

# 이동 컴퓨팅 서비스를 위한 부분 연결 관리자의 설계 및 프로토타이핑

김 평 중<sup>†</sup> · 윤 석 환<sup>††</sup> · 진 성 일<sup>†††</sup>

## 요 약

이동 컴퓨팅 서비스들 효율적으로 제공하기 위해서는 이동 호스트와 이동 서버간에 높은 에러율과 낮은 전송률을 가진 무선 매체를 사용하는 문제와 이동 호스트의 전력 소비를 줄이거나 네트워크 접속 비용을 줄이기 위하여 갑자기 단절되는 문제를 해결하여야 한다. 이를 위하여 본 논문에서는 부분 연결 관리자의 구조를 제안하고 설계 및 구현하였다. 특히, 사용자가 요청한 사항을 신속하게 처리하기 위하여 우선 순위에 따라 제한된 개수만큼 이동 에이전트를 생성하여 서버 시스템으로 병렬로 전송하고 결과들을 취합하도록 하였다. 제안된 부분 연결 관리자는 이동 컴퓨팅 서비스인 장터 서비스에 적용함으로써 이동 컴퓨팅 환경에 이동 에이전트 기술이 적합하며, 이동 컴퓨팅 환경에서 발생할 수 있는 부분 연결 문제를 극복할 수 있음을 일었다

## Design and Prototyping of Partial Connection Manager for Mobile Computing Service

Pyeong-Jung Kim<sup>†</sup> · Seok-Hwan Yoon<sup>††</sup> · Seong-Il Jin<sup>†††</sup>

## ABSTRACT

We must solve problems caused by mobile computing environments for supporting efficient mobile computing services. The problems is the use of wireless medium having high error rates and low bandwidth, and the sudden network disconnection to reduce the power consumption of a mobile host and the cost of the network connection. For these problems, we proposed the architecture of the partial connection manager and designed and implemented that in this paper. In particular, The partial connection manager creates the limited number of mobile agents according to priority and sends them in parallel to server systems and combines results of them to process rapidly the user request. By applying the proposed partial connection manager to the mobile computing services called cybermarket, we knew that the mobile agent technique could be suited to the mobile computing environment and overcome the partial connection problem caused by the mobile computing environment.

### 1. 서 론

이동 컴퓨팅에서는 호스트 이동성과 이동 호스트의 하드웨어 특성 그리고 무선 네트워크로 인하여 호스트가 고정된 환경의 분산 시스템에서 발생하지 않는 재로

운 문제점들이 발생된다. 특히, 이동 컴퓨팅 서비스를 효율적으로 제공하기 위하여 이동 호스트와 이동 서버 간에 높은 에러율과 낮은 전송률을 가진 무선 매체를 사용하는 문제와 이동 호스트의 전력 소비를 줄이거나 네트워크 접속 비용을 줄이기 위하여 갑자기 단절되는 문제를 해결하여야 한다[1, 2]. 본 논문에서 이러한 이동 컴퓨팅 문제점을 부분 연결 문제로 정의한다.

부분 연결 문제는 이동 에이전트 기술을 사용하여

† 김 회 원 : 도림중북대학교 컴퓨터정보과 교수  
†† 중신회원 : 정보통신연구원 용지1팀 팀장  
††† 중신회원 : 충남대학교 컴퓨터학과 교수  
논문집수 1999년 10월 22일, 심사완료 2000년 6월 5일

효율적으로 해결할 수 있다. 이동 에이전트는 실행 코드가 목표 달성을 위해 호스트를 이동하며 작업을 수행한다. 이동 에이전트는 네트워크 연결을 계속 유지한 상태로 작업하지 않고 실행 코드 자체가 시스템간을 이동하며 임무를 수행하기 때문에 네트워크 연결이 불안정하거나 부하가 많이 걸리는 환경에서 유용하게 활용될 수 있다[2-4].

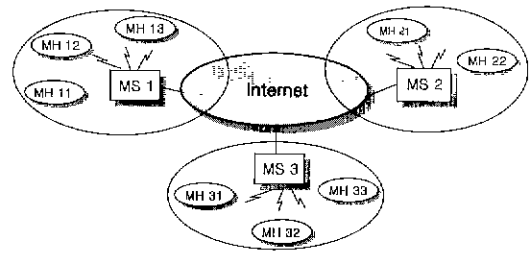
이동 에이전트 패러다임이 이동 컴퓨팅 서비스에 적합하다 하더라도 부분 연결 문제를 해결해주는 것은 아니다. 이 문제에 대하여 이동 에이전트 기술을 활용하여 해결하기 위한 시도는 Gray와 Brewington에 의하여 이루어졌다. Gray는 이동 에이전트 시스템에 이동 컴퓨팅 환경의 비연결 연산을 지원하기 위하여 랩톱 도킹 시스템(laptop docking system)을 제안하였다[2]. 한편, Brewington은 원하는 정보를 찾을 때까지 전체 기대 시간을 최소화하기 위하여 방문 리스트를 결정하는 Traveling Agent Problem을 해결하기 위하여 이동 에이전트 계획 시스템을 제안하였다. 이 시스템은 이동 컴퓨팅 환경을 고려한 것은 아니지만, 계획 모듈, 네트워크 탐지 모듈, 및 윈도우 페이지 모듈로 구성되어 Traveling Agent Problem을 해결하고 있다[5].

