

RFID 무선 프로토콜 필터링의 설계와 확률적 인식 과정 시뮬레이션

정희원 박 현 성*, 종신회원 김 중 덕*o

Design of RFID Air Protocol Filtering and Probabilistic Simulation of Identification Procedure

Hyun-Sung Park* *Regular Member*, Jong-Deok Kim*o *Lifelong Member*

요 약

RFID 리더에서의 효율적인 필터링은 RFID 시스템의 성능에 있어 중요한 요소이다. RFID 기술 사용이 보편화 되면 태그 및 리더의 수가 크게 늘고 리더가 필터링 없이 수집한 태그 정보 모두를 그대로 RFID 응용 시스템으로 전달할 경우 통신망 및 응용 시스템에 부하를 증가시켜 성능에 악영향을 미치기 때문이다. 본 논문은 RFID 무선 프로토콜 구간 필터링과 무선 인식 후 필터링이라는 단계적인 필터링 기능 조합으로 효과적으로 필터링을 수행하는 방법을 제안한다. 특히 계산 기하학을 활용하여 무선 프로토콜 구간의 필터링을 효율적으로 수행할 수 있는 구체적 방법론을 제시하였다. 시뮬레이션을 통해 제안한 방법의 성능을 검증하였는데 무선 프로토콜 구간의 필터링 없이 수행한 순차적인 필터링에 비해 성능이 50% 정도 개선됨을 확인하였다.

Key Words : RFID, Filtering, Air Protocol, Air Interface, Tag Identification, Reader Protocol, Anti-Collision, Middleware, EPC, Tag

ABSTRACT

Efficient filtering is an important factor in RFID system performance. Because of huge volume of tag data in future ubiquitous environment, if RFID readers transmit tag data without filtering to upper-layer applications, which results in a significant system performance degradation. In this paper, we provide an efficient filtering technique which operates on RFID air protocol. RFID air protocol filtering between tags and a reader has some advantages over filtering in readers and middleware, because air protocol filtering reduces the volume of filtering work before readers and middleware start filtering. Exploiting the air protocol filtering advantage, we introduce a geometrical algorithm for generating air protocol filters and verify their performance through simulation with analytical time models. Results of dense RFID reader environment show that air protocol filtering algorithms reduce almost a half of the total filtering time when compared to the results of linear search.

I. 서 론

일상 생활에서 점차 IT 기술이 도입됨으로 인하여 사물에 유일한 ID를 부여하여 유비쿼터스 환경

으로 변화하기 위한 시도들이 차츰 현실화되고 있다. 그 대표적인 예로 RFID (Radio Frequency Identification) 기술이 사용하는 고유한 ID인 EPC (Electronic Product Code) 태그를 들 수 있는데

※ 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음

(This work was supported for two years by Pusan National University Research Grant)

* 부산대학교 컴퓨터공학과 이동통신 연구실(jjesse, kimjd*)@pusan.ac.kr(° : 교신저자)

논문번호 : KICS2008-12-547, 접수일자 : 2008년 12월 11일, 최종논문접수일자 : 2009년 4월 23일

EPCglobal 등의 표준화 기관과 WalMart 등과 같은 국제적 대형 유통 물류회사들의 주도로 빠르게 보급되고 있다.

RFID 기술을 사용하는 장소가 늘어감에 따라 실시간으로 처리해야 할 EPC 태그 정보의 양도 급격하게 늘어나고 있다. 이러한 추세로 인하여 많은 종류의 태그들이 대량으로 혼재하게 되고 이는 다수의 RFID 리더로부터 입력되는 대규모의 RFID 태그 정보 중에서 상위 응용에서 필요로 하지 않는 불필요한 정보도 같이 입력되게 됨을 의미한다. 즉, 상위 응용들에서 필요한 정보는 주어진 입력과 비교하였을 때 아주 적은 비율이 되는 경우가 많다. 따라서 가까운 미래에 구축될 것으로 보이는 지구적 규모 (global identification)의 RFID 망의 핵심 기능 중 하나는 불필요한 태그 정보를 중도에서 삭제하고 필요한 정보만을 “필터링” 하여 미들웨어나 상위 응용으로 전달하는 것이 매우 중요하다고 할 수 있다.

필터링이란 기본적인 정책에 따라 원하는 정보를 받아들이고 그렇지 않은 정보는 걸러내는 역할을 한다. RFID 필터링은 여러 가지 종류가 있을 수 있지만 본 논문에서는 개체에 부여된 EPC를 비트 단위의 논리곱 연산을 통해 수행되는 저 수준의 필터링에 관심을 두었다. 이는 마치 인터넷 라우터(Router)나 방화벽(Firewall)에서 많은 수의 패킷을 처리(Packet Classification)하기 위해 필터링을 수행하는 것과 유사하며 이에 대한 연구는 RFID 기술에 앞서 많은 연구가 진행되어 왔다^[1,2]. 그에 반해 아직 많은 수의 RFID 태그 ID를 빠르게 필터링 할 수 있는 기술에 관한 연구는 매우 부족한 편이다.

그림 1은 RFID 시스템 아키텍처^[3]의 한 예로써 대규모로 입력되는 태그 정보 처리를 위한 환경을 나타낸다. 그림의 제일 좌측은 여러 가지 종류의 태그로

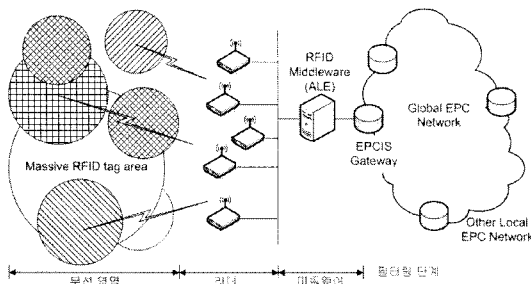


그림 1. 큰 규모의 RFID 시스템의 구성. 다양한 종류의 태그들이 혼재하고 다수의 RFID 리더가 많은 수의 태그 정보를 읽어 미들웨어로 전송.

