할 방식(SDMA), Beacon 채널 이용 방식, 시분할 방식(TDMA), 주파수 도약(Frequency Hopping) 방식, 캐리어 감지 방식(CSMA) 등으로 나뉜다. 공간 분할 방식은 리더의 방향성이나 배치를 통해 매체 접근을 허용하는 방식이고, 시분할 방식은 각 리더들이 채널 사용 시간을 분배하여 사용하는 방식이다. Colorwave나 Enhanced Colorwave 방식이 시분할 방식에 속하지만, 다수의 채널이 사용되는 경우 적용하기가 어렵다. 주파수 도약 방식은 주파수 분할 방식(FDMA)의 일종으로 미국에서 사용하는 방식이다. 미국에서는 902MHz~928MHz를 사용하고 있으며 한 채널의 점유 대역폭이 500kHz이고 50개 이상의 채널이 사용 가능하다. 이처럼 채널 대역폭이 넓고, 사용 가능한 채널수가 많은 경우

주파수 도약 방식이 유리하다.

반면에 CSMA 방식은 채널 사용이 필요한 경우에만 사전감지를 통하여 채널을 점유하는 방식으로서, LBT 방식이 이에 속한다. LBT 방식은 주파수 대역이 좁을 때 유리한 방식이며, 유럽에서 채택하여 사용하고 있다. 유럽에서는 865MHz~868MHz 대역을 사용하고 있으며, 한 채널의 점유 대역폭은 200kHz이고 총 채널수도 15개밖에 되지 않는다<sup>[6]</sup>.

우리나라에서는 908.5MHz~915MHz의 대역폭을 사용하고 있으며 200kHz의 27개 채널이 사용 가능하다. 국내 무선설비 기술기준에서는 LBT 방식과 주파수 도약 방식을 모두 허용하고 있는데<sup>[8]</sup>, 국내 RFID 주파수 대역이 좁고 채널수가 많지 않기 때문에 본 논문에서는 밀집 리더 환경에 더 적합한 LBT 방식에 초점을 맞추고 있다.

LBT 방식을 사용하는 RFID 시스템은 채널이 사용 중이라고 판단할 때, 해당 채널을 계속해서 모니터링할 것인지 또는 채널을 변경하여 모니터링할 것인지에 결정을 해야 한다. 그러나 앞서 언급했듯이 국내 무선설비 기술기준에서는 이에 대한 구체적인 방법을 정의하고 있지 않다.

LBT를 통하여 하나의 채널이 유희할 때까지 계속해서 기다렸다가 경쟁을 통해 채널을 사용하는 방식은 리더가 동일 채널을 선택하는 비율에 따라 효율이 좌우된다. 즉, 리더들이 서로 다른 채널을 골고루 선택할 경우 채널 이용 효율이 높지만, 동일 채널을 선택하는 비율이 높은 경우에는 채널 이용 효율이 낮아지게 된다. 이와 같은 방식은 하나의 컨트롤러가 각 리더에게 채널 사용 여부를 결정해 주는 중앙 집중 방식(Centralized control)에 적합하지만, 리더들이 각자 채널을 선택하는 분산 방식(Distributed control)에서는 부적합하다. 중앙 집중 방식에서는 리더에게 채널이 중복되지 않도록 효율적인 할당이 쉽게 가능하지만, 분산 방식에서는 이와 같은 채널 할당이 쉽지 않기 때문이다. 다만 중앙 집중 방식에서는 하나의 컨트롤러가 모든 리더들의 채널 사용을 감시할 수 있어야 하기 때문에 실제 환경에 적용이 제한적이라는 단점이 있다.

반면에 리더가 LBT를 통하여 채널이 사용 중이라고 판단할 경우 무조건 랜덤하게 타 채널을 선정하여 호핑하는 방식은 호핑으로 인한 오버헤드를 고려했을 때 효율적이라고 할 수 없다. 특히 밀집 리더 환경에서 리더의 전송 요구량이 많은 경우에는 지나치게 많은 횟수의 호핑이 발생할 수 있다. 일반적으로 리더가 타 채널

로 한번 호핑하기 위해서는 5~20ms정도의 시간이 소요된다.

본 논문에서는 이와 같은 문제를 해결하기 위해 채널에 대기하거나 호핑하는 것을 확률적으로 정해주는 방법을 사용한다. 즉, LBT를 통해 채널이 사용 중이라고 판단되는 경우 채널 모니터링을 통하여 구해진 채널 호핑 확률을 적용한다. 이 호핑 확률은 리더 밀도를 예측하여 채널 환경에 적합하게 정해진다. 이 방식은 무분별한 채널 호핑으로 인한 오버헤드를 막고 효율적으로 비어있는 채널을 찾아갈 수 있도록 설계되어 앞에서 지적한 문제점을 해결할 수 있다. 본 논문에서는 제안하는 알고리즘을 확률적 채널 호핑 알고리즘이라고 정의한다.

### III. 확률적 채널 호핑 알고리즘

확률적 채널 호핑 알고리즘은 밀집 리더 환경에서 리더의 충돌 확률을 줄이고, 채널 이용률을 높이기 위해 LBT 이용시 타 채널로 호핑(hopping)하는 확률을 동적으로 정하는 방식을 사용한다. 타 채널이 비어있을 확률이 높은 경우에는 호핑 확률을 높이는 것이 유리하며, 타 채널 역시 사용 중일 확률이 높은 경우에는 호핑 확률을 작게 하는 것이 유리하다. 호핑 확률을 정의할 때 타 채널이 사용 중일 확률을 예측하기 위해 채널의 이용률을 모니터링하고, 경쟁을 통해 채널을 획득할 확률을 연속적으로 계산하여 사용한다. 또한, 경쟁 윈도우(contention window: CW)의 크기를 채널 환경에 따라 동적으로 할당하여 경쟁 윈도우 크기에 의한 오버헤드(overhead)를 줄인다. 호핑 확률을 정의하고 경쟁 윈도우 크기를 변화시킴으로써 데이터 처리율(throughput)을 높이고, 리더가 채널을 점유하기 위한 대기시간을 최소화하는 것을 목표로 한다.

