인터넷 응용 서비스의 통계에 근거한 트래픽 분석

정태수* · 최진섭** · 정중수** · 김정태*** · 김대영****

Traffic Analysis of Statistics based on Internet Application Services

Tae-Soo Jeong* · Jin-Sub Choi** · Joong-Soo Chung** · Jung-Tae Kim*** · Dae-Young Kim****

요 약

오늘날 인터넷의 백본 발달과 더불어 수많은 응용 서비스들이 사용되고 있다. 이러한 응용 서비스는 인터넷 초기 출현 시에는 웹, 파일전달, 이메일 등의 well-known 포트를 사용한 서비스가 주축을 형성하였다. 그러나 최근 인터넷의 폭발적인 사용과 다양한 컨텐츠의 요구로 unwell-known 포트를 사용한 서비스가 매우 많이 등장하였다. 인터넷 트래픽을 모니터링하여 (un)well-known 포트를 사용하는 패킷의 PDU 정보의 분석 후, 응용 서비스의 유형 및 통계 정보를 구하는 기법은 트래픽 분석자에게 매우 유용한 정보이다. 본 논문에서는 우선 TCP와 UDP 위에서 동작하는 (un)well-known 포트를 사용하는 패킷들의 통계 정보를 ethereal에서 제시된 netflow 및 tcpdump 기법으로 수집하여, 사용빈도가 높은 포트의 응용 서비스 제시와 동작과정을 분석하였다. 이후 분석된 응용 서비스의 면밀한 통계를 위해 원시 데이터 트래픽을 수집하는 에이전트와 에이전트로부터 전달받은 트래픽을 BNF(Backus-Naur Form) 기법을 활용하여 서버에 적용하는 과정을 제안하였다. 또한 제안된 과정을 안동대학교 네트워크 환경에 적용하여 인터넷 트래픽 서비스 유형과 응용 서비스의 면밀한 통계 결과를 제시하여 트래픽 분석자에게 매우 유용한 정보를 제공하였다.

ABSTRACT

A number of Internet application services are used with the development of Internet backbone nowadays. Well-known services such as WWW, FTP, email are provided at first time. Tremendous unwell-known services are presented according to the demands of various contents. After analyzing PDU information of the packet using unwell-known port travelling on the internet, searching internet service type and its statistical data is provided with internet traffic analyst as very useful information. This paper presents the mechanism to extract the internet application services operated on (un)well-known port of UDP or TCP used occasionally through netflow and tcpdump method introduced by ethereal and the operation scheme of the service. Afterwards to get the detailed statistics of the analyzed application service, the agent and the server environment, the agent gathering raw data traffics and the server adapting the traffic received from the agent BNF(Backus-Naur Form) method, is also introduced. Adapting the presented mechanism over LAN of Andong national university, the internet traffic service type and the detailed statistics of the analyzed application services which provides with internet traffic analyst are presented as very useful information.

키워드

트래픽, 인터넷, 컨텐츠

*한국전자통신연구원 **안동대학교 공과대학 전자정보산업학부

***목원대학교 공과대학 전자정보보호공학부 ****충남대학교 정보통신공학부

접수일자 : 2004. 5. 1

Ⅰ. 서 론

오늘날 인터넷의 백본망과 라우터의 급속한 발달과 더불어 수많은 응용 서비스들이 사용되고 있으며, 향후에도 다양한 컨텐츠가 요구되고 있다. 한편 종단 사용자 관점에서 살펴보면 이더넷 LAN 카드를 장착한 사용자 수는 LAN 뿐만 아니라 공중망의 ADSL(Asynchronous Digital Subscriber Loop) 까지 확대되어 가는 추세이다.

인터넷 초기에는 TCP(Transmission Control Protocol)나 UDP(User Datagram Protocol) 프로토콜을 활용하는 응용 서비스 중 웹, 파일전달, 이메일 등의 well-known 포트를 사용하는 서비스가 주축을 형성하였다. 그러나 최근 인터넷의 폭발적인 사용과 다양한 컨텐츠의 요구로 unwell-known 포트를 사용하는 서비스가 매우 많이 등장하였다. 다양한 서비스의 등장과 네트워크의 발달로 인터넷 상에서 동작하는 프로토콜의 면밀한 분석을 위하여 국내외에서는 프로토콜 분석장비가 많이 출시되고 있다[1,2,3]. 이러한 프로토콜 분석장비는 대다수 TCP/IP 프로토콜 슈트만 분석하고 있다. 아울러 많은 네트워크 관련 기관들은 앞 다투어 새로운 응용 서비스의 출현으로 unwell-known 포트를 사용하는 서비스의 속성을 제시하고 있다[4]. 일반적으로 unwell-known 포트를 사용하는 서비스는 TCP나 UDP의 well-known 포트번호 사용범위를 제외한 임의의 포트번호를 사용한다. 최근에는 방화벽 등의 공격을 피하기 위해 well-known 포트번호를 활용하여 임의의 서비스를 제공하기도 하는 실정이다.

