
RFID 상황인식 시스템을 위한 UPnP 서비스

김동균*, 전병찬**, 이상정***

UPnP Services for RFID Context-Aware System

Dong-Kyun Kim *, Byung-Chan Jeon**, Sang-Jeong Lee***

요 약

본 논문에서는 UPnP를 이용하여 RFID 서비스 발견 및 제어를 할 수 있는 상황인식 서비스 시스템을 제안한다. 제안된 시스템은 자원이 부족한 RFID 태그의 서비스를 위한 미들웨어 플랫폼으로 UPnP를 적용한다. UPnP를 적용하여 시스템을 구축하면 상황인식 서비스의 개발이 용이하며 별도의 설정 없이도 RFID 디바이스 서비스를 제공할 수 있다. 또한, RFID의 상황인식 데이터의 신뢰도를 높이기 위하여 SQL 기반의 클리닝(cleaning) 모듈을 설계하고 구현한다. 제안 개발된 클리닝 모듈을 적용하여 RFID 시스템의 태그에 대한 순수 인식률을 60~80% 정도에서 98% 이상으로 향상 시켰다.

제안된 UPnP 기반 RFID 상황인식 서비스 검증을 위하여 물류서비스에 대한 상황인식 시나리오를 제시하고, RFID 서비스 아키텍처와 UPnP 서비스 발견 플랫폼을 구현 하였다. 또한, 구현된 시나리오 상에서 UPnP 서비스 광고 메시지의 네트워크 혼잡 기여도와 RFID 시스템의 태그에 대한 순수 인식률을 실험하고 분석하여 제안된 방식의 타당성을 검증한다.

ABSTRACT

In this paper, it is presented to utilize UPnP as RFID service discovery and control mechanism for context-aware services. Using UPnP, it is possible to achieve easy deployment of context-aware services and to provide zero-configuration for RFID services. In addition, SQL based cleaning module which raises detection rates is developed since context-aware applications heavily rely on streams of data gathered from RFID tags. Using the cleaning technique, detection rates are improved from 60-80% to 98% or more.

In order to verify RFID context aware service based on the UPnP, sample context-aware scenario for physical distribution services is implemented on UPnP over RFID system. The impacts of UPnP messages for service advertisements on network congestion and SQL cleaning module are experimented and analyzed, and the result show the good correctness and validity of the proposed system.

키워드

UPnP, 서비스, RFID, 상황인식

* 순천향대학교 컴퓨터학부 시간강사

** 청운대학교 컴퓨터학과 전임강사

*** 순천향대학교 컴퓨터학부 교수 (교신저자)

1. 서 론

상황인식 서비스는 특정한 사람, 공간, 시간, 이벤트 등의 주변 상황에 적합한 서비스를 제공한다. 즉, 인식된 상황에 기반하여 사용자와 서비스간의 연관성을 맺어 주는 것이다. 상황인식 서비스 시스템은 주변 상황을 인식하고 상황 정보를 수집 처리하여 이질의 네트워크들을 경유해서 각 사용자에게 서비스를 제공한다. 이를 위해서는 상황에 대한 인식기술이 확보되어야 하고 상황 인식 후 서비스의 제공이 사용자의 개입 없이 자동으로 이루어져야 한다.

객체에 대한 식별정보를 기반으로 객체의 관독, 추적, 관리를 하는 RFID 기술은 최근 제조 및 물류 유통 분야를 필두로 많은 분야에서 연구개발이 활발히 진행되고 있다[1]. RFID 시스템 활용 범위가 커질수록 사용자는 더 많은 RFID 태그들을 설정하고 관리해야 하는 문제가 발생한다. 또한, RFID의 다양한 응용에 따라 각각 종속적인 프로그램을 작성해야 한다. 이러한 문제는 UPnP(Universal Plug and Play)를 적용하면 해결할 수 있다[2]. 사용자는 RFID 태그의 특별한 설정 없이 사용자의 휴대용 기기(PDA)나 PC 등을 네트워크에 연결만 하면 즉시 RFID 서비스를 제공받을 수 있으며, 타 네트워크 및 타 시스템과의 상호운용을 용이하게 하여 다양한 응용에 대해 독립적으로 프로그램을 작성할 수 있다. 그러나, RFID 태그는 자원이 매우 부족하여 직접 RFID 서비스를 제공할 수 없으며, TCP/IP 상에서 동작하는 UPnP를 RFID 태그에 직접 적용할 수도 없는 문제점이 있다.

본 논문에서는 UPnP를 이용하여 RFID 서비스 발견 및 제어를 할 수 있는 상황인식 서비스 시스템을 제안한다. 제안된 시스템은 자원이 부족한 RFID 태그의 서비스를 위한 미들웨어 플랫폼으로 UPnP를 적용한다. UPnP를 적용하여 시스템을 구축하면 상황인식 서비스의 개발이 용이하며 별도의 설정 없이도 RFID 디바이스 서비스를 제공받을 수 있다. 또한, RFID의 상황인식 데이터의 신뢰도를 높이기 위하여 SQL 기반의 클리닝(cleaning) 모듈을 설계하고 구현한다. 제안 개발된 클리닝 모듈을 적용하여 RFID 시스템의 태그에 대한 순수 인식률을 60~80% 정도에서 98% 이상으로 향상 시켰다.

제안된 UPnP 기반 RFID 상황인식 서비스 검증을 위하여 물류서비스에 대한 상황인식 시나리오를 제시하고, RFID 서비스 아키텍처와 UPnP 서비스 발견 플랫폼

을 구현 하였다. 또한, 구현된 시나리오 상에서 UPnP 서비스 광고 메시지의 네트워크 혼잡 기여도와 RFID 시스템의 태그에 대한 순수 인식률을 실험하고 분석하여 제안된 방식의 타당성을 검증한다.

