

무선 센서 네트워크를 위한 속성 기반 네이밍 구조

정의현*

An Attribute-Based Naming Architecture for Wireless Sensor Networks

Eui-Hyun Jung**

요약

최근에 응용 개발의 복잡성을 숨겨주고, 상위 응용계층에서 필요로 하는 기능을 추상화하여 제공하는 센서 미들웨어에 대하여 많은 연구진들이 주목하고 있다. 미들웨어를 설계하기 위한 요건들은 다수이지만, 그 중에서도 속성 기반 네이밍의 지원은 매우 중요하게 여겨진다. 하지만 현재 연구된 대부분의 미들웨어는 센서 네트워크의 특성을 제대로 고려하지 않고 설계했을 뿐만 아니라 속성 기반 쿼리에 대한 확장성을 제공하는데 한계가 있었다. 본 연구에서는 가상 대응체(Virtual Counterpart) 개념을 적용하여 속성 기반 네이밍을 싱크 노드의 미들웨어에서 가상의 센서 노드가 지원해주는 구조를 제안한다. 개별 센서 노드에서 속성 기반 쿼리를 처리하는 기존의 데이터 중심 미들웨어와는 달리, 제안된 구조에서는 센서 노드에 대응되는 가상의 센서 노드를 싱크 노드의 미들웨어에서 운용하고, 속성 기반 쿼리를 가상의 센서 노드가 물리적 센서 노드를 대행하여 처리한다. 이러한 접근 방안을 취함으로써, 하부 네트워크에 비 종속적인 속성 기반 네이밍이 가능해졌으며, 확장이 용이해졌다.

Abstract

Recently, a lot of researchers focus on the sensor middleware that hide the complexity of application developments and provide the abstraction of functions to upper application layer. Although there are several factors to design sensor middleware, the attribute-based naming is considered to be an essential factor among them. However, most existing researches were not designed to reflect the characteristics of sensor networks and have the limitation of attribute-based query extension. This study adopts the concept of Virtual Counterpart to suggest the structure where attribute-based naming is supported by virtual sensor nodes of the middleware on the sink node. Unlike traditional data-centric middleware in which individual sensor nodes process attribute-based query, virtual sensor nodes mapped to physical sensor nodes are running on the middleware of the sink node and process attribute-based query as a proxy of the physical sensor. This approach enables attribute-based naming independent of physical infrastructure and easy extensibility.

▶ Keyword : 속성 기반 네이밍(Attribute-based Naming), 센서 미들웨어(Sensor Middleware), 무선 센서 네트워크(Wireless Sensor Network)

• 제1저자 : 정의현
• 접수일 : 2007. 7.18, 심사일 : 2007. 8.13, 심사완료일 : 2007. 9.1.

* 안양대학교 컴퓨터학과 교수

※ 이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-003-D00365).

1. 서론

지난 수년간 센서 네트워크에 대한 폭발적인 관심으로 인하여, 센서 네트워크의 운영체제와 하드웨어 구성, 그리고 네트워크에 대한 많은 연구가 경쟁적으로 수행되었다 [1]. 그러나 아직까지 센서 네트워크의 응용개발은 운영체제나 센서 노드에서 제공하는 저수준의 기능을 이용하여 이루어지고 있으며, 센서네트워크의 개발과정은 특정 응용을 개발 초기부터 고려하여 이에 의존적인 형태로 진행되고 있다 [2]. 이러한 특정 응용에 종속적인 센서 네트워크 개발은 무선 센서 네트워크의 개발과 배포 등을 어렵게 하고 센서 네트워크 제작에 대한 전체적인 비용을 증가시켜서 기 설치된 센서 네트워크에 다양한 응용을 적용하려는 산업계의 요구를 충족시키지 못하고 있다 [3]. 또한 센서 네트워크에서의 새로운 경향은 멀티 프로그래밍(Multi programming)이 가능한 센서 네트워크를 구현하는 것이다. 즉, 하나의 센서 네트워크에 여러 개의 응용을 올리는 것을 중요하게 생각한다. 과거에는 센서 하드웨어와 운영체제에 의존적이며 특정 응용에 종속적인 형태로 센서 네트워크의 기능이나 기술적 적용 가능성을 보여주는 것에 초점을 맞췄다면 앞으로의 추세는 다양한 응용분야를 지원할 수 있으며 사용자와 상황의 요구에 따라 동적으로 적용할 수 있는 유연한 구조로 다양한 센서 응용을 지원하는 센서 네트워크를 개발하는 것이 필요하다 [2][3].

