

확장성을 고려한 웹 기반 SNMP 망 관리 시스템의 구현

민 경 주[†] · 권 택 근^{††}

요 약

SNMP는 네트워크의 장비들 사이에서 관리 정보를 주고받는 응용 계층 프로토콜로 동작 방법은 매우 간단하다. SNMP는 PDU를 통해서 관리 정보의 이름과 값을 서로 교환한다. 지금까지 네트워크 관리는 SNMP를 이용해 콘솔 기반으로 관리되었기 때문에 매우 불편했다. 이 논문에서는 새롭게 디자인된 사설 MIB와 GUI를 통한 웹 기반 SNMP 망 관리 시스템을 구현한다. 또한 확장성 문제를 해결하기 위해 가상 에이전트를 제안하고 구현한다. 제안된 가상 에이전트를 사용함으로써, 무분별한 트랩 발생에 대한 필터링과 계층적인 망 관리가 가능하게 된다.

An Implementation of Web-Based, Scalable SNMP Network Management System

Kyoung J. Min[†] · Taeck G. Kwon^{††}

ABSTRACT

SNMP is a simple, application layer protocol and that is sending and receiving the management information among network devices. It exchanges the management information by the PDUs that have the management information names and values. Until now, network management is so inconvenience because of console-based management by using SNMP. This paper shows the Web-based SNMP network management system that is using the designed private MIB and GUI. In this paper, we have proposed a virtual SNMP agent supporting scalability of network management. With the virtual agent, large-scale network can be managed simply by trap filtering and hierarchical management.

키워드 : SNMP, NMS(망 관리 시스템), Scalable(확장성), 가상 에이전트(Virtual Agent), 트랩 필터링(Trap Filtering)

1. 서 론

IP 망의 규모가 커지고, 또 그것에 참여하는 망 장비가 다양해짐에 따라 그것들을 효율적으로 관리할 수 있는 망 관리기능이 필요하게 되었다. 이러한 망 관리 기능을 제공하기 위한 대표적인 표준 프로토콜이 SNMP(Simple Network Management Protocol)인데, 그 구조의 단순성과 구현의 용이함으로 개발 작업이 활발히 진행되어 망 관리 표준으로 자리잡았다[1]. 망 관리의 필요성을 인식한 많은 교육기관과 기업에서 SNMP를 이용한 망 관리 시스템을 개발하게 되었다. 하지만 망의 규모가 급속히 확대되면서, 기존의 콘솔 기반 관리는 현실적으로 기능을 상실하게 되어 망 관리자 입장에서 관리를 효율적으로 하기 위한 EMS(Element Management Station)의 개발 필요성과 확장성에 대한 고려가 대두되었다. 또한 GUI(Graphic User Interface)를 이용한 EMS의 개발에 자바 언어(Java Language)를 이용함으로써 플랫폼

(Platform)에 독립적인 망 관리 시스템의 개발은 필수적이라 할 수 있다.

이러한 이유로 본 연구에서는 SNMP를 이용한 망 관리 시스템을 구현하였는데, 개발의 신속성과 개발비용의 절감을 위해, 에이전트는 리눅스 플랫폼에서 구현되었고, 매니저는 플랫폼에 독립적인 자바 언어 기반 SNMP(Java-SNMP)를 이용한 EMS를 개발하였다. 또한 매니저와 에이전트 사이에 가상 에이전트를 두어 최근의 망 관리 핵심 목표[2-4] 중 하나인 확장성 문제 해결이 가능하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 SNMP 에이전트의 기본 용어 및 구조와 동작을 소개한다. 3장에서는 CNU-SNMP(Chungnam National Univ.-SNMP) 구현에 관한 상세한 사항과 확장성 해결을 위한 설계에 관한 상세한 사항을 소개한다. 결론 및 향후 연구과제는 4장에서 기술한다.

2. SNMP의 구조와 동작

SNMP는 망 관리자가 위치하는 매니저의 영역인 관리국(Management Station), 매니저의 요청에 대해 적절한 행동

† 정 회 원 : 한국전자통신연구원

†† 정 회 원 : 충남대학교 정보통신공학부 교수

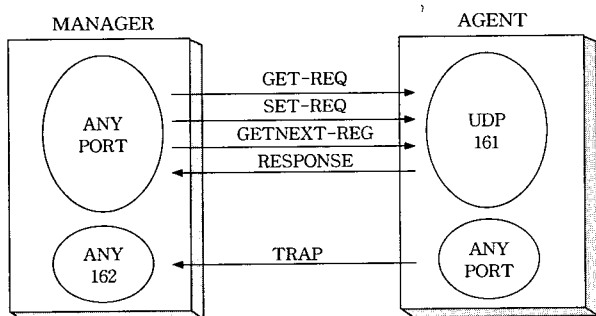
논문접수 : 2002년 1월 2일, 심사완료 : 2002년 3월 28일

을 취하고, 매니저에게 응답을 하는 피관리 에이전트(Managed Agent), 망 관리 대상에 대한 개념적 구조체인 MIB, 매니저와 에이전트 사이의 통신을 위한 프로토콜인 SNMP 네 개의 요소를 가지고 동작한다[5-7]. SNMP의 버전은 1, 2, 3으로 크게 분류된다. 버전 1은 SNMP의 기본 동작인 GET, SET, TRAP을 지원한다. 버전 1은 두 가지 큰 문제점을 갖는데, 이는 보안 문제를 다루지 않는다는 점과, 라우팅 테이블의 정보를 요청할 때 반복적인 정보의 요청으로 인해 과도한 대역폭의 낭비라는 문제점을 갖는다. 이를 개선한 버전 2는 송신할 때 DES(Data Encryption Standard) 알고리즘을 사용하고, 사용자 인증 시에는 MD5(Message Digest 5) 알고리즘을 사용함으로써 보안 문제를 다루게 된다. 또한 과도한 대역폭의 낭비를 방지하기 위해 bulk 요청을 한다. 또한 버전 1에서 에이전트가 트랩을 발생시켰을 때 매니저가 트랩을 받았음을 알리는 Inform 메시지를 보냄으로써 트랩의 효율성을 높인다. 또한 버전 3에서는 버전 2의 기능에 사용자 인증, 보안, 접근 제어를 강화한다[5, 8].

