

센서 네트워크 응용을 위한 능동적 다중 에이전트 미들웨어 설계 및 구현

이 연 식[†] · 장 민 석^{††}

요 약

본 논문에서는 다양한 센서 네트워크 응용을 위한 능동적 에이전트 미들웨어를 제안한다. 이를 위하여, 이동 에이전트가 네이밍 에이전트의 네임 스페이스의 메타 테이블로부터 제공되는 이주 대상 센서 노드들을 차례로 방문하여, 사용자 조건, 용도 및 필요에 따른 다양한 능동규칙을 통하여 센서 데이터를 수집 및 전송하며, 임의의 발생 사건들(센서 데이터 값 및 시간 등의 변화)에 상응하는 조치들을 직접 실행할 수 있는 능동규칙 탑재 이동 에이전트를 설계 구현한다. 또한, 기존의 센서 데이터 관련 규칙 및 상황 데이터베이스 시스템과의 연동을 통하여 다양한 능동적 센서 네트워크 응용에의 적용 가능성을 보인다.

키워드 : 에이전트 미들웨어, 능동규칙 시스템, 센서 네트워크 응용

Design and Implementation of the Active Multi-Agent Middleware for the Sensor Network Application

Yonsik Lee[†] · Minseok Jang^{††}

ABSTRACT

In this paper, we suggest the active multi-agent middleware for the sensor network application. For this, firstly we design and implement the active rule based mobile agent middleware. The mobile agent in the proposed system visits the destination sensor nodes according to the migration list offered by the meta table in the name space of the naming agent, acquires and transmits sensor data according to the purpose and needs through the active rules, and directly executes the actions corresponding to the optional events(changed sensor data and/or time etc.). And then, we show the potential applicability of the active rule based mobile agent middleware in various active sensor networks through the interaction with the rule base system and context database system.

Keywords : Agent Middleware, Active Rule System, Sensor Network Application

1. 서 론

USN 상에서의 다양한 응용 개발을 위해서는 센서 노드 하드웨어, 네트워크 프로토콜과 같은 하부구조에 대한 상세한 지식이 요구되며, 또한 센서 노드 내의 제한된 자원을 고려한 설계 패러다임이 필요하다. 이러한 문제를 해결하기 위하여 센서 네트워크 응용과 센서 노드 하드웨어의 중간에 위치하여 둘 사이의 유연한 통합을 지원하는 다양한 USN

미들웨어가 요구된다. 또한, USN 관련 상황인식과 반응, 센서 간 정보 협력 및 자율적인 대응체계 구축 등은 복잡적, 상승적으로 발생하므로[1,2], 향후 USN 환경에서 변화에 따른 실시간 반응 원격 자율 제어, 예외 상황 대책 기능, 부수 작업 자율구동 등 보다 고도의 응용들을 효율적이고 효과적으로 개발하기 위해서는 코드 및 프로시저의 능동성, 자율성, 이동성 등을 지원하는 미들웨어 기반이 필수적으로 요구된다[1,2,5,8]. 이 개념을 확장시킨 것이 능동규칙 탑재 및 능동규칙 시스템 연동 에이전트 모델로서 응용프로그램뿐만 아니라 미들웨어나 운영체제를 구성하는 모듈들을 자동적으로 업데이트 할 수 있도록 하는 구조의 미들웨어이다 [1,8,11]. 이는 센서 네트워크를 관리하고 정보 취득 명령을 내리며 보고를 접수하여 사용자 및 관리자에게 표현하고 상응하는 조치를 수행하도록 하는 사용자노드(센서 데이터 서버 역할 포함)는 이러한 능동성과 자율성을 갖는 이동 소프

※ 본 논문은 2010년도 교육과학기술부 및 한국산업기술진흥원의 지역혁신인력양성사업과 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(No.2009-0074891)을 받아 수행된 연구임.

