이동 에이전트 미들웨어를 이용한 중복 센서 데이터 제거

Elimination of the Redundant Sensor Data using the Mobile Agent Middleware

이 정 수* 이 연 식**
Jeongsu Lee Yonsik Lee

요 약

센서 네트워크 시스템의 센서 노드들은 싱크 노드와 무선으로 송수신하며 사람이 일일이 접근하기 힘든 방대한 지역의 센서 데이터를 획득 및 전송한다. 하지만 센서 노드들의 중복 센서 데이터의 비효율적인 반복 전송은 전체 시스템의 수명을 짧게 하고, 이때 발생하는 많은 양의 데이터들은 사용할 때 다시 선별해야 하는 번거로움이 있다.

본 논문에서는 네이밍 에이전트의 네임 스페이스의 메타 테이블로부터 제공되는 이주 대상 노드들을 차례로 방문하여, 사용자 조건에 따라 중복 센서 데이터를 제거하고, 용도 및 필요에 따라 센서 데이터를 수집 및 전송함으로써 센서 데이터의 과잉 송수신을 막고 전체 시스템의 수명을 늘릴 수 있는 이동 에이전트 미들웨어를 설계 및 구현한다. 또한, 실제 환경에서 발생할 수 있는 상황을 고려한 조건 및 제한들을 적용한 이동 에이전트를 이용한 실험을 통하여 중복 센서 데이터의 제거 및 데이터 수집의 효율성을 보이고, 향후 제안된 이동 에이전트 미들웨어에 능동 규칙을 탑재하거나 능동 규칙 시스템과의 연계를 통하여 다양한 능동적 센서 네트워크 응용에의 적용 가능성을 보인다.

ABSTRACT

The sensor nodes of sensor network system are capable of wireless communication with sink nodes. They also acquire and transmit sensor data in broad region where people cannot access easily. However, the transmission of redundant data from sensor nodes reduces the lifetime of the entire system and substantial amount of resulted data needs to be resorted before implementing them to the specific applications.

In this paper, the mobile agent middleware to eliminate the redundant sensor data is designed and implemented. In the proposed system, the mobile agent visits the destination sensor nodes according to the migration list offered by the meta table in the name space of the naming agent, eliminates the redundant sensor data corresponding to user condition, and acquires and transmits sensor data according to the purpose and needs. Thus, the excess transmission of the sensor data is avoided and the lifetime of the entire system can be extended.

Moreover, the experiments using the mobile agent middleware with the conditions and limitations that are possible in real situation are done to verify the successful elimination of the redundant sensor data and the efficiency of the data acquisition. Also, we show the potential applicability of the mobile agent middleware in various active sensor networks through the active rule based mobile agent middleware or the interaction with the active rule system.

☞ keyword : 이동 에이전트 미들웨어(Mobile Agent Middleware); 센서 네트워크(Sensor Network); 중복 센서 데이터 제거(Redundant Sensor Data Elimination)

1. 서 론

센서 네트워크상의 센서 노드는 다양한 실세계 정보를 센싱 할 수 있는 센서와 이를 처리할 수 있는 최소한의 프로세싱 리소스 그리고 다른 센서 노드와의 협동적인 데이터 처리를 위한 무선 네트워크 기능을 포함하고 있다[1,2,3,11].이러

* 준 회 원 : 군산대학교 대학원 컴퓨터정보공학과(공학석사)
jsjs2000@kunsan.ac.kr
*** 정 회 원 : 군산대학교 컴퓨터정보공학과 교수
yslee@kunsan.ac.kr(교신저자)
[2011/01/26 투고 - 2011/02/15 심사 - 2011/04/15 심사완료]

한 센서 네트워크 환경에서는 제한된 자원과 전력 등의 한정된 환경 내에서 통신하기 때문에, 중복된 데이터의 전송이나 비효율적인 센싱 주기 등은 센서 노드의 수명을 짧게 만들고, 시스템 전체의 성능에 악영향을 끼친다. 광범위한 지역의 수많은 센서 노드들이 수집하는 센서 데이터는 이웃한 근처 센서 노드와 중복되는 데이터가 발생할 확률이 높으며, 센서 네트워크 환경의 특성상 이러한 불필요한 데이터를 모두 주기적으로 전송하는 것은 시스템의 효율적인 면이나 수명 연장을 위해서도 개선되어야 할 사항이다[1,2,4,11,13]. 또한 센서 노드들이 일제히 싱크 노드를 향해 데이터를 보내는 상황을 피하고, 인접 노드의 중복된 데이터를 필터링 하여 데이터를 전송할 수 있는 시스템의 개발이 필요하다[1,2,4,6,11-3].

