

碩士學位論文

CORBA 기반의 Mobile Agent 기술을 이용한 네트워크 관리



濟州大學校 大學院

제주대학교 중앙도서관
情報工學科
JEJU NATIONAL UNIVERSITY LIBRARY

吳 亮 訓

1999年 12月

CORBA 기반의 Mobile Agent 기술을 이용한 네트워크 관리

指導教授 宋 旺 徹

吳 亮 訓

이 論文을 工學 碩士學位 論文으로 提出함



吳亮訓의 工學 碩士學位 論文을 認准함

審査委員長

안 기 증



委 員

송 왕 철



委 員

이 동 희



濟州大學校 大學院

1999年 12月

CORBA based Network Management using Mobile Agent Technology

YangHoon Oh

(Supervised by professor WangCheol Song)



제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING**

**DEPARTMENT OF INFORMATION ENGINEERING
GRADUATE SCHOOL
CHEJU NATIONAL UNIVERSITY**

1999. 12.

목 차

SUMMARY	1
I. 서 론	1
II. 분산 네트워크 관리	4
1. 네트워크 관리의 개요	4
2. SNMP 기반의 네트워크 관리	5
3. 분산 네트워크 관리	7
4. 분산 객체 관리 기술	8
5. Mobile Agent 기술	10
III. 네트워크 관리 시스템 설계	18
1. 코드 이동성을 이용한 네트워크 관리	18
2. 관리 시스템 요구사항	20
3. CORBA 기반의 Mobile Agent를 이용한 네트워크 관리	22
IV. 시스템 구현 및 검증	25
1. 프로토타입 시스템 구현 환경	25
2. Mobile Agent의 생성	26
3. 네트워크 관리 시스템 동작 과정	27
4. 네트워크 관리 정보	29
5. 시스템 검증 및 결과	30
V. 결 론	34
참고문헌	35

Summary

Network management is needed to control and optimize the operation of network elements. In general, a centralized network management architecture is used to manage a network. However, this type of architecture has caused many problems in network extension, real time monitoring, performance, etc. In order to overcome its weaknesses, studies on network management technology have been introduced. One of the solutions to the problems of the centralized network management architecture is mobile agent technology, which has been highlighted these days in computer-related fields.

The mobile agent is an itinerary software program that autonomously runs on behalf of its operator in a network environment. It has been widely applied to various kinds of computer engineering. In this thesis, a network management architecture using the mobile agent technology in a distributed network environment is proposed.

After being created by a managing system, the mobile agent loads a proxy, that is, a distributed object that acts as a substitute for an SNMP agent. In order to get the management information that a manager requires, the mobile agent locates a remote system and operates on it. Only the concise results from the operation are sent to the managing system. The proposed architecture is based on CORBA which provides several transparent characteristics in the distributed environment. In addition, the proxy object that represents the SNMP agent enables the mobile agent and the SNMP agent to converse with each other. In addition, this architecture provides an efficient network management method by which it can carry out an automated management function and make it easy to add a new function and modify the existing functions.

I. 서 론

최근 컴퓨터 하드웨어 기술이 비약적인 발전과 정보 네트워크의 저변 확대로 많은 사용자들이 네트워크를 통해 필요한 정보를 얻고 있다. 네트워크 사용자들은 사용중인 서비스들이 안정적이고 지속적으로 유지되길 원하며 더불어 다양한 기능을 갖춘 새로운 서비스들을 요구하고 있다. 이러한 사용자의 요구들을 충족시키기 위해서 관리자들은 네트워크 상에 존재하는 여러 시스템 자원들을 사용하여 서비스들의 효율적인 이용을 돕고있다. 따라서, 네트워크는 더욱더 복잡해지고 이질적인 장비와 운영체제로 상호연동 되어지고 있다.

이런 시점에서 사용자들에게 유익한 서비스들을 지속적으로 제공해주기 위해 하부구조인 네트워크 자체를 관리하기 위한 관리 기능들이 요구되어지고 있다. 관리 기능으로는 초기화 등에 관련된 구성관리 기능, 장비들의 결함 탐지를 위한 장애관리 기능, 네트워크의 지속적인 성능유지를 위한 성능관리 기능, 사용자 관리를 위한 계정관리 기능 그리고 접근제어를 위한 보안관리 기능들이 있다(Morris, 1996. Aiko. 1995.). 그리고, 이러한 관리 기능들을 수행하기 위해 네트워크 장비들 간의 관리 정보 교환을 위한 효과적인 구조를 요구하게 되었다.

표준화된 네트워크 관리체계로는 ISO(International Standard Organization)의 CMIP/CMIS(Common Management Information Protocol/Common Management Information Service)와 IETF(Internet Engineering Task Force)의 SNMP(Simple Network Management Protocol)등이 제안되었으며, 이러한 표준으로 인해 시스템은 구성요소간의 호환성 및 상호연결성을 보장받을 수 있으며 일관된

관리 기능을 수행 할 수 있는 토대를 마련하게 되었다(David, 1997. Morris, 1996. William, 1993. William, 1996.). 그러나 제안된 관리 체계들은 중앙집중형 관리 구조에서 오는 관리 시스템의 성능 저하와 네트워크 확장 등의 문제를 노출하고 있다. 표출된 한계를 극복하기 위한 방법으로 클라이언트/서버 기술을 이용한 분산 관리 기법이 연구되었다(OMG, 1999. 한, 1998). 이러한 연구로서 웹을 이용한 네트워크 관리 기술은 웹 브라우저를 사용하므로 관리자와의 인터페이스가 용이하며 관리 시스템의 위치와 플랫폼에 무관한 관리 인터페이스를 제공한다. 이러한 연구로는 기업망 감시와 관리를 위한 WBEM(Web-Based Enterprise Management)과 SUN사의 Java를 이용한 JMAPI(Java Management Application Interface) 등이 있다. 그러나 웹을 이용한 네트워크 관리는 애플릿의 보안과 지연문제 등 문제점을 나타내고 있다(조, 1998. Jan, 1997).

OMG(Object Management Group)에서는 분산 객체를 관리하기 위한 공통적인 구조로 CORBA(Common Object Request Broker Architecture)를 이용한 관리 구조가 제시된바 있다. CORBA는 분산환경에서 소프트웨어 버스의 역할을 하며 사용자들의 사용하는 소프트웨어, 하드웨어의 플랫폼에 구애받지 않고 필요한 정보를 획득할 수 있게 한다(OMG, 1999. 한, 1998.).

Mobile Agent 기술은 인공지능분야에서 연구되기 시작하여 Agent 기술에 이동성을 부여한 기술로 네트워크 상의 이동성과 자율적인 작업을 가능하게 한다. 이는 기존의 RPC(Remote Procedure Calling) 방식을 탈피하여 실행 코드 자체가 목적지 시스템으로 이동하여 실행되어진다. 따라서 Mobile Agent를 이용한 관리는 관리의 분산, 동적인 관리 기능의 추가, 자동화된 관리 등의 특징을 제공한다(Mistubishi, 1998. MIAMI, 1998. Andrzej, 1998.).

CORBA와 Mobile Agent 기술의 접목은 분산환경에서 네트워크 상

의 위치 투명성을 얻을 수 있으며, 관리의 분산과 효율적인 자원의 사용, 그리고 네트워크 관리에 유연성을 부여할 수 있다. 따라서 이러한 구조는 효과적인 네트워크 관리 모델로 연구되고 있다(Andrzej, 1998. Larry, 1997.).

따라서, 본 논문에서는 효율적인 네트워크 관리를 위해 CORBA를 기반으로 하는 분산환경에서 Mobile Agent를 이용한 관리 구조를 제안하였다.

본 논문의 II장은 분산 네트워크 관리에 대한 전반적인 개념과 최근 각광 받고 있는 Mobile Agent 기술에 관하여 소개하며 III장에서는 CORBA 기반의 Mobile Agent를 이용한 네트워크 관리 구조를 제안하고 IV장에서는 제안된 구조에 의해 네트워크 관리 시스템의 프로토타입의 설계 및 구현 그리고 V장에서 결론을 맺는다.



II. 분산 네트워크 관리

지난 수년간 네트워크는 설비, 서비스 그리고 사용량 등에서 양적 질적으로 괄목할 만한 성장을 했다. 이러한 네트워크를 안정적이고 최적인 상태로 유지하기 위해 네트워크 관리가 중요성을 더해가고 있다. 따라서 ISO, ITU 그리고 IETF 등의 국제 단체에서 네트워크 관리를 위한 서비스, 프로토콜 그리고 관리 구조에 관한 연구를 수행하고 있다 (Morris, 1996. Aiko, 1995. William. 1993.).

1. 네트워크 관리의 개요

네트워크 관리는 네트워크의 동작을 감시하고 변화하는 사용자의 요구를 수용하여 네트워크를 최적화하기 위해 필요하다. 네트워크의 본질적인 기능은 사용자들이 네트워크 상에 위치한 시스템들에 대한 접근, 상호 간의 자료 교환 등에 필요한 기본적인 기능들을 직접적으로 사용자들에게 지원하는 것이다. 네트워크 관리는 장비들의 초기화 기능, 현재 상태에 대한 감시와 수정 기능들을 포함하고 있어야 한다.

