

능동적 관리 프로그램 생성 기능을 지원하는 네트워크 관리 프레임워크 설계

김 은 희[†] · 이 명 진^{**} · 류 근 호^{***}

요 약

네트워크의 규모가 확장되면서, 다양한 네트워크 장비들로 구성되어 있는 복잡한 네트워크 관리를 위해 많은 비용이 소모되고 있다. 이 논문에서는 네트워크 구성 요소 및 서비스 관리를 위해 필요한 관리 프로그램의 개발비용 및 시간을 절감하기 위하여 능동적 관리 프로그램 생성 기능을 지원하는 네트워크 관리 시스템을 제안한다. 제안한 시스템은 실시간 네트워크 관리를 위해 구성관리, 성능관리 및 장애관리의 기본 구성요소와 이들 기본 구성요소를 통해 관리 대상이 되는 네트워크 구성요소들에 대한 정보 생성을 능동적으로 지원해주기 위한 능동적 관리 프로그램 생성기로 구성된다. 제안하는 시스템은 네트워크 구성 요소들에 대한 정보를 가지고 SNMP 매니저와 함께 네트워크 관리를 수행하게 되는데, 이때 필요한 네트워크 구성요소들에 대한 관리 프로그램 생성을 기존의 수작업에 의한 방법에서 능동적으로 생성해 준다. 우리는 제안하는 시스템을 실제 네트워크에 구축하여 운영함으로써 네트워크 객체에 대한 확장 또는 네트워크 관리 객체에 대한 관리 프로그램 생성 시 개발 기간을 단축하는 효과와 개발비용 및 유지비용이 감소되는 것을 확인하였다.

키워드 : 네트워크 관리 시스템, 네트워크 관리 프로그램

A Design of Network Management Framework employing active management program generation facilities

Eun Hee Kim[†] · Myung Jin Lee^{**} · Keun Ho Ryu^{***}

ABSTRACT

As the scale of network expands largely, most of users consume much cost for complex network that composes various network environments. In this paper, we propose a network management framework employing active management program generation facilities in order to reduce the cost and time for developing a network management program. The proposed framework consists of several basic components including configuration management, performance management, and fault management for real time network management. Besides those components, it has an added active program generator in order to actively create information of network objects that are managed through the basic components.

Our framework manages the network with SNMP manager using the information of network objects. This information is generated automatically instead of the existing manual manner. We shows that the cost and time for developing a network management program is reduced through the construction and operations on the real network.

Key Words : Network Management System, Network Management Program

1. 서 론

초고속 통신망의 보급과 네트워크 환경의 급속한 변화로 기하급수적으로 네트워크 장비가 증가되고 있으며, 정보 시스템의 확장과 네트워크 기술의 발달로 네트워크는 점차적

으로 진화되어 왔으며, 이에 따라 다양한 네트워크 장비들이 대규모 네트워크 환경을 구성하게 되었다. 또한 네트워크 서비스의 품질 향상을 위해서 이러한 장비들에 대한 관리 및 유지 보수는 필수적이다. 이러한 네트워크 환경은 대부분 구조가 간단하고 구현이 용이한 SNMP(Simple Network Management Protocol)를 기반으로 관리되고 있다 [1, 2, 3]. SNMP 기반의 네트워크 관리는 네트워크 환경의 규모가 복잡해지고 다양해짐에 따라 구성 관리와 관리 도구 개발 과정에서 많은 문제를 야기한다[4]. 매니저/에이전트 기반의 구조를 가지는 SNMP 네트워크 관리 시스템은 집중

* 이 연구는 정보통신부 대학 IT연구센터 육성 지원사업의 연구비 지원으로 수행되었음

† 준 회원 : 충북대학교 전자계산학과 박사 수료

** 정 회원 : 가림정보기술(주) 대표이사

*** 통신회원 : 충북대학교 전기전자컴퓨터공학부 교수(교신저자)

논문접수 : 2005년 9월 26일, 심사완료 : 2006년 8월 29일

화된 관리 체계로 인하여 하나의 매니저가 관리할 수 있는 범위가 제한적이다[5, 6, 7, 8]. 그리고 네트워크 관리 시스템과 관련된 트래픽은 실제 사용자들에게 제공되는 서비스를 위한 트래픽이 아님에도 불구하고 사용자들에게 제공되는 서비스에 큰 오버헤드를 야기할 수 있으며, 네트워크 관리를 위해 요구되는 CPU와 메모리 등의 하드웨어 비용이 증가하는 문제를 갖는다. 기존의 SNMP를 기반으로 한 시스템들은 네트워크 관리를 위해 네트워크 관리 대상 객체에 대한 관리 프로그램의 개발 작업을 선행해야 한다. 그러나 기존 시스템들의 네트워크 관리 프로그램들은 개발 작업에서 다음과 같은 몇 가지 문제점들을 가진다. 첫째, 수작업으로 인한 개발비용 및 개발 시간의 증가 문제이다. 네트워크에서 관리 대상이 되는 네트워크 구성 요소들은 네트워크의 상태 변화에 따라 추가, 변경 또는 삭제 될 수 있다. 이러한 구성 요소의 변경이 발생하는 경우 관리 대상이 되는 구성 요소에 대한 관리 정보는 수정되거나 새롭게 생성되어야 한다. 기존의 네트워크 관리 시스템은 구성 요소 변경에 따른 관리 프로그램을 수작업으로 생성하기 때문에 개발비용의 증가와 개발 시간의 지연을 초래한다. 둘째, 개발된 네트워크 관리 대상 객체를 관리하기 위한 프로그램은 수작업으로 인한 많은 오류를 포함하게 된다. 즉, 수작업으로 발생할 수 있는 철자 오류, 문법적인 오류, 논리적인 오류 등을 수정하는 데 많은 시간이 소요되는 실정이다. 셋째, SNMP와 관련된 객체에 대한 정보 및 연산 제공이 미흡하다. SNMP와 관련된 정보는 네트워크 장비를 생산하는 제조업체에서 관리 정보 베이스(MIB: Management Information Base) 구조로 제공된다[6, 8]. 그러나 네트워크의 운영 과정에서 기본적인 정보 외에도 구성된 네트워크에 필요한 정보를 추가해야 하는 경우가 종종 발생한다. 이러한 문제를 해결하기 위하여 제조업체에서 제공하는 기본적인 MIB 정보 외에 네트워크 구성 환경에 맞는 관리 정보를 추가적으로 제공하여야 한다. 이와 같은 문제점들을 해결하기 위하여, 이 논문에서는 네트워크 장비의 관리를 위해 능동적 관리 프로그램 생성 기능을 지원하는 네트워크 관리 프레임워크를 제안한다.