2.1 SNMP 관리 모델

SNMP의 주된 동작은 GET-REQUEST, SET-REQUEST, TRAP의 세 가지로 크게 분류할 수 있다. 매니저는 에이전트에게 관리에 필요한 대상에 대한 요청을 하게 되고, 이에 에이전트는 모듈에 정의된 절차를 수행한 결과를 매니저에게 응답하게 된다. 이때 필요한 행동이 GET-REQUEST이고, 에이전트는 이 요청에 대해 응답을 하게 된다.

SET-REQUEST는 매니저가 에이전트의 시스템의 설정을 바꾸기 위한 요청이다. 마지막으로 GET/SET-REQUEST가 매니저가 에이전트에게 요청을 하는 메시지만큼 반대로 TRAP은, 에이전트가 매니저에게 보내는 메시지로 시스템 등에 문제가 발생하면, 매니저의 요청에 관계없이 비동기적으로 매니저에게 보고하는 메시지이다. 이는 매니저와 에이전트 사이에 특정한 포트를 통해서 이루어지는데, 이에 대한 내용은 (그림 1)과 같이 메시지를 보낼 때는 포트의 영향이 없지만, 받을 때는 161번이나 162번을 통해서 받게 된다[9-12].



(그림 1) 매니저-에이전트 통신을 위한 포트설정

2.2 MIB

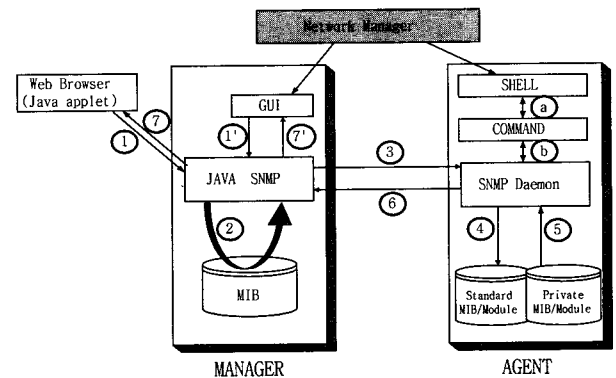
MIB는 루트로부터 시작하는 망 관리 대상에 대한 개념적인 트리 형식의 구조체이다. 각각의 관리 대상은 하나 또는 그 이상의 대상 객체로 구성되고, 특정 객체 식별자(Object Identifier = OID)에 의해 확인된다. 트리 형식의 구조에서 SNMP는 최종 노드인 리프만을 읽고 쓸 수 있다. 이때 트리에서 깊이가 증가함에 따라 '.'로 구분하게 된다. 가령 시스템의 위치를 나타내는 sysLocation은 'iso.org.dod.internet.mgmt.mib-2.system.sysLocation'으로 식별하는데, 실제로는 '1.3.6.1.2.1.1.6'과 같은 연속된 정수로 표현한다. 이때, system은 리프가 아니기 때문에 읽기/쓰기가 허용되지 않고, 리프인 sysLocation만을 읽기/쓰기를 하게 된다[11-13].

3. CNU-SNMP의 설계 및 구현

CNU-SNMP의 구현 환경은 저비용과 신속한 개발환경을 위해 리눅스 플랫폼에서 에이전트가 설계되었고[14], 에이전트와 가상 에이전트에 사용된 리눅스는 펜티엄 프로 200MHz에 와우 리눅스 7.0, 커널 버전 2.4.0이 사용되었다. 또한 윈도우즈 환경에서의 망 관리를 위해 자바-SNMP를 매니저로 사용하였다[15]. 이때 매니저에 사용된 운영체제는 Windows 2000이고 펜티엄4 1GHz에서 동작한다. GUI는 FOUNDARY Networks사의 라우터를 위한 사용자 매뉴얼[16]을 참고하여 구현되었다.

3.1 매니저 - 에이전트의 구성

CNU-SNMP에서 구현한 SNMP의 매니저와 에이전트 사이에서 EMS 동작과정은 (그림 2)와 같다.

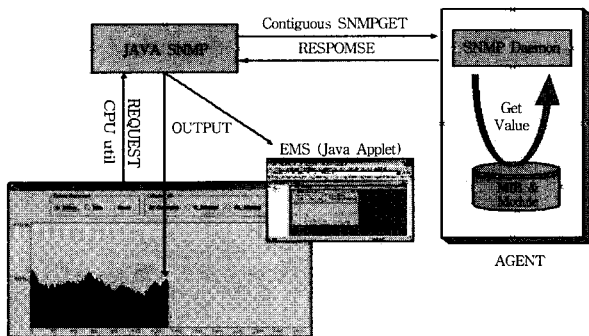


(그림 2) 매니저-에이전트간 EMS 동작과정

(그림 2)에서 보는바와 같이 매니저는 에이전트에게 GUI나 원격접속을 통한 콘솔에서 명령어를 통해 에이전트에게 정보를 요청한다. CISCO ISO 12.0 표준에 의한 CLI(Command Line Interface)를 통해 메시지를 요청하게 되는 불편을 없애기 위해, 명령어를 단순화시켜 콘솔 상에서 관리가 가능하게 되었다[17]. 이를 위해 별도의 셸을 구현하였는데

본 논문에서는 다루지 않는다. 또한 사용자 관점에서 망을 관리하기 위해 자바 언어를 이용한 GUI를 통해 관리함으로써 효율적인 망 관리가 가능하게 된다. 이러한 CNU-SNMP에서 매니저-에이전트간 EMS 메시지 전송 흐름을 보면 다음과 같다.

