효율적인 비동기 전송을 지원하기 위한 RTLS 미들웨어의 확장

(API Extension of RTLS Middleware for Efficient Asynchronous Transmission)

박 재 관* 홍 봉 희** 이 승 철***
(Jae Kwan Park) (Bong Hee Hong) (Seung Chul Lee)

요 약 최근, 많은 기업에서 실시간 자산 관리를 위해 RTLS 시스템을 구축하고 있다. RFID와 달리, RTLS 태그는 이동 과정과 한정되지 않고 임의의 위치에서 지속적으로, 자동적으로 인식된다. 그러나, RTLS 미들웨어의 표준 API는 2가지 한계점이 있다. 미들웨어가 애플리케이션으로 불필요한 데이터를 포함하는 대용량의 데이터를 전달해야 한다는 것과 미들웨어에서 애플리케이션으로 질의 결과를 전달하는 방식에서 동기 방식만을 지원한다는 문제가 그것이다. 이 논문에서는 이러한 문제를 해결하기 위해, 다양한 질의에 대해 애플리케이션으로 전달되는 데이터 량을 줄이기 위한 질의 타입별 정제 조건을 명세할 수 있는 SessionSpec을 정의하고 실시간 이벤트 처리를 위한 비동기 방식 지원 방법을 제안한다. 또한, 이러한 방법을 적용한 RTLS 미들웨어를 설계하고 구현하여 그 결과를 확인하였다.

키워드 : RTLS, 미들웨어, 질의처리, 스트리밍 데이터

Abstract Recently many global enterprises build RTLS system for their environments. RTLS is used to detect object at real time. Unlike RFID, RTLS tags are read automatically and continuously, independent of the process that moves the tags. The proposed functionality of standard API has two problems. When middleware provides data to application, it sends a huge amount of data that may be useless. When only an application requests for data, the middleware replies result data in synchronous mode. This paper proposes a method to reduce an amount of data transferring from middleware to application and an addition communication mode to support real-time event processing in the middleware. Also, we designed and implemented an RTLS middleware applying the proposed methods.

Keywords : RTLS, Middleware, Query Processing, Streaming Data

1. 서 론

RTLS(Real Time Location System) 시스템은 실내 또는 실외의 제한된 공간 내에서의 RFID[1]기술 또는 무선 랜 기술을 활용하여 움직이는 사람이나 사물에 RTLS 태그를 부착하여 RTLS 판독기(Reader)의 인식을 통해 실시간으로 위치를 파악하는 시스템이다. 이러한 RTLS 시스템을 최근 많은 기업들이 RTLS 환경을 도입하거나 도입을 적극 검토 중에 있다. RTLS 시스템의 도입으로 RTLS 태그를 부착한 사물의 현재 위치를 실시간으로 확인 할 수 있게 함으로써 사물의 관리에 있어 지연 요소를 최소화하고 의사결정의 신속성을 높인다[2].

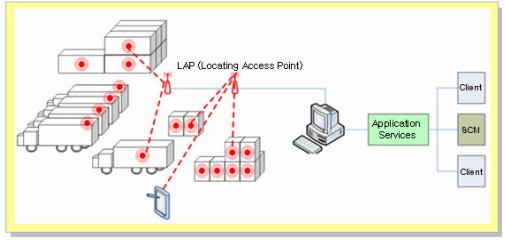

그림 1. RTLS 시스템

† 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음(This work was supported for two years by Pusan National University Research Grant)

* LG전자 전자기술원 선임연구원, jkpark183@lge.com
** 부산대학교 컴퓨터공학과 교수, bhhong@pusan.ac.kr(교신저자)
*** LG전자 PDP사업부, sclee@lge.com

논문접수 : 2008.11.24 수정일 : 1차 2008.12.31 / 2차 2009.02.23 심사완료 : 2009.03.11

이러한 RTLS 시스템은 다양한 애플리케이션에서 사용되고 있다. 예를 들어 항만 관리 시스템[3]에서 컨테이너 및 차량의 실시간 위치 추적을 통해 이동 대상 컨테이너의 위치 및 차량의 위치를 파악하여 항만 운영의 효율적인 관리를 할 수 있다. 또한, 고가의 의료 장비가 외부로 유출 될 경우의 막대한 자산 손실을 가져 올 수 있는 의료 관리[4] 에서 실시간으로 위치를 확인함으로써 자산 손실을 막을 수 있으며, 보안업체에서 제한 구역에 비 승인자가 통과 시, 이를 즉시 확인 가능하기 때문에 통제할 수 있게 한다.

이렇게 RTLS 시스템으로 관리함으로써 업무를 자동화하여 사람이 직접 관리할 수 없는 부분까지 제어할 수 있도록 하여 효율을 높일 수 있다. 하지만 이런 업무 자동화를 위해 기업의 모든 애플리케이션들이 각각 위치 데이터를 수집하고 애플리케이션에서 필요한 데이터를 정제하여 사용한다면 애플리케이션 마다 중복되는 모듈들이 탑재되어야 하는 자원의 낭비가 초래된다. 이를 방지하기 위해서는 미들웨어(Middleware)의 사용으로 RTLS 시스템 상에 존재하는 모든 데이터의 흐름을 미들웨어가 관장하도록 하여 애플리케이션의 질의에 대하여 수집 및 정제하여 미들웨어로부터 데이터를 제공받는 구조가 필요하다. 이로 인해 RTLS 시스템을 이용하는 미들웨어에 공통적으로 필요한 기능을 ANSI(American National Standard Institute) 371[5]에서는 필요한 Application Programming Interface(API)를 제정을 하였으며, ISO(International Standard Organization) 24730-1[6]에서 이 API를 국제 표준으로 제정하여 사용하고 있다. 이API는 크게 2가지 기능으로 데이터를 제공하고 있다. 첫 번째는 Stateless Query로써, 이것은 현재 RTLS 태그 별 현재 위치 및 상태 정보를 즉시 제공하는 API이다. 두 번째는 Session-Based Query로써 애플리케이션이 미들웨어로 질의를 세션으로 등록을 하고 애플리케이션이 결과를 요청 할 때 RTLS 데이터를 제공하는 API이다. 이 두 가지 API는 모두 SOAP 표준 인터넷 프로토콜을 사용하여 질의를 요청하고, RTLS 데이터를 즉시 응답하는 동기 방식을 기술하고 있다.

