

최적 경로 탐색을 이용한 이동 에이전트의

적응적 이주

이연식, 김광종, 최영춘
군산대학교

yslee@kunsan.ac.kr, kkjkim@kunsan.ac.kr, Choi@kunsan.ac.kr

Adaptive Migration of Mobile Agent Using Optimal Path Searching

Yon-sik Lee, Kwangjong Kim, Youngchun Choi
Kunsan University

요약

본 논문에서는 이동 에이전트의 이주 노드들에 대한 최적 경로 탐색을 통한 적응적 이주 경로 기법을 설계한다. 적응적 이주 경로 기법은 이동 에이전트가 사용자로부터 동일한 작업 요청을 부여 받았을 경우 일반적으로 전체 노드를 순회하지 않고 이주할 노드를 선택함으로써 수행되어진다. 그러나 이주 수행시 네트워크 환경 변화 및 부하로 인해 트래픽이 증가된 경우에는 스케줄링된 목적 노드로의 이주가 원활히 수행되지 못하므로 노드 순회 시간 비용이 증가하게 된다. 따라서 본 논문에서는 이러한 상황을 해결하기 위해 최적으로 이주 노드들의 경로를 지정하는 경로 탐색 알고리즘을 통해 에이전트의 노드 이주 시간 비용을 최소화할 수 있는 이주 기법을 제안한다. 제안된 기법은 이동 에이전트의 이주 신뢰성을 확보하며 순회 작업 처리 시 효율성을 높일 수 있다.

1. 서론

최근 발전하고 있는 분산 모바일 기술들 중 이동 에이전트 기술은 최근 정보 검색 및 분산처리 이동 컴퓨팅, 전자상거래, 네트워크 관리 등의 다양한 분야에 응용이 확산되고 있는 추세이다[1,2,3]. 그러나 이동 에이전트 시스템들은 동적 환경변화 및 이주비용에 대한 특징을 고려하지 못하고 있으며 결함에 대한 처리를 제공하지 못하는 문제점 때문에 이동 에이전트 이주 성능이 감소된다. 또한, 순차적으로 노드를 방문하여 결과를 누적시키며 이주하는 단순 구조로서 예측치 못한 네트워크 부하, 노드의 장애 및 서비스 부재와 같은 문제 발생 시 이주 중인 에이전트들이 특정 상황 변화에 대처하지 못하므로 이주비용을 증가시킨다[4,5,6,7,8]. 그러므로 이동 에이전트는 여러 노드로 이주할 때 이주할 목적지의 선정을 어떻게 할 것인지에 대한 방법이 요구된다 이는 이동 에이전트가 자율적으로 목적지를 결정할 수도 있고, 개발자가 개발할 때 직접 지정할 수도 있다 다른 방법으로는 이주할 목적지들의 전체 경로를 미리 지정하여 관리할 수 있다. 따라서 이러한 분산 환경에서의 특정한 상황 변화에 능동적으로 대처하여 사용자의 요구 정보에 대해 빠른 결과 제공 및 노드의 위치투명성에 의해 이주의 신뢰성을 제공하여 이동 에이전트의 효율적 이주와 성능을 최대화하기 위한 이주 노드 경로 기법이 요구된다.

본 논문에서는 네트워크 트래픽이 증가된 시점에서의 에

이전트 이주 시 사용자에게 의한 이주 스케줄 지정 방식이 아닌 네트워크 트래픽 감지를 이용한 목적 노드까지의 최적 경로 탐색 알고리즘을 통해 능동적인 경로 조정을 수행하는 적응적 이주 기법을 설계한다 이는 이동 에이전트의 노드 이주 시간 비용을 최소화할 수 있다

본 논문의 구성은 2장에서 관련 연구로서 기존 에이전트 시스템의 이주 기법에 대해 설명하고 3장에서는 제안된 최적 경로 탐색 방법의 알고리즘과 이를 통한 경로 탐색에 관한 수행과정을 기술한다 그리고 4장에서 제안된 최적 경로 탐색 알고리즘을 통한 적응적 이주 경로 기법의 구조와 수행과정을 제시한다 마지막 5장에서는 결론 및 향후 연구방향에 대해 기술한다

