PAPER

Identification of Attack Nodes from Traffic Matrix Estimation

Yuichi OHSITA[†a)], **Shingo ATA**[††b)], ***Members, and*** **Masayuki MURATA**[†††c)], ***Fellow***

SUMMARY Distributed denial-of-service attacks on public servers have recently become more serious. The most effective way to prevent this type of traffic is to identify the attack nodes and detach (or block) attack nodes at their egress routers. However, existing traceback mechanisms are currently not widely used for several reasons, such as the necessity of replacement of many routers to support traceback capability, or difficulties in distinguishing between attacks and legitimate traffic. In this paper, we propose a new scheme that enables a traceback from a victim to the attack nodes. More specifically, we identify the egress routers that attack nodes are connecting to by estimating the traffic matrix between arbitral source-destination edge pairs. By monitoring the traffic variations obtained by the traffic matrix, we identify the edge routers that are forwarding the attack traffic, which have a sharp traffic increase to the victim. We also evaluate the effectiveness of our proposed scheme through simulation, and show that our method can identify attack sources accurately.

key words: *Distributed Denial of Service (DDoS), Traceback, Traffic matrix, Simple Network Management Protocol (SNMP)*

1. Introduction

The recent rapid growth and the increasing utility of the Internet are making security issues increasingly important. Denial-of-service (DoS) attacks are one of the most serious problems and must be resolved as soon as possible. These attacks prevent users from communicating with service providers and have damaged many major web sites all over the world.

The number of attacks has been increasing, and the techniques used to attack servers have become more complex. In the distributed denial-of-service (DDoS) attacks often seen recently, multiple distributed nodes attack a single server concurrently. A malicious user tries to hack the remote nodes by exploiting the vulnerabilities of the software running on them, installs an attack program on the hijacked nodes, and keeps them waiting for an order to attack a victim server. When the malicious user sends a signal to them, they begin to attack the same server. Even if the rate of attack for each node is small, the attack traffic can cause serious damage to the victim server when the number of hijacked nodes is large.

Identifying the attack sources is one of effective (and ideal) solutions to cut off the link to the attacker or filter attack packets by an edge router connected to the attacker. However, because attackers can easily spoof the source address fields of the attack packets, it is quite hard to identify the attack sources by only checking the source address of the attack packets.

For this reason, several methods for identifying the attack sources are proposed. In general, these methods are called *IP tracebacks*. One of them is proposed in [1], where a router generates an ICMP traceback packet to the destination of the packet with a low probability at the event of packet forwardings. The victim identifies the actual source of the packet by using the received ICMP traceback packets. Other methods are proposed in [2]–[4], in which a router marks IP packets to be forwarded with identification information instead of generating ICMP packets. The victim can identify the source of the packets using the identification information. Another method is proposed in [5], [6], where each router stores packet digests. The victim queries its upstream routers to see whether an attack packet has passed through them or not.

However, these methods have several problems. One of them is that they cannot work with legacy routers because they need router support. Another is that they may erroneously identify legitimate clients as attack sources. This is because these methods can only identify the source nodes of IP packets. Since there is no difference between legitimate and attack packets, identifying attack packets from the mixture of attack and legitimate traffic is difficult.

In DoS attacks, attackers send a large number of packets to exhaust the network resources. When an attack starts, there is a rapid increase in the traffic from the attack sources to the victim. Several methods use such increase in traffic in the network to detect attacks [7]–[10]. By using traffic volumes which can also be monitored by legacy routers, we identify edge routers connecting to the attackers without any change in core network. Then, deploying only edge routers supporting IP traceback, we identify attack nodes by using IP traceback from the identified edge routers. In addition, identification of the attack sources by monitoring the

[†]Graduate School of Economics, Osaka University, 1-7 Machikaneyama, Toyonaka, Osaka, 560-0043, Japan

[††]Graduate School of Engeneering, Osaka City University, 3-3-138, Sugimoto, Sumiyoshi, Osaka, 568-8585, Japan

[†††]Graduate School of Information Science and Technology, Osaka University, 1-5, Yamadaoka, Suita, Osaka, 565-0871, Japan

a) E-mail: y-ohsita@econ.osaka-u.ac.jp

b) E-mail: ata@info.eng.osaka-cu.ac.jp

c) E-mail: murata@ist.osaka-u.ac.jp

increased traffic can distinguish the attackers from the legitimate clients, which do not sharply increase traffic. However, there are only a few papers about identification of attack sources by monitoring the increase in traffic.

Lakhina et al propose a method for identifying the attack sources by monitoring the traffic on each link in the network [8]. In this method, the measured loads of all links are separated into normal and abnormal subspaces. The normal subspace indicates the time-of-day variation of the traffic. Other variations are categorized into the abnormal subspace. We then identify the attack source that explains the largest amount of anomalous subspace. Although this method can identify the attack source in a single attacker case, this method has difficulty in identifying multiple attack sources such as DDoS, because we need to consider all possible cases by changing the number of attackers. It requires a huge computation overhead.

The anomaly detection methods using traffic volumes between all source/destination pairs are also proposed. The traffic volumes between every pair of ingress and egress points are typically described as a traffic matrix. The method proposed in [10] uses the compact summary of the per-flow statistics and detect anomalies by comparing the difference between the actual summary and the forecasted summary obtained by the forecast models. Another method proposed in [9] separates the traffic matrix into normal and abnormal subspaces. Since this method separate traffic volumes between all source/destination pairs into normal and abnormal subspaces, we can identify traffic between source/destination pairs having large abnormal subspaces as attack traffic.

However, these methods assume that the traffic matrix can be monitored accurately. Though Cisco's NetFlow [11] can monitor the flow statistics and periodically export the monitored statistics to the central storage device, the process of NetFlow in routers consumes CPU time to identify flows of recieved packets. The perfomance analysis of NetFlow [12] says the resource consumption would increase according to the number of active flows passing the router. Especially, DDoS attack traffic contains many of spoofed packets which lead the large number of flows having a single packet. As a result, during DDoS attacks, the activation of NetFlow has possibility to consume very large amount of CPU time. Though [13] proposes the distributed method to monitor traffic data and separate them into normal and abnormal subspaces, this method needs router support and cannot work with legacy routers. On the contrary, the objective of our work in this paper is to estimate the edge-to-edge traffic matrix accurately under the assumption that we cannot monitor the amount of traffic for all edge pairs and therefore we only measure the load of network links.

Estimating traffic matrix from the measurement of link loads is also proposed in some literatures e.g., [14], however, existing traffic matrix estimation methods are not suitable to apply the identification of attacks because the assumption used in these methods may decrease the accuracy of estimation of traffic volumes including the attack traffic. We describe in detail in Subsection 2.1.

In this paper, we propose a new method for identifying attack sources by estimating the increase in traffic between each source and destination. In this method, we can estimate the increase in traffic accurately by focusing not on the total rate of traffic but on the increase in traffic. In addition, our method can work with existing routers because we can obtain link load data through Simple Network Management Protocol (SNMP).

