

논문 2012-49TC-1-5

유비쿼터스 컴퓨팅 환경을 위한 RFID/WSN 통합 관리 시스템에 관한 연구

(A Study on The RFID/WSN Integrated system
for Ubiquitous Computing Environment)

박 용 민*, 이 준 혁**

(Yong-Min Park and Jun-Hyuk Lee)

요 약

유비쿼터스 컴퓨팅 환경을 구현하기 위해서 가장 핵심이 되는 기술은 USN(Ubiquitous Sensor Network)기술로써 RFID(Radio Frequency Identification)와 무선 센서 네트워크(Wireless Sensor Network WSN)를 이용한 다양한 센서 기술과 프로세서 집적 기술 그리고 무선 네트워크 기술을 이용해서 실제 물리적 환경 정보를 원격에서 손쉽게 수집하고 모니터링 하는 것을 가능하게 함으로써 기존 가상 공간의 IT 기술을 다양한 실제 환경에서 확대할 수 있도록 한다. 그러나 RFID와 WSN은 이러한 기술적 유사성과 상호 영향에도 불구하고 별 개의 연구로 인식되었으며 RFID와 WSN의 기술적인 융합에 대한 연구는 미비한 수준이다. 이러한 문제점을 인식한 EPCglobal에서는 국제 표준인 EPCglobal Network를 기반으로 RFID와 WSN 기술을 효율적으로 통합/연동할 수 있는 EPC 센서 네트워크(EPC Sensor Network)를 제안하였다. 제안된 EPC 센서 네트워크는 미들웨어에서 복합 이벤트 처리(Complex Event Processing) 기법을 이용하여 RFID와 WSN을 통해 발생하는 데이터를 단일 환경에서 통합하고, 발생된 이벤트를 EPCglobal Network를 기반으로 연동하는 기술이다. 하지만 이러한 EPC 센서 네트워크는 미들웨어에서 복합 이벤트를 검출하기 위해 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하지 못하는 경우에도 계속적인 검사를 수행하기 때문에 연산 비용이 증가되는 문제점이 있다. 또한 센서 데이터의 표현을 위해 RFID 태그의 EPC를 기반으로 센서 데이터를 하나의 결합된 쌍으로 표현한다. 이것은 센서 데이터를 처리하는데 있어 반드시 EPC가 있어야 함을 의미한다. 즉, 센서 데이터만을 위한 관리 기능은 제공하지 못하는 문제점이 있다. 이러한 EPC 센서 네트워크의 문제점들을 해결하기 위하여 본 논문에서는 RFID/WSN 통합 관리 시스템을 제안한다. 제안하는 RFID/WSN 통합 관리 시스템은 첫 번째, RFID와 WSN 데이터를 인터넷 표준 프로토콜인 SIP(Session Initiation Protocol) 기반의 통합 관리 시스템을 제안하였다. 두 번째, 미들웨어에서 불필요한 복합 이벤트 검출 연산을 위하여 복합 이벤트의 최소 조건을 정의하고 조건을 만족하는 경우에만 복합 이벤트를 검출하는 알고리즘을 제안하였다. 제안하는 기법들의 성능을 평가하기 위하여 SIP 기반의 통합관리시스템을 구현하여 성능을 평가하였다.

Abstract

The most critical technology to implement ubiquitous health care is Ubiquitous Sensor Network (USN) technology which makes use of various sensor technologies, processor integration technology, and wireless network technology-Radio Frequency Identification (RFID) and Wireless Sensor Network (WSN)-to easily gather and monitor actual physical environment information from a remote site. With the feature, the USN technology can make the information technology of the existing virtual space expanded to actual environments. However, although the RFID and the WSN have technical similarities and mutual effects, they have been recognized to be studied separately, and sufficient studies have not been conducted on the technical integration of the RFID and the WSN. Therefore, EPCglobal which realized the issue proposed the EPC Sensor Network to efficiently integrate and interoperate the RFID and WSN technologies based on the international standard EPCglobal network. The proposed EPC Sensor Network technology uses the Complex Event Processing method in the middleware to integrate data occurring through the RFID and the WSN in a single environment and to interoperate the events based on the EPCglobal network.

* 정회원-교신저자, 삼육보건대학 의료정보시스템과 (Sahmyook Health University)

** 정회원, 한국정보통신기술대학 정보통신설비과 (Korea Information & Communication Polytechnic College)

※ 본 연구는 2011년도 삼육보건대학 교육역량강화사업의 학술연구지원사업에 의해 연구되었음.

접수일자: 2011년11월4일, 수정완료일: 2012년1월17일

However, as the EPC Sensor Network technology continuously performs its operation even in the case that the minimum conditions are not to be met to find complex events in the middleware, its operation cost rises. Moreover, since the technology is based on the EPCglobal network, it can neither perform its operation only for the sake of sensor data, nor connect or interoperate with each information system in which the most important information in the ubiquitous computing environment is saved. Therefore, to address the problems of the existing system, we proposed the design and implementation of USN integration management system. For this, we first proposed an integration system that manages RFID and WSN data based on Session Initiation Protocol (SIP). Secondly, we defined the minimum conditions of the complex events to detect unnecessary complex events in the middleware, and proposed an algorithm that can extract complex events only when the minimum conditions are to be met. To evaluate the performance of the proposed methods we implemented SIP-based integration management system.

Keywords : USN, CEP, Middleware, Ubiquitous Computing, SIP

I. 서 론

유비쿼터스 컴퓨팅을 위한 핵심 기술은 RFID와 WSN을 이용하여 실제 물리적 환경 정보를 원격에서 손쉽게 수집하고 모니터링하여 기존 가상 공간의 IT 기술을 다양한 실제 환경으로 확대할 수 있는 기술이다.

이러한 RFID와 WSN은 식별기술이므로 위치를 관리하는 프레임워크와 같이 동작하여 물체의 식별 및 위치 추적 시스템을 구축할 수 있다. 기존의 연구에서는 EPC를 기반한 특별한 미들웨어를 사용하는 기법이 소개되었다. 그러나 미래에는 다양한 네트워크 기술들이 인터넷 프로토콜(IP)을 기반으로 수렴하여 이른바 All IP 네트워크를 구성할 것으로 기대된다. 따라서 IP 기반의 USN 관리 시스템은 적은 비용으로 설치가 가능하고 기존의 IP 기반의 네트워크와의 쉬운 연동을 가능하게 할 것이다^[1~3].

그러나 RFID와 WSN은 이러한 기술적 유사성과 상호 영향에도 불구하고 별개의 연구로 인식되었으며 RFID와 WSN의 기술적인 융합에 대한 연구는 미비한 수준이다. 또한 지금까지의 WSN은 미리 준비된 환경에서 제한된 사용자만을 위한 특정 어플리케이션을 위한 서비스로 제공되었기 때문에 인터넷을 통한 연동은 제공되지 않았다. 따라서 무선 센서 네트워크를 인터넷으로 확장해서 다양한 형태의 서비스를 제공하지 못했으며 다른 센서 네트워크간의 데이터를 공유할 수 있는 표준화가 마련되지 못했다^[4~5].

이러한 문제점을 인식한 EPCglobal에서는 Auto-ID Lab에서 RFID의 인터넷 아키텍처에 WSN을 결합한 EPC 센서 네트워크를 제안하였다. EPC 센서 네트워크는 국제 표준인 EPCglobal Network를 기반으로 RFID와 WSN 기술을 인터넷과 통합하는 아키텍처를 제공함으로써 언제 어디서나 인터넷을 기반으로 다양한 센서 네트워크와 RFID 서비스를 제공하는 유비쿼터스

컴퓨팅 환경을 제공하는데 목적을 두고 있다[6-8]. EPC 센서 네트워크는 RFID와 WSN을 통해 발생하는 데이터를 단일 환경에서 통합 처리하기 위해 미들웨어에서 복합 이벤트 처리(Complex Event Processing) 기법을 이용하여 EPCglobal Network에서 센서 데이터를 처리할 수 있게 된다. 복합 이벤트 처리 방식은 여러 이벤트 소스로부터 발생한 이벤트를 대상으로 이벤트들의 영향을 분석하여 대응되는 액션을 처리하는 방식으로 가장 핵심이 되는 기술은 어떻게 사용자에게 의미있는 이벤트(복합 이벤트)를 검출하는가이다^[9~10]. 복합 이벤트는 기본 이벤트의 조합으로 이루어진 이벤트로, 복합 이벤트를 구성하는 모든 이벤트가 발생하여야지만 응용 계층으로 결과를 전달할 수 있다. EPC 센서 네트워크에서 제시하는 복합 이벤트 처리 방식은 아직까지 표준으로 정의되진 않았지만, 여러 논문을 통해 가능성을 제시하였다. 지금까지 제시된 복합 이벤트 검출 알고리즘은 오토마타 기법을 이용한 SASE 알고리즘^[11]과 복합 이벤트의 패턴을 그래프 형태로 구성하고 발생한 이벤트를 검출하는 RCEDA 알고리즘^[12]이 대표적이다. 이러한 알고리즘은 공통적으로 발생하는 모든 기본 이벤트를 상위노드로 전달하여 비교 연산을 수행하기 때문에 기본 이벤트의 수가 증가함에 따라 복합 이벤트를 검출하기 위한 연산 비용이 크게 증가하는 문제점을 가지고 있다.

또한 EPC 센서 네트워크는 EPCglobal Network를 기반으로 하기 때문에 반드시 식별자인 EPC(Electronic Product Code)를 통해 연계 및 연동이 이루어진다. 따라서 데이터 저장소인 EPCIS (EPC-Information Server)에는 RFID EPC를 기반으로 센서 데이터를 하나의 결합된 쌍으로 저장된다. 즉, EPC를 기반으로 저장되기 때문에 센서 데이터만을 위한 관리 기능은 제공하지 못하는 문제점을 가지고 있다.

본 논문에서는 기존 시스템의 문제점들을 해결할 수

있는 SIP(Session Initiation Protocol) 기반의 USN 통합 관리 시스템을 제안하였다. 이를 위해 첫 번째, 제안하는 시스템은 인터넷 표준 프로토콜 중 SIP[13]를 기본 프로토콜로 사용하기 때문에 다른 인터넷 표준 프로토콜과 완벽하게 호환이 가능하다. SIP는 응용계층에서 세션을 열거나 닫는 역할을 하는 프로토콜로, 인터넷 전화 서비스(VoIP)나 IMS(IP Multimedia Subsystem)^[14] 등에서 세션을 관리하는 프로토콜로서 널리 쓰이고 있다. 이러한 세션을 관리하는 기능에 더해, SIP는 인터넷에서 호스트의 이동성을 지원한다. 제안하는 시스템은 이러한 SIP의 이동성 지원 특성을 활용하여 RFID 태그 및 센서 노드의 위치를 추적하게 된다. 일반적으로 RFID 태그 및 센서 노드의 하드웨어 용량은 IP 스택을 올리기에 충분하지 않다. 또한 SIP 메시지를 주고받을 수 있는 기능을 기대할 수도 없다. 따라서 본 시스템은 기존 SIP 아키텍처들을 이루는 요소 외에 추가로 두 가지 핵심 컴포넌트인 UUA(USN User Agent)와 UNS(USN Name Server)를 제안하였다. UUA는 제한된 능력을 가진 RFID 태그 및 센서 노드를 대신해서 위치 갱신 과정을 수행하는 역할을 한다. UNS는 RFID 식별자 및 센서 노드의 식별자와 SIP URI(Universal Resource Identifier) 사이의 변환을 담당한다. 두 번째, 미들웨어에서 대용량 RFID와 WSN 데이터 스트림에 대한 효율적인 복합 이벤트 처리를 위해 기존 연구의 문제점을 해결한 새로운 알고리즘을 제안하였다. 제안하는 알고리즘은 복합 이벤트를 질의 형태로 등록하고 기본 이벤트가 발생하면 질의 색인을 통하여 복합 이벤트에 접근한다. 질의는 복합 이벤트를 구성하는 기본 이벤트의 발생 여부만을 비교하고 복합 이벤트를 구성하는 기본 이벤트가 모두 발생 할 때 실제 복합 이벤트 검출 연산을 수행한다.