이러한 표준 RTLS 미들웨어 API는 오직 2가지의 기능을 개략적으로 제시하여 단지 현재 위치에 대한 질의만을 제공하기 때문에 미들웨어로써의 기능이 부족하다. 따라서 대부분의 데이터 처리를 각 애플리케이션에서 처리해야하며, 그로 인해 다양한 애플리케이션들의 질의 처리 시에 애플리케이션의 처리 비용 증가, 미들웨어에서 애플리케이션으로의 전송 데이터 량의 증가 문제가 발생한다. 또한 표준API는 동기 전송 방식만을 사용함으로써 즉시 전송이 필요한 데이터라 하더라도 애플리케이션의 요청이 있을 때 까지 전송이 지연되는 문제가 있다.

이 논문에서는 표준을 준수하고, 표준 API의 한계점을 보완하여 RTLS 미들웨어를 설계 및 구현 하였다. 먼저, 애플리케이션의 질의 타입에 따라 데이터에 대한 정제 및 수집에 대해 기술할 수 있는 방법(SessionSpec, SessionReportSpec)을 정의하여 질의 타입에 따라 애플리케이션이 원하는 데이터만을 제공하도록 하였다. 또한 기존 API를 확장하여 전송 방식에 대한 명세를 할 수 있도록 하였으며, Subscribe라는 API의 추가를 통해 즉시 보고가 필요한 데이터 및 애플리케이션의 요청이 없어도 주기적으로 전송하는 비 동기 전송 방식을 지원하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하며, 3장에서는 문제를 정의한다. 4장에서는 질의 분석과 문제 정의를 기반으로 RTLS 미들웨어 확장 방법을 기술한다. 5장에서는 RTLS 미들웨어의 구현 및 실험에 대하여 기술하며, 마지막으로 6장에서 결론을 기술한다.

2. RTLS 미들웨어 관련 국제 표준

표준의 API(ISO 24730-1)는 역할에 따라 Stateless Query와 Session-Based Query로 분류되며 미들웨어와 애플리케이션간의 메세지 교환은 SOAP-RPC를 이용해서 이루어지게 된다. 아래의 그림 2는 미들웨어와 애플리케이션 간의 SOAP-RPC 메시지를 간략하게 나타낸 것이다.

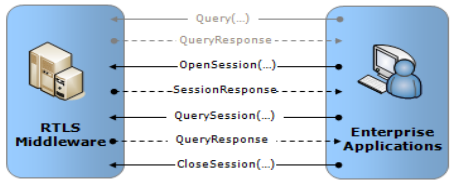

그림 2. 표준 Application Programming Interface

Stateless Query는 RTLS 미들웨어에서 질의의 조건에 맞는 모든 RTLS 데이터를 애플리케이션에게 전달하며 하나의 RTLS 태그 당 최대 하나의 RTLS 데이터가 전달이 된다. 표준 RTLS API는 애플리케이션이 Stateless Query를 요청할 때 마다 QueryResponse로 결과를 반환하며, 반환되는 결과는 이전의 Stateless Query의 영향을 받지 않는 API이다.

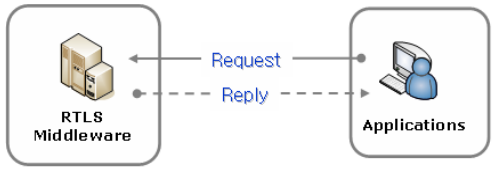

그림 3. Stateless Query의 동작 방식

Session-Based Query는 애플이케이션이 OpenSession을 통하여 Session을 등록하면, Session에 주어진 질의의 조건들에 따라 RTLS데이터들을 수집하며, 애플리케이션이 QuerySession을 통하여 질의를 하면, 이전의 Query-Session 이 후부터 새로운 QuerySession 명령이 들어올 때까지 수집된 RTLS 데이터들을 애플리케이션에게 전달하게 된다. RTLS 미들웨어가 OpenSession에 의해서 주

어진 Session 에 대해 RTLS 데이터를 수집하는 기간은 최종적으로 Session이 종료되는 시점으로 CloseSession 이 호출될 때 까지이다.

그림 4. Session Based Query의 동작 방식

기존의 AeroScout의 Mobileview[4], WhereNet의 Visibility[7]과 같은 RTLS 시스템에서는 애플리케이션이 미들웨어에서 처리해야 할 기능을 사용하여 환경에 맞는 데이터를 정제하고 수집하고 있다. 또한 대다수의 벤더들이 미들웨어 표준 API를 사용하지 않고 자체 API를 사용하고 있는 상황이다.