In Section 2, we explain an overview of our proposed method. In Section 3 we evaluate our method. In Section 4 we conclude by briefly summarizing the paper and mentioning some of the future works we intend to do.

2. Identification of attack sources by estimating traffic matrix

Our method identifies attack sources by estimating the increases in traffic between every pair of sources and destinations. We estimate the increases in traffic from the monitored link load. In the estimation of the traffic matrix, we don't focus on the total amount of traffic, but only focus on the amount of increase from the previous measurement. The reason why we use only the increases in traffic for the traffic estimation is discussed in the next subsection. In this section, we first show a brief overview of our proposed scheme.

Fig. 1 shows an overview of our proposed method. In our method, we introduce a control node to perform the process of identification of attack sources. We call this node a *monitoring node*, and we also define the region where the monitoring node controls as a *monitored network*. The monitoring node identifies the attack sources by periodically performing the following operations.

1. Obtains the statistics of the link load data from all routers in the monitored network.
2. Estimates a matrix of the increase in traffic between all arbitrary pairs of edge routers in the monitored network.
3. Identifies the attack sources from the estimated traffic increase matrix.

We can obtain link load data through SNMP. SNMP is supported by essentially every device in IP networks and is used to monitor or manage the device. That is, our method can work with existing routers.

The interval for obtaining the statistics affects the time for identifying the attack sources. If we set the

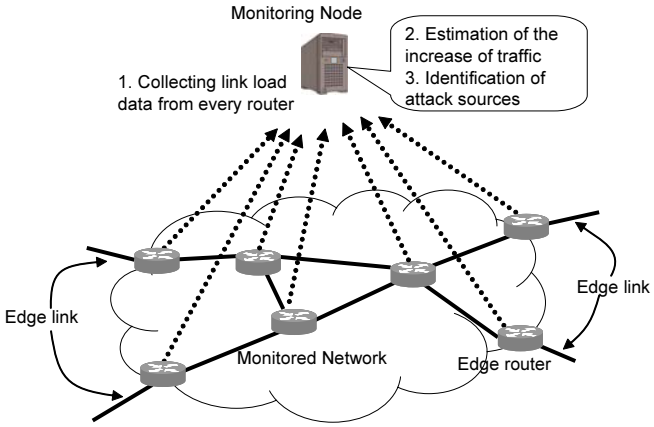


Fig. 1 Overview of proposed method

interval to a larger value, the identification takes more time. On the other hand, if we set the interval to a smaller value, the loads on the routers increase though we can identify attack the sources soon after the attack starts. Thus, we should set this interval to as small value as possible without high loads on routers. In general, to avoid high loads on routers, the interval of SNMP is set to 5 minutes. Therefore, we set this interval to 5 minutes in our evaluation described in Section 3.

In the following sections, we describe the details about how to estimate the increase in traffic and how to identify the attack sources.

2.1 Estimation of Increase in Traffic

First, we assign a set of links outside the monitored network as E. We call these links *edge links*. The router, which is connected by an edge link, is called the *edge router*. We assign a set of all the links in the monitored network, including the edge links, as L.

Traffic matrix T is defined as the $|E| \times |E|$ sized matrix, whose element $t_{i,j}$ $(i, j \in E)$ indicates the amount of traffic traversing from edge link i to edge link j. We can obtain the link loads from each router through SNMP. The link loads can be denoted by the $2|L|$-size link load matrix X as follows:

$$X = \begin{bmatrix} x_1^{\mathrm{f}} \\ x_1^{\mathrm{b}} \\ x_2^{\mathrm{f}} \\ x_2^{\mathrm{b}} \\ \vdots \\ x_{|L|}^{\mathrm{f}} \\ x_{|L|}^{\mathrm{b}} \end{bmatrix}. \tag{1}$$

In matrix X, elements x_l^{f} $(l \in L)$ and x_l^{b} $(l \in L)$ indicate the amount of traffic on link l in the forward and backward directions respectively, because most of the network links are bidirectional. We only use the words forward/backward to distinguish the direction of the link. Therefore, there is no policy for determining the forward or backward direction of each link. However, we must distinguish between the ingress and egress traffic. To distinguish between them, we denote the ingress traffic measured on edge link i as x_i^{in} $(i \in E)$ and egress traffic measured on the edge link j as x_j^{out} $(j \in E)$.

2.1.1 Traffic Matrix Estimation using *Gravity Model*

We estimate the traffic matrix of each pair of edge links from the link loads and routing information in monitored network. [14] uses a *gravity model* to estimate the traffic matrix. The gravity model assumes that traffic from a source to a destination is proportional to the total traffic at the source and at the destination. Using this model, we estimate the traffic matrix from

$$t_{i,j} = x_i^{\mathrm{in}} \frac{x_j^{\mathrm{out}}}{\sum_{\forall k \in E} x_k^{\mathrm{out}}} \quad (i, j \in E) \ , \tag{2}$$

where x_i^{in} is the element of X corresponding to the amount of ingress traffic to the monitored network measured on the edge link i and x_j^{out} is the egress traffic measured on the edge link j.

However, we cannot estimate increases in traffic accurately using Eq. (2) as follows. We assume that an attack traffic whose rate is t_{attack} traverses from i to j. We also assume legitimate traffic $t_{i,j}$ can be accurately estimated by Eq. (2). Traffic from i to j, including the attack traffic is estimated from

$$t'_{i,j} = (x_i^{\mathrm{in}} + t_{\mathrm{attack}}) \frac{x_j^{\mathrm{out}} + t_{\mathrm{attack}}}{\sum_k x_k^{\mathrm{out}} + t_{\mathrm{attack}}}, \tag{3}$$

where $t'_{i,j}$ is the traffic traversing from i to j including attack traffic. Then, the increased traffic by the attack is estimated by

$$t'_{i,j} - t_{i,j} = \frac{t_{\mathrm{attack}}^2 + t_{\mathrm{attack}}(x_i^{\mathrm{in}} + x_j^{\mathrm{out}})}{\sum_{k \in E} x_k^{\mathrm{out}} + t_{\mathrm{attack}}}, \tag{4}$$

where $t_{i,j}$ is the legitimate traffic from i to j. Fig. 2 shows a simple example. In this example, we assume the total rate of traffic in the monitored network is 6 GBytes/sec, both x_A^{in} and x_B^{out} are 1 GBytes/sec. We also assume the attack traffic from the edge link A to B has the rate of 1 GBytes/sec. From Eq. (4), the total traffic, including the attack traffic from edge link i to j is estimated as 0.55 GBytes/sec, which is quite different from the attack rate (1 GBytes/sec).