본 논문의 세부 구성은 다음과 같다. 제 II장에서는 EPC 센서 네트워크 및 문제점을 설명하고, 제 III장에서는 제안하는 RFID/WSN 통합 관리 시스템과 복합 이벤트 검출 알고리즘에 관하여 설명한다. 제 IV장에서 제안하는 시스템의 성능 평가를 수행하고 결과를 설명한다. 제 V장에서는 비교 및 고찰을 설명하고 마지막으로 제 VI장에서 결론을 맺는다.

II. 관련 연구

(1) EPC 센서 네트워크

RFID는 각 사물들의 글로벌한 식별을 가능하게 하고 RFID에서 사실 상의 표준인 EPCglobal Architecture Framework는 RFID 데이터를 처리하기 위한 프레임워크로서 만들어졌다. 하지만 EPCglobal Architecture Framework는 RFID 데이터만을 위해서 디자인된 아키텍처이므로 사물들의 주변 환경 정보와 같은 데이터를 제공하는 WSN에 대해서는 고려하지 않았다. 반대로 WSN의 경우 다양한 센서 네트워크가 존재하면서 하나의 통일된 규격이 존재하지 않는다. 이를 보완하기 위해 RFID와 WSN를 통합한 미들웨어 프레임워크인 EPC 센서 네트워크가 제안되었다. EPC 센서 네트워크는 RFID의 인터넷 아키텍처에 WSN를 결합한 인프라로 정의 될 수 있으며 인터넷 연동 기술로써 EPCglobal의 아키텍처 프레임워크를 기반으로 하고 있다. EPC 센서 네트워크에서는 논리적 컴포넌트의 기능을 정의하는 대신에 각각의 통신 인터페이스를 정의하고 있으며 이들 인터페이스를 변경함으로써 RFID 뿐 만 아니라 WSN의 센싱 특성을 통합한 다양한 형태의 RFID가 지원될 수 있도록 하였다. 그림 1은 EPC 센서 네트워크의 구조를 보여준다^[15~17].

EPC 센서 네트워크 미들웨어는 이벤트 기반의 복합 이벤트 처리를 이용한 이벤트 처리 컴포넌트로 EPCglobal Network에서 센서 데이터의 처리에 사용된다. EPC 센서 네트워크에서의 복합 이벤트 처리 연구는 기존 EPCglobal Network의 ALE 미들웨어 외에 자체적인 미들웨어 구조로 디자인하였다. 복합 이벤트 처리는 여러 이벤트 소스로부터 발생한 이벤트를 대상으로 이벤트들의 영향을 분석하여 대응되는 액션을 처리하는 방식으로 가장 핵심이 되는 기술은 어떻게 사용자에게 의미있는 이벤트(복합 이벤트)를 검출하는가이다. EPC 센서 네트워크에서 지금까지의 이벤트 검출 알고리즘은 SASE 또는 RCEDA 기법을 이용하여 USN 데

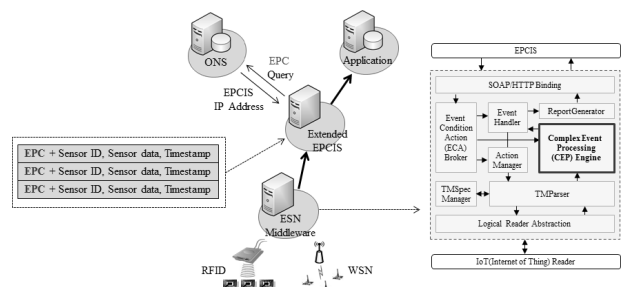


그림 1. EPC 센서 네트워크 구조
Fig. 1. The structure of EPC sensor network.

이터를 통합하기 위한 복합 이벤트 처리 알고리즘이 제안되었다.

일반적으로 복합 이벤트 처리에서 이벤트는 기본 이벤트(primitive event)와 복합 이벤트(complex event)로 구성된다. 기본 이벤트는 하나의 이벤트로 표현되며, 복합 이벤트는 기본 이벤트의 연관성을 기초로 만들어지고 의미있는 데이터로 변환하여 응용 계층으로 전송한다. 복합 이벤트 언어는 DBMS에서 사용하는 SQL과 비슷한 구조를 가지고 있다^[18~19].

SASE는 기본 이벤트의 패턴과 조건 그리고 시간 간격의 구성을 통해 복합 이벤트를 정의하기 위한 언어를 제안하였다. SASE는 5가지의 처리 과정을 수행하여 복합 이벤트의 적합성을 검사한다. 먼저 등록된 복합 이벤트는 오토마타 형태로 구성하고 이벤트가 발생함에 따라 이벤트의 상태 변화를 스택에 저장한다. 이때, 변화가 발생하기 전의 상태는 포인터로 연결된다. 이벤트 발생에 따라 상태의 변화가 오토마타의 최종 단계에 도착하게 되면 포인터를 따라 발생한 이벤트의 조합을 구성한다. 이 처리 과정을 SSC(Sequence Scan and Construction)이라 한다. SSC를 통하여 구성된 이벤트 조합은 실제 복합 이벤트의 후보 집합으로 SELECTION 연산을 수행하여 복합 이벤트의 조건을 만족하는지 검사한다. 또한, WINDOW 연산을 통해 시간적인 조건을 만족하는지 검사한다. 마지막으로 NG(negation) 연산을 통하여 발생하지 않는 데이터를 처리하게 된다. 모든 조건을 만족하는 이벤트 조합은 복합 이벤트를 등록된 응용 계층에게 전달한다.

하지만 SASE는 복합 이벤트의 패턴이 SEQUENCE 연산으로 한정되어 수행된다. SASE에서는 복합 이벤트를 오토마타의 형태로 정의하며 기본 이벤트가 발생함에 따라 복합 이벤트의 상태를 단계별로 나누어 변화시킨다. 이벤트의 상태는 발생하는 기본 이벤트의 순서에 따라 결정됨으로 순서를 고려하는 SEQUENCE 연산만이 오토마타로 구성될 수 있다. SEQUENCE 이외의 연산은 오토마타의 형태로 구성과 복합 이벤트의 상태를 단계별로 정의하는 명확한 기준을 제공하지 못한다. 따라서 SASE는 다양한 형태의 복합 이벤트를 처리할 수 없는 문제점이 있다. 또한 SASE에서는 각각의 복합 이벤트는 인스턴스 스택에 기본 이벤트를 저장하고 발생한 시간에 따라 포인터를 연결하는 구조를 가지고 있다. 따라서 같은 기본 이벤트로 구성되어 있는 복합 이벤트도 각각의 기본 이벤트를 저장하고 있어야 한다.

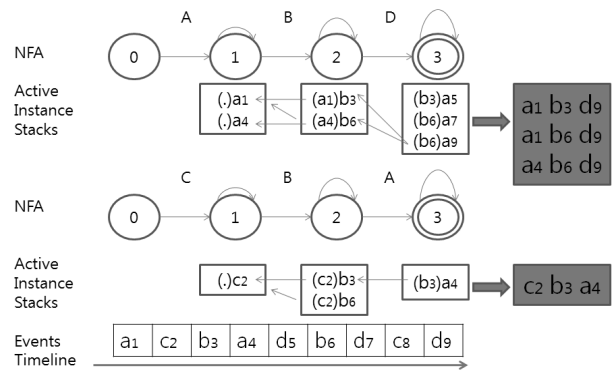


그림 2. 동일한 이벤트로 생성된 복합 이벤트
Fig. 2. Complex event generated by the same event.

이는 데이터의 중복 저장을 의미한다. 그림 2는 이러한 문제점을 나타낸 것이다.

RCEDA 알고리즘은 그래프를 이용하여 복합 이벤트 처리기법을 제안하였다. RCEDA에서는 복합 이벤트를 효율적으로 처리하기 위하여 복합 이벤트를 정의하는 규칙을 제안하였고 복합 이벤트를 구성하기 위해 기본 이벤트를 조합하기 위한 연산자를 정의하였다. 이러한 연산자는 AND, OR, NOT과 같은 기본 연산자와 순차적인 이벤트를 묶는 SEQ, 시간 조건이 포함된 TSEQ 연산, 동일한 이벤트의 연속적인 처리를 위한 SEQ+, TSEQ+ 연산 등이 있다. 실제 복합 이벤트를 처리하기 위해 어플리케이션에서 등록된 복합 이벤트는 계층적인 그래프로 저장한다. 그래프의 단말노드는 기본 이벤트로 구성되어 있으며 비 단말노드는 이벤트의 연산자로 구성되어 있다. 기본 이벤트가 발생하면 그래프의 단말노드에서부터 이벤트의 조건을 검사하여 상위 노드로 전달된다. 이때, 발생하지 않는 이벤트의 연산자가 포함된 노드를 만난 경우는 상위노드에서 Pseudo 이벤트를 발생 시킨다. 발생 하지 않는 연산은 하위 노드에서 상위노드로 이벤트를 전달하는 시점을 선정 할 수 없기 때문에 발생한 Pseudo 이벤트는 정해진 시간에 하위 노드로 질의를 전달하고 조건을 만족하는 경우 다시 상위 노드로 전달된다. 이벤트가 최종 루트노드에서 조건을 만족하게 되면 복합 이벤트를 등록된 응용 계층에게 최종 이벤트가 전달된다. 하지만 그림 3에서 처럼 E_1, E_2, E_3, E_4 의 이벤트로 구성된 복합 이벤트가 있고 이 중 기본 이벤트 E_1, E_3, E_4 만 빈번히 발생하고 E_2 이벤트는 발생하지 않는다면, 중간단계의 후보 집합인 E_3, E_4 의 조합 결과는 계속 생성되지만 최종 복합 이벤트는 발생하지 않는다.

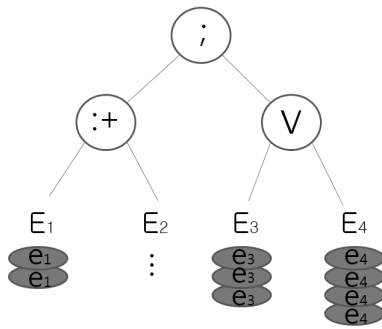


그림 3. RCEDA에서 이벤트 처리과정의 문제점
Fig. 3. Process in the event of a problem RCEDA.

