

센서 데이터 획득을 위한 이동 에이전트 설계

이연식, 이정수
군산대학교 컴퓨터정보공학과
e-mail : {yslee, jsjs2000}@kunsan.ac.kr

Design of Mobile Agent for Sensor Data Acquisition

Yonsik Lee, Jeongsu Lee
Dept. of Computer Information Engineering, Kunsan National University

요 약

센서 네트워크 환경에서 센서 노드들은 무선 통신 능력을 이용하여 접근하기 어려운 시공간적 환경에서 센싱 데이터를 송수신하며 각종 데이터를 수집, 분석, 감시 및 처리를 용이하게 한다. 이러한 센서 노드들은 동적으로 변화하는 환경에 대한 적응력이 떨어지고, 센서 데이터의 과잉 및 중복 송수신으로 인한 전력이나 네트워크 대역폭 관련 등의 문제점을 가진다. 이에 본 논문에서는 센서 노드를 이주하며 중복 데이터를 제거하고, 사용자나 어플리케이션의 요구에 적합한 데이터만을 수집 및 전송하여 데이터의 과잉 송수신으로 인한 전력낭비와 네트워크 부하를 줄일 수 있는 이동 에이전트를 설계한다. 또한 기존 멀티 에이전트 시스템과의 연계를 통한 이동 에이전트의 이주 알고리즘을 제시하고, 실제 이동 에이전트의 통신 수행과정을 보임으로써 설계 방법의 유효성을 보인다.

1. 서론

센서 네트워크는 Ad-hoc 네트워크의 일종으로 센싱 기능과 계산처리, 그리고 무선통신 능력을 가지고 있는 센서 노드와 게이트웨이 또는 데이터 집중국 역할을 하는 싱크 노드로 구성된 무선 네트워크로 정의된다. 기본적으로 센서 노드는 감지된 정보를 싱크 노드로 전달하고, 싱크 노드는 인터넷과 같은 기존의 네트워크를 통하여 사용자에게 해당 정보를 제공한다.

일반적으로 센서 네트워크는 센싱 지역에 분산되어 있는 센서 노드들의 수집 데이터를 주변 센서 노드들과의 협업을 통해 싱크 노드로 전송한다[1, 5, 7, 10].

이러한 센서 네트워크에서는 제한된 자원과 전력 등의 한정된 환경 내에서 통신하기 때문에, 중복된 데이터의 전송이나 비효율적인 센싱 주기 등은 센서 노드의 수명을 짧게 만들고, 시스템 전체의 성능에도 큰 악영향을 끼친다. 센서 노드들이 광범위한 센싱 지역에서 인근 센서 노드와 중복된 데이터가 발생하는 경우 이를 선별하여 전송하고, 주변 상황에 따라 능동적으로 센싱 주기나 전송 주기를 변경할 수 있는 센서 네트워크 시스템의 개발이 필요하다[3, 4, 7]. 이를 위해서는 멀티 에이전트 시스템의 이동 에이전트와의 연계를 통해 이동 에이전트가 각 센서 노드들을 순회하며 해당 센서 노드가 센싱하는 데이터를 확인하고 선별하여 데이터를 전송하는 이동 에이전트 기반의 센서 네트워크 시스템이 적합하다. 멀티 에이전트 시스템은 분산 환경에서 네트워크 부하나

대기시간 감소 등의 문제를 해결하는데 주로 사용되어 왔으며, 효율적인 정보 검색과 다양한 응용 분야에 적용할 수 있는 시스템이다[2, 8, 9].

본 논문에서는 멀티 에이전트 시스템의 이동 에이전트에 대해 분석하고, 센서 네트워크에서 중복 데이터의 송수신으로 인한 전력낭비 및 네트워크 과부하를 막고 변화하는 환경에 능동적으로 적응 가능한 이동 에이전트 모델을 설계한다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련연구로 센서 네트워크상의 OS 와 멀티 에이전트 시스템과의 연계 및 이주를 위한 방법을 설명하고, 3 장에서 센서 네트워크 환경에서의 이동 에이전트 모듈과 이주 알고리즘을 설계한다. 그리고 4 장에서는 이동 에이전트의 실제 이주 및 통신 수행 과정을 보이고, 5 장에서 결론을 맺는다.

