MEASUREMENT AND ANALYSIS OF IP NETWORK TRAFFIC

Cen Zhiwei Gao Chuanshan Cong Suo Han Liangxiu
Dept. of Computer Science, Fudan University, Shanghai 200433, China
zhiwei_cen@yahoo.com

ABSTRACT

Traffic measurement and analysis are crucial to the design, operation and maintenance of wide-area networks based on Internet Protocol. This paper has outlined some important issues in real time traffic measurement and analysis. Firstly we give a brief survey on the related work of Internet traffic measurement, and then we focus on the general measurement methodologies. At last we present several visualizing results of our analysis upon some TCP traffic data.

KEYWORDS

Measurement, Performance Analysis, Discrete Wavelet Transform

1 INTRODUCTION

Ever since the Internet has been popular, both the transmission speed and the response time remain a crucial problem. The research of the infrastructure of wide-area internet is far from the requirement of the rapid, tremendous, and ceaseless growth of the network. However, the knowledge about the network performance is important for us to forge a deep insight of the network and thus help us to develop more sophisticated network operations and protocols.

Attempts to adequately track and monitor the Internet were greatly diminished in early 1995 when the NSF (National Science Foundation) relinquished its stewardship role over the Internet. As a result of this, current Internet industry lacks the ability to evaluate trends, identify performance problems beyond the boundary of a single ISP, or prepare systemically for the growing expectations of its users.

Now the situation is changing. Spawned mainly by end users with an interest in verifying performance of their Internet service, numerous independent activities in the area of network performance measurement are emerging. Today, there are a wealth of work dedicated to the realm of network performance measurement and analysis. The most related include NIMI [3], IPMA project [4], NAE workshop [5], NAI [6], Surveyor [7], San-qi Li's SMAQ tool [8] and Vern Paxson's thesis work [9].

In the analyzing stage, traffic models play a significant role. Accurate models enhance our understanding of the complicated network characteristics and behaviors and allow us to study the effect of various model parameters on network performance through simulation [2]. There are many tools to find these models, among which discrete wavelet transform is an effective one.

"The key issue in the design or selection of a congestion management scheme is the traffic pattern, and traffic patterns are dependent upon the application [1]." In this paper we focus first on the general measurement methodologies, then present several visualizing results of our analysis upon some TCP traffic data. The TCP traces we use are collected from Fudan University Campus Network, as well as some other universities and institutions. We also take the traces of other researchers as our examples.

The rest of this paper is organized as follows: In section 2 we outline the general method to measure the network. Section 3 presents our analysis results of the specific traffic using discrete wavelet transform. Section 4 concludes the paper and states some future work.

2 MEASUREMENT METHODOLOGY

Getting traffic data of real wide-area network is important to the research. Only through systematic collected data can we draw any reasonable conclusion about network behavior. Generally, there are several ways to collect usable data in real networks. Each has its advantages and disadvantages in different circumstances.

(1) Server Logs

Web servers can be configured to record information about all client requests. For example, most web servers have a log file called *access*, in which all the requests and responses of the server are recorded. Each line of the access log contains information on a single request for a document. The log entry for a normal request can be of the form:

hostname – [dd/mm/yyyy:hh:mm:ss tz] request status bytes

From each log entry, it is possible to determine the name of the host machine making the request, the time that the request was made, and the name of the requested document. The entry also provides some information about the server's response to this request, such as whether the server was able to satisfy the request (if not, a reason why the response was unsuccessful is given) and the number of bytes transmitted by the server, if any bytes were transmitted.

An example of a line from an access log is:

cs.fudan.edu.cn – [15/Aug/1999:14:50:05 - 0600] "GET/HTTP/1.1" 200 1200

This request came from the host cs.fudan.edu.cn at 14:50:05 on August 15, 1999. The requested document was the home page ("/") of the Web server. The status code 200 means that the request was successfully completed by the server, and 1,200 bytes were transferred from the server to cs.fudan.edu.cn.

Then we can use the data from the access logs to characterize web traffic.

(2) Passive Monitoring

Another way to gather the traffic data is to choose an appropriate site and passively capture every IP packet through it. It is important to choose a good monitoring point. The ideal site for obtaining these traces would be one through which many TCP connections pass, and it will be better if these connections are between a wide variety of Internet hosts. Thus during a connection there would be many data bytes for transmission.

Then the best point would be a major gateway site, which makes long TCP connections with sites all over the Internet. However, it is often hard to gain access to such sites. In this paper, we have selected the Campus Network Center of Fudan University. We monitored the site for about two hours and 200 million TCP packages were caught. This result is fairly passable.