RCEDA에서 기본 이벤트가 발생하면 하위 노드에서 상위 노드로 전달된다. 이벤트를 전달 받은 상위노드는 하위 노드에서 발생한 이벤트를 저장하고 하위 노드들로부터 전달받은 이벤트를 이용하여 중간 단계의 후보 집합을 생성하여 상위 노드로 전달해야 한다. 따라서 상위 노드는 모든 중간 단계의 후보 집합 이벤트를 저장하기 위한 추가적인 구조가 필요하다. 실제 복합 이벤트의 발생에 영향을 주지 않는 불필요한 중간 단계 후보 집합도 상위 노드로 전달되어 저장하기 때문에 불필요한 저장비용을 소모한다.

RCEDA는 여러 복합 이벤트들의 그래프 공유에 대한 문제점이 발생한다. 복합 이벤트는 서로 다른 복합 이벤트와 그래프의 노드를 공유할 수 있다. 하지만 연산을 구성하는 연산자와 하위노드 연산시간이 모두 일치하지 않으면 공유할 수 없다. 따라서 그래프의 단말 노드들은 공유가 용이하지만 비 단말 노드는 쉽게 공유되지 않는다.

EPC 센서 네트워크는 글로벌 식별자인 EPC를 사용하고 EPCglobal Network를 다양한 UNS를 지원하도록 확장함으로써 RFID 태그만을 지원하고 있는 EPCglobal Network와는 달리 UNS의 다양한 종류의 센서와 액추에이터(actuator)가 활용 가능하다. 즉, 센서 데이터의 표현을 위해 모든 RFID 태그나 센서 네트워크 노드는 EPC를 기반으로 센서 데이터 또는 RFID 태그의 식별 정보는 해당 EPC와 결합된 쌍으로 표현된다. 따라서 그림 4와 같이 EPC를 기반으로 센서 데이터까지도 공유하고 사용자에게 제공할 수 있는 데이터베이스인 Extended EPCIS를 제안하였다. Extended EPCIS는 EPCglobal Network에서 EPC 데이터만을 저장할 수 있는 EPCIS를 확장한 개념으로 EPC 센서 네트워크 미들웨어로부터 전달되는 RFID 및 센서 데이터

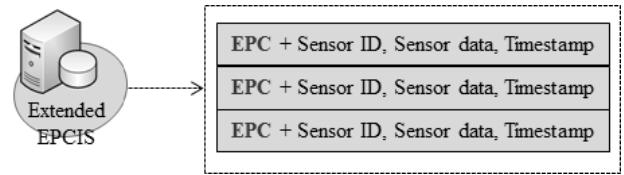


그림 4. 확장된 EPCIS 구조
Fig. 4. Structure of extended EPCIS.

를 저장하고 이를 필요한 사용자 및 응용에서 표준 인터페이스를 통해서 공유할 수 있도록 해주는 분산 데이터베이스 시스템이다. 하지만 EPC를 기반한 관리 시스템은 RFID EPC 식별자가 반드시 존재해야지만 운용할 수 있는 시스템으로 단일 센서 데이터만을 위한 기능은 제공하지 못하는 문제점이 있다.

III. 제안하는 RFID/WSN 통합 관리 시스템

본 장에서는 기존 시스템에서의 문제점들을 해결하기 위하여 새롭게 제안하는 SIP 기반의 RFID/WSN 통합관리 시스템에 대하여 기술한다. 제안하는 시스템은 미들웨어에서 비트맵 인덱스 기반의 복합 이벤트 검출 알고리즘, 제한된 능력의 RFID 태그 및 센서 노드를 대신해 위치 등록과정을 수행하는 UUA(USN User Agent), RFID 태그 및 센서 노드의 위치 추적을 위한 이름 변환 서비스(name resolution service)를 제공하는 UNS(USN Name Server)를 제안하였다.

(1) SIP 기반 RFID/WSN 통합 관리 시스템 모델

제안하는 관리시스템 구조가 그림 5에 나타나 있다. 일반적인 SIP 구조는 SIP 서버와 UA로 구성 된다. SIP 서버는 기능에 따라 다시 proxy/redirect, registrar 서버로 분류된다. proxy/redirect 서버는 자신이 받은 SIP 메시지를 목적지로 전송하거나 받은 메시지를 목적지로 가기 위해 거쳐 갈 수 있는 다른 곳을 지정해서 알려주게 된다. registrar는 이동성을 지원하기 위해 RFID 태그 및 센서 노드의 위치 정보를 저장하고 있는 서버이다. 제안하는 관리 시스템에서 registrar는 RFID 태그 및 센서 노드의 현재 위치 및 상태를 저장하고 있다. 제안하는 관리 시스템은 일반적인 SIP 구조에 RFID와 센서 네트워크를 통해 발생된 데이터를 단일 환경에서 통합하기 위해 EPC 센서 네트워크와 마찬가지로 복합 이벤트 처리를 기반으로 한 미들웨어를 제안하였다. 일반적으로 RFID 태그 및 센서 노드의 하드웨어 용량은 IP

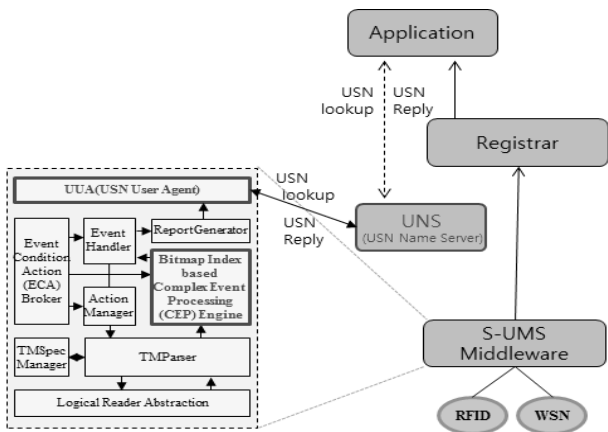


그림 5. 제안하는 SIP 기반 USN 통합 관리 시스템
Fig. 5. The proposed USN management system SIP-based.

스택을 올리기에는 충분하지 않다. 또한 SIP 메시지를 주고받을 수 있는 시그널링 기능을 기대할 수도 없다. 따라서 제안하는 시스템은 그림 5와 같이 기존 SIP 아키텍처들을 이루는 요소 외에 추가로 두 가지 핵심 컴포넌트인 UUA와 UNS를 제안한다. UUA는 제한된 능력을 가진 RFID 태그 및 센서 노드를 대신해서 위치 갱신 과정을 수행하는 역할을 한다. UNS는 RFID EPC 식별자 및 센서 노드의 유일한 식별자인 MAC 주소와 SIP URI(Universal Resource Identifier) 사이의 변환을 담당한다.

(2) 복합 이벤트 처리를 위한 시스템 구성도

기존 연구의 문제점을 해결하기 위해 제안하는 복합 이벤트 검출 알고리즘은 복합 이벤트를 구성하는 기본 이벤트들이 모두 발생할 때 복합 이벤트를 검출하도록 검출 시점을 조절한다. 제안하는 복합 이벤트 처리를 위한 모델은 그림 6과 같이 복합 이벤트를 구성하는 모든 기본 이벤트들이 발생되었는지를 판별하기 위해 비

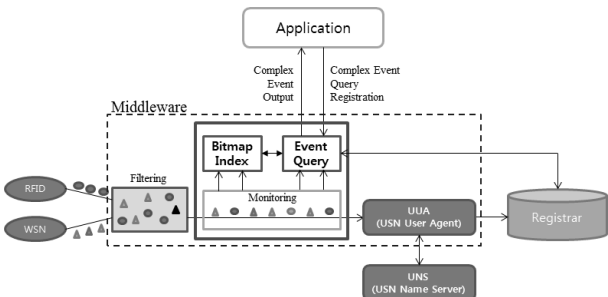


그림 6. 복합 이벤트 처리를 위한 모델
Fig. 6. The proposed model for complex event processing.

트맵 인덱스를 구성하며, 복합 이벤트를 위한 질의는 복합 이벤트를 구성하는 기본 이벤트중 마지막 이벤트를 트리거 이벤트로 정의하고, 트리거 이벤트를 기준으로 복합 이벤트 질의를 등록한다. 복합 이벤트 질의는 연산자의 특성에 의해 트리 형태로 구성된다. 실제 기본 이벤트가 발생하면 비트맵 인덱스를 통해 발생시간을 기록하며, 만약 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하면 등록된 복합 이벤트를 검출하기 위해 복합 이벤트 트리를 검사한다.

(3) 비트맵 인덱스

논문에서는 비트맵 인덱스를 이용하여 기본 이벤트의 발생 이력을 저장한다. 그림 7과 같이 비트맵 인덱스는 논리적인 그리드 영역에 대응하는 비트 열을 가진 복합 이벤트 연산 시간동안 유지하는 2차원 배열 형태를 가진다.

복합 이벤트 질의에서는 기본 이벤트의 최종 발생 시간만을 저장한다. 과거에 발생한 기본 이벤트의 정보는 복합 이벤트 질의에서 유지되지 않는다. 따라서 복합 이벤트를 검출하기 위하여 과거에 발생한 기본 이벤트의 이력을 조사해야 한다. 기본 이벤트가 발생하게 되면 그리드 영역을 통과하는 기본 이벤트는 발생한 시간대에 일치하는 비트열에 해당 그리드 영역에 대응하는 비트를 갱신한다. 기본 이벤트가 발생하게 되면 기본 이벤트는 발생한 시간대에 일치하는 비트열에 해당 그리드 영역에 대응하는 비트를 갱신한다. 비트열은 단순히 어느 시간대에 어떤 이벤트가 발생했는지의 정보만을 유지하고 실제 발생한 기본 이벤트는 데이터베이스

	Event						
	a	b	c	d	e	f	g
T ₀	0	1	0	0	0	0	0
T ₁	1	0	0	1	0	1	0
T ₂	1	1	0	0	0	0	1
T ₃	0	0	0	0	0	1	0
T ₄	0	1	0	1	0	0	0
T ₅	0	0	1	0	0	0	0
T ₆	0	0	0	0	1	1	0
⋮							
T _t	0	0	0	0	1	0	0

그림 7. 비트맵 인덱스 구조
Fig. 7. Structure of bitmap index.

에 저장하게 된다. 즉, 비트맵 인덱스는 기본 이벤트의 발생 유·무와 데이터베이스에 저장된 이벤트를 검색하기 위한 인덱스로 사용된다. 예를 들어 T_i 시간에 e 이벤트가 발생하였다면 비트열을 '1'의 값으로 기록된다.

(4) 이벤트 질의(Event Query)

이벤트 질의는 응용 계층에서 등록된 복합 이벤트를 효율적으로 처리하기 위하여 복합 이벤트 정보를 저장하는 구조이다. 이벤트 질의는 응용 계층에서 등록된 복합 이벤트를 연산자의 특성에 의해 트리 형태로 구성한다. 또한 등록된 복합 이벤트를 구성하는 모든 이벤트 중 마지막 이벤트를 트리거 이벤트로 정의하며, 이 트리거 이벤트를 기준으로 복합 이벤트 질의를 등록하게 된다. 트리거 이벤트는 정의 1과 같다.