이루어진 산재된 태그들을 표현하였고 또한 여러 종류의 태그들을 읽어 들이기 위해 필요한 해당 리더들이 위치해 있다. 리더들에서 읽어 들인 정보는 하나 혹은 둘 정도의 미들웨어 (ALE; Application Level Event^[4])로 집중되고 미들웨어가 최종적으로 판단한 태그 정보는 최상위 응용 혹은 EPC Information Services (EPCIS)를 통해 다른 로컬 EPC 네트워크와 결합하여 글로벌 EPC 네트워크에 참여한다.

RFID 시스템에서 필터링이 일어날 수 있는 구간은 무선(Air) 인식 과정, 리더 처리과정, 그리고 미들웨어 처리 과정^[5]에 걸쳐 단계적으로 이루어질 수 있는데 본 연구에서는 크게 무선 구간과 유선 구간으로 나누어 무선 인식 과정에서 수행되는 저 수준의 필터링에 중점을 두었다. 무선 프로토콜 구간에서의 필터링은 리더나 미들웨어 구간에서의 필터링에 비해 태그 정보의 입력량 자체를 줄일 수 있는 특징을 가진다. 그러므로 입력되는 대규모 RFID 태그 정보 중에서 시스템이 불필요하게 읽어 들이는 태그 정보를 무선 인식 구간에서 필터링하여 미리 제거할 수 있고 이는 상위 단계인 리더와 미들웨어에서 처리해야 하는 필터링 연산의 부하를 줄일 수 있는 효과를 보인다. 무선 인식 구간에서의 필터링은 최근 단일 표준으로 중요성이 부각되고 있는 RFID C1G2 (Class 1 Generation 2)^[6]의 Select 명령에 기초한 것으로 특정 태그들을 선별적으로 선택 가능하다는 특징을 이용할 것이다.

II. 배경 설명

본 장에서는 RFID 무선 인식 과정에서 적용할 무선 프로토콜 필터 생성 알고리즘을 고안하여 제안하려고 한다. 먼저 필터링의 대상이 되는 EPC 태그 구조와 기본적인 필터링이란 무엇인지 정의한다. 그리고 기하학적으로 2차원 공간에서 필터를 사각형으로 표현하여 어떻게 하면 효과적으로 원하는 영역만을 선택하고 불필요한 영역을 무선 구간에서 미리 제거할 수 있을지에 대해 설명할 것이다.

2.1 RFID EPC 태그 구조

RFID 산업화 및 기술을 주도하고 있는 EPCglobal은 태그와 리더를 비롯하여 아키텍처^[3] 별로 RFID 표준을 제정하고 있다.

표 1은 EPCglobal에서 공표한 TDS (Tag Data Standard) 표준^[7]에서 필터링의 소스가 되는 태그 데이터 중 96비트로 이루어진 GID-96 (General ID)의 예를 들었다. 태그 데이터 구조는 기존 바코드

표 1. EPC 태그 데이터 중 TDS 1.3 표준에서 규정한 GID-96 태그 구조의 예

	Version Header	General Manager Number	Object Class	Serial Number
GID-96	8	28	24	36

(Barcode)의 체계가 된 EAN-UCC (European Article Numbering - Uniform Code Council)에서 정의한 다양한 형식을 수용하고 있으나 그 종류에 관계없이 공통적으로 (1) Version 정보, (2) 제조사 정보를 나타내는 Domain Manager, (3) 제품 정보를 나타내는 Object Class, (4) 제품의 일련 번호 인 Serial Number의 네 가지 필드로 구성된다.

2.2 RFID 리드 필터링 (Read Filtering)

RFID 필터링의 종류는 필터링 성격에 따라 몇 가지로 분류되는데 본 연구에서는 가장 저 수준의 필터링인 비트 단위 필터링을 다룬다⁸⁾. RFID 리더 (Reader)나 미들웨어에서 수행될 수 있는 리드(Read) 필터링은 EPC 코드에 대해 두 헥사 스트링의 필터값 (V)과 마스크(M)를 이용하여 단순히 마스크에 '1'로 세팅된 비트들에 대해서 비트단위 논리곱 연산을 통해 실제 태그 ID가 같은지 비교 한다. 우리는 수식 1을 표 2의 예제로 그림 2와 같이 기하학적 관점으로 도식하여 그 의미를 설명하고자 한다.

$$\begin{aligned} & \text{If } ((V \text{ bit and } M) \equiv (T \text{ bit and } M)) \\ & \text{Then } \text{TagIDMatchesTheFilter}() \end{aligned} \quad \text{수식 1}$$

태그 데이터 구조는 기본적으로 4가지 필드를 가지고 있으므로 4차원의 공간으로 대응시킬 수 있으나 본 연구에서는 2차원 공간으로 단순화 하였다. 이 때 2차원을 대표하는 두 가지 필드는 태그 구조의 Domain Manager 필드와 Object Class 필드를 대상으로 하였다. Domain Manager 필드는 기업체를 식별하는 번호를 나타내고 Object Class 필드는 제품의 종류를 식별하는 필드를 의미하므로 두 필드가 태그 구조에서 가장 빈번히 필터링되는 부분이라 할 수 있고 필터링 목적에 가장 부합된다고 할 수 있다 (즉, 어느 회사의 어떤 제품에 대한 정보만 연고자 필터링 한다).

표 2는 태그의 Domain Manager 필드와 Object Class 필드에 대해 D1, D2로 단순화 한 리드 4개의 리드 필터들이다. 리드 필터들은 프리픽스(Prefix) 부분과 Don't care 부분(*)으로 표현할 수 있다.