2. 관련연구

기존 이동 에이전트 시스템들의 이주 기법들은 단순히 에이전트 코드를 필요에 따라 전송하기 때문에 동적인 환경 변화에 적절히 대응하지 못하며 이주에 필요한 데이터의 크기 그리고 결함에 대해 적절히 대처하지 못하는 문제점이 있다. Ajanta[8,10]의 경우, 에이전트의 이주는 네트워크를 통해 이동할 에이전트를 클래스만을 이동시키는 이주 기법을 사용한다 이러한 클래스 로딩 기법은 수행에 필요한 코드만을 이동하는 방법으로 전송할 데이터양을 최소화함으로써 네트워크 부하를 감소시키지만 요구한 클래스만을 이동시키는 방법을 사용함으로써 이주 데이터의 크기를 고려하지 못하는 문제점과 결함에 대한 처리를 하지 못한다는 문제점이 있다 JAMES[11]은 이주 정보에 따라 필요한 코드를 각 에이전트 시스템에 메시지 형태로 전송하고 메시지를 수신한 에이전트

본 연구는 한국과학재단 특정기초연구(R01-2007-20989-0) 지원으로 수행되었음

시스템에서는 자신이 필요한 코드를 자신의 캐시메모리나 이전 에이전트 시스템의 캐시메모리 또는 자신의 보조기억장치에서 찾는 프리패칭 기법을 이용하여 이주할 때 발생할 수 있는 네트워크 트래픽을 개선하였다. 그러나 이주 노드에 대한 이주 스케줄 지정과 관련하여 순서대로 노드를 방문하고 방문한 노드에서 검색한 데이터를 포함한 이주 문제를 가지고 있다[9,11,12,13]. 이와 같이 기존 이동에이전트 시스템은 복잡하고 다양한 사용자 요구를 처리하는 에이전트의 작업 수행에 있어 이주 노드를 재지정해야 하는 불가피한 상황이나 통신망 혹은 노드의 결점과 같은 특정한 문제 발생 시 동적으로 에이전트의 이주를 수행할 수 없으므로 발생할 수 있는 여러 가지 요인들에 대해 능동적으로 대처할 수 없으며 방문해야 하는 노드의 수나 검색되는 데이터의 양이 조금이라도 많은 경우에는 노드 순회 이주 시간 비용이 증가되어 에이전트를 사용하는 것 자체가 비효율적일 수 있다.

3. 최적 경로 탐색 방법

3.1 시스템 구조

제안 이주를 지원하는 시스템은 네이밍 에이전트를 이용하여 각 에이전트간의 협력 관계를 유지하며 이주 스케줄 정보를 제공한다. 서버 푸시 에이전트(SPA), 클라이언트 푸시 에이전트(CPA), 이동 에이전트(MA) 등의 각 에이전트는 네이밍 에이전트에 등록되고 각 에이전트의 요청으로부터 이름에 의한 에이전트 참조를 지원한다. 또한, 서로 다른 에이전트들의 위치를 유지하여 분산 환경에 존재하는 객체에 대한 투명성을 보장한다. 이는 결과적으로 분산 객체의 통합을 유도하여 이주의 신뢰성을 제공하며 에이전트의 성능을 향상시키는 효율적 이주를 지원한다. 클라이언트 브라우저, 시스템 모니터링 에이전트(SMA)는 클라이언트 푸시 에이전트의 일부 구성 모듈로 운영된다. 그림 1은 제안 기법을 위한 시스템 구조이다.

