웹 기반의 인터넷/인트라넷 네트워크 트래픽 모니터링 및 분석 시스템
(Web-based Internet/Intranet Network Traffic Monitoring and Analysis System)

권순선, 김재영, 홍원기
포항공과대학교 컴퓨터공학과
{jay, jwkhong}@postech.ac.kr

요 약

인터넷과 월드와이드웹(WWW)의 보급과 확장과 함께 네트워크 기반의 응용 프로그램의 사용량이 늘어남에 따라 인터넷/인트라넷 네트워크 트래픽은 증가일로에 있으며 어느 호스트나 어느 응용 프로그램이 가장 많은 트래픽을 유발하는지를 알아내는 일은 한정된 네트워크 자원을 효율적으로 활용하는데 매우 중요하고도 핵심적인 일이 되었다. 본 논문에서는 월드와이드웹을 이용하여 사용하기 쉽고, 어떤 곳에서나 쉽게 적용할 수 있도록 이동성을 극대화한 WebTrafMon 이라는 이름의 네트워크 트래픽 분석 및 관찰 도구의 설계 및 구현 방법에 대해 고찰해 본다. 웹 기반 기술은 언제 어디서나, 쉽게 사용할 수 있는 웹 브라우저를 통해 사용자들이 복잡한 사용자 인터페이스를 통하지 않고도 쉽게 원하는 결과를 얻어갈 수 있게 한다. WebTrafMon 은 이러한 웹 기반 기술의 장점을 살려 트래픽 양의 측정뿐만 아니라 해당 트래픽의 유형과 트래픽의 진원지 및 목적지까지 함께 나타내어 준다.

1. 서론

오늘날, 점점 더 많은 시스템이 네트워크에 직접 연결되고 다양한 응용 프로그램들이 네트워크와 관련되어 개발되는 경우가 많아지면서 네트워크 트래픽은 날로 증가일로에 있다. 이것은 네트워크 회선의 부족이나, 응답 시간의 저하 등과 같은 많은 문제를 불러일으키는 수준까지 이르렀으며, 이를 해결하기 위한 다양한 네트워크 모니터링 및 분석 시스템의 개발되는 계기가 되었다.

왜 트래픽을 분석해야 하는가? 그것은 더욱 많은 시스템들이 인터넷에 접속되면서 네트워크 그 자체도 디스크나 메모리처럼 관리가 필요한 귀중한 자원 중의 하나로 취급되어야 하기 때문이다. 물론 그에 따라 다양한 네트워크 자원을 관리하고 네트워크 사용을 감시하는 것은 관리자들에게 무거운 짐이 되었다.

네트워크 트래픽 모니터링은 데이터를 수집하고 분석하는 기능을 제공하는 것이다. 네트워크 상태 정보는 네트워크 상에 전송되는 패킷을 검사하여 획득되고 분석 작업은 이러한 정보를 기반으로 좀 더 확장된 정보를 제공하게 된다. 한정된 네트워크 자원을 효율적으로 활용하기 위해서는 네트워크로부터 정확하고 믿음직한 정보를 얻어야 한다. 예를 들면 다음과 같다.

- 얼마나 많은 트래픽이 유발되었는가?
- 어떤 형태의 트래픽이 유발되었는가?
- 어떤 곳에서 얼마만큼의 트래픽이 유발되었는가?
- 어떤 시스템이나 응용 프로그램이 병목현상을 일으키고 있으며 최대 트래픽은 얼마이고 언제 그러한 현상이 일어나는가?

만약 네트워크 관리자가 이러한 종류의 질문에 믿음직한 답변을 해내지 못한다면 귀중한 네트워크 자원들이 쓸모없이 낭비될 것이다. 따라서, 이러한 맥락 하에서 네트워크 관리자들이 좀 더 네트워크를 쉽게 관리할 수 있게 하기 위한 여러 가지 툴들이 개발되어 왔다. 뒤에서 살펴볼 MRTG [6], Etherfind [20], Argus [17], NFSwatch [18], TCPdump [19] 등이 그러한 예이다.

그러나, 예를 들어 MRTG 의 경우, 하루, 일주일, 한달, 일년 등 다양한 주기로 트래픽 정보를 사용자에게 제공해 주고 있다는 장점이 있으나 어느 호스트가 트래픽을 유발했으며 어느 프로토콜이 병목 현상을 일으켰는가 하는 등의 관리자가 문제 해결에 가장 필요로 하는 정보들을 제공해주지 못한다는 결정적인 단점을 지니고 있다.

이 논문에서는 새로운 웹 기반의 네트워크 트래픽 모니터링 분석 시스템인 WebTrafMon 이라는 툴의 설계 및 구현에 관해 설명한다. 이 시스템은 네트워크 트래픽을 자세히 보여주며 사용자들이 각각의 네트워크 계층별로 트래픽 정보를 볼 수도

있고, 발신지 정보 및 목적지 정보도 함께 볼 수 있다. WebTrafMon 은 웹 기반의 사용자 인터페이스를 채택하여 사용자들이 트래픽 정보를 일반적이고 쉬우며, 간편하게 구할 수 있는 웹 브라우저를 통해 접근하도록 하였다. 웹 인터페이스는 시스템이나 위치에 독립적으로 다양한 정보를 가장 일반적으로 보여줄 수 있는 방법이다. 사용자들은 오직 인터넷에 연결되어 있고, 웹 브라우저만 있으면 된다. 어떤 특별한 훈련 과정도 필요 없으며 클라이언트 쪽에서 특별히 새로운 프로그램을 설치하거나 설정할 필요도 없다. 누구든지 데이터 및 분석 결과에 접근할 수 있으며 일반적인 웹 기반의 사용자 인증 시스템을 통해 그러한 접근을 통제할 수도 있다 [15, 16].

WebTrafMon 은 네트워크의 가장 하위 계층에서부터 가장 상위 계층까지 단계적으로 패킷으로부터 헤더 정보를 추출하여 낸다. 이러한 추출 과정에서 네트워크에 다른 부담을 주는 과정은 전혀 없으며 트래픽 정보는 실시간으로 추출된다.

WebTrafMon 은 트래픽 정보를 근원지와 목적지를 기준으로 웹 인터페이스를 통해 보여줄 수 있다는 점에서 기존의 툴들과는 차이점이 있다. WebTrafMon 은 트래픽 정보를 네트워크 계층으로부터 최상위 어플리케이션 계층까지 단계적으로 나타낼 수 있는 기능도 포함하고 있다. 엔터프라이즈 네트워크의 관리에는 어떤 호스트가 네트워크 대역폭을 특정 시간에 많이 소비하는지 알아내는 것도 매우 중요한 일이다. WebTrafMon 은 그러한 용도에도 잘 부합되는 것으로서 트래픽 정보를 근원지와 목적지로 구분하여 나타낼 수 있는 기능이 네트워크 대역폭 관리에 특히 유용하게 사용될 수 있다.

