메타데이터를 이용한 능동규칙 이동에이전트의 정 방향 이주

이연식* · 이준호**

Forward Migration of an Active Rule Mobile Agent using the Meta_data

Yonsik Lee* · Junho Lee**

이 논문은 이 논문은 2011년도 정부의 재원으로 한국연구재단(No. 2009-0074891)의 지원과
중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(No. C0028654)의 지원으로 수행한 연구임.

요 약

이동에이전트의 노드 이주 방법은 분산 시스템의 전체 성능에 큰 영향을 줄 수 있는 요소가 되므로, 이러한 이동에이전트의 센서 네트워크 내에서의 효율적 이주를 위한 방법이 요구되며, 이를 위하여 다양한 센서 네트워크 구성요소들(서버, 싱크 및 센서노드들) 관련 데이터들을 수집 및 저장하여 일관된 네이밍 서비스를 제공해야 할 필요가 있다. 따라서 본 논문에서는 센서데이터 서버의 정보가 저장되는 *MetaData*와 싱크노드들과 그들에 연결되어있는 센서노드들의 다양한 정보가 저장되는 *SubMetaData* 부분으로 나누어 메타테이블을 설계 구현하고, 이러한 메타테이블의 정보들을 이용한 RMI 기반의 네이밍 기법을 적용하여 능동규칙 이동에이전트의 정 방향 이주 방법을 구현함으로써 효율적인 센서 네트워크 응용 환경 구축 가능성을 제시하였다. 또한, 본 논문에서는 네이밍 에이전트를 J2EE 모델 기반의 RMI-IIOP(Internet Inter-ORB Protocol) 기술을 적용하여 설계 및 구현함으로써, 새로운 센서 네트워크 환경에 적합한 등록, 해제 및 검색 등을 수행할 수 있도록 하였다.

ABSTRACT

Since a migration method of the mobile agent is a factor that affects the overall performance of the entire distributed system, it is necessary to find efficient migration methods of the mobile agent within the sensor network and to collect and store data related to various components(server, sink and sensor node) of the sensor network, thereby providing consistent naming services. Accordingly, this paper presents design and implementation of *MetaTable* that is divided into *MetaData* where information on the sensor data server is stored and *SubMetaData* where various types of information on sink nodes and data on sensor nodes connected with the sink nodes is stored. Moreover, the paper also presented the implementation of forward migration of an active rule mobile agent applying the naming method based on RMI that used the meta_table and proposed the possibility of constructing efficient sensor network application environment. In this paper, for registration, release and retrieval methods suitable for new sensor network environment, we designed and implemented the naming agent by applying J2EE model based on RMI-IIOP(Internet Inter-ORB Protocol) technique.

키워드

에이전트 미들웨어, 이동에이전트, 정 방향 이주, 메타데이터

Key word

Agent Middleware, Mobile Agent, Forward Migration, Meta_data

* 정회원 : 군산대학교 교수(교신저자, yslee@kunsan.ac.kr)
** 정회원 : 삼성전자(주) 네트워크사업부 연구원

접수일자 : 2012. 02. 06
심사완료일자 : 2012. 03. 05

Open Access http://dx.doi.org/10.6109/jkiice.2012.16.7.1567

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ⅰ. 서 론

센서 네트워크 미들웨어의 주요한 개념인 추상화는 센서 네트워크를 위해서 필수적인 데이터의 수집, 이벤트 처리 메커니즘, 전력 관리 그리고 네트워킹 등의 기능을 사용자에게 숨겨서 다양한 응용들을 하위 계층에 대한 고려 없이 쉽게 개발할 수 있도록 지원하는 개념이다. 센서 네트워크의 특성상 중복된 데이터의 전송이나 비효율적인 센싱 주기 등은 센서노드의 수명을 짧게 만들고, 시스템 전체의 성능에 악영향을 끼친다. 또한 이러한 특성을 기반으로 이루어지는 센서 네트워크 관련 응용 개발은 운영체제나 센서노드의 저수준의 기능을 이용하여 특정 응용에 의존적으로 이루어지고 있는 실정이므로, 사용자의 요구에 따라 동적으로 적용 가능한 유연한 구조의 센서 네트워크 미들웨어에 관한 연구들이 진행되고 있다[1,2,3,4]. 그 중 일부 연구들이 미들웨어의 주요 특징인 추상화를 이용하여 센서 네트워크상에서의 필수적인 데이터의 수집, 이벤트 처리 메커니즘, 전력 관리 및 네트워킹 등의 기능을 사용자에게 숨겨서 다양한 응용들을 하위 계층에 대한 고려 없이 쉽게 개발할 수 있도록 수행되었다[1,3,9,11].