이 논문을 효율적으로 기술하기 위하여 다음과 같이 구성하였다. 2장에서는 네트워크 관리 기법에 관한 기존 연구들을 고찰한 후, SNMP 기반 네트워크 관리 시스템의 문제점 및 해결 방안을 기술한다. 3장에서는 능동적 관리 프로그램 생성 기능을 지원하는 네트워크 관리 프레임워크를 제안하며, 제안하는 프레임워크의 설계 및 각 구성 요소들에 대해서 기술한다. 4장에서는 제안한 프레임워크의 구현과 실제 네트워크 환경에서의 운용을 통해 유용성을 분석한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

이 장에서는 기존의 대표적인 SNMP 네트워크 관리 시스템과 현재 상용화 되어있는 네트워크 관리시스템들의 문제점을 분석하고 해결 방안을 제안한다.

SNMP 기반의 네트워크 관리 시스템은 매니저/에이전트 기반 구조로서 매니저는 에이전트에게 관리에 필요한 대상에 대한 정보를 요청 하게 되고, 이에 에이전트는 모듈에 정의된 절차를 수행한 결과를 매니저에게 응답하게 된다. SNMP 기반의 네트워크 관리 시스템은 많은 장점을 가지고 있지만 대규모 네트워크 관리에 있어서는 구성요소와 관련된 정보를 전송하는 능력에 한계를 드러냈다. 이러한 단점을 보완하여 거대한 네트워크에서 생산되는 많은 관리 데이터를 효율적으로 전달하고 처리하기 위한 방안으로 XML을 네트워크 관리에 이용하려는 연구들이 시도되고 있다. XML 기반 네트워크 관리는 SNMP MIB을 XML로 재 정의하고 관리를 위한 데이터의 전달을 XML 문서의 형태로 주고받는다[4]. 네트워크 장치를 관리하기 위해 많은 장치들은 내장된 웹 서버를 이용하고 있다. 이러한 내장 웹 서버를 장치 관리에서 네트워크 관리로 확장하여 이용하려는 연구가 진행되었다[4, 10]. 이들 연구에서는 XML을 이용해서 데이터를 전달하는 방법이 제안되었으며, 웹 서버를 내장하고 있는 에이전트인 WBM(Web Based Management)[9]을 사용하여 네트워크 장치 관리를 한다. 이들 시스템은 복잡한 네트워크 구조를 관리할 때 표준화된 데이터 처리 및 전달 방법을 제공하는 등의 많은 장점을 가지고 있지만, 이들 시스템은 기존의 SNMP 에이전트만이 장착된 장비를 관리하는 방안을 제공하지 못한다. SNMP 에이전트는 인터넷 전반에 걸쳐 널리 사용되고 있지만, 모든 에이전트들이 XML을 지원하는 에이전트인 내장 웹 서버 에이전트로 변경하는 것은 불가능하다. Martin-Flatin[10]은 웹 기반의 통합 네트워크 관리 구조에 관한 연구로서, 데이터 통합을 위한 방법으로 XML을 네트워크 관리에 사용하는 방법을 제시하고 있다. 하지만, XML 기반 매니저의 장점을 이용하면서 기존의 SNMP 에이전트를 변경하지 않고 통합해서 관리하는 방법이 요구되고 있다.

지금까지 제안된 중·소규모의 네트워크 환경을 관리하기 위한 대표적인 네트워크 관리 시스템으로는 WhatsUp Gold[11], SolarWinds Engineers Edition[12], VisualRoute[13] 등이 있다.

WhatsUp Gold는 윈도우즈 NT 서비스와 같이 구동되며, 시스템의 접근을 위하여 웹 콘솔만을 사용한다. 이러한 웹 콘솔에 의한 접근은 완벽한 네트워크 관리 시스템과 시스템을 구성하는 NT 서비스로 운영된다. SolarWinds Engineers Edition은 강력한 네트워크 도구로 핑 모니터(ping monitor)와 서브넷 계산기 등 단순한 기능에서부터 훨씬 복잡한 성능 관리 및 주소 관리 기능에 이르는 수많은 기능들을 제공한다. VisualRoute는 각 홉(hop)의 다음 측정에 대한 경로 추적을 실행함으로써 경로에서 발생하는 오류를 진단한다.

지금까지 진행되어온 SNMP 기반의 네트워크 관리 시스템들은 데이터 전송 방식에 초점을 맞추어 연구가 진행되어왔다. 그러나 네트워크 기술이 발달함에 따라 새롭고 다양한 네트워크 구성 요소들이 지속적으로 출현하고 있기 때문에 네트워크 관리 시스템 내에서 새로운 네트워크 구성 요

소를 관리하기 위한 네트워크 관리 프로그램의 개발이 필수적이다. 특히 네트워크 관리 프로그램의 추가 개발 및 유지/관리를 위해 소요되는 비용이 전체 네트워크 관리 시스템의 유지/관리 비용에 큰 영향을 준다. 기존의 네트워크 관리 시스템들은 새로운 네트워크 구성 요소가 등장할 때 마다 수작업에 의해 네트워크 구성 요소를 관리하는 프로그램을 개발하고 있다. 또한 네트워크 장비의 추가 구성 및 확장 시 요구되는 정보를 제공하기 위하여 기존의 네트워크 구성 요소 관리를 위한 프로그램을 자주 수정해 주어야 한다. 그러나 기존의 네트워크 관리 시스템은 이러한 기능을 제공하지 못하고 있다.

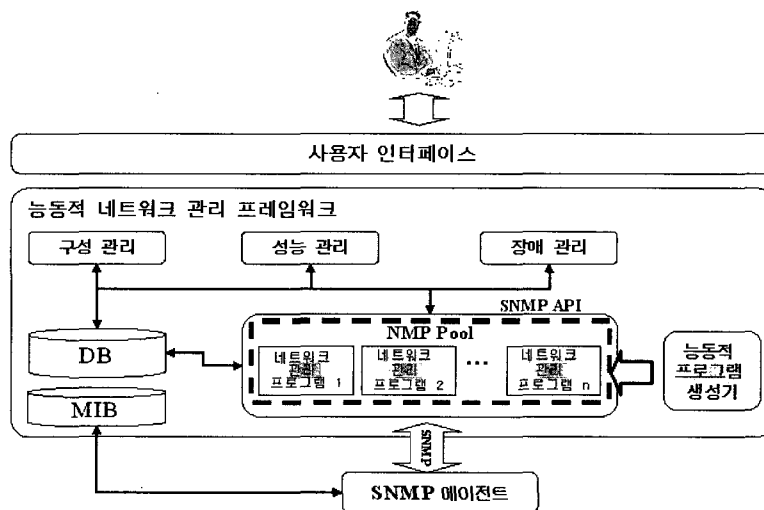
이 논문에서는 기존 연구의 문제점을 해결하기 위하여 네트워크 대상 객체 관리를 위한 능동적 프로그램 생성 기능을 지원한다. 능동적 프로그램 생성 기능이란 새로운 네트워크 대상 객체를 관리하기 위해 필요한 네트워크 관리 프로그램을 자동으로 생성해 주는 것을 의미하는 것으로서 기존의 수작업으로 인해 발생했던 많은 에러들에 대한 문제까지 해결할 수 있다.

