

분산환경에서 Jini 기반 SNMP 에이전트의 설계

○

이성동*, 김원태**, 배민수*, 안광선*

*경북대학교 컴퓨터공학과

**가야대학교 컴퓨터공학과

e-mail:holyeast@comeng.ce.knu.ac.kr

Design of Jini based SNMP agent in distributed environment

○

Seong-Dong Lee*, Won-Tae Kim**, Min-Soo Bae*, Gwang-Sun Ahn*

*Dept of Computer Engineering, Kyungpook National University

**Dept of Computer Engineering, Ka-Ya University

요 약

분산 환경에서 네트워크 관리 시스템을 구현하기 위한 연구가 오래 전부터 진행되어 왔다. 기존의 경우 CORBA, Java의 RMI와 네트워크 관리 시스템을 연동하여 게이트웨이를 만들거나 보편적인 인터페이스인 웹 기반의 관리 시스템이 연구되었다. 또 웹 기반의 관리 시스템은 WBEM, JMAPI 등으로 표준화하려는 노력이 있었다. 본 논문에서는 기존의 관리 시스템이 가지는 문제점을 해결하기 위해 Jini 기반 SNMP 에이전트의 구조를 제안한다. 본 시스템은 기존 시스템이 폴링 기반이어서 발생하는 자원 낭비를 배제하여 성능 개선이 예상된다.

1. 서론

산업 사회의 고도화와 그에 따른 컴퓨터 산업의 발전으로 컴퓨터의 사용 인구가 폭발적으로 증가하였으며, 이로 인해 개인의 생활은 컴퓨터와 밀접한 관계를 갖게 되었다. 특히 90년대 중반부터 WWW 서비스의 등장으로 네트워크의 트래픽 증가는 급격히 일어나고 있다. 이는 네트워크의 고속화에 대한 요구를 촉발시켰으며, 네트워크 관리에 대한 요구를 함께 증가시키고 있다. 이런 상황에서 점차 복잡해지고 있는 네트워크를 유지, 운영하기 위해서는 네트워크 관리 시스템의 사용이 더욱 절실하게 되었다.

대표적인 네트워크 관리 시스템은 SNMP(Simple Network Management Protocol)[1, 2]와 국제표준기구(ISO)의 CMIP(Common Management Information Protocol)[3]이 있는데, SNMP는 구조가 간단하고 구현이 용이해 많은 업체에 채택되고 있어 사실상 네트워크 관리 시스템의 표준이 되었다. 이런 SNMP를 분산환경에 구현하기 위해 CORBA[4], Java 기반의 RMI(Remote Method Invocation)[5]와 연동[6]하거나 혹은 Sun의 JMAPI(Java Management API)[7]와 WBEM(Web-Base Enterprise Management)[8]과 같이 웹 기반으로 표준화하려는 등 많은 연구가 있었다. 그러나 분산환경에서 흔히 등장하는 가장 큰 문제 중 하나는 협력하는 컴포넌트 중 일부가 예고 없이 동작하지 않는 경우가 생긴다. 이렇게 장애가 발생한 리소스에 접근하려는 프로그램은 타임아웃 될

때까지 계속 리소스로의 접근을 시도하게 되므로, 결국 에러가 발생했을 때에 예외 처리를 제대로 해주지 않아 프로그램 자체가 다운된다. 따라서 반복적인 리소스의 접근으로 인한 자원의 낭비를 유발하게 된다.

본 논문에서는 위의 문제점을 해결할 수 있는 분산 환경에서의 Jini[9, 10] 기반 SNMP 에이전트를 제안한다. Jini 기반의 네트워크 관리는 플랫폼에 관계없이 쉽게 사용할 수 있으며, 분산 환경에서의 타임 아웃된 장비에 대한 반복적인 접근을 배제하여 자원의 낭비를 막을 수 있고, 언제든지 Jini 네트워크에 연결되기만 하면 다른 설치나 부수적인 절차 없이 즉시 사용 가능한 장점을 가진다. 본 논문의 구성은 1장은 서론이며, 2장은 SNMP와 Jini 기술에 대해 상술한다. 그리고 3장에서는 Jini 기반의 SNMP 에이전트의 구조를 제안하고, 마지막으로 4장에서 결론을 맺는다.