[정의 1] 트리거 이벤트(Trigger Event)

트리거 이벤트는 복합 이벤트를 추출할 수 있는 기준 시점을 제공하는 이벤트로 연산자에 따라 정의되며 SEQUENCE 연산의 경우 마지막에 위치한 이벤트를 트리거 이벤트로 정의하며, 다른 연산자에 의한 연결된 이벤트들은 마지막 항의 연결된 모든 이벤트를 트리거 이벤트로 정의한다.

예를 들어, 그림 8과 같이 복합 이벤트를 위한 질의 Q1, Q2, Q3가 응용 계층에서 등록되었다면, 이벤트 질의는 트리형태로 구성되며, 트리거 이벤트를 기준으로 복합 이벤트 질의를 등록하게 된다. 여기서 트리거 이벤트는 d, c, h가 된다. 또한 그림 8에서와 같이 하나의 트리거 이벤트(d)는 다른 복합 이벤트들 사이에서 공유될 수 있다.

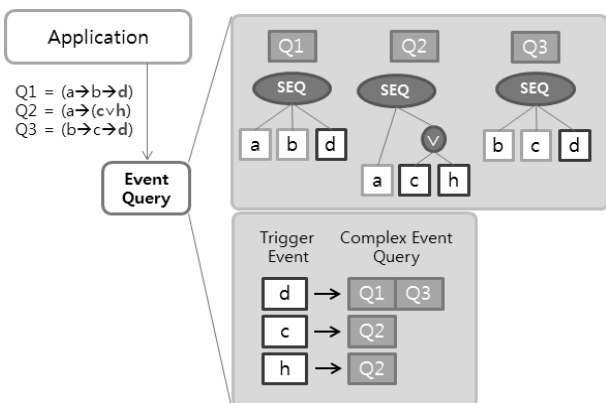


그림 8. 이벤트 질의
Fig. 8. Event query.

(5) 제안하는 복합 이벤트 검출 과정

그림 9는 복합 이벤트 검출을 위한 처리 과정을 나타낸다. 먼저 복합 이벤트 질의는 트리 형태와 트리거 이벤트를 기준으로 복합 이벤트 질의를 등록한다. 수신된 스트림 데이터에서 트리거 이벤트인 d의 기본 이벤트가 발생하였다면, 비트맵 인덱스에 발생한 시간을 기록하며, 이벤트 질의를 통해 자신을 포함한 복합 이벤트 질의인 Q1과 Q3에 접근한다. Q1과 Q3 각각의 질의는 d 이벤트를 제외한 다른 이벤트의 발생 유/무를 식별하기 위해 비트맵 인덱스를 통해 검색한다. 비트맵 인덱스의 검색은 복합 이벤트 질의가 최초 등록된 시점에서 트리거 이벤트가 발생한 시점까지를 시간 범위로 하여 그 범위 내에서 각 질의를 구성하는 이벤트의 발생 유/무를 검색한다. 만약 모든 이벤트가 발생하였다면 각각의 질의는 연산을 수행하며 그렇지 못하다면 연산을 수행

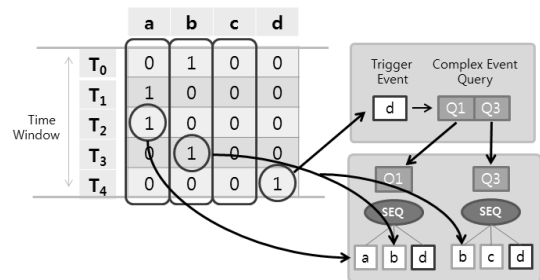


그림 9. 복합 이벤트 검출을 위한 처리 과정
Fig. 9. For the detection of complex event processing.

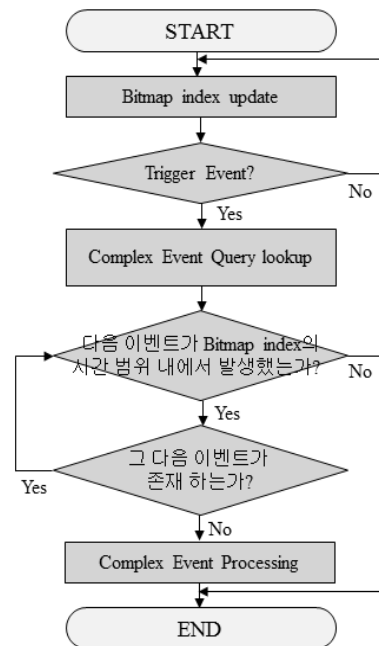


그림 10. 복합 이벤트 검출을 위한 흐름도
Fig. 10. The flow chart for complex event detection.

```

1 algorithm event_Detection(event_list) //이벤트 처리 과정
2 BIT nowBit = 공백 비트열 생성
3 while(event_list 검사)
4   발생한 이벤트를 nowBit에 할당
5 end while
6 while(nowBit의 길이에 대해)
7   if(bit==1)
8     해당 bit에 대응하는 그리드들 찾고 그리드 영역에
9     등록된 복합 이벤트의 값을 갱신
10    if(bit==Trigger event)
11      check_query(Qid)
12      //복합 이벤트의 조건을 검사
13      if(검사 조건을 만족할 경우)
14        make_event_set(Qid)
15        //복합 이벤트를 생성
16      end if
17    end if
18  end while
19  발생한 모든 이벤트를 디스크에 저장
  
```

그림 11. 복합 이벤트 검출 알고리즘
Fig. 11. Complex event detection algorithm.

하지 않는다.

그림 9에서 Q1과 Q3는 SEQUENCE 연산자로 발생 순서가 중요하다. 따라서 Q1에서 d 이벤트가 T4에서 발생하였다면, 그 다음 순서인 b 이벤트는 T0~T4 시간 범위에서 발생 유/무를 검색한다. b 이벤트는 T3 시점에서 발생하였으므로, a 이벤트를 b 이벤트의 발생 시점을 기준으로 T0~T3 시간범위에서 다시 발생 유/무를 검색한다. 그 범위의 T2에서 발생하였기 때문에 Q1 질의는 모든 이벤트가 발생하였으므로 연산을 수행한다. Q3의 경우 마찬가지로 d 이벤트가 T4에서 발생하였다면 c 이벤트는 T0~T4 시간범위에서 발생 유/무를 검색한다. c 이벤트는 그 시간범위에서 발생하지 않았으므로 연산을 수행하지 않는다. 그림 10과 그림 11은 복합 이벤트 검출을 위한 흐름도 및 알고리즘을 나타낸다.

(6) UUA

UUA는 제한된 능력을 가지고 있는 RFID 태그 및 센서 노드를 대신해서 SIP 시그널링을 담당하는 개체이다. SIP 구조의 관점에서 볼 때 UUA는 SIP 클라이언트의 한 종류이다. 그림 12와 그림 13과 같이 UUA는 RFID 태그와 센서 노드의 위치 및 상태 등록, 추적 등의 두 가지 주된 기능을 수행한다.

위치 등록 과정을 위해 우선 RFID 태그 및 센서 노

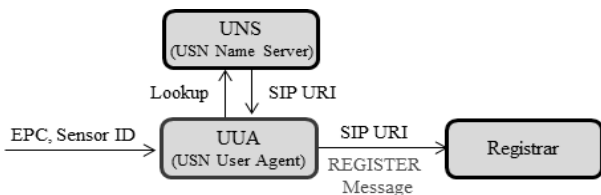


그림 12. RFID 태그 및 센서 노드의 위치 등록
Fig. 12. The location register of RFID tags and sensor nodes.

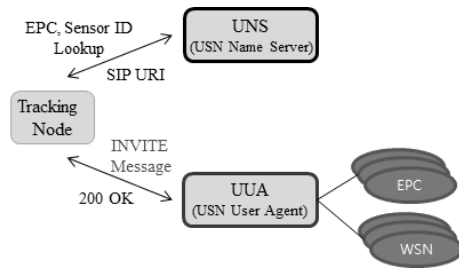


그림 13. RFID 태그 및 센서 노드의 위치 추적
Fig. 13. The location tracking of RFID tags and sensor nodes.

드를 위한 SIP URI가 필요하다. 따라서 UUA는 UNS를 참조함으로써 SIP URI를 알아내게 된다. UNS에게서 SIP URI를 받게 되면 UUA는 RFID 태그 및 센서 노드 홈 도메인에 있는 레지스트라에게 REGISTER 메시지를 보냄으로써 RFID 태그 및 센서 노드의 현재 위치 및 상태를 등록하게 된다. RFID 태그 및 센서 노드의 현재 위치 및 상태 정보, 예를 들면 UUA의 도메인 이름은 REGISTER 메시지의 contact 필드에 들어있다.

RFID 태그 및 센서 노드의 위치 추적 과정에서도 UUA는 중요한 역할을 수행한다. 추적 노드가 RFID 태그 및 센서 노드의 현재 위치를 알고 싶어할 때, 추적 노드는 RFID 태그 및 센서 노드의 현재 위치를 관리하고 있는 UUA에게 INVITE 메시지를 보내게 되고, 이를 받은 UUA는 이에 대한 응답으로 RFID 태그 및 센서 노드의 현재 위치를 알려주게 된다. 바꾸어 말하면, 만약 추적 노드가 관심 있는 RFID 태그 및 센서 노드의 위치를 추적하기 위해 추적 노드가 INVITE 메시지를 UUA에게 보내게 되면, 그 UUA는 추적 노드에게 RFID 리더의 ID 및 싱크 노드의 그룹 ID를 200 OK 메시지의 contact 필드에 담아 보냄으로써 RFID 태그 및 센서 노드의 현재 위치를 알려주게 된다. 여기서 RFID 리더의 ID 및 센서의 그룹 ID란 리더 및 센서 네트워크를 구별하기 위한 식별자이다. 이 200 OK 메시지를 받은 추적 노드는 contact 필드를 분석함으로써 RFID 태그 및 센서 노드의 현재 위치를 알아낼 수 있게 된다.

(7) UNS

UNS는 RFID 태그의 EPC 및 센서 노드의 고유 식별자인 MAC 주소와 SIP URI 사이의 사상을 저장하고 있는 분산 데이터베이스로 정의할 수 있다. UNS는 인터넷의 DNS와 비슷한 조회 서비스를 제공한다. 즉, UNS는 RFID 태그의 고유 식별자인 EPC 및 센서 노드

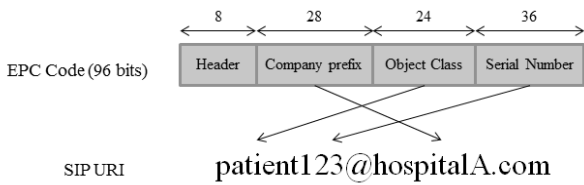


그림 14. EPC를 SIP URI로 변환
Fig. 14. EPC into the SIP URI.

의 고유 식별자인 MAC 주소를 입력으로 받아서 SIP URI를 돌려주는 기능을 수행한다. UUA나 추적 노드는 자신들이 관심 있는 RFID 태그의 EPC 및 센서 노드의 MAC 주소를 UNS에게 보내서 위치 갱신이나 추적 과정에서 사용될 SIP URI를 얻게 된다. 즉, UNS는 자신이 받은 EPC 및 MAC 주소에 대한 SIP URI를 생성하거나 캐쉬된 URI를 찾아내서 그것을 UUA나 추적 노드에게 돌려주는 역할을 수행한다. UNS에서 RFID 태그 및 센서 노드를 위한 SIP URI를 만드는 과정은 그림 14와 같이 RFID의 경우 SIP URI를 만드는 과정은 EPC의 구조를 활용하여 이루어진다.