II. UPnP기반 RFID 상황인식

상황인식 서비스의 제공을 위해서는 주변 상황을 인식하는 기술과 인식된 정보를 수집하고 처리하며 처리된 데이터를 여러 노드와 이질의 네트워크를 경유하여 최종적으로 인간이 인지할 수 있는 형태로 제공해야 한다. 이러한 과정은 인간의 개입 없이(어떤 설정 없이) 자연스럽게 제공되어야 한다.

사용자 및 사물에 대한 상황 지원 환경을 제공하기 위해서는 적절한 시스템 플랫폼이 필요하다. 이는 서비스를 제공하는 서버와 같은 미들웨어 단계에서의 사용자 및 사물에 대한 에이전트 서비스의 일반적인 기능을 제공함으로써 해결된다. 미들웨어 단계에서의 기능 구성은 기본적인 상황인식을 위한 상황 관리 기술과 이벤트 발생 시 적절한 처리 절차를 가지는 이벤트 서비스 기술, 다양한 응용 단말을 위하여 네트워크 및 메시지에 대한 매핑의 효율적인 수행을 보증하는 브로커 서비스 기술 등으로 구성될 수 있다.

객체의 인식 기술로는 사물의 인식을 통해 존재 유무를 판단하는 RFID 기술이 각광받고 있으며, RFID 기술은 물류, 유통, 군사, 식품, 안전 등의 분야에서 광범위하게 사용되고 있다[1]. RFID 기술로 인식된 정보를 자동으로 발견하여 정보를 수집하고 이를 처리하는 미들웨어 기능이 필요한데 이를 위해서는 홈네트워크 분야에서 사용되는 UPnP 미들웨어가 대안이 될 수 있다[3]. UPnP는 표준 네트워크 아키텍처의 프로토콜에 의해 정의되고 특정 운영체제, 프로그래밍 언어, 혹은 물리적 매체 등에 독립적이고, 애플리케이션이 사용할 API를 지정하지 않기 때문에 벤더들은 자신의 필요에 맞는 API를 개발할 수 있는 특징을 가지고 있다[4].

본 논문에서는 상황인식 서비스를 제공하기 위하여 객체의 인식 기술로 RFID 시스템을 사용하고 인식된 상황에 맞는 서비스의 제공을 사용자의 개입 없이 자동 설정 및 자동 발견하기 위하여 UPnP를 적용한다.

UPnP 기반의 RFID 상황인식 서비스를 제공하기 위

해서는 선결되어야 할 두 가지 문제점이 있다. 첫째는 RFID 시스템의 태그 인식의 신뢰도를 확보하는 것이고 둘째는 인식된 상황에 맞는 디바이스 및 서비스를 제공하기 위하여 UPnP를 사용하는데 있어서 RFID 태그들은 매우 제한적인 자원의 한계를 가지고 있다는 점이다.

상황인식에 대한 신뢰도 확보는 중요하다. 왜냐하면 모든 상황인식 서비스는 상황인식을 기반으로 서비스하기 때문에 잘못된 상황인식으로 인하여 원하지 않는 서비스가 제공될 수 있다. 그리고 상황인식 서비스에 대한 전반적인 신뢰도를 확보할 수 없으며 사용자에게는 서비스 제공에 따른 혼란만 가중될 수 있다. 본 논문에서 인식 기술로 사용하는 RFID 시스템의 인식 성공률은 60~80%이다. 이와 같은 수준의 인식률로는 상황인식 서비스의 신뢰도를 보장할 수 없다. 따라서 RFID 태그에 대한 인식률을 향상시키는 방법이 제시되어야 한다.

상황인식 자체의 신뢰도가 중요하기 때문에 기존의 상황인식 서비스들은 대부분 어떤 상황을 인식하고 신뢰도를 높이며 인식된 상황에 맞는 서비스의 제공에 초점이 맞춰져 연구 및 개발이 이루어지고 있다[5, 6, 7, 8, 9]. 그러나, 기존 연구에서는 상황인식 자체는 자동으로 이루어지지만 서비스의 제공은 수동으로 이루어진다. 즉, 서비스의 제공을 위해서는 사용자에게 서비스에 대한 사용자 설정(서비스 설정, 서비스 프로그램 인스톨 등)을 요구한다. 사용자의 개입 없이 자동으로 상황을 인식하고 인식된 상황에 맞는 서비스를 자동으로 발견하며 이를 서비스할 수 있는 시스템이 필요하다. 따라서 본 논문에서는 서비스를 자동으로 발견하고 이 서비스를 제공하기 위하여 UPnP를 적용한다. 그러나, UPnP는 TCP/IP 상에서 운용될 수 있으므로 매우 제한적인 자원의 한계를 가지고 있는 RFID 태그들에서는 운용이 불가능하다. RFID 시스템을 UPnP와 연동하려면 RFID 태그들을 제어 관리할 수 있는 방법이 필요하다.

III. 서비스 시나리오 및 시스템 설계

본 장에서는 서비스 시나리오와 RFID 상황인식 시스템을 위한 UPnP 서비스 아키텍처를 설계하고 제안한다. 제안된 아키텍처에는 WMS (Warehouse Management System) 서버, RFID 리더 및 안테나, 유무선 AP, PDA 및 PC 등으로 구성된다. WMS 서버에는 RFID 데이터를 저

장하는 데이터베이스와 RFID UPnP 디바이스, 웹 프로그램 및 웹 서버로 구성된다. 설계 제안된 UPnP 서비스 디바이스 및 컨트롤포인트의 동작을 설명한다.

3.1 서비스 시나리오

설계 제안하는 물류관리 시스템은 물류의 총괄 처리를 담당하는 WMS 서버와 데이터베이스, 제품의 물류를 실시간으로 확인할 수 있는 실시간 모니터링 PC 및 PDA, 제품을 인식하기 위한 RFID 리더/안테나 및 제품에 부착된 태그와 유무선 네트워크를 지원하는 무선 AP(Access Point)들로 구성된다.