본 논문에서는 위의 특징들을 수용하기 위하여 원격에서 이동에이전트를 이용하여 자동으로 센서노드들이 동작 할 수 있도록 할 뿐만 아니라, 제한된 자원을 조절하여 활용할 수 있는 알고리즘을 포함하고, 중복 데이터 제거 모듈을 통하여 데이터가 통신의 주체가 되는 새로운 방식의 통신 등을 지원하는 이동에이전트 미들웨어를 사용자의 요구에 의한 이벤트 중심의 능동적 데이터 처리 방식으로 설계 및 구현한다.

이를 위하여, 센서 네트워크 미들웨어 상의 능동규칙 이동에이전트 시스템과 에이전트의 이주 대상 서버들의 메타데이터를 가지고 있는 메타테이블을 시스템에 적합하도록 설계한다. 또한 센서 네트워크 환경에서 싱크노드뿐만 아니라 센서노드까지 능동규칙을 탑재한 이동에이전트가 각 센서노드들을 정 방향으로 이주하며 센서데이터를 획득하고, 인접 센서노드들과의 데이터 비교를 통한 중복 데이터 처리 및 센싱 데이터에 의한 인접 노드를 제어하는 능동규칙의 실행을 실험을 통하여 보임으로써 구현 시스템의 응용 가능성을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서 센서 네트워크상의 이동에이전트 미들웨어 시스템의 구조와 센서 데이터 처리 과정을 보이고, 3장에서는 센서 네트워크 응용 환경에 적합한 네이밍 에이전트의 메타테이블을 설계 및 구현하고, 네이밍 서비스를 이용한 서버리스트 생성과정을 설명한다. 4장에서는 이동에이전트 구조와 이주 리스트에 따른 능동규칙 이동에이전트의 정 방향 이주 방법을 보인다. 5장은 실험 및 결과를 보이고, 마지막으로 결론 및 향후 연구 방안을 제시한다.

Ⅱ. 센서 네트워크상의 능동규칙 이동에이전트 미들웨어 시스템

능동규칙 이동에이전트 미들웨어는 원격에서 이동에이전트를 이용하여 사용자의 개입 없이 자동으로 센서노드들이 동작 할 수 있도록 할 뿐만 아니라, 제한된 자원의 활용을 조절하여 사용할 수 있는 알고리즘과 능동규칙을 적용하여 센서데이터에 따른 능동적인 작업을 수행할 수 있는 시스템이다[7,8].

능동규칙 시스템과의 연동을 위하여 센서데이터 서버에 RMI통신을 위한 푸시 에이전트를 탑재하였으며, 각 센서 네트워크 구성요소(서버, 싱크 및 센서노드 등)의 위치, 명칭 및 속성 정보는 네이밍 에이전트의 네임 스페이스에 저장된다. 사용자와 관리자는 요구사항에 적합한 서버 관련 정보를 네임 스페이스의 메타테이블을 통하여 검색하고, 이동에이전트를 통해 특정 명령의 전달 및 규칙을 실행할 수 있도록 하는 등의 센서 네트워크 미들웨어 기능을 수행한다. 센서데이터 서버에 탑재된 능동규칙 시스템은 에이전트와 센서 네트워크를 연동하여 대리자와 같은 역할 수행의 장점과 더불어 사용자 또는 관리자의 요구를 이벤트로 보고 저장된 Rule Base에 따라 센서데이터를 제공하거나 센서데이터의 필터링이나 rule의 추가, 삭제 등이 가능하게 하고, 이외에도 고정적인 간격으로 수집되는 센서데이터를 이벤트로 간주하여 Rule Base을 기반으로 능동적인 제어 및 관리가 가능하도록 한다. 다음 그림 1은 능동규칙을 적용한 센서데이터 처리 과정을 보여준다.