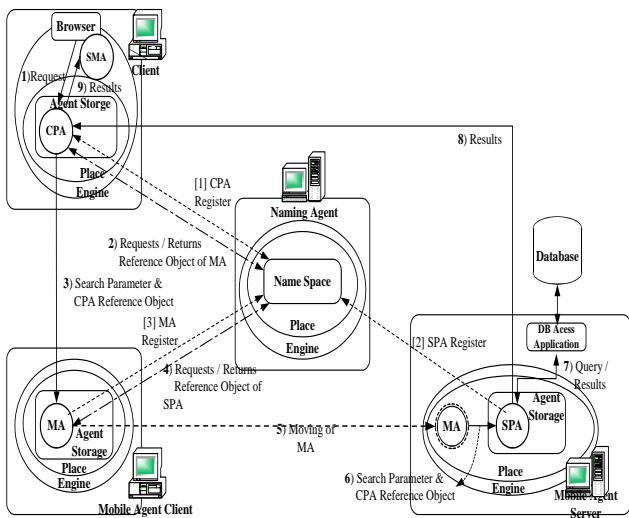


그림 1. 시스템 구조

에이전트간 협력을 위한 초기 작업은 에이전트 이름이 네이밍 에이전트에 등록됨으로써 수행되며 사용자의 이벤트 발생에 따라 상호 정보를 교류하기 위한 응답과 대로 이루어진다. 시스템에서의 수행 과정은 네이밍 에이전트로부터 분산된 해당 객체 정보를 요청하여 반환받으며 이를 기준으로 이주 노드 경로가 결정된다. 이렇게 결정된 노드 경로에서 이동 에이전트는 해당 서버로의 이주를 통해 자원 접근을 위한 실행을 요구하고 서버측에 위치하는 서버 푸시 에이전트와의 협력 작업을 수행한 후, 다음 이주 노드들에 대한 네트워크 트래픽을 측정하여 측정값의 결과를 통해 최적 경로 노드를 탐색하여 해당 노드로 이주한다.

3.2 최적 경로 탐색 알고리즘

최적 경로 탐색은 네트워크 트래픽 감지를 통해 현재의 노드에서 에이전트 이주 테이블의 각 목적 노드들 중 이주 경로가 최적인 노드를 판단하여 이주를 수행함으로써 이주 시 소요되는 시간 비용을 최소화한 에이전트 프레임워크이다. 제안 방법은 능동적 이주 스케줄 지정 방식을 사용함으로써 이동 에이전트가 과도한 트래픽 발생 경로를 회피하여 최적 경로를 통해 이주를 수행할 수 있도록 지원한다.

최적 경로 탐색을 통해 에이전트의 이주를 지원하는 기법은 다양한 사용자 요구나 문제들을 보다 효율적으로 처리함은 물론 네트워크 환경 변화에 동적으로 대처하여 이동 에이전트의 이주 성능을 향상시킨다. 최적 이주 경로의 탐색은 Ping 프로그램을 이용하여 서버 푸시 에이전트 객체 참조자 리스트의 각 노드에 패킷을 보내 평균 왕복 시간과 패킷 분실 수를 추출한다. 추출된 값과 서버 푸시 에이전트 객체 참조자 리스트의 IP, Port, 참조 객체자의 Key 값을 Vector V를 생성하여 V에 저장한다. V에 저장된 정보의 개수는 서버 푸시 에이전트 객체 참조자 리스트의 개수와 동일한 N 개이다. N 개의 정보들 중 최적 이주 경로를 탐색하기 위해 버블 정렬(Bubble Sort) 방식을 이용하여 최소의 패킷 분실 수와 최소의 평균 패킷 왕복 시간을 갖는 노드 정보 순으로 Vector V를 정렬한다. 이는 V의 노드 정보들 중 1번째에 위치한 정보의 패킷 분실 수와 2번째에 위치한 정보의 패킷 분실 수를 비교하여 2번째의 패킷 분실 수가 1번째 패킷 분실 수보다 작거나 두 분실 수가 똑같을 때 두 노드 정보들을 교환한다. 이러한 방법으로 N번째까지 수행하면 최상위에는 패킷 분실 수가 가장 적은 노드 정보들이 위치한다. 또한, 동일한 패킷 분실 수를 갖는 노드 정보들끼리도 패킷 평균 왕복 시간이 작은 정보들이 먼저 위치하게 된다. 이렇게 정렬된 노드 정보들은 또다시 패킷 분실 수가 5보다 작은 노드 정보들까지 재정렬을 수행하게 된다. 재정렬 방법은 1번째 패킷 분실 수와 2번째 패킷 분실 수의 차이가 1이고, 1번째의 평균 패킷 왕복 시간이 2번째의 평균 패킷 왕복 시간보다 2배 이상일 경우 1번째 정보와 2번째 정보를 교환한다. 이러한 방법으로 5번째의 노드 정보까지 재정렬을 수행했을 때 Vector V의 1번째 노드 정보가 최적 이주 노드 경로이다. 최적 이주 경로의 조정은 선택된 Vector V의 1번째 노드 정보와 서버 푸시 에이전트 객체 참조자 리스트의

