# Interval Weighted Load Balancing Method for Multiple Application Gateway Firewalls

B. K. Woo[1], D. S. Kim[1], S. S. Hong[1], K. H. Kim[1], and T. M. Chung[1]

[1]Real-Time Systems Laboratory,
School of Electrical and Computer Engineering,
SungKyunKwan University,
Chon-chon dong 300, Chang-an gu, Suwon, Kyung-ki do,
Republic of Korea
{bkwoo, dskim, sshong, byraven, tmchung}@rtlab.skku.ac.kr

**Abstract.** Firewalls are installed at network perimeters to secure organization's network as alternatives to general gateways. Because of potential performance problems on the gateways, load balancing technique has been applied. However, compared to general gateways, firewalls require more intelligent load balancing method to process massive network traffic because of their relatively complex operations. In this paper, we analyze the inherent problems of existing simple load balancing methods for firewalls and propose the interval weighted load balancing (IWLB) to enhance the processing of massive network traffics. The IWLB deals with network traffics in consideration of the characteristics of application protocols to achieve more effective load balancing. We observed that the IWLB outperforms other simple load balancing methods during our simulation. Therefore, we expect that the IWLB is suitable to balancing loads for multiple firewalls at a network perimeter.

## 1 Introduction

While the explosive growth of the Internet made it possible to exchange massive information, it caused some negative effects such as the increase of network traffics and security threats. Thus, organizations have a burden to deal with massive network traffic and protect their network from any malicious security threats.

The typical solution to counteract various security threats is deploying the firewall which applies policy-based access control to network traffic at a network perimeter [5, 6, 7]. However, a single firewall cannot operate properly when massive network traffics are applied but also become the dangerous security hole of networks [8]. Furthermore, as shown from DDoS attacks, such as the Trinoo, the Nimda worm, and so forth, using massive packet deliveries or service requests, these attacks degrade network services and performance and create serious operational failures, as well [12, 13].

To make firewall more robust and stable with massive network traffics, it is inevitable to install multiple firewalls to distribute network traffics by using various load

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2331, pp. 1051–1060, 2002.

balancing methods. Thus, the performance and fairness of a load balancer is the critical factor to estimate the network performance in the environment above.

Although there are many existing load balancing methods based on different principles such as Round robin, Hashing, and Randomization, most of them have flaws to be deployed for firewalls because they do not consider the characteristics of application protocols as the parameter for load balancing.

In this paper, we propose the interval weighted load balancing (IWLB) method, which is designed to use the characteristics of application protocols as the parameter for load balancing decision, to overcome the critical problems of existing load balancing methods. Furthermore, the IWLB is able to enhance network performance and give more robustness in the environment that multiple firewalls are inevitable. With the analysis of simulation results, we observed that the IWLB outperforms other load balancing methods when it is deployed as the principle for the load balancing to distribute traffics.

This paper is organized as follows. In chapter 2, we issue the potential problems of existing load balancing methods when they are applied for multiple firewalls. In chapter 3, we introduce the principle and mechanism of the IWLB method. Our simulation model is described in chapter 4, and the analysis of simulation results and the comparison to other load balancing methods are presented in chapter 5. At last, we close this paper with conclusions in chapter 6.

## 2 Simple Load Balancing Methods

Under the environment that multiple firewalls are installed in a network for the load balancing purpose, it is desirable to build an appropriate load balancing principle to guarantee the performance of the individual firewall and network security. The most important factor for successful load balancing is to distribute service requests fairly to multiple firewalls.

However, the fairness must be defined differently in this environment because a firewall could discard incoming service requests in terms of its access control policy. That is, even though the load balancer of multiple firewalls distributes incoming service requests fairly, the active session distribution on multiple firewalls can be distributed unevenly. The existing load balancing methods show the limit to overcome this problem because they do not regard the characteristics of application protocols as the parameter for fair load balancing.

In the following sections, we issue the critical flaws of most widely used load balancing methods when they are deployed as the load balancer of multiple firewalls.

**Round Robin.** Round robin method distributes incoming service requests by simply allocating them to the next available firewall in the rotational manner. That is, this method does not consider the load or number of active sessions currently allocated to the individual firewall. Thus, it can cause the potential load imbalance and increase the skew when a specific application requires long service time [1].

