

이동 에이전트 통신을 위한 효율적인 위치 추적 방법

송 상 훈[†]

요 약

이동 에이전트에 대한 위치 추적 기능은 네트워크에서 움직이는 개체에 메시지를 전달하기 위한 것이다. 대부분의 위치 추적 방법들은 이동 에이전트의 현재 위치 정보를 가지고 있는 릴레이 노드를 사용하여 메시지를 전달할 수 있게 한다. 본 논문은 도메인 기반의 프록시를 이용한 효율적인 위치 추적 방법을 제안한다. 이동 에이전트들이 이동 할 호스트를 무작위로 결정하지 않고 같은 도메인에 있는 호스트들을 먼저 방문하는 특징을 이용하여 위치 정보 등록 비용과 메시지 전달 비용을 최소화 하는 것이다. 도메인 기반 프록시는 위치 정보 등록 비용을 줄이는데 효과적이지만, 프록시 체인의 길이가 길어지면 메시지 전달 비용을 높일 수 있는데 프록시 체인의 길이를 압축 시켜 메시지를 효율적으로 전달하는 방법을 제안한다.

An Efficient Location Tracking Method for Mobile Agent Communication

Sang Hoon Song[†]

ABSTRACT

The provision of location tracking for mobile agents is designed to deliver a message to a moving object in a network. Most tracking methods exploit relay stations that hold location information to forward messages to a target mobile agent. In this paper, we propose an efficient location tracking method for mobile agents using the domain-based proxy as a relay station. Mobile agents can reduce the length of their migration paths by visiting hosts in the same domain first, rather than selecting hosts randomly. The proposed method exploits the domain-based moving patterns of mobile agents and minimizes registration and message delivery costs. Since the long proxy chain can contribute to the high message delivery cost, we also propose a compressing method to reduce the message delivery cost.

키워드 : 이동 에이전트(Mobile Agents), 이동 에이전트 통신(Mobile Agent Communication), 위치 추적(Location Tracking), 프록시(Proxy), 도메인(Domain)

1. 서 론

이동 에이전트는 사용자를 대신하여 네트워크에 연결된 호스트들을 자발적으로 방문하여 필요한 정보를 찾거나, 정보 전달, 또는 최소 비용의 예약 업무 등의 주어진 작업을 처리할 수 있는 소프트웨어를 뜻한다. 이동 에이전트 시스템은 분산 시스템에서 클라이언트/서버 방식의 여러 가지 문제점들을 해결할 수 있는 새로운 방식으로 온라인 쇼핑, 실시간 장치 제어, 분산처리 작업 등 여러 가지 분야에 응용할 수 있는 기술이다[1, 2].

이동 에이전트는 미리 정해진 경로 또는 이동 에이전트에 의해 현재 입수된 정보에 따라 동적으로 결정되는 경로를 따라 노드들을 방문하여 작업을 수행해 나간다. 직접 코

드가 이동되므로 여러 가지 장점이 있다. 예를 들어 원격 호스트에 처리할 데이터의 크기가 아주 큰 경우에, 원격 호스트에 이동 에이전트를 보내어 처리한 결과만 전송하면 네트워크 부하를 감소시킬 수 있다. 네트워크상에서 발생할 수 있는 불규칙적인 통신 지연은 원격 시스템을 실시간으로 처리하는 것을 어렵게 만든다. 하지만 이동 에이전트는 원격 시스템으로 이동하여 직접 실시간으로 제어를 할 수 있어 네트워크의 통신 지연 등에 의한 문제점을 해결할 수가 있다. 그리고 연결 상태가 자주 끊기는 이동 장치를 사용하는 응용 분야에도 적합하다.

이러한 여러 가지 장점에도 불구하고 이동 에이전트 시스템은 아직 여러 가지 해결해야 될 문제들이 남아있다. 보안 문제, 이동 에이전트의 이동성에 따른 통신 문제, 이동 에이전트의 위치 정보 관리 등에 대한 시스템 지원에 관한 연구들이 진행되어 오고 있다. 원격지의 대용량의 데이터를

[†] 종신회원 : 세종대학교 컴퓨터공학과 교수
논문접수 : 2004년 3월 20일, 심사완료 : 2004년 6월 11일

네트워크 통신을 통하여 처리하지 않고 이동 에이전트를 원격 호스트에 보내어 로컬 데이터로 처리할 수 있다는 것이 큰 특징 중 하나이다. 이런 특징 때문에 이동 에이전트 시스템에서 통신 문제는 그렇게 중요하지 않다고 주장할 수도 있을 수 있지만, 다음과 같은 여러 가지 상황에서 효율적인 통신 방법은 꼭 필요하다.

예를 들어 한 사용자에게 의해 이동 에이전트들이 네트워크에 투입될 때 매개 변수가 정해졌다고 하자. 어느 시점에 사용자가 네트워크에 퍼져서 실행 중인 이동 에이전트의 매개 변수를 변경해야 될 경우가 있다. 또는 투입된 이동 에이전트들 중의 하나가 이미 사용자에게 결과를 전달하면 더 이상 다른 이동 에이전트들이 원격 호스트들과 네트워크의 자원을 소모하지 않도록 모든 이동 에이전트들의 동작을 마치고 사라지게 해야 한다. 대부분의 경우에 많은 양의 메시지 전송이 필요한 것은 아니지만, 효율적인 메시지 전송이 이루어 져야 한다[3].