본 논문에서는 부분 연결 관리자를 통하여 부분 연결 문제를 해결하고 이동 컴퓨팅 서비스인 장터에 적용하였다. 부분 연결 관리자는 네트워크 모니터링 모듈, 디렉토리 서비스 모듈과 병렬 계획 모듈 등 3개의 모듈로 구성하였고 이러한 모듈들은 이동 서버의 고정된 에이전트로 구현된다. 이를 구현하기 위한 기본 시스템으로 IBM의 Tokyo Research Lab.(TRL)에서 개발한 자바 기반의 Aglets를 사용하였으며[3, 6], Windows98과 Windows NT 시스템에서 자바 가상 머신의 API를 사용하여 구현하였다. 결론적으로 부분 연결 관리자를 이용함으로써 이동 컴퓨팅 서비스를 효율적으로 제공할 수 있음을 보여주고 있다.

본 논문의 구성은 다음과 같다. 2장에 이동 컴퓨팅 시스템 모델과 이동 컴퓨팅 서비스를 기술하고, 3장에서는 이동 컴퓨팅의 고유 특성으로부터 발생하는 문제점과 부분 연결 문제를 정의한다. 4장에서는 부분 연결 문제를 해결하기 위하여 이동 에이전트 기술을 이용한 부분 연결 관리자를 기술하고 5장에서 이를 설계 및 프로토타입핑 하고 6장에서 장터 서비스에 적용하였다. 마지막으로 7장에서는 결론 및 향후 연구를 기술한다.

## 2. 이동 컴퓨팅 서비스

이동 컴퓨팅 시스템 모델은 (그림 1)과 같이 기존의 고정 네트워크(인터넷)과 더불어 이동 관련 개체(이동 호스트, 이동 서버, 그리고 무선 네트워크)가 추가되어 구성된다[1, 7, 8]. 예를 들어 *MH 11*이 *MH 33*에게 메시지를 전달하는 과정을 살펴보면, *MH 11*은 먼저 무선 접속을 통하여 *MS 1*에게 메시지를 전달한다. *MS 1*은 인터넷을 통하여 *MH 33*을 현재 서비스하고 있는 *MS 3*에게 전달하고, 마지막으로 *MS 3*은 무선 인터페이스를 통하여 메시지를 *MH 33*에게 전달한다.



(그림 1) 이동 컴퓨팅 시스템 모델

Marsh는 이동 컴퓨팅 서비스를 다음과 같이 적어도 4개 이상으로 분류하고 있다. 첫째, 이동 호스트의 사용자는 언제 어디서든지 데이터베이스에 질의, 전자도서(잡지, 신문, 등)로부터 정보 검색, 하이퍼텍스트나 대화형 문서의 탐색등을 포함하는 정보 브라우징이다. 둘째, 이동 호스트의 사용자는 어떤 장소에서든지 비동기적 통신을 이용하여 전자 우편이나 문서들을 보내고 받는 것을 포함하는 개인 통신이다. 셋째, 협력 컴퓨팅 모형에 근거하여 다수의 사용자가 동시에 화이트보드를 사용하거나 대화적인 게임을 하는 다수대화이다. 넷째, 현장에서 이동 호스트를 사용하여 데이터를 등록하고 평가하는 데이터 입력이다[9].

Lmiclinski는 수직적 응용과 수평적 응용으로 구분하고 있다. 수직적 응용은 택배, 메일 추적, 환동 매지 등의 서비스에 사용되고 있다. 수평적 응용은 메일 응용과 정보 서비스로 나누고 메일 응용은 시간과 위치에 관계없이 메일을 송수신하고, 기존에 정의된 조건(비행기 시간, 교통 정보 등)에 따라 경보를 울리거나 사용자 프로파일에 따라 전자 뉴스 서비스, 지역에 따른 교통 정보나 기상 정보 서비스, 사용자의 포트폴리오를 사용하여 주식 정보 서비스를 포함한다. 정보서

비스는 위치에 따라 관련 정보 선택(가장 가까운 병원) 서비스, 지역 정보 서비스(지역 옐로우 페이지, 영화, 백화점 서비스) 그리고 형태 기반 트랜잭션 서비스(판매 및 주문 서비스) 등을 포함한다[10].

