Fast Detection of Distributed Global Scale Network Attack Symptoms and Patterns in High-speed Backbone Networks

Sun Ho Kim, *Non-Member* and Byeong-hee Roh, *Member, KSII*
Graduate School of Information and Communications, Ajou University,
San 5 Wonchon-Dong Youngtong-Gu, Suwon, 443-749, Korea
[E-mail: {shkim11, bhroh}@ajou.ac.kr]
*Corresponding author: Byeong-hee Roh

Received May 2, 2008; revised May 20, 2008; accepted May 26, 2008; published June 25, 2008

Abstract

Traditional attack detection schemes based on packets or flows have very high computational complexity. And, network based anomaly detection schemes can reduce the complexity, but they have a limitation to figure out the pattern of the distributed global scale network attack. In this paper, we propose an efficient and fast method for detecting distributed global-scale network attack symptoms in high-speed backbone networks. The proposed method is implemented at the aggregate traffic level. So, our proposed scheme has much lower computational complexity, and is implemented in very high-speed backbone networks. In addition, the proposed method can detect attack patterns, such as attacks in which the target is a certain host or the backbone infrastructure itself, via collaboration of edge routers on the backbone network. The effectiveness of the proposed method are demonstrated via simulation.

***Keywords*:** Network security, DDoS, attack traffic modeling, attack detection

Digital Object Identifier 10.3837/tiis.2008.03.001

1. Introduction

There have been several threats on the Internet infrastructure by malicious network attacks such as [1] and [2]. Attacks on the Internet infrastructure can cause enormous damage, since different infrastructure components of the Internet have implicit trust relationships [3].

Traditional security solutions against network attacks have been focused on individual packets or flows [3][4][5][6]. However, they require very high computational complexity, and cannot be directly applied to high-speed backbone network environments. In order to solve the computational complexity problems, network based anomaly detection schemes have been also proposed [7][11][12][13]. In those schemes, significant deviations from normal behavior of aggregate traffic pattern at a high-speed link are used as key criteria to detect attack symptoms. Those methodologies focused on detecting network attacks on individual links for securing end-networks connected to those links. However, in order to react against distributed global-scale network attacks, the attack pattern as well as symptom should be detected at the same time. It is noted that distributed global-scale network attacks can be more precisely visible in the backbone domain rather than at any individual link.

In this paper, we propose an efficient method for detecting distributed global-scale network attack symptoms and patterns on a high-speed Internet backbone network. The proposed attack detection method is extended from our previous work of [7], in which attack detection is carried out at an incoming link of an edge router. Since the detection of attack symptoms is performed at the aggregate traffic level, this can result in much lower computational complexity than existing schemes. In addition, it can also detect attack patterns, such as attacks in which the target is a certain host or the backbone network infrastructure itself, via collaboration of edge routers on the backbone network.

The rest of the paper is organized as follows. In Section 2, the proposed method for detecting network attack symptoms and patterns at the aggregate traffic level is explained. Experimental results are given in Section 3. Finally, we conclude the paper in Section 4.

2. Detection of Attack Symptoms and Patterns

2.1 Network Model

Fig. 1 shows the network model considered in this paper. N edge routers, E_0, E_1, ..., E_{N-1}, are connected to a backbone network, and send traffic information associated with detection of attack symptoms and patterns to a global detection system (GDS). A detailed description on the mechanism of edge routers and GDS will be explained in the next sections.

2.2 Detection of Attack Symptoms on an Outgoing Link

To reduce the computational complexity of attack detection, the method operated at the aggregate traffic level has been proposed [7]. However, since the method of [7] detects the attack symptom at an incoming link of an edge router connected to a backbone network, it has the limitation to figure out the pattern of the detected attack. In order to react against network attacks, the attack pattern as well as symptom should be detected at the same time. To solve the problem, we extend the method of [7] to be fitted for the network model of **Fig. 1**.

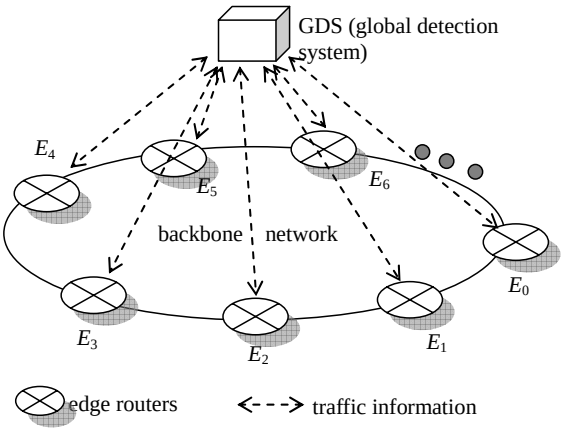


Fig. 1. Network model

In [7], two measures for the attack detection have been derived from the extensive investigation of actually captured data at an Internet backbone network. *IP spoofing* and *scanning* attack types have been considered in the investigation. The two measures are the packet count-to-the traffic volume ratio (CVR) and the average power spectrum (APS). The former reflects the dynamics of traffic volumes from the view points of burstiness, while the latter does the self-similar nature of network traffic.

