윈도우 기반의 네트워크 관리 시스템 설계 및 구현

김진천*

A Design and Implementation of Windows-Based Network Management System

Jin-chun Kim*

이 논문은 2002년도 경성대학교 특별과제연구비에 의하여 연구되었음

요 약

오늘날 정보통신 기술의 발전에 따라 기업, 공공기관 뿐만 아니라 교육기관 등에서 네트워크를 이용한 업무가 증가되고 있으며 이러한 업무의 증가에 따라 네트워크의 관리가 필요한 실정이다. 하지만 대부분의 네트워크 관리를 위한 소프트웨어가 네트워크에 대한 전문 지식을 필요로 하거나 유닉스를 기반으로 하여 잘 사용되지 않는 고급 기능들까지 포함하고 있어 효율적이지 못하다. 이에 본 연구에서는 비전문가라 하더라도 손쉽게 사용할 수 있으며 네트워크 관리를 위한 데이터베이스 기능과 네트워크 관리에 꼭 필요한 기능만을 포함하는 마이크로소프트사의 윈도우 기반의 저가형 네트워크 통합 관리 시스템을 설계 및 구현하였다.

ABSTRACT

Network management has become very important issue due to the broad usage of network. However most of network management software are based on UNIX and very complicated to use. Therefore a simple and compact Network Management System which can be used easily on PC environment is needed.

In this paper we address the design and implementation of windows-based network management system which can be easily used.

키워드
망 관리 시스템, 윈도우 기반, SNMP, TCPIP, NMS

Ⅰ. 서 론

오늘날 정보통신기술의 발달에 따라 인터넷에 대한 관심과 활용이 폭발적으로 증가하고 있다. 따라서 IP를 할당하거나 네트워크 장비의 사용 상황을 모니터링 하는 등의 네트워크 관련 업무의 필요성이 부각되고 있고 그 업무량 또한 방대하기 때문에 네트워크를 보다 효율적이고 손쉽게 관리 할 수 있는 네트워크 관리 시스템의 요구가 증가되고 있다.

현재 대부분의 네트워크 관리 시스템은 유닉스 기반으로 네트워크에 대한 학문적인 지식을 필요로 하며 일반적으로 잘 사용되지 않는 고급 기능까지 많이 포함하고 있어 사용법이 어렵기 때문에 효율적으로 사용되지 못하고 있다. 이에 PC를 기반으로 하여 비전문가라 하더라도 손쉽게 사용할 수 있으며 네트워크 관리를 위한 데이터베이스 기능과 네트워크 관리에 꼭 필요한 기능만을 포함하는 저가형 네트워크 관리 시스템을 본 연구에서 설계 및 구현하였다.

*경성대학교 전기전자 · 컴퓨터공학부

접수일자 : 2003. 2. 7

II. 관련 연구

2.1 Network Management System

NMS는 한 지점에서 네크워크에 있는 장비들의 구성 및 상태를 확인, 관리하고 나아가 분산되어 있는 여러 네트워크를 통제하는 시스템으로 일반적으로 표 1. 과 같은 주요 기능을 포함한다.

표 1. 네트워크 관리 시스템의 주요 기능
Table. 1 major function of NMS

기능 영역	기 능
실행 관리	가용성, 응답시간, 사용량, 에러량, 처리속도등 성능 분석에 필요한 통계 데이터를 제공하는 기능
구성 관리	네트워크상의 장비와 전반적인 물리적 구조를 Mapping하는 기능
계정 관리	각 노드별로 사용 현황을 측정하는 기능
결함 관리	문제의 검색, 추출 및 해결을 제공하는 기능
보안 관리	정보의 제어 및 보호 기능

2.2 Simple Network Management Protocol

네트워크 관리에 이용되는 프로토콜에는 SNMP[1], CMIP 등이 있으며 이 중에서 SNMP는 구현이 쉽고 간단해 오늘날 가장 일반적인 네트워크 관리 프로토콜로 사용되고 있으며, CMIP는 구현의 복잡성, 방대함으로 인해 아직도 네트워크 관리의 중심으로 자리잡지 못하고 있다.

