공 학 석 사 학 위 논 문

RFID 태그 대량기록 성능의 향상을 위한 병행쓰기 기법

2011년 2월

부 경 대 학 교 대 학 원

I T 융 합 응 용 공 학 과

임 영 준

공 학 석 사 학 위 논 문

RFID 태그 대량기록 성능의 향상을 위한 병행쓰기 기법

지도교수 송 하 주

이 논문을 공학석사 학위논문으로 제출함.

2011년 2월

부 경 대 학 교 대 학 원

I T 융 합 응 용 공 학 과

임 영 준

임영준의 공학석사 학위논문을 인준함.

2011년 2월 23일

주 심 이학박사 신 상 욱 (인)

위 원 공학박사 권 오 흠 (인)

위 원 공학박사 송 하 주 (인)

목 차

1. 서 론 ····· 1

1.1. 연구의 필요성 ····· 1

1.2. 논문의 구성 ····· 3

2. 관련연구 ····· 4

2.1. RFID ····· 4

2.2. RFID 태그에 대한 쓰기 ····· 6

2.3. 태그 메모리 쓰기에 대한 다양한 프로토콜 ····· 7

2.3.1. 추정 쓰기 프로토콜 ····· 7

2.3.2. RFID 미들웨어의 쓰기 프로토콜 ····· 12

2.3.3. CQI 기반의 재처리 모델 ····· 12

3. 대량의 RFID 태그에 대한 병행쓰기 기법 ····· 19

3.1. 다수의 리더를 사용한 병행쓰기 기법 ····· 19

3.2. 개별 안테나에서의 interleaving 기법 ····· 20

3.3. 다수의 리더에서의 interleaving 기법 ····· 25

4. 실험 및 고찰 ····· 29

4.1. 실험 환경 ····· 29

5. 결 론 ····· 37

표 차 례

표 1 개별 태그의 reply 형태 ····· 7
표 2 단계별 처리 과정 ····· 9
표 3 불확실 상황 발생 후 읽기 작업의 결과값 ····· 12
표 4 UNSURE를 위해 미들웨어에 저장되는 테이블 ····· 14
표 5 병행쓰기 알고리즘 ····· 22
표 6 reply 형태의 분류 ····· 25
표 7 쓰기 작업의 결과 분류 ····· 26

PUKYONG NATIONAL UNIVERSITY
부경대학교

그림 차례

그림 1 본 논문에서 제안하는 시스템의 구조 ····· 2
그림 2 수동형 태그의 데이터 전송 ····· 6
그림 3 추정 쓰기 프로토콜의 동작 순서 ····· 11
그림 4 UNSURE를 포함한 트랜잭션 종료 요청 ····· 15
그림 5 불확실 리스트를 위한 CQI ····· 18
그림 6 불확실한 실행들에 대한 재처리 과정 순서 ····· 19
그림 7 시스템 구성도 ····· 20
그림 8 알고리즘의 도식화 ····· 24
그림 9 다수의 안테나 사용에 의한 충돌 발생 상황 ····· 27
그림 10 단순 쓰기와 interleaving 쓰기의 성공률 비교 ····· 32
그림 11 단순 쓰기 작업의 쓰기 시도 횟수 조정을 통한 성공률 비교 ···· 33
그림 12 interleaving 값에 따른 'WRERR'과 'NOTAG'의 개수 ····· 34
그림 13 interleaving 값의 따른 리더 2대의 성공률 ····· 36

PUKYONG NATIONAL UNIVERSITY 부경대학교

RFID 태그 대량기록 성능의 향상을 위한 병행쓰기 기법

임 영 준

부경대학교 대학원 IT융합응용공학과

요 약

RFID는 유비쿼터스 환경을 구축하기 위한 핵심적인 기술이다. 특히 RFID가 널리 사용됨에 따라 대량의 태그에 대해 데이터를 신속하게 기록하는 기술이 필요하다. 한편 태그 메모리 내에 데이터를 쓰기 위한 작업은 무선 주파수 기술이라는 한계로 인해 부정확한 정보가 기록되어지기 쉽다. 이를 위해 본 논문에서는 대량의 RFID 태그에 데이터를 기록함에 있어 효율적으로 태그 데이터를 쓰기 위한 기법을 제시한다. 본 기법을 통해 미들웨어에 연결된 다수의 리더를 이용하여 태그에 대한 병행기록(쓰기) 작업을 한다. 다수의 태그에 대한 쓰기 작업을 리더별로 분산시키고 분산된 태그들에서 다시 그룹을 지어 쓰기 작업을 실시한다. 이를 통해 대량의 쓰기 작업에 대한 실패율을 줄이고 태그에 쓰인 데이터에 대한 정확성을 확보한다.

Parallel Writing Scheme to Enhance Massive RFID Tag Write Performance

Young Jun Lim

Department of IT Convergence and Application Engineering,

The Graduate School,

Pukyong National University

Abstract

RFID Technology plays an important role in building a ubiquitous environment. As RFID tags are widely used, it is necessary to enhance bulk tag writing. However, tag memory writing does not provide good performance because of the limitations of radio frequency technology. In this thesis, we propose parallel writing scheme to enhance the performance of massive RFID tag writes . In this scheme, parallel writing operation is performed by multiple readers connected to a middleware. Write operations are performed by readers and antennas by tag groups. Proposed scheme can reduce the errors and ensure the integrity of the tag data in writing.

1. 서 론

1.1. 연구의 필요성

RFID(Radio Frequency Identification)기술은 무선 주파수를 이용하여 태그의 데이터를 읽어 사용하는 기술이다[1]. 안테나를 통해 인식되는 다양한 형태의 태그들을 사물에 부착하여 사물에 대한 정보 획득 및 식별, 관리 등이 가능하도록 응용하여 사용할 수 있다. 예를 들어 공정관리나 가축 관리. 그리고 각종 교통 지불 시스템 등에 이용될 수 있다.

이러한 RFID 기술에서 태그의 데이터는 중요히 여겨진다. RFID는 태그 내의 데이터를 사용하는 기술이기 때문에 태그 내의 데이터가 부정확하거나 없다면 기술의 사용이 불가능하기 때문이다. 이러한 태그의 데이터를 얻는 방법은 다양한 태그 종류 중 수동형(passive) 태그의 경우 다음과 같은 과정이 필요하다[2]. 먼저, 리더의 안테나 범위에 태그가 들어오게 되면, 안테나에서는 해당 태그의 데이터를 얻기 위해 무선 주파수를 전송한다. 다음으로 무선 주파수를 태그 내의 전원으로 변경하여 태그 내의 데이터를 다시 안테나로 전송하게 된다. 이러한 일련의 과정을 통해 리더는 태그의 데이터를 인식하게 되며, 인식된 데이터를 유용하게 사용할 수 있게 되는 것이다. 또한 리더는 태그의 읽기뿐만 아니라 쓰기 또한 가능하다. 태그에 대한 쓰기를 위해서는 읽기보다 가까운 위치에 태그가 존재하여야 하며, 리더는 안테나를 통해 사용자가 태그에 쓰고자 하는 데이터를 무선 주파수를 통해 쓰기 작업 실시한다.

하지만, 위와 같은 태그의 쓰기 작업 시 문제점이 발생할 수 있다. 리더와 수동형 태그의 경우 그 인식 범위는 10미터 내이다[3]. 이 범위 내에

위치하지 않는다면 태그에 대한 읽기나 쓰기 동작이 제대로 수행되지 않는다. 또한 다수의 태그가 협소한 공간에 위치할 경우 태그들 간의 충돌로 인해 인식률은 낮아진다[4]. 그뿐만 아니라 리더의 범위 내에 태그가 있다고 할지라도 RFID 기술이 무선 주파수를 기반으로 하는 기술이므로 모든 태그가 리더에 읽혀지지 않는다[5]. 그리고 리더에서 단일 태그에 대한 쓰기 작업은 실패할 확률이 낮으나 다수의 태그에 대한 쓰기 작업은 실패할 확률이 높다. 그 이유는 리더에서 다수의 태그에 대한 접근은 무선인식 기술을 바탕으로 하므로 간섭이나 회절과 같은 전파측면에서의 문제와 태그들 간의 상호간섭이 발생할 수 있기 때문이다. 또한 쓰기 작업을 하고자 하는 태그들이 좁은 간격으로 붙어 있는 경우가 발생할 수 있으며, 이와 같은 문제로 인해 쓰기 작업의 실패가 발생하며, 이러한 실패로 인해 태그 내에 데이터가 정확하게 써지지 않는 문제가 발생할 수 있다.

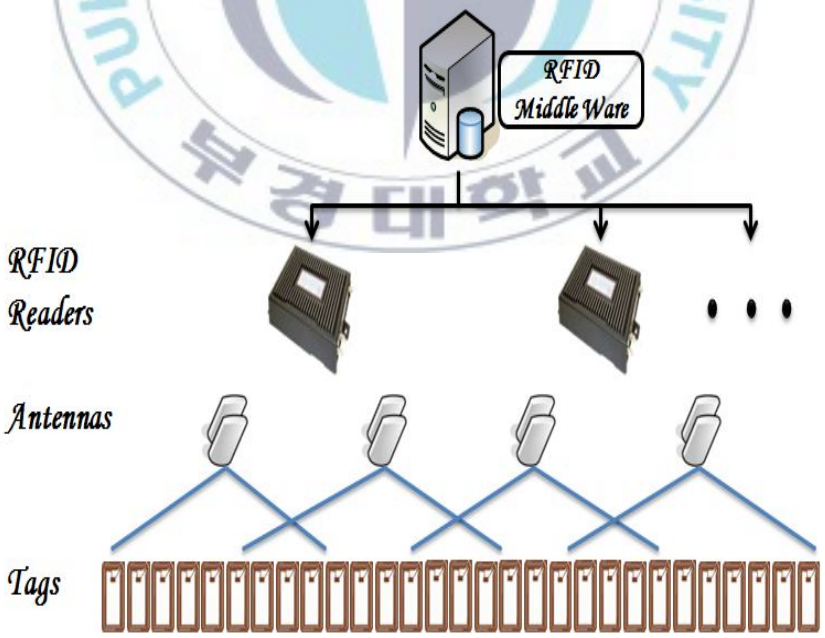

그림 1 본 논문에서 제안하는 시스템의 구조

본 논문에서는 이러한 다수의 태그에 대한 일련의 쓰기 작업에 대한 문제점을 제시하고 이를 개선하기 위해 [그림 1]과 같은 시스템을 사용하는 방법을 제안한다. 다수의 태그에 대한 쓰기 작업을 다수의 리더를 통해 실행하며, 보다 효율적인 리더의 사용과 쓰기 작업의 실패 확률을 줄이기 위한 기법을 제시한다.