번째 노드 정보를 비교했을 때 동일한 노드 정보가 아니면 서버 푸시 에이전트 객체 참조자를 재구성한다 또한, 선택된 최적 이주 경로를 통해 에이전트 이주 시 통신망 결손으로 인해 이주가 불가능할 경우 다음 순위의 이주 경로로 에이전트를 이주시킨다 그림 2는 최적 경로 탐색 알고리즘이다

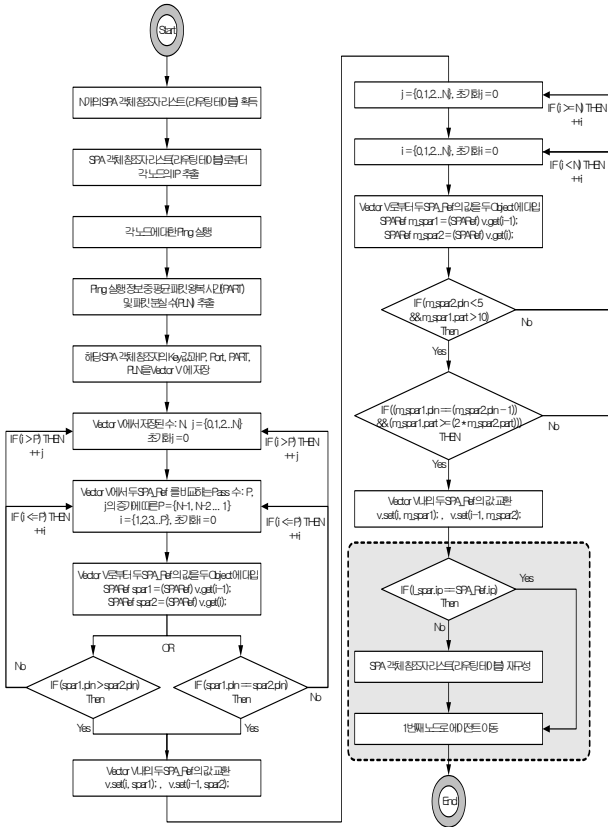


그림 2. 최적 경로 탐색 알고리즘

3.3 최적 경로 탐색 수행 과정

이동 에이전트 시스템에서 사용자 요구사항에 대한 작업 처리 시간은 에이전트가 부여받은 임무를 수행하는 비즈니스 로직의 실행 시간과 분산된 노드들 간을 이주하는데 걸리는 에이전트 이주 시간이 포함된다 이주 노드 탐색 방법은 네트워크 부하가 발생하였을 경우 이동 에이전트의 이주 시간을 단축시키기 위한 방법으로 이주할 노드들 중 최소의 네트워크 소요 시간을 갖는 경로를 선택하는 것이다 즉, 네트워크의 부하 발생 요인인 트래픽을 측정하여 과다한 트래픽이 발생하는 경로는 회피하고 최소의 네트워크 지연을 갖는 경로를 선택하여 보다 효율적인 작업 처리가 이루어지도록 하기 위한 방식이다.