이에 본 논문에서는 센서 노드를 이주하며 중복된 데이터를 필터링 하여 싱크노드로 전송하며, 능동 규칙의 탑재 및 능동 규칙 시스템과의 연동이 가능한 이동 에이전트 미들웨어를 설계 구현한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서 기 개발한 멀티 에이전트 시스템의 능동적 센서 네트워크 연동을 위한 확장 구조와 에이전트 미들웨어의 특징을 설명하고, 3장에서는 본 논문에서 구현한 이동 에이전트 미들웨어 플랫폼, 이동 에이전트 구조 및 이주 방식, 중복 데이터 처리 알고리즘을 제안한다. 4장에서는 실제 센서 노드 및 센서 게이트웨이와 발생 가능한 조건 및 제한들을 적용한 이동 에이전트를 이용한 실험을 통하여 중복 데이터 제거 모듈을 탑재한 이동 에이전트 미들웨어의 응용 가능성 및 효율성을 보이고, 마지막 5장에서 결론 및 향후 연구 과제를 제시한다.

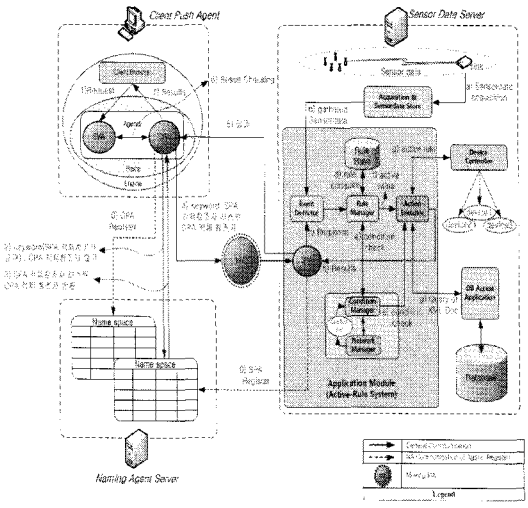
2. 멀티 에이전트 시스템 및 에이전트 미들웨어

2.1 멀티 에이전트 시스템

각 에이전트들의 상호간 협력과 상호 보완적 관계를 통해 분산 환경의 정보 공유 및 통합을 용이하게 하는 멀티에이전트 시스템은 센서 네트워크 환경에서 기존의 정보들뿐만 아니라 SDS (Sensor Data Server)를 통한 센서 정보들까지 검색하여 사용할 수 있는 형태로 제공한다. 멀티 에이전트 시스템은 네트워크 트래픽 감지 및 능동적인 데이터 전달방식을 제공하는 푸시 에이전트, 메타 데이터 형식의 인덱스로 구성되어 센서 데이터 및 객체화된 센서 네트워크상의 구성 장치 정보를 테이블로 저장하여 보다 정확한 관리가 되도록 설계된 네이밍 에이전트, 클라이언트 푸시 에이전트와 연계되어 시스템 자원 관리 및 필터링 기능을 이용해 센서 네트워크와의 연동을 유연하게 설정 가능한 시스템 모니터링 에이전트, 네트워크 트래픽 감소 및 순회 검색 수행 시간의 단축을 통한 효율 향상을 유도하는 이동 에이전트 등으로 구성된다[3-5,7,8]. 특히, 네이밍 에이전트는 모든 에이전트들의 위치정보, 이름, 등록객체 및 검색 키워드 등의 메타 데이터와 각 센서 네트워크 구성요소(서버, 싱크 및 센서 노드 등)의 위치, 명칭 및 속성 정보를 네임스페이스에 저장하고 이들을 통합 운영, 관리 및 서비스 하는 역할을 수행함으로써 분산된 여러 시스템을 하나의 시스템으로 통합하고, 여러 서버와 데이터를 관리함으로써 이를 이용한 다양한 응용을 가능하게 하는 주요 구성 요소로써, 이동 에이전트를 통해 특정 명령의 전달 및 규칙을 실행할 수 있도록 하는 등의 센서 네트워크 미들웨어 기능을 지원하는 주요 역할을 수행한다[5,8-10]. 다음 (그림 1)은 센서 네트워크 환경과 연동된 멀티 에이전트 시스템의 구조를 나타낸다.

2.2 에이전트 미들웨어

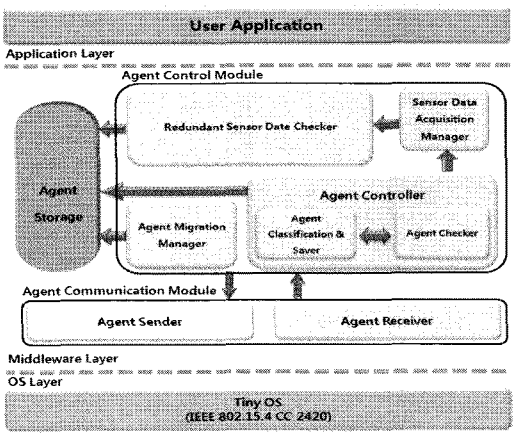
기존 센서 네트워크 관련 응용 개발은 운영체제나 센서 노드의 저수준의 기능을 이용하여 특정 응용에 의존적으로 이루어지고 있는[1,2,3,5,11,12] 실정이므로, 사용자의 요구에 따라 동적으로 적용 가능한 유연한 구조의 센서 네트워크