Let us assume that time is partitioned into discrete periods, Δ, then Δ is a basic unit of traffic measurement. Let c_n^{ij} and v_n^{ij} (i,j=0,1,2,…,N-1, and n=0,1,2,…) be the packet count and traffic volume of aggregate traffic from E_i to E_j measured during the *n*-th Δ period, respectively. Let LΔ be the detection period, which is the time period that the detection algorithm is applied to. Each detection period consists of *L* independent consecutive Δs. And, we define packet counts and traffic volumes measured during the *m*-th detection period as vectors $\vec{\boldsymbol{c}}^{\,ij}(m) = \left[c_{mL}^{ij}, \quad \cdots, \quad c_{(m+1)L-1}^{ij}\right]$ and $\vec{\boldsymbol{v}}^{\,ij}(m) = \left[v_{mL}^{ij}, \quad \cdots, \quad v_{(m+1)L-1}^{ij}\right]$ (m=0,1,2,…), respectively. Then, the APS for vector $\vec{\boldsymbol{c}}^{\,ij}(m)$ is derived as follows;

$$\overline{P}^{ij}(m) = \sum_{k=0}^{L-1} \phi_{mk}^{ij} \tag{1}$$

where $\Psi_{mk}^{ij} = [\phi_{mk}^{ij}, \cdots, \phi_{m(L-1)}^{ij}]$ is obtained via the DFT of $\vec{\boldsymbol{c}}^{\,ij}(m)$, such that; $\Psi_{mk}^{ij} = L^{-2}\left|\mathrm{DFT}(\vec{\boldsymbol{c}}^{\,ij}(m))\right|^2$. Note that the APS is a metric associated with the effect of self-similarity due to network attacks. Xiang et al [11] also used the self-similarity to detect attack symptoms by calculating the Hurst parameter directly. However, the only self-similar nature is considered for their detection process, but it is insufficient to reflect the packet count dynamics. So, we use the second parameter, CVS, to reflect the packet dynamics.

CVR is given by;

$$\overline{R}^{ij}(m) = \frac{\vec{\boldsymbol{c}}^{\,ij}(m) \bullet \vec{\boldsymbol{e}}}{\vec{\boldsymbol{v}}^{\,ij}(m) \bullet \vec{\boldsymbol{e}}} \tag{2}$$

where $\vec{\boldsymbol{e}} = [1,1,\cdots 1]^T$, and $[\bullet]^T$ indicates a transpose matrix.

The proposed method for detecting network attack symptoms using the two metrics given in Eq. (1) and Eq. (2) is as follows. Let $x_P^{ij}(m)$ and $x_R^{ij}(m)$ be weighted averages of the APS and the CVR measured during the m-*th* detection period, respectively, and given by;

$$x_P^{ij}(m) = \alpha_P^{ij} x_P^{ij}(m-1) + (1-\alpha_P^{ij})\overline{P}^{ij}(m) \quad , \text{m=0,1,2,…} \quad (3)$$

$$x_R^{ij}(m) = \alpha_R^{ij} x_R^{ij}(m-1) + (1-\alpha_R^{ij})\overline{R}^{ij}(m) \quad , \text{m=0,1,2,…} \quad (4)$$

where α_P^{ij} and α_R^{ij} are constant values between 0 and 1.

In general, it is known that normal traffic flows in a stationary state vary within a predictable range. Let $\delta_P^{ij}(m)$ and $\delta_R^{ij}(m)$ be tolerances for weighted averages of APS and CVR of normal traffic within a certain stationary state, respectively. It is assumed that the tolerances $\delta_P^{ij}(m)$ and $\delta_R^{ij}(m)$ are determined based on the normal traffic statistics prior to actual measurement for detection [13], according to the administrative policy of the network operators.

In the case in which the measured $x_P^{ij}(m)$ and/or $x_R^{ij}(m)$ exceeds tolerances $\delta_P^{ij}(m)$ and/or $\delta_R^{ij}(m)$, we assume that a symptom of network attacks has manifested. In order to determine the case in which the network infrastructure is currently being attacked, we define three states; NORMAL, ALERT, and ATTACK. The NORMAL state is the state in which there is no attack. The ALERT state is the one in which there is a possible attack symptom, but the decision about the attack symptom has not yet been made. The ATTACK state means that it has been inferred that network resources are being attacked. The transition between these states is shown in **Fig. 2**.

