

JNDI를 이용한 Mobile Agent 기반 망관리 시스템

김정철* · 송왕철*

*제주대학교

Network Management System Based upon Mobile Agent using JNDI

Jung-chul Kim* · Wang-cheol Song*

*Cheju National University

E-mail : kjchul@infowork.cheju.ac.kr

요 약

본 논문에서는 JNDI를 이용한 네이밍 서비스와 디렉토리 서비스를 적용한 Mobile Agent 기반 SNMP 망관리 시스템을 제안한다. Mobile Agent 기반의 망관리 시스템은 기존의 중앙집중식 망관리의 단점들을 해결한다. 또한 분산환경에서 관리자가 직접 대리자를 참조하고 접근하여 관리자의 필요에 따라 Mobile Agent를 파견할 수 있도록 JNDI를 이용하여 네이밍 서비스와 디렉토리 서비스를 지원하는 시스템을 설계 및 구현하였다.

ABSTRACT

This paper supposes the SNMP network management system based upon Mobile Agent using the Naming and Directory Service, which is JNDI(Java Naming Directory Interface) technology. The Mobile Agent network system is expected to provide a solution for the conventional centralized network system. So, We have designed and implemented the network management system based upon Mobile Agent using JNDI

1. 서 론

기존의 중앙집중식의 SNMP(Simple Network Management Protocol)를 사용한 망관리시스템은 관리자가 관리대상의 대리자로부터 SNMP 메시지를 전송하고 전송 받으며 관리자는 대리자로부터 전송 받은 메시지를 분석하여 망관리를 수행하게 된다.^[1] 이러한 중앙집중형 망관리 모델은 간결한 망 구성 및 관리가 가능하고 보안이 용이하며, 망 관리에 적합한 모델이기는 하나, 중앙의 관리자에 모든 관리 기능이 집중되어 있기 때문에 망 확장성의 제한, 실시간 관리 기능의 제한, 서비스의 동적인 추가의 어려움 등의 문제점을 가지게 되었다. 이러한 문제점들을 해결하기 위하여 SNMP표준안에서 계층적 관리모델 등을 제시하기는 하였지만 별다른 효과가 없었다.^[2]

Mobile Agent는 분산환경에서 최근 부각되고 있는 연구분야이며 Mobile Agent 고유의 자율성, 비동기성, 이동성 등은 망관리에 적용하기 적합한 특성으로 Mobile Agent를 망관리에 적용함으로써 중앙집중식의 망관리의 단점들을 해결할 수 있다.^[3]

네이밍 & 디렉토리 서비스는 분산환경에서 객체를 참조하거나 객체의 속성을 참조하여 객체를 찾을 수 있도록 하는 서비스이다.^[4]

본 논문에서는 관리자가 Mobile Agent를 파견하고자 하는 관리대상을 참조할 수 있도록 하기 위하여 JNDI(Java Naming Directory Interface)를 이용하였으며, 관리자가 네이밍 서비스와 디렉토리 서비스를 받아 Mobile Agent를 생성 또는 파견할 수 있도록 하는 시스템을 설계하고 구현하였다.

II. 분산망 관리

분산망관리를 위해 가장 널리 쓰여지는 프로토콜인 SNMP의 중앙집중식 망관리의 단점은 Mobile Agent 기술을 망관리에 접목시킴으로써 많은 부분이 해결될 것으로 보인다.