(그림 1) 멀티 에이전트 시스템 구조

미들웨어에 관한 많은 연구들이 진행되고 있다[1,3,10-13]. 센서 네트워크 미들웨어는 개발 시 여러 특징들을 고려해야 하지만[1,2,13], 본 논문에서는 주위 상황에 맞는 데이터 수집을 위한 자동 설정기능, 센서 노드들의 신뢰성을 위한 자원 활용 조절기능, 속성(메시지, 규칙) 기반 통신이 요구되는 데이터 중심의 통합적 처리기능과 이벤트를 중심으로 데이터를 처리하는 publish/subscribe형태의 기능적 특징만을 고려한다.

본 논문에서 제안하는 이동 에이전트 미들웨어는 위의 특징들을 수용하기 위하여 원격에서 이동 에이전트를 이용하여 사용자의 개입 없이 자동으로 센서 노드들이 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 지원한다. 또한 사용자의 요구에 의한 이벤트 중심의 능동적 데이터 처리 방식으로 설계 및 구현함으로써, 향 후 능동 규칙 탑재뿐만 아니라 능동규칙 시스템과의 무리 없는 연동이 가능하도록 한다.



(그림 2) 이동 에이전트 미들웨어 플랫폼 구조

3. 센서 네트워크 연동 이동 에이전트

3.1 이동 에이전트 미들웨어 플랫폼

센서 네트워크상의 싱크 노드 및 센서 노드들은 이동 에이전트를 효율적으로 이주 시키고, 각 노드에서 이동에이전트의 역할을 수행하기 위해 각각의 노드에 이동 에이전트 플랫폼을 필요로 한다. 본 논문에서 구현한 이동 에이전트 미들웨어 플랫폼의 구조는 다음 (그림 2)와 같다.

구성요소는 이동 에이전트의 송수신을 담당하는 *Agent Communication Module*과 에이전트의 이주, 데이터 수집, 중복 데이터 제거 등의 작업을 수행하는 *Agent Control Module*로 이루어진다.

* ***Agent Communication Module***

- *Agent Sender*: Dest_Node를 통해 다른 노드나 싱크노드로 에이전트를 송신
- *Agent Receiver*: 이동 에이전트 수신 및 해당 이동 에이전트의 ID등을 확인하기 위한 Agent Checker 호출

* ***Agent Control Module***

- ***Agent Migration Manager***: 이동 에이전트로부터 이주 리스트(이주 대상 전체 노드 목록)를 받아 에이전트 이주를 관리하며, 다음 방문 노드를 선택하고 해당 노드를 이동 에이

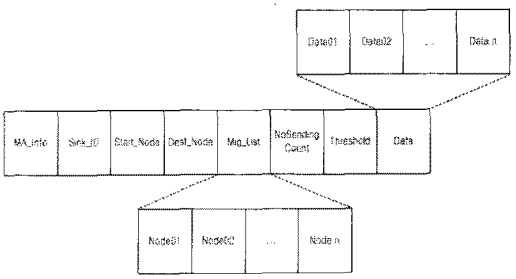
전트 이주 시 Dest_Node에 저장

- *SensorData Acqusition Manager*: 센서 데이터 수집 관리
- *Agent Checker*: 이주해 온 이동 에이전트의 ID를 사용하여 이주의 정확성 확인
- *Agent Classification & Saver*: 이동 에이전트를 특징에 따라 분류, 이전 노드의 센서 데이터를 획득, 이주 리스트를 *Agent Checker*에 송신, 에이전트를 저장소에 저장
- *Redundant Sensor Data Checker*: 중복되는 센서 데이터를 제거하기 위한 모듈(행 후 이동 에이전트에 탑재하여 수행할 수 있는 능동규칙의 조치 실행기 역할)
- *Agent Storage*: 이주해 온 이동 에이전트 저장 및 다른 노드로 이주하기 직전 재조정 후 이주

3.2 이동 에이전트 구조

이동 에이전트는 이주 리스트를 가지고 대상 센서 노드들로 순차적으로 이주하며 센서 데이터 획득, 전송 및 요구되는 처리를 수행한다. 이동 에이전트는 최초 싱크 노드로부터 이주하며 센서 노드에 전송된 후에 이동 에이전트 정보인 MA_Info와 도착노드인 Dest_Node가 올바른지 확인하고, 센서 데이터 수집 결과를 다른 노드나 싱크 노드로 전송한다. 이동 에이전트가 가지고 이동할 센서 데이터들의 저장을 위하여 Data 필드를 설정하고, 이들 값들을 이용하여 중복 센서 데이터를 제거하고 요구에 적합한 데이터만을 싱크 노드로 보내도록 설계 구현 한다. 중복 제거 및 향 후 능동 규칙 탑재를 위한 이동 에이전트의 구조는 다음 (그림 3)과 같다.