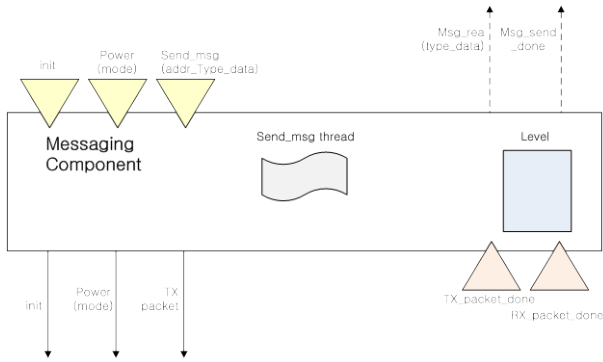
2. 관련연구

2.1 센서 네트워크

센서 노드는 마이크로 컨트롤러를 내장한 소형 컴퓨터 시스템으로 센싱 응용처리와 센서 노드 간 통신 등을 위한 운영체제가 필요하다. 센서 운영체제는 자원이 제한적인 센서 노드 하드웨어에서 수행되어야 하므로 크기가 작고 전력 소모가 적어야 하며, 센서 노드 간에 저 전력 통신을 제공하면서도 프로세서와 메모리를 효율적으로 관리하도록 설계되어야 한다[1, 3, 4, 7, 10].

현재 센서 네트워크 운영 체제로는 TinyOS, MANTIS,

SOS, Nano Qplus 등의 다양한 운영체제가 연구 개발 중에 있다[10]. 이중 TinyOS 는 현재 가장 널리 쓰이고 있는 컴포넌트 기반, 이벤트 구동방식의 센서 네트워크 운영체제로 본 논문에서는 TinyOS 를 사용하여 센서 네트워크를 구성하였으며, 이동 에이전트 또한 TinyOS 상에서 구동되도록 설계하였다. 다음 [그림 1]은 TinyOS 의 메세징 컴포넌트 구조를 나타낸다 [10].

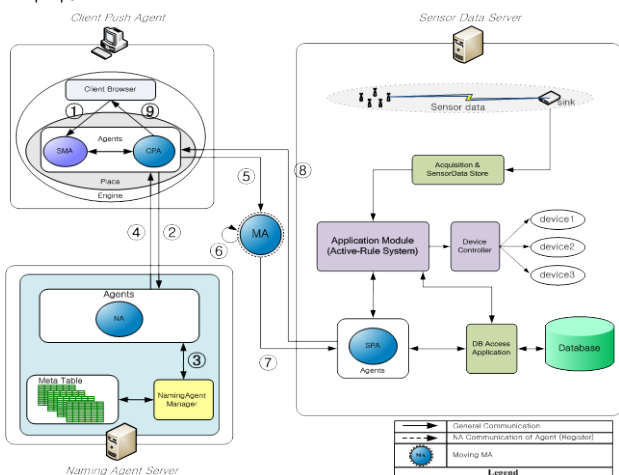


[그림 1] TinyOS 의 메세징 컴포넌트

TinyOS 의 가장 큰 장점은 재사용 가능한 컴포넌트들로 구성된다는 점과, CPU 가 사용되지 않을 동안에는 sleep 상태로 들어간다는 점이다. 불필요한 전력 소모를 줄여 제한된 전력을 가진 센서 노드의 수명을 늘릴 수 있다. TinyOS 상에서 개발, 구동되는 이동 에이전트 또한 위의 장점을 가지며, 멀티 에이전트 시스템과의 협업을 통해 보다 다양한 응용에 이용될 수 있다.

2.2 멀티 에이전트 시스템의 이동 에이전트

다음 [그림 2]는 멀티 에이전트 시스템의 전체 구조이다.