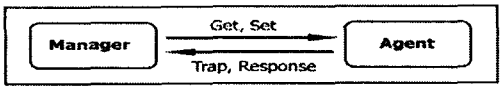

그림 1. SNMP 기본통신
Fig. 1 Basic Communication of SNMP

SNMP는 SGMP(Simple Gateway Monitoring Protocol)[2]에 기초를 두고 있으며 UDP(User Datagram Protocol)상에서 작동하는 비동기식 요청/응답 메시지 프로토콜로서 기본적으로 3가지 기능만을 수행한다. 그림 1에서 SNMP가 수행하는 3가지 기능은 아래와 같다.

- **GET** : 에이전트의 MIB의 값을 가져온다.
- **SET** : 에이전트의 MIB의 값을 변경한다.
- **TRAP** : 에이전트의 특정 상황 발생을 관리자에게 알린다.

SNMP는 관리자의 물음에 에이전트가 응답하는 것이 기본이고 에이전트는 특수한 상황이 발생한 경우에 관리자에게 발생을 알리고 필요한 정보를 함께 보낸다. SNMP기반의 TIP/IP 네트워크 관리 모형은 그림 2. 와 같이 구성되어 진다.

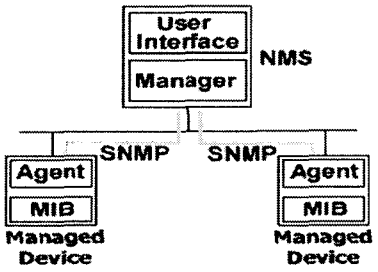

그림 3 SNMP기반의 TCP/IP 네트워크 관리 모형
Fig. 2 SNMP based TCP/IP 네트워크 Management Model

2.3 Management Information Base

네트워크 관리에는 정보 저장 및 검색 메커니즘이 요구된다. TCP/IP 기반의 네트워크를 관리할 때 이와 같은 역할을 하는 것이 MIB[3]이다. MIB는 관리되는 네트워크 요소의 각 오브젝트들의 계층적이며 구조적인 집합을 의미한다. 이런 MIB은 네트워크 장비들의 운영 체제가 유지하는데 NMS는 이런 MIB의 오브젝트 값을 읽음으로써 노드의 자원을 감시하고, 그러한 값들을 변경함으로써 노드의 자원을 조정할 수 있다.

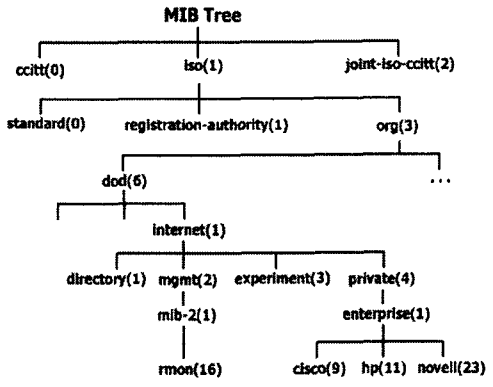

그림 3. MIB 오브젝트트리
Fig. 3 MIB Object Tree

MIB의 각 오브젝트는 계층적인 OBJECT IDENTIFIER로서 식별되며 명명 규약은 계층적인 오브젝트 형태의 구조를 명확히 나타낼 수 있어야 한다. MIB의 각 오브젝트들은 정수의 연속된 나열로 표시된다. 그림 3은 MIB 오브젝트 트리의 일부이다.

II. 시스템 설계

3.1 시스템 모듈 구성

본 연구에서 구현된 시스템의 주요 모듈은 그림 4와 같이 네트워크 모듈, 데이터베이스 모듈, 로그 모듈, 모니터 모듈로 나누어진다. 네트워크 모듈에서 수집한 네트워크 상황에 대한 정보는 데이터베이스 모듈에서 저장 및 기존의 데이터와 통합되어 분석되고 분석된 결과는 로그 모듈에 의해서 저장되어 진다. 또 현재의 상황을 계속해서 추적 감시하는 모니터 모듈에 결과를 전달하게 되는데 모니터 모듈에서는 이를 시각적인 방법으로 도식화하여 현시점의 네트워크 상황을 정확하게 리포팅한다.