그림 1은 물류 창고관리 시스템의 작업 흐름을 나타내는 것이다. 공장 내의 직원들은 RFID 태그가 내장된 사원증을 모두 착용하고 있는 것으로 가정하고 공장 내부의 주요 위치(출입구, 지게차 등)에는 RFID 리더 및 안테나가 배치되어 있어 태그(제품, 직원)의 움직임을 실시간으로 모니터링하고 있다고 가정한다. 또한, 공장 내부는 유무선 네트워크가 갖추어져 있으며 RFID를 위한 UPnP 디바이스가 구축되어 있다고 가정한다.

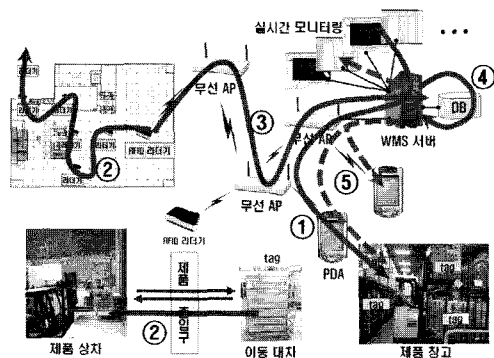


그림 1. 물류 창고관리 시스템의 작업 흐름도
Fig. 1 Processing Flow for Warehouse Management System

그림 1의 ①은 전체 공장의 물류를 관리하는 관리자가 자신의 컴퓨터에서 작업자에게 제품 출하를 요청한다. 관리자의 컴퓨터에서 출하 요청을 하면 출하 요청 명령이 WMS 서버에 전달되고 이를 서버에서 해석하여 관리자가 원하는 제품의 위치와 그 위치에서 가장 가까운 위치의 작업자에게 작업을 지시하는 명령을 무선 AP들을 경유하여 전달한다. 작업자는 전달 받은 출하 요청을

보고 자신이 출하해야 할 제품의 위치를 PDA에서 확인하고 그 위치로 이동한다. 출하할 제품을 지게차에 부착된 RFID 리더로 다시 읽는 것으로 확인하고 이 제품을 출입구를 통하여 출하장으로 이동한다.

그림 1의 ②는 작업자의 지게차가 제품을 출하장까지 이동하는데 여러 출입구를 통과하게 되는 것을 나타낸다. 이때 각 출입구에 설치된 리더들이 계속해서 출하될 제품의 이동경로를 추적하게 된다. 리더들은 제품에 부착된 태그를 인식하여 실시간으로 WMS 서버와 데이터베이스에 정보를 전송한다. 출입구에는 태그의 인식률을 높이기 위하여 최소 2개 이상의 안테나가 배치된다.

그림 1의 ③은 리더에서 인식된 제품(태그)의 정보를 WMS 서버와 데이터베이스에 전송하는데 공장 내에 구축된 무선 AP들을 경유하여 전달한다. 전달되는 태그의 정보는 아무런 처리 없이 데이터베이스에 저장된다.

실시간으로 리더에서 전송되어 데이터베이스에 저장된 데이터는 태그에서 읽혀진 원본 데이터이다. 이 원본 데이터는 각 안테나에서 제품을 각각 인식한 것에 대한 데이터이다. 이것을 그대로 분석해 보면 출입구를 통과하는 제품(태그)의 20% 정도가 인식되지 않는다. 따라서 데이터베이스에 저장된 원본 데이터를 WMS 서버가 읽어 클리닝 데이터 처리 후 다시 데이터베이스에 인식이 향상된 제품 데이터를 저장한다. 이것을 나타낸 것이 그림 1의 ④이다.

그림 1의 ⑤는 처리된 결과를 WMS 서버가 최종적으로 관리자 및 작업자에게 알려 주는 것을 나타내고 있다. 이것은 일반 프로그램으로 처리하거나 UPnP 이벤트로 처리한다.

그림 2는 작업자가 출하될 제품이 아닌 다른 제품을 출하하거나 출하 명령이 없는 제품의 이동이 감지될 때 각 출입구에 설치된 리더에 의해 이를 인식하여 WMS 서버에 데이터를 전송하고 WMS 서버에서는 잘못된 제품 이동을 감지한다. 잘못된 제품의 이동을 감지한 WMS 서버는 관리자에게 이 사실을 알린다. 여기서 UPnP 이벤트가 발생하게 된다. 관리자는 자신의 컴퓨터가 아닌 다른 장소에 있더라도 UPnP 이벤트를 수신 받을 수 있다. 관리자의 사원증(태그)을 인식한 리더에 의해 관리자의 위치가 파악되고 관리자 위치에서 가장 가까운 컴퓨터, PDA 혹은 디스플레이할 수 있는 기기(예: TV)에 이벤트가 알려진다. 이것은 UPnP의 자동 서비스 발견 및 이벤트 알릴 기능으로 가능하다.

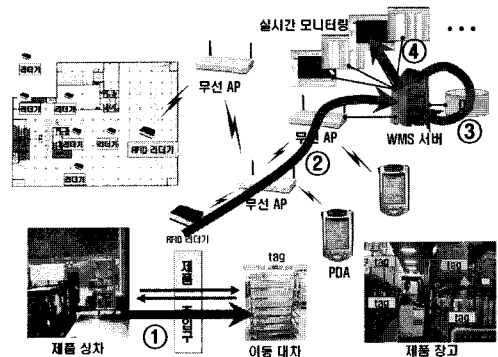


그림 2. 물류 창고관리 시스템의 이벤트 발견
Fig. 2 Event Discovery for Warehouse Management System

3.2 RFID 상황인식 시스템을 위한 UPnP 서비스 아키텍처

지능적인 상황인식 서비스를 제공하기 위하여 본 논문에서는 RFID 상황인식 시스템을 지원하는 UPnP 서비스 아키텍처를 설계 제안한다. 제안된 시스템에서는 상황인식을 위하여 RFID를 이용하고 서비스의 제공 및 자동 발견을 위해 UPnP 미들웨어를 사용한다.