As previously mentioned, when attack traffic is injected, the estimated increase in traffic is proportional to the total rate of traffic monitored at the source. That is, the gravity model is infeasible for directly estimating the attack traffic because the impact of the attack traffic is distributed among the edge links that have legitimate traffic to the victim. As a result, the estimated attack rate is significantly lower than the rate of the attack traffic that is really generated.

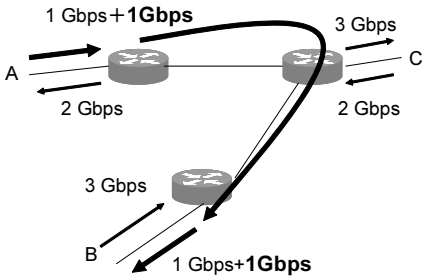


Fig. 2 Simple example of DoS attack

2.1.2 Traffic matrix estimation focusing on increased traffic

To accurately estimate the increase in traffic, we propose a matrix estimation method focusing not on the total rate of traffic but on the increase in traffic.

First, we calculate the increases in traffic on each link from

$$G_n = X_n - \bar{X}_n, \tag{5}$$

where G_n is the $2|L|$-size vector in which the elements $g^{\mathrm{f}}_{i,n}$ $(i \in L)$ and $g^{\mathrm{b}}_{i,n}$ $(i \in L)$ indicate the increase in traffic on link i in the forward and backward directions at time n, respectively. X_n is the link load vector at time n and $\bar{X}_n$ is the $2|L|$-size vector in which $\bar{x}^{\mathrm{f}}_{i,n}$ is the average rate of legitimate traffic on the link i in the forward direction before time n and $\bar{x}^{\mathrm{b}}_{i,n}$ is one on the same link in the backward direction. We explain how to calculate $\bar{X}_n$ in Subsection 2.1.4.

Then, by using G_n, we estimate the increases in traffic between every pair of sources and destinations. The increase in traffic can be shown as a $|E| \times |E|$ matrix F_n whose element $f_{i,j,n}$ $(i, j \in E)$ indicates the increase in traffic traversing from edge link i to edge link j.

Eq. (2) cannot be used to estimate the traffic increase matrix from G_n, which may include negative values, because it supports only positive values. Therefore, we modify Eq. (2) to support negative values. We define the traffic increase matrix F_n, having the traffic increase $f_{i,j,n}$, from edge link i to j between the time $n-1$ and n. The value of $f_{i,jn}$ is calculated from

$$f_{i,j,n} = \begin{cases} g^{\mathrm{in}}_{i,n} \dfrac{g^{\mathrm{out}}_{j,n}}{\sum_{\{k:(g^{\mathrm{out}}_{k,n}>0)\}} g^{\mathrm{out}}_{k,n}} & (g^{\mathrm{in}}_{i,n}>0, g^{\mathrm{out}}_{j,n}>0) \\ -\left| g^{\mathrm{in}}_{i,n} \dfrac{g^{\mathrm{out}}_{j,n}}{\sum_{\{k:(g^{\mathrm{out}}_{k,n}<0)\}} g^{\mathrm{out}}_{k,n}} \right| & (g^{\mathrm{in}}_{i,n}<0, g^{\mathrm{out}}_{j,n}<0) \\ 0 & (\text{others}). \end{cases} \tag{6}$$

Focusing on the increase in the traffic, we can eliminate the effect of the amount of legitimate traffic and estimate the increase in the traffic more accurately. That is, we can estimate that the increase in traffic from attack sources to the victim is large by checking the increase in traffic when the attack starts. If the monitored network suffers from multiple attacks whose sources and victims are different, some traffic from different sources to different destinations concurrently increases. In this case, the estimated increase in traffic is proportional to the increase in traffic measured at the sources. That is, traffic from a source of an attack to a victim of another attack is estimated as increased. However, we can identify the attack sources that generate the attack traffic, even if we could not identify the victim node exactly where the attack source sends the attack traffic to.

2.1.3 Modification of traffic matrix

Although F_n is a $|E| \times |E|$ matrix, F_n can be denoted as following the $|E|^2$-size vector;

$$F_n = \begin{bmatrix} f_{1,1,n} \\ f_{1,2,n} \\ \vdots \\ f_{1,|E|,n} \\ f_{2,1,n} \\ \vdots \\ f_{|E|,|E|,n} \end{bmatrix} \tag{7}$$

Due to the fact that the total amount of traffic on the link is the summation of the traffic of flows that are passing the link, F_n and G_n satisfy

$$G_n = AF_n, \tag{8}$$

where A is a $2|L| \times |E|^2$ routing matrix which is given by

$$A = \begin{bmatrix} a^{\mathrm{f}}_{1,1,1} & a^{\mathrm{f}}_{1,2,1} & \cdots & a^{\mathrm{f}}_{|E|,|E|,1} \\ a^{\mathrm{b}}_{1,1,1} & a^{\mathrm{b}}_{1,2,1} & \cdots & a^{\mathrm{b}}_{|E|,|E|,1} \\ a^{\mathrm{f}}_{1,1,2} & a^{\mathrm{f}}_{1,2,2} & \cdots & a^{\mathrm{f}}_{|E|,|E|,2} \\ a^{\mathrm{b}}_{1,1,2} & a^{\mathrm{b}}_{1,2,2} & \cdots & a^{\mathrm{b}}_{|E|,|E|,2} \\ \vdots & \vdots & \ddots & \vdots \\ a^{\mathrm{f}}_{1,1,|L|} & a^{\mathrm{f}}_{1,2,|L|} & \cdots & a^{\mathrm{f}}_{|E|,|E|,|L|} \\ a^{\mathrm{b}}_{1,1,|L|} & a^{\mathrm{b}}_{1,2,|L|} & \cdots & a^{\mathrm{b}}_{|E|,|E|,|L|} \end{bmatrix}. \tag{9}$$

$a^{\mathrm{f}}_{i,j,k}$ $(i, j \in E, k \in L)$ is equal to 1 if the traffic from edge link i to edge link j traverses on link k in the forward direction, or set to zero otherwise. In a similar way, $a^{\mathrm{b}}_{i,j,k}$ $(i, j \in E, k \in L)$ is equal to 1 if the traffic from edge link i toedge link j traverses on link k in the backward direction or zero otherwise. Matrix A can be obtained by monitoring the routing messages, such as the Link State Advertisement (LSA) of OSPF [15] or by simulating routing [16].

The traffic matrix estimated by the gravity model cannot satisfy Eq. (8) because Eq. (6) does not use the traffic statistics on the internal links of the monitored

network, but uses only the traffic measurements of the edge links. Therefore, we adjust the traffic matrix estimated by the gravity model to satisfy Eq. (8). We can obtain the final estimation results for F_n from

$$F_n = F'_n + pinv(A)(G_n - AF'_n), \tag{10}$$

where F'_n is the $|E|^2$-size vector indicating the results estimated by Eq. (6), and $pinv(A)$ is a pseudo-inverse matrix of A. We can obtain the least squares solution of $X = AT$ by multiplying X by the pseudo-inverse matrix of A. That is, by Eq. (10), we obtain F_n which satisfies $Gn = AF_n$ and minimizes $|F_n - F'_n|^2$. In this paper we obtain $pinv(A)$ by using a function of Scilab [17].