기존의 툴들은 어느 호스트에서 트래픽이 유발되는지 알 수 있는 정보를 제공해주는 기능을 가지고 있지 않았다. 그러나 트래픽의 진원지를 밝혀내는 것은 네트워크의 사용량이 늘어나 병목현상이 빈번한 근래의 환경에 있어 매우 중요한 정보라고 할 수 있다. 예를 들어보자. 대역폭을 많이 차지하는 AOD(Audio On Demand), VOD(Video On Demand) 서버 몇 대만으로도 해당 서브넷 전체가 포화되는 현상이 생길 수 있는 것이다. 많은 데이터 전송을 필요로 하는 멀티미디어 정보가 점점 더 일반화되면서 트래픽 관리는 더욱 중요한 이슈로 대두되는 것은 현재의 추세로 보아서 당연한 일이라고 할 수 있다. 따라서 각각의 프로토콜에 관한 자세한 정보는 물론 각각의 호스트에 관한 정보는 네트워크 관리에 매우 중요한 요소라 할 수 있다. WebTrafMon 은 이 두 가지 형태의 자료를 모두 사용자에게 보여줄 수 있도록 설계되었다.

WebTrafMon 은 프로브(probe)와 뷰어(viewer)로 나눌 수 있는데 프로브는 네트워크 패킷으로부터 정보를 얻어내어 로그 파일에다 저장하고 뷰어는 사용자와 상호대화식으로 작동하면서 웹 브라우저를 통해 사용자에게 정보를 제공한다.

2. 관련 연구

본 장에서는 기존에 개발되어 사용되는 다양한 네트워크 모니터링 도구의 특성을 알아보고 각 툴이 제공하는 기능을 비교 분석한다.

2.1 MRTG

MRTG(Multi-Router Traffic Grapher) [6]는 네트워크 링크 간의 트래픽 부하량을 측정하는 도구로서 HTML 을 이용하여 네트워크 트래픽을 쉽게 파악할 수 있는 GIF 이미지를 생성한다. MRTG 는 PERL 과 C 언어로 이루어져 있으며 여러 다양한 유닉스 플랫폼과 윈도우즈 NT 에서 작동하며 널리 사용되고 있다.

MRTG 는 SNMP [9]를 이용하여 트래픽 정보를 읽어들이는데 이렇게 읽어들인 트래픽 정보를 좀 더 빠른 처리를 위해 C 언어로 작성된 프로그램이 처리하여 해당 네트워크의 트래픽 정보를 분석한다. 출력은 그래프 형태로 나타나게 되는데 웹 페이지에 포함되므로 단순히 웹 브라우저만 있으면 사용자는 언제 어디서나 그 정보를 읽을 수 있다는 장점이 있다.

MRTG 는 또한 장기간의 네트웍 트래픽 정보를 알아볼 수 있게 하는 기능도 제공한다. MRTG 는 트래픽 정보에 대한 로그를 가지고 있기 때문에 하루치의 트래픽 정보는 물론 일주일, 한달, 심지어는 1 년 동안의 트래픽 정보를 한꺼번에 표시할 수 있다는 장점이 있다. 이렇게 장시간의 트래픽 정보를 모니터링할 수 있음에도 불구하고 로그 정보를 담고 있는 파일들의 총 크기는 일정 한도를 넘어서지 않도록 특수한 알고리즘으로 디자인되었기 때문에 하나의 시스템으로부터 여러 군데의 네트워크 세그먼트를 동시에 모니터링할 수 있다는 장점이 있다.

그렇지만 MRTG 는 어느 호스트나 어느 어플리케이션이 얼마만큼의 트래픽을 유발했는지에 관한 정보를 제공해주지 못한다는 단점이 있다. SNMP MIB 변수들은 그같은 정보를 지원하지 않기 때문이다. 또한 트래픽 발신지, 목적지 정보는 물론 각각의 프로토콜과 관련한 정보들 역시 제공하지 않는다는 단점이 있다.

2.2 패킷 캡처링 도구

현재까지 여러 다양한 패킷 캡처링 도구들이 개발되어 왔다. 그중 대표적으로 많이 쓰이는 도구들을 여기서 소개한다.

2.2.1 Etherfind

SunOS 운영체제에서 제공하는 Etherfind [20]는

소프트웨어 패킷 모니터링 툴이다. 이것은 네트워크 인터페이스를 통해 해당 호스트로 오는 패킷은 물론 그렇지 않은 패킷까지 모두 읽어들여 필요한 정보를 얻어서 파일로 저장하는 역할을 한다. 여기서 제공되는 정보에는 프로토콜 타입, 크기, 그리고 패킷의 발신지와 수신지 등이 있다.

그렇지만 이것은 텍스트 기반의 인터페이스로 작동하기 때문에 불편한 점이 있고 시스템에 관해 특정한 권한이 있는 사용자만이 이 툴을 이용할 수 있다는 단점이 있으며 분석 기능을 제공해주지 않고 있어 실제로 트래픽 정보를 알아내기 위해서는 사용자가 다시 한번 데이터를 기반으로 정보를 조합해 내는 과정이 필요하기 때문에 여러모로 불편하다. 그림 1 은 이 Etherfind 를 실제로 사용하는 예이다.

```
# etherfind -p -i le0 -src nyssa -o -dst nyssa
    icmp type
    lnth proto source destination src port dst port
    60 tcp leela.acs.ohio nyssa login 1021
    118 udp tardis nyssa 652 684
    60 tcp leela.acs.ohio nyssa login 1021
```

그림 1. Etherfind 사용 예

2.2.2 NFSwatch

NFSwatch [18]는 NFS 파일 서버의 기능을 모니터링하기 위해 만든 것으로서 현재 들어오는 모든 네트워크 트래픽을 여러 가지로 분류해서 보여준다. 기본적으로는 NFSwatch 는 자기 자신을 향하는 트래픽 외에도 특정 호스트를 지정하면 해당 호스트로 향하는 트래픽 정보까지 보여줄 수 있는 기능을 가지고 있다는 것이 특징이다. 또한 특정 호스트와 호스트 사이에 교환되는 패킷 정보도 보여줄 수 있다.

그러나, 이렇게 자기 자신이 목적지가 아닌 패킷을 모니터링하기 위해서는 데이터링크 계층에서의 특별한 처리가 필요한데 NFSwatch 는 어느 특정 운영체제에 종속적인 방법을 취하고 있어 운영체제 독립성의 측면에서 문제가 있다. 또한 Etherfind 와 마찬가지로 텍스트 기반이고 분석 기능을 제공하지 않기 때문에 불편한 점이 있다.

