Journal of Korea Multimedia Society Vol. 11, No. 4, April 2008(pp. 544-553)

센서 네트워크에서 헬스케어 이동성 에이전트 모듈 설계

남진우[†], 정영지[††]

요 약

통상적인 센서 네트워크는 센서노드로부터 고정된 정보를 수집하는 일괄적인 방식을 사용한다. 하지만 이러한 방식은 헬스케어 환경에서 주요 센싱대상인 사람의 상태와 위치변화, 주변 환경의 변화에 능동적으로 대처하기 어렵다. 헬스케어 환경을 지원하기 위해서는 사람의 상태와 주변상황의 변화에 따른 특정정보를 제공해주어야 한다. 또한 사람의 이동에 따른 네트워크 타입의 변화에 대처하여야 한다. 본 논문에서는 대표적인 다중 에이전트 미들웨어인 Agilla 모델과 주변노드간의 클러스터를 구성함으로써 네트워크의 효율성을 보장하는 LEACH 프로토콜에 대해 분석한다. 이 분석 결과를 기반으로 기존 LEACH 프로토콜의 취약점인 노드 이동성을 보장하는 LEACH_Mobile 프로토콜을 제안한다. 또한 노드의 동적 기능변경을 지원하는 Mobility Agent Middleware와 LEACH_Mobile 라우팅 모듈을 구성요소로 이동노드의 동적 기능변경을 지원하는 Mobility Agent 모듈을 설계한다. 그리고 LEACH_Mobile 프로토콜의 시뮬레이션 결과를 통해 네트워크의 데이터 전송 성공률에 대한 성능 향상을 보여준다.

Design Healthcare Mobility Agent Module in Sensor Network

Jin-Woo Nam[†], Yeong-Jee Chung[††]

ABSTRACT

The general sensor network uses bundle method to collect fixed information from sensor node. However, this method is difficult to actively cope with major sensing objects in healthcare environment including status and position change of person and change of surrounding environment. In order to support the healthcare environment, certain information should be provided in accordance with the change of status of person and surrounding circumstance and also must cope with the change of network type by movement of person. This paper analyzes LEACH protocol which guarantees effectiveness of network as it constitutes clusters between Agilla model that is a representative multi agent middle ware and surrounding nodes. Based on the result of this analysis, it suggests LEACH_Mobile protocol which guarantees node mobility that is the weakest point of LEACH protocol. Moreover, it designs mobility agent middleware which supports dynamic function change of node and mobility agent module which supports dynamic function change of mobile node as components of LEACH_Mobile routing module. In addition, it is definitely increase performance which in mobile node of transfer data rate through LEACH_Mobile protocol of simulation result.

Key words: Healthcare Service(헬스케어 서비스), Sensor Network(센서 네트워크), Agent(에이전트), Routing Protocol(라우팅 프로토콜)

※ 교신저자(Corresponding Author) : 남진우, 주소 : 전라북도 익산시 신용동 344-2(570-749), 전화 : 063)850-6887, FAX : 063)856-8009, E-mail : pote333@wku.ac.kr
접수일 : 2007년 10월 18일, 완료일 : 2008년 3월 4일
[†] 준회원, 원광대학교 컴퓨터공학과 석사과정

[††] 정회원, 원광대학교 공과대학 전기전자 및 정보공학부 정교수
(E-mail : yjchung@wku.ac.kr)
※ 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (지방연구중심대학육성사업/헬스케어기술개발사업단).

1. 서 론

센서 네트워크는 매우 작은 크기의 독립된 무선 센서들을 물리적 공간에 배치하여 주위의 상황정보를 무선으로 감지, 관리할 수 있는 기술을 의미한다. 일반적으로 센서 네트워크는 센싱 지역에 분산되어 있는 센서 노드들의 수집 데이터를 주변 센서노드들과의 협업을 통해 수집노드로 전송한다. 초기의 센서 네트워크 개념은 특정 환경에서의 빠른 배치 및 랜덤 토폴로지 구성을 주축으로 군용 및 재해, 오지 등의 환경에 대한 모니터링에 이용되어 왔다. 그러나 최근 대두되고 있는 유비쿼터스 컴퓨팅 연구에 따른 센서 네트워크 개발의 증가로 인해 인간의 행동에 관련된 상업적 응용에 까지 범위가 확장되었다. 이러한 응용은 헬스케어 홈, 병원, 요양원 등의 시설에서 헬스케어를 위한 서비스 환경 구축을 필요로 하게 되었다. 헬스케어 환경을 지원하기 위해서는 사람의 이동과 주변상황의 변화에 따라 능동적으로 센서노드의 기능을 변경하여 고정된 정보가 아닌 상황에 따른 특정 정보를 제공해주어야 한다. 즉, 동적이고 이동성을 지닌 센싱대상에 대한 대량의 정보 수집, 교환, 처리가 요구된다. 하지만 고정된 위치의 센서노드가 주변 상황의 정보를 수집하여 고정된 라우팅 경로를 통해 수집노드로 전송하는 기존 방식으로는 사람의 이동에 따른 위치변화와 상태정보 그리고 환경변화에 대처하기 어렵다. 이러한 센서 네트워크를 보장하기 위해서는 센서노드에서 지원하는 범위의 동적인 기능 변경과 사람의 이동과 주변 환경의 변화에 따른 노드들 간의 네트워크 타입의 변화에 대처하여 센싱정보에 대한 정보전달의 효율성을 고려해야 한다.

본 논문에서는 노드의 동적인 기능변경을 위해 다중 에이전트를 지원하며 노드 간 에이전트의 이동성을 보장하는 Agilla와 다수의 노드가 밀집된 공간에서 주변노드간의 클러스터를 구성함으로써 네트워크의 측정 가능성과 제한된 전원문제, 그리고 정보전달의 효율성에 있어 유용한 해결방안을 제시하는 계층적 클러스터 알고리즘인 LEACH 프로토콜에 대해 분석한다. 그리고 모바일 에이전트 기반의 미들웨어 모듈과 LEACH 프로토콜의 클러스터 알고리즘을 따르며 노드의 이동성을 보장하는 LEACH-Mobile[1] 프로토콜을 지원한 라우팅 모듈을 설계한다. 이와 같은 모듈들의 설계를 바탕으로 헬스케어 응용 서비스를 위한 이동 센서네트워크를 지원하는 Mobility Agent 모듈 설계를 제안한다. 또한, LEACH_Mobile 프로토콜의 시뮬레이션 결과를 통해 이동 노드가 증가함에 따른 데이터 전송 성공률의 명확한 성능 향상을 보여준다.

2. 관련 연구

헬스케어 환경에서 밀집분포 되어 있는 대량의 센서노드들은 이동성을 가진 대상에 대한 대량의 정보 수집, 교환, 처리가 요구된다. 이 같은 요구사항은 인간의 이동에 따른 위치정보를 기반으로 동일한 기능의 센서노드들을 배치시켜 처리할 수 있으나 이런 방식은 기능에 대한 제한을 가지므로 다양한 처리를 수행하기 힘들다. 이에 대한 해결책으로 노드 간 에이전트의 전이를 통해 노드의 기능을 변경하여 대상의 이동에 따른 문제를 해결할 수 있다.

본 절에서는 대표적인 이동성 지원 다중 에이전트 미들웨어인 Agilla와 센서 네트워크의 정보전달의 효율성에 있어 유용한 해결방안의 하나인 LEACH 프로토콜을 분석한다.

