웹 기반의 트래픽 모니터링 및 분석 시스템의 설계와 구현

(Design and Implementation of a Web-based Traffic Monitoring and Analysis System)

이 명 섭 † 박 창 현 ††
(Myung-Sub Lee) (Chang-Hyeon Park)

요 약 최근 들어 TCP/IP 프로토콜을 사용하고 있는 통신망들은 전 세계적으로 엄청나게 증가하고 있다. 특히 인터넷과 WWW(World Wide Web) 서비스의 증가는 통신망에서의 트래픽 증가를 더욱 가속화 하고 있다. 본 논문에서는 먼저 네트워크 트래픽 모니터링 및 분석을 위한 요구사항을 조사하고, 웹 기반의 트래픽 모니터링 및 분석 시스템을 설계하고 구현한다. 또한 SNMP 분석 파라미터들을 정의하고 이를 이용하여 네트워크 트래픽 분석을 수행한다. 마지막으로 수집된 트래픽 결과를 기반으로 분석 형태에 따라 GUI(Graphic User Interface)형태로 표현한다. 본 논문에서 제시한 시스템은 웹을 이용하여 플랫폼에 종속되지 않고, 분석 파라미터에 따라 인터넷 트래픽 상황을 분석할 수 있도록 하여 사용자가 자신의 호스트에서 직접 인터넷 트래픽을 관리할 수 있다.

키워드 : 트래픽 모니터링, SNMP, 트래픽 모니터링 에이전트, 트래픽 분석 에이전트

Abstract Within the past decade, TCP/IP network environment has been explosively widespread all over the world. As the internet and the WWW expand their boundaries, the network traffic caused by data transfers over the internet has also increased. In this paper, we present the design and implementation of a WebTraMAS(Web-based Traffic Monitoring and Analysis System) which can resolve the shortcomings of current management approaches, particularly on the network traffic monitoring and analysis. The WebTraMAS presented in this paper performs the network management activities based on the parameters related to the MIB-II of SNMP and the parameters related to the QoS such as network performance and fault. The proposed WebTraMAS, implemented using the WWW technology, is able for the network manager to manage the network easily and platform independently with the performance analysis of internet traffic.

Key words : Traffic monitoring, MIB-II, SNMP, Traffic monitoring agent, Traffic analysis agent

1. 서 론

현대의 통신망은 규모의 거대화, 구조의 복잡성 증가, 그리고 관리 기술의 이질성과 같은 문제로 사람의 손에 의한 수동관리가 한계에 달하고 있으며, 이질적인 장치들 간의 효율적인 통합관리 기술에 대한 요구가 높아지고 있다[1].

† 비 회 원 : 영남대학교 컴퓨터공학과
skydream@cse.yu.ac.kr
†† 종신회원 : 영남대학교 컴퓨터공학과 교수
park@cse.yu.ac.kr
논문접수 : 2001년 11월 5일
심사완료 : 2002년 8월 20일

이러한 문제를 해결하기 위하여 TINA-C(TINA Consortium)의 TINA(Telecommunications Information Networking Architecture)[2], ITU-T의 TMN (Telecommunications Management Network)[3][4], IETF의 SNMP(Simple Network Management Protocol) [5][6]와 같은 망 관리 체계들이 정립되고 있다.

그러나 TINA와 TMN과 같은 망 관리 체계는 기능 세분화, 계층화(Layering) 및 MIB(Management Information Base)/MIT(Management Information Tree) 정의와 같은 구조의 복잡성으로 인해 구현과 운용에 어려움이 따른다. 특히 IP 네트워크가 주류인 현재의 인터넷 및 인트라넷의 경우 IP 네트워크 관리용으로 일반적

으로 선택되는 SNMP체계로는 트래픽 모니터링이 한계가 있다. 왜냐하면 SNMP Agent는 NE(Network Element)의 입력 및 출력 트래픽 양에 대한 정보만을 얻을 수 있으며, 서비스 종류별 트래픽 점검은 수행할 수 없다.

본 논문에서 설계 및 구현하는 WebTraMAS(Web-based Traffic Management System)는 SNMP와 함께 패킷 획득(Packet capture)방식을 통하여 네트워크에서 전송중인 패킷을 조사하여 서비스 종류별 트래픽 양을 계산하고, 그 특성치를 분석하여 망 관리자에게 보고한다. 트래픽 양의 계산 방법은 MIB-II[7]에서 system, interface, ip, snmp 그룹의 관리 변수를 Case 다이어그램[8]에 따라 분석하여 선로 이용률, 에러 수신률, 인터페이스 패킷 송수신률, 인터페이스 패킷 송수신 손실률, 입출력 트래픽률, 관리 트래픽 이용률 등의 분석 파라미터를 이용한다. 또한 WebTraMAS는 ICMP(Internet Control Message Protocol)를 통하여 특정 네트워크 장비의 장애 상태를 파악하여 장애가 발생한 장비를 실시간으로 망 관리자에게 보고한다. 모니터링된 트래픽은 데이타베이스에 저장되고, 망 관리자의 요청이 있을 때 제공되어진다.

WebTraMAS는 SNMP를 이용함으로써 기존의 SNMP를 지원하는 망 장치 관리에 수정없이 사용이 가능하며, 패킷 획득 방식을 이용함으로써 SNMP가 지원하지 못하는 서비스별 트래픽 모니터링이 가능하다. 또한 ICMP를 통하여 SNMP가 설치되어 있지 않은 망 장치의 상태 정보를 얻을 수 있는 장점이 있으며, 트래픽 조절 기능을 두어 트래픽 과부하 시 TCP 트래픽의 경우 윈도우 사이즈 제어기능을 이용하여 트래픽 흐름을 제어한다.

또한, 망 관리자와의 인터페이스로 웹을 사용하여 어떤 컴퓨터에서도 실행이 가능하므로 망 관리자가 망관리 시스템이 설치된 특정 위치에 상주할 필요 없이 자유로이 이동할 수 있다. 구현한 시스템은 웹서버, 데이타베이스 서버, 모니터링 에이전트 및 관리 서버로 구성되어, 각각 독립적으로 동작할 수 있도록 분산 서버 방식을 도입하여 설계하였다. 분산 서버 방식은 각기 다른 프로세스가 여러 시스템으로 나뉘어 실행되므로, 실시간 관리가 요구되는 네트워크 관리 환경에서 기존의 시스템 보다 향상된 성능을 나타내게 된다는 장점이 있다.

2. 관련 연구

기존의 웹 기반의 트래픽 점검 시스템은 MRTG (Multi Router Traffic Grapher)[9], WebTrafMon (Web-based Network Traffic Monitoring and Analysis System)[10], RTFM(Realtime Traffic Flow Management)[11] 등이 있다.

MRTG는 SNMP를 이용하여 측정중인 장비의 입력/출력 패킷의 양을 웹을 통하여 그래프로 보여주는 트래픽 모니터링 도구로써, 크게 SNMP Manager 기능과 웹접속 기능으로 그 기능을 나눌 수 있다. SNMP Manager는 관리 대상 장비에서 구동중인 SNMP Agent로 패킷에 대한 정보를 SNMP 관리 명령으로 내리고 그 결과를 수신하여 LOG파일에 저장하는 기능을 수행하며, 웹 접속 부분은 수집된 패킷 정보를 웹 브라우저에서 읽어 갈 수 있도록 그래프를 위한 그림과 HTML 문서를 생성해 내는 기능을 수행한다. MRTG는 LOG파일을 통하여 최대 1년까지의 트래픽의 양을 그래프로 표시할 수 있지만 특정 서비스에서 이용한 트래픽 양을 나타내지는 못한다는 단점을 가지고 있다. 또한 어떤 호스트가 트래픽을 유발했으며, 어느 프로토콜이 병목현상을 일으켰는가 하는 등의 관리자가 문제 해결에 가장 필요로 하는 정보들을 제공해 주지 못한다는 결정적인 단점이 있다.