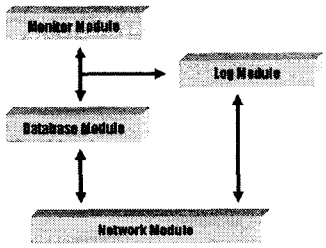

그림 4. 주요 모듈 구성도
Fig. 4 Major modules

3. 2 시스템 모듈별 기능

1) Network Module

네트워크 상에 있는 각종 장치들에 관한 정보를 수집하는 모듈이다. 사용되고 있는 IP 나 각종 장치들의 종류 및 특성에 관한 자료뿐만 아니라 각 장비의 문제의 발생 여부까지도 다양한 방법을 이용하여 수집한다. 이렇게 수집한 자료는 데이터 베이스 모듈에 전달되어 진다.

2) Database Module

Network Module로부터 입수한 네트워크 장치들에 관한 정보를 저장 및 분석하는 모듈이다. 각각의 장비들에 대해서 입수한 정보를 통해 문제의 발생 여부 및 현 상태를 진단한다. 또한 네트워크의 복잡한 연결상태를 시각적으로 도시화하기 위한 상대적인 위치에 관한 분석도 이루어진다.

3) Log Module

네트워크 상의 문제나 각종 이벤트를 저장하여 문제를 해결할 수 있는 정보를 제공한다. 네트워크 모듈로부터 전달받은 긴급한 이벤트나 데이터베이스 모듈에 의해 분석되어진 중요한 결과를 저장하여 관리하고 있는 네트워크에 대한 시간적인 기록을 남긴다.

4) Monitor Module

데이터베이스 모듈로부터 분석된 결과를 입수하여 네트워크 장비들의 상태를 감시하고 각 장비의 트래픽 등을 시각적으로 도식화하는 모듈이다.

3.3 시스템 모듈별 세부 설계

앞에서 설명한 4가지의 모듈은 그림 5. 와 같이 다시 세분화되어 각각의 세부모듈로 구분되어 질 수 있고 각각의 모듈은 객체지향 프로그래밍에서 하나의 클래스로 구성되어 진다.

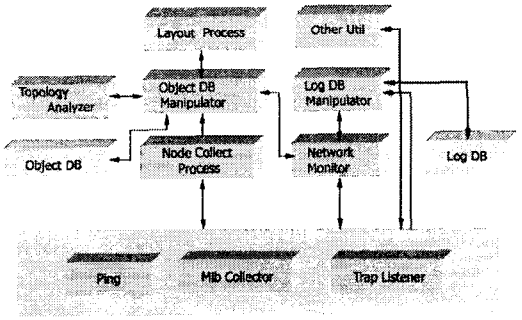

그림 5. 구현을 위한 모듈 구성도
Fig. 5 Modules for implementation

네트워크 상에 있는 각종 장치들에 관한 정보를 수집하는 모듈인 네트워크 모듈은 Ping과 Mib Collector 그리고 Trap Listener를 나타낸다.

Ping은 네트워크 상에서 각 에이전트의 존재 여부를 결정하는 데 사용되어지며 Node Collector Process는 MIB Collector를 통해 각 SNMP 에이전트에 MIIB값을 쿼리 하고 에이전트의 응답을 검사하여 에이전트의 추가, 삭제, 변경을 동적으로 행한다.

이때 Node Collector Process에 의한 에이전트 정보는 Object DB Manipulator에게 전달되고 이 데이터들은 Topology Analyzer를 통해 각 에이전트 또는 오브젝트들이 맵상에서의 위치를 부여받게 되며 이러한 정보들은 Object DB에 저장 될 뿐만 아니라 Layout Process에게도 전달되어 맵을 그리게 된다.

Net Monitor역시 MIB Collector를 통해 MIB값을 쿼리하고 응답을 검사하여 Log DB Manipulator를 통해 데이터를 Log DB에 저장하게 된다.

Trap Listener는 현재 NMS가 설치 되어 있는 시스템으로 전달되는 트랩 메시지를 전달 받는 기능을 한다. Trap Listener를 통해 수신된 메시지는 Log DB Manipulator를 통해 Log DB에게 전달된다.

