

RTLS를 위한 위치 데이터 스트림 처리기 개발

Development of Location Data Stream Processor for RTLS+

이승철*, 홍봉희, 김기홍, 박재관

Seung-chul Lee*, Bong-hee Hong, Gi-hong Kim, Jae-kwan Park

부산대학교 컴퓨터공학과

{gatsby*, bhong, buglist, jkpack}@pusan.ac.kr

요약

최근 항만 물류 및 자산 관리 분야에서 실시간 위치 정보를 처리하는 RTLS(Real Time Locating System)시스템이 도입되고 있다. 이러한 시스템에서 RTLS 서버는 태그를 부착한 이동 객체들의 위치 데이터 스트림을 일정 시간 동안 수집하여 애플리케이션으로 전달한다. 이 때 위치 정보는 전파 굴절 현상으로 인해 오차가 발생하며, 이동 객체에 부착된 태그는 수 초 마다 위치 정보를 보고하기 때문에 시스템의 과부하를 초래하게 된다.

본 논문에서는 표준과의 호환성을 고려하고, 요구사항을 반영한 위치 데이터 스트림 처리기를 설계 및 개발하였다. RTLS 시스템의 전파 굴절 현상으로 야기되는 비정상적인 위치 오차를 보정하기 위해 맵 매칭 기법을 도입하여 위치 데이터 스트림의 신뢰성을 제공하며, 위치 변화가 없는 객체의 위치 데이터 스트림을 빠르게 정제하는 정지 상태 제거 필터를 개발하여 질의 처리 시 부하를 줄인다. 또한 각 애플리케이션의 질의 결과에 무의미한 위치 정보를 배제하는 중요 위치 수집기를 개발하여 시스템 성능을 향상시킨다.

1. 서론

RTLS(Real time Location System)는 실내 또는 실외의 제한된 공간 내에서 RFID 기술 및 무선 랜 기술을 활용하여 이동 객체의 위치를 실시간으로 인식하여 관리하기 위한 시스템이다. 현재 헬스케어 애플리케이션에서 환자, 직원과 의료기의 위치를 파악하여 환자의 관리, 의료기의 분실을 최소화하여 의료 환경 개선의 목적으로 구축되어 있으며, 놀이동산의 미아 찾기 애플리케이션에 사용되어 놀이동산에 발생하는 미아를 찾기에 소비되는 인력 감소의 목적으로도 사용되고 있다. 또한 항만운영시스템에서 컨테이너 및 컨테이너 차량에 대하여 RTLS 태그를 부착하여 위치를 추적함으로써 항만운영 시스템

에 차별화되고 효율적인 시스템을 구축하고 있다. 이렇게 다양한 애플리케이션에서 제품의 추적과 자산의 모니터링을 위해 RTLS 시스템이 채택되고 있으며, 이러한 RTLS 시스템의 구축을 위해서는 태그들의 위치 데이터 스트림을 관리하는 RTLS 서버는 필수적이다. RTLS 서버는 애플리케이션 환경이 요구하는 정보만을 제공할 수 있어야 하며 RTLS 태그로부터 올라오는 위치 데이터 스트림을 수집, 저장 및 정제하여 애플리케이션에게 제공해야 한다.

본 논문에서는 RTLS 시스템의 API 표준을 담당하는 ANSI INCITS 371-3의 바탕으로 표준을 준수하고, 데이터 스트림을 실시간으로 처리하기 위한 방법, 비정상적

+ “이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임”(지방연구중심대학육성사업/차세대IT기술연구사업단)

인 위치 데이터 스트림을 처리하기 위한 위치 보정방법과 RTLS 표준 API의 확장으로 어플리케이션이 보고 받은 위치 정보에 정제 단계를 거치지 않도록 하기 위한 RTLS 서버 구조를 설계하고 개발하였다.