2.2.3 TCPdump

TCPdump [19]는 인터넷 게이트웨이 성능을 높이기 위한 측정 도구로 처음 개발되었는데 최근에 개발된 버전은 libpcap 이라는 운영체제 독립적인 패킷 캡처링 라이브러리를 이용하여 다양한 운영체제에서 널리 사용되고 있다. TCPdump 는 네트워크 상에서 전송되는 패킷의 헤더 정보를 분석하여 보여주는 역할을 하는데 사용자는 그 정보를 수동으로 분석하여 네트워크의 상태를 파악해야 한다. 따라서 장시간의 네트워크 상태를 파악하는 데는 적당하지 않고, TCPdump 가 여러 가지 다양한 옵션들을 가지고 패킷 정보를 자세하게 보여주는 장점을 가지고 있긴 하지만 텍스트 기반에, 분석 기능을 제공하지 않는다는 것, 그리고 출력 결과가 매우 복잡하다는 것이 커다란 단점이다. 그림 2 는 TCPdump 의 실제 출력 결과를 예로 든 것이다.

```
22:27:52.875202 ohhara.postech.ac.kr.6255 >
        crystal.hiper.co.kr.7871: S
        2171642993:2171642993(0) win 512 <mss
        1460>

22:27:52.915202 210.111.183.206.dtspcd >
        belle.postech.ac.kr.dtspcd: udp 30

22:27:52.925202 xgs21.postech.ac.kr.x11 >
        ohhara.postech.ac.kr.12382: P

        1792:1824(32) ack 1289 win 4096
```

그림 2. TCPdump 의 사용 예

2.3 Argus

Argus [17]는 IP 네트워크의 감사 툴이라고 할 수 있다. 이 툴은 지정한 네트워크 인터페이스 디바이스를 읽어들여 해당 인터페이스로 들어오는 모든 패킷을 분석하고 패킷의 정보를 저장한다. 이렇게 저장된 정보는 다양한 형태로 가공되어 사용자에게 보여지는데 관리자의 입장에서 일목요연하게 정보를 제공하기 때문에 주로 네트워크 보안 및 설정이 잘못되었는지, 해킹 시도가 있었는지 등을 감지하는 용도로 많이 쓰였다.

그렇지만 이것 역시 개별 프로토콜에 대한 자세한 정보를 보여주지 못하고, 웹 기반이 아닌 텍스트 기반이라는 점 때문에 사용자 요구 조건을 충족시켜주지 못했다. 다음 표 1 에서 요구 조건과 각 툴들이 지원하는 기능들에 대해서 살펴보도록 한다.

표 1. 기존에 개발된 툴의 특성

	가	나	다	라	마
Web-based	Yes	No	No	No	No
Analysis	Yes	No	No	No	Yes
Host Information	No	No	No	No	Yes
Protocol Information	No	No	No	No	No

가: MRTG, 나: Etherfind, 다: NFSwatch,
라: TCPdump, 마: Argus

표 1 에서와 같이 지금까지 여러 가지 기존에 개발된 툴들을 살펴보았으나 어느 것도 요구조건을 완전히 만족시켜주는 것은 없었다. 따라서 WebTrafMon 이라는 이름의 새로운 웹 기반 네트워크 트래픽 모니터링 시스템을 개발하게 된 것이다.

3. 시스템 요구조건

다음은 엔터프라이즈 네트워크 모니터링 및 분석 시스템이 갖추어야 할 가장 중요한 요구조건들을 간추려본 것이다.

3.1 운영체제 독립성

저수준의 패킷 캡처링 루틴은 운영체제에 독립적이어야 한다. 각각의 플랫폼에서 제각각 다른 방법의 네트워크 디바이스 인터페이스를 제공하기 때문에 상위 계층에서 보았을 때 추상화가 되어 있어야 한다.

3.2 웹 기반의 사용자 인터페이스

사용자 인터페이스는 사용자들이 시스템을 좀더 쉽고 편리하게 이용할 수 있도록 설계되어야 한다. 이러한 이유 때문에 웹 기반의 인터페이스는 최적의 솔루션이 될 수 있는 것이다. 웹 기반의 인터페이스는 또한 한가지 더 중요한 강점을 가지고 있다. 웹 기반의 시스템을 사용하기 위해서 사용자들은 네트워크에 연결하여 웹 브라우저를 사용할 수 있기만 하면 된다는 것이다. 언제나 어디서나 누구라도 일반적인 웹 브라우저를 통해 시스템을 사용할 수 있다.

3.3 패킷 캡처링 과정의 신뢰성

고속의 네트워크에서는 전송되는 패킷의 개수가 매우 많다. 그러한 데이터를 분석하는 것은 굉장히 많은 처리시간을 요할 수 있는데 그래서 패킷 캡처링 과정은 최대한 효율적으로 작성되어야 하는 것이다. 고속 네트워크에서는 패킷을 제때 처리하지 못하는 일이 발생할 수도 있다. 시스템 자체의 처리 시간이 모든 패킷을 캡처링할 수 있을 정도로 빠르지 못하다면 분석 결과를 완전히 신뢰할 수는 없을 것이다. 그렇지만 모든 패킷을 제대로 캡처한다는 것은 사실 매우 어려운 일이다 [8].

3.4 프로토콜 정보 분류

네트워크에서는 여러 다양한 계층이 존재하는데 예를 들어 HTTP [4, 5], FTP, Telnet, SNMP [9] 등등 매우 다양한 어플리케이션 계층의 프로토콜들이 계속해서 생겨나고 있다. 모든 패킷들은 정해진 프로토콜에 따라 전송되기 때문에 그러한 프로토콜을 계층적으로 정리할 수 있다. 따라서 네트워크 모니터링 툴은 모든 가능한 정보를 계층적으로 분류하여 사용자에게 보여줄 수 있도록 하는 것을 이상적인 목표로 삼을 필요가 있다.

3.5 이동성

패킷 캡처링 툴은 쉽게 다른 네트워크 세그먼트에 설치하여 적용할 수 있어야 한다. 만약 어떤 사람이 특정한 네트워크 세그먼트를 모니터링하고자 한다면 그는 모니터링 시스템을 노트북이나 데스크탑 컴퓨터에 설치하여 두고 그 세그먼트에 접속만 하면 금방 네트워크 상태를 모니터링할 수 있을 정도로 쉽게 적용할 수 있어야 한다.

3.6 보안

보안 문제는 매우 중요하게 다루어져야 한다. 귀중한 데이터의 유실을 막고 해커들의 침입으로부터 데이터로의 접근을 차단할 수 있어야 한다. 가끔 보안 설정을 소홀히 하여 해커들로부터 귀중한 데이터를 도용당하는 경우가 많이 있는데 그런 일이 없도록 하기 위해서 인증된 사용자만이 시스템에 접근할 수 있도록 할 필요가 있다.