1.2. 논문의 구성

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보고, 3장에서는 본 논문에서 제안하는 다수의 리더를 사용한 쓰기 기법을 제시한다. 4장에서는 제안된 기법에 대한 실험 및 고찰에 대해 알아보며, 5장에서는 결론으로 끝을 맺는다.

PUKYONG NATIONAL UNIVERSITY 부경대학교

2. 관련연구

2.1. RFID

RFID(Radio Frequency IDentification)는 전자기장 신호들의 탐지를 바탕으로 한 무선 센서 기술이다[6]. 보통의 RFID 시스템은 세 가지로 구성되는데 안테나나 코일, 디코더가 포함되는 트랜시버와 전기적으로 유일한 정보로 프로그램 되어 있는 RF 태그로 대표되는 트랜스폰더로 나뉜다. 이 중 안테나에서 방출되는 무선 신호는 태그에 데이터를 쓰거나 읽을 수 있도록 작동될 수 있다. 또한 안테나는 태그와 트랜시버 사이에 통신을 설정한다. 그리고 트랜시버는 데이터 획득을 책임진다. 안테나는 보통 리더라는 이름으로 트랜시버와 디코더를 합쳐서 부르게 된다. 이러한 리더는 고정형 리더와 핸드 핼드형 리더 등의 종류가 있다. 이러한 리더로부터의 무선 주파수는 약 100 피트부터 그 이상의 범위에서 작동할 수 있다.

RFID 시스템에서 태그는 리더와 함께 중요한 역할을 담당한다. RFID 태그는 크게 수동형 태그와 능동형 태그로 나뉜다[6]. 능동형 태그의 경우 내부에 자체 전력원을 가지고 있으며 리더에서 태그로의 신호 세기가 낮다. 하지만 태그에서 리더로의 신호세기는 강하다. 능동형 태그는 주로 넓은 범위에 RFID 시스템을 적용하기 위해 사용되며 태그 내의 저장 공간 또한 크다. 이와 반대로 수동형 태그의 경우 자체 전원이 없어 리더로부터 수신되는 신호를 바탕으로 전원을 생성해 내며 리더의 범위 내에 포함될 경우에만 발견된다. 능동형 태그와는 반대로 리더로부터 태그로의 신호 세기는 강한 반면 태그로부터 리더로의 신호 세기는 약하다. 또한 통신 범위가 작으며 데이터 저장 공간 또한 작다. 이 외에 주파수 대역별로 각 국가간에 사용되는 태그들의 차이가 있으며 수동형 태그와 능동형 태그의 중간

형태인 반수동형 태그가 있다.

이러한 RFID 태그에 정보를 쓰기 위해서는 리더의 안테나 범위 내에 태그가 위치하여야하며, 리더로부터 보내지는 신호에 의해 태그 내에 데이터를 쓸 수 있다. 이때 태그 간 간섭 현상이나 리더의 안테나들 간의 전파 측면에 문제에 의해 태그에 대한 데이터 쓰기 능력은 달라질 수 있다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

2.2. RFID 태그에 대한 쓰기

RFID 시스템에서 태그 내부의 메모리에 사용자가 원하는 데이터를 쓰기 위한 방법 중 저주파수를 사용하는 패시프 태그의 경우를 살펴보면 다음과 같다[7]. [그림 2]에서 보듯 먼저 RFID 리더는 전자기적 신호를 태그에 브로드 캐스팅한다. 그리고 태그의 안테나는 리더로부터 신호를 받는다. 그리고 이를 태그 내의 캐패시터에 저장한다. 캐패시터에 에너지가 충분히 충전되게 되면 이를 코일에 전달한다. 그리고 태그의 코일은 태그의 정보가 포함된 인코딩된 무선 주파수를 리더로 전달하게 된다.

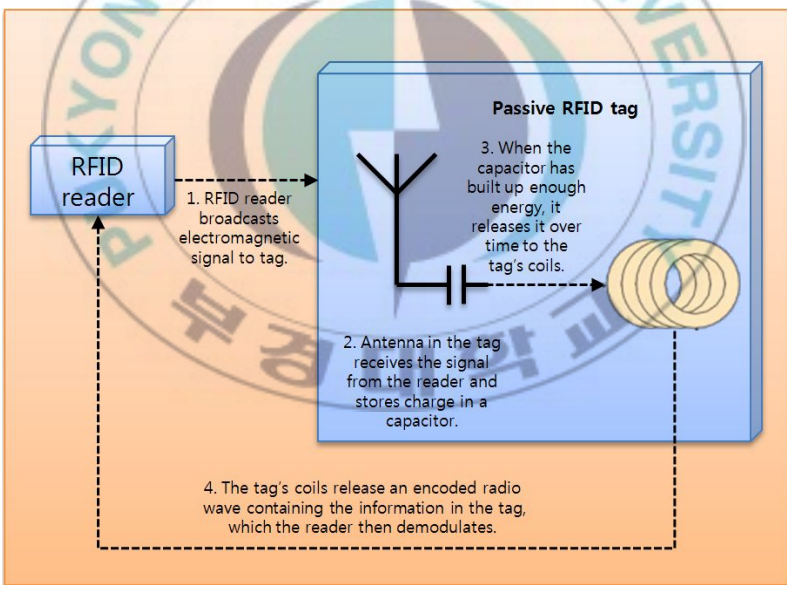

그림 2 수동형 태그의 데이터 전송

그리고 이와 유사하게 리더에서는 태그 내에 쓰고자 하는 정보를 포함한 전자기적 신호를 태그에 보내게 되고 태그는 이를 바탕으로 자신의 내부에 위치한 메모리의 특정 영역에 전달받은 데이터를 쓰게 된다.

2.3. 태그 메모리 쓰기에 대한 다양한 프로토콜

2.3.1. 추정 쓰기 프로토콜

태그에 대한 쓰기 작업의 불확실성을 제거하기 위한 방법으로 제대로 써지지 않았거나 부분적인 데이터만 쓰인 태그에 대한 쓰기 작업을 다시 시도하는 방법이 있다[8]. 이러한 방법은 미들웨어 레벨에서 리더로부터 얻어지는 쓰기 작업에 따른 태그의 에러 발생 메시지를 분석한다. 그리고 이러한 에러 메시지를 구분하여 부정확한 정보를 가지거나 쓰기 작업이 실패한 것으로 여겨지는 태그의 리스트를 유지하며 다른 리더에서 해당 태그가 발견될 경우 다시 쓰기 작업을 시도하는 방식이다. 이 방식은 태그에 대한 쓰기 연산을 세 가지 단계로 나누어 처리하는데 차례대로 Normal Phase, Reprocessing Phase, Aborting Phase이다. Normal Phase는 쓰기 연산이 미들웨어에 등록되면서 시작되게 된다. [표 1]에는 개별 태그로부터의 reply가 정리되어 있다.

표 1 개별 태그의 reply 형태

nww / reply 형태	0	n보다 작음	n
Success	X	X	성공
Temporarily Failed	실패	미완료	X
Permanently Failed	실패	미완료 실패	X
No Response	미완료	미완료	X

[표 1]에서 *nww*는 *Number of Written Words*로서 실제 태그에 쓰고자 하는 사용자가 태그에 쓰고자 했던 워드의 수를 의미하는 것이며 이 표에서는 태그로부터 자신에 쓰여진 워드의 개수로 리더로의 reply에 포함된 것이다. 즉, 0일 경우는 태그에 워드 단위로 데이터가 하나도 써지지 경우이며 n보다 작은 것은 원래 쓰고자했던 n 워드 전체가 써지지 않았다는 것이다. 그리고 n은 정상적으로 다 쓰인 경우를 의미한다. reply 형태의 경우 크게 4개로 나뉜다. *Success*는 성공한 것으로서 reply에 포함된 *nww*는 당연히 n이다. 그리고 *Temporarily Failed*의 경우 불충분한 전원 공급과 같은 환경적 요인에 의한 일시적인 실패를 의미하는 것으로 nww가 0일 경우는 실패. n보다 작을 경우는 부분적인 쓰기 작업으로 미완료로 분리한다. Permanently Failed의 경우 영구적인 실패로서 태그 메모리 초과나 태그에 대한 접근 권한이 없을 경우 발생하는 것이다. Temporarily Failed와 마찬가지로 nww가 0일 경우는 실패로 분류한다. 하지만 nww가 n보다 작을 경우에는 미완료 실패로 분류하는데 이는 Reprocessing Phase로 해당 태그 정보를 보내기 위해 정의된 것이다. 다음 reply 형태인 No Response는 쓰기 작업에 대한 reply를 리더가 받지 못한 경우를 의미한다. 따라서 이 상태에서는 리더에 의한 쓰기 작업의 실행 여부를 알 수 없다. 주로 충돌, 다수의 리더에 의한 중복된 접근, 태그 주변에 장애물, 반사 등으로 인해 발생 할 수 있다.

표 2 단계별 처리 과정

Normal Phase
1. Sends a write message to the reader when the tag is observed at specified reader. 2. If the result is either *Success* or *Fail*, return the result to

the client and stop.

3. Else if the result is *Incomplete*, go to the reprocessing phase

4. Else if the result is *Failed-as-incomplete*, go to the aborting phase

Reprocessing Phase

1. Send *Presumed Completed* to the client

2. Reprocess the operation by sending the write message to the reader when the tag is re-observed at any reader.

3. If the result is *Success*, send *Success* to the client and stop.

4. Else if the result is *Incomplete*, continue the reprocess

5. Else if the result is *Failed-as-incomplete*, go to the aborting phase

Aborting Phase

1. Send *Presumed Failed* to the client

2. Send a write message of initial data to the reader when the tag is re-observed at any reader.

3. If the result is *Success*, send *Failed* to the client and stop.

4. Else if the result is *Incomplete*, continue the reprocess

[표 2]에서 정의된 과정을 살펴보면 다음과 같다. 먼저, Normal Phase는 특정 리더 범위에 태그가 발견되게 되면 리더는 쓰기 메시지를 전송한다. 그리고 쓰기 작업의 결과를 reply를 통해 분석하는데 Success, Temporarily Failed, Permanently Failed, No Response로 나눌 수 있고 이는 클라이언트에게 전달된다. reply를 분석하여 미완료일 경우에는 Reprocessing Phase로 해당 태그 정보를 보내어 쓰기 작업을 재시도하며

미완료 실패일 경우는 Aborting Phase로 태그 정보를 보내게 된다.