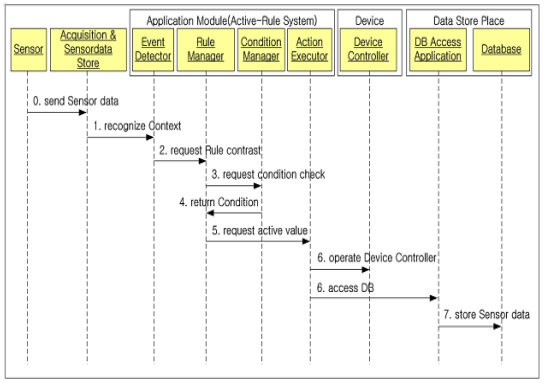

그림 1. 능동규칙을 적용한 센서데이터 처리 과정
Fig. 1 Sensor Data Processing Applying Active Rule

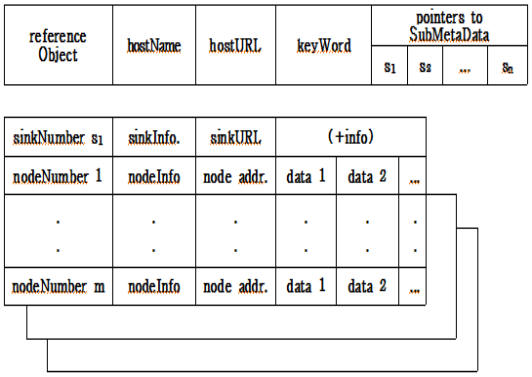

그림 2. 센서 네트워크 응용을 위한 메타테이블
Fig. 2 The Meta_table for Sensor Network Application

Ⅲ. 네이밍 에이전트

3.1. 네이밍 에이전트의 메타테이블 구조

네이밍 에이전트는 네임 스페이스(Name Space)에 등록된 센서데이터 서버, 서버 푸시 에이전트, 클라이언트 푸시 에이전트, 이동에이전트의 정보를 수집하여 통합된 네이밍 서비스의 기능을 제공한다[6,7,12]. 이러한 센서데이터 서버를 포함한 네이밍 서비스를 위하여 싱크 및 센서노드의 식별이 필요함에 따라 USN 노드들의 주소 및 네임체계, 네임 리졸브(resolve) 및 비동기적 데이터처리 체계가 요구되는데[5,10,12], 본 논문에서는 주소체계를 인터넷망과 USN망으로 된 2계층으로 구분하고, 데이터 서버와 연동된 USN은 싱크노드와 센서노드 계층으로 구성하여 각 계층의 노드에 일련번호를 부여하고 번호를 조합한 주소를 사용한다. 주소의 형식은 Data server IP 주소::sink node 번호.sensor node 번호(예: 202.31.147.40::1.2)와 같다.

본 논문에서는 이와 같은 주소체계를 사용하여 센서네트워크상의 싱크 및 센서노드까지 검색이 가능하고, 또한 노드의 수에 따라 확장이 가능한 형태의 새로운 메타테이블을 다음 그림 2와 같이 센서데이터서버의 정보가 저장되는 MetaData와 싱크노드들과 그들에 연결되어 있는 센서노드들의 정보가 저장되는 SubMetaData부분으로 나누어 설계 구현한다.

그림 2의 윗부분은 센서데이터 서버 관련 정보를 나타내는 MetaData이며, 아랫부분은 MetaData로부터 포인터로 연결되어 해당 서버에 관련된 싱크 및 센서노드 관련 정보를 나타내는 SubMetaData 테이블이다. SubMetaData의 싱크노드(sinkURL) 및 센서노드(node addr.)의 주소를 이용하여 직접적인 실시간 센서데이터 획득 및 이동에이전트의 이주를 통한 규칙 실행을 가능하게 한다.

3.2. 네이밍 서비스를 이용한 서버리스트 생성

클라이언트 푸시 에이전트는 객체 참조자를 이용하여 이동에이전트를 원하는 서버 푸시 에이전트나 센서데이터 서버에 보내게 된다. 에이전트 이주를 위한 서버리스트를 생성하기 위하여 먼저 네이밍 서비스는 등록을 원하는 서버 푸시 에이전트나 센서데이터 서버의 요청을 받아 순서대로 이들을 네임 스페이스에 원격객체로 등록한다. 객체 참조자를 얻는 과정은 클라이언트(클라이언트 푸시 에이전트)가 RMI 프레임워크에서 제공하는 네이밍 레지스트리와 네이밍 서비스를 이용하여 네이밍 스페이스 내의 객체 참조자를 얻어낼 수 있으며, 메타테이블 내에 있는 여러 서버들의 목록을 클라이언트 푸시 에이전트에서 입력한 키워드로 검색하여 이에 해당하는 서버들의 리스트를 반환한다. 다음 그림 3은 이동에이전트의 이주를 위한 서버리스트 생성 알고리즘이다.

```
Server List Generation
Input : keyWord
Output : Server List
Process :
NamingSearch (keyWord);
GET MetaDatae;
WHILE(MetaData.HasNext())
  IF (keyWord == MetaData.keyWord)
      THEN
          Server List[][] = hostName, hostURL, keyWord,
                            sinkNumber, sinkURL, sinkInfo;
          Return Server List[][];
      ELSE
          Return Null;
```

