

Lightweight and Migration Optimization Algorithms for Reliability Assurance of Migration of the Mobile Agent

Yon-Sik Lee*

*Professor, School of Computer Information and Communication Engineering, Kunsan National University, Kunsan, Korea

[Abstract]

The mobile agent, which handles a given task while migrating between the sensor nodes, moves including the execution commands and task processing results. This increases the size of the mobile agent, causing the network to load, leading to the migration time delay and the loss of migration reliability. This paper presents the method of lightening the mobile agent using distributed object technology and the algorithm for exploring and providing the optimal migration path that is actively performed in the event of network traffic, and it proposes a method to ensure the reliability of the mobile agent migration by applying them. In addition, through the comparative analysis experiments based on agent size and network traffic for the migration time of mobile agent equipped with active rules in sensor network-based mobile agent middleware environment, applying the proposed methods proves to ensure the autonomy and migration reliability of the mobile agent.

▶ **Key words:** Mobile agent, Lightweight of agent, Distributed object, Optimal migration path, Migration reliability

[요 약]

이동성 객체인 이동에이전트는 이주 시 주어진 작업처리 명령들과 작업수행 결과 탑재에 의한 크기 증가로 인하여 네트워크의 부하를 유발함으로써 이주시간 지연 및 이주 신뢰성을 훼손한다. 본 논문에서는 분산객체 기술을 이용한 이동에이전트의 경량화 방법과 네트워크 트래픽 발생 시 능동적으로 최적의 이주경로를 탐색 및 제공하는 알고리즘을 제시하고, 에이전트 이주 시 이들을 적용함으로써 이동에이전트의 이주 신뢰성을 보장하는 방법을 제안한다. 또한, 센서 네트워크 기반 이동에이전트 미들웨어 환경에서 능동규칙 탑재 이동에이전트의 이주시간에 대한 에이전트의 크기와 네트워크 트래픽 발생에 따른 비교 및 분석 실험을 통하여, 제안 방법 적용 시 이동에이전트의 자율성과 이주 신뢰성이 보장됨을 입증한다.

▶ **주제어:** 이동에이전트, 에이전트 경량화, 분산 객체, 최적 이주경로 탐색, 이주 신뢰성

-
- First Author: Yon-Sik Lee, Corresponding Author: Yon-Sik Lee
 - Yon-Sik Lee (yslee@kunsan.ac.kr), School of Computer Information and Communication Engineering, Kunsan National University
 - Received: 2020. 02. 19, Revised: 2020. 04. 13, Accepted: 2020. 04. 13.

I. Introduction

센서 네트워크의 단말을 구성하고 있는 센서노드(싱크노드 포함)들 사이에서 이동에이전트가 이주할 경우, 노드 장애나 수동적 이주순서 제공으로 인하여 분산 환경에서의 이주, 주어진 작업 실행 및 결과 전송 등에 요구되는 시간비용이 증가하게 된다. 또한 에이전트의 이주 자체만을 보장하는 기존의 이주방법들에서는 노드의 결손 및 장애 발생이나 이주를 위한 정보의 부재로 인하여 에이전트가 무한 대기상태나 고아상태가 되어 불가피하게 폐기되어진다[1,2]. 이를 해결하기 위하여 경로 조정을 위한 후위 복구 및 다수의 무결점 이주 기법들이 제시되었지만 [2,3,4], 이들은 발생된 장애를 처리하는 방식으로써 노드들의 상호 감시기능과 무결점 접속환경이 요구될 뿐 아니라 이주 신뢰성을 보장하기 어렵다. 따라서 에이전트의 이주 신뢰성 보장 및 이주 최적화를 위한 에이전트의 구조 및 최적경로 탐색에 대한 연구들이 필요하다[5,6,8].

본 논문은 이주 신뢰성 확보를 위하여 물리적인 방해요소들을 해소하고 능동적으로 최적경로를 탐색 및 제공하는 알고리즘을 제안하여, 이동에이전트의 자율성과 효율적 기능 수행을 위한 이주 최적화 방법을 설계 및 구현한다. 또한, 분산 객체 개념을 적용한 이동에이전트와 네트워크 트래픽 부하의 실시간 측정을 통하여 에이전트의 이주 시 네트워크의 부하 감소를 유도하는 에이전트 경량화 방법을 제안한다.

논문의 구성은 2장에서 분산객체 개념을 적용한 이동에이전트와 센서 네트워크 기반의 이동에이전트 미들웨어 환경에서의 이동에이전트 기능 수행을 설명하고, 3장에서 이동에이전트의 이주 및 최적경로 탐색 및 조정 알고리즘을 제안하고, 4장에서는 제안 방법들을 적용하여 에이전트의 이주시간을 에이전트 크기와 네트워크 트래픽을 기준으로 기존 시스템과 비교 실험한 결과를 보이고, 마지막으로 결론을 제시한다.