3. 문제정의

3.1 지속적인 대량의 데이터 처리 문제

GeoSensor 데이터스트림 시스템에서 S를 스트림이라 하고 R을 릴레이션이라고 할 때, 공간 조인의 대상이 될 수 있는 요소들은 다음과 같이 정의할 수 있다.

RTLS 미들웨어는 애플리케이션이 일반적으로 빈번하게 사용하는 질의(위치, 알람, 수량 질의 등)에 대하여 데이터를 제공해야 하지만 RTLS 표준은 위치 질의만을 정의함으로써 애플리케이션이 미들웨어로부터 받은 RTLS 데이터를 정제해야 하는 단점이 있으며 많은 양의 데이터 전송으로 인해 미들웨어와 애플리케이션 간의 네트워크 트래픽을 증가시키는 문제점이 있다.

예를 들어 애플리케이션이 “제한 구역에 들어오는 이동체의 위치를 보고하라”라는 질의를 등록한다고 가정하자. 그림 5와 같이, 최초 제한 구역 내에 노트북1~4까지의 위치를 보고 받았고, 이후에 방문자가 제한 구역으로 들어가서 보고 받은 데이터는 노트북 1~4와 방문자의 위치이다. 이 경우 애플리케이션은 이전 정보와 중복된 노트북 1~4의 위치까지 보고받게 되어 이를 정제를 해야한다. 따라서 RTLS 미들웨어는 지속적인 대량의 데이터 스트림에 대하여 애플리케이션의 질의에 적합한 결과만을 전달하기 위한 구조가 필요하다.

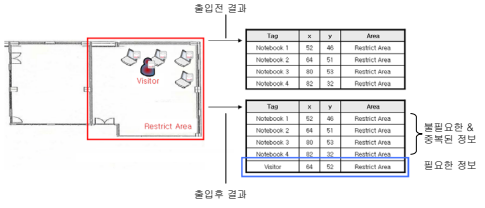

그림 5. RTLS 표준 질의의 문제점

3.2 RTLS표준의 전송 방식의 문제

또한 RTLS 표준은 애플리케이션에게 데이터를 전송하기 위해 동기 방식(On demand)을 제공하고 있다. 이러한 동기 방식은 애플리케이션이 미들웨어에게 데이터를 요청해야만 RTLS 데이터를 응답하는 방식만으로써 애플리케이션에게 즉시 제공되어야 할 데이터에 대해서 지연이 발생하게 된다. 그림 6과 같이 외부인에 의해 노트북이 도난을 당하는 상황을 예를 들면, 창고 안에 있는 노트북이 누군가에 의해서 창고 밖으로 운반되어 도난이 의심되는 상황이다. 노트북에 부착된 RTLS 태그는 신호를 보내어 창고 밖에 위치되었음을 알리고, 미들웨어는 애플리케이션에 즉시 보고를 해야 하는 상황이다. 하지만 RTLS 표준은 애플리케이션이 요청이 있어야지만 응답을 하는 동기 방식만을 제공함으로써 실제 데이터가 발생한 시간보다 늦은 시간에 보고하게 된다. 그로 인해 실시간 물품 관리에 있어 문제점을 야기시킬 수 있다.

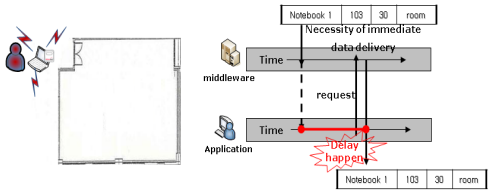

그림 6. 전송 방식 문제의 예

이러한 전송 지연에 있어 간단하게 해결하는 방법으로 애플리케이션에서 계속해서 미들웨어로 데이터를 요청하는 방식이 있을 수 있다. 하지만 언제 일어날지 모르는 상황에 대하여 계속해서 데이터를 요청하는 방식은 애플리케이션의 질의 요청에 대한 비용이 낭비되어 비효율적이다. 따라서 RTLS 미들웨어는 애플리케이션에게 효율적으로 데이터를 제공할 수 있도록 애플리케이션이 요청하는 시점에 데이터를 전송하는 동기 방식 및 요청이 없이 즉시 보고가 필요한 데이터에 대하여 비 동기 전송방식을 지원하여야 한다.

4. 효율적인 비동기 전송을 지원하는 RTLS 미들웨어 API의 확장

이 논문에서 제시하는 RTLS 미들웨어는 애플리케이션의 질의를 분석하여 질의에 알맞은 데이터만 제공하는 구조로 설계하기 위하여, 질의 타입의 확장과 이러한 질의 타입의 정제 조건을 명세할 수 있는 SessionSpec을 정의하였으며, 질의 타입의 수집 조건을 명세할 수 있는 SessionReportSpec을 정의하였다. 또한, 데이터의 보고 지연 문제는 비 동기 전송 방식에 대한 조건을 등록 할 수 있는 확장 API인 OpenSession 과 등록한 조건을 활성화/비활성화 시키는 Subscribe/UnSubscribe를 제공하여 이를 해결하는 미들웨어 구조를 설계하였다.

4.1 질의 타입의 확장 및 질의 관리자

4.1.1 질의 타입 분석

기존 RTLS 벤더의 솔루션을 분석한 결과, RTLS 미들웨어는 각각의 애플리케이션에서 환경에 의존적인 질의를 제외하면 공통적으로 필요로 하는 질의 타입은 다음과 같이 현재 위치에 대한 질의, 특정 상황에 대한 질의, 수량에 대한 질의로 요약된다. RTLS 미들웨어는 3가지 질의 타입에 대하여 효율적으로 질의 결과를 제공할 수 있는 구조로 설계 되어야 하며 이에 대하여 애플리케이션의 사용할 수 있는 API를 제공해야 한다.