본 논문에서는 우선 TCP와 UDP 위에서 동작하는 (un)well-known 포트를 사용하는 패킷들의 통계 정보를 ethereal에서 제시된 netflow기법[5]으로 수집하여 사용빈도가 높은 포트 번호의 추출과 그들의 응용 서비스 대응 관계를 제시하였다. 아울러 tcpdump기법[6]으로는 netflow 기법에서 제시된 포트 번호 추출뿐만 아니라 원시 데이타의 수집 및 분석을 통해 포트 번호에 대한 응용 서비스 추출 관계를 추가로 제시하였다. 또한 제시된 응용 서비스의 동작과정을 분석 한 후 분석된 서비스의 면밀한 통계를 위해 원시 데이터 트래픽을 수집하는 에이전트와 에이전트로부터 전달받은 트래픽에 분석된 동작과정을 기반으로 BNF(Backus-Naur Form) 기법[7]을 서버에 적용하는 과정을 제안하였다. 또한 제안된 과정을 안동대학교 네트워크 환경에 적용하여 인터넷 트래픽 서비스 유형과 응용 서비스의 면밀한 통계 결과를 제시하여 트래픽 분석자에게 매우 유용한 정보를 제공하였다. 분석을 위해 사용된 환경으로는 펜티움 II 프로세서 800 MHz PC 기반의 Linux 기반 OS를 축으로 하였다.

Ⅱ. 분석 시스템 설계

어떤 기관에서 외부 인터넷 망과 접속되는 관문국 라우터(본고에서는 라우터라고만 함)에서 송, 수신되는 트래픽을 분석하여야 그 기관에서 사용하는 트래픽 패턴을 파악 할 수 있다. 따라서 라우터를 통한 송, 수신 트래픽을 측정 및 분석하기 위해서는 라우터나 그에 접속된 허브등의 장바를 통해 PC에서 수행한다. 트래픽 측정은 ethereal에서 제시된 netflow 및 tcpdump 기법을 활용하였고, 면밀한 통계를 얻기 위해 서버 및 에이전트 환경을 활용하였다.

1. netflow 기법을 사용한 경우

netflow는 TCP나 UDP 패킷의 포트번호별 사용통계를 구하는 방법만 제시될 뿐 실제 PDU 내부의 원시 데이터는 측정되지 않는다. netflow를 이용한 트래픽 측정방법은 라우터가 일정 시간동안 수집한 트래픽을 인터넷 주소 등이 할당된 온-라인 상의 특정 PC로 전달한다. 이를 위해 라우터의 환경 설정을 다음과 같은 명령을 수행하여 global configuration 모드 상태로 한다.

```
# configure
# ip cef
(config)# interface ATM1/0 (설정하고 싶은 인터페이스)
(config-if)# ip route-cache flow
(config-if)# exit
(config)# ip flow-export destination 203.255.255.248 2055
          (cflowd가 설치된 시스템의 IP주소)
(config)# ip flow-export version 5 peer-as
          (netflow 버전이 5이고 AS는 인 접한 AS에 대한
           정보를 얻고자 함. 만약 원래의 소스 정보를
           얻으려면 origin으로 설정)
```

통상 어떤 인터페이스에 netflow를 활성화하면 해당 인터페이스로 입, 출력되는 트래픽에 대해서 그 정보를 생성해 준다. netflow 기법을 활용하기 위한 환경으로는 arts++와 cflowd가 사용된다. 이를 위해서는 우선 FreeBSD를 설치한 후 http://www.caida.org의 arts++와 cflowd 다운받아 설치한다. 설치가 완료되면 통상 설치된 파일은 /usr/local/arts 아래에 존재 한다. cflowd 설치 후에는 설정 파일을 환경에 맞도록 수정을 해 주어야 한다. cflowd가 설치된 디렉토리로 이동하면 etc디