이동 에이전트 시스템의 기본 통신은 분산 시스템의 일반화된 통신 방법인 메시지 전달 방식이라고 가정한다. 메시지를 이동 에이전트에 전달하기 위해서는 이동 에이전트의 위치를 추적할 수 있는 기능이 있어야 한다. 즉, 이동 에이전트가 어느 한 호스트에서 다른 호스트로 이동할 때마다 갱신된 현재의 위치 정보가 어딘가에 저장되어 송신자의 메시지가 갱신된 정보에 따라 이동 에이전트에 전달될 수 있어야 한다. 이동 에이전트의 위치를 추적할 수 있는 정보를 저장하는 장소와 방법에 따라 위치 정보를 등록하는 비용과 메시지 전달 비용이 정해진다. 이동 에이전트 위치 정보 추적 시스템의 목표는 이동 에이전트에게 효율적으로 메시지 전달을 위해서 위치 정보를 등록하는 비용과 메시지 전달 비용을 최소화하는 것이다.

본 논문에서는 도메인 기반의 프록시라 불리는 릴레이 노드를 이용한 위치 추적 방법과, 프록시 체인의 길이를 압축하여 메시지 전송 비용을 최소화하는 방법을 제안한다. 여기서 도메인이란 구조적 그리고 물리적으로 노드간 거리가 가까워서 네트워크 지연이 작은 호스트들의 집합이라고 볼 수 있다. 이동 에이전트들은 이동 시에 무작위로 호스트를 결정하지 않고 같은 도메인에 있는 호스트들을 먼저 방문함으로써 전체 이동 경로를 줄일 수 있다. 제안한 방법은 이동 에이전트의 이런 특징을 이용하여 위치 정보 등록 비용도 최소화하고 메시지 전송 비용도 최소화할 수 있는 효율적인 방법이다. 본 논문에서는 이동 에이전트가 메시지 전달 속도보다 빠르게 계속 이동하게 되면 메시지는 계속 이동 에이전트를 따라가게 되어 메시지 전달이 보장되지 않는 경우는 다루지 않는다. 이 문제는 이동 에이전트의 메시지 전달 보장 문제로서 또 하나의 연구 분야이다. 본문의

내용에서 노드와 호스트는 같은 의미로 사용된다.

2장은 관련 연구 및 동기에 대하여 기술한다. 3장은 도메인 기반의 프록시 개념에 대하여 설명하고 4장에서 프록시 체인 압축 방법에 대하여 설명한다. 5장에서 시뮬레이션 및 성능 분석을 기술하고 6장에서 결론을 맺는다.

2. 관련 연구 및 동기

이동 에이전트 위치 추적에 관한 여러 가지 방법들이 최근에 발표되어 왔다[3-5]. 이 방법들의 공통된 개념은 현재의 위치 정보를 저장하고 있는 릴레이 노드(relay node)를 사용하는 것이다. 송신자는 이동 에이전트의 위치를 몰라도 이동 에이전트의 릴레이 노드에 메시지만 보내면 된다. 릴레이 노드는 이동 에이전트의 위치 정보를 가지고 있어서 송신자에게 투명한 메시지 전달 서비스를 한다. 투명한 서비스란 송신자는 이동 에이전트가 현재 이동 중이거나, 릴레이 노드가 이동 중에 있는 경우에도 관계없이 똑 같은 방법으로 메시지를 보낼 수 있도록 하는 것이다. 즉, 이동 에이전트나 릴레이 노드가 이런 상태 정보를 송신자에게 알려주어, 송신자가 이를 해결하도록 하는 것이 아니다. 이동 에이전트와 협력하여 작업을 하는 송신자들은 이동 에이전트의 홈을 알고있다고 가정하고 홈 서버도 릴레이 노드로 간주한다. 송신자가 릴레이 노드를 통해서 얻은 위치 정보를 이용하여 이동 에이전트에 메시지를 직접 전달하는 것은 고려하지 않는다. 릴레이 노드에 3가지 형태가 있는데, 홈 서버, 메일 박스, 그리고 포인터 체인이 있다.

홈 서버 방식은 이동 에이전트의 홈 서버가 이동 에이전트의 현재 위치 정보를 저장하고 있고 송신자로부터 보내진 메시지를 이동 에이전트의 현 위치로 전달하는 방법이다. 이동 에이전트가 새로운 호스트로 이동할 때마다 새로운 위치 정보를 홈 서버에 알려야 한다. 이 프로토콜은 간단하지만 위치 정보를 등록하는 비용이 높다. 특히 이동 에이전트가 홈 서버에서 멀리 떨어진 경우에 비용이 아주 커진다. 메시지 전달에 홈 서버 외에 다른 중간 릴레이를 이용하지 않으므로 메시지 전달 비용은 가장 작다. 송신자가 홈 서버에서 멀리 떨어진 경우는 삼각형의 라우팅 경로에 의해 전달 비용이 커지는 것은 이동 에이전트의 홈을 송신자의 홈과 가까운 곳으로 옮겨 메시지 전달 비용을 최소화할 수 있다[6].

포인터 체인 방식은 다수의 릴레이 노드를 사용하는 방식으로 이동 에이전트가 방문하는 경로의 호스트들은 경로상의 다음 호스트 위치를 가리키는 포인터 정보를 가지고 있다. 이동 에이전트가 새로운 호스트로 이동할 때마다 그 호스트를 연결해줄 수 있는 포인터를 만들어 놓는다. 홈 서

버는 포인터 체인의 첫 번째 노드가 된다. 홈 서버에서 멀리 떨어져 있는 한 도메인에 속하는 호스트들 간에 이동이 있을 때 홈 서버 방식에 비하여 위치 정보 등록 비용은 아주 작다. 그러나 이동한 경로들을 따라서 메시지가 전달이 되기 때문에 이동이 많이 된 후에 메시지 전달 비용은 매우 높아지는 단점이 있다.