이러한 최적 이주 노드 탐색 방식의 수행 과정은 먼저 사용자로부터 입력받은 검색 키워드를 이용하여 네이밍 에이전트에 등록된 모든 서버 푸시 에이전트 객체들 중 해당 정보 서비스를 지원할 수 있는 서버 푸시 에이전트 객체 참조자 리스트를 전달받는다 이 때, 서버 푸시 에이전트 객체 참조자 리스트는 적중 문건 수가 높은 순

로 정렬되어 있다. 처음 시작 노드에서 적중 문건 수가 우선인 이주 경로 테이블내의 서버 푸시 에이전트 객체 참조자를 이용하여 모든 원격 노드에 Ping을 실행한다. Ping을 실행하여 얻은 정보들 중 각 노드 간 패킷의 평균 왕복 시간인 평균값과 패킷의 분실 수인 분실값을 추출하며 노드까지의 경로들 중 패킷의 평균 왕복 시간과 패킷의 분실 수가 최소인 노드 경로를 찾기 위해 추출된 값들을 비교한다. 추출된 값을 통해 얻어진 노드 경로가 이주 경로 테이블에서 최상위의 서버 푸시 에이전트 객체 참조자에 해당하는 노드 경로인지 검사한다 만약, 최상위의 서버 푸시 에이전트 객체 참조자에 해당하는 노드까지의 경로가 아닌 경우 패킷 왕복 시간과 패킷 분실 수의 값이 최소인 노드 경로들 순으로 이주 경로를 재구성한다. 이 때, 패킷의 평균 왕복 시간 및 분실된 패킷의 수를 이용한 값 비교 시 패킷 분실 수가 적은 것을 우선으로 한다. 이동 에이전트는 이주 경로 테이블의 최상위 노드를 선택하여 에이전트를 이주시킨다

이주 노드 탐색 방식의 수행 과정 중 경로 탐색은 Ping을 통해 얻어진 패킷의 평균 왕복 시간과 패킷 분실 수의 값을 비교하여 이주 경로 테이블에서 최상위의 서버 푸시 에이전트 객체 참조자에 해당하는 노드까지의 경로가 최적인지를 판단하는 과정이다 Ping을 통해 얻어진 값들의 비교 수행을 통해 최상위의 서버 푸시 에이전트 객체 참조자에 해당하는 노드 경로가 최적이지만 다른 노드 경로들의 순서가 바뀌어 있어도 이주 경로를 재구성하지 않는다 이는 에이전트가 분산된 노드들을 이주하며 작업을 수행함으로써 이주한 노드들의 위치에 따라 이들의 순서가 새롭게 바뀔 수 있기 때문이다 경로 조정 과정은 최상위의 서버 푸시 에이전트 객체 참조자에 해당하는 노드 경로가 최적이지 아닌 경우에만 패킷의 평균 왕복 시간과 패킷 분실 수의 값이 최소인 노드 경로들 순으로 이주 경로를 재구성한다 이 때, 패킷의 평균 왕복 시간 및 분실된 패킷의 수를 이용한 값 비교 시 패킷 분실 수가 적은 것을 우선으로 이주 경로 테이블을 재구성하게 되는데 이는 통신망의 결손 발생으로 인한 이동 에이전트의 이주 보장을 위해서이다

4. 최적 경로 적응적 이주 기법

4.1 이주 기법

최적 경로 탐색을 통한 적응적 이주 기법은 먼저 사용자가 원하는 작업을 요청 시 네이밍 에이전트의 메타 데이터에 등록된 각 노드의 객체 정보를 얻어와 이주를 수행한다. 이때 얻어진 각 노드의 객체 정보는 적중 문건 수나 적중률을 기준으로 내림차순 정렬이 되어 있으므로 정보가득률에 따른 임계값을 설정하여 이주할 노드의 수를 제한한다 이러한 노드의 제한은 유효정보에 대한 유효정보 확보율을 높일 수 있으므로 결과반환 시간과 이주비용을 감소시킬 수 있으며 사용자가 위치 정보를 알고 있지 않더라도 단지 사용자가 원하는 서비스 정보에 대한 키워드만 알고 있으면 서비스를 제공받을 수 있도록 위치 정보 투명성을 제공한다 그림 3은 최적 경로 탐색을 통한 제안 이주 기법의 구조이다