이러한 유연한 구조의 센서 네트워크 개발의 일환으로써 센서 네트워크 운영체제 레벨에서의 소프트웨어 진화와 동적인 소프트웨어 갱신에 대한 연구가 수행되었으며 최근에는 센싱 데이터 추상화(abstraction)를 통해 다양한 응용 개발을 지원할 수 있는 센서 미들웨어(Sensor Middleware)에 특히 많은 연구가 수행되고 있다 [4][5]. 추상화는 센서 네트워크를 위해서 필수적인 데이터의 수집, 이벤트 처리 메커니즘, 전력 관리 그리고 네트워킹 등 센서 네트워크의 필수적인 기능을 포함하며 최종적으로는 센싱 결과들이 마치 하나의 블랙박스과 같이 인식될 수 있도록 해서 다양한 센서 응용을 쉽게 개발할 수 있도록 도와준다. 이러한 미들웨어 개념의 도입은 센서 네트워크 응용이 개발되고 배포되고 운영되는 전체 흐름의 변화를 의미한다.

센서 네트워크에서 데이터 수집과 보고는 센서간의 통신을 통해서 이루어지게 된다. 그러나 센서간의 통신은 일반 데이터 통신에서처럼 개별 센서들과의 종단 통신(End-to-end communication)으로 수행되지는 않는다. 이는 IP와

같은 식별자를 개별 센서에게 주기 어렵다는 점과 전송계층(transport layer)에 대응되는 센서 네트워크의 프로토콜이 개발되지 않았다는 등의 이유에 기인한다. 이러한 이유 때문에 개별 노드간의 통신보다는 센서 노드에 의해 생성된 데이터의 특성에 의해 통신을 구성하는 새로운 영역의 통신인 데이터 중심(data-centric)의 통신이 새로 주목받고 있다 [3][6].

데이터 중심의 통신은 노드의 데이터를 얻어오는 것에 초점이 맞추어져 있는 기존 노드 중심의 통신과 달리 센싱 태스크(task)를 적합한 센서에 할당해주고, 태스크를 할당받은 센서들로부터 데이터를 수집하는 것에 초점이 맞추어져 있다. 이를 위해서는 무엇보다도 속성 기반 네이밍(Attribute-based naming) 기능이 센서 네트워크에서 제공되어야 한다. 속성 기반 네이밍은 원하는 속성을 센서 네트워크에 부여하면, 센서 네트워크가 그 속성에 해당하는 센서들을 선택하고 그 센서들에서만 데이터를 얻는 구조를 갖는다 [7].

예를 들어 "북부 지역의 습도가 30% 이상인 센서들의 온도 데이터를 구하라"라는 태스크는 "북부", "습도 30% 이상", "온도 데이터"라는 속성을 갖고 있는 센싱 태스크라 할 수 있다. 이러한 속성 기반 네이밍은 다수의 센서의 데이터를 융합(fusion)하여 현상을 관측하는 센서 네트워크의 특징과 잘 부합되는 구조이며, 특정 현상에 대한 쿼리를 효과적으로 처리할 수 있다. 그러나 이러한 좋은 특성에도 불구하고 속성 기반 네이밍에 대한 연구는 아직 초기 단계이다. 그 이유는 속성 기반 네이밍을 효과적으로 하부의 노드 기반 중심의 통신 구조와 연동하는 것이 쉽지 않기 때문이다. 즉 실제 데이터를 수집하는 것은 노드 기반 중심의 통신을 통해서 수집되어야 하기 때문에 속성을 어떻게 하부 노드의 개별 속성과 연결할 수 있는지에 대해서는 아직 여러 연구들이 진행 중에 있다.

기존 연구 중, Directed Diffusion[8]은 싱크(sink)의 질의 방송(broadcast)에 기반을 둔 데이터 중심적 라우팅 기법으로 질의 유포 및 처리 응용에 적합한 특성을 가진다. 이 연구에서는 속성(attribute)과 값(value)의 쌍으로 구성된 interest라는 질의가 싱크로부터 네트워크 전체에 배포되어야 한다. 즉 질의가 있을 때마다 네트워크 전체에 질의가 배포되게 되는데, 이렇게 되면 센서 네트워크의 트래픽이 증가하게 되고 각 노드들의 에너지 소비율도 증가하게 되는 문제점을 갖고 있다. SINA[9]와 Cougar[10]는 본 연구와 비슷하게 속성 기반 네이밍을 지원하는 데이터 중심 미들웨어이다. SINA는 SQTL(Sensor Query and Tasking Language)이라는 스크립트 언어를 이용하여 속성 기반 네이밍을 구현하였는데 이것은 정형화된 구조를 갖는 SQL((Structured Query Language)을 기반으로 하고 있다. 즉, 이것은 사전에 정해놓은 속성과 관련된 쿼리(query)만을 처

리할 수 있다는 것과 허부 네트워크가 속성 기반 네이밍에 맞추어 종속된다는 문제점을 갖고 있다. 따라서 허부 네트워크가 다른 네트워크 구조를 사용할 수 없기 때문에 범용성 있는 속성 기반 네이밍 구조를 갖추기 어렵다. 즉, 속성 기반 네이밍 구조에 종속적인 센서 네트워크 밖에는 적용이 안 된다는 점이 문제이다. Cougar 또한 데이터 베이스 기반의 미들웨어로 쿼리에 대해 확장성을 제공하는데 어려움이 있다. 쿼리에 대한 최적화와 확장성을 제공하기 위해 다양한 기술들을 적용시키고 있지만 그로 인해 미들웨어가 더욱 복잡해 졌을 뿐만 아니라 이것으로 쿼리에 대한 최적화는 가능하나 확장성에 대한 완벽한 해결책은 찾지 못하였다.