메일 박스 방식은 이동 에이전트 마다 메시지 전달을 중계하는 메일 박스를 이용한다. 메일 박스는 이동 에이전트와 분리되어 있어 이동 에이전트와 다른 장소에 머물 수 있고 이동 에이전트가 이동한 경로를 따라 이동이 가능하다. 그리고 메일 박스의 이동은 수동적이다. 아직 이동하는 방법 및 이동할 노드에 관한 연구는 발표되지 않고 있다. 릴레이 노드 역할을 하는 메일 박스는 이동 에이전트의 위치 정보를 저장하고 있고 현 메일 박스의 위치는 항상 홈 서버에 갱신되어야 한다. 메시지 송신은 홈 서버를 통하여 메일 박스, 그리고 이동 에이전트로 전달이 된다. 홈 서버 방식과 같이 이동 에이전트가 이동할 때마다 메일 박스에 위치 정보를 등록하는 것은 똑같다. 메시지 전송 시에 송신자는 항상 홈 서버를 통해서 메일 박스의 위치를 알아야 하기 때문에 홈 서버를 사용하여 전달하는 것과 비용이 비슷하다고 볼 수 있다.

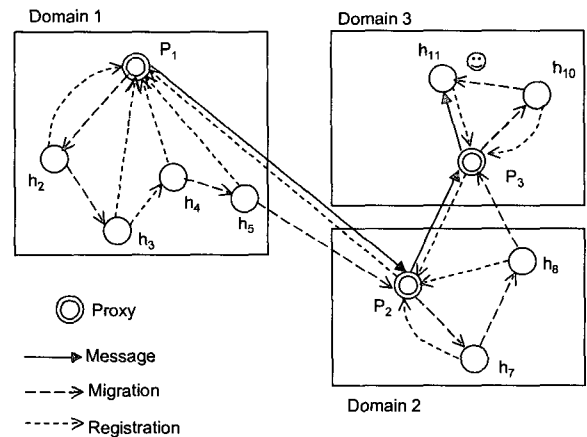
3. 도메인 기반의 프록시

위치 정보 등록의 비용을 줄이기 위해서 현재 위치 정보를 등록하여 저장하는 릴레이 노드의 위치가 현 호스트와 가까워야 되므로, 물리적 또는 구조적으로 가까운 호스트들을 도메인으로 나누어 도메인마다 하나의 프록시에 의해서 서비스하게 한다. 한 이동 에이전트 플랫폼이 속할 도메인은 플랫폼 설치 시에 네트워크의 구조를 보고 설치자에 의하여 설정되고, 각 도메인마다 유일한 도메인 번호가 지정되어 있다고 가정한다. 같은 도메인 내의 호스트들은 구조적으로 거리가 가깝기 때문에 하나의 프록시를 정하여서 릴레이 서비스를 하게 한다. 즉, 도메인 내의 이동은 도메인 내의 정해진 프록시에 위치 정보를 등록한다.

한 도메인의 프록시는 동적으로 정해지는데, 이동 에이전트가 이동 시에 새로운 도메인으로 들어갈 때 방문하는 첫 호스트가 프록시 역할을 하게 한다. 같은 도메인 내에서 각 이동 에이전트 마다 서로 다른 프록시의 서비스를 받게 된다. 같은 도메인 내의 호스트로 이동하는 경우는 위치 정보 등록 비용이 아주 작다. 메시지 전달은 도메인들을 연결하는 프록시 체인을 통하여 이동 에이전트가 있는 도메인의 프록시로 전달되고, 그 프록시에서 이동 에이전트로 전달된다. 동적으로 프록시를 정하고, 프록시 체인을 구성하려면, 에

이전트가 이동 시에 현 도메인 번호와 현 프록시 정보를 가지고 이동하면 된다.

(그림 1)은 도메인 기반의 프록시 체인 방법에서 이동 에이전트가 이동할 때의 등록과 메시지 전달 경로를 보여주고 있다. 이동 경로는 굵은 점선으로 나타내고, 위치 정보 등록은 가는 점선으로 나타냈다. 메시지 전달 경로는 프록시 P1, P2, P3를 거쳐서 이동 에이전트에 전달되는 실선으로 나타냈다. 주목할 점은 도메인 내에서의 위치 정보 등록 비용이 작고 메시지 경로가 포인터 체인 방식에 비하여 짧아진 것이다. 도메인 1과 도메인 2의 거리가 먼 경우에 홈 서버 방식의 경우는 도메인 2에 속하는 h7, h8에서의 위치 정보 등록 비용은 높다. 도메인 2의 P2에 도착했을 때의 위치 정보 등록 비용은 홈 서버 방식의 비용과 같다. 하지만 h7, h8으로 이동 시에 위치 정보 등록 비용은 프록시와 같은 도메인에 있는 호스트와의 거리가 가깝기 때문에 위치 정보 등록 비용은 작아지게 된다. 그리고 포인터 체인 방식에 비하여 메시지 전달되는 경로의 길이가 짧기 때문에 메시지 전달 비용이 낮지만, 홈 서버 방식보다는 여전히 크다.

