

센서 태그 데이터의 필터링을 위한 Edge Manager의 설계

이준호⁰ 류우석 홍봉희
부산대학교 컴퓨터공학과
{junos52⁰, wsryu, bhhong}@pusan.ac.kr

Design of Edge Manager for Filtering Sensor Tag Data

Junho Lee, Wooseok Ryu, Bonghee Hong
Department of Computer Engineering, Pusan National University

요 약

RFID 기술은 기존의 바코드 기술보다 발전된 무선의 비접촉 인식 기술로서, 유비쿼터스 컴퓨팅의 핵심 기술로 간주되고 있다. RFID 태그는 기존의 단순한 사물의 인식 기능 중심에서 사물의 상태 및 환경 정보를 감지할 수 있는 센서 태그로의 발전으로 물류 프로세스의 양적, 질적 향상을 도모할 수 있게 되었다. 센서 태그는 환경 정보를 센싱하기 위하여 배터리를 내장하고 있으며, RF Transceiver의 내장 유무에 따라 자체적으로 신호를 보낼 수 있는 active 센서 태그와 리더로부터의 신호를 사용하는 semi-passive 센서 태그로 구분된다. Semi-passive 센서 태그는 배터리를 부착함으로써 passive 태그에 비해 인식률과 인식거리가 향상되었고 active 태그에 비해 단가가 매우 저렴하여 센서 태그가 부착된 물품의 상태를 모니터링하고 환경을 감시하는 다양한 응용에 사용될 수 있다. 이러한 응용의 요구에 따라 Edge Manager는 기존의 passive 태그는 물론 센서 태그를 지원함으로써 상위 응용에게 정제된 결과를 전달할 필요성이 있다.

본 논문에서는 특히 semi-passive 센서 태그를 지원하는 Edge Manager의 설계를 위하여, 센서를 사용한 또 다른 활용 분야인 센서 네트워크에서의 질의 유형을 분석하고, semi-passive 센서 태그의 특징을 고려한 요구사항을 분석한다. Semi-passive 센서 태그는 센서 네트워크의 센서 노드와는 달리 태그 레벨에서 필터링과 병합을 수행할 수 없으므로 Edge Manager에서 이러한 기능이 제공되어야 한다. 본 논문에서는 Edge Manager에서의 센서 태그 데이터에 대한 질의를 위한 방법으로 EPCglobal ALE 표준 명세의 ECSpec을 확장하는 방법을 제안하고, 센서 태그 데이터의 특성을 고려한 필터링 기법과 병합(aggregation) 기법을 적용한 질의 처리가 가능한 Edge Manager의 구조를 제시한다.

1. 서 론⁺

RFID(Radio Frequency Identification) 기술은 유통, 물류 환경에서 기존의 바코드 기술보다 발전된 무선의 비접촉 인식 기술과 네트워크화된 관리 메커니즘을 제공함으로써 유비쿼터스 컴퓨팅의 핵심 기술로 간주되고 있다. Edge Manager는 리더와 가까운 Edge 단에서 다양한 물리적 리더들을 지원하고, 이들로부터 끊임 없이 발생하는 스트림 형태의 데이터를 필터링하고 수집하기 위한 역할이 필수적이다. RFID 관련 기술에 대한 사실상의(de facto) 표준을 제정하고 있는 EPCglobal은 이러한 기능의 제공을 위해 EPC Network Architecture Framework[1]를 제시하였고 스트림 데이터의 필터링과 수집을 담당하는 구성요소로서 'Application Level Events (ALE)' 인터페이스를 정의하였다[2].

한편, RFID 태그는 사물의 고유한 ID를 읽고 식별하는 기존의 단순한 기능에서 최근에는 태그가 부착된 사물의 환경 정

보(온도, 습도, 압력, 조도 등)를 감지할 수 있는 기능으로 발전하고 있다. 이와 같이 사물의 ID 획득과 환경 정보의 센싱이 가능한 RFID 태그를 '센서 태그'라고 한다. 센서 태그는 인식률 개선과 인식거리 확대, 센싱을 위해서 배터리를 가지고 있는 것이 특징이다. 센서 태그는 RF Transceiver의 내장 유무에 따라 자체적으로 신호를 보낼 수 있는 active 센서 태그와 리더로부터의 신호를 사용하는 semi-passive 센서 태그로 구분된다. 특히 semi-passive 센서 태그는 기존의 passive 태그에 배터리를 부착함으로써 인식률과 인식거리가 향상되었고 active 태그에 비해 단가가 매우 저렴하여 물류과정에서 혈액, 식의약품과 같이 환경에 민감한 물품의 상태를 모니터링하거나 환경을 감시하는 다양한 응용에 사용될 수 있다.