현재 위치(location) 질의는 RTLS 미들웨어가 보고 해야 하는 가장 기본적인 질의로써 현재 위치에 대한 질의이다. 예를 들어 물류 관리에서 “창고에 있는 물품들을 보고하라.”라는 질의, 의료 관리에서는 “환자의 위치를 보고하라.” 질의, 항만관리에서는 “컨테이너의 위치를 보고하라.” 와 같은 질의가 이에 해당된다. 알람(Alert) 질의는 알람 질의의 특정 장소에 출입하는 태그가 발생하면 즉시 보고하는 질의이다. 예를 들어 물류 관리에서 “창고 안으로 들어오는 태그에 대하여 보고하라.” 라는 질의가 이에 해당된다. 수량(Quantity) 질의는 특정 지역내의 RTLS 태그를 부착한 이동체의 수량을 보고하는 질의로써 특정 지역 내에 위치한 이동체의 수량을 보고하는 질의이다. 예를 들어 물류 관리에서 “창고내의 물품에 대하여 수량을 보고하라.”라는 질의가 이에 해당된다.

4.1.2 질의 타입의 확장

표준 API에서는 질의 타입으로 현재 위치 처리 질의만 가능한 질의 타입인 RTLS_Query를 제공하고 있다. RTLS_Query는 현재 위치에 대한 질의 조건으로 크게 3가지 조건을 정의 할 수 있으며 그 의미는 표 1과 같다. 그리고 그림 7은 현재 질의 처리를 위한 <Location-Type>이라는 위치에 대한 조건, <FilterType>이라는 태그에 대한 조건을 명시하고 </Fields> 라는 질의의 결과 형태 결정 조건을 제시하고 있는 XML예제이다.

표 1. 질의 조건

조 건	질 의	설 명
어디서	</LocationType>	RTLS 태그가 읽힌 위치 조건
무엇을	</FilterType>	어떤 태그인지 태그의 조건
어떻게	</Fields>	질의의 결과 형태 결정 조건