† 중신회원 : 군산대학교 컴퓨터정보공학과 교수

†† 정 회 원 : 군산대학교 컴퓨터정보공학과 교수

논문접수 : 2011년 2월 16일

수정일 : 1차 2011년 4월 1일

심사완료 : 2011년 4월 4일

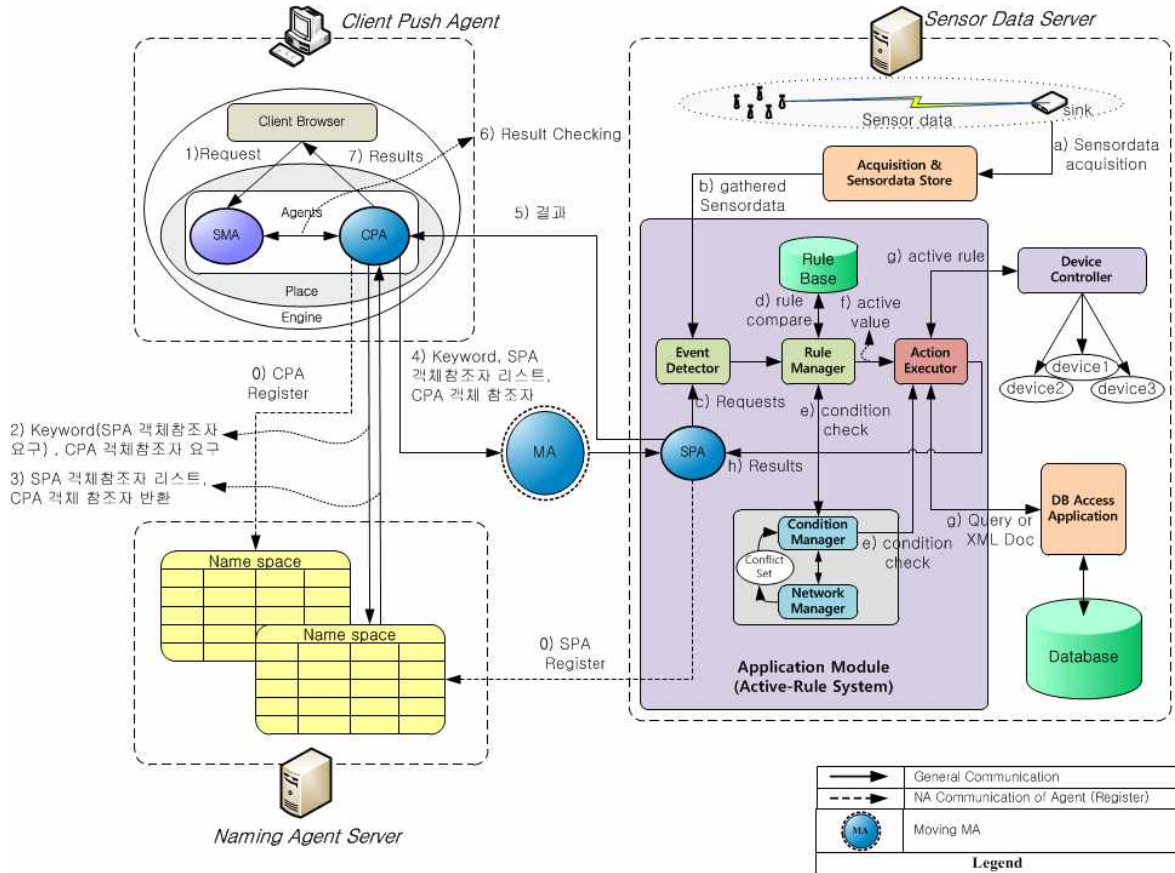
트웨어 모듈의 탑재 및 운용을 가능하게 하며, 센서로부터 수집되는 대량의 센싱 데이터를 정제, 요약하여 의미있는 데이터를 생성하고 기존 응용시스템과의 연동 및 통합을 담당하는 새로운 형태의 미들웨어 시스템이다. 따라서 본 논문에서는 이동 에이전트가 멀티 에이전트 시스템의 네이밍 에이전트의 메타 테이블을 이용하여 이주 대상 센서 노드들을 방문하고, 다양한 능동규칙을 적용하여 센서 데이터를 수집 및 전송하며, 임의의 발생 사건들에 상응하는 조치들을 직접 실행할 수 있는 능동규칙 탑재 이동 에이전트 미들웨어를 설계 구현한다. 또한, 기존의 센서 데이터 관련 규칙 및 상황 데이터베이스 시스템과의 연동을 통하여 다양한 능동적 센서 네트워크 응용에의 적용 가능성을 보인다.