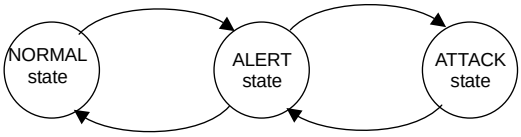


Fig. 2. Transition between states for the detection of the attack symptom

Let define $s^{ij}(m)$ and $a^{ij}(m)$ be the attack state and counter for traffic from an edge router E_i to E_j in the m-*th* detection period, respectively. Attack state $s^{ij}(m)$ is NORMAL, ALERT, or ATTACK, as shown in **Fig. 2**. The attack counter in the m-*th* detection period $a^{ij}(m)$ is increased or decreased according to the attack state in the previous detection period $s^{ij}(m-1)$, as well as the weighted averages of APS and CVR in the current detection period $x_P^{ij}(m)$ and $x_R^{ij}(m)$. And, attack state $s^{ij}(m)$ is determined according to attack counter $a^{ij}(m)$. The algorithm for detecting attack symptoms using $s^{ij}(m)$ and $a^{ij}(m)$ is shown in **Table 1**.

Table 1. Algorithm for the detection of attack symptoms on an outgoing link from E_i to E_j

<variables>
$a^{ij}(m)$: attack counter for representing the volume of attacks in m-*th* detection period $s^{ij}(m)$: attack state in the m-*th* detection period Alert_Threshold : threshold value for a transition between ALERT and ATTACK states Attack_Threshold : maximum value of attack_count
<main algorithm>

At the end of the m-*th* detection period (m=1,2,3,….), update $x_P^{ij}(m)$ and $x_R^{ij}(m)$ using (3) and (4). Then, the state is determined by the following sequence;

```
if ( s^ij(m-1) == NORMAL )
    if ( ( x_P^ij(m) > δ_P^ij(m) AND x_R^ij(m) ≤ δ_R^ij(m) ) OR ( x_P^ij(m) ≤ δ_P^ij(m) AND x_R^ij(m) > δ_R^ij(m) ) )
        s^ij(m) = ALERT;
        a^ij(m) = a^ij(m-1) + 1;
    elseif ( x_P^ij(m) > δ_P^ij(m) AND x_R^ij(m) > δ_R^ij(m) )
        s^ij(m) = ALERT;
        a^ij(m) = a^ij(m-1) + 2;
    endif
elseif ( s^ij(m-1) == ALERT )
    if ( ( x_P^ij(m) > δ_P^ij(m) AND x_R^ij(m) ≤ δ_R^ij(m) ) OR ( x_P^ij(m) ≤ δ_P^ij(m) AND x_R^ij(m) > δ_R^ij(m) ) )
        a^ij(m) = a^ij(m-1) +1;
    elseif ( x_P^ij(m) > δ_P^ij(m) AND x_R^ij(m) > δ_R^ij(m) )
        a^ij(m) = a^ij(m-1) + 2;
    elseif ( x_P^ij(m) ≤ δ_P^ij(m) AND x_R^ij(m) ≤ δ_R^ij(m) )
        a^ij(m) = a^ij(m-1) - 2;
    else
        a^ij(m) = a^ij(m-1) -1;
    endif
    if ( a^ij(m) > Alert_Threshold )
        s^ij(m) = ATTACK;
    elseif ( a^ij(m) ≤ 0 )
        s^ij(m) = NORMAL;
        a^ij(m) = 0;
    endif
elseif ( s^ij(m-1) == ATTACK )
    if ( x_P^ij(m) > δ_P^ij(m) AND x_R^ij(m) > δ_R^ij(m) )
        a^ij(m) = MIN (= a^ij(m-1) +1, Attack_Threshold );
    elseif ( x_P^ij(m) ≤ δ_P^ij(m) AND x_R^ij(m) ≤ δ_R^ij(m) )
        a^ij(m) = a^ij(m-1) -2;
    else
        a^ij(m) = a^ij(m-1) -1;
    endif
    if ( a^ij(m) ≤ Alert_Threshold )
        s^ij(m) = ALERT ;
    endif
endif
```

2.3 Detection of Global-Scale Attack Patterns on a Backbone Network

For the detection of global-scale attack patterns on a backbone network, each router E_i (i=0,1,2,...,N-1) sends the GDS attack counters and states on all its outgoing links, $a^{ij}(m)$ and $s^{ij}(m)$ (j=0,1,2,...,N-1, j≠i), respectively, at the end of every m-*th* detection period. Then, in the m-*th* detection period, the GDS has NxN attack counter and state matrices $\mathbf{A}(m)$ and $\mathbf{S}(m)$, respectively;