(그림 2)와 같이 관리자는 GUI (①)나 웹 브라우저(①')를 통해 매니저인 자바-SNMP에게 정보를 요청한다. 자바-SNMP는 MIB를 참고하여(②) 에이전트에게 요청을 하게 되고(③), 에이전트는 MIB나 MIB 모듈에 구현된 절차를 따라(④) 시스템 내에서 값을 가져와(⑤) 자바-SNMP에게 전달한다(⑥). 자바-SNMP는 GUI(⑦)나 웹 브라우저(⑦')에게 값을 보내면, 받은 값을 화면에 출력해 준다. CNU-SNMP에서 구현된 모든 형태의 EMS 시스템은 위와 같은 절차를 거쳐 관리를 한다. 또한 에이전트와의 직접 원격 접속을 통해 단순화한 명령어 체계를 사용하여 콘솔 상에서도(⑧, ⑨) 효율적으로 관리할 수 있다. 실제로, 에이전트의 C.P.U. 사용률을 가져와 GUI나 웹 브라우저를 통해 관리자가 실제로 화면으로 보게 되는 절차가 (그림 3)과 같다.

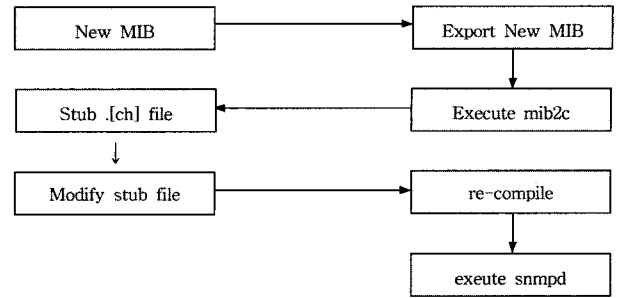


(그림 3) EMS 동작과정의 예 (C.P.U. 사용률)

3.2 에이전트

매니저의 요청에 대해 적절한 행동을 한 결과에 대한 결과를 보내주기 위해서 에이전트가 필요하다. 이러한 에이전트의 행동을 규정하기 위해서는 MIB와 그 모듈이 필요한데, 이러한 MIB는 RFC에서 제공하는 표준 MIB와 각 기업에서 자신의 목적에 맞게 정의한 사설 MIB로 구분할 수 있다. 본 논문의 구현에서는 표준 MIB를 포함하고, 망 관리 시스템에 적합한 프레임워크를 보이기 위해 리눅스에서 사설 MIB와 그 모듈을 정의한다. 이러한 에이전트에서 사설 MIB를 정의하고 모듈을 정하는 절차가 다음의 (그림 4)와 같이 새로운 MIB를 정의하고 문법을 검사하게 된다. 새로운 MIB를 기존의 MIB에 추가한 후에, MIB에 맞는 스티브 파일(.ch)이 생성하는 펄(Perl)로 구현된 mib2c를 실행한다. 매니저의 요구에 적합한 모듈을 스티브 파일에 추가한 다음, 재 컴파일 후 데몬을 실행하면 새로운 MIB와 그 모듈의 추가가 완성된다.

망 관리 시스템은 정보의 모니터링, 통계 정보, 트랩 관리,



(그림 4) MIB 및 모듈 추가 절차

정보의 변경과 같이 크게 네 가지로 분류할 수 있다. 이러한 분류의 프레임워크를 보이기 위한 정의 및 구현된 사설 MIB는 다음과 같다.

3.2.1 정보의 모니터링

정보의 모니터링에는 관리 정보에 대한 값을 얻어오는 정보요청과 성능평가로 구분할 수 있다. 첫째, 정보요청은 매니저의 요청에 대해 해당 OID의 값을 매니저에게 보내주는 역할로 모든 MIB는 정보요청을 만족한다. 대표적인 예가 CNU-SYSTEM-MIB이다. CNU-SYSTEM-MIB는 여러 개의 에이전트를 관리하기 위해 공간적으로 멀리 떨어진 호스트에 대한 정보를 매니저가 알 필요가 있을 때, 호스트의 시스템정보를 포함한, 메모리의 사용에 대한 정보, C.P.U.와 O.S.의 상세 정보 등을 포함한다. 둘째, 성능평가의 대표적인 예가 CNU-CPU-MIB와 CNU-MEMORY-MIB이다. 이 중 CNU-CPU-MIB를 살펴보면 다음과 같다.

CNU-CPU-MIB에는 순간 C.P.U. 사용률, 5분 평균 사용률, 15분 평균 사용률, 그리고 과도한 사용률을 보일 때 매니저에게 트랩을 발생시키기 위해 트랩을 보내는 기준이 되는 값(cpuWarning), 타이머(cpuTimer)값으로 구성된다. 여기서 타이머를 두는 것은 C.P.U. 사용률이 cpuWarning(매니저에 의해 읽기/쓰기 가능) 주위에서 진동하거나, cpuWarning이 상을 계속 유지할 때 무분별한 트랩 발생을 방지하기 위해 초 단위로 설정하는 값이다. 설명한 바와 같이, CNU-CPU-MIB는 성능평가 기능뿐만 아니라, 트랩 발생기능을 함께 포함하고 있다.