응용이 다양해짐에 따라 센서 태그의 요구가 증대되고 있는 상황에서 기존의 RFID 태그와 센서 태그를 모두 지원하는 시스템의 필요성이 높아지고 있지만 센서 태그의 지원을 위한 표준의 지원은 미비한 실정이고 각 시스템을 따로 운용해야 하는 것이 현실이다. 따라서 본 논문에서는 기존의 passive 태그를 위한 Edge Manager 시스템에서 센서 태그를 지원하

⁺ “이 논문은 2007년 정부(교육인적자원부)의 재원으로 한국 학술진흥재단의 지원을 받아 수행된 연구임”(지방연구중심대학육성사업/차세대물류IT기술연구사업단)

는, 그 중에서도 passive 태그와 유사한 semi-passive 태그의 지원이 가능한 Edge Manager 시스템의 아키텍처를 제안한다.

Edge Manager에서 semi-passive 센서 태그를 지원하기 위한 요구사항으로써, 본 논문에서는 센서 네트워크에서의 질의 유형을 분석하고 semi-passive 센서 태그의 특성을 고려하여 Edge Manager에서의 센서 태그 데이터에 대한 질의 표현 방법을 제안한다. 그리고 제안된 질의 표현 방법을 바탕으로 센서 태그 데이터를 필터링하여 사용자에게 병합된 결과를 제공할 수 있는 Edge Manager의 구조를 설계한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하며, 3장에서는 센서 태그를 적용하기 위한 요구사항을 분석하고 그에 따른 문제점을 정의한다. 4장에서는 Edge Manager에서 센서 태그 데이터에 대한 질의를 표현하는 방법을 제시하고, 5장에서 센서 태그를 지원하기 위한 Edge Manager 구조의 요구사항을 기술하며, 6장에서는 이를 종합하여 센서 태그 특성을 고려한 Edge Manager의 구조에 대해서 기술한다. 마지막으로 7장에서는 결론 및 향후 연구를 기술한다.

2. 관련 연구

이 장에서는 센서를 활용한 분야으로써 일반적인 센서 네트워크의 질의 유형과 상용 미들웨어로써 오라클의 Sensor Edge Server의 질의 처리 방식에 대해 기술한다.

2.1 센서 네트워크의 질의 유형

센서 네트워크는 다수의 '센서 노드'들 간의 멀티 홉 라우팅 방식으로 구성되는 네트워크이다[3]. 센서 네트워크에서는 리더가 필요 없으므로 질의 처리 방식은 RFID 환경과 크게 다르나, 환경 정보를 획득한다는 점에서 센서 태그와 유사하므로 센서 네트워크에서 사용되는 질의를 기술한다.

```

SELECT      {attributes, aggregates}
FROM        {Sensordata S}
WHERE       {predicate}
GROUP BY   {attributes}
HAVING     {predicate}
DURATION   time interval
EVERY      time span e
    
```

그림 1 Query Template

센서 네트워크에서는 그림 1와 같이 SQL에 기반한 선언적 질의를 사용한다[3]. 일반적으로 센서 네트워크에서는 환경 감시의 목적으로 오랜 시간 동안 주기적으로 정보를 획득하는 것이 필요할 수 있으므로 이를 위해 DURATION과 EVERY 절이 추가된 것이 특징이다. 이를 바탕으로 유지는 다음과 같은 종류의 질의를 사용할 수 있다[4].

- Historical Query : 과거에 네트워크로부터 축적된 데이터에 대한 일회성 질의
- Snapshot Query : 현재 시점의 데이터에 대한 일회성 질의
- Long-running Query : 일정 시간 간격 동안 주기적

으로 데이터를 획득하는 연속 질의

표 1은 센서 네트워크와 ALE 표준 명세에서의 질의와 aggregation 유형을 비교한 것이다. EPCglobal의 ALE 명세에서 제공되는 인터페이스에서도 이와 유사한 형태의 질의를 제공한다. Historical Query와 Snapshot Query는 일회성 질의로 ALE 인터페이스의 poll()과 immediate()로 대응될 수 있으며, Long-running Query는 연속 질의로 ALE 인터페이스의 subscribe()와 대응될 수 있다.