II. Sensor Network based Mobile Agent Middleware

1. Mobile agent and push operation

이동에이전트는 임의 공간에 분포되어 있는 센서노드들 사이를 이주하는 이동성 객체로서, 이주 시 주어진 작업을 처리하기 위한 능동규칙과 같은 실행로직과 작업처리결과

의 탑재로 인하여 크기가 증가하므로 네트워크의 부하를 유발하여 이주시간 지연 문제를 유발한다[7,8]. 이를 해결하기 위하여, 본 논문에서는 분산객체 개념을 사용하여 실행로직은 Sensor Data Acquisition Manager와 Active Rule Checker에 의해 Sensor Data Server에 연동된 규칙베이스나 Sensor Data Server로부터 푸시기능으로 전달 받고, 수집된 센싱 데이터나 작업처리결과는 즉시 서버(싱크노드)에 푸시기능으로 송신 후 이주하도록 하였으며, 호출 모듈만을 이동에이전트 객체로 구현하는 에이전트 경량화 방법을 적용하여, 이들의 상호 협력과정을 통하여 주어진 작업을 효과적으로 처리하는 방법을 적용한다. 또한, Agent Generator에 의해 생성되고 이주 리스트(이주 대상노드들에 대한 객체 참조자 리스트)를 통하여 이주하는 이동에이전트와 관련된 응용시스템에서는 이동에이전트의 이주 최적화 및 신뢰성 보장방법이 필수적으로 요구된다. 이주순서는 노드들의 위치나 주소를 통한 정해진 순서로 이주하거나, 임의의 노드에서 주어진 임계값을 적용한 규칙들의 실행결과에 따라 이주순서를 변경하기도 한다. 이러한 이주 과정에서 네트워크의 트래픽이나 노드의 장애로 인하여 이주시간 지연이나 무한 대기상태 등이 발생할 경우, 이를 능동적으로 해결하여 이동에이전트의 이주 신뢰성을 확보하여야만 한다[5,8]. 본 논문에서는, 이동에이전트를 기본적인 스레드 형태로 동작시키고, 필수적으로 필요한 정보 반환 기능 (getAgentProfile()), 에이전트 실행 기능 (run()), 이주 기능 (gotoNext()), 도착 시 수행 기능 (onArrive()), 이주 전 수행 기능 (onLeave()) 등의 인터페이스와, 이주의 신뢰성 확보를 위한 최적경로 탐색 메소드 (getOptimalPath())와 노드 장애 시 이주 대상노드 재지정 메소드 (getNodeAdjusted()) 등을 구현하여 적용한다.

이동에이전트 시스템 운영 시 서버에서 제공하는 푸시기능은 네트워크 트래픽 감소를 위하여 클라이언트에게 정보를 제공하는 기능으로서[12,14], 네트워크 트래픽 분석 기반의 데이터 전송 유연성을 통하여 안정성과 신뢰성을 보장하는 정보서비스를 제공한다. 이와 같은 푸시기능은 이동에이전트를 통하여 사용자 요구사항을 전달받고, 객체 참조자를 이용하여 규칙 실행로직을 위한 규칙베이스를 검색한다. 또한 Ping 프로그램 실행을 통하여 이주 대상노드들에 대한 패킷의 평균 왕복시간 (ART: Average Return Time)과 패킷 분실 수 (NLP: Number of Lost Packets)를 산출하여, 이 값들이 주어진 임계값의 범위를 벗어날 경우에는 클라이언트에게 보내는 처리결과의 전송을 임의의 시간 동안 지연시킨 후 다시 이주 대상노드들에 대한 Ping 실행 정보를 분석하여 결과를 전송하도록 한다.

이 때 이동에이전트가 처리결과를 보유한 채 다음 노드로 이주하는 것이 아니라, 푸시 기능을 통하여 결과를 사용자에게 전달하도록 함으로써 이동에이전트를 일정 크기로 유지하도록 한다.

2. Mobile agent middleware

공공건물에 사용되는 다양한 센서노드들은 이질적인 수많은 노드들로 구성되며 수요자 행위 및 환경 설정에 따른 다양한 기능들을 통제 및 운영이 요구되므로, 응용 레벨에서 이를 제어하는 시스템의 설계 및 구현이 필요하다 [7,10]. 또한 인접한 시공간 내에서 강한 의존성과 연관성을 갖는 센서 데이터들의 특성으로 인하여, 센서 네트워크와 응용시스템 사이에서의 독립성 유지 및 유연한 통합을 보장하는 미들웨어가 반드시 필요하다[9,13,14].

본 논문에서는 이동에이전트의 이주를 통하여 센서 데이터를 수집·저장·전송하고, 향후 학습과 능동규칙을 통하여 지능적 자율조치를 실행하는 지능형 이동에이전트 미들웨어 시스템 구축을 위하여 다음 Fig. 1과 같은 센서 데이터 서버와 이동에이전트 미들웨어의 연동 환경과 이동에이전트의 수행 패턴을 제안한다.