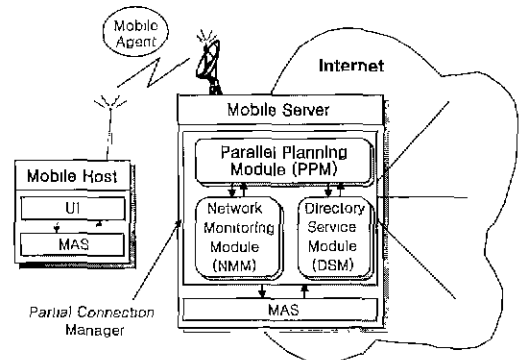
### 3. 부분 연결 문제

이동 컴퓨팅 서비스를 위하여 다음과 같은 이동 컴퓨팅 고유 특성으로부터 발생하는 문제점들을 해결하여야 한다. 첫째, 위치 설정 문제이다. 이동 호스트와 통신하기 위해서는 먼저 그 호스트의 현재 위치를 설정해야 한다. 호스트의 이동성으로 말미암아 네트워크 접속 위치가 계속 변화되고, 그에 따라 이동 호스트의 IP 주소(여기에서는 인터넷의 접속점 역할)가 계속 변하는 문제가 발생된다. 본 논문에서는 이 문제를 IETF IP Mobility Support[8]를 활용함으로써 해결하는 것으로 가정하였다 둘째, 무선 매체를 사용한다. 이동 서버는 자신의 셀 영역에 위치하고 있는 모든 이동 호스트들에게 메시지를 방송하는 방식으로 전달한다. 이러한 무선 매체는 상대적으로 높은 에러율과 낮은 전송률을 갖는다 셋째, 호스트 이동성은 단절 현상을 발생시킨다. 일반적으로 이동 호스트는 독립된 전원을 사용하기 때문에 전원의 효율성을 위하여 의도적으로 단절하는 경우가 많다. 또한, 호스트가 셀과 셀 사이를 이동할 때 핸드오프가 발생하기 때문에 의사성 단절이 발생한다. 넷째, 기존의 분산 알고리즘들은 트리나 스타와 같은 네트워크 위상의 논리 구조를 활용하고 있지만 이동 호스트가 이동함에 따라 이 구조를 변화시키기 때문에 필요할 때마다 동적으로 재구성되어야 한다[1, 2]. 본 논문에서는 상기의 문제점들 중 이동 호스트와 이동 서버간에 높은 에러율과 낮은 전송률을 가진 무선 매체를 사용하는 문제와 이동 호스트의 전력 소비를 줄이거나 네트워크 접속 비용을 줄이기 위하여 갑자기 단절되는 문제를 부분 연결 문제로 정의하고, 이동 에이전트 기술을 사용하여 효율적으로 해결하고자 한다.

### 4. 부분 연결 문제를 위한 이동 에이전트

부분 연결 문제에 대하여 이동 에이전트 기술을 활용하여 해결하기 위한 시도는 Gray와 Brewington에 의하여 이루어졌다. Gray는 이동 에이전트 시스템에 이동 컴퓨팅 환경의 비연결 연신을 지원하기 위하여

랩톱 도킹 시스템을 제안하였다. 이 시스템에서는 모든 호스트에 dock을 갖고있고 각 dock에는 dockmaster를 갖고 있다. 만약 이동 에이전트가 어떤 호스트를 방문하지 못할 경우 스스로 dockmaster의 큐에 들어간다 dockmaster가 그 호스트와 재연결에 성공하면 그 호스트를 가기 위해 대기하고 있는 모든 이동 에이전트들을 보낸다[2]. Brewington은 Traveling Salesman Problem과 유사한 Traveling Agent Problem을 해결하기 위하여 이동 에이전트 계획 시스템을 제안하였다 이 문제는 원하는 정보를 찾을 때까지 전체 기대 시간을 최소화하기 위한 방문 리스트를 결정하는 것이다. 이 시스템은 이동 컴퓨팅 환경을 고려한 것은 아니지만, 계획 모듈, 네트워크 탐지 모듈, 및 옐로우 페이지 모듈로 구성되어 Traveling Agent Problem을 해결하고 있다[5].



(그림 2) 이동 서버에서 부분 연결 관리지

본 논문에서는 부분 연결 문제를 해결하기 위하여 부분 연결 관리자를 이동 서버의 이동 에이전트 시스템(MAS)에 추가한다 부분 연결 관리자는 (그림 2)에서와 같이 병렬 계획 모듈, 디렉토리 서비스 모듈, 및 네트워크 모니터링 모듈 등 3가지 모듈로 구성된다. (그림 3)에서처럼 사용자 요구 사항을 담은 이동 에이전트가 이동 호스트로부터 이동 서버에 도착하면 제일 먼저 병렬 계획 모듈에 사용자 요구 사항을 전달한다. 병렬 계획 모듈은 이동 에이전트가 주어진 작업을 완성하도록 서버의 방문 리스트를 디렉토리 서비스 모듈에게 묻는다 디렉토리 서비스 모듈은 Infoseek나 Altavista처럼 우선 순위를 갖춘 서버의 방문 리스트를 반환한다. 우선 순위는 원하는 정보를 찾을 확률이다. 본 논문에서는 확률을 측정할 수 있다는 가정을 하였다. 병렬 계획

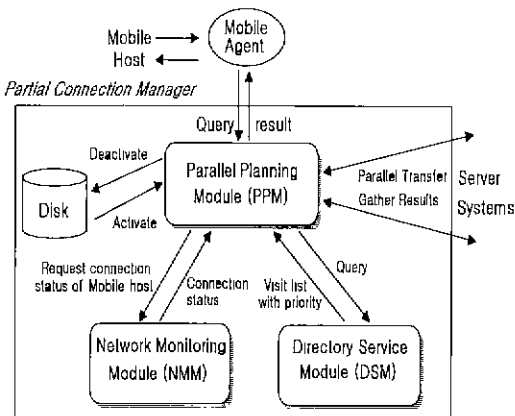
모듈은 우선 순위에 따라 제한된 개수만큼 이동 에이전트를 생성하여 병렬로 전송한다. 병렬 계획 모듈은 무한정 결과를 기다릴 수 없기 때문에 타임아웃을 주어 제한된 시간에 되돌아온 결과만을 수집한다. 이 결과를 분석하여 향후 디렉토리 서비스를 위하여 방문 이력을 디렉토리 서비스 모듈에 전송한다. 디렉토리 서비스 모듈은 방문 분석 결과를 테이블에 저장하여 향후 방문 이력 결과를 이용한다. 수집된 결과를 이동 호스트로 보내기 위하여 네트워크 모니터링 모듈에게 연결 상태를 질의한다. 네트워크 모니터링 모듈은 Ping과 같은 도구를 사용하여 주기적으로 이동 호스트와의 연결 상태를 갖고 있다. 만약 연결 상태이면 즉시 이동 에이전트를 보내 수집된 결과를 반환하고, 그렇지 않은 경우 비활성화 하여 이동 서버의 디스크에 저장한다. 이동 호스트가 다시 연결하여 결과를 요청하면 병렬 계획 모듈은 결과를 포함하고 있는 이동 에이전트를 빈환함으로써 부분 연결 문제를 해결한다

에이전트를 병렬로 전송하고, 이동 컴퓨팅 서비스를 제공하기 위하여 네트워크 연결 상태를 모니터링 하는 점이다.