Reprocessing Phase에서는 Presumed Completed라는 메시지를 클라이언트로 보내게 된다. 이는 쓰기 작업이 완전한 완료가 아닌 완료로 추정이 된다고 알려주는 형태이다. 그리고 쓰기 작업이 미완료된 태그가 다른 리더에서 발견이 될 경우 해당 태그에 대한 쓰기 메시지를 보내는 명령을 재처리한다. 만약 결과가 Success일 경우는 클라이언트에게 성공했다는 메시지가 전달되고 정지된다. 이와 다르게 태그의 reply가 미완료일 경우에는 쓰기 작업의 재시도가 다시 이루어지고 미완료 실패일 경우에는 Aborting Phase로 진행된다.

Aborting Phase는 클라이언트에게 Presumed Failed를 전송하면서 시작된다. 그리고 해당 태그가 어느 리더에서 발견될 경우 초기 데이터를 쓰기 위한 쓰기 메시지를 전송한다. 결과가 Success일 경우는 Success 메시지를 클라이언트에게 보내고 정지되며, 미완료일 경우 재처리를 진행한다.

이러한 방식은 해당 태그에 대한 정보를 미들웨어 레벨에서 관리하며 쓰기 작업에 대한 태그의 reply를 분석하여 성공과 실패를 추정하여 클라이언트에게 보고한다. 그리고 재시도 방식을 통해 태그에 대한 쓰기 작업을 완료하려고 하며 쓰기 작업이 불가능한 태그에 대해서는 초기 데이터 값으로 변환하려는 쓰기 시도 또한 하고 있다. [그림 3]은 위와 같은 과정을 도식화한 것이다.

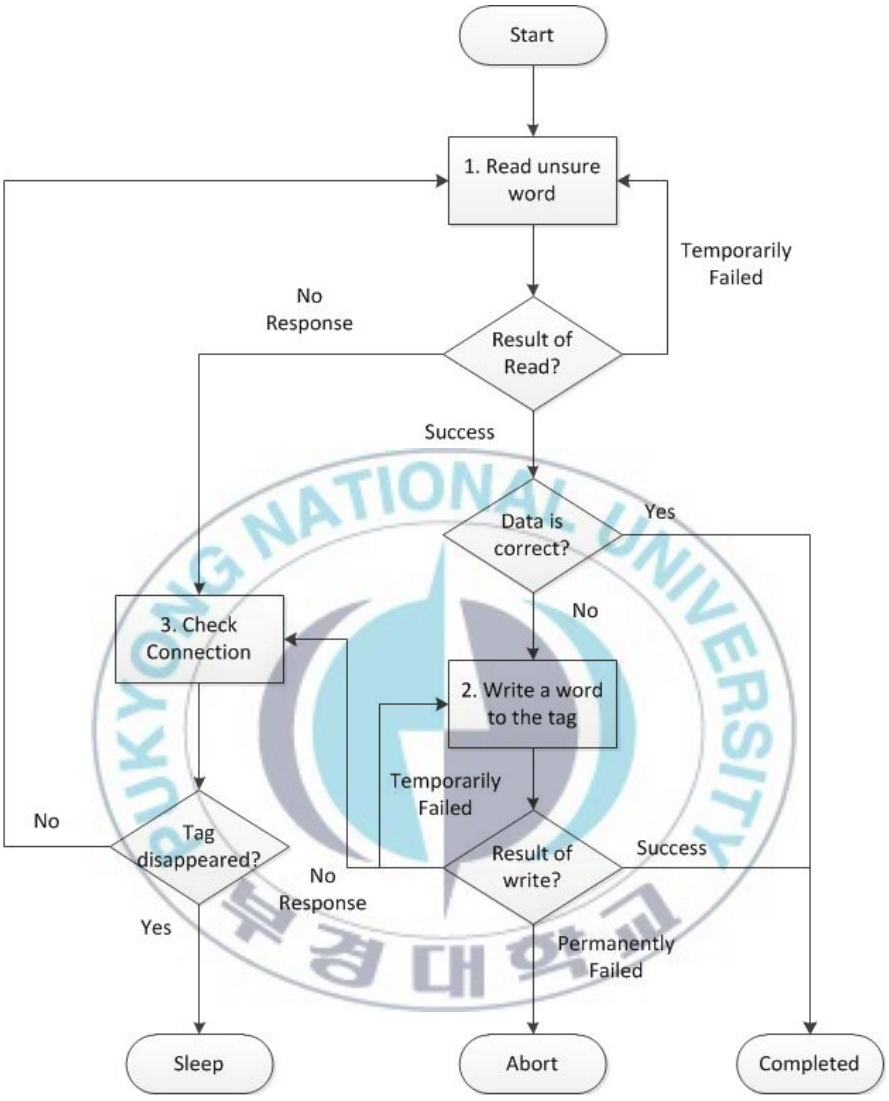

그림 3 추정 쓰기 프로토콜의 동작 순서

2.3.2. RFID 미들웨어의 쓰기 프로토콜

미들웨어 레벨에서 태그에 대한 쓰기 작업을 보장하기 위한 기법으로 트랜잭션을 이용한 방식이다[9]. RFID가 무선 주파수를 사용한 기술임으로 인해 발생되는 무선 단절 문제가 있으며 이로 인해 메시지 유실의 상황이 발생하게 된다고 정의한다. 그리고 이러한 상황을 해결하기 위해 불확실(이하 UNSURE) 리스트와 준완료 상태를 통한 트랜잭션의 일관성을 보장하는 기법을 제안하고 있다.

먼저, 리더에서 태그에 대한 쓰기 작업을 수행한 후 태그에서 리더로 전달되는 오류는 다양할 수 있다. 이때 태그가 리더의 인식 영역에서 사라진 경우와 태그로부터의 reply를 리더가 받지 못한 경우를 태그의 UNSURE 상태로 정의한다. 이를 통해 리더는 쓰기 작업을 수행한 후, 태그가 사라졌거나 태그로부터 reply를 받지 못한 경우 미들웨어에 UNSURE 메시지를 반환한다.

미들웨어에서는 이러한 UNSURE 메시지를 받게 되면 다음과 같이 처리한다. 먼저 UNSURE 상황에 대한 분석을 수행하는데 [표 3]과 같이 정의할 수 있다.

표 3 불확실 상황 발생 후 읽기 작업의 결과값

UNSURE 발생 상황의 종류	추가적인 읽기 작업 시 결과값
리더의 쓰기 메시지가 유실된 경우	태그에 쓴 정보와 불일치
태그 기록 성공 후 태그의 reply 메시지가 유실된 경우	태그에 쓴 정보와 일치
태그 기록 실패 후 태그의 reply 메시지가 유실된 경우	태그에 쓴 정보와 불일치
태그가 리더의 인식 영역에서 사라진 경우	읽기 reply 불가

[표 3]과 같이 정의될 수 있는 다양한 형태로 UNSURE 상황이 정의될 수 있다. 이러한 UNSURE 상황으로 인해 태그에 대한 쓰기 작업이 미완료일 경우를 트랜잭션의 개념으로 보고 준완료의 개념을 통한 방법을 제시한다.

먼저, 쓰기 작업에서의 완벽한 트랜잭션의 개념은 미들웨어에서 쓰기 작업을 태그에 대해 시행하게 되고 완료가 되면, 애플리케이션이 미들웨어에 트랜잭션의 종료 요청을 알리게 되고 미들웨어는 리더에 대한 자원을 해제하고 트랜잭션이 되는 것이다. 이때 애플리케이션이 트랜잭션에 대한 종료를 요청했을 때, 해당 트랜잭션 상에서 UNSURE 상태를 갖는 태그 정보에 대한 불일치를 해결하기 위해서 UNSURE 상태의 작업과 이를 포함하는 트랜잭션 정보를 아래의 표와 같은 테이블에 저장한다. [표 4]는 이러한 UNSURE 상태를 저장하기위해 미들웨어 저장되는 테이블의 형식을 보여준다.

표 4 UNSURE를 위해 미들웨어에 저장되는 테이블

TABLE NAME	FIELD NAME	TYPE	설명
TRANSACTION_INFO_LIST	TID	TRANSACTION_ID_TYPE	트랜잭션 ID
	Notification URI	List<URI>	트랜잭션의 수행 결과를 통지 받을 클라이언트 목록
	spec	spec	트랜잭션의 작업을 정의하는 CCSpec 명세
	TIME_STAMP	TIME_STAMP	다음 작업 초과 시간
	state	state	트랜잭션의 상태

			REQUEST ACTIVE PRECOMMIT
UNSURE_LIST	TAGID	String	태그 식별자
	TID	TRANSACTION_ID_TYPE	트랜잭션 식별자
	field	String	기록할 정보의 위치
	data	String	기록할 정보

애플리케이션이 트랜잭션의 종료를 요청한 경우 UNSURE 상태의 작업을 고려하여 다음과 같은 작업을 수행한다.

1. UNSURE LIST를 조회하여 해당 트랜잭션과 관련된 UNSURE 상태의 작업이 있는 지 확인한다.

2. UNSURE 상태의 작업이 없다면 TRANSACTION_INFO_LIST에서 해당 트랜잭션을 제거하고 애플리케이션에게 COMMIT을 보고 한다.

3. UNSURE 상태의 작업이 있다면 TRANSACTION_INFO_LIST에서 해당 트랜잭션의 state를 'PRECOMMIT'으로 변경하고 애플리케이션에게 PRECOMMIT을 보고한다. UNSURE를 포함한 트랜잭션 종료 요청한다.

[그림 4]는 이러한 과정을 도식화한 것이다.

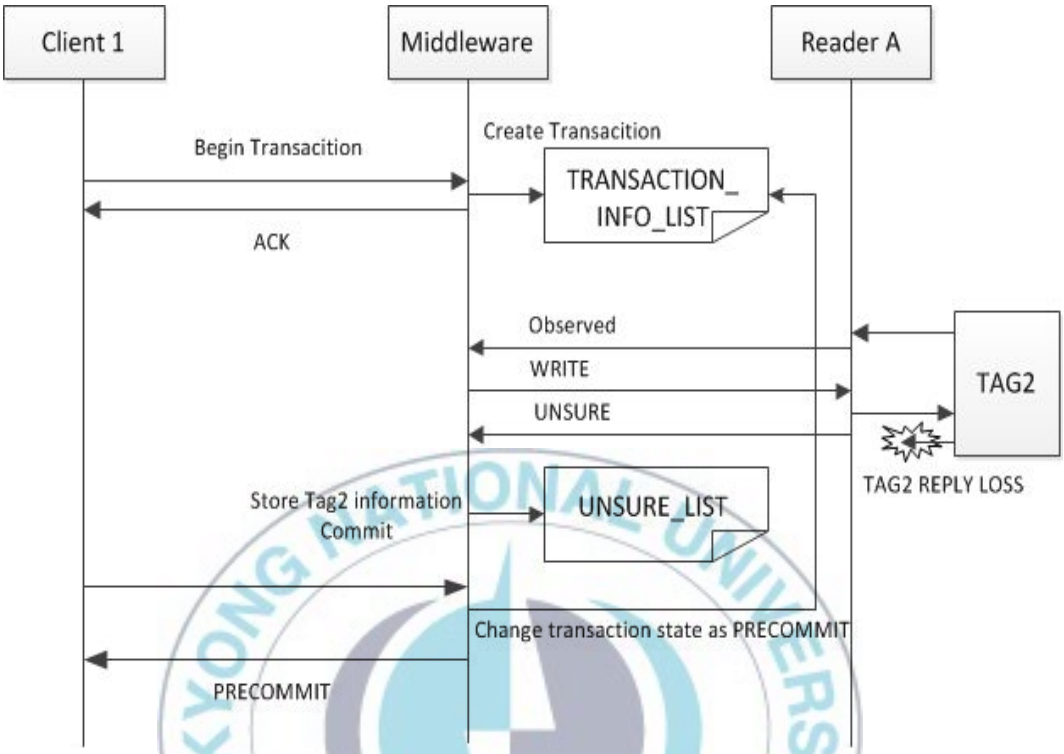

그림 4 UNSURE를 포함한 트랜잭션 종료 요청

다음과 같이 쓰기 작업이 제대로 완료되지 못했다고 여겨질 경우 PRECOMMIT을 해당 클라이언트에게 전송하게 되는데 트랜잭션의 준완료 후 UNSURE 처리는 다음과 같다.