웹 기반의 트래픽 모니터링 분석 시스템인 Web TrafMon은 언제 어디서나 쉽게 사용할 수 있는 웹 브라우저를 통하여 사용자들이 복잡한 사용자 인터페이스를 통하지 않고 원하는 결과를 얻을 수 있으며, 각 네트워크의 계층별 트래픽 정보와 발신지, 목적지 정보를 볼 수 있다. 이 시스템은 크게 두 개의 모듈로 동작하는데, 첫 번째는 Packet Capture 모듈로써 네트워크 패킷으로부터 각각의 네트워크 계층에 대한 정보를 추출하고 미리 정의된 로그 파일에 저장하는 기능을 가진다. 두 번째는 분석 모듈(viewer)로써 데이타 리더, 분석기, 뷰 제어기로 구성되어 있다. 데이타 리더는 로그 파일로부터 데이타를 읽어 들이는 기능을 수행하고, 분석기는 뷰 제어기가 요청한 데이타를 분석하는 역할을 수행하며, 뷰 제어기는 사용자와 상호 대화적으로 작동하며 사용자가 원하는 정보를 보여주는 역할을 수행한다. 그러나 이 시스템 또한 특정 서비스에서 이용한 트래픽 양을 나타내지 못하며 네트워크 이용률, 에러율 등을 측정하지 못한다는 단점을 가진다.

RTFM은 IETF에서 연구 중인 실시간 트래픽 플로우를 측정하는 시스템으로 네트워크 상의 트래픽을 실시간으로 capture 하여 플로우 단위로 통계 및 각종 수치를 보여주기 위한 시스템이다. RTFM 구조는 다음과 같은 구성 요소를 가진다.

① Meter : network monitoring point에서 실제

network flow를 요구 사항에 맞도록 읽어 들여 측정 데이타를 만드는 모듈

② Meter Reader : meter로부터 측정 데이타를 읽어 들이는 모듈

③ Manager : meter를 configure하고 meter reader를 제어하는 관리 모듈

④ Analysis Application : 측정데이타를 분석하여 사용자가 원하는 형태로 출력하는 모듈

RTFM에서는 meter가 측정한 데이타를 가져오기 위한 방법으로 SNMP를 사용하며, 이를 위한 meter MIB를 정의하였다. 최근에는 meter MIB을 확장하는 작업이 진행 중이며, traffic flow와 이에 대한 동작을 정의하고 표현하기 위한 표준 언어의 작성도 함께 진행되고 있다. RTFM 또한 SNMP 지원 호스트에 대해서만 분석이 가능하며 특정 서비스에서 이용한 트래픽 양을 나타내지 못하며 네트워크 이용률, 에러율 등을 측정하지 못한다는 단점을 가진다.

3. 요구사항

네트워크 트래픽의 증가 원인으로는 첫째 네트워크 사용자 수의 증가와 둘째 다양한 네트워크 응용 프로그램의 개발 및 사용으로 인한 네트워크 사용량의 증가를 들 수 있다. 특히, 웹의 발전은 네트워크를 사용하지 않던 사람들에게 네트워크를 접해 볼 기회를 제공했을 뿐 아니라 이미 네트워크를 사용하던 사람들의 네트워크 사용양도 크게 높이고 있다. 네트워크 트래픽을 분석하기 위해서는 네트워크 분석기가 호스트 정보와 네트워크 트래픽 정보를 가지고 네트워크 트래픽의 주된 양을 어떤 호스트가 차지하는지, 네트워크를 사용하는 호스트 수는 어느 정도인지, 호스트가 주로 사용하고 있는 서비스는 무엇인지 등의 정보를 알려 줄 수 있어야 한다. 그러나 tcpdump[12]와 같은 분석기들은 텍스트 기반에서 실시간으로 호스트 정보, 프로토콜 정보, 트래픽 양 정보 등을 화면에 보여주는 것에 그치고 있어서, 어떤 호스트가 네트워크 트래픽의 대부분을 차지하는지, 사용되고 있는 호스트 수는 어느 정도인지 등의 문제를 파악하기 어렵다. 이러한 문제점을 생각해 볼 때 네트워크 모니터링 및 분석 시스템은 기본적으로 다음과 같은 요구사항을 만족해야 한다

① 호스트 정보 분석 기능이 있어야 한다. 특정 시간동안 어떤 호스트들이 네트워크 트래픽을 발생시켰고 그중 가장 많은 트래픽을 발생시킨 호스트는 무엇인지 알 수 있어야 한다.

② 현재 네트워크 상에서 제공되고 있는 서비스의 종류를 파악하기 위해서 사용되고 있는 전송 프로토콜 및 어플리케이션 계층의 프로토콜을 분석할 수 있어야 한다.

③ 관리자의 편의성을 위해 웹 기반의 사용자 인터페이스를 제공해야 한다.

요구사항 ①, ②, ③을 만족하는 네트워크 분석 시스템으로는 ntop[13]과 WebTrafMon이 있다.

ntop은 웹 기반에서 호스트를 기준으로 해서 각 호스트들이 어떤 프로토콜을 사용하는지 분석해 주는 기능이 있고 WebTrafMon도 웹 기반에 서 시간 별로 호스트 정보를 제공 하므로 둘 다 요구 사항 ①, ②, ③을 만족한다. 그러나 ntop이나 WebTrafMon은 하루 동안 매 시간 별 네트워크 트래픽 정보만을 제공하고 24시간이 지나가면 분석해 놓은 데이타가 없어지기 때문에 장기간의 네트워크 트래픽을 분석할 수 없다. 장기적인 네트워크 트래픽 분석 계획을 세우기 위해서는 매 시간별 데이타 뿐만 아니라 하루 단위, 한달 단위, 일년 단위 등으로 네트워크 트래픽을 분석해 줄 수 있어야 한다. 장기적인 네트워크 분석기로는 MRTG가 있으나 관련연구에서 소개했듯이 MRTG는 네트워크 트래픽의 총량을 보여 줄 뿐 요구사항 ①, ②같은 정보를 제공해 주지 않기 때문에 네트워크 트래픽의 정확한 원인을 진단할 수 없다.

ntop과 WebTrafMon으로는 분리된 네트워크의 트래픽을 분석할 수 없다. 분리된 네트워크 트래픽을 분석하기 위해서는 한 노드와 다른 노드의 네트워크 트래픽을 합쳐서 분석할 수 있어야 한다. 그리고 패킷 캡쳐와 패킷 분석을 동시에 하므로 시스템 과부하시 패킷 손실이 발생한다. 또한 네트워크에서 트래픽이 과다하게 발생할 때 이를 조절해 줄 수 있는 기능이 없다. 이러한 문제들을 고려해 볼 때 요구사항 ①, ②, ③ 이외의 다음과 같은 요구사항이 필요하다.

④ 네트워크 트래픽을 매시, 매일, 매달, 매년 단위 등으로 장기간 분석해서 자동으로 보여줄 수 있어야 한다.

⑤ 여러 네트워크 노드에서의 네트워크 트래픽을 합쳐서 분석할 수 있어야 한다.

⑥ 패킷 캡쳐가 독립적으로 이루어져 패킷 분석으로 인한 시스템 과부하시의 패킷 손실이 없어야 한다.

⑦ 트래픽 조절 기능이 있어야 한다.