2.1.4 How to estimate average of legitimate traffic

Our method for estimating the increase in traffic uses the average rate of legitimate traffic. The rate of legitimate traffic varies according to the time of day. To follow the time-of-day variation of this traffic, we assume that the average rate of legitimate traffic $\bar{X}_n$ is basically estimated by the exponentially weighted average of the monitored traffic rate from

$$\bar{X}_{n+1} = \alpha X_n + (1-\alpha)\bar{X}_n \quad (0 < \alpha < 1) \ . \tag{11}$$

Note here, other estimation method (e.g. AutoRegressive Integrated Moving Average (ARIMA) model) can also be used to estimate the average rate of legitimate traffic. However, according to [10], exponentially weighted average is almost as accurate as ARIMA model though it is very simple. For this reason, we use the exponentially weighted average described above.

However, when the traffic increases suddenly and rapidly (we call these *spikes* throughout the rest of this paper), $\bar{X}_n$ becomes large after the spike. The large $\bar{X}_n$ value causes difficulties in the identification of the increase in traffic after the spike, because the larger $\bar{X}_n$ value makes the impact of $(X_n - \bar{X}_n)$ small, even for cases of increases in traffic. For this reason, we must estimate the average of the legitimate traffic without the effect of spikes.

We can eliminate the effect of spikes by updating only the elements of $\bar{X}_n$ corresponding to the link on which the increase in traffic is under a threshold. However, as described in the previous subsection, our method assumes the situation covered by Eq. (8). For this reason, we should update $\bar{X}_n$ by satisfying Eq. (8).

For this purpose, we update $\bar{X}_n$ using an element from estimated F_n, which is not rapidly increasing. First, we extract the element not increasing rapidly from F_n. We denote the $|E| \times |E|$ matrix of the extracted elements as $\hat{F}_n$. Each element $\hat{f}_{i,j,n} \quad (i,j \in E)$ is defined by

$$\hat{f}_{i,j,n} = \begin{cases} f_{i,j,n} & (f_{i,j,n} < \mu_{i,j} + \beta\sigma_{i,j}) \\ 0 & (\text{others}) \end{cases} \ . \tag{12}$$

where $\mu_{i,j}$ is the average of the last J values of $f_{i,j,k} \quad (i,j \in E, n-J < k \le n)$ and $\sigma_{i,j}$ is the variance of the last J values of $f_{i,j,k} \quad (i,j \in E, n-J < k \le n)$. β is the parameter by which we can set the threshold. By Eq. (12), when the traffic from i to j sharply increases at time n beyond the threshold, $\hat{f}_{i,j,n}$ is zero, while in other cases, $\hat{f}_{i,j,n}$ is $f_{i,j,n}$.

After that, we update $\bar{X}_{n+1}$ with the following equation.

$$\bar{X}_{n+1} = \alpha(\bar{X}_n + A\hat{F}_n) + (1-\alpha)\bar{X}_n \tag{13}$$

In Eq. (13), we calculate the increase in traffic on each link from $\hat{F}_n$ by $A\hat{F}_n$. Using the increase in traffic, we calculate the amount of traffic at time n as $\bar{X}_n + A\hat{F}_n$. Then, we update $\bar{X}_{n+1}$ as the weighted average of the monitored traffic using the amount of traffic at time n.

With the above stated equations, we can update $\bar{X}_{n+1}$ without the effect of any spikes in F_n. By deciding whether each element of F_n should be used to update, we can satisfy Eq. (8).

The above model to estimate the average of legitimate traffic uses three parameters, α, β and J. If the change of legitimate traffic causes large entries in G_n, the legitimate traffic is errorneously identified as an attack. To avoid this errorneous detection, we should set the parameters to the value which can minimize G_n during no attack times.

α indicates the weight to current measurement. Setting α to a large value, we cannot eliminate the impact of temporal change of traffic on the estimated average of legitimate traffic. As a result, the impact enlarges the increase of legitimate traffic from the estimated average of legitimate traffic. However, setting α to a small value, the estimated average of legitimate traffic cannot follow the periodic change of traffic. Therefore, we use the traffic data monitored before to set α to adequate value. By using the traffic data monitored before, we set α to the value which can minimize the squared errors between monitored traffic rate and its estimated rate.

By β, we can set the sensitivity to detect the spikes. If we set β to a large value, the spike also affect the estimated average of legitimate traffic. On the other hand, if we set β to a small value, the periodical change of traffic may mistakenly be identified as a spike. As a result, we cannot udpate the estimated average of legitimate traffic. Therefore, we use the traffic data monitored before and set β to as small value as possible without identifying the periodic change of the monitored traffic as a spike.

J is the number of monitored data used for setting a threshold to detect spikes. By setting J to large value, we use more monitored data. However, setting J to large value needs more memories to store the monitored data. Therefore, we set J to as large value as possible.

2.2 Identification of attack sources

When an attack starts, the traffic sharply increases from the attackers to the victim. Moreover, the larger the increase is, the more serious the impact on the network resources is. We identify the sources increasing the traffic on the victim as attack sources. However, when many attack sources are widely distributed, the impact of the attack is serious, even if each attack source generates a small rate of attack traffic. Thus, the identification of the attack sources, by setting a static threshold to the increase in traffic, is not sufficient. Instead of setting a threshold, we identify the attack sources by comparing the increase in traffic from each edge link to the victim. When the victim detects an attack, it is reasonable enough to assume that the source generating more traffic to the victim has more likelihood of being considered an attack source. With this assumption, we identify attack sources from the nodes generating a lot of traffic to the victim node. We also use the total rate of traffic to detect the event of an attack. By using the total rate of attack traffic, we can identify the attack sources even in cases of DDoS. The total rate of attack traffic can be estimated from the increase of the egress traffic to the victim.

When an attack starts, the egress traffic increases with the rate of the attack traffic. However, the rate of legitimate traffic may also change according to the time-of-day. Assuming the increase of egress traffic to the victim is attack traffic may be an overestimation of the attack traffic, because an increase in egress traffic includes both legitimate and attack traffic. As a result of this overestimation, the source node sending only legitimate traffic may be mislead as an attack source. For this reason, we estimate the rate of the attack traffic $\tilde{g}^{\text{out}}$ from results of traffic estimation. When an attack to edge link j starts at the time n, $\tilde{g}^{\text{out}}$ is estimated from

$$\tilde{g}^{\text{out}} = g^{\text{out}}_{j,n} - \mu^{\text{out}}_{j} - \gamma, \tag{14}$$

where $g^{\text{out}}_{j,n}$ is the egress traffic on edge link j to the outside of the monitored network, μ^{out}_{j} is the average of the last J values of $g^{\text{out}}_{j,k}$ $(n - J \leq k < n)$, and γ is the parameter indicating the variation in the rate of the legitimate traffic. In this equation, μ^{out}_{i} represents the effect of the time-of-day variation of the legitimate traffic and γ mitigates the effect of the other variations of the legitimate traffic. By adequately setting γ, we can estimate $\tilde{g}^{\text{out}}$ as the value which may be a little smaller than the actual attack rate, but is never larger than the actual attack rate.