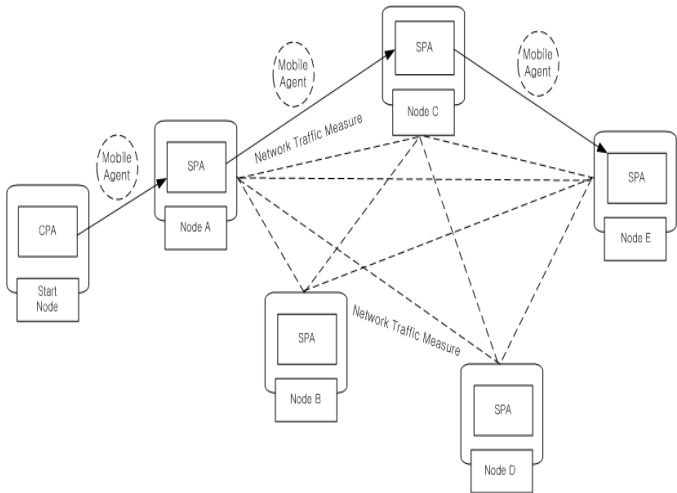


그림 3. 이주 기법 구조

그림 3에서와 같이 이동 에이전트의 노드 이주는 에이전트가 생성되기 전 사용자가 원하는 정보의 키워드를 입력 시 이를 네이밍 에이전트 서버에게 전달하여 네이밍 에이전트에 등록된 각 서버의 객체 정보를 전달받아 에이전트를 생성 후 이주를 수행한다 이때 전달받은 정보는 각 서버가 보유하고 있는 문서에서 적중 문건 수와 적중률을 기준으로 내림차순 정렬된 정보이다 그러므로 이동 에이전트는 동일 요청 작업을 수행하기 위해 전체 노드를 순회하지 않더라도 사용자가 원하는 결과를 반환하여 전체 네트워크 소요시간과 이동 에이전트의 성능을 향상시킬 수 있다. 그러나 각 해당 노드로 이주 시 네트워크 환경 변화와 부하로 인해 트래픽이 증가된 경우 이주 스케줄에 있는 각 노드들의 네트워크 트래픽 측정 을 통해 노드의 이주 순서는 재조정된다

4.2 이주 수행과정

이주 수행 과정은 각 에이전트의 상호협력력을 기반으로 이루어지며 사용자 요구 정보에 대한 검색의 신뢰성을 제공한다. 각 에이전트 관련 정보 등록 과정은 각 시스템들의 초기 기동 시, 각각의 시스템 내에 포함된 에이전트, 플래스, 에이전트 시스템에 대한 객체들을 네이밍 서비스를 지원하는 네이밍 에이전트에 우선적으로 등록된다. 등록된 객체 정보들은 에이전트들 간 통신이나 이동 에이전트의 이주 시 원격 시스템들 간의 연결이 가능하도록 위치 투명성을 제공하며 이동 에이전트가 분산된 각 노드들을 이주하며 작업을 처리할 수 있도록 지원한다. 이러한 이주 수행 시나리오는 다음과 같다
먼저, 클라이언트 브라우저를 통해 사용자가 요구하는 정보들을 검색하기 위한 키워드를 입력받는다 브라우저를 통해 입력된 검색 키워드가 클라이언트 푸시 에이전트에 전달되면, 클라이언트 푸시 에이전트는 네이밍 에이전트로부터 이동 에이전트의 객체 참조자와 자기 자신의 객체 참조자를 획득한다 이렇게 획득한 이동 에이전트의 객체 참조자를 이용하여 에이전트 클라이언트 내의 이동 에이전트와 연결한 후 검색 키워드와 클라이언트 푸시 에이전트의 객체 참조자를 전달한다 에이전트 클라이언트는 전달된 검색 키워드를 이용하여 네이밍 에이