3.2.2 통계 정보

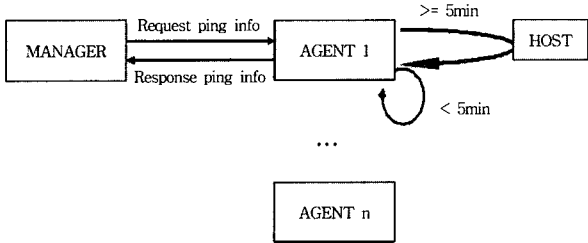
통계 정보는 이후에 발생할 수도 있는 문제나 향후 추가적인 개발에 사용할 목적으로 관리되는 항목이다. 이러한 통계 정보의 대표적인 예로 다음의 두 가지가 있다.

- CNU-STAT-MIB

CNU-STAT-MIB는 에이전트가 받은 패킷의 수와 매니저로부터 받은 메시지의 종류(get, set, get-next, response 등), 받은 메시지에서 발생한 에러의 상황, 에이전트가 매니저에게 보낸 패킷의 수 등 메시지와 패킷에 대한 세부사항을 관리한다.

• CNU-PING-MIB

CNU-PING-MIB의 동작 과정을 살펴보면 다음의 (그림 5)와 같다.

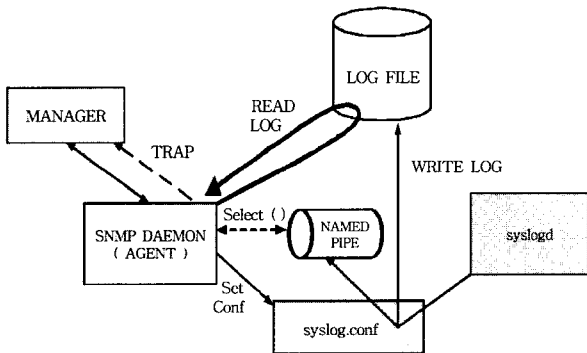


(그림 5) CNU-PING-MIB의 동작

(그림 5)에서 보는 바와 같이 매니저는 에이전트에게 ICMP 메시지를 통해 망의 상태를 요청하게 된다. 이때 에이전트는 ICMP 메시지를 통한 모든 호스트에 대한 ICMP 정보를 자신이 파일로 저장하여 관리를 한다. 이는 무분별한 ICMP 메시지의 사용으로 인해 네트워크의 부하를 증가시키기 때문에 부하를 줄이기 위한 방법으로, 요청한 호스트에 대한 값이 최근 5분 이내의 정보를 요청하게 되면, 에이전트는 자신의 파일에 저장된 내용을 가져와 매니저에게 전달하게 된다. 반면 5분 이상의 정보는 네트워크의 변화를 충족시키지 못한다는 판단 하에 호스트에게 ICMP 메시지를 보내고, 그 값을 파일로 저장하면서 매니저에게 전달한다. 이러한 CNU-PING-MIB는 네트워크의 부하를 줄이고, ICMP 메시지를 통해 얻은 결과로 호스트에 대한 통계정보 관리를 가능하게 해준다.

3.2.3 트랩의 발생

망 관리 시스템에서 가장 중요하게 인식되는 부분이 트랩이다. 트랩은 에이전트에 문제가 발생했을 때 매니저에게 자신의 상황을 알리는 기능으로 대표적인 예가 CNU-SYS-LOG-MIB와 진술한 CNU-CPU-MIB가 있다.



(그림 6) CNU-SYSLOG-MIB의 동작

CNU-SYSLOG-MIB는 시스템에 발생하는 시스템 로그를 매니저가 관리하기 위해 만들어진 MIB 및 그 모듈이다. 시스템 로그의 설정과 로그의 쓰기/읽기 그리고, 트랩을 발

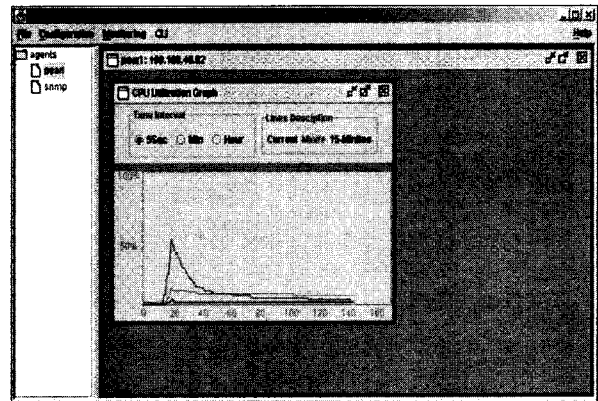
생시키는 절차가 (그림 6)에 도시되어있다.

그림에서 살펴보는 바와 같이, 매니저는 에이전트에게 시스템 로그에 대한 enable /disable을 설정할 수 있고, 로그 파일의 위치와 위험 수준 등을 설정할 수 있다. 이때 syslog daemon은 설정된 설정과일을 참조하여 로그파일에 시스템에 발생한 로그를 기록하게 되는데, 로그가 발생하면 에이전트가 알 수 있도록 네임드 파이프 (named PIPE)라 불리는 FIFO에도 로그를 남기게 된다. 에이전트는 네임드 파이프를 select()함으로써 로그가 새로 발생했음을 알 수 있고, 매니저에게 트랩을 발생시킨다. 매니저는 트랩 메시지를 확인하여, 에이전트 n 으로부터 트랩이 왔다는 사실과 시스템로그에 관한 트랩임을 알고, 에이전트에게 EMS를 통해 로그에 관한 MIB정보를 가져오게 된다.