There are many monitoring platforms. Among them we choose the *TcpDump* based *WinDump*. TcpDump is a UNIX tool coded by Van Jacobson, Craig Leres and Steven McCanne, all of the Lawrence Berkley National Laboratory, University of Californian, Berkley. TcpDump will print out the headers of packets on a network interface that match the boolean expression given by the user. WinDump is an equivalent on Win32 platform and is extended to adapt to the operating system. Users can build complicated boolean expressions to decide which data packets to catch. The expression consists of one or more primitives. Primitives usually consist of an identifier preceded by one or more qualifiers. There are three different kinds of qualifier:

type qualifiers say what kind of thing the identifier name or number refers to. Possible types are *host, net* and *port*.

dir qualifiers specify a particular transfer direction to and/or from the identifier. Possible directions are *src, dst, src or dst* and *src and dst*.

proto qualifiers restrict the match to a particular protocol. Possible protos are: *ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp* and *udp*.

The output of WinDump is protocol dependent. For example, the general format of a TCP protocol line is:

src > dst: flags data-seqno ack window urgent options

src and *dst* are the source and destination IP addresses and ports. *Flags* are some combinations of S (SYN), F (FIN), P (PUSH), or R (RST) or a single "." (no flags). *data-seqno* describes the portion of sequence space covered by the data in this packet. *Ack* is sequence number of the next data expected by the other direction on this connection. *Window* is the number of bytes of receive buffer space available in the other direction on this connection. *Urg* indicates there is "urgent" data in the packet. *Options* are TCP options.

(3) Active Probe

Although it is easy to perform passive monitoring, it has some obvious deficiencies. First, passive monitoring techniques can miss packets if the throughput of the monitoring system is insufficient, or if there is not enough buffering in the monitor to handle the occasional burst of packets. Another problem with passive monitoring is that it occasionally accepts packets with checksum errors. While it is theoretically possible to have the tracing program check the header checksums on each packet, in practice this would slow down the program and thus make it lose many packets.

While passive measurements provide views of the performance of individual links or nodes, active measurements provide complementary views of the performance of a path consisting of several links and nodes. This measurement data is used for traffic engineering, performance debugging, network operations, and to measure compliance with performance targets.

3 RESULTS VISUALIZATION

Once the raw data have been collected, one needs some way to tidy them up. Since the trace traffic often has tremendous size, they are usually reduced or stored in data warehouses. Besides these, we can also make some pretreatment to the data we have got.

In [12], the Web server access log is classified according to possible responses to client requests. There are four categories: *Successful, Not Modified, Found*, and *Unsuccessful*. Since the *Successful* responses are responsible for the majority of the data bytes transferred