전트로부터 서버 푸시 에이전트의 객체 참조자 리스트를 획득한다. 이 때, 서버 푸시 에이전트의 객체 참조자 리스트는 검색 키워드와 네이밍 에이전트의 메타 테이블 정보와의 비교를 통해 해당 객체 참조자들이 추출된다 에이전트 클라이언트는 획득한 서버 푸시 에이전트의 객체 참조자 리스트 중 우선순위에 의해 이주 경로가 최적인 노드를 선택한 후, 해당 노드로 에이전트 객체를 이주시킨다. 에이전트 서버로 이주한 이동 에이전트는 서버 푸시 에이전트에 검색 키워드를 전달하여 작업 실행 모듈을 실행시킨다 그 후, 이동 에이전트는 또다시 푸시 에이전트의 객체 참조자 리스트에서 최적의 경로에 위치한 서버 푸시 에이전트를 탐색한 후 해당 노드로 이주를 수행한다. 한편, 서버 푸시 에이전트는 작업 실행 모듈의 흐름에 따라 데이터베이스 핸들러를 이용하여 데이터베이스에 접근한다 데이터베이스 핸들러는 데이터베이스를 검색한 후, 해당 검색 결과를 반환한다. 서버 푸시 에이전트는 데이터베이스로부터 반환된 검색 결과를 클라이언트 푸시 에이전트에게 전달한다 클라이언트 푸시 에이전트는 서버 푸시 에이전트들로부터 전달받은 검색 결과를 필터링 한 후, 사용자 브라우저에 디스플레이 한다.

아래 그림 4는 사용자로부터 요구사항을 입력받아 이를 처리하여 사용자에게 서비스해줄기까지의 제안 기법의 이주 실행을 위한 각 에이전트간의 통신 과정이다

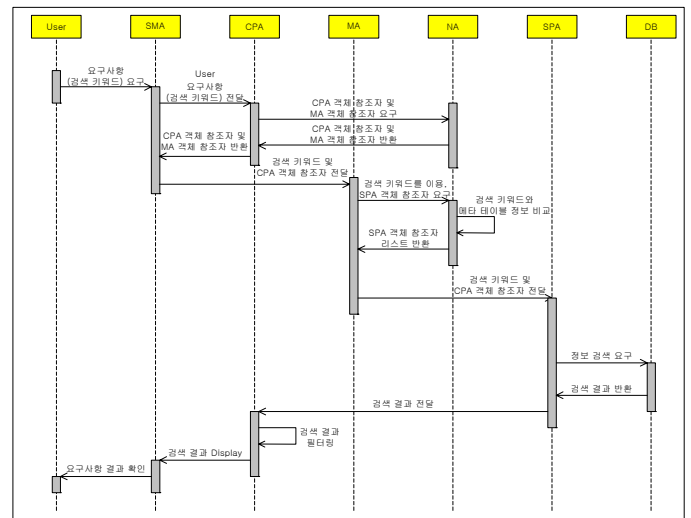


그림 4. 이주 통신 과정

이와 같이 최적 경로 탐색 적응적 이주 실행을 위한 각 에이전트의 통신 과정은 사용자로부터 요구사항을 입력받아 이를 처리하고자 네이밍 에이전트의 객체 참조자 리스트의 정보 요청 및 반환과 이러한 정보를 통해 정보 가득물의 임계값에 의해 이주해야 할 노드 수를 결정하며 결정된 이주 노드로 이동 에이전트가 이주를 수행하여 사용자가 요구한 작업의 결과를 반환함과 동시에 각 해당 노드로 이주시 네트워크 환경 변화에 대처함으로써 발생하는 문제를 최소화할 수 있으며 또한 이주 신뢰성을 유지할 수 있다.