렉토리상의 cflowd.conf.example와 cfdcollect.conf.example 파일을 각각 cflowd.conf와 cfdcollect.conf 파일로 변경 한 후 그 내용을 사용자 요구에 맞도록 수정한다.

```
flowd.conf 설정방법
OPTIONS {
  LOGFACILITY:        local6
 TCPCOLLECTPORT:  2056
  PKTBUFSIZE:        1048576
  TABLESOCKFILE:
    /usr/local/arts/etc/cflowdtable.socket
  FLOWDIR:
    /usr/local/arts/data/cflowd/flows
  FLOWFILELEN:          1000000
  NUMFLOWFILES:         10
  MINLOGMISSED:         1000
}

COLLECTOR {
   HOST:              211.248.0.136
   # IP address of central collector
   ADDRESSES: { 211.248.0.136 }
   AUTH:              none
}

CISCOEXPORTER {
   HOST:              211.248.0.254
   ADDRESSES:  { 211.248.0.254 }
   CFDATAPORT: 2055
   SNMPCOMM: 'public'
   LOCALAS:    0
   COLLECT:
     { protocol, portmatrix,
       netmatrix, flows }
}
```

netflow를 이용하여 트래픽을 측정하면 arts.k(k는 일반적으로 20031005와 같은 연월일로 표기됨)과 같은 파일이 생긴다. 그림 1과 같이 측정된 파일은 일반적으로 분석하기에 용량이 너무 크므로 arts.k1, arts.k2 .. 등 몇 개로 분리한 후 각각 파일의 트래픽 통계를 sport_1.txt(source port의 약칭임), sport_2.txt, dport_1.txt(destination port의 약칭임), dport_2.txt 등으로 포트 번호별로 통계를 취하는 절차와 방법은 다음과 같다.

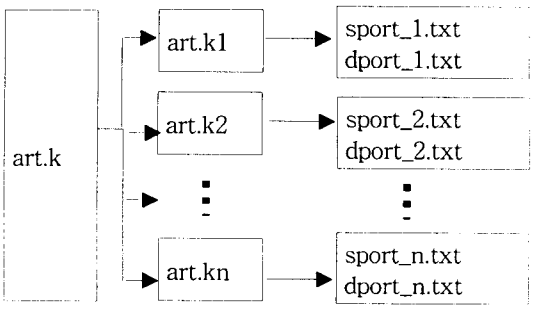


그림 1. 파일 분리 작업
Fig. 1. File Division Procedure

1단계: artsportms를 이용해 포트별 데이터로 변환한다. 본 예에서 infile은 arts.k1이고 이 명령어를 수행한 결과는 port_1.txt 파일이 된다.

```
usage: artsportms [-p] [-s srcPort] [-d dstPort] infile(s)
```

2단계: 리눅스나 유닉스의 sort 명령어를 통하여 파일을 정렬한다. sort -k 5,5는 다섯 번째 파라메터인 Bytes 순으로의 정렬을 의미한다. 이 명령어를 수행한 결과는 sort_Result_by_Bytes 파일이 된다.

```
cat ./port_1.txt | sort -k 5, 5 > sort_Result_by_Bytes
ex) port_1.txt 부분중
srcPort  dstPort  Pkts    Pkts/sec  Bytes      Bits/sec
-------  -------  ------  --------  ---------  ----------
20       1570     164412  843.138   246458565  1.01111e+07
2806     2090     11015   56.4872   15478324   635008
```

3 단계: 리눅스나 유닉스의 sort 명령어를 통하여 발신지와 착신지 포트별로 sorting한다.

```
cat ./sort_Result_by_Bytes | sort -k 2, 2 > dport_1_byte.txt
ex) sport_1_byte.txt 부분중
0  1230  1  0.00333333  52  1.38667
0     7  1  0.00332226  284  7.54817
```

4단계: arts.k의 분리된 파일을 sport_1_byte.txt, sport_2_byte.txt 등 파일을 모두 합산 후 다시 정렬한다.

```
cat ./sport_1_byte.txt >> sport_total.txt
cat ./sort_total.txt | sort -k 1, 1 > sport_total_byte.txt
```

5 단계: 각 포트별 모든 트래픽을 합산하는 프로그램을 C 언어로 작성하여 포트별 트래픽 통계치를 구한다.

```
while(1)    {
   while( (c = fgetc(read)) != '\n')   {
      if(c == EOF)    {
         check(buf, &num, i, write);  //port 번호와 트래픽 양 추출
          fprintf(write, " %10d - %15.1lf  \n", num, sum);
          fclose(write);
          return;
      }
          buf[i++] = c;
   }
}
```