그림 3. 에이전트 이주를 위한 서버리스트 생성 알고리즘
Fig. 3 Server List Generation Algorithm for Agent Migration

Ⅳ. 이동에이전트 구조 및 정 방향 이주

4.1. 이동에이전트 구조

이동에이전트는 이주 대상 목록인 서버리스트를 가지고 센서노드들 사이를 이주하며 센서데이터를 획득, 전송하며 센서데이터를 수집하는 역할을 한다. 이러한 이동에이전트는 싱크노드뿐만 아니라 센서노드를 이주하며 적절한 데이터만을 선택적으로 수집할 수 있고, 중복 데이터 수집을 막을 수 있다[3,4,6,8,10]. 본 논문에서 제시한 이동에이전트는 센서데이터를 수집한 후 능동규칙과의 연동 과정을 거친 후 규칙에 적용된 작업을 수행하는 정 방향 이주를 수행한다. 다음 그림 4는 이동에이전트의 구조를 나타낸다.

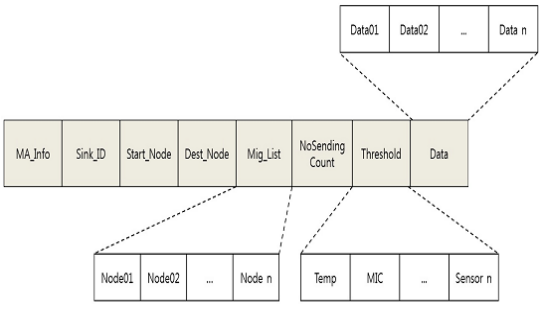

그림 4. 이동에이전트 구조
Fig. 4 A Structure of Mobile Agent

이동에이전트 구조에서 MA_Info는 이동에이전트의 정보로써 이동에이전트를 수신하는 노드에서 해당 에이전트를 확인 후 수신할 수 있도록 이동에이전트를 식별하는 식별자로 쓰인다. Sink_ID는 해당 이동에이전트가 처음 이주를 시작한 싱크노드의 ID로써 최종적으로 이동에이전트가 돌아가야 하는 싱크노드를 나타낸다. Start_Node는 현재 이동에이전트가 어디로부터 온 것 인지를 나타내고, Dest_Node는 목적지 노드로써, 센서노드에서는 이 Dest_Node가 자신인지 확인 후 에이전트를 수신하게 된다. Start_Node와 Dest_Node가 정확해야 노드에서 이동에이전트 역할 수행 후 다음 이주 노드를 결정할 수 있으며, Mig_List의 모든 노드를 정확히 이주 할 수 있다. Mig_List는 모든 이주 노드의 리스트를 나타낸다. Mig_List를 가지고 Agent Migration Manager는 방문한 노드와 아직 방문하지 않은 노드를 구분하여 다음 방문할 노드를 결정한다. Data 는 센서노드에서 수집한 센서데이터를 저장하는 공간이다. 본 이동에이전트는 인접한 2개 노드의 센서데이터를 가지고 이동하며, 해당 센서데이터를 기반으로 중복 데이터 처리를 한다. NoSendingCount는 중복 데이터 처리를 한 횟수로써, 중복 처리 횟수가 많아짐으로써 해당 영역 내의 데이터 분포 특성을 파악할 수 없음을 방지하기 위한 값이다. Threshold는 센서별 중복 데이터 처리의 임계값이며, 본 논문에서는 온도와 조도의 임계값으로 구성한다.

