제목 : 분산 시스템 환경에서의 로드 밸런싱을 통한 웹기반 네트워크 트래픽 모니터링 및 분석
성명 : 홍순화, 김재영, 조범래, 홍원기
소속기관 : 포항공과 대학교 컴퓨터 공학과
키워드 : 네트워크 모니터링
주소 : 경북 포항시 포항공대 정보통신연구소 453호 DPNM 연구실
우편번호 : 790-784
전화 : 054-279-5654
전자우편 주소 : {padosori, jay, brcho, jwkhong}@ postech.ac.kr

분산 시스템 환경에서의 로드 밸런싱을 통한 웹기반 네트워크 트래픽 모니터링 및 분석

홍순화, 김재영, 조범래, 홍원기
{padosori, jay, brcho, jwkhong}@ postech.ac.kr

요약

최근 인터넷 사용자가 계속해서 증가하고 네트워크 기반의 응용 프로그램이 다양하게 개발됨에 따라 네트워크 트래픽이 증가하고 있다. 이에 따라 네트워크 트래픽을 분석하고 모니터링 할 수 있는 네트워크 분석기들이 소개되고 있다. 그러나 기존의 네트워크 분석기들은 실시간 혹은 매 시간 단위의 정보만을 확인 할 수 있어 장기간의 네트워크 트래픽 모니터링에는 적합하지 않으며 하나의 시스템에서 패킷 캡처와 패킷 분석을 동시에 하므로 시스템 과부하시 패킷 손실이 일어나고 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 없는 단점이 있다. 따라서 본 논문에서는 이러한 단점을 보완한 네트워크 모니터링 및 분석 시스템인 WebTrafMon II의 설계 및 구현을 설명한다. WebTrafMon II는 웹기반에서 호스트 정보, 프로토콜 정보, 어플리케이션 정보를 하루 동안 매 시간 별로 분석해 주는 WebTrafMon의 장점을 살리면서 데이터베이스를 이용하여 장기간의 네트워크 트래픽 정보를 저장하고 로드 밸런싱을 사용하여 패킷 캡처, 패킷 분석, 웹 뷰어 등의 모듈을 분산 시스템에서 독립적으로 실행하여 시스템 과부하로 인한 패킷 손실을 막고 패킷 캡처를 독립적으로 설계하여 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 있다.

1. 서론

오늘날 네트워크 트래픽은 계속해서 증가하고 있다. 이러한 네트워크 트래픽의 증가로 인해 네트워크 회선의 부족이나 네트워크 응답 시간의 저하 등의 문제가 발생 할 수 있으며 이를 대비하기 위해 네트워크 모니터링 및 분석 시스템의 중요성이 부각되고 있다.

여러 네트워크 분석기가 개발되어 사용되고 있으나 단순한 실시간 정보로 인해 네트워크 모니터링에 적합하지 못하거나 장기간의 네트워크 계획을 세우는데 적합하지 않다. 예를 들어, tcpdump [3]는 패킷 한 개의 정보를 실시간으로 나열하고 있어 네트워크 모니터링에 적합하지 않으며 ntop [4]은 네트워크 트래픽 정보를 하루 동안 매 시간 별로 모아 분석하므로 하루동안의 모니터링에는 적합하나 하루 이상의 장기간 모니터링에는 적합하지 않다.

또, 기존의 네트워크 분석기들은 한 시스템에서 패킷 캡처와 패킷 분석을 동시에 하므로 시스템 과부하시 패킷 손실이 발생하며 한 시스템에서만 패킷 캡처가 가능하도록 설계되어 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 없다.

이에 따라 본 논문에서는 기존의 네트워크 분석기들의 장단점을 파악하여 네트워크 트래픽을 사용자에게 정확하고 편리하게 분석해 줄 수 있는 네트워크 모니터링 및 분석 시스템의 요구사항을 살펴보고 그러한 요구사항을 반영한 WebTrafMon II의 설계 및 구현을 설명한다.

WebTrafMon II의 기반이 되는 WebTrafMon [1]은 웹기반에서 하루동안 매 시간 별로 호스트 정보, 프로토콜 정보, 어플리케이션 정보 등을 분석해 주는 장점을 갖는 네트워크 분석기이다. WebTrafMon도 기존의 네트워크 분석기들처럼 장기간의 네트워크 모니터링이 지원되지 않는 단점과 한 시스템에서 패킷 캡처와 패킷 분석을 동시에 하여 시스템 과부하시 패킷 손실이 발생하고 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 없는 단점이 있다.

WebTrafMon II는 WebTrafMon의 장점을 살리면서 단점을 극복하기 위해 데이터베이스를 사용하여 장기간의 네트워크 트래픽 정보를 저장하고 분산 시스템에서의 로드 밸런싱을 사용하여 패킷 캡처와 패킷 분석을 독립시켜 시스템 과부하로 인한 패킷 손실을 막았다. 또, 패킷 캡처를 독립적으로 설계해 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 있다.