본 논문은 센서 네트워크의 특성을 고려하여 속성 기반 네이밍을 제공하는 데이터 중심 미들웨어에 대하여 설명한다. 센서 노드에 대응되는 가상의 노드로 구성된 가상의 네트워크를 싱크 노드의 미들웨어에서 운용하고, 속성 기반 쿼리를 가상의 센서 네트워크 층에서 처리하는 구조를 제안한다. 이러한 구조는 센서 노드의 미들웨어를 가볍게 하여 전력 에너지원의 제한성을 가진 센서 네트워크에 적용을 용이하게 하고 가상의 센서 네트워크에서 쿼리를 처리함으로써 데이터 형식이 보다 간략화 되어 네트워크 트래픽이 감소되고 궁극적으로 전체 네트워크의 에너지 효율성을 높일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 미들웨어의 문제점과 제안된 구조에 관해서 설명한다. 3장에서는 제안된 미들웨어를 통한 속성 기반 네이밍 구조에 관해 설명하고, 4장에서는 기존 미들웨어와의 비교를 통한 제안된 미들웨어의 특성을 정성적으로 살펴본 후, 5장에서 결론을 맺는다.

II. 이슈 및 구조

2.1 기존 미들웨어의 문제점

기존의 미들웨어는 일반적인 컴퓨팅 환경에서의 미들웨어와 마찬가지로 각각의 센서 노드들에 설치된 미들웨어에서 직접 모든 데이터를 처리하였다. 하지만 낮은 컴퓨팅 파워와 메모리 용량이 제한된 센서에 덩치가 큰 소프트웨어인 미들웨어를 그대로 적용시키기엔 무리가 있다. 뿐만 아니라 기존의 데이터베이스를 기반으로 한 데이터 중심 미들웨어의 속성 기반 쿼리는 센서 네트워크에 최적화 되어 있지 않아서 쿼리의 파라미터를 확장하는데 많은 문제점이 있었다. 이는 복잡한 쿼리의 처리를 위해서는 여러 번의 데이터 통신이 필요함을 의미하며, 통신을 최적화해야만 하는 센서

네트워크에서는 치명적 약점이 될 수 있다.

제안된 시스템은 센서 네트워크를 위한 데이터 중심 미들웨어로써 센서 네트워크에 최적화시키기 위하여 모든 센서 노드들에 같은 미들웨어를 적용시키는 것이 아니라 비교적 많은 자원을 가지고 있는 싱크(sink) 노드의 미들웨어 상에 가상의 센서 네트워크를 구축하고 실제 센서 노드에 대응되는 가상 센서에서 쿼리를 처리하는 구조로 설계하였다. 따라서 네트워크의 데이터 전송량을 최소화할 수 있고, 데이터베이스 기반이 아니므로 사용자가 얼마든지 필요에 따라 쿼리 파라미터를 확장할 수 있다.

2.2 미들웨어의 구조

현재 연구된 센서 네트워크를 고려한 미들웨어의 경우 전력 에너지원의 제한성을 가진 센서 네트워크의 특성을 고려하지 않고 일반적인 인프라 기반의 미들웨어처럼 모든 센서 노드들에 똑같은 역할을 하는 미들웨어를 적용시켰다. 제안된 시스템은 센서 네트워크의 특성을 고려하여 일반적으로 센서 노드들보다 많은 자원을 가지고 있는 싱크 노드의 미들웨어에 가상 대응체의 역할을 하는 가상 센서(Virtual Sensor)들로 구성된 가상 네트워크(Virtual Networks)를 구축하여 실제 센서 노드에서 처리할 다양한 처리를 가상 센서에서 하게 함으로써 실제 센서 노드의 미들웨어 구조를 단순화 시켰다.

제안된 시스템은 그림 1에서 보는 바와 같이 네트워크 층과 응용 층 사이에 독립적으로 위치하므로 다양한 센서 네트워크 프로토콜(LEACH, BCDP, PEGASIS 등)이 위치할 수 있다. 싱크 노드의 미들웨어는 쿼리 해석기(Query Interpreter)와 가상 네트워크(Virtual Network)로 구성된다. 실제 구현은 센서 네트워크 시뮬레이터인 J-Sim 시뮬레이터[11]를 통하여 실제 센서들과의 통신을 에뮬레이션 하였다.