Then, we identify source i as attack source when source i satisfies

$$\sum_{(k:f_{k,j,n} > f_{i,j,n})} f_{k,j,n} \leq \tilde{g}^{\text{out}}, \tag{15}$$

where $f_{i,j,n}$ is the element of the estimated traffic increase matrix F_n corresponding to the traffic from edge link i to victim edge link j. Before using Eq. (15), we must first sort out the set of $f_{k,j,n}(1 \leq k \leq N)$ by descending order based on their values. We then calculate the total of the top m traffic to the victim node. We compare the total top m traffic with the estimated egress traffic $\tilde{g}^{out}$. We increment m by one and calculate the total top m traffic until the total traffic exceeds $\tilde{g}^{out}$. Finally, we identify these m nodes as the attack sources.

Let us denote the actual rate of attack traffic as t^{attack} and that the sum of the top m increases of the egress traffic to the victim as $t^{\text{top}(m)}$. If $t^{\text{top}(m)}$ is smaller than $\tilde{g}^{\text{out}}$ and $t^{\text{top}(m+1)}$ is larger than $\tilde{g}^{\text{out}}$, then we can identify $m+1$ attack sources. In this case, the total rate of attack traffic from the identified attack sources is $t^{\text{top}(m+1)}$, which is larger than $\tilde{g}^{\text{out}}$. That is, the rate of the attack traffic from the unidentified attack sources is at most $t^{\text{attack}} - \tilde{g}^{\text{out}}$, which is calculated from $\gamma + \mu - f^{\text{normal}}$ where f^{normal} is the increase in legitimate traffic. Therefore, by setting γ adequately, we can identify most of the attack sources and limit the rate of attack traffic from the unidentified attack sources.

2.3 Calculation time of our method

In our method, we estimate the traffic increase matrix from Eq. (6), Eq. (10) and Eq. (13). The calculation time for Eq. (6) is $O(|E|^2)$ because we should estimate the traffic transmitted between every pair of ingress and egress points. Although Eq. (10) needs the value of $pinv(A)$, we do not have to calculate $pinv(A)$ each time, since A seldom varies. The calculation times for Eq. (10) and Eq. (13) are $O(|L||E|^2)$, because they include the products of a $2|L| \times |E|^2$ matrix and a $|E|^2$-sized vector. That is, the calculation time for estimating the traffic increase matrix is $O(|L||E|^2)$.

To identify the attack sources, we check whether the candidate satisfies Eq. (15). The number of candidates is $|E|$. If $f_{k,j,n}(1 \leq k \leq N)$ are sorted by descending order, we check the condition at most $|E|$ times. Using a quicksort algorithm, we can sort $|E|$ elements by less than $O(|E|^2)$ comparisons. That is, the calculation time for identifying the attack sources using the estimated matrix is $O(|E|^2)$.

Consequently, the calculation time for our method is $O(|L||E|^2)$. However, in a large network, we can reduce the calculation time by using a link load on only a part of the links, not on all links. This can be done by taking A and X_n from a part of links.

3. Evaluation

We evaluate our method by using simulations. In our simulation, we use the backbone topology of Abilene

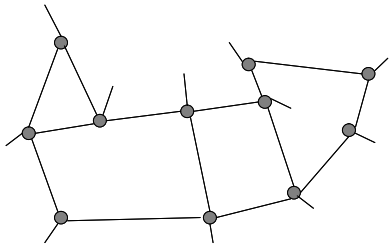


Fig. 3 Backbone Topology of the Abilene

(11 nodes and 14 links) shown in Fig. 3 for the monitored network. We assume that each node in Fig. 3 has one edge link. That is, in this simulation, the purpose of our method is to extract nodes connecting to attackers from 11 nodes in Fig. 3. We use the traffic data captured for 24 hours with 5 minutes interval on the Abilene backbone network for the legitimate traffic in the simulation.The sampling rate of the data is 1:100 (that is, one out of every 100 packets is sampled). In this simulation, we use packets/sec to measure the traffic rate, because attacks sending a number of small packets (including SYN flood attacks, which are the most frequent attacks [18]) affect packets/sec more significantly than byte/sec.

In our simulations, we set α to 0.3 and β to 3, which allows a time-of-day variation of the traffic.

3.1 Accuracy in estimating the increase of traffic

First, we validate that our method can accurately estimate the increase in traffic. Fig. 4 shows the time-dependent variation of the arrival rate of each packet between a source and a destination. Fig. 5 compares the actual time-dependent variation of the increase in arrival traffic with its estimated rate. Comparing Fig. 4 and Fig. 5, we can see that by monitoring the increase in traffic, we can eliminate the time-of-day variation of the traffic. That is, by monitoring the increase in traffic, we can identify the attack sources without the effect of a time-of-day variation in the traffic. From Fig. 5, we also see that in the cases where a rapid increase in traffic occurs, our method can accurately estimate it.

We performe another simulation to evaluate accuracy when attacks from several sources start. We inject attack traffic from randomly selected five sources to a single destination. Fig. 6 and Fig. 7 compare the results of the estimations with actual values. The horizontal axis is the actual rate of traffic and the vertical axis is the estimated value. In Fig. 6, the attack rate from each source is 4000 packets/sec. In this case, 25% of all packets to the victim are attack packets. In Fig. 7, the attack rates from each source is 10000 packets/sec. In this case, about a half of all packets to the victim are attack packets. The lines in both figures show $x \pm 1000$. These figures show we can accurately estimate the increase in traffic. Even for large attacks, we can estimate the increase in traffic with an error rate of less than 1000 packets/sec.