네트워크 관리는 다양한 네트워크 시스템의 초기화 등에 관련된 구성 관리, 네트워크 구성 장비의 고장을 검출하고 지역화 시키며 고장을 수리하는 장애 관리, 네트워크의 성능의 지속적인 유지와 향상을 위한 성능관리, 네트워크에 관련된 사용자에 대한 정보 관리를 위해 필요한 계정 관리, 네트워크 장비 및 정보의 적절한 사용을 위한 보안 관리 등으로 분류하고 있다(Morris, 1996. William, 1996. Aiko 1995.).

네트워크 관리를 위한 표준으로 ISO에서 OSI 네트워크 관리 구조로 CMIP/CMIS를 제안하였고, 인터넷 관리를 위해 IAB의 IETF에서는

SNMP를 제안한바 있다(William, 1993, William, 1996.).

2. SNMP 기반의 네트워크 관리

인터넷의 성장은 네트워크 관리 프로토콜을 개발하는데 결정적인 역할을 했다. IETF는 인터넷 관리를 위한 프로토콜로 SNMP를 개발하여 발표하게 되었다. SNMP는 빠르게 전파되어 제조업체 사이의 사실상 표준으로 자리 잡았으며, 성능과 보안 기능을 추가한 SNMPv2가 발표되었고, 현재 SNMPv3에 대한 연구가 진행 중이다(William, 1996. David, 1997. 정, 1996.).

1) SNMP 네트워크 관리 구조

SNMP 네트워크 관리 모델은 Fig. 1과 같은 요소들로 구성되어 있다.

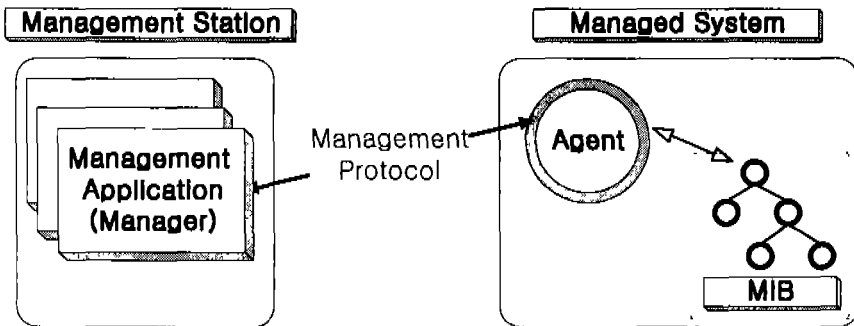


Fig. 1 The SNMP model of a managed network

- 관리 스테이션(Management Station)
- 대리자(Management Agent)

- 관리 정보(Management Information Base)
- 네트워크 관리 프로토콜(Network Management Protocol)

관리 스테이션은 일반적으로 독립적인 장치로 네트워크 관리 업무를 수행하는 관리자와 네트워크 관리 시스템과의 인터페이스 역할을 수행한다. 대리자는 호스트, 브리지, 라우터 그리고 허브와 같은 네트워크 구성 장비에 위치하며 관리 스테이션이 요구하는 정보를 검색하여 응답하는 기능을 한다. 또한 장비의 비정상적인 동작으로 인해 발생하는 이벤트의 보고 기능을 수행한다. 네트워크 상의 실제 자원들은 객체로 추상화되며 이러한 객체들이 집합이 MIB이다. 이러한 MIB는 대리자가 실제 자원에 접근할 수 있는 연결고리를 제공한다. 관리 스테이션과 대리자는 네트워크 관리 프로토콜로 연결되어지며 TCP/IP를 사용하는 네트워크를 관리하기 위해 사용되는 프로토콜이 SNMP이다. SNMP는 비 접속 전송 프로토콜인 UDP(User Datagram Protocol) 상에서 구현되며, 관리 스테이션이 대리자에게 객체의 값을 검색하기 위한 GET, 객체의 값을 지정하기 위한 SET, 그리고 대리자가 관리 스테이션으로 이벤트 통지를 위한 TRAP등의 서비스 프리미티브로 구성된다. Fig. 2는 시스템 간 SNMP를 이용한 동작을 보여주고 있다.

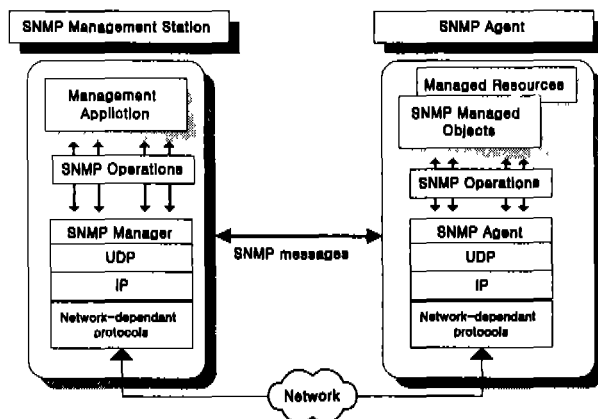


Fig. 2 The role of SNMP

초기 SNMP 개발은 단순화에 초점을 두어 개발되었다. 이로 인해 SNMPv1에서 성능과 보안에 관한 문제점들이 표출되기 시작했다. SNMPv2에서는 성능 향상을 위해 패킷의 최대 크기를 초과하는 경우 응답 가능한 패킷의 범위 내에서 여러 패킷으로 나누어 전송할 수 있는 GetBulk PDU가 제안되었으며, Inform PDU를 이용한 중간 단계 관리자를 도입하여 계층화된 관리를 가능하게 하였다. 또한 보안 기능 강화를 위해 Parties와 Context에 기반한 완벽한 보안 메커니즘을 제시하였다. Parties는 프로토콜 엔티티와 유사하며 다종의 Parties가 독립적인 SNMP 서브 시스템에 의해 활성화되며 Parties는 각각 다른 방법으로 조정된다(Case J, 1990. William, 1996. David, 1997.).

3. 분산 네트워크 관리

초기 시스템 관리는 중앙에 있는 대형 컴퓨터가 모든 서브 시스템을 관리하는 대형 호스트 중심의 중앙 집중형 관리 방법이다. 이 방식은 중앙의 컴퓨터에 모든 작업부하가 걸리는 부하의 집중화 문제를 초래했다(German, 1996. Yemini, 1996.). 또한 모든 서브 시스템의 정보를 중앙의 관리 시스템으로 전송한 후 처리하므로 실시간 관리에 제한이 있었다. 이러한 문제들을 해결하고자 클라이언트/서버 기술을 이용하여 일정하게 작업을 중형 컴퓨터로 분배하여 처리하게 하는 분산 네트워크가 개발되었다. 분산 네트워크 기술은 시스템 자원의 공유가 가능하고 서비스 제공자와 요청자가 분리되며 시스템 규모의 확장과 조정에 유연성을 부여한다. 네트워크 환경이 분산 환경으로 변화됨에 따라 네트워크 관리 구조 또한 기존의 중앙 집중형 네트워크 관리구조에서 분산 네트워크 관리로 변화하였다. 분산 네트워크 관리에서는 중앙의 관리 시스템을 대신하여 관리와 서비스 기능을 동시에 지닌 스테이션들을 통해

관리 기능들을 수행하게 한다. 이러한 구조는 기존 구조에서의 중앙의 관리 시스템 지니는 잠재적인 장애점이 문제를 제거할 수 있으며, 지역적으로 산재해 있는 관리정보를 중앙의 관리 스테이션으로 전송하는데 사용되는 네트워크 비용을 감소시킬 수 있다. 따라서 기존의 중앙 집중형 관리 구조에 비해 좀더 효율적인 네트워크 관리를 가능하게 한다. 또한, 분산환경에서 이질적인 시스템들을 지원하며 객체의 위치 투명성과 접근 투명성을 지원하는 CORBA를 미들웨어로 이용하려는 연구와 네트워크 환경에서 이동이 가능한 Agent를 통해 관리 작업을 대신 수행하는 Mobile Agent 기술이 효율적인 네트워크 관리를 위해 연구되어지고 있다.

4. 분산 객체 관리 기술

분산 객체는 분산 처리 기술과 객체 지향 기술을 바탕으로 하여 다양한 플랫폼 상에서 접근이 가능하고, 프로그램 언어에 독립적인 분산 환경에서 동작하는 프로그램이다. 분산 객체를 관리하기 위한 구조는 OMG에서 제안한 CORBA와 객체지향 개념에 컴포넌트 개념을 추가한 MicroSoft의 DCOM(Distributed Component Object Model)등이 있다 (한, 1998).