2.1 Agilla (Mobile Agent Middleware)

Agilla는 무선 센서 네트워크상에서 노드 간 에이전트의 전이를 통해 해당노드의 특정기능을 수행하는 모바일 에이전트 기반의 미들웨어이다. Agilla는 각 노드에서 다중 에이전트를 지원하며 유연성 있게 센서노드들의 상태정보가 퍼지는 것을 컨트롤하고 에이전트에게 Tuple Space와 Neighbor List의 리소스를 제공한다. Tuple Space는 노드에서 수집된 센싱정보의 집합으로써 노드에 종속적이며 에이전트들 간의 통신을 위한 decoupled-style을 제공한다. 튜플은 타입과 값을 가진 필드들의 규정된 세트이며 타입에는 정수, 문자열, 센싱정보 등을 포함한다. 튜플은 에이전트의 템플릿에 대한 패턴매칭을 통해 노드의 수집정보를 제공한다. 또한, Tuple Space는 노드의 센서타입을 명세한 튜플을 두어 에이전트의 전이 시 해당노드의 센서타입을 인지하도록 정의한다. 만약, 노드가 온도센서를 가지고 있다면 Tuple Space에 'temperature tuple'을 삽입한다. Neighbor List는 한 홉 간의 인접노드 주소를 포함하며 원격접근을 통한 인접노드의 센싱정보에 대한 요청과 각

노드들의 상황에 따른 에이전트의 전이에 이용된다. Agilla 어플리케이션은 네트워크상에 뿌려진 수많은 에이전트들로 이루어져 있다. 에이전트가 해당노드에 전이하면 자신의 상태와 코드를 저장하고 리액션 템플릿을 Tuple Space에 삽입시켜 튜플과의 패턴매칭에 따른 매칭 시 리액션 기능을 발동시켜 해당 기능을 수행하게 된다. 그림 1은 Agilla 모델을 보여준다[2,3].

2.2 LEACH

LEACH 프로토콜은 주변노드들 간의 자기구성을 통한 계층적인 구조를 형성하는 클러스터링 알고리즘을 기반으로 네트워크의 범위성을 용이하게 하여 정보전달의 효율성을 높이고 센서노드간의 에너지 소모를 균등하게 함으로써 네트워크 생존시간을 최대화하여 제한된 전원에 대한 유용한 해결책을 제시한다. LEACH 프로토콜 기반의 네트워크는 임의의 클러스터를 구성하고 클러스터마다 하나의 헤드 노드를 선출한다. 클러스터 내부의 일반 노드들은 클러스터 헤드 노드로 데이터를 전송하고 클러스터 헤드 노드는 이를 병합하여 싱크에게 직접 전송한다. 에너지 소모가 균등하게 이루어지도록 일정 시간 마다 클러스터를 재구성하고 헤드 노드를 재 선출한다. 클러스터 내부의 일반 노드들은 클러스터 헤드 노드로 데이터를 전송하고 클러스터 헤드 노드는 이를 병합하여 싱크에게 직접 전송한다.

LEACH의 실행 단계는 라운드로 이루어져 있으며 클러스터를 구성하는 Set-Up Phase[4]로 시작하여 베이스 스테이션에 데이터를 전송하는 Steady-State Phase가 이어진다. Set-Up Phase에서는 모든 노드가 다음의 확률함수를 사용하여 클러스터 헤드를 결정하고 클러스터 헤드가 이웃 노드에 광고 메시지를 Broadcast하여 클러스터를 구성한다.

$$P_i(t)=\begin{cases}\dfrac{k}{N-k^*\left(\mathrm{r}\bmod\dfrac{N}{k}\right)} & : C_i(t)=1\\ 0 & : C_i(t)=0\end{cases}$$

위 식에서 i는 노드의 식별자, t는 시각, N은 전체 노드의 수, k는 클러스터의 수, r은 라운드를 나타낸다. Ci는 최근 r mod (N/k) 라운드 동안 클러스터 헤드였다면 0이고, 아니라면 1이다. 즉, 최근 r mod (N/k) 라운드 동안 헤드를 한 번이라도 했다면 다시 뽑힐 확률은 없는 것이다. 이와 같은 확률함수를 통해 자신이 클러스터 헤드라면 이를 주변의 이웃노드에게 브로드캐스트 한다. 이때 매체 접근 제어(MAC) 프로토콜은 CSMA 방식을 사용한다. 광고 메시지를 수신한 비 헤드노드들은 자신이 속할 클러스터의 헤드 노드를 선택하고 이를 헤드 노드에게 알린다. Steady-State Phase는 주어진 시간슬롯 동안 클러스터의 모든 노드가 최소한 한번은 클러스터 헤드에 데이터를 전송하도록 하기 위해 프레임 단위로 나뉘어져 있다. 각 프레임은 데이터 전송을 위해 클러스터의 노드 수에 기반한다. 클러스터가 구성되고 TDMA 스케줄이 정해지면 데이터 전송이 시작된다. 여기서 모든 노드는 주어진 시간에 클러스터 헤드에 데이터를 전송한다고 가정한다. 모든 비 클러스터 헤드 노드는 에너지 소모를 최소화하기 위해 주어진 시간 단위까지 turn-off 된다. 모든 데이터가 수신되면 클러스터 헤드는 모든 수신 데이터를 하나의 신호에 적재하기 위한 신호 처리를 수행한다. 이로써 Steady-State Phase를 마치고 다음 라운드 단계인 Set-Up Phase가 반복된다[5].

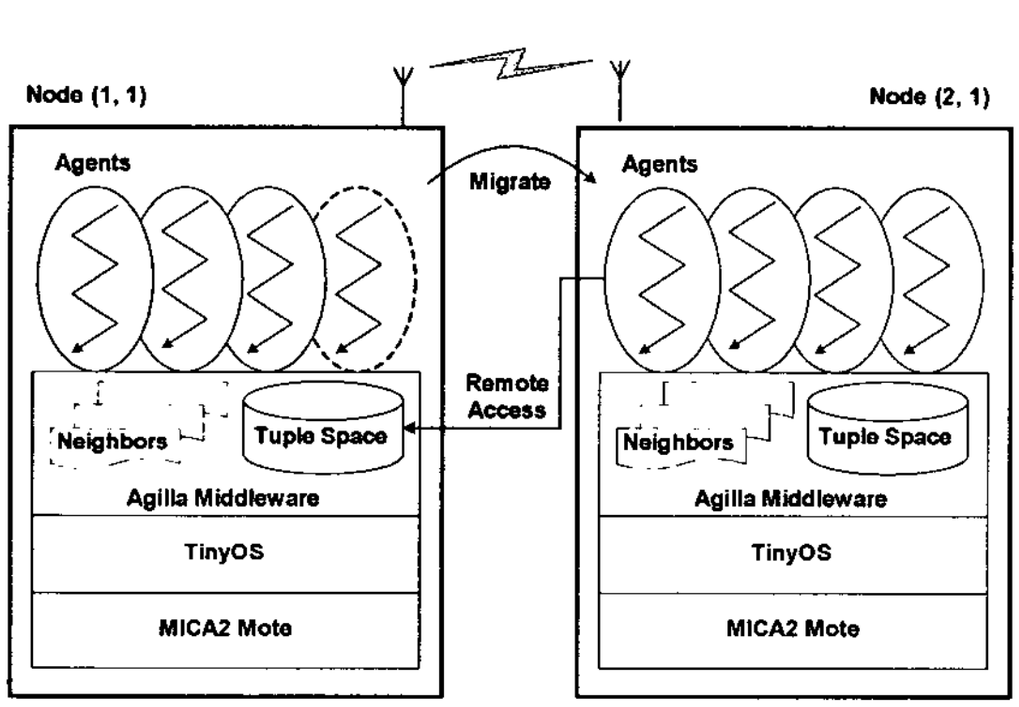

그림 1. The Agilla Model

3. 헬스케어 응용 서비스를 위한 Mobility Agent 모듈 설계

최근 헬스케어는 환자 중심의 질병, 증상을 완화와 치료에서 점차 일반인의 건강을 증진하고 질병을 예방하는 개념으로 변화되고 있다. 이를 위해서 정보통신 기술과 의료 서비스를 융합하여 언제 어디서나 예방, 진료, 치료 그리고 사후관리를 받을 수 있는 헬스케어 응용 서비스가 제공되어야 한다. 즉, 댁내

의 사람과 사물 같은 객체의 위치를 인식하고 이를 기반으로 연속적인 거주자의 상황정보에 따른 맞춤형 헬스케어 서비스를 제공해야 한다[6].