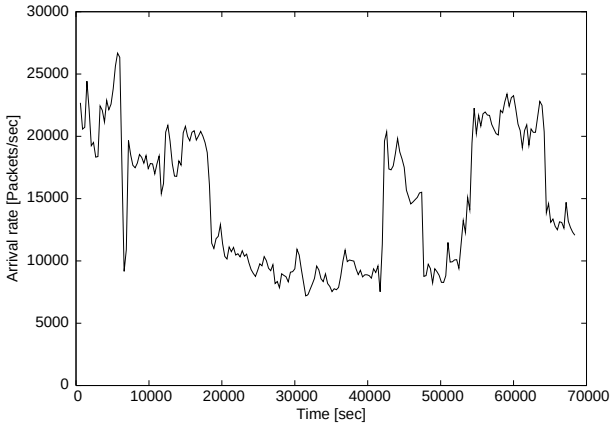


Fig. 4 Time-dependent variation of arrival rate of packets

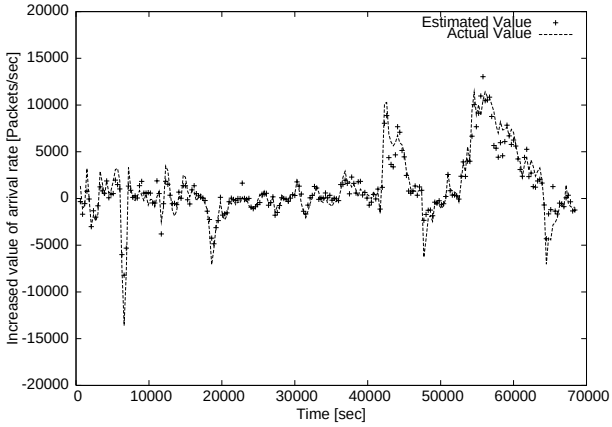


Fig. 5 Time-dependent variation of increase of arrival rate of packets between source and destination.

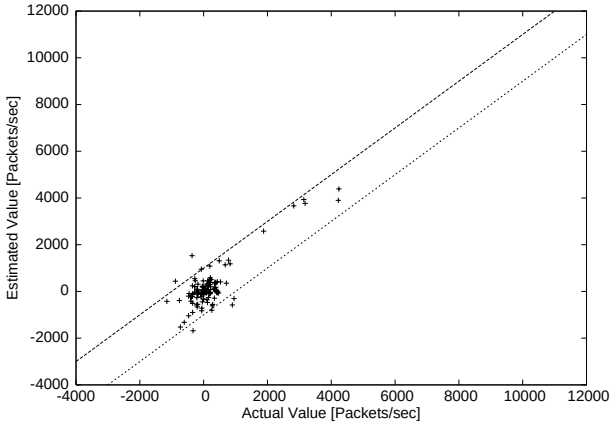


Fig. 6 Estimated value vs. Actual value (4000 packets/sec attack injected)

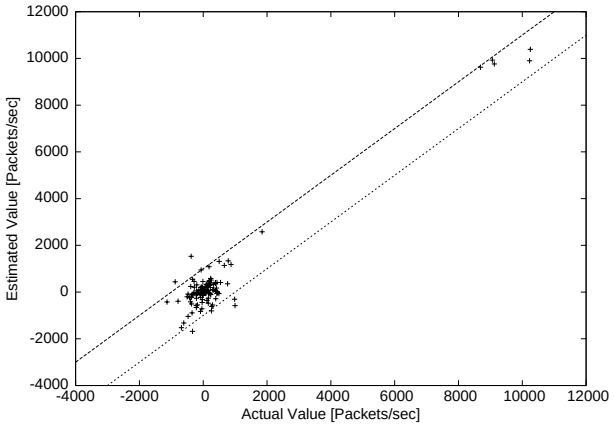


Fig. 7 Estimated value vs. Actual value (10000 packets/sec attacks injected)

3.2 Accuracy of identification of attack sources

3.2.1 Definition of false-positive and false-negative

The accuracy of our method for identifying attack sources is evaluated by two metrics, false-positive and false-negative. We define false-positive as a case where a source not generating attack traffic is erroneously identified as an attack sources. We define false-negative for cases where an attack source cannot be identified. That is, the number of false-positives indicates the number of sources erroneously identified as attack sources and the number of false-negatives indicates the number of attack sources that cannot be identified. We also define the false-negative and false-positive rates as follows:

$$\text{false-negative rate} = \frac{\text{\# of false-negative}}{\text{total \# of attack sources}}$$

$$\text{false-positive rate} = \frac{\text{\# of false-positive}}{\text{total \# of sources not generating attack traffic}}$$

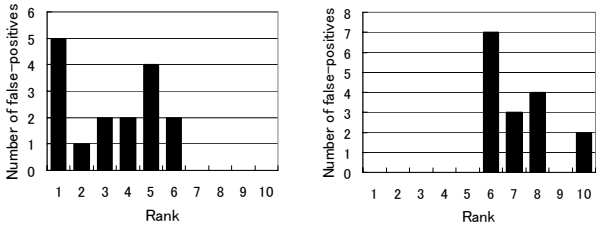
3.2.2 Number of attack sources vs. false-positives and false-negatives

We simulate our method to identify attack sources, changing the number of attack sources. We inject attack packets at 14 different times which are randomly selected. We change the number of attack sources from one to five and attack sources are randomly selected. In this simulation, we set the attack rates so that 25% of all packets to the victim are attack packets and the attack rate from each attack source is equal. We set γ to 6000 packets/sec.

Table 1 shows the total number of false-positives and false-negatives of 14 attacks and their rates. From these results, we can accurately identify the attack sources regardless of the number of attack sources.

Table 1 Number of attack sources vs. false-positives and false-negatives

# of attack sources (total # of attack sources)	# of false-negatives (false-negative rate)	# of false-positives (false-positive rate)
1 (14)	0 (0.00)	12 (0.09)
2 (28)	0 (0.00)	6 (0.05)
3 (42)	2 (0.04)	12 (0.12)
4 (56)	6 (0.14)	14 (0.16)
5 (70)	14 (0.20)	16 (0.21)



(a) Rank of estimated increase vs. number of false-positives

(b) Rank of actual increase vs. number of false-positives

Fig. 8 Rank of increase in traffic vs. number of false-positives

However, there are a few false-positives. Therefore, we investigate such false-positives. Figure 8 shows where these false-positives are ranked in estimated increase in traffic and actual increase in traffic when the number of attack sources is five. In this figure, the horizontal axis is the rank order in estimated increase in traffic and actual increase in traffic and vertical axis is the number of false-positives correnponding to the rank order. From this figure, though actual increases in traffic from the sources mistakenly identified are ranked 6th or lower, estimated ones are ranked 5th or higher. That is, the reason of these false-positives is estimation errors. These estimation errors are caused by the rapid increase in traffic traversing to the link that is near the link to the victim. In these cases, the rapid increases cause errors because most of the path of the increased traffic is common with the path from the source of the increased traffic to the victim.