4.2. 이동에이전트의 정 방향 이주

정 방향 이주 방식의 이동에이전트는 해당 센서노드에서의 작업을 모두 마치면 다음 노드로 이주한다. 이동에이전트는 Agent Migration Manager가 Mig_List에서 Dest_Node를 선택하여 Agent Communication Module에 있는 Agent Sender를 통해 이주한다. 다음 노드를 결정하는 것은 이동에이전트의 이동 거리나 정확도, 중요도, 혹은 결로가 생겼을 때 회피방법 등 다양한 라우팅 방법으로 이루어 질 수 있으며, 본 논문에서는 이주방식에 대해서만 구현한다. 다음 그림 5는 센서네트워크상에서 이동에이전트의 정 방향 이주 알고리즘이다.

```
Agent Migration Algorithm
INPUT : nextNodeNum, ma_data, reading
OUTPUT:
PROCESS:
IF (nextSend == TRUE && reading == NREADINGS) THEN
    IF(!sendbusy && sizeof ma_data <= call AMSend.maxPayloadLength()) THEN
        memcpy(call AMSend.getPayload(&send_data), &local, sizeof ma_data);
        call AMSend.send(nextNodeNum, &send_data, sizeof ma_data) == SUCCESS
```

그림 5. 이동에이전트의 정 방향 이주 알고리즘
Fig. 5 Mobile Agent (Forward) Migration Algorithm

이동에이전트의 정 방향 이주방식은 라디오 통신을 사용하며, TinyOS의 액티브 메시지 구조를 통해 이루어진다. IEEE 802.15.4 호환 모뎀을 사용하는 라디오 통신은 모트의 핵심 기능 중 하나로 모트에서 센서를 통해 수집한 데이터를 다른 모트로 전송하며 네트워크를 형성한다. IEEE 802.15.4는 라디오 통신을 위해 물리 계층의 프레임과 MAC계층의 패킷 구조를 정의하고 있으며, TinyOS는 라디오 스택의 추상화 과정을 통해 IEEE 802.15.4의 패킷 및 프레임 구조에 대응하는 새로운 액티브 메시지 구조를 정의해 사용자에게 제공한다. 액티브 메시지는 message_t 구조체로 표현되며 상위 응용에서 라디오 스택으로 데이터를 전달하거나 라디오 스택의 내부 상태를 상위 응용에게 알릴 때 사용하며, message_t의 내부 구성은 프레임 및 패킷에 대응하는 필드 외에도 라디오 모뎀을 운영하는데 필요한 추가 정보가 메타데이터란 형식으로 message_t에 포함된다. 액티브 메시지를 이용한 데이터 송수신은 상위 응용에서 ActiveMessageC 컴포넌트로 라디오 스택을 초기화 하고, AMSenderC와 AMReceiverC의 인스턴스를 메시지 유형에 대한 전달 인자와 함께 생성한 후 해당 컴포넌트에 접근하는 인터페이스를 이용하여 이루어진다. TinyOS에서 제공하는 message_t 구조체는 사용자 데이터 영역의 경우 TOSH_DATA_ LENGTH 매크로 상수에 의해 28로 정의하고 있으므로, 사용자는 TOSH_DATA_ LENGTH 매크로 상수의 크기를 재정의해 최대 116 바이트까지 늘릴 수 있다. 이동에이전트는 이 영역을 사용하며, 이동에이전트 mobileAgent_t 구조체를 message_t 액티브 메시지에 담아 보냄으로 이주를 하게 된다.

Ⅴ. 실험 및 결과

5.1. 이동에이전트의 정 방향 이주 실험 환경 및 과정

제시된 이동에이전트 미들웨어의 실험은 실제 센서노드인 Hmote2420 모델을 사용하였다. Hmote2420의 MCU는 TI사의 MSP430F1611이고, RF칩은 CC2420이다. 실험 시 사용한 통신 주파수 대역은 2405MHz이고, 송/수신 강도인 RF Power는 [Output Power 0 dBm, Current Consumption 17.4mA]이다. OS는 TinyOS-2.x를 사용하고 개발을 위해 Cygwin 툴을 사용하였다. 이동에이전트의 이주 순서는 '모든 노드가 서로 통신이 가능한 거리 안에 있다'는 가정 하에 실험 하였다. 실험 센서데이터는 온도와 조도 데이터이며 싱크노드를 포함하여 총 7개의 노드를 사용하였다. 실험은 각 센서노드에 이동에이전트 플랫폼 이미지를 업로드 한 후 센서 값의 확인과 전등의 상태를 확인하기위한 시뮬레이션을 통하여 수행하고, 이동에이전트의 이주와 데이터 전송 여부는 각 센서노드에 부착된 Led 전구의 on/off 상태로 확인한다. 본 논문에서는 다음 표 1과 같이 센서노드의 센서데이터 값들(온도(℃), 조도(lux))을 센서별로 고정시킨 상태로 실험하였다.