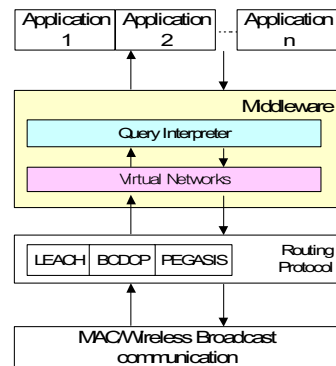


그림 1. 미들웨어의 구조
Fig 1. Architecture of the Middleware

2.2.1 쿼리 해석기

쿼리 해석기는 센서 응용에서 요청하는 다양한 질의를 가상 네트워크 상의 가상 센서들에게 전달하는 역할을 담당한다. 센서 네트워크에서는 센서 응용이 개별 센서와 중단간 통신을 하는 것을 가급적 배제하기 때문에, 속성 기반 쿼리를 적절히 가상 센서에 전달하는 것이 매우 중요하다. 쿼리 해석기는 센서 응용으로부터의 질의를 미들웨어 상의 가상 센서들에게 브로드캐스트하는 역할을 하게 된다. 또한 쿼리 해석기는 가상 센서들이 전달한 값을 취합하고, 쿼리 결과에 따른 부가 연산을 처리하는 역할을 담당한다. 쿼리 해석기가 부가적인 연산을 처리할 수 있기 때문에, 쿼리 확장이 필요한 경우에는 단순히 쿼리 해석기만을 확장하면 된다.

2.2.2 가상 네트워크

가상 네트워크는 그림 2와 같이 가상 대응체[12]의 역할을 하는 가상 센서들로 구성된다. 가상 네트워크에서는 가상 대응체 역할을 하는 가상 센서를 실제 센서 노드와 논리적으로 1:1 대응하는 구조를 갖게 함으로써, 실제 센서 노드의 속성과 데이터를 대응하는 가상 센서가 유지하도록 하고 있다. 이러한 구조에서는 센서 응용이 요구하는 다양한 질의를 싱크에서 분석할 수 있으며 실제 센서 노드 대신 다양한 처리를 할 수 있다. 가상 센서는 싱크 노드 내의 메모리 내에 위치하며 실시간으로 대처 할 수 있도록 각각의 가상 센서는 쓰레드으로써 동작한다. 가상 센서들은 다른 가상 센서들과 통신이 가능하고 가상 센서들이 그룹을 형성할 수 있으며 형성된 그룹의 데이터를 다른 가상 센서에서 이용할 수 있다.

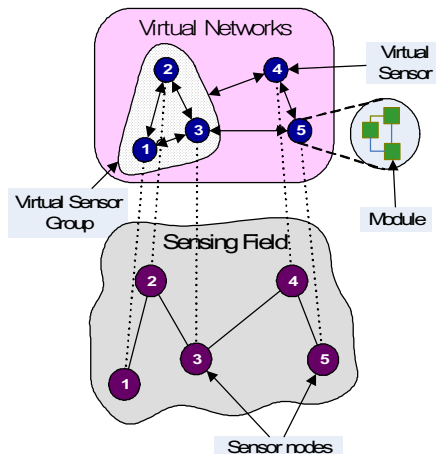


그림 2. 가상 네트워크의 구조
Fig 2. Architecture of the Virtual Network

III. 속성 기반 네이밍

기존의 연구된 속성 기반 네이밍은 정형화된 구조로 설계되었다. 그래서 응용, 데이터, 노드의 속성 등을 추가하거나 수정하기 위해서는 구조 전체를 바꿔야만 했을 뿐만 아니라 쿼리에 대한 확장성도 제공하지 못하여 복잡한 속성 기반 쿼리의 경우 정형화된 쿼리 문맥에 맞추어 여러 번 보낼 수 밖에 없었다. 제안된 시스템은 데이터를 추상화 하고 확장성을 제공할 수 있는 속성 기반 쿼리 문맥을 정의하여 다양한 속성 기반 네이밍을 가능하게 하였다.

3.1 데이터 추상화

가상 센서는 실제 센서 노드로부터 일반적인 데이터를 받아서 "Attribute"로 추상화된 데이터로 구성하여 저장하게 된다. 이렇게 저장된 데이터들은 향후 센서 애플리케이션이 원하는 속성 기반 쿼리에 대해서 속성 기반 네이밍이 가능하게 된다. "Attribute"는 "Property"와 "Sensing Data"로 구분하였는데 "Property"는 센서 노드의 특성으로 노드의 위치정보를 나타내는 "Location"과 에너지 관련 데이터를 정의하는 "Energy"로 구분된다. 그리고 "Sensing Data"는 센서 노드가 센싱하는 데이터로써 "Temperature"와 "Humidity", 두 가지 센서로 정의 하였다. 그림 3은 각 클래스들의 관계를 보여주고 있다.