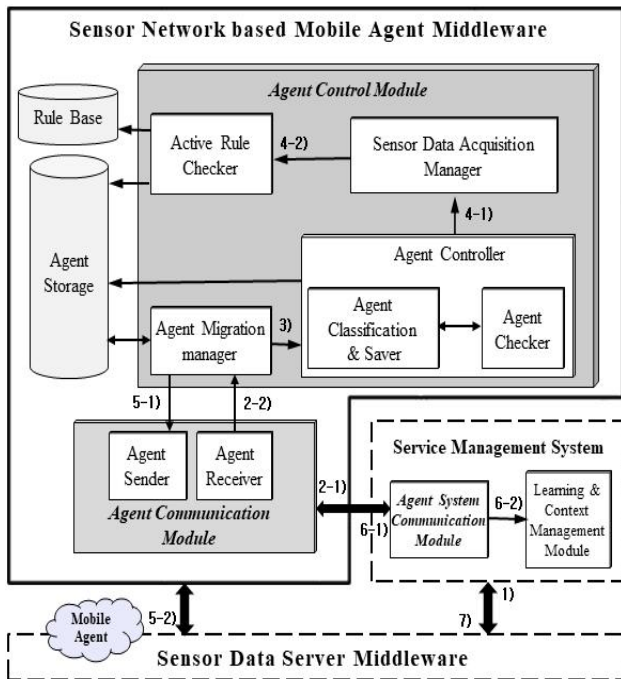


Fig. 1. The process of performing the functions of the mobile agent in sensor network based mobile agent middleware

센서 데이터 서버와 이동에이전트 미들웨어 연동 환경에서 이동에이전트의 이주, 규칙 실행 및 결과 전송을 위한 기능들의 수행 과정은 다음과 같다.

① Sensor Data Server Middleware를 통한 서비스 요청이 발생하면, ② Service Management System은 센서 데이터 및 메타데이터 요구 질의를 Mobile Agent Middleware의 Agent Communication Module을 통하여 Agent Migration Manager에 전송

③ Agent Control Module의 Agent Migration Manager는 저장된 에이전트의 분류에 따라 각 에이전트의 이주를 위한 경로 조정 및 선택 후 Agent Communication Module을 통해 이동에이전트를 이주 대기하거나 능동규칙 탑재 불필요 시에는 이주

④ 이주 대상 이동에이전트는 분류 및 저장 후 Sensor Data Acquisition Manager에 의해 능동규칙 적용 여부를 검사하여 필요 시 Sensor Data Server에 연동된 Rule Base로부터 능동규칙을 탑재

⑤ 규칙 탑재 후 지정된 경로에 따라 해당 센서노드로 이주하여 센싱 데이터 수집 및 탑재된 능동규칙 실행 후 복귀하거나 결과 전송 후 이주

⑥ Service Management System 내의 Agent System Communication Module은 Agent Communication Module로부터 수신한 센싱 데이터 및 메타데이터를 Learning & Context Management Module에 전송

⑦ 학습 등을 통한 결과를 Sensor Data Server Middleware를 통해 응용시스템에 전달

Fig. 1과 같은 센서 네트워크 기반의 이동에이전트 미들웨어 환경에서 이동에이전트는 TinyOS의 액티브 메시지를 통하여 이주한다. TinyOS는 추상화 과정을 통하여 IEEE 802.15.4의 프레임에 대응하는 액티브 메시지의 정의를 사용자에게 제공하므로, 본 논문에서는 라디오 스택과 상위 응용 간의 데이터 전달을 위하여 mobileAgent_t 구조체로 표현되는 액티브 메시지를 이용하여 에이전트를 이주하도록 한다.

본 논문에서 정의하여 사용하는 이동에이전트는 다음 Fig. 2와 같이 식별자, 이주 시작 및 복귀 노드 식별자(서버 또는 싱크노드), 이전 노드 및 목적 노드, 이주 리스트, (절대/상대) 임계값 및 센싱 데이터 등의 정보를 보유하며, 응용 시스템에 적합한 능동규칙 설계 내용에 따라 보유정보들의 변경 및 추가가 가능하다.

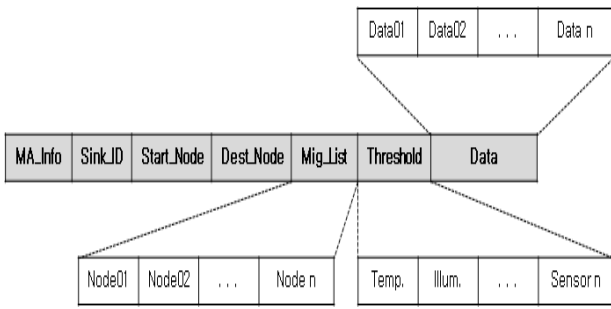


Fig. 2. Structure of mobile agent

Agent Migration Manager는 이주 리스트 (Mig_List)를 통하여 방문대상노드를 결정하고, 필요 시 경로조정 방법을 적용하여 Agent Communication Module의 Agent Sender를 통해 에이전트를 이주시킨다. 또한, 이주 대상 노드들을 식별하기 위하여 각각의 대상에 대한 검색과 확장이 가능한 주소체계와 모든 이주 대상 개체들의 정보를 보유하는 다음 Fig. 3과 같은 메타테이블을 사용한다.