표준 API는 현재 위치에 대한 질의만을 제공하고 있어서 애플리케이션에서 필요로하는 상황을 인지하고 처리할 수 없으므로 알람(Enter, Leave, Idle), 수량(Quantity)에 대한 질의는 제공하지 못하고 있다. 따라서 이러한 질의를 제공하기 위한 질의 타입을 ENTER_Query, LEAVE_Query, Idle_Query, Quantity_Query로 확장하였다. 질의 타입에 따른 수집 방법을 각각 명시할 수 있도록 하여 애플리케이션의 질의에 대하여 적합한 데이터만을 전달하여 데이터의 중복을 제거하였다. 표 2는 질의 타입에 대한 설명이다.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:Queryxmlns:ns3="http://schemas.xmlsoap.org/soap/encoding/"xmlns:ns2="http://www.autoid.org/iso24730-1/RTLS-schema">
    <QueryType>RTLS_Query</QueryType>
<FilterType>
        <LocationType>
            <ZoneID>zone3</ZoneID>
</LocationType>
        <TagID>=100</TagID>
    </FilterType>
<Fields>TagID BatteryLow X Y</Fields>
    <SortBy>
        <Field>TagID</Field>
        <Order>asc</Order>
    </SortBy>
</ns2:Query>
```

그림 7. RTLS_Query의 XML예제

표 2. QueryName의 확장

Query Type	설 명
RTLS_Query	이동체의 현재 위치에 대한 질의
ENTER_Query	이동체가 특정 지역에 들어가면 보고하는 질의
LEAVE_Query	이동체가 특정 지역에서 나가면 즉시 보고하는 질의
IDLE_Query	이동체가 일정시간 움직임이 없으면 즉시 보고 하는 질의
QUANTITY_Query	특정 지역내의 수량에 대한 질의

4.1.3 질의 관리자

질의 타입의 확장 정의로 인해 이동체의 현재 위치에 대한 질의 처리 구조에서 확장 정의한 질의 타입에 대해서도 처리하는 구조가 필요하다. 확장한 질의 타입 또한 이동체의 위치를 기반으로 하고 있어 먼저 이동체의 위치에 대한 필터링을 거치고, 각각의 질의 타입의 조건에 맞는 정제 과정을 거쳐야 한다. 이에 이동체의 위치에 대한 질의 처리를 위해 연속 질의 색인 기법인 VCR-Based Query Index[8]를 사용하여 삼각 측량을 통해 위치를 결정하는 위치 결정 엔진(Location Engine)으로부터 획득되는 끊임없이 발생하는 위치 데이터 스트림 데이터[9]을 효율적으로 처리하였다. RTLS 태그 식별자를 위한1차원 질의 색인과 위치 정보(x, y)를 위한 2차원 질의 색인을 사용하였으며, 각 질의 타입의 처리를 위해 질의 관리자를 각각의 질의 타입에 따라 필터링하는 구조로 설계하였다.

질의가 등록되는 과정은 다음과 같다. 애플리케이션은 원하는 질의 타입으로 질의를 등록하면, 태그 및 위치에 대한 질의 조건은 연속 질의 색인에 등록되고, 질의 타입의 질의 조건은 질의 관리자에 등록되게 된다. 이 후 위치 데이터가 입력되면 연속 질의 색인에서 태그 식별자 및 위치(x, y)에 대한 정제를 하게 되고, 다음으로 질의 관리자에서 연속 질의 색인에서 처리된 결과인 위치 데이터로부터 각 질의 타입에 적합한 필터링 단계를 거치게 된다. 그림 8은 각각의 질의 타입에 따라서 처리되는 과정을 나타낸다. RTLS_Query는 연속 질의 색인의 결과인 위치 데이터와 동일하여 별도의 필터링 작업이 불필요 하지만, ENTER_Query와 LEAVE_Query는 RTLS 태그에 대한 1차원 연속 질의 색인 처리 결과와 위치에 대한 2차원 연속 질의 색인에서 전달되는 결과로 RTLS 태그 별 가장 최근의 태그 위치와 비교하여 처리하게 된다.

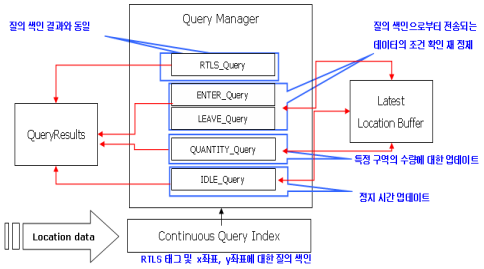

그림 8. 질의 매니저의 동작 방식

QUANTITY_Query와 IDLE_Query는 가장 최근 위치를 저장하는 위치 버퍼(Location Buffer)를 사용하여 위치 정보에 대한 2차원 연속 질의 색인의 처리 결과가 전달되면 특정 위치 내의 이동체의 수량을 업데이트하여 유지하고 Idle_Query는 위치 버퍼의 가장 최근 위치와 비교하여 변동 유무를 확인 후, RTLS 태그 정지 시간을 업데이트하여 처리한다.

4.2 SessionSpec과 SessionReportSpec 정의

SessionSpec과 SessionReportSpec은 질의 타입인 RTLS_Query뿐만 아니라 ENTER_Query, LEAVE_Query, IDLE_Query, QUANTITY_Query에 대한 필터링 조건과 전송 타입을 명세 한다. 그림 9는 애플리케이션이 SessionSpec을 등록하고, 결과로 SessionReportSpec에 명시된 수집조건의 정보로 SessionReports를 전송 받는 구조를 나타낸 것이다.

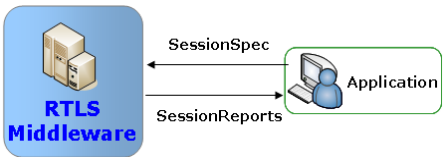

그림 9. SessionSpec과 SessionReport