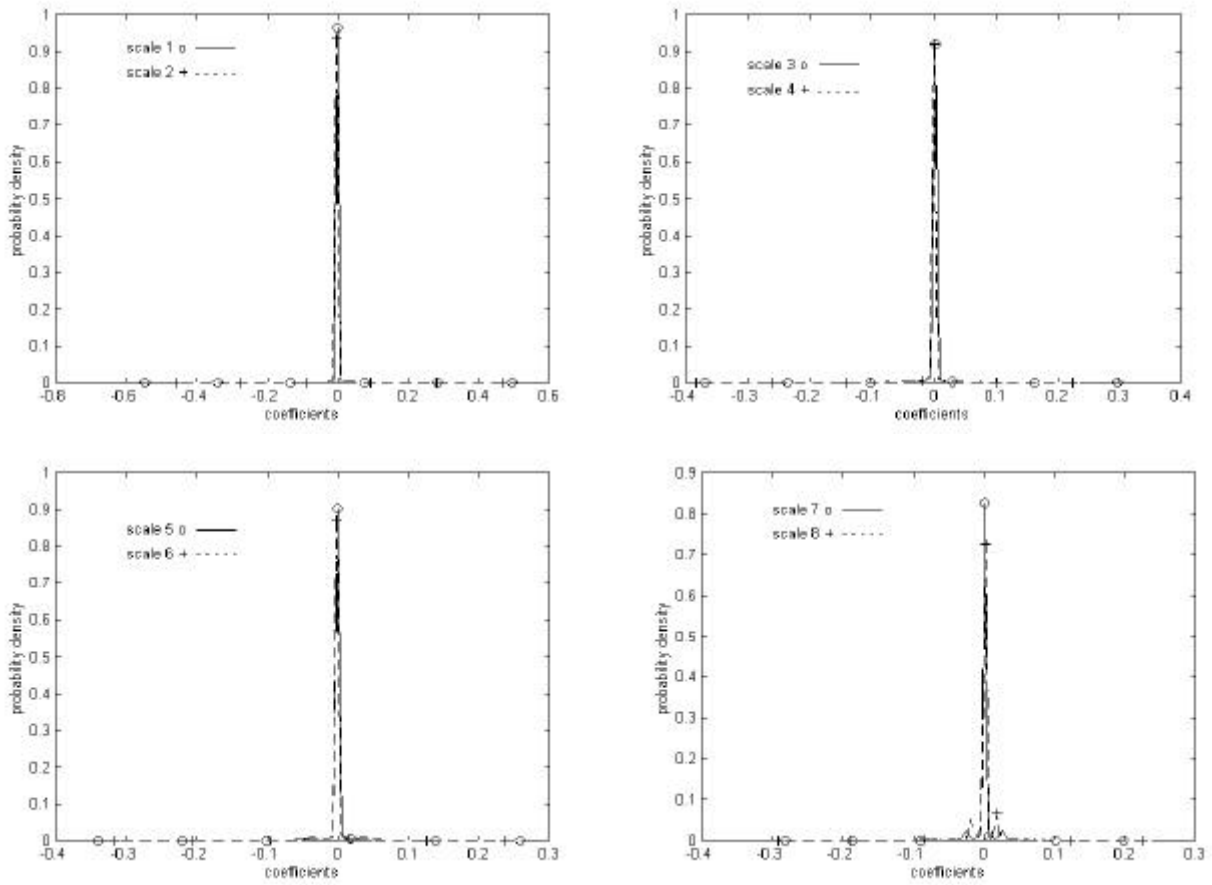

Fig. 3.1 The pdf of wavelet coefficients of fd-pkt-1

by the server, we can only use these responses for analyses, while other responses can be ignored.

With the traffic data collected by *WinDump*, we perform the following sanitizing process: (1) Map the IP address into virtual ones in order to shield the sensitive information of the inspected object; (2) Divide the raw data into five categories. They are TCP data segments, SYN/FIN/RST segments of TCP, UDP segments, IP encapsulation packets, and other kinds of packets. Since our objective concerns mainly TCP traffic, the time stamp, source IP address (sanitized), destination IP address (sanitized), source TCP port, destination TCP ports and segment length (in bytes, zero for pure ACK segments). For other data only time stamps are recorded.

In this section, we introduce Haar Wavelet at first, which is the mathematical technique in our traffic modeling process, and then we explain the traffic analysis process in steps. Finally, the graphs of the results are presented.

(1) Haar Wavelet

The fundamental idea behind *wavelets* is to analyze according to *scale*. Wavelet algorithms process data at different scales (resolutions). The wavelet analysis procedure is to adopt a wavelet prototype function, which is called *mother wavelet* or *analyzing wavelet*. Temporal analysis is performed with a contracted, high frequency version of the prototype wavelet, while frequency analysis is performed with a dilated, low frequency version of the same wavelet. Since the original signal can be represented in terms of a wavelet expansion, data operations can be performed on the corresponding wavelet coefficients.

The inherent scaling property of the wavelets to "localize" a signal in both time and scale is well suited for analyzing multi-fractal processes. Wavelet analysis provides an effective tool that studies the scale dependent properties of data directly via the coefficients of scale-time wavelet decomposition ([10, 11]).

The *discrete wavelet transform* (DWT) can represent signal X (t) in terms of shifted and dilated versions of a prototype band pass wavelet function ψ (t) and shifted version of a low pass scaling function ϕ (t). Based on the theory of *multi-resolution analysis* (MRA), the information in X (t) can be rewritten as a collection of details at different resolutions and a low-resolution approximation:

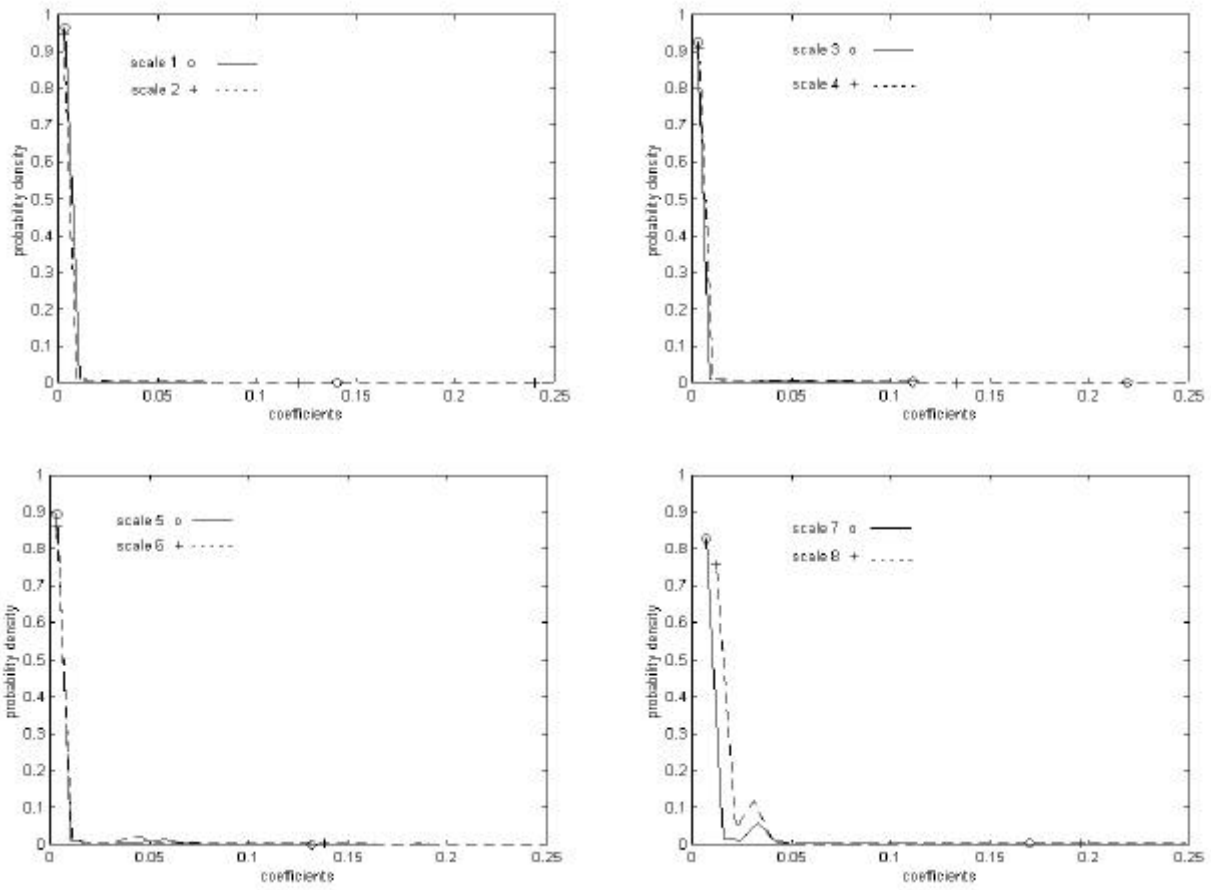

Fig. 3.2 The pdf of scaling coefficient of fd-pkt-1

$$X(t) = approx_J(t) + \sum_{j=1}^{J} detail_j(t)$$

$$= \sum_k a_X^J(k)\boldsymbol{f}_{J,k}(t) + \sum_{j=1}^{J}\sum_k d_X^j(k)\boldsymbol{y}_{j,k}(t) \quad (3\text{-}1)$$

These coefficients are defined through inner products of X with two sets of functions (scaling functions and wavelet functions):

$$a_X^j(k) = < X, \overset{\circ}{\boldsymbol{f}}_{j,k} >$$

$$d_X^j(k) = < X, \overset{\circ}{\boldsymbol{y}}_{j,k} > \quad (3\text{-}2)$$

Intuitively (from (3-1)), the DWT divides a signal into different frequency components and provides a way to analyze each component with a resolution matched to its scale. The inherent scaling localization ability of wavelets makes them an ideal mathematical tool for investigation the global and local scaling behavior.

In this paper, we will adopt the Haar wavelet transform to analyze the web traffic. For the Haar wavelet, the prototype scaling and wavelet functions are given by:

$$\boldsymbol{f}(t) = \begin{cases} 1, & 0 \le t < 1 \\ 0, & else \end{cases}$$

$$\boldsymbol{y}(t) = \begin{cases} 1, & 0 \le t < 1/2 \\ -1, & 1/2 \le t < 1 \\ 0, & else \end{cases} \quad (3\text{-}3)$$

Then the scaling and wavelet transform coefficients can be computed using:

$$a_X^j(2k-1) = 2^{-1/2}(a_X^{j-1}(k) + d_X^{j-1}(k))$$

$$a_X^j(2k) = 2^{-1/2}(a_X^{j-1}(k) - d_X^{j-1}(k)) \quad (3\text{-}4)$$

A simple constraint to guarantee that the process is positive can be derived from the above functions: $|d^j_X(k)| < a^j_X(k)$.