표 1 센서 네트워크와 ALE 표준 명세의 비교

	센서 네트워크	ALE 표준 명세
일회성 질의	Snapshot Query Historical Query	poll, immediate
연속 질의	Long-running Query	subscribe
질의 조건	highest/lowest =, >, <, ≥, ≤	value, *, [lo-hi], mask
Aggregation	AVG, COUNT, SUM MIN, MAX, MEDIAN	COUNT

센싱 정보는 측정된 수치들로 나타나므로 센서 네트워크에서의 질의 조건은 센서 노드에서 측정된 수치를 특정 임계값과 비교하여 만족하는 센서 노드의 데이터를 획득하거나 최대/최소값을 가지는 센서 노드의 데이터를 획득하는 것에 목적이 있지만, ALE 표준 명세에서는 특정 패턴과 비교하여 일치하는지 여부를 판단하여 태그 데이터를 필터링한다. 필터링된 데이터에 대해서 센서 네트워크에서는 평균값, 합, 중간값, 최대/최소값, 갯수를 구하는 병합 연산(aggregation)을 지원하지 않지만 ALE 표준 명세에서는 태그 population에 대한 COUNT만을 지원한다.

2.2 Oracle Sensor Edge Server

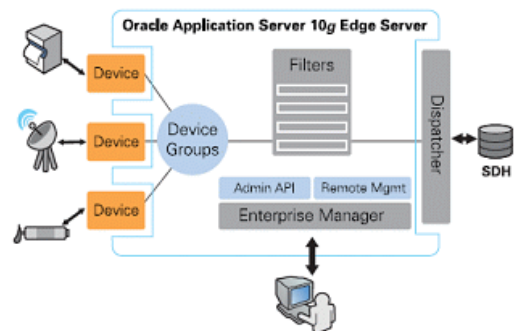


그림 2 Oracle Sensor Edge Server

오라클의 Sensor Edge Server(SES)는 물리적 환경에서 일어나는 이벤트의 수집을 담당하는 영역에 위치하는 오라클의 RFID Infrastructure의 한 구성요소이다[5]. SES의 아키텍처는 그림 2과 같이 다양한 센서와의 통신을 담당하는 Device

Driver, 수집된 데이터를 가공하는 Filter, 이를 적절한 저장소로 전송하는 Dispatcher, 그리고 관리 기능을 지원하는 Enterprise Manager로 구성된다.

SES에는 기본적으로 데이터를 가공하기 위한 몇 가지 필터가 내장되어 있다. ALE 표준 명세에도 정의되어 있는 리더에 대한 필터, 태그 ID에 대한 필터를 제공하지만 대부분의 필터는 데이터의 여과보다는 물리적 이벤트에 비즈니스 컨텍스트를 부여하는 가공에 목적을 두고 있다. 그림 3과 같이 내장 필터를 확장한 사용자 정의 필터에서도 여과 기능의 파라미터는 제공하지 않는다.



그림 3 내장 필터와 커스텀 필터

SES에서 수집되어 가공된 데이터는 SDR(Sensor Data Repository)에 저장되어 Enterprise Application이 사용할 수 있게 한다. 미들웨어로서의 SES는 데이터의 필터링 기능과 Aggregation 기능을 제공하지 않는다.

3. 센서 태그 질의 분석 및 문제 정의

3.1 센서 태그의 데이터 분석

센싱 정보가 센서 태그의 메모리에 기록되는 방식을 분석하기 위하여 Alien의 BAP(Battery Assisted Passive, ALB-2484) 태그의 메모리 구조를 분석하였다[6]. ALB-2484 태그는 온도를 센싱할 수 있는 태그로 태그 ID를 저장하기 위한 메모리라는 별도로 센싱 정보만을 저장하기 위하여 메모리(NVRAM, 4K)를 가지고 있으며 구조는 그림 4와 같이 선형적이다.

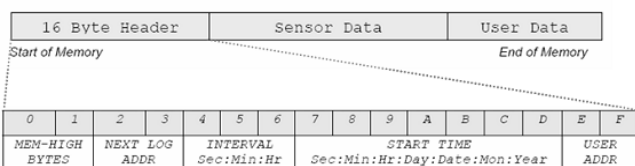


그림 4 ALB-2484 태그 메모리의 구조

이 메모리는 센싱 정보를 로깅하는 용도로 리더의 영역 밖에서 센싱 정보를 계속적으로 저장하기 위해서 사용된다. 리더 영역 내에 있는 경우에는 온도 센서로부터 센싱 정보를 직접 획득한다.