6단계: stotal_byte.txt와 dtotal_byte.txt에서 바이트 크기순으로 다시 정렬한다.

```
cat ./stotal_byte.txt | sort -k 3, 3 > sport.txt
cat ./dtotal_byte.txt | sort -k 3, 3 > dport.txt
```

7단계: 포트별 해당 응용 서비스를 추출하는 프로그램을 C언어로 작성한다.

```
while(1)    {
    while( (c = fgetc(read)) != '₩n')  {
        if(c == EOF)    {
            check(buf, &num, i, write);  //port 번호와 트래픽 양 추출
            fprintf(write, "  %10d  -  %15.1lf  ₩n", num, sum);
            fclose(write);
            return;
        }
            buf[i++] = c;
    }
    fseek(read1, 0, SEEK_SET);
    if(temp < 65000 )   {
        while(--temp != -1)   {        //port 번호의 응용 찾기
            while( (c = fgetc(read1)) != '₩n');
        }
        if( (c = getc(read1)) != '₩n')   {
            fscanf(read1, "%s", ptr);  //port 번호와 응용 추출
            fprintf(write, "%20d   %c%s   ₩n", q, c,ptr);
        }
    }
}
```

8단계: 6 단계를 거쳐서 생긴 트래픽 양의 순으로 정렬된 포트번호들과 7 단계를 통해 얻은 결과인 응용 서비스를 통해 포트번호와 응용 서비스를 대응시킨 결과에 포트별 트래픽 비율을 계산한다.

실행결과

4662	eDonkey-2000	21.6%
138	netbios-dgm	5.3%
20	ftp-data	5.1%
80	http	4.8%
9298	-	3.9%
6891	-	2.8%
1494	ica	2.5%

2. tcpdump 기법을 사용한 경우

tcpdump 기법은 netflow처럼 포트별 해당 응용 서비스를 추출할 수 있을 뿐 아니라 TCP나 UDP 패킷의 PDU 내부의 원시 데이터를 얻을 수 있으며, 이것으로 부터 그 포트의 응용 서비스를 유추할 수 있다. 대부분은 Linux, FreeBSD 배포판에는 tcpdump와 pcap(이는 라이브러리 임)이 포함되어 있다. 단 최신 버전이 아니라면 http://www.tcpdump.org에 가서 pcap library와 tcpdump를 다운받아 설치한다. 분석을 위해 파일로 출력 하려면 통상 다음과 같은 tcpdump의 일반적인 명령어에 옵션으로 처리한다.

```
tcpdump [-adefInNOpqRStuvxX ][ -ccount ]
[ -Cfile_size][ -Ffile ][ -iinterface][ -m module ]
[ -rfile ][ -ssnaplen][ -Ttype ][ -wfile]
[ -Ealgo:secret][ expression ]
```

[expression]은 분석자가 원하는 파일을 생성하는 중요한 옵션으로 분석자는 이 옵션을 어떻게 사용하느냐에 따라 생성되는 파일이 달라질 수 있다.

가. 많이 사용되는 트래픽의 포트번호를 구하는 방법

tcpdump 환경 설정 후 트래픽 포트번호별 사용 통계를 구하는 방법과 절차는 다음과 같다.

1단계: Coral reef(crl_flow)를 이용하여 tcpdump를 통한 통계자료를 플로우 자료로 변환한다. filename을 받아서 outfile로 출력한다. 아래 예에서는 dump1.bin을 1.txt로 출력한다.

```
crl_flow -b -Tf15 -o (outfile) pcap:(filename)
 ex) crl_flow -b -Tf15 -o 1.txt dump1.bin
```

2단계: 상기 플로우 자료를 t2_convert를 이용하여 proto_ports_table 을 생성한다. flowfile은 1.txt이다.

```
t2_convert Proto_Ports_Table < (flow file) > output
#프로토콜, ok필드, src 포트, dst 포트, 패킷수, 바이트수, 플로우수
      6      1     8080     3421      9     3020      1
```

3단계: 2 단계의 결과를 이용하여 TCP, UDP, 별로 port Matrix를 생성한다. 이와 같이 처리한 후에 perl 프로그램을 작성하여 내림차순으로 많이 활용되는 패킷의 응용 포트명을 구한다. PFILE은 1.txt이다.

```
while (<PFILE>) {
        chop;
        next if ( (/^₩s*#/) or (/^₩s*$/) );
        my ($protocol, $ok, $src_port, $dst_port,
           $pkts, $bytes, $flows) = split();   //출력
        next if ($ok ==0);
        $ident = sprintf "%10s %10s", $src_port, $dst_port;
        if ($protocol==6) {  //TCP의 경우
                $tcpcounts {$ident} +=1;
                if ($tcpcounts {$ident} ==1) {
                        push (@tcpports, $ident);
                }
                $tcppkts {$ident} +=$pkts;
                $tcpbytes {$ident} +=$bytes;
                $tcpflows {$ident} +=$flows;
        }
}
```

상기 프로그램 수행 결과 src 포트(source의 발신지를 의미함), dst 포트(destination의 착신지를 의미함), 패킷수, 바이트수, 프로우수로 모두 누적시켜 추후 응용 서비스 판단 시 유용하게 사용된다.