3.7 실시간 정보 및 트래픽 변동패턴 정보 처리

네트워크 모니터링 시스템은 실시간 온라인 정보 및 축적된 정보를 통하여 트래픽 변동패턴 정보도 동시에 보여줄 수 있어야 한다. 트래픽 변동패턴 정보를 분석하면 장기간의 네트워크 상태를 조망할 수 있으며 실시간 정보를 통해서 단기간의 네트워크 문제를 해결할 수 있게 된다. 이러한 두 가지 방식을 제공함으로서 네트워크 문제의 해결을 좀더 신속하게 해줄 수 있게 된다.

4. WebTrafMon 설계

앞에서 살펴본 요구조건들을 기반으로 웹 기반의 엔터프라이즈 네트워크 모니터링 및 분석 시스템을 설계해 본다. 설계 구조는 그림 3 과 같이 나타낼 수 있다.

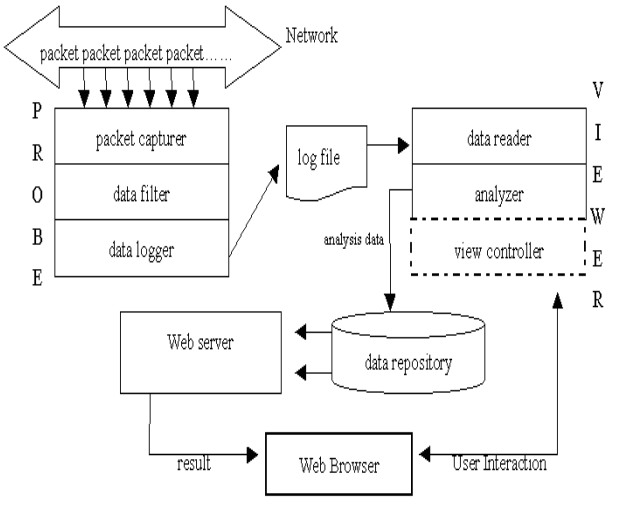

그림 3. 웹 기반의 네트워크 트래픽 모니터링 시스템의 설계

전반적인 시스템 구성은 크게 프로브와 뷰어 두 가지로 나눌 수 있다. 프로브는 네트워크 패킷으로부터 각각의 네트워크 계층에 대한 정보를 추출하고 미리 정의된 로그 파일에 그 정보를 저장한다. 로그 파일은 뷰어의 분석 루틴을 통해 읽혀지

며 사용자는 웹 브라우저를 통해 저장된 정보에 접근하게 된다.

4.1 프로브 설계

프로브는 패킷으로부터 데이터를 추출해서 로그 파일에 저장한다. 네트워크 상의 패킷 정보를 모두 추출하기 위해 프로브를 promiscuous 모드로 동작한다. 이것은 이더넷 자체의 특성상 이더넷이 패킷 브로드캐스팅 방법으로 동작하기 때문에 가능한 일이다 [2, 3]. 모든 패킷을 캡처하는 것은 프로브의 가장 기본적이면서도 중요한 역할이다. 따라서 이 부분은 운영체체에 독립적인 방법으로 작성되어야 할 필요가 있으며 자세한 구조는 뒤에 살펴보고 우선 각각의 계층별로 추출되는 정보에 대해서 좀더 자세히 고찰한다.

4.1.1 MAC 계층

Medium Access Control(MAC) 계층에서 보았을 때 패킷은 최대한의 속도로 캡처되어야 한다. 그렇지만 이 시스템은 하드웨어적으로 동작하는 것이 아니라 소프트웨어적으로 동작하는 것이기 때문에 운영체제 커널의 영향을 받게 되므로 모든 패킷을 안정적으로 캡처하는 것은 실제로는 불가능하다. 따라서 패킷 캡처 루틴은 최대한 효율적으로 설계할 필요가 있으며 하드웨어 드라이버로부터 상위 계층으로의 추상화 또한 필요하다.

이 계층에서 추출되는 정보는 네트워크 트래픽 모니터링에서 가장 중요하게 사용되는 패킷의 크기 정보이다.

4.1.2 네트워크 계층

이 계층은 IP(Internet Protocol) 및 ARP(Address Resolution Protocol), RARP(Reverse ARP) 등과 관련이 있다. 만약 어떤 패킷이 IP 기반의 패킷이라면 IP 프로토콜은 발신지 정보와 목적지 정보를 포함하고 있기 때문에 여기서부터 패킷의 발신지 및 목적지 정보를 추출할 수 있고 IP 기반의 패킷이 아니라면 패킷의 크기만을 알아본 뒤 더 이상 분석하지 않아도 된다 [7].

4.1.3 전송 계층

이 계층에서는 TCP(Transmission Control Protocol) 및 UDP(User Datagram Protocol) 정보가 추출된다. TCP 는 신뢰성 있는 전송을 보장하는 반면에 UDP 는 네트워크에 부담을 많이 주지 않고 그 대신 신뢰성을 보장하지는 않는다. 프로브는 이 계층에서 얻어낸 정보 역시 로그파일에 함께 저장하게 된다 [3, 7].

4.1.4 어플리케이션 계층

어플리케이션 계층은 포트 정보와 직접적으로 관련을 가진다. 예를 들어 포트 번호 23 번은 텔넷(Telnet)에 의해 사용되고 있는데 이러한 포트 정보는 RFC 1700 [1]에서 일부 그 사용을 정의하고 있다. IANA 에서는 포트 번호를 크게 세 가지로 분류하고 있는데 다음과 같다 [12].

1. Well-known ports: 포트 번호 0 번부터 1023 번까지는 IANA 에서 그 사용 용도를 지정하며 예를 들어 80 번 포트는 HTTP 에 의해 사용되도록 결정되어 있다.

2. Registered ports: 포트 번호 1024 부터 49151 까지를 의미하며 IANA 에 의해 통제되고 있지는 않지만 효율적인 인터넷 사용을 위해 특정 응용 프로그램이 사용할 포트 번호를 등록하게 된다.

3. Dynamic ports: 49151 부터 65535 까지를 의미하며 이 포트 번호는 임의로 선택된다.

응용 프로그램 계층에서는 정말로 다양한 프로토콜들이 존재한다. 예를 들어 HTTP 의 경우 웹의 인기가 높아지면서 매우 다양한 방향으로의 개발이 이루어져 왔는데 그에 따라 HTTP 가 많은 대역폭을 소비하게 되는 결과를 낳았다. 인터넷 관련 응용 프로그램들이 지속적으로 다양하게 개발되고 있기 때문에 네트워크 모니터링 시스템은 그러한 응용 프로그램들에 대한 정보 또한 잘 파악하고 있어야 한다.