EPC의 일반 관리자 번호는 SIP URI의 홈 도메인을 만들어내는데 사용된다. 즉, 일반 관리자 번호는 상품의 제조사를 알려주게 되고, 따라서 UNS는 이를 사용해 RFID 태그의 홈 도메인을(예를 들면 hospitalA.com) 알아낼 수 있게 된다. 다음으로 객체 클래스를 조사함으로써 UNS는 상품의 종류를(예를 들면 patient) 알아낼 수 있고, 또한 순서 번호(예를 들면 123)가 RFID 태그를 유일하게 구별하기 위해 사용될 수 있다. 그 결과, UNS는 patient123@hospitalA.com과 같은 형태의 SIP URI를 만들어낼 수 있게 된다.

그림 15과 같이 WSN의 경우 센서 노드의 유일한 식별자인 MAC 주소를 활용하여 이루어진다. MAC 주소는 IEEE Address 또는 Extended Address라고도 부르는데 본 논문에서는 MAC 주소로 표기한다.

센서 노드의 MAC 주소는 전체 64 비트의 크기를 가지며 이중 24 비트는 기관 고유 식별자(OUI : Organizational Unique Identifier)를 나타내며, 40비트는

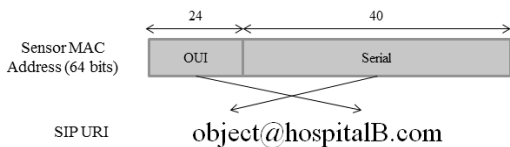


그림 15. 센서 ID를 SIP URI로 변환
Fig. 15. Sensor ID into the SIP URI.

개체의 번호를 나타낸다. 여기서 기관 고유 식별자는 SIP URI의 홈 도메인을 만들어내는데 사용된다. 즉, 기관 고유 식별자는 센서 노드가 사용되는 기관을 알려주게 되고, 따라서 UNS는 이를 이용하여 센서 노드의 홈 도메인을 (예를 들어 hospitalB.com) 알아낼 수 있게 된다. 다음으로 개체의 번호는 센서 노드를 유일하게 식별하기 위해 사용된다. 그 결과, UNS는 object@hospitalB.com과 같은 형태의 SIP URI를 만들어낼 수 있게 된다. 성능이나 실용적인 이유에 의해서 UNS는 분산적이고 계층적인 방법으로 설치될 수 있다. 이러한 경우 UNS에게 조회 서비스를 요청할 때 UUA나 추적 노드는 우선 로컬 UNS에게 먼저 질의를 하게 된다. 만약 로컬 UNS가 이 요청을 직접 처리할 수 있다면, 로컬 UNS가 SIP URI를 돌려주게 된다. 그러나 만약 로컬 UNS가 센서 노드의 홈 도메인에 대한 정보를 가지고 있지 않은 경우, 이 질의를 처리할 수 없고 즉, SIP URI를 만들 수 없다. 이러한 경우에는 로컬 UNS는 이 질의를 해결하기 위해 UNS 계층에서 더 상위에 있는 UNS에게 재귀적으로 질의를 하게 된다. UNS 계층에서 상위에 있는 UNS가 이 질의를 처리하게 되는 경우, 그 결과를 로컬 UNS에게 전달하게 된다. 결과를 받으면 로컬 UNS는 이 결과를 질의 한 UUA나 추적 노드에게 전달하고, 앞으로 있을지 모를 질의에 대비해서 이 결과를 자신의 캐쉬에 저장하게 된다.

IV. 성능 평가 및 결과

제안하는 시스템의 성능 평가를 위해 그림 16과 같이 시스템을 구축하였다. 미들웨어에서 복합 이벤트 처리 엔진은 기존 이벤트 검출 알고리즘과 제안하는 이벤트 검출 알고리즘을 시뮬레이션을 통해 비교 분석 한다. 또한 UUA, UNS를 통해 Application을 구현한다.

미들웨어에서의 복합 이벤트 검출 성능 평가에 사용된 시스템은 Intel Dual Core 2.4GHz 프로세스와

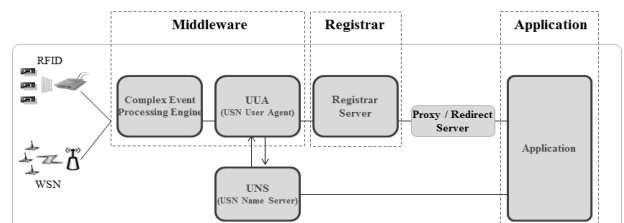


그림 16. 성능 평가를 위한 시스템 구성도
Fig. 16. System configuration for performance evaluation.

2Gbyte의 메모리를 가지고 있으며 운영체제는 Fedora Linux 9를 사용하였다. 복합 이벤트 처리를 위한 모듈은 NS2 시뮬레이터를 이용하여 구현하였다. 성능 평가는 응용 계층으로부터 등록되는 복합 이벤트의 수와 초당 발생하는 기본 이벤트의 수에 따라 기존 연구에서 복합 이벤트 검출을 위한 평균 처리 시간과 제안하는 알고리즘에서의 평균 처리 시간을 비교한다. 또한 기본 이벤트 수를 변경하여 복합 이벤트 검출을 위한 연산 수를 비교 분석한다. 기존 연구인 SASE 기법은 SEQUENCE 연산 이외에 다양한 연산을 포함하는 복합 이벤트를 처리하지 못하는 단점이 있다. 따라서 제안하는 복합 이벤트 처리 기법의 성능 평가는 RCEDA 기법과의 비교 평가를 수행한다. 제안하는 시스템 구현을 위해 표 1과 같이 RFID는 키스컴의 KIS900-W4CH Dev Kit, WSN는 한백전자의 심전도(ECG), 맥박, 혈압, 체온을 측정할 수 있는 바이오 센서 장비를 사용하였다. 레지스트라는 리눅스 기반의 MySQL 데이터베이스를 사용하였으며, Proxy/Redirect 서버는 SIP 2.0 버전을 사용하였다.

그림 17은 미들웨어에서 복합 이벤트 처리를 위한 성능 평가 모델을 나타낸다. 성능 평가 모델은 복합 이벤트 발생기를 통해 복합 이벤트를 복합 이벤트 처리 엔진에 등록하며, 기본 이벤트 발생기는 RFID와 WSN 통해 발생하는 기본 이벤트를 나타낸다. 복합 이벤트 처리 엔진에서는 등록된 복합 이벤트를 기반으로 발생하는 기본 이벤트를 검출하는 기능을 수행한다.

표 1은 복합 이벤트 검출을 위해 기본 이벤트와 복합 이벤트 발생을 위한 파라미터를 나타낸다. 기본 이벤트 발생기는 초당 10,000~50,000개를 발생할 수 있도록 설계 하였으며, 이벤트 형식은 RFID는 (ID, L, T)로 구성되며, WSN은 (ID, L, D, T)로 구성된다. ID는 각 태그 및 센서 노드의 식별자를 나타내며, L은 센싱 위

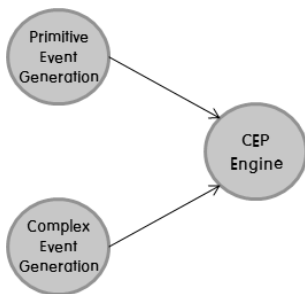


그림 17. 성능 평가 모델
Fig. 17. Performance evaluation model.

표 1. 성능평가를 위한 파라미터

Table 1. Parameter for performance evaluation.

Parameters for primitive event generation		
Parameter	Description	Values
<i>P</i>	Primitive events occur per second (event/sec)	[10,000-50,000]
<i>T</i>	Number of primitive event types (ID, L, D, T)	[1-100]
θ_1	Random number generate of occurrences of event types	[1-100]

Parameters for complex event generation		
Parameter	Description	Values
<i>C</i>	Number of complex events registered	[100-500]
<i>S</i>	Length of the sequence in each complex event	[2-10]
<i>PE</i>	Number of primitive event in each complex event	[1-20]
<i>O</i>	Number of operator in each complex event	[0-20]
<i>L</i>	Number of tree level in each complex event	[1-3]
θ_2	Random number generate of PE, O, L	

치, T는 센싱 시간을 나타내며, WSN의 D는 센싱 데이터를 나타낸다. 또한 복합 이벤트 발생기는 복합 이벤트 등록을 100~500개로 하였으며, 복합 이벤트의 순서 길이는 SEQUENCE 연산자를 통해 이벤트의 발생 순서를 정의하는 것으로, 2~10의 길이를 가질 수 있도록 설계하였다. 또한 각 복합 이벤트의 기본 이벤트 수는 2~10개의 기본 이벤트 수를 가질 수 있도록 설계하였으며, 연산자 수는 0~20개의 연산자를 가질 수 있도록 설계하였다.

(1) 복합 이벤트 검출을 위한 처리 시간

제안하는 복합 이벤트 검출을 위한 평균 처리 시간은 등록되는 복합 이벤트 수와 기본 이벤트의 발생으로 복합 이벤트 처리 엔진에서 처리 시간을 측정한다. 식 1은 등록되는 복합 이벤트 수(*C*)를 나타낸 것으로 *Min*은 등록되는 복합 이벤트의 초기값을 나타내며, *Max*는 등록되는 복합 이벤트의 최대값을 나타낸다. *rand()*는 복합 이벤트의 초기값과 최대값의 범위에서 무작위로 수를 추출한다. 또한 식 2는 각 복합 이벤트를 구성하는 기본 이벤트 수(*PE*)를 나타낸 것으로 복합 이벤트 수와 같이 일정한 범위에서 무작위로 추출한 수 있도록 정의 된다. P_{Max} 는 각 복합 이벤트를 구성하는 기본 이벤트의 최대값을 나타내며 P_{Min} 은 각 복합 이벤트를 구성하는 기본 이벤트의 초기값을 나타낸다.

$$C = (rand() \% (Max - Min) + Min) \tag{1}$$

$$PE = (rand() \% (P_{Max} - P_{Min}) + P_{Min}) \tag{2}$$

제안하는 복합 이벤트 검출 알고리즘은 비트맵 인덱

스를 통해 복합 이벤트를 구성하는 모든 기본 이벤트의 발생 유·무를 관리한다. 따라서 평균 처리 시간은 비트맵 인덱스를 검색하는 시간을 포함한다. 식 3은 비트맵 인덱스를 검색하는데 걸리는 시간(T_{search})을 나타낸다. $K=1$ 인 경우는 복합 이벤트를 구성하는 모든 기본 이벤트가 발생한 경우를 나타내며 $K=0$ 인 경우는 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하지 않는 경우를 나타낸다. $B_{time}[k]$ 는 비트맵 인덱스에서 k 번째 기본 이벤트를 검색하는데 걸리는 시간을 나타내며, 1에서 PE 의 범위에서 발생한 기본 이벤트의 검색 시간의 합이 비트맵 인덱스를 검색하는 전체 시간이 된다.

$$T_{search} = \begin{cases} K=1, & \sum_{1 \leq k \leq PE} B_{time}[k] \\ K=0, & 0 \end{cases} \quad (3)$$

1개의 복합 이벤트를 처리하는 시간(C_{time})은 식 4와 같이 비트맵 인덱스를 검색하는 시간(T_{search})과 1개의 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하였을 경우 기본 이벤트를 처리하는 시간($P_{time}[k]$)의 합으로 나타낼 수 있다. 기존 알고리즘과의 차이점은 비트맵 인덱스를 검색하는 걸리는 시간과 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하는 경우만 복합 이벤트 검출 연산을 수행하는 것에 차이점이 있다.

$$C_{time} = T_{search} + \begin{cases} K=1, & \sum_{1 \leq k \leq PE} P_{time}[k] \\ K=0, & 0 \end{cases} \quad (4)$$

전체 등록된 복합 이벤트를 검출하는데 걸리는 평균 처리 시간($C_{totaltime}$)은 식 4에 의해 전체 등록된 복합 이벤트를 처리하는데 걸리는 시간을 등록된 복합 이벤트 수로 나누어서 식 5와 같이 계산 할 수 있다. 여기서 Min 은 등록된 복합 이벤트의 최소값이며 C 는 등록된 복합 이벤트 수를 나타낸다.

$$C_{totaltime} = \frac{\sum_{Min \leq i \leq C} C_{time}[i]}{C} \quad (5)$$

초당 30,000개의 기본 이벤트가 발생하는 상황에서 복합 이벤트의 수를 100~500개로 변경시켜 등록하고 복합 이벤트 검출을 위한 평균 처리 시간을 측정한다. 또한 복합 이벤트를 500개로 고정한 후 기본 이벤트를 10,000~50,000개로 변경 시키며 실험을 반복 수행한다.