본 논문의 구성은 다음과 같다. 2장에서 능동적 센서 네트워크와 연동된 기 구축된 멀티 에이전트 시스템의 확장 구조와 설계 대상 능동규칙 시스템의 특징을 설명하고, 3장에서는 구현한 센서 네트워크 연동 이동 에이전트 미들웨어 플랫폼의 구조와 이동 에이전트 이주 및 능동규칙 실행 방법을 설명한다. 4장에서는 실제 센서 노드 및 센서 게이트웨이를 대상으로 중복 데이터 제거를 위한 능동규칙을 탑재한 이동 에이전트의 이주 및 실행 여부를 실험을 통하여 보임으로써 제안 시스템의 응용 가능성 및 효율성을 보이고, 5장에서 결론을 제시한다.

2. 센서 네트워크 연동 멀티 에이전트 시스템

본 논문에서 제안하는 능동적 에이전트 미들웨어는 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, 능동규칙 탑재 및 처리를 통하여 데이터가 통신의 주체가 되는 새로운 방식의 통신 등을 지원한다. 또한 사용자의 요구에 의한 이벤트 중심의 능동적 데이터 처리 방식으로 설계 및 구현함으로써, 향후 다양한 센서 네트워크 응용들과 능동규칙 시스템과의 무리없는 연동 가능성을 제공한다. 제안 시스템은 기 개발[7,9]한 검색용 멀티 에이전트 시스템을 (그림 1)과 같이 센서 데이터 서버(Sensor Data Server)를 통하여 센서 네트워크와 연동된 구조로 확장하여 구현한다.

센서 네트워크 환경에서의 멀티 에이전트 시스템은 각 에이전트들의 상호 협력 및 보완 관계를 통해 분산 환경의 정보 공유 및 통합을 용이하게 하며, 센서 데이터 서버의 센서 데이터 정보들까지 검색하여 사용할 수 있는 형태로 구성된다. 멀티 에이전트 시스템은 네트워크 트래픽 감지 및 능동적인 데이터 전달방식을 제공하는 푸시 에이전트, 메타 데이터 형식의 인덱스로 구성되어 센서 데이터 및 객체화된



(그림 1) 센서 네트워크 연동 멀티 에이전트 시스템 구조

센서 네트워크상의 구성 장치 정보를 테이블로 저장하여 정확히 관리하도록 설계된 네이밍 에이전트, 시스템 자원 관리 및 필터링 기능을 이용해 센서 네트워크와의 연동을 유연하게 설정 가능한 시스템 모니터링 에이전트, 네트워크 트래픽 감소, 순회 검색 수행 시간의 단축 및 센서 노드에서의 규칙 실행을 유도하는 이동 에이전트 등으로 구성된다 [5,7,9]. 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 네임 스페이스에 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 요소으로써, 이동 에이전트를 통해 특정 명령의 전달 및 규칙을 실행할 수 있도록 하는 등의 센서 네트워크 미들웨어 기능을 지원하는 주요 역할을 수행한다 [7,9,10]. 또한, 센서 데이터 서버의 응용 모듈인 능동규칙 시스템은 사건 검출기, 규칙 관리자(조건 관리자, 네트워크 관리자) 및 조치 실행기를 주요 구성 요소로 포함하며, 센서 노드나 싱크 노드로부터 들어오는 센싱 정보를 사건으로 처리하여 기존 데이터베이스 및 규칙베이스와 연동되어 필요한 조치를 실행하도록 하는 역할을 한다.