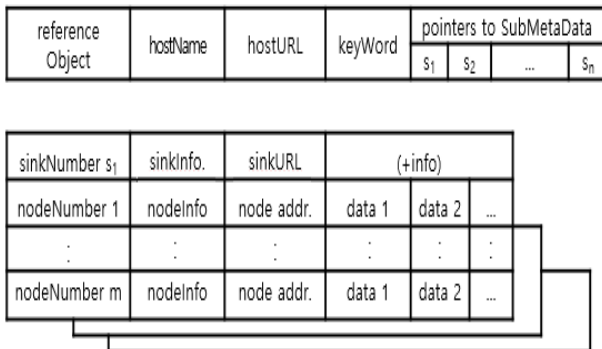


Fig. 3. Meta table for sensor network environment

Fig. 3의 윗부분은 센서 데이터 서버 정보를, 아랫부분은 해당 서버에 관련된 싱크노드와 센서노드 관련 정보를 표시하는 메타테이블이다. 메타테이블의 싱크노드 및 센서노드 주소를 이용하여 이동에이전트의 이주, 데이터 획득 및 능동규칙의 실행이 이루어진다. 각 노드들에 대한 네이밍 서비스를 위한 주소체계로, 센서 데이터 서버와 연동된 싱크 및 센서노드 계층으로 구성하고, 각 노드에 부여된 번호를 조합한 주소 (Data server IP: sink node #. sensor node #, 예: 202.31.147.40:1.2)를 사용할 수 있으나, 본 논문에서는 각 서버에 센서 모듈을 부착하여 이주경로 최적화 및 경로조정 알고리즘에 대한 실험만을 위하여 인터넷 주소망의 주소(예: 202.31.147.40)만을 사용한다.

다음 Fig. 4는 이러한 센서 네트워크 기반의 이동에이전트 미들웨어 환경에서 이동에이전트의 이주 과정을 보여주는 흐름도이다.

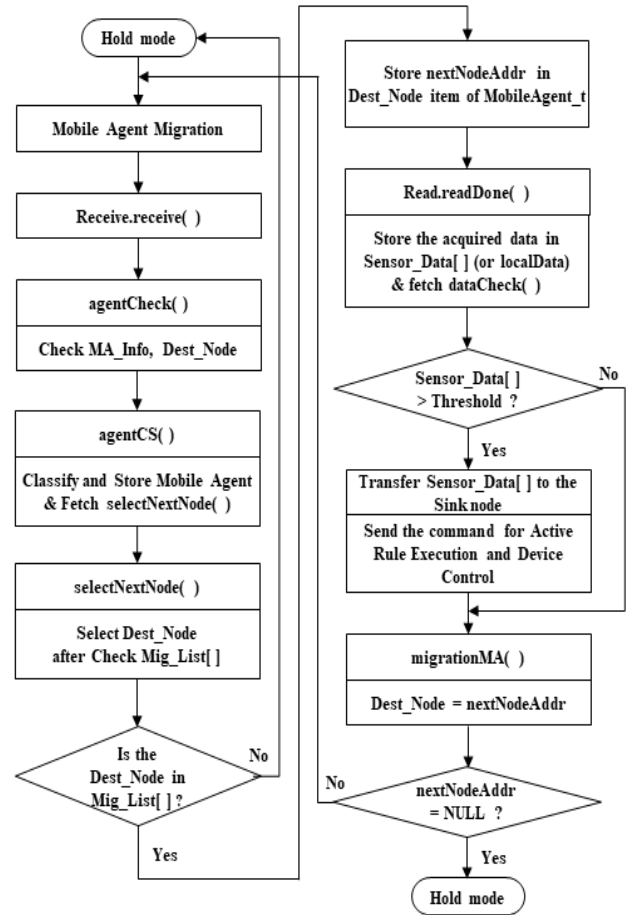


Fig. 4. Process of mobile agent migration

위와 같은 이주과정을 통하여 이동에이전트에 의한 실시간 데이터 획득, 비교, 분석 및 조치 실행을 가능하도록 함으로써 이동에이전트 미들웨어 시스템의 응용성과 유용성을 확보하도록 한다.

III. Migrating Optimization Algorithms of Mobile Agents

센서 네트워크 환경에서 이동에이전트는 센서노드들 간을 임의의 능동규칙 등을 탑재하고 주어진 순서대로 이주하면서 필요한 동작을 수행한다. 이러한 이동에이전트의 기능 수행 과정 중 적절한 이주가 불가능할 경우 이동에이전트의 최적 이주경로를 보장할 수 있는 방법이 필수적으로 요구되며, 이는 에이전트 이주에 대한 신뢰성 보장에 중요한 기능으로 작용한다.

1. Optimal migration path search

이동 에이전트의 요구 기능에 대한 처리시간은 능동규칙을 포함한 탑재 로직 실행시간과 노드들 간의 이주시간을 포함하므로, 트래픽을 분석하여 지연시간을 최소화하는 최적의 이주경로 탐색이 필수적으로 요구된다.