1. SNMP기반 망관리

TCP/IP 네트워크 환경에서 가장 많이 사용되

고 있는 관리방법인 SNMP 관리시스템은 관리스테이션, 관리대리자, MIB(Management Information Base), 네트워크 관리 프로토콜로 구성되어 있다. 관리스테이션은 데이터 분석 및 장애복구 위한 프로그램, 네트워크의 현황을 관찰할 수 있는 인터페이스, 리모트에 있는 네트워크 요소의 관리 능력, MIB를 통한 획득한 정보의 테이블화 및 이스화가 기본기능이다. 호스트, 브리지, 라우터 및 허브 등에 관리대리자가 구동되어, 긴급사항 또는 관리스테이션의 요구에 응답할 수 있다. 관리대상 객체의 현황을 데이터 변수로 표현한 MIB는 관리스테이션을 위해 관리대리자에서 객체를 액세스하는 접근점으로 MIB값의 변경을 통해 에이전트의 동작대상 및 구성변경을 수행할 수 있다. 네트워크 관리 프로토콜에 의해 관리스테이션 관리대리자사이의 링크를 형성하는 주요 기능은 다음과 같다.

- Get : 관리 스테이션이 관리대리자를 통해 객체의 값을 조회.
- Set : 관리 스테이션이 관리대리자를 통해 객체의 값을 세팅.
- Trap : 에이전트가 특별 이벤트를 관리 스테이션을 보고.

SNMP는 어플리케이션 레벨(Application-Level) 프로토콜로서 UDP(User Datagram Protocol)상에서 동작하도록 구성되어 있다. 그림 1. 에서 보는 것같이 관리스테이션이 GetRequest, GetNextRequest, SetRequest의 유형으로 메시지를 보내면 에이전트의 응답은 GetResponse에 의해 수행된다.

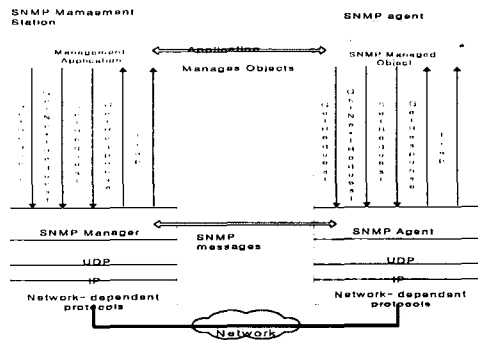


그림 1. SNMP의 역할

2. Mobile Agent 기반 망관리

Agent는 사용자를 대신하여 사용자가 원하는 어떤 일을 수행해주는 프로그램이라고 할 수 있다. 또한 Mobile Agent는 네트워크 환경을 이동하면서 Agent 기능을 수행 할 수 있는 프로그램을 의미한다. 이동 에이전트 시스템은 원격 시스

템의 코드를 호출하는 기존의 클라이언트/서버 개념, 클라이언트의 요청이 있을 때 코드를 보내 주어 실행하는 방식인 Code-on-demand 방식과는 달리 Mobile Agent가 네트워크를 통해 해당시스템으로 전송되어 실행되는 방식이며 자신의 판단에 의해 이동한다. Mobile Agent는 Script 언어로 작성되고 Interpreter로 수행된다는 것이 특징이다.

Mobile Agent가 해당 네트워크 장비(Network Element)로 직접 이동해서 수행되고, 관리정보를 획득하고 원하는 NE(Network Element)로 이동할 수 있기 때문에 중앙집중형 망관리 모델에 비해 다음과 같은 특징을 가진다.

- NMS의 부하 분산 : 이동 에이전트가 직접 NE로 이전 되어 NE에서 오퍼레이션을 하여 그 결과 값을 가지고 오므로 NMS의 부하 감소.
- 확장성 제공 : 네트워크 오버헤드를 감소시켜 망관리 범위를 확대.
- 실시간 관리의 용이 : 여러진단, 긴급복구 등의 실시간의 요구되는 망관리 경우 대처능력의 탁월.
- 서비스 추가의 용이 : 새로운 서비스를 능동적으로 대처.
- 망연동의 유연성 : 서비스의 동적인 추가로 서로 이질적인 타 망과의 연동에 유연.
- 효율적인 망 운용 : 각각의 NE 에 맞는 망관리 기능을 이동 에이전트를 통해 감시함으로써 NE의 특성을 충분히 살리는 최적화 된 망관리가 수행되고, 불필요한 MO(Managed object) 요소가 감소.