3. 능동적 네트워크 관리 프레임워크 구조

제안한 능동적 네트워크 관리 프레임워크는 대규모의 복잡한 네트워크를 효율적으로 관리하기 위해서 구성 관리, 성능 관리, 장애 관리, 능동적 프로그램 생성기, SNMP API 및 데이터베이스 인터페이스로 구성되어 있다. 제안하는 능동적 네트워크 관리 프레임워크의 전체적인 구조는 (그림 1)과 같다. 제안하는 프레임워크의 구성 요소들 중에서 구성 관리(CM: Configuration Management)는 전체 네트워크상의 관리 대상에 대한 정보 및 현재 상태를 관리하는 기능을 수행하게 되며, 성능 관리(PM: Performance Management)는 CPU나 메모리등과 같은 하드웨어 정보를 주기적으로 관리하는 기능을 수행한다. 장애 관리(FM: Fault Management)

는 네트워크 관리 대상에서 발생하는 장애들을 모니터링 하여 실시간으로 장애 정보를 제공하는 기능을 수행한다. SNMP API는 SNMP 에이전트와 통신을 하며 네트워크 관리 객체들을 관리하는 역할을 수행한다. (그림 1)에서와 같이 구성 관리, 성능 관리 및 장애 관리는 각 모듈들의 기능을 구현하기 위하여 필요한 네트워크 정보를 SNMP API를 통하여 네트워크 관리 객체 내에 위치한 SNMP 에이전트로부터 얻는다. 또한 SNMP API는 네트워크 관리 프로그램들의 집합인 NMP Pool을 포함하고 있다. 네트워크 관리 프로그램은 실질적으로 각각의 SNMP 에이전트로부터 정보를 수집하고 SNMP 라이브러리 역할을 수행하며, 네트워크 관리 객체의 종류마다 해당 에이전트를 관리하기 위한 네트워크 관리 프로그램이 별도로 존재하여야 한다. 그러나 네트워크 기술이 발달함에 따라 새로운 네트워크 장비들이 속속 등장하고 있고, 이러한 장비들을 관리하기 위한 네트워크 관리 프로그램을 매번 개발하여야 하기 때문에 네트워크 관리 시스템을 유지/관리하기 위한 비용이 증가한다.

제안하는 프레임워크에서는 능동적 프로그램 생성기[19, 20]를 통하여 새로운 네트워크 장비를 신속하게 관리할 수 있도록 네트워크 관리 프로그램의 자동 생성 기능을 지원한다. 능동적 프로그램 생성기는 구성 관리, 성능 관리 및 장애 관리 모듈로부터 호출되는 바이너리 코드 형태의 네트워크 관리 프로그램을 생성한다. 생성된 네트워크 관리 프로그램은 사용자가 원하는 MIB 객체에 대한 SNMP 연산을 수행하는 코드가 포함된 것으로서 구성 관리, 성능 관리 및 장애 관리의 기능을 수행하기 위하여 필요한 MIB 객체에 대해 SNMP Get/Set 연산을 수행한다. 아울러 실제 SNMP 연산을 수행하기 위해서는 제공되는 SNMP 라이브러리 메소드를 호출하고, 호출 결과로 생성되는 값을 구성 관리, 성능 관리 및 장애 관리에게 반환한다. 데이터베이스는 SNMP API를 통하여 얻어진 네트워크 관리 객체에 대한 정보를 저장한다. 이 정보는 향후 각각의 네트워크 관리 객체에 대한 과거 정보가 요청되었을 때, 데이터베이스로부터



(그림 1) 전체 능동적 네트워크 관리 프레임워크 구조

해당 정보를 읽어온다. 다음 절부터는 각 구성 요소의 기능과 처리 과정에 관하여 상세히 기술한다.

3.1 구성 관리

구성 관리는 네트워크에 포함된 네트워크 장비와 각 장비의 구성에 대한 모든 내용을 관리하는 기능을 수행한다. 구성 관리에 필요한 모든 정보들은 노드, 인터페이스 및 서비스 별로 구분 되어 데이터베이스 테이블로 저장 관리한다. 또한 제안한 프레임워크에서는 네트워크 내의 관리 대상들을 네트워크를 구성한 조직의 부서 또는 지역으로 분류하여 그룹 단위로 관리한다. 그룹의 분류는 네트워크 매니저가 네트워크의 구성 환경에 맞게 생성할 수 있다.

구성 관리에서는 sysDescr, sysObjectID, sysName, sysContact, sysLocation, sysServices와 같은 노드 구성 정보와 ifDescr, ifType, ifMTU, ifSpeed, ifPhyAddress, ifAdminStatus, ifOperStatus등과 같은 인터페이스 정보들을 사용한다. 여기서 "sys"라는 접두어로 시작하는 속성들은 시스템과 관련된 속성임을 의미하며, "if"라는 접두어는 인터페이스 속성임을 나타낸다. 먼저 노드 관리를 위해 사용자가 지정한 IP 주소 대역에서 관리 대상 노드를 검색한다. 이때 IP 주소 대역은 배열을 사용하여 처리하고, 사용자는 지정한 IP 주소 대역(IP 시작 주소, IP 끝 주소), SNMP 포트 번호 및 SNMP 커뮤니티 문자열을 입력한다. 사용자는 입력된 정보를 이용하여 관리 대상 노드를 검색하고, 검색 결과를 노드 관리 테이블에 저장한 후 검색된 노드의 개수를 반환한다. 주어진 IP 주소 대역에서 관리 대상 노드 검색은 IP 시작 주소부터 IP 끝 주소까지 해당하는 IP 주소마다 ping 명령을 수행하여 네트워크 도달 가능성을 검사한다. 이때 ping 명령이 성공하면 SNMP를 통해 시스템에 대한 정보를 수집하는 SNMP get 연산을 수행한다. 만약 ping 명령이 실패하게 되면 다음 IP 배열의 주소로 이동하여 ping 명령을 수행한다. 이 과정을 IP 주소 대역에 저장된 모든 IP 주소에 대해서 반복 수행한다.

3.2 성능 관리

성능 관리에서는 일정 수준으로 네트워크 서비스를 제공하기 위해 네트워크 트래픽을 측정하고 설정하는 기능을 수행한다. 즉 네트워크에서 발생하는 상황을 모니터링하고, 네트워크 디자인을 조정하여 네트워크 트래픽 성능을 최적화한다. 성능 관리는 노드 성능 관리, 인터페이스 성능 관리 및 서비스 성능 관리로 이루어지며, 네트워크에서 발생하는 트래픽을 분석하여 그 결과를 제공한다.