확률적 채널 호핑 알고리즘에서는 리더 간 충돌 확률을 줄이기 위해 랜덤 백오프를 사용한다. 랜덤 백오프는 채널 감지 후 채널에 동시 접속으로 인해 충돌이 발생하는 것을 막기 위한 방식으로, 채널 감지 이후에 경쟁 윈도우 크기 내에서 랜덤하게 선택된 슬롯 값만큼 대기한 후에 전송을 시도하는 방식이다. 즉, 제일 작은 슬롯 값을 선택하는 리더가 채널을 선점하는 방식이다. 경쟁(contention)에서 실패한 리더는 다음 경쟁 상황에서 줄어든 만큼의 슬롯 값을 사용함으로써 다른 리더에 비해 우선순위를 가질 수 있다. 랜덤 백오프는 경쟁 윈도우의 크기에 의해 성능이 좌우된다. 경쟁 윈도우의

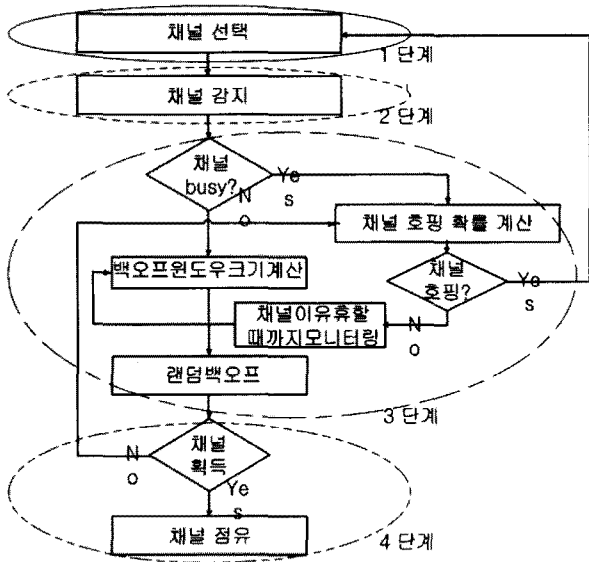


그림 1. 제안하는 알고리즘의 흐름도  
Fig. 1. Flow chart of the proposed algorithm.

크기가 클 경우 충돌 확률은 줄어들지만, 비어있는 슬롯들에 의해 오버헤드가 크다. 반대의 경우 빈 슬롯에 의한 오버헤드는 적지만 충돌 확률이 줄어든다. 따라서 리더의 수 등에 의한 가변적인 경쟁 윈도우 크기 조절이 필요하다.

리더는 호핑 시에 랜덤하게 채널을 선택하게 되는데, 사용 중인 같은 채널을 반복해서 선택하게 되는 것을 방지하기 위해 채널의 호핑 순서를 사전에 정하여 호핑하는 방식을 사용한다.

이와 같이 제안하는 알고리즘은 여러 가지 메커니즘이 복합적으로 구성되어 있다. 그림 1은 이러한 각각의 메커니즘들의 흐름도이다. 그림 1에서 보듯이 동작 과정은 동작 순서에 따라 크게 네 가지 단계로 구성된다. 첫째 단계는 각 리더가 사용할 채널을 설정하는 단계이다. 리더들은 처음 동작을 시작할 때 랜덤하게 채널을 선택한 이후 제안하는 알고리즘에 따라 자신들이 사용할 채널을 결정하게 된다. 둘째 단계는 리더들이 결정한 채널을 감지하는 단계이다. 채널 감지를 위해 LBT 방식이 적용되며 채널이 현재 사용되고 있는지 여부를 판단하는 단계이다. 셋째 단계에서는 채널 감지를 통하여 채널의 사용 여부를 판단하게 되고 그에 따라 채널을 호핑할 것인지 또는 현재 채널에 대기할 것인지를 결정하는 단계이다. 마지막 단계는 최종적으로 리더가 채널을 점유하여 사용하는 단계이다. 그림 2에서는 8개의 리더와 4개의 채널이 존재하는 환경에서 제안하는 알고리즘의 동작 원리를 나타내는 하나의 예를 보여주고 있다. 붉은 색 화살표는 채널의 호핑을 나타내고,

파란 색 화살표는 백오프 윈도우의 크기가 다음 경쟁 시에 그대로 옮겨가는 것을 나타낸다. 각 단계에 대한 자세한 설명은 다음과 같다.

1. 채널 선택 단계

리더는 태그와의 통신을 하기 위해서는 어느 채널을 사용할지 먼저 선택해야 한다<sup>[9]</sup>. 기본적으로 확률적 채널 호핑 알고리즘에서는 리더가 채널 점유를 마친 후에 타 채널로 호핑하는 방식을 사용한다. 그리고 리더는 다른 리더에 의해 채널이 사용되고 있는 경우, 호핑 확률에 따라 타 채널로 호핑한다. 호핑 확률을 구하는 방법은 이후에 자세히 설명한다. 채널 호핑시 선택되는 채널은 랜덤하게 선택되는 것을 기준으로 한다. 그러나 단순히 랜덤으로 채널을 선택하는 것은 효율적이지 못하다. 채널의 이용률이 높은 경우, 리더는 비어있는 채널을 찾기 위해 다수의 채널 호핑이 이루어질 수 있기 때문이다. 이 때 채널을 단순히 랜덤하게 선택하여 호핑할 경우 사용 중인 같은 채널을 여러 번 확인하게 되는 비효율적인 문제가 발생할 수 있다. 또한, 모든 채널이 사용 중인 경우에는 비어있는 채널을 찾지 못하고 계속해서 호핑할 가능성이 있다. 이러한 문제를 해결하기 위해 확률적 채널 호핑 알고리즘에서는 채널 호핑을 처음 시작할 때 현재 채널 이외의 모든 채널에 대해 호핑 순서(Hopping Sequence)를 정하고 그 순서대로만 호핑한다. 호핑 순서는 채널 호핑이 새로 시작될 때마다 랜덤하게 정해지게 된다. 이와 같은 방식을 사용하면 앞에서 언급한 같은 채널을 반복해서 방문하는 문제나 채널 루핑(looping) 현상을 해결하고, 모든 채널을 한 번씩만 검색하는 것이 가능하다. 이 순서대로 모든 채널을 검색해도 비어있는 채널을 찾지 못한 경우에는 원래 채널에 대기하여 채널 점유를 시도하게 된다.