Network Module로부터 입수한 네트워크 장치들에 관한 정보를 저장 및 분석하는 모듈인 데이터베이스 모듈은 Object DB Manipulator, Log DB Manipulator 그리고 Object DB, Log DB로 구분 될 수 있다. Object DB Manipulator에 의해 처리되는 정보는 맵의 갱신과 유지를 위한 정보들이며 Log DB Manipulator에 의해 처리되는 정보는 사용자의 확인 또는 처리를 요하는 정보들이다. 즉, Log DB상에 저장 되어지는 데이터들은 warning, 또는 critical 상태를 의미하는 데이터 들이며 Trap Listener로부터 전달 받게 되는 trap 메시지 또한 포함한다.

Net monitor에 의해 생성되는 데이터들은 Log db와 object db 양쪽 모두에 저장되어 질 수 있으며 object db에 저장되어지는 정보들은 맵의 갱신과 유지를 위한 정보들이다.

다음은 각 데이터베이스의 스키마이다.

1) Log DB

표 2. 로그 데이터베이스
Table 2. Log Database

log_source	status	source ip	identifier	value

- log_source : log의 제공지가 Net monitor인지 trap listener 인지를 구별한다.
- status : log의 상태 정보를 나타내며 normal, warning, critical의 3가지가 있다. 각각의 정의는 본 문서에서 로그표시기를 논의할 때 설명한다.
- source_ip : log가 발생된 에이전트의 ip를 나타낸다.
- identifier : log 식별자로써 본 문서의 로그표시기를 논의 할 때 설명한다.
- value : value는 실제 로그의 값이다.

2) Object DB

표 3. IP 테이블
Table 3. IP Table

obj_id	ip	man_status	ip_operation

- obj_id : 네트워크상의 오브젝트의 아이디이며 각 오브젝트들은 이 아이디로 식별 된다.
- ip : ip이며 한 오브젝트가 복수개의 ip를 가질 수 있다.
- man_status : IP의 관리 여부를 나타낸다.
- ip_operation : IP의 동작 여부를 나타낸다.

표 4. 오브젝트 테이블
Table 4. Object Table

obj_id	obj_name	obj_type	obj_level	obj_pos

op_status	man_status	view_status	agent

- obj_id : 네트워크상의 오브젝트의 아이디이며

각 오브젝트들은 이 아이디로 식별 된다.

- *obj_name* : 각 오브젝트들에게 부여 되어진 이름이다.
- *obj_type* : 오브젝트 타입을 나타내며 세그먼트, 라우스, 스위치, 허브, pc가 있다.
- *obj_level* : 오브젝트의 맵상에서의 레벨을 나타낸다.
- *obj_pos* : 오브젝트의 맵상에서의 위치를 나타낸다.
- op_status : 오브젝트가 어떠한 상태에 있는지를 알려준다. 오브젝트가 가질 수 있는 상태는 *normal, warning, critical*의 3가지 상태가 있다.
- man_status : 오브젝트의 management 여부를 나타낸다.
- view_status : 오브젝트가 맵상에서 보여지는지의 여부를 나타낸다.
- agent : 오브젝트가 SNMP agent인지의 여부를 나타낸다.

Layout Process는 Object DB의 데이터를 Topology Analyzer가 처리하여 맵상에서의 위치를 결정하게 되면 이를 실제적으로 맵상에 처리하는 역할을 하는 모듈이다. 또한 사용자의 맵 편집시 이 데이터들을 Object DB Manipulator를 통해 Object DB에 전달하는 모듈이다.

Node Collect Process는 맵을 그리기 위한 Mib 데이터 및 에이전트의 정보를 수집하는 모듈이다. 프로그램 실행시 Node Collect Process가 동작하여 각 오브젝트들의 정보를 수집하기 시작한다.

각 오브젝트의 정보 수집을 위한 동작은 최초 ping 테스트 부터 시작된다. 일단 ping을 테스트한 후 응답이 없는 오브젝트들은 제외된다. 하지만 node collect process는 계속적으로 맵의 유지와 갱신을 위해 정보를 수집 하기 때문에 한번 제외 된다고 해서 다시 맵상의 오브젝트로 나타내어 질 수 없는 것은 아니다.

ping 테스트에 응답이 있는 오브젝트들에 대해서는 다음으로 SNMP agent가 서비스 되고 있는지를 살펴 본다. 이때 SNMP agent가 운영되고 있지 않으면 이 오브젝트는 PC로 간주한다.