2. 관련연구

여러 네트워크 분석기들이 실제로 사용되고 있다. 관련연구에서는 각 네트워크 분석기들의 특징을 살펴보고 장단점을 파악한다.

2.1 Ntop

ntop은 Deri Luca가 1998년부터 지금까지 오픈 프로젝트로 개발하고 있는 네트워크 모니터링 및 분석 시스템이다. ntop은 'network top'이라는 뜻으로 유닉스의 시스템 자원 상태를 보여주는 top 명령어를 네트워크 모니터링 및 분석 시스템에 응용했다는 뜻이다. ntop은 웹기반에서 호스트 정보, 프로토콜 정보, 어플리케이션 정보 등을

분석해 준다. 호스트 중심으로 호스트에서 나가고 들어오는 패킷의 양 이라든가 호스트에서 많이 사용되는 프로토콜을 확인하기 편리하다. ntop은 시스템 과부하로 인한 패킷 손실을 막기 위해 패킷 분석 모듈에 쓰레드를 쓰는 등 시스템 자원을 적게 사용하도록 만들었다. 현재 ntop은 하루 24시간을 기준으로 해서 매 시간 별 네트워크 트래픽을 분석해 주는 기능이 있다. 호스트 중심의 분석 기능은 조사한 네트워크 분석기들 중 가장 뛰어났으나 장기간 네트워크 트래픽 분석을 제공하지 못하는 단점이 있다.

2.2 MRTG

MRTG(Multi-Router Traffic Grapher) [5]는 네트워크 링크 간의 트래픽 부하량을 측정하는 도구로서 5분 단위에서 1년 단위까지 네트워크 트래픽의 총 양을 확인하는 도구이다. 웹기반으로 동작하여 네트워크 트래픽을 확인하기에 편리할 뿐만 아니라 C와 Perl로 작성되어 유닉스 플랫폼 뿐 만 아니라 윈도우즈 NT에서도 동작하는 장점이 있다. 또 MRTG는 snmp MIB 정보를 사용하여 패킷 캡처를 하지 않으므로 패킷 손실 없이 정확한 정보를 제공해 준다. 그러나 MRTG는 네트워크 트래픽의 총 양에 대한 정보를 제공해 줄 뿐 어떤 호스트에서 어느 정도의 트래픽을 발생시켰는지, 어떤 어플리케이션이 어느 정도의 트래픽을 발생시켰는지, 어떤 프로토콜이 사용되었는지 등의 정보를 제공해 주지 못하는 단점이 있다. 이러한 단점으로 인해 네트워크의 문제점을 파악 할 수 있는 네트워크 분석기로는 적당하지 않다.

2.3 Ethereal

ethereal [6]은 실시간 혹은 파일에 저장해 놓은 네트워크 트래픽 정보를 분석해 주는 프로그램이다. GTK+ 기반으로 유닉스 플랫폼의 X 윈도우 화면에서 동작하며 MS 윈도우즈용도 있다. 여러 다양한 프로토콜 분석을 지원하는 장점이 있다. Gerald Comb가 50명 이상의 프로그래머들과 함께 오픈 프로젝트로 진행되고 있다. ethereal은 보안 등을 위한 실시간 네트워크 모니터링에 적합한 프로그램으로 장기간의 네트워크 모니터링에는 적합하지 않다.

2.4 NNStat

NNStat [7]은 Robert T. Braden과 Annette L. DeSchon이 1988년에 release 2.2를 발표한 프로그램으로 여러 네트워크 노드에서의 트래픽을 합쳐서 분석 할 수 있으며 사용자가 지정하는 장기간의 네트워크 트래픽 분석도 지원한다. NNStat은 SAA(Statistics Acquisition Agent)와 SCH(Statistics Collection Host)로 구성되어 있다. SAA는 여러 네트워크 노드에서 네트워크 트래픽을 모으는 모듈이고 SCH는 분석하는 모듈이다. 각 노드에 SAA가 하나씩 있게 되고 각각의 SAA에서 모은 데이타를 SCH에 보내준다. SAA에는 분석 시간을 설정하는 기능이 있어 사용자가 분석 시간을 설정 할 수 있다. NNStat은 SunOS 4.0 NIT(Network Interface Tap) 인터페이스를 사용하는 시스템에만 지원되는 단점이 있다.

2.5 UniMon

OSI 7 계층의 프로토콜을 정확히 분석하는 것을 목표로 갖는 UniMon [8]은 NNStat처럼 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 있는 장점이 있다. UniMon은 현재의 네트워크 문제점 파악에 적당한 프로그램으로 장기간의 네트워크 트래픽 분석에는 적당하지 않다.