**Hashing.** Hashing method distributes incoming service requests by hashing the information of service request packets such as source address, source port number, and so on [1]. However, when considering that the flooding attacks generally occur with the identical source information, Hashing method can be vulnerable to those attacks. Moreover, when deployed Hashing mechanism is exposed to attackers, the overall network can be plunged into the fatal situation. We strongly suggest not using Hashing method for the load balancer of multiple firewalls.

**Randomization.** Randomization method distributes requests to each node according to the value of pseudo random number [1]. In this method, the fine algorithm for random number generation is the key to the successful load balancing. Like other methods mentioned above, Randomization method does not consider the characteristics of application protocols or sessions. Therefore, it is hardly expected that this method is suitable for the load balancer of multiple firewalls.

## 3 The Interval Weighted Load Balancing

As mentioned in chapter 2, simple load balancing methods are not suitable for the fair load balancing of multiple application gateway firewalls, because they do not take the characteristics of application protocols in consideration. Thus, it is necessary to consider the characteristics of application protocols or sessions as the parameter in order to design more stable and efficient load balancer.

The proposed load balancing method named the interval weighted load balancing (IWLB) makes use of the weight value allocated to each application protocol. In the IWLB, the weight of an application protocol is defined as the interval in the order of firewalls in order to decide which firewall will process a current incoming request. That is, the IWLB decides the firewall for the current service request by adding the weight of the application protocol to the order of the previously selected firewall for the previous service request that has the same protocol as the current one. Since the initial value of the weight, based on the former research about the traffic pattern of application protocols [14], is assigned to each application protocol, the IWLB keep track of the weight value by calculating it periodically in a statistical manner.

To give a specific example how the IWLB obtains the weight value of each protocol, let's suppose the following situation. There are 8 firewalls and all of them can serve 4 different application protocols: HTTP, FTP, SMTP, and TELNET. And let's the weight of HTTP assigned by the IWLB is 3 at this moment. If 6 HTTP requests arrived at the IWLB load balancer sequentially, the order of firewalls to service these requests will be $Fw_1 \rightarrow Fw_4 \rightarrow Fw_7 \rightarrow Fw_2 \rightarrow Fw_5 \rightarrow Fw_8$. That is, the IWLB decides the firewall to service these requests by adding the weight value of HTTP to the previously selected firewall in a rotational manner. Fig. 1 depicts this example and difference from the conventional round robin method.

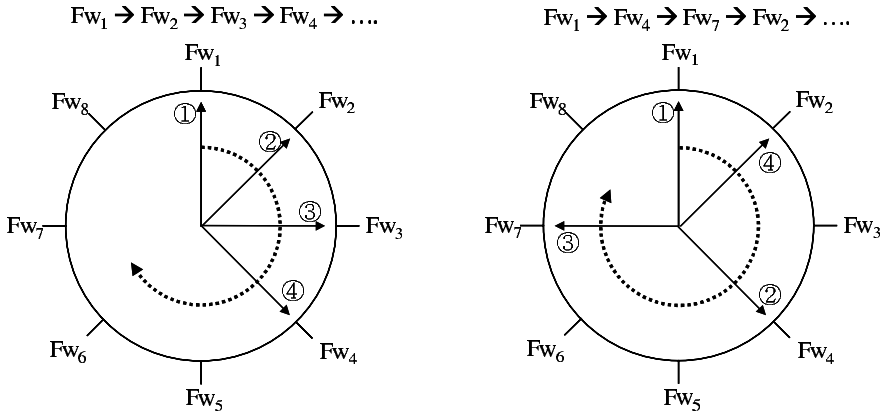


**Fig. 1.** The comparison of the selection sequence for Round robin and the IWLB

The IWLB uses the standard weight value to calculate the weight value of each application protocol, the standard weight value is set to 1 and assigned to the application protocol that has the largest average service time. After deciding the standard weight value, the weight values of other application protocols are decided as the ratio of their average service time to the average service time of the application protocol that has the standard weight value. Finally, the calculated ratio values must be rounded to a nearest integer value to be used as the weight values for application protocols.

Table 1 explains the rule that the IWLB generates the weight values of application protocols.