만약 운영되고 있다면 이 오브젝트는 네트웍 장비이다. SNMP agent가 운영되고 있다면 MIB 값들을 쿼리해서 얻음으로써 각 오브젝트들의 구분이 가능한데, 만약 MIB 값중 system.sysServices라는 이름을 가진 MIB가 존재한다. 이 MIB 값이 1로 세팅되어 있으면 이 오브젝트는 허브로 간주한다.

그렇지 않다면 다음 구분으로 넘어가게 되며 ip.ipAdEntAddr 이라는 MIB의 값은 현재 대상 오브젝트가 가지고 있는 IP의 리스트를 나타 내는데 이 값중 0.0.0.0, 127.0.0.1을 제외한 IP가 하나이면 이 오브젝트는 스위치이고 복수개이면서 MIB IP그룹 중 현재 오브젝트가 gatway로 사용 되는지 여부를 나타내는 ipForwarding의 값이 1이면 라우터이다. 그리고 라우터의 IP 테이블과 넷 마스크를 조사하여 세그먼트를 확장해 나간다.

이렇게 하여 얻어진 오브젝트의 값은 절대적일 수는 없지만 네트워크 관리자의 오판이나 실수가 아니라면 정확한 오브젝트의 유형을 얻을 수 있다. 이렇게 도출한 오브젝트의 유형을 기준으로 Object DB에 현재 조사하는 오브젝트의 값을 저장하게 된다.

IV. 시스템 구현

본 연구를 통해 구현된 시스템은 메인 프로그램과 MIB 컴파일러, MIB 탐색기, 트래픽 모니터, 로그 표시기, 기타 유틸리티로 구성 되어 진다. 본 장에서는 각 프로그램의 구현 결과와 기능에 대해 설명한다.

4.1 메인 프로그램

메인 프로그램은 네트워크에 존재하는 모든 자원을 맵이라는 구조로 통합하여 관리한다. 맵은 오브젝트들의 집합으로 구성된 정보이며, 관리 대상이 되는 네트워크의 토폴로지를 구조화하여 그래픽으로 나타내고 포함된 오브젝트를 아이콘으로 형상화하여 표시한다. 아이콘화 된 오브젝트는 작동 상태에 따라 서로 다른 색상으로 표시되어 사용자가 쉽고 편리하게 네트워크의 운용 상태를 인지하도록 하는데 도움을 준다.

맵은 네트워크에 있는 노드들을 IP 주소 체계에 따라 그룹화 하여 이를 대표하는 오브젝트라는 개념

으로 표현된다. 오브젝트는 네트워크 주소와 네트워크 마스크에 따라 식별되는 세그먼트 와 라우터, 스위치, 허브, PC와 같은 실제 물리적 장비를 나타내는 오브젝트로 표현된다.

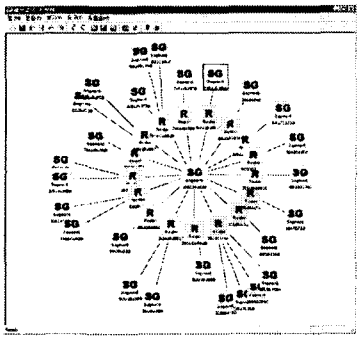

그림 6. 메인 프로그램
Fig. 6 Main Program

메인 프로그램은 자동으로 오브젝트들을 실시간으로 네트워크의 상태를 감시 할 수 있으며 관리자가 임의로 맵상의 오브젝트를 편집 가능하다. 또한 특정 에이전트의 접속확인과 관리를 위한 Ping, Telnet, MIB 정보 액세스를 위한 도구를 제공한다.

4.2 MIB 컴파일러

MIB들로부터 MIB Tree를 구성하기 위해서는 MIB들을 정의한 구문을 분석하여 필요한 정보만으로 요약한 데이터파일이 필요하다. 이러한 역할을 수행하는 것이 MIB 컴파일러이며 MIB 컴파일러로부터 생성되어진 자체 데이터 파일은 MIB 탐색기가 MIB 트리를 구성하는 데 사용되어 진다.