3.2.3 γ vs. false-positive and false-negative

We evaluate the relationship between γ and the false-positives or false-negatives in our method by using a simulation with various values of γ. In this simulation, we inject attack packets from randomly selected five sources at randomly selected 14 different times. Fig. 9 shows the results. In this figure, we inject two kinds of attacks. First, the attack rate from each source is

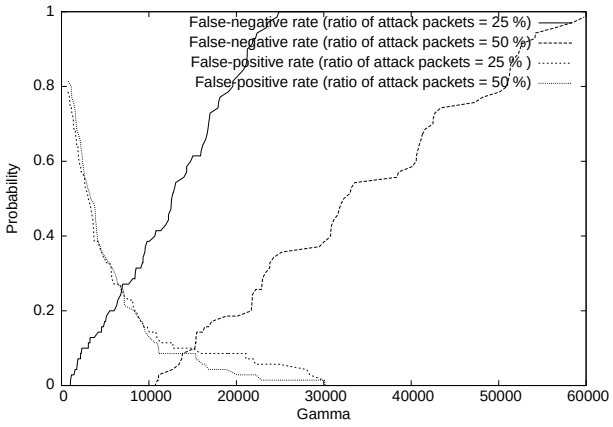


Fig. 9 γ vs. false-negative and false-positive

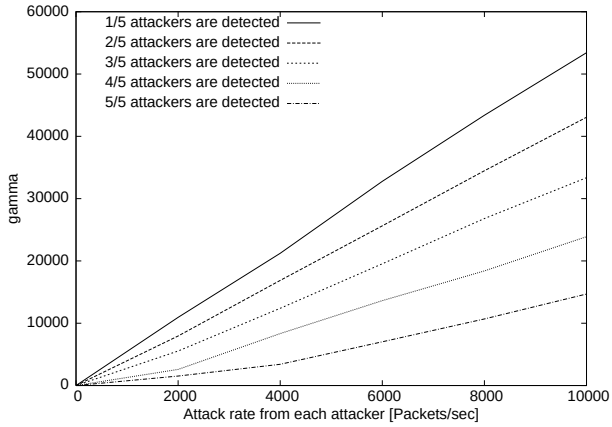


Fig. 10 Relationship between attack rate and γ to identify attachk sources

4000 packets/sec. In this case, 25 % of all packets to the victim are attack packets. In the second one, the attack rate from each source is 10000 packets/sec. In this case, about a half of all packets to the victim are attack packets. From Fig. 9, we can see that the proposed method can reduce the number of false-positives by setting γ to a larger value. However, a large γ causes many false-negatives. In addition, when comparing two kinds of attacks, we can also see that if we set γ to the same value, we have less false-negatives in cases of larger attacks than in smaller attacks. From this figure, we can also see that the number of false-positives is almost the same, regardless of the injected attack rate. That is, the attack rate does not affect the number of false-positives.

3.2.4 γ vs. attack rate from unidentified attack sources

To evaluate the relationship between γ and the total rate of attacks from unidentified attack sources, we simulate our method to identify attack sources, changing the attack rate. In this simulation, we inject attack packets from randomly selected five sources at randomly selected 14 different times and the attack rate from each source is equal.

In Fig. 10, the horizontal axis is the total rate of the attack traffic. Each line shows γ, which can identify one of the five attack sources, two of the five attack sources, three of the attack sources, four of the attack sources and all of the attack sources at all time. From this figure, we can see that a smaller γ is needed to identify attack sources for smaller attacks or to identify more attack sources. This figure also shows that even when we set γ to the same value, we can identify more attack sources for large attacks. For example, by setting γ to 10000 packets/sec, we can identify only one attack source when the attack rate from each attacker is 2000 packets/sec. However, by setting γ to the same value, we can identify four attack sources when the attack rate is 6000 packets/sec.

Fig. 11 shows the relationship between γ and the total rate of attack traffic from unidentified attack sources. In this figure, the three lines indicate the false-positive rate and the maximum and average of the total rate of attack traffic from the unidentified attack sources. From this figure, we can see that by setting γ to a smaller value, the attack rate from unidentified attack sources can be small while a smaller γ causes more false-positives. We can also see that the average of the total rate of attack traffic from unidentified attack sources is near γ. That is, the total rate of attack traffic from unidentified attack sources is closely related to γ.

However, in some cases, the total rates of attack traffic from unidentified attack sources are higher than γ. There are two reasons for this. First one is caused by the decrease of legitimate traffic to the victim. In this case, our method underestimates the total attack rates to the victim. Another reason is caused by errors in our method for estimating the increases in traffic. Our method for estimation has errors in the range of ± 1000 packets/sec. That is, the estimated increase in traffic from an attack source may be 1000 packets/sec less than the actual increase, while the difference from one to another attack source may be 1000 packets/sec larger than the actual increase. In this case, this error causes 1000 packets/sec attack traffic from unidentified attack sources. However, we can accurately identify attack sources sending attack traffic whose estimated rate is larger than $\gamma + \mu - f^{\mathrm{normal}}$. That is, by adequately setting γ, we can identify attack sources even when the estimated increases have several errors.

As previously mentioned, the total rate of attack traffic from unidentified attack sources is closely related to γ. That is, by defining the maximum attack rate that does not affects the network resources, we can adequately set γ to limit the total attack rate from unidentified attack sources to the defined maximum attack

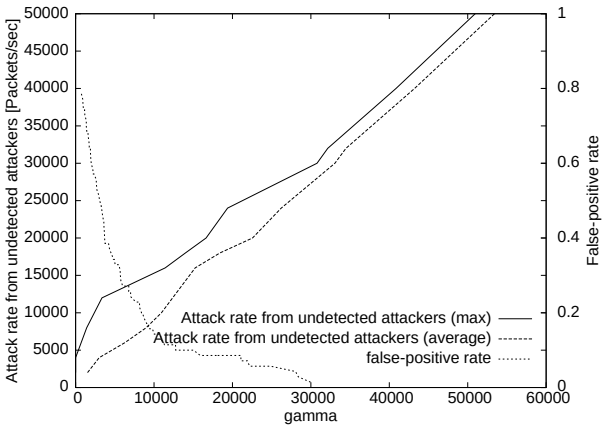


Fig. 11 γ vs. total rate of traffic from unidentified attack sources

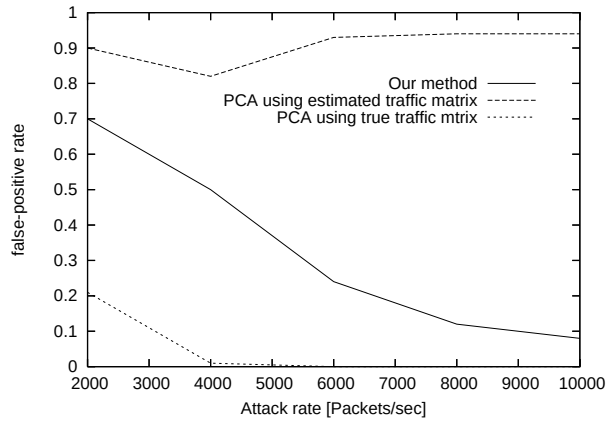


Fig. 12 Comparison of our method and PCA method using estimated traffic matrix

rate.

3.2.5 Comparison with existing method

Finally we compare our method with exisiting method.