[그림 2] 멀티 에이전트 시스템 구조

멀티 에이전트 시스템[2, 8, 9]에서 이동 에이전트는 시스템을 구성하고 있는 SPA(ServerPushAgent), SDS(SensorDataServer), NA(NamingAgent) 그리고 CPA(ClientPushAgent) 등의 다양한 에이전트들과 상호 협력하여 이주하며, 네이밍 서비스를 이용하여 사

용자 요구에 적합한 결과를 제공 할 수 있도록 한다. 또한 현재의 위치에서 이주 대상 서버 리스트의 각 목적지 중 최적 경로 탐색을 통해 이주 경로가 최적인 노드를 판단하여 이주를 수행함으로써 작업 처리 시 소요되는 시간을 최소화한다[2, 9].

멀티 에이전트의 초기 작업으로써 멀티 에이전트의 각 구성요소인 CPA, SPA, SDS, NA 등은 상호 연동 구조를 형성하기 위하여 NA 에 이름과 각각의 특성 정보들이 네이밍 서비스를 통하여 등록 된다. 그리고 클라이언트 브라우저의 이벤트 발생에 따라 각 에이전트는 상호 정보를 교류하기 위해 응답과 대기 를 반복하게 되며 최종 결과를 사용자에게 제공한다.

다음 [그림 3]은 멀티 에이전트 시스템에서 이동 에이전트의 이주를 위한 서버리스트 생성 알고리즘 이다.

Migration ServerList

Input : *Keyword*

Output : *ServerList*

Process:

NamingSearch (*Keyword*);

GET MetaTable;

WHILE(MetaTable.HasNext())

IF(*Keyword* == MetaTable.Keyword) THEN

ServerList[][] = Name, URL, Keyword,

SinkNum, SinkInfo,

SinkURL;

Return ServerList[][];

ELSE

Return Null;

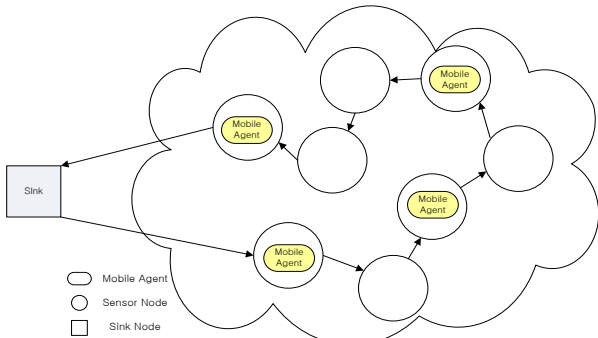
[그림 3] 이주 서버 리스트 생성 알고리즘

NA 에서 생성 되어진 이주 서버 리스트가 이동 에이전트에게 전달 되면, 이동 에이전트는 최적 경로 탐색 과정을 거쳐 리스트를 재조정하고, SPA 나 SDS 로 이주하게 된다. 이주 후 에이전트는 요청된 자료가 있는지 서버내의 자료를 검색하고 해당되는 자료가 있으면 이를 CPA 쪽으로 반환하고, 리스트 내의 다음 서버로 이주한다.

3. 센서 네트워크 환경의 이동 에이전트 모듈 설계

변화하는 환경에 대한 적응력 감소, 중복 데이터 처리, 네트워크 과부하 등의 문제들을 해결하기 위해 이동 에이전트 기반의 센서 네트워크 환경을 제안한다. 이동 에이전트는 싱크 노드 뿐만 아니라 센서 노드를 이주하며 적절한 데이터만을 선택적으로 수집할 수 있고, 중복 데이터 수집을 막을 수 있다. 이동 에이전트 기반 센서 네트워크 환경은 다음과 같은 장점을 갖는다. 첫째, 이동 에이전트를 이용하여 환경에 대한 정보를 직접 감지하고, 동적 변화에 따른 센서 데이터 수집 방법 및 주기를 변경할 수 있어, 새로운 어플리케이션에 따른 데이터 변화와 동적인 주변 환경 변화에 빠르게 적응 할 수 있다. 둘째, 불필요하고 중복되는 데이터는 제거과정을 거쳐 선택적으로 데이터를 수집함으로써 네트워크 부하를 줄인다. 이동 에이전트를 이용하여 각각의 센서 노드로 이주 후 선택적으로 필

요한 데이터만을 획득 함으로써 네트워크 부하를 줄일 수 있다. 다음 [그림 4]는 이동 에이전트가 센서 네트워크 환경에서 각 센서 노드를 이주하는 과정을 나타낸다.