2. 채널 감지 단계

리더는 선택된 채널로 호핑한 이후, 그 채널을 LBT 방식으로 감지한다. LBT는 RFID 시스템의 무선 설비 기준에 따라 5ms 동안 이루어지며, 채널의 사용 유무를 판단하기 위해서는 최소한 5ms의 시간이 필요하다<sup>[8]</sup>. 확률적 채널 호핑 알고리즘에서는 채널이 사용 중이라고 판단되는 경우와 비어있다고 판단되는 경우에 각각의 메커니즘이 존재한다. 각각의 메커니즘은 다음 단계에서 설명한다.

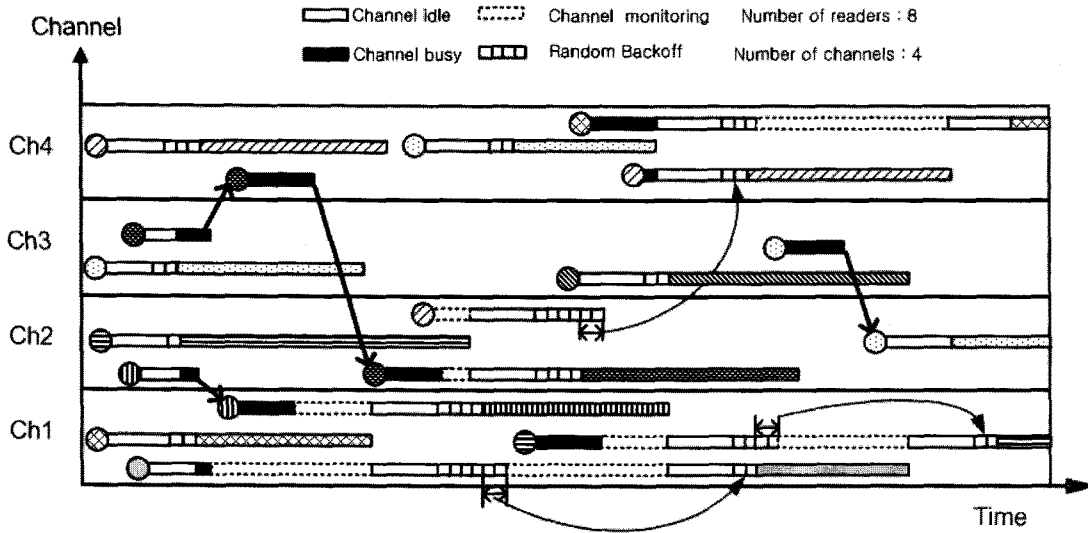


그림 2. 제안하는 알고리즘의 동작 원리 (예)  
 Fig. 2. Operation of the proposed algorithm (Example).

3. 채널 호핑 및 채널 사용 준비 단계

가. 채널 호핑 확률 적용 단계

리더가 채널을 감지하여 사용 중이라고 판단될 경우 적용되는 단계이다. 이 때 리더는 두 방식 중 하나를 선택하게 된다.

첫째, 현재 채널에서 채널이 유희해질 때까지 기다리는 방식이다. 채널이 유희해지면 리더는 채널 점유 시도를 할 수 있다. 이 경우 리더는 채널을 사용하고 있는 리더의 전송이 끝날 때까지 기다려야 한다. 타 채널이 비어있을 확률이 클 경우 사용 중인 채널에서 대기하는 것은 비효율적일 수 있다. 또한, 대기 이후 다른 리더와의 경쟁에서 채널을 점유하지 못하게 된다면 또 다시 채널이 유희할 때까지 기다리게 되어 대기 시간이 더욱 길어질 가능성도 있다. 따라서 채널이 유희해질 때까지 그 채널에서 무조건 대기하는 것은 효율적이지 못한 방식이라고 할 수 있다.

둘째, 현재 채널이 아닌 타 채널로 호핑하는 방식이다. 리더는 다른 채널의 사용 정보에 대해 알 수가 없으므로 랜덤하게 호핑하는 방식이 사용된다. 이 경우 타 채널이 비어있을 확률이 높으면 비어있는 채널을 찾아 호핑하는 것이 유리할 것이다. 그러나 리더의 사용 요구량이 높아 비어있는 채널이 존재할 가능성이 낮은 경우에는 비어 있는 채널을 찾지 못하고 계속해서 호핑만 반복되는 경우가 발생할 수 있다. 또한, 리더가 호핑하는 데에는 일정 시간의 오버헤드가 필요하기 때문에 무조건 호핑하는 방식은 비효율적이라고 볼 수 있다.

이와 같이 채널이 사용 중인 경우 무조건 현재 채널에 대기하거나 타 채널로 호핑하는 방식은 좋지 않다. 상황에 따라 두 가지 방식 중 더 적합한 하나를 선택한다면 더 좋은 성능을 보일 수 있을 것이다. 따라서 채널이 사용 중인 경우 채널의 호핑 확률을 계산하여 적용한다. 이 때 채널의 호핑 확률을 계산하기 위해 채널의 리더밀도(리더의 수/채널의 수)에 따른 특징이 중요하게 작용한다.

채널의 리더밀도가 1이하일 경우 이러한 환경을 스파스(Sparse) 모드라고 하며, 같은 채널에서 다른 리더와의 경쟁이 많이 이루어지지 않는다. 따라서 LBT 이후에 백오프 경쟁에 의해 채널을 점유하게 될 확률이 크게 되며, 채널 점유에 실패하게 되더라도 타 채널이 비어있을 확률이 크다. 따라서 이 경우에는 같은 채널에 대기하는 것보다 타 채널로 호핑하는 것이 유리하다. 따라서 호핑 확률을 높게 설정하는 것이 좋다.