표 2. 태그의 두 가지 필드를 2차원 공간에서 필터링 할 6개의 리드 필터 프리픽스(Prefix)

리드 필터	D1	D2
f1	00*	11*
f2	10*	11*
f3	11*	10*
f4	01*	01*
f5	1*	0*
f6	0*	*

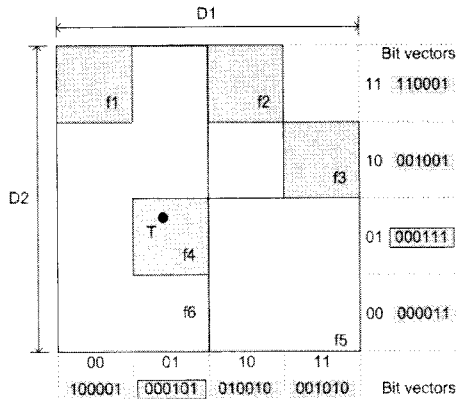


그림 2. 기하학적 관점에서 바라본 필터링

그림 2는 표 2의 6개의 리드 필터들을 차원에 대한 비트맵(Bitmap)들로 대응하여 표현하였는데 Lakshman과 Stiliadis가 제안한 비트 벡터 (Bit-Vector) 알고리즘⁹⁾을 활용하여 기하학적으로 표현하였다. 각 차원은 비트 단위로 가장 긴 프리픽스의 길이(W) 만큼 세분화하여 필터들의 영역을 표현할 수 있다. 즉, W=2이므로 2⁰ 부터 2^W 까지 00, 01, 10, 11의 간격 (Interval)을 생성한다. 각 차원(D1, D2)에 대하여 생성된 간격에 리드 필터들의 프리픽스가 존재하는 영역을 표시하면 사각형 모양으로 표현될 수 있다. 사각형들이 모두 만들어지면 각 필터가 해당 구간에서 부합(Match)하는 유무를 해당 비트의 자리 수로 표현할 수 있다 (총 6개의 필터가 존재하므로 6 비트만큼의 벡터로 표현됨). 이렇게 필터들의 비트맵을 미리 준비해 두면, 예를 들어 실제 입력된 태그가 T (01100111)라 할 때 D1에서 01 구간의 비트 벡터(000101)과 D2에서 01 구간의 비트 벡터(000111)를 구할 수 있다. 그리고 수식 1의 비트단위 연산 000101 & 000111 = 000101을 통하여 {f4, f6}가 부합하는 필터들임을 알 수 있고 우선 순위에 의해 f4가 가장 부합하는 필터로 결정 된다.

2.3 무선 프로토콜 필터링과 Select 명령

RFID 리더 (또는 interrogator)들은 산재되어 있는 태그 정보를 읽어 들이기 위해 세 가지 기본 단계 (Select, Inventory, Access)를 거친다 (C1G2 표준 - ISO 18000-6 Type C와 호환)^[6]. Select 명령은 리더가 태그들에게 하나 이상의 Select 명령을 보내어 어떤 태그들이 인식(Inventory) 과정에 참여할 것인지 선택한다. 리더의 Select 명령에서 MemBank 필드는 태그의 메모리를 선택하고 Pointer 필드는 시작 비트를 찾아 길이(Length 필드) 만큼 마스크(Mask 필드) 하여 논리합(\cup), 논리곱(\cap), 논리부정(\sim)을 통해 부합하는지 부합하지 않는지 판단한다. 이 때 부합하는 태그들만 플래그(Flag)들이 세팅 될 수 있다.

플래그는 두 가지 플래그 - SL (Selected) 플래그와 Inventoried 플래그를 가지고 있다. SL 플래그는 참여시키거나 제외시키는 역할을 하는 반면 Inventoried 플래그는 참여한 태그가 4가지 세션(Session) 중 하나의 세션 값을 A 또는 B로 세팅 한다. 이렇게 SL (Selected) 플래그와 Inventoried 플래그가 세팅 되어 있는 태그들만 특정 리더로부터 Query 명령을 받아서 인식 과정에 참여할 수 있다.

2.4 무선 프로토콜 필터(F)와 리드 필터(f)의 관계

그림 3은 2차원 공간 (Euclidean space: R^2)에서 필터들을 기하적으로 도식하여 필터링의 관계를 설명하기 위한 그림이다. 그림 3의 좌측 예에서와 같이 만약 리더나 미들웨어에서 $\{f_1, f_2, \dots, f_6\}$ 의 리드(Read) 필터들이 존재한다면 각각의 리드 필터를 직사각형으로 표현할 수 있음을 그림 2에서 살펴보았다. 오른쪽 그림은 이 리드 필터들에 대해서 선별적인 리더의 Select 명령을 이용하여 생성된 무선 프로토콜 필터 $\{F_1, F_2\}$ 들을 이용하여 불필요한 부분(Redundant Area: ϵ)을 무선 프로토콜 구간에서 미리 제거하는 예이다. 이러한 과정을 통하여 특정 태그들을 선별해 내는 과정을 무선 프로토콜 필터링이라고 한다. 본 연구는 무선 프로토콜 구간에서 사용되는 필터를 대문자 F로 정의하고 리더와 미들웨어에서 사용되는 필터를

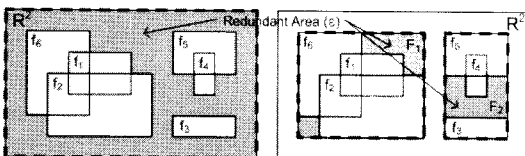


그림 3. (좌측) 리드 필터 $\{f_1, f_2, \dots, f_6\}$ 에 대해 (우측) 무선 프로토콜 필터 $\{F_1, F_2\}$ 가 사용된 예제

소문자 f로 정의하여 의미를 구분하였다.

III. 계산 기하학을 활용한 RFID 무선 프로토콜 필터들의 생성

앞 장에서 설명한 바와 같이 RFID 무선 프로토콜 필터들은 리더의 Select 명령을 활용하여 전체 태그들을 선별적으로 선택 한다. 이 때 인식이 불필요한 영역(ϵ)을 제거하기 위한 방법 중 가장 쉽게 생각할 수 있는 것은 리드 필터들을 둘러싸고 있는 바운딩 박스(Bounding Box)를 구하여 외곽의 불필요한 영역을 제거하는 방법일 것이다. 그런데 만약 다수의 Select 명령을 사용하여 리드 필터들을 더 정교하게 선택한다면 불필요한 영역을 더 줄일 수 있을 것이다. 그러나 Select 명령의 처리에 시간 비용이 발생하므로 하나의 Select 명령이 걸리는 시간에 비례하여 처리 시간이 선형적으로 증가하게 된다. 또한 다수의 Select 명령이 서로 겹쳐져 새롭게 중복되는 영역(Duplication Area: δ)을 발생시킬 수 있다. 즉, 이상적인 선택이란 가장 적은 Select 명령 횟수로 중복 영역과 불필요한 영역을 최소화하는 것이라고 할 수 있을 것이다. 본 장에서는 기하학적인 도식을 통하여 무선 프로토콜 필터를 어떻게 생성할지에 대한 알고리즘을 제안한다.