1. 먼저 UNSURE_LIST를 조회하여 해당 태그가 UNSURE_LIST에 존재하는지 확인한다.

2. 만약 해당 태그가 UNSURE_LIST에 존재하지 않으면, 현재 등록된 무선인식 트랜잭션의 대상이 되는지 확인한다. 트랜잭션의 대상이 아닌 태그 이벤트는 무시된다. 해당 태그가 트랜잭션의 대상이면 트랜잭션의 작업을

수행한다.

3. 만약 해당 태그가 UNSURE_LIST에 존재하면, UNSURE 상태의 작업을 수행한다. 리더로부터 태그의 수행 결과를 받게 되면 미들웨어는 해당 작업을 UNSURE_LIST에서 제거한다.

4. 만약 UNSURE_LIST에 3의 트랜잭션이 더 이상 존재하지 않게 되면 미들웨어는 TRANSACTION_INFO_LIST에서 해당 트랜잭션을 제거하고 애플리케이션에게 COMMIT을 보고한다.

위와 같이 태그에 대한 쓰기 작업을 트랜잭션 개념으로 보고 준완료의 메시지를 클라이언트에게 전달하여 쓰기 작업을 실패의 알린다. 그리고 준완료 상태인 쓰기 작업 트랜잭션을 다시 시도함으로서 태그에 대한 쓰기 작업을 완료하는 기법이다.

2.3.3. CQI 기반의 재처리 모델

RF 통신 기반의 RFID 시스템에서 태그에 대한 데이터 쓰기 작업의 확실성을 높이기 위해 Continuous Query Index(이하 CQI)기법을 사용한 모델이다[10]. 쓰기 작업이 성공하지 못한 태그가 다시 발견되게 되면 쓰기 작업을 재시도하는 방식으로서 CQI를 통해 태그들에 대한 검색 성능을 향상시켰다.

쓰기 작업이 실패했다고 여겨지는 실행에 대해 재시도를 하기 위한 방법으로 Asynchronous Wake-up(이하 AW) 방식 사용된다. AW 재시도 방식은 성공하지 못한 실행에 대해 이를 다른 작업들과 분류하여 관리하는 것으로 이에 해당하는 태그가 발견되면 AW 모델은 미완료의 실행 작업을 wake-up하여 재실행을 하게 된다.

쓰기 작업이 실패했다고 여겨지는 태그는 별도의 리스트로 관리가 되고 해당 태그가 재발견 되었을 때 CQI를 사용하여 리스트를 검색한다. 이 때 매칭되는 태그가 있을 경우 해당 명령을 재실행하게 되는 과정을 거친다. CQI는 리스트에서 일차원 인덱스를 구성하게 되는데 이때 인덱스의 키는 태그의 아이디인 tid이다. 그리고 CQI의 구조를 위해 확장된 해싱 방법을 사용한다. 이러한 구조를 보여주는 것이 [그림 5]이다.

CQI를 통한 쓰기 작업의 결과가 불확실하다고 여겨지는 태그들이 리스트를 효율적으로 관리하며 이러한 리스트들에 대한 쓰기 작업을 재시도하는 과정은 다음과 같다. 불확실 리스트 내의 태그가 재발견되면 태그로부터 데이터를 읽게 된다. 이때 데이터가 정상일 경우는 종료가 된다. 읽기 작업에 대해 태그의 reply가 No Response일 경우 연결 상태를 확인하며 태그가 사라지게 되면 다시 태그가 발견되기를 기다리며 그렇지 않을 경우는 다시 태그에 대한 읽기를 시도한다. 읽기가 성공 후 데이터가 정확하지 않

을 경우는 태그에 쓰기 작업을 시도 하고 성공하지 못한 경우는 다시 연결 상태를 확인하며 성공 후에는 종료가 된다. [그림 6]은 이를 순서도로 표현한 것이다.

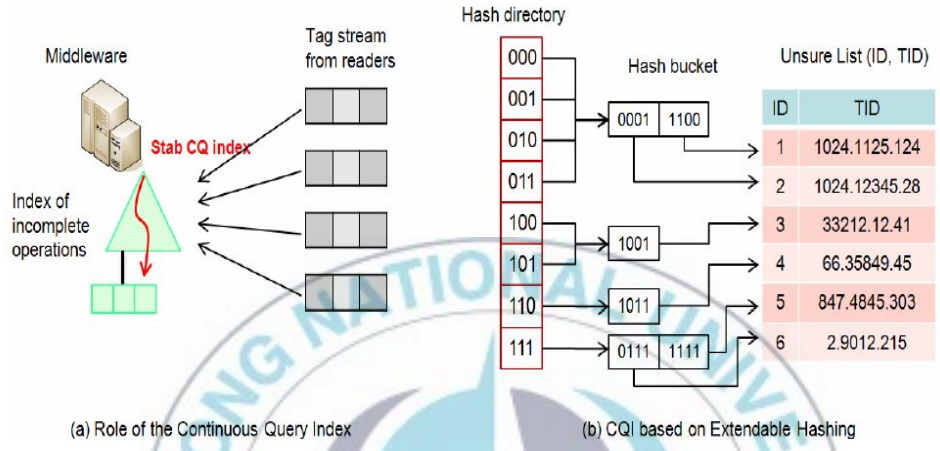

그림 5 불확실 리스트를 위한 CQI

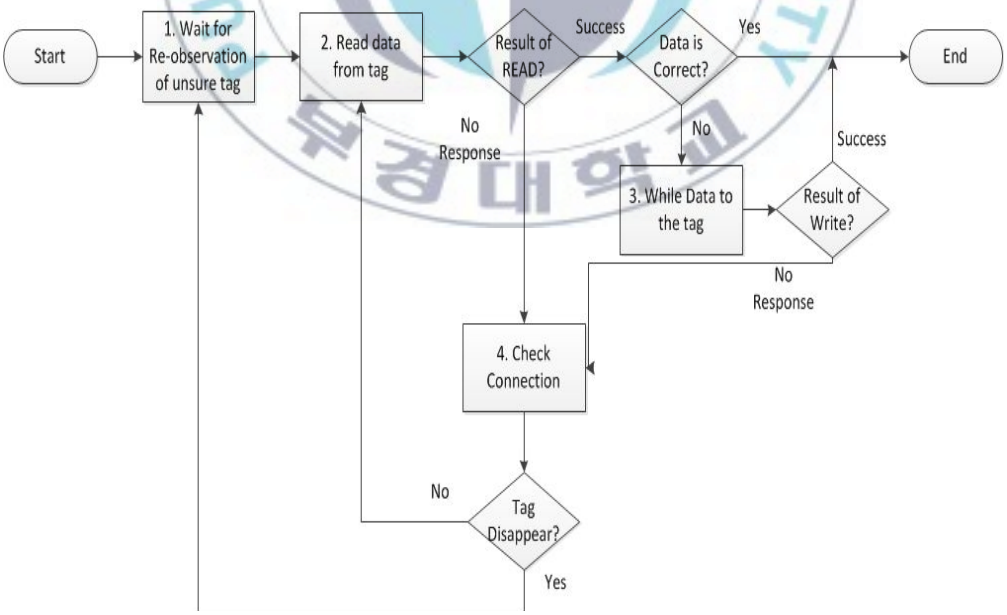

그림 6 불확실한 실행들에 대한 재처리 과정 순서

3. 대량의 RFID 태그에 대한 병행쓰기 기법

3.1. 다수의 리더를 사용한 병행쓰기 기법

본 논문에서 제안하는 시스템은 [그림 7]과 같이 구성된다. 리더의 연결과 쓰기 작업을 시행할 수 있는 미들웨어가 있으며, 다수의 리더가 미들웨어에 연결되어 있다. 쓰기 작업을 위한 태그는 리더의 안테나 범위에 분포한다. 이 때 태그는 각 안테나의 적절한 범위 내에 좁은 간격으로 배치되어 있다. 그리고 태그에 대한 EPC ID 및 태그 ID는 알고 있는 상태이다.