3.2 센서 태그 질의 특성

본 논문은 기존의 passive 태그의 ID뿐만 아니라, semi-passive 센서 태그의 ID와 센싱 정보에 대한 필터링과 병합을 지원하는 Edge Manager 시스템을 대상으로 하고 있다.

semi-passive 센서 태그(이하 센서 태그)의 지원을 위하여 고려해야 할 센서 태그의 몇 가지 특징은 다음과 같다. 첫째, 센서 태그는 태그의 ID와 센싱 정보를 동시에 가지고 있다. 이것은 센서 태그에 대해 질의를 할 때, 태그 ID 혹은 센싱 정보만을 고려하거나, 두 가지 모두를 고려할 수 있다는 것을 의미한다. 둘째, 센서 태그는 센서 노드와 달리 CPU와 OS를 가지고 있지 않으므로 센싱 정보를 태그 레벨이 아닌 미들웨어 또는 상위 애플리케이션 레벨에서 필터링하고 가공할 수 있어야 한다. 셋째, 센서 태그는 물류, 유통 환경에서 수송을 위해 리더의 인식 영역을 벗어날 수밖에 없는 경우가 빈번히 발생한다. 센서 태그가 다시 리더의 인식 영역 내에 들어왔을 때, 인식 영역 밖에 있는 동안의 센싱 정보에 대한 질의가 필요한데, 이 경우 Edge Manager는 Historical Query를 통해 이러한 센싱 정보를 모두 획득하고 병합해야 하므로 그림 5와 같이 다수의 센서 태그에 대한 병합뿐만 아니라, 하나의 센서 태그가 가지고 있는 전체 센싱 정보에 대한 두 번의 병합이 필요하다. 넷째, Historical Query를 사용하여 센서 태그의 메모리에 저장된 대량의 센싱 정보를 획득하기 위해서는 리더와 태그 사이에 여러 번의 데이터 전송이 이루어진다. 그러므로 센싱 정보의 완전한 전송을 보장하기 위해서는 부분적으로 전송되는 데이터를 임시로 저장하고 데이터 손실이 발생하는 경우를 관리할 수 있어야 한다.

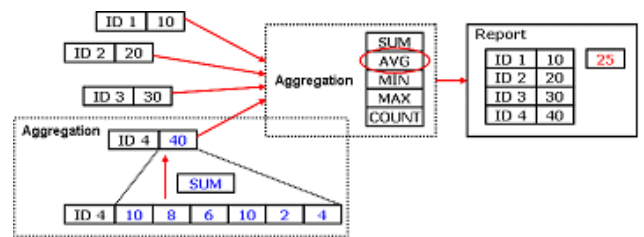


그림 5 센서 태그의 병합 연산

3.2 문제 정의

passive 태그를 지원하는 Edge Manager에서 센서 태그를 지원하기 위해서는 첫째, 센서 태그의 질의에 적합한 필터링 조건을 표현할 수 있어야 한다. ECSpec(Event Cycle Specification)은 passive 태그에 사용되는 질의로 ALE 표준 명세에 정의되어 있는데, 태그 데이터 수집을 위한 리더 조건(ECBoundarySpec), 사용자에게 보고될 리포트 형태를 정의한 조건(ECReportSpec) 내에 필터링을 위한 태그 ID 및 사용자 메모리의 패턴 조건(ECFilterSpec), 군집화를 위한 정보

(ECGroupSpec)등을 포함하고 있다. 그림 6은 EPCglobal에서 제시한 ECSpec의 예이다.

9.5.1 ECSpec: Door42PassThrough			
Description:		TODO: Description of why included	
Parameters:			
readers	DockDoor42		
includeSpecInReports			
ECBoundarySpec:			
startTrigger		stopTrigger	
repeatPeriod	50 MS	duration	10 MS
stableSetInterval			
ECReportSpec			
reportName	DeckDoorPassThrough	reportSet	ADDITIONS
reportIfEmpty	false	reportOnlyOnChange	true
output	MEMBERS		
filter	urn:epc:pat:gl:96:20,*		
group			