3.1 알고리즘 데이터 구조

우리는 알고리즘의 자료 구조로써 Bentley가 제안한 2차원 공간에서 세그먼트 트리(Segment Tree)를 선택하였다^[10]. 그림 4의 세그먼트 트리는 각 차원 마다 하나씩 생성하는데 생성된 세그먼트 트리를 각각 가로 탐색선(Scan Line)과 세로 탐색선으로 라인 스위핑(Line Sweeping) 하여 교차점(Intersection)을 계산함으로써 각각의 세그먼트들이 존재하는 구간에 관한 정보를 얻을 수 있다. 이 때 라인 스위핑은 탐색 선이 사각형의 꼭지점(Vertex)를 만났을 때 마다 수행된다. 트리 구조를 사용함으로써 가능한 시간복잡도 $O(\log n)$ 를 얻을 수 있으며 저장복잡도 $O(n \log n)$ 를 얻을 수 있다.

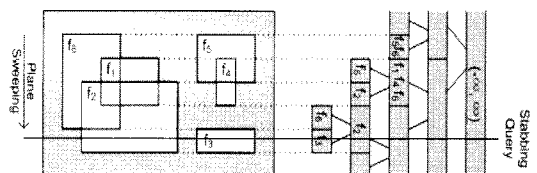


그림 4. Bentley가 제안한 2차원에서의 세그먼트 트리 알고리즘과 탐색선을 이용한 교차점 확인

3.2 알고리즘 1단계: 바운딩 박스 구하기 (BB)

알고리즘 1단계는 단순히 바운딩 박스(BB: Bounding Box)를 구하는 것이다. 그림 5의 왼쪽 예와 같이 필터링하고자 하는 리더 필터들이 가운데로 모여 있다면 간단하면서도 효과적으로 불필요한 공간을 제거할 수 있다. 그러나 오른쪽 예에서와 같이 리더 필터들 중 하나가 각 차원의 영역 가장 자리에 걸쳐 있다면 알고리즘이 얻을 수 있는 효과는 없을 것이다.

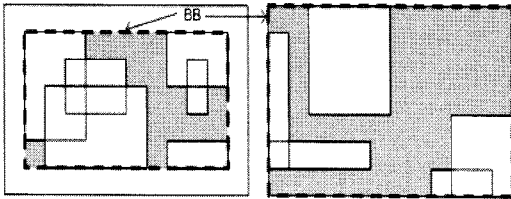


그림 5. 바운딩 박스 (F_1)를 구하는 두 가지 예

3.3 알고리즘 2단계: 독립된 바운딩 박스 구하기 (IBB)

알고리즘 2단계는 1단계의 단점을 보완하고자 고안한 방법으로 독립된 바운딩 박스 구하기(IBB - Isolated Bounding Boxes)이다. 이 방법은 다수의 Select 명령을 통하여 다수의 선택 영역을 구하는 것인데 선택하는 기준은 중첩(Overlay)된 리더 필터들의 고립성(Isolation) 여부이다. 이 때 그림 7에서와 같이 각각의 세그먼트 트리에서 라인 스위핑을 통해 사각형의 변(edge)을 만날 때 마다 교차점 정보를 계산하여 중첩된 사각형들의 정보를 구한 후 독립된 바운딩 박스를 그릴 지의 여부를 판단한다.

알고리즘 2단계의 이득은 그림 6의 왼쪽 그림에서와 같이 불필요한 영역을 다수의 Select 명령을 통해 추가적으로 제거할 수 있다는 것이다. 그러나 다수의 Select 명령은 그 만큼의 추가적인 시간 부하(overhead)를 가져오며 오른쪽 그림에서와 같이 새로운 중복 영역을 발생시킬 수 있는 가능성을 가지므로 리더 필터들의 생김새에 따라 오히려 알고리즘의 성능 저하를 보일 수 있다.

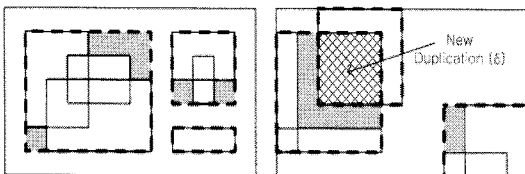


그림 6. 다수의 Select 명령을 이용하여 독립된 바운딩 박스들을 구하는 예

3.4 알고리즘 3단계: 면적 비교를 통한 독립 바운딩 박스 세분화 하기 (IBBSS)

알고리즘 3단계 역시 2단계의 단점을 보완하기 위해 제안한 방법으로 면적 비교를 통한 바운딩 박스를 세분화 하는 알고리즘 (IBBSS - Isolated Bounding Boxes with Separate Selects)이다. 이 방법은 기존의 불필요한 영역과 새롭게 중복되는 영역과의 크기를 비교하여 세분화된 바운딩 박스를 독립적으로 구할 것인지 아닌지를 결정한다.

그림 7은 1단계부터 3단계 알고리즘 까지 수행한 결과를 보여준다. 그림 6에 비하여 더 많은 Select 명령을 사용한 반면 더 적은 불필요한 공간(δ)과 새로이 중복되는 공간(ϵ)을 가지게 되었다. 하나의 격자를 면적 1이라고 한다면 왼쪽 그림은 4번의 Select 명령을 통한 면적 $\delta + \epsilon = 6$ 의 불필요한 공간을 가지고 오른쪽 그림은 4번의 Select 명령과 면적 $\delta + \epsilon = 3$ 의 불필요한 공간을 가진다. 그러나 공간의 면적만으로 비교하였을 때 두 그림 모두 전체 필터링 영역에 비해 상당 부부의 불필요한 리더 필터링 영역을 무선 프로토콜 필터링을 통하여 제거하였다고 볼 수 있다. Select 비용에 대한 계산은 다음 장에서 설명한다.