### 5. 부분 연결 관리자의 설계 및 프로토타입핑

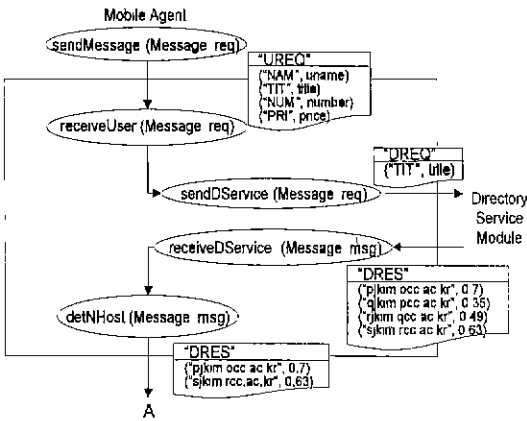
#### 5.1 병렬 계획 모듈

병렬 계획 모듈은 디렉토리 서비스 인터페이스 부분, 병렬 전송 부분, 및 네트워크 모니터링 인터페이스 부분으로 구분한다. 디렉토리 서비스 인터페이스 부분은 (그림 4)와 같이 이동 호스트로부터 도착한 이동 에이전트의 사용자 요구 메시지 UREQ를 receive User(Message req) 함수를 통하여 접수한다. 병렬 계획 모듈은 sendDService(Message req) 함수를 통하여 키워드인 title을 DREQ 메시지에 담아 디렉토리 서비스 모듈로 보내고 응답 함수인 receiveDService(Message msg) 함수를 통하여 DRES 메시지를 받는다. DRES 메시지는 도메인 이름과 우선 순위를 받는다. 현재 우선 순위가 50% 이상인 서버들 수만큼 이동 에이전트를 생성하여 병렬로 이동시킨다. 여기에서 디렉토리 서비스 모듈은 Inforseek나 Lycos처럼 우선 순위를 확률로 계산할 수 있음을 가정하였다. 병렬 전송 부분은 (그림 5)와 같이 setTime(int timeOut) 함수를 사용하여 타임아웃을 설정하고 시간 내에 도착한 이동 에이전트 결과만을 취합한다. 병렬 전송은 우선 순위에 따라 개수가 결정된다. 만약 타임아웃 이후에 도착한 이동 에이전트의 결과는 무시하기 위하여 이동 호스트 주소, Master ID, Aglet ID를 테이블에 보관한다. 이동 에이전트의 이동은 dispatch(URL url) API를 이용한다. 각 이동 에이전트의 수행 결과들은 취합하기 전에 향후 디렉토리 서비스를 위하여 방문 이력을 디렉토리 서비스 모듈에 전송한다. 네트워크 모니터링 인터페이스 부분은 (그림 6)과 같이 수집된 결과를 이동 호스트로 보내기 위하여 네트워크 모니터링 모듈에게 sendNMonitoring(URL url)을 통하여 연결 상태를 질의한다. 연결 상태이면 바로 보내고 그렇지 않으면 deactivateAglet(long duration) 하여 디스크에 저장한다. 이동 호스트가 다시 연결하여 요청한 결과를 요구하면 그 이동 호스트로 가기 위하여 대기하고 있던 모든 이동 에이전트를 보내야한다. 이를 위하여 각 이동 호스트마다 대기 테이블을 갖고 있다

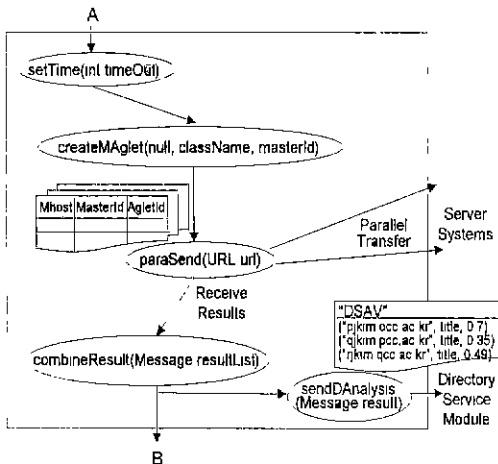


(그림 3) 부분 연결 관리자의 데이터 흐름

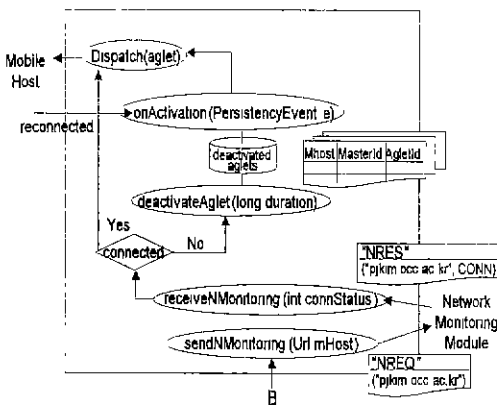
본 논문의 부분 연결 관리자는 Gray의 랩톱 도킹 시스템과 비교해보면, 이동 서버에만 부분 연결 관리자가 있어 비대칭형 구조를 갖고 있고, 디렉토리 서비스가 방문 이력을 이용하며, 이동 서버에게 서비스를 요구한 이동 호스트들만 네트워크 연결 상태를 모니터링하고, 병렬로 서버들에게 이동 에이전트를 보내는 점이다. Brewington의 이동 에이전트 계획 시스템과 비교해보면, 전체 기대 시간을 줄이기 위하여 방문 계획을 세우지 않고 우선 순위가 있는 서버들에게만 이동