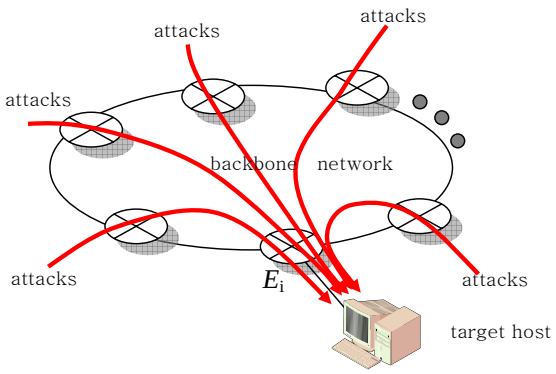
$$\mathbf{A}(m) = \left[a^{ij}(m)\right] \text{ , m=0,1,2,3,…, i,j=0,1,2,3,…,N-1} \quad (5)$$

$$\mathbf{S}(m) = \left[s^{ij}(m)\right] \text{ , m=0,1,2,3,…, i,j=0,1,2,3,…,N-1} \quad (6)$$

With matrices $\mathbf{A}(m)$ and $\mathbf{S}(m)$, the following two patterns of global-scale attack patterns are detected;

• **Concentrated Attack** : An attack concentrated on a target host or network

In an attack concentrated on a target host or network, shortly concentrated attack, a large number of packets from many infected hosts are sent to the target host or network to disable it. An example traffic flow for this kind of attack pattern is shown in **Fig. 3**(a), in which a target host is connected to a network via an edge router E_i. After vulnerable hosts in other networks are infected, all attack packets generated from these infected hosts are concentrated on router E_i via all other edge routers connected to these infected hosts. Accordingly, among all elements of attack counter matrix $\mathbf{A}(m)$, values of $a^{ji}(m)$ (j=0,…,N-1, j≠i), in the shaded region in **Fig. 3**(b), are increased. Then, after attack symptoms on attack links are detected using the algorithm shown in **Table 1**, the attack state matrix **S**(m) takes the form shown in **Fig. 3**(c), such that only elements $s^{ji}(m)$ (j=0,…,N-1, j≠i) of **S**(m) represent ATTACK while others are NORMAL. Likewise, with matrices $\mathbf{A}(m)$ and $\mathbf{S}(m)$, the pattern of concentrated attacks is easily detected.



(a) Traffic flow

$$\mathbf{A}(m) = \begin{bmatrix} a^{0,0} & \cdots & a^{0,i} & \cdots & a^{0,N-1} \\ a^{1,0} & \cdots & a^{1,i} & \cdots & a^{1,N-1} \\ a^{2,0} & \cdots & a^{2,i} & \cdots & a^{2,N-1} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a^{N-1,0} & \cdots & a^{N-1,i} & \cdots & a^{N-1,N-1} \end{bmatrix}$$

(b) $\mathbf{A}(m)$

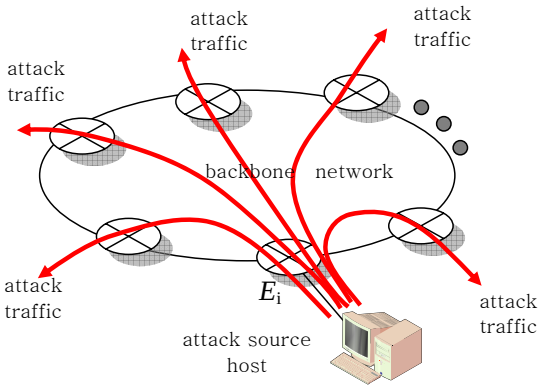
$$\mathbf{S}(m) = \begin{bmatrix} 0 & \cdots & 0 & s^{0,i}=2 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & s^{1,i}=2 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & s^{2,i}=2 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & s^{N-1,i}=2 & 0 & \cdots & 0 \end{bmatrix}$$

(c) $\mathbf{S}(m)$ (0:NORMAL, 1:ALERT, 2:ATTACK)

Fig. 3. An example pattern of a concentrated attack

• **Dispersed Attack** : An attack dispersed over the entire network

In an attack in which attack packets are dispersed over the entire network, as shown in **Fig. 4**(a), shortly dispersed attack, a certain malicious host connected to an edge router disperses a large number of packets over the entire network via other edge routers, in order to disrupt and disable the backbone network itself, or find and infect vulnerable hosts in the entire network. In this case, since a large number of attack packets are sent from an edge router, for example, E_i, to all other edge routers, only a^{ij} (j=0,…,N-1, i≠j) of $\mathbf{A}(m)$, in the shaded region in **Fig. 4**(b), are increased. Then, as a result of attack symptom detection, attack states $s^{ij}(m)$ (j=0,…,N-1), in the shaded region in **Fig. 4**(c), are ATTACK, while others are NORMAL.