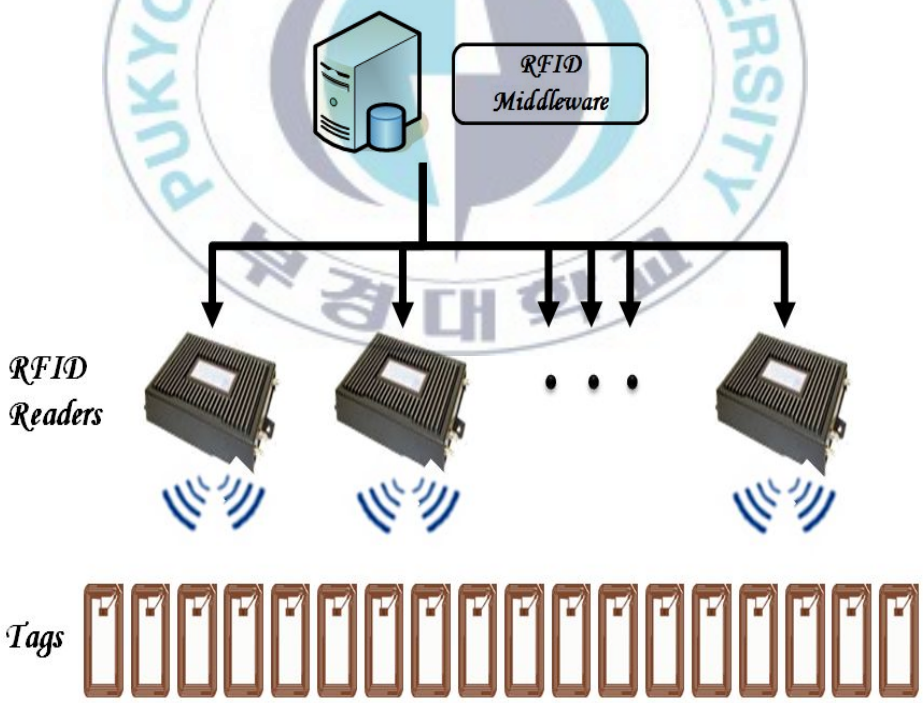

그림 7 시스템 구성도

본 논문에서 제안하는 기법은 보다 효율적인 태그 쓰기 작업을 위해 interleaving 개념을 도입한다. 리더가 최대로 쓰기 작업을 수행할 수 있는 태그의 개수 내에서 데이터를 모든 태그에 동시에 쓰는 것이 아니라 전체의 태그를 일정한 간격을 둔 태그 단위로 그룹을 지정하여 쓰기 작업을 수행하는 것이다. 이때 interleaving이란 일종의 간격으로서 태그가 협소한 공간에 위치할 경우 태그 간의 간섭 현상을 최소화하기 위해 도입하는 것이다. 안테나의 범위 내에 태그가 좁은 간격으로 배치된다고 가정했을 때 리더를 통해 일련의 태그 또는 개별 태그에 대한 쓰기 작업을 실시할 경우 그 실패 확률은 높을 수 있다. 이를 위해 interleaving을 사용하여 interleaving의 값만큼 태그 간의 간격을 두어 쓰기 명령을 수행하는 것이다. 모든 태그에 대해 서로 다른 데이터를 쓰게 될 경우에도 interleaving을 통해 각 태그에 대한 데이터를 고려하여 쓰기 작업을 실시한다.

3.2. 개별 안테나에서의 interleaving 기법

본 논문에서 제시하는 방법은 [표 5]와 같은 알고리즘을 바탕으로 수행된다. 쓰기 작업을 하고자 하는 태그의 리스트를 N이라고 하고, 미들웨어 연결된 리더의 개수를 K라 한다. 그리고 태그에 쓰고자 하는 데이터의 리스트를 v라 한다. 그리고 한 리더가 쓰기 작업을 수행할 수 있는 태그의 최대 개수를 M이라 한다.

리더별로 태그 쓰기 작업을 할당하기 위해 N/K 연산을 수행하여 쓰기가 가능한 리더 별로 태그를 할당한다. 그리고 이 연산의 결과로 리더 당 쓰기 작업 리스트인 T를 구한다. 이 T는 리더 별로 쓰기 작업을 해야 하는 태그의 수를 의미한다. 이 후 태그 간의 간격을 의미하는 interleaving 값을 입력한다. 이는 사용자가 임의로 지정하는 것으로서 태그 간의 간격을 지정하여 실

제 쓰기 작업 시 사용하게 된다. interleaving 값을 통해 태그 리스트에서 간격을 고려하여 태그들을 분류하게 되며 동일 간격을 가지게 된 태그들은 각각의 리스트 G에 삽입되게 된다. 예를 들어 20개의 태그에서 interleaving 값을 1로 지정할 경우 한 번에 쓰기 작업을 시도하게 되는 태그의 개수는 10개가 되며 태그 간의 간격은 1이 된다. 이에 따라 G는 2개가 만들어지게 된다. 그리고 리스트 G에 들어있는 각각의 태그에 대해 쓰기 연산을 실행하게 되며, 이 후 태그에서 리더로 전달되는 reply의 형태에 따라 태그들을 분류하며, 재시도가 가능한 태그에 대해서는 재시도를 실시하며 이 후 쓰기 작업의 성공과 실패를 보고하게 된다. 또한 각각의 태그 또는 모든 태그에 대한 쓰기 작업은 동일한 데이터를 쓰고자 하는 경우와 각각 다른 데이터를 쓰고자 하는 두 가지 경우가 있을 수 있다. 이를 위해 동일 데이터의 쓰기 작업은 리스트 G 내의 모든 태그에 대해 쓰기 작업을 실시한다. 이와 다르게 각 태그에 대해 모두 다른 데이터를 쓰고자 하는 경우에는 각 태그에 맞는 데이터를 매칭시켜 쓰기 위해 g를 사용하는데 이는 태그 리스트 G에서 선택된 하나의 개별적인 태그를 가리키는 것이다. 따라서 개별 태그인 g에 쓰고자 하는 데이터를 각 태그에 쓰게 된다.

표 5 병행쓰기 알고리즘

```
Algorithm MassWrite(N,K,V,M)
N is a tag list
K is a number of readers
v is a data list which be written into tags
M is a number of maximum tags which can handled by reader
begin
    T = N/K
    for each reader
    interleaving_factor x
```

```
      categorize each tags by x
      G[] is a list of all tags which are categorized by x
      for each G
      if (v is a one value)
         reply = lowWrite(G, v)
      else
      g is a each tag in G
      for each g in G
         reply = lowWrite(g, v)
      endif
      if (reply.status = WROK)
         return success;
      else if (reply.status = WRERR)
         return reWrite(reply.temporarilyfailedtaglist, v)
      else if ( reply.status = NOTAG)
         return failed;
      end if
end

reWrite(T, v)
    reply = lowWrite(T,v)
    if( reply.status = WROK )
      return success;
    else if( reply.status = WRERR )
      return reWrite( reply.temporarilyfailedtaglist, v )
    else if ( reply.status = NOTAG )
      return failed;
    end if
end
```

[그림 8]의 알고리즘의 도식화에서 보듯 일련의 태그 리스들에 값들이 다음과 같이 일정한 인덱스 번호를 가진 형태로 배열되어 있다. 그리고 각 리더에 할당된 태그들의 인덱스 번호를 앞서 구한 interleaving_factor로 나눈 값들이 나오게 된다. 예로 0과 1, 2의 값들이 나온다고 가정했을 때 같은 값을 가진 태그들이 리스트 G에 각각 삽입되게 된다. 이렇게 되면 태그와 태그 사이에 약 2개의 태그만큼의 interleaving이 생기게 된다. 이 후 리스트 G 내부의 태그들에 대해 쓰기 작업을 실시하게 된다.

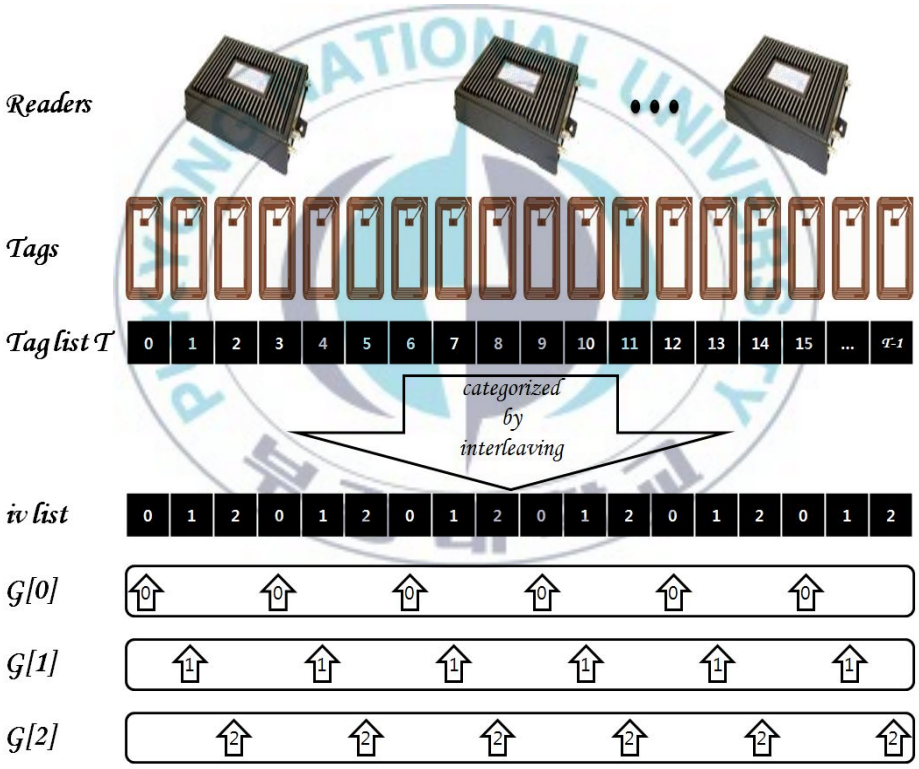

그림 8 알고리즘의 도식화

쓰기 작업 이후 태그로부터의 reply를 통해 쓰기 작업의 재시도를 하게 된다. 재시도 작업을 실시하게 되는 태그의 선별은 쓰기 작업 이후 개별 태그로

부터의 reply 형태에 따라 분류가 가능하게 된다. reply는 본 논문에서 실험을 위해 사용한 장비인 Intermec 사의 IF5 fixed reader의 프로토콜을 따른다[11]. 쓰기 작업 이후의 reply의 형태는 [표 6]과 같이 크게 5가지로 나눌 수 있다. 첫 번째 형태는 쓰기 작업이 성공한 경우로서 태그의 reply는 'WROK'이다. 두 번째 형태는 쓰기 작업이 실패한 경우로서 'WRERR'이라는 reply가 있다. 세 번째와 네 번째 형태는 EPCglobal Class 1 Gen 2 태그에 대한 메모리 접근에 대한 오류의 reply로서 'ADERR'과 동일 태그에 대한 메모리 접근 권한이 없는 경우 발생하는 reply로서 'PRVERR'가 있다. 마지막 형태는 없는 존재하지 않는 태그에 대한 쓰기 작업을 시도했을 때 발생하는 'NOTAG'가 있다.

표 6 reply 형태의 분류

reply	설명
WROK	쓰기 작업의 성공
WRERR	쓰기 작업의 실패
ADERR	EPCglobal Class 1 Gen 2 태그에서의 태그 메모리 접근 에러
PRVERR	EPCglobal Class 1 Gen 2 태그에서의 메모리 접근 권한 에러
NOTAG	존재하지 않는 태그에 대한 쓰기 작업 시도 시 발생

이러한 reply 형태 중 'ADERR', 'PRVERR'의 경우 EPCglobal Class 1 Gen 2 태그라는 특정 태그에서의 형태이므로 쓰기 작업의 재시도를 위한 부분에서 제외하였다. 각 reply 특성 또한 태그 메모리 접근 에러나 메모리 권한 접근 에러로서 쓰기 작업의 재시도를 위해서는 사용자의 추가적인 작업이 필요한 것 또한 제외의 이유가 된다.