본 논문에서는 헬스케어 환경을 지원하기 위하여 사람의 이동에 따른 연속적인 위치변화와 상태정보 그리고 환경변화에 대한 동적 기능변경과 이동성을 지원한 Mobility Agent 모듈을 설계한다.

3.1 Mobility Agent 모듈 설계

본 논문에서는 헬스케어를 위한 센서 네트워크를 지원하기 위해 이동노드의 동적인 기능변경을 보장하는 Mobility Agent 모듈을 설계하였다. Mobility Agent 모듈은 사람의 상태와 주변상황의 변화에 따른 노드의 기능변경을 지원하는 동적 기능변경 모듈인 Mobility Agent Middleware를 설계하여 센서노드 간의 에이전트 전이를 통한 동적 기능변경을 지원하였다. 또한 사람의 이동에 따른 네트워크 타입의 변화와 정보 전달의 효율성을 보장하기 위해 LEACH 프로토콜의 클러스터 알고리즘을 따르며 노드의 이동성을 보장하는 LEACH_Mobile 프로토콜을 지원하는 이동노드 지원 라우팅 모듈인 LEACH_Mobile 라우팅 모듈을 Mobility Agent Middleware의 하위 계층으로 구성하여 노드의 이동성을 보장하고 정보전달의 효율성을 지원하였다. 그림 2는 Mobility Agent Module Architecture를 보여준다.

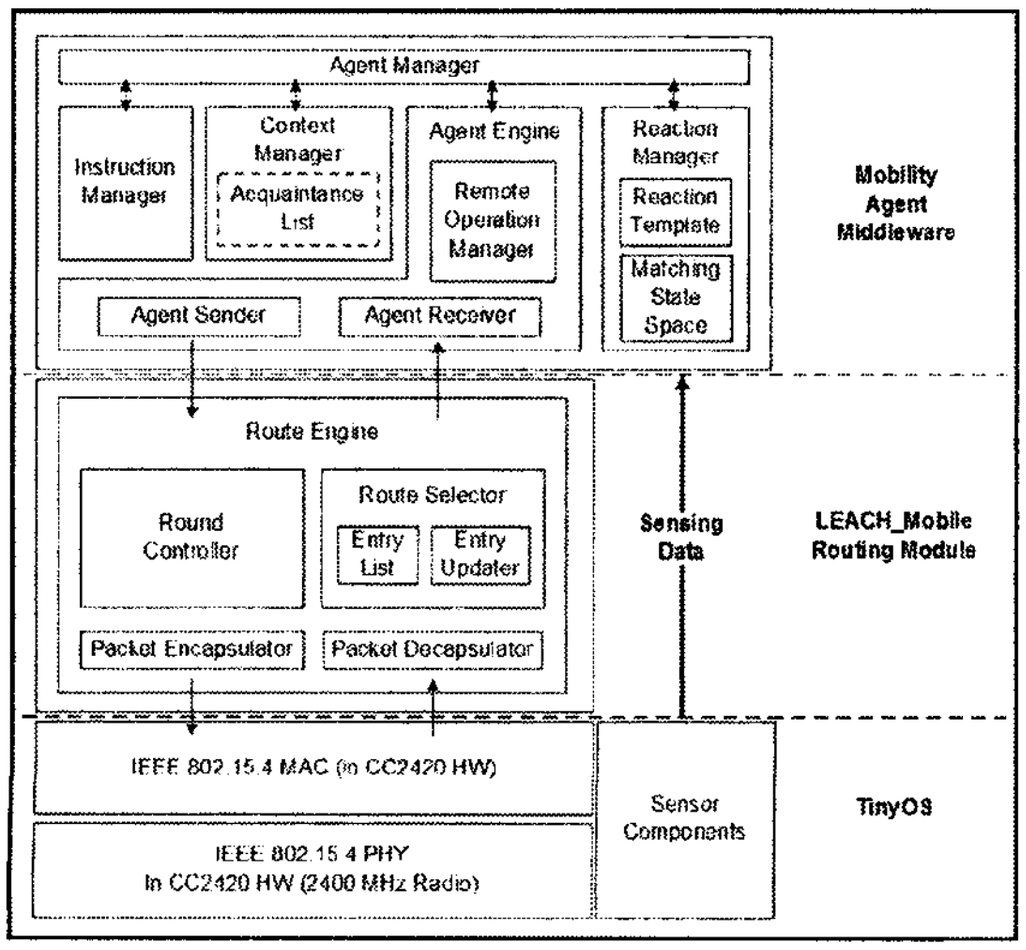

그림 2. Mobility Agent Module Architecture

3.2 이동노드 지원 라우팅 모듈 설계

LEACH 프로토콜이 계층적 클러스터를 구성하여 좋은 에너지 효율성과 동적 자기 구성이라는 장점을 갖지만, 헬스케어를 위한 노드의 이동성을 지원하는 환경에서 데이터 전송 성공률 면에 대한 이득을 갖지 못한다. LEACH는 Set-Up Phase에서 클러스터가 재구성되고 클러스터 헤드가 선택된 후 Steady-State Phase에서 베이스 스테이션으로 실제 데이터가 전송되기 위해 클러스터의 구성이 유지되어 클러스터 상의 노드들이 이동했을 시에 대한 변화에 대응하지 못한다. 이 같은 문제점은 클러스터 헤드가 선택된 후 노드의 이동성을 지원하지 못하는 결과를 초래한다[7]. 이는 LEACH 프로토콜에 이동 노드가 현재 클러스터의 참여에 대한 확인을 요청함으로써 Set-Up Phase 이후 노드의 이동성을 보장하는 LEACH_Mobile 프로토콜로써 해결 가능하다.

3.2.1 LEACH_Mobile 프로토콜

LEACH 프로토콜에서는 클러스터 헤드가 Steady-State Phase인 동안 TDMA 스케줄에 의한 데이터 수신을 기다리지만, LEACH-Mobile에서는 클러스터 헤드 노드가 데이터 수집을 위한 데이터 전송 요청 메시지를 비 클러스터 헤드 노드에 전송한다. 요청을 받은 노드는 데이터를 전송하고, 클러스터 헤드는 프레임이 끝날 때 마다 데이터를 수신하지 못한 노드를 비 수신 노드 리스트에 추가한다. 다음 프레임이 끝날 때까지 비수신 노드 리스트의 노드로부터 데이터를 전송 받지 못하면 해당 노드를 클러스터 멤버에서 제거하고 새로 참여하는 노드에 이 시간 슬롯을 할당한다. 이 후 새로 생성된 TDMA 스케줄은 모든 클러스터의 멤버 노드에 전송된다. 이 방법은 데이터 요청 메시지에 응답하지 않는 노드는 이동하여 현재 클러스터 영역을 벗어났다고 가정한 것이다.