본 논문에서는 요구사항 ①, ②, ③뿐만 아니라 ④, ⑤, ⑥, ⑦도 반영할 수 있는 네트워크 모니터링 및 분석 시스템을 설계하고 구현한다. 이 시스템의 가장 큰 특징은 요구사항 ④, ⑤, ⑥을 반영하기 위해 패킷 모니

터링 시스템, 관리서버, 분석 시스템을 각각 모듈로 분리하여 동작시키며, ⑦을 반영하기 위해 TCP 패킷에 대해 윈도우 조절(window size control)기능을 수행한다.

4. 제안 시스템

본 논문에서 구현한 WebTraMAS(Web-based Traffic Monitoring and Analysis System)는 네트워크 상의 자원들을 관리하기 위하여 관리 대상을 설정하고, 이들에 대한 정보를 수집, 분석하고, 분석된 자료를 기반으로 원활한 트래픽의 흐름이 이루어지도록 하는 웹 기반의 그래픽 사용자 인터페이스 시스템을 의미한다. 시스템 구성은 크게 관리서버, 트래픽 모니터링 에이전트, 트래픽 분석 에이전트, Configurator, 클라이언트 프로그램으로 구성되어 있다. 그림 1에서 WebTraMAS의 전체 구성도를 보인다.

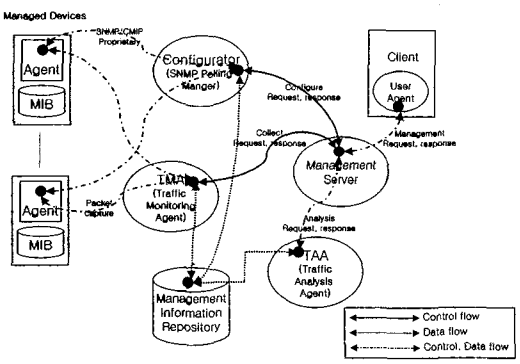

그림 1 전체 시스템 구성도

4.1 관리 서버(Management Server)

관리 서버는 클라이언트로부터의 요구를 처리하며 이러한 처리를 위한 연결 설정이나 메시지 생성, 각 요청별 데이타 처리 및 전송, 분석 항목에 대한 처리, TAA와의 연결설정 및 요청 정보 전달, 이미지 생성, 분석 결과 웹 페이지 생성, 요청 정보 관리 등을 처리한다. 관리서버의 동작과정을 보면 초기 상태, 대기 상태, 결과 처리 상태, TAA연결 상태, GUI 처리 상태의 5가지 상태를 가진다. 초기 상태는 서버를 가동시켜 데몬으로서 클라이언트의 연결을 기다리고 있는 상태이고, 이 상태에서 클라이언트로부터 접속 요구가 들어오면 서버는 대기 상태로 가서 클라이언트로부터의 요구 메시지를 기다린다. 클라이언트로부터 메시지를 수신하면 메시지 종류에 따라 각 요구를 처리할 수 있는 결과 처리 상태로 가고, 여기에서 필요에 따라 TAA와의 연결을 통해 클라이언트의 요청을 처리하는 TAA 연결 상태로 가거나 HTML&Graph처리를 위해 GUI처리상태로 간다. 관리 서버의 상태천이와 동작은 그림 2에서 보인다.

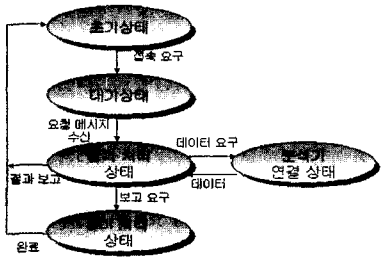

그림 2 관리 서버 구성도

4.2 트래픽 모니터링 에이전트(TMA)

망 관리자에게 보다 정확한 네트워크 트래픽 현황을 보여주기 위해서는 가능한 다량, 다중의 정보를 수집하여야 한다. 네트워크의 트래픽에 대한 정보는 크게 두 가지로 나뉠 수 있다.

첫째, 모니터링 중인 네트워크와 외부 네트워크 간에 전송된 총 트래픽 양이 있다. 이 정보는 모니터링 중인 네트워크가 과부하 상태에 빠졌을 때 외부 망과의 연계 때문인지 아니면 모니터링 중인 네트워크 내부의 트래픽 때문인지 구분을 명확히 하게 한다.

둘째, 모니터링 중인 네트워크에서 전송중인 트래픽의 네트워크 서비스별 양이다. 이 정보는 모니터링 중인 네트워크가 과부하 상태일 때 어느 서비스에 영향을 가장 많이 받았는지를 알 수 있게 된다. WebTraMAS는 위의 두 가지 정보를 얻는데 SNMP와 패킷 획득 방법을 사용한다. SNMP는 모니터링 중인 서브 네트워크에서 외부로 입출력되는 총 패킷 양을 모니터링 하는데 사용하며, 라우터 혹은 ATM Switch와 같이 SNMP Agent가 설치된 네트워크 장비의 특정 포트(port)를 모니터링 한다. 기본적으로 이 모듈에서는 유닉스 시간 데몬(Timing Daemon)인 Cron에 폴링 스크립트와 폴링 기간의 정보를 입력으로 받아 CF(Crone File)을 생성하고, 이를 Cron에 넘겨 일정 기간 동안 폴링 스크립트를 수행시켜 분석에 필요한 자료를 수집한다. 패킷 획득은 모니터링 중인 네트워크에서 전송되고 있는 트래픽을 서비스 단위로 검사하는데 사용되며, Libpcap 라이브러리를 이용하여 구현한다. Libpcap 라이브러리는 유닉스 계열의 다양한 운영체제에서 쓸수 있도록 운영체제에 독립적인 API를 제공하는 장점이 있어 대부분의 네트워

크 분석기에 사용된다. 패킷 획득 시스템의 패킷 수집과 데이타베이스 저장 부분은 별도의 프로세스로 동작을 하며 둘 사이의 데이타 공유를 위하여 공유 메모리 기법을 사용한다.

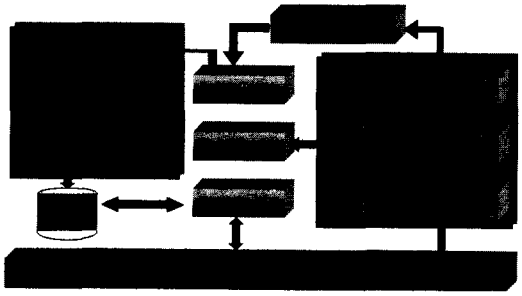

그림 3 트래픽 모니터링 에이전트 구성도

그림 3은 트래픽 모니터링 에이전트의 구성도로 패킷 획득 모듈은 네트워크 디바이스에서 패킷 필터링 과정을 거쳐서 패킷을 획득하고 일련의 루틴에 따라 헤더를 풀어낸 후 공유 메모리 버퍼에 저장을 한다. 이때, 데이타베이스 변환 모듈에서 이 데이타를 읽어서 데이타베이스 테이블에 맞는 형태로 일련의 변환 과정을 거쳐 일정 간격으로 데이타베이스에 저장을 한다.

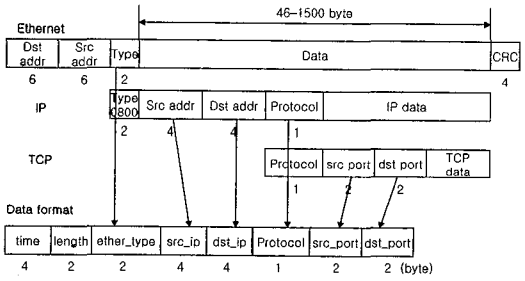

그림 4 패킷 획득의 데이타 형태

그림 4는 이더넷 프레임에서 정보를 추출해서 데이타베이스에 저장되는 데이타 형태를 설명한다. 데이타 형태에서 time은 패킷을 캡처해서 데이타베이스에 저장할 때의 시간이며, length는 패킷 전체의 크기에 CRC값(4 byte)을 더한 값이다. time과 length를 제외한 나머지 정보는 모두 패킷에서 가져오는데 ether_type은 이더넷 헤더에서 정보를 가져오고, src_ip, dst_ip, protocol 등은 IP 프로토콜 헤더에서, src_port, dst_port 등은 TCP/ UDP 헤더에서 정보를 가져온다. 데이타 포맷의 일정 필드를 사용하지 않는 프로토콜은 0으로 채운다.