반대로 채널의 리더밀도가 1이상일 경우 이러한 환경을 밀집 모드 또는 덴스(Dense) 모드라고 하며, 다른 리더와의 채널 점유 경쟁이 많이 발생할 수 있다. LBT 이후에 백오프 경쟁에서는 채널 점유의 성공률이 비교적 낮으며, 타 채널로 호핑하더라도 비어있는 채널이 존재할 가능성이 낮다. 따라서 같은 채널에 대기하는 것이 타 채널로 호핑하는 것보다 유리할 수 있으며, 이 경우에는 호핑 확률을 낮게 설정하는 것이 좋다.

이와 같이 채널의 리더밀도에 따라 타 채널로의 호핑 확률을 동적으로 적용하는 것이 유리하며, 본 연구에서는 리더환경의 밀도를 측정하기 위해 백오프 경쟁에서

의 성공률과 채널의 이용률을 예측하여 확률 계산의 파라미터로 사용한다.

첫 번째 파라미터로 리더가 채널에서 백오프 중 다른 리더와의 경쟁에서 지게 될 확률(CLP : Contention Loss Probability)을 정의한다.

$$CLP(i, n) = (1 - \alpha) \cdot CLP(i, n-1) + \alpha \cdot CL(i, n) \quad (1)$$

where  $CL = \begin{cases} 1, \text{unsuccessful contention} \\ 0, \text{successful contention} \end{cases}$

식(1)은 지수 평균(Exponential Averaging) 방식을 사용하고 있는데, 이는 최근 값에 일정한 가중치를 두어 이전 값에 더해가는 방식이다. 이 방식은 단순한 산술 평균보다 리더의 수나 데이터 전송량이 변화하는 모바일 환경에 적응력을 보일 수 있다. 식 (1)에서  $i$ 는 리더의 ID를,  $n$ 은 계산 횟수를 의미한다. 또한, CL(Contention Loss)는 경쟁에 의해 얻어지는 현재 값을 의미하며 경쟁에서 이기면 0, 지게 되면 1값을 가지게 된다. 결과적으로 CLP는 리더 간의 경쟁에서 지게 될 확률을 나타낸다.

두 번째 파라미터로 식(2)와 같이 각 리더의 채널 사용 정도를 모니터링하여 계산된 채널 이용률( $U$  : Utilization)을 정의한다.

$$U(i, n) = (1 - \alpha) \cdot U(i, n-1) + \alpha \cdot U_i(i, n) \quad (2)$$

마찬가지로 모바일 환경에 적응력을 보일 수 있도록 지수 평균 방식을 사용한다.  $U_i$ 는 리더가 일정 시간(5초) 간격으로 측정된 채널의 이용률로, 채널이 사용 중인 경우 1, 채널이 유향한 경우 0 이 두 가지로 표현된다. 리더는 항상 자신이 속해있는 채널을 모니터링하여  $U_i$ 를 얻어내며, 이를 이용하여  $U$ 를 계산할 수 있다.  $U$ 는 채널이 얼마만큼 사용되고 있는지를 나타내는 것으로 리더환경의 밀도가 증가할수록 높은 값을 보이게 된다.

본 연구에서 호핑 확률(HP : Hopping Probability)은 현재 채널이 사용 중인 경우, 비어있는 채널을 찾기 위해 타 채널로 호핑하게 될 확률을 의미한다.

$$HP(i, n) = 1 - CLP(i, n) \cdot U(i, n) \quad (3)$$

리더의 호핑 확률은 CLP와  $U$ 가 클수록 작게 설정하는 것이 유리함을 앞에서 설명한 바 있고, 두 파라미터 모두 채널의 밀도를 대변할 수 있다. 따라서 CLP와  $U$ 가 증가함에 따라 HP는 감소하게 되고 따라서 HP를 식(3)과 같이 정의한다. 이와 같은 과정은 p-persistent

알고리즘에서 전송하고자 하는 데이터 량에 따라 최적의 처리율을 나타낼 수 있도록  $p$ 값을 선택해주는 과정과 유사하다고 할 수 있다.

#### 나. 채널 사용을 위한 백오프 단계

리더가 채널을 감지하여 유향하다고 판단될 경우 적용되는 단계이다. 이때에는 백오프 윈도우를 계산하여 랜덤 백오프 과정을 거치게 된다.

각 리더들은 둘 이상의 리더가 동시에 접속하여 충돌이 나는 것을 방지하기 위해 백오프 방식을 사용하게 된다. 그러나 경쟁 리더의 수가 많지 않은 경우 지나치게 큰 백오프 윈도우 크기를 선택하여 사용하는 경우 그 시간만큼 비효율적으로 낭비하는 경우가 발생할 수 있다. 따라서 채널의 이용률에 따라 백오프 윈도우 크기를 동적으로 정해주는 방식이 효율적일 수 있다. 따라서 백오프를 수행하기 전에는 항상 백오프 윈도우를 계산해주는 과정이 선행되며 그 후에 랜덤 백오프를 수행한다. 백오프 윈도우 크기(BWS : Backoff window size)는 식(4)와 같이 앞서 정의한  $U$ 를 이용하여 정해지게 된다.

$$BWS(i, n) = \text{Round}(\text{Max\_BWS} - \text{Min\_BWS}) \cdot U(i, n) + \text{Min\_BWS} \quad (4)$$

Max\_BWS와 Min\_BWS는 사전에 정의된 BWS의 최대값과 최소값을 의미한다. BWS는 이 두 값 사이의 정수로 정의된다. Round()는 반올림함수를 나타낸다.

#### 4. 채널점유 단계

랜덤 백오프를 통해 리더는 다른 리더들보다 작은 백오프 윈도우를 선택하게 되면 채널을 점유할 수 있다. 그러나 다른 리더가 먼저 채널을 점유하게 되면 앞에서 구한 채널 호핑 확률에 따라 타 채널로 호핑하게 된다. 호핑을 하게 된다면 앞에서 설명한 호핑 순서에 따라 호핑 채널이 결정된다.