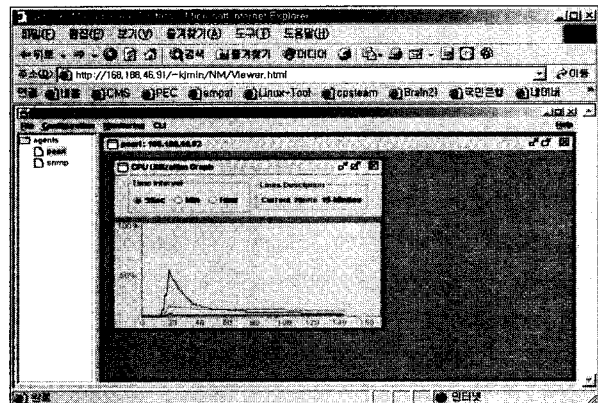
3.2.4 정보의 변경

정보의 변경은 SNMP의 대표적인 set 요청을 통해 에이전트의 정보를 매니저가 수정하는 것으로, MIB에서 읽기/쓰기로 접근 권한이 있어야하고, 인증 되지 않은 사용자가 변경하는 것을 방지하기 위해 인증 과정을 거치게된다. 이러한 정보의 변경은 진술한 대부분의 MIB에서 부분적으로 사용된다.

3.3 EMS



(그림 7-A) 망 관리 창

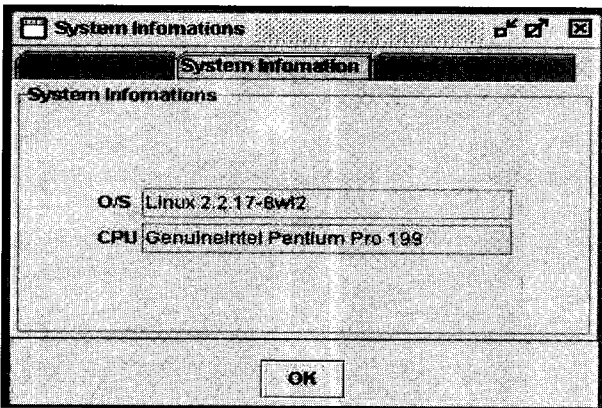


(그림 7-A) 웹 지원을 위한 망 관리 창

매니저의 요청을 사용자 인터페이스의 관점에서 EMS를 이용하여 기존의 텍스트 기반 망 관리를 효율적으로 하기 위해 에이전트에게 메시지 요청을 하고, 트랩을 받게 된다. 이러한 EMS 또한 망 관리의 방향이 되는 정보의 모니터링, 통계 정보, 트랩 관리, 정보의 변경의 네 가지로 분류할 수 있다. 이러한 망 관리를 위해 관리 창(Viewer)이 생성되는데, 자바 언어를 이용한 어플리케이션(application)과 웹을 지원하기 위한 자바 애플릿(Java applet)의 두 가지 방법이 있다. 이러한 두 가지 방법이 다음의 (그림 7-A)과 (그림 7-B)에 있다. 이러한 망 관리 창을 통해서 모든 망 관리가 이루어지게 된다.

3.3.1 정보의 모니터링

에이전트에게 시스템 정보를 요청하여 얻어온 값을 GUI를 통해 화면에 출력해 주는 시스템 정보의 예가 (그림 8)에 나타나 있다.

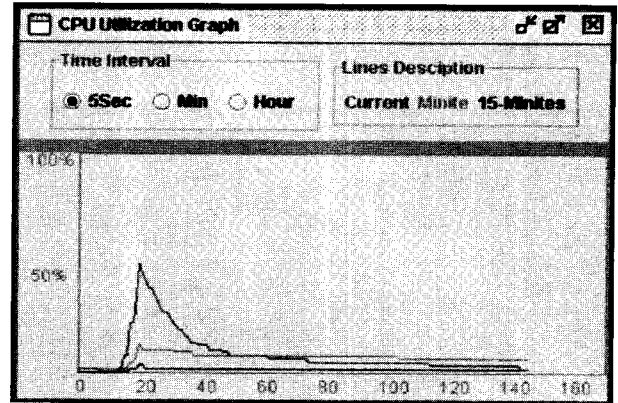


(그림 8) 정보 요청을 통한 시스템 정보

위의 그림과 같은 정보를 얻기 위해, 관리자는 복잡한 MIB 구조를 알아야 할뿐만 아니라, 복잡한 명령어체계를 콘솔에서 입력해야만 했다. 이러한 단점을 극복하기 위해 GUI를 사용하는데, 사용자에게 친근한 GUI를 통해 버튼의 선택을 통해서 원하는 정보를 가져올 수 있다. 위의 예에서는 에이전트의 O.S.와 C.P.U.의 상세 사항에 대한 값을 얻어오는 경우인데, O.S.와 C.P.U. 정보를 갖는 OID에 대한 결과물이다. 이와 유사한 예로 네트워크 정보, 메모리 사용량, 시스템 일반 정보, 메시지 통계 정보 등이 있다.

반면 C.P.U. 사용률과 같이 현재의 사용률 변화를 그래프 형식으로 보는 예가 다음의 (그림 9)에 나타나 있다.

이러한 C.P.U. 사용률은 같은 OID에 대한 연속적인 요청에 대한 응답을 그래프로 나타낸 것으로 시스템 모니터링의 기능을 수행하며 설정된 값 이상이 되었을 때 트랩을 발생하는 기능을 포함한다. 요청하는 시간 간격을 조절함으로써 네트워크의 부하를 줄일 수 있다. 이와 유사한 예로 메모리 사용률의 변화량이 있다.