(그림 1)의 센서 데이터 서버 내의 응용 모듈인 능동규칙 시스템은 규칙 요소(사건, 조건, 조치)들에 대한 하부 시스템들로 구성되며, 이들 하부시스템들 사이의 결합 관계가 규칙시스템의 구조를 이루는 중요한 요소이다. 사건기반규칙을 채택하는 규칙시스템의 구성요소는 규칙을 정의하고 관리하는 규칙언어, 규칙언어를 컴파일 하는 규칙 컴파일러, 사건 검출을 담당하는 사건 검출기, 규칙의 조건을 평가하는 조건 평가기, 조치 실행기 및 이들 각각의 요소들을 관리하는 규칙 관리기로 구성되며, 시스템의 특성에 따라 다양한 형태로 구성할 수 있다[3,4].

사건 검출기는 규칙에서 명세된 사건을 검출하는 역할을 담당하는 부분이며, 사건은 임의의 센서 노드들의 센싱 데이터 발생 시 묵시적으로 기동 된다. 그러나 사건 기반 규칙은 사건을 명시적으로 명세할 수 있으며, 이 경우 사건의 명세는 특정 데이터나 상황을 명세 하는 기본 사건(Primitive Event)과 이들을 결합한 복합 사건들에 대해서도 명세한다. 이러한 복합 사건들은 규칙이 다양한 상황을 표현할 수 있도록 하여 준다.

규칙의 조건부는 임의의 술어나 연산으로 구성된다. 술어는 조건의 진위여부를 판단하기 위하여 다양한 관련 연산을 포함할 수 있고, 그 연산 대상에 따라 연산 부담이 증가할 수 있다. 능동성 부여에 따라 전체 시스템의 부하 증가 및 성능 저하를 방지하기 위하여 효율적인 조건 평가기법이 요구된다. 규칙의 실행과정은 주로 센싱 데이터 발생 시 시스템의 요청에 의하여 반복적으로 수행되며, 질의 내용이 미리 알려져 있는 특성이 있으므로, 이러한 특성은 효율적인 조건 평가에 있어 중요한 의미를 갖는다.

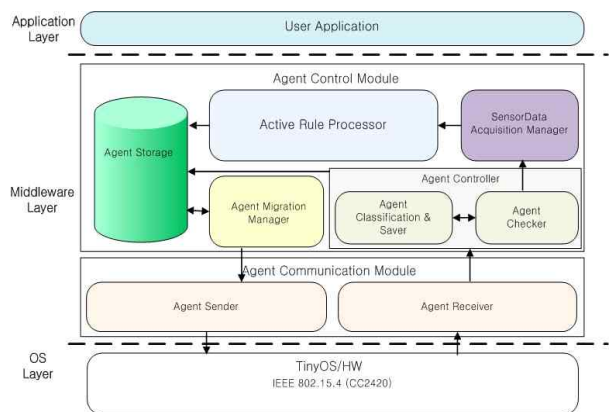
조치 실행기는 조건의 평가가 참으로 평가되었을 경우 규칙에서 명세된 내용의 조치를 실행시키는 부분이다. 규칙언어에서 명세할 수 있는 다양한 형태의 조치들은 단일 문장 형태이거나, 임의의 센서 노드를 제어하거나, 외부 응용을 동작시키는 세 가지 형태로 분류할 수 있으며, 조치 실행부는 이러한 조치를 수행시킬 수 있는 구조를 가져야 한다.

규칙은 메타 데이터 형태로 취급하며, 규칙베이스라는 특별한 영역에 저장한다. 이때 각각은 규칙 식별자를 이용하여 식별할 수 있으며, 규칙관리자의 요구에 의하여 식별된 규칙의 조치 실행부를 로딩하거나 외부 응용프로그램을 호출한다. 규칙관리자는 규칙이 정의되어 규칙베이스에 등록된 이후 규칙이 삭제될 때까지 규칙을 총괄하는 역할을 수행한다. 이때 규칙관리자가 하는 주요한 역할은 규칙의 등록 및 삭제, 활성화 및 비활성화, 각 규칙 요소들 사이의 통제와 규칙 실행과정의 관리 등이다.