4.2 뷰어

WebTrafMon 뷰어는 세 가지 요소로 구성되어 있는데 데이터 리더, 분석기, 뷰 제어기로 나눌 수 있다. 데이터 리더는 로그 파일로부터 데이터를 읽어들이고 분석기는 뷰 제어기가 요청한 데이터를 분석하는 역할을 한다. 뷰 제어기는 사용자와 상호대화적으로 작동하며 사용자가 무엇을 원하는지를 파악한다.

뷰어측에서는 프로브에서 생성된 로그 파일을 가지고 모든 가능한 정보를 표현하게 되는데 주로 네트워크 대역폭 소비에 관한 정보에 중점을 둔다.

- 각각의 네트워크 계층 프로토콜 정보
- 각각의 수송 계층 프로토콜 정보
- 어플리케이션 계층 프로토콜 정보
- 발신지 및 목적지 트래픽 정보
- 발신지 트래픽 정보
- 목적지 트래픽 정보

사용자 인터페이스는 어떤 것이든 적용할 수 있지만 좀더 쉽고 효율적으로 동작할 수 있게 하기 위해 웹 기반의 인터페이스를 채택하였다. 웹기반 인터페이스는 이식 작업 등의 번거로움이 없고 운영체제에 독립적이기 때문이다.

뷰어는 로그파일을 읽어들여 사용자에게 분석 결과를 보여주며 뷰어를 설계함에 있어 모든 가능

한 보안 문제들을 함께 고려하였다. 그래서 가장 일반적으로 쓰이는 패스워드 점검 방식으로 사용자를 인증할 수 있게 시스템을 구성하였다.

5. 구현

본 논문에서는 설계를 실제로 구현하기 위해 웹 기술을 채택하였다. 여기서 좀더 자세한 WebTrafMon 의 구현에 대해서 살펴보도록 하자.

5.1 프로브 구현

프로브에서는 libpcap [13]을 이용하여 패킷 캡처링 루틴을 구현하였다. libpcap 은 운영체제에 독립적인 API 로서 다양한 플랫폼에서 동일한 소스코드를 적용할 수 있게 한다. 패킷 정보를 분석하기 위해서 TCP/IP 계층 구조에 기반을 둔 로그파일 포맷을 정의했는데 다음 그림 4 는 그러한 로그파일의 예를 보여준다.

```
346   164.124.96.18   141.223.82.4    udp   telnet
64    141.223.82.4    141.223.82.26   tcp   http
112   rarp
64    arp
74    141.223.99.99   141.223.82.28   icmp
```

그림 4. WebTrafMon 로그파일 예제

각각의 부분이 의미하는 바는 다음과 같다. 각각의 행은 패킷 하나하나에 대한 크기, 발신지, 목적지, 프로토콜 정보 등을 담고 있으며 첫 번째 열이 의미하는 것은 MAC 계층으로부터 추출된 패킷 크기 정보이다. 두 번째 열은 패킷의 발신지 정보이고 세 번째 열은 목적지 정보를 의미하는데 만약 어떤 패킷이 IP 를 사용하지 않는 패킷이라면 발신지 정보나 목적지 정보는 없으므로 해당 프로토콜에 대한 정보만을 출력하고 더 이상 처리하지 않는다.

그림 4 의 예에서는 3 행과 4 행에서 분석된 패킷이 각각 ARP, RARP 기반의 패킷임을 알 수 있다. 그 다음 열에서는 전송 계층 프로토콜 정보를 나타내는데 가장 일반적으로 많이 쓰이는 TCP 또는 UDP 프로토콜임을 표시해 준다. 만약 전송 계층 프로토콜 중에서 TCP 나 UDP 를 사용하지 않는 패킷에 대해서는 해당 프로토콜 정보를 출력해 준다. 위의 예에서는 마지막 패킷이 ICMP 기반의 패킷임을 알 수 있다. 마지막 열은 어플리케이션 계층에서의 정보를 나타낸다. 예에서는 HTTP 나 FTP 프로토콜을 이용하는 패킷임을 알 수 있는데 이렇게 패킷의 크기정보, 발신지 및 목적지 정보, 프로토콜 정보를 분류해 둠으로서 뷰어측에서 이 파일을 참고로 사용자들에게 원하는 정보를 보여줄 수 있게 된다.

5.2 뷰어 구현

뷰어는 프로브에서 생성한 로그파일을 분석하는 역할을 한다. 여기서는 펄 스크립트 언어를 사용하여 CGI [11]를 통해 사용자의 웹 브라우저와 통신하는 방법을 채택했다. 펄 스크립트는 로그파일로부터 각각의 행, 열을 읽어들여 원하는 정보를 계산해 준다. 다음과 같은 정보를 보여줄 수 있도록 설계하였다.

- 발신지 호스트
- 목적지 호스트
- 발신지 호스트와 목적지 호스트의 조합
- 네트워크 계층 프로토콜
- 전송 계층 프로토콜
- 어플리케이션 계층 프로토콜

뷰어는 각 행으로부터 두 개 혹은 세 개 정도의 열을 읽어들여 이러한 정보를 추출해낸다.

보안을 위해서는 패스워드 점검 과정을 추가하였다. 이것은 뷰어 자체적으로 구현할 수도 있고 웹 서버 쪽에서 구현할 수도 있는데 WebTrafMon 에서는 웹 서버의 설정을 고쳐서 패스워드 점검을 할 수 있게 구현하였다. 예를 들어 아파치 웹 서버는 간단하고도 쉬운 패스워드 점검 루틴을 추가할 수 있는 명령어를 제공한다 [10].

6. 시스템 적용

지금까지 살펴보았던 요구조건 및 구현 방법을 기반으로 실제로 WebTrafMon 이라 이름 붙인 시스템을 개발하였다. 패스워드 점검 과정을 추가해 두었기 때문에 접근이 허락된 사람은 누구라도 네트워크의 상태가 이상하다고 생각되면 즉시 시스템을 이용하여 문제를 파악하고 해결책을 찾아낸다. 그림 5 는 개발된 WebTrafMon 의 초기 화면이다. 크게 두 개의 창으로 나뉘어져 있는데 왼쪽 창은 WebTrafMon 이 보여줄 수 있는 정보들을 나타내고 오른쪽 창은 실제 정보가 정해진 형식으로 보여지게 된다.