본 논문에서는 reply의 형태 중 앞서 얘기한 두 가지 형태의 reply 형태와

쓰기 작업의 성공을 알리는 'WROK', 존재하지 않는 태그에 대한 쓰기 작업 시도 시 발생하는 'NOTAG'를 제외한 'WRERR'의 reply를 반환하는 태그에 대한 쓰기 작업을 재시도하는 것으로 한다.

이러한 분류를 통해 본 논문에서의 reply에 따른 쓰기 작업 결과의 형태를 다음과 같이 나눈다. 먼저, 태그에 대한 쓰기 작업이 성공했다는 의미를 담은 reply인 'WROK'는 성공으로 간주하며, 쓰기 작업의 실패를 나타내는 'WRERR'은 쓰기 작업의 재시도를 위해 별도의 리스트에 삽입한다. 그리고 쓰기 작업이 불가능하다고 여겨지는 상태를 나타내는 reply인 'NOTAG'는 완전한 실패로 간주한다.

이에 따라 크게 상태를 나누면 [표 7]과 같이 Success와 Temporarily Failed, Failed로 나누며, Temporarily Failed는 쓰기 작업의 재시도가 가능한 'WRERR'을 반환한 태그에 대해서는 재시도가 불가능한 'NOTAG'를 반환한 태그에 대해서는 Failed를 할당한다. 이를 통해 Success는 사용자에게 성공으로 보고를 하며, Temporarily Failed의 경우 쓰기 작업 재시도를, 'NOTAG'는 실패를 각각 사용자에게 보고한다.

표 7 쓰기 작업의 결과 분류

쓰기 작업의 결과	설명
Success	'WROK'를 반환한 태그
Temporarily Failed	'WRERR' 쓰기 작업 재시도 가능 태그
Failed	'NOTAG' 쓰기 작업 재시도 불가능 태그

3.3. 다수의 리더에서의 interleaving 기법

다수의 리더를 사용할 경우 리더의 안테나 간의 간섭과 충돌 현상이 발생할 수 있는데 아래의 그림과 같이 각 리더에 결합되어 있는 안테나들에서 그러한

현상이 발생할 수 있다[12]. RFID는 무선 주파수를 사용한 기술이므로 이러한 문제가 발생하며 태그에 대한 접근이 원활하지 않을 수 있다. 본 논문에서 제안하는 다수의 태그에 대한 쓰기 기법의 경우 태그와 안테나들이 다수가 사용됨으로 인해 이러한 문제가 발생할 확률이 더욱 높다.

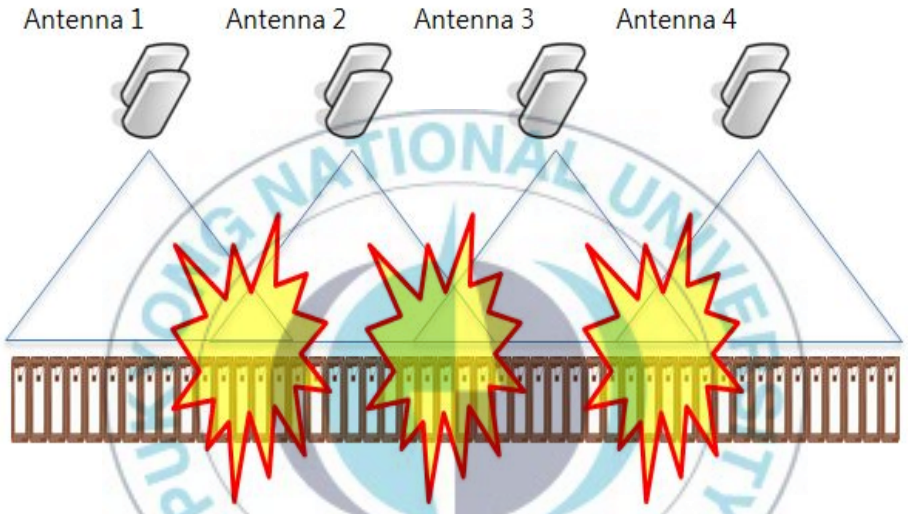

그림 9 다수의 안테나 사용에 의한 충돌 발생 상황

충돌 부분은 [그림 9]에서 보듯 각 안테나에서 태그로 보내지는 무선 신호가 겹치는 곳에서 발생할 수 있다. 안테나 1과 안테나 2가 동시에 태그에 대한 쓰기 작업을 실시한다고 했을 때, 이 두 개의 무선 신호가 겹치는 부분에서는 태그에 대한 접근 문제가 발생할 수 있다. 쓰기 작업뿐만 아니라 쓰기 작업 후 태그로부터 리더로 전달되는 reply 메시지 또한 문제가 발생할 수 있다. 또한 다수의 리더를 사용한다고 하였을 때 개별 리더의 안테나 사이의 충돌 문제뿐만 아니라 리더 1의 안테나 2와 리더 2의 안테나 3의 무선 신호 또한 충돌이 발생할 가능성이 있다. 이를 해결하기 위해서는 다수의 리더 또

는 다수의 안테나를 통한 태그로의 접근에 대한 방안이 필요하다. 또한 쓰기 작업과 쓰기 작업 후 reply에 대한 고려도 필요하다.

이를 해결하기 위해 본 논문에서는 두 가지 기법을 제시한다. 리더별 안테나의 경우 최대 4개까지 지원할 수 있는데 이러한 안테나들의 쓰기 작업 시도를 시간 별로 나누어 진행하는 시간 분할 방법과 주파수 특성을 이용하여 주파수를 분할한 각 채널 별로 쓰기 작업을 시행하는 주파수 분할 방법이다.

시간 분할 방법의 경우 리더의 작동 시간을 분할하는 것이다. 위의 그림에서 충돌이 발생하는 부분에 대해 안테나 작동의 우선순위를 정하는 방법으로 해결이 가능하다. 예를 들어 리더1의 안테나 1과 리더 2의 안테나 3의 안테나를 동시에 작동시키며, 작동이 마무리되면 리더 1의 안테나 2와 리더 2의 안테나 4를 작동시킨다. 이렇게 되면 안테나의 범위가 겹쳐 충돌이 발생하는 부분을 피할 수 있게 되며 동시에 쓰기 작업을 진행함으로서 리더 사용의 효율성을 높일 수 있다. 리더가 추가적으로 설치가 되고 안테나의 개수 또한 늘어나는 경우에도 시간 분할 방법을 사용할 경우 간섭 및 충돌 현상 없이 작업이 가능하다.

두 번째 방법은 주파수 분할 방법이다. EPC UHF Gen2 규격에 태그는 두 리더가 동시에 활성 상태인 경우에도 작동해야만 한다. 이러한 환경을 고밀도 환경(Dense Interrogator Environment)라고 하는데 고밀도 환경에서 발생하는 충돌 방지 방법은 여러 가지가 있으나 본 논문에서는 주파수 분할 방법을 사용한다.

주파수 분할 방법의 경우 기존의 주파수 영역에서 여러 개의 채널로 나누어 각각의 안테나에 대해 다른 채널을 할당하여 사용하는 방법이다. 이렇게 사용할 경우 여러 개의 안테나를 동시에 사용하더라도 채널이 다르므로 무선 신호의 간섭 및 충돌 현상을 줄일 수 있다. 시간 분할 방법과 다르게 주파수 분할 방법은 서로 충돌이 발생하는 리더 1의 안테나 1과 안테나 2를 동시에 사용

할 수 있다는 것과 리더 1의 안테나 2와 리더 2의 안테나 3 또한 동시에 사용이 가능하다.

이와 같은 방법으로 리더 당 개별 안테나의 작동 시간 분할과 주파수 분할 방법을 사용하게 되면 다음과 같은 장점이 있다. 먼저, 다수의 리더를 사용할 수 있는 점이다. 개별적으로 하나의 리더만을 사용하는 것이 아니라 여러 개의 리더를 동시 사용이 가능하다는 점이다. 둘째, 리더와 안테나 간에 발생하는 간섭 및 충돌 현상을 해결할 수 있다. 또한 다수의 리더에 부착되는 다수의 안테나 사용이 가능해진다. 각 안테나별로 별도의 주파수를 할당하는 방법을 사용하는 것이다. 이러한 방법으로 다수의 태그에 대한 쓰기 작업 시 문제를 제거함으로서 본 논문에서 제안하는 다수의 리더를 사용한 병렬 쓰기 기법이 가능해진다.

PUKYONG NATIONAL UNIVERSITY
부경대학교

4. 실험 및 고찰

4.1. 실험 환경

실험의 환경은 다음과 같다. 먼저, 실험에 사용한 RFID 리더는 Intermec 사의 IF5 fixed reader로서 고정형의 리더이며 태그는 EPCglobal Class 1 Gen 2 태그이다. 모두 동일한 태그이며 태그의 ID와 EPC ID는 모두 알고 있는 상태로 실험을 진행하였다. 실험을 위해 테스트 프로그램을 작성하여 리더와 연결하였으며 각 태그에서 리더로 전달되는 reply 분석과 함께 본 논문에서 제안한 기법을 실험하였다. 실험을 위해 안테나를 고정시킨 후 안테나의 범위 내에 태그들을 좁은 간격으로 배열시키고 시행하였다. 태그들은 일렬로 모두 배열되어 있으며 태그들 간의 간격은 0.5cm로 하여 실험을 진행하였다. 태그와 리더의 간격은 50cm로서 쓰기 작업을 위해 일률적으로 설정하였다. 이와 더불어 지속적인 쓰기 작업은 태그의 성능 저하를 가져오게 된다. 이로 인해 지속적인 쓰기 작업은 실험의 결과에 영향을 주기 때문에 각 실험별로 태그의 교체를 통해 실험의 결과에 영향을 최소화시켰다.

실험은 단순 쓰기 작업과 interleaving 기법을 비교하기 위해 몇 가지의 형태로 진행되었으며 정확한 실험을 위해 리더에서의 읽기와 쓰기의 시도 횟수는 1로 고정하여 시험하였다. 그리고 이러한 환경에서 성공률 비교 실험을 진행하였다. 또한 리더 2대를 사용하여 안테나 interleaving과 주파수 분할 방법의 필요성을 증명하였다.