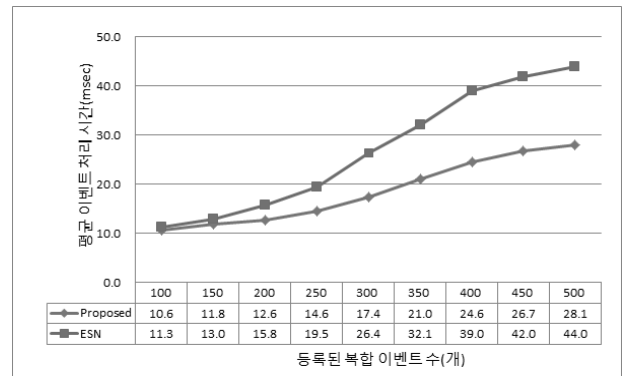


그림 18. 복합 이벤트 수에 따른 평균 처리 시간
Fig. 18. Average processing time according to the number of complex event.

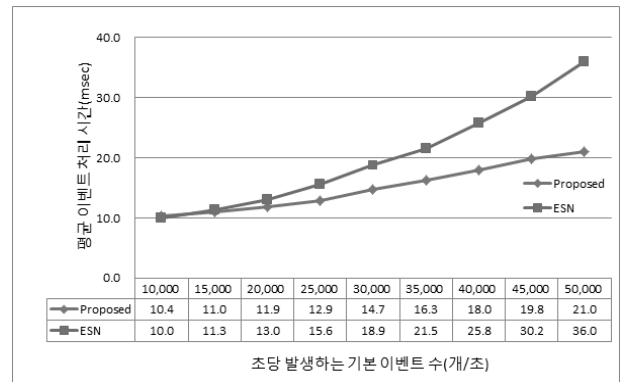


그림 19. 기본 이벤트 수에 따른 평균 처리 시간
Fig. 19. Average processing time according to the number of primitive event.

그림 18은 등록된 복합 이벤트에 따른 평균 처리 시간을 나타낸다. 기존 알고리즘은 복합 이벤트에 따라 처리 시간이 제안한 알고리즘에 비해 비교적 크게 증가하는 것을 볼 수 있다. 반면 제안하는 알고리즘은 복합 이벤트에 따라 평균 처리 시간이 기본 알고리즘에 비해 크게 증가하지 않는다. 이유는 복합 이벤트가 증가함에 따라 기존 알고리즘에서 처리해야 하는 복합 이벤트의 수는 크게 증가하나 제안하는 알고리즘은 증가율이 적다.

그림 19는 기본 이벤트의 변화에 따른 복합 이벤트의 평균 처리 시간을 나타낸다. 이번 실험 역시 제안하는 알고리즘을 통해 조건을 만족하는 복합 이벤트의 수가 현저히 줄기 때문에 기본 이벤트를 증가시켜도 제안하는 알고리즘의 처리시간은 변화가 크게 증가하지 않음을 보여준다. 하지만 기존 알고리즘에서는 기본 이벤트가 증가함에 따라 복합 이벤트를 처리하는 시간이 제안하는 알고리즘에 비해 상대적으로 증가함을 볼

수 있다.

(2) 복합 이벤트 검출을 위한 연산 수

연산 수는 등록된 복합 이벤트를 검출하는데 발생한 연산의 수를 나타내며, 각 복합 이벤트를 구성하는 연산자 수에 따라 발생한 연산 수를 나타낼 수 있다. 식 6은 각 복합 이벤트를 구성하는 연산자 수(PO_{num})를 나타내는 식으로 PO_{Max} 는 각 복합 이벤트를 구성하는 연산자의 최대값을 나타내며, PO_{Min} 는 각 복합 이벤트를 구성하는 연산자의 초기값을 나타낸다. $rand()$ 는 각 복합 이벤트를 구성하는 연산자의 초기값과 최대값의 범위에서 무작위로 수를 추출한다. 따라서 모든 복합 이벤트 검출을 위해 사용되는 연산 수($PO_{totalnum}$)는 식 7과 같다.

$$PO_{num} = (rand())\%(PO_{Max} - PO_{Min}) + PO_{Min} \quad (6)$$

$$PO_{totalnum} = \begin{cases} K=1, & 1 \leq k \leq C \\ K=0, & 0 \end{cases} \sum_{k=1}^C PO_{num}.[k] \quad (7)$$

등록된 복합 이벤트의 수에 따라 초당 이벤트를 처리하는 평균 연산 수를 측정한다. 실험은 전과 동일한 환경에서 실시한다. 그림 20은 복합 이벤트의 변화에 따라 실험 평가한 결과이다. 초당 30,000개의 기본 이벤트가 발생할 때 제안하는 알고리즘은 기존에 비하여 복합 이벤트 검출에 필요한 연산 수가 적음을 알 수 있다. 즉, 이벤트의 복잡도가 다양하면 본 논문에서 제안하는 알고리즘이 처리하는 복합 이벤트 검출 연산의 수가 현저히 감소함을 볼 수 있다.

그림 21은 기본 이벤트를 변화 시켜가며 제안하는 알

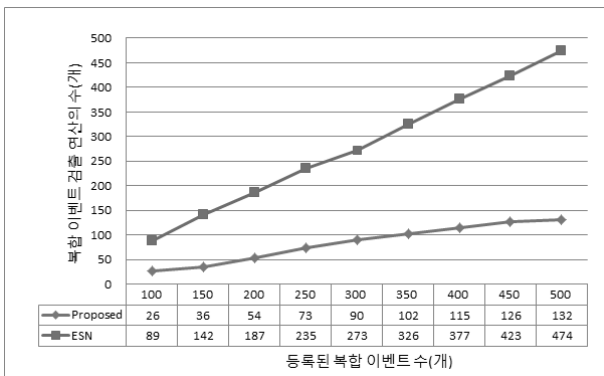


그림 20. 복합 이벤트 수에 따른 연산 수
Fig. 20. Operations according to the number of complex event.

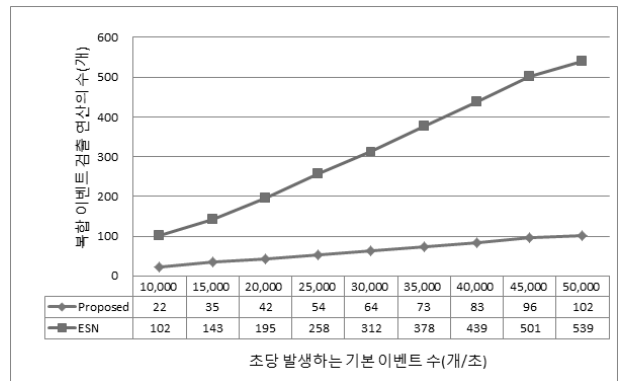


그림 21. 기본 이벤트 수에 따른 연산 수
Fig. 21. Operations according to the number of primitive event.

고리즘과 기존알고리즘을 비교한다. 기본 이벤트가 증가함에 따라 기존과 제안하는 알고리즘 모두 초당 발생하는 이벤트에 따라 증가하는 모습을 볼 수 있다. 하지만 기본 이벤트의 증가에 따라 제안하는 알고리즘이 기존 알고리즘보다 증가량이 현저히 적음을 알 수 있다. 기존 알고리즘은 증가된 기본 이벤트에 따라 대응하는 복합 이벤트의 수가 증가되기 때문이다. 하지만 제안하는 알고리즘은 기본 이벤트의 수가 증가하여도 복합 이벤트의 최소 조건을 만족하는 복합 이벤트의 수는 크게 증가하지 않는다.

(3) 복합 이벤트 검출을 위한 저장 비용

저장 비용이란 복합 이벤트를 검출하는데 필요한 메모리 용량을 나타내는 것으로 복합 이벤트 처리를 위해 등록된 복합 이벤트는 이진 트리 형태로 등록된다. 따라서 각 복합 이벤트 검출을 위한 저장 비용(C_{size})은 이진 트리의 공간 복잡도(Space Complexity)도 공식을 이용하여 식 8과 같이 계산한다. 여기서 $2^{T_{level}}$ 은 트리 레벨에 따른 전체 노드 수를 나타낸 것으로 복합 이벤트를 구성하는 기본 이벤트와 연산자를 나타낸다. 여기에 -1을 통해 나타난 값은 기본 이벤트를 제외한 상위 연산자수를 나타낸다. 따라서, 복합 이벤트 검출을 위한 저장 비용($C_{totalsize}$)은 식 9와 같이 각 복합 이벤트의 저장 비용을 전체 합의 값으로 나타낼 수 있다. 기존 알고리즘은 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하지 않아도 부분적으로 발생하는 기본 이벤트를 처리하기 위해 계속적으로 저장하여 처리하는 반면, 제안하는 알고리즘은 복합 이벤트를 구성하는 모든 기본 이벤트가 발생하여야지만 저장하여 처리하게 된다.

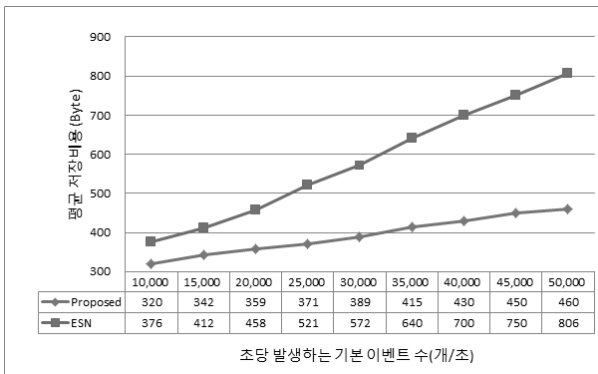


그림 22. 기본 이벤트 수에 따른 저장 비용
Fig. 22. Cost according to the number of primitive event.

$$C_{size} = 2^{T_{level}} - 1 \quad (8)$$

$$C_{totalsize} = \begin{cases} K=1, & 1 \leq \sum_{k \leq C} C_{size}[k] \\ K=0, & 0 \end{cases} \quad (9)$$

그림 22는 발생하는 기본 이벤트 수에 따른 평균 저장비용을 나타낸다. 제안하는 알고리즘은 복합 이벤트를 구성하는 모든 기본 이벤트가 발생할 때만 수행하므로 저장 비용역시 기존 알고리즘에 비해 적음을 알 수 있다.