여러 네트워크 노드의 트래픽을 합쳐서 분석을 하기 위해서는 모니터링 에이전트가 여러 곳에서 실행되어야 한다. 따라서 트래픽 모니터링 에이전트와 트래픽 분석 에이전트를 독립적인 모듈로 설계하여 시스템의 과부하가 일어나지 않도록 한다.

4.3 트래픽 분석 에이전트(TAA)

트래픽 모니터링 에이전트에 의해 수집된 트래픽 정보는 수집된 시간과 함께 데이타베이스에 저장되며, 트래픽 분석 에이전트에 의해 트래픽 분석이 이루어진다. 트래픽 분석 에이전트는 트래픽 모니터링 에이전트가 저장한 정보를 IP와 non-IP 데이타로 분류해서 데이타베이스에 저장하는 모듈과 수집된 트래픽 데이타에 대한 통계적 특성을 계산해내는 기능 모듈로 년, 월, 일 단위의 트래픽 양을 계산하는 통계 분석기로 구성된다. 망 관리자는 통계 분석기를 통하여 년, 월, 일 단위의 트래픽 양을 점검해 볼 수 있으며, 평균 트래픽이 가장 높은 기간을 파악할 수 있고, 또한 해당 시점에서 가장 많이 사용된 네트워크 서비스 정보와 가장 많은 대역을 사용한 네트워크 단말을 알 수 있다. 이 정보를 바탕으로 네트워크에 부하를 가장 많이 주는 서비스와 단말에 대한 관리가 가능하다.

4.4 Configurator

네트워크 모니터링에 필요한 네트워크 장비들의 정보와 네트워크상의 구성정보를 읽어서 데이타베이스에 저장한다. 네트워크 구성정보는 모니터링에 필요한 장치 정보들뿐만 아니라 각 장치의 네트워크 인터페이스 정보들을 포함한다. Configurator는 ping과 snmpd를 함께 체크하고 결과 값을 데이타베이스에 저장하는 데몬 형태로 구성되며, AutoScan기능을 통하여 현재의 네트워크에서 관리 대상이 되는 호스트 이름과 시스템의 ON/OFF상황, 관리대상 IP 주소의 정보와 SNMP 데몬의 유무를 확인한다. Configurator에 수집된 정보는 동적 서브넷 리포팅 기능에 이용되며 동작방식은 다음과 같다.

① default로 주어진 seed값을 읽는다.
② 데몬 프로세스를 시작한다.
③ snmptable을 이용하여 seed값에 해당하는 IP 주소의 라우터 테이블 정보를 읽는다.
④ 라우터 테이블의 서브네트워크 IP들의 정보를 데이타베이스에 저장한다.
⑤ 데이타베이스 정보를 이용하여 각 서브네트워크 마다 ping과 snmpget을 전송한다.
⑥ IP를 키 값으로 하여 ④의 결과 값을 디렉토리 구조로 저장한다.

⑦ 모든 서브네트워크 IP를 체크한 다음 랜덤하게 ping과 snmpget을 체크한다.

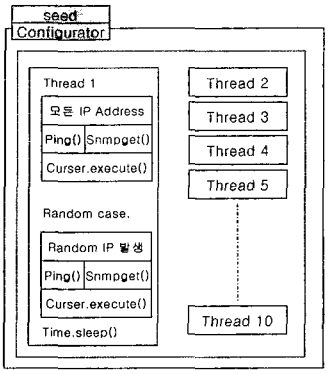

그림 5 Configurator 구조

④번 실행 시에 여러 개의 thread를 사용하여 하나의 서브넷 안에 있는 각각의 호스트들을 체크하는 시간을 줄였으며, 또한 랜덤하게 ping과 snmpget을 체크할 때, 한 번 발생한 난수에 대해 다음번에 발생하지 않도록 제거를 시켜 모든 IP들에게 패킷이 균등하게 전달되도록 구현 한다. 그림 5에서 Configurator의 설계화면을 보인다.

4.5 클라이언트 프로그램

WebTraMAS의 클라이언트 프로그램은 사용자 GUI와 사용자 에이전트(user agent)로 구성된다. 사용자 GUI는 기본적으로 웹 브라우저를 이용한다. 이는 웹 서비스가 가능한 장소 어디에서나 네트워크 트래픽을 점검할 수 있게 하므로 망 관리자가 특정 관리 시스템에 상주하여야 하는 불편을 없앨 수 있다. 네트워크 트래픽 정보를 사용자에게 편리하게 제공하기 위해 두 가지 방식으로 구현되었다. SNMP 지원 호스트에 대한 사용자 GUI는 C언어와 gd 라이브러리를 이용하였으며, 패킷 획득에 관련된 사용자 GUI는 Java Swing AIP를 사용하여 네트워크 트래픽의 양을 보여주는 그래프를 그렸다. 또한 데이타베이스의 내용을 웹 서버로 전달하기 위해 C-CGI와 JDBC를 사용하여 웹과 데이타베이스를 연동하였다.

사용자 에이전트는 클라이언트 프로그램과 관리 서버 사이에 존재하여 메시지 전달을 중계하며, 외부 유저가 WebTraMAS를 사용하는 망 관리자의 접근 권한을 감사한다. 사용자 에이전트의 인증 기능은 전통적인 Login ID와 패스워드에 의한 사용자 인증 방법을 사용한다.

5. 트래픽 분석 파라메터 정의

네트워크 트래픽 모니터링의 가장 중요한 목적 중에 하나는 사용자에게 효율적이고 신뢰성있는 네트워크 환경을 제공하는 것이다. 이러한 목적을 이루기 위해서는 네트워크 장치들의 지속적인 모니터링을 통하여 획득한 패킷의 분석이 이루어 져야 하며, 분석된 결과는 네트워크 관리자에게 자동적으로 보고 되어야 한다. 이번 단락에서는 본 논문에서 제시한 망 관리 시스템에서 사용한 트래픽 분석 파라메터 계산식을 제시한다.

5.1 FDDI 이용률

FDDI 네트워크의 이용률을 측정하기 위해서는 FDDI 네트워크 내의 모든 라우터의 입·출력 트래픽 량을 측정하여 그들의 합을 2로 나누어 줘야 한다. FDDI 네트워크의 최대 전송률은 100Mbps이기 때문에 식 (1)과 같이 이용률을 측정할 수 있다.

일반적으로 FDDI 네트워크의 이용률이 90% 이상일 때 과부하로 측정할 수 있다. 일시적으로 이용률이 90%를 초과할 때는 큰 문제가 되지 않지만, 평균 이용률이 90%이상일 때는 네트워크 전체 과부할 상태로 판명하여 네트워크를 재구축하거나 업그레이드가 이뤄져야만 한다.

Formula (1) : FDDI Utilization(%)

$$\frac{1}{2}\sum_{devices}[(total\_bit\_sent + total\_bits\_received)/bandwidth]$$

$$\frac{1}{2}\sum_{devices}\left[\frac{[ifInOctets_{(x+t)} - ifInOctets_{(x)} + ifOutOctet_{(x+t)} - ifOutOctet_{(x)}]*8}{(sysUpTime_{(x+t)} - sysUpTime_{(x)})*ifSpeed*100}\right]$$

본 논문에서 제시하는 식 (1) - (4)에서 x는 이전 폴링 시간을 의미하며, t는 폴링 주기를 의미한다.