제안하는 RTLS 서버는 데이터 스트림을 효과적으로 처리하기 위해서 연속 질의 색인을 도입하여 질의처리 시간을 감소시켜 실시간 처리를 지원한다. 또한 정지 상태인 객체로부터 발생하는 데이터 스트림에 대한 정제 기법을 개발하여 RTLS 서버의 부하를 줄인다. RTLS 시스템의 전파 굴절 현상으로 야기되는 비정상적인 위치 정보를 보정하기 위하여 이동 객체가 존재 할 수 없는 영역에 위치하는 경우, 맵 매칭 기법을 도입하여 위치를 보정시켜 위치 데이터 스트림의 신뢰성을 제공하였으며, RTLS 서버의 공통 API를 확장하여 각 어플리케이션에서 질의에 무의미한 위치 정보를 배제하는 중요 위치 수집기를 개발하여 시스템 성능을 향상시켰다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 소개하며, 3장에서 기본적인 요구 사항을 분석하며, 4장에서 대상 환경 및 문제를 정의한다. 5장에서는 RTLS 서버의 구조 설계에 대하여 기술하고, 6장에서 결론 및 향후 연구에 대하여 기술한다.

2. 관련 연구

이 장에서는 RTLS 서버를 설계하기 위해 필요한 RTLS 서버의 API 표준 명세와 스트림 데이터를 다루는 질의 색인 관련 연구, 위치 정보를 보정하기 위한 맵 매칭 기법을 기술한다.

2.1 RTLS 서버 API 표준

RTLS 시스템의 전체 구조는 [그림 1]과 같으며 하드웨어 부분의 RTLS Reader와 RTLS 태그로 구성되고, 소프트웨어 부분으로 RTLS 서버로 구성된다. RTLS 서버의 API 표준 ANSI INCITS 371-3[1]에 제정되어 있다.

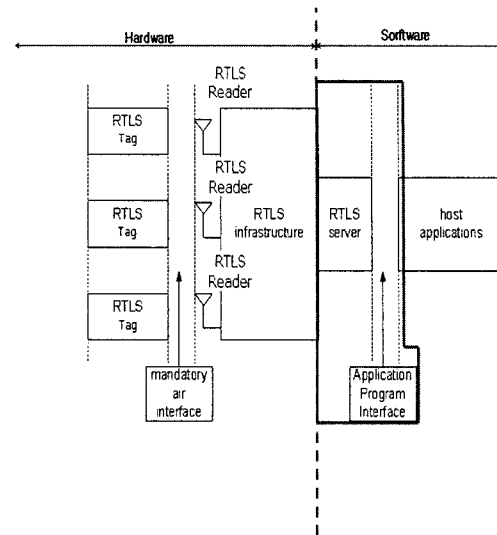


그림 1 RTLS 시스템 전체 구성 요소
RTLS 서버의 API는 [표 1]에서처럼 Stateless Query와 Session-based Query로 나뉘어진다.

(a) Stateless Query : 어플리케이션의 질의에 RTLS 태그 별 가장 최근 값만으로 한 번의 결과 값을 반환한다.

(b) Session-based Query : 어플리케이션과 RTLS 서버간의 연결 상태를 지속적으로 유지하여 어플리케이션의 질의 조건에 맞는 결과 값을 임시적으로 서버에 저장하여 어플리케이션이 원하는 시점에 하나 이상의 값들로 질의를 수행하여 결과 값을 반환한다.

2.2 질의 색인 방법론

끊임없이 발생하는 데이터 스트림을 효율적으로 처리하기 위해서는 데이터 스트림에 적합한 질의 처리 기법을 적용시켜야 한다. 기존의 DBMS에서 사용하는 데이터 처리 방식과는 달리 데이터를 색인하는 대신 질의를 색인하는 연속 질의 색인 기법이 사용된다. 이러한 연속 질의 색인 기법으로는 셀을 기반으로 한 CQI(Cell-based Query Indexing Scheme)[3]와 가상의 사각형 구조를 사용한 Virtual Construct Rectangle-Index[2]가 있다.