첫째, 노드 성능 관리는 CPU 및 메모리 사용율에 대한 관리를 수행한다. 노드 성능은 주기적으로 노드에서 CPU와 메모리에 대한 사용율을 SNMP를 통하여 노드 성능 관리 테이블에 저장한다. 성능 관리를 위한 시간 주기는 5분, 10분 등 매니저가 네트워크 환경에 맞게 설정할 수 있다. 수집된 성능 값이 초기에 설정한 임계치를 넘어가게 되면 TCA(Threshold Crossing Alert)를 발생시킨다. TCA는 성

능 관리를 조절하기 위한 속성으로서 네트워크 관리 대상으로부터 발생하는 상황을 확인할 수 있도록 제공되는 정보를 말한다.

둘째, 인터페이스에 대한 항목도 주기적으로 인터페이스에서 SNMP를 통하여 성능 관리 테이블에 저장한다. 성능과 관련하여 수집된 값들은 임계치에 따라 관리한다. 인터페이스 성능 관리를 위해 인터페이스 입력 이용율, 입력 처리율, 입력 에러율 및 입력 폐기율을 이용한다. 인터페이스에 대한 성능 관리 항목들은 다음과 같이 관리한다.

- 인터페이스 입력 이용율: 인터페이스 입력 이용율이 임계치를 초과한 경우로서 이용율은 식 1에 의해 계산된다.

$$ifInUsage = \frac{(ifInOctets \times 8)}{(ifSpeed \times interval)} \times 100 \quad (식 1)$$

위 식에서 ifSpeed는 전체 네트워크 속도를 나타내며, 단위는 bps(bit per second)이다. interval은 수집 주기로서 단위는 초(sec)이다. ifInOctets은 입력 옥텟수를 의미한다.

- 인터페이스 입력 처리율: 인터페이스 입력 처리율이 임계치 미만으로 하락한 경우로서 식 2에 의해 계산된다.

$$ifInThroughput = \frac{(ifInTotalPkts - (ifInDiscardPkts + ifInErrorPkts))}{ifInTotalPkts} \times 100 \quad (식 2)$$

위 식에서 ifInTotalPkts은 전체 패킷 수를 나타내며 전체 패킷은 (ifInUcastPkts - ifInNUcastPkts) 로 얻을 수 있다. ifInDiscardPkts은 입력 폐기 패킷 수를 나타낸다. ifInErrorPkts은 입력 에러 패킷 수를 의미한다.

- 인터페이스 입력 폐기율: 인터페이스 입력 폐기율이 임계치를 초과한 경우로서 식 3과 같다.

$$ifInDiscardRate = \frac{ifInDiscardPkts}{ifInTotalPkts} \times 100 \quad (식 3)$$

- 인터페이스 입력 에러율: 인터페이스 입력 에러율이 임계치를 초과한 경우로서 식 4와 같다.

$$ifInErrorRate = \frac{ifInErrorPkts}{ifInTotalPkts} \times 100 \quad (식 4)$$

이러한 인터페이스의 입력 트래픽 정보들은 수집 대상이 되는 노드의 IP 주소, 노드의 인터페이스 목록, 이전 주기에서 수집된 성능 임계치(preValue) 및 사용자가 입력한 임계치(curValue) 등을 기반으로 수집되며, 이들 정보는 SNMP get 연산을 통하여 수행된다. 인터페이스에 대한 트래픽 정보는 사용자가 입력한 임계치 값과 이전 주기에서 수집된 성능 임계치 값의 차를 계산하여, 트래픽정보 = (curValue

- *preValue*), 현재 주기에 측정된 입력 트래픽 정보를 산출한다.

마지막으로 서비스 성능 관리는 현재 네트워크 서비스의 상태 정보 및 네트워크 서비스를 사용하고 있는 노드와 인터페이스 정보들을 테이블로 저장하여 관리한다. 네트워크에서 발생하는 트래픽은 요일 별, 시간 별로 특성이 다르게 나타난다. 즉 주말에는 주중보다는 적은 수의 트래픽이 발생하며, 시간대 역시 업무 시간대(am 8~pm 6)에는 비업무 시간대(pm 7~익일 am 7) 보다는 많은 수의 트래픽이 발생한다. 만약 임계치를 주중의 업무 시간대와 같이 트래픽 발생량이 많은 조건에 맞추어 값을 주게 되면 주말이나 주중의 비업무 시간대에는 비효율적인 자원 낭비가 발생하게 된다. 요일 별 및 시간 별 임계치를 효율적으로 관리할 수 있도록 적응형 임계치를 적용한다. 적응형 임계치란 요일 별 및 시간 별로 임계치를 조절하여 CPU와 메모리 사용율을 최적화하기 위한 임계치 변수이다. 적응형 임계치를 계산하기 위해서 이전에 수집된 샘플 데이터를 기반으로 임계치를 결정하게 된다. 샘플 데이터 $S = \{s_0, s_1, s_2, \dots, s_{23}\}$ 가 있다고 가정하자. 여기서 s_i 는 시간대별(0시~23시) 수집 데이터이다. 같은 시간대에 발생하는 트래픽의 최대, 최소, 평균을 각각 *Max*, *Min*, *Avg*라고 한다. 이때 적응형 임계치를 구하는 계산식은 식 5와 6과 같다.

$$LowerBound = Min(Avg \times 20\%) \quad (\text{식 5})$$

$$UpperBound = Max(Avg \times 20\%) \quad (\text{식 6})$$

여기서 *LowerBound*, *UpperBound*는 각각 적응형 임계치의 하한, 상한 경계치를 의미하며, 이때 측정된 값은 적응형 임계치의 두 경계 값과 비교하여 TCA 발생 여부를 결정한다.

3.3 장애 관리

장애 관리에서는 관리 대상인 노드, 인터페이스 및 서비스에 대한 장애를 자동으로 모니터링 하여 실시간으로 정보를 제공하는 기능이다. 제안한 시스템에서는 장애를 4등급(Critical, Major, Minor, Warning)으로 구분하여 체계적인 장애 관리 기능 제공한다. 장애 등급에서 Critical은 가장 심각한 장애로서 시스템에 위급한 문제가 발생하였음을 나타낸다. Major는 중요한 장애로서 시스템에 중대한 문제가 발생하였으며, 네트워크 운용에도 문제가 발생하였음을 나타낸다. Minor는 일반적인 장애로서 시스템에 문제가 발생하였지만 네트워크 운용에는 문제가 없음을 나타낸다. 마지막으로 Warning은 경고성 장애로서 네트워크 운용을 위해서는 시스템에 점검이 필요한 경우를 의미한다. 노드, 인터페이스 및 서비스에 대한 장애 발생에 대한 이력 정보를 관리하기 위해 데이터베이스 테이블로 저장 관리한다. 장애 관리는 장비에서 발생하는 SNMP Trap을 받아서 장애 처리를 한다. 처리되는 장애 정보로는 ICMP 메시지 도달성 테스트 정보로서 Ping Success, Ping Fail 항목 등이 있고, 수

신되는 SNMP Trap 정보로서 노드가 하드웨어적으로 재시작하는 coldStart, 노드가 소프트웨어적으로 재시작하는 warmStart, 인터페이스가 down을 나타내는 linkDown, 인터페이스가 up되는 linkup, 그리고 허가되지 않는 SNMP 접근을 의미하는 AuthenticationFailure항목 등이 있다.