(2) Traffic analysis process

We apply the Haar wavelet technique to analysis the inter-arrival time and packet length of actual Internet

traffic.

Let discrete process $\{X^{(N)}(k)\}$ denotes the traffic process (packet inter-arrival time). We make it directly relate to the finest scale scaling coefficients: $\{X^{(N)}(k)\} = \{a^N_X(k)\}$ instead of multiplying a factor $2^{-n/2}$.

To characterize the parameters, we perform the following fine-to-coarse decomposition:

1) Begin with the finest scale j=N-1. Compute the Haar DWT of $\{ a^{j+1}_X(k)\}$ to generate $\{ a^j_X(k) \}$ and $\{ d^j_X(k) \}$; Meanwhile, analyze the probability density function (pdf) of random variable $\{ d^j_X(k) \}$.

2) Iterate step 1 until the coarsest scale j=0 is reached.

(3) Examining the Results

The traffic data presented here are called *fd-pkt-1*. It was collected from the campus Network Information Center of Fudan University. The WWW server and BBS server of the campus are responsible for most of the traffic. The traffic is from 11:17 to 12:48 on April 25, 2000. Of the 1,866,561 data packets, 1,793,558 are TCP segments. The maximum arrival time interval is 1070.172 ms. The minimum is 0.006 ms. The average is 3.068 ms. The standard deviation is 17.756 ms.

The maximal size of the TCP segments is 1460 bytes and the minimum is 0 bytes. The average is 606 bytes. The standard deviation is 579 bytes.

The pdfs of wavelet coefficients at different scales are shown in Fig. 3.1. It shows that this traffic exhibits an underlying conservative cascade construction ([10, 11]) by the DWT analysis. In different scales, the pdfs of wavelet coefficients also can be described as a truncated normal distribution with zero mean.

The pdfs of scaling coefficient of fd-pkt-1 are shown in Fig. 3.2, from which we can see that the pdf of scaling coefficient of fd-pkt-1 presents the nature of Weibull distribution at different scales.

4 CONCLUSIONS

In this paper we have outlined the important issues in real time traffic measurement. There has been a considerable development in the measurement approaches and we have made a summary of them. Then we introduce three categories of network measurement: server logs, passive monitoring, and active probe. Finally, we adopt Discrete Wavelet Transform as a useful tool to analyze the traffic data and get some visualized results. Future work includes improving the measurement methods to present more reliable and representative traffic data, and explain the visualized data in order to fit them into proper traffic models.

Reference

[1] R. Jain, Myths about Congestion Management in High-Speed Networks, DEC-TR-724, Oct '90.

[2] V. J. Ribeiro, R. H. Riedi, M. S. Crouse, and R. G. Baraniuk, Simulation of nonGaussian Long-Range-Dependent Traffic using Wavelets, SIGMETRICS' 99, Atlanta, Georgia, USA.

[3] Vern Paxson, Jamshid Mahdavi, Andrew Adams, and Matt mathis, An Architecture for Large-Scale Internet Measurement, IEEE Communications Magazine, August 1998.

[4] C. Labovitz, et al., The Internet Performance Measurement and Analysis Project, http://www.merit.edu/ipma.

[5] Kc Claffy, Internet measurement and data analysis: topology, workload, performance and routing statistics, NAE' 99 workshop.

[6] Towards a systemic understanding of the Internet organism: a framework for the creation of a network analysis Infrastructure, 1998, http://moat.nlanr.net/NAI.

[7] Sunil Kalidindi, Matthew J. Zekauskas, Surveyor: An Infrastructure for Internet Performance Measurements, June 1999, http://telesto.advanced.org/~kalidindi/papers/INET/inet99.html.

[8] San-qi Li, Sangkyu Park, and Dogu Arifler, SMAQ: A Measurement-Based Tool for Traffic Modeling and Queuing Analysis, IEEE Communications Magazine, August 1998.

[9] Vern Paxson, End-to-End Routing Behavior in the Internet, In Proceedings of ACM SIGCOMM' 96, August 1996.

[10] Gao Chuanshan, Cong Suo, Wu Gansha, Traffic Measurement, Simulation and Analysis, FTF' 99.

[11] Gao Chuanshan, Cong Suo, Wu Gansha, Measurement-Based Multifractal Traffic Modeling, Accepted at the ISCA 13th International Conference on Parallel and Distributed Computing Systems, August 2000.

[12] Martin F. Arlitt, Carey L. Williamson, Web Server Workload Characterization: The Search for Invariants, SIGMETRICS '96 5/96 PA, USA.