본 논문에서는 최적경로 탐색을 위하여, 먼저 8개의 이주 대상노드들로 구성된 이주 리스트 획득 후 각 노드의 IP들을 추출하고, JAVA의 Runtime class의 exec() 메소드와 InputStream class를 이용하여 실행 정보를 획득하는 Echo Request 메시지를 이용하여 각 노드들의 동작 상태와 패킷 송수신 시간을 검사한다. 검사 결과에 따라 패킷의 평균 왕복시간과 패킷 분실 수를 구하고, 이주 지연 해결과 이주 보장을 위하여 패킷 손실률이 적은 노드들 순으로 라우팅 테이블을 재구성하여 각 센서노드에서 이동 에이전트의 이주경로를 결정한다. 만일 패킷 분실 수가 같을 경우에는 평균 왕복시간이 최소인 노드를 선택하도록 한다.

Fig. 5는 제안한 최적경로 탐색방법을 통한 이주경로 결정과정의 예를 보여준다. 초기 이주 리스트는 사용자의 의도에 따라 응용시스템에서 요구되는 환경변수를 적용한 DOI를 기준으로 SN51-> SN54-> SN42-> SN46-> SN44-> SN55-> SN53-> SN45 순으로 설정하였으나, 제안한 최적경로 탐색방법을 적용하여 이주시킨 결과 SN51-> SN54-> SN46-> SN42-> SN44-> SN53-> SN55-> SN45 순으로 이주함을 나타낸다. 여기서 초기 이주 리스트는 응용에 따라 소규모 구역 내의 위치 및 환경적 상황 특성을 기준으로 설정이 가능하다.

2. Re-designating migration target nodes

임의의 노드에 이주하여 실행 중인 이동 에이전트가 노드 장애 및 결손으로 인하여 다음 이주 대상노드를 결정하지 못한 경우에는 이주 신뢰성 보장을 위하여 이주 대상노드를 재지정할 필요가 있다. 본 논문에서는 3.1절에서 제안한 최적경로 탐색방법을 적용한 이주 방법에 추가적으로 노드 복제 방법을 적용하여 이주 지연 문제를 해결하고 이주의 신뢰성을 확보하도록 한다. 노드 복제 방법은 이동 에이전트가 이주한 현재의 노드에서 라우팅 테이블에서 제공한 목적 노드로 이동 시 자신을 복제하여 현재 노드에 대기시켜 두고, *System.currentTimeMillis()* 함수를 이용하여 *pre_time*과 *cur_time*으로 대기시간을 측정하고, 다음 노드로 이주한 후 삭제 신호를 수신할 때 까지 대기하도록 한다. 만일 이주 대상인 노드에 장애 발생 시 복제한 이동 에이전트는 주어진 대기시간 경과 후 $(cur_time - pre_time) > 300,000ms$ 이주 대상노드를 장애 발생으로 판단하고, 라우팅 테이블의 우선순위에 따라 다음의 최적노드를 이주 대상노드로 선정하여 이주하도록 한다. 이주 대상노드로 이주하여 주어진 작업을 수행한 경우에는 이주 전 노드에 복제 에이전트 삭제 신호를 송신하여 이를 삭제하며, 이와 같은 방법을 각 노드에서 반복적으로 적용한다.

위와 같은 이주 대상노드의 재지정 방법은 센서노드를 포함한 네트워크 환경 장애에 대한 능동적 대응을 통하여 이동 에이전트의 이주 신뢰성을 보장한다. 다음 Fig. 6은 Fig. 5에서의 최적경로 탐색방법을 적용하여 이주하는 과정에서 SN 44 노드에서 장애 발생 시 이주 대상노드를 능동적으로 재지정하는 과정을 보여준다.