SessionSpec의 구성은 표 3과 같다. TagID는 RTLS 태그에 대한 정제, x, y는 좌표값에 대한 정제를 명세하며 timeInterval은 Idle_Query타입을 등록할 때 정지시간을 명세한다. SessionSpec 내의 SessionReportSpec은 RTLS 수집 정보를 포함한다.

표 4. SessionReportSpec의 구성

SessionReportSepc	IncludeField : List Control : SubscriptionControl
SubscriptionControl	Duration : time AlertTrigger : Boolean ReportSet : ReportSetType
ReportSetType	CURRENT DELETION ADDITION

ReportSetType의 CURRENT는 그림 10와 같이 특정 지역 내에서의 발생하는 데이터를 전송하는 명세를 담당하며, DELETION은 그림 11와 같이 특정 지역에 빠져나간 데이터에 대해서만 명세를 하도록 하며, ADDITION은 그림 12과 같이 특정 지역으로 들어가는 데이터에 대해서만 명세를 담당하고 있다.

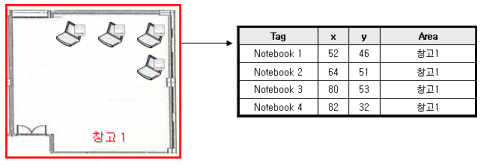

그림 10. SessionSpec과 SessionReport

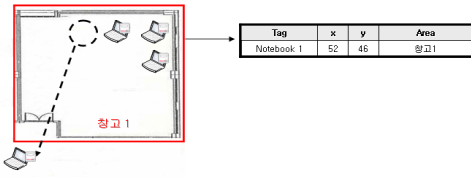

그림 11. SessionSpec과 SessionReport

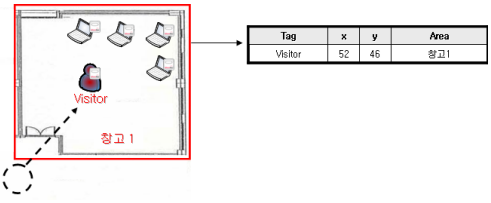

그림 12. SessionSpec과 SessionReport

4.3 확장 API

기존 미들웨어 표준 API에서는 오직 두 가지 기능인 Stateless Query와 Session-Based Query로만 현재 위치

에 대한 질의를 처리 하였다. 이 두 가지의 기능 중 Session-Based Query는 애플리케이션이 질의를 등록하고 그 질의에 대한 RTLS 데이터를 정제 및 수집하여, 요청이 있을 시 RTLS 미들웨어에서 수집한 데이터를 전송하는 동기 방식만을 사용하였다. 그런데, 요청이 있을 때만 데이터를 제공하는 동기 전송 방식의 경우에는 데이터 보고에 있어 지연이 발생한다. 이러한 문제점을 해결하기 위하여 기존의 OpenSession() API를 확장하고, 새로운 API인 Subscribe를 정의하여 비 동기 전송 방식을 지원함으로써 데이터 보고에 있어 지연 문제를 해결하고 애플리케이션의 요청 없이 데이터를 전송하는 구조로 설계하였다.

표 5. Extended API

Extended API	설 명
OpenSession(SessionSpec) : SessionID	SessionSpec 등록
Subscribe(SessionID, notificationURI)	SessionSpec 활성화
UnSubscribe(SessionID)	SessionSpec 비활성화

표 5는 확장한 API를 설명하고 있다. 첫 번째로 확장한 API인OpenSession은 인자로써 SessionSpec을 가질 수 있게 확장하여 필터링 조건에 대한 명세 외에 전송방식에 대한 명세(Duration, AlertTrigger)도 포함할 수 있도록 하였다. 두 번째로 확장정의한 API인 SubScribe()는 등록한 SessionSpec을 활성화하고 비 동기 전송 방식으로 제공되는 데이터를 수신하기 위해 수신할 애플리케이션의 notificaitonURI인자를 가지고 있다.

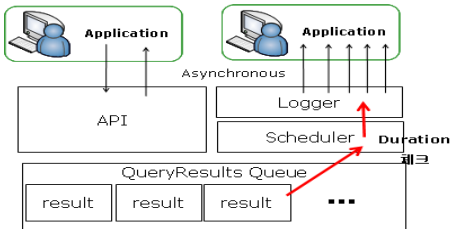

그림 13. Duration에 의한 전송 구조

그림 13은 Duration을 사용하여 애플리케이션에게 데이터를 제공하는 구조를 나타낸다. 애플리케이션의 질의 결과 데이터들은 질의 색인을 거쳐 QueryResults에 저장되고, Scheduler는 각 애플리케이션이 명세한 Duration을 체크하여 Duration의 조건을 만족하는 데이터를 Logger에 전송하게 되며, Logger는 애플리케이션이notification-URI에 명세한 프로토콜(Protocol) 에 따라 각 애플리케이션에게 비 동기 전송 방식으로 제공하는 구조를 가지고 있다.

그림 14은 Alert에 의한 전송 구조이다. AlertTrigger가 설정된 애플리케이션의 질의들은 Alert Listener에게 등록되며, Alert Listener는 즉시 보고가 필요한 데이터에

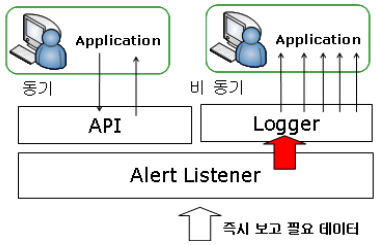

그림 14. Alert에 의한 전송 구조

대하여 Logger를 통해 비 동기 전송 방식으로 제공하는 구조를 가지고 있다.

그림 15는 확장 API OpenSession과 Subscribe를 사용하여 질의를 등록하고 활성화 하여 비 동기 전송방식으로 SessionReport를 전송 받는 흐름을 나타낸 것이다. 애플리케이션에서 확장된 OpensSession으로 SessionSpec을 등록하면 SessionSpec에 대한 sessionID를 RTLS 미들웨어는 반환하고, 이 반환된 sessionID와 데이터를 받을 자신의 URI를 이용하여 확장 Subscribe를 통해 등록된 SessionSpec을 활성화 시킨다. 이 후 애플리케이션은 SessionSpec에 대한 즉시 보고가 필요한 데이터 및 일정 기간 동안 수집한 데이터를 의미하는 SessionReports 데이터 타입으로 비 동기 전송 된다.

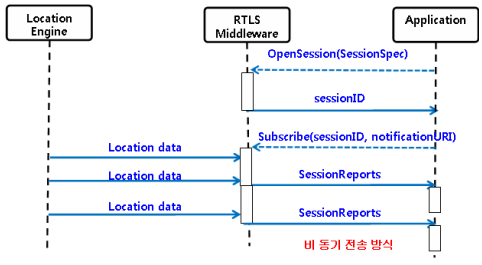