(4) UUA 구현

제안하는 UUA는 제한된 능력을 가지고 있는 RFID 태그 및 센서 노드를 대신하여 SIP 시그널링을 담당하는 개체이다. 그림 23과 같이 수신된 USN 데이터에서

RFID				
Company	Object	Serial	Reader ID	Timestamp
0000C56	00016C	000169DC0	F409	2011-02-06-15:00:24
0000A89	00018A	000876AC8	F407	2011-02-06-15:00:25
0000A89	00034A	000457FA0	F407	2011-02-06-15:01:03
0000B59	00015B	000128AB4	F407	2011-02-06-15:01:16
0000B59	00091B	000894CB9	F409	2011-02-06-15:01:18
0000A89	00052A	000587DA5	F407	2011-02-06-15:02:01
0000C56	00037C	000567BC4	F407	2011-02-06-15:02:34
0000A89	00011A	000498BA6	F409	2011-02-06-15:02:39

WSN				
OUI	Serial	Sensing Data	Group ID	Timestamp
0050C1	37C0010003	000169DC0	2000	2011-02-06-15:05:54
0050B1	37B0010006	000876AC8	2000	2011-02-06-15:07:12
0050B1	37B0010002	000457FA0	3000	2011-02-06-15:08:55
0050A1	37A0010015	000128AB4	4000	2011-02-06-15:13:42
0050C1	37C0010008	000894CB9	2000	2011-02-06-15:15:56
0050B1	37B0010009	000587DA5	3000	2011-02-06-15:19:06
0050D1	37D0010007	000567BD4	4000	2011-02-06-15:20:38
0050A1	37A0010005	000498BA6	2000	2011-02-06-15:21:19

그림 23. UUA 구현 화면
Fig. 23. UUA screen implementation.

우선 RFID 태그와 센서 노드를 위한 SIP URI가 필요하다. 따라서 UUA는 UNS를 참조함으로써 SIP URI를 알아내게 된다. 또한 UUA는 RFID 리더와 싱크 노드 수준으로 객체를 관리한다. 따라서 UUA는 USN 데이터에서 RFID 리더 ID, 센서 네트워크의 싱크 노드 그룹 ID 정보를 관리한다.

ID 정보는 추적 과정에서 추적 노드가 RFID 태그 및 센서 노드의 현재 위치를 알고 싶어할 때 추적 노드는 태그의 현재 위치를 관리하고 있는 UUA에게 INVITE 메시지를 보내게 되고, 이를 받은 UUA는 이에 대한 응답으로 태그의 현재 위치를 알려주게 된다. 다시 말하면, 만약 추적 노드가 관심 있는 태그의 특정 UUA 관리 아래에 있는 RFID 리더 및 센서 노드의 근처에 있는 경우, 그 UUA는 추적 노드에게 리더의 ID 주소 및 싱크 노드의 그룹 ID를 200 OK 메시지의 contact 필드에 담아 보냄으로써 태그의 현재 위치를 알려주게 된다. 이 200 OK 메시지를 받은 추적 노드는 contact 필드를 분석함으로써 태그의 현재 위치를 알아낼 수 있게 된다.

(5) UNS 구현

UNS는 EPC 및 MAC 주소와 SIP URI 사이의 사상을 저장하고 있는 분산 데이터베이스로 정의할 수 있다. UNS는 인터넷의 DNS나 EPCglobal Network의 ONS와 비슷한 조회 서비스를 제공한다. 즉, UNS는 EPC 및 MAC 주소를 입력받아서 SIP URI를 돌려주는 기능을 수행한다. 그림 24는 UNS의 구조를 나타낸 것으로 RFID 태그의 EPC와 센서 노드의 MAC 주소를

RFID EPC Mapping to SIP URI						
Company	Object	Serial	SIP-User	@	SIP-Domain	Timestamp
0000C56	00016C	000169DC0	00016C000169DC0	@	0000C56	2011-02-06-15:36:12
0000A89	00018A	000876AC8	00018A000876AC8	@	0000A89	2011-02-06-15:37:22
0000A89	00034A	000457FA0	00034A000457FA0	@	0000A89	2011-02-06-15:39:05
0000B59	00015B	000128AB4	00015B000128AB4	@	0000B59	2011-02-06-15:41:12
0000B59	00091B	000894CB9	00091B000894CB9	@	0000B59	2011-02-06-15:42:31
0000A89	00052A	000587DA5	00052A000587DA5	@	0000A89	2011-02-06-15:44:39
0000C56	00037C	000567BC4	00037C000567BC4	@	0000C56	2011-02-06-15:44:41
0000A89	00011A	000498BA6	00011A000498BA6	@	0000A89	2011-02-06-15:46:11

WSN Sensor ID Mapping to SIP URI					
OUI	Serial	SIP-User	@	SIP-Domain	Timestamp
0050C1	37C0010003	0050C1	@	37C0010003	2011-02-06-15:32:54
0050B1	37B0010006	0050B1	@	37B0010006	2011-02-06-15:35:32
0050B1	37B0010002	0050B1	@	37B0010002	2011-02-06-15:46:12
0050A1	37A0010015	0050A1	@	37A0010015	2011-02-06-15:57:42
0050C1	37C0010008	0050C1	@	37C0010008	2011-02-06-16:02:37
0050B1	37B0010009	0050B1	@	37B0010009	2011-02-06-16:13:29
0050D1	37D0010007	0050D1	@	37D0010007	2011-02-06-16:27:38
0050A1	37A0010005	0050A1	@	37A0010005	2011-02-06-16:58:19

그림 24. UNS 구현 화면
Fig. 24. UNS screen implementation.

SIP URI 형태로 변환하여 저장된 화면을 나타낸다.

UUA나 추적노드는 자신들이 관심 있는 태그의 EPC 및 센서 노드의 MAC 주소를 UNS에게 보내서 위치 갱신이나 추적 과정에서 사용하게 될 SIP URI를 얻게 된다. 즉, UNS는 자신이 받은 전자상품코드에 대한 SIP URI를 생성하거나 캐쉬된 URI를 찾아내서 그것을 UUA나 추적 노드에게 돌려주는 역할을 수행한다.

V. 비교 및 고찰

EPC 센서 네트워크와 제안하는 관리 시스템의 컴포넌트들에 대한 비교가 표 2에 나타나 있다. 첫 째로, 개체에 대한 식별자로 EPC 센서 네트워크는 EPCglobal Network를 기반으로 하기 때문에 EPC를 사용하며, 제안하는 시스템은 SIP 식별자를 통해 데이터를 전송한다. 두 번째로, 두 시스템 모두 ONS와 UNS라는 조회 서비스가 존재한다. EPC 센서 네트워크는 EPC-주소의 사상(mapping)을 알려주며, 제안하는 시스템은 EPC, MAC 주소의 사상을 알려주는 역할을 수행한다. 또한 두 서비스 모두 분산적인 방법으로 구현될 수 있다. 그러나 ONS는 EPC와 관련된 정보들을 관리하는 Extended EPCIS의 주소를 돌려주는 반면에 UNS에서는 RFID 태그와 센서 노드에 대응하는 SIP URI를 돌려준다. 세 번째로, 두 시스템 모두 미들웨어에서 USN 데이터를 통합하기 위해 복합 이벤트 처리 방식을 사용한다. 하지만 복합 이벤트를 검출하기 위한 알고리즘으로 EPC 센서 네트워크는 그래프 기반의 알고리즘을 사용하는 반면 제안하는 시스템은 비트맵 인덱스를 통해 이벤트를 검출한다. 네 번째로, 망 개방 여부이다. EPC 센서 네트워크는 자신만의 구성요소와 프로토콜로 다른 네트워크와의 연계가 어렵다. 하지만 제안하는 시스템

은 인터넷 표준 프로토콜인 SIP를 기반으로 하기 때문에 개방적인 망 구성이라 할 수 있다. 다섯 번째로, 두 시스템 모두 위치 추적이나 다른 목적을 위해 EPC 및 센서 정보와 관련된 정보를 저장할 필요가 있다. 이러한 목적을 위해서 EPC 센서 네트워크는 Extended EPCIS를 통해 저장되며, 제안하는 시스템에서는 Registrar를 통해 정보를 저장한다.

다음은 제안하는 관리 시스템과 EPC 센서 네트워크를 재사용성, 신장성, 확장성의 관점에서의 비교와 다른 분야와 관련지어 제안하는 관리 시스템의 장점을 설명한다.

첫째로, 제안하는 관리 시스템은 SIP 기반의 VoIP 네트워크와 같은 기존에 존재하는 네트워크 인프라 구조를 그대로 다시 사용할 수 있다. RFID 태그와 센서 노드를 관리하기 위해서 EPC 센서 네트워크는 자신만의 특별한 구조로 되어 있고, 특별한 컴포넌트를 필요로 한다. 이것은 인터넷 표준 프로토콜인 SIP에 기반하고 있는 제안하는 관리 시스템과는 다른 부분이다. 제안하는 관리 시스템은 SIP를 아무런 수정 없이 그대로 사용하고 있기 때문에, 기존에 SIP를 위해 설치된 SIP proxy/redirect 서버, registrar 등을 그대로 재사용할 수 있다. 최근 SIP는 VoIP 서비스나 IP 멀티미디어 서비스 시스템(IMS) 등에서 시그널링 프로토콜로서 널리 쓰이고 있다. 따라서 제안하는 관리 시스템을 기존의 인프라 구조와 통합하는 것이 가능하고, 이러한 재사용성은 시스템의 설치비용의 감소와 빠른 개발을 가능하게 한다.

두 번째로 제안하는 관리 시스템은 신장성이 뛰어나다. 제안하는 관리 시스템이 기반하고 있는 프로토콜인 SIP는 HTTP와 같이 텍스트를 기반으로 하는 프로토콜이다. 이는 응용 레벨이나 사용자 레벨에서 새로운 서비스를 추가하는 것을 용이하게 한다. EPC 센서 네트워크의 경우는 특별한 목적을 가진 미들웨어를 기반으로 한 솔루션으로 개발되었다. 그러므로 EPC 센서 네트워크는 웹이나 무선 인터넷 접속 등 다른 서비스와의 통합 및 연계가 어렵다. 반면에 SIP에 기반 한 관리 시스템은 쉽게 신장이 가능하고, 다른 인터넷 기반의 서비스와의 통합이 쉽다.

세 번째로 제안하는 관리 시스템은 확장성이 좋다. 제안하는 관리 시스템은 SIP 서버나 SNS를 관리하는 중앙 서버가 존재하지 않는다. proxy나 redirect 서버 등의 SIP 서버는 많은 도메인 상에 분산되어 분포하고 있고, 각각의 서버는 자신의 도메인 상에서 자신의 고

표 2. 컴포넌트 비교
Table 2. Comparison of component.