(그림 3)에서 MA_Info는 해당 에이전트를 확인 후 수신할 수 있도록 하는 이동 에이전트의 식별자이다. Sink_ID는 해당 이동 에이전트의 이주 시작 및 복귀를 위한 싱크 노드의 ID이다. Start_Node는 현재 이동 에이전트가 어디로부터 온 것 인지를 나타내고, Dest_Node는 목적지 노드로써, 센서 노드에서는 이 Dest_Node가 자신인



(그림 3) 이동 에이전트 구조

지 확인 후 에이전트를 수신한다. Agent Migration Manager는 Mig_List의 내용에 따라 다음 방문할 노드를 결정한다. Data는 센서 노드에서 수집한 센서 데이터를 저장하는 공간으로서 확장성을 가지며, Threshold와 NoSendingCount는 규칙 처리에 필요한 임계값이나 필요한 횟수를 저장하기 위한 부분이다.

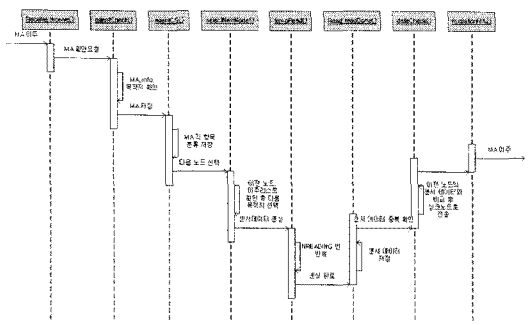
3.3 이동 에이전트 이주

이동 에이전트의 이주는 에이전트가 센서 노드에서 정해진 조치를 수행하면, *Agent Migration Manager*가 Mig_List를 참조하여 Dest_Node를 선택하고 Agent Sender를 통해 이루어진다. 이주 시 다음 이주 대상 노드의 선택은 이동 거리, 정확도, 중요도, 또는 결로 회피방법 등 다양한 라우팅 방법을 적용하여 결정될 수 있지만 본 논문에서는 이주방식에 대해서만 구현한다. 다음 (그림 4)는 센서 네트워크상에서의 이동 에이전트 이주 알고리즘이다.

에이전트의 이주는 message_t 구조체로 표현되는 TinyOS의 액티브 메시지 구조를 통해 이루어지며, 이는 MAC 계층의 프레임 제어와 패킷의 순서 번호 및 주소 체계 필드를 가지는 헤더와 유효성 검사를 위한 풋터를 제외하고 최대 116바이트를 사용자 데이터 영역으로 사용할 수 있다. 이동 에이전트의 이주는 이 영역을 사용하여 mobileAgent_t 구조체를 message_t 액티브 메시지에 탑재하여 보냄으로써 이루어진다. 다음 (그림 5)는 이동 에이전트의 이주 과정을 나타낸다.

```
=======================================
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);
   IF (call AMSend.send(nextNodeNum,
   & send_data, sizeof ma_data) == SUCCESS)
   THEN sendbusy = TRUE;
=======================================
```

(그림 4) 이동 에이전트 이주 알고리즘



(그림 5) 이동 에이전트 이주 과정

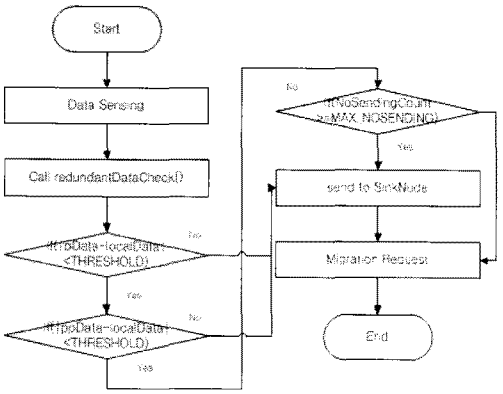
에이전트의 이주는 message_t 구조체로 표현되는 TinyOS의 액티브 메시지 구조를 통해 이루어지며, 이는 MAC 계층의 프레임 제어와 패킷의 순서 번호 및 주소 체계 필드를 가지는 헤더와 유효성 검사를 위한 풋터를 제외하고 최대 116바이트를 사용자 데이터 영역으로 사용할 수 있다. 이동 에이전트의 이주는 이 영역을 사용하여 mobileAgent_t 구조체를 message_t 액티브 메시지에 탑재하여 보냄으로써 이루어진다. 다음 (그림 5)는 이동 에이전트의 이주 과정을 나타낸다.