클러스터 헤드 노드가 데이터 요청 메시지로 자신의 클러스터에 속한 노드의 참여 여부를 검사하는 동안, 각 이동 노드들은 자신이 속할 클러스터를 결정한다. 클러스터가 구성되고 클러스터 헤드가 결정되면, 센싱노드는 데이터 요청 메시지에 대한 응답으로 데이터를 전송한다. 프레임이 끝날 때까지 데이터 요청 메시지를 TDMA 스케줄에 의해 할당된 시간 슬롯에서 수신하지 못하면, 프로토콜 수행 과정을 다음 프레임으로 진행시킨다. 다음 프레임이 끝날 때까

지 이동 노드가 데이터 요청 메시지를 수신하지 못하면, 클러스터 참여 요청 메시지를 Broadcast한다. 참여 요청 메시지를 수신한 클러스터 헤드 노드는 Set-Up Phase와 같이 광고 메시지를 전송한다. 이 과정이 완료되면 이동 노드는 현재 라운드에서 속하게 될 클러스터를 결정하게 된다. 이 결정과정은 수신된 광고 메시지의 수신 세기에 기반하며 이동 노드가 자신이 속할 클러스터를 결정하게 되면 해당 클러스터 헤드 노드에 클러스터에 참여하였음을 공지한다. 이동 노드가 새로 참여하게 된 해당 클러스터의 헤드 노드는 클러스터 멤버 리스트와 TDMA 스케줄을 갱신하고, 자신에 속한 모든 멤버 노드에 TDMA 스케줄을 전송한다. 새로 참여한 이동 노드를 비롯한 모든 멤버 노드들은 다음 프레임부터 TDMA 스케줄을 갱신한다. 그림 3은 LEACH-Mobile 프로토콜의 메시지 교환 과정을 보여준다.

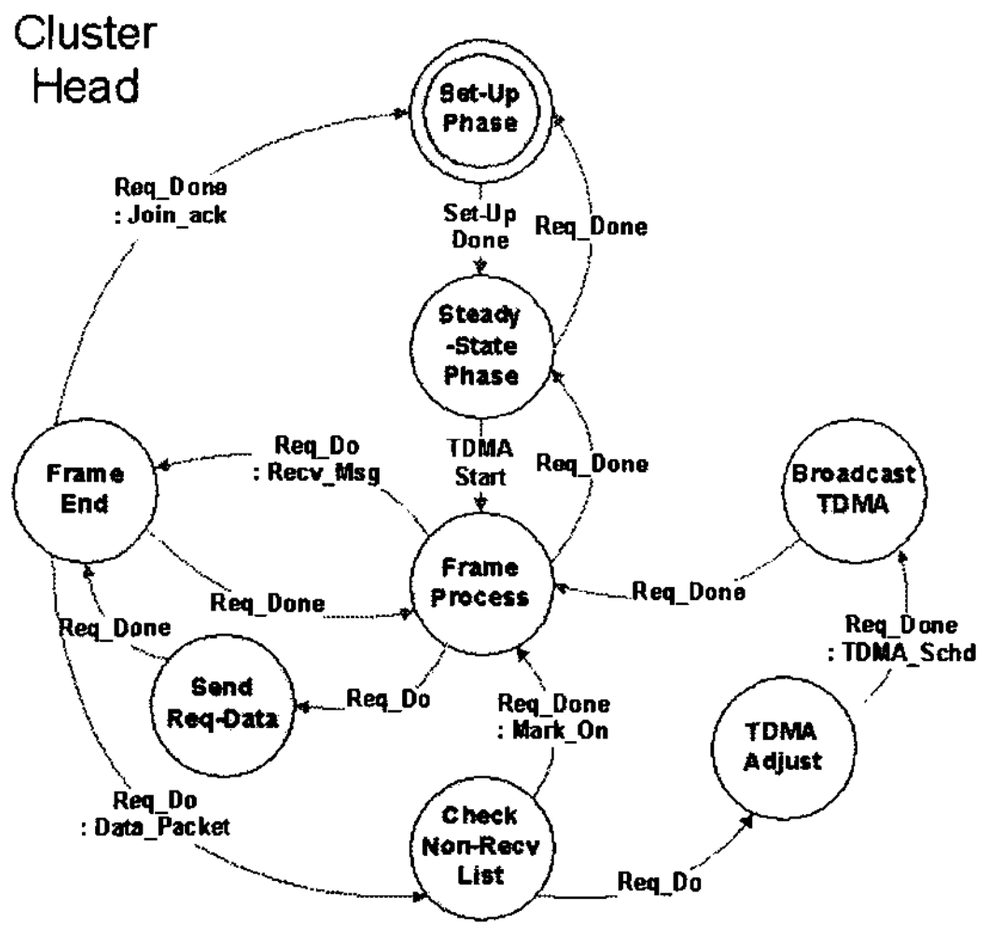

그림 3. LEACH_Mobile 프로토콜의 메시지 교환 과정

3.2.2 LEACH_Mobile Routing Module

LEACH_Mobile Routing Module은 헬스케어 환경에서 이동 노드를 지원토록 제안된 LEACH_Mobile 프로토콜을 설계한 모듈로써 이동하는 노드가 클러스터 헤드로부터 이동노드에 데이터 요청 메시지를 전송하여, 특정 클러스터 헤드 노드와 통신이 가능한지의 여부로서 클러스터의 멤버가 될 노드를 결정하고 클러스터를 재구성하여 multihop path를 제공한다. 그림 4는 LEACH_Mobile Routing Module을 보여준다.

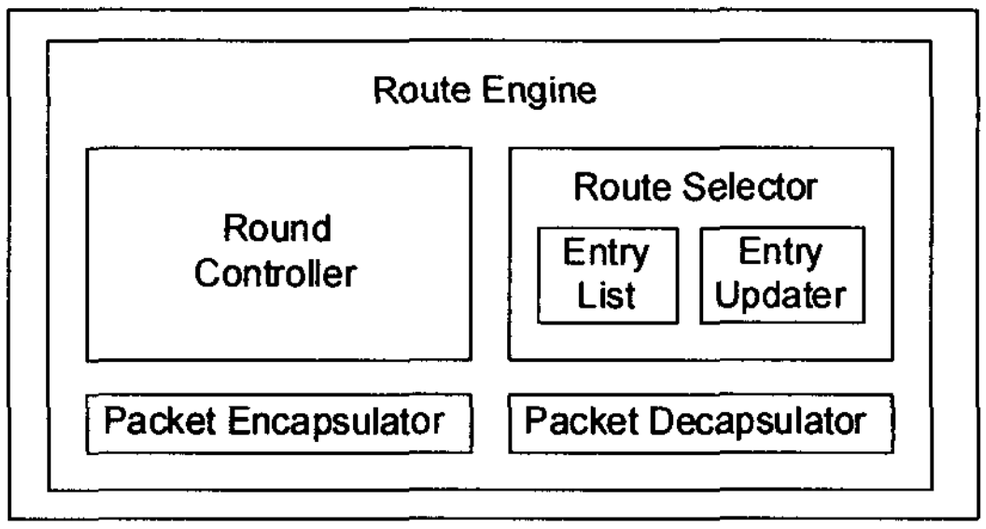

그림 4. LEACH_Mobile Routing Module

Route Engine은 TDMA 스케줄에 따른 라운드 컨트롤을 위한 Round Controller와 트리 구소에서 상위 노드의 존재여부를 판단하여 최신의 상황에 맞는 라우팅 경로를 제공하는Route Selector를 포함하며, 클러스터 헤드 선출을 위한 광고 메시지, 비 클러스터 헤드에 대한 데이터 전송 요청 메시지, 이동노드의 클러스터 참여 요청 메시지 등을 초기화 및 캡슐화하며 각 노드에 주어지는 TDMA 스케줄의 시간슬롯 조정 등의 라우팅에 관련된 전반적인 관리 역할을 수행한다.