[그림 4] 이동 에이전트 이주 과정

이동 에이전트를 이용하여 센서 데이터를 획득할 경우 불필요하거나 중복된 데이터를 삭제할 뿐만 아니라, 여러 어플리케이션에 따른 각기 다른 데이터를 수집할 수 있다. 예를 들어 A 라는 프로그램은 Temperature 데이터를 10 진수 5 자리의 숫자로 원할 수 있고, B 라는 프로그램은 Humidity 데이터를 10 진수 3 자리의 percent 데이터로 원할 수도 있다. 이동 에이전트를 이용하지 않을 경우에는 이런 데이터들을 서버나 기타 다른 응용프로그램에서 변환하거나 정규화 해주는 과정이 있어야 하지만 이동 에이전트를 이용할 경우에는 해당 프로그램에 맞게 데이터를 변환하여 수집할 수 있다.

또한 중복 데이터 제거나 수집 주기를 변경할 수 있다. 인접된 노드들에 중복이나 유사 데이터가 많을 경우 네트워크 부하를 유발할 수 있다. 이럴 경우엔 이동 에이전트를 이용하여 각 노드로 이주 후 센싱된 데이터를 확인하여 인접 노드와 중복되는 데이터는 삭제하고, 차이(특정범위)가 있는 데이터만을 추출하여 싱크로 전송하게 하여 네트워크 트래픽을 감소 시키고, 서버 측의 부하를 줄일 수 있다.

다음 [그림 5]는 이동 에이전트의 노드 이주 알고리즘을 나타낸다.

```

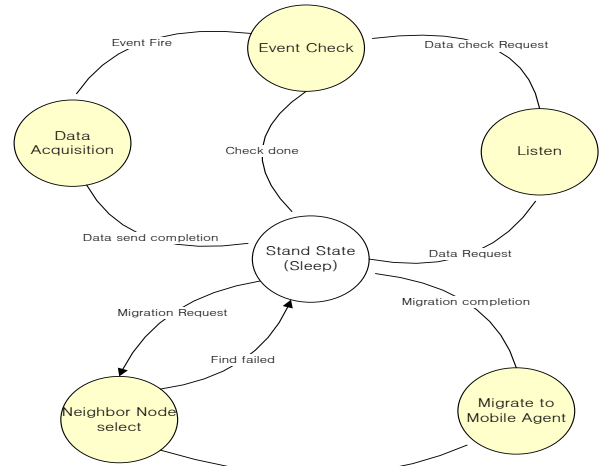
Mobile Agent Migration
Input : VisitedList
Output : SensorData
Process:
IF (Node_ID == VisitedList)
    MA Migration Neighbor Node;
ELSE IF (Node_ID != VisitedList)
    MA Acquisition SensorData;
    MA Migration Neighbor Node;
ELSE IF(ThisNode = LastNode)
    MA Acquisition SensorData;
    MA Migration SinkNode;
END
    
```

[그림 5] 이동 에이전트 이주 알고리즘

이동 에이전트는 센서 노드로 이동하면 우선 방문

했던 노드인지 확인 후 방문하지 않은 노드라면 현재 노드가 센싱하고 있는 센서 데이터를 수집한다. 만일 센서 데이터 수집시에 조건이 있다면 해당 조건 또한 체크 하게 된다. 데이터 수집 후 이동 에이전트는 인접 노드를 검색하여 해당 노드로의 이주를 반복하고, 마지막으로 모든 노드를 이주한 후에 싱크 노드로 돌아 오게 된다.

다음 [그림 6]은 이동 에이전트의 상태천이를 보여 준다.