3. 센서 네트워크 연동 이동 에이전트

3.1 이동 에이전트 미들웨어 플랫폼

이동 에이전트들의 노드 간 효율적 이주와 노드에서의 규칙 실행 등 역할 수행을 위하여 각 노드는 이동 에이전트 플랫폼을 필요로 한다.



(그림 2) 이동 에이전트 미들웨어 플랫폼 구조

본 논문에서 구현한 센서 네트워크 연동 이동 에이전트 미들웨어 플랫폼의 구조는 크게 이동 에이전트의 송수신을 담당하는 Agent Communication Module과 에이전트의 이주, 데이터 수집, 중복 데이터 제거 등의 작업을 수행하는 Agent Control Module로 구성되며, 위 (그림 2)와 같다.

Agent Communication Module은 이동 에이전트 이주 시 다음 목적지인 Dest_Node를 통해 다른 노드나 싱크노드로 에이전트를 송신 하는 역할을 하는 Agent Sender와 이주해 오는 이동 에이전트를 수신하고 해당 이동 에이전트의 ID등을 확인하기 위해 Agent Checker등을 호출하는 역할을 하는 Agent Receiver로 구성되어 있다. Agent Control Module은 Agent Migration Manager, SensorData Acquisition

Manager, 그리고 Agent Classification & Saver와 Agent Checker로 이루어진 Agent Controller 및 Active Rule Processor로 구성되어 있다. Agent Migration Manager는 이동 에이전트로부터 이주 리스트를 받아 에이전트의 이주를 관리한다. 이주 대상으로 선택된 노드는 이동 에이전트 이주 시에 Dest_Node에 저장된다. SensorData Acquisition Manager는 센서 데이터를 수집하여 이주 시 이동 에이전트의 Data부분에 저장한다. Agent Controller의 Agent Checker는 이동 에이전트 확인기로 이동 에이전트가 이주해 오면 해당 에이전트의 ID를 확인하고 이주의 올바름을 확인하는 역할을 수행하며, Agent Classification & Saver는 이동 에이전트의 각 부분을 분류하고 저장하는 역할을 담당한다. Active Rule Processor는 이동 에이전트에 탑재되어 수행될 수 있는 여러 가지 능동규칙들의 처리기로서, 이전 노드 또는 현재 노드에서 센싱된 데이터를 사건으로 받아들여 조건에 맞는 조치를 해당 노드에서 직접적으로 실행하는 역할을 한다. 또한, 이는 규칙베이스와 연동된 외부의 능동규칙 시스템과 연계되어 상황에 적합한 다양한 규칙들을 실행할 수 있다. Agent Storage는 센서 노드로 이동 에이전트가 이주해 오면 저장하고, 다른 센서 노드로 이주하기 직전 이곳에서 이동 에이전트를 꺼내 재조정 후 이주 시킨다. 이러한 플랫폼 상에서의 역할을 수행하는 이동 에이전트는 센서 데이터를 수집한 후 요구에 적합한 데이터만을 싱크 노드로 보내도록 구현하며, 이러한 능동적인 규칙들을 탑재할 수 있는 구조로 구성한다[10].

이동 에이전트 구조는 이동 에이전트 식별자, 이동 에이전트 복귀 싱크 노드 ID, Start_Node, Dest_Node, 이주 대상 노드 목록, Data 필드 등으로 구성한다. 다양한 능동규칙을 실행을 위하여 Data 필드 수를 확장할 수 있으며 규칙 처리 시 필요한 임계값이나 카운터 필드를 추가할 수 있도록 하여, 다양한 능동규칙 탑재 및 실행을 위한 확장성을 제공한다.