(그림 4) 병렬 계획 모듈의 디렉토리 서비스 인터페이스 부분



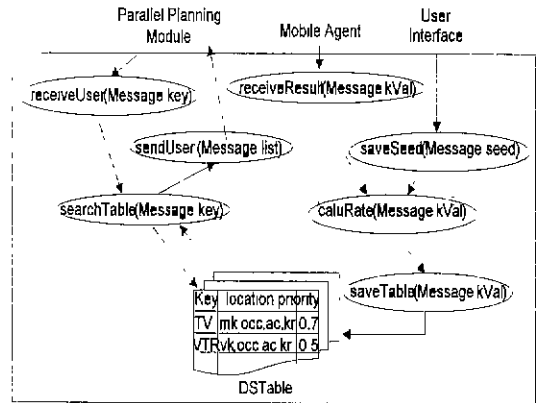
(그림 5) 병렬 계획 모듈의 병렬 전송 부분



(그림 6) 병렬 계획 모듈의 네트워크 모니터링 인터페이스 부분

52 디렉토리 서비스 모듈

디렉토리 서비스 모듈은 (그림 7)과 같이 병렬 계획 모듈의 요구를 receiveUser(Message key)로 접수받아 searchTable(Message list)을 통하여 방문될 서버 시스템 리스트를 검색한 후 sendUser(Message list)를 통하여 반환한다. 디렉토리 서비스 테이블을 유지하기 위하여 처음에 이동 서버의 운영자가 사용자 인터페이스를 사용하여 방문될 서버 시스템의 초기 scdd 주소를 입력하면 saveSeed(Message seed)를 통하여 입력된다. 향후 이동 에이전트의 방문 결과를 분석하여 저장한다. 또한, 이동 에이전트가 서버 시스템을 방문한 후 방문 이력 결과를 receiveResult(Message kVal)를 통하여 입력한다. 우선 순위를 계산하기 위하여 calcuRate(Message input)를 사용한다. 디렉토리 서비스 모듈은 이동 에이전트의 방문 횟수가 증가함에 따라 사용자 요구에 근접한 정보를 보유하게 된다.

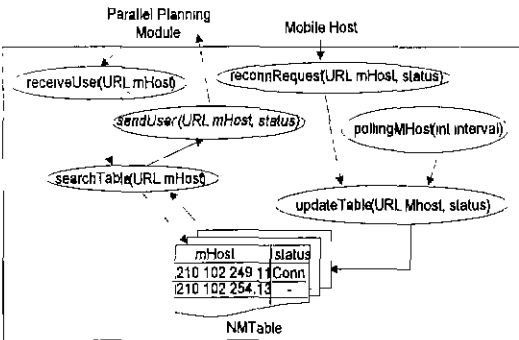


(그림 7) 디렉토리 서비스 모듈

53 네트워크 모니터링 모듈

네트워크 모니터링 모듈은 (그림 8)과 같이 병렬 계획 모듈의 요구를 receiveUser(URL mHost)로 접수받아 searchTable(URL mHost)에 따라 DTable을 검사한 후 이동 호스트의 무선 네트워크 연결 상태를 sendUser(URL mHost, int status)를 통하여 반환한다. DTable의 값을 정확히 유지하기 위하여 주기적인 폴링 방법과 이동 호스트로부터 재연결 요청 방법 등을 사용한다. 첫째, 주기적인 폴링 방법은 pollingMHost(int interval)을 통하여 이루어진다. pollingMHost(int interval)는 이동 서버에 등록된 이동 호스트만을 상대로 interval 간격으로 ping을 사용하여 폴링한다. 이 방법은 추가적인 네트

워 트래픽을 발생시키기 때문에 이를 최소화시키면서 정확한 정보를 유지하는 것이 중요하다. 현실적으로 폴링 방법은 이동 호스트가 실제적으로 연결되어 있더라도 네트워크 접속 현상 때문에 연결되지 않은 경우도 존재한다 따라서 이를 보완할 수 있는 방법이 필요하다. 현재 interval 간격은 1분으로 설정되어 있다. 둘째, 이동 호스트로부터 재연결 요청이 온 경우이다. reconn-Request(URL mHost, status) 이벤트가 자기 셀에 존재하는 이동 호스트인지 아니면 다른 셀에서 온 이동 호스트인지를 구분한다. 이동 호스트가 재연결 요청을 할 때마다 현재의 네트워크 주소를 알리기 때문에 update-Table(URL mHost, status)를 통하여 NMTable을 갱신한다. 그리고 병렬 계획 모듈에도 재연결 이벤트를 보내 대기중인 비활성 이동 에이전트를 보내도록 한다