그림 3은 제안된 RFID 상황인식 시스템을 위한 UPnP 서비스 아키텍처를 보여주고 있다. 제안된 시스템은 크게 WMS 서버와 RFID 리더/안테나, 웹 클라이언트(PC), 유무선 AP, 무선단말(PDA)로 구성된다. WMS 서버에는 RFID UPnP 디바이스와 RFID 데이터베이스(별도의 서버로 구성될 수 있음), 웹 프로그램 및 웹 서버로 구성된다. WMS 서버 내의 RFID UPnP 디바이스는 non-UPnP 디바이스인 RFID 리더의 데이터를 수신하여 RFID 데이터베이스에 저장하는 RFID 이벤트 관리자와 RFID 원본 데이터의 인식률을 향상시키는 클리닝 모듈 그리고 외부 UPnP 컨트롤포인트에게 RFID 서비스를 제공하는 RFID 서비스 모듈로 구성된다.

객체(제품)에 태그가 부착되어 있고 이 객체가 이동을 시작하면 각 출입구에 부착되어 있는 RFID 리더가 이를 인식한다. 인식된 데이터는 리더부터 WMS 서버의 RFID 어댑터가 수신하여 RFID 원본 데이터 수집기를 거쳐 RFID 원본 데이터베이스에 저장된다. 이 과정은 수시로 빈번히 일어나며 non-UPnP 디바이스인 RFID 리더의 이벤트를 데이터베이스에 저장함으로써 RFID UPnP 디바이스가 사용할 수 있는 기반이 조성된다. 이것은

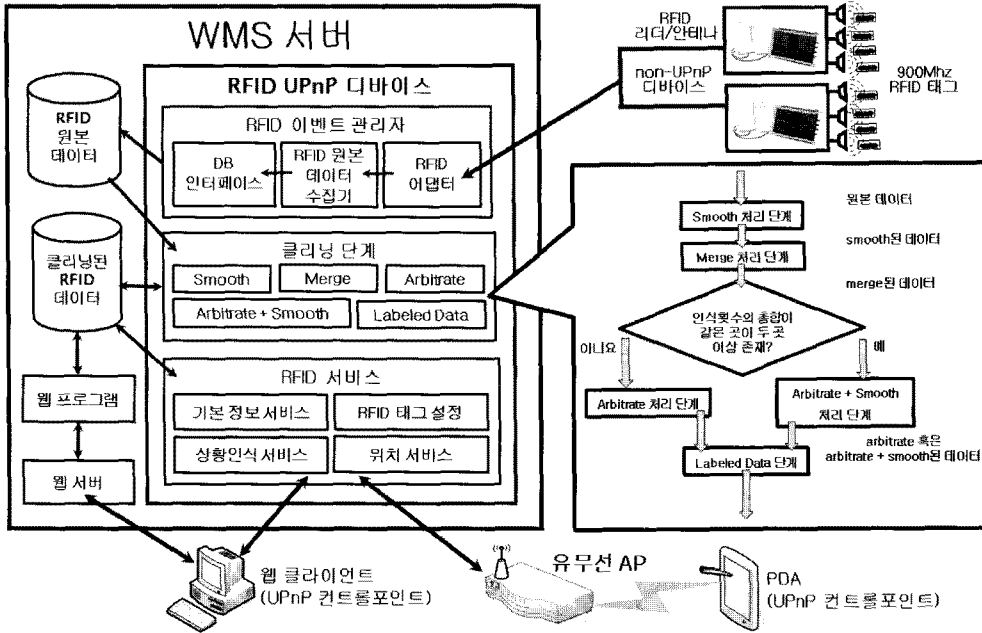


그림 3. RFID 상황인식을 위한 UPnP 서비스 시스템
Fig. 3 UPnP Service System for RFID Context-Aware

non-UPnP 디바이스의 UPnP 네트워크에 편입시키는 브리지 역할을 담당한다. 저장된 RFID 원본 데이터는 약 20% 이상의 부정확한 데이터를 포함하고 있기 때문에 이를 보정해 주는 것이 필요하다. RFID UPnP 디바이스의 클리닝 단계에서 이를 보정하게 되는데 총 5단계를 거치게 된다. 5단계는 그림 3의 오른쪽 순서도에 표현된 것과 같이 RFID 원본 데이터베이스에서 원본 데이터를 수신하여 smooth, merge, arbitrate 의 순서로 클리닝을 진행한다. arbitrate + smooth 처리 단계는 merge 단계에서 생성된 데이터 중에서 arbitrate 단계로 클리닝이 되지 않는 특수한 데이터를 보정하기 위한 처리 단계이다. 즉, 클리닝 단계를 거친 후에도 남아있는 false positive(태그가 실제로 존재하지 않지만 인식한 경우, 한 객체가 두 구역 이상의 안테나에서 중복 인식된 경우)를 보정해주는 역할을 한다. 각 단계의 클리닝 모듈은 SQL 기반으로 작성되어있어 RFID 데이터베이스와 연동하여 동작한다. RFID 데이터베이스는 RFID 원본 데이터와 클리닝 처리를 거친 보정된 RFID 데이터를 저장하고 관리한다.