3.2 이동 에이전트 이주 및 능동규칙 실행

이동 에이전트는 센서 노드에서 수행할 지정된 처리를 마치면 다음노드로 이주한다. 이동 에이전트의 노드 이주는 Agent Migration Manager가 Mig_List를 참조하여 Dest_Node를 선택하고 Agent Communication Module의 Agent Sender를 통해 수행한다. 이주는 다음 목적지 노드인 Dest_Node를 Mig_List에 있는 노드들 중에서 선택한 후에 이루어지며, 다음 노드의 선택은 이동 에이전트의 이동 거리, 정확도, 중요도 및 경로 회피방법 등 다양한 라우팅 방법을 적용하여 결정될 수 있다.

이동 에이전트의 이주는 message_t 구조체로 표현되는 TinyOS의 액티브 메시지 구조를 통해 이루어진다. 액티브 메시지는 MAC 계층의 프레임 제어와 패킷의 순서번호 및 주소체계 필드를 가지는 헤더와 유효성 검사를 위한 풋터를 제외하고 최대 116바이트를 사용자 데이터 영역으로 사용할 수 있다. 이동 에이전트의 이주는 이 영역을 사용하여 이동

에이전트 mobileAgent_t 구조체를 message_t 액티브 메시지에 탑재하여 보냄으로써 이루어진다.

본 논문에서는 이동 에이전트에 능동규칙 탑재 및 실행과정을 보이기 위하여, 중복 센서 데이터 처리 규칙을 탑재하고 이를 실행하는 과정을 실험한다. 실험은 두 개의 직전 노드들의 센서 데이터 값들을 이용하여 중복 데이터를 처리하고, 일정 주기마다 센서 데이터를 전송할 수 있도록 NoSendingCount를 두어 다양한 형태의 센서 데이터 값들의 분포 환경에도 일반적으로 적용할 수 있도록 하였다. 이동 에이전트가 이주하면서 가져온 인접 노드 2개의 데이터를 pData와 ppData란 이름으로 저장하고, 현재 센서 노드의 데이터를 localData에 저장 후 중복 제거 모듈을 호출한다. 중복 제거 모듈은 pData와 ppData를 차례로 localData와 비교하여 THRESHOLD값 이상 차이가 있을 경우에만 싱크 노드로 전송하고, 그렇지 않을 경우에는 NoSendingCount 값을 1 증가시키고 이주를 요청한다. 또한, NoSendingCount 수가 MAX_NOSENDING보다 크면 중복 여부와 관계없이 현재 센서 노드의 데이터를 싱크 노드로 전송하도록 함으로써, 센서 분포 정도, 센서 데이터 값들의 변화 정도 및 해당 지역의 특성에 적합하도록 다양하게 능동규칙을 최적화할 수 있음을 보인다.

4. 실험

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 사용하였으며, 실험 장소는 데이터 손실을 최소화 할 수 있는 임의의 장소를 선택하였다. 실험은 온도 데이터를 사용하였고 윈도우 상에서 TinyOS 개발을 위해 Cygwin 툴을 사용하였다. 각 센서 노드에 이동 에이전트 플랫폼 이미지를 업로드 한 후 사용자 viewer를 실행하고, 에이전트의 이동은 viewer를 통해 확인 가능하나 각 센서 노드에 부착된 Led로도 이동 에이전트의 이동과 센서 데이터 전송 여부를 확인 할 수 있다.

다음 (그림 3)는 7개의 센서 노드들을 이용하여 중복 센서 데이터 처리 알고리즘 수행 결과를 보인 것이다. 이동 에이전트는 각 노드로 이주한 후 센서 데이터를 싱크 노드로 전송한다. 이때 중복 데이터 제거 모듈은 현재 노드의 센서 데이터 전송 여부를 결정한다. (그림 3)의 (A-1)과 (B-1)은 중복 제거 모듈을 탑재 하지 않은 이동 에이전트가 각 노드를 1 ~ 7 번순으로 순차적으로 이주하면서 센서 데이터를 싱크 노드로 전송한 화면이다. 즉 (A-1)과 (B-1)은 실제 센싱 데이터 값을 그대로 싱크노드로 보낸 결과를 보인 것이다. (A-2)와 (B-2)는 이동 에이전트에 중복 제거 모듈을 탑재하고, 중복 데이터 처리를 위한 임계값인