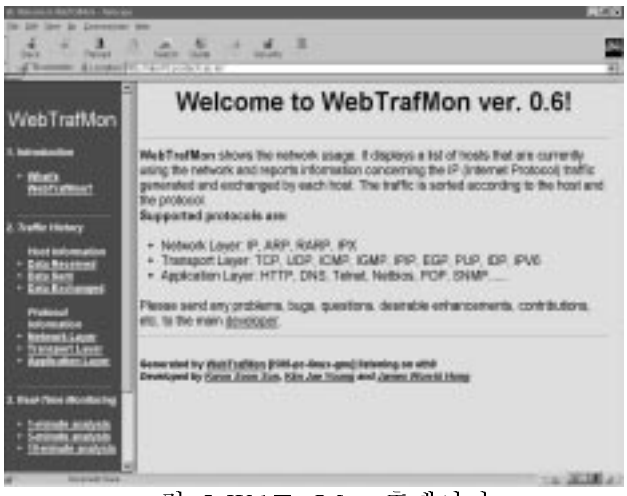

그림 5. WebTrafMon 홈페이지

왼쪽 창에서 "Data Received"는 목적지(destination) 호스트 정보를 나타내고 "Data Sent"는 발신지(source) 호스트 정보를 나타내며 "Data Exchanged"는 발신지 호스트와 목적지 호스트 간의 패킷 교환을 표시한다. "Protocol Information"에서는 각 계층별로 정의된 프로토콜 정보를 나타내고 "Real-Time Monitoring"은 현재 네트워크의 상태를 실시간으로 파악하고 싶을 때 이용한다.

그림 6은 발신지 정보인 "Data Sent" 화면을 나타낸 것이다. 왼쪽 창은 그대로 고정되어 언제든지 사용자의 입력을 받을 준비를 하고 오른쪽 창이 두 개로 갈라져 실제 정보가 오른쪽 아랫 부분에 나타나게 되며 각각의 창은 좀더 쉽게 볼 수 있게 하기 위해 크기를 변화시킬 수도 있다.

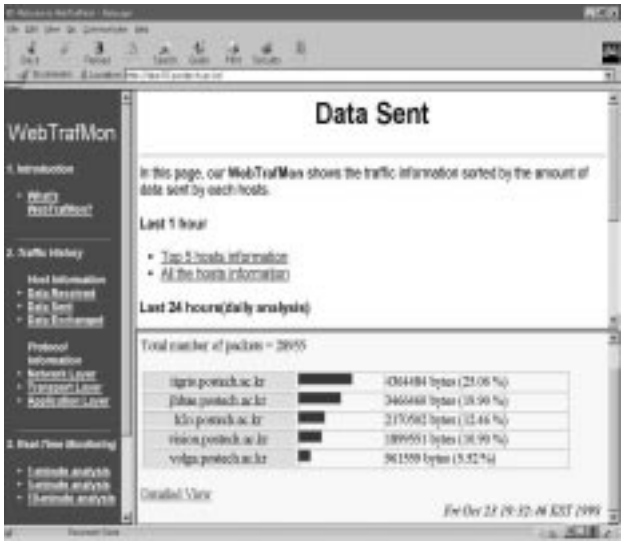

그림 6. 발신지 호스트 트래픽 정보

그림 7은 목적지 호스트 정보인 "Data Received" 화면을 나타낸 것이다. 바로 앞에서 살펴본 발신지 호스트 정보와 형태가 거의 비슷하다는 것을 알 수 있다.

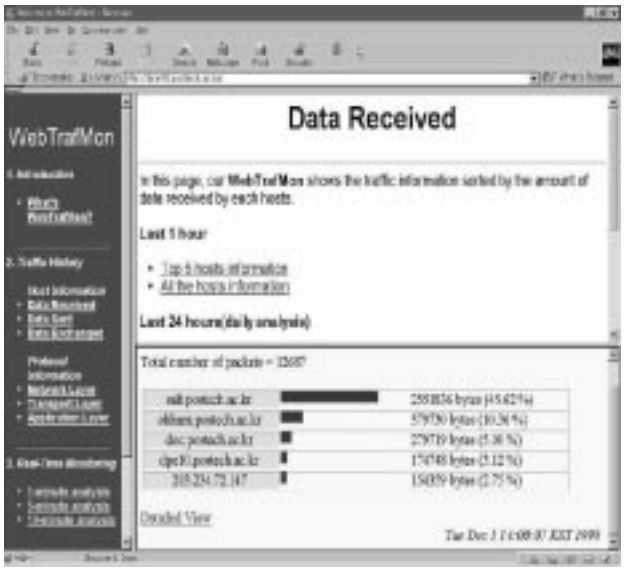

그림 7. 목적지 호스트 트래픽 정보

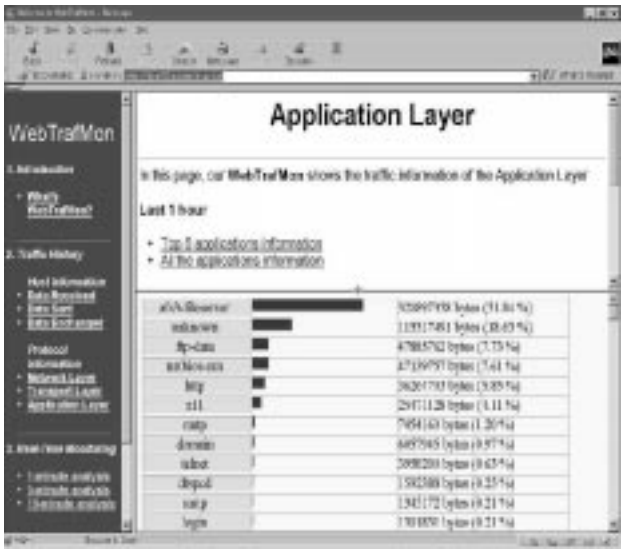

그림 8. 어플리케이션 프로토콜 계층 트래픽 정보

각각의 프로토콜과 관련된 정보들 역시 비슷한 형태로 출력되어 일관성을 유지하였다. 그림 8은 어플리케이션 계층의 트래픽 정보를 나타낸 것이다. TCP/IP 기반의 패킷 헤더를 차례대로 분석하다 보면 가장 상위 계층에 있는 어플리케이션 계층에서 포트정보를 뽑아낼 수 있는데 그 포트정보를 기반으로 어플리케이션 프로토콜을 구분해 내는 것이다.

실시간 정보를 보기 위해서는 "Real-Time Monitoring"을 선택하면 되는데 기본적인 출력 모양은 앞에서 살펴본 그림들과 거의 비슷하며 원하는 정보를 직접 선택하여 줄 수 있다는 것이 다른 점이다. 그림 9는 실시간으로 네트워크 트래픽을 분석하는 예를 보여준다. 오른쪽의 6가지 항목이 히스토리 정보를 나타내 주는 왼쪽 창의 6가지 항목과 일치한다는 것을 알 수 있을 것이다.