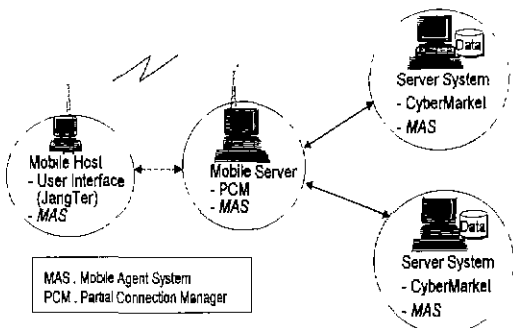


(그림 8) 네트워크 모니터링 모듈

6. 부분 연결 관리자의 적용 : 장터 서비스

6.1 장터 시스템 구성

시골 장터 서비스를 위한 전체적인 시스템은 (그림 9)

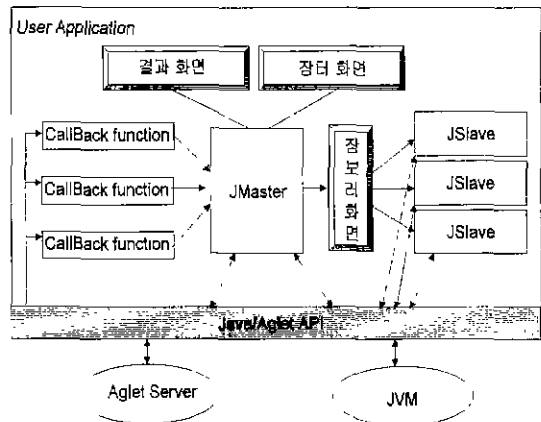


(그림 9) 장터 시스템 구성

와 같이 구성된다. 이동 호스트에는 MAS와 사용자 인터페이스인 JangTer가 장착되고, 이동 서버에는 MAS와 부분 연결 관리자, 서버 시스템에는 MAS와 간단한 거래 정보 시뮬레이션을 제공하는 CyberMarket이 장착된다.

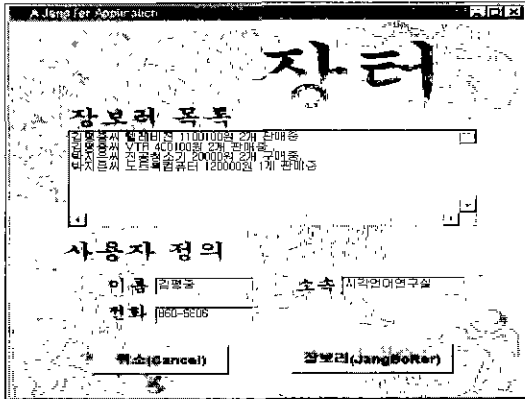
6.2 사용자 인터페이스

사용자 인터페이스는 (그림 10)과 같이 JMaster를 중심으로 정터 화면, 장보러 화면, 및 결과 화면으로 구성하고, Aglets가 이벤트 모델이기 때문에 이를 처리하는 콜백 함수, 각 품목마다 장보기를 위하여 이동 에이전트인 JSlave, 하단에 이들을 지원하는 자비 가상 머신 API와 Aglets API 등으로 구성한다.

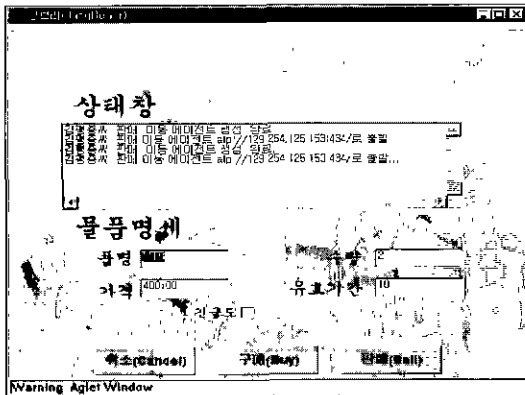


(그림 10) JMaster 구성 요소들간의 관계

사용자 인터페이스를 수행 관점에 보면, 우선 이동 호스트의 사용자가 JMaster를 시작하면 onCreate()이 불리고 이 함수 내에서 (그림 11)과 같은 장터 화면을 디스플레이 한다. 장터 화면에서 사용자 정보(이름, 소속, 전화)를 Rfp 클래스에 저장하여 JMaster 클래스에게 전달한다. 사용자가 장보러 버튼을 누르면 (그림 12)의 장보러 화면이 디스플레이 된다. 장보러 화면에서 사용자가 물품 정보(품명, 수량, 가격, 유효 기간, 및 긴급도)를 Rfp 클래스에 저장하여 JMaster 클래스에게 전달한다. 이 화면에서 구매 버튼이나 판매 버튼을 누를 때마다 이동 에이전트인 JSlave를 생성하여 Rfp 정보를 메시지(RFPTPSLAVE)를 전달하고 이동 서버로 전송한다. JMaster는 여러 품목에 대한 정보를 보냈을 경우 여러 개의 JSlave를 생성하여 보낸다.