### IV. 시뮬레이션

#### 1. 성능 분석 파라미터

밀집 리더 환경에서 사용되는 RFID 리더 충돌 알고리즘의 성능을 평가하기 위해 크게 두 가지 측면에서 성능을 측정하여 평가한다.

첫째, 채널에서 정상적으로 데이터 전송이 이루어진 비율, 즉, 처리율(throughput)을 측정한다. 이는 전체 시

간 중에서 채널의 유휴 구간과 충돌 구간을 제외한 시간의 비율을 의미한다. 처리율을 리더가 채널을 점유하여 사용하고자 하는 시간, 즉, 채널에서 전송이 되기 위해 시도되는 시간의 비율(offered load: G)과 비교하면 얼마나 효율적으로 데이터가 전송되고 있는지를 평가할 수 있다.

둘째, 리더가 데이터 전송을 위해 트리거 된 이후 채널 점유에 성공할 때까지 걸리는 시간(waiting time)을 측정한다. 이는 리더가 비어있는 채널을 얼마나 빠르게 검색하여 데이터 전송을 시도하는 데까지 걸리는 시간을 측정하는 것으로, 짧은 시간에 전송이 시작될수록 좋은 성능이 나타난다고 볼 수 있다.

이러한 두 가지의 성능 평가 항목에 의해 각 알고리즘의 성능을 평가한다.

### 2. 시뮬레이션 환경

알고리즘의 성능을 평가하기 위해 매트랩(matlab)을 사용하여 시뮬레이터를 구현하였다. 시뮬레이션에서 사용된 파라미터들은 표 4와 같다.

전송 후 대기 시간은 국내 무선 설비 기준에서 규정하는 값으로<sup>[8]</sup>, 리더의 채널 사용 이후에 다시 채널을 선점하기 위해 시도할 때까지 의무적으로 기다려야 하는 시간을 의미한다. 데이터는 푸아송 분포(Poisson Distribution)를 따른다고 가정하며, 경쟁 윈도우의 Max와 Min 값은 32와 8이 사용된다. CLP와 U를 계산하는데 사용되는  $\alpha$ 는 0.2이고, 리더는 충돌을 감지하지 못한다고 가정한다. G(offered load)는 모든 리더들의 채널 사용에 대한 시간 요구량을 의미한다.

표 4. 시뮬레이션 파라미터  
Table 4. Simulation parameters.

| 파라미터                  | 값                     |
|-----------------------|-----------------------|
| LBT time (ms)         | 5                     |
| 시뮬레이션 시간 (s)          | 10000                 |
| 전송 후 대기 시간 (s)        | 0.1                   |
| 데이터 arrival           | 푸아송 분포                |
| 리더의 서비스 시간 (s)        | 푸아송 분포 (average: 0.5) |
| CW(Contention Window) | Min : 8, Max : 32     |
| 채널 수 (개)              | 4, 8                  |
| 리더 수 (개)              | 4, 8, 16              |
| G                     | 0.1 ~ 1.0             |
| $\alpha$              | 0.2                   |

### 3. 성능 분석

알고리즘의 성능을 평가하기 위해 LBT만을 사용하

는 기존의 알고리즘과 확률적 채널 호핑 알고리즘의 성능을 비교한다. 확률적 채널 호핑 알고리즘은 채널 선택 단계에서 채널의 호핑 순서를 사전에 정하여 사용하는 방식의 성능 개선 효과를 분석하기 위해, 이 기능을 제외한 다른 기능은 확률적 채널 호핑 알고리즘과 동일하게 동작하는 알고리즘을 확률적 채널 호핑 알고리즘(단순 호핑)으로 정의하고, 각각 시뮬레이션 하였다. LBT만을 사용하는 알고리즘은 리더가 LBT를 사용하여 채널을 감지하고, 채널이 사용 중인 경우 채널에 대기하였다가 채널이 비게 되면 점유하여 사용하는 방식이다. 채널의 점유가 끝난 이후에는 다른 채널로 호핑하게 된다. 본 논문에서는 이 알고리즘을 기본 LBT 알고리즘이라고 정의한다. 기본 LBT 알고리즘에서는 백오프 알고리즘을 사용하지 않는다. 따라서 동시에 채널에 대기하는 리더수가 두 개 이상이면 충돌이 발생하여 처리율이 낮아진다. 또한, 실제로는 리더의 채널 점유요구가 충돌에 의해 실패한다면 해당 요구가 다시 이루어져야 하지만, 이로써 충돌에 의해 리더에 쌓이는 요구량이 포화상태(saturation)에 이르러 정확한 성능 비교가 어려워질 수 있다. 두 알고리즘의 성능 비교를 위하여 충돌로 인한 재요구는 발생하지 않는 것으로 가정한다.

#### 가. 처리율(Throughput)

그림 3~5는 리더의 처리율을 나타낸다. G값이 증가할수록 처리율은 선형적으로 증가하지만, 기본 LBT 알고리즘의 경우 채널이 4개이고, 리더의 수가 4개, 8개일 때 약 45%까지 증가한다. 리더가 8개인 경우 G가 1에 가까워질수록 약간의 감소가 이루어지는데 이는 G가 증가함에 따라 충돌이 많이 발생했기 때문이다. 채널이 8개, 리더가 16인 경우도 마찬가지로의 결과를 보여준다. 여기에서 처리율이 선형적으로 계속 증가하지 않는 이유는 리더간의 충돌 또는 일부 채널에서 발생하는 이용률 저하 때문에 나타나는 결과이다. 앞에서 언급했듯이 사용 중인 채널에 두 개 이상의 리더가 채널 점유를 위해 대기 중인 경우에는 반드시 충돌이 발생하기 때문에 낮은 처리율을 보이게 된다. 또한, 이용률이 저하되는 이유는 비어있는 채널이 있음에도 효율적으로 그 채널을 찾아내지 못하고 사용 중인 채널에 대기하는 경우가 발생하기 때문이라고 할 수 있다. 반면에 확률적 채널 호핑 알고리즘에서는 리더 수와 상관없이 처리율이 80% 이상으로 증가함을 볼 수 있다. 이는 본 알고리즘을 사용하면 리더가 비어있는 채널을 효율적으로 찾아