2.6 기타

tcpdump와 tcpslice [9]는 실시간 네트워크 분석로 tcpslice는 다양한 포맷으로 네트워크 트래픽 정보를 파일에 저장 할 수 있는 특징을 가지고 있다. snoop [10]은 Etherfind의 새로운 버전으로 SunOS 5.x에 기본적인 패키지로 들어가 있다. snoop은 tcpdump와 기능이 비슷하며 역시 실시간 분석기이다. 그외 argus [11], arpwatch [12], nsfwatch [13], drawbridge [14] 등은 네트워크 보안을 목적으로 하기에 약간의 네트워크 분석 기능을 지원하나 네트워크 모니터링에는 불편하다. 또 MS 윈도우즈 전용 프로그램으로 ewatch [15]와 sniffer pro [16]같은 프로그램이 있다. ewatch와 sniffer prot는 상용 프로그램으로 실시간 분석을 위주로 편리한 UI를 제공해 주는 것이 장점이나 장기간의 네트워크 트래픽을 분석 할 수 없다.

2.7 프로그램 비교사항

위에서 소개한 내용을 정리해 보면 다음 표1과 같다.

	분석방식	분석범위	여러노드에서의패킷캡쳐	호스트 분석가능	웹기반
MRTG	일괄처리	현재상황 매시, 매일, 매주, 매달	no	no	yes
ntop	일괄처리	현재상황 매시	no	yes	yes
ethereal	실시간 일괄처리	현재상황	no	yes	no
tcpdump	실시간	현재상황	no	yes	no
snoop	실시간	현재상황	no	yes	no
NNStat	일괄처리	지정된시간동안	yes	yes	no
UniMon	실시간	현재상황	yes	yes	no
tcpslice	실시간	현재상황	no	yes	no
argus	실시간	현재상황	no	no	no
arpwatch	실시간	현재상황	no	no	no
nfswatch	실시간	현재상황	no	no	no
drawbridge	실시간	현재상황	no	no	no
ewatch	실시간	현재상황	no	yes	no
sniffer pro	실시간	현재상황	no	yes	no
WebTrafMon	일괄처리	현재상황 매시	no	yes	yes

표 1. 네트워크 분석기 비교

표1에서 ‘분석방식’을 ‘실시간’과 ‘일괄처리’로 나누었다. 실시간은 패킷 하나가 들어올 때마다 정보를 보여주는 방식을 말하며 일괄처리는 일정한 시간동안의 트래픽을 모아서 분석한뒤 보여주는 방식을 말한다. ‘분석범위’의 매시는 과거 1시간 동안을 분석해서 보여주는 것을 말한다. ‘여러 노드에서의 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.

표1에서 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 필요하여 WebTrafMon II는 분석 방식을 일괄처리, 분석 범위는 매시, 매일, 매달, 매년으로 정했다. 또 여러 노드에서의 패킷 캡처를 지원하는 프로그램이 NNStat과 UniMon밖에 없기에 여러 네트워크 노드에서의 패킷 캡처를 지원하도록 정하였다.

3. 요구사항

네트워크 트래픽이 증가하는 원인으로는 첫째 점점 더 많은 시스템이 네트워크에 연결되어 사용자가 증가하는 것과 둘째 웹(WWW)을 비롯한 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 만 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, 사용되고 있는 호스트의 수는 어느 정도인지, 호스트가 주로 사용하고 있는 서비스는 무엇인지 등의 정보를 알려 줄 수 있어야 한다. 그러나 tcpdump와 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 정보를 분석 할 수 있어야 한다. 특정한 시간 동안 어떤 호스트들이 네트워크 트래픽을 발생시켰고 그 중 가장 많은 트래픽을 발생시킨 호스트는 무엇인지 알 수 있어야 네트워크의 문제점을 파악 할 수 있다.

2) 사용되고 있는 전송 프로토콜(e.g, arp, ip, udp, tcp) 및 어플리케이션 계층의 프로토콜(e.g, ftp, snmp, telnet)을 분석 할 수 있어야 한다. 프로토콜을 분석해야 어떤 서비스가 사용되는 지 알 수 있다.

3) 웹기반의 사용자 인터페이스를 제공해야 한다. 웹기반의 사용자 인터페이스를 통해 여러 사람이 언제 어디서나 정보를 쉽게 확인 할 수 있다.

요구사항 1), 2), 3)을 만족하는 네트워크 분석기로는 ntop과 WebTrafMon이 있다.

ntop은 웹기반에서 호스트를 기준으로 해서 각 호스트들이 어떤 프로토콜을 사용하는지 분석해 주는 기능이 있고 WebTrafMon도 웹기반에서 시간 별로 호스트 정보와 프로토콜 정보를 제공하므로 둘 다 요구사항 1), 2), 3)을 만족한다.

그런데 ntop이나 WebTrafMon으로는 장기간의 네트워크 트래픽을 분석 할 수 없다. 왜냐하면 ntop이나 WebTrafMon은 하루 동안 매 시간 별 네트워크 트래픽 정보만을 제공하기에 24시간이 지나가면 분석해 놓은 데이타가 없어지기 때문이다. 장기적인 네트워크 계획을 세우기 위해서는 매 시간 별 데이타 뿐 만 아니라 하루 단위, 한달 단위, 일년 단위 등으로 네트워크 트래픽을 분석해 줄 수 있어야 한다.