(a) Traffic flow

$$\mathbf{A}(m) = \begin{bmatrix} a^{0,0} & a^{0,1} & \cdots & a^{0,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a^{i-1,0} & a^{i-1,1} & \cdots & a^{i-1,N-1} \\ a^{i,0} & a^{i,1} & \cdots & a^{i,N-1} \\ a^{i+1,0} & a^{i+1,1} & \cdots & a^{i+1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a^{N-1,0} & a^{N-1,1} & \cdots & a^{N-1,N-1} \end{bmatrix}$$

(b) **A**(m)

$$\mathbf{S}(m) = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ s^{i,0}=2 & s^{i,1}=2 & \cdots & s^{i,N-1}=2 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

(c) **S**(m) (0:NORMAL, 1:ALERT, 2:ATTACK)

Fig. 4. An example pattern of a dispersed attack

3. Experimental Results

3.1 Experimental Setup

An ns-2 simulator [9] with the network configuration shown in **Fig. 5** was used for the experiments. The backbone network consisted of five edge routers;E_0, E_1, …, E_4 with a full-meshed connection between them. Each edge router was connected to an AS router, which is an intermediate router between the backbone and other networks. Nodes N, AC and AD generated traffic which was normal, concentrated and dispersed, respectively. Node S received all traffic from other routers or networks.

Once a vulnerable host is infected, it generates attack packets according to the infected attack pattern. As the number of infected hosts in a network increases, we can assume that

aggregate attack traffic sent to the backbone network also increases, in proportion to the number of infected hosts. Based on this assumption, we constructed the following aggregate attack traffic generation model, such that the number of infected hosts over time was approximately in accordance with the worm propagation model proposed by Weaver [10]. Unlike Weaver's original model, for the convenience of the simulation, we partitioned time into segments of discrete periods, as shown in **Fig. 6**, in which the vertical axis represents the ratio of the aggregate attack traffic volume to the link capacity as a percentage. For the generation of aggregate attack traffic, we used the self-similar traffic generation model given in [8]; the Hurst parameter was 0.99, and the average number of packets in each partitioned time segment is shown in **Fig. 6**.

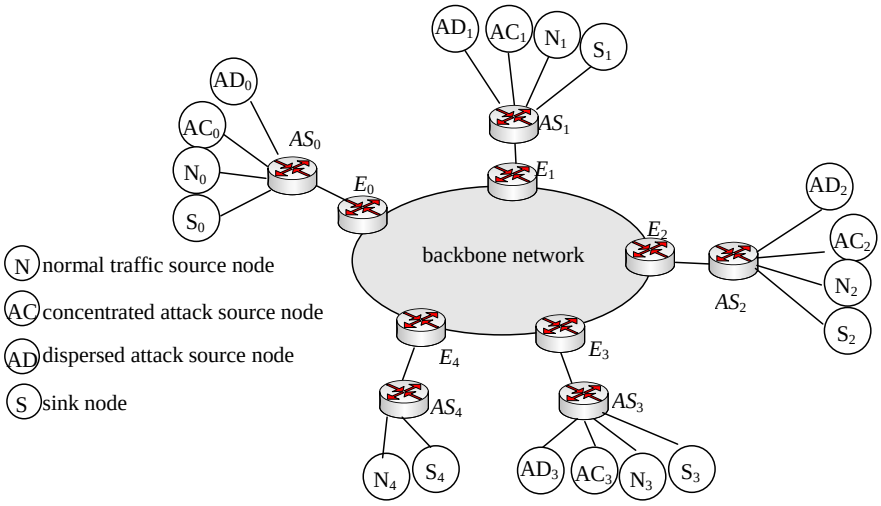


Fig. 5. Simulation network configuration

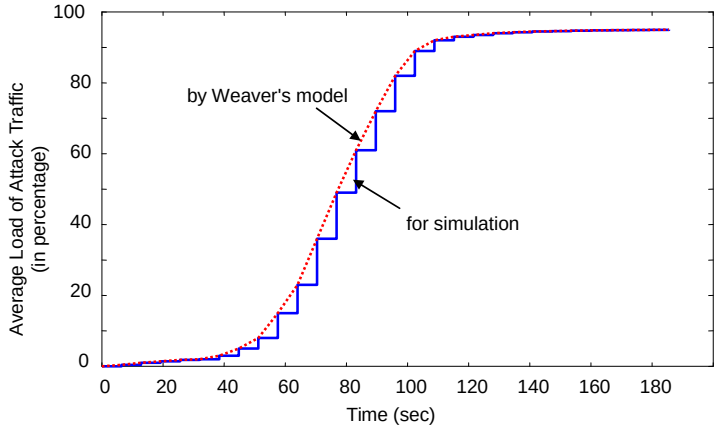
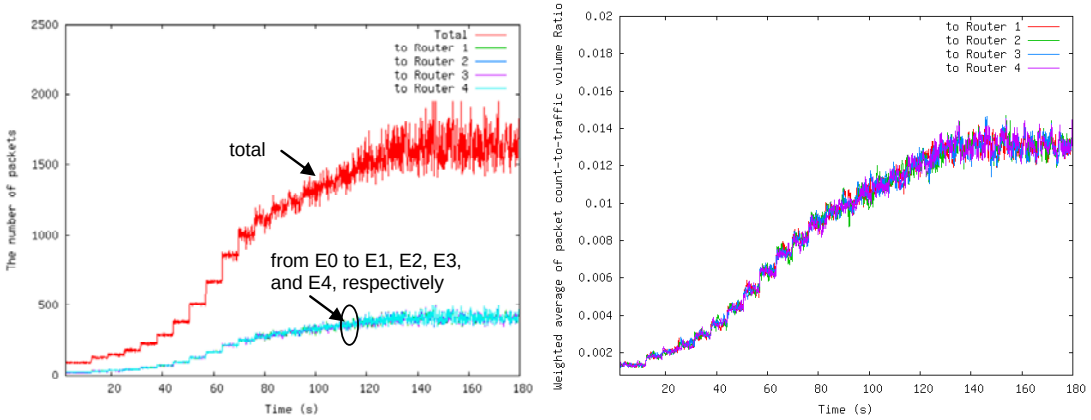