*ifInOctets*은 인터페이스에 수신된 옥텟의 총 수, *ifOutOctets*은 인터페이스를 벗어나서 전송되는 옥텟의 총 수, *sysUpTime*은 시스템이 마지막으로 재 초기화된 이후의 시간, *ifSpeed*는 인터페이스의 현재 대역폭으로 bps(bit per second)로 표시, *ifInError*는 패킷에 포함된 오류 때문에 상위 계층에 전달되지 못한 도착 패킷의 수, *totalPktIn*은 전체 도착 패킷의 수를 의미한다.

5.2 이더넷 이용률

이더넷은 브로드캐스팅(broadcasting) 전송방식(특히, CSMA/CD)을 사용한다. 본 논문에서는 FDDI 백본 상에서 네트워크 인터페이스의 입·출력 트래픽을 분석할 뿐만 아니라, 네트워크 디바이스의 상태를 모니터링 한다. 네트워크 트래픽을 분석하기 위해서 본 논문에서는

각각의 컴포넌트 네트워크에 따라 분석 식을 다르게 사용한다. 이더넷 네트워크의 이용률을 측정하기 위해서는 모든 입·출력 트래픽을 더한 다음, 트래픽을 총 합을 최대 전송속도로 나눠주면 된다. 식 (2)에서 이더넷 트래픽 분석 식을 보인다.

Formula (2) : Ethernet Utilization(%)

$$[(Total\_bit\_sent + total\_bits\_received)/bandwidth]$$
$$\left[\frac{[ifInOctets_{(x+t)} - ifInOctets_{(x)} + ifOutOctets_{(x+t)} - ifOutOctets_{(x)}]*8}{(sysUpTime_{(x+t)} - sysUpTime_{(x)})*ifSpeed*100}\right]$$

5.3 시리얼 링크 이용률

시리얼 링크 이용률의 경우 전송방식이 반 이중 방식(half-duplex)과 전 이중 방식(full-duplex)의 두 가지 경우가 있다. 반 이중 시리얼 링크의 경우 입력 트래픽의 합이 링크의 이용률이 되며, 전 이중 시리얼 링크의 경우, 링크의 연결 방향에 따라 두 개의 라인으로 구성된다. 식 (3)에서 각 링크 연결 방향에 따라 전 이중 시리얼 링크의 이용률 측정을 보인다.

일반적으로 시리얼 링크의 경우 이용률이 90%이상이면 네트워크 과부하 상태로 판명한다. FDDI 네트워크와 마찬가지로, 네트워크 이용률이 일시적으로 90%를 초과하는 경우에는 큰 문제가 없으나. 평균 이용률이 90%이상 초과 시에는 네트워크에 심각한 문제가 있다고 판단하여 네트워크 재구축이나 업그레이드가 반드시 필요하다.

Formula (3) : Serial Link Utilization(%)

$$Max[(total\_bit\_sent + total\_bits\_received)/bandwidth]$$
$$Max\left[\frac{[(ifInOctets_{(x+t)} - ifInOctets_{(x)}),(ifOutOctets_{(x+t)} - ifOutOctets_{(x)})]*8}{(sysUpTime_{(x+t,x)} - sysUpTime_{(x)})*ifSpeed*100}\right]$$

또 다른 네트워크 성능 측정치로 트래픽 에러율을 들 수 있다. 일반적으로 에러율이 전체 대역폭에 1%이상이면 네트워크 관리자는 네트워크 인터페이스를 조사해 봐야 한다. 식 (4)에서 에러률 측정을 보인다.

Formula(4) : Error Rate(%)

$$\left[\frac{ifInError_{(x+t)} - ifInError_{(x)}}{totalPktIn_{(x+t)} - totalPktIn_{(x)}}\right]$$

where
totalPktIn = ifInUcastPkts + ifInBroadcastPkts + ifInMulticastPkts

식 (4)에서 *totalPktIn*은 전체 도착 패킷의 수를 의미하며, 도착 유니캐스트 패킷의 수와 도착 브로드캐스트 패킷의 수 그리고 도착 멀티캐스트 패킷의 수를 합하여 측정할 수 있다.

6. 시스템 구현

본 논문에서 설계한 망 관리 시스템의 구현은 SNMP를 지원하는 네트워크 디바이스들을 관리하기 위한 부분과 SNMP를 지원하지 않는 네트워크 디바이스를 관리할 수 있도록 크게 두 가지 형태로 구성되며 개발 환경은 다음과 같다.

6.1 개발환경

웹 기반의 네트워크 관리 시스템은 다음과 같은 환경에서 개발하였다. Linux Kernel 2.2.17 운영체제와 Intel Pentium-III 800Mhz, 128M 메모리의 컴퓨터를 관리서버, 모니터링 에이전트, 분석 에이전트로 각 1대씩 사용하였으며, 클라이언트를 위해 Window 2000 서버 1대, 그리고 각 네트워크 노드로 5대의 PC와 라우터, 스위치, 허브 장비를 사용하였다. 트래픽 모니터링 툴로 LibPcap 0.4a6을 사용하고, SNMP 에이전트는 Ucd-snmp-4.2를 사용하였다. 그래픽 표현을 위해서 gcc 9.61, gdchart 0.94, Python1.5.2, Pmw0.8이 사용되었다. 실시간 트래픽 정보 표현을 위해서는 JDK2.0과 Swing 라이브러리가 사용되었다. 그 외에 웹 서버로는 Apache Web Server 1.2.9와 CGI 프로그램 작성을 위해 Perl 5.004_01, HTML을 사용하였다. 데이타베이스로는 MySQL 3.23.41이 사용되었으며, 웹과 DB연동을 위하여 JDBC와 C-CGI가 사용되었다.

6.2 SNMP 모니터 구현

SNMP를 지원하는 네트워크 디바이스들을 관리하기 위해 MRTG(Multi-Router Traffic Grapher) 라이브러리를 이용하여 추가적인 기능을 구현하였다. 추가된 기능으로는 동적 서브넷 리포팅 기능, CPU 로드 모니터링 기능 등을 구현하였다. 동적 서브넷 리포팅 기능은 Configurator에 의해 수집된 IP 정보를 기반으로 라우터 테이블의 값을 리스트에서 파싱하여 데이타베이스에 저장한 다음 저장된 IP정보를 기반으로 Python과 Pmw를 이용하여 구현한다. 그림 6에서 동적 서브넷 리포팅 실행 화면을 보인다.

CPU 로드 모니터링은 Configurator에서 ping 데몬을 이용하여 SNMP 데몬의 유무를 판단한 다음에 SNMP 지원 호스트나 장비에 대해 snmpget()을 수행한다. 표 1에서 수행 코드를 보인다.

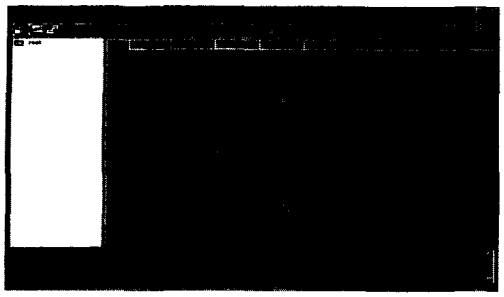

그림 6 동적 서브넷 리포팅 실행화면

표 1 CPU 로드 모니터링 수행 코드

```
#!/usr/local/bin/perl/perl
1. system "/usr/local/bin/snmpget 165.229.193.46
   \ public .1.3.6.1.4.1.2021.10.1.3.1
   | awk '{print $3}' > temp";
2. system "/usr/local/bin/snmpget 165.229.193.46
   \ public .1.3.6.1.4.1.2021.10.1.3.2
   | awk '{print $3}' >> temp";
3. open( fileHandle, "<temp" )
   || die "Cannot open $fileName.\n";
4. $value1 = <fileHandle> ; # 1분 평균 load
5. $value2 = <fileHandle> ; # 5분 평균 load
6. chop($value1); # 개행문자 버림
7. chop($value2); # 개행문자 버림
8. close( fileHandle );
9. $one_minute = $value1 * 100 ; #백분율로 변경
10. five_minute = $value2 * 100 ;
11. print "$one_minute\n$five_minute\n";#결과출력
```

그림 7에서 SNMP 모니터의 전체적인 동작 과정을 보인다.