그림 6 ECSpec의 예

그러나 ECSpec은 센서 태그를 수집할 리더를 정의하는 논리적 리더 설정과 질의 수행을 위한 시간 조건의 표현은 가능하나, 센싱 정보의 필터링 조건에 해당하는 최대/최소값, 비교 연산자(=, >, <, ≤, ≥), 범위 등을 모두 표현할 수 없다. 또한 ECSpec에는 수집된 태그의 군집화(grouping)를 위한 조건을 표현할 수는 있지만 태그의 population에 대한 "COUNT" 병합 연산만 지원하고 있다. 센서 태그를 위해서는 태그의 population뿐만 아니라 센싱 정보에 대한 "SUM", "AVG", "MIN", "MAX", "COUNT", "MEDIAN"과 같은 다양한 병합 연산의 유형을 표현할 수 있어야 한다.

둘째, 센서 태그의 메모리에 저장된 대량의 센싱 정보를 전송함에 있어서 완전성을 보장해야 한다. Alien사의 ALB-2484 센서 태그의 경우 메모리의 크기는 4KBytes이고 리더와 태그 사이에 전송되는 패킷의 사이즈는 16Bytes이므로 전체 메모리에 저장되어 있는 센싱 정보를 획득하기 위해서는 여러 번의 전송이 필요하다. 리더와 태그의 정보 교환은 RF 통신으로 이루어지므로 태그가 리더의 인식 영역을 벗어나거나 통신상의 장애로 메모리를 읽는 과정에서 데이터의 손실이 발생할 수 있다. Historical Query는 태그 내 센싱 정보에 대한 병합 연산을 요구하므로 데이터의 손실이 발생하거나 전송이 완료되지 못한 경우 정확한 병합 연산의 결과를 기대하기 어렵다. 그러므로 센싱 정보의 완전한 전송을 보장하기 위하여 전송 과정에서 중간 결과를 저장하고 손실된 데이터에 대한 관리가 가능한 Edge Manager의 아키텍처 설계가 필요하다.

마지막으로 태그 ID에 대한 필터링과 함께 센싱 정보에 대한 태그 내 병합 연산과 다수의 태그간의 병합 연산을 효율적으로 수행할 수 있는 아키텍처 구조의 설계가 필요하다.

4. 센서 태그 데이터를 위한 필터링 조건의 표현

기존의 ECSpec으로는 센서 태그 데이터에 대한 필터링 조건을 표현할 수 없으므로 우선 ECSpec과 유사하면서 센서 태그를 위한 새로운 질의에 대한 명세(Specification)을 정의하는 방법을 생각해볼 수 있다. 그러나 새로운 명세에 대해 필요한 추가적인 API는 표준과 맞지 않는다는 제약이 있다. 또한 센서 태그에 대한 질의에 필요한 조건은 센싱 정보에 대한 필터링 조건, 보고 조건을 제외하고 ECSpec과 모두 동일하

로 두 명세간 구성 요소간의 중복이 발생한다. ECSpec은 태그 수집, 수집 대기, 등록 완료 등의 상태에 기반하여 동작하는데, 이러한 중복으로 인해 새로운 명세는 ECSpec과 동일한 상태로 관리될 수 있음에도 불구하고 새로운 명세의 상태를 관리하기 위한 추가적인 프로세스가 필요하다. 이러한 제약으로 인해서 기존의 ECSpec을 확장하는 방법을 제안한다. ALE 표준 명세에 정의된 ECSpec의 각 구성요소들은 확장 지점(Extension Point)을 지원하므로 센서 태그를 지원하기 위하여 기존 ECSpec의 태그 필터링 조건을 확장하여 센서 태그 데이터에 대한 필터링 및 병합 연산의 표현을 가능하게 한다. ECSpec을 확장함으로써 기존 Edge Manager의 ECSpec 관련 컴포넌트들을 재사용 할 수 있고, ALE 표준 인터페이스와도 호환을 유지할 수 있다.

센서 태그 데이터에 대한 필터링 조건을 ECSpec에 확장시키기 위하여 표 2와 같이 몇 가지 연산자와 병합 연산 타입을 정의한다. 병합 연산은 다양한 응용에 따라 추가가 가능하다. 또한 Historical Query와 달리 Snapshot Query와 Long-running Query는 수집 시점의 현재 데이터만 필요로 하므로 이를 구분하기 위한 식별자를 정의한다.