- 실험 1 : 단순 쓰기 작업과 다양한 interleaving 값에 따른 쓰기 작업 1회 후의 성공률 비교
- 실험 2 : 단순 쓰기 작업의 쓰기 시도 횟수 조정을 통한 성공률 비교
- 실험 3 : interleaving 값에 따른 태그 인식률 비교
- 실험 4 : 리더 2대를 사용한 단순 interleaving 기법 사용에 의한 성공률 비교

4.2. 실험 결과 및 고찰

실험은 네 가지의 형태로 진행되었다. 첫 번째 실험은 모든 태그에 대한 단순 쓰기 작업과 본 논문에서 제안하는 interleaving 기법을 비교하였다. 단순 쓰기 작업의 경우 쓰기 작업의 재시도가 없으며 interleaving 기법의 경우 쓰기 작업의 재시도가 포함된다. 이를 위해 40개의 태그를 사용하였으며 두 방법의 비교를 위해 실행 후 쓰기 작업의 성공률을 실험을 통해 측정하였다. 쓰기 작업의 재시도의 경우 'WRERR'을 반환하는 태그는 최종적으로 'WROK' 또는 'NOTAG'를 반환할 때까지 진행되었으며, 작업의 성공률은 태그에 대한 쓰기 작업이 성공했다는 reply인 'WROK'를 전달한 태그의 개수를 전체 개수로 나누어 성공률을 표시하였다. 실패로 분류되는 태그는 최종적으로 'NOTAG'를 반환한 태그들로 구성하였다.

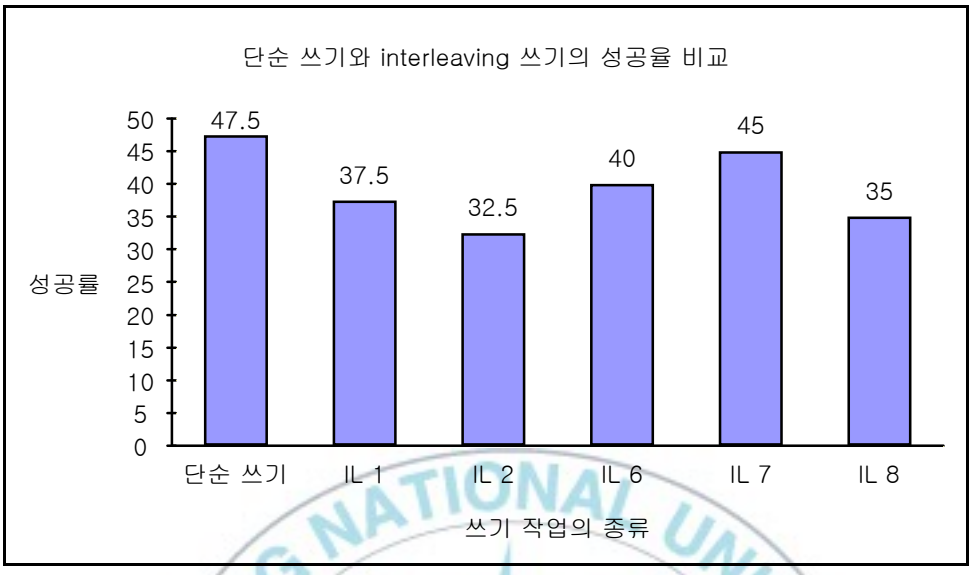

그림 10 단순 쓰기와 interleaving 쓰기의 성공률 비교

[그림 10]은 단순 쓰기와 interleaving 값을 1, 2, 6, 7, 8로 변경하면서 진행한 실험의 결과이다. 단순 쓰기는 47.5%의 성공률을 보였으며 interleaving 값이 1일 경우에는 37.5%, 2일 경우 32.5%, 6일 경우 40%, 7일 경우 45, 8일 경우 35%로 나타났다. 이는 재시도를 하지 않고 쓰기 작업을 한번만 시행했을 때의 결과 값이다. 단순 쓰기의 경우 평균 성공률이 47.5%로서 절반에 못 미치는 결과를 보였고 interleaving을 고려한 쓰기 작업은 이보다 낮은 성공률 보였다. 이는 재시도를 하지 않은 첫 데이터로서 단순 쓰기보다는 성공률이 떨어짐을 알 수 있다.

이와 비교를 위해 단순 쓰기 작업을 리더의 쓰기 작업 시도 횟수를 변경하여 재실험을 실시해보았다. 실험은 이더넷을 통한 리더의 연결을 통해 시행하였으며 쓰기 작업의 시도 횟수를 각각 1번, 5번, 10번, 15번, 20번으로 나누어 시행하였다.

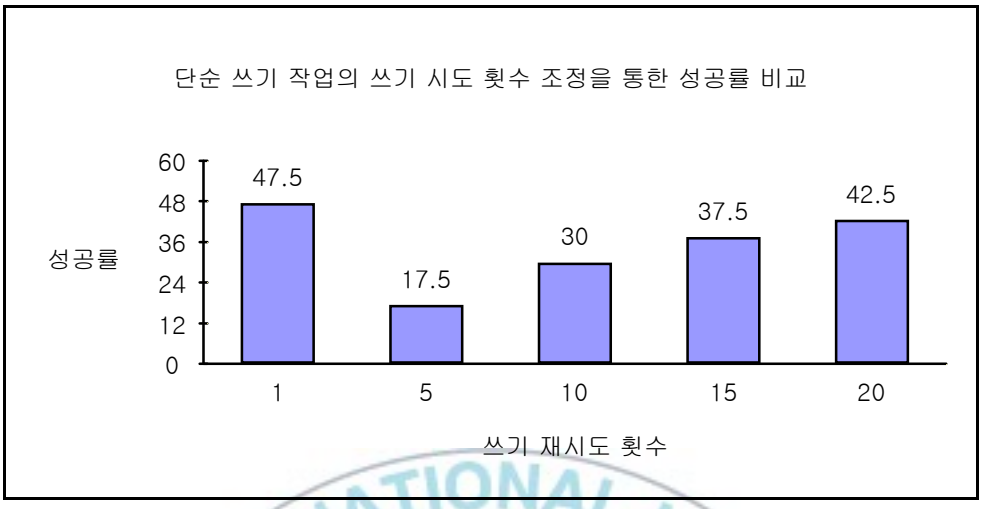

그림 11 단순 쓰기 작업의 쓰기 시도 횟수 조정을 통한 성공률 비교

실험의 결과는 [그림 11]과 같다. 맨 처음 시도의 경우 47.5%라는 성공률을 보였지만 시도 횟수를 증가시킬수록 쓰기 작업의 성공률이 낮아짐을 알 수 있다. 이는 지속적으로 태그에 대한 쓰기 작업 횟수가 증가하면서 성공률이 떨어지는 것으로 생각된다. interleaving과의 비교를 위해 태그를 변경하여 진행하였으나 interleaving을 사용한 쓰기 작업에 비하여 재시도 횟수를 증가시킨다고 하여 성공률이 증가하지 않음을 알 수 있었다. 실험에 사용된 Intermec사의 IF5 리더의 쓰기 동작 방식은 알 수 없으나 미루어보아 전체 태그에 대한 쓰기 작업을 지속적으로 시행하는 것으로 판단된다.

이와 함께 실험 1의 결과인 [그림 10]에서 interleaving 값이 각각 6과 7, 8 일 경우에는 성공률이 증가하는 것을 볼 수 있다. 하지만 interleaving 값 1과 2와는 조금의 차이가 있음을 알 수 있는데 이는 interleaving 값이 작을수록 태그 인식 수준이 낮아져 실제 쓰기 작업을 시도 후 'NOTAG'가 많이 발생할 것이라 추측하였다.

이를 증명하기 위해 실험 3을 실시하였다. 실험 2는 interleaving 값에 따른 태그 인식률을 비교하는 것으로서 이를 통해 interleaving 값 간의 차이점에 대해 분석하였다. 각각의 interleaving 값들에 대한 첫 번째 쓰기 작업 이후 재시도가 필요한 태그들의 개수와 재시도의 횟수를 측정한 것이며, interleaving 값 1부터 8까지 차례대로 측정한 결과는 [그림 12]와 같다.

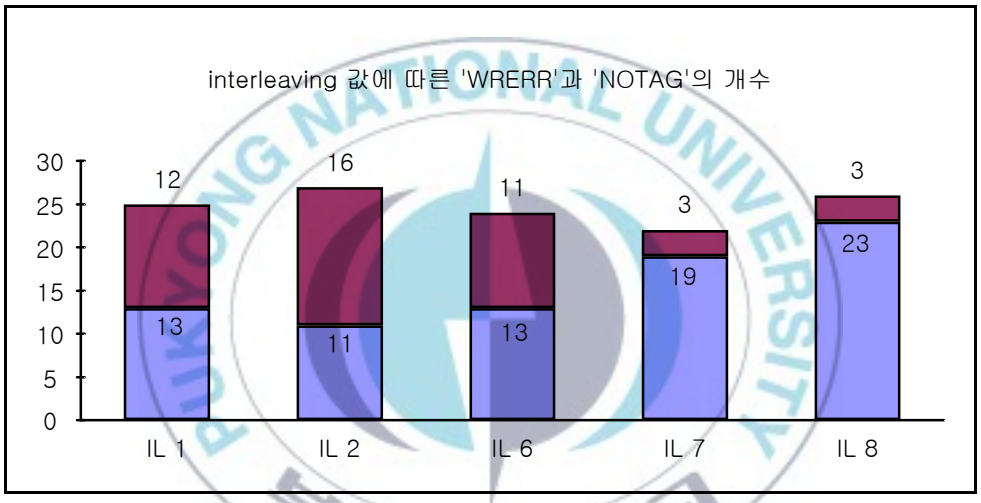

그림 12 interleaving 값에 따른 'WRERR'과 'NOTAG'의 개수

실험의 결과를 살펴보았을 때 쓰기 작업이 실패했다고 여겨지는 reply의 형태인 'WRERR'은 interleaving의 값이 점차 증가할수록 함께 증가한다는 것을 알 수 있다. 하지만 이와 반대로 인식되지 않는 태그를 나타내며 최종적으로는 쓰기 작업이 실패했다고 여겨지는 reply 형태인 'NOTAG'의 경우 그 값이 점차 줄어드는 것을 알 수 있다.

위 실험 결과에 따라 본 논문에서 제안한 interleaving 기법을 사용할 경우 다음과 같은 장점이 있다는 것을 알 수 있다. 첫째, 쓰기 작업의 성공

률을 단순 쓰기 작업에 비해 월등히 높일 수 있다는 점이다. 단순 쓰기 작업의 평균 성공률이 절반에도 못 미치는 반면 interleaving을 고려한 쓰기 작업에서는 높다는 것을 알 수 있다. 두 번째, 다수의 태그가 밀접한 환경에서 본 기법을 사용하게 될 경우에도 쓰기 작업의 성공률을 높일 수 있다는 점이다. 세 번째로는 interleaving 값의 조절을 통해 밀접한 태그들 간의 간섭 현상을 줄여 태그의 인식률을 높일 수 있다는 점이다.