장기적인 네트워크 트래픽 정보를 제공하는 네트워크 분석기로는 MRTG가 있으나 관련연구에서 소개했듯이 MRTG는 네트워크 트래픽의 총양을 보여 줄 뿐 요구사항 1), 2)같은 호스트 정보, 프로토콜 정보 등을 보여 주지 못해 네트워크 트래픽의 원인을 정확히 진단 할 수 없다.

ntop과 WebTrafMon으로는 나누어진 네트워크의 트래픽을 분석 할 수 없다. 나누어진 네트워크의 트래픽을 분석하기 위해서는 한 노드와 다른 노드의 네트워크 트래픽을 합쳐서 분석 할 수 있어야 한다. NNStat [13]이 여러 네트워크 노드의 트래픽을 분석 할 수 있는 기능을 제공한다. 그러나 NNStat은 SunOS 4.*의 NIT(Network Interface Tap)기반에서 개발되어 SunOS 4.*에서만 동작하는 단점이 있다.

ntop이나 WebTrafMon은 패킷 캡처와 패킷 분석을 동시에 하므로 시스템 과부하시 패킷 손실이 발생한다. ntop의 경우 패킷 분석 모듈이 시스템 자원을 적게 쓰도록 만들었으나 패킷이 많이 들어 올 경우 패킷 캡처를 독립적으로 하는 시스템 보다 시스템 과부하로 인한 패킷 손실의 위험이 높다.

이런 문제들을 고려해 볼 때 요구사항 1), 2), 3) 이외에 다음과 같은 요구사항이 필요하다.

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

5) 여러 네트워크 노드에서의 네트워크 트래픽을

합쳐서 분석 할 수 있어야 한다.

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

본 논문에서는 요구사항 1), 2), 3) 뿐 만 아니라 4), 5), 6)도 반영 할 수 있는 네트워크 모니터링 및 분석 시스템을 설계 및 구현하였다. 이 시스템의 가장 큰 특징은 요구사항 4), 5), 6)을 반영하기 위해 분산 시스템에서의 로드 밸런싱을 사용했다는 점이다. 패킷 캡처, 패킷 분석, 웹 뷰어 등으로 모듈을 나누고 이 세 모듈을 각각의 머신에서 동작하게 함으로서 여러 네트워크 노드에서 패킷 캡처를 할 수 있고 패킷 분석시의 시스템 과부하가 패킷 캡처에 영향을 주지 않는다.

4. 설계

앞에서 살펴본 요구조건들을 기반으로 WebTrafMon II를 설계한다. 설계 구조는 그림1과 같다.

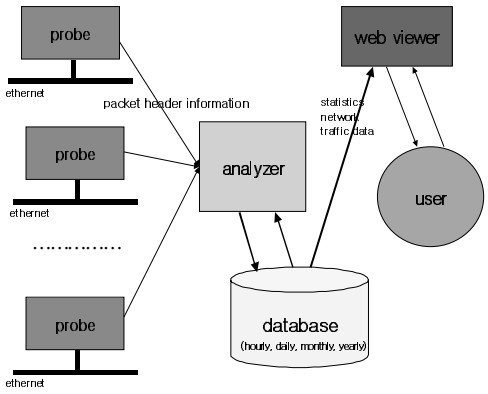

그림 1. WebTrafMon II의 구조

그림1과 같이 WebTrafMon II는 패킷을 캡처해서 파일에 저장하는 패킷 캡처 모듈(probe)과 파일에 저장된 패킷 정보를 분석해서 데이터베이스에 저장한 뒤 매시, 매일, 매달, 매년 단위로 분석하는 패킷 분석 모듈(analyzer), 그리고 분석된 정보를 웹에서 보여주는 웹 뷰어(web viewer) 모듈로 구성되어 있다.

4.1 패킷 캡처 모듈(probe)의 설계

프로브는 네트워크 상의 패킷을 캡처한 뒤 패킷 헤더로부터 원하는 정보를 추출하여 로그 파일에 저장하는 모듈이다. 그림2는 이더넷 프레임에서 정보를 추출해서 로그 파일에 저장하는 내용을 설명한다.