최초 대기모드에서 에이전트가 이주되면 Receive.received() 메서드가 호출되고, 에이전트가 올바르게 도착한 것인지 확인하기 위하여 agentCheck() 메서드를 호출한다. agentCheck() 메서드가 호출되면서 인자로 넘긴 msg에서 에이전트 ID인 MA_Info와 목적지 노드인 Dest_Node를 확인한다. Dest_Node는 이전 노드에서 이동할 때의 목적지 노드로써 현재 노드와 비교하여 동일한지를 확인한다. 올바르게 도착한 에이전트이면 agentCS()를 호출하고, 아니면 종료하고 대기모드로 전환한다. agentCS() 메서드는 이동 에이전트를 분류 및 저장하기 위한 메서드로 massge_t 형태로 온 메시지를 다시 MobileAgent_t 형태로 저장한다.

Start_Node에 현재 노드의 번호를 부여하고, Dest_Node는 selectNextNode()에서 결정하도록 비워둔다. Mig_list[] 에서 현재 노드의 위치를 확인하고 다음 이주 대상 노드의 번호를 nextNodeNum으로 반환하고, 이를 MobileAgent_t 메시지의 Dest_Node에 저장한다. tempRead()는 센서 데이터 획득 메서드로써 헤더파일 내의 NREADINGS 횟수만큼 센서 데이터를 획득하며, 센서 데이터 획득 후 Read.readDone() 이벤트가 발생된다. Read.readDone()에서 획득한 센서 데이터는 이동 에이전트의 Sensor_Data[]에 저장하고, NREADINGS 횟수만큼 수행 후 데이터 중복 처리 메서드인 dataCheck() 메서드를 호출한다. 중복 데이터 제거(3.4절) 모듈 실행 후 중복이 아닐 경우 현재 노드의 센서 데이터 값을 싱크 노드로 전송 요청하고, 에이전트가 다음 노드로 이동 할 수 있도록 migrationMA() 메서드를 호출하며, 만일 중복 데이터로 처리되면 전송하지 않고, migrationMA()를 통해 nextNodeNum 노드로 에이전트를 이주시킨다. 에이전트가 이주한 후 노드는 다시 대기상태로 전환된다.

3.4 중복 데이터 제거

센서 데이터의 특성상 인접한 센서 노드의 데이터는 서로 비슷하거나 같을 경우가 많으므로, 이러한 센서 데이터를 모두 싱크 노드로 전송함



(그림 6) 중복 데이터 제거 과정

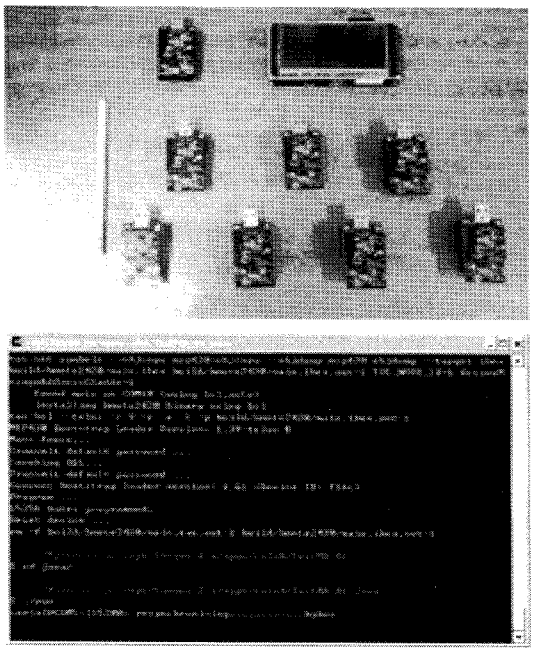
으로써 생기는 데이터 과부하, 센서 노드의 수명 단축 및 대역폭 오버헤드 등의 문제를 해결하기 위하여 중복 데이터 제거가 필요하다. 본 논문에서는 이동 에이전트가 센서 노드로 이주를 완료한 후 센서 데이터를 획득하면 중복 데이터 제거 알고리즘이 실행되도록 구현한다.