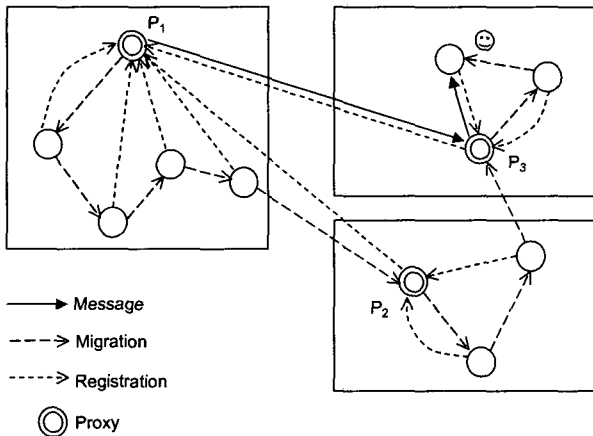


(그림 1) 도메인 기반의 프록시 체인

4. 프록시 체인 압축

메시지 전달은 프록시 체인을 통하여 이동 에이전트가 있는 도메인의 프록시에 전달되고, 그 프록시에서 이동 에이전트가 방문하고 있는 호스트로 전달된다. 메시지 전달 횟수가 많지 않을 때는 도메인 기반 프록시 방법이 효율적이지만, 메시지 전달 횟수가 많아지면서 프록시 체인의 길이도 메시지 전달 지연에 영향을 미치기때문에 프록시 체인의 길이를 압축하는 방법을 사용하여 메시지 전달 비용을 줄여야 한다. 간단한 방법으로 도메인을 옮길 때마다 프록시 체인에 연결하지 않고 홈 서버 방식처럼 홈 서버에 도메인의 프록시를 연결한다. (그림 2)는 프록시 P3에서 등록을 P2에

하지 않고 P_1 에 등록을 한 것을 보여주고 있다. 따라서 메시지가 많은 경우에도 메시지 전달 비용은 프록시 체인의 길이가 1이므로 최적의 비용으로 전달할 수가 있을 것이다. 도메인이 바뀔 때마다 새로운 도메인의 프록시를 홈에 연결하여 릴레이 노드 역할을 하게 함으로써 전체 비용을 최소화할 수 있다. 하지만 이동하는 도메인이 계속 홈에서 멀어질 경우는 가까운 도메인의 프록시에 연결하는 경우에 위치 정보 등록 비용을 줄일 수 있다.



(그림 2) 간단한 프록시 체인 압축

최소화하려는 전체 비용은 메시지 전달 회수와 위치 정보를 등록하는 프록시 간의 거리에 의해 결정이 된다. 메시지 수를 예측할 수 있고 프록시 간의 거리를 알 수 있으면 최적의 프록시 체인으로 압축 시킬 수 있을 것이다. 네트워크 노드들 간의 거리 정보가 필요한 여러 가지 응용 분야에 인터넷의 기하학적인 모델링을 통하여 계산된 절대 좌표계를 이용한 GNP(Global Network Positioning) 방법이 제안되었다[7, 8]. 예상되는 메시지 수는 응용 프로그램마다 다르지만 이동 에이전트를 반복 실행한 후에 만들어지는 통계자료를 이용하여 예측할 수 있다. 예를 들어 이동 에이전트가 정해진 매개 변수 없이 호스트들을 방문하여 작업하고, 중간에 사용자가 작업을 중지하라는 것 외에 다른 메시지가 없는 경우에는 거의 0에 가까운 값으로 예측할 수 있다. 또는 작업 시간이 너무 길어지는 경우에 사용자가 중간에 중지 시키는 메시지를 보낼 확률이 높아지면 1에 가까운 값을 예측할 수 있을 것이다.

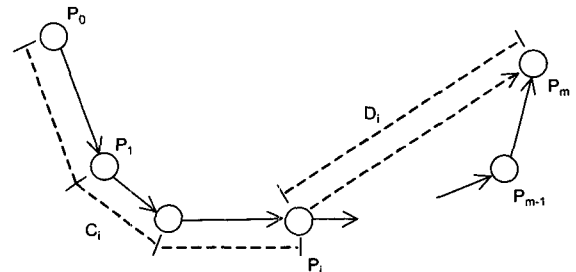
프록시 체인 압축 방법을 설명하기 위해 다음 변수들을 소개한다.

- N : 예측되는 메시지 수
- D_i : i 번째 프록시에서 현 프록시까지 거리
- C_i : 홈에서 i 번째 프록시까지 프록시 체인을 통한 거리
- R_i : i 번째 프록시에 등록하는 비용

i 번째 프록시에 연결을 함으로써 예상되는 전체 비용은 다음과 같다.

$$N \times (D_i + C_i) + R_i$$

(그림 3)에서 마지막 프록시 노드가 새로 연결되는 프록시인데 i 번째 노드에 연결하여 체인의 길이를 압축할 경우를 보여주고 있다. 에이전트는 P_0, P_1, \dots, P_{m-1} 의 프록시 번호와 홈에서 각 프록시까지의 프록시 체인 거리인 C_0, C_1, \dots, C_{m-1} 을 가지고 이동해야 한다. 각 노드에서 P_i 까지의 거리인 D_i 는 알고 있고, R_i 값은 D_i 에 비례하기 때문에 각 프록시 P_i 에서의 위 전체 비용을 계산하여 최소화가 되는 프록시 P_i 를 찾으면 된다.