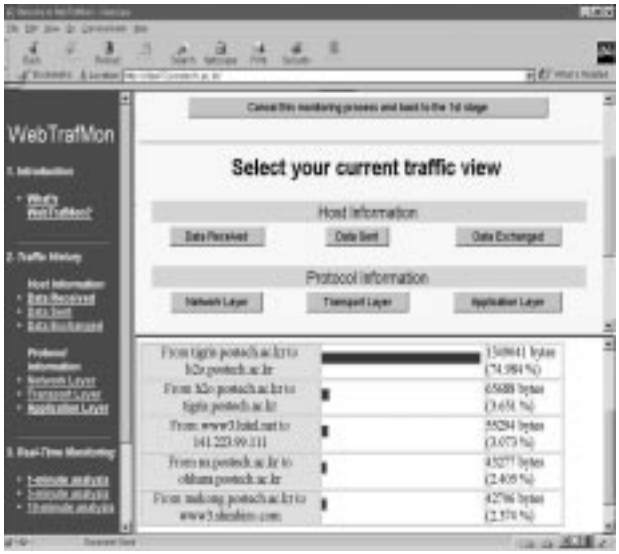

그림 9. 실시간 네트워크 상태 모니터링

실제로 본 시스템을 시험해 본 네트워크 환경에는 네트워크 대역폭의 대부분을 차지하는 AFS[14] 파일 서버가 있었는데 WebTrafMon을 이용하여

네트워크를 모니터링한 결과 예상했던 대로 네트워크 대역폭을 가장 많이 사용하는 응용 프로그램이었음이 명확히 밝혀졌다.

그런데, 고속의 네트워크 선로 상에는 수없이 많은 패킷들이 여러 가지 정보를 가지고 이곳 저곳으로 전송되는데 이런 방법으로 어느 특정 지점에서 네트워크 패킷을 캡처하여 분석하게 되면 짧은 시간 안에 많은 수의 패킷을 처리해야 한다는 제한조건이 따른다. 결과적으로는 자료의 개수가 많아지기 때문에 뷰어 측에서 처리시간이 늘어나게 된다는 약점으로 작용하므로 자료의 개수를 최소한으로 줄이면서 뷰어 측에서 제공하는 분석 결과는 최대한 정확하게 유지하는 방법을 연구하였다.

따라서 프로브 측에서 모든 패킷을 한꺼번에 처리하는 것이 아니라 일정 주기를 두고 통계적인 방법을 적용할 수 있게 일부러 패킷을 건너뛰며 처리하도록 설정을 바꾸었다. 예를 들자면 동일한 네트워크에서 동일한 상태의 패킷 정보를 수집함에 있어 한쪽은 모든 패킷을 다 읽어들이게 설정하고, 다른 한쪽은 모든 패킷을 다 읽어들이는게 아니라 하나씩 건너뛰어 읽어들이게 한다면 데이터의 개수가 절반으로 줄어들 것이다. 이렇게 패킷을 샘플링하여 몇 개의 패킷을 건너뛰고 몇 개의 패킷을 읽어들일지 지정해 줌으로서 데이터의 개수를 줄일 수가 있었다.

물론 이런 방법은 네트워크 트래픽 상태 정보를 부정확하게 하는 요인이 된다. 그렇지만 아주 자세한 정보를 필요로 하는 것이 아니라 장기간에 걸친 네트워크의 전반적인 상황을 파악하고 병목현상을 자주 일으키는 호스트를 찾아내는 등의 일반적인 경우라면 별로 문제가 되지 않는다. 이런 방법은 하드웨어 시스템 사양이 좋지 못한 환경에서 낮은 사양의 하드웨어로도 거의 비슷한 결과를 산출해낼 수 있게 하며 하드웨어 사양이 좋다고 하더라도 고속 네트워크 환경 하에서는 같은 시간 동안 전송되는 패킷의 개수가 그만큼 많을 것이기 때문에 두 경우 모두 시스템의 부담을 덜어줄 수 있는 좋은 방법이라 생각된다.

요즈음은 여러 가지 다양한 멀티미디어 정보들이 서비스되는 경향이 있어 AOD 서버나 VOD 서버가 네트워크에 뜻하지 않은 부담 요인으로 작용하는 경우도 많다. 만약 어떤 네트워크 세그먼트 안에 있는 복수의 사용자가 동시에 AOD 서버로부터 음악을 전송 받아 감상하게 되면 곧 그 네트워크는 포화 상태에 이르게 될 정도로 멀티미디어 정보는 상당히 많은 대역폭을 차지한다. 그럴 때 네트워크 관리자는 WebTrafMon 을 이용하여 어느 특정 호스트로부터 특별히 많은 패킷이 전송되는지 알아내어 조치를 취할 수 있게 된다.

WebTrafMon 을 자동으로 실행할 수 있게 하여 주기적으로 네트워크 상태를 모니터링 하는데 이용하면 장기간의 네트워크 상태에 대한 경향을 파악할 수 있게 된다. 이러한 WebTrafMon 의 특징은 네트워크를 좀더 자세하고 효과적으로 파악할 수 있는 또다른 방법을 제시해 주었다.

7. 결론 및 향후과제

본 논문에서는 WebTrafMon 이라는 이름의 웹 기반의 네트워크 트래픽 모니터링 및 분석 시스템의 개발 과정을 설명하였다. 왜 웹 기반인가? 오늘날 컴퓨터 네트워크의 급속한 성장은 주로 인터넷의 급속한 사용 증가에 기인한다. 인터넷은 정보를 찾고, 사용하는 새로운 방법을 제시하였다. 인터넷을 더욱 인기 있게 만든 것은 웹이다. 웹을 사용하는 것은 다른 어떤 텍스트 기반의 툴을 사용하는 것보다 쉽다. 웹 기반의 시스템은 웹 서핑을 해본 사람들이라면 누구에게나 친근한 느낌을 주며 인터넷에 연결된 사람이라면 누구라도 사용할 수 있다는 장점이 있다. 필요한 것은 오직 하나. 인터넷에 연결되어 있어야 한다는 것뿐이다. 요즘의 상황에 비추어 본다면 이것은 전혀 어려운 요구조건이 아님을 잘 알 수 있을 것이다. 이러한 사실들로부터 이상적인 네트워크 트래픽 모니터링 및 분석 시스템은 웹 기반의 시스템이어야 한다는 것을 확신하게 하였고 개별적인 특정 운영체제를 사용함으로서 발생하는 여러 가지 문제들을 해결할 수 있는 발판을 마련해 주었다.

쉽고 강력한 시스템을 개발하고자 노력했으며 웹 기반의 인터페이스를 채택하는 것 이외에 트래픽 분석 구조를 각각의 네트워크 계층별로 정의하였다. 이러한 정의 방법이 사용자들에게 좀 더 쉽게 네트워크의 상태를 이해할 수 있는 가장 표준적이면서도 효과적인 방법 중의 하나라고 확신한다.