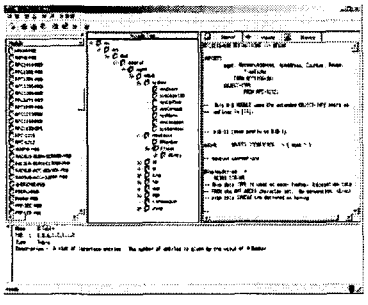

그림 8. MIB 컴파일러
Fig. 8 MIB Compiler

4.3 MIB 탐색기

MIB 탐색기는 MIB 컴파일러를 통해 컴파일 된 MIB 정보 파일을 기반으로 하여 MIB 트리를 구성하고 각 트리의 노드에 대하여 사용자가 지정한 에이전트로부터 MIB 값을 전달받아 이를 보여주는 도구로 특정 에이전트로 부터의 Trap 메시지의 Listen 도 가능하다.

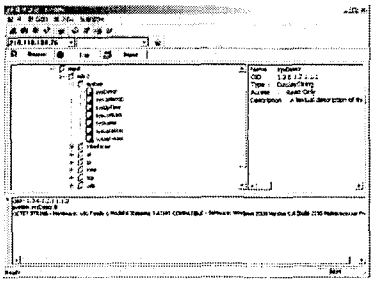

그림 9. MIB 브라우저
Fig. 9 MIB Borwser

4.4 트래픽 모니터

사용자가 특정 에이전트의 트래픽을 연속적으로 모니터 하고자 할 때 매초 단위로 이를 그래프화 시켜 주는 도구로 특별한 설정 없이 원하는 오브젝트의 in, out 패킷종류를 선택하기만 하면 이를 그래프도 나타내어 준다.

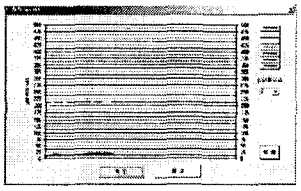

그림 10. 트래픽 모니터
Fig. 10 Traffic Monitor

4.5 로그 표시기

로그 표시기는 맵상에 나타나 있는 에이전트의 로그를 표시하며 에이전트에 이상 징후가 관찰 되었을 때 이를 일정 기간 단위로 남긴 로그를 사용자에게 보여주는 도구로 MIB Collector에 의해 수집되는 정보 및 Trap정보를 로그로 저장한다. 관리자는 로그

표시기를 통해 네트워크상의 이상 징후를 발견 할 수 있다.

4.6 기타 유틸리티

기타 도구로는 ping, telnet, traceroute가 있으며 이러한 도구는 관리자가 네트워크 관리시 꼭 필요한 기본적인 도구들로서 Windows 운영체제에서 제공되고 있는 Command line 명령어들을 GUI환경, 즉 윈도즈 기반의 프로그램으로 바꾼 것이다. 이러한 도구의 제공으로 Command line상에서 실행의 번거로움을 해결하고 한 프로그램의 실행으로 필요한 도구들을 모두 사용 할 수 있게 된다.

V. 결 론

본 연구를 통해 네트워크에 문제나 오류 발생시 네트워크 관리자가 이를 발견하고 나아가 사전에 네트워크에서 발생 할 수 있는 문제를 미리 예방 할 수 있는 NMS를 설계 및 구현 하였다.본 시스템은 기존의 유닉스 기반의 시스템들과는 달리 Windows를 기반으로 네트워크 관리에 꼭 필요한 기능만을 포함하여 비전문가라 하더라도 손쉽게 사용할 수 있으므로 중·소규모의 네트워크 관리에 적용할 수 있는 NMS이다.

참고문헌

[1] J. Case, M. Fedor, M. Schoffstall, J. Davin. "A Simple Network Management Protocol (SNMP)" RFC 1157, May1990

[2] Davin, J., J. Case, M. Fedor, and M. Schoffstall, "A Simple Gateway Monitoring Protocol", RFC 1028, Proteon, University of Tennessee at Knoxville, Cornell University, and Rensselaer Polytechnic Institute, November 1987.

[3] McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1066, TWG, August 1988.

저자소개

김진천(Jin-Chun Kim)

1983년 2월 한양대학교 전기공학과(공학사)

1985년 8월 미국 미시간주립대학교 전자 및 시스템공학과(공학석사)

1996년 2월 한국과학기술원 전산학과(공학박사)

1988년 4월~1996년 2월 삼성종합기술원 선임연구원

1996년 3월~현재 경성대학교 전기전자·컴퓨터공학부 부교수

※관심분야: 멀티미디어 통신, ATM 스위치 구조, 컴퓨터 구조, 초고속 네트웍