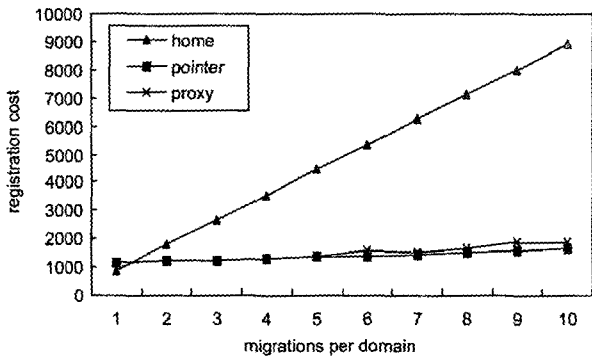


(그림 3) 프록시 체인 압축

일반적으로 프록시 체인의 길이는 포인터 체인에 비하여 아주 짧기 때문에 최소 비용을 갖는 P_i 를 찾는데 걸리는 계산은 무시할 만큼 작다.

5. 시뮬레이션 결과 및 고찰

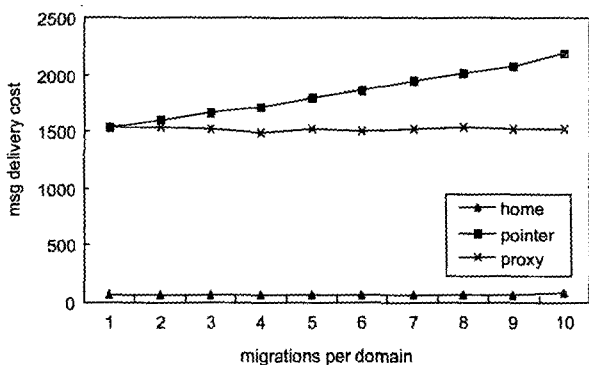
네트워크 연결은 2차원 그리드 형태로 이루어져 있고, 각 호스트의 위치는 2차원 평면 그리드의 (x, y) 좌표로 나타낸다고 가정한다. 두 호스트 간의 거리는 두 좌표 사이의 기하학적 거리로 나타낸다. 위치 정보 등록과 메시지 전달 비용은 두 노드간의 거리와 데이터 크기에 의해 정해지는데, 위치 정보 등록을 위한 데이터는 메시지 크기의 1/4로 가정하였다[9]. 시뮬레이션에서 이동에이전트가 네트워크의 호스트들을 이동하면서 필요한 위치 정보 등록 비용과 메시지 전달 비용을 계산한다. 한 도메인의 홈 노드에서 시작하여, 한 도메인 내에서 매개변수로 정해진 수 만큼의 호스트들을 방문한 후에 또 다른 도메인으로 옮겨가면서 이동을 계속한다. 에이전트가 이동하는 도메인들의 위치와 도메인 내의 호스트들의 위치들은 균등 분포에 의해 임의로 선택되고 도메인 내의 호스트 수와 메시지 수를 변화하면서 비용들을 계산한다. 네트워크 과부하에 의한 추가되는 통신 지연은 없다고 가정한다.



(그림 4) 위치 정보 등록 비용

(그림 4)는 도메인 수가 20일 때, 도메인 내에서 방문한 호스트 수에 따른 위치 정보 등록 비용을 나타낸다. 총 이동한 호스트 수는 방문한 도메인 수에 도메인 내에서 이동한 호스트 수를 곱하면 된다. 도메인과 호스트들이 임의로 선택된 경우에 도메인 당 방문한 호스트가 하나이면 위치 정보 변경 비용은 세 가지 방식 모두가 비슷하게 된다. 이것은 무작위로 선택한 경우에 홈 노드에서 모든 도메인과의 거리와 임의의 두 도메인 간의 거리는 평균적으로 비슷하게 되기 때문이다.

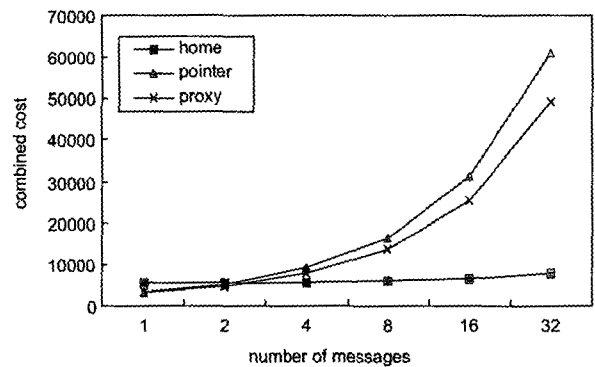
홈 서버 방식은 위치 정보 등록 비용이 포인터 체인 방식과 도메인 기반의 프록시 방법에 비하여 도메인 내에서 방문한 호스트가 많아지면서 훨씬 커지는 것을 보여준다. 홈 서버에서 멀리 떨어진 도메인의 호스트들에서 위치 정보 등록 비용이 크기 때문이다. 하지만 도메인 내의 프록시를 통한 위치 정보를 등록하는 도메인 기반 프록시 방법은 위치 정보 등록 비용이 포인터 체인 방식 보다 약간 크지만 큰 차이가 없음을 알 수 있다. 도메인 기반 프록시의 장점을 보여주는 것이다.