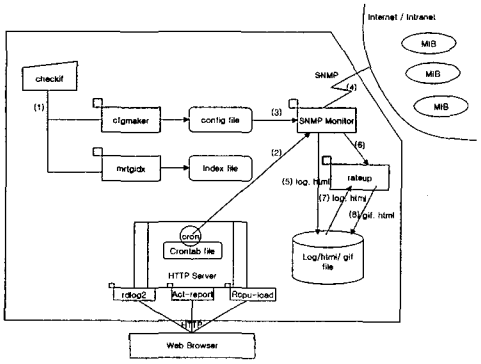

그림 7 SNMP 모니터 시스템 구조

그림 7은 전체 시스템 구조에서 SNMP지원 네트워크 디바이스를 관리하는 단계를 설명하고 있으며, 각각의 세부 모듈과 동작 과정은 다음과 같다.

- cfgmaker : 앞에서 설명한 Configurator와 동일한 기능을 수행한다. 전체 네트워크의 네트워크 디바이스를 발견하고, 데이타베이스를 구성하는 역할을 수행한다.
- mrtgidx : 네트워크 구성 데이타베이스를 이용하여 각각의 네트워크 디바이스들의 구성 정보를 파악할 수 있는 HTML파일을 생성한다.
- checkif : 네트워크 인터페이스를 찾아낸다.
- rateup : 모니터링 된 파일들을 검색하여 그래프를 생성한다.
- active-report : 모니터링 된 파일들을 검색하여 리포트를 작성한다.
- rdlog2 : 네트워크 맵을 생성시켜주고, 링크 이용률, 에러율 등을 계산한다.
- rcpu-load : CPU 로드 그래프를 생성하고, 네트워크 디바이스의 임계 값을 알려준다.
- contab : 에이전트 폴링 시간과 리포트 생성에 필요한 시간을 생성시켜주는 스크립트

(1) 단계 : 초기 네트워크 디바이스 검색 단계

네트워크 디바이스들을 모니터링 하기 위해 네트워크 디바이스의 기본 정보를 가지고 초기 네트워크 구성 파일을 생성한다. 여기에서는 cfgmaker와 checkif 모듈이 사용된다. 그와 동시에 mitgidx모듈은 네트워크 디바이스들에 대한 하이퍼링크 정보를 포함하는 HTML 파일을 생성한다.

(2) 단계

시스템 Crontab파일 내에 정의된 trigger함수를 기반으로 다양한 관리 함수가 trigger된다.

(3)-(5) 단계

cfgmaker에 의해 생성된 구성 정보를 이용하여 SNMP 모니터는 각 관리 에이전트들에게 보낼 request 들을 생성하여 보내고 에이전트들로부터 트래픽 관련 데이타를 받는다. 받은 데이타는 데이타베이스에 로그파일과 같은 형태로 저장되어 트래픽 분석 자료로 이용된다.

에이전트에게 보낼 request 구조를 생성하기 위해 Configurator에서 얻어온 정보를 기반으로 SNMP 지원 호스트에 대해 표 2와 같이 네트워크 디바이스에 object ID 값을 알아내기 위해 에이전트 명과 community 명을 이용하여 snmpget 쿼리를 보낸다. snmpget에 의해 object ID값을 가져 온 다음 각 ibject ID를 상수로 하여 필요한 정보를 가져와서 데이타베이스에 저장한다.

(6)-(8) 단계

SNMP 모니터는 로그 파일형태의 데이타를 검색하기

위하여 rateup 프로세스를 트리거하고 트래픽 분석 작업을 수행한다. 분석된 트래픽 데이타는 그래프나 HTML 문서형태로 표현된다. 그림 8에서 분석된 데이타를 기반으로 생성된 트래픽 모니터링 그래프를 보인다.

표 2 snmpget 질의 형식

```
def snmpget (agent, community, objids):
    # initialize objid & vals lists
      encoded_oids = []
      encoded_vals = []
    # create a SNMP session
      session = pysnmp.session(agent, community)
    # encode objids
      for objid in objids:
    # encode an objid
      encoded_oids.append(session.encode_oid
      (session.str2nums(objid)))
    # build a packet
      packet = session.encode_request
('GETREQUEST', encoded_oids,
encoded_vals)
    # send SNMP request and recieve a response
      packet = session.send_and_receive (packet)
    # parse a response packet
      (encoded_oids, encoded_vals) =
session.decode_response (packet)
    # decode objs & vals
      index = 0
      return
session.decode_value(encoded_vals[index])
```

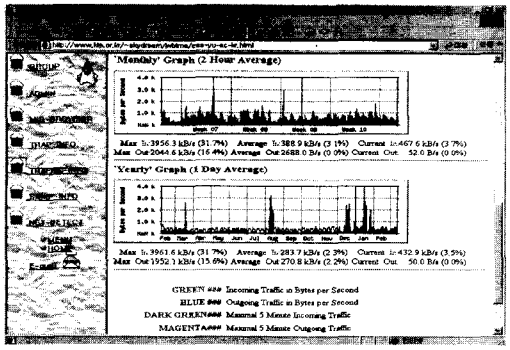

그림 8 트패릭 모니터링 그래프

그림 8에서는 일별, 주별, 월별, 년별 트래픽 량을 모니터링 한 결과로서 녹색 라인은 초당 입력 트래픽을 바이트 단위로 나타내고, 푸른색 라인은 초당 출력 트래픽 량을 바이트 단위로 나타낸다. 진한 녹색과 심홍색 그래프는 5분 동안의 최대 입출력 트래픽 량을 나타내고 있다.

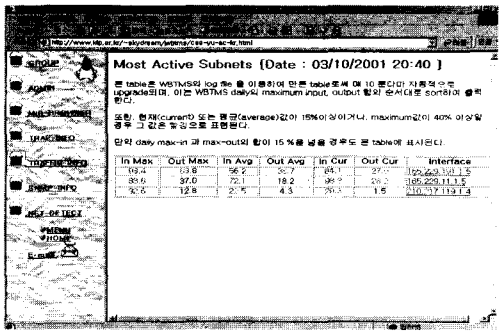

그림 9 네트워크 임계치 리포트

(9) 단계

rdlog2, Act-report, Rcpu-load는 맵 생성기, 링크 사용률, 라우터 CPU 로드 모니터링에 사용된다. 그림 9에서 Act-report를 이용하여 네트워크 디바이스의 링크 사용률을 표현한 형태를 보인다.

그림 9는 SNMP 모니터의 로그 파일을 이용하여 작성된 테이블로써 매 10분마다 자동적으로 업그레이드되며, 이는 일별 최대 입출력 트래픽 량의 합의 순서대로 정렬하여 출력한다. 또한, 현재 또는 평균 트래픽 값이 15%이상이거나 최대 값이 40%이상일 경우 그 값은 빨강으로 표시되며 트래픽의 흐름제어가 필요하다.

6.3 패킷 획득 시스템 구현

패킷 획득 시스템은 모니터링 중인 네트워크에서 전송되고 있는 트래픽을 서비스 단위로 검사하는데 사용되며, LibPcap 라이브러리와 Java를 이용하여 구현한다.