(그림 9) 연속적인 정보요청을 통한 성능 감시

3.3.2 통계 정보

SNMP를 통해 에이전트는 메시지에 대한 통계 정보를 수집하고 관리한다. 이러한 통계는 향후 개발이나 발생 가능성이 있는 문제점을 해결하는데 도움을 주는데, 대표적인 통계정보가 (그림 10)에 나타나 있다.

The screenshot shows a window titled 'Snmp Statistics' with two columns: 'In Packets' and 'Out Packets'. Each column contains a list of statistics with corresponding values.

In Packets		Out Packets	
Total in-packets size	23099	Total out-packets size	23099
Bad SNMP version errors	0	Too big errors	0
Unknown community name	0	No such name errors	0
Illegal operation for community	0	Bad values errors	0
Encoding errors	0	General errors	0
Number of requested variables	0	Response	85
Number of altered variables	0	Trap	165
Get request	21969		
Get next	872		
Set request	18		

(그림 10) 에이전트의 메시지 통계 정보

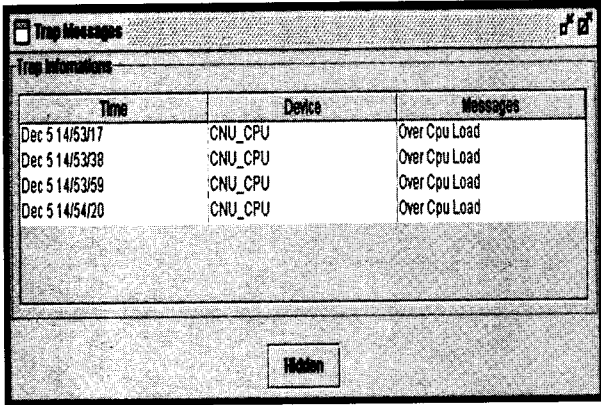
그림에서 보는 바와 같이 에이전트가 관리하는 통계정보는 에이전트로 들어오고 나가는 모든 메시지와 받은 메시지를 분석하여 에러 발생시 발생원인 등에 대한 정보를 포함하고 있다.

3.3.3 트랩의 발생

트랩은 이전의 GET/SET-REQUEST와 달리 매니저의 요청과 관계없이 에이전트 자신에게 문제가 발생했을 때 매니저에게 비동기적으로 알리는 것을 의미한다. 에이전트는 자신에게 문제가 발생하면 매니저에게 상황을 보고해야만 한다. 이것은 시스템의 다운과 같은 치명적인 문제를 예방할 수 있기 때문에 매우 중요하다고 할 수 있다.

위에서 정의한 CNU-CPU-MIB와 CNU-SYSLOG-MIB가 트랩을 발생하는 대표적인 예인데, CNU-CPU-MIB에서 같이 에이전트의 C.P.U. 사용률이 매니저가 정한 이상의 값을 유지하게 되면 cpuTimer에 설정된 시간 간격으로 매니저에

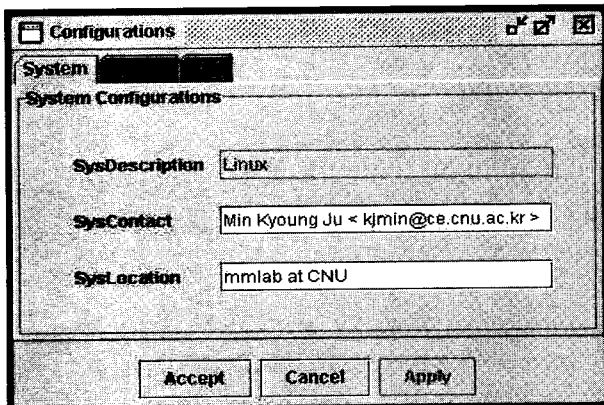
게 알리게 된다. 이러한 트랩이 발생한 것을 매니저가 브라우저를 통해 본 트랩 발생 상황이 (그림 11)에 나타나 있다.



(그림 11) 트랩 메시지

3.3.4 정보의 변경

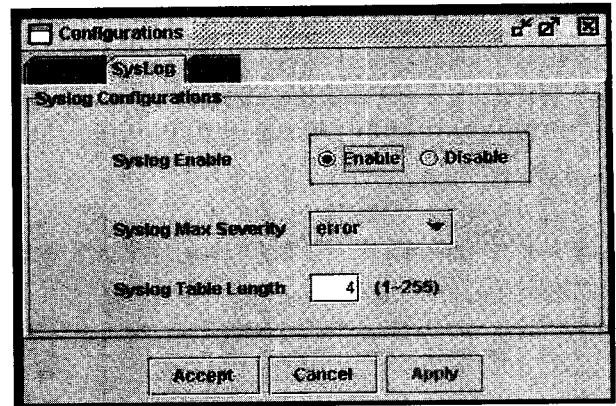
매니저가 SET-REQUEST를 통해서 에이전트의 정보를 변경하는 것을 의미한다. SET-REQUEST는 요청한 사용자가 망 관리 시스템의 사용 권한을 갖는지에 대한 검사를 하고, 인증된 사용자만이 사용할 수 있다. SET-REQUEST는 GETREQUEST와의 역할을 공유하도록 설계되었다. 이러한 요청에 대한 예가 다음의 (그림 12)에 나타나 있다.