3.4 SNMP API의 동작

SNMP API는 매니저/에이전트 구조로서 SNMP 에이전트의 정보를 얻기 위하여 사용한다. 또한 SNMP 매니저의 핵심적인 역할을 담당한다. SNMP 매니저는 에이전트 그룹을 제어하고, 직접적으로 개인의 에이전트 관리 정보나 그들의 특수한 요소를 운용하고 교환하는 능력을 가지고 있다. 매니저는 에이전트 보다 더 복잡적이고, 일련의 에이전트 정보를 수집하며, 정보를 관련시킨다. 또한 일련의 에이전트 동작을 수행하는 하나의 직접적인 Set 연산을 호출한다. SNMP 에이전트는 관리 대상 객체, 즉 호스트, 라우터, 브리지, 허브와 같은 네트워크 장비에 설치되어 관리 시스템의 요구에 따라 관리 정보를 송달하거나 관리 시스템의 액션 요구를 수행하며 문제 발생 시 자동적으로 장애 상황을 관리 시스템에 통보한다. 또한 에이전트는 대부분 수동적이며, 간단한 요소로서 원격지 프로세스의 직접적인 연결로 동작된다. 이들 이벤트는 Trap을 호출하며, Trap의 사용은 한정되어 있다. 에이전트는 단지 네트워크 요소에서 주어진 특정 자원들을 관리한다. 유효 범위는 에이전트를 상대적으로 단순하고, 저렴한 비용으로 동작하게 한다.

3.5 데이터베이스 인터페이스

데이터베이스 인터페이스는 구성 관리, 성능 관리 및 장애 관리에서 필요한 데이터베이스 접근을 위해 사용자 관점의 인터페이스를 제공한다. 즉, 데이터베이스에 저장된 데이터에 대하여 검색, 삽입, 삭제, 갱신 연산을 위한 인터페이스와 결과 셋에 대한 처리 인터페이스를 제공한다. 이 인터페이스를 통하여 구성, 성능 및 장애 관리는 처리해야 할 데이터에 대하여 각 테이블에 원하는 연산을 데이터베이스 관리 시스템과 독립적으로 구현할 수 있다. 또한 각 관리 모듈에서 발생하는 데이터베이스에 대한 접근은 동시성에 따른 병목 현상 제거의 요구를 갖게 된다. 즉 데이터베이스에 대한 삽입 연산이 지속적으로 많이 발생하고, 데이터베이스 접근에 대한 병목 현상이 발생할 수 있기 때문에 데이터베이스 인터페이스는 성능 및 장애에 대한 데이터를 삽입하기 위한 연산을 메시지 방식으로 추상화하여 제공한다.

3.6 능동적 프로그램 생성기

능동적 프로그램 생성기는 관리를 필요로 하는 네트워크 대상 객체들에 대한 네트워크 관리 프로그램을 능동적으로 생성해준다. 생성된 네트워크 관리 프로그램은 구성 관리, 성능 관리, 장애 관리 모듈로부터 호출되는 바이너리 코드 형태의 프로그램이다. 제안하는 능동적 프로그램 생성기는 NE Basic Info Handler, MIB Handler, Template Handler,

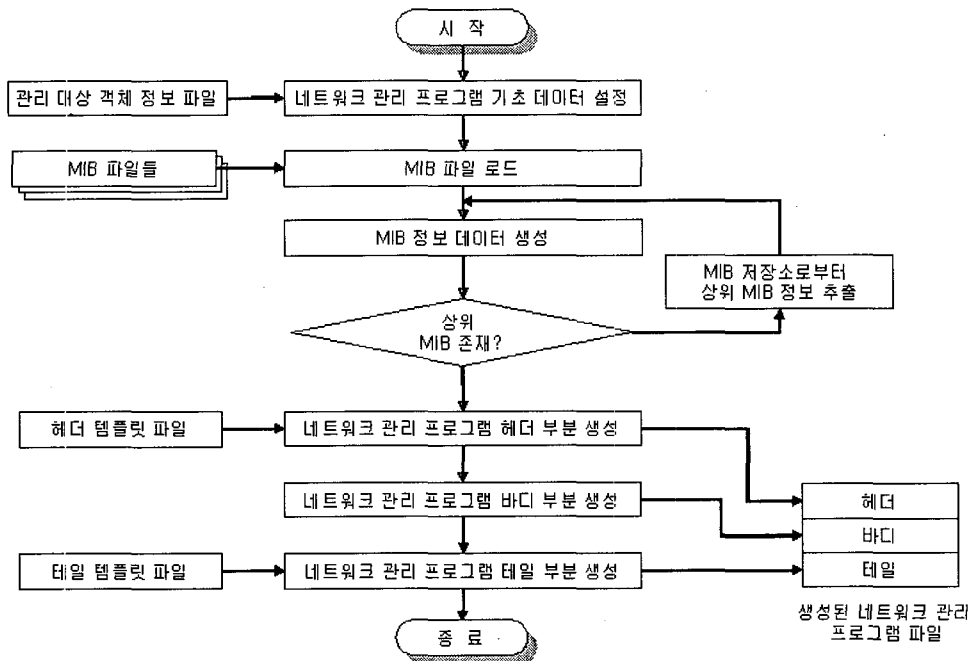
그리고 Operation Handler 로 이루어져 있으며, 각 구성요소의 기능은 다음과 같다.

- NE Basic Info Handler:** 먼저 네트워크 관리 객체에 대한 네트워크 관리 프로그램을 생성하기 위해서는 관리 대상 객체에 대한 기본 정보가 필요하다. NE Basic Info Handler 에서는 생성될 네트워크 관리 프로그램에 대한 정보를 생성하여 저장하는 기능을 담당한다. 네트워크 관리 프로그램에 대한 기초 정보 생성을 위해 몇 가지 기본 정보를 필요로 한다. 필요한 기본 정보들로는 1) 생성될 네트워크 관리 프로그램 클래스 이름 및 구성요소 프로그램의 이름, 2) 네트워크 개발자가 지정하고자 하는 네트워크 구성요소의 관리정보 파일 이름, 3) 특정 네트워크 구성요소에 관한 관리 객체의 이름, 4) SNMP Set 연산에 관한 메소드의 생성 여부 5) 관리정보에 포함된 특정한 수치 데이터를 문자 데이터와 상호 전환하는 메소드의 생성여부를 표시하는 데이터 등과 같은 속성들로 구성되어 있다.
- Operation Handler:** SNMP 수행을 위한 연산들을 관리하는 기능을 한다. SNMP 프로토콜은 4 가지 연산(Get, Get Next, Set, Trap)을 수행한다. Get 연산은 장비의 상태 및 가동 시간 등의 관리 정보를 읽어 들이고, Get Next 연산은 정보가 계층적 구조를 가지므로 매니저가 장비에 조회를 해서 해당 층보다 하위 층의 정보를 얻어, 장비의 MIB을 조작하여 장비를 제어한다. 그리고 SNMP 매니저는 요청을 보내 다시 초기화 시키거나, 프로그램에 따라 스스로를 다시 재구성하기 위해 Set연산을 수행을 통해 매니저에게 임계치나 이벤트 등을 보고

한다. 마지막으로 장비내의 SNMP 에이전트는 Trap 연산을 통해 경고나 고장 통지등 매니저가 미리 설정한 유형의 보고서를 생성할 수 있다.