# src 포트, dst 포트, 패킷수, 바이트수, 플로우수				
8080	3421	110	87493	12
4662	3830	234	12056	49
80	8211	1584	576576	1584

4단계: 상기에서 얻어진 port matrix에서 src또는 dst측이 well-known 포트번호이면 반대쪽을 ephemeral 포트로 간주하여 반대쪽 포트 번호를 알려진 포트 번호로 치환한다.

# src 포트, dst 포트, 패킷수, 바이트수, 플로우수				
8080	8080	110	87493	12
4662	4662	234	12056	49
80	80	1584	576576	1584

5단계: 이제 상기 자료를 이용하여 src 포트 또는 dst 포트로 요약한다.

# 포트, 패킷수, 바이트수, 플로우수, 대응된 포트번호수				
8080	110	87493	12	1
4662	234	12056	49	1
80	1695	582852	1598	2

6단계: 상기 결과를 엑셀 환경에서 포트별 서비스를 대응 시킨다.

나. 원시 데이터를 바탕으로 분석 파일생성

tcpdump를 이용하여 캡쳐된 패킷들은 위에서 언급된 "II장 2절 가 항"의 2단계를 거치면서 많이 사용되는 트래픽의 포트번호를 구하여, 포트별 프로우를 분석하기 위해서는 각각의 발, 착신 포트별 패킷을 모아야 한다. 분석 파일 생성은 tcpdump 내부 명령을 사용하여 2 단계에서 구한 트래픽의 포트번호를 바탕으로 원시 데이터를 추출한다.

예를 들어 src, dst port[PortNum] 같은 특별한 조건을 만족시키는 패킷들을 dump_file 로부터 축출하여, 용의한 분석을 하기 위해 각각의 조건에 맞는 파일을 생성하는 과정은 다음과 같다. 즉, tcpdump를 이용한 port(dst,src)별 파일생성은 다음과 같은 절차를 따른다.

- tcpdump -enx -s 1500 -w dump_file.dat 명령을 사용하여 dump_file.dat 라는 파일을 생성한다.
- 그리고 분석자가 dump_file.dat 로부터 src port 1570 인 패킷만을 골라서 port_1570.dat 파일 생성하고자 하면 다음과 같은 명령을 사용할 수 있다.

```
tcpdump -enx -s 1500 -r dump_file.dat
-w port_1570.dat src port 1570
```

이와 같은 명령을 사용하면 port_1570.dat 파일에는 src port 1570에 해당하는 모든 패킷이 입력된 순서대로 위치한다.

- 같은 방법으로 dst port 1570에 대한 패킷들을 원한다면 다음과 같다.

```
tcpdump -enx -s 1500 -r dump_file.dat
-w port_20.dat dst port 1570
```

- 또한 src, dst 와 상관없이 port 1570 에 대한 모든 패킷을 원한다면 다음과 같은 명령을 사용한다.

```
tcpdump -enx -s 1500 -r dump_file.dat
-w port_20.dat port 1570
```

[expression] 옵션은 은 여러 가지 조건을 비교적 자유롭게 조합할 수 있다. 이런 방법으로 생성된 파일들은 초기에 dumpfile 보다 분석이 용의하여, 분석자 입장에서 매우편리하게 이용할 수 있다.

3. 면밀한 분석 기법을 사용한 경우

앞서 언급한 netflow나 tcpdump는 각각의 패킷별 트래픽을 측정하거나 분석하는데 사용되었다. 이렇게 구한 응용 포트들은 IANA에서 권고된 포트 번호 사용 규약[4]을 지키지 않은 경우가 빈번하다. 80 포트는 웹 서비스를 권고하지만 방화벽 등을 통과하려는 목적으로 임의의 응용 서비스의 포트로 활용되기도 한다. 따라서 IANA에서 정의된 많이 사용되는 well-known 포트중 임의의 응용 서비스 사용에 대한 통계를 구하는 것은 매우 중요하다. 이를 위해 트래픽 플로우 단위의 기반으로 측정되어 실제 어떠한 특정 서비스가 얼마만큼의 트래픽을 발생시키는지를 알아볼 수 있도록, 분석된 응용 서비스의 면밀한 통계를 위해 BNF 기법을 활용하여 에이전트 및 서버에 적용 과정을 그림 1처럼 제안하였다.