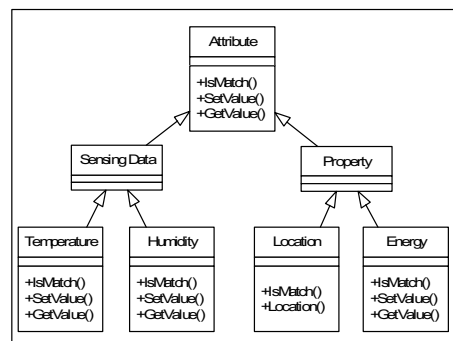


그림 3. 클래스 관계도
Fig 3. Class Diagram

3.2 속성 기반 쿼리의 구조

속성 기반 쿼리의 문법은 그림 4와 같이 정의하였다.



그림 4. 속성 기반 쿼리 문법
Fig 4. Syntax of Attribute-based Query

문법에서 ①항은 쿼리의 종류를 나타낸다. 'GET'은 데이터를 센서 노드로부터 받아오는 것을 의미하고 'PUT'은 데이터를 센서 노드에 보내는 것을 의미한다. ②항은 쿼리의 함수에 해당한다. 현재 정의된 함수는 합, 평균, 개수, 최대값, 최소값 등이며, 이러한 함수는 추가가 가능하다. ③항은 ②항에 어떤 함수가 들어오는지에 따라서 변화되는 값이다. 주로 함수의 결과로 어떤 것이 반환되어야 하는지를 지정하는 값이다. 예를 들어 온도가 25도 이상인 노드들의 수를 알고 싶은 경우에는, 다음과 같은 쿼리를 작성하면 된다.

```
GET Count Temperature OpGE 25
```

속성 기반 쿼리 문법의 세부 항목에 대한 자세한 설명은 표 1과 같다.

표 1. 쿼리 문법의 구성 요소
Table 1. Component of Query Syntax

	파라미터	의미
①	GET	데이터를 받을 때
	PUT	데이터를 보낼 때
② FUNCTION	Average	속성에 대한 평균값
	Sum	속성에 대한 합계
	Min	속성에 대한 최소값
	Max	속성에 대한 최대값
	Count	속성을 갖고있는 노드 수
	Choose Find	속성을 갖고있는 노드선택 속성을 갖고있는 노드 리스트
③ ATTRIBUTE	Location	위치
	Range	지역
	Temperature	온도
	Humidity	습도
	Battery	배터리 양
OPERATION	OpGT	>
	OpGE	>=
	OpLT	<
	OpLE	<=
	OpOr	
	OpAnd	&
	OpEQ	=
	OpNE	!=
	OpIncluded	Range에 포함됨

	OpExcluded	Range에 포함되지 않음
VALUE	Position Area Scalar	위치 관심 지역 스칼라 값

만약 얻고자 하는 데이터가 다음과 같고, 센서 노드들이 표 2와 같이 구성되어 있다면 처리 절차는 다음과 같다.

- ① x:300, y:400 과 x:500, y:100좌표 안에 있는 노드들을 찾아라.
- ② ①번에 해당하는 노드들 중 온도가 15도 이상이거나 습도가 60%이상인 노드들의 수를 세어라.

표 2. 예제 센서 노드의 위치와 속성
Table 2. Location and attributes of sample sensor nodes

	위치(x,y)	습도(%)	온도(섭씨)
센서노드 1	354,458		32
센서노드 2	430,170		13
센서노드 3	590,326	59	
센서노드 4	392,277	84	

(1) 싱크 노드의 미들웨어상의 쿼리 인터프리터는 가상 센서가 해석할 수 있도록 속성 기반 쿼리를 가상 네트워크 상에 모든 가상 센서들에게 전송한다. 그림 5는 전송되는 쿼리의 모습이다.

①	②	③
Get	Count	((LOCATION OpIncluded AREA (300, 400, 500, 100)) OpAnd (TEMP > 15) OpOr (HUMIDITY > 60))

그림 5. 전송되는 쿼리
Fig 5. Broadcasted Query

(2) 가상 센서들은 전송받은 쿼리의 ③ 항을 분석하여 쿼리에 자신의 속성이 포함되는지 아닌지를 판단하고 포함될 때만 대응하는 실제 센서 노드에 데이터를 요구한다. LOCATION 속성을 연산하면, 표 2에서는 센서 노드 1, 2, 4가 해당한다.

(3) 해당하는 가상 센서들은 실제 센서 노드로부터 데이터를 받아서 센싱 데이터에 해당하는 부분을 연산한다. 온도와 습도 부분을 판단하면, 표 2에서의 결과는 센서 노드 1, 4가 해당한다. 해당하는 부분이 ③ 항을 모두 만족하는 경우에는 쿼리 인터프리터로 자신의 데이터를 전송한다.

(4) 쿼리 인터프리터는 전송받은 데이터를 취합한 후, 쿼리의 ②항을 연산한 후 그 결과 값을 센서 응용에게 전달한다.

다. ②항이 "Count"이므로 쿼리 인터프리터는 전송받은 데이터를 센 후에, 결과 값을 만들어 센서 응용으로 전송한다.