- MIB Handler:** 네트워크 관리 프로그램 생성을 위한 MIB 정보 트리를 구축하기 위한 기능을 담당한다. MIB 정보 트리란 관리 대상이 되는 MIB 객체들이 트리 형태로 구성된 것으로서, MIB 정보 트리로부터 네트워크 구성요소 핸들러 대상이 되는 MIB 객체들의 식별자 값을 추출하고 제공되는 MIB 파일의 내용을 파싱 하여 트리를 구성하게 된다.
- Template Handler:** 생성되는 네트워크 관리 프로그램에서 공통적으로 사용되는 전제 부(Template Header)와 종료 부(Template Tail)를 정형화한 정보를 제공하는 기능을 한다. 전제 부에서는 네트워크 관리 프로그램의 명칭 및 필요한 변수를 정의하고, 종료 부에서는 디버깅을 위해 설정된 메소드에 관한 소스 코드를 정의한다.

(그림 2)는 능동적 프로그램 생성기의 수행 절차를 나타낸 것이다. NE Basic Info Handler를 통해 생성될 네트워크 관리 대상 객체에 대한 기초 데이터를 생성한다. 기초 데이터가 생성되면 MIB Handler에 의해 MIB 데이터베이스로부터 네트워크 장비와 함께 제공되는 MIB 파일과 사용자 지정 MIB 파일을 불러들여 MIB 트리를 구성한다. 구성된 MIB 트리와 Template Handler에서 Template Header와 Template Tail 을 붙여서 소스 코드를 생성 한 후 Operation Handler 로부터 SNMP와 관련된 연산들을 추가하여 네트워크 관리 프로그램을 생성한다.



(그림 2) 능동적 프로그램 생성기의 수행 절차

4. 구현 및 평가

이 장에서는 제안한 능동적 관리 프로그램 생성 기능을 지원하는 네트워크 관리 프레임워크를 구현하고, 능동적 프로그램 생성기를 통해 작성된 네트워크 관리 프로그램과 기존의 수작업에 의해 개발된 네트워크 관리 프로그램간의 결과 비교를 통해 제안하는 프레임워크의 신뢰성 및 정확성을 검증한다.

4.1 구현 환경

제안하는 능동적 네트워크 관리 프레임워크의 설계 및 구현은 윈도우즈 운영 체제를 기반으로 이루어졌으며, 네트워크 상태를 모니터링 할 수 있도록 이더넷 카드가 내장된 Pentium-IV 시스템에서 개발하였다. 프로그램의 개발은 자바 및 JSP를 이용하였으며, 모니터링 결과를 웹을 통하여 서비스할 수 있도록 Apache Tomcat 4.0을 사용하여 구현하였다. 또한 네트워크 관리 정보를 저장 및 유지하기 위해 데이터베이스는 MySQL을 사용하였다.

4.2 구현 결과

제안한 프레임워크의 인터페이스는 구성관리(그룹관리, 노드관리, 인터페이스, 서비스 관리), 장애관리, 성능관리로 구성되어 있다. (그림 3, 4, 5)는 구성 관리 중에서 그룹 관리 기능, 노드 관리 기능 및 인터페이스 관리 기능을 위한 인터페이스를 각각 나타낸다.

그룹관리

검색조건 : 그룹 이름 검색어 : 지역

그룹명	그룹설명	기능
마포지역	마포지역 네트워크 구성에 사용된 장비들 마포지역의 담당자는 홍길동이다.	수정 삭제
송파지역	송파지역 네트워크 구성에 사용된 서버들 담당자는 박영수이다.	수정 삭제
여의도 지역	여의도 네트워크 구성에 사용된 서버들	수정 삭제

전체개수:3 1/1 pages

(그림 3) 그룹 관리 기능

인터페이스관리

검색조건 : 인터페이스 IP주소 검색어 : 210.125 인터페이스 상태 : 전체 인터페이스 유형 : 전체

그룹명	노드명	노드IP주소	ifIndex	인터페이스 IP주소	유형	상태	기능
전산실	SYSTEM	210.125.144.116	16777219	210.125.144.116	ethernet-csmacd	ON	보기
마포지역		210.125.144.37	1	210.125.144.37	ethernet-csmacd	ON	보기
전산실	CASWEL-NVMLOVE	210.125.144.103	2	210.125.144.103	ethernet-csmacd	ON	보기
송파지역	B.43.su48	210.115.161.251	322	210.125.144.1	Unknown	ON	보기

전체개수:4 1/1 pages

(그림 4) 인터페이스 관리 기능

노드관리

검색조건 : 노드 IP주소 검색어 : 210.115 노드 상태 : 전체 노드 유형 : 전체

그룹명	노드명	IP주소	노드유형	상태	개수	기능
송파지역	john	210.115.167.100	Unix Server	ON	1	보기 수정 삭제
전산실		210.115.167.112	Printer	ON	1	보기 수정 삭제
마포지역	rtab	210.115.167.101	Unix Server	ON	1	보기 수정 삭제
송파지역	ALGO-SERVER	210.115.167.121	Windows Server	ON	1	보기 수정 삭제
여의도 지역	KKIDO	210.115.167.203	Windows Server	ON	1	보기 수정 삭제
송파지역	com.scien	210.115.167.129	Unix Server	ON	1	보기 수정 삭제
여의도 지역	alcere	210.115.167.119	Unix Server	ON	1	보기 수정 삭제
송파지역	PDAPROCLUB	210.115.167.165	Windows Server	ON	1	보기 수정 삭제

전체개수:31 1/4 pages

(그림 5) 노드 관리 기능

장애관리

카테고리 : 전체 발생대상 : 인터페이스 발생시간 : 2003-04-29 날짜 : 2003-04-1 부터 2003-04-9 까지

장애명	severity	발생대상	발생개체	발생시간	지속시간	기능
dataWrong	⊕	▲	B.37-s48-29	20030401 00:29:45		미결
dataRight	⊕	▲	B.37-s48-29	20030401 00:34:45		미결
dataWrong	⊕	▲	B.37-s48-29	20030402 23:54:55		미결
dataRight	⊕	▲	B.37-s48-29	20030402 23:59:55		미결
dataWrong	⊕	▲	B.37-s48-29	20030403 00:04:55		미결
dataRight	⊕	▲	B.37-s48-29	20030403 00:09:55		미결
dataWrong	⊕	▲	B.37-s48-29	20030403 22:14:59		미결
dataRight	⊕	▲	B.37-s48-29	20030403 22:19:59		미결