3.3 동적 기능변경 모듈 설계

헬스케어 환경을 지원하기 위해서는 사람의 상태와 주변상황 변화에 따른 노드의 능동적 기능변경이 요구된다. 본 절에서는 노드 간 에이전트 전이를 통해 동적 기능변경을 지원하는 Mobility Agent Middleware를 설계하였다. Mobility Agent Middleware는 각 노드 상에서 다중 에이전트를 지원

하며 노드 간 에이전트의 전이를 통해 노드의 기능변경을 지원해주는 일종의 가상머신 모듈이다. Mobility Agent Middleware는 특정 기능을 수행하고 있던 노드에 에이전트가 도착하면 에이전트의 메모리를 할당하며 해당노드가 에이전트의 특정기능에 따른 센싱이 가능한 노드인지 검증하고 기능수행이 가능하다면 에이전트의 명령코드를 삽입하고 Agent Engine의 스케줄링을 통해 에이전트의 명령을 실행하게 된다. 에이전트의 명령은 세 가지 종류로 나뉘는데 일반 명령(general_instruction)의 경우, 해당노드 상에서 처리되는 보편적인 명령으로 즉각적인 실행 후 다음 명령코드로 권한을 넘겨주며 리액션 명령(reaction_instruction)의 경우, 인접노드의 센싱정보에 대한 원격요청을 하여 인접노드들의 센싱정보를 획득하거나 자신의 리액션 템플릿에 리액션 코드를 삽입하여 센싱정보가 매칭되면 리액션 이벤트를 발생시켜 해당 이벤트에 따른 명령코드를 실행시킨다. 그리고 전이 명령(migration_instruction)의 경우, 인접노드들의 위치정보를 확인하여 해당 에이전트를 인접노드로 전이시키고 전이된 에이전트의 메모리를 반환함으로써 수행노드에 대한 다른 에이전트들의 전이를 보장한다. 그림 5는 Mobility Agent Middleware의 상태천이를 보여준다.

Mobility Agent Middleware는 에이전트의 기능에 대한 전반적인 관리를 해주는 Agent Manager와 에이전트의 기능을 실행해주는 Agent Engine, 에이전트에 명세된 명령코드를 관리하는 Instruction Manager, 인접노드의 주소와 위치에 대한 관리를 하는 Context Manager, 마지막으로 노드의 센싱정보에 대한 관리를 해주며 매칭되는 정보의 처리를 요청하는 Reaction manager로 이루어진다. 그림 6은 Mobility Agent Middleware를 보여준다.

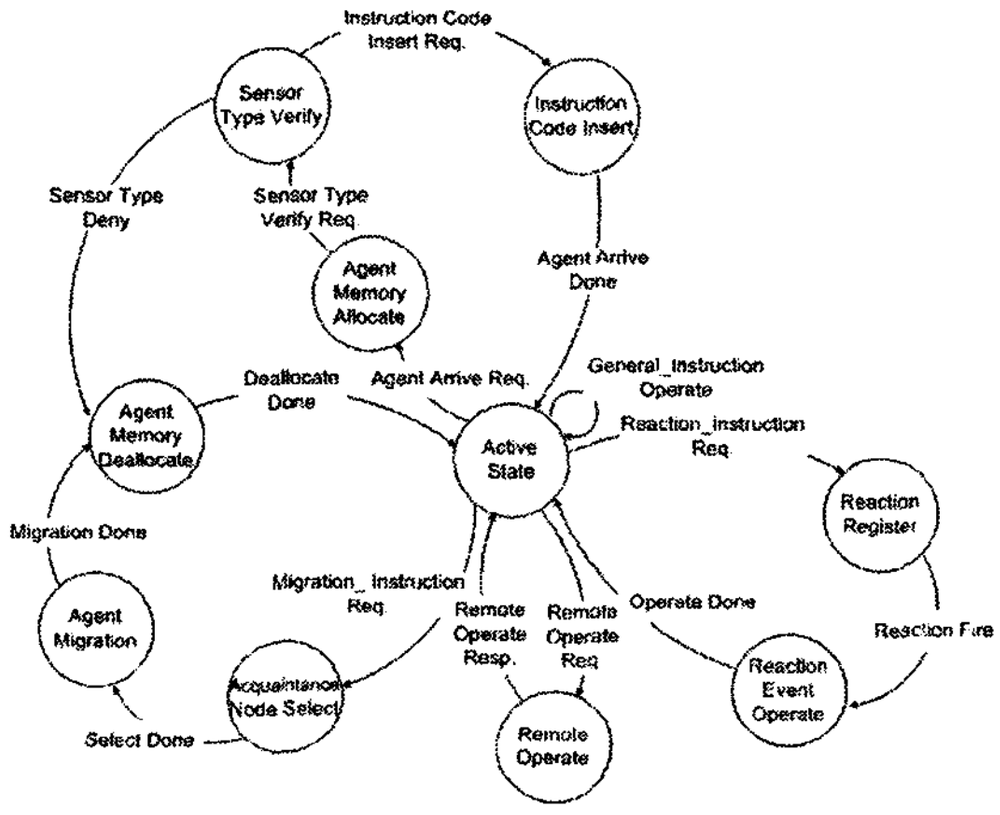

그림 5. Mobility Agent Middleware State Transition Diagram

Agent Manager는 등록된 에이전트의 전이에 따른 메모리 할당과 반환, 실행에 대한 준비, 그리고 이벤트 발생에 따른 명령을 Agent Engine에게 통보하여 에이전트에 대한 전반적인 관리를 수행한다. Agent Engine은 가상 머신 커널로 노드의 모든 에이전트 간 동시병행을 지원한다. 그 구성요소로 인접노드의 Matching State Space에 대한 원격 작동을 담당하는 Remote Operation Manager와 에이전트의 송수신을 담당하는 Agent Sender, Agent Receiver를 포함하며 에이전트의 기능에 대한 실제적인 수행을 조정한다. Instruction Manager는 에이전트가 전이되면 에이전트에 명세된 명령코드를 비축하여 이벤트의 발생이나 에이전트의 실행에 따른 명령코드를 Agent Manager에게 전달하는 방식으로 명령코드에 대한 관리를 수행한다. Reaction manager는 Matching State Space와 Reaction Template을 구성요소로 포함하고 하나의 matching state가 삽입될 때마다 리액션 템플릿에 매칭되는지 체크한다. 만약, 새로운 matching state가 리액션 템플릿과 매칭되면 이에 따른 리액션 이벤트를 발생시켜서 Agent Manager에게 통보하여 노드의 수집정보와 특정 이벤트의 발생에 대한 관리를 수행한다. Context Manager는 구성요소로 인접노드들의 위치정보에 대한 리스트인 Acquaintance List를 포함한다. Agent Manager의 요청에 따라 인접노드들의 주소와 위치를 축적하거나 제공하는 역할을 수행한다.