1) 분산 객체 환경 CORBA

1989년 OMG는 객체지향 기술을 기반으로 이종의 분산 환경에서 응용프로그램간의 통합된 관리를 위한 표준으로 OMA(Object Management Architecture)를 발표하였다. OMA는 응용 프로그램간의 결합과 객체의 생성, 소멸, 저장, 그리고 트랜잭션 기능 등 분산 객체

환경에서 필요한 모든 서비스를 지칭한다. OMA는 분산환경에서 통신을 담당하는 CORBA, 각종 기본 기능들을 정의하는 CORBAServices 그리고 추가적으로 제공되는 기능에 대한 CORBAFacility로 구성된다.

분산환경에서 객체간의 통신을 담당하는 CORBA는 OMA의 핵심으로 서버 스키텔레톤, 클라이언트 스텔레톤, 인터페이스 저장소, 구현 저장소 객체 어댑터, ORB 인터페이스, ORB로 구성되며 서버 구현 객체와 클라이언트간 통신을 할 수 있게 한다(OMG, 1999. Rovert, 1998.). Fig. 3은 OMG의 CORBA의 핵심인 OMA를 개념적으로 보여준다.

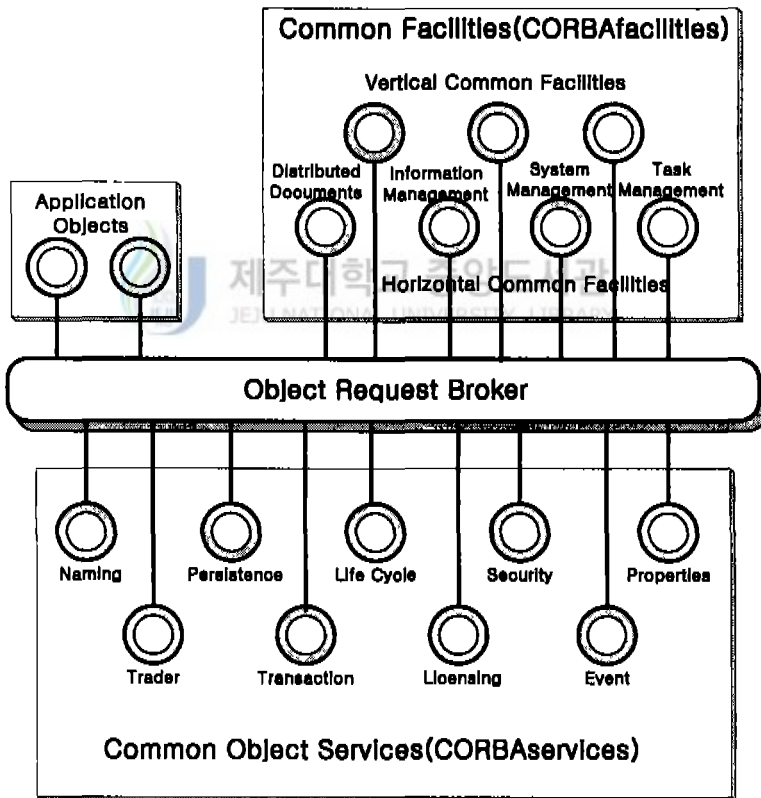


Fig. 3 The OMG Object Management Architecture

CORBA services는 IDL(Interface Definition Language)로 정의된 시스템 레벨 서비스로 ORB의 기능을 보완한다. 생명주기 서비스, 명명 서비스, 사건 전달 서비스, 지속성 서비스, 객체관계 서비스, 객체 외형화 서비스, 트랜잭션 서비스, 동시성 제어 서비스, 라이선싱 서비스, 객체 특성 서비스, 질의 서비스, 트레이딩 서비스, 보안 서비스, 시간 서비스 등의 서비스들이 있다(OMG, 1999. Rovert, 1998.).

CORBA Facilities는 CORBA 시스템에서 제공하는 시스템 기능이 보다 사용자측면에서 제공되는 응용 프로그램 차원의 상호 연동을 위한 서비스들이다. 수평 지원 기능과 수직 지원 기능으로 나뉘지면 전자는 거의 모든 도메인에서 공통적으로 필요한 분산 관리, 정보관리, 시스템 관리, 그리고 작업 관리 기능 등을 지원하며, 후자는 특정 분야에서 필요한 기능들을 지원한다(Rovert, 1998. OMG, 1999.).

OMG CORBA를 이용한 분산 객체 관리는 다음과 같은 특징을 갖는다.

- 정적인 호출과 동적인 호출
- 고급 언어 바인딩
- 자기서술 시스템
- 지역/원격 투명성
- 내장된 보안과 트랜잭션
- 다형성 메시지
- 기존 시스템과의 공존

CORBA에서 제공하는 여러 투명성 특징과 다양한 서비스를 이용하기 위해 분산환경에서 CORBA를 미들웨어로 하는 네트워크 관리가 연구되어지고 있다.(Larry, 1997. 조, 1996. 한, 1998.).

5. Mobile Agent 기술

Agent는 관리자의 작업을 대신 수행하는 독립적인 소프트웨어 프로그램을 의미하며 인공지능 연구에서 비롯되었다. 이러한 Agent는 속성에 따라 여러 종류로 분류하고 있다. Agent를 분류하는 속성으로는 지능성, 이동성, 사회성 등이 있다(Mitsubishi, 1998. Colin, 1995. J.E. White, 1996).

1) Mobile Agent의 개념

Mobile Agent는 Agent의 여러 속성들 중에 이동성 속성을 부각시킨 것으로 네트워크 상을 자유롭게 이동하면서 관리자의 작업을 대신하는 처리하는 대리자를 의미한다. 따라서 Mobile Agent는 인터넷 등과 같은 네트워크 환경에서 사용자를 대신해서 정보를 수집하기 위해 네트워크 사이를 자율적으로 이동하면서 필요한 정보를 수집해서 제공해준다.

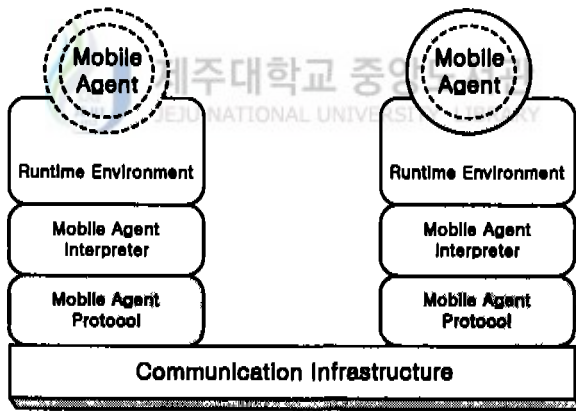


Fig. 4 Model of Mobile Agent system

Fig. 4는 개념적인 Mobile Agent 시스템 모델을 보여 주고 있다. 이러한 Mobile Agent 시스템을 구현하기 위해서는 실제로 Mobile Agent의 기능을 구현할 수 있는 프로그램 언어, 이를 실행 할 수 있는 인터프리터, 그리고 Mobile Agent들 사이의 통신을 지원하기 위한 프

로토콜이 필요하다.

Mobile Agent를 구현하는 프로그램 언어는 사용자들이 원하는 기능을 담은 Mobile Agent를 쉽게 구현 할 수 있어야 하며 Java, Perl, TCL, Lisp 그리고 Telescript 등의 스크립트 언어가 고려되었다. 인터프리터는 프로그램화된 Mobile Agent가 시스템의 플랫폼에 관계없이 동작할 수 있도록 지원하며 Mobile Agent 프로토콜은 Mobile Agent들이 하부 네트워크 프로토콜에 상관없이 Mobile Agent 서로 간의 정보교환 및 이동할 수 있는 수단을 제공해야 한다(Jonathan, 1997. Steven, 1997.).

2) Mobile Agent 플랫폼

현재 Mobile Agent 기술을 이용하기 위해 여러 연구 단체와 업계에서 Mobile Agent 시스템을 위한 플랫폼들을 개발하였다.

Java를 기반으로 하는 미츠비시의 Concordia, IBM의 ASDK(Aglets Software Developer Kit), Gernal Magic의 Odyssey, Objectspace Voyager 등과 OMG의 MASIF 표준에 따라 개발된 IKV++의 grasshopper, TCL을 확장한 스크립트 언어인 Agent-TCL을 이용하는 Dartmouth 대학의 D'Agent 등이 개발되었다(Nui maynooth, 1998.).

(1) Concordia

Concordia는 미츠비시에서 연구중인 Mobile Agent 플랫폼으로 이식성과 인터넷상에서의 접근 용이성, 풍부한 개발도구 등의 특성을 갖는 Java 환경에서 Mobile Agent를 생성할 수 있게 한다. 즉 기존의 Java 언어와 표준 API를 사용해서 새로운 서비스를 제공하는 Mobile Agent 코드를 작성할 수 있게 한다.