(1) 망 관리자는 웹 브라우저를 통하여 관리서버 쪽으로 start_monitoring() 원격지 메소드를 통하여 패킷 획득을 실행시킨다.
(2) TMA는 모니터링 시작 명령을 전달받으면 GCC와 LibPcap 라이브러리를 이용하여 관리하고 있는 LAN Segment를 모니터링 한다.
(3) 모니터링 결과는 누적 분석을 위하여 데이타베이스에 저장된다.
(4) TAA는 TMA가 저장한 원시 모니터링 데이타를 검색하여 분석한다.

패킷 획득 방식의 네트워크 모니터링은 네트워크에서 전송중인 트래픽의 네트워크 서비스별 양뿐만 아니라 패킷 길이별 분포 등을 측정할 수 있다. 이 정보는 모니터링 중인 네트워크가 과부하 상태일 때 어느 서비스에 영향을 가장 많이 받았는지, 그리고 패킷 길이 분포가 어떠할 때 네트워크 과부하 상태가 발생하는지를 알 수 있게 된다.

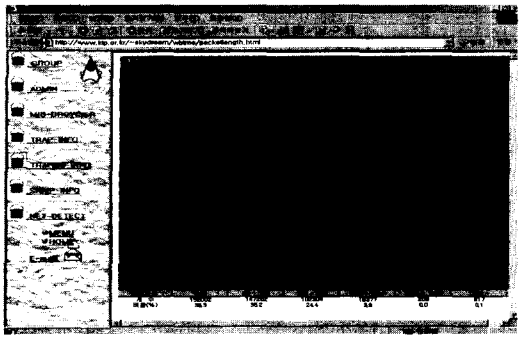

그림 10 패킷 길이별 분포현황

그림 10의 패킷 길이 분포는 네트워크에 과다 트래픽을 유발하는 패킷의 길이별 분포를 찾아내어 분포가 큰 패킷에 대해 흐름제어를 수행한다.

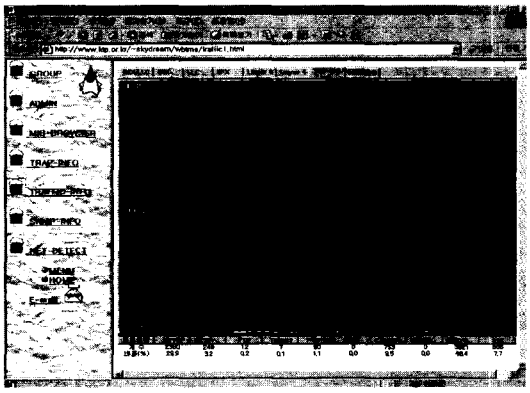

그림 11 프로토콜별 분포현황

그림 11의 프로토콜별 트래픽 분포 현황을 조사하여 관리 서버에서는 특정 프로토콜별 사용 내역을 감시할 수 있다.

6.4 트래픽 조절 시스템 구현

TCP 패킷의 경우 헤더에는 수신 측의 윈도우 버퍼 크기가 들어가게 된다. 이 때 윈도우 크기가 매우 클 경우 송신 측에서는 한꺼번에 윈도우 크기에 해당하는 만큼의 데이타를 전송하게 된다. 이로 인해 내부 네트워크에는 많은 패킷이 발생하여 충돌을 유발하게 된다. 본 논문에서는 중간 노드에서 수신 측의 TCP 패킷의 윈도우 사이즈를 줄여 변조함으로써 송신 측의 데이타 전송을 제어할 수 있다. 그림 12에서 중간노드에서 TCP패킷의 변조 과정을 보인다. 송신 측에서 데이타를 보낸 다음 이에 대한 TCP패킷의 윈도우 사이즈가 4000일 경우 윈도우 크기 제어를 거쳐 TCP패킷을 위한 윈도우 사이즈를 25%정도 줄임으로써 송신 측의 데이타 전송량을 줄인다. 본 논문에서는 패킷 구분기로부터 윈도우 크기제어 모듈이 불려지면 이 모듈에서는 정책구조체로부터 해당 클래스의 윈도우 비율(Window Rate)값을 얻어내어 TCP 패킷의 윈도우 크기를 윈도우 비율만큼 줄여준다.

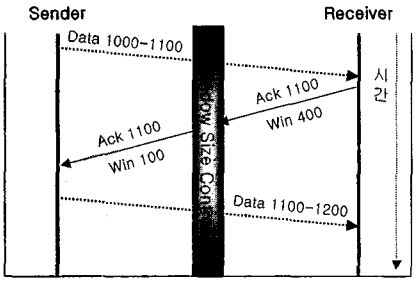

그림 12 중간노드에서 TCP패킷 변조

이때 TCP 헤더가 변경되었으므로 TCP 체크썸(Checksum)까지 수행한다. 그림 13에서 각 서비스 별 실시간 패킷 분석 현황과 트래픽 조절 화면을 보인다.

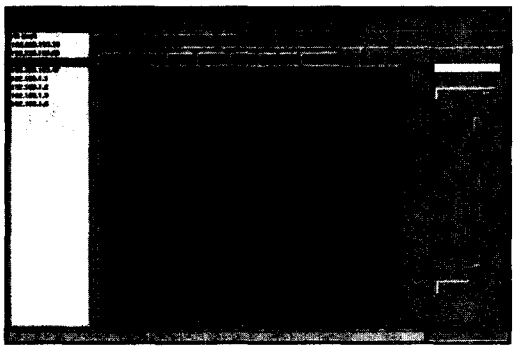

그림 13 각 서비스별 실시간 패킷 분석 현황과 트래픽 조절 화면

그림 13에서 상단의 버튼을 이용하여 각 서비스별 실시간 분석이 가능하며, 우측의 스크롤 바를 이용하여 TCP 패킷의 윈도우 사이즈 제어 기능을 수행할 수 있다. 상단 우측의 대역폭 제어 부분은 UDP 패킷의 전송 품질 보장을 위해 현재 구현중인 부분이다.

표 3에서 본 논문에서 설계하고 구현한 WebTraMAS와 기존의 네트워크 트래픽 분석 시스템을 비교 설명한다.

표 3 네트워크 트래픽 분석 시스템 비교

	분석 방식	분석범위	여러노드 패킷캡처	호스트 분석 기능	웹 기반	트래픽 제어 기능
MRTG	일괄 처리	현재상황 매시, 매일, 매주, 매달	no	no	yes	no
ntop	일괄 처리	현재상황 매시	no	yes	yes	no
ethereal [14]	실시간, 일괄 처리	현재상황	no	yes	no	no
tcpdump	실시간	현재상황	no	yes	no	no
RTFM	실시간	현재상황	no	no	yes	no
WebTraf Mon	일괄 처리	현재상황, 매시	no	yes	yes	no

표 3에서 '분석방식'을 '실시간'과 '일괄처리'로 나누었다. 실시간은 패킷 하나가 들어올 때마다 정보를 보여주는 방식을 말하며 일관처리는 일정한 시간동안의 트래픽을 모아서 분석한뒤 보여주는 방식을 말한다. '분석범위'의 매시는 과거 1시간 동안을 분석해서 보여주는 것을 말한다. '여러 노드 패킷 캡처'는 여러 네트워크 노드의 트래픽을 합쳐서 분석해 줄 수 있는 지의 여부이며 '호스트 분석 기능'은 호스트 별 패킷 사용량 등의 정보를 제공하는 것을 의미한다. 표 3에서 보면 분석 범위가 장기간이면서 호스트 정보를 제공해 주는 프로그램이 없다는 것을 발견 할 수 있다. 일정한 간격의 장기간 네트워크 트래픽 정보를 제공하는 것은 MRTG밖에 없는데 MRTG는 호스트 정보를 제공하지 못한다. 따라서 장기간의 네트워크 트래픽 정보를 제공해 주면서 호스트 정보를 분석해 줄 수 있는 프로그램이 필요하며 WebTraMAS는 분석 방식을 일괄처리, 실시간으로 수행하며, 분석 범위는 매시, 매일, 매달, 매년으로 정하여 트래픽 분석을 수행하였다. 또한 여러 노드에서 패킷 캡처를 지원하도록 설계하였으며, 트래픽 제어 기능을 수행할 수 있도록 설계하였다.