실험 4는 본 논문에서 제안한 안테나 interleaving과 주파수 분할 기법에 대한 필요성을 증명하기 위해 진행된 실험이다. 이 실험에는 두 대의 동일한 Intermec사의 IF5 fixed reader가 사용되었으며 리더 별로 안테나는 2개를 연결하였다. 그리고 태그는 총 80개를 사용하여 진행되었으며 앞선 실험과 마찬가지로 태그 간 간격과 태그와 리더의 안테나 사이 간격은 각각 0.5cm와 50cm로 동일하다.

안테나의 배치는 리더 당 안테나 두 대를 먼저 배열하고 다음 리더의 안테나 2대를 배열하는 방식으로 하였으며, 안테나의 범위는 겹치는 것으로 실험 환경을 구성하였다. 실험은 interleaving의 값을 1과 8로 하여 진행하였으며 이 두 값에 대해 각 리더 당 성공률을 비교하였다. 이에 대한 결과는 [그림 13]과 같다.

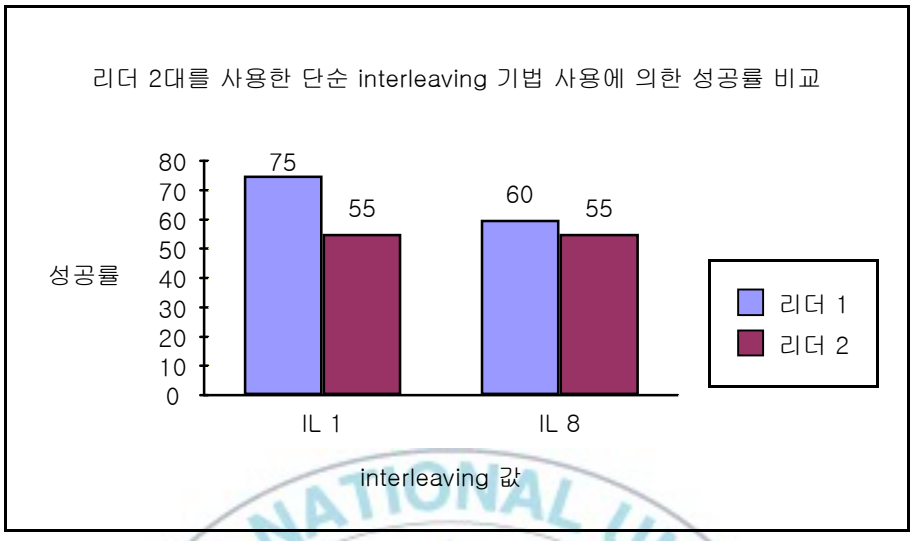

그림 13 interleaving 값의 따른 리더 2대의 성공률

실험의 결과 interleaving 값 1과 8에 따라 어느 정도에 차이는 있으나 리더 1대를 사용했을 때의 성공률보다는 확실히 떨어지는 것을 알 수 있다. interleaving 1의 경우가 평균적으로는 약 60%, interleaving 8의 경우 약 55% 정도의 성공률을 보였다. 이는 실제 리더 1대에서 시행하였던 단순 쓰기 작업의 성공률을 조금 상회하는 정도의 수준이다. 다음과 같은 결과가 나오게 된 것은 안테나간의 간섭 현상에 의한 것이라고 볼 수 있다. 실제로 리더 2대에 연결된 안테나 4대는 서로 전파 영역이 겹치는 부분이 있으며, 특히 리더 1의 안테나와 리더 2의 안테나가 겹치는 부분에서의 쓰기 작업에 더욱 문제가 있는 것으로 알 수 있었다. 이러한 점을 볼 때 본 논문에서 제안했던 다수의 리더를 사용하기 위한 안테나 interleaving 기법과 안테나별 주파수 분할 기법이 필요하다는 것을 알 수 있다.

현재 가장 광범위하게 사용되고 있는 Alien사나 Intermec사의 RFID 리더 모두 본 논문에서 제안한 기법의 사용이 불가능하다. 예로, Alien사의

9800 RFID 리더의 경우 리더에 연결된 안테나 한 개만이 쓰기 작업을 위해 사용될 수 있다[13]. 또한 본 논문의 실험에 사용된 Intermec사의 RFID 리더의 경우에도 특정 안테나에 쓰기 작업 할당이 불가능하며 주파수 분할 방법 또한 지원하지 않음을 확인하였다. 이를 통해 본 논문에서 제안하는 다수의 리더를 사용하기 위한 두 가지 기법을 지원하는 RFID 리더의 개발이 필요하며, 이를 통해 다수의 리더와 다수의 안테나를 사용한 보다 안정적인 쓰기 작업이 가능할 것으로 보인다.

5. 결 론

본 논문에서는 다수의 리더를 이용한 다수의 태그에 대한 병렬쓰기 작업의 효율성을 위한 기법을 제시하였다. 일련의 과정을 통해 쓰기 작업을 실시할 경우 다음과 같은 장점이 있다. 다수의 리더를 사용한 쓰기 작업이므로 효율적인 리더의 사용이 가능하다. 미들웨어 연결된 다수의 리더를 사용하여 태그에 대한 쓰기 작업이 가능하기 때문이다. 또한 개별적인 태그에 대한 쓰기 작업이 아닌 다수의 태그에 대한 쓰기 작업을 시행하므로 빠른 속도로 쓰기 작업을 완료할 수 있다. 그리고 본 논문에서 사용한 기법인 interleaving 기법을 통해 보다 안정적이고 확실한 태그 쓰기 작업이 가능하다. 기존 쓰기 방식의 경우 대량의 태그가 좁은 범위 내에 위치할 경우 다수의 태그에 대한 쓰기 작업의 실패율이 높아질 경우가 있다. 이때 interleaving 기법을 사용하여 부분적으로 쓰기 작업을 실시함으로서 태그 내 데이터의 일관성을 보장할 수 있다. 그리고 좁은 범위 내에 위치하는 태그들의 경우 간섭 현상 등에 의해 인식률에 문제가 발생할 수 있는데, 이를 interleaving을 통해 일정한 간격을 두어 쓰기 작업을 실시함으로 쓰기 작업의 확실함을 보장할 수 있는 것이다. 이를 통해 다수의 리더를 사용하여 다수의 태그에 대한 병렬쓰기 방식을 사용함으로 보다 효율적인 태그 메모리에 대한 쓰기 작업이 가능해진다. 이와 더불어 현재의 RFID 리더에서 지원하지 않고 있는 안테나별 쓰기 작업의 할당과 안테나 또는 리더 간의 주파수 분할 방법을 지원하게 된다면 본 논문에서 제안된 interleaving 기법의 효율적인 사용이 가능할 것으로 보이며, 이를 통해 안정적이며 확실한 쓰기 작업이 가능할 것이다.

참고문헌

[1] R. Want, An Introduction to RFID technology, Pervasive Computing, IEEE, Volume 5, Issue 1, pp 25-33, 2006

[2] EPCglobal, Reader Protocol 1.1 Recommended Specification April 10, 2006

[3] R. Weinstein, RFID: a technical overview and its application to the enterprise, IT professional, vol. 7, no. 3, pp. 27-33, 2005

[4] C Floerkemeier, M Lampe, Pervasive Computing, Lecture Notes in Computer Science, Volume 3001/2004, 188-193, 2004

[5] K. P. Fishkin, B. Jiang, M. PHilipose, and S. Roy, I sense a disturbance in the field: unobtrusive detection of interactions with RFID tagged object, Lecture Note in Computer Science, Volume 3205, pp 268-282, 2004

[6] Konstantinos Domdouzis, Bimal Kumar, Chimay Anumba, Radio-Frequency Identification(RFID) applications: A brief introduction, Advanced Engineering Informatics, Volume 21, Issue 4, October 2007, Pages 350-355

[7] Ron Weinstein, RFID: a technical overview and its application to the enterprise, IT professional, June 2005, pp 27-33

[8] Wooseok Ryu and Bonghee Hong, A Presumed Write Protocol for Ensuring Complete Access to RFID Tag Memory, Ubiquitous Information Technologies & Applications, ICUT '09. Proceedings of the 4th International Conference on, 2009

[9] 류우석, 홍봉희, 최병운, 태그 메모리 접근을 위한 RFID 미들웨어의 쓰기 프로토콜, 데이터베이스연구 제 25권 제 2호, 2009.8

[10] Wooseok Ryu and Bonghee Hong, A Reprocessing Model Based on Continuous Queries for Writing Data to RFID Tag Memory, Database Systems for Advanced Applications, Lecture Notes in Computer Science, Volume 5463/2009, 201-214, 2009

[11] Intermec Technologies Corporation, IF5 Fixed Reader Programmer's Reference Guide

[12] 빌 글로버, 히만슈 바트 공저, 서환수 역, 실무자를 위한 RFID 이해와 활용, 한빛미디어, 2007년 2월

[13] Alien Technology Corporation, Reader Interface Guide All Fixed Readers, September 2008

PUKYONG NATIONAL UNIVERSITY 부경대학교

Thank to...

"The best way to predict

the future is to invent it!"

처음 대학원 입학과 함께 저의 좌우명으로 삼았던 말이었습니다. 좌우명처럼 대단한 미래를 만들지는 못했지만 희망 가득한 미래를 설계할 수 있었던 소중한 시간이었습니다.

먼저, 부족한 실력임에도 칭찬해주시고 아껴주시고 보살펴주셨던 송하주 교수님께 감사드립니다. 그리고 항상 밝은 미소로 저에게 좋은 말씀 많이 해주신 조우현 교수님께도 감사드립니다. 또한 2년 동안 함께 했던 연구실 동기들에게도 감사드립니다. 늘 뼈가 되고 살이 되는 조언을 아끼지 않으셨던 멘토 명환형, 맏형이자 연구실의 지주였던 우리 대표 태용, 부족한 저를 항상 도와주었던 우리 승혁, IDB의 CTO 우리 성진, 까칠한 매력에 멋진 우리 영민, 큰 키지만 싱겁지 않고 구수한 우리 주호형, 늘 웃는 모습에 노력하는 우리 재형, 가장 잘따르고 마음이 맞았던 우리 주형, 이미 졸업했지만 항상 좋은 말씀 아끼지 않았던 화춘형, 늘 즐겁게 만들어주었던 종민. 모두가 있어 행복했고 많이 배울 수 있었던 연구실 생활이었습니다.

항상 사랑하고 세상 무엇과도 바꿀 수 없는 소중한 우리 가족들에게도 감사드립니다. 가족이 있어 지금의 제가 꿈을 꿀 수 있었습니다.

어떤 시간과도 바꿀 수 없는 너무나 소중하고 가슴 깊이 담을 추억들이 가득한 여러분과의 시간을 잊지 않겠습니다. 좋은 말, 예쁜 말들보다 이말 하나 드리고 싶습니다. 감사하고 고맙습니다.

2010년 12월

임영준 드림