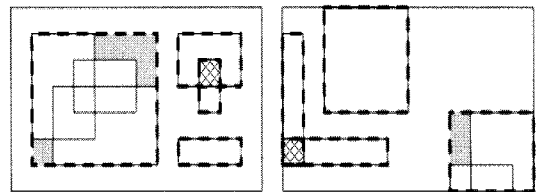


그림 7. 면적 비교를 통한 바운딩 박스 세분화 하기. 1단계부터 3단계 알고리즘 까지 수행한 결과까지 수행한 결과

IV. 확률론을 이용한 태그 필터링 모델링

본 장에서는 계산 기하학을 사용하여 생성한 RFID 무선 프로토콜 필터들의 효과를 살펴보기 위하여 필터링을 확률적으로 모델링하고자 한다. RFID C1G2^[6]에서 태그가 인식되는 과정은 Select, Inventory, Access 단계에서 순차적으로 진행된다. 기하학적으로 생성한 무선 프로토콜 필터들은 Select 단계에서 작용하여 다음 Inventory 단계에서 태그 인식 충돌 방지(Anti-Collision)를 위해 입력되는 태그 ID의 양을 상당히 줄여 줄 수 있다. 그러나 다수의 Select 명령은 하나의 Select 명령을 처리하는데 걸리는 시간에 대해 선형적인 증가를 가져오므로 필터링 시간 모델을 세워 성능을 비교해보려고 한다.

4.1 확률 기반의 프레임 크기 결정

태그를 인식하기 위한 프레임 크기(N)와 전체 태그의 수(n)에 대하여 태그 간 충돌을 확률적인 계산을 통해 모델링 한다 (Random-Slotted Collision Arbitration). 충돌 해결을 위한 프레임 크기(N)와 전체 태그의 수(n)의 결정은 S. R. Lee et al^[11]의 $N = n + 1$ 을 기초로 모델링을 시작한다. 그리고 결과에 대하여 태그를 (1)인식한 경우, (2) 충돌한 경우, (3)응답이 없는 경우로 나누어 확률적인 모델링을 완성한다.

$a_r^{N,n}$ 을 프레임 크기 N과 인식하지 못한 태그 수를 n이라 할 때 시스템 효율(System Efficiency)은 수식 2와 같이 계산된다.

$$\begin{aligned} \text{시스템 효율}(SE) &= \frac{\text{슬롯 당 1개의 태그만 접근한 슬롯}}{\text{현재 프레임의 크기}} \\ &= \frac{a_1^{N,n}}{N} \end{aligned} \quad \text{수식 2}$$

시스템 효율의 최대치를 구하기 위해 수식 2를 태그 수 n에 관하여 미분하면,

$$\begin{aligned} \frac{d(SE)}{dn} & \quad \text{수식 3} \\ &= \left(1 - \frac{1}{N}\right)^{n-1} + n \left(1 - \frac{1}{N}\right)^{n-1} \ln\left(1 - \frac{1}{N}\right) = 0 \end{aligned}$$

수식 3을 전개하여 정리하면 프레임 크기 N과 태그 수 n의 관계식 수식 4를 구할 수 있다^[11].

$$\begin{aligned} N &= \frac{1}{1 - e^{-\frac{1}{n}}} = \frac{e^{\frac{1}{n}}}{e^{\frac{1}{n}} - 1} \\ &\approx \frac{1 + \frac{1}{n}}{1 + \frac{1}{n} - 1} = n + 1, \quad n \gg 1 \end{aligned} \quad \text{수식 4}$$

결과적으로 n이 충분히 클 때 인식을 위한 프레임의 크기 N도 충분히 커야지만 최대 성능을 발휘할 수 있음을 알 수 있다. 앞으로 우리는 $N = n + 1$ 의 관계식을 태그 인식 과정에 대한 확률적 시뮬레이션의 평가를 위한 계산에 활용한다.

4.2 인식한 경우, 충돌한 경우, 응답이 없는 경우의 확률

일반적으로 임의의 수 (Random Variable) R이 이항 분포 (Binomial Distribution)를 따를 때 $R \sim B(n, p)$, r 횟수 만큼 성공할 확률 질량 함수 (Probability Mass Function)는 아래와 같다. 이 때 성공할 확률은 p는 $\frac{1}{N}$ 이다.

$$B_{n, \frac{1}{N}}(r) = \binom{n}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{n-r} \quad \text{수식 5}$$

한 번의 리더 명령에서 하나의 태그가 응답하여 인식에 성공할 확률은 p_1 , 태그 응답이 없어서 인식에 실패할 확률은 p_0 이며 수식 6과 7에서 각각 모델링할 수 있다.

$$p_1 = B_{n, \frac{1}{N}}(1) = \frac{n}{N} \left(1 - \frac{1}{N}\right)^{n-1} \quad \text{수식 6}$$

$$p_0 = B_{n, \frac{1}{N}}(0) = \left(1 - \frac{1}{N}\right)^n \quad \text{수식 7}$$

수식 4에서 $N = n + 1$ 을 프레임 윈도우 크기로 사용하였으므로 대입하여 $B_{n, \frac{1}{N}}(1)$ 과 $B_{n, \frac{1}{N}}(0)$ 을 구할 수 있고 계산에 의해서 두 값은 서로 같다는 것을 알 수 있다. 따라서 충돌할 확률 p_k 는 자동적으로 수식 8에서 구해질 수 있다 (확률 p_k 는 충돌 외 에러를 비롯한 p_1 과 p_0 를 제외한 모든 상황을 포함한다고 가정).

$$\begin{aligned} p_1 &= p_0 \\ p_k &= 1 - (p_1 + p_0) \end{aligned} \quad \text{수식 8}$$

위 식들을 통하여 우리는 p_1, p_0, p_k 에 비율(ratio)을 수식 9를 이용하여 구할 수 있다. 따라서 이론적으로 리더의 프레임 윈도우 크기가 태그의 개수들 만큼 큰 $N = n + 1$ 가정에서 한 번의 리더 사이클 동안 인식 시도를 한 번만 수행한다면 약 37%의 인식률을 가진다.