그림 15. Extend API를 사용한 비 동기 방식의 흐름

5. RTLS 미들웨어 구현 및 실험

5.1 RTLS 미들웨어의 구조

그림 16은 RTLS 미들웨어의 구조를 블록 다이어그램으로 나타낸 것이다. 획득 서비스, 프로세스, 질의 서비스 계층으로 세분화되며, 획득 서비스 계층에서는 위치 결정 엔진(Location Engine)으로부터 전송되는 데이터를 획득을 담당하며, 프로세스 계층에서는 애플리케이션에 의미 있는 데이터만을 제공하기 위한 정제 및 수집을 담당하며, 질의 서비스 계층에서는 API를 제공하며, 애플리케이션의 질의에 대한 결과를 전송하는 역할을 한다.

5.2 RTLS 미들웨어의 구현 및 실험

RTLS미들웨어는 다양한 환경에 적용 가능해야 한다. 이는 미들웨어와 운영체제나 하드웨어에 비 종속적이어야 한다는 의미가 된다. 이를 위해 논문의 RTLS 미들웨어

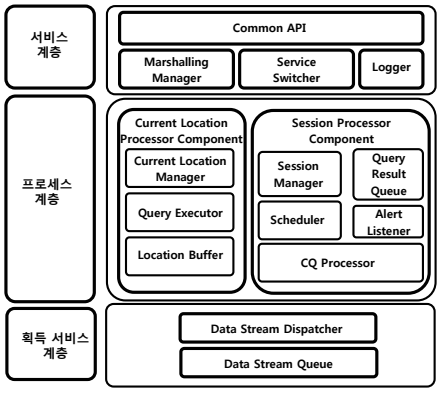

그림 16. RTLS 미들웨어 블록 다이어그램

구현에 Java언어를 사용하여 플랫폼에 독립적으로 동작할 수 있도록 하였다. Pentium IV 2.66Ghz, 1GB RAM, Windows XP OS에서 구현하였다.

5.2.1 RTLS 미들웨어 및 모니터링 툴

RTLS 미들웨어의 API는 표준에 따라 SOAP-RPC로 구현되었다. 이를 위해 미들웨어는 별도의 웹 컨테이너가 필요하기 때문에 Jetty Web 6.0 웹 컨테이너[10] 기반으로 구현되었다. 또한, RTLS 미들웨어는 애플리케이션이 직접 사용하는 것이 아닌 응용 소프트웨어를 통해 사용되므로 사용자가 직접 미들웨어의 내부 동작을 눈으로 쉽게 확인하는 것은 불가능하다. 이에 사용자가 미들웨어의 정상 동작을 쉽게 확인할 수 이도록 그림 17과 같이 별도의 RTLS 미들웨어 모니터를 구현하였다. RTLS 미들웨어 모니터를 통해 위치 결정 엔진으로부터 획득되는 스트림 데이터를 질의 색인을 거쳐 정제를 하고 질의 매니저를 통해 각 질의 타입 별로 다시 정제되어 마지막 최종으로 결과 데이터를 수집하는 데이터의 흐름을 관찰할 수 있도록 하였다.

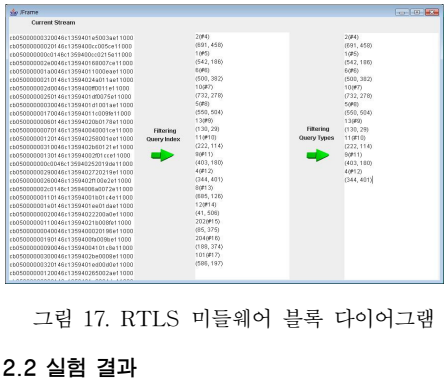

그림 17. RTLS 미들웨어 블록 다이어그램

5.2.2 실험 결과

그림 18는 표준에 의한 정제 및 수집 방법과 SessionSpec에 의한 정제 및 수집 방법에 대하여 애플리케이션에게 전송되는 데이터 량을 비교한 것이다. 전송 데이터 량은 다양한 질의 (RTLS_Query, EENTER_Query, LEAVE_Query, IDLE_Query, QUANTITY_Query)에 대하여 질의 조건을 랜덤으로 설정(보고 주기는 3초)하여, 기존 표준 API를 이용할 경우와 SessionSepc을 이용할 경우에 대하여, 미들웨어에서 응용으로 전달되는 위치 데이터의 평균 개수이다. 전체 태그 량을 1000개로 제한하였으며, 한 태그당 보고 주기를 3초로 하였을 때 SessionSpec에 의한 정제 및 수집 방법이 애플리케이션에게 전송되는 데이터 량이 줄어 들었음을 알 수 있다.

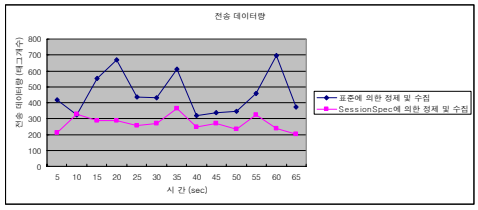

그림 18. 애플리케이션에 전송되는 데이터 량

그림 19는 전송방식에 따른 CPU 점유율에 대하여 나타낸 그림이다. 이 결과는 그림 18의 질의들을 수행하는 상황에서, 응용이 미들웨어로 등록한 질의에 대한 결과를 얻기 위해서 0.5초, 1초 마다 결과를 확인하는 주기적 connection 형태의 기존 방법과 확장된 미들웨어의 비동기 처리 방법을 비교한 것으로 미들웨어에 요구되는 부하를 측정한 결과이다. 즉시 보고가 필요한 데이터를 보고받기 위해 기존 API의 QuerySession을 사용하여 1초 및 0.5초로 동기 전송을 사용하여 질의를 수행하는 것은 CPU 사용에 있어 비효율적으로 나타났으며, 확장 API인 Subscribe를 사용하여 비 동기 전송을 사용하는 것이 CPU 점유율에 있어 안정적으로 나타났다. 제안된 미들웨어는 위치 데이터가' 인식될 때 확장된 component에 의해 기존 미들웨어보다 CPU 연산이 증가 하지만 제안된 미들웨어에 의해 불필요한 데이터가 필터링되므로 관리되는 데이터 량이 줄어들게되고 비동기 방식 전송을 지원하게 됨으로써 감소되는 부하가 훨씬 더 크기 때문에 그림 19와 같은 결과를 보였습니다.

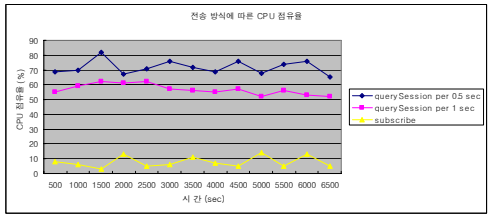

그림 19. 전송 방식에 따른 CPU 점유율

6. 결 론

최근 다양한 환경에서 RTLS 시스템을 도입하고 있다. 이러한 RTLS 시스템에서 미들웨어는 필수적이지만 표준

API의 기능만으로는 지속적으로 전송되는 대량의 데이터에 대해 효율적으로 질의를 처리하기 어렵다. 즉, 애플리케이션에게 불필요한 데이터를 포함하여 제공함으로써 2차 정제를 해야 하며, 미들웨어와 애플리케이션간에 네트워크 부하가 증가하게 된다. 또한 애플리케이션의 요청이 있을 때만 데이터를 제공하는 동기 전송 방식만을 기술하여 즉시 보고를 해야 하는 데이터에 대해 지연을 발생시킨다.