그림2.은 에이전트의 활동 구조를 나타내고 있다.

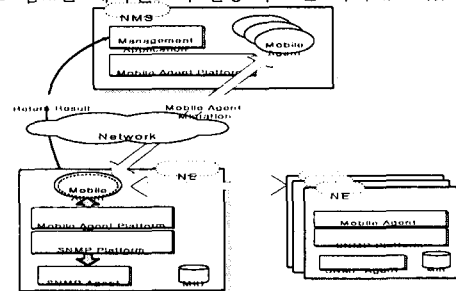


그림 2. Mobile Agent를 이용한 망관리구조

III. JNDI

본 논문에서는 관리자가 대리자의 정보를 직접 참조하여 Mobile Agent를 파견 또는 이동시킬 수 있도록 하기 위하여 JNDI에서 제공하는 네이밍 & 디렉토리서비스를 사용한다.

1. 네이밍 & 디렉토리 서비스

네이밍 서비스는 분산된 객체의 객체 참조를 쉽게 얻어올 수 있도록 객체 참조를 한곳에 모아

참조할 수 있도록 한다. 사용자 관점에서는 혼합된 네임을 구성하는 name-space가 있는 것이다. 즉, 객체의 이름을 가지고 분산된 객체를 쉽게 찾기 위한 객체를 이름과 함께 등록하고, 클라이언트는 네이밍 서비스 서버에 접속하여 등록된 이름을 사용해서 객체에 대한 객체 참조를 얻는다.

디렉토리 서비스는 인터넷에서 사용자, 기계, 네트워크, 서비스, 응용프로그램에 대한 정보를 제공하는 아주 중요한 역할을 한다. 디렉토리 서비스는 다양한 엔트리의 정돈과 일치작업을 하기 위해 인간의 이해 가능한 namespace를 제공하기 위해 네이밍 기능을 통합하고 있다. 기업의 컴퓨터 네트워크 환경은 다른 여러 혼합된 namespace를 나타내는 몇개의 네이밍 기능으로 구성되어 있는데 그 예로 전자 메일을 사용할때 메일을 보내고자하는 받는 사람의 이름을 정해주어야 한다. 각 객체들은 네이밍 서버에 자신의 정보를 등록하고 클라이언트는 네이밍 서비스 서버에 접속하여 등록된 이름을 사용하여 객체에 대한 참조를 얻는다. 네이밍 서비스는 대부분 디렉토리 서비스로 확장 되어지는데 네이밍 서비스가 이름으로 주어진 객체를 조사하도록 제공해 주는데 반면 디렉토리 서비스는 그 객체의 속성(attributes)을 가지게 해준다. 그러므로 디렉토리 서비스는 조회하는 것 이외에 객체의 속성 혹은 주어진 속성의 객체(object)까지도 조회할 수가 있다. 네이밍 & 디렉토리 서비스는 사용자, 기계, 네트워크, 서비스 응용프로그램 관한 다양한 정보를 널리 공유하므로써 인트라넷 과 인터넷에서 중요한 역할을 한다.

네이밍 서비스는 등록된 객체를 그림과 같이 계층적인 트리 형태로 관리하게 된다. 이러한 트리의 구조로 관리되는 객체의 이름은 시스템 내부의 파일 시스템의 파일 이름처럼 계층적인 이름을 가지게 된다. 이러한 계층적인 트리 구조를 네이밍 그래프(Naming Graph)라 한다.

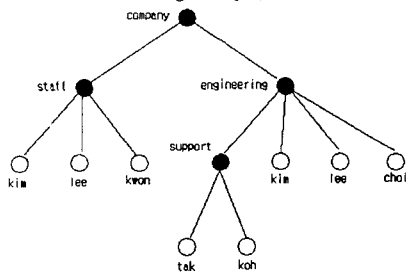


그림 3. 네이밍 그래프

그림 3.의 네이밍 그래프에서 각각의 사원은 다음에서 처럼 컨텍스트와 객체이름으로 구성된 자기만의 합성된 객체이름을 가지게 된다. 그림 3.와 같이 staff라는 부서에 일하는 kim라는 사원은 자신만의 독특한 합성이름인 company.staff.ki-m 라는합성이름(compound

Name)을 가지게 된다.