$$\begin{aligned} p_1 : p_0 : p_k &= p_1 : p_0 : 1 - (p_1 + p_0) \\ &\approx 0.37 : 0.37 : 0.26 \\ p_1 : p_{(0,k)} &= p_1 : (1 - p_1) \approx 0.37 : 0.63 \end{aligned} \quad \text{수식 9}$$

4.3 모든 태그를 인식하기 위한 인식 시도 횟수 모델링

앞서 인식한 경우(p_1), 응답이 없는 경우(p_0), 충돌한 경우(p_k)에 대해 계산한 확률적인 비율 값을 가지고 한 번의 리드 사이클 동안 모든 태그를 인식하기 위해 필요한 인식 시도 횟수(c)를 구해 본다. n_c 를 c^{th} 번째 시도에 참여하는 아직 인식되지 않은 태그 수라고 가정하면 수식 10과 같이 인식 시도 횟수에 따른 과정을 모델링 할 수 있다.

$$\begin{aligned} n_1 &= \lfloor n \rfloor \\ n_2 &= \lfloor n_1 \rfloor (1-p_1) \\ &\dots \\ n_c &= \lfloor n_{c-1} \rfloor (1-p_1) = 0 \end{aligned} \quad \text{수식 10}$$

수식 10은 기하 급수(Geometry Series)의 형태를 가지므로 수식 11의 좌변으로 일반화 할 수 있고 우변에 임의로 1을 더해서 c 를 구하고자 하였다.

$$n(1-p_1)^{c-1} = 1 \quad \text{수식 11}$$

$$c = \left\lceil \frac{\ln\left(\frac{1}{n}\right)}{\ln(1-p_1)} \right\rceil \quad \text{수식 12}$$

따라서 필요한 모든 태그를 인식하기 위해 필요한 총 인식 시도 횟수는 수식 12를 이용하여 구할 수 있으며 앞서 계산한 p_1 , p_0 , p_k 확률 값들과 함께 필터링이 소비하는 시간 비용을 모델링 하는데 사용한다.

V. 태그 인식과 필터링 소요 시간 모델링

본 장에서는 RFID 태그 인식과 리더 혹은 미들웨어에서 수행되는 필터링까지 소요되는 전체 시간을 모델링 한다. 총 소요되는 시간은 무선 프로토콜 과정에서 소요되는 시간과 인식 후 리더 혹은 미들웨어에서 수행하는 필터링에 소요되는 시간으로 구분 가능하다. 따라서 수식 13에서 τ_s 는 태그 Select 과정에 소비되는 시간, τ_i 는 인식(Inventory) 과정에서 소비되는 시간, 그리고 τ_f 는 RFID 리더 혹은 미들웨어에서 필터링에 수행되는 시간의 합을 의미한다.

$$\tau = \tau_s + \tau_i + \tau_f \quad \text{수식 13}$$

5.1 Select 단계에서 소비되는 시간(τ_s) 추정

Select 단계에서 소비되는 시간(τ_s) 추정하기 위해서 C1G2 표준⁶⁾에 의하여 Select 명령을 두 번 사용하는데 걸리는 시간으로 모델링 하였다. 첫 번째 Select는 태그의 SL (Selected) 플래그를 이용하여 인식에 참여할 태그를 선택하기 위한 명령이다. 두 번째 Select 명령은 Inventoried 플래그의 A와 B 상태를 변경하는데 필요한 명령이다⁶⁾.

그림 8은 두 번의 Select 명령을 사용하여 인식에 참여할 모든 태그를 선택하는 과정을 나타낸다. 즉, 한 번의 Select 명령을 처리하는데 걸리는 시간을 θ 라 한다면 총 2θ 만큼의 시간을 하나의 무선 프로토콜 필터(F)가 소비한다. 따라서 다수의 무선 프로토콜을 사용하면 필터(F) 수 공급하기 2θ 만큼의 시간을 소비한다고 할 수 있다.

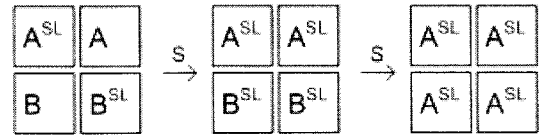


그림 8. Select 단계에서 두 번의 Select 명령을 사용하여 모든 태그 선택

5.2 인식 단계에서 소비되는 시간(τ_i) 추정

인식(Inventory) 단계에서 소비되는 시간(τ_i)를 추정하기 위해서 C1G2 표준 그림 9의 소요 시간에 대한 권고 사항⁶⁾에 따라 모델링 하였다. 인식 단계의 첫 번째 인식 시도에서 소요되는 시간을 τ_{i1} 이라고 할 때 ω_1^Q , ω_0^Q , ω_k^Q 를 그림 9로부터 수식 14를 표현할 수 있다. 인식 단계의 두 번째 인식 시도에 걸리는 시간 τ_{i2} 부터는 Query 명령 대신에 QueryRep (Query Repeat) 명령이 사용되므로 ω_1^{QR} , ω_0^{QR} , ω_k^{QR} 를 그림 9로부터 구하여 수식 15에 정리하였다.

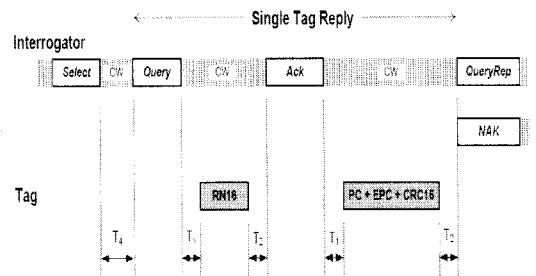


그림 9. C1G2 표준에서 권고하는 리더와 태그 사이의 명령과 응답에 따른 시간

$$\tau_{r_1} = np_1\omega_1^Q + np_0\omega_0^Q + np_k\omega_k^Q \quad \text{수식 14}$$

$$\tau_{r_2} = n_1p_1\omega_1^{QR} + n_1p_0\omega_0^{QR} + n_1p_k\omega_k^{QR} \quad \text{수식 15}$$

따라서 인식 단계의 두 번째 인식 시도부터 모든 태그를 인식하기 위해 c^{th} 까지 시도한다면 걸리는 시간은 수식 16으로 나타낼 수 있다.