Fig. 6. Worm propagation model used in the simulation

3.2 Case I: Dispersed Attack

Traffic Generation: In a dispersed attack, a large number of attack packets are sent from an edge router with many infected hosts, to all other edge routers. In order to represent this case, we assumed that at the start of the simulation there was no infection, but after 10 seconds, an attack packet was generated only by node AD_0 in AS0, according to the worm propagation

model shown in **Fig. 6**. And, the packets generated by AD_0 were evenly distributed to all other edge routers E_1, E_2, E_3, E_4 via E_0. **Fig. 7** shows traffic characteristics generated by AD_0. In addition, as background traffic, normal traffic was generated by all nodes N_i (i=0,1,2,3,4) with the Hurst parameter of 0.9. And, the following values are used for the experiment: α_p = α_r =0.9, Alert_Threshold=5, Attack_Threshold = 10.



(a) packet counts (b) CVR

Fig. 7. Characteristics of attack traffic generated by AD_0

Experimental Results: **Fig. 8** shows attack state matrix **S**(0) maintained by GDS at the start of the simulation. Since there was no infection at the start time, all elements of **S**(0) were 0, which means the NORMAL state.

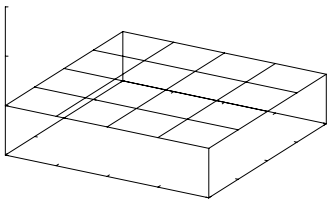
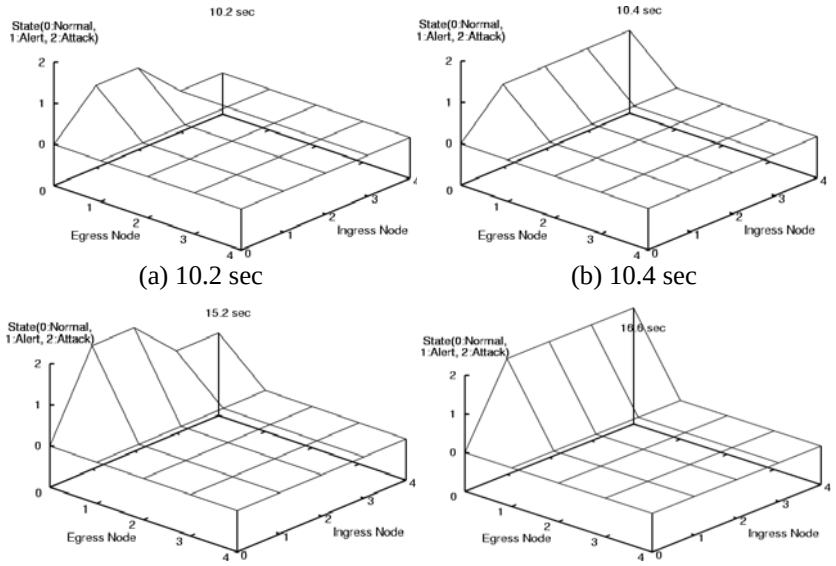


Fig. 8. Initial attack state matrix **S**(0) (no infection)

Fig. 9 shows the variation of **S**(m) monitored by GDS after the attack has begun. Note that each edge router sent the information on attack counter and states of its all outgoing links to GDS, then, with this information, GDS obtained the attack state matrix **S**(m), as shown in **Fig. 9**. In the simulation scenario, all attack packets were distributed from edge router E_0 to all other edge routers on the backbone network. Accordingly, as time advanced, only these states of s^{0j} (j=1,2,3,4) were changed to ATTACK via ALERT, while other states remained NORMAL. Finally, as shown in **Fig. 9**(d), from **S**(m), GDS could easily detect symptoms and corresponding patterns of the dispersed attack.