표 2 연산자 및 Aggregation 유형

연산자	MAX	최대값
	MIN	최소값
	RANGE	범위 [lo-hi]
	GT	초과
	GE	이상
	EQ	동등
	LT	미만
LE	이하	
Aggregation 형태	AVG, SUM, MIN, MAX, COUNT, MEDIAN, etc.	

ECFilterSpec은 태그에 대한 여과 조건을 정의하는 클래스로 센서 태그 데이터를 위해서 확장된 ECFilterSpec의 XML 예는 그림 7과 같다.

```

<filterSpec>
  <filterList>
    <filter>
      <includeExclude>INCLUDE</includeExclude>
      <fieldSpec> <fieldname>epc</fieldname> </fieldSpec>
      <patList> <pat>urn:epc:pat:sgtin-64:0.0080876.043242.*</pat>
      </patList>
    </filter>
  </filterList>
  <extension>..... </extension>
</filterSpec>

<extension>
  <sensorFilter type="temperature" isHistorical="TRUE">
    <historicalQuery> <aggregation>AVG</aggregation> </historicalQuery>
    <query>
      <aggregationList>
        <aggregation>COUNT</aggregation>
        <aggregation>MIN</aggregation>
      </aggregationList>
      <conditionList>
        <condition>
          <operator>LT</operator>
          <value>30</value>
        </condition>
      </conditionList>
    </query>
  </sensorFilter>
</extension>
    
```

그림 7 ECFilterSpec의 확장 예

'sensorFilter' 요소는 센서와 질의의 타입을 결정한다. isHistorical 속성이 TRUE이면 'historicalQuery' 요소는 aggregation 형태를 반드시 기술해야 하고, FALSE이면 생략

되거나 NULL일수 있다. 'query' 요소는 필터링 조건과 최종적으로 보고될 센서 태그에 대해서 적용할 병합 연산형태를 기술한다. 필터링 조건과 병합 연산의 형태는 응용에 따라 복수개의 등록이 가능하므로 리스트의 형태로 관리한다.

```

<reports>
<report reportName="report1">
  <group>
    <groupList>
      <member>
        <tag>um:epc:tag:gid-96:10.50.1000</tag>
        <extension>
          <sensor type="temperature">20</sensor>
        </extension>
      </member>
      <member>
        <tag>um:epc:tag:gid-96:10.50.1001</tag>
        <extension>
          <sensor type="temperature">10</sensor>
        </extension>
      </member>
    </groupList>
    <extension>
      <aggregationResultList>
        <aggregationResultMember>
          <aggregation type="MIN">10</aggregation>
        </aggregationResultMember>
        <aggregationResultMember>
          <aggregation type="COUNT">2</aggregation>
        </aggregationResultMember>
      </aggregationResultList>
    </extension>
  </group>
</report>
</reports>
    
```

그림 8 ECReports의 예

그림 8은 센서 태그의 질의 수행의 결과로 보고 받은 ECReports의 예이다. 태그의 ID가 보고되는 영역이 확장되어 센싱 정보가 같이 보고되며, 병합 연산의 결과가 군집화 영역의 확장을 통해 보고된다.

5. 센서 태그를 위한 컴포넌트의 설계

센싱 정보를 획득하기 위해서는 센서 태그의 인식이 먼저 선행되어야 하고 태그 ID 필터링을 거친 태그에 대해서 메모리 읽기를 시도하는 명령을 전달하여 센싱 정보를 수집한다 [7]. 수집되는 센싱 정보는 이미 태그 ID 필터링 과정을 거친 태그의 정보이므로 재차 필터링 과정을 거치지 않고 센싱 정보를 관리하는 모듈로 전달되어야 한다. 그러므로 센서 태그 리더에 센싱 명령을 전달하고 그 결과로써 태그에 저장된 센싱 정보를 수집하는 역할을 하는 컴포넌트가 필요하다. 만약 이러한 구조를 가지지 않고 태그의 인식 단계에서 모든 센싱 정보를 획득한다면 태그 ID를 필터링하는 과정에서 제거될지도 모르는 태그의 불필요한 정보 전송이 발생할 것이다. 특히 Historical Query를 수행하는 경우 이러한 구조는 불필요한 데이터 전송량을 대폭 감소시킬 수 있다.