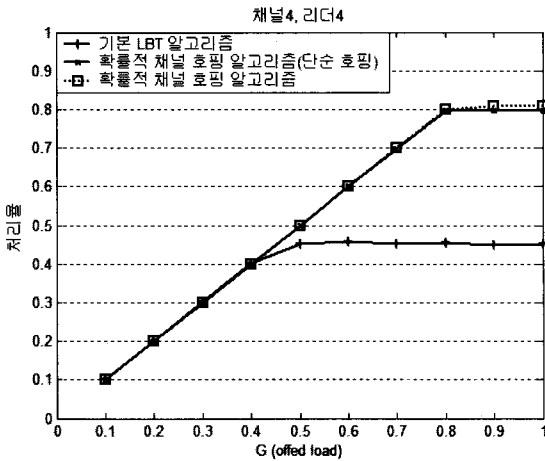


그림 3. 각 알고리즘의 처리율(채널수: 4, 리더수: 4)  
Fig. 3. Throughput of algorithms(4 channels, 4 readers).

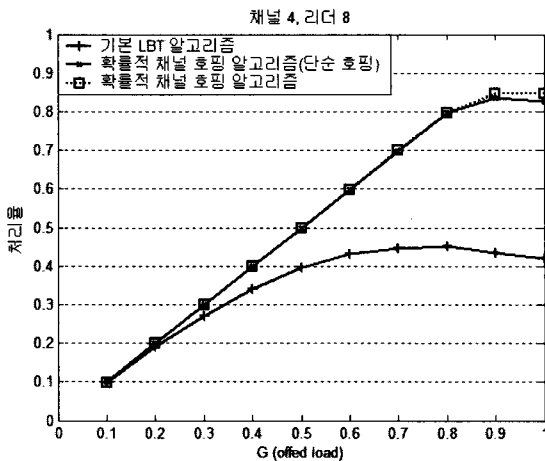


그림 4. 각 알고리즘의 처리율(채널수: 4, 리더수: 8)  
Fig. 4. Throughput of algorithms(4 channels, 8 readers).

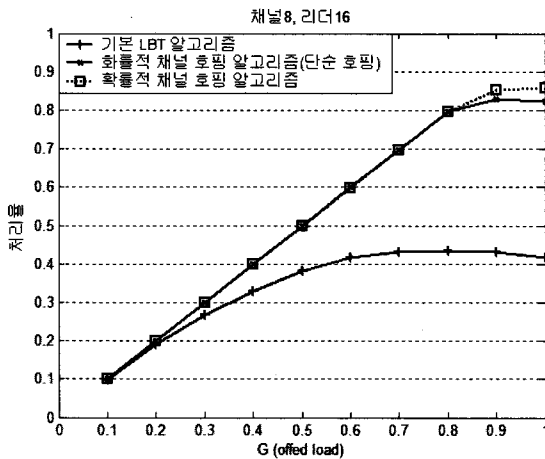


그림 5. 각 알고리즘의 처리율(채널수: 8, 리더수: 16)  
Fig. 5. Throughput of algorithms(8 channels, 16 readers).

가고 있기 때문에 높은 처리율을 얻을 수 있다고 해석할 수 있다. 또한, 채널 호핑 순서를 정하여 적용하는 방식이 단순하게 랜덤으로 호핑하는 방식보다 약 1% 정도 높게 나타나고 있다. 여기에서 처리율이 90% 이상으로 나타나지 않는 것은 채널 센싱을 위한 오버헤드, 채널 호핑을 위한 오버헤드, 그리고 약간의 리더 충돌에 의한 것으로 볼 수 있다.

나. 대기 시간(Waiting time)

그림 6~9은 리더의 전송 요구가 발생한 후 채널을 획득하여 태그와 정보를 주고받기 시작할 때까지의 대기 시간에 대한 누적 분포를 나타내는 그래프이다. 리더의 전송 요구량이 적을 경우와 많을 경우를 비교하기 위하여 G=0.4인 경우와 G=0.7인 경우에 대한 결과를 비교하였다. 다만 기본 LBT 알고리즘에서는 전송 요구량이 증가하더라도 처리율이 0.5에 채 미치지 못하므로 G가 0.7인 경우 리더는 전송 요구가 포화 상태에 이르게 된다. 따라서 전송 요구량 G가 0.4인 경우만을 도시하였다.

우선 리더의 요구량이 적은 G=0.4인 경우, 각 히스토그램은 0.1초 단위의 분포를 나타내고 있다. 먼저 기본 LBT 알고리즘의 경우 채널이 사용 중일 때 리더는 타 채널로 호핑하지 못하고 해당 채널에서 전송이 끝날 때까지 대기해야 한다. 이는 대기 시간이 충분히 길어질 수 있는 요인이 된다. 그래프를 살펴보면 채널이 4개, 리더가 4개, 8개인 경우와 채널이 8개, 리더가 16개인 경우 모두 약 3초까지 대기 시간이 분포하고 있음을 알 수 있다. 반면에 제안하는 알고리즘의 경우에는 채널이

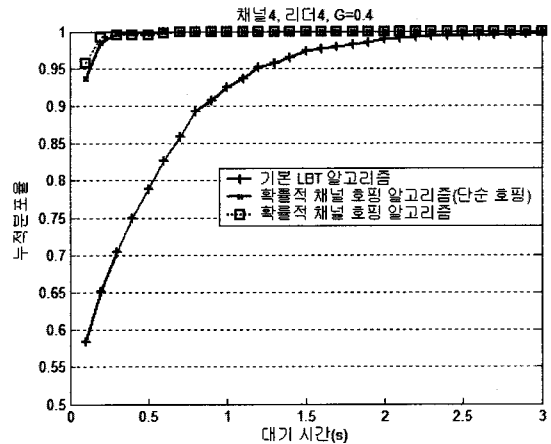


그림 6. 각 알고리즘의 대기 시간(채널수: 4, 리더수: 4, G=0.4)  
Fig. 6. Waiting time of algorithms(4 channels, 4 readers, G=0.4).