CQI-Index는 셀 기반의 그리드 방식을 적

(a) Stateless Query
QueryResponse Query(queryName, filters, fields, sortBy)
(b) Session-based Query
SessionResponse OpenSession(queryName, filters, fields)
QueryResponse QuerySession(SessionID)
void CloseSession(SessionID)

표 1 RTLS 서버 API

용한 기법으로 이동 객체에 대한 영역 질의를 색인화 한다. 전체 영역을 일정 크기의 셀로 분할하고 각 셀은 2개의 ID 리스트인 Full 리스트와 Part 리스트를 가진다. Full 리스트에 속하는 검색 시 빠른 성능을 보장하지만 Part List로부터 얻어진 결과 질의의 셋들이 실제로 이동 객체를 포함하는지 비교하기 위한 정제 단계가 필요하여 검색 성능이 저하되는 단점이 있다.

VCR-Index는 이동 객체에 대한 영역 질의를 색인한다. 영역 질의 조건을 2차원의 사각형으로 표현되는데 이것을 미리 정의된 VCR(Virtual Construct Rectangle)이라는 가상의 사각형 구조를 이용하여 분할하고 해당 구조의 ID 리스트에 질의의 ID를 삽입하는 기법이다. 검색방법은 질의 점을 포함하는 모든 분할 구조의 ID 리스트를 탐색하는 것이다.

2.3 맵 매칭(Map Matching) 기법

맵 매칭[4]이란 LBS(Location based Service)에서 주로 사용되는 기법으로 실제로 이동 객체가 이동한 만큼의 거리와 방향이 지도상에 나타난 지점과 일치하지 않을 때 지도위에서 이동 객체의 위치를 수정해 주는 기법을 말한다. 보고되는 위치 정보가 존재할 수 없는 영역일 때 과거의 위치, 속력, 방향에 의한 정보로 이동 객체가 위치 할 수 있는 곳으로 맵칭 시키게 된다.

3. RTLS 서버의 요구사항 분석

RTLS 서버는 다음과 같은 기본적인 요구 사항을 만족하는 구조로 설계가 되어야 한다.

▶ 표준 공통 API 지원하는 구조 :

표준에서 제시하는 공통 API는 어플리케이션에서 한 번의 질의를 통해 태그별 현재 위치를 검색하는 Stateless Query와 어플리케이션의 질의를 RTLS 서버에 유지하는 Session-based Query가 존재한다. 이에 RTLS 서버는 이와 같은 API를 지원하는 구조를 가져야 한다.

▶ 위치 데이터 스트림 관리 : RTLS 서버는 끊임없이 발생하는 위치 데이터 스트림을 실시간으로 처리하고 관리해야 한다. 어플리케이션의 질의 조건에 만족하는 위치 데이터 스트림을 효율적으로 처리하기 위해서는 고성능 질의 처리 모듈이 요구된다.

4. 대상 환경 및 문제 정의

4.1 대상 환경

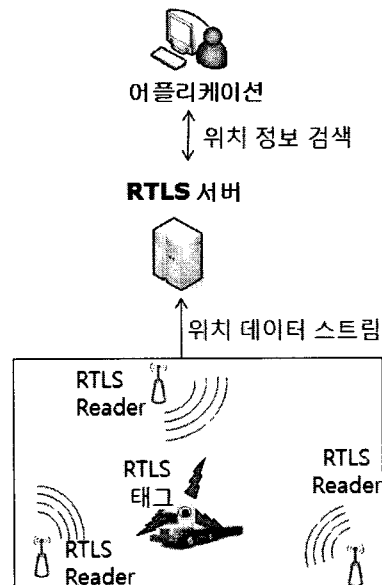


그림 2 RTLS 구성 요소

RTLS 서버는 기본적으로 [그림 2]와 같

이 이동 객체에 부착된 RTLS 태그로부터 위치 데이터 스트림을 수집하고 어플리케이션으로부터 원하는 질의 조건을 입력받아 위치 데이터 스트림을 제공할 수 있는 정보 형태로 가공하여 요청한 어플리케이션에 제공한다.

4.2 문제 정의

RTLS 서버는 위치 데이터 스트림을 수집하고 어플리케이션으로부터 질의를 받아 그 질의에 해당하는 위치 정보를 제공하는 역할을 한다. 이에 RTLS 서버는 다음과 같은 문제들이 나타날 수 있다.