센서 태그는 리더 영역에서 벗어나더라도 계속적으로 센싱 작업을 수행하여 메모리에 저장한다. 그리하여 장거리 운송을 거치는 동안 대량의 센싱 정보가 메모리의 축적될 수 있고 Historical Query를 사용하여 전체 혹은 일부분의 메모리 데이터를 읽을 필요가 있다. 메모리의 크기에 비해서 리더와 태그 사이에 전송되는 데이터 크기는 아주 작으므로 여러 번의 센싱 정보 전송이 필요하고 이 과정에서 센서 태그의 리더 영역

이탈 또는 무선 통신 장애로 인해 데이터의 손실이 있을 수 있다. 센싱 정보에 대한 정확한 병합 연산을 보장하기 위하여 센싱 정보의 완전한 전송이 이루어져야 하는데 이를 위해서 전송중인 센서 태그에 대한 센싱 정보를 임시로 저장하는 저장소가 필요하고 데이터 손실이 일어난 경우 리더에 손실된 데이터를 재요청하는 기능을 해야 한다.

센싱 정보의 수집이 완료되면 병합 연산을 수행한다. 하나의 태그 내의 센싱 정보에 대한 병합은 센싱 정보 필터링 과정 이전에 수행되어야 한다. 병합이 완료되면 센싱 정보 필터링을 수행하고 질의의 결과로 수집된 다수의 센서 태그에 대한 병합 연산을 수행하여 최종 리포트를 생성한다. 만약 Historical Query가 아닌 Snapshot Query나 Long-running Query라면 태그 내 병합 연산은 수행하지 않고 곧바로 필터링 과정을 수행한다. 이를 종합한 알고리즘은 그림 9와 같다.

```

Algorithm for Filtering Sensor tag data

1:  matched query Q ← execute patternFilter()
2:  if a query Q has sensor filter conditions then
3:    if historical query then
4:      execute Aggregation
5:    endif
6:  execute sensorFilter()
7:  endif
    
```

그림 9 센서 태그 필터링 알고리즘

6. Edge Manager의 설계

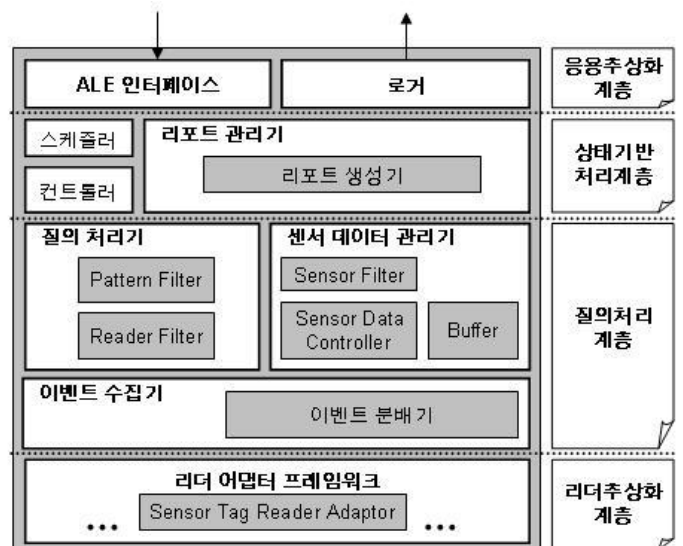


그림 10 블록 다이어그램

그림 10은 5장의 요구사항을 반영하여 센서 태그를 지원하는 Edge Manager의 구조를 블록 다이어그램으로 표현한 것이다. Edge Manager를 구성하는 컴포넌트들은 그 역할에 따라 다양한 리더의 지원을 위한 공통 인터페이스를 지원하는 리더 추상화 계층, 수집된 이벤트를 질의에 등록된 조건에 따라 처리하는 질의처리 계층, Edge Manager의 동작을 상태기반으로 스케줄링하고 질의 수행의 결과 리포트를 생성하는 상

태기반 처리 계층, 그리고 질의의 등록 및 결과 보고를 위한 인터페이스를 제공하는 응용 추상화 계층으로 구분된다[8]. 기존의 passive 태그만 지원하던 Edge Manager를 계층적 구조로 설계함에 따라 다양한 리더 장치와 응용에 대한 확장이 용이하다. 따라서 Edge Manager에서 센서 태그를 지원하기 위해서는 센서 태그 리더에 대한 인터페이스를 어댑터 형식으로 리더 추상화 계층에서 지원해야 하며, 태그 ID에 대한 질의 처리와 함께 센서 태그 데이터에 대한 질의를 처리할 수 있는 컴포넌트가 질의 처리 계층에 추가되어야 한다. 센서 데이터 관리기는 센서 태그에 대한 질의 처리를 담당하는 컴포넌트로 구성 요소는 다음과 같다.