3.3 Case II: Concentrated Attack

Traffic Generation: In a concentrated attack, a large number of packets from many infected hosts are concentrated on a target host, to disable it. In order to show that the proposed method can detect the pattern of such an attack, we generated attack packets as follows. It was assumed that the target host is connected to E_2 via AS_2, and nodes AC_0, AC_1, AC_3 generate attack packets sent to the target host. In order to show the effectiveness of the proposed method, it was also assumed that no attack packets were generated by AS_4, because AS_4 was perfectly secure. The characteristics of attack traffic generated by AC nodes are as same as those by AD_0, as described in Section 3.2 and shown in **Fig. 7**. Unlike AD0 described in Section 3.2, however, all packets generated by AC nodes are sent only to edge router E_2. For convenience, to show worm propagation via the entire backbone network, the start times of worm propagation in AS0, AS1, AS2 and AS3 are set to 10, 30, 50, and 70 seconds, respectively, as shown in **Fig. 10**. There was no infected host until 10 seconds after the simulation had begun. We call each time interval between the start times of consecutive worm propagation a period, as shown in **Fig. 10**. As mentioned previously, the worm propagation model in each AS is in accordance with the model shown in **Fig. 6**. Note that since AS4 was secure, no infection appeared in AS4.



(a) 10.2 sec (b) 10.4 sec

(c) 15.2 sec (d) 15.5 sec

Fig. 9. Variation of **S**(m) after the attack has begun, at 10 sec. (Case I)

Experimental Results: The attack state matrix during Period-1 was the same as that shown in **Fig. 8**, since there was no attack packet in that period. In Period-2, attack packets were generated by AS_0 only, and sent to E_2 via E_0. Accordingly, $s^{0,2}$ of $\mathbf{S}(m)$ was changed to ATTACK via ALERT as shown in **Fig. 11**. This means that an attack was currently in progress, from E_0 to E_2. Likewise, in Period-3 and Period-5, because all attack packets from E_1 and E_3, respectively, were distributed to E_2, $s^{1,2}$ and $s^{3,2}$ were gradually changed to ATTACK, as shown in **Fig. 12** and **Fig. 13**, respectively. All attack packets generated by AC_2 were internally distributed to the target host in AS_2, and $s^{2,2}$ was not changed. In addition, since AS4 was secure, no attack packets were generated by E4, accordingly $s^{4,2}$ remained NORMAL. **Fig. 13**(b) shows the final attack state for our attack simulation

scenario, and it is clear that a host or network connected to E_2 was attacked by E_0, E_1, and E_3, which constitutes a concentrated attack.

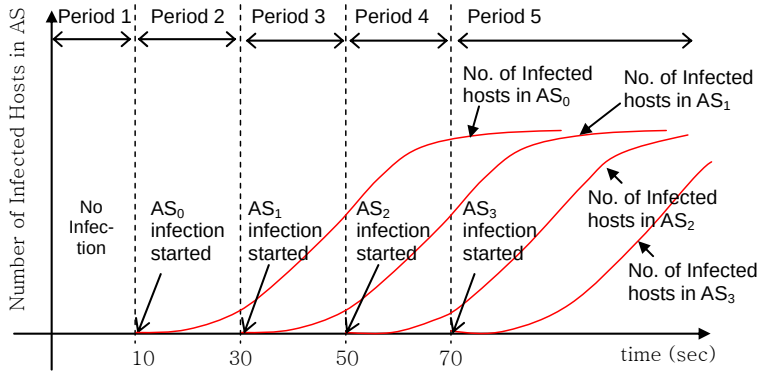
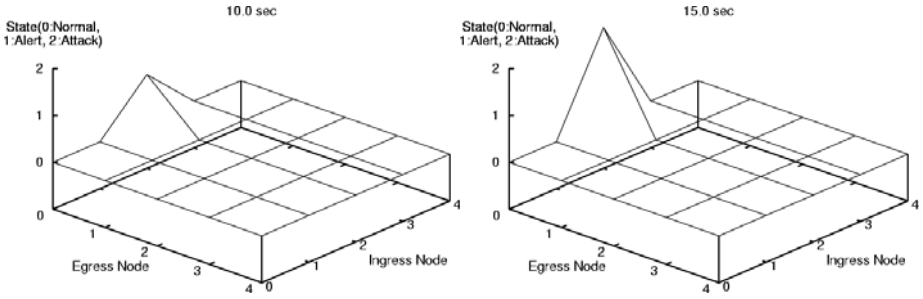
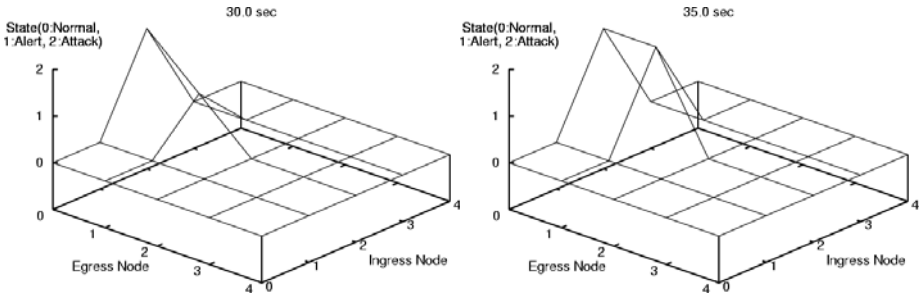