센서 데이터의 중복을 제거하기 위해서 이동 에이전트가 이전 센싱 데이터들을 모두 가지고 이동하며, 서로 다른 형태의 센서 데이터들의 중복을 제거하기 위한 알고리즘을 적용할 수 있지만, 이는 에이전트의 크기가 점점 커짐에 따른 부하와 이동 시간의 지연으로 인한 트래픽 부담이 발생한다. 또한 많은 센서 데이터를 한 번에 처리할 만큼 센서 노드의 처리 능력이 양호하지 않은 실정이다. 따라서 본 논문에서 구현한 이동 에이전트 미들웨어는 두 개의 Data 필드에 직전의 두 개의 센서 데이터를 저장 후 이동하여 이들 값을 기반으로 중복 센서 데이터를 처리하고, 센서 데이터들이 싱크 노드로 송신되지 않는 횟수가 임의의 수 이상이 될 경우 지역성 확인을 위하여 해당 노드의 센서 데이터를 싱크 노드로 무조건 전송하기 위한 카운터인 NoSendingCount를 두어 다양한 형태의 센서 데이터 값들의 분포 환경에도 일반적으로 적용할 수 있도록 하였다. 다음 (그림 6)은 이동 에이전트의 중복 데이터 제거 과정을 나타낸다.

```
=======================================
INPUT : pData, ppData, localData,
        NoSendingCount
OUTPUT :
PROCESS :
IF ( | localData - pData | >= THRESHOLD)
THEN
  sendSinkNode(SINKNODENUM,
  local_data);
  NoSendingCount ++;
ELSE
 IF ( | localData - ppData | >= THRESHOLD)
 THEN
   sendSinkNode(SINKNODENUM, local_data);
   NoSendingCount++;
  IF (NoSendingCount >= MAX_NOSENDING)
  THEN
    sendSinkNode(SINKNODENUM, local_data);
    NoSendignCount = 0;
=======================================
```

(그림 7) 중복 센서 데이터 제거 알고리즘

이동 에이전트가 가지고 온 인접 노드의 센서 데이터를 pData와 ppData에 저장하고 현재 노드의 센서 데이터를 localData에 저장하며, NoSendingCount는 중복 데이터를 제거한 횟수를 나타내는 카운터로 싱크 노드로 센서 데이터를 보내지 않는 노드에서 1씩 증가한다. 중복 데이터 제거 과정은 우선 pData, ppData 및 localData를 저장 후 중복 제거 모듈을 호출한다. 중복 제거 모듈은 pData와 ppData를 현재 노드의 센서 데이터 값인 localData값과 비교한 차이를 헤더파일에 정의된 임계값(THRESHOLD)과 비교하여 임계값 이상이면 현재 노드의 센서 데이터 값을 싱크 노드로 전송 요청하고, 임계값 미만이면 중복 데이터로 처리하여 전송하지 않고, NoSendingCount를 1 증가시킨 후 이동 에이전트 이주를 요청한다. 또한, NoSendingCount를 확인하여 데이터를 전송하지 않은 횟수가 MAX_NOSENDING 이상이면 현재 센서 노드의 데이터를 무조건 싱크 노드로 전송하고, 미만이면 다음 노드로 이동 에이전트를 이주시킨다. 다음 (그림 7)은 이동 에이전트의 중복 센서 데이터 제거 알고리즘을 나타낸다.

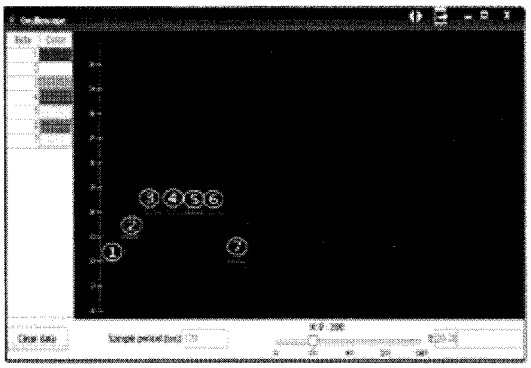


(그림 8) 실행 화면

위 알고리즘 적용 시 THRESHOLD와 MAX_NOSENDING은 센서 분포 정도, 센서 데이터 값들의 변화 정도 및 해당 지역의 특성에 알맞은 값으로 최적화함으로써 가장 효율적으로 수행될 수 있다.

4. 실험 및 평가

제시된 이동 에이전트 미들웨어의 실험은 실제 센서 노드인 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. 사용 OS는 TinyOS-2.x를 사용하고, 윈도우 상에서 TinyOS 개발을 위해 Cygwin 툴을 사용하였다. 실험 데이터는 온도 데이터 이며 싱크 노드를 포함하여 총 8개의 노드로 실험을 하였다. 다음 (그림 8)은 실험 센서 노드들과 최초 실행 화면이다.



(그림 9) 실험 결과 1

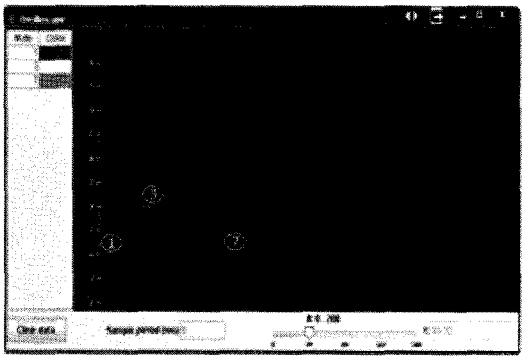
각 센서 노드에 이동 에이전트 플랫폼 이미지를 업로드 한 후 사용자 viewer를 실행 한다. 이동 에이전트의 이동은 viewer 를 통해 확인 가능하나 각 센서 노드에 부착된 Led로도 이동 에이전트의 이동과 센서 데이터 전송 여부를 확인 할 수 있도록 하였다.