- Sensor Data Controller: 센서 태그에 센싱 정보 수집 명령을 전달, 이벤트 수집기로부터 센싱 정보를 수집하는 역할을 하고 Buffer에 저장된 중간 전송을 관리한다.
- Sensor Filter : 센싱 정보에 대한 필터링을 수행
- Buffer : Historical Query의 결과로 수집되는 대량의 센싱 정보의 완전한 전송을 보장하기 위한 임시 저장소

질의 처리기는 Reader Filter와 Pattern Filter로 구성되어 있고 passive 태그와 센서 태그의 ID에 대해서 필터링을 수행한다. 수집된 태그 이벤트가 센서 태그의 이벤트라면 질의 처리기는 필터링 조건을 만족하는 태그 ID를 센서 데이터 관리기로 전달한다. 센서 데이터 관리기는 Sensor Data Controller를 통하여 해당 태그 ID에 대한 센싱 정보를 획득한 후, Historical Query라면 정확한 병합 연산 결과를 위한 완전한 전송을 보장하기 위하여 Buffer를 사용하여 센싱 정보의 수신을 관리하고 수신이 완료되면 1차 병합 연산을 수행한다. 그리고 센싱 정보에 대한 질의 조건을 만족하는 결과를 리포트 관리기로 전달한다.

이벤트 수집기로 수집되는 이벤트는 센싱 정보를 포함하거나 포함하지 않을 수 있다. 센싱 정보가 포함된 이벤트는 Sensor Data Controller가 내린 명령의 수행 결과 이므로 태그 ID를 필터링하는 질의 처리기로 전송하지 않고 이벤트 분배기를 통해서 센서 데이터 관리기로 전달할 수 있게 한다.

리포트 관리기에서는 질의 수행의 결과를 사용자에게 보고하기 위한 형태로 리포트를 생성하며, 센서 태그 질의인 경우, 2차 병합 연산을 수행한다. 생성된 리포트는 다양한 로거(TCP, HTTP, FILE)를 통해 사용자에게 보고된다.

7. 결론 및 향후 연구

RFID 기술이 발전함에 따라 RFID 태그의 기능이 점차 확대되고 있는 가운데 본 논문에서는 semi-passive 센서 태그를 지원하는 Edge Manager의 구조를 제시하였다. 센서를 소자로 사용하는 또 다른 분야인 센서 네트워크에서의 질의를 분석하고, semi-passive 센서 태그의 특징을 고려함으로써 Edge Manager에 적용하기 위한 요구사항을 도출하고, 이를 바탕으로 RFID 관련 기술 표준 단체인 EPCglobal의 ALE 표

준 명세의 ECSpec을 확장하여 질의를 표현하는 방안을 제안하였다. 그리고 semi-passive 센서 태그의 질의 처리를 위해 센싱 정보에 대한 필터링을 수행하고 센싱 정보의 완전한 전송을 보장하기 위한 컴포넌트를 추가한 Edge Manager 시스템을 설계하였다. 향후 연구로써 다양한 타입의 센서 지원을 위한 동향 분석 및 연구, 리더 영역 밖에 장기간 위치하는 경우 태그의 제한된 메모리로 인해 발생할 수 있는 제약에 관한 해결 방법, active 센서 태그의 지원을 위한 관련 연구가 필요하다.

7. 참고 문헌

- [1]EPCglobal Inc., "The EPCglobal Architecture Framework", September 2007
- [2]EPCglobal Inc., "The Application Level Events (ALE) Specification, Version 1.1" February 2008
- [3]Yong Yao, Johannes Gehrke, "Query Processing for Sensor Networks" CIDR 2003
- [4]Philippe Bonnet, Johannes Gehrke and Praveen Seshadri, "Querying the Physical World", October 2000
- [5]Oracle Application Server 10g : Sensor Edge Server
- [6]Alien Technology Corporation, "Alien Nanoscanner Reader Tag Guide"
- [7]Sanjay Sarma, Daniel W. Engels, "Technical Report on the future of RFID tags and Protocols", AutoID-Lab, June 2003
- [8]최병운, 홍봉희, 류우석, 안성우, 안준환, "다양한 무선인식 소자를 지원하는 동적 재구성가능 미들웨어의 개발", 한국정보과학회 데이터베이스연구회, KDBC2007, pp 414-420