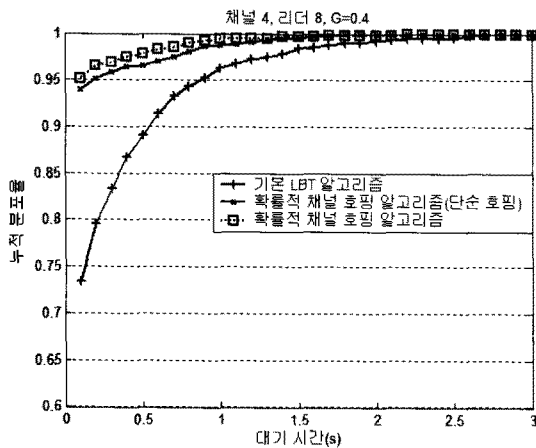


그림 7. 각 알고리즘의 대기 시간(채널수: 4, 리더수: 8, G=0.4)

Fig. 7. Waiting time of algorithms(4 channels, 8 readers, G=0.4).

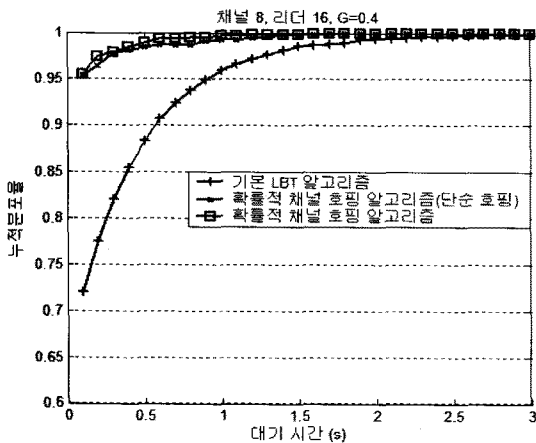


그림 8. 각 알고리즘의 대기 시간(채널수: 8, 리더수: 16, G=0.4)

Fig. 8. Waiting time of algorithms(8 channels, 16 readers, G=0.4).

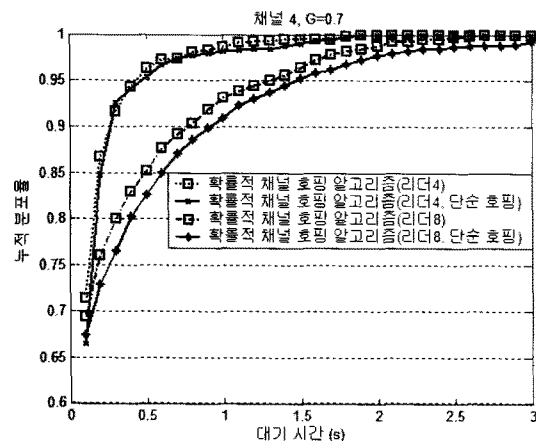


그림 9. 각 알고리즘의 대기 시간(채널수: 4, 리더수: 4, 8, G=0.7)

Fig. 9. Waiting time of algorithms(4 channels, 4, 8 readers, G=0.7).

사용 중인 경우 효율적으로 빠르게 비어있는 채널로 찾아간다. 따라서 대기 시간이 매우 짧게 나타난다. 그래프를 살펴보면 호핑 순서를 적용하는 경우와 적용하지 않는 경우 채널이 4개, 리더가 4개일 때 0.1초 이내에 각각 95%와 93% 이상, 8개일 때 95%와 94% 이상이 분포하는 것을 볼 수 있다. 그리고 리더수가 4개, 8개일 때 각각 0.5초 이내, 2초 이내에 모든 분포가 나타나는 것을 볼 수 있다. 채널이 8개, 리더가 16개인 경우에도 이와 유사하게 0.1초 내에 95% 이상 분포하고 2초 이내에 모든 분포가 나타나고 있다. 이는 확률적 채널 호핑 알고리즘이 기본 LBT 알고리즘에 비해 대기 시간 측면에서 매우 많은 성능 차이가 있음을 의미한다.

리더의 요구량이 많은 G=0.7인 경우, 기본 LBT 알고리즘의 waiting time은 리더 요구의 포화에 의해 무한히 길어질 수 있으므로 앞에서 언급한대로 시뮬레이션 결과에서 제외하였다. 그림 9는 확률적 채널 호핑 알고리즘에서 호핑 순서를 적용하는 경우와 적용하지 않는 경우 리더가 4개일 때와 8개일 때의 결과를 나타낸다. 이를 살펴보면 앞의 그래프에서보다 전반적으로 대기시간 분포가 길게 나타나고 있다. 각 히스토그램은 0.1초 간격의 분포를 나타낸다. 리더의 수가 4, 8개인 경우 모두, 0.1초 내에 약 70%가 분포하고 있으며 3초 이내에 모두 분포하고 있다. 기본 LBT 알고리즘의 대기 시간 분포가 무한히 길게 나타날 수 있음을 고려해보면 큰 성능 향상이 이루어졌음을 알 수 있다.

결과적으로 확률적인 호핑은 대기 시간 측면에서 성능 향상에 큰 역할을 하며 같은 처리율이라도 짧은 대기 시간을 갖는 확률적 랜덤 호핑 알고리즘이 훨씬 좋은 성능을 보인다고 판단할 수 있다.

## V. 결 론

최근 RFID의 사용 범위가 커지게 되면서 다양한 응용 분야에 걸쳐 사용되고 있다. 최근에는 단일 리더가 아닌 다수의 리더가 동작할 수 있는 환경을 요구하는 시스템들이 많아지는 추세이다. 국내에서는 RFID를 위한 대역폭이 상당히 작은 편이기 때문에 많은 리더들이 한정된 대역폭을 나누어 사용해야 하는 밀집 리더 환경에서 동작할 수 있는 RFID 시스템이 필요하다. 따라서 본 논문은 밀집 리더 환경 하에서 동작하는 RFID 시스템의 간섭을 회피하는 기술을 개발하고, 간섭회피 기술의 성능을 평가하는 시뮬레이터를 개발한다.