Concordia는 Fig. 5와 같이 Manager components들로 구성된 Concordia 서버, 통신 하부구조를 제공하는 Agent Manager, 제공된 서비스들을 관리하는 Administrator, 적절한 사용자를 식별하는 Security manager, 네트워크 상에서 Agent들의 상태를 관리하는 Persistence manager, 이벤트를 처리하는 Event manager, 일정관리를 위한 Queue manager, Concordia 네트워크 상의 네이밍 서비스를 위한 Directory manager, 다양한 기종 사이의 인터페이스 역할을 하는 Service Bridge 그리고 Concordia Mobile Agent를 개발하는데 필요한 라이브러리인 Agent tools library로 구성된다.

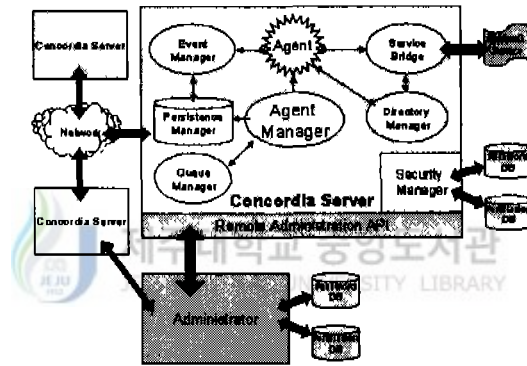


Fig. 5 Concordia architecture

Concordia는 고품질, 고성능 그리고 경제적인 Mobile Agent 응용프로그램들을 개발할 수 있게 하며 사용자의 개인 통신 장비에서 적은 비용과 다양한 이식성을 제공하며 공중망 상에서 Intranet 형식의 안전성을 보장하고 적은 네트워크 대역폭을 통한 작업을 허락한다(Mitsubishi, 1998.).

(2) IBM의 ASDK

ASDK는 IBM사에서 개발한 Java를 기반으로 하는 Mobile Agent

플랫폼으로 클라이언트/서버 기술을 대신하여 전자상거래에 이용을 목적으로 하고 있다. 클라이언트가 API를 통해 원격지 서버와 상호작용을 위한 Client Application Environment, 사용자 인터페이스, 다양한 운영체제 그리고 메시지 전송 서비스들과 바이딩을 위한 Agent Environment, 통신 설비를 통해 Mobile Agent의 송수신을 위한 Messaging sub-system 등을 기본 구조로 한다. ASDK는 J-AAPI(Java Aglet Application Programming Interface), Tahiti, ATCI(Agent Transfer and Communication Interface), ATP(Agent Transfer Protocol) 그리고 Fiji로 구성되어있다. J-AAPI는 Agent와 실행환경사이의 인터페이스를 담당하는 표준으로 Aglets의 생성, message 조작, Aglets의 초기화, 발송, 재시작, 복사, 배치, 활성화 및 비활성화에 관련된 메소드를 제공한다. Tahiti는 응용프로그램으로 Aglets 서버 역할과 사용자와의 인터페이스를 제공한다. ATCI는 AEB의 전송계층 API로 Aglets을 목적지까지 전송하는 기능을 수행한다. ATP는 HTTP 프로토콜 상에서 구현되는 응용계층 프로토콜로 Aglets 간 메시지 전송이나 Mobile Agent에 기반한 분산 시스템에서 Aglets의 전송을 담당한다. 또한, Fiji는 J-AAPI의 확장 라이브러리로 웹 브라우저의 플러그인과 같은 방식으로 동작된다(Colin, 1995., Danny B. 1998).

(3) IKV++ Grasshopper

Grasshopper는 Java를 사용하며, OMG의 MASIF 표준을 따르는 Mobile Agent 플랫폼이다. 클라이언트/서버 구조와 Mobile Agent 기술을 통합한 기술로 전자 상거래나 동적인 정보 검색 등의 분산 응용프로그램 개발을 가능하게 한다. 또한 산업계 표준인 OMG의 MASIF를 따르는 Mobile Agent 플랫폼이다(IKV++, 1999).

Grasshopper는 Agent, Mobile Agent를 위한 실행환경인 Agency,

Agency 내의 논리적인 기능 블록인 Place, 그리고 분산 객체 환경 관리를 위한 Region으로 구성된다. Fig. 6은 분산환경을 지원하는 Grasshoper의 구조를 보여준다.

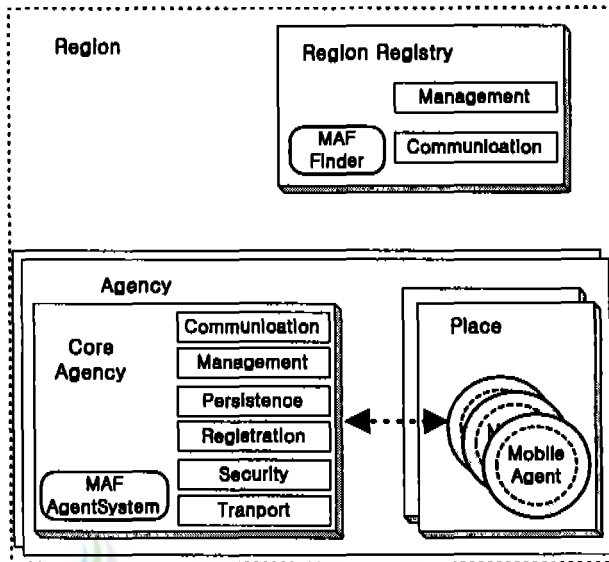


Fig. 6 Structure of Grasshoper distributed agent environment

(4) Objectspace Voyager

Objectspace의 Voyager는 순수 Java 언어를 이용한 Mobile Agent를 위한 Object Request Broker이다. 전통적인 방법인 클라이언트/서버 기술과 향상된 Agent 기술을 이용하여 분산 객체를 지원하므로 네트워크 상에서 응용프로그램을 유연하게 작성 할 수 있다. Voyager는 개발 초기부터 객체나 Agent의 이동성을 지원하도록 개발되었으므로 객체나 Agent들이 한 시스템에서 다른 시스템으로 이동이 가능하며 이동 후에도 계속적인 작업 수행이 가능하다. 따라서 연결이 해제된 이후에도 Mobile Agent는 클라이언트를 대신하여 독립적으로 동작할 수 있다(Graham, 1999.).

Voyager는 정규 Java 문법을 사용하여 원격지 객체나 Mobile Agent와 통신을 할 수 있으며 원격지 실행을 위한 다른 자바 클래스들의 수정이 필요 없으며 상대적으로 작은 크기로 구성이 되어 있으므로 다른 ORB에 비해 빠른 응답시간을 갖는다.

또한 Voyager는 Fig. 7과 같이 융통성 있는 구조로 되어있다.

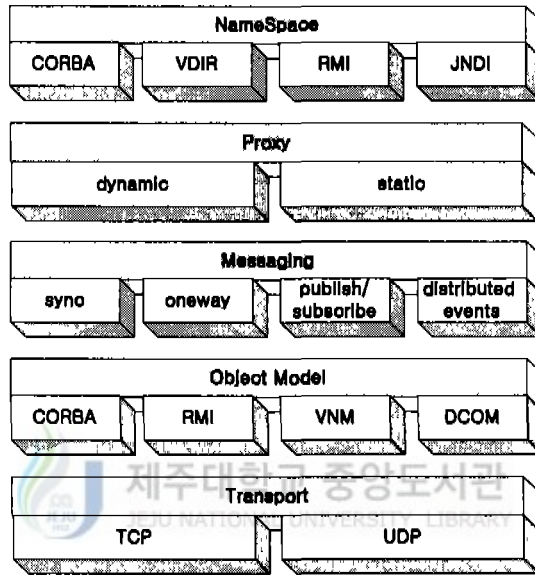


Fig. 7 Universal architecture of Voyager

IIOP(Internet Inter-ORB Protocol)상의 CORBA, ORPC(Object Remote Procedure Call)상의 DCOM(Distributed COM) 그리고 JRMP(Java Remote Method Protocol)상에서 동작하는 RMI(Remote Method Invocation)등의 이질적인 ORB와도 투명하게 동작이 가능하고, SunMicrosoft의 RMI가 일시적인 네이밍 서비스를 지원하는 반면 고유의 네이밍 서비스를 가지고 있으므로 지속적인 네이밍 서비스를 지원한다. 그리고 객체의 pass-by-value 속성, 동적인 클래스 로딩과 향상된 메세징을 지원한다(Graham, 1999.).

3) Mobile Agent 기술의 장점

Mobile Agent 기술은 관리자 입장이나 시스템 관점에서 여러 가지 이득을 제공한다. Mobile Agent 기술은 기존의 RPC(Remote Procedure Calling)를 이용한 클라이언트/서버 기술에 비교하여 향상된 효율과 융통성을 보인다. 전송되어지는 자료가 원시 자료가 아닌 처리되어진 결과만을 전송하므로 네트워크에 걸리는 부하를 줄일 수 있고, 실제 대상과 실시간 지역처리를 통해 네트워크 지연을 극복할 수 있으며, 하부의 프로토콜에 무관하게 동작되어진다. 그리고 비동기적이고 자율적인 실행이 가능할뿐더러 동적으로 실행 환경변화에 적응하여 결함에 강한 견고한 프로그램을 작성할 수 있게 한다. Fig. 8은 RPC 기반의 클라이언트/서버 기술과 Mobile Agent 기술의 실행 특성을 보여주고 있다.