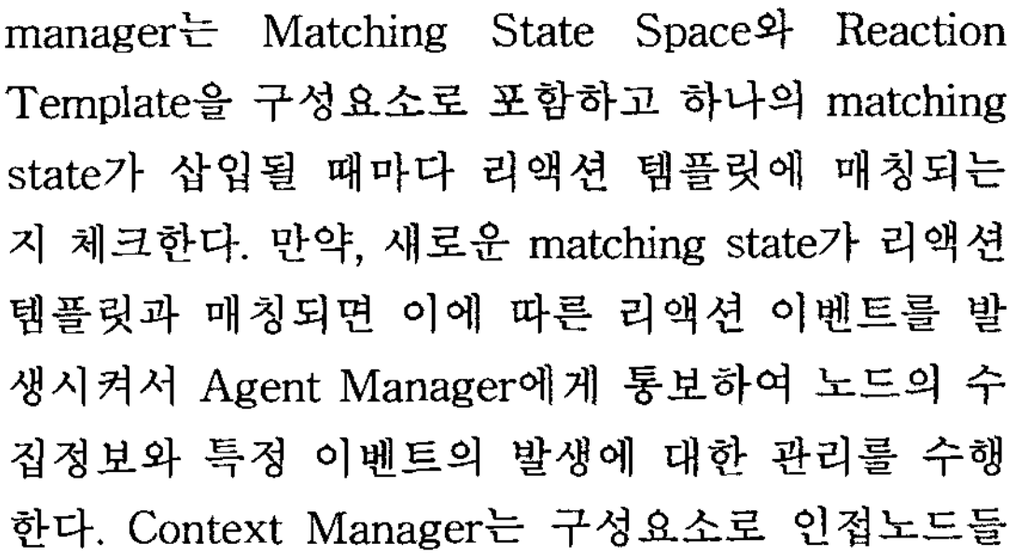

그림 6. Mobility Agent Middleware

3.4 Mobility Agent Middleware와 LEACH_Mobile Routing Module 간 인터페이스

Mobility Agent Module을 구성하는 Mobility Agent Middleware와 LEACH_Mobile Routing Module은 인터페이스를 통한 메시지 교환으로 상호간의 통신을 지원한다. 두 모듈간의 메시지는 에이전트의 전이에 따른 메시지와 인접노드에 대한 Remote Operation 메시지, 그리고 인접노드의 위치정보 요청 메시지로 분류된다. 그림 7은 Mobility Agent Middleware와 LEACH_Mobile Routing Module간의 메시지 플로우를 보여준다.

노드 간 에이전트의 전이에 따른 메시지의 경우, 에이전트의 전이 시 미들웨어는 agent_send() 메시지를 통해 라우팅 모듈로 에이전트를 전달하며 이때 메시지의 인자로 목적노드의 ID와 해당 에이전트의 ID와 상태정보, 코드, 에이전트의 기능을 수행하기위한 리액션 템플릿 등을 갖는다. 그리고 에이전트를 수신한 노드의 라우팅 모듈은 agent_receive() 메시지를 통해 미들웨어로 에이전트를 전달하여 에이전트의 전이과정을 수행한다.

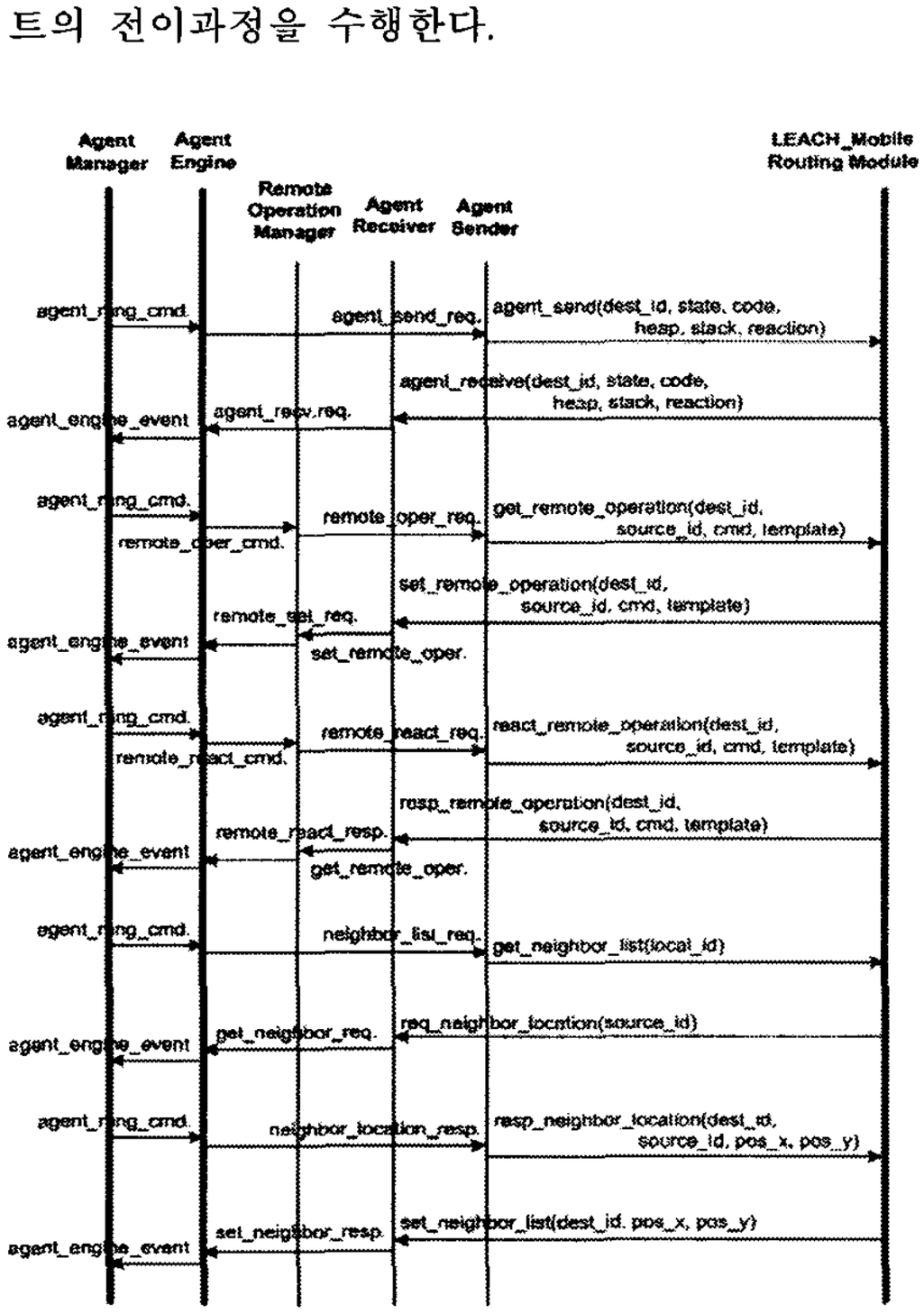

그림 7. Mobility Agent Middleware와 LEACH_Mobile Routing Module 간 메시지 플로우

인접노드의 상황정보에 대한 요청 메시지의 경우, 상황정보에 대한 요청 시 미들웨어는 get_remote_operation() 메시지를 상황정보에 대한 요청노드 ID와 현재 노드 ID, 명령코드 그리고 목적노드의 리액션 템플릿에 삽입되어 상황정보의 확인에 이용되는 템플릿을 인자로 하여 라우팅 모듈로 전달하고 이를 수신한 목적노드의 라우팅 모듈은 set_remote_operation() 메시지를 미들웨어로 전달하여 목적노드의 상황정보에 대한 리액션 템플릿을 삽입하게 된다. 목적노드의 상황정보에 대한 Reaction 이벤트가 발생하면 미들웨어는 라우팅 모듈로 react_remote_operation() 메시지를 전달하여 원격 작동(remote operation)을 요청한 노드로 확인 메시지를 전송하게 된다. 원격 작동을 요청한 노드는 확인 메시지를 수신하면 라우팅 모듈에서 미들웨어로 resp_remote_operation() 메시지를 전달하여 상황정보에 대한 응답을 수신하게 된다.