이 논문에서는 애플리케이션이 미들웨어로 등록하는 일반 질의인 현재 위치에 대한 질의, 특정 상황 발생 질의, 수량에 대한 질의 등에 따라 데이터에 대한 필터링 조건을 기술할 수 있는 SessionSpec 과 SessionReportSpec을 정의하여 질의 타입에 따라 정제 및 수집 방법을 다르게 설계하여 질의 타입에 맞는 데이터만을 제공하도록 하였으며, 또한 기존 OpenSession API를 확장하여 SessionSpec과 SessionReportSpec을 등록 할 수 있게 하여 전송 방식에 대한 명세를 할 수 있도록 하였고, Subscribe라는 API를 추가하여 데이터를 보고 받을 애플리케이션의 정보를 명세할 수 있도록 함으로써 즉시 보고가 필요한 데이터를 효율적으로 전송할 수 있도록 하고, 애플리케이션의 요청이 없어도 주기적으로 전송하는 비 동기 전송 방식을 지원하였다.

최근 항만물류, 병원, 물류창고 등 다양한 환경에서 RTLS 시스템을 도입하고 있습니다. 이러한 환경에 이 논문에서 제안한 확장된 미들웨어가 효율적으로 적용될 수 있으며, 향후 RFID 및 센서 네트워크 시스템과 융합된 컨버전스 시스템으로 발전할 것으로 예상되므로 이를 위한 연구가 필요합니다.

참 고 문 헌

[1] 정태수, 김영일, 이용준, RFID미들웨어 기술 동향 및 응용," 전자통신동향분석, 제20권 제3호, 2005, pp. 81-91.

[2] 백한진, "RTE 구현을 위한 전략, " 삼성SDS IT컨설팅팀, 2004.

[3] 박두진, 최영복, "RTLS를 활용한 유비쿼터스 항만운영시스템 구축방안," 한국콘텐츠학회논문지 제6권 제12호, 2006, pp. 128-135.

[4] AeroScout, http://www.aeroscout.com/, 2007.

[5] BSR INCITS 371.3 ANSI, Part3 : Application Programming Interface(API), 2003.

[6] ISO/IEC 24730-1, Part1 : Application Program Interface(API), 2005.

[7] Wherenet, http://www.wherenet.com/, 2007.

[8] Kun-Lung Wu et al., "Processing Continual Range Queries over Moving Object Using VCR-Based Query Indexes," in Proc. mobiQuitous, 2004, pp.226-235.

[9] 이미영, 김명준, "이벤트 기반 서비스 기술 동향," 전자통신동향분석, 제 21권 제 5호, 2006, pp. 61-68.

[10] Jetty6 - Jetty Web Server, http://www.mortbay.org/, 2007.

[11] Gustafsson.f, Gunnarsson.f, "Positioning using time-difference of arrival measurements," in Proc. IEEE Conf. on ICASSP, 2003, pp. VI-553-6.

[12] Kaemarungsi.K, Krishnarmurthy.P, "Properties of indoor received signal strength for WLAN location fingerprinting," in Proc. IEEE Conf. on MOBIQUITOUS, 2004, pp. 14-23.

[13] Donald Carney et al., "Monitoring Stream - A New Class of Data Management Applications," VLDB, 2002, pp. 215-226.

[14] Brenda M. Michelson, "Event-Driven Architecture Overview," Patricia Seybold Group, 2006.

[15] EPCglobal, EPC Information Services (EPCIS) Version 1.0 Specification, 2007.

[16] EPCglobal, The Application Level Events (ALE) Specification Version 1.0, 2005.

[17] Ekahau, http://www.ekahau.com/, 2007.

[18] 김정준, 김관교, 김동오, 이기영, 한기준, "공간 DSMS 기반 RTLS의 설계 및 구현," 한국공간정보시스템학회논문지, 제10권 제4호, 2008, pp. 44-55.

[19] 강홍구, 박치민, 홍동숙, 한기준, "공간 센서 데이타의 효율적인 실시간 처리를 위한 공간 DSMS의 개발, " 한국공간정보시스템학회논문지, 제9권 제1호, 2007, pp. 45-57.

박 재 관

1999년 부산대학교 컴퓨터공학과 졸업(학사)
2001년 부산대학교 대학원 컴퓨터공학과 졸업(공학석사)
2008년 부산대학교 대학원 컴퓨터공학과 (공학박사)
2008년~현재, LG전자기술원 선임연구원
관심분야 : RTLS 시스템, 유비쿼터스 시스템, 모바일 임베디드 DBMS

홍 봉 희

1982년 서울대학교 전자계산기공학과 졸업(학사)
1984년 서울대학교 대학원 전자계산기공학과 졸업(공학석사)
1988년 서울대학교 대학원 전자계산기공학과 졸업(공학박사)
1987년~현재, 부산대학교 컴퓨터공학과 교수
관심분야 : 공간 데이터베이스, RFID 시스템, RFID 데이터베이스

이 승 철

2006년 신라대학교 컴퓨터정보과학부 졸업(학사)
2008년 부산대학교 대학원 컴퓨터공학과 졸업(공학석사)
2008년~현재, LG전자 연구원
관심분야 : RTLS, RTLS 미들웨어, 공간 데이터베이스