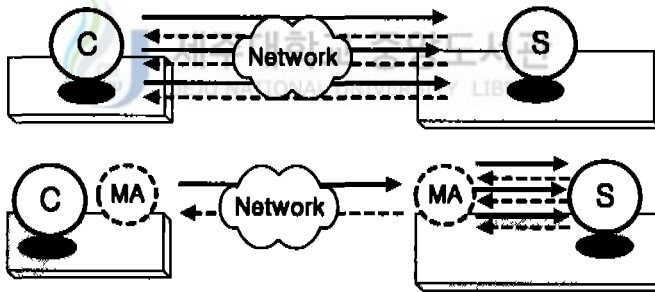


Fig. 8 Client/Server model vs Mobile Agent model

이러한 장점들로 인해 Mobile Agent 기술은 다양한 분야에 적용되어지고 있다. 특히 전자 상거래, 분산 정보 검색, 온라인 쇼핑 그리고 네트워크 관리 분야에서 Mobile Agent 기술을 이용하려는 연구들이 이뤄지고 있다(Danny, 1998. Graham, 1999. Pham, 1998.)

Ⅲ. 네트워크 관리 시스템 설계

1. 코드 이동성을 이용한 네트워크 관리

중앙 집중적인 네트워크 관리 방식을 탈피하고 분산환경에서 클라이언트/서버 기반의 관리 방식에서 오는 여러 문제점을 극복하여 효율을 극대화하기 하기 위한 시도로 실행 코드 자체를 관리 대상 시스템으로 이동하여 실행하려는 시도, 즉 연산의 이주에 관한 시도들이 네트워크 관리분야에 적용되기 시작하였다. (Yemini, 1996. German, 1996. Mario, 1997.)

1) Code-On-Demand(COD) 방식

Code-On-Demand 방식은 실행 시점에 필요한 코드들을 코드 서버에서 다운로드 받고, 동적인 링크과정을 통해 관리 프로그램을 동작하는 방법이다.

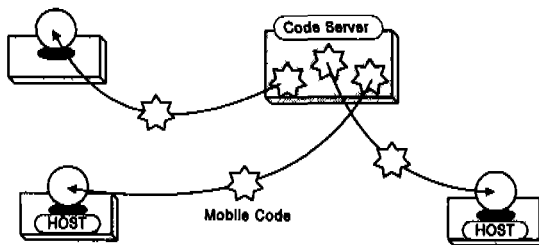


Fig. 9 Code-On-Demand(COD) paradigm

Fig. 9처럼 COD는 이동이 가능한 코드를 제공하는 코드 서버, 응용 프로그램에 적용되어지는 코드 그리고 동적으로 링크해서 사용할 수 있는

호스트들로 구성되며, 이러한 방식은 코드 자체에 이동성을 지닌 Java와 WWW 브라우저를 이용한 관리 기술에 적용되었으나 제한적인 코드의 이동성에 의존한 한계를 나타내고 있다.

2) Remote Evaluation(REV) 방식

Fig. 10의 Remote Evaluation 방식은 RPC의 확장된 방식으로 호스트가 실행에 요구되어지는 코드를 요구할 수 있으며 응용 프로그램의 실행 제어 또한 이동된 시스템에서 이뤄진다. 따라서 이 방식은 제어 정보, 코드 그리고 자료를 대상 시스템으로 이동한 뒤 이동된 지역 시스템의 자원을 이용하여 작업을 수행한다.

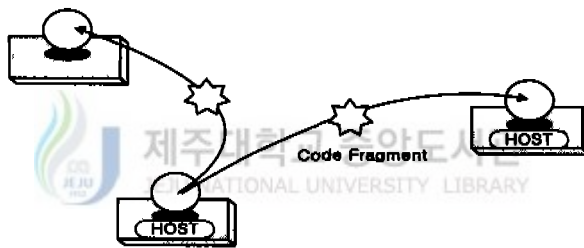


Fig. 10 Remote-Evaluation(REV) paradigm

3) Mobile Agent 방식

Mobile Agent 기술을 이용한 관리는 실행 코드 자체에 이동성과 자율적인 처리 능력을 부여하여 관리 대상 시스템들 사이를 이동하여 작업을 수행하는 방식이다. Fig. 11은 Mobile Agent를 이용한 방식을 보여주고 있다.

Mobile Agent는 자신이 수집한 데이터를 지역 호스트에서 처리를 해서 필요한 정보만 전송하므로 상대적으로 적은 네트워크 대역폭을 점유

한다. 그리고 관리 대상 시스템에서 동작되므로 실시간 관리가 가능하다. Mobile Agent는 이동된 시스템의 자원들을 사용하므로 관리 시스템은 CPU와 메모리 등의 시스템 자원을 보다 효율적으로 사용할 수 있게 된다. 또한 관리자가 해야 할 일들을 Mobile Agent가 대신 처리하므로 관리자가 네트워크 관리 작업에 필요한 시간을 단축시킬 수 있다. 그리고 서로 다른 장비와 환경으로 구성된 네트워크 환경에서 편리하게 응용프로그램을 개발할 수 있게 한다(Andrzej, 1998. German, 1996. Pham, 1998.).

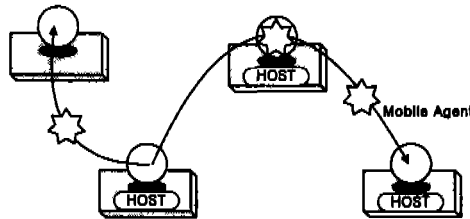


Fig. 11 Mobile Agent paradigm



2. 관리 시스템 요구사항

네트워크를 구성하는 여러 장비들은 자체의 고유한 역할을 수행하기 위해 필요한 기능들로 최적화 되어있다. 따라서 관리 대상 시스템을 위해 필요한 몇 개의 요소들이 요구되어진다. 따라서 본 논문에서 제안된 CORBA 기반의 Mobile Agent를 이용한 관리 시스템은 Mobile Agent 플랫폼, SNMP 대리자를 위한 프록시 그리고 분산 환경 지원을 위한 ORB로 구성된다.

1) Mobile Agent 환경

관리 대상 장비들은 Mobile Agent가 직접적으로 실행할 수 있는 환경을 제공하지 못한다. 따라서 관리 대상 시스템에서 Mobile Agent 환경을 지원해야 한다. Mobile Agent 환경은 관리자의 관리 작업을 대신 수행하는 Mobile Agent의 생성, 중지, 재시작, 삭제, 이동 등 Mobile Agent의 생명주기에 관한 관리 기능을 수행하며 또한 여러 Mobile Agent간의 필요한 정보들을 교환하기 위한 통신기능을 지원한다. 즉 Mobile Agent 환경은 하부의 통신 프로토콜과 무관하게 Mobile Agent가 이동할 수 있는 프로토콜, 이동된 Mobile Agent가 관리 기능을 수행하도록 하는 인터프리터와 런 타임 실행 환경으로 구성된다.

2) SNMP 대리자를 위한 프록시

SNMP는 시스템 관리 프로토콜로 UDP 상에서 ASN.1 문법에 따라 정의되며 SNMP PDU 교환을 통해 관리 기능을 수행 한다. 또한 Mobile Agent는 IDL 타입으로 정의되어 구현된 분산 객체로서 RPC 스타일의 오퍼레이션을 수행한다. 따라서 Mobile Agent는 직접적으로 SNMP 대리자와 오퍼레이션을 할 수 없다. 이를 극복하기 위한 방법으로 SNMP 대리자를 위한 프록시를 두어 이를 해결하였다.

SNMP 대리자를 위한 프록시는 분산 객체로서 관리 스테이션의 관리 대상 시스템에 투명하게 접근 할 수 있는 수단을 제공하고 이동된 Mobile Agent가 시스템 관리를 위해 수행하는 RPC 스타일의 CORBA 오퍼레이션들을 메시징 방식으로 동작하는 SNMP 대리자가 인식할 수 있는 방식으로 중계하는 역할을 수행한다. 즉 SNMP 대리자가 사용하는 GET, SET, TRAP 등의 SNMP 오퍼레이션들을 분산 객체인 Mobile Agent가 수행 할 수 있게 한다.

3) 분산 환경을 위한 미들웨어

분산 환경은 LAN 또는 WAN등으로 구성된 네트워크 환경을 기반으로 원격지에 있는 시스템간의 상호 작업을 통해 자원 및 정보의 공유 등에 필요한 구성요소들의 집합을 의미한다. 즉 네트워크 상에서 상호 동작을 위해 필요한 하부 구조를 제공하는 것이다. 본 논문에서는 분산 환경을 지원하는 미들웨어로서 CORBA를 사용하였다.