인접노드의 위치정보에 대한 요청 메시지의 경우, 미들웨어는 라우팅 모듈로 get_neighbor_list() 메시지를 메시지 송신 노드의 ID를 인자로 하여 전달하고 인접노드들에게 위치정보에 대한 응답을 보내게 한다. 이 요청 메시지를 수신한 노드의 라우팅 모듈에서는 req_neighbor_location() 메시지를 미들웨어로 전달하고 자신의 노드 위치정보를 resp_neighbor_location() 메시지를 통해 라우팅 모듈에 전달하여 위치정보를 요청한 노드로 전송한다. 이 경우 해당노드는 자신의 위치정보를 갖고 있다는 것을 가정한다. 위치정보를 요청한 노드에서 위치정보가 수신되면 라우팅 모듈에서는 set_neighbor_list() 메시지를 통해 위치정보를 미들웨어로 전달하여 위치정보에 대한 요청과정을 완료한다.

4. LEACH_Mobile 시뮬레이션

본 논문에서 제안한 프로토콜의 성능 검증을 위해 이동성을 지원하는 LEACH-Mobile 프로토콜과 LEACH 프로토콜을 네트워크 시뮬레이터 NS-2에 구현하였다.

4.1 라디오 모델

본 논문에서는 시뮬레이션을 위해 [4]에서 논의된 *First Order Radio Model*을 사용하였다. 이 라디오 모델

에서 라디오는 데이터 송수신에 $E_{elec} = 50nJ/bit$, 전송 증폭에 $\epsilon_{amp} = 100pJ/bit/m^2$의 에너지를 소비한다. 라디오는 전력 제어가 가능하며 수신자에 도달하기까지 필요한 최소 에너지를 증가시킬 수 있다. 또한, 원하지 않는 전송의 수신을 피하기 위해 라디오를 turn-off 할 수 있다. 아래의 식은 k bit의 메시지를 거리 d에 전송할 때, 송수신에 필요한 비용을 표현한 식이다.

Transmitting:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d)$$

$$E_{Tx}(k,d) = E_{elec} * k + \epsilon_{amp} * k * d^2$$

Receiving:

$$E_{Rx}(k) = E_{Rx-elec}(k)$$

$$E_{Rx}(k) = E_{elec} * k$$

LEACH와 LEACH-Mobile은 에너지 비용 계산에 같은 상수(E_{elec},ϵ_{amp}, and k)들을 사용한다. 위 라디오 모델에서 LEACH는 모든 노드가 베이스 스테이션에 메시지가 도달하기 위한 충분한 에너지를 소모한다고 가정하지만, 실제 상황에서 각 노드는 제한된 전원과 베이스 스테이션 사이의 거리에 의해 라디오 전송에 제약을 가질 수 있다.

이런 이유로 본 논문의 시뮬레이션에서는 수신 전력에 임계치를 설정함으로써 LEACH와 LEACH-Mobile의 라디오 모델에서 전송 거리를 제한하였다. 노드가 데이터를 수신할 때 수신된 전력이 감지 임계치 ($P_{r-detect}$)를 넘지 못하면, 패킷을 수신하지 못했다고 가정하고 패킷을 버린다. 수신된 전력이 감지 임계치를 넘지만 수신 성공 임계치 ($P_{r-thresh}$)를 넘지 못하면, 수신 실패로 간주한다. 수신 전력이 수신 성공 임계 ($P_{r-thresh}$)를 넘으면 패킷 수신에 성공한 것으로 한다[5].

이 같은 경우에 LEACH와 LEACH-Mobile은 전송거리에 제한을 갖기 때문에 멀티 홉 라우팅이 필요하게 된다. LEACH 프로토콜에서 멀티 홉 라우팅을 고려한 LEACH-M 혹은 M-LEACH 프로토콜은 [7]에서 논의되었다. 본 논문에서는 LEACH와 LEACH-Mobile은 베이스 스테이션에 도달하기 까지 멀티 홉 라우팅을 사용한다고 가정하였다.

4.2 시뮬레이션 결과

헬스케어 응용 서비스를 위한 센서 네트워크에서는 주로 사람의 상태와 행동에 대한 센싱정보를 처리하므로 이동성 노드의 데이터 전송 성공률 면에서 높은 성능을 보여야 한다.

이러한 측면에서 LEACH_Mobile의 성능을 검증하기 위해 본 논문에서는 50m × 50m의 네트워크 영역에 100개의 노드를 랜덤 배치하여 LEACH_Mobile과 LEACH에 대한 시뮬레이션을 수행하였다. 베이스 스테이션은 전체 50m × 50m 필드에서 (25, 25) 지점에 위치하게 했다. 시뮬레이션은 각 노드로부터 베이스 스테이션에 도달하는데 성공한 데이터 패킷의 수와 데이터 패킷 전송에 소모된 에너지량을 LEACH와 LEACH_Mobile에 이동 노드의 수를 25%, 50%, 75%로 증가시키면서 수행하였다. 여기서 이동 노드는 적은 에너지 레벨을 갖고, 고정 노드는 지속적이고 높은 에너지 레벨을 갖도록 설정하였다.

그림 8은 이동 노드가 증가함에 따른 데이터 전송 성공률의 변화를 보여준다. 그림 9는 동일한 매개변수를 가진 에너지 소모량을 보여준다. 그래프에서 보이듯이 LEACH_Mobile 프로토콜은 이동노드가 증가함에 따라 데이터 전송 성공률 면에서 LEACH 프