네이밍 서비스는 company.staff.kim과 같은 합성이름과 실제 객체의 객체참조를 관리한다. 이러한 객체참조는 그림 4.에서와 같이 객체가 실제로 위치하는 IP 주소와 클라이언트의 호출을 기다리는 port번호, 객체의 IDL 인터페이스의 저장소 ID, 또한 객체를 식별할 수 있는 유일한 객체의 위치를 알고자 하는 클라이언트에게 객체 참조를 반환하고, 클라이언트는 반환된 객체 참조를 이용하여 서버와 통신하게 된다.

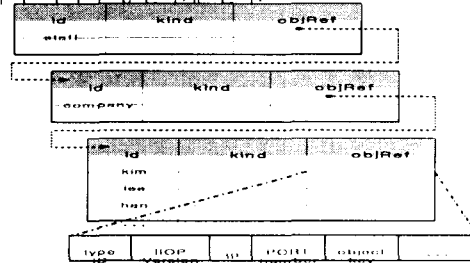


그림 4. 합성이름과 객체사이의 참조관계

2. JNDI

JNDI(Java Naming Directory Interface) 네이밍 & 디렉토리 서비스를 제공하는 Java API(Application Program Interface)이다. JNDI은 선 마이크로시스템에 의해 개발되었으며 JNDI을 사용하므로써 자바 어플리케이션은 자바 객체를 저장하고 검색할수 있다. 게다가 JNDI은 어트리뷰트(attributes)을 사용하는 객체를 검색하거나 객체에 그 어트리뷰트를 연결 하는 것 같은 기본적인 디렉토리 오퍼레이션을 수행하기 위한 방법을 제공한다. JNDI은 네이밍 & 디렉토리 서비스 구현에 독립적이다. 이것은 공통된 API를 사용하므로써 각각 각색의 네이밍 & 디렉토리 서비스를 자바응용프로그램을 통해 액세스할 수 있도록 하여준다.

여러 네이밍 & 디렉토리 서비스 제공자는 공통된 API 뒤에서 단락없이 접속될 수 있다. 이것은 Java 응용프로그램이 LDAP, NDS, DNS 그리고 CORBA 같이 존재하는 다양한 네이밍 & 디렉토리 서비스를 이용할 수 있도록 해준다. JNDI을 사용하므로써 자바 응용프로그램은 강력하고 인식성이 강한 응용프로그램을 만들 수 있다. 그 응용프로그램이 채택되어 환경과 잘 접합될 수 뿐만 아니라 자바의 객체 모델의 이점을 이용할 수 있다.

그림 5. 는 JNDI 구조를 보여 주고 있다.

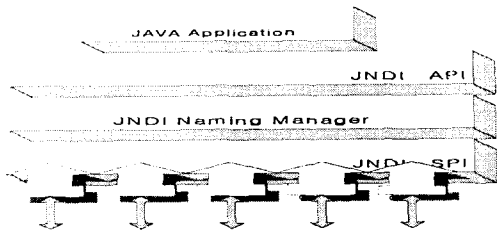


그림 5. JNDI의 구조

- The Naming Interface 제공 : javax.naming
- The Directory Interface 제공 : javax.naming.directory
- TheEvent Interface 제공: javax.naming.event
- The LDAP Interface 제공: javax.naming.ldap
- The Service Provider Interface 제공: javax.naming.spi