그림 4, 5, 6, 7은 클리닝 모듈의 각 단계별 SQL문을 보여준다.

```
select sum(read_count), count(object_epc), object_epc from Raw_data
where timestamp <= (select max(timestamp) from Raw_data)
and timestamp >= (select max(timestamp)-2 from Raw_data)
and object_epc = 'u.1'
and reader_epc = '1.1' group by object_epc;
```

그림 4. smooth 처리를 위한 SQL문
Fig. 4 SQL for Smooth Cleaning

```
select reader_epc from Smooth_data
where timestamp = (select max(timestamp) from Smooth_data)
and object_epc = 'u.1';
```

그림 5. merge 처리를 위한 SQL문
Fig. 5 SQL for Merge Cleaning

```
select location_id from Merge_data
where timestamp = (select max(timestamp) from Merge_data)
and object_epc = 'u.1' group by location_id;
```

그림 6. arbitrate 과정에서 중복이 발생하는지 확인하는 SQL문
Fig. 6 SQL for Checking for the Duplication during Arbitration

```
select location_id, object_epc, count(object_epc) from Merge_data
where timestamp >=(select max(timestamp)-2 from Merge_data)
and object_epc = 'u.1' group by location_id;
```

그림 7. arbitrate+smooth 처리를 위한 SQL문
Fig. 7 SQL for Arbitrate+Smooth Cleaning

RFID UPnP 디바이스의 RFID 서비스 모듈은 클리닝 처리 후 보정된 데이터(클리닝된 RFID 데이터베이스)를 이용하여 사용자에게 서비스를 제공한다. 제공되는 서비스는 어떠한 설정도 없이 자동으로 발견되어 제공된다.

RFID UPnP 디바이스가 제공하는 주요 서비스는 표 1에서 나타낸 것과 같이 기본 정보 서비스, RFID 태그 설정 서비스, 상황인식 서비스, 위치 인식 서비스 등이다.

표 1. RFID 디바이스 서비스
Table. 1 RFID Device Services

서비스	내용
기본 정보 서비스	- UPnP 네트워크 내에서 관리하는 RFID 태그들의 기본적인 정보를 제공. <사용 예> - RFID 태그의 기본정보, 태그의 개수, 리더의 개수, 리더 배치, 지원하는 서비스 목록 등
RFID 태그 설정	- RFID 태그의 설정 값을 얻거나 새로운 값을 설정하는 기능을 수행 <사용 예> - 객체에 RFID 태그를 부착할 때 태그에 기본값 설정 - 이벤트 임계값 설정
상황 인식 서비스	- 상황인식 기능을 설정하거나 해제 기능을 수행 <사용 예> - “출하 명령이 없는 제품(객체)가 출하장으로 이동하면 이벤트를 발생하라.” 와 같이 정의하면 해당 동작이 발생시 관리자에게 이벤트 알림
위치 서비스	- 각 RFID 태그(객체)의 위치 정보를 제공 <사용 예> - 창고에 적재되어 있는 제품의 위치를 알려 줌으로써 자동으로 재고 조사가 이루어짐

웹 서버는 외부에 존재하는 PC로 상황인식 서비스 시스템을 모니터링하고 제어하기 위하여 해당 웹 서비스를 제공한다. PDA와 PC에 존재하는 UPnP 컨트롤포인트는 WMS 서버의 RFID UPnP 디바이스로부터 상황인식 데이터 서비스를 UPnP 프로토콜을 통하여 제공받는다. 사용자가 필요한 서비스를 자동으로 찾아주며, 찾은 서비스에 대한 제어 프로그램도 자동으로 UPnP 디바이

스에 의해 제공 받는다. 또한, 상황인식 서비스 이벤트가 발생하면 RFID UPnP 디바이스가 모든 컨트롤 포인트에게 알려준다(notification).

IV. 상황인식 서비스 구현 및 성능평가

4.1 상황인식 서비스 구현

본 논문에서는 자재 관리 시스템을 RFID와 UPnP 기반으로 구현하였다. 기자의 이동 상황을 실시간으로 확인할 수 있으며, 기자의 등록, 검색, 대여, 이벤트 알림(도난) 등의 서비스 등을 지원한다. 그림 8은 테스트를 위한 구현 구성도를 나타내고 있다.

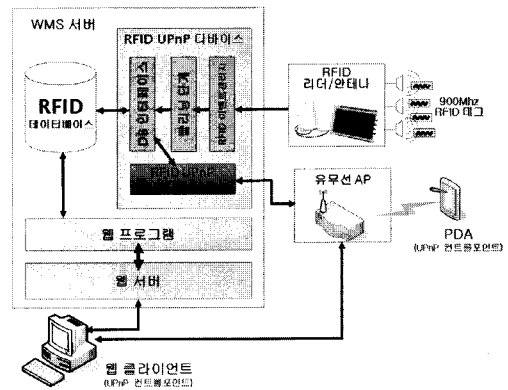


그림 8. 구현 구성도
Fig. 8 Block Diagram of Implementation

PDA 컨트롤포인트 구현 환경은 UPnP 서비스를 위하여 Intel사의 UPnP Tool을 이용하여 구현하고 개발 툴로는 Microsoft Visual Studio .NET 2003의 C#을 사용하였다 [10]. WMS 서버는 리눅스 운영체제에 MySQL을 이용하여 RFID 데이터 DB를 구축하고 gcc 개발 툴을 이용하여 클리닝 모듈을 개발하였다. RFID 관리 서버의 RFID 데이터 디바이스는 컨트롤포인트와 마찬가지로 Intel사의 UPnP Tool을 이용하여 구축하였다.

RFID 리더는 4개의 ALR 9611BC 서클러 안테나를 포함하는 ALIEN사의 ALR-9800이다[11]. 리더는 900Mhz 대의 주파수를 사용하며 태그는 EPC Class 1, Gen2 규약을 따르는 ALIEN사의 ALL-9460 “Omni-Squiggle”을 사용하였다. 복수의 안테나가 중첩되는 논리 공간에 대한

태그 인식을 할 때 전파 발산 시기가 겹쳐(신호 중첩) 신호 감쇄로 인한 성능 저하를 막기 위하여 안테나는 순차적으로 신호를 발생하도록 설정한다. 리더의 read rate는 10Hz로 설정하였으며, 수신된 태그 데이터를 TCP/IP를 통해 서버로 전송하도록 notify 모드를 설정하였다. 그의 설정은 리더 장치의 기본 값으로 하였다.