(그림 5) 메시지 전달 비용

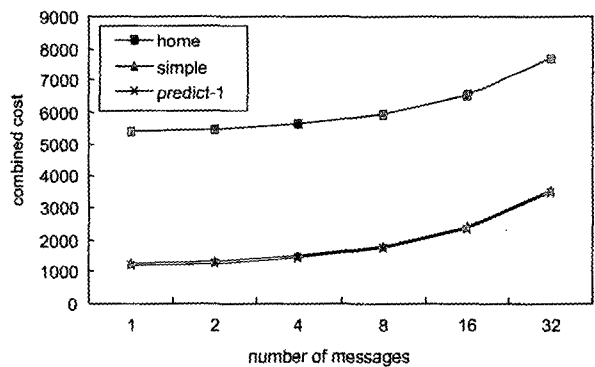
(그림 5)는 홈 서버에서 최종 호스트까지의 한 개의 메시지 전달 비용을 나타낸다. 홈 서버 방식은 현재 이동 에이전트의 위치에 다른 릴레이 노드 없이 직접 전달하기 때문

에 비용이 가장 작다. 도메인 기반 프록시 방법은 도메인 당 방문한 호스트가 한 개이면 포인터 체인 방식과 같게 되는 것을 확인할 수 있다. 그러나 도메인 당 방문한 호스트 수가 증가함에 따라 포인터 체인 방식은 체인의 길이가 늘어나 선형 적으로 증가하게 된다. 도메인 기반의 프록시는 프록시 체인의 길이가 포인터 방식과 달리 도메인 내의 방문한 호스트 수와 관계가 없기 때문에 증가하지 않는 것을 알 수 있다.



(그림 6) 메시지 전달 수에 따른 성능 비교

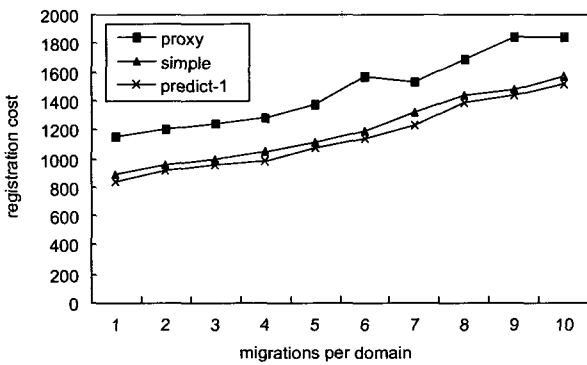
(그림 6)은 도메인 당 방문하는 호스트 수를 6으로 고정시키고 메시지 수의 증가에 따른 총 비용을 보여준다. 메시지 수가 증가함에 따라 프록시 방법과 포인터 체인 방식에서 메시지 전달 비용은 등록 비용을 압도하게 되어 전체 비용을 좌우한다. 포인터 체인의 길이가 프록시 체인 보다 길기 때문에 비용 증가율은 더욱 심하다. 홈 서버 방식은 여전히 등록 비용이 메시지 전달 비용을 압도하기 때문에 비용 증가율을 아주 낮다. 메시지 수가 작을 때는 프록시 방법이 효율적이지만 메시지 수가 증가함에 따라 전체 비용이 증가하므로 메시지 수가 많아지면, 4장에서 설명한 방법으로 프록시 체인을 압축해서 메시지 전달 비용을 줄여야 한다.



(그림 7) 홈 서버 방식과 압축 방법의 메시지 수에 대한 성능 비교

(그림 7)은 홈 서버 방식과 두 가지 압축 방법을 비교한

것이다. simple 압축 방법은 메시지 수를 예측하지 않고 도메인을 옮길 때마다 항상 홈 노드에 프록시를 연결함으로써 프록시 체인의 길이를 1로 유지하는 방법이다. Predict-1 압축 방법은 메시지 수를 항상 1로 예측하여 압축할 수 있는 최적의 프록시 연결 지점을 찾아 연결하는 방법이다. 이동하는 도메인들이 균등 분포에 의하여 임의로 선택이 되기 때문에 도메인을 옮길 때 앞 도메인과 새 도메인 간의 거리와 홈 노드와 새 도메인 간의 거리는 어느 쪽이 크다고 예측할 수 없다. 메시지 수에 따른 전체 비용은 두 압축 방법에서 거의 차이가 없게 나타나고, 홈 서버 방식과 마찬가지로 메시지 증가에 따른 성능 저하가 없어 압축 방법이 효율적임을 확인할 수 있다. 홈 서버 방식과의 차이는 거의 등록 비용의 차이에서 나타나는 것이다. 즉, 프록시 압축 방법은 홈 서버 방식의 단점인 등록 비용을 줄이고, 프록시 체인의 길이를 짧게 만들어 전체 비용을 최소화 하는 것을 보여주고 있다.