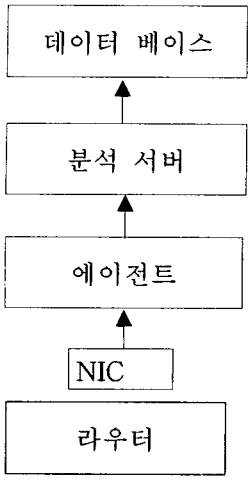


그림 2. 면밀한 분석 환경
Fig. 2. Environment of Detailed Analysis

라우터로부터 트래픽을 수집한 에이전트는 분석 서버에게 가공할 형태로 전달한다. 이후 서버는 분석자가 쉽게 접근하도록 포트의 정확한 통계 정보를 데이터베이스화 한다.

- 에이전트 기능: 원시 IP 패킷을 플로우 단위로 패킷들을 조립하여 서버로 보내주는 역할을 한다.
- 서버 기능: 에이전트에서 캡처한 플로우를 포트명으로 메핑하여 자체 BNF 기법으로 응용 인식을 수행하여 통계 자료를 데이터베이스화 하여 보여준다.

III. 분석 방법 및 결과

본 고에서 제안된 트래픽 분석 환경은 국립 안동대학교 전산망을 활용하였으며, 전산망은 외부 인터넷 망과 DS3 급의 전송로를 통한 시스코 7507 라우터 장비와 접속된다. 내부망은 다시 기가비트 이더넷 망으로 서브넷과 접속 되고 있으며, 안동대학교 전산망을 통한 트래픽 분석을 위하여 netflow와 tcpdump 기법을 사용하였다. netflow 기법은 7507 라우터에서 수집한 패킷들의 정보를 내부망을 통해 분석하는 PC에게 전달해 준다. tcpdump 기법은 7507 라우터에 허브를 우선 접속한다. 이후 허브에 임의의 한 포트를 할당하여 트래픽 분석용 PC를 접속시킨다. 따라서 tcpdump 기법은 netflow 기법보다 번잡하나, 패킷 트래픽의 원시 데이터까지 추출할 수 있는 장점이 있다. 많이 사용되는 well-known 포트 중 임의의 응용 서비스 사용에 대한 통계를 구하기 위한 면밀한 분석을 서버 및 에이전트 환경으로 수행하였다.

1. netflow 기법을 사용한 경우

그림 1은 국립 안동대학교 네트워크에서 netflow를 이용한 분석 환경의 구성도를 나타내고 있다. 분석 환경은 시스코 라우터 7507(7507 라우터로 명함)과 교내 인터넷 상에 접속된 netflow가 설치된 PC(PC로만 명함)로 구성된다.

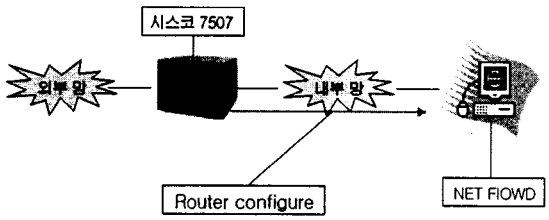


그림 3. netflow 시험 환경
Fig. 3. Netflow Test Environment

7507 라우터는 PC에게 인터넷 주소를 한개 할당하여 UDP를 이용하여 취합된 트래픽을 PC에게 전달한다. 이후 PC에서는 트래픽을 II장 1절에 서술된 개념을 적용하여 분석하였다. 안동대학교 교내 망에서 상기의 방법을 활용하여 평일 낮 시간인 2004년 3월 3일 측정한 트래픽의 포트별 통계 자료를 표 1과 같이 정리하였다.

표 1. 트래픽 포트별 통계 자료
Table 1. Statistical Analysis of Traffic Port

응용 서비스 명칭	사용포트	총 바이트 수(Mbyte)	비율(%)
웹	80	63.4	42.8
당나귀	4662	43.3	30.6
ftp(데이타)	20	5.02	4.56
벅스뮤직	1755	3.01	2.56
	8080	2.68	2.29
	6891	2.61	2.7
구루구루	9292	2.07	1.8
Winmx, 타키, 세이클럽 메신저	6699	1.90	1.7
ftp(제어)	21	1.68	1.5

2. tcpdump 기법을 사용한 경우

그림 2는 국립 안동대학교 네트워크에서 tcpdump를 이용한 분석 환경의 구성도를 나타내고 있다.