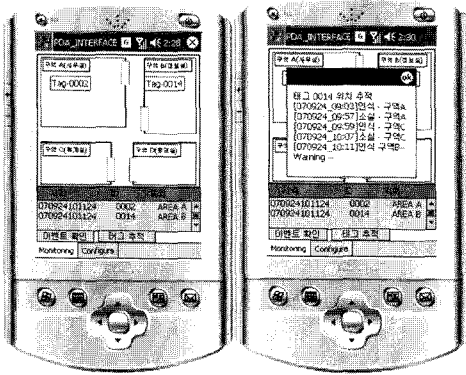


그림 9. 모니터링 위치 추적
Fig. 9 Monitoring for Location Tracking



그림 10. 물류 창고관리 웹 인터페이스
Fig. 10 Warehouse Web Interface

그림 9는 RFID 태그(제품, 직원)에 대한 관리자 모니터링을 PDA로 구현한 것이다. 각 사무실의 구역이 표시되며 각 사무실 내의 직원이 표시되고 있다. 한 직원의 일정 기간 동안의 이동 경로를 추적하기 위한 경로 모니터링도 가능하다.

그림 10은 구현된 자재 관리 웹 인터페이스의 일부로 물품, 등록번호, 자재의 위치 및 상태 등을 나타내고 있다. 구현된 웹 인터페이스 및 웹 프로그램은 APM (Apache, PHP, MySQL) 기반에서 구현되었다.

4.2 실험 및 성능평가

4.2.1 UPnP 서비스 실험 및 성능 평가

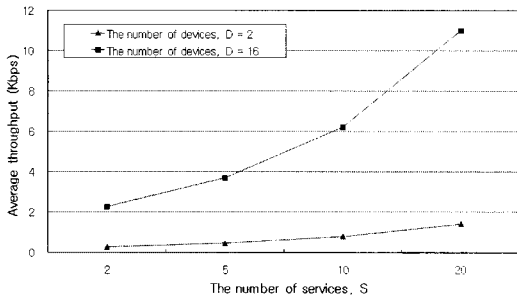
실험을 위하여 WiFi가 내장된 노트북 컴퓨터를 사용했다. 이 노트북 컴퓨터에는 RFID UPnP 디바이스를 자동으로 찾고 서비스를 받을 수 있는 컨트롤포인트를 내장했다. 16대의 PC를 이용하여 UPnP 디바이스 서비스를 구현했고 802.11b(대역폭 10Mbps)가 내장된 노트북 컴퓨터를 이용하여 UPnP 컨트롤포인트를 구현하였다.

RFID 서비스 환경에서 UPnP 디바이스가 주위에 많을 때에는 UPnP 광고(advertisement) 메시지가 다량으로 발생한다. 이 경우에는 네트워크 혼잡이 발생할 수 있다. 본 논문에서는 UPnP 서비스 자동발견 시 네트워크 혼잡 정도를 실험을 통하여 테스트 하였다.

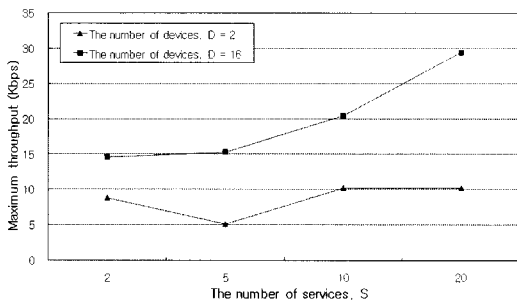
블루투스와 같이 무선 네트워크에서는 메시지가 전달되는 UPnP 광고가 많아지면 네트워크 혼잡문제가 발생할 수 있다(이더넷 대역폭 최소 10Mbps, 블루투스 대역폭 최대 1Mbps). 또한, UPnP 광고 메시지가 모바일 기기들의 배터리를 소모하거나 수면(sleep)에 들어가는 것을 방해하여 배터리 소모를 가속화시킬 우려도 발생한다. 이것은 UPnP 광고 메시지가 주기적으로 네트워크에 브로드캐스트 함으로써 발생한다.

UPnP 디바이스는 주기적으로 3+2d+k NOTIFY(ss에:alive) 메시지를 발송한다(d:임메디드 디바이스들의 수, k:디바이스 내의 임메디드 서비스의 수). 이것을 광고 세트라고 한다[12, 13]. UPnP 디바이스는 신뢰성 없는 UDP 멀티캐스트를 통하여 광고 세트를 전송하기 때문에 좀더 확실한 광고 메시지의 전달을 위해 중복하여 광고 세트를 발생한다. 따라서 같은 광고 메시지를 2번씩 광고 그룹으로 발송한다.

광고 메시지는 일시에 발생하는 것을 방지하기 위하여 cache-control 값의 1/3에서 2/3 사이에서 랜덤하게 발생시키도록 설계되었고 보통 cache-control 값은 30분(1800초)으로 설정되어 있다. 본 논문에서는 가혹 실험을 위하여 cache-control 값을 쿼고 값의 반인 15분(900초)으로 설정하여 테스트하였다.



(a) 평균 메시지 처리량



(b) 최대 메시지 처리량

그림 11. 서비스 수와 디바이스 수를 증가시키는데 따른 메시지 처리량 측정

Fig. 11 Measurement of the number of messages increasing with the number of services and devices

그림 11은 서비스 수와 디바이스 수를 증가시키면서 이에 따른 메시지 처리량을 측정하여 나타내주고 있다. UPnP 디바이스 수는 16대의 PC를 가지고 2대 16대 두 가지를 측정하였고 하나의 UPnP 디바이스에는 2, 5, 10, 20개의 서비스들이 존재하여 서비스 수의 변화에 따른 메시지 처리량 변화를 측정하였다.