**Table 1.** The average service time and weight of application protocols

| Application protocol | Average service time | Calculation rule | Weight |
|---|---|---|---|
| HTTP | 970 | $W_{http} = 3090/970 \fallingdotseq 3.18$ | 3 |
| FTP | 3090 | $W_{ftp} = 1$ (standard weight) | 1 |
| SMTP | 790 | $W_{smtp} = 3090/790 \fallingdotseq 3.91$ | 4 |
| TELNET | 430 | $W_{telne} = 3090/430 \fallingdotseq 7.18$ | 7 |

In Fig. 2, we depict how the IWLB balances the service requests with the example of Table 1. Load Distribution of each application protocol starts at the first firewall, i.e., $Fw_1$.

On the other hand, if the total number of firewalls is divisible by the weight value of a certain protocol, all of the incoming requests of the protocol would be assigned to the same firewall. To prevent this phenomenon, the weight for each application protocol must be the prime to the total number of firewalls.

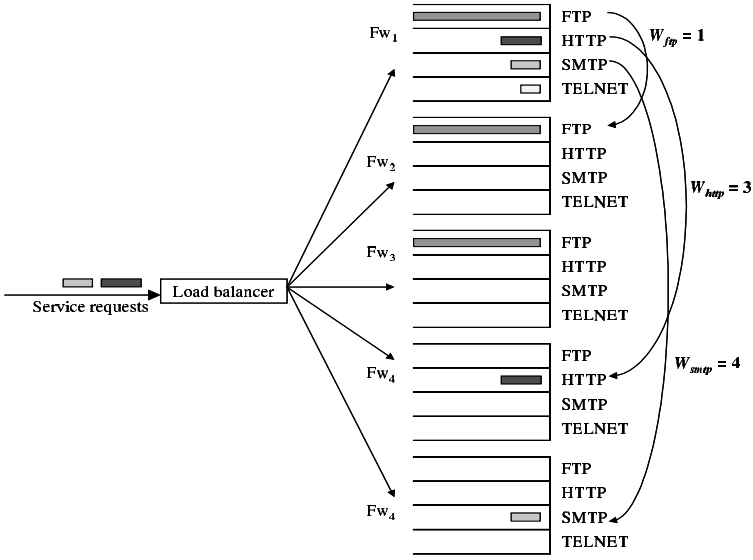


**Fig. 2.** The distribution of incoming requests using the IWLB

## 4 Modeling and Simulation

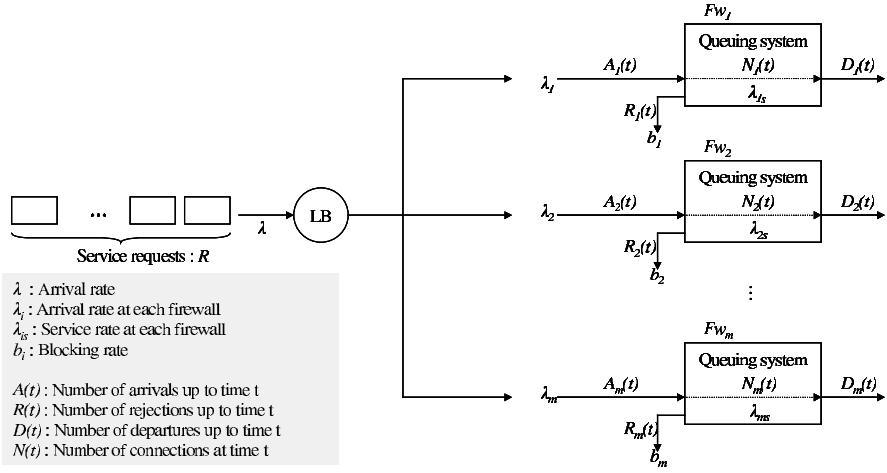


**Fig. 3.** The modeled load balancer and multiple firewalls

It is widely known that TCP-based application protocols occupy the large portion of the Internet traffic. McCreary announced that some TCP-based application protocols such as HTTP, FTP, SMTP, and TELNET occupy 42.52%, 2.59, 1.70%, and 0.13%,

of the overall Internet traffic, respectively in his recent research [14]. From this fact, we can deduce easily that the performance of a proxy firewall is directly influenced by TCP-based application protocols.

To build our simulation model, we chose 4 representative TCP-based application protocols: HTTP, FTP, SMTP, and TELNET. In our model, the load balancer distributes service requests to a group of firewalls modeled by the queuing system. To compare the performance of existing load balancing methods to that of the IWLB, we include 3 other load balancing methods, Round robin, Hashing, Randomization, in our simulation. Fig. 3 depicts the simulation model of the load balancer and multiple firewalls.