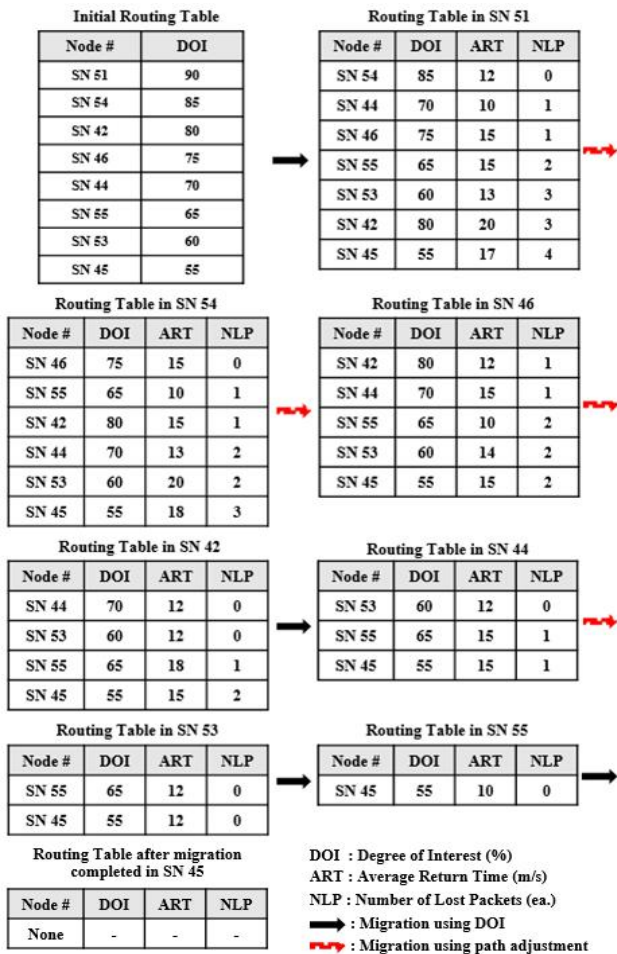


Fig. 5. Path adjustment process for optimal migration path search

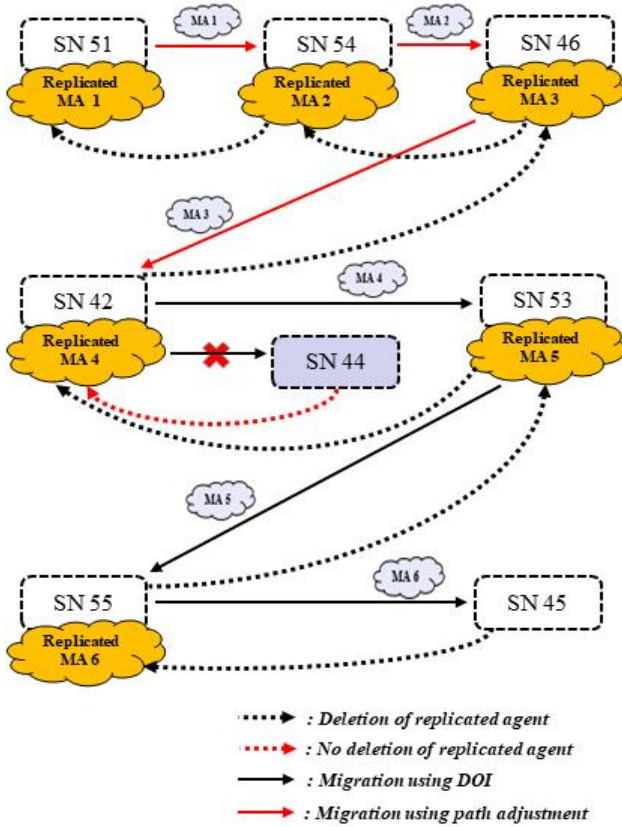


Fig. 6. Re-designating migration target nodes in the event of a node (SN 44) failure

IV. Experiments and results

본 장에서는 제안한 최적 이주경로 탐색방법을 기존 Aglets 시스템과 비교하여, 이동에이전트의 이주 방법에 따른 각 노드들로의 이주시간을 실험을 통하여 분석한다. Aglets은 미리 정의된 메소드를 보유한 프레임워크를 제공하므로, 에이전트 시스템 개발을 위하여 미리 정의된 메소드를 재 정의하여 원하는 기능을 구현할 수 있으며, HTTP의 상위 프로토콜로써 에이전트 이주 시 사용되는 응용계층 프로토콜인 Agent Transfer Protocol을 자체적으로 개발하여 사용한다[11].

실험 환경은 OMG의 MAF 명세와 OrbixWeb 3.2를 사용하고, JBuilder와 JDK를 사용하여 프로그램을 구현하였다. 또한, 이동에이전트의 이주시간 측정을 위한 실험은 100 Mbps 대역폭의 네트워크 시스템에서 노드 간 거리는 10m ~ 20m 사이로 설정하여 실행하였다. 구현에 사용된 시스템들은 Window 10 Pro OS 환경에서 Intel Core i5/i7 CPU와 12~18GB 메모리의 PC들을 사용하였고, 센서노드들의 주소는 각 PC의 IP 주소 (202.31.147.**를 사용하였다.

이동에이전트의 작업처리 시간은 이주시간과 해당 노드에서의 능동규칙을 통한 작업처리 시간의 합으로 결정되므로, 이주시간만을 비교 측정하기 위하여 이동에이전트 객체의 receive_agent() 메소드 호출 시점의 차이 값을 사용한다. 이동에이전트의 크기에 따른 이주시간 분석을 위하여, 동일한(고정적) 이주 계획에 따른 이주시간을 측정하여 제안 시스템과 비교 시스템 간의 성능을 평가한다.

1. Experiment on migration time depending on agent size

Aglets은 실행 모듈과 작업처리결과를 포함한 채로 이주하므로[11], 노드 간 이주 시 이들의 크기가 누적되어 증가하므로 이주시간이 지속적으로 증가하는 현상이 발생한다. 그러나 본 논문에서 제안한 이동에이전트의 이주 모델은 실행 모듈을 분리하고, 작업을 처리한 결과를 포함하지 않고 이주를 수행함으로써 Fig. 7에서 보이는 것처럼 실행 모듈의 크기에 따라 약간의 차이가 있지만 거의 일정한 이주시간이 보장됨을 알 수 있다.