Mobile Agent와 SNMP 대리자를 위한 프록시는 분산 객체로서 원격지 서버에서 네트워크에 투명하게 접근을 할 수 있으며 다양한 하드웨어 플랫폼 상에서도 동작이 가능하며 또한 상속이나 다형성 같은 객체지향 기술의 장점들을 제공해 준다. 따라서 분산 환경을 지원하는 CORBA는 이기종 시스템들을 통합 관리 할 수 있는 방법을 제공한다. 관리 스테이션은 CORBA에서 제공하는 네이밍 서비스 등과 같은 서비스를 통해 관리 대상 시스템 접근이 용이하게 한다.



3. CORBA 기반의 Mobile Agent를 이용한 네트워크 관리

분산객체를 지원하는 CORBA환경에서 Mobile Agent를 이용한 네트워크 관리는 CORBA에서 제공하는 네이밍 서비스, 생명 주기 서비스, 보안 서비스 등 기존 서비스들을 사용할 수 있으며 분산 환경에서의 위치 투명성 및 접근 투명성을 보장받을 수 있다. 이러한 장점으로 인해 CORBA를 기반으로 하고 Mobile Agent를 이용한 네트워크 관리가 활발히 연구되어지고 있다.

본 논문에서는 CORBA 환경에서 Mobile Agent 기술을 이용한 네트워크 관리 구조를 제안하였다. 분산 환경을 지원하는 CORBA 기반으로

하고 Mobile Agent 기술을 이용한 네트워크 관리 시스템은 관리자 역할을 수행하는 관리 스테이션과 관리 대상 시스템들로 구성된다. 관리 스테이션은 네트워크 관리자가 관리를 위한 여러 기능들을 사용할 수 있게 하는 사용자와의 인터페이스를 위한 관리 응용 프로그램과 Mobile Agent 실행을 위한 Mobile Agent 인터프리터, Mobile Agent의 생성, 삭제 등 라이프사이클과 관리 대상시스템으로 Mobile Agent를 이동시키기 위한 Mobile Agent 플랫폼, 분산 객체 기술을 지원하기 위한 ORB 그리고 시스템 관리를 위한 SNMP 대리자와 오퍼레이션을 수행하기 위한 프록시로 구성된다.

또한, 관리 대상 시스템 Mobile Agent 플랫폼, mobile Agent 인터프리터, ORB 그리고 SNMP 대리자로 구성된다.

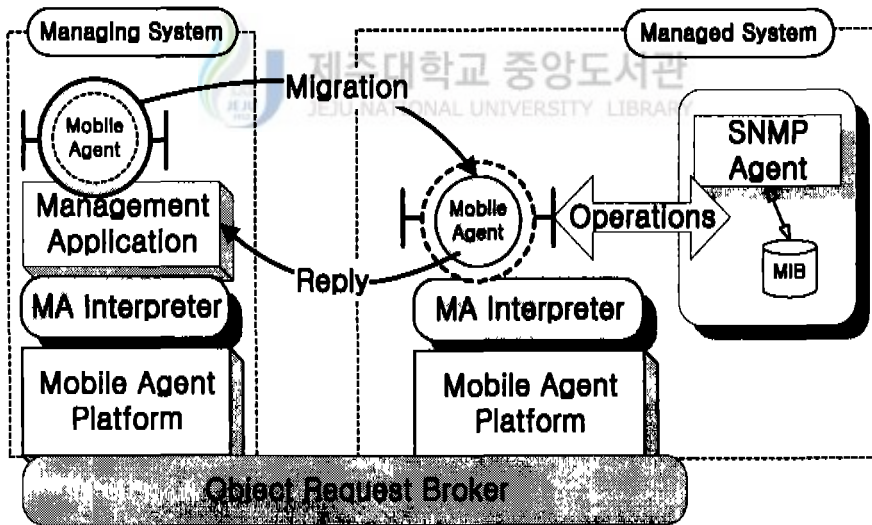


Fig. 12 Architecture of network management system

Fig. 12의 CORBA 기반의 Mobile Agent를 이용한 네트워크 관리 시스템 구조에서는 Mobile Agent의 생성을 위해 코드 이동성 특성을 갖는 Java가 사용되었으며, Mobile Agent의 이동과 Mobile Agent 사이의 통신을 위해 Voyager Mobile Agent 플랫폼이 이용되었으며 시스템 관리를 지원하는 SNMP 대리자와의 통신을 위해 CORBA 객체화된 SNMP 대리자 프록시를 두었다. CORBA 객체화된 프록시를 통해 Mobile Agent는 분산환경에서의 여러 서비스와 투명성 특성을 이용할 수 있도록 하였다.

제안된 관리 시스템 구조에서 관리 시스템은 CORBA 네이밍 서비스를 이용하여 관리 시스템을 찾고 이후 관리자의 작업을 대신 수행하는 Mobile Agent를 ORB를 통해 관리 대상 시스템으로 전송한다. 관리 대상으로 이동된 Mobile Agent는 Java 가상 머신에 의해 실행되며 CORBA 객체화된 SNMP 대리자 프록시와 오퍼레이션을 통해 관리 작업을 수행한다. Mobile Agent는 얻어진 결과를 관리 시스템으로 반환한다.

본 논문에서 제안된 CORBA 기반의 Mobile Agent 기술을 이용한 네트워크 관리 구조는 분산 객체의 네트워크 투명성과 Mobile Agent의 자율적인 이동과 작업 수행 특성으로 인해 효율적인 네트워크 관리를 가능하게 한다.

IV. 시스템 설계 및 구현

기존의 네트워크관리 시스템들은 중앙의 관리 시스템이 모든 네트워크 장비를 관리하는 중앙 집중형 관리 구조를 사용하였거나 클라이언트/서버 기술에 기반을 둔 분산 관리 구조를 사용하고 있다. 본 논문에서는 중앙 집중형 관리 구조에서 오는 단점을 보완하고 최근 각광 받고 있는 분산 객체 기술인 CORBA와 Mobile Agent 기술을 접목한 네트워크 관리 구조를 제안하였다.

1. 프로토타입 시스템 구현 환경

본 논문에서 제안된 CORBA 기반의 Mobile Agent 기술을 이용한 네트워크 관리 시스템은 교내 이더넷 상에 존재하는 호스트 장비들과 일반 PC들을 실험 대상으로 이용하였다.

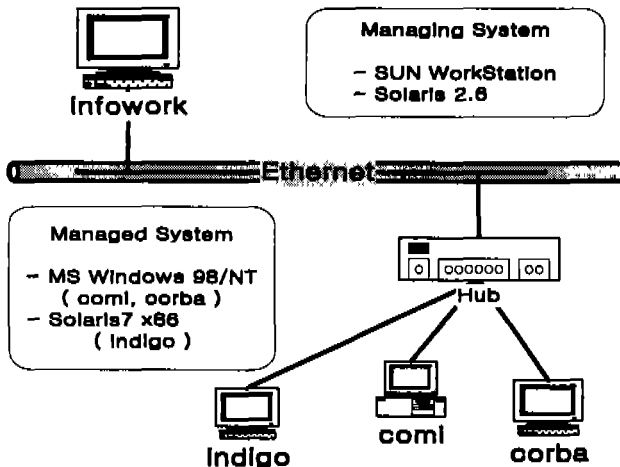


Fig. 13 Configuration of testing system environment

Fig. 13은 프로토타입 시스템 구현환경을 나타내고 있다.

관리 스테이션은 호스트 장비 중에 워크스테이션으로 SUN Ultra-2 플랫폼으로 Solaris 2.6을 운영체제로 하고 있으며 다양한 클라이언트 환경을 위해 Intel x86상의 Solaris7 운영체제를 사용하는 호스트와 Microsoft Windows98/NT를 운영체제로 하고 있는 일반 PC들이 사용되었다.

제안된 시스템 구현을 위해 다음과 같은 소프트웨어들이 사용되었다. Mobile 코드 생성을 위해 SunMicrosystems의 Java 1.2, Mobile Agent를 위한 플랫폼으로 Objectspace의 Voyager ORB 3.1.2, SNMP 대리자로 UCD-SNMP 4.0.1, SNMP 대리자와 오퍼레이션을 하기 위해 AdventNet SNMPv1 클래스가 사용되었다(AdventNet, 1998. Graham, 1999.).

2. Mobile Agent의 생성



제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

Mobile Agent를 생성하기 위한 작업으로 우선 IDL(Interface Definition Language)로 Mobile Agent를 위한 인터페이스와 Mobile Agent가 수행해야하는 관리 작업들을 위한 인터페이스를 작성한다. 생성된 인터페이스를 Voyager JavaToIDL 컴파일러(Cgen)를 통해 Java 언어로 매핑한다. 사용자가 인터페이스를 구현하는 구현 파일을 작성하고 이를 자바 컴파일러로 컴파일 과정을 통하여 Mobile Agent를 생성한다. Fig. 14는 Mobile Agent 생성을 위한 과정이다.