그림 11의 (a)에서와 같이 서비스 수가 증가되면 평균 메시지 처리량도 선형적으로 증가되는 것을 알 수 있다. 최대 평균 트래픽은 11Kbps의 평균 대역폭을 보이는 디바이스당 서비스 수가 20개인 디바이스 16개인 경우에서 발생했다. 이것은 WiFi 대역폭 11Mbps나 블루투스 대역폭 1Mbps에도 미치지 못한다. 따라서 UPnP의 광고 메시지에 대한 평균 트래픽은 네트워크 대역폭의 영향을 받지 않는다고 볼 수 있다.

그림 11의 (b)는 (a)와 같이 서비스 수를 증가하면서 최대 메시지 처리량을 측정하였다. 최소 5Kbps(디바이스 수 D=2, 서비스 수 S=5)와 최대 29Kbps(디바이스 수 D=16, 서비스 수 S=20) 사이의 결과 값이 측정되었다. 최

소값이 D=2, S=2가 아니고 D=2, S=5인 것은 cache-control 값 안에서 랜덤하게 서비스에 대한 광고 메시지가 발생하는데 기인한다. cache-control 값의 1/2에서 2/3 사이에서 랜덤하게 메시지가 발생하므로 일부 구간에서 메시지가 우연히 겹치는 경우가 발생하는데 이때 메시지 처리량도 함께 증가하므로 실험과 같이 데이터가 측정될 수 있다. 즉, 최대 메시지 처리량(트래픽)은 선형적으로 증가하는 그래프를 그리지만 반드시 디바이스 수와 서비스 수가 적다고 해서 낮은 처리량을 가진다고는 보장되지 않는다. 그러나, 대체적으로 선형적인 증가 그래프를 그리는 것은 확실하다. UPnP의 광고 메시지에 대한 최대 트래픽도 WiFi나 블루투스의 대역폭 안의 범주에 있으므로 큰 영향을 받는다고는 볼 수 없다.

4.2.2 RFID 클리닝 실험 및 성능 평가

SQL 기반의 클리닝 모듈 성능을 평가하기 위하여 그림 12와 같이 논리적인 영역을 A, B로 나누고 각 구간 사이의 통로 양 옆에는 각각 2개의 안테나를 배치한다. 안테나는 정확한 실험을 위하여 높이 및 각도를 동일하게 고정한다. 실험에 사용하는 리더의 인식범위는 일반적으로 반지름이 6m인 부채꼴 형태이다. 안테나 배치 시 안테나 간의 거리를 3.5m로 설정하여 실제 상황보다 데이터 중복인식이 많이 발생하도록 배치하였다. 논리적 영역 A, B에는 각각 태그를 지닌 사용자가 존재하며, 사용자 A는 1분 간격으로 반대편 논리 영역으로 이동한다. 또한, 사용자 B는 2분 간격으로 반대편 논리 영역으로 이동한다[14].

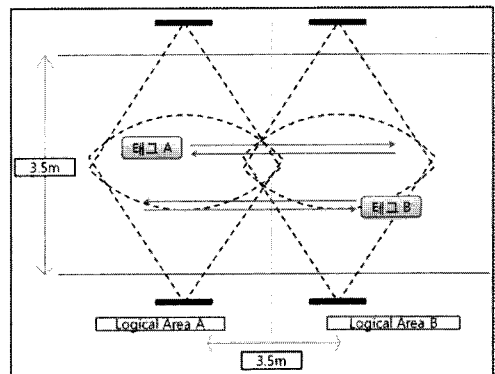


그림 12. 실험 구성도

Fig. 12 Block Diagram for Experimentation

실험에는 하나의 리더에 4개의 안테나로 진행하였다. 안테나의 전파 발산 시기를 겹치지 않게 하기 위해 순차적으로 신호를 발생하도록 설정하였다. 리더의 read rate는 10Hz로 설정하고, 수신된 데이터를 TCP/IP로 전송할 수 있도록 notify 모드를 설정하였다. 이 외의 리더 설정은 기본 값으로 하였다. 데이터베이스는 MySQL을 사용한다.

실험결과와 인식률을 측정 한 방법은 인식된 구간(단위: 초)을 전체 인식 구간으로 나눈 후 백분율로 나타내었다. 이 때, 두 개의 안테나 중 하나의 안테나에만 인식된 경우라도 인식이 된 것으로 간주한다. 이렇게 계산한 인식률은 그림 13의 원본 인식률이다. 원본 인식률이 일반적인 환경에서의 인식률(60~70%)보다 높게 나타난 것은 실험환경 자체의 이상적인 조건 때문이라고 추론할 수 있다.

클리닝 처리는 smooth, merge, arbitrate 순으로 하며 arbitrate 단계까지 거친 후의 인식률은 그림 13에서와 같이 향상된 것을 알 수 있다. 3단계의 클리닝을 거친 인식률은 상당히 향상되었으나, 논리적으로 불가능한 중복 인식이 아직도 남아있음을 확인할 있었다. 이러한 문제를 해결하기 위해서 arbitrate+smooth 단계를 한번 더 수행한다. 이 처리 단계를 수행한 후 데이터를 살펴보면 arbitrate 단계 이후에도 나타났던 중복인식이 대부분 사라진 것을 확인할 수 있었다.

중복을 의도적으로 발생시킨 것을 제외하고 실험의 환경이나 여러 조건들은 상당히 이상적인 상태에서 실시되었다. 실제 인식률 향상은 상당하였고, 태그 인식률은 태그 A에서 12.3%, 태그 B에서 25%가 향상되었다.