표 1. 실험에 사용된 센서데이터의 값
Table. 1 Sensor Data for Experiments

	node 1	node 2	node 3	node 4	node 5	node 6
온도 (℃)	18	24	26	30	34	32
조도 (lux)	70	65	55	30	50	40

실험은 이동에이전트가 센서노드로 이주를 완료한 후 센서데이터를 획득하면 온도 데이터의 중복 제거 과정과 조도 데이터에 의한 인접 노드를 제어하는 능동규칙의 실행 여부 판단과정으로 이루어진다. 온도 데이터의 중복 처리 규칙은 직전에 방문한 노드와의 데이터 차이가 THRESHOLD 보다 클 경우에만 싱크노드로 전송하는 것이다. 또한, 인접 노드 제어 규칙은 현 노드에서 획득한 조도 데이터가 주어진 THRESHOLD 보다 작을 경우에만 싱크노드로 값을 전송하고, 해당 센서노드에 연결된 전등의 전원과 이주 리스트상의 이전 노드와 다음 노드의 전원을 동시에 on 시키는 것이다. 구현

된 중복된 온도 데이터 제거 알고리즘에서는 광범위한 영역의 센서데이터들의 전체적인 변화 및 분포 관련 추이를 표현하기 위하여, 임계 횟수(NoSendingCount) 이상 중복으로 판단될 경우 무조건 싱크노드로 해당 데이터를 전송하도록 하였다. 또한 인접 노드 제어 실험을 위하여 센서에 연결된 전원의 가동 여부를 시뮬레이션 프로그램으로 구현하였다. 다음 그림 6은 이동에이전트의 중복 데이터 제거 및 인접 노드 제어 과정을 나타낸다.

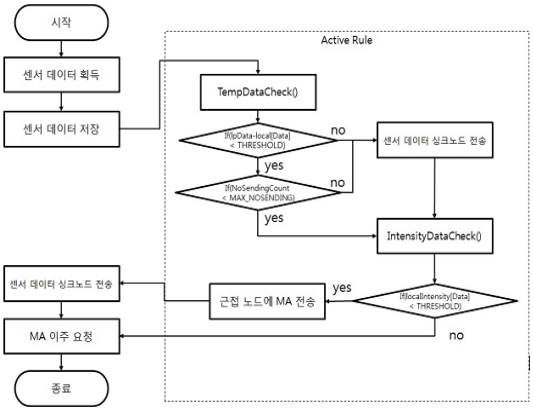

그림 6. 중복 데이터 제거 및 인접노드 제어 과정
Fig. 6 Process of Redundant Data Elimination and Adjacent Node Control

여기서, 주어진 THRESHOLD나 MAX_NOSENDING 등의 값들은 센서데이터의 변화가 심할수록 더욱 작은 값을 적용시킴으로써 전체적인 센서데이터들의 변화 추이를 보다 자세히 알 수 있듯이, 이들 값들을 관심지역 내의 센서 분포 정도나 특성 등에 적합하게 설정함으로써 다양한 센서 응용들에 적용할 수 있도록 하였다.

5.2. 실험 결과

중복 데이터 제거를 위한 온도의 THRESHOLD 값은 2이고 조도의 THRESHOLD 값은 45로 설정하였다. 다음 그림 7은 이동에이전트가 각 노드를 정 방향으로 이주하여 센서데이터를 단순히 싱크노드로 전송하고 다른 작업을 수행하지 않은 화면이다.

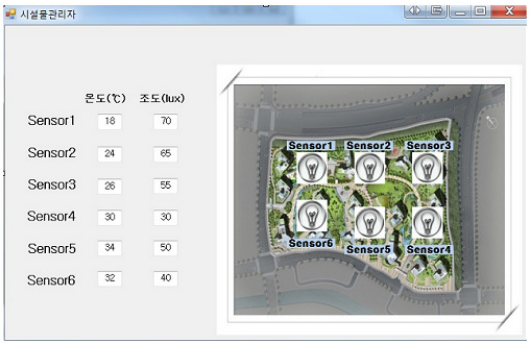

그림 7. 능동규칙을 탑재하지 않은 이동에이전트의 노드 이주 결과
Fig. 7 The Result of Mobile Agent Migration without Active Rule Execution

다음 그림 8은 2가지 능동규칙 모듈을 탑재한 이동에이전트가 4번 센서노드까지 이주 결과이다.