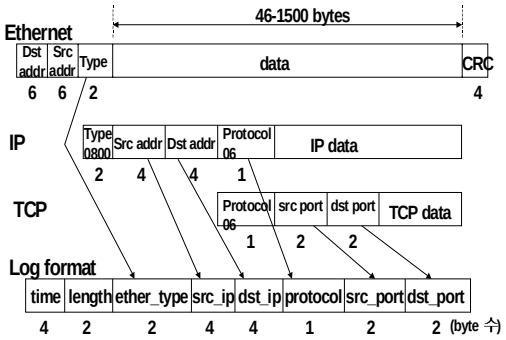

그림 2. 프로브의 로그 정보

로그 포맷의 time은 패킷을 캡처해서 로그 파일에 저장할 때의 시간이며 length는 패킷 전체의 크기에 CRC값(4 바이트)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. 로그 포맷의 일정 필드를 사용하지 않는 프로토콜은 사용하지 않는 필드를 0으로 채운다. 예를 들어, ARP같이 IP기반이 아닌 프로토콜은 time, length, ether_type 등의 값만 저장한뒤 사용하지 않는 src_ip, dst_ip, protocol, src_port, dst_port 등은 모두 0으로 저장한다. 또, ICMP같이 IP기반이지만 포트 정보를 사용하지 않는 프로토콜은 time, length, ether_type, src_ip, dst_ip 등의 값만 저장한뒤 src_port, dst_port 등을 0으로 저장한다. 포트 번호 중 0번이 있으나 분석할 때 TCP/UDP 프로토콜 여부를 먼저 체크한 뒤 포트 분석을 하므로 TCP/UDP의 포트 번호 0번과 빈값을 채워넣는 0값과 구분할 수 있다.

그림1과 같이 여러 네트워크 노드의 트래픽을 합쳐서 분석하려면 프로브를 여러곳에서 실행시킬 수 있어야 한다. 따라서 프로브를 패킷 분석 모듈과 독립적인 모듈로 설계하였다. 독립적으로 설계된 프로브와 패킷 분석 모듈이 로그 정보를 공유하기 위해 프로브는 매 시간마다 로그 파일을 새로 만들고 패킷 분석 모듈은 한 시간 전의 로그 파일을 읽어 분석한다.

4.2 패킷 분석 모듈(analyzer)의 설계

어낼라이저는 프로브가 저장한 파일 정보를 IP와 non IP 데이타로 분류해서 데이터베이스에 저장하는 로그 변환기와 분류된 데이타를 가지고 분석하고 통계를 내서 매시, 매일, 매달, 매년의 데이타를 만들어 주는 DB분석기로 나누어진다. 그림3은 어낼라이저의 구조이다.

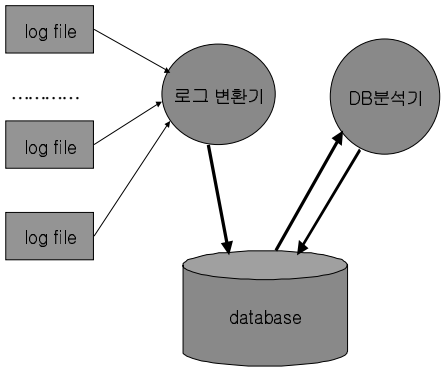

그림 3. 어낼라이저의 구조

여러 노드에서 프로브를 실행해서 매 시간마다 각각의 로그 파일이 만들어지면 로그 변환기가 각각의 로그 파일들의 정보를 합쳐서 데이터베이스에 넣는다. 그 후 DB분석기가 데이터베이스에 저장된 정보를 매 시간마다 분석하여 호스트 정보, 프로토콜 정보, 어플리케이션 정보를 만들어서 데이터베이스에 저장한다. 그리고 이렇게 매 시간마다 저장된 데이타를 가지고 하루마다, 한달마다, 1년마다 통계를 내서 데이터베이스에 저장한다. 그림4는 로그 변환기와 DB분석기에 의한 네트워크 트래픽 데이터베이스 테이블의 생성과 변화를 나타낸다.

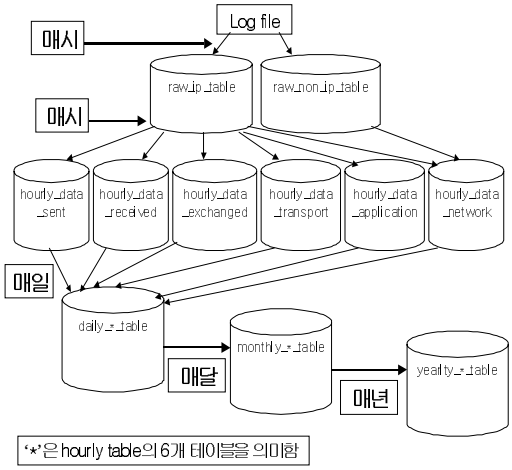

그림 4. 어낼라이저의 데이타 이동

로그 변환기가 로그 파일을 분석해서 raw_ip_table과 raw_non_ip_table을 만든다. raw 테이블 두 개를 이용해서 DB분석기가 호스트, 프로토콜, 어플리케이션 등에 관련된 데이타를 분석하여 hourly 테이블을 만들어 낸다. 그 다음 부터는 hourly 테이블을 가지고 차례로 통계를 내서 daily, monthly, yearly 테이블을 만든다. 호스트 정보는 data_sent_table, data_received_table, data_exchanged_table 등 3개의 테이블로 나누었고 프로토콜 정보는 network_table과 transport_table로 나누었다. 그림5에 raw 테이블과 네트워크 테이블을 예로 해서 hourly, daily, monthly, yearly 테이블의 데이터베이스 스키마를 나타냈다.