$$\tau_{r_c} = n_{c-1}p_1\omega_1^{QR} + n_{c-1}p_0\omega_0^{QR} + n_{c-1}p_k\omega_k^{QR} \quad \text{수식 16}$$

그러므로 인식 단계에서 소요되는 총 시간은 수식 17과 같다.

$$\tau_i = \tau_{r_1} + \sum_{j=2}^c \tau_{r_j} \quad \text{수식 17}$$

5.3 리드 필터링에 소비되는 시간(τ_f) 추정

리드 필터링에 소비되는 시간(τ_f)을 추정하기 위해서는 먼저 리드 필터링을 RFID 리더에서 직접 수행하는 경우와 미들웨어에서 집중적으로 수행하는 경우로 나누어 볼 수 있다. 따라서 우리는 무선 프로토콜 구간을 지난 후 RFID 리더 이상의 계층에서 일어나는 필터링은 해당 시스템 상황에 매우 의존적이므로 하나의 리드 필터(f)에 대해 소비되는 시간을 상수 α 로 두었다. 상수 α 는 해당 리더 혹은 미들웨어에서 어떤 알고리즘을 사용하는지에 따라 달라질 것이지만 우리는 크게 특별한 알고리즘 없이 순차적으로 필터링을 수행하는 경우와 최대 $O(n \log n)$ 의 탐색 성능을 가지는 트리 기반의 알고리즘을 사용하여 필터링 하는 경우의 2가지로 구분하여 수식 18에 정리하였다 (m 은 리드 필터(f)들의 개수).

$$\begin{aligned} \text{순차적 검색 } \tau_f &= \alpha mn \\ \text{트리기반 검색 } \tau_f &= \alpha n \log m \end{aligned} \quad \text{수식 18}$$

VI. 시뮬레이션과 평가

본 장에서는 앞 장에서 구한 무선 프로토콜 구간에서 부터 리더를 거쳐 미들웨어 응용까지 구한 시간 모델을 적용하여 시뮬레이션을 수행하고 무선 프로토콜 필터(F)의 성능을 검증할 것이다.

6.1 총 소요시간(τ)에서 τ_s , τ_i , τ_f 비율 비교

태그 인식과 필터링에 소요되는 총 시간(τ)의 측정

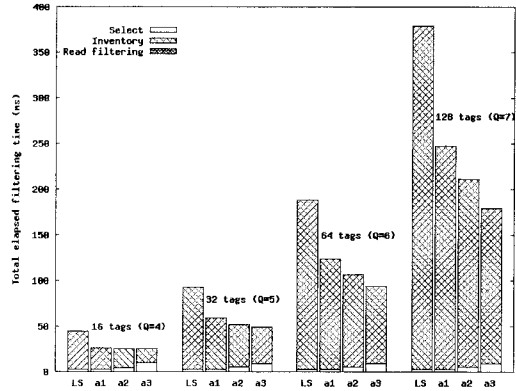


그림 10. 단일 리더 환경에서의 필터링 소요 시간 비교

과 τ_s , τ_i , τ_f 비율 비교를 위하여 우리는 지난 연구 결과를 토대로 최소한의 상수 값으로 $\alpha = 0.01\mu s$ 로 설정하였다.

그림 10은 모델링을 구현한 시뮬레이션을 수행하여 구한 결과 값으로 하나의 RFID 리더가 5개의 리드 필터를 가지고 $2^4, 2^5, 2^6, 2^7$ 개수의 태그들을 인식하는데 걸린 시간을 비교하였다. 히스토그램의 τ_i 가 가장 높은 비율을 차지하였는데 원인은 한 리드 사이클 당 모든 태그를 인식하기까지 인식 시도를 수행한다고 가정하였기 때문으로 판단된다. 또한 리드 필터링에 소요되는 τ_f 가 아주 낮은 비율로 그래프에 드러나지 않는 이유는 최소한의 상수 값으로 $\alpha = 0.01\mu s$ 를 매우 낮게 설정하였고 단일 리더 환경의 실험으로 리드 필터링의 부하가 크지 않은 상황이기 때문이다.

제시한 무선 프로토콜 필터링 알고리즘을 단계적으로 수행한 경우 순차적 검색(linear search)으로 필터링 한 경우와 비교하여 약 30% 정도 필터링 시간을 절약할 수 있음을 알 수 있다. 따라서 무선 프로토콜 구간에서 수행하는 알고리즘이 다수의 Select 명령으로 추가적인 비용을 발생시키더라도 인식과정에 필요한 비용과 비교하였을 때 상대적으로 낮은 시간 비용을 지불한다는 것을 알 수 있다. 왜냐하면 무선 프로토콜 필터(F)는 리드 필터(f)에 대해 구한 값이므로 태그의 수가 증가하더라도 필터 수에는 변함이 없기 때문이다 (즉, 동일한 시간 비용을 지불한다)

그림 11은 그림 10의 실험에서 인식 단계에서 필요로 한 총 리드 시도 횟수를 비교하였다. 왜냐하면 τ_i 가 전체 시간 비용(τ)에서 가장 높은 비중을 차지하였기 때문이다. 이 그림에서도 무선 프로토콜 구간에서 수행한 알고리즘이 인식 단계에 참여하는 태그 수를 줄여 주기 때문에 알고리즘의 단계가 높아질수록 순

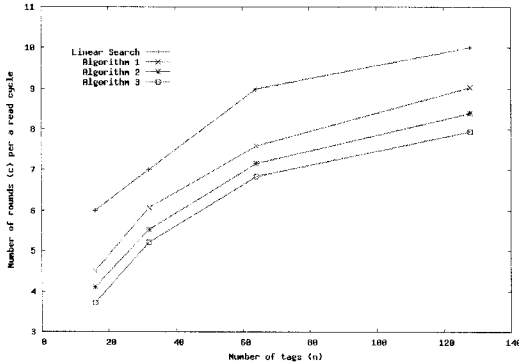


그림 11. 그림 10의 실험에 대한 인식 단계에서 인식 시도 횟수의 비교

차적 검색의 필터링에 비해 적은 수의 인식 시도를 보여준다.