2. 관련연구

2.1 SNMP

SNMP는 TCP/IP 네트워크에서의 이기종 네트워크 관리를 위한 Internet 프로토콜이며, 단순하여 시스템에 부담을 적게 준다. SNMP는 관리자가 원격 호스트나 게이트웨이의 에이전트에게 네트워크 상내에 대해서 물어보고 원격 호스트의 동작을 제어하도록 한다. SNMP는 관리자와 에이전트 사이에 어떻게 메시지를 전달하는지를 정의하고, 관리자가 TCP/IP Internet에서의 문제점을 찾아서 해결하는데 사용된다. 명령체계도 Get, Set 그리고 Trap으로 구성되어

서 아주 간단하며, 기본적으로 MIB(Management Information Base)[11]라는 관리 정보 집합을 사용하도록 하고 있다. 관리 정보 집합인 MIB는 해당 피관리 시스템에 대한 정보 수집의 중요한 자료가 되는 것으로 이를 기반으로 하여 모든 네트워크 관리 시스템들이 개발되고 있다.

일반적인 SNMP의 구조는 그림 1과 같으며, 관리자는 통계자료를 결정하기 위해서 변수로부터 값을 읽고, 관리자는 에이전트를 통제하기 위한 값을 MIB 변수에 저장한다. SNMP 메시지는 에이전트가 MIB 변수의 어떠한 값을 읽어와 하고 저장해야 하는 지에 대한 내용을 가지고 있다. 에이전트는 관리자의 SNMP 메시지에 지정된 명령을 지역(local) 데이터에 맞는 관리동작을 수행한다. 에이전트는 현재 운용중인 네트워크에 대한 현황을 파악하고 이를 개선시키기 위해서 관리 정보를 수집하여 분석을 시행한 후, 이에 대한 문제점을 파악하고 이를 의사 결정 과정을 통해 해당 피관리 시스템을 제어하는 형태를 갖는다. 그리고 MIB 변수는 ISO의 표준인 ASN.1(Abstract Syntax Notation One)[12]을 사용하여 표현한다.

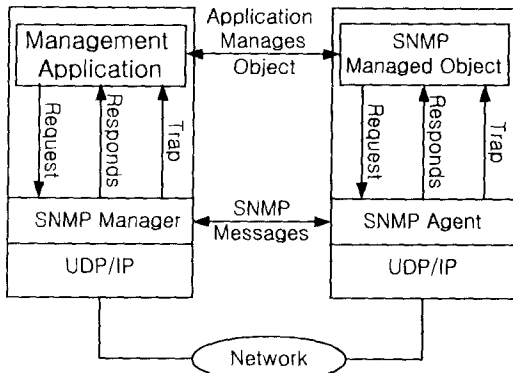


그림 1. SNMP 시스템 구조

2.2 Jini

Jini 기술은 Sun Microsystems에서 개발된 차세대 접속기술이다. Jini는 네트워크상의 플러그 앤 플레이(Plug & Play : PnP)의 개념을 가능케 해 주는 프로토콜이며 클래스 라이브러리이다. 즉, 어떤 Jini 기반의 디바이스가 하드웨어든 소프트웨어든 네트워크에 연결이 되기만 하면 그 네트워크에 연결된 사용자나 다른 디바이스는 즉시 이를 감지하여 그 디바이스를 사용할 수 있게 해 주는 기술이다. 이때 Jini를 장착한 장비들은 하나의 서비스 개념으로 처리되는데 이러한 서비스들이 네트워크를 통해 상호 연결되어서 서로 서비스를 공유할 수 있는 Jini 연합체를 형성하게 된다. 여기서 Jini 연합체는 Jini를 장착한 기기끼리 특정한 업무를 실행, 처리하기 위해서 필요한 서비스의 집합으로서 서로간에 상호 서비스를 이용할 수 있도록 해주는 기능을 가지고 있다. Jini는 서비스 제공자가 자신의 서비스를 Lookup 서비스에 등록한

후 서비스 이용자가 그 Lookup 서비스에 필요한 서비스들을 검색한 후 그 서비스를 이용하는 방식을 취한다.

Jini에서는 Lease[13]라는 개념을 도입하였는데, 이 Lease는 특정 자원에 대한 시간 제약이 있는 허가권이다. 즉 원하는 서비스에 대해서 일정 시간만큼의 사용허가를 받고, 계속 서비스 받기를 원하면 사용자 측에서 연장을 신청하고, 만약 서비스 연장신청이 없으면 더 이상 서비스를 원하지 않거나 네트워크 장애로 간주하여 더 이상 서비스를 해주지 않게 된다. 폴링을 하는 대신 Lease 개념을 사용함으로써 시스템의 부하가 분산되어서 성능 향상을 기대할 수 있고, 효율적인 시스템 구성이 가능하다.