실험 중 센서 데이터 값들은 시간에 따라 변할 수 있으므로 중복 데이터 처리 알고리즘을 적용하기 위하여 센서 데이터 값들을 노드 내부에 고정 값으로 저장해 두고 실험하였다.

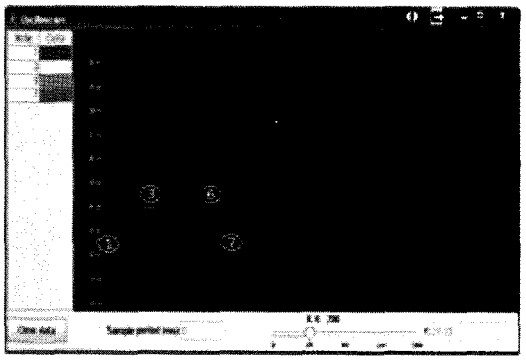
다음 (그림 9)의 실험 결과 1은 이동 에이전트가 각 노드를 이주하면서 중복 데이터를 처리하지 않고 획득한 센서 데이터를 싱크 노드로 전송한 화면으로, 센서 노드들을 ①번에서 ⑦번 까지 오름차순으로 순차적 이주한 결과를 보인다.

(그림 10)의 실험 결과 2는 이동 에이전트에 임계값인 THRESHOLD를 2로 설정한 중복 제거 모듈을 탑재하여 실험한 결과 화면으로, 직전 노드와의 온도 차이가 주어진 임계값 이상인 경우만 싱크 노드로 전송하도록 한 결과이다.

이러한 실험 2의 환경에서는 인근 노드들의 센서 데이터 값들이 연속적으로 임계값 이하를 유지하고 있을 때, 이들 값들이 중복으로 처리되어 넓은 영역의 센서 데이터 값들이 전송되지 않기 때문에 지역적 특성을 파악하기 어려울 수 있다. 따라서 일정 횟수 이상으로 중복 처리가 계속되면 센서 데이터 값을 싱크 노드로 보낼 필요가



(그림 10) 실험 결과 2



(그림 11) 실험 결과 3

있다. 이를 해결하기 위하여 NoSendingCount를 두어 그 값 이상으로 중복 처리가 발생하면 무조건 센서 데이터를 싱크 노드로 보내도록 구현하였다. 다음 (그림 11)의 실험 결과 3은 실험에서 MAX_NOSENDING값을 3으로 하여 이동에이전트를 이용한 센서 데이터 중복을 처리한 결과 화면이다.

(그림 11)의 실험 결과 3의 ③번 노드로 부터의 과정은 다음과 같다. ③번 노드에 이동 에이전트가 도착하여 중복 처리 알고리즘을 실행하면, ②번 노드의 센서 데이터 값인 pData와 ①번 노드의 값인 ppData 등 두 개의 값들과 ③번 노드에서 획득한 센서 데이터 값을 비교하여 임계값인 THRESHOLD이상의 값이 있으므로 ③번 노드의 값을 싱크로 전달한다. 전송 후 ③번 노드의 데이터는 pData에, ②번 노드의 데이터는 ppData에 저장하고, ②번 노드에서 1로 할당된 NoSending Count값을 0으로 변경한 후 이동 에이전트에 적재 한다. 그 후 ④번부터 ⑥번 노드까지는 센서 데이터 값들이 임계값 미만이므로 싱크 노드로 전달하지 않고 NoSendingCount값만 ⑥번 노드에서 3이 되며, 알고리즘 실행 시 MAX_NOSENDING값을 3으로 설정하였으므로 ⑥번 노드에서 센서 데이터 값을 싱크 노드로 보내고, NoSendingCount를 0으로 바꾼 뒤 ⑦번 노드로 이주한다.

위의 실험 결과들에서 알 수 있듯이, 이동 에이전트에 탑재한 중복 처리를 위한 능동규칙에 임계값과 NoSendingCount값 등을 적용함으로써, 센서 노드들의 분포에 따른 센서 데이터 값들의 변화 형태를 손상시키지 않고 불필요한 중복 센서 데이터 값들의 전송을 줄일 수 있는 제안 방법의 유효성을 증명한다.

5. 결론 및 향후 연구과제

일반적인 센서 노드들의 비효율적인 반복적 데이터 전송은 전체 시스템의 수명을 짧게 하고, 많은 양의 데이터들은 이를 활용할 때 다시 선별해야 하는 번거로움이 있다. 이에 본 논문에서는 센서 노드를 이주하며 중복 데이터를 제거하고, 사용자의 필요에 따라 센서 데이터를 수집 및 전송하여 중복된 센서 데이터의 과잉 송수신을 막고, 전체 시스템의 수명 연장을 기대할 수 있는 이동 에이전트 미들웨어를 설계 및 구현하였다.