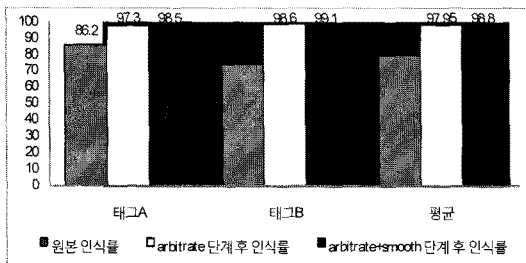


그림 13. 실험 인식률

Fig. 13 Experimentation Recognition Rate

V. 결 론

상황인식 서비스 시스템은 주변 상황을 인식하고 상황 정보를 수집 처리하여 이질의 네트워크를 경유해서 각 사용자에게 서비스를 제공한다. 상황인식 서비스를 원활히 제공하기 위해서는 몇 가지 문제점을 해결해야 한다. 상황인식 서비스를 편리하게 이용하기 위하여 서비스 자동 발견 및 제어 동작에 대한 서비스 정의가 용이해야 할 필요가 있다. 또한, 상황정보를 수집하는 센서는 자원이 부족한 경우가 대부분이므로 센서에서 직접 서비스를 제공할 수는 없다. 그리고 상황인식 애플리케이션은 센서에서 수집된 데이터를 신뢰할 수 있어야 한다.

본 논문에서는 이와 같은 문제들을 해결하고 상황인식 서비스를 제공할 수 있도록 RFID와 UPnP 기반 상황인식 서비스 시스템을 설계 제안하고 구현하였다. 제안된 시스템은 상황인식을 위해 RFID 기술을 사용하여 객체를 인식하였다. 그러나, RFID 태그의 값을 수집한 원본 데이터는 실험을 통하여 확인해 본 결과 약 80.15%만 인식하고 약 19.85%는 잘못된 객체 인식 결과를 보였다. 제안된 시스템에서는 잘 못된 객체 인식값(RFID 원본 데이터)을 클리닝 기법을 이용하여 알고리즘을 세우고 값을 보정하였다. smooth, merge, arbitrate, arbitrate+smooth 순으로 클리닝한 최종 결과는 평균 98.8%의 인식률을 보였다. 18.65%의 인식률 향상이 있었다.

자원이 부족한 RFID 태그의 데이터 서비스와 상황인식 서비스의 자동 발견 및 제어 동작에 대한 정의의 지원을 위해 본 논문에서는 UPnP 미들웨어를 적용하여 시스템을 제안한다. UPnP를 적용하여 시스템을 구축하면 상황인식 서비스의 개발이 용이하며 설정 없이도 RFID 디바이스 서비스를 제공받을 수 있다. 설계 구현된 UPnP 서비스 디바이스는 RFID 상황인식 및 클리닝과 이에 맞는 서비스를 사용자의 개입 없이 자동으로 제공한다. UPnP 디바이스의 광고에 대하여 실험을 통해 성능을 확인하였다. 실험은 현재의 네트워크 대역폭 내에서 원활히 통신할 수 있는 수준으로 평가되었다.

참고문헌

[1] RFID Journal, <http://www.rfidjournal.com/>

[2] H. Song, D. Kim, K. Lee, J. Sung, "UPnP-based sensor network management architecture," Second International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2005), Osaka, JAPAN, April 13 - 15, 2005.

[3] 김동균, 윤희성, 이상정, "UPnP 기반 RFID 서비스 발전", 제28회 한국정보처리학회 추계학술발표대회 논문집 하권 p.1092-1095, 2007년 11월

[4] UPnP Forum, <http://upnp.org>

[5] ORACLE Sensor-based Service, <http://www.oracle.com/technologies/rfid/index.html>

[6] Ahmad Rahmati, Lin Zhong Dep. of ECE, Rice University. Reliability Techniques for RFID-Based Object Tracking Applications

[7] E. Rahm et al.. Data cleaning: Problems and current approaches. IEEE Data Eng. Bull., 23(4):3-13,2000

[8] Anish Das Sarma, Shawn R. Jeffery, Michael J. Franklin, Jennifer Widom. Estimating Data Stream Quality for Object-Detection Applications. Technical Report No. UCB/EECS-2005-23

[9] S. R. Jeffery, M. Garofalakis, and M. J. Franklin. Adaptive Cleaning for RFID Data Streams. Under submission to SIGMOD, 2006.

[10] Intel Software for UPnP Technology, <http://www.intel.com/technology/upnp>

[11] Alien ALR-9800 915MHZ RFID Reader. <http://www.alientechnology.com/readers/alr9800.php>

[12] Y.Liong, Y.Ye, "Effect of UPnP advertisements on User Experience and Power Consumption", Consumer Communications and Networking Conference, 2005. CCNC. 2005 Second IEEE 3-6 Jan. 2005 Page(s):91-97

[13] Dong-Kyun Kim and Sang-Jeong Lee, "UPnP-Based Telematics Service Discovery for Local Hot-Spots", Proceedings of 2007 International Conference on Multimedia and Ubiquitous Engineering, April 2007.

[14] 윤희성, 김동균, 이상정, "SQL 기반 RFID 클리닝 모듈 설계", 제28회 한국정보처리학회 추계학술발표대회 논문집 하권 p.1088- 1091, 2007년 11월

저자소개



김동균(Dong-Kyun Kim)

1997년 금오공과대학교 기계공학과 (공학사)

1996-2000년 동양그룹 (주)동양매직 가전연구소 연구원

2002년 순천향대학교 컴퓨터공학과(공학사)

2004년 순천향대학교 전산학과(공학석사)

2008년 순천향대학교 전산학과(공학박사)

※관심분야: IP 네트워크, 신재생에너지, 임베디드 시스템



전병찬(Byung-Chan Jeon)

1992년 한밭대학교 전자계산학과 (공학사)

1994년 수원대학교 전산계산학과 (이학석사)

2001년 순천향대학교 전산학과(공학박사)

2002년-현재 청운대학교 컴퓨터학과 전임강사

※관심분야: 컴퓨터 구조, 마이크로프로세서 응용, 모바일 네트워크



이상정(Sang-Jeong Lee)

1983년 한양대학교 전자공학과 (공학사)

1985년 한양대학교 전자공학과 (공학석사)

1988년 한양대학교 전자공학과(공학박사)

1988년-현재 순천향대학교 컴퓨터학부 교수

1999-2000년 미국 University of Minnesota 방문교수

※관심분야: 네트워크 응용, 컴퓨터 구조