IV. 시스템 평가

4.1 구현 용이성

기존에 연구된 미들웨어를 센서 네트워크에 구현하기 위해서는 우선 모든 센서 노드들에 미들웨어를 적용시키기 위하여 네트워크를 전체적으로 수정해야만 하고 모든 센서 노드들이 일반적인 센서 노드의 어플리케이션보다 비교적 복잡하고 무거운 미들웨어를 적용시킬 수 있을만한 컴퓨팅 자원을 가지고 있어야 한다. 뿐만 아니라 현재 연구된 센서 네트워크를 고려한 대부분의 미들웨어는 최적화된 서비스를 제공하기 위해서 네트워크 층 또는 MAC층까지 운용하므로 그 부분까지 수정해야 하는 번거로움이 있다.

그러나 제안된 시스템은 핵심 부분인 가상 네트워크를 싱크 노드의 미들웨어에 구축함으로써 실제 센서 노드의 미들웨어 부분에서 처리해야 할 부분을 가상 네트워크상의 가상 센서에서 처리해 줄 수 있게 하였다. 따라서 기존 물리 센서들은 변동할 부분이 거의 필요 없게 된다. 제안된 시스템은 네트워크층과 애플리케이션층 사이에 독립적으로 존재하기 때문에 기존에 사용하고 있는 네트워크 프로토콜 및 MAC 프로토콜을 그대로 사용할 수 있다. 즉, 다른 계층의 영향을 전혀 주지 않고 쉽게 적용이 가능하다.

4.2 추상화

속성 기반 네이밍을 지원하는 기존의 데이터 중심 미들웨어[9][10]는 모두 데이터베이스를 기반으로 데이터를 관리하고 SQL(Structured Query Language)문을 이용하여 속성 기반 쿼리를 만들었다. 따라서 쿼리에 대한 확장성을 제공하기 위해서는 데이터베이스 스키마를 변경해야만 했고 이것은 전체 네트워크 구조에 영향을 끼쳤다. 그리고 SQL문은 센서 네트워크 속성을 제대로 반영할 수 없어서 센서 네트워크에 적합한 쿼리를 만드는데 많은 어려움이 있었다. 예를 들어 다양한 종류의 센서 노드들에 쿼리를 보낼 때는 각각의 종류별로 쿼리를 만들어서 여러 번 보내야 하는 번거로움이 있었고 특정 지역 안에 있는 노드를 찾기 위해서는 쿼리 내용이 매우 복잡해져야 했다.

제안된 시스템은 표 3과 같이 센서 네트워크의 속성을

제대로 반영할 수 있도록 속성 기반 쿼리를 최적화 하였다. 뿐만 아니라 전체 네트워크 구조에 영향을 끼치지 않고 쿼리의 파라미터를 확장할 수 있게 설계하여 사용자가 보다 쉽게 확장하여 쿼리문을 만들 수 있다.

표 3 SINA와의 쿼리 비교
Table 3. Query Comparison with SINA

쿼리 내용
① x:300, y:400 과 x:500, y:100좌표 안에 있는 노드들을 찾아라 ①번에 해당하는 노드들 중 온도가 15도 이상이거나 습도가 60%이상인 노드들을 찾아라
제안 시스템
GET FIND ((LOCATION Included AREA (300, 400, 500, 100)) And (TEMP > 15) Or (HUMIDITY > 60))
SINA
1. SELECT x, y FROM nodes WHERE x=>300 and x<=500 and y<=400 and y)=100 2. SELECT temperature FROM temperatureTable WHERE temperature > 15 3. SELECT humidity FROM humidityTable WHERE humidity > 60

표 3에서 보는 바와 같이 제안된 시스템은 센서 네트워크의 속성을 최적화하여 보다 간단하게 사용자가 원하는 쿼리를 만들 수 있다. 반면에 SINA는 SQL문을 사용하므로 각 항목마다 쿼리를 따로 만들어야 하며 센서 네트워크의 속성을 고려하지 않으므로 복잡한 구조를 갖는다. 또한 쿼리 내용을 센서 노드의 미들웨어에서 직접 처리하는 기존의 데이터 중심 미들웨어와는 달리 제안된 시스템은 싱크노드의 미들웨어에서 처리하여 해당하는 센서 노드에게만 데이터를 요구한다. 뿐만 아니라 센서 노드에게 쿼리문을 보낼 필요 없이 데이터를 요청만 하면 되므로 메시지 형식이 표 3과 같이 매우 간단해 질 수 있다. 이렇듯 메시지 형식이 간단해지고 해당하는 노드에게만 데이터를 요청하는 메시지를 보냄으로써 전체 네트워크 트래픽이 줄어들게 되고 전체 네트워크의 에너지 효율성을 높일 수 있게 된다.