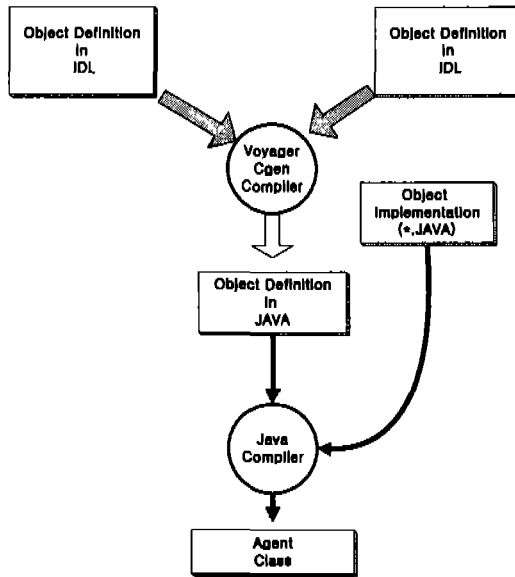


Fig. 14 Creating Mobile Agent

3. 네트워크 관리 시스템 동작 과정



본 논문에서 구현된 네트워크 관리 시스템은 Fig 15와 같은 절차로 동작되어진다.

Voyager ORB는 분산환경을 지원하는 Mobile Agent 플랫폼이며 SNMP 프록시는 분산객체로서 SNMP 대리자와 Mobile Agent 사이의 SNMP 오퍼레이션을 가능하게 한다.

Mobile Agent는 다른 시스템으로 이동되어지며 이후 관리 대상 시스템에서 시스템 관리 프로토콜을 이용하는 SNMP 대리자와의 오퍼레이션을 통하여 관리자의 관리 기능을 대신 수행하며 얻어진 결과에서 관리자가 요구하는 정보만 관리 시스템으로 보내주게 된다.

구현된 네트워크 관리 시스템의 동작 과정은 다음과 같은 절차를 따른다.

- ①. Mobile Agent 생성
- ②. Voyager ORB에 등록
- ③. Mobile Agent가 목적지 시스템으로 이동
- ④. Mobile Agent 실행
- ⑤. SNMP 프록시 로딩
- ⑥. SNMP 대리자 프록시를 통한 관리 정보 획득
- ⑦. 결과를 관리 시스템으로 반환

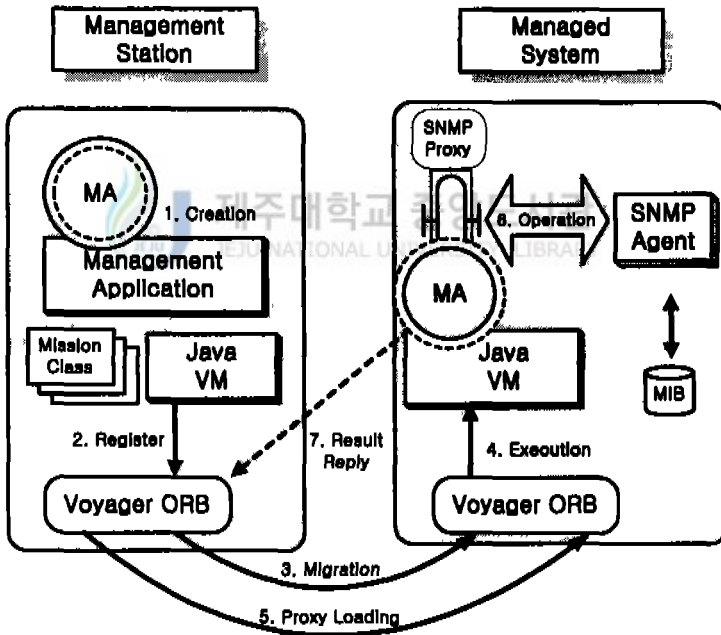


Fig. 15 Procedure of network management system

4. 네트워크 관리 정보

Mobile Agent는 네트워크 관리자를 대신하여 관리에 필요한 정보를 관리 대상 시스템으로부터 얻어오게 된다. 이때 사용되는 관리정보들은 RFC(Request for Comments) 1213에 정의된 MIB-II와 RFC 1514의 HOST RESOURCES MIB를 사용하였다(Case.J, 1990, P. Grillo, 1993). Mobile Agent는 SNMP 대리자를 CORBA 객체화한 프록시를 이용하여 SNMP 오퍼레이션들을 통해 MIB 내에 존재하는 여러 속성값들을 접근할 수 있다.

Fig. 16은 네트워크 관리 시스템에서 사용되어지는 MIB 트리를 보여주고 있다.

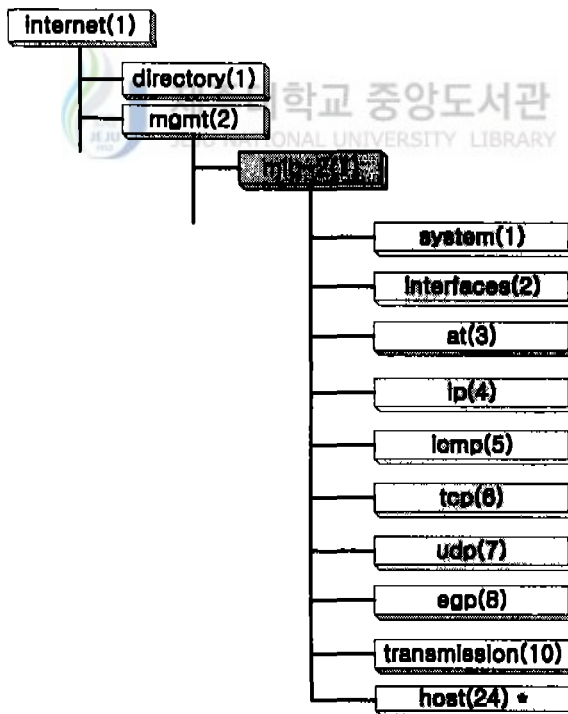


Fig. 16 MIB tree

사용된 그룹들에 대한 설명을 Table 1과 같다.

Table 1 MIB groups

MIB 그룹	설 명
system	관리 대상 시스템의 일반적인 정보 제공
interfaces	물리적인 인터페이스에 대한 총괄적인 정보 제공
ip	IP 계층 동작에 관한 트래픽 정보 제공
icmp	ICMP의 피드백 정보 제공
tcp	TCP segment에 관한 정보 제공
udp	UDP datagram에 대한 정보 제공
host	호스트 컴퓨터 자원에 관한 정보 제공

Mobile Agent는 system 그룹 내에 정의된 MIB 값을 통해 관리 대상을 식별해 낼 수 있으며, interfaces 그룹은 전반적인 네트워크 인터페이스의 상태를 확인할 수 있다. 그리고 ip, icmp, tcp, udp 그룹은 인터넷을 위해 사용되어지는 프로토콜들의 동작을 알아 볼 수 있다. 또한 추가된 host 그룹을 통해 시스템 기억장치, 프로세스 등 시스템 자원의 사용 정보를 얻어낼 수 있다.

5. 시스템 검증 및 결과

본 논문에서는 SNMP 대리자를 CORBA 객체화한 프록시를 찾기 위해 OMG에서 제공하는 COSNaming 서비스 대신에 Voyager에서 지원하는 Voyager 네이밍 서비스를 이용하였다.

본 논문에서 Mobile Agent가 프록시 Object를 통해 MIB-II내의 system, interfaces, ip, icmp, tcp, udp 등의 그룹에서 시스템 관리에 필요한 여러 값들을 추출하고 이를 관리자가 원하는 정보들만 처리해서 관리 스테이션에 알려 주게 된다.

Fig. 17은 관리 스테이션이 Mobile Agent를 통하여 현재 네트워크 인터페이스에서의 입출력되는 상태를 알아보는 화면이다.

```

Get System Information
System Name      Indigo
System Description SunOS Indigo 5.7 Generic_106542-02 186pc
Contact         baram@cheju.ac.kr

Get interfaces information

===== General Interfaces Information =====
- Interfaces name      lprb0
- Interfaces type      1
- Largest protocol Unit 1500 Octets
- Interfaces capacity  10000000
=====

== Incoming Error Report ==
Incoming Error Rate  .0%
Outgoing Error Rate  .0%
Stop Mobile worker

```

Fig. 17 Snapshot of managing system

이를 위해 MIB-II 내에 있는 system 과 interfaces 그룹에 있는 정보들을 획득하여 관리 대상 시스템을 식별하고 네트워크 인터페이스 상태 정보를 관리 시스템에 전달한다.