전체개수:20 1/3 pages

(그림 6) 장애 관리 기능

NMS 성능관리

최근 3일간 CPU 사용률 Top 5

노드명	노드IP	CPU(%)	Memory(%)
GALMIT	192.168.0.1	6.90	68.98

최근 3일간 CPU 사용률 TCA 발생건수 Top 5

노드명	노드IP	발생건수(건)	지속시간(초)
GALMIT	192.168.0.1	8	1,499

최근 3일간 인터페이스 이용률 Top 5

노드명	인터페이스	이용률(%)	발생시간(분)	지속시간(초)
GALMIT	16777219(192.168.0.1)	0.84	0.92	179
GALMIT	16777218(192.168.0.1)	0.84	0.85	192

최근 3일간 인터페이스 이용률 TCA 발생건수 Top 5

노드명	인터페이스	발생건수(건)	지속시간(초)
GALMIT	16777219(192.168.0.1)	1	192

(그림 7) 성능 관리 기능

(그림 6)과 (그림 7)은 장애관리와 성능관리를 위한 인터페이스를 보여준다.

4.3 능동적 프로그램 생성기를 통한 네트워크 관리 프레임워크의 평가 및 분석

이 절에서는 다양한 종류의 네트워크 관리 객체에 대해 능동적으로 생성된 네트워크 관리 프로그램과 수작업에 의해 작성된 네트워크 관리 프로그램의 결과를 분석하고, 제안된 프레임워크를 통해 생성된 네트워크 관리 프레임워크의 신뢰성을 평가한다. 네트워크 관리 프로그램의 기능은 관리하고자 하는 네트워크 관리 객체로부터 관리 정보를 가져오는 역할을 수행한다. 프로그램의 신뢰성은 다양한 종류

〈표 1〉 네트워크 관리 객체의 예

노드 유형	노드 IP	노드 이름
라우터 (HANA)	211.196.xxx.127	ROUTER
윈도우즈 서버	211.196.xxx.133	KT-9Z25FJCHPIZ8

의 네트워크 관리 객체들로부터 정확한 정보가 획득되는 지를 비교하여 평가한다. 이 논문에서는 제안하는 프레임워크를 검증하기 위하여 표 1처럼 서로 다른 종류의 네트워크 관리 객체로부터 정보를 획득하여, 그 결과를 비교하였다.

실험 방법으로는 엔트리 객체인 ifEntry 값을 5분 주기로 수집하고, 각 주기에 측정된 값의 변화를 측정하였다. 24시간 정보를 수집하였고, 각 주기별로 수집된 항목은 입력 옥텟 수, 입력 유니캐스트의 패킷 수, 입력 비 유니캐스트 패킷 수, 출력 이용률, 출력 옥텟 수, 출력 유니캐스트 패킷 수, 출력 비 유니캐스트 패킷 수, 출력 이용률 순으로 수집하였다.

(그림 8)과 (그림 9)는 같은 라우터로부터 수작업에 의해 생성된 네트워크 관리 프로그램과 제안된 프레임워크를 통해 생성된 네트워크 관리 프로그램에 대한 결과를 각각 보여준다. 동일한 네트워크 관리 객체인 라우터에 대하여 제안된 프레임워크를 통해 생성된 네트워크 관리 프로그램과 수작업으로 작성한 네트워크 관리 프로그램의 수행 결과 모두 동일한 네트워크 관리 정보를 수집하였다. 이러한 수행 결과는 이 논문에서 제안하는 프레임워크의 결과에 대한 신뢰성을 검증하는 것이다. 또한 제안된 프레임워크를 통해 생성된 네트워크 관리 프로그램을 이용하여 라우터의 성능을 모니터링 한 결과 아무런 문제없이 수행됨을 확인하였다.

(그림 10)과 (그림 11)은 윈도우 서버에 대하여 두 개의 네트워크 관리 프로그램을 이용하여 수집한 네트워크 관리 정보를 보여 준다. 동일한 네트워크 관리 객체인 윈도우 서버에 대하여 제안된 프레임워크를 통해 생성된 네트워크 관리 프로그램과 수작업으로 작성한 네트워크 관리 프로그램의 수행 결과 역시 동일한 네트워크 관리 정보를 수집하였다.

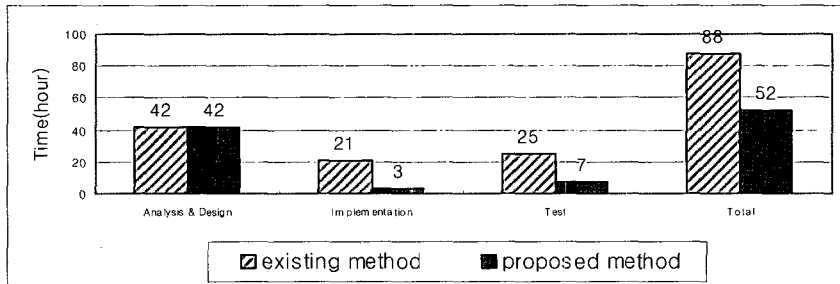
수작업과 제안된 프레임워크를 이용하여 생성된 네트워크 관리 프로그램에 대해서 오류 및 생성 시간과 같은 성능에 대한 비교는 프로그램을 생성하기 전에 수집되는 정보의 변수가 많기 때문에 직접 비교하는데 문제가 있었다. 따라서

(그림 9) 라우터(HANA)에서 수집된 데이터 (제안된 프레임워크)

(그림 10) 윈도우 서버에서 수집된 데이터 (수작업)

(그림 8) 라우터(HANA)에서 수집된 데이터 (수작업)

(그림 11) 윈도우 서버에서 수집된 데이터 (제안된 프레임워크)



(그림 12) 라우터 장치관리를 위해 필요한 관리 프로그램 개발에 소요되는 전체적인 시간을 분석

제안된 프레임워크를 검증하기 위해서 라우터 장치에 대한 관리 프로그램 개발을 위한 전체적인 시간을 비교하였다. (그림 12)는 프로그램 개발에 대해 분석 및 설계단계, 구현 단계, 테스트 단계 및 전체적인 소요 시간 단계로 구분하여 기존의 방식과 제안된 방식을 각각 비교 하였다. 비교 한 결과 제안된 방식은 구현과 테스트 단계에서 많은 시간 감소를 보였다. 전체적으로 기존의 수작업에 의한 방법에 비해 37%정도 개발 시간을 줄일 수 있었다. 또한 우리는 제안한 프레임워크의 신뢰성과 운용에 대한 안정성을 검증하기 위해서 한 달 이상 실세계 응용 분야에 적용한 결과, 다양한 네트워크 관리 객체들에 대하여 능동적으로 생성된 네트워크 관리 프로그램의 실행 상의 오류 없이 정상적으로 동작함을 확인하였다.