(그림 12) 정보의 변경

SET-REQUEST를 실행하게 되면 위의 그림과 같이 이전 에이전트가 가지고 있는 정보를 화면에 보여주게 된다. 이때 값을 변경하게 되면 Apply버튼이 활성화되어 SET-REQUEST를 수행하게 된다. 이러한 SET-REQUEST는 망 관리 시스템이 처음 설정될 때는 최초로 NULL값을 화면에 보여주게 되고, 관리자에 의해 입력된 정보가 에이전트에게 반영이 된다.

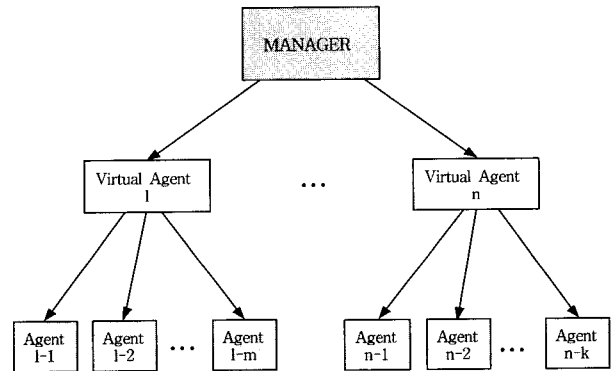
SET-REQUEST의 대표적인 사용은 트랩과 연관성을 갖는다. 에이전트에게 문제가 발생했을 때 트랩을 발생시키는 수준과, 관리하는 트랩의 테이블 개수등에 관한 정보를 설정할 수 있다. 이러한 예가 (그림 13)에 나타나 있다.



(그림 13) 시스템 로그의 설정

3.4 확장성을 고려한 설계

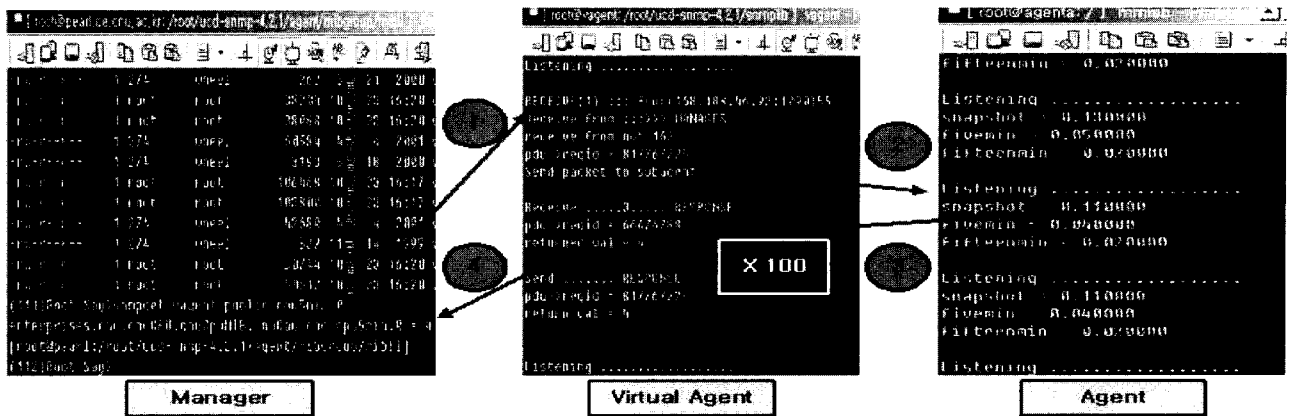
기존의 SNMP를 통한 망 관리는 매니저와 에이전트가 직접 연결된 구조를 갖게 된다. 이는 매니저가 다수의 에이전트를 관리할 때 매니저에게 과도한 부하를 야기하게 된다. 이를 해결하기 위해 매니저와 에이전트 사이에 가상 에이전트를 두는데, 이러한 계층적 SNMP 관리 모델이 (그림 14)에 나타나 있다.



(그림 14) 확장성을 고려한 관리 모델

3.4.1 정보의 요청 - 응답

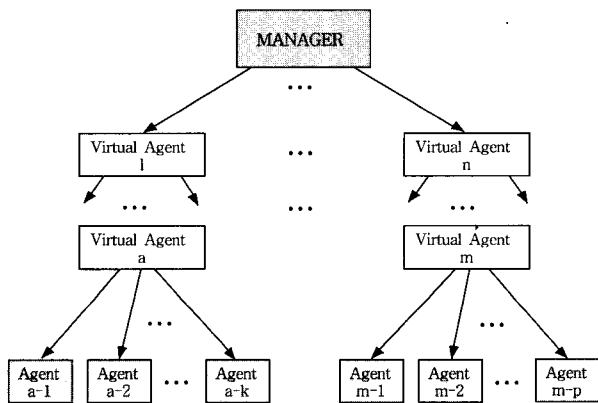
매니저는 가상 에이전트를 통해 하위 에이전트에게 메시지를 요청하게 된다. 또한 에이전트는 기존의 직접연결 모델에서와 같이 아무런 변화가 없어 투명성을 보장할 수 있다. 이는 매니저가 직접 에이전트를 관리할 수 있음을 의미한다. 실험 모델에서는 CNU-CPU-MIB를 통해 실험되었다. 매니저가 가상 에이전트에게 정보를 요청하면 가상 에이전트는 하위 에이전트에게 같은 메시지를 요청하게 된다. 요청에 대해 응답 받은 에이전트의 정보를 가상 에이전트가 취합하여 매니저에게 전달하게 된다. 정보가 숫자인 것에 대해 실험 모델에서 사용하였는데, 각각의 에이전트 정보에 대한 평균값을 전달하게 된다. 이러한 매니저-가상 에이전트-에이전트 간의 메시지 전달 과정이 (그림 15)에 나타나 있다.