	EPC 센서 네트워크	제안하는 시스템
식별자	EPC	SIP URI
프로토콜	SOAP	SIP
이벤트 검출 알고리즘	오토마타	비트맵 인덱스
망 개방 여부	폐쇄적	개방적
네이밍 서비스	ONS	UNS
저장소	Extended EPCIS	Registrar

유한 기능을 수행하고 있다. 또한 UNS는 DNS와 마찬가지로 분산적인 방식으로 구현될 수 있다. 제안하는 관리 시스템에서의 EPC와 센서 노드의 MAC 주소 조회는 DNS에서의 질의와 유사하게 처리된다. 즉, EPC와 센서 노드의 MAC 주소 조회는 우선 로컬 UNS에서 처리되고, 만약 로컬 UNS가 이를 해결하지 못하는 경우에는 UNS 계층에서 좀 더 위에 있는 UNS가 이 요청을 처리하게 된다. 반면에 ONS는 좀 더 중앙 집중적인 방식으로 동작한다. 비록 ONS가 루트 ONS와 로컬 ONS라는 두 개의 계층으로 구성되어 있지만, ONS 조회에 있어서 시작점이 되는 것은 루트 ONS이다. 다시 말하면, 모든 EPC 조회 서비스는 루트 ONS에게 질의를 보냄으로써 시작된다. 이는 잠재적으로 루트 ONS를 성능저하의 병목지점으로 만들 것이다.

네 번째 제안하는 관리 시스템의 장점은 비용 측면에서 찾을 수 있다. RFID 태그 및 센서 노드는 개체의 자동식별 및 환경 정보 수집을 가능하게 하므로 이를 공급망 관리(Supply Chain Management, SCM)에 활용하게 되면 현재 기업환경에서 RFID 만을 적용한 관리 시스템에서 USN 을 대상으로 한 포괄적인 관리의 효율성 및 관리 비용의 절감의 효과를 가져 올 것이다. 이를 위해서 EPCglobal에서는 EPC 센서 네트워크를 제시하였다. 그러나 이는 상당히 고가의 컴포넌트 및 미들웨어이므로 USN 관리 시스템을 사용하기 위해서 EPCglobal에 많은 돈을 로열티로 지불해야 할 것이다. 그러나 제안하는 관리 시스템은 인터넷 표준에 기반한 개방형 구조이므로 EPC 센서 네트워크와 같은 기능을 제공하면서도 저렴한 가격으로 구현될 수 있다. 만약 제안하는 관리 시스템을 공급망 관리에 사용한다면 EPCglobal에 지불하는 비용을 절약할 수 있을 것이다.

다섯 번째 장점은 인프라 및 서비스 분야에 대한 파급효과이다. 제안하는 관리 시스템은 RFID 태그 및 센서 노드의 위치를 관리해주는 시스템으로 USN을 활용한 서비스로 분류할 수 있다. 즉, 제안하는 관리 시스템의 구축은 USN에 대한 수요를 발생시키고 결과적으로 USN 산업의 발전을 촉진시킬 것이다. 또한 제안하는 관리 시스템은 VoIP 서비스, IMS 서비스와 함께 시너지 효과를 발휘하면서 성장할 수 있다. 이는 첫 번째에서 제시한 재사용성과 밀접한 관련이 있다. 제안하는 관리 시스템을 구축하기 위해 설치한 SIP 서버는 VoIP 서비스를 제공하거나 IMS를 구축하는데 사용될 수 있다. 반대로 VoIP 서비스나 IMS를 위해 설치된 SIP 서

버를 제안하는 관리 시스템을 구축하기 위해 사용할 수 있다. 즉, 제안하는 관리 시스템, VoIP, IMS 중 어느 한 시스템의 구축은 나머지 다른 시스템의 구축을 쉽고, 빠르고 저렴하게 만드는 긍정적인 영향을 미칠 것으로 기대 된다.

VI. 결 론

최근 국·내외 다양한 환경에서 RFID와 WSN을 이용한 많은 연구가 진행되고 있다. 하지만 RFID와 WSN은 그 기술적인 발달에도 불구하고 표준의 부재와 현재의 기술적 한계 때문에 데이터의 공유와 융합을 위해 이를 지원할 수 있는 인프라스트럭처가 필요하게 되었다.

본 논문에서는 RFID와 WSN 데이터를 인터넷을 기반으로 한 통합 관리 시스템을 제안하였다. 제안하는 관리 시스템은 효율적인 이동성을 지원하는 인터넷 표준 프로토콜인 SIP에 기반하고 있다. SIP 기반의 USN 통합 관리 시스템을 위해 첫 번째, 대용량 RFID와 WSN 데이터를 단일 환경에서 처리하기 위해 미들웨어에서 비트맵 인덱스 기반의 복합 이벤트 검출 알고리즘을 제안하였다. 제안하는 검출 알고리즘은 복합 이벤트를 구성하는 기본 이벤트의 발생 유·무 판별하기 위해 비트맵 인덱스로 관리하고 복합 이벤트를 구성하는 기본 이벤트를 이벤트 질의를 통해 트리 구조로 등록하여 관리한다. 복합 이벤트를 구성하는 모든 기본 이벤트가 비트맵 인덱스를 통해 발생하면 이벤트 질의를 통해 복합 이벤트 검출을 위한 연산을 수행한다. 제안하는 복합 이벤트 검출 알고리즘은 모든 기본 이벤트가 발생할 때만 검출 연산을 수행하기 때문에 연산의 수를 감소시킬 수 있고 불필요한 후보 집합을 생성하지 않아 복합 이벤트 검출을 위한 처리 시간이 줄어들게 된다. 실험을 통해 제안하는 알고리즘이 기존의 EPC 센서 네트워크에서 제안하는 검출 알고리즘에 비해 초당 발생하는 기본 이벤트 수와 복합 이벤트 등록 수에 따라 평균 처리 시간은 6.8msec, 검출에 필요한 연산 수는 평균 226개로 기존 알고리즘에 비해 적게 나타나 우수한 성능을 나타내었다. 이를 통해 USN 환경에서 제안하는 알고리즘이 기존 알고리즘 보다 평균 처리 시간 및 검출을 위한 연산 수에서 효과적으로 복합 이벤트 검출을 수행함을 확인할 수 있었다. 두 번째, RFID 태그 및 센서 노드를 대신하여 SIP와 관련된 메시지를 처리하기 위해 UUA를 제안하였다. 또한 EPC 및 센서 노드의 MAC

주소와 SIP URI 사이의 글로벌 네이밍 서비스를 제공하기 위해 UNS라는 네이밍 서비스를 제안하였다. EPCglobal에서 제안하는 RFID와 WSN 관리 시스템인 EPC 센서 네트워크와 비교했을 때, 제안하는 관리 시스템은 재사용성, 신장성, 확장성 등의 장점을 가지고 있다. 이는 제안하는 관리 시스템이 SIP를 사용한 분산된 구조를 가지고 있기 때문이다.

참 고 문 헌

- [1] 이기욱, 성창규, “유비쿼터스 센서 네트워크 기반의 상황 정보 모니터링 시스템 구현”, 한국컴퓨터정보학회논문지, 제11권, 제5호, 2006. 8.
- [2] R.Want. “Enabling ubiquitous sensing with RFID,” Computer IEEE , vol.37, pp. 84-86, April. 2004.
- [3] 오지성, 이정욱, 진정우, 노재정, “EPC 센서 네트워크-based Product and Process Traceability System in the Food Supply Chain,” 한국경영정보학회 International Conference, 2007. 3.
- [4] 임명환, 박용재, “RFID/USN 서비스 시장 전망 및 적용 사례 분석,” 한국전자과학회, 제17권 제3호, pp. 3-12, 2008. 3.
- [5] 표철식, 채종석, “차세대 RFID/USN 기술 발전 전망,” 한국통신학회:정보통신, 제24권 제8호, pp. 7-13, 2007. 8.
- [6] 성종우, 김대영, “RFID와 USN 통합 인프라스트럭처를 위한 EPC 센서 네트워크,” 한국통신학회, 제23권, 제12호, pp. 27-46, 2007. 12.
- [7] 이경대, 임장관, 성종우, 김성훈, 하민근, 김대영, “EPC 센서 네트워크에서의 데이터 처리와 정보 공유를 위한 미들웨어,” 한국정보과학회 2010 학술발표논문지, 제37권, 제2호, pp.19-22, 2010. 2.
- [8] 정수호, 김규백, 김성훈, 김영주, 김대영, “EPC 센서 네트워크를 위한 ZigBee Reader 설계 및 구현,” 한국정보과학회 2010 학술발표논문지, 제37권, 제2호, pp.28-31, 2010. 2.
- [9] Guangqian Zhang, Li Zhang, “Study of CEP-Based RFID Data Processing Model,” Computer society IEEE, pp. 185-193, Feb. 2008.
- [10] Louis Perrochon, “Enlisting Event Patterns for Cyber Battlefield Awareness,” DARPA Information Survivability Conference & Expositoin, Vol. 2, pp.1411-1420, Feb. 2000.
- [11] D. Gyllstrom, E. Wu, H. J. Chae, Y. Diao, P. Stahlberg, and G. Anderson SASE: Complex Event Processing over Streams. In Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR 2007)
- [12] F. Wang, S. Liu, and Y. Bai, “Bridging Physical and Virtual Worlds: Complex Event Processing for RFID Data stream”, Proc International Conference on Extending Database Technology, 2006.
- [13] J. Rosenberg et al, “SIP: Session Initiation Protocol,” RFC 3261, IETF, June. 2002.
- [14] G. Camarilli, M. Garcia-Martin, “The 3G IP Multimedia Subsystem(IMS) Merging the Internet and the Cellular Worlds,” John Wiley & Sons, August 2004.
- [15] 박성진, “EPC SENSOR NETWORK 기반의 효과적인 응용-애플리케이션 개발을 위한 프레임워크 설계 및 구현,” 대한산업공학회 춘계학술대회논문집, pp. 14-16, 2009.
- [16] Weixin Wang, Jongwoo Sung, Daeyoung Kim, “Complex Event Processing in EPC Sensor Network Middleware for Both RFID & WSN,” Object Oriented Real-Time Distributed Computing, 2008 11th IEEE International Symposium on, pp. 165-169, May 2008.
- [17] Sanchez Lopez, T., D. Kim, “A Context Middleware Based on Sensor and RFID Information,” Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on, pp. 331-336, March. 2007.
- [18] Brenda M. Michelson, “Event-Driven Architecture Overview”, Patricia Seybold Group, Feb. 2007.
- [19] 김종익, “동적 데이터 흐름 처리를 위한 이벤트 스트림 관리 기술 개발,” 교육과학기술부 기초연구사업, pp. 23-26, 2009. 3.

저 자 소 개

박 용 민(정회원)

대한전자공학회 논문지

제42권 TC편 제10호 참조



이 준 혁(정회원)

1993년 동의대학교 전자통신
공학과 학사졸업

1997년 광운대학교 전자통신
공학과 석사졸업

2009년 광운대학교 전자통신
공학과 박사졸업

<주관심분야 : 신뢰성, 정보통신, 의료정보공학>