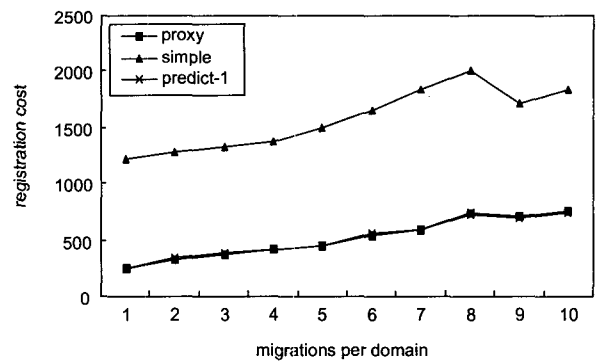


(그림 8) 균등 분포에 의한 임의의 도메인 선택 시 등록 비용 비교

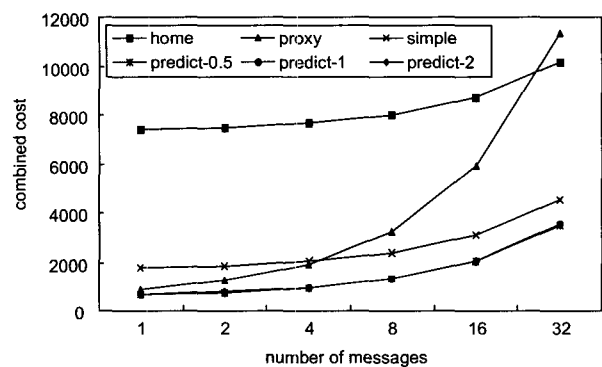
(그림 8)은 균등 분포에 의한 도메인들을 임의로 선택한 경우에 프록시, simple 압축, 그리고 predict-1 압축 방식의 등록 비용을 비교한 것이다. 균등 분포의 경우에 무작위로 선택된 도메인과 노드들에 의하여 홈 노드와 이동 경로 상의 프록시들과의 평균 거리는 59.8이고, 프록시들 간의 평균 거리가 76.6으로 나타나서 평균 거리가 약 16.8 정도의 차이가 나타났다. 즉, 이렇게 무작위로 선택된 노드들의 위치 때문에 압축없는 프록시 방법이 simple 방법보다 등록 비용이 커지고, predict-1 압축 방법은 simple 압축 방법과 비슷한 결과를 보여주고 있다.

그러나 이동 경로가 균등 분포에 의하여 임의로 선택이 되지 않고, 여러 도메인들이 하나의 클러스터를 이루면서 이러한 클러스터들이 서로 멀리 떨어져 있을 때, simple 압축 방법은 등록 비용이 커져서 비효율적이게 된다. 따라서 프록시 체인의 길이와 메시지 수를 고려한 압축 방법을 이용해

야 효율적으로 등록 비용을 줄일 수 있다. 실제 인터넷은 국가별로 클러스터, 지역 별 클러스터를 이루고 있어 이러한 구조에 더 가깝다고 볼 수 있다. 즉, 도메인을 옮길 때 simple 압축 방법보다는 메시지 수를 예측하고 프록시 간의 거리 정보를 이용하여 최적의 프록시 체인으로 압축하여 전체 비용을 줄일 수 있다. (그림 9)는 클러스터된 도메인 그룹들이 있는 경우의 등록 비용을 보여 주고 있다. 클러스터된 경우에 홈의 위치에서 이동 경로 상의 프록시들과의 평균 거리는 80.3이고, 프록시들 간의 평균 거리가 17.4로 선택이 되었다. 즉, 홈에서 거리가 멀리 떨어진 곳에 도메인들이 클러스터된 경우에 홈과의 거리가 멀기 때문에, 도메인을 바꿀 때마다 홈에 연결을 하는 simple 압축 방법은 비효율적이다. 예상한대로, predict-1 압축 방법과 압축하지 않는 프록시 방법의 경우가 simple 압축 방법에 비하여 등록비용이 작아지는 것을 알 수 있다.



(그림 9) 클러스터된 도메인 선택 시 등록 비용 비교



(그림 10) 클러스터된 경우의 프록시 압축 방법 성능 비교

(그림 10)은 도메인들이 클러스터된 경우에 홈 서버, 프록시, 그리고 프록시 압축 방법들의 성능을 보여주고 있다. Predict-n에서 n은 예측한 메시지 수를 나타낸다. 압축하지 않은 프록시 방법은 등록 비용은 작지만 프록시 체인의 길이에 의하여 전체 성능을 저하시키게 되는 것을 알 수 있다. 그리고 홈 서버 방식은 높은 등록 비용이 전체 비용을 압

도하고 있다. 압축 방법에서 n 값이 큰 경우는 메시지 전달 경로인 프록시 체인의 길이를 짧게해야 하고, n 이 0에 가까우면 등록 비용이 작아지는 프록시에 연결을 해야한다. (그림 12)에서 나타난 것처럼 n 이 0.5 그리고 1 이상인 경우에서 실제 메시지 수에 관계 없이 성능들이 거의 비슷하게 나타나서 그림에서 차이를 알아볼 수 없을 정도이다. 동적으로 메시지 수를 예측하는 것이 어렵기 때문에 n 은 실행할 때에 프로그램 성격에 따라 매개 변수로 정할 수 있는데, 메시지가 전혀 예상되지 않는 경우를 제외하면 1 정도의 값을 이용해서 프록시 체인을 효과적으로 압축할 수 있음을 알 수 있다. 즉 메시지 수를 전혀 예측할 수 없는 경우에는 predict-1 압축 방법을 사용하여 등록비용과 메시지 전달에 관계되는 프록시 체인의 길이를 비슷한 비중으로 반영하게 되어 효율적으로 전체 비용을 줄일 수 있다.