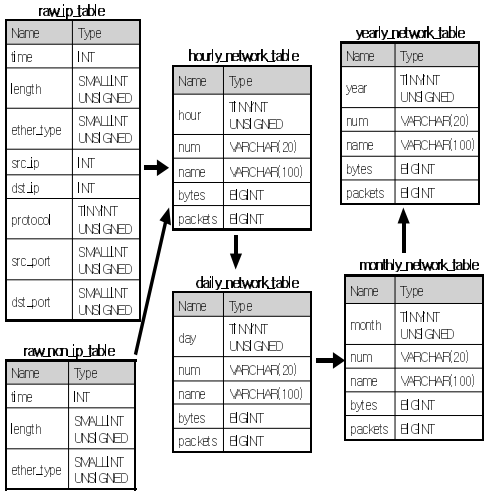

그림 5. 데이터베이스 스키마 일부

4.3 웹 뷰어(web viewer) 모듈의 설계

웹 뷰어는 어낼라이저가 분석 및 통계를 내서 저장한 매시, 매일, 매달, 매년의 네트워크 트래픽 데이타를 웹에서 보여주는 모듈이다. 데이터베이스의 스키마 내용을 적절하게 보여 줄 수 있도록 웹 뷰어에서도 매시, 매일, 매달, 매년으로 나누어 정보를 확인 할 수 있도록 설계했다. 그림6은 웹뷰어의 설계를 설명한다.

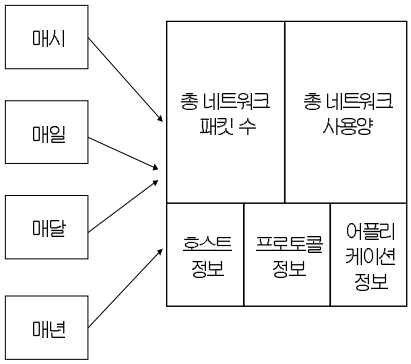

그림 6. 웹 뷰어의 설계

그림6과 같이 매시, 매일, 매달, 매년의 정보에서 먼저 각각의 네트워크 트래픽의 총 패킷 수와 총 패킷 사용양을 보여주고 각각에서 호스트 정보, 프로토콜 정보, 어플리케이션 정보를 보여줄 수 있도록 설계했다.

5. 구현

WebTrafMon II의 구현은 분산 시스템 환경에서 로드 밸런싱을 사용했다. 로드 밸런싱을 사용한 이유 중 하나가 시스템 과부하로 인한 패킷 손실을 막기 위함이다. 하나의 시스템에서 패킷을 캡처하면서 동시에 분석 할 경우 시스템 과부하로 인한 패킷 손실의 위험이 높아진다. 따라서 분산 시스템에서 패킷 캡처하는 모듈과 패킷 분석하는 모듈을 각각 하나의 시스템에서 동작하게 하여 이러한 위험을 없앴다. 분산 시스템을 사용한 다른 이유는 여러 네트워크 노드의 트래픽을 합쳐서 분석 할 수 있는 환경을 만들기 위해서이다. 그림7은 WebTrafMon II의 로드 밸런싱을 설명해 준다.

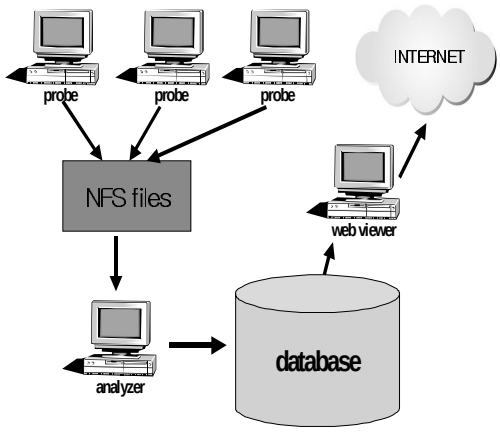

그림 7. WebTrafMon II의 시스템 구조

그림7과 같이 여러개의 프로브가 네트워크 노드의 트래픽 정보를 NFS 환경에서 각각의 로그 파일에 저장한다. 이렇게 저장된 각각의 로그 파일들을 어낼라이저가 분석하여 데이터베이스에 저장하고 웹 뷰어는 분석된 데이타를 사용자에게 보여준다. 로그 파일과 데이터베이스도 프로브, 어낼라이저, 웹 뷰어 등과 독립되야 하므로 NFS 파일 서버를 사용하여 하나의 시스템에서 동작시켰다.

5.1 패킷 캡처 모듈(probe)의 구현

프로브를 구현하기 위해서 libpcap 라이브러리의 promiscuous 모드를 사용했다. libpcap 라이브러리는 유닉스 계열의 다양한 운영체제에서 쓸 수 있도록 운영체제에 독립적인 API를 제공하는 장점이 있어 대부분의 네트워크 분석기에 사용된다. libpcap이 운영체제에 독립적이기에 WebTrafMon II도 다양한 운영체제에서 사용이 가능하다.