따라서 능동적 프로그램 생성 기법에 의해 생성된 네트워크 관리 프로그램의 신뢰성이 보장되며, 생성된 네트워크 관리 프로그램 상의 오류가 발생하지 않음을 알 수 있다. 그리고 제안하는 프레임워크를 통해 프로그램 개발에 소요되는 인적, 물적 비용 및 네트워크 관리 시스템의 유지/관리를 위한 시간적 비용이 크게 절감됨을 확인하였다.

5. 결 론

기존의 네트워크 관리 시스템은 새로운 네트워크 장비나 관리 대상이 추가되면 관리 대상 객체에 대한 네트워크 관리 프로그램을 수작업에 의해서 생성한다. 또한 지속적으로 새로운 네트워크 관리 객체가 개발되고 있기 때문에 각각의 객체마다 일일이 관리 프로그램을 생성하는 것은 많은 비용이 소모된다. 따라서 이 논문에서는 능동적 프로그램 생성 기능을 지원하는 네트워크 관리 프레임워크를 제안하였다. 제안된 프레임워크는 실시간 네트워크 관리를 위해 구성관리, 성능관리 및 장애관리의 기본 구성요소와 이들 기본 구성요소를 통해 관리 대상이 되는 네트워크 객체들에 대한 정보 생성을 능동적으로 지원해주기 위한 능동적 프로그램 생성기로 구성되어 있다. 또한 네트워크 관리 객체에 대한 정보를 가지고 SNMP 매니저와 함께 네트워크 관리를 수행하게 되는데, 이때 필요한 네트워크 관리 객체들에 대한 네트워크 관리 프로그램 생성을 기존의 수작업에 의한 방법에서 능동적으로 생성해 준다. 이 연구의 실용성을 검증하기 위하여, 우리는 실제 네트워크 상에 제안된 프레임워크를 구축하고 운영하였다. 그 결과로서 네트워크 객체에 대한 확장이나 추가 시, 네트워크 관리 객체에 대한 네트워크 관리 프로그램 생성에 대한 개발 기간을 단축하는 효과와 개발비용 및 유지 보수비용이 감소되는 결과를 얻었다. 또한 기존의 네트워크 관리 시스템에서 수작업에 의해 생성된 네트워크 관리 프로그램에 포함되는 많은 오류 문제 해결을 통해, 제안된 능동적 네트워크 관리 프레임워크는 네트워크 관리 프로그램의 자동화 생성 기능뿐만 아니라 정확성 향상에도 기여하므로 능동적인 네트워크 관리를 위해 활용할 수 있다.

참 고 문 헌

- [1] J. D. Case, M. S. Fedor, M. L. Schoffstall, and J. R. Davin, "A Simple Network Management," RFC 1098, <http://www.faqs.org/rfcs/rfc1098.html>, April 1989.
- [2] J. Case, M. Fedor, M. Schoffstall, and C. Davin, "The Simple Network Management Protocol," RFC 1157, <http://www.faqs.org/rfcs/rfc1157.html>, May 1990.
- [3] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1907, <http://www.faqs.org/rfcs/rfc1907.html>, January 1996.
- [4] H. T. Ju, S. H. Han, Y. J. Oh, J. H. Yoon, H. J. Lee, and J. W. Hong, "An Embedded Web Server Architecture for XML-Based Network Management," in Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS2002), Florence, Italy, pp. 5-18, April 2002.
- [5] K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II," RFC 1213, March 1991.
- [6] M. Rose and K. McCloghrie, "Concise MIB Definitions," RFC 1212, <http://www.faqs.org/rfcs/rfc1212.html>, March 1991.
- [7] W. Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2," Third edition, Addison-Wesley, Reading, MA, USA, 1999.
- [8] F. Strauss, "A Library to Access SMI MIB Information," <http://www.ibr.cs.tu-bs.de/projects/libsmi/>, 2002.
- [9] Consortium, WBEM homepage, <http://wbem.freerange.com/>, 2000.
- [10] J. P. Martin-Flatin. "Web-Based Management of IP Networks and Systems," Ph.D. thesis, Swiss Federal Institute of Technology, Lausanne (EPFL), October 2000.
- [11] WhatsUp Gold, <http://www.ipswitch.com>.
- [12] Solar Winds Engineers Edition, <http://www.solarwindsuk.net>.
- [13] VisualRoute, <http://www.visualroute.com>.
- [14] HP OpenView, <http://openview.com>.
- [15] SunNet Manager, <http://www.sun.com>.
- [16] SPECTRUM, <http://www.aprisma.com>.
- [17] TAPS, <http://www.kdccorp.co.kr>.
- [18] MonaLisa, <http://www.inti.co.kr>.
- [19] 이명진, 김은희, 신문선, 이웅재, 류근호, "네트워크 관리 프로그램 자동생성 프레임워크," 정보처리학회 논문지, 제12-C권 제 6호, pp933-940, 2005년 10월.
- [20] E. H. Kim, M. J. Lee, and K. H. Ryu, "AGNC: An Automatic Generation of Network Components for Dynamic Network Management," ICIC 2006, pp165-170, August 2006.



김 은 희

e-mail : ehkim@dblab.chungbuk.ac.kr
2001년 삼척대학교 정보통신공학과
(공학사)
2003년 충북대학교 대학원 전자계산학
전공(이학석사)
2003년~현재 충북대학교 전자계산학과
박사 수료

관심분야 : 접근제어, 네트워크 침입 탐지 시스템, 무선
센서네트워크 보안



이 명 진

e-mail : mjlee@paran.com
1984년 충북대학교 계산통계학과
(이학사)
1986년 숭실대학교 대학원 전자계산학
전공(공학석사)
2005년 충북대학교 대학원 전자계산학
전공(이학박사)

1989년~2001년 KTI 초고속 망관리 사업팀(망관리시스템 부장)
2001년~2001년 성화통신 기술연구소(S/W 연구실장)
2002년~현재 가림정보기술(주) 대표이사
관심분야 : USN, 망관리시스템, 미들웨어 시스템



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산학과 (이학사)
1980년 연세대학교 대학원 전산전공
(공학석사)
1988년 연세대학교 대학원 전산전공
(공학박사)

1976년~1986년 육군군수 지원사 전산실(ROTC 장교),
한국전자통신연구원(연구원), 한국 방송대 전산학과
(조교수) 근무

1989년~1991년 Univ. of Arizona Research Staff (TempIS
연구원, Temporal DB)

1986년~현재 충북대학교 전기전자컴퓨터공학부 교수

관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal
GIS, 유비쿼터스 컴퓨팅 및 스트림 데이터 처리,
지식기반 정보검색 시스템, 데이터마이닝, 바이오
인포매틱스