Let $R$ be a set of service requests, which is generated with the arrival rate $\lambda$, arrived at the load balancer of multiple firewalls. We suppose that the inter-arrival time between two adjacent service requests and service durations of requests are exponentially distributed with the arrival rate $\lambda$ and the service rate $\mu$, respectively. Additionally, each element of $R$ contains its application protocol for the IWLB.

After service requests are distributed to a group of firewalls by various load balancing methods, each firewall processes the allocated service requests. When we suppose that $m$ firewalls installed in our model, the summation of the service requests allocated to each firewall equals the $R$ if there is no blocking of requests at the load balancer. Therefore, as shown in the equation (1), the summation of the arrival rate at each firewall equals the arrival rate at the load balancer.

$$\lambda = \sum_{i=1}^{m} \lambda_i \tag{1}$$

The service request allocated to a firewall can be blocked in terms of its access policy. If we assume that the blocking rate, $b_i$, on service requests at $Fw_i$ denoted as the $i$th firewall, then the rate of service requests processed by $Fw_i$, $\lambda_{is}$ would be defined as the equation (2)

$$\lambda_{is} = \lambda_i(1 - b_i) \tag{2}$$

If service requests are blocked by the access policy of a firewall unpredictably, the load of firewalls will be distributed unevenly irrelevant of the fair distribution of service requests by the load balancer. Since no simple load balancing methods are able to cope with this situation, the fluctuation of workload between firewalls is inevitable. In the IWLB, if a service request is blocked, the firewall signals to a load balancer. When the load balancer receives the signal from the firewall, it allocates the next service request to the firewall once more to prevent the fluctuation of the load between firewalls.

We regarded each firewall as an M/M/c/c queuing system independently with the same capacity. The capacity of $i$th firewall, $C_i$, means the maximum number of active sessions that the firewall can handle with concurrently. If the number of active sessions exceeds the $C_i$, then the allocated service request to firewall will be queued for a later service.

## 5 Simulation Result and Analysis

In the simulation, we assume that the load balancer interacts with 4 firewalls. We generated 200,000 service requests to monitor the fair distribution of service requests and applied 3 simple load balancing methods and the IWLB to the load balancer in our model. During the simulation, we supposed that the propagation delays from the load balancer to firewalls are ignorable.

For the analysis of simulation results, we monitored the summation of service time and the waiting time of service requests in the queue at each firewall every second. If we analyze these values between firewalls, we would judge whether the applied load balancing methods distributed the service requests optimally to firewalls.

Firstly, we compared the response time of the IWLB to that of other simple load balancing methods. **Fig. 4** depicts the maximum response time of Hashing, Randomization, Round robin, and the IWLB, respectively. We sampled the response times of each firewall every 1 second and select the maximum response time among the sampled value. The graph shows that the maximum response time of the IWLB is remarkably lower than others. Additionally, the fluctuation of its curve is relatively narrower than that of other simple methods. Note that we put the results of each load balancing method together in a graph for comparison.

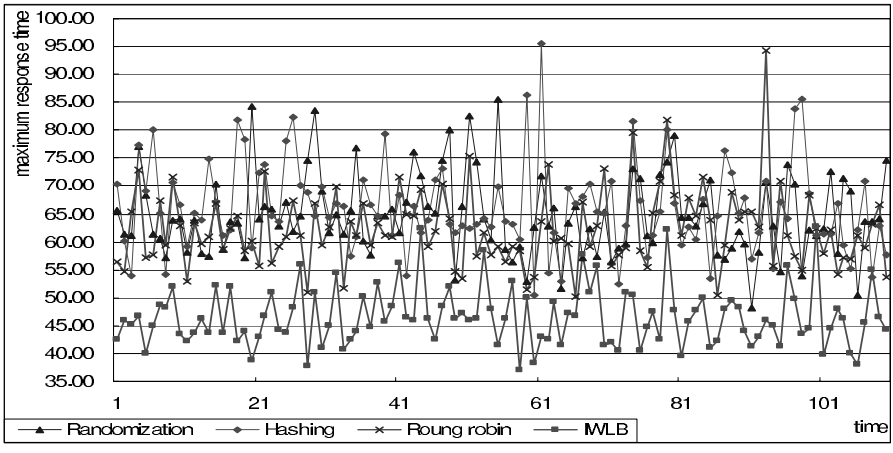


**Fig. 4.** Maximum response time

For more sophisticated comparison, we calculated the mean response time of each load balancing method and the mean response time of each method is depicted in **Fig. 5**. While the mean values of Round robin, Randomization, Hashing methods are not much different each other, those of the IWLB are quite different from them. The IWLB shows very lower mean response time during our simulation.