3. SNMP 에이전트의 구조

본 논문에서 제안한 시스템의 구조는 그림 3과 같다. 구성요소는 Jini 인터페이스, SNMP 데몬, 관리자에게 보내어질 유저 인터페이스, MIB, Document의 5가지 부문이다. Jini 인터페이스 부분은 Jini 네트워크의 연결과 등록 등의 작업을 하며, Lookup 서비스로부터 proxy 역할인 registrar를 받아들인다. SNMP 데몬은 기존의 SNMP 에이전트의 역할을 수행하며, 관리자의 요구에 대한 응답을 담당한다. 관리정보는 MIB에 저장되어 있고, Document에는 운영에 필요한 정보가 들어있다. 그리고 유저 인터페이스가 에이전트 쪽에 위치하여서 에이전트 쪽의 모듈이 관리자의 모듈에 비해 상대적으로 키지게 되며, 관리자는 Jini 인터페이스만을 가지므로 간단한 구조를 가진다.

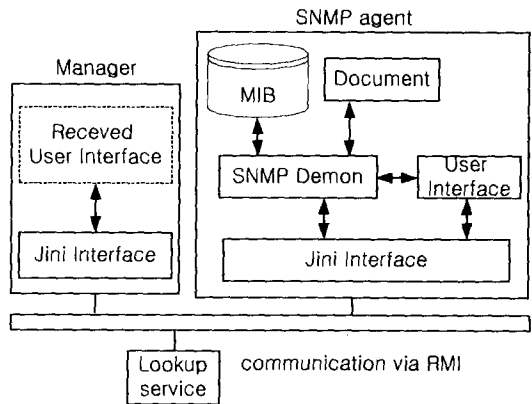


그림 2. Jini 기반 SNMP 시스템의 구조

Jini 기반으로 네트워크관리 시스템을 구성하면 폴링방식을 원천적으로 배제하며, Lease 메커니즘을 통해 자원을 시간 개념으로 관리하므로 기존의 분산환경에서의 폴링에 따른 자원 낭비와 하나의 시스템에 많은 부하가 걸리는 문제점을 해결할 수 있게 된다.

본 논문에서 제안한 시스템의 관리자와 SNMP 에이전트 사이의 동작 과정을 UML의 시퀀스 다이어그램으로 표현하면 그림 2와 같다. Jini 기반의 여러 SNMP 에이전트들이 Lookup 서비스를 발견하고

Lookup 서비스의 연결 프록시인 registrar를 받아 이를 이용하여 해당 패관리 대상의 정보와 자신의 속성을 담은 서비스 프록시 객체를 Lookup 서비스에 등록한다. 그 후 관리자가 Lookup 서비스를 발견하고 에이전트와 동일한 방법으로 연결 프록시인 registrar를 이용하여 여러 SNMP 에이전트 중 관리하고자 하는 대상을 결성하여 에이전트에서 생성한 서비스 프록시를 받는다. 유저 인터페이스와 데이터가 담겨있는 프록시를 관리자의 자바 가상머신에서 실행하여 에이전트의 네트워크 관리 정보를 알 수 있으며, 관리자의 명령은 Lookup 서비스를 통해 에이전트에 전달되며, 명령에 대한 응답과 상태변경 및 트랩은 Jini의 리모트 이벤트를 이용하여 관리자에게 전달된다.

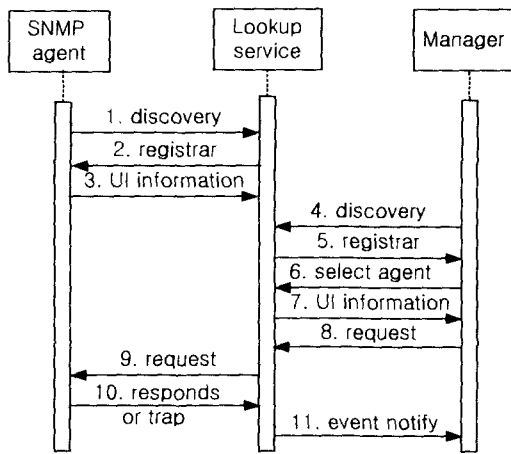


그림 3. 관리자와 에이전트의 동작 과정

4. 결론

네트워크의 규모가 커지고 네트워크를 이용한 서비스의 증대됨에 따라 이들 전산 네트워크를 효율적으로 관리하기 위한 네트워크 관리 시스템의 필요성과 중요성에 대한 인식이 높아가고 있고, 기존의 네트워크 관리 시스템의 성능을 향상하고자 하는 노력이 확산되고 있다.