(그림 11) 장터 화면



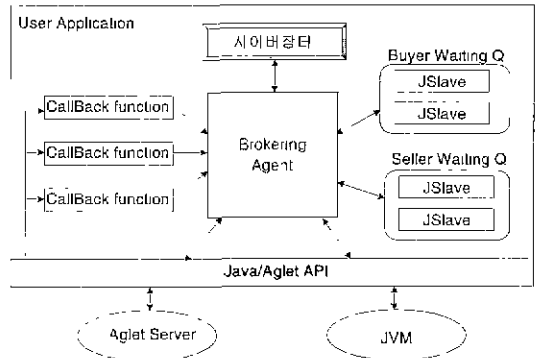
(그림 12) 장보러 화면

사용자가 이전에 요청했던 작업의 수행 결과를 알고자 할 경우 이동 호스트가 이동 서버에 연결되어 있거나 재 연결 요청을 함으로써 이루어진다. JSlave 수행 결과는 콜백 함수에 의하여 JMaster에게 보고한다. JSlave는 자기 주인인 JMaster\_Id를 갖고 있어 올바르게 보고한다. 수행 결과를 보고할 때에도 BUYRESULTTOSLAVE, SELLRESULTTOSLAVE, SELLERTOSLAVE, BUYERTOSLAVE 등의 메시지를 통하여 보고된다. JMaster는 JSlave의 리스트(JSlave\_Id, JMaster\_Id)를 관리함으로써 서로의 상관 관계를 준다. JSlave가 되돌아왔을 경우 JMaster에게 결과보고 메시지를 띄워 보고하고 은퇴한다.

### 6.3 서버 시스템

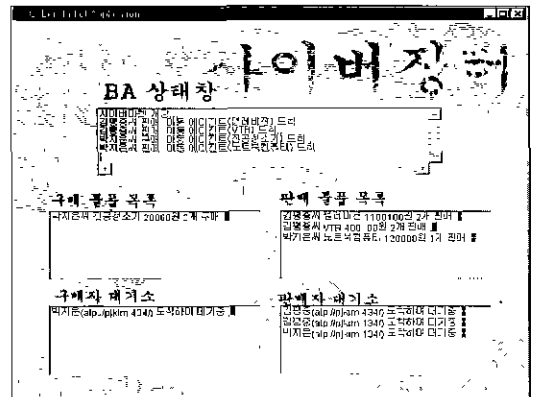
사이버장터는 (그림 13)과 같이 Brokering을 중심으

로 사이버장터 화면, Aglets가 이벤트 모델이기 때문에 이를 처리하는 콜백 함수, 구매 이동 에이전트가 도착하여 쉬는 대기열, 판매 이동 에이전트가 도착하여 쉬는 대기열, 하단에 이들을 지원하는 자바 가상 머신 API와 Aglets API 등으로 구성한다. 이동 에이전트 JSlave가 도착하면 요구 정보를 Brokering에게 메시지를 통하여 넘겨주고 대기열로 들어간다.



(그림 13) Brokering 구성 요소들간의 관계

사이버장터를 수행 관점에 보면, 우선 관리자가 사이버장터 프로그램을 시작하면 BA 클래스의 on Creation()이 불리고 이 함수 내에서 (그림 14)와 같은 사이버장터 화면을 디스플레이 한다. 사이버장터 화면은 상태 창, 구매 물품 목록, 구매자 대기소, 판매 물품 목록, 판매자 대기소 등으로 구성된다. 이동 에이전트가 사이버장터에 도착한 다음, REGITOBTA 메시지를 통하여 사이버장터에 등록하면 상태 창에 도착한 이동 에이전트의 정보를 기록하고, 새로운 스레드를 생성하여 주어진



(그림 14) 사이버장터 화면

작업을 수행한다. 각 쓰레드에서 작업 수행이 완료되면 BUYRESULTTOSLAVE, SELLRESULTTOSLAVE, BUYERTOSLAVE, SELLERTOSLAVE 등의 메시지를 통하여 대기중인 이동 에이전트에게 수행 결과를 알린다. 이동 에이전트는 수행 결과를 가지고 이동 서버로 이동한다.

## 7. 결론 및 향후연구

본 논문에서는 이동 컴퓨팅 서비스를 효율적으로 지원하기 위하여 부분 연결 관리자의 구조를 제안하고 이를 설계 및 구현하고 강터 서비스에 적용하였다. 이동 컴퓨팅 환경에서 발생하는 네트워크 단절에 의한 부분 연결 문제를 해결하기 위하여 이동 서버에 이동 에이전트 기술을 이용한 부분 연결 관리자가 추가되었다. 부분 연결 관리자는 네트워크 모니터링 모듈, 디렉토리 서비스 모듈과 병렬 계획 모듈의 3개의 고정된 에이전트로 구성하였다. 병렬 계획 모듈은 이동 호스트로부터 이동 에이전트를 받아 사용자 요구 사항을 만족시킬 수 있는 적절한 서버 시스템을 디렉토리 모듈을 이용하여 방문 리스트를 작성한다. 이동 서버는 방문 리스트 내 서버 시스템들로 이동 에이전트를 병렬로 전송하고 지정된 제한 시간 내에 도착한 이동 에이전트들로부터만 수집된 결과를 분석하여 이를 이동 호스트에게 전송하도록 구현하였다. 이 때, 네트워크 모니터링 모듈을 이용하여 이동 서버와 이동 호스트 사이의 연결 상태를 감시하여 연결 여부에 따라 이동 에이전트를 전송하거나 비활성화시킨 후 디스크 내 각 이동 호스트 별 대기열에 저장하였다. 이동 서버의 이동 호스트 사이의 연결 여부는 서비스를 요구한 이동 호스트들에 대해서만 주기적인 폴링을 이용하여 모니터링 하였다. 한편, 이동 호스트가 재연결 요청을 하는 경우에는 이동 호스트가 새로운 IP 주소를 함께 부분 연결 관리자로 보내도록 하였다. 두 방법에 의해 단절된 이동 호스트와 재연결된 경우에는 병렬 계획 모듈이 이동 호스트별 대기열에 존재하는 비활성화된 이동 에이전트들을 활성화시켜 이동 호스트로 수집된 결과를 반환할 수 있도록 하여 부분 연결 문제를 해결하였다.