Now, from the two results of the IWLB, we can judge that the IWLB outperformed other simple load balancing methods.

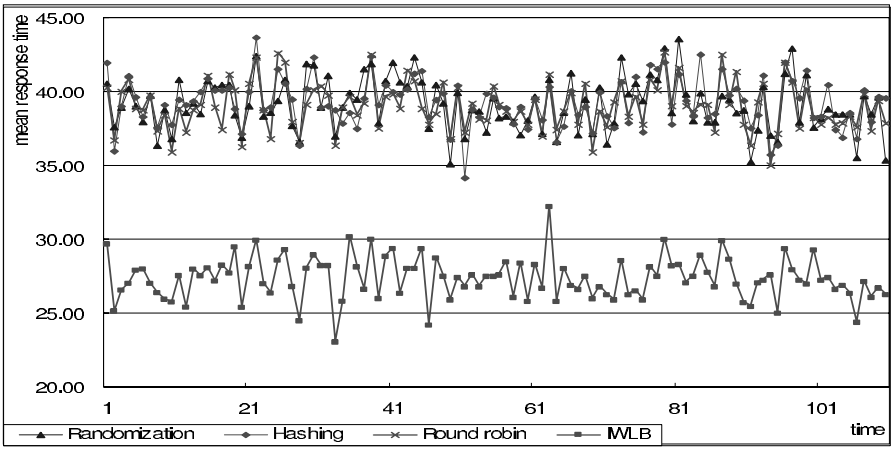


**Fig. 5.** Mean response time

In **Fig. 6**, we calculated the variance of the response time of each load balancing method. In this figure, we can see that the variance of the IWLB is lower than others, too. According this, we can conclude the response time of the IWLB is more stable than that of other simple methods.

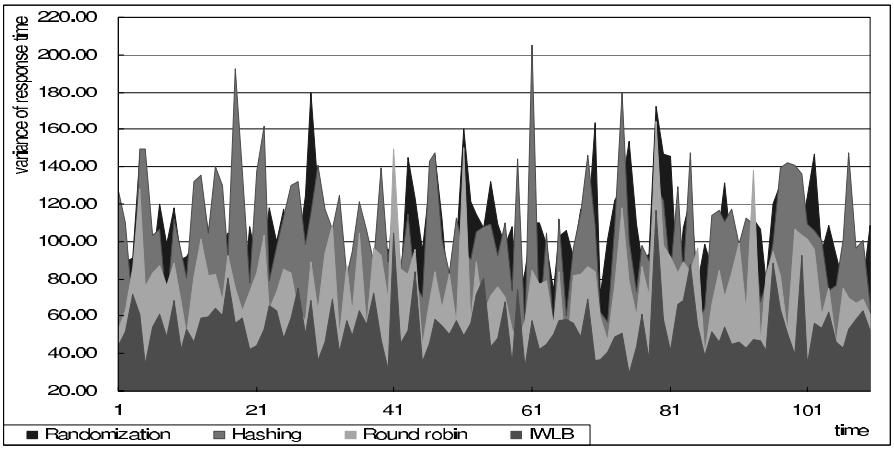


**Fig. 6.** The variance of response time

From the comparison, we can deduce two facts. One is that the distribution of service requests by simple load balancing methods causes the skewed load distribution among firewalls because they do not consider the blocking of requests in terms of the access policy of firewalls. The other is that the load balancing principle of the IWLB, considering the characteristics of application protocols, shows the positive effect for the fair traffic distribution.

As **Table 2** shows, we compared the mean and maximum buffer size of each firewall. In the case of the IWLB, the mean and maximum buffer sizes are greatly smaller

than those of other simple methods. That is, the IWLB balances loads most fairly among simulated methods and can reduce the memory resource of firewalls. Note that the mean values of buffer size are calculated to the second decimal place.