프로브에서 time을 저장함으로서 매시, 매일, 매달, 매년의 데이타 뿐 만 아니라 특정 시간대의 사용자가 요구하는 시간 간격의 정보도 보여줄 수 있다.

캡처한 패킷 정보를 데이터베이스에 저장하는 것보다 파일에 저장하는 것이 더 빠르고 데이터베이스에 직접 저장할 경우 프로브가 동작하는 시스템에서 과부하로 인한 패킷 드랍이 발생할 수 있기에 로그 파일을 거쳐 데이터베이스에 정보를 저장하도록 구현했다. 로그 파일을 사용할 경우에는 어낼라이저가 동작하는 시스템에서 로그 정보를 데이터베이스로 저장하므로 프로브가 동작하는 시스템에 부하를 주지 않는다. 로그 파일을 텍스트 파일이나 바이너리 파일로 만들 수 있는데 텍스트 파일은 눈으로 로그 정보를 확인할 수 있는 장점은 있으나 저장하는데 바이너리 파일보다 시간이 많이 걸리고 저장한 내용을 데이터베이스로 옮기는 작업이 쉽지 않아 텍스트 파일보다 더 빠르게 저장되고 데이터베이스에 쉽게 저장할 수 있는 바이너리 파일을 사용하였다.

5.2 패킷 분석 모듈(analyzer)의 구현

어낼라이저는 C 언어와 mysql 데이터베이스를 사용해서 구현했다. 어낼라이저는 매시가 되면 프로브에서 저장한 바이너리 파일을 읽어 IP기반 패킷과 non IP기반 패킷으로 나누어 데이터베이스에 저장한다.

어낼라이저는 때가 되면 매시, 매일, 매달, 매년의 네트워크 트래픽 정보를 자동으로 분석해서 데이터베이스에 저장해야 하므로 unix의 cron 명령어를 이용해서 매시, 매일, 매달, 매년마다 어낼라이저가 자동으로 동작하도록 했다.

또 웹기반에서 사용자에게 TCP/UDP 포트 번호와 포트 이름을 입력받을 수 있도록 했다. 이렇게 구현한 이유는 다양한 어플리케이션이 개발되는 것을 고려해서 어플리케이션 계층 분석의 확장성과 정확성을 높이기 위함이다. 사용자가 특정한 포트 번호와 포트 이름을 입력해 주면 분석 할 때 이것이 반영되어 잘 알려진 포트 번호외에 최근 개발된 어플리케이션까지 분석할 수 있다.

5.3 웹 뷰어(web viewer) 구현

웹 뷰어는 아파치 서버를 사용해서 웹서버를 동작시켰으며 네트워크 트래픽 정보를 사용자에게 편리하게 제공하기 위해 C 언어와 gd 라이브러리를 사용하여 네트워크 트래픽의 총 양을 보여주는 그래프를 그렸다. 또 데이터베이스의 내용을 아파치 웹서버로 전달하기 위해 C-CGI를 사용하여 웹과 데이터베이스를 연동하였다.

6. 구현 화면

시간 별 데이타를 가지고 예를 든다. 다음 그림8은 매 시간 별 데이타를 보여준다.

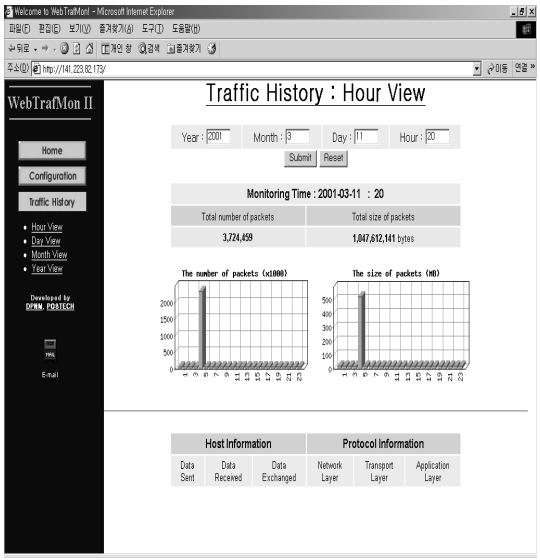

그림 8. Hour View의 첫 화면

그림8의 두 그래프는 한시간 동안의 총 패킷 수와 총 네트워크 트래픽 양을 보여준다. 제일 위의 시간창에서 2001년 3월 11일 20시의 데이타를 선택했다는 것을 알 수 있고 3월 11일 하루의 매 시간 별 데이타를 보여준다. 그림8에서 Data Sent를 선택하면 그림9와 같이 3월 11일 20시에 데이타를 가장 많이 보낸 호스트 10개를 보여준다. 그림9에서 141.223.82.171라는 호스트가 2001년 3월 11일 20시에 가장 많은 네트워크 트래픽을 보냈음을 알 수 있다.