6.2 밀집한 RFID 환경에서의 시뮬레이션

첫 번째 실험은 태그 수와 리더 필터 수가 많지 않은 상황에서 리더 한 대가 수행하는 필터링을 무선 프로토콜 필터링과 대비하여 그 효과를 비교하였다. 두 번째 실험은 많은 양의 태그들에 대해 다수의 RFID 리더가 존재하는 밀집한 RFID 환경에서 다수의 필터를 가지고 동작하는 상황을 가정하였다. 그림 12는 32개의 임의로 배치되어 있는 RFID 리더들이 2¹¹, 2¹², 2¹³, 2¹⁴의 태그 수에 대해 1024개의 리더 필터(f)를 가지고 실험한 결과이다.

그림 12에서 많은 수의 태그들에 대하여 단순히 바운딩 박스만을 구하는 첫 번째 단계만으로는 성능을 크게 향상시킬 수 없었다. 그러나 알고리즘을 2단계까지 적용하였을 때 약 40% 정도의 시간이 절약되었으며 3단계까지 적용하였을 경우 약 50% 가량 성능 향상을 나타내었다. 또한 다수의 RFID 리더가 존재하므로 다수의 리더로부터 하나의 미들웨어로 인식한 태

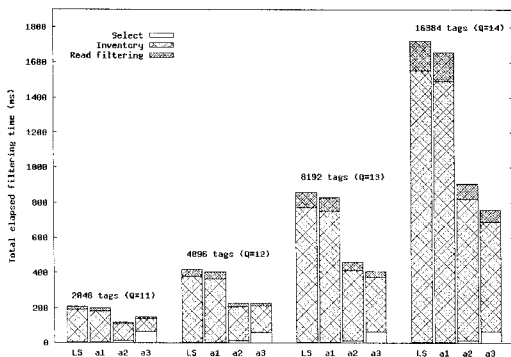


그림 12. 밀집한 RFID 환경에서의 필터링 소요 시간 비교

그 정보가 집중하여 리더 필터링을 수행하는데 걸리는 시간 τ_f 가 증가함을 보였다.

VII. 향후 연구와 결론

본 논문은 유비쿼터스 환경으로 변해 가는 과도기에서 사물에 유일한 ID를 부여하여 자동으로 인식, 인지하는 기반 기술을 제공하기 위한 방법에 주목하였다. 그 대표적인 예로 RFID 기술에서 필터링의 중요성을 부각하였고 특히 무선 프로토콜 구간에서의 필터링에 초점을 맞추어 계산 기하학적인 알고리즘을 적용하여 보였다. 제한한 알고리즘은 시뮬레이션을 통하여 밀집한 RFID 환경에서 일반적인 순차적인 필터링에 비해 약 절반 가량 필터링 성능을 개선한 예를 제시하였다. 본 논문이 기존 연구에 대해 기여할 수 있는 바를 아래와 같이 정리하였다.

- RFID 시스템에서 필터링이 발생할 수 있는 과정에서 무선 프로토콜 구간에서 필터링을 수행하는 방법론을 제시하였다. 무선 프로토콜에서의 필터링은 상위 계층에 전달되는 태그의 양 자체를 줄여 줄 수 있다는 장점을 시뮬레이션을 통해 보였다.
- RFID 인식 과정에서 인식, 불인식, 충돌 과정을 확률적인 방법을 이용하여 계산하였고 인식과 필터링에 소비하는 시간을 모델링하여 시뮬레이션하는 방법론의 예를 제시하였다.

참 고 문 헌

- [1] P. Gupta, N. McKeown, "Algorithms for Packet Classification", *IEEE Network*, March-April 2001, Vol. 15, No. 2, pp. 24-32.
- [2] D. Eppstein, S. Muthukrishnan, "Internet packet filter management and rectangle geometry", *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, Washington 2001, pp. 827-835.
- [3] EPCglobal, "The EPCglobal Architecture Framework", Final Version of 1 July 2005.
- [4] EPCglobal, "The Application Level Events (ALE) Specification", Version 1.0, February 2005.
- [5] Y. Bai, F. Wang, P. Liu, "Efficiently Filtering RFID Data Streams", *In Proc. of the First International VLDB Workshop on Clean*

Databases (CleanDB'06), Seoul, Korea, September 11, 2006.

[6] EPCglobal, "EPC™ Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz", Version 1.1.0, December 17, 2005.

[7] EPCglobal, "Tag Data Standards", Version 1.3, Standard Specification, September 9, 2005.

[8] EPCglobal, "Reader Protocol 1.1", Recommended Specification, April 10, 2006.

[9] T. V. Lakshman, D. Stiliadis, "High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching", *Proceedings of ACM SIGCOMM*, September 1998, pp. 191-202.

[10] M. de Berg, M. van Kreveld, M. Overmars. "Computational Geometry: Algorithms and Applications", *Springer-Verlag*, 2nd rev. ed. 2000.

[11] S. R. Lee, S. D. Joo, C. W. Lee, "An enhanced dynamic framed slotted aloha algorithm for RFID tag identification", *Proceedings of MobiQuitous*, 2005, pp. 166-174.

[12] H. S. Park, J. D. Kim, "Design and Implementation of a High-Speed RFID Data Filtering Engine", *EUC Workshops, LNCS*, Vol. 4097, pp. 423-434, Springer, 2006.

[13] 권성호, 모회숙, 최길영, 표철식, 채종석, "Gen2 기반 RFID시스템에서의 충돌방지 알고리즘에 관한 연구", *한국통신학회논문지*, 제6B호, 2006, pp. 561-571.

박 현 성 (Hyun-Sung Park)

정회원



1999년 2월 경성대학교 산업공학과 졸업

2007년 8월 부산대학교 컴퓨터공학과 석사

2007년 9월~현재 부산대학교 컴퓨터공학과 박사과정

<관심분야> 무선통신, RFID/USN

김 종 덕 (Jong-Deok Kim)

종신회원



1994년 2월 서울대학교 계산통계학과 졸업

1996년 2월 서울대학교 전산학과 석사

2003년 2월 서울대학교 컴퓨터공학과 박사

2004년 2월~현재 부산대학교

정보컴퓨터공학과, 조교수

<관심분야> 무선통신, 이동통신망, RFID/USN