그림 8. 중복 데이터 처리 및 조도 값을 통한 인접 노드 제어 실험 결과 1
Fig. 8 The Results of Active Rule Execution - 1

그림 8의 (A)는 시뮬레이션 프로그램의 초기화면이고, (B)는 이동에이전트가 4번 센서노드까지 방문 후의 결과이다. 센서 1번과 2번의 온도 데이터 차이가 THRESHOLD인 2보다 크기 때문에 싱크노드로 전송하였으며, 조도 데이터는 조도의 THRESHOLD인 45보다 크기 때문에 싱크노드에 전송이 되지 않았다. 센서 3번의 경우 [표 1]의 데이터 값에서 볼 수 있듯이 온도 데이터가 26으로 센서 2번 데이터와 차이가 2보다 크지 않으므로 싱크노드로 전송되지 않았으며, 조도 데이터 또한 THRESHOLD보다 크기 때문에 싱크노드에 전송되지 않았다. 센서4번에서는 조도가 THRESHOLD 이하이므로 싱크노드에 전송되고, 해당 센서노드와 인접한 두 노

드들로 에이전트가 이동하여 연결된 전등을 on 시킴을 알 수 있다.

다음 그림 9는 이동에이전트가 위와 동일한 방법으로 각 센서노드를 한 번씩 방문 후 싱크노드로 돌아온 후 결과이다.

그림 9. 중복 데이터 처리 및 조도 값을 통한 인접 노드 제어 실험 결과 2
Fig. 9 The Results of Active Rule Execution - 2

이와 같이 이동에이전트가 정 방향으로 이주하면서 중복 데이터를 처리하고 인접 노드를 제어하는 능동규칙을 실행함을 실험을 통하여 보임으로써, 센서데이터를 모두 싱크노드로 전송하는 과정에서 발생하는 노드의 수명 단축, 데이터 과부하 및 대역폭 오버헤드 등의 문제를 해결할 수 있는 센서 네트워크상에서의 능동규칙을 탑재한 이동에이전트 미들웨어의 유효성을 제시하였다.

VI. 결론 및 향후 연구

센서 네트워크 환경에서 데이터 수집, 중복 데이터 제거 및 노드 제어 등의 규칙 모듈을 탑재한 이동에이전트 미들웨어를 이용함으로써 비효율적인 센서데이터의 과잉 송수신 방지 및 시스템 수명 연장 등을 기대할 수 있다. 그러나, 기존 센서 네트워크 환경에서 센서 관련 데이터들을 효율적으로 이용할 수 있는 새로운 메타데이터 수집, 저장, 관리 및 네이밍 서비스 방법과 다양한 상황과 환경에 능동적으로 대처하기 위한 능동규칙 탑재 이동에이전트의 이주 관련 연구가 요구된다.

본 논문에서는 센서 네트워크 환경에서 센서데이터 서버, 싱크 및 센서노드의 위치정보와 다양한 특성 등을 포함하는 서브 메타테이블을 추가하여, 센서 네트워크를 연동한 능동규칙 에이전트 시스템에 맞도록 새로운 메타테이블을 설계 및 구현 하였다. 제안한 메타테이블은 센서데이터서버의 정보가 저장되는 MetaData와 싱크노드들과 그들에 연결되어있는 센서노드들의 정보가 저장되는 SubMetaData부분으로 나누어 설계 구현되었다. 이러한 메타테이블을 적용하여 센서 네트워크 환경에서 센서노드까지 능동규칙을 탑재한 이동에이전트가 이주하며 센서데이터를 획득하고, 사용자나 관리자의 특별한 조작 없이 센서노드들을 이주 하며 센서데이터에 따라 2가지 이상의 서로 다른 능동규칙을 실행 하는 능동규칙을 탑재한 이동에이전트를 구현하였다. 또한, 이동에이전트의 이주 방식은 이주 대상들을 포함하는 서버리스트를 생성하여, 그에 따라 차례로 이주하는 정 방향 이주 방식으로 구현하였으며, Hmote2420 센서노드들을 사용하여 센서데이터 수집, 중복 데이터 제거 및 인접 노드 제어 모듈 등의 능동규칙을 탑재한 이동에이전트의 이주 및 규칙 실행과정과 결과를 보임으로써 센서 네트워크 환경에서의 이동에이전트 미들웨어의 유효성을 보였다.