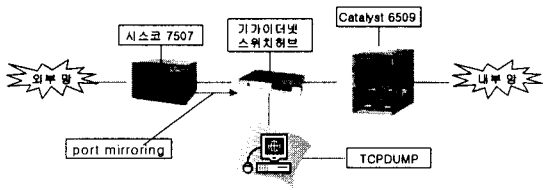


그림 4. 시스템의 구성도
Fig. 4. Configuration of system

위 그림과 같이 7507 라우터에서 포트 미러링 기법을 적용하여 기가빗 이더넷 허브에서 포트출력으로 포트 통계 정보는 netflow 기법을 사용한 것과 동일하묘, 추가로 원시 데이터를 추출하여 분석하는 방법을 도출하였다.

3. 면밀한 분석 기법을 사용한 경우

netflow 기법에서 제시된 포트 번호 추출과 포트 번호에 대한 응용 서비스 추출 관계의 제시 후 응용 서비스의 동작과정을 분석하였다. 분석된 서비스의 면밀한 통계를 위해 원시 데이터 트래픽을 수집하는 에이전트와 에이전트로부터 전달받은 트래픽에 분석된 동작과정을 기반으로 BNF(Backus-Naur Form) 기법을 서버에 적용하는 과정을 안동대학교 네트워크 환경에 적용환경을 그림 4와 같이 수행하였다. 7507 라우터에서는 허브로 연결되어 외부와 송, 수신되는 2개의 포트를 NIC 카드로 에이전트와 접속한다. 라우터에 송, 수신되는 트래픽을 서버에게 전달하면 서버는 데이터베이스화하여 사용자는 외부의 인터넷 환경에서 인터넷 트래픽 정보를 분석 할 수 있도록 하였다.

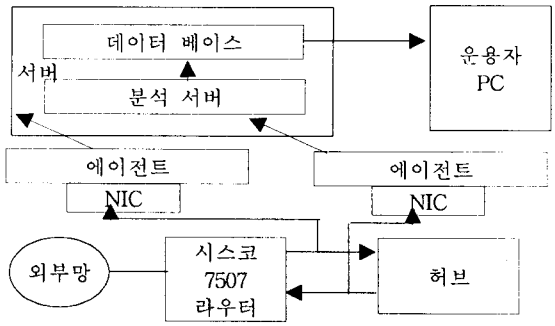


그림 5. 면밀한 분석 환경
Fig. 5. Environment of Detailed Analysis

표 1의 응용 포트를 그림 4에 의한 면밀한 분석 결과를 표 2에 나타내었다.

표 2. 안동대 네트워크에서의 응용 트래픽의 면밀한 분석
Table 2. Detailed analysis of the analyzed application traffics at Andong National University

응용 서비스 명칭	사 용 포트	총 바이트수/ 순수 컨텐츠 (Mbyte)	비율(%)
웹	80	63.4/56.4	89
당나귀	4662	43.3/41.1	95
ftp(데이타)	20	5.02/50.0	
벅스뮤직	1755	3.01/3.00	
	8080	2.68	
	6891	2.61	
구루구루	9292	2.07/2.07	
Winmx, 타키, 세이클럽 메신저	6699	1.90/1.80	
ftp(제어)	21	1.68/1.67	

표 2에서 웹을 사용하는 well-known 포트 80은 방화벽 등을 의식하여 약 11%가 웹이 아닌 서비스를 사용하였으며, well-known 포트 4662가 5%가 당나귀가 아닌 다른 응용 서비스를 수행하고 있었다. 또한 20 및 21번 포트의 FTP 서비스 및 여타의 다수 포트는 본래의 서비스 정보 외에는 거의 사용되지 않는다는 결론을 얻을 수 있다.

IV. 결 론

본 논문에서는 TCP와 UDP 위에서 동작하는 (un)well-known 포트를 사용하는 패킷의 PDU 정보에 의한 응용 서비스의 유형을 찾는 트래픽 분석 기법을 수행하였다. 이러한 분석을 위하여 수많은 트래픽 중 활용도가 많은 응용 서비스를 추출하기 위하여, 안동대학교 네트워크에서 ethereal에서 제시된 netflow 및 tcpdump 기법을 활용하였다.