3. JNDI를 이용한 Mobile Agent 망관리

Mobile Agent 기반 망관리는 앞 절에서 언급한 것과 같이 관리자가 자신의 역할을 Mobile Agent 에 일임하여 직접 대리자가 있는 NE에 파견하고 Mobile Agent 는 일임받은 임무를 수행하여 관리자의 역할을 대신 하는 구조를 가진다. 관리자가 Mobile Agent를 생성하고 파견하는 것에 대해 기존의 방법들은 관리자가 대리자를 선택하는 부분에 있어 어떠한 참조 또는 정보를 제공받지 못하였으며 분산환경에서 망의 투명성을 제공받지 못하였다. 본 논문에서 제안한 구조에서는 관리자는 네이밍 혹은 디렉토리 서버로부터 네임 혹은 속성들을 참조할 수 있기 때문에 분산환경에서 중요하게 여겨지는 망 투명성과 위치 투명성을 제공받을 수가 있다.

IV. 시스템 설계 및 구현

JNDI를 이용하여 네이밍 & 디렉토리 서비스를 Mobile Agent 기반 망관리에 제공하기 위하여 제안한 구조와 구현결과는 다음과 같다.

1. 시스템 구조

그림 6.은 본 논문의 이동 에이전트 기반 망 구조에 대하여 나타내고 있다.

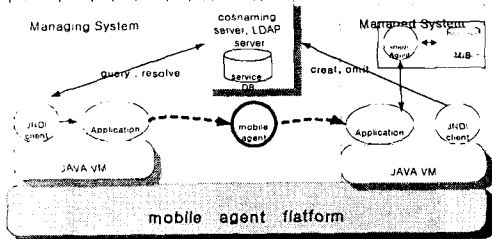


그림 6. 시스템 설계구조

망관리를 하기위해 관리자는 JNDI를 이용한 미리 갖추어진 네이밍 서버로부터 DB로부터 이동 할 피관리시스템(NE)의 위치(IP)와 port번호, NE의 속성등을 참조 받아 대상 해당 NE에 Mobile Agent를 파견시킬 수 있다. Mobile Agent는 NE의 대리자와 SNMP 오퍼레이션을 수행하여 관리자의 작업을 대신하고 결과를 관리시스템에게 보고한다.

2. 시스템 구현 환경

구현한 시스템의 환경은 다음과 같다.

- managing system의 환경
 - 하드웨어 : x86계열 Intel Pentium 프로세서
 - 운영체제 : 솔라리스2.7,
- managed system의 환경
 - 하드웨어 : SUN 워크스테이션(SUN Ultra-2), x86계열 Pentium dual 프로세서
 - 운영체제 : 솔라리스 2.6, NT
- 시스템 구현에 사용된 언어(도구)
 - 2 SDK v1.2(JAVA 언어)
 - JAVA beans(AdventNet SNMPv1 class)
 - voyager(이동에이전트 생성 프레임)
 - JNDI(Java Naming Directory Service Interface)
 - LDAP server program, COSNaming server

구현한 시스템은 그림 7.과 같은 순서로 동작하며 각각의 과정은 다음과 같다.

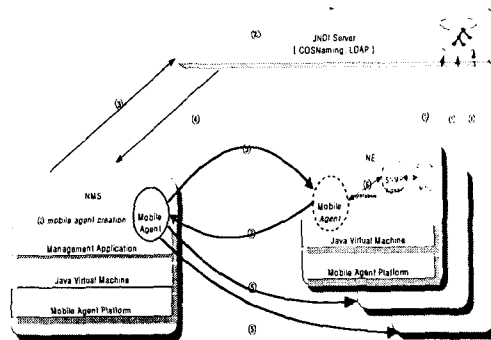


그림 7. 설계시스템 동작과정

피관리시스템이 JNDI서버에게 자신의 정보를 등록시킨다(①). NMS의 응용프로그램은 Mobile Agent를 생성하여(②) JNDI Server에 등록된 Mobile Agent는 등록된 피관리시스템의 정보를 네이밍 서비스를 받아(③·④) 피관리시스템으로 이동하여(⑤) Mobile Agent에 부여된 의무는 SNMP 대리자와 통신한다(⑥). SNMP 대리자는 MIB-II 값을 가져와 오퍼레이션을 수행한다. 그리고 수행된 결과를 관리시스템으로 넘긴다(⑦).