본 연구의 결과는 센서 네트워크 응용 개발에 있어서 다음과 같은 장점을 제공한다. 첫째, 이동에이전트를 이용하여 환경에 대한 정보를 직접 감지하고, 센서데이터 수집 방법 및 주기 변경 등의 동적인 환경 변화에 빠르게 적응 할 수 있다. 둘째, 응용에 적합한 다양한 능동규칙의 탑재 및 실행을 통하여 네트워크 부하 경감, 수명 연장 및 개발 적응성을 향상시킬 수 있다. 또한, 이동에이전트를 이용하여 센서데이터를 획득할 경우 여러 어플리케이션에 따른 데이터 변환 및 정규화가 필요 없이 해당 프로그램에 맞게 데이터를 변환하여 수집할 수 있다.

향후에는 이동에이전트의 이주 경로의 안전성 확보를 위한 라우팅 방법 및 다양한 규칙들의 추가, 제어 및 관리가 가능한 능동규칙 시스템과의 연동에 요구되는 부품 기술들에 대한 연구가 필요하다.

참고문헌

[1] 황재각, 표철식, “USN 미들웨어 기술 개발 동향,” 한국전자파학회지, 제 19권, 제 6호, pp. 51-59, 2008.

[2] 원광호, 황태호, 김동순, 김태현, “WSN 기술 동향 및 응용기술,” 정보통신학회지, 제 25권, 제 10호, pp. 33-41, 2008
[3] Pratik K. Biswas, Hairong Qi, Yingyue Xu, “A Mobile-Agent-Based Collaborative Framework for Sensor Network Applications,” Mobile ad hoc and Sensor Systems(MASS)2006 IEEE, pp. 650 - 655, Oct. 2006.
[4] Konstantopoulos C. et al., ″Effective Determination of Mobile Agent Itineraries for Data Aggregation on Sensor Networks,″ IEEE Transactions on Knowledge and Data Engineering, Vol. 22, pp. 1679-1693, 2010.
[5] 최신일, 문석재, 엄영현, 국윤규, 정계동, 최영근, “분산 센서 네트워크에서 모바일 에이전트를 이용한 효율적인 데이터 수집,” 한국정보과학회 가을학술발표논문집, 제 33권, 제 2호(B), pp. 138-142, 2006.
[6] 이연식, 장민석, “센서 네트워크 응용을 위한 능동적 다중 에이전트 미들웨어 설계 및 구현”, 한국정보처리학회논문지(A), 제 18-A권, 제 4호, pp. 159-164, 2011
[7] 이정수, 최영춘, 이연식, “센서 네트워크 응용을 위한 네이밍 에이전트 설계,” 정보통신분야학회 합동학술대회논문집, pp. 147-150. 2009.
[8] 이연식, 이정수, “센서데이터 획득을 위한 이동에이전트 설계,” 정보처리학회 춘계학술발표대회논문집, 제 17권, 제 1호, pp. 1070-1073, 2010.
[9] 김대영, 성종우, 송형주, 김수현, “센서 네트워크 미들웨어 기술,” 전자공학회지, 제 32권, 제 7호, pp.800-814, 2005.
[10] Jong-Wan Yoon et al., ″Agent-based Sensor Network Middleware using Reputation Mechanism over Heterogeneous Network Environments,″ 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), pp. 373-376, 2010.
[11] Heimfarth T. et al., ″Experimental Analysis of a Wireless Sensor Network Setup Strategy Provided by an Agent-oriented Middleware,″ 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 820-826, 2010.
[12] 정의현, “무선 센서 네트워크를 위한 속성 기반 네이밍 구조,” 한국컴퓨터정보학회논문지, 제 12권, 제 4호, pp. 95-102, 2007.

저자소개

이연식(Yonsik Lee)

1982년 전남대학교 전자계산학과(이학사)
1984년 전남대학교 대학원 전자계산학과(이학석사)
1994년 : 전북대학교 대학원 전산응용공학전공 (공학박사)
1997년 ~ 1998년 : University of Missouri 교환교수
2004년 ~ 2005년 : Ohio State University 교환교수
1986년 ~ 현재 : 군산대학교 컴퓨터정보공학과 교수
※관심분야 : 객체지향 시스템, 능동규칙 시스템, 센서 네트워크 에이전트 미들웨어, USN 응용

이준호(Junho Lee)

2010년 군산대학교 컴퓨터정보과학과(이학사)
2010년~2012년 삼성소프트웨어 멤버십 정회원
2012년 군산대학교 대학원 컴퓨터정보공학과 (공학석사)
2012년 현재 삼성전자(주) 네트워크사업부 연구원
※관심분야 : USN, 능동규칙 에이전트 미들웨어, 영상처리