4.3 기존 시스템 비교 및 실험

표 4는 현재 가장 대표적인 데이터 중심 미들웨어인 SINA와 제안된 시스템을 비교한 자료이다. 제안된 시스템은 메시지 크기가 고정적이고 속성기반 쿼리가 메시지 내에

존재하지 않기 때문에 SINA의 메시지보다 간략해질 수 있다. 뿐만 아니라 가장 대응체 개념을 이용하여 센서 네트워크 특성에 맞게 미들웨어를 최적화 하였고 데이터 베이스 기반이 아닌 새로운 쿼리 문맥을 개발함으로써 쿼리의 확장성을 제공하고 센서 네트워크에 맞게 최적화 하였다. 제안된 시스템은 SINA처럼 Network Response Implosion에 대한 해결책은 제공하지 않지만 현재 개발된 다양한 라우팅 프로토콜(LEACH, LEACH-C, BCDP, PEGASIS 등)을 이용하여 해결할 수 있다.

표 4 SINA와의 특성 비교
Table 4. Characteristics Comparison with SINA

	제안 시스템	SINA
메시지 크기	고정적	가변적
쿼리 파라미터 확장성	O	X
쿼리 최적화	O	X
쿼리 확장성 제공	O	X

SINA와 제안 시스템의 통신 효율을 비교하기 위하여 노드 100개를 가정하였을 때, 트래픽 패턴을 비교하였다. 먼저, 제안 시스템의 경우에는 센서 노드들이 주기적으로 데이터를 싱크 노드에 보고하고, 외부 센서 애플리케이션은 데이터 요청을 싱크 노드에만 하는 구조로 되어 있다. 따라서 센서 애플리케이션이 아무리 많은 데이터 요청을 하여도 실제 센서 네트워크에서는 이로 인한 트래픽이 유발되지 않는다. 따라서 제안 시스템의 경우에 트래픽은 다음과 같은 식을 따르게 된다.

$$traffic(Proposed) = \sum^p (n * \alpha)$$

n은 센서 노드의 총 개수이고, α는 하나의 센서가 싱크로 보내는 데이터의 크기, p는 단위 시간당 센서 네트워크에서 싱크 노드로 보고하는 횟수이다. 즉, 제안 시스템에서는 전체 트래픽이 시간에 따라 선형적으로 증가하는 구조를 갖고 있으며, 센서 애플리케이션의 데이터 쿼리의 수와는 무관하다. 이 부분은 센서 애플리케이션의 단위 시간당 쿼리의 수가 많은 경우에는 장점이 되지만, 적은 경우에는 싱크에 최신의 데이터를 유지하기 위해 센서 네트워크로부터 데이터를 자주 가져와야 하는 문제점을 갖게 된다.

SINA는 제안된 시스템과 달리 비동기적으로 트래픽이 발생하게 된다. 센서 애플리케이션이 데이터를 요청하는 경우에 센서 네트워크로 쿼리가 전달되고, 이 쿼리가 센서 노드에서 해석된 뒤에 데이터가 수집되어 다시 센서 애플리케이션으로 전달된다. 이 과정에서 발생하는 트래픽은 다음의 식을 따르게 된다.

이전으로 전달된다. 이 과정에서 발생하는 트래픽은 다음의 식을 따르게 된다.

$$traffic(SINA) = \sum^q (n * \beta + n * \alpha * \gamma)$$

일단 센서 애플리케이션에서 요청이 있는 경우에는 전체 n개의 노드로 β 바이트 크기를 가진 쿼리가 전송된다. 이렇게 전송된 쿼리는 각 노드에서 해석되고, 노드가 해당 쿼리와 일치하는 경우에는(γ의 확률), 센서가 α 바이트의 센싱 데이터를 전달하게 된다. 식에서 q는 센서 애플리케이션이 단위 시간당 쿼리를 요청 횟수이다. 즉, SINA의 경우에 총 트래픽은 쿼리 요청이 있을 시, 쿼리 문을 전체 센서 노드에 보내는 트래픽과 이 트래픽이 각 센서에 도착했을 때, 일치된 데이터를 보유한 센서 노드가 센서 애플리케이션에 이를 보고하는 트래픽의 합이라고 할 수 있다.

두 시스템을 비교하기 위해, n=100, β=256 바이트, α=20 바이트, γ를 10%로 잡고, p와 q를 변화시키면서 두 시스템의 트래픽을 비교한 것이 표 5이다. 비교에서 보는 바와 같이, SINA의 경우에는 1시간당 6회의 쿼리가 있는 경우, 즉 10분당 1번의 쿼리를 한 경우에 이미 154,800 바이트의 트래픽이 발생한다. 이에 비해, 제안된 시스템에서는 6회의 데이터 갱신이 일어난 경우에 20,000 바이트의 트래픽이 발생한다. 최대치로 60회의 갱신, 즉 1분마다 한 번씩 센서 네트워크에서 데이터를 보고하는 경우에도 120,000 바이트가 발생할 뿐이다. 이 점이 의미하는 것은 SINA의 경우에는 비동기적이어서 언제나 최신의 데이터를 갖고 올 수 있지만, 비동기적으로 10분마다 1번씩 최신의 데이터를 가져오는 경우가, 1분마다 최신의 데이터를 갱신하는 제안 시스템에 비해서도 이미 트래픽이 넘음을 알 수 있다. 또한 2분마다 데이터를 갖고 오는 경우에는 제안 시스템에 비해, 약 12배 이상의 트래픽이 발생함을 알 수 있다. 따라서 본 논문에서 제안된 시스템이 트래픽 감소에 큰 강점을 갖고 있다고 결론 지을 수 있다.