발신지 정보와 목적지 정보를 보여주는 것은 우리가 개발한 시스템을 특별한 것으로 만들어주는 또 하나의 특징이다. 엔터프라이즈 네트워크 관리자들에게는 네트워크가 병목 현상을 일으킬 때 어떤 프로토콜에 의한 패킷이 병목 현상을 일으키는지 알아보는 것보다는 어떤 호스트에서 전송되는 패킷이 병목현상을 일으키는지 알아내는 것이 문제 해결에 효과적인 경우가 훨씬 많다. WebTrafMon 은 그러한 정보를 잘 보여줄 수 있으며 네트워크 사용 현황에 대해서 좀더 잘 파악할 수 있게 해줄 것이다.

지금까지 살펴본 것이 WebTrafMon 이 표현할 수 있는 정보들이었다. WebTrafMon 의 기본적인 요소들은 이미 구현이 끝났지만 성능향상을 위해 수정하거나 고쳐야 할 것은 아직도 많이 남아있다. 예를 들면 좀더 빠른 네트워크 환경에서 이 시스템을 적용하기 위해서는 프로브가 좀더 효율적으

로 작동하여야 하며 뷰어 측에서도 대용량의 데이터를 효과적으로 분석할 수 있는 방법을 재고해 보아야 한다. 그렇지만 이러한 것들은 네트워크 상태를 미리 예측할 만한 신뢰할 만한 방법이 없다는 점에서 매우 어려운 일이다.

WebTrafMon 을 SNMP 기반으로 동작하는 MRTG [6]와 연동하는 것도 좋은 아이디어가 될 수 있다. 네트워크에서 병목 현상이 일어날 때마다 MRTG 에서 자동으로 WebTrafMon 을 동작시킬 수 있게 시스템을 설정한다면 문제가 되는 특정 시점에서 문제 해결 방법을 좀더 쉽게 제시할 수 있게 될 것이다.

패킷 샘플링과 관련한 통계적 처리방법에 대한 보다 심도있는 연구도 필요하다. WebTrafMon 에서는 간단하게 N 개의 패킷마다 한 개씩 패킷을 읽어들이는 방법으로 로그파일의 크기를 1/N 으로 줄일 수 있었지만 좀더 많은 경우에 대해서 테스트하고 가장 정확성이 높은 결과를 찾아내는 일이 필요하다.

장기간의 모니터링이 가능하도록 하는 기능을 추가해야 한다. 현재로서는 한시간 전의 네트워크 상태와 하루(24 시간)전의 네트워크 상태만을 보여주고 있지만 이를 확장하여 일주일, 한달 전의 상태까지 보여줄 수 있게 하는 작업이 필요하다.

그리고 마지막으로 WebTrafMon 은 현재 일반적인 공유 이더넷 기반에서만 작동하고 있는데 스위칭 이더넷, FDDI, ATM 등 다양한 네트워크 환경에서도 작동할 수 있게 하는 것 또한 활용도를 높일 수 있는 좋은 방법이 될 것이다.

참 고 문 헌

[1] J. Raynolds, J. Postel, "Assigned Numbers", RFC1700, Network WG, October 1994.

[2] Andrew S. Tanenbaum, *Computer Networks*, Prentice-Hall, 1996.

[3] William Stallings, *Data and Computer Communications*, Prentice-Hall, 1997.

[4] W3C, "HTTP/1. 1 Performance Overview", http://www.w3.org/pub/WWW/Protocols/HTTP/Performance/.

[5] T. Berners-Lee, R. Fielding, H. Frystyk, "HyperText Transfer Protocol - HTTP/1.0", RFC 1945, IETF HTTP WG, May 1996.

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

[7] W. Richard Stevens, *TCP/IP Illustrated, Volume 1*, Addision-Wesley, 1994.

[8] Gary R. Wright, W. Richard Stevens, "TCP/IP Illustrated, Volume 2", Addison-Wesley, 1994.

[9] David Perkins, Evan McGinnes, *Understanding SNMP MIBs*, Prentice-Hall, 1997.

[10] Apache Development Group, "Apache Web Server 1.31", http://www.apache.org.

[11] NCSA, "The Common Gateway Interface", http://hoohoo.ncsa.uiuc.edu/cgi.

[12] IANA, "Protocol Numbers", ftp://ftp.isi.edu/iana/assignments/protocol-numbers.

[13] Laurence Berkeley National Laboratory, "libpcap 0.46", ftp://ee.lbl.gov.

[14] Transarc Inc., "AFS", http://www.transarc.com.

[15] J. W. Hong, J. Y. Kong, T. H. Yun, J. S. Kim, J. T. Park and J. W. Baek, "Web-based Intranet Services and Network Management", IEEE Communications Magazine, Vol. 35, No. 10, October 1997, pp. 100-110

[16] J. Won-Ki Hong, S. U. Park, Y. M. Kang and J. T. Park, "Enterprise Network Traffic Monitoring, Analysis and Reporting Using Web Technology", Accepted to be published in the Journal of Network and Systems Management, Plenum Press, 1999.

[17] Carter Bullard, "argus-1.7.beta.1b", ftp://ftp.sei.cmu.edu/pub/argus.

[18] Dave Curry and Jeff Mogul, "nfswatch-4.3", http://ftp.lip6.fr/pub2/networking/nfs/.

[19] Lawrence Berkley National Laboratory, "tcpdump 3.4a6", ftp://ftp.ee.lbl.gov.

[20] Craig Hunt, *TCP/IP Network Administration*, O'Reilly and Associates, Inc., 1992

권 순 선

1997 연세대학교, 전파공학 학사
1999 포항공과대학교, 전자계산학 석사
1999-현재 삼성전자 연구소 근무
관심분야: 인터넷 서비스 관리

김 재 영

1994 포항공과대학교, 전자계산학 학사
1996 포항공과대학교, 전자계산학 석사
1996-1998 포항공과대학교 전자계산소 연구원
1998-현재 포항공과대학교 컴퓨터공학과 박사과정
관심분야: 네트워크 및 분산 시스템 관리, 분산처리, CORBA, 인터넷 서비스 관리

홍 원 기

1983 Univ. of Western Ontario, 전산학 학사
1985 Univ. of Western Ontario, 전산학 석사
1985-1986 Univ. of Western Ontario, 전산학과 강사
1986-1991 Univ. of Waterloo, 전산학 박사
1991-1992 Univ. of Waterloo, Post-Doc fellow
1992-1995 Univ. of Western Ontario, 연구교수
1995-현재 포항공대 컴퓨터공학과 부교수
관심분야: 분산처리, 네트워크 및 분산 시스템 관리, CORBA, Internet 관리.