제안한 이동 에이전트 미들웨어 시스템은 기존의 정보 검색용 다중 에이전트 시스템의 네이밍 에이전트의 네임 스페이스의 메타 테이블을 센서 네트워크 시스템에 적합하도록 변경하고, 그로부터 제공되는 이주 대상 노드들을 차례로 방문하여 사용자 조건에 따라 중복 센서 데이터를 제거하고, 용도 및 필요에 따라 센서 데이터를 수집 및 전송함으로써 센서 데이터의 과잉 송수신을 막고 전체 시스템의 수명 연장을 유도하도록 한다. 또한, 실제 센서 노드 Hmote2420 및 X-Hyper320WN 센서 게이트웨이와 실제 환경에

서 발생할 수 있는 상황을 고려한 조건 및 제한들을 적용한 이동 에이전트를 이용한 실험을 통하여 중복 데이터 제거 모듈을 탑재한 이동 에이전트 미들웨어의 응용 가능성 및 효율성을 보였다. 제안된 이동 에이전트 미들웨어 시스템은 향후 다양한 환경 및 상황에 적합한 능동 규칙 탑재 및 능동 규칙 시스템과의 연계를 통하여 다양한 능동적 센서 네트워크 응용에의 적용 가능성을 보장한다. 향후 추가적으로 센서 네트워크 응용을 위한 능동 규칙 설계, 능동 규칙 탑재 및 경량화, 능동 시스템과의 연동 및 효율적 라우팅 방법 등이 요구된다.

Acknowledgement

본 논문은 2009년도 정부의 재원으로 한국연구재단(No. 2009-0074891)의 지원과 교육과학기술부 및 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임.

참 고 문 헌

[1] 황재각, 표철식, "USN미들웨어 기술 개발 동향," 한국전자파학회지, 제 19권, 제 6호, pp. 51-59, 2008.

[2] 원광호, 황태호, 김동순, 김태현, "WSN 기술 동향 및 응용기술," 정보통신학회지(정보와 통신), 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 adhoc 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] Yonsik Lee, Kwangjong Kim, "Optimal Migration Path Searching using Path Adjustment and Reassignment for Mobile Agent," Proc. of 4th International Conference on Networked Computing and Advanced Information Management(NCM2008), pp. 564-569, Sep. 2008.

[6] 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.

[7] 남진우, 정영지, "센서 네트워크에서 헬스케어 이동성 에이전트 모듈 설계," 멀티미디어학회논문지, 제 11권, 제 4호, pp. 544-553, 2008.

[8] 김광종, "객체 다중 복제 기반의 멀티캐스트 이동 에이전트 시스템," 군산대학교 대학원 박사학위논문, 2004.

[9] 이정수, 최영춘, 이연식, "센서 네트워크 응용을 위한 네이밍 에이전트 설계," 정보통신분야학회 합동학술대회논문집, pp. 147-150. 2009.

[10] 이연식, 이정수, "센서 데이터 획득을 위한 이동 에이전트 설계," 정보처리학회 춘계학술박표대회논문집, 제 17권, 제 1호, pp. 1070-1073, 2010.

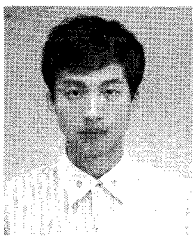
[11] 김대영,성종우, 송형주, 김수현, "센서 네트워크 미들웨어 기술," 전자공학회지, 제 32권, 제 7호, pp.800-814, 2005.

[12] 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.

[13] Heimfarth T. et al., "Experimental Analysis of a Wireless Sensor Network Setup Strategy

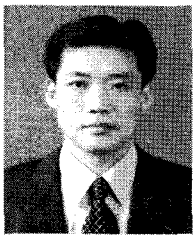
Provided by an Agent-oriented Middleware," 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 820-826, 2010.

◐ 저 자 소 개 ◑



이 정 수

2009년 군산대학교 컴퓨터정보공학과(공학사)
2011년 군산대학교 대학원 컴퓨터정보공학과 (공학석사)
관심분야 : 지능형 에이전트 미들웨어, 센서 네트워크, 능동시스템
E-mail : jsjs2000@kunsan.ac.kr



이 연 식

1982년 전남대학교 전자계산학과 졸업(학사)
1984년 전남대학교 대학원 전자계산학과 졸업(이학석사)
1994년 전북대학교 대학원 전산응용공학전공 졸업(공학박사)
1997년~1998년 University of Missouri(Kansas City) 교환교수
2004년~2005년 Ohio State University 교환교수
1986년~현재 군산대학교 컴퓨터정보공학과 교수
관심분야 : 객체지향시스템, 능동시스템, 센서 네트워크 에이전트 미들웨어, USN 응용
E-mail : yslee@kunsan.ac.kr