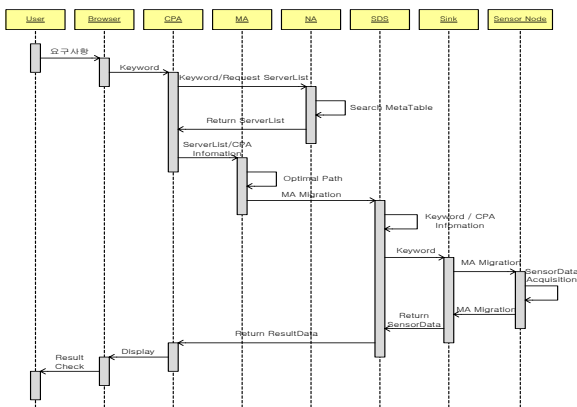
[그림 6] 이동 에이전트 상태 천이도

이동 에이전트는 어떠한 작업을 수행하기 전까지는 항상 대기(Sleep) 상태가 된다. 에이전트가 위치해 있는 센서 노드 또한 대기상태가 된다. 이동 에이전트에게 이주 요청(Migration Request) 이 들어오면 에이전트는 인근의 노드(Neighbor Node)를 선택하여 이동한다. 센서 노드로 이동한 에이전트는 실행되기 전까지 다시 대기하고, 센서 데이터 요청이 들어오면 현재 센싱되는 데이터를 확인한다. 센서 데이터가 사용자가 응용 프로그램이 요청한 데이터에 해당한다면(Event Fire) 데이터 수집 혹은 싱크 노드로 전송을 한다.

센서 네트워크 환경에서 이동 에이전트는 환경의 특성상 운영체제와 미들웨어 또는 응용프로그램의 경계가 뚜렷하지 않다. 센서 노드 같은 협소한 메모리 공간을 가진 환경에서는 운영체제와 미들웨어는 서로 유기적으로 작동하며 하드웨어를 제어한다. 이동 에이전트는 센서 노드와 센서 노드 사이에서 또 센서 노드와 싱크 노드 사이에서 동작하며 사용자에게 알맞은 데이터를 제공한다.

4. 이동 에이전트 통신 수행 과정

다음 [그림 7]은 이동에이전트 기반의 센서 네트워크 상에서의 에이전트 및 메시지 이동 과정을 나타낸다.



[그림 7] 이동 에이전트의 통신 수행 과정

본 논문에서 제시한 이동 에이전트를 기반으로 센서 네트워크 응용의 예를 보이면 다음과 같다.

예를 들어 클라이언트 브라우저를 통해 “temperature ≥ 25”라는 센서 데이터 검색 키워드를 사용자가 입력하였다면, CPA 는 이 검색 키워드를 NA 에 전달한다. 메시지를 전달받은 NA 는 전달 받은 내용 “temperature”를 MetaTable 에 등록된 서버들 중 SDS 를 찾아 SDS 만 가지고 있는 SinkMetaTable 을 검색한다. SinkMetaTable 내의 SinkInfo 항목에 센서 데이터의 정보가 담겨 있기 때문에 “temperature”라는 정보를 SinkInfo 항목들과 비교 후 해당되는 SDS 리스트를 CPA 측에 반환한다. CPA 에 반환되는 SDS 리스트에는 해당 서버의 Name, URL, Keyword, SinkNum, SinkInfo, SinkURL 등의 Meta Data 가 포함되어 있다. MetaTable 검색 결과로 반환 받은 SDS 리스트를 CPA 는 MA 에게 CPA 의 ReferenceObject, Name, URL 등의 정보와 함께 전달한다. MA 는 최적 경로 탐색 및 조정 과정을 거쳐 해당 SDS 로 이주하여, “temperature ≥ 25” 검색 정보와 결과를 반환 받을 CPA 의 정보를 전달하고, 리스트의 다음 서버로 이주한다. SDS 는 포함된 싱크 노드 중 “temperature” 데이터를 반환하는 싱크 노드에게 “temperature ≥ 25” 메시지를 전달한다. 이때 싱크 노드는 이동 에이전트에게 “temperature ≥ 25” 에 해당하는 데이터를 반환할 것을 요구한다. 이동에이전트는 해당 메시지를 가지고 센서 노드로 이주를 시작한다. 이동 에이전트는 이주와 데이터 체크를 반복하며 다수의 센서 노드들을 순회하며 “temperature ≥ 25”에 해당하는 데이터를 찾아 싱크 노드로 반환한다. 또한 “temperature ≥ 25”에 해당되지만 인접 노드와 중복되는 데이터는 삭제하여, 중복 전송을 방지한다. 이동 에이전트의 모든 센서 노드 순회가 완료 되면 싱크 노드는 에이전트가 전송한 데이터들을 SDS 에 전송하고, SDS 는 최초 정보를 요구한 CPA 에게 결과를 반환하고 통신을 마친다.