▶ 위치 오차

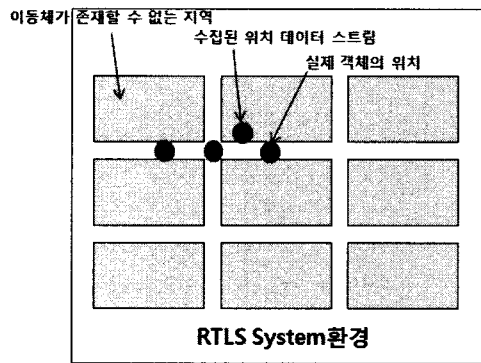


그림 3 위치 오차 문제

RTLS 시스템의 전파적인 특성으로 비정상적인 위치 데이터 스트림 수집하는 경우가 발생한다. 이로 인해 RTLS 서버의 신뢰성을 감소시킨다.

▶ 대량의 데이터 스트림 발생

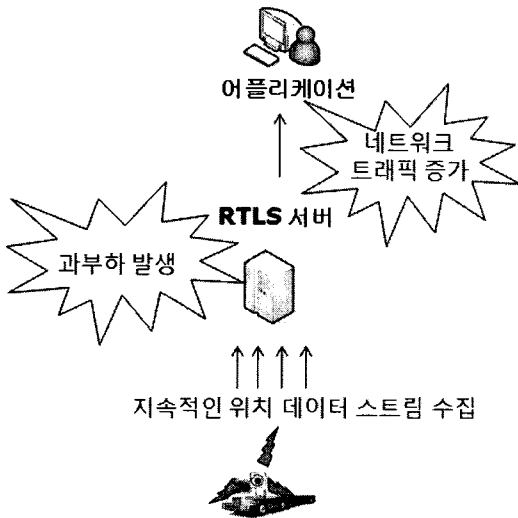


그림 4 대량의 데이터 스트림 문제
RTLS 시스템은 수백 개에서 수만 개의

이동 객체의 위치 정보를 관리한다. 이에 각 이동 객체들은 수초 단위로 위치 데이터 스트림을 지속적으로 RTLS 서버로 보고하게 되어, 위치 데이터 스트림 처리 시 과부하를 초래하게 되며 어플리케이션에 위치 정보를 전송 시 네트워크 트래픽을 증가 시키게 된다. 또한 어플리케이션의 질의는 이동 객체가 일정거리 이상으로 이동한 데이터에 대해서만 알기를 원하는 형태가 다수이므로 [그림 5]과 같이 어플리케이션은 제공받은 대량의 위치 정보로부터 무의미한 데이터를 정제하여 중요 위치만 알아내야 한다.

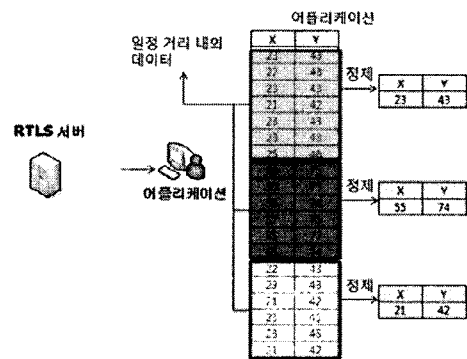


그림 5 어플리케이션의 일정 거리 내의 데이터 정제

5. RTLS 서버의 설계

이 장에서 3장의 기본적인 요구 사항과 4장의 문제 정의를 통하여 분석된 사항들을 고려하여 RTLS 서버를 설계한다.

5.1 질의 처리

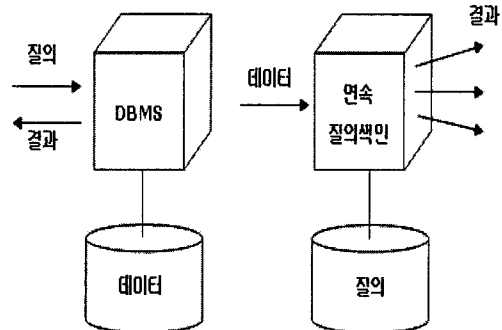


그림 6 기존 DBMS 처리 방식과 질의 색인 처리 방식

대량의 데이터 스트림에 대하여 질의 처리를 하기 위하여 질의 색인 구조를 도입하였다. RTLS 서버에 Session-based Qu

ery에 질의가 들어오면 질의 조건을 데이터화 시키고, 실시간으로 수집되는 위치 데이터 스트림을 질의화시켜 처리함으로써 효과적으로 질의 처리를 한다.