제안된 부분 연결 관리자를 검증하기 위하여 시골 장터 시스템에 적용하여 구현하였다. 이를 위하여 이동 호스트에는 MAS와 사용자 인터페이스인 JangTer를 장착했고, 이동 서버에는 MAS와 부분 연결 관리자, 서버 시스템에는 MAS와 간단한 거래 정보 시물레이션을 제공하는 CyberMarket을 장착하였다. 시골 장

터에서 물품을 거래하는 것을 시물레이션 하는 장터 서비스를 통하여 이동 컴퓨팅 환경에 이동 에이전트 기술이 적합하며, 부분 연결 문제 등의 이동 컴퓨팅 환경에서 발생할 수 있는 문제점을 극복할 수 있음을 알았다. 향후에는 이동 에이전트의 병렬 탐색과 순회 탐색의 동적 방식 결정 문제, 그리고 웹 서버 데이터를 효율적으로 분석하여 필요한 데이터만을 필터링 하는 기법에 대한 연구가 필요하다.

## 참 고 문 헌

- [1] K. Brown and S. Singh, "A Network Architecture for Mobile Computing," *IEEE INFOCOM'96*, pp 1388-1396, Mar. 1996.
- [2] R. Gray, D. Kotz, S. Nog, D. Rus and G. Cybenko, "Mobile Agents for Mobile Computing," *Proc. of the 2nd Aizu Int'l Symposium on Parallel Algorithms/Architectures Synthesis (pAs97)*, Fukushima, Japan, p17, Mar. 1997.
- [3] D. B. Lange and M. Oshima, Programming and Deploying Mobile Agents with Aglets, *Addison-Wesley*, ISBN : 0-201-32582-9, 1998
- [4] J. White, "Mobile Agents White Paper," General Magic White paper, <http://www.genmagic.com/agents/Whitepaper/whitepaper.html>, p.28, 1996.
- [5] B. Brewington, R. Gray, and K. Moizumi, "Mobile Agents in Distributed Information Retrieval," *Intelligent Information Agents*, Springer Verlag, <http://agent.cs.dartmouth.edu/papers/>, 1999.
- [6] M. Oshima, G. Karjoth and K. Ono, Aglets Specification 1.1 Draft. <http://www.tri.ibm.co.jp/aglets/spec1.1.html>. Sep. 1998
- [7] G. H. Cho, "Location and Routing Optimization Protocols Supporting Internet Host Mobility," *Ph.D. dissertation*, Univ of Newcastle, UK., Dec. 1995
- [8] C. E. Perkins and D. B. Johnson, "Mobility Support in Ipv6," *Proc. of 2nd Int'l Conf. on Mobile Computing and Networking (Mobicom'96)*, 1996.
- [9] B. Marsh, F. Douglass and R. Caceres, "System Issues in Mobile Computing," *Matsushita Lab., Technical Report MITL-TR-50-93*, Feb. 1993.
- [10] T. Lmielinski and B. R. Badrinath, "Mobile Wireless Computing : Solutions and Challenges in Data Management," *Communications of the ACM*, Vol.37, No. 10, Oct. 1994





### 김 평 중

e-mail : pjkim@ccc.ac.kr

1985년 충남대학교 계산통계학과  
(학사)

1995년 KAIST 전산학과  
(공학석사)

1995년 전자계산기조직응용기술사

2000년 충남대학교 컴퓨터과학과(이학박사)

1987년~1988년 포항종합제철(주)(전산기술직)

1988년~1998년 한국전자통신연구원(선임연구원)

1998년~현재 도림중북과대학 컴퓨터정보과(조교수)

1998년~현재 한국정보처리학회 학회지 편집위원

관심분야 : 이동에이전트, 전자상거래, 컴퓨터네트워크,  
분산멀티미디어



### 진 성 일

e-mail : sijin@cs chungnam.ac.kr

1978년 서울대학교 계산통계학과  
졸업(학사)

1980년 한국과학기술원 전산학과  
졸업(석사)

1994년 한국과학기술원 전산학과  
졸업(박사)

1983년~1989년 충남대학교 전산학과 조교수

1987년~1989년 Northwestern대학 객원 교수

1989년~1994년 충남대학교 전산학과 부교수

1994년~현재 충남대학교 컴퓨터과학과 교수

1996년~현재 충남대학교 소프트웨어연구센터 소장

관심분야 : 데이터베이스, 멀티미디어, 성능분석, CALS



### 윤 석 환

e-mail : yoonsh@iita.re.kr

1982년 아주대학교 산업공학과  
졸업(공학사)

1984년 건국대학교 산업공학과  
(공학석사)

1992년 품질관리 기술사 자격획득

1996년 아주대학교 산업공학과(박사)

1986년~1997년 한국전자통신연구원 책임연구원

1998년~현재 정보통신연구진흥원 책임연구원  
(용지 1팀장)

1998년~현재 한국정보처리학회 이사 겸 학회지 편집  
위원장

관심분야 : 분산처리, 소프트웨어 공학, 품질보증, 개발  
체계, ERP