5. 결론 및 향후 연구 과제

본 논문에서는 센서 네트워크 환경에서 이동 에이전트를 이용하여 센서 데이터를 수집 및 전송하여 중복 데이터의 송수신을 방지하고, 새로운 사용자나

어플리케이션에 알맞은 데이터를 동적으로 변환하여 전송할 수 있는 이동 에이전트 모델을 제안하였다. 이것은 기존의 센서 노드에서 단순히 데이터를 센싱하여 송수신하는 정적인 환경과 달리 센서 데이터를 선별하여 송수신할 수 있다는 점과 센서 노드에서 싱크 노드로 데이터 송수신시에 새로운 어플리케이션이 요구하는 형태의 데이터로도 제공이 가능한 시스템이다. 또한 이동 에이전트를 이용한 센서 네트워크 시스템은 기존의 정적인 시스템에 비해 전원관리 문제나, 네트워크 대역폭할당 문제, 과부하 문제 등을 해결할 수 있는 시스템이다. 그러나 이동 에이전트를 이용한 센서 네트워크 시스템은 프로그램의 크기가 커지고, 전송 되는 데이터의 양이 더 많아 진다는 문제를 가질 수 있기 때문에 향후에는 에이전트 모델의 크기를 줄이고, 전송 데이터의 효율적인 송수신에 대한 연구가 진행되어야 할 것이다.

참고문헌

- [1] Chien-Liang Fok, Gruia-Catalin Roman, Chenyang Lu, "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications," 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05), pp. 653-662, Jun. 2005.
- [2] 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. 2-4, Sep. 2008.
- [3] Min Chen, Taekyoung Kwon, Yong Yuan, and Victor C.M.Leung, "Mobile Agent Based Wireless Sensor Networks," Journal of Computer, pp. 14-21, Apr. 2006.
- [4] 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.
- [5] 최신일, 문석재, 엄영현, 국윤규, 정계동, 최영근, "분산 센서 네트워크에서 모바일 에이전트를 이용한 효율적인 데이터 수집," 한국정보과학회 가을학술발표논문집, 제 33 권, 제 2 호(B), pp. 138-142, 2006. 10.
- [6] 남진우, 정영지, "센서 네트워크에서 헬스케어 이동성 에이전트 모듈 설계," 멀티미디어학회논문지, 제 11 권, 제 4 호, pp. 544-553, 2008. 04.
- [7] 황재각, 표철식, "USN 미들웨어 기술 개발 동향," 한국전자과학회지, 제 19 권, 제 6 호, pp. 51-59, 2008. 11.
- [8] 이정수, 최영춘, 이연식, "센서 네트워크 응용을 위한 네이밍 에이전트 설계," 정보통신분야학회합동 학술대회논문집, pp. 147-151, 2009. 11.
- [9] 김광중, 고현, 이연식, "네이밍 에이전트의 메타데이터를 이용한 이동에이전트의 적응적 이주 경로 기법," 한국컴퓨터정보학회논문지, 제 12 권, 제 3 호, pp. 165-175, 2007. 07.
- [10] 원광호, 황태호, 김동순, 김태현, "WSN 기술 동향 및 응용기술," 정보통신학회지(정보와통신), 제 25 권, 제 10 호, pp. 33-41, 2008