6. 결 론

본 논문은 이동 에이전트 시스템에서 도메인 기반의 프록시를 이용한 효율적인 위치 정보 추적 방법을 제안하였다. 이동 에이전트들이 이동 시에 무작위로 호스트를 결정하지 않고 같은 도메인에 있는 호스트들을 먼저 방문하는 이동 패턴을 이용하여 위치 정보 등록 비용과 메시지 전송 비용을 최소화 하는 것이다. 도메인 내의 프록시는 이동 에이전트가 처음 도메인으로 이동할 때에 동적으로 결정된다. 도메인 내의 이동은 해당 프록시에 위치 정보 등록을 함으로써 위치 정보 등록 비용이 작아지고 프록시 체인의 길이를 줄임으로써 메시지 전송 비용을 줄인다. 홈 서버, 포인터 체인, 그리고 제안한 도메인 기반의 프록시와 프록시 압축 방법을 시뮬레이션을 통하여 성능을 분석하였다. 시뮬레이션은 도메인 당 방문하는 호스트들의 수, 도메인 수, 그리고 메시지 수를 변경하면서 위치 정보 변경 비용 및 메시지 전달 비용을 계산한다. 도메인 당 방문하는 호스트가 많아질수록 도메인 기반 프록시와 홈 서버 방식의 위치 정보 변경 비용은 더욱 차이가 커진다. 그리고 방문한 도메인이 많아질수록 프록시 체인의 길이도 성능에 영향을 미치므로 도메인을 이동 할 때마다 프록시 체인을 압축함으로써 더욱 성능을 향상 시킬 수 있음을 시뮬레이션을 통하여 검증하였다.

참 고 문 헌

- [1] D. B. Lange and M. Oshima, "Seven Good Reasons for Mobile Agents," *Communication of the ACM*, Vol.42, No.3, pp.88-89, March, 1999.
- [2] D. B. Lange and M. Oshima, *Programming and deploying Java mobile agents with Aglets*, Addison-Wesley, 1998.
- [3] A. Murphy and G. P. Picco, "Reliable Communication for Highly Mobile Agents," *Agent Systems and Architectures/Mobile Agents (ASA/MA)'99*, pp.141-150, October, 1999.
- [4] J. Cao, X. Feng and S. K. Das, "Mailbox-Based Scheme for Mobile Agent Communications," *IEEE Computer*, pp. 54-60, September, 2002.
- [5] J. Baumann and K. Rothermel, "The Shadow Approach : An Orphan Detection Protocol for Mobile Agents," Springer-Verlag, Berlin, Germany, Vol.1477, pp.2-13, 1998.
- [6] G. Kunito, Y. Okumura, K. Aizawa and M. Hatori, "Tracking Agent : A New Way of Communication in a Multi-Agent Environment," *Proc. of IEEE 6 th Int'l Conf. on Universal Personal Comm.*, Vol.2, pp.903-907, 1997.
- [7] T. S. Eugene Ng and Hui Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches," in *Proceedings of IEEE INFOCOM'02*, New York, NY, June, 2002.
- [8] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. F. Gryniwicz and Y. Jin, "An architecture for a global Internet host distance estimation service," in *Proceedings of IEEE INFOCOM '99*, New York, NY, Mar., 1999.
- [9] Jiannong Cao, Xinyu Feng, Jian Lü, Henry Chan and Sajal K. Das, "Reliable Message Delivery for Mobile Agents : Push or Pull," *ICPADS 2002*, Taiwan, ROC., pp.314-320, December, 2002.
- [10] Objectspace, "Objectspace Voyager Core Technology," <http://www.objectspace.com>
- [11] A. Pham and A. Karmouch, "Mobile Software Agents : An Overview," *IEEE Communications magazin*, Vol.36, No.7, pp.26-37, July, 1998.
- [12] T. K. Shih, "Agent Communication Network - A Mobile Agent Computation Model for Internet Applications," *Proc. 1999 IEEE Int'l Symp on Computers and Communications*, pp.425-431, 1999.
- [13] V. A. Pham and A. Karmouch, "Mobile Software Agents : An Overview," *IEEE Communications Magazine*, pp.26-37, July, 1998.
- [14] N. Minar, K. H. Kramer and P. maes, "Cooperative Mobile Agents for Dynamic network Routing," in *Software Agents for Future Communication Systems*, Springer-Verlag, ISBN 3-540-65578-6, 1999.
- [15] A. Bar-Noy, I. Kessler and M. Sidi, "Mobile Users : To Update or Not To Update?," *ACM/Baltzer J. Wireless Network*, Vol.1, No.2, pp.175-195, July, 1995.

[16] A. Fuggetta, G. P. Picco and G. Vigna, "Understanding Code Mobility," IEEE Transactions on Software Engineering, Vol.24, No.5, May, 1998.

[17] P. Bellavista, A. Corradi, C. Stefanelli, "Mobile Agent Middleware for Mobile Computing," IEEE Computer, Vol. 34, No.3, pp.73-81, March, 2001.

[18] Prakash, R., Raynal, M., Singhal, "An Efficient Causal Ordering Algorithm for Mobile Computing Environments," Proceedings of the 16th International Conference on Distributed Computing Systems, Hong Kong, IEEE Computer Society, May, 1996.

[19] Kenjiro Cho and Ken Birman, "A Group Communication Approach for Mobile Computing," Proc. of the Workshop on Mobile Computing Systems and Applications. Santa Cruz, California. Dec., 1994.

[20] P. Bellavista, et al., "A Secure and Open Mobile Agent Programming Environment," Proc. ISADS '99, IEEE Press, Tokyo, Japan, 1999.



송 상 훈

e-mail : song@sejong.ac.kr

1977년 연세대학교 전자공학과(학사)

1979년 KAIST 전산학과(석사)

1992년 University of Minnesota 전산학과
(박사)

1992년~1996년 세종대학교 컴퓨터공학과
조교수

1997년~현재 세종대학교 컴퓨터공학과 부교수

관심분야 : 분산/병렬처리, 컴퓨터 보안, 컴퓨터구조, computer arithmetic