5.2 위치 보정

[그림 7]의 (a)는 수집되는 위치 데이터 스트림을 나타낸다. 이렇게 비정상적인 데이터 스트림으로 위치 오차가 발생하게 되는데 이를 처리하기 위하여 맵 매칭 기법을 사용하여 위치 될 수 없는 영역에 위치 데이터 스트림이 발생했을 때 [그림 7]의 (b)처럼 이를 과거 위치, 속력, 방향에 의한 정보를 유지하는 데이터를 통해 위치를 보정한다.

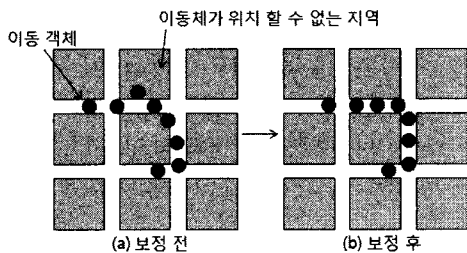


그림 7 맵 매칭 기법

5.3 대량의 위치 데이터 스트림 처리

RTLS 서버로 보고되는 위치 데이터 스트림에서 정지 상태인 객체를 정제하여 질의 처리 시 부하를 줄인다. 또한 어플리케이션의 질의가 이동 객체의 이동이 일정 거리 이상인 데이터에 대해서만 보고받기 원하는 형태가 다수 이므로 RTLS 서버 API의 확장하고, 중요 위치 수집기를 개발하여 일정 거리를 이동한 객체에 대해서만 데이터를 수집하고, 전송하여 네트워크 트래픽을 감소시켜 어플리케이션에 질 높은 서비스를 제공한다.

▶ 정지 상태의 객체 데이터 스트림 제거

질의 처리의 과부하와 네트워크 트래픽의 증가하는 것을 감소시키기 위해 정지 상태의 위치 데이터 스트림의 제거가 필요하다. [그림 8]처럼 연속 질의 색인을 처리하기 전에 정지 상태 제거 필터를 사용함으로써 연속 질의 색인에서 처리하는 위치 데이터 스트림의 양은 감소하게 되

어 연속 질의 색인의 부하를 감소 시켰으며 어플리케이션에 전달되는 데이터 량도 감소하게 된다.

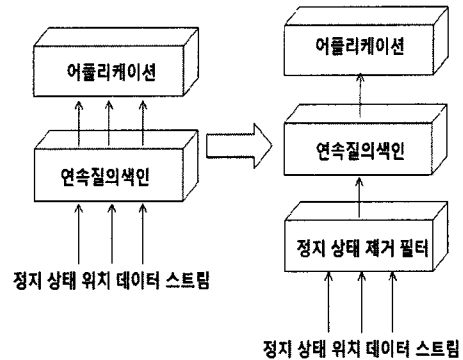


그림 8 정지 상태 제거 필터

▶ 중요 위치 수집기 개발

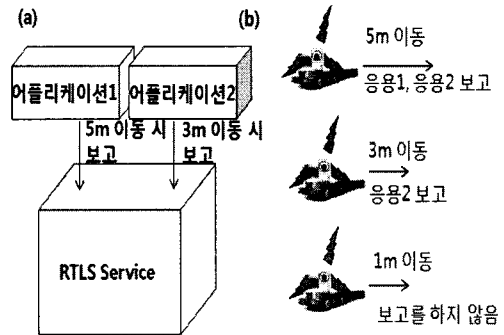


그림 9 중요 위치 수집기 개발

각 어플리케이션들 마다 이동 객체의 이동이 얼마 이상일 때 보고하라는 조건이 다르다. [그림 9]의 (a)에서 어플리케이션 1에서는 객체의 이동이 5m 이상일 때 보고를 원하고, 어플리케이션2에서는 객체의 이동이 3m 이상일 때 보고를 원한다. 하지만 RTLS 서버의 위치 데이터 스트림의 보고 주기대로 보고하게 되면 각 어플리케이션이 보고 받은 대량의 위치 정보를 정제해야 한다. 그러므로 API의 확장으로 [그림 9]의 (b)와 같이 객체의 이동이 어플리케이션에서 원하는 만큼 이상 이동하였을 때 보고하는 중요 위치 수집기를 개발하여 어플리케이션에 질 높은 서비스를 제공한다.