(그림 15) 가상 에이전트를 통한 메시지의 전달

3.4.2 트랩 필터링

확장성 관리 모델에서 가상 에이전트를 두는 가장 큰 이유는 에이전트로부터 발생하는 트랩으로 인한 과도한 부하를 줄이고, CNU-SYSLOG-MIB에서 설정된 위험 수준으로 인해 무시되는 트랩을 관리하기 위해서이다. 에이전트가 매니저로 보내는 모든 트랩을 가상 에이전트가 먼저 받게 된다. 이에 대해 가상 에이전트는 매니저에 의해 설정된 값 이상에 대해서만 트랩을 매니저에게 전달하게 된다. 이렇게 트랩을 중간에서 필터링 함으로써, 매니저를 하나만 두고 관리가 가능하게 된다. 이렇게 토폴로지에 따라 그룹핑하여 통합 관리함으로써 다음의 (그림 16)와 같은 모델로의 확장을 가능하게 한다.



(그림 16) 확장 가능한 관리 모델

이러한 확장 가능한 관리 모델은 현대와 같이 망의 규모가 급증하는 현실에 확장성의 문제를 해결할 수 있게 된다.

4. 결론 및 향후 연구과제

우리는 본 논문에서 효율적인 망 관리를 위해 피 관리 에이전트를 설계하고, 이러한 에이전트를 효율적으로 관리하기 위한 망 관리시스템, 망의 규모가 급증하는 현실에 대

응하기 위한 방안으로 가상 에이전트를 두는 모델의 프레임워크를 구현하여 보았다. 리눅스에서 설계된 에이전트는 빠른 개발환경과 경제성을 갖고, 자바-SNMP는 플랫폼에 독립성을 제공한다. 이러한 망 관리 시스템은 기존의 텍스트 기반에서 사용자 관점의 망 관리를 가능하게 하고, 가상 에이전트 관리 모델을 통한 망 관리는 확장성의 문제를 해결할 수 있음을 보였다.

향후 과제로는, 원하는 환경에 맞는 사설 MIB의 설계와 모듈의 구현이 필요하다. 또한 에이전트는 리눅스 플랫폼만을 대상으로 하고 있어 라우터, 허브와 같은 실제 네트워크 디바이스에 적용하기 위해 플랫폼에 적합한 관련 연구가 더욱 진행되어야 한다.

참고 문헌

- [1] 김건웅, 김익배, 진명숙, 송병권, 안순신, "KU-NMS : SNMP를 이용한 전산망 관리 시스템", 한국정보과학회논문지 제22권 제3호, pp.454-461,
- [2] M. J. Wren and J. A. Gutiérrez, "Agent and Web-based Technologies in Network Management," IEEE Global Telecommunications Conference, Vol.3, pp.1877-1881, December, 1999.
- [3] J. Yang, P. Wang and J. Wu, "A scalable, Web-based Architecture for Hierarchical Network Management," IEEE Global Telecommunications Conference, Vol.3, pp.1882-1888, December, 1999.
- [4] H. J. Lin and C. H. Wang, "Distributed Network Management by Http-based Remote Invocation," IEEE Global Telecommunications Conference, Vol.3, pp.1889-1893, December, 1999.
- [5] M. T. Rose, "The Simple Book- An Introduction to Management of TCP/IP-based Internets," Prentice Hall Inc., 1991.
- [6] W. Stalling, "SNMP, SNMPv2, and CMIP," Addison Wesley Publishing Complay, 1993.

[7] A. B. Artzi, A. Chandna, and U. Warriar, "Network Management of TCP/IP Networks : Present and Future," IEEE Network Magazine, pp.35-42, July, 1990.

[8] S. M. Klerer, "The OSI Management Architecture : an Overview," IEEE Network, Vol.2, No.2, pp.20-29, March, 1988.

[9] RFC1155, "Structure of Management Information(SMI)".

[10] RFC1157, "Simple Network Management Protocol(SNMP)".

[11] RFC1212, "Concise MIB definitions".

[12] RFC 1213, "Management Information Base(MIB-II)".

[13] D. Perkins and E. McGinnis, "Understanding SNMP MIBs," Prentice Hall, 1997.

[14] Net-SNMP, <http://net-snmpp.sourceforge.net/>, November, 2001.

[15] AdventNet, <http://www.adventnet.com/products/index.html#nms>, November, 2001.

[16] FOUNDRY NETWORKS Inc., "IronView Network Management User's Guide," FOUNDRY NETWORKS, September, 2000.

[17] CISCO Systems, <http://www.cisco.com/warp/public/477/SNMP/snmp-indx.html>, November, 2001.



민 경 주

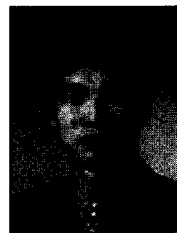
e-mail : meanrace@etri.re.kr

2000년 충남대학교 컴퓨터공학과 졸업
(학사)

2002년 충남대학교 대학원 컴퓨터공학과
(공학석사)

2002년~현재 한국전자통신연구원

관심분야 : 망 관리, 통신 프로토콜 표준 연구, VoIP, 초고속 통신, 멀티미디어 통신



권 택 근

e-mail : tgkwon@ce.cnu.ac.kr

1988년 서울대학교 컴퓨터공학과(학사)

1990년 서울대학교 대학원 컴퓨터공학과
(공학석사)

1996년 서울대학교 대학원 컴퓨터공학과
(공학박사)

1992년~1998년 LG 전자 R&D 센터 연구원

1998년~현재 충남대학교 정보통신공학부 조교수

관심분야 : 멀티미디어 통신, 멀티미디어 서버, 초고속 통신망, 시스템 소프트웨어