표 5 SINA와의 트래픽 비교
Table 5. Traffic Comparison with SINA

p	q	traffic(Proposed)	traffic(SINA)
6	1	12,000	25,800
10	3	20,000	77,400
20	6	40,000	154,800
30	10	60,000	258,000
40	20	80,000	516,000
60	30	120,000	774,000

V. 결론

본 연구에서는 가상 대응체 개념을 센서 네트워크에 적용하여 속성 기반 네이밍을 싱크 노드의 미들웨어상에서 가상의 센서 노드가 처리해주는 구조를 제안하였다. 이것은 기존의 여러 계층을 바인딩 함으로써 가능했던 물리적인 통신망기반의 방식이나 센서 노드에서 직접 속성 기반 쿼리를 처리하는 기존의 데이터 중심 미들웨어와는 달리, 센서 네트워크의 데이터 중심 통신과 저전력을 고려한 속성 기반 네이밍 기능을 제공하게 된다. 또한, 이 방식은 속성 기반 네이밍 뿐만 아니라 실시간을 요하는 여러 가지 응용들을 적용할 수 있고 이종의 다양한 센서 노드들을 쉽게 관리/운영 할 수 있다. 그리고 센서 노드들의 변경 없이 기존 시스템 위에 새로운 구조의 센서 애플리케이션을 쉽게 구축할 수 있다. 하지만 싱크 노드에는 많은 메모리와 에너지를 요구하는 구조가 된다는 단점을 가지고 있다. 그러나 일반적으로 싱크 노드는 다른 센서 노드들에 비해서 고정적인 구조를 갖고 있는 경우가 많기 때문에 그러한 구조에 적합하도록 구성될 수 있을 것으로 기대된다. 향후 속성 기반 쿼리의 확장과 가상 센서들 간의 결합 동작(composite operation)에 대한 연구가 진행되어야 할 것이다.

참고문헌

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," IEEE Computer, pp.41-49, vol.37, no.8, Aug. 2004.

[2] 김영만 "센서 네트워크 미들웨어 구조 및 연구현황," 정보과학회지 제22권 제12호

[3] S. Hadim and N. Mohmed, "Middleware: middleware challenges and approaches for wireless sensor networks," IEEE Distributed Systems vol.7, no.3, Mar. 2006.

[4] B. Krishnamachari, D. Estrin and S. Wicker, Impact of Data Aggregation in Wireless Sensor Networks, Preceedings of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02), pp. 575-578, 2002.

[5] 나선용, 이상정, 김동균, 최영길, "무선 센서 네트워크를

이용한 지능형 홈 네트워크 서비스 설계," 컴퓨터정보학회 논문지, 11권 5호, 2006.

[6] 이기욱, 성창규, "유비쿼터스 센서 네트워크 기반의 상황 정보 모니터링 시스템 구현," 컴퓨터정보학회 논문지, 11권 5호, 2006.

[7] J. Heidemann, F. Silva, C. Intagaonwivat, R. Govindan, D. Extrin, and D. Ganesan, "Building efficient wireless sensor networks with low level naming," In Proceedings of the Symposium on Operating Systems Principles, pp. 146-159, Banff, Alberta, Canada, October 2001

[8] C. Intanagonwivat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," IEEE Trans. Networking, vol. 11, no. 1, pp. 2-16, Feb. 2003.

[9] C. Shen, C. Srisathapornphat, C. Jaikeo, "Sensor Information Networking Architecture and Applications," IEEE Personal Communications, Vol.8, No.4, pp. 52-59, Aug. 2001.

[10] Y. Yao and J.Gehrke. "The Cougar Approach to In?Network Query Processing in Sensor Networks," SIGMOD Record, Vol. 31, No. 3, Sept. 2002.

[11] J-Sim Sensor Network Simulator, <http://www.j-sim.org/>.

[12] ROMER, K., MATTERN, F., DUBENDORFER, T., AND SENN, J. Infrastructure for Virtual Counterparts of Real World Objects, Apr. 2002.

저자 소개



정의현

1999년 2월 : 한양대학교 전자공학 박사
 2004년 ~ 현재 : 안양대학교 컴퓨터학과 전임강사
 관심분야 : 시맨틱 웹, 디지털 컨버전스