5.4 블록 다이어그램

[그림 10]은 요구 사항을 바탕으로 RTLS 서버의 구조를 블록 다이어그램을 나타낸 것이다. RTLS 서버의 구조는 3계층으로 나누어 구성되어 있다. Capture Layer는

위치 데이터 스트림을 수집하는 역할을 담당하며, Processor Layer는 Capture 인터페이스로부터 수집된 위치 데이터 스트림을 정지 상태 제거 필터, 맵 매칭 필터, Session-based Query 컴포넌트, Stateless Query 컴포넌트, Query 매니저를 통하여 위치 데이터 스트림을 처리하며, Service Layer는 어플리케이션들과 통신하는 API로 구성되어 있다.

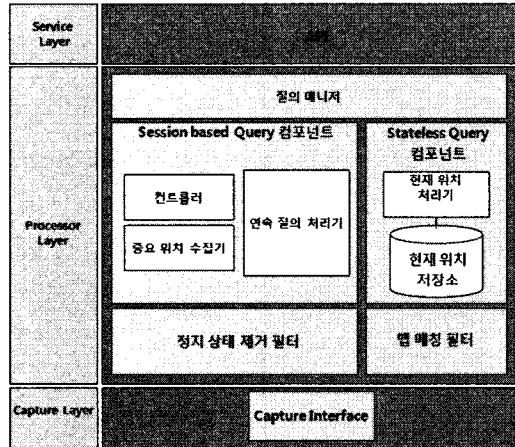


그림 10 RTLS 서버 블록 다이어그램

6. 결론 및 향후 연구

RTLS 시스템은 이동 객체의 위치를 실시간으로 인식하여 이동 객체의 추적과 모니터링을 하기 위한 서비스이다. 실시간으로 위치 데이터 스트림을 수집하여, 어플리케이션에게 제공하기 위해 연속 질의 색인을 도입하여 위치 데이터 스트림을 처리하며, RTLS 전파 굴절 현상으로 인한 비정상적인 위치 데이터 스트림을 보정하여 어플리케이션에게 보다 신뢰성이 높은 위치 데이터 스트림을 제공한다. 또한 정지 상태 제거 필터를 사용하여, 연속 질의 색인에서 처리하는 위치 데이터 스트림의 양을 감소시켜 RTLS 서버의 질의 처리 시 부하를 줄이며, 확장 API와 중요 위치 수집기를 개발하여 어플리케이션에 질 높은 서비스를 제공한다.

향후 본 논문에서 설계된 RTLS 서버를 바탕으로 보다 효율적인 질의 처리 기법과 중복 제거 기술을 설계하고, 보다 많은 API 확장의 통하여 위치 데이터 스트림을

효율적으로 제공하는 RTLS 서버의 설계가 필요하다.

7. 참고 문헌

- [1] American National Standard Institute, Inc., Real Time Location Systems(RTLS)-Part 3:Application Programming Interface, 2003
- [2] Kun-Lung Wu, Shyh-Kwei Chen, Philip S. Yu, "Processing Continual Range Queries over Moving Objects Using VCR-Based Query Indexes," mobiquitous, pp. 226-235
- [3] D.V Kalashnikov, S.Prabhakar. W.G. Aref, and S.E.Hambrusch, Efficient Evaluation of continuous range queries on moving objects, In Proc. of 13th DESA, 2002
- [4] C.E. White, D. Bernstein, and A.L. Kornhauser. Some map matching algorithms for personal navigation assistants. Transportation Research Part C, 8:91-108, 2000.