**Table 2.** The comparison of the buffer size of firewalls

| | Mean buffer size | Max buffer size |
|---|---|---|
| Round robin | 8.13 | 40 |
| Randomization | 8.21 | 35 |
| Hashing | 8.12 | 39 |
| IWLB | 0.14 | 15 |

During the analysis of simulation results, we observed that the IWLB outperformed other simple load balancing methods in many aspects. Moreover, the simulation results explain that the IWLB proved that it is able to cope with massive traffic loads adequately. Consequently, it is strongly required to consider the weight of application protocols for fair load balancing to counteract massive service requests. Furthermore, when the load balancer interacts with firewalls, it should be able to manage the request blocking by the firewall to prevent the fluctuation of workloads between them. We expect that the IWLB meets to these requirements successfully.

## 6 Conclusion and Further Studies

Although many organizations are deploying firewalls for the purpose of network security, it is doubtable that they can process massive network traffics without performance degradation. To make it worse, considering the trends of preferring an application gateway firewall that performs more sophisticated operations, it is obvious the performance degradation will be more serious for massive network traffic. Because the performance degradation or malfunction of firewalls implies the critical security flaws, it is strongly required to make the firewall more tolerate against massive network traffic.

Several researches paid attention to deploying multiple firewalls and load balancers to counteract massive network traffic. It was unsuccessful to manage them because of the inherent drawbacks of existing simple load balancing methods for firewalls.

In this paper, we proposed the enhanced load balancing method, the IWLB, to manage overloaded network traffic efficiently. Since the IWLB makes use of the weight values of application protocols, calculated by the statistical traffic pattern of application protocols, for the load distribution, it is optimized for the load balancing method for multiple application gateway firewalls.

As shown in our simulation results, we observed that the IWLB outperformed other simple load balancing methods on the load distribution of massive TCP-based application service requests. From these results, we expect that the IWLB would be suitable for the load balancer for the network that deploying multiple application gateway firewalls is inevitable.

At this moment, we are planning to extend our evaluation model to various application protocols and performance measurement to various aspects such as resource usage, packet loss, and so on. Additionally, we will evaluate the scalability and tolerance of the IWLB when some firewalls are not functioning properly.

## References

1. Rajkumar, B.: High Performance Cluster Computing: Architecture and Systems, Volume 1, Prentice Hall PTR, (1999)
2. Leon-Garcia, A.: Probability and Random Process for Electrical Engineering, 2nd Ed., Addison Wesley Publishing Company, Inc., (1994)
3. Molloy, K.M.: Fundamentals of Performance Modeling, Macmillan Publishing Company, (1989)
4. Law, M.A., Kelton, W.D.: Simulation Modeling & Analysis 2nd ed., McGraw-Hill Book Co., (1991)
5. Cheswick , R.W., Bellovin, M.S.: Firewalls and Internet Security : repelling the willy hacker, Addison Wesley, (1994)
6. Chapman, D.B., Zwicky, D.E.: Building Internet Firewalls, O Reilly & Associations, Inc., (1996)
7. Hare, C., Siyan, K.: Internet Firewalls and Network Security - 2nd ed., New Readers, (1996)
8. Kostic, C., Mancuso, M.: Firewall Performance Analysis Report, Computer Sciences Corporation, Secure Systems Center – Network Security Department, (1995)
9. Haeni, E. R.: Firewall Penetration Testing, The George Washington University, Cyberspace Policy Institute, (1997)
10 Test Final Report – Firewall Shootout Networkd+Interop, KeyLabs Inc., 28 May 1998.
11. Foundry ServerIron Firewall Load Balancing Guide, Foundry Networks, Inc., (2001)
12. Carnegie Mellon University, CCERT Advisory CA-2001-26 Nimda Worm, CERT/CC, http://www.cert.org/advisories/CA-2001-26.html, (2001)
13. Carnegie Mellon University, CERT Incident Note IN-99-07: Distributed Denial of Service Tools, CERT/CC, http://www.cert.org/incident_notes/IN-99-07.html, (1999)
14. McCreary, S., Claffy, K.: Trends in wide area IP traffic patterns - A view from Ames Internet Exchange, Proceedings of 13th ITC Specialist Seminar on Internet Traffic Measurement and Modeling, Monterey, CA. 18-20, (2000)