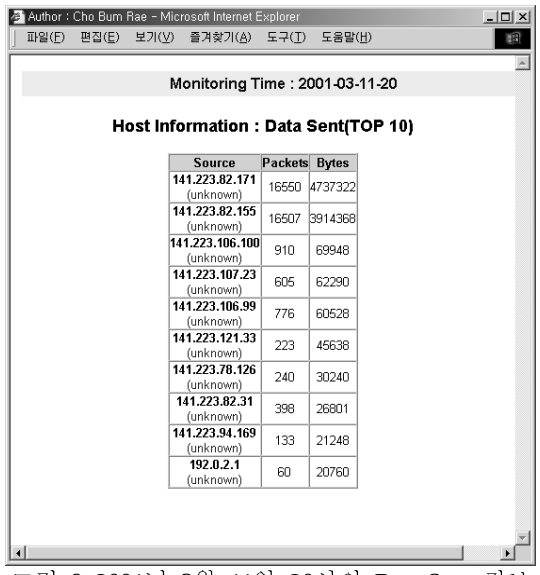

그림 9. 2001년 3월 11일 20시의 Data Sent 정보

7. 결론 및 향후 과제

현재 WebTrafMon II는 네트워크 트래픽을 장기간 분석하여 때에 따라 보여 줄 수 있도록 분산 시스템에서 구현되어 있다. WebTrafMon II는 다른 네트워크 분석기들과 달리 시간 단위에서 매일, 매달, 매년 단위의 장기간 네트워크 트래픽 데이타를 분석해 줄 수 있으며 로드 밸런싱을 사용하여 하나의 시스템에서 독립적인 패킷 캡처를 하므로 시스템 과부하시의 패킷 손실을 막을 수 있고 여러 네트워크 노드에서의 트래픽을 합쳐서 분석 할 수 있다는 장점이 있다. 또 사용자에게 어플리케이션 계층의 정보를 입력받아 어플리케이션 계층 분석의 확장성을 갖는 것도 장점이다.

앞으로의 연구는 실시간 네트워크 트래픽을 분석하는 기능과 사용자가 원하는 시간 범위 내에서의 데이타를 분석해 주는 기능을 추가하는 것이다. 또 데이터베이스를 이용한 호스트와 어플리케이션의 연관관계를 분석하거나 호스트와 프로토콜, 어플리케이션의 종합적인 분석도 필요하다.

마지막으로 사용자에게 정보를 좀 더 빨리 보여 줄 수 있도록 어낼라이저와 웹 뷰어 등의 성능을 개선하고 사용자가 편리하게 사용 할 수 있는 화면 설계 및 개선이 요구된다.

[참고문헌]

[1] 권순선, 김재영, 홍원기, “웹 기반의 인터넷/인트라넷 네트워크 트래픽 모니터링 및 분석 시스템”, KNOM Review 제 2권 제 1호, 1999년 4월, pp.1-10.

[2] R. Enger and J. Reynolds, “FYL on a Network Management Tool Catalog”, IETF RFC 1470, June 1993.

[3] Lawrence Berkley National Laboratory, “tcpdump 3.6”, http://www.tcpdump.org.

[4] L. Deri and R. Carbone, “Monitoring Networks Using Ntop”, Released paper in http://luca.ntop.org, Jan 29th 2001.

[5] Tobias Oetiker and Dave Rand, “MRTG: Multi Router Traffic Grapher”, http://www.mrtg.org.

[6] Ethereal Homepage, http://www.ethereal.com.

[7] Robert T. Braden and Annette L. DeSchon, “NNStat: Internet Statistics Collection Package”, USC/Information Sciences Institute Marina del Rey, California, November 28, 1988.

[8] Werner Erhard, Michael M. Gutzmann and Hastings M. Libati, “Network Traffic Analysis and Security Monitoring with UniMon”, Proceedings of the IEEE Conference on, 2000, pp.439-446.

[9] Lawrence Berkeley National Laboratory, “tcpslice-1.1a3”, ftp://ftp.ee.lbl.gov/tcpslice.tar.Z.

[10] sun’s snoop web page , “http://www.sun.com/products/sunray1/ts-sysmon.html”.

[11] Carter Bullard, “argus-1.7.beta.1b”, ftp://ftp.sei.cmu.edu/pub/argus.

[12] Lawrence Berkley National Laboratory, “arpwatch 2.0”, ftp://ftp.ee.lbl.gov/arpwatch.tar.Z.

[13] Dave Curry and Jeff Mogul, “nfswatch-4.3”, ftp://ftp.lip6.fr/pub2/networking/nfs.

[14] David K. Hess and Douglas Lee Schales, David R. Safford, “drawbridge 2.0”, http://www.certcc.or.kr/tools/index.html.
[15] ewatch homepage, “http://ewatch.hangkong.ac.kr”.
[16] sniffer pro homepage, “http://www.softseek.com”.