Fig. 18은 관리 스테이션이 Mobile Agent를 Solaris7상의 호스트에서 MIB-II 내의 system 그룹을 통해 시스템의 기본 정보를 획득하고 IP, ICMP, TCP, UDP등의 프로토콜 별 상태를 모니터링하는 화면이다. 이를 위해 Mobile Agent는 ip, icmp, tcp, udp 그룹에서 필요한 정보들을 얻어내고 이를 바탕으로 손실 및 폐기 등 에러가 발생한 정보를 추출하고 에러율을 알려 준다.

```

InfoworkK ./RunAgent
Voyager Running at Infowork 8000

At remote worker, home = tcp://indigo:8000
Starting Mobile worker
Get system Information.
System Name : Indigo
System Description Sun SNMP Agent,
Contact : System administrator
Get IP information
IP Receiving Error Rate : 0%
IP Success Receiving Rate 49%
IP Outgoing Discards Rate 0%
Get ICMP information
ICMP Incoming Error Rate : 21%
ICMP Outgoing Error Rate 50%
Get TCP Information ...
TCP Incoming Error Rate 0%
TCP Retransmission Rate : 0%
Get UDP Information .....
UDP Incoming Error Rate : 0%
Stop Mobile worker

```

Fig. 18 Snapshot of managing system

Fig. 19는 Mobile Agent를 이용하여 기존의 MIB에 추가된 Host Resources MIB를 이용하여 관리 대상 시스템으로부터 현재 하드디스크 기억장치의 사용 현황을 알아보도록 하였다.

```

At remote worker, home = tcp //indigo 8000
Starting Mobile worker
Get System Information

System Name indigo
System Description SunOS indigo 5.7 Generic_106542-02 i86pc
Contact barani@chejul.cheju.ac.kr

Get Host Disk Information

Searching Mount point
Allocated Partition
Used Storages
==== System Disk Usages ====
/proc 0 % Used Total 0 KBytes 0 KBytes used
/ 11 % Used Total 261615 KBytes 31299 KBytes used
/usr 55 % Used Total 481283 KBytes 265660 KBytes used
/dev/fd 0 % Used Total 0 KBytes 0 KBytes used
/var 21 % Used Total 123567 KBytes 26238 KBytes used
/home 10 % Used Total 4072102 KBytes 445962 KBytes used
/opt 9 % Used Total 675056 KBytes 66001 KBytes used
/usr/local 22 % Used Total 675056 KBytes 155012 KBytes used
/usr/openwin 48 % Used Total 481269 KBytes 208826 KBytes used
/tmp 0 % Used Total 78705 KBytes 2 KBytes used
/vari 0 % Used Total 0 KBytes 0 KBytes used
Stop Mobile worker

```

Fig. 19 Snapshot of managing system

Host Resource MIB(RFC 1514)는 기존에 제공되는 SNMP에 포함되어 있지 않으므로 이를 위해 별도의 Host Resource MIB를 대상 시스템에 추가로 설치하였다.



V. 결 론

본 논문은 기존의 중앙집중형 관리구조에서 관리 스테이션의 성능 저하문제와 네트워크 확장성 문제를 해결하고자 CORBA를 이용한 분산 환경에서 Mobile Agent 기술을 적용한 효율적인 네트워크 관리를 제안하였다. 따라서 본 논문에서는 관리자의 노동 집약적인 네트워크 관리 작업들을 Mobile Agent가 대신하여 수행하고 관리 작업의 수정만으로 원하는 결과를 얻어낼 수 있으므로 동적인 관리작업이 가능함을 알 수 있었다. 분산 객체 기술인 CORBA를 사용하므로 관리 대상 시스템에 대한 네트워크 상의 위치 투명성을 확보할 수 있다. 그런데, SNMP 프로토콜은 시스템 관리 목적으로 만들어진 프로토콜이므로 CORBA에서 제공하는 Object 서비스를 이용하기 위해 CORBA 객체화된 프록시 객체를 만들어 분산 환경에서 관리 시스템과 통신이 가능하도록 하였다. Mobile Agent는 관리 대상 시스템에서 인터페이스의 상태 정보와 각 프로토콜별 상태 정보 그리고 현재 기억장치의 사용 현황을 알아보는 관리 작업을 수행하게 하였다. Mobile Agent에서 관리 기능을 담고 있는 관리 작업의 수정을 통해 다양한 관리 기능을 수행하는 시스템을 통해 분산환경에서 효율적인 네트워크 관리가 가능함을 볼 수 있었다.

앞으로, OMG에서 표준화가 진행중인 MASIF(Mobile Agent System Interoperability Facilities)에 부합하는 Mobile Agent 인터페이스 개발을 통한 기존의 CORBA 서비스와의 연동을 위한 노력이 요구되어진다.

참고문헌

Graham Glass, 1999, "Voyager ORB Guide 3.0, 3.1.1, 3.1.2"

Graham Glass, CTO ObjectSpace, 1999 "Reducing development effort using the ObjectSpace Voyager ORB"

Graham Glass, CTO ObjectSpace, 1999 "The ObjectSpace Voyager universal ORB"

Nui Maynooth, 1998, " A review of State of the Art int Mobile Agent System."

Danny B. Lange, Mitsuru Oshima, 1998, 「 Programming and Deploying Java Mobile Agents with Aglets 」, Addison Wesley

David Perkins, Evan McGinnis, 1997, 「 Understanding SNMP MIBs 」, prentice Hall

Larry T. Chen, Tatsuya Suda, 1997, "Designing Mobile Computing Systems Using Distributed Objects", IEEE Communication Magazine

Rovert Orfail, Dan Harkey, 1998, 「 Client/Server Programming with JAVA and CORBA 2nd Edition 」, Joh Wieg & Sons. Inc.,

Mario Baldi, Silvano Gai, and Gian Pietro Picco, 1997, "Exploiting Code Mobility in Decentralized and Flexible Network Management"

Morris Sloman, 1996, 「 Network and distributed systems management 」 , Addison Wesley

MIAMI consortium, 1998, "Mobile Intelligent Agents for Managing the Information Infrastructure"

Mitsubishi Electric ITA Horizon Systems Lab., 1998, " Mobile Agent Computing-A White Paper"

Steven Versteeg, 1997, " Language for Mobile Agents"

**IKV++, 1999, "A CORBA environment supporting Mobile Agent",
IKV++ GmbH**

Aiko Pras, 1995, " Network management architecture"

Andrzej Bieszczad, Bell Lab., 1998 " Mobile Agent for Network Management"

AdventNet, 1998, "AdventNet SNMP API Tutorial"

William Stallings, 1996 「 SNMP SNMPv2 and RMON 」 , Addison Wesley

William Stallings, 1993 「 SNMP SNMPv2 and CMIP 」 , Addison
Wesley

OMG, 1997, "Mobile Agent System Interoperability Facilities
Specification"

OMG, 1999, <http://www.omg.org/>

Yechiam Yemini, Sushil da Silva, 1996, "Towards programmable
networks "

K. Rothermel, F. Hohl, N. Radouniklis, 1998, "Mobile Agent
Systems : What is Missing?"

German S. Goldszmidt, 1996, " Distributed Management by
Delegation"

정보통신부, 1996, 「 단순망관리 규약(SNMPv2) 표준 」

조수현, 1998, "CORBA를 이용한 웹 기반의 인터넷 서비스 관리 시스
템의 구현 및 설계"

Jan Vitek, 1997, "New Paradigms for Distributed Programming"

Jonathan Dale, 1997, "A Mobile Agent Architecture for distributed
Information Management"

J. E. White, 1996, " Mobile Agents White Paper "

Case. J, 1990, 「 A Simple Network Management Protocol(SNMP)
RFC1157 」

P. Grillo, 1993 「 Host Resources MIB RFC 1514 」

Colin G. Harrison, David M. Chess, Aaron Kershenbaum, 1995,
"Mobile Agents: Are they a good idea?"

Vu Anh Pham, Ahmed Karmouch, 1998, "Mobile Software Agents
: An Overview"

한국전산원, 1998, "분산 객체 기술 표준 연구"



감사의 글

항상 시작에는 두려움이 따르고, 끝에는 어딘가 모르는 부족함을 느끼곤 합니다.

늘 세심한 배려와 지도로 저의 부족함을 채워주시고 지도해주신 송왕철 지도 교수님께 감사 드립니다. 그리고 결실을 맺기까지 지도편달을 아끼지 않으신 김장형 교수님, 안기중 교수님,곽호영 교수님, 이상준 교수님, 변상용 교수님과 이봉희 교수님께도 감사의 마음을 전합니다.

충고와 조언으로 저의 앞길에 희망을 주신 경호형님, 권호형님 그리고 수많은 늦은 밤을 같이 했던 강석영님과 망 관리 연구실의 여러 식구들에게도 고마운 마음을 전합니다. 그리고, 많은 도움을 주신 수경누나와 외국어 교육관 여러 식구들에게도 감사함을 전합니다.

두 번의 겨울을 같이 보낸 대학원 동기들과 학기동안 룸메이트였던 승범이와 여러 후배들에게도 늘 좋은 일이 함께 하길 기원합니다.

그 동안 늘 바쁘다는 이유로 소홀했던 여러 친구들과 새 식구가 된 매형들 그리고 나의 가족들에게 자그마한 저의 결실로나마 미안한 마음을 대신 합니다.

마지막으로 늘 기쁨을 주었던 나의 사랑스런 박박이와 뽀뽀이 뒤편에서 저의 앞길을 위해 기도하셨던 할머니와 든든한 나의 후원자인 어머니님 그리고 돌아가신 아버지님 영전에 이 논문을 바칩니다.

1999년 겨울 아라굴에서