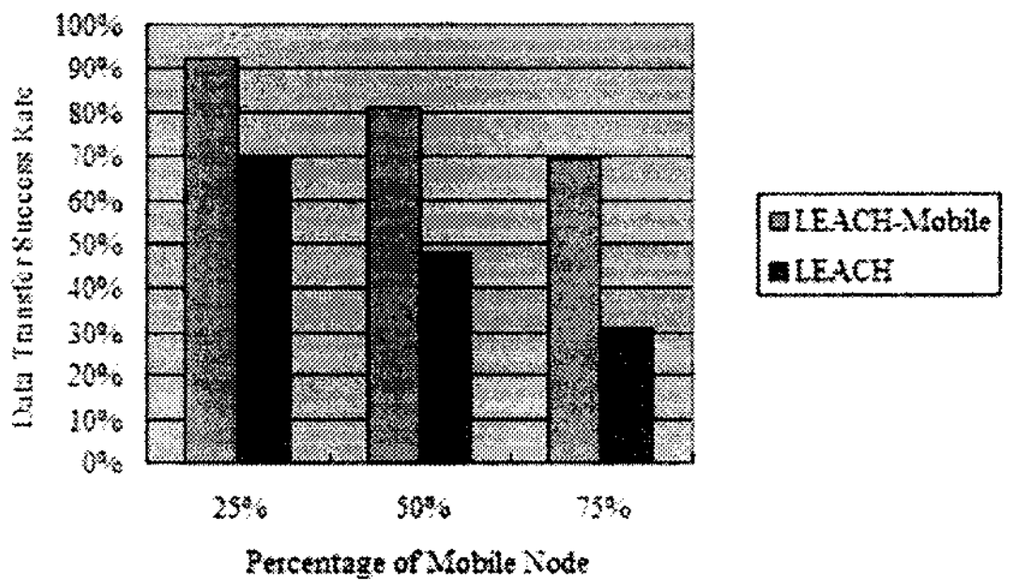

그림 8. 데이터 전송 성공률 시뮬레이션 결과

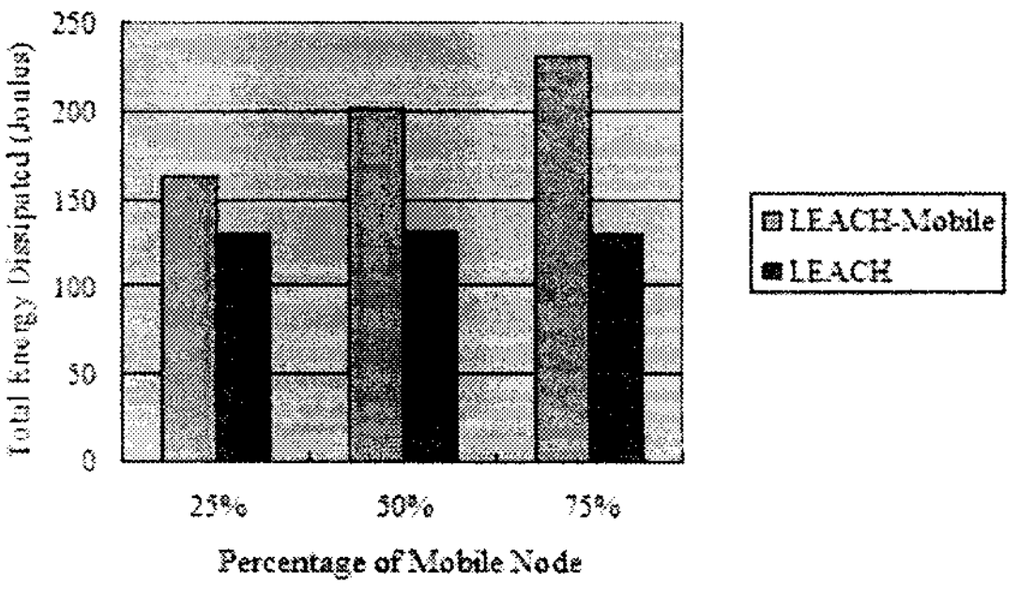

그림 9. 에너지 소모량 시뮬레이션 결과

로토콜에 비해 30%에서 120%의 명확한 성능 향상을 가져온다. 그러나 전체 전송 데이터 패킷 수가 증가함에 따라 LEACH와 비교하여 에너지 소모량 또한 증가함을 알 수 있다. 이러한 결과를 통해 LEACH_Mobile 프로토콜은 LEACH 프로토콜에 비해 네트워크 서비스의 질(QoS) 측면에서 전송 성공률에 대한 성능 향상을 가져온다고 기대된다.

5. 결 론

본 논문에서는 헬스케어 환경에서 특정 목적에 따른 노드의 동적 기능변경을 제공하며 다양한 환경 및 상황에 따른 서비스 재구성을 지원하는 Mobility Agent 모듈을 설계 제안하였다.

본 논문에서 제안한 Mobility Agent Module은 헬스케어 홈, 병원, 요양원 등의 헬스케어 환경에서의 응용 서비스를 가정하였다. 헬스케어 환경에서는 사람의 이동을 고려한 건강상태 및 상황정보 등 중요한 센싱정보를 처리하게 된다. 그에 따라 헬스케어 환경에 대한 응용 서비스를 제공하는 모듈은 노드에 대한 에너지 효율성보다는 센싱대상의 이동에 따른 데이터 전송 성공률 면에서 높은 성능을 보여야 한다. 이러한 요구조건에 대해 Mobility Agent 모듈은 LEACH_Mobile 라우팅 프로토콜을 사용함으로써 높은 데이터 전송 성공률을 보장한다. 이러한 성능을 기반으로 Mobility Agent 모듈은 센서노드들의 이동에 따른 정보전달의 효율성과 센서노드들 간의 센싱 정보를 처리하는 에이전트들의 이동성을 모두 고려하여 보다 안정되고 동적인 헬스케어 응용 서비스를 지원할 수 있다.

향후 본 연구를 기반으로 헬스케어 환경에서 센싱 노드에 대한 이동성과 센싱기능의 동적 기능 변경을 지원하는 Mobility Agent 모듈을 구현한다. 또한, 구현한 모듈의 성능 분석을 통해 다양한 서비스 목적에 따른 동적 기능 변경 미들웨어의 효율성을 입증한다.

참 고 문 헌

[1] 김도성, 정영지, "모바일 센서 노드 지원을 위한 자기구성 라우팅 프로토콜," *한국정보처리학회 춘계 학술발표대회 논문집*, 제13권, 제1호, pp. 1295-1298, 2006.

[2] Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu, "Mobile Agent Middleware for Sensor Networks," *In Proceedings of the 4th International Conference on Information Processing in Sensor Networks*, pp. 382-387, 2005.

[3] Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu, "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Application," *In Proceedings of the 24th International Conference on Distributed Computing Systems*, pp. 653-662, 2005.

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient Routing protocols for wireless microsensor networks," *In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Vol.2, pp. 908-918, 2000.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Comm. IEEE*, Vol.1, No. 4, pp. 660-670, 2002.

[6] 안동인, 신창선, 주수종, "헬스케어 홈 서비스를 위한 실내위치 기반의 상황정보 지원 시스템," *한국컴퓨터종합학술대회 논문집*, 제33권, 제1호, pp. 64-66, 2006.

[7] V. Mhatre and C. Rosenberg, "Homogeneous vs Heterogeneous Clustered Sensor Networks," *A Comparative Study, 2004 IEEE International Conference on Communications*, Vol.6, pp. 3646-3651, 2004.

남 진 우

2008년 2월 원광대학교 전기전자 및 정보공학부 컴퓨터공학과 졸업 (공학사)
2008년 3월~현재 원광대학교 컴퓨터공학과 석사과정
관심분야 : 센서 네트워크, RFID, 유비쿼터스 컴퓨팅

정 영 지

1982년 연세대학교 전기공학과 졸업 (공학사)
1984년 연세대학교 전기공학과 졸업 (공학석사)
1993년 연세대학교 전기공학과 졸업 (공학박사)
1984년~1987년 금성계전(주) 연구소 주임 연구원
1987년~1993년 삼성 종합기술원 소재부품 연구소 선임 연구원
1993년~1995년 한국 전자 통신 연구소 이동통신 기술 연구단 무선 제어 연구실 선임 연구원
1997년~1999년 Visiting Professor at MPRG, Virginia Tech.
1995년~2006년 3월 원광대학교 공과대학 전기전자 및 정보공학부 부교수
2006년 4월~현재 원광대학교 공과대학 전기전자 및 정보공학부 정교수
관심분야 : 이동통신 네트워크, 센서 네트워크, 텔레매틱스, LBS