본 논문에서는 Jini 기술을 이용한 SNMP 에이전트를 제안하였다. Jini는 Sun에서 제안하고 있는 차세대 접속기술로서 Java를 이용하여 네트워크 상의 장비나 소프트웨어를 동적으로 연계시키는 기술이다. 특정한 컴퓨터의 하드웨어나 소프트웨어에 구애 없이 Jini를 채택하고 있는 장비들이 어떠한 인위적인 조작이나 설치와 같은 절차 없이 네트워크에 연결되기만 하면 시간과 장소에 상관없이 네트워크 상에서 서로 간의 상호작용을 통해 여러 가지의 작업을 할 수 있다.

Jini를 SNMP 에이전트에 적용하면 부수적이며 수동적인 인스톨 과정 없이 플러그 앤 플레이 방식으로 쉽게 네트워크 연결만으로 사용이 가능하고, 또한 적용 장비가 네트워크 장비에 국한된 것이 아니라 프린터나 홈 네트워크 장비 등에 쉽게 적용할 수 있어서 확장성이 우수하다. 또한 협력 컴포넌트의 연결 상황

을 확인하기 위해 계속 폴링(polling)하여 다운된 리소스에 대한 접근으로 인한 시스템의 부하를 줄일 수 있어서 성능 향상이 가능하다. 또한 Java의 환경을 사용하므로 애플릿으로 작성하거나 AWT, Swing, Java2D 등을 이용한 GUI 구성이 가능하며 전송되어 질 객체가 Serialization 되어서 전송되어 안전하다. 그리고 확장성이 우수하여 부수적인 노력 없이 홈 네트워크의 관리에 적용할 수 있다. 그러나 본 논문의 제한점은 Jini의 특성상 자바 가상 머신과 RMI가 반드시 필요하므로 현실점에서 실용화하기에 제약이 있으며, 유저 인터페이스가 에이전트 쪽에 위치하고 있어서 에이전트 모듈이 관리자에 비해 커지게 된다. 또 아직 구현을 하지 않아서 그에 따른 성능 평가가 부족하다. 차후 과제로는 실제로 구현하여 성능 평가를 해보고 실제 구현상의 문제점을 연구하고, Lego Mindstorm에 적용하여서 홈 네트워크의 관리에 관한 연구가 이뤄져야 한다.

참고문헌

- [1] RFC 1157, *A Simple Network Management Protocol(SNMP)*, May, 1990.
- [2] W. Stallings, *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*, Addison Wesley, 1999.
- [3] ISO 9596/CCITT X.711, *Common Management Information Protocol - CMIP*, 1991.
- [4] P. Haggerty and K. Seetharaman, "The Benefits of CORBA-based Network Management," *Communications of the ACM*, Oct. 1998.
- [5] Sun Microsystems, *Java Remote Method Invocaton Specification*, Available at <http://www.javasoft.com/product/jdk/rmi/>, 1998.
- [6] J. Park, N. Ban and T. Kim, "Java-based Network Management Environment," *IEEE*, 1998.
- [7] Sun Microsystems, *Java Management API architecture*, 1996.
- [8] WBEM consortium, "Web-Based Enterprise Management Protocol: HyperMedia Management Protocol Overview," April, 1996.
- [9] N. Jim, *Guide to JINI Technologies Jini tutorial*, Available at <http://pandonia.canberra.edu.au/java/jini/tutorial/Jini.xml>, 1999.
- [10] W. K. Edwards, *Core Jini*, Sun microsystem press, 1999.
- [11] MIB RFC 1213. "An enhancement to MIB I for use with network management protocols in TCP/IP based networks," 1991.
- [12] Information processing system - Open Systems Interconnection, *Specifacation of Abstract Syntax Notation One(ASN.1)*, International Organization for Standardization, 1997.
- [13] Sun Microsystems, *Jini Distributed Leasing Specification*, Available <http://www.javasoft.com/product/jini/>, 1999.