그림 8.은 동작의 수행결과이다.

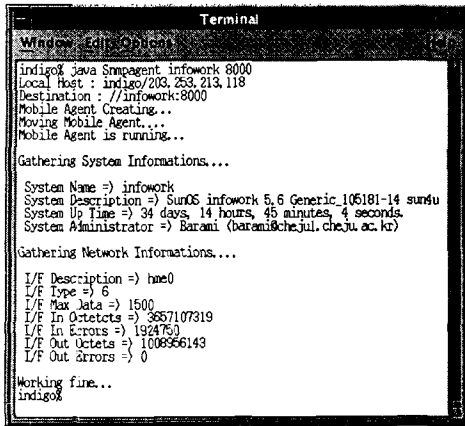


그림 8. 관리자시스템에서의 실행 결과

V. 결 론

본 논문에서는 Mobile Agent가 NE에 존재하는 네트워크 객체에 쉽게 접근하기 위해 네이밍 & 디렉토리 서비스를 이용하는 방안을 제시하였다. 관리자는 여러 NE들을 관리하면서 NE들의 특성, NE에서 제공하는 MIB, 위치 등의 어트리뷰트므로 네이밍 & 디렉토리 서버를 통해 객체를 참조할 수 있고 접근할 수 있었다. 그 결과로 관리자는 대리자에 대한 세밀한 정보 혹은 망 특성과 위치 특성에 따른 부분들을 고려하지 않아도 됨에 따라 Mobile Agent 기반의 망관리에 망투명성과 위치투명성을 제공할 수 있었으며 이것은 분산환경에서 무척이나 유리한 점으로 작용한다.

향후 연구 과제로는 차후 관리 기능의 추가 기능은 보안 문제를 해결한 Mobile Agent 시스템 개발이 필요하겠고, 망관리의 글로벌에 따른 복잡성을 제거하기 위해 Mobile Agent의 위치파악을 위한 또 다른 시스템 개발이 필요하겠으며, 에이전트가 이용 할 데이터 베이스의 구축이 당면 과제로 남아 있다. 이런 문제가 해결이 되면 Mobile Agent를 이용한 망관리 모델은 더욱더 효과적이고 앞으로 지속적인 발전을 가질 수 있는 망관리 모델이 될 것으로 고려된다.

참고 문헌

- [1] William Stllings, 「SNMP SNMPv2 and RMON」, Addison-Wesley Publishing Company, pp.84~ 120. pp.145~196, 1996
- [2] R. J. Post. 1995. Manager-Manager MIB analysis and implementation, University of Twente TIOS. pp 67

- [3] Sun Microsystem, inc., 「Java Naming and Directory Interface Application Program Interface(JNDI API, SPI)」, 1999
- [4] 조일권, 이동에이전트를 이용한 망관리 방법에 관한 연구, 석사학위논문, pp.16-17, 1997
- [5] 조수형, corba기반에서 SNMP와의 연도에 관한 설계 및 구현, 석사학위논문, pp.12-13, 1999
- [6] 이인화, Corporate LAN상에서 최적의 네트워크 관리 구현 방안, 석사학위논문, pp.4-6, 1994
- [7] 정보통신부, 「단순망 관리 규약(snmpp) 표준」 pp.8-37, 1993
- [8] 한국전산원, 「통합망관리 표준화 연구」 pp.114-119, 1995
- [9] Advent Network Management inc., Advent SNMP Package」
- [10] Morris Sloman, 「Network and Distributed System Management」 pp.250~280, 1996
- [11] ObjectSpace inc., 「Voyager ORB3.0 Developer Guide」, 1999