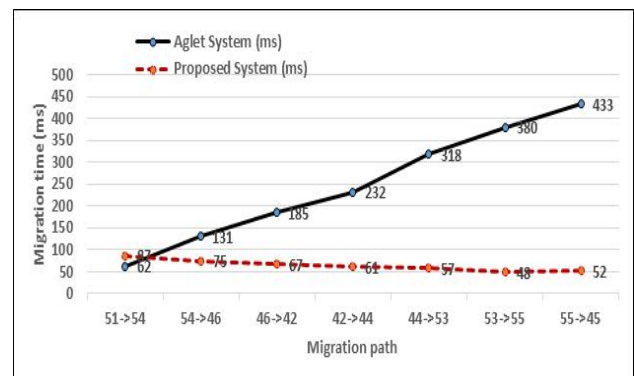


Fig. 7. Migration times depending on agent size

2. Experiment on migration time depending on network traffic

네트워크 트래픽에 따른 이주시간 비교 실험은 주어진 라우팅 테이블에서 제공하는 이주경로에 따라 이주하는 과정에서, 네트워크 트래픽을 탐지하고 ART와 NLP를 이용하는 노드 재지정 알고리즘을 적용한 후 이주시간의 변화를 측정하여 성능을 비교 평가한다. 실험은 8개의 노드 간에 이동에이전트를 이주시키면서 이주경로 상에 트래픽을 발생시켜 진행한다.

Aglets은 시스템의 특성 상 이주 시 네트워크와 이주 대상노드의 상태들을 고려하지 못함으로써 이주의 효율성과 신뢰성을 보장할 수 없으며 고정된 이주계획에 따라 이동에이전트가 이주하지만[11,12], 제안 시스템은 트래픽을

감지하면 최적경로 탐색 알고리즘을 실행 후 대상노드를 결정하여 이주하도록 한다.

실험 결과 Fig. 8과 같이 Aglets은 이주가 진행될수록 크기 또한 증가하므로 트래픽의 영향을 많이 받아 이주시간이 더욱 증가함을 알 수 있다. 제안 시스템은 트래픽을 탐색하고 최적경로 탐색을 위한 소요시간으로 인하여 이주 초기에는 이주시간이 지연되는 현상을 보이지만, 이주가 진행될수록 라우팅 테이블의 객체 참조자수가 감소됨에 따라 최적경로 탐색시간이 감소되어 전체적인 이주시간이 감소되는 장점을 보인다.

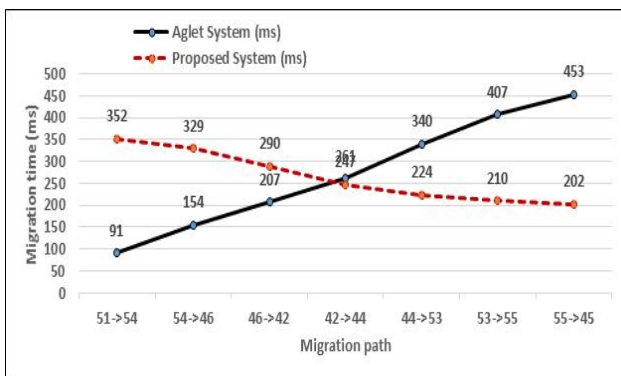


Fig. 8. Migration times depending on network traffic

위 실험들의 결과를 통하여, 제안된 최적 이주경로 탐색 알고리즘을 적용한 이동에이전트의 이주는 기존 이동에이전트 시스템과 비교하여 에이전트의 경량화에 따른 이주 시간 뿐만 아니라, 네트워크의 트래픽 발생 경우에도 동적인 최적 이주경로 제공으로 이주의 효율성과 신뢰성을 보장함을 알 수 있다. 다만, 최적 이주경로 탐색을 위한 시간이 소요됨은 본 실험 환경과 같은 소규모 환경 보다 이동에이전트가 원거리에 배치되어 있는 노드 간을 이주해야 하는 환경에서 효과적으로 적용될 수 있음을 나타낸다.

V. Conclusions

본 논문에서는 센서노드들 간을 이주하면서 주어진 작업을 처리하는 이동성 객체인 이동에이전트는 실행 로직과 작업처리결과 탐색으로 인하여 크기가 증가되어 네트워크의 부하를 유발하여 이주시간 지연 및 이주 신뢰성을 훼손한다. 본 논문에서는 분산객체 기술을 이용한 이동에이전트의 경량화 방법과 네트워크 트래픽 발생 시 능동적으로 이주 최적경로를 탐색 및 제공하는 이주 신뢰성 보장 알고리즘을 제안하였다. 이를 위하여, 이동에이전트의 이