The method proposed in [9] separates traffic matrix into normal and abnormal subspaces by applying Principal Component Analysis (PCA). When the square sum of abnormal subspaces is larger than the threshold, it is detected as attacks. These abnormal subspaces also can be used to identify attack sources. Using abnormal subspaces, we can identify all attack sources by identifying the sources having the largest abnormal subspaces, until the squared sum of the abnormal subspaces of traffic which are not identified as attack becomes less than the threshold. In this simulation, we use two traffic matrix for PCA method. One is the true traffic matrix. Another is estimated by the method proposed in [14]. By using estimated traffic matrix, we compare our method with PCA method in the conditions that we can only monitor link loads.

We compare our proposed and PCA methods by simulation. In this simulation, we inject attack packets from randomly selected five sources at randomly selected 14 different times and the attack rate from each source is equal. We compare the false-positive rates when we set the thresholds so that false-negative rates are less than 0.1. Fig. 12 shows the results. In this figure, the horizontal axis is the attack rate from one attack source and the vertical axis is the false-positive rate. From this figure, false-positive rate of PCA method with the true traffic matrix is low. That is, in the case of monitoring traffic matrix accurately, PCA method identifies attack sources accurately. However, false-positive rate of PCA method using estimated traffic matrix is quite high. This is because the method proposed in [14] cannot estimate the traffic matrix accurately in the case of attacks. Because the increase in traffic matrix estimated by the method proposed in [14] is proportional to the total rate of traffic monitored at the source, we mistakenly identify sources having large amount of traffic or cannot identify attack sources having small amount of traffic. As a result, PCA method cannot identify attack sources accurately in the case that we can monitor only the link utiliztions.

On the other hand, our method can identify attack sources accurately. This is because our method estimates not the total amount of traffic but the increase in traffic. By focusing on the increase in traffic, we can accurately estimate the increase in traffic caused by the attacks and identify attack sources. From this figure, we can also see that our method can identify attack sources more accurately when the attack rate is larger. This is because larger attack causes the significant increase in traffic. As a result, because the increase in traffic caused by the attack is much larger than the time-dependent variations of legitimate traffic, we can identify the sources easier.

This way, to detect attack sources, traditional traffic matrix estimation method is insufficient and we need to use the estimation method focusing on the changes in traffic caused by attacks. In our method, we can identify attack sources accurately by focusing on the increase in traffic.

4. Conclusion and future works

In this paper, we have proposed a new method for identifying attack sources by estimating traffic matrices. Our method periodically collects link load data from each router through SNMP and estimates the increase in traffic between each source and destination. When attacks start, our method identifies the sources of the attack using the estimated increase. We have also shown that our method can accurately identify attack sources without any false-positives by setting the adequate parameters of γ.

One of our future works are to set γ and sampling rate automatically.

Acknowledgement

We acknowledge all the people that permitted the Abilene Netflow data to be available on-line, and particularly Mark Fullmer.

References

[1] B. Wang and H. Schulzrinne, "A denial-of-service-resistant IP traceback approach," in *Proceedings of IEEE Symposium on Computers and Communications*, vol. 1, pp. 351–356, June 2004.

[2] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proceedings of ACM SIGCOMM 2000*, pp. 295–306, Aug. 2000.

[3] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proceedings of IEEE INFOCOM 2001*, vol. 2, pp. 878–886, Apr. 2001.

[4] K. Law, J. C. Lui, and D. K. Yau, "You can run, but you can't hide: An effective methodology to traceback DDoS attackers," in *Proceedings of International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, pp. 433–440, Oct. 2002.

[5] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-packet IP traceback," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 721–734, Dec. 2002.

[6] T.-H. Lee, W.-K. Wu, and T.-Y. W. Huang, "Scalable packet digesting schemes for IP traceback," in *Proceedings of IEEE International Conference on Communications 2004*, pp. 1008–1013, June 2004.

[7] A. Soule and K. S. andNina Taft, "Combining filtering and statistical methods for anomaly detection," in *Proceedings of Internet Measurement Conference 2005*, pp. 331–344, Oct. 2005.

[8] A. Lakhina, M. Crovella, and C. D. February, "Diagnosing network-wide traffic anomalies," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 219–230, Aug. 2004.

[9] A. Lakhina, M. Crovella, and C. Diot, "Detecting distributed attacks using network-wide flow traffic," in *Proceedings of FloCon 2005 Analysis Workshop*, 2005.

[10] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: Methods, evaluation, and applications," in *Proceedings of ACM SIGCOMM Internet Measurement Conference 2003*, pp. 234–247, Oct. 2003.

[11] "Cisco NetFlow." available at `http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_%home.html`.

[12] Cisco, "NetFlow performance analysis." available at `http://www.cisco.com/en/US/tech/tk812/technologies_white_paper0900aecd8%02a0eb9.shtml`.

[13] L. Huang, X. L. Nguyen, M. Garofalakis, M. Jordan, A. Joseph, and N. Taft, "Distributed PCA and network anomaly detection." Technical Report UCB/EECS-2006-99, Electrical Engineering and Computer Science Department, University of California Berkeley, July 2006.

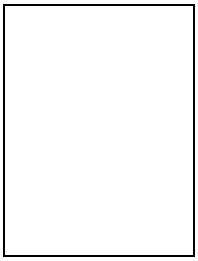
[14] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proceedings of ACM SIGMETRICS*, pp. 206–217, June 2003.

[15] D. Watson and C. Labovitz, "Experiences with monitoring OSPF on a regional service provider," in *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 204–213, 2003.

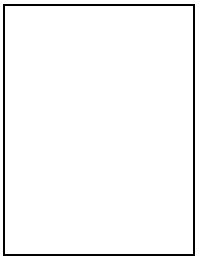
[16] A. Fedmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "NetScope: Traffic engineering for IP networks," pp. 11–19, Apr. 2000.

[17] "Scilab development team." available at `http://www-rocq.inria.fr/scilab/`.

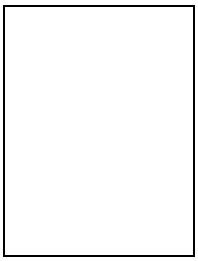
[18] "Symantec internet security threat report." available at `http://www.symantec.com/enterprise/threatreport/index.jsp`, Mar. 2005.



Yuichi Ohsita received the M.E. degree in Information and Computer Science from Osaka University, Japan, in 2005. He is now an Assistant Professor in the Graduate School of Economics at Osaka University. His research interests include traffic matrix estimation and countermeasure against DDoS attacks. He is a member of IEEE.



Shingo Ata received M.E. and Ph.D. degrees in Informatics and Mathematical Science from Osaka University in 1998 and 2000, respectively. He is an Associate Professor in Information and Communication Engineering at Osaka City University, Japan. His research works include networking architecture, design of communication protocols, and performance modeling on communication networks. He is a member of IEEE and ACM.



Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an Associate Professor in the Graduate School of Engineering Science, Osaka University, and from April 1999, he has been a Professor of Osaka University. He moved to Graduate School of Information Science and Technology, Osaka University in April 2004. He has more than three hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, The Internet Society and IPSJ.