5. 결론 및 향후 연구과제

본 논문에서는 네트워크 트래픽 감지를 이용한 이주 노드들에 대한 최적 경로 탐색을 통해 사용자 요구 처리에 있어 최소의 이주 시간을 가지는 이동 에이전트의 적응적 이주 기법을 설계하였다 또한, 네트워크 환경 변화 및 트래픽 가중 문제를 해결하기 위해 Ping을 이용한 최적 경로 탐색 알고리즘을 제시하였으며 사용자에게 의한 이주 스케줄 지정 방식에서 탈피하여 능동적인 이주 스케줄 지정이 가능하도록 하였다 이를 통해 특정 상황들에 동적으로 대처할 수 있을 뿐만 아니라 자동적인 로드 조절이 가능해짐으로써 이동 에이전트가 최소 이주 시간 비용을 갖도록 최적 이주 경로를 지정할 수 있다 이에 사용자의 요구 정보에 대해 빠른 결과 제공 및 노드의 이주 신뢰성을 제공하여 이동 에이전트의 성능을 최대화할 수 있다.

향후 연구 과제로는 제안된 기법을 적용한 정보 검색 시스템의 개발이 요구되며 최적 경로 탐색을 위해 수행되는 Ping의 대체 방법 및 능동적인 이주 스케줄에 대한 접근 및 갱신 시간 증가에 따른 문제를 해결할 수 있는 방법에 대한 연구와 평가가 계속되어야 한다

Design for Agent-based White Pages", AAMAS'04, pp.1476-1477, 2004.

- [10] Steven, P.F., Martin, L.G., Reed, L., "Agent behavior architecture a MAS framework comparison", AGENTS, International Conference on Autonomous Agents, pp. 86-87, 2002.
- [11] L. M. Silva and P. Simoes, "JAMES : A platform of Mobile Agents for the Management of Telecommunication Networks", The International Workshop on Intelligent Agents for Telecommunication Application, pp.76-95, 1999.
- [12] 김광중, 김영자, 고현, 이연식, "네트워크 소요시간 최소화를 위한 이동 에이전트의 멀티캐스트 이주 모델 구현", 한국정보처리학회논문지(D) 제12-D권 제2호, April, 2005.
- [13] 김광중, 김영자, 고현, 이연식, "데이터 검색의 적중률 향상을 위한 이중 캐시의 푸시 에이전트 모델 설계", 한국컴퓨터정보학회논문지 제10권, 제6호, March, pp.153-166, 2006.

참고문헌

- [1] P. Metes, "A Pattern based development tool for mobile agents", SIGCSE 05', St.Louis, Missouri, USA, pp.72-75, 2005.
- [2] Haizheng, Z., Bruce, C.W., Brian, L., Victor, L., "A Multi-Agent Approach for Peer-to-Peer Based Information Retrieval System", AGENTS, International Conference on Autonomous Agents, pp.456-463, 2004.
- [3] T. Marian, B. Dumitriu, M. Dinsoreanu and I. Salomie, "A framework of reusable structures for mobile agent development", Mobile Computing and Communications Review, Vol.10, No.2, pp.112-116, 2004.
- [4] P. Metes, "A Pattern based development tool for mobile agents", SIGCSE 05', St.Louis, Missouri, USA, pp.72-75, 2005.
- [5] Tino, S., Peter, B., Ryszard, K., "Towards Autonomous Mobile Agents with Emergent Migration Behaviour", AAMAS'06, pp.585-592, 2006.
- [6] T. Schlegel, P. Braun and R. Kowalczyk, "Towards Autonomous Mobile Agents with Emergent Migration Behaviour", AAMAS 06', May 8 - 12, Hakodate, Hokkaido, Japan, pp.585-592, 2006.
- [7] Y. Berbers, B. D. Decker and W. Joosen, "Infrastructure for mobile agents", ACM AGENTS 04', pp.173-180, 2004.
- [8] A. R. Tripathi, N. M. Karnik, T. Ahmed, R. D. Singh, A. Prakash, V. Kakani, M. K. Vora and M. Pathak, "Design of the Ajanta Systems for Mobile Agent Programming", Journal of Systems and Software, Vol.62, No.2, pp.123-140, 2002.
- [9] Todd, W., "Naming Services in Multi-Agent Systems: A