본 논문에서 구현한 WebTraMAS를 통한 네트워크 트래픽 관리를 실험하기 위해서, Windows Media Encoder 7.0을 사용하여 높은 대역폭을 가지는 동영상 스트림 데이타 타입의 advanced stream format(asf) 파일을 생성하여 전송 데이타로 이용하였으며, 각 종단 시스템에서는 패킷 생성기를 사용하여 임의로 트래픽을 발생시켰다. 그림 14, 그림 15는 이러한 실험 데이타를 이용하여 실제 트래픽 제어를 적용한 결과를 그래프로 나타낸 것이다. 발생된 트래픽은 모두 외부로 흘러나가는(outbound) 패킷이며, 그 대역폭은 1.2Mbyte/sec 보다 크도록 하였다. 그림 14에서 각 클래스에 정책을 적용하기 전에는 동일한 전송 량으로 전송이 이루어지다가 정책 1에서 클래스 1과 클래스 2에 각각 70%, 30%의 비율로 전송 량을 조절한 결과 7:3의 비율로 전송이 이루어지는 것을 보이고 있다. 정책 2에서 전송 량을 각각 30%, 70% 비율로 재조정한 결과 전송 량이 3:7의 비율로 전송이 이루어짐을 볼 수 있다. 그림 15에서는 클래스 1, 클래스 2, 클래스 3, 클래스 4에 각각 40%, 30%, 15% 10%의 비율로 전송 량을 주었으며, 기본 클래스에는 5%의 전송 량을 배정한 경우의 그래프 화면이다. 그림 15에서와 같이 클래스 3, 클래스 4가 유휴상태에 있을 때 클래스 1과 클래스 2가 유휴 대역폭을 할당받아 사용하다가 클래스 3, 클래스 4가 유휴 상태에서 깨어나면 클래스 1과 클래스 2는 자신에게 주어진 대역폭을 사용함을 보여준다.

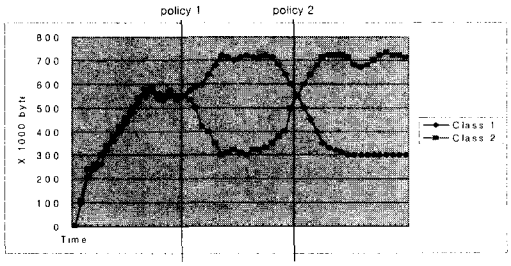

그림 14 정책 적용 전·후의 트래픽 그래프

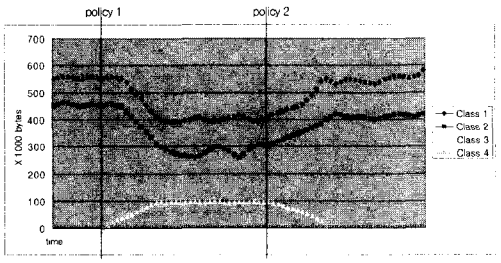

그림 15 타 클래스 휴지·활동 시 트래픽 그래프

7. 결 론

망 관리자는 본 논문에서 제안한 망 관리 시스템을 통하여 인터넷 혹은 인트라넷에서의 서비스별 트래픽 현황을 웹을 통해 손쉽게 파악할 수 있다. 이를 통하여 현재 망에 부하를 주고 있는 서비스를 찾아 낼 수 있으며, 차후 망을 관리하는데 근거 자료로 사용할 수 있다.

그리고 지정한 용량을 초과하는 트래픽이 발생했을

때 망 관리자는 본 논문에서 제안한 시스템의 보고 기능을 통해 실시간으로 알 수 있으므로 빠른 시간에 망을 정상화 할 수 있다. 또한 가장 많은 트래픽을 발생시키는 노드와 해당 노드가 사용 중인 서비스를 조사 할 수 있으며, 이를 통하여 망 관리자는 특정 노드의 네트워크 사용 내역을 감시할 수 있다. 이는 부정 사용자를 검출하여 네트워크 대역의 공평한 사용을 위한 기초 자료로 사용될 수 있다.

참 고 문 헌

[1] F. Barillaud, Luca Deri, Metin Feridun, "Network Management using Internet Technologies," Proc. of the 5th IEEE/IFIP International Symposium on Integrated Network Management (IM'97), San Diego CA, May 1997, pp. 61-70.

[2] Seong-Woo Kim, Young-Tak Kim, "TINA based Performance Management Architecture for Broadband Networks," ICOIN-14, pp. 2C-1.1 - 2C-1.6, Jan 2000.

[3] ITU-T Rec. M.3010, "Principles for a Telecommunications Management Network," Oct. 1992.

[4] Dicakara K. Udupa, "Network Management Systems Essentials," McGraw-Hill, 1996.

[5] David Perkins, Evan McGinnis, Understanding SNMP MIBs, Prentice Hall, 1997.

[6] J. Case. C. Prtridge, "Case Diagrams : A First Step to Diagrammed Management Information Bases," ACM Computer Communication Review, March 1989.

[7] K. McCloghrie, M. Rose, "Management In -formation Base for Network Management of TCP/IP- based Internets : MIB-II," RFC1213, March 1991.

[8] William Stallings, SNMP, SNMPv2, SNMPv3 and RMON : 3rd Ed. Addison -Wesley, 1999.

[9] Tobias Oetiker and Dave Rand, "MRTG : Multi Router Traffic Grapher," http://ee-staff.ethz.ch/~oetiker/webtools/mrtg

[10] 포항공과대학교, "WebTrafMon : Web-based Network Traffic Monitoring and Analysis System," http://amazone.postech.ac.kr/research/00/webtrafmon/english

[11] Auckland Univ, "RTFM : Realtime Traffic Flow Management," http://www.ackland.ac.kr/net/Internet/rtfm

[12] Lawrence Berkley National Laboratory, "tcpdump 3.5," http://www.tcpdump.org

[13] L.Deri and R.Carbone, "Monitoring Networks Using Ntop," Released paper in http://lica.ntop.org, Jan. 29th 2001.

[14] Ethereal webpage, http://www.ethereal.com

이 명 섭

1996년 경일대학교 공과대학 컴퓨터공학과(공학사). 1998년 영남대학교 공과대학 컴퓨터공학과(공학석사). 2002년 영남대학교 공과대학 컴퓨터공학과(박사수료). 1999년 ~ 2001년 경동정보대학 인터넷정보계열 전임강사. 관심 분야는 데이타 마이닝, 에이전트 시스템, 망관리, 네트워크 보안

박 창 현

1986년 경북대학교 공과대학 전자공학과 전산학전공(공학사). 1988년 서울대학교 자연과학대학 계산통계학과 전산학 전공(이학석사). 1992년 서울대학교 자연과학대학 계산통계학과 전산학 전공(이학박사). 1992년 ~ 1993년 서울대학교 컴퓨터신기술공동연구소 특별연구원. 1998년 ~ 1999년 University of Maryland, Institute of Advanced Computer Systems, Visiting Researcher. 1993년 ~현재 영남대학교 컴퓨터공학과 부교수. 관심 분야는 인공지능, 지식기반 시스템, 데이타 마이닝, 에이전트 시스템, 지능형 망관리 시스템