THRESHOLD를 2로 설정한 결과로써, 이동에이전트가 이주하여 도착한 센서의 데이터와 직전에 방문한 두 개의 센서 노드의 데이터 값들(pData, ppData) 중 하나라도 차이가 임계값인 2이상일 경우에만 싱크노드로 센싱된 데이터를 송신한 결과이다. (A-2)에서 최초 1번 노드는 인접 노드의 센서 데이터를 가지고 있지 않기 때문에 싱크 노드로 바로 전송하였고, 3번 노드는 직전 전 노드인 1번 노드의 센서 데이터(ppData)와의 차이가 2이상이기 때문에 해당 데이터를 싱크 노드로 전송하였다. 또한 (A-2)에서 7번 노드의 경우 직전 방문 노드인 6번 노드의 센서 데이터(pData)와의 차이가 2이상이므로 해당 노드의 센서 데이터를 싱크로 전송하였다. (B-2)도 동일한 알고리즘에 의한 수행 결과이다.

(A-2)와 (B-2)와 같이 중복 처리 알고리즘만을 적용할 경우, 센서 노드들의 배치에 따라 직전 두개 노드들의 센싱 데이터 값이 계속 임계값 미만으로 비교될 경우 중복 처리가 계속 되므로 싱크 노드로의 데이터 전송이 발생하지 않기 때문에 응용 및 상황에 따라 필요한 센서 데이터를 획득하지 못할 수 있다. 따라서, (A-3)과 (B-3)은 이를 방지하기 위하여 NoSendingCount를 추가하여 센서 데이터를 전송하지 않은 노드들의 수가 MAX_NOSENDING 보다 크게 될 경우에는 무조건 싱크 노드로 송신하게 한 실험 결과이다. MAX_NOSENDING 값을 3으로 하여 실험한 결과인 (A-3)의 6번 노드에서 NoSendingCount가 3이 되기 때문에, 6번 노드의 센서 데이터는 싱크 노드로 전송하고, NoSendingCount를 0으로 변경한 후 7번 노드로 이주한다. 7번 노드에

서는 이전 6번 노드와의 센서 데이터 차이가 THRESHOLD 이상이 되므로 유효한 데이터로 간주하여 싱크 노드로 해당 센서 데이터를 전송한다.

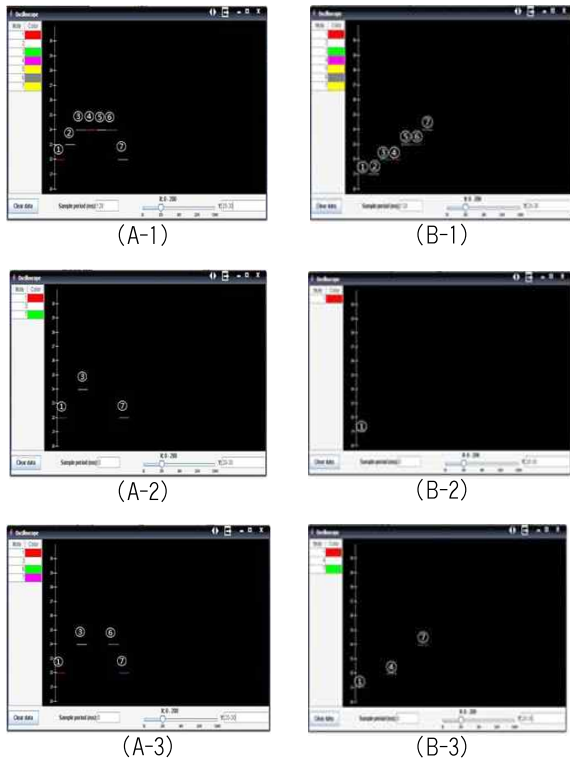
실험 결과 (A-3)과 (B-3)에서 알 수 있듯이, 이들 경우 원래의 센서 데이터 값들의 변화 형태를 유지할 수 있으며, 불필요한 중복 센서 데이터 값들의 전송을 줄일 수 있음을 보임으로써 제안 방법의 유효성을 나타낸다.