주와 작업수행 환경을 지원하고, 향 후 학습과 능동규칙을 통하여 지능적 자율조치를 실행하는 지능형 이동에이전트 미들웨어 시스템 구축을 위하여 센서 데이터 서버와 이동에이전트 미들웨어의 연동 환경을 구축하였다. 또한 이주 대상노드들에 대한 패킷의 평균 왕복시간과 패킷 분실 수를 이용하는 최적경로 탐색 및 경로 재조정 알고리즘과 로드 복제 알고리즘을 적용하여 이주 지연 문제 해결 및 이주 신뢰성을 확보하도록 하였다. 또한, 센서 네트워크 기반 이동에이전트 미들웨어 환경에서 능동규칙을 탑재한 이동에이전트의 이주시간 분석 및 규칙 실행 실험을 통하여, 제안 방법들이 기존 시스템 비교하여 이동에이전트의 자율성과 이주의 효율성 및 신뢰성을 보장함을 확인하였다.

향 후 사용자의 의도 및 상황에 따른 학습과 능동규칙 등을 적용하여 지능적 자율조치를 실행하는 지능형 이동에이전트 미들웨어 시스템 구축과 이주 신뢰성 기반의 이동에이전트 응용 시스템에 대한 연구가 요구된다.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (018R1D1A1B07051045)

REFERENCES

- [1] P. Ardakani, "A Mobile Agent Routing Protocol for Data Aggregation in Wireless Sensor Networks," *International Journal of Wireless Information Networks*, 24(1), pp. 27-41, 2017 <https://doi.org/10.1007/s10776-016-0327-y>
- [2] H. Lee, "Reducing response time with dynamic load balancing and agent pool in multi-agent system," Thesis(M.A) Sungkyunkwan University, 2012
- [3] H. Ito, Y. Miyagi, "A Study on Distance-Based Control of Mobile Agents for Formation Avoiding Entire and Partial Reflection," *Proceedings of ICCAS 2019*, pp. 753-758, 2019 <https://doi.org/10.23919/iccas47443.2019.8971714>
- [4] G. Damianos, et al., "Mobile agent itinerary planning for WSN data fusion: considering multiple sinks and heterogeneous networks," *International Journal of Communication Systems*, 30(8), 2017 <https://doi.org/10.1002/dac.3184>

- [5] Y. Yang, et. al., "Detecting and resolving deadlocks in mobile agent systems," *Journal of Computer Languages*, 42, pp. 23-30, 2017 <https://doi.org/10.1016/j.jvlc.2017.08.002>
- [6] D. Kim, J. Lim, S. Kim, "Design of Mobile-based Security Agent for Contents Networking in Mixed Reality," *Journal of Convergence for Information Technology*, 9(3), pp.22-29, 2019
- [7] S. Feng, "WSN Deployment and Localization Using a Mobile Agent," *Wireless Personal Communications*, 97(4), pp. 4921-4931, 2017 <https://doi.org/10.1007/s11277-017-4747-5>
- [8] K. Lingaraj, et al., "OMMIP: An optimized multiple mobile agents itinerary planning for wireless sensor networks," *Journal of Information and Optimization Sciences*, 38(6). pp. 1067-1076, 2017 <https://doi.org/10.1080/02522667.2017.1374740>
- [9] S. Sasirekha, S. Swamynathan, "Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network," *Journal of Communications and Networks*, 19(4), pp. 392-401, 2017 <https://doi.org/10.1109/jcn.2017.000063>
- [10] K. Jin, et. al., "Wi-Fi RSSI-Based Indoor Location Detection System with IoT Device," *Journal of KICIS*, 42(12), pp. 2346-2349, 2017 <https://doi.org/10.7840/kics.2017.42.12.2346>
- [11] G. Cabri, et. al., "Strong agent mobility for aglets based on the IBM JikesRVM," *Proceedings of the 2006 ACM symposium on Applied computing*, pp. 90-95, 2006 <https://doi.org/10.1145/1141277.1141298>
- [12] H. Park, "Design of the Agent Migration Information System for Shortest Migration Order," *The KIPS Transactions : Part A*, 9A(4), pp. 555-562, 2002 <https://doi.org/10.3745/kipsta.2002.9a.4.555>
- [13] X. Zhang, "A Node Localization Algorithm based on Wireless Sensor Network," *International Journal of Performability Engineering*, 14(4), pp. 821-830, 2018 doi: 10.23940/ijpe.18.04.p24.821830
- [14] Y. Lee, M. Jang, "Location Trigger System for the Application of Context-Awareness based Location services," *Journal of the KSCI*, 24(10), pp. 149-157, 2019 <https://doi.org/10.9708/jksci.2019.24.10.149>

Authors



Yon-Sik Lee received the B.S. and M.S. degrees in Computer Science from Chonnam National University, Korea, in 1982 and 1984, respectively. And, his Ph.D. degree in Computer Application Engineering from

Chonbuk National University, Korea, in 1994. Dr. Lee joined the faculty of the School of Computer Information and Communication Engineering, Kunsan National University, Kunsan, Korea, in 1986. He is interested in sensor network middleware, active rule system, agent system and cloud computing.