본 논문에서 제안한 확률적 채널 호핑 알고리즘은 기

존 LBT를 사용하는 방식에 구체적인 동작 기법을 적용한다. 호핑 확률을 정의할 때 타 채널이 사용 중인 확률을 예측하기 위해 채널의 이용률을 모니터링하고, 경쟁을 통해 채널을 획득할 확률을 연속적으로 계산한다. 경쟁 윈도우의 크기를 리더 수에 의한 채널 환경에 따라 동적으로 할당하여 경쟁 윈도우 크기에 의한 오버헤드를 줄인다. 또한, 사용 중인 같은 채널을 반복해서 선택하게 되는 것을 방지하기 위해 채널의 호핑 순서를 사전에 정하여 호핑하는 방식을 사용한다.

확률적 채널 호핑 알고리즘을 검증하기 위해 매트랩을 사용하여 다중 채널 환경에서 사용 가능한 시뮬레이터를 개발하였고, 확률적 채널 호핑 알고리즘과 기본 LBT 방식의 성능을 처리율과 대기 시간측면에서 비교하였다. 또한, 이 비교를 통해 제안하는 알고리즘의 성능을 평가하고 검증하였다.

확률적 채널 호핑 알고리즘은 차후 실제 리더에 탑재하여 활용이 가능할 것으로 보이며, 개발된 시뮬레이터 또한, 밀집 리더 환경에서 다양한 알고리즘을 검증하는데 활용할 수 있을 것이라 사료된다.

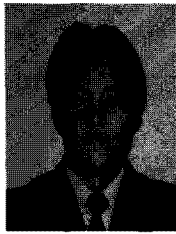
## 참 고 문 헌

- [1] EPC Global, "EPC Radio-Frequency identity Protocols class-1 generation-2 UHF RFID protocols for communications at 860MHz-960MHz version 1.0.9", EPC global, Sept. 2004.
- [2] S. R. Lee, C. W. Lee, "An Enhanced Colorwave Reader Anti-collision Algorithm in RFID system", in Proc. of ITC-CSCC 2006, pp. 145-148, July 2006.
- [3] H. Dai, S. Lai, H. Zhu, "A Multi-Channel MAC Protocol for RFID Reader Networks", in Proc. of Wicom 2007, pp. 2093-2096, Sept. 2007.
- [4] S. Jain, S. R. Das, "Collision Avoidance in a Dense RFID Network", in Proc. of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization, pp. 49-56, Sept. 2006.
- [5] J. H. Mo, H. S. W. So, J. Walrand, "Comparison of Multi-Channel MAC protocols", IEEE Transactions on mobile computing, vol. 7, pp. 50-65, Jan. 2008.
- [6] ETSI, "Electromagnetic compatibility and Radio spectrum Matters (ERM); Improved spectrum efficiency for RFID in the UHF Band", ETSI TS 102 562 V1.1.1, Mar. 2007.
- [7] 안동현, 양훈기, 양성현, 김영수, "RFID 리더의 다중접속 프로토콜", 한국통신학회, 한국통신학회지 정보와 통신, 제24권, 제5호, 124-134쪽, 2007년 5월.
- [8] 방송통신위원회 전파연구소, "방송·해상·항공·전기통신사업용 외의 기타업무용 무선설비의 기술기준", 전파연구소, 전파연구소고시 제2006-84호, 2006년 8월.
- [9] S. Anusha, "Minigating the Reader Collision Problem in RFID Networks with Mobile Readers", in Proc. of ICON 2005, Vol. 1, pp. 16-18, Nov. 2005.

저 자 소 개



목 치 영(학생회원)  
 2006년 아주대학교 전자공학과  
 학사 졸업.  
 2008년 아주대학교 전자공학과  
 석사 졸업.  
 <주관심분야 : WSN, RFID,  
 Traffic Engineering>

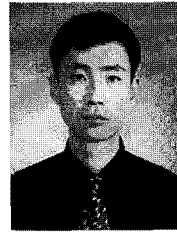


최 진 철(학생회원)  
 2005년 아주대학교 전자공학과  
 학사 졸업.  
 2007년 아주대학교 전자공학과  
 석사 졸업.  
 2007년 3월~현재 아주대학교  
 전자공학과 박사과정.  
 <주관심분야 : Ubiquitous Sensor Networks,  
 MANET, RFID, Traffic Engineering>



이 채 우(정회원)  
 1985년 서울대학교 제어계측  
 학사 졸업.  
 1988년 한국과학기술원  
 전자공학과 석사 졸업.  
 1995년 University of Iowa 박사  
 졸업.  
 1985년 1월~1985년 12월 (주)금성통신 연구원.  
 1988년 9월~1999년 3월 한국통신 선임연구원.  
 1999년 3월~2001년 9월 Lucent Technologies  
 Korea 이사.  
 2001년 9월~2002년 2월 한양대학교 겸임교수  
 2002년 3월~현재 아주대학교 전자공학과  
 부교수.

<주관심분야 : 광대역 통신망, Ubiquitous  
 networking, Traffic Engineering>



권 성 호(정회원)  
 1992년 (중)연변대학교  
 전자공학과 졸업.  
 2001년 대구대학교 정보통신  
 공학과 공학석사.  
 2005년 대구대학교 정보통신  
 공학과 공학박사.

2005년 4월~현재 한국전자통신연구원  
 선임연구원  
 <주관심분야: RFID, 지능형 컴퓨팅, 인지과학>



최 길 영(정회원)  
 1985년 경북대학교 전자공학과  
 학사 졸업.  
 1987년 경북대학교 전자공학학  
 석사 졸업.  
 1987년 2월~현재 한국전자통신  
 연구원 팀장

<주관심분야: 광대역통신망, 네트워크 QoS,  
 RFID/USN>



모 희 숙(정회원)  
 1989년 전남대학교 전산학과  
 학사 졸업  
 2002년 충남대학교 컴퓨터과학과  
 석사 졸업  
 1990년 2월~현재 한국전자통신  
 연구원 책임연구원.

<주관심분야: RFID/USN, Ad-Hoc, 위성관제>