5. 결론

본 논문에서는 실제 센서 노드 Hmote2420 및 X-Hyper 320WN 센서 게이트웨이와 실제 환경에서 발생할 수 있는 상황을 고려한 조건 및 제한들의 적용 예로 중복 데이터 제거 모듈이라는 능동규칙을 탑재한 이동 에이전트의 이주 및 규칙 실행 실험을 통하여 센서 네트워크 환경에서의 능동적 다중 에이전트 미들웨어의 응용 가능성 및 효율성을 보였다. 제안한 능동적 다중 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 따른 능동규칙을 통하여, 용도 및 필요에 따라 센서 데이터를 수집, 전송 및 조치를 실행한다. 또한, 제안 시스템은 향후 다양한 환경과 응용에 적합한 능동규칙 탑재 및 실행뿐 아니라 능동규칙 시스템과의 연계를 통하여 다양한 능동적 센서 네트워크 응용에의 적용 가능성을 보장하며, 향후 추가적으로 최적의 능동규칙 설계를 위한 온톨로지, 에이전트 경량화 및 효율적 라우팅 방법 등이 요구된다.

참고 문헌

- [1] 황재각, 표철식, "USN미들웨어 기술 개발 동향," 한국전자과학 회지, 제19권, 제6호, pp.51-59, 2008.
- [2] 원광호, 황태호, 김동순, 김태현, "WSN 기술 동향 및 응용기술," 정보통신학회지(정보와통신), 제25권, 제10호, pp.33-41, 2008
- [3] 황정희, 신예호, 류근호, "능동 규칙의 실행 의미를 반영한 종료 분석," 컴퓨터정보통신연구, 제9권, 제1호, pp.93-102, 2001.
- [4] 신 문선, 이 명진, "Active network management system with automatic generation of network management program using triggers," 한국인터넷정보학회논문지, 제10권, 제1호, pp.19-31, 2009.
- [5] 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.
- [6] 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.



(그림 3) 능동규칙(중복 센서 데이터 처리) 수행 실험 결과

[7] 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.

[8] 최신일, 문석재, 엄영현, 국윤규, 정계동, 최영근, "분산 센서 네트워크에서 모바일 에이전트를 이용한 효율적인 데이터 수집," 한국정보과학회 가을학술발표논문집, 제33권, 2호(B), pp. 138-142, 2006.

[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 춘계학술발표대회논문집, 제17권, 제1호, pp.1070-1073, 2010.

[11] 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.

[12] 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.



이 연 식

e-mail : yslee@kunsan.ac.kr

1982년 전남대학교 전자계산학과(이학사)

1984년 전남대학교 전자계산학과(이학석사)

1994년 전북대학교 전산응용공학전공

(공학박사)

1995년~1997년 군산대학교 교무부처장

1997년~1998년 University of Missouri(Kansas City) 교환교수

1999년~2001년 군산대학교 전자계산소장

2004년~2005년 Ohio State University 교환교수

1986년~현 재 군산대학교 컴퓨터정보공학과 교수

관심분야: 객체지향 시스템, 능동규칙 시스템, 센서네트워크

에이전트 미들웨어, USN 응용



장 민 석

e-mail : msjang@kunsan.ac.kr

1989년 연세대학교 전자공학과(공학사)

1991년 연세대학교 전자공학과(공학석사)

1997년 연세대학교 전자공학과(공학박사)

2002년~2003년 한국대학교육협의회 국내

교류 파견연구교수

2009년~2010년 연구교수

1997년~현 재 군산대학교 컴퓨터정보공학과 교수

관심분야: 웹기반기술/응용, USN 응용