추출된 트래픽의 분석을 위하여 ethereal 트래픽 분석 장치를 활용하였다. 추출된 트래픽 서비스의

포트 번호로 서비스명을 알기 위하여 IANA에서 제시된 포트 사용 번호를 먼저 점검하였다. IANA 기관에서 제시되거나 경험적으로 알고 있는 정의된 서비스의 포트명이면 응용 서비스를 PC에 다운로딩하고 인터넷 환경에서 수행한다. 아울러 트래픽 분석기를 활용하 패킷을 캡쳐하여 그 특성을 분석하였다. 또, 추출된 포트 중 어느 서비스인지도 모르면 tcpdump를 수행하며, PDU 내부의 원시 데이타를 추적하여 그 서비스 명을 파악하였다. 서비스 명을 찾은 다음 그 서비스를 인터넷 환경에 접속된 PC에 다운로딩하고 이의 수행과 더불어 패킷을 캡쳐하여 서비스의 특성을 분석하였다. 또한 제시된 응용 서비스의 동작과정을 분석 한 후 분석된 서비스의 면밀한 통계를 위해 원시 데이터 트래픽을 수집하는 에이전트와 에이전트로부터 전달받은 트래픽에 분석된 동작과정을 기반으로 BNF 기법을 서버에 적용하는 과정을 제안하였다. 또한 제안된 과정을 안동대학교 네트워크 환경에 적용하여 인터넷 트래픽 서비스 유형과 응용 서비스의 면밀한 통계 결과를 제시하여 트래픽 분석자에게 매우 유용한 정보를 제공하였다.

향후 현재의 분석 기법으로는 서비스명의 포트 번호를 알고 있으면 분석이 용이하며, 서비스 회사나 기관에서 동일한 서비스의 포트 명을 바꾸면 그 포트의 서비스를 자동으로 추적하는 기능이 필요하다.

참고문헌

[1] "Sniffer_Pro Protocol Analyzer User manual", http://www.snifferpro.com.

[2] "PA100 Protocol Analyzer User manual", C&C 인스투루먼트, 2000, http://www.cncinst.com

[3] "EtherealProtocol Analyzer User manual", http://www.ethereal.com

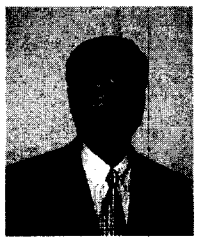
[4] "rfc 1700: Assigned Number", http://www.iana.com

[5] http://www.caida.org

[6] http://www.tcpdump.org

[7] "rfc2234: Augmented BNF for Syntax Specification: ABNF"

저자소개



정태수(Tae-Soo Jeong)

1981년 2월: 경북대학교 전자공학과 학사
1983년 2월 : 경북대학교 대학원 전자공학과 석사
1983년 3월 ~ 현재 : 한국전자통신연구원(ETRI) 책임연구원, IP 네트워킹기술팀장
※관심 분야: 인터넷 트래픽 측정 및 제어, BcN, 트래픽 엔지니어링, QoS 제어책임연구원, 인터넷 트래픽제어팀 팀장

최진섭(Jin-Sub Choi)

2004년 8월: 안동대학교 정보통신학과 학사
2004.6-현재: 한국전력 근무

정중수(Joong-Soo Chung)

1981년 2월: 영남대학교 전자공학과 학사
1983년 2월 : 연세대학교 대학원 전자공학과 석사
1993년 8월 : 연세대학교 대학원 전자공학과 박사
1983년 3월 ~ 1994년 2월 : 한국전자통신연구원(ETRI) 선임연구원
1994년 3월 ~ 현재 : 안동대학교 전자정보산업학부 부교수
※관심 분야: 인터넷 트래픽 제어, 무선통신망

김정태(Jung-Tae Kim)

1989년 2월: 영남대학교 전자공학과 학사
1991년 8월 : 연세대학교 대학원 전자공학과 석사
2001년 8월 : 연세대학교 대학원 전자공학과 박사
1991년 8월 ~ 1996년 2월 : 한국전자통신연구원(ETRI) 선임연구원
2002년 10월 ~ 현재 : 목원대학교 전자정보보호공학부 교수
※관심분야 : Microwave photonics, Optically fed wireless communication system design, Information security system design, Network Security, ASIC Design.

김대영(Dae-Young Kim)

1975년 2월: 서울대학교 전자공학과 학사
1977년 2월 : KAIST 전기 및 전자공학과 석사
1983년 2월 : KAIST 전기 및 전자공학과 박사
1983년 5월 ~ 현재 : 충남대학교 정보통신공학부 교수
※관심 분야: 인터넷 트래픽 제어, 무선통신망