Fig. 10. Time frame for worm propagation in AS_0, AS_1, AS_2, AS_3. (No worm in AS_4)



(a) 10 sec (b) 15 sec

Fig. 11. Variation of $\mathbf{S}(m)$ during Period-2 (Case II)



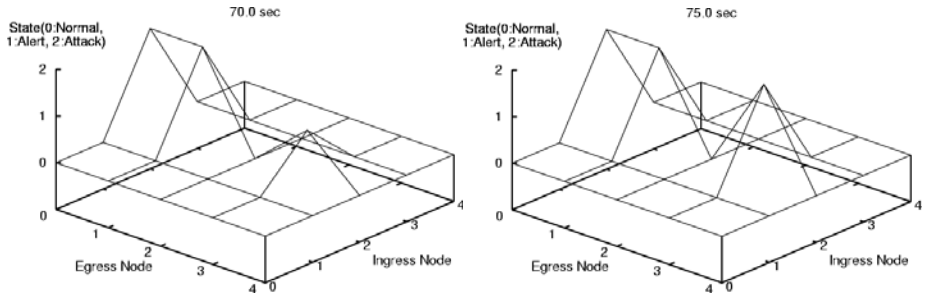
(a) 30 sec (b) 35 sec

Fig. 12. Variation of $\mathbf{S}(m)$ during Period-3 (Case II)

3.4 Case III: Hybrid of Dispersed and Concentrated Attacks

Traffic Generation: In this case, both dispersed and concentrated attack packets were generated simultaneously. Dispersed attack packets from each node AC were generated in the same manner as in Section 3.2. Unlike Section 3.2, however, nodes AC_1, AC_2, and AC_3 as well as AC0 generated dispersed attack packets, and these packets were evenly distributed to other edge routers. In addition, worm propagation for dispersed attacks from AS_0, AS_1, AS_2 and AS_3 started at different times according to the time frame, as shown in **Fig. 10**. On

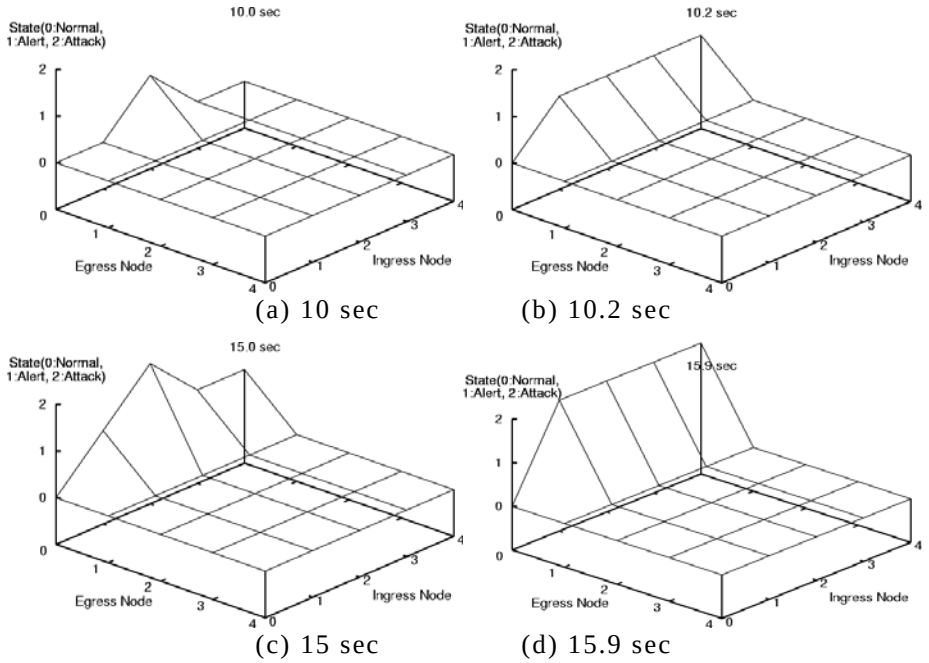
the other hand, concentrated attacks were performed in the same manner as in Section 3.3. And, it was also assumed that AS4 was secure, thus, no attack packet was generated from E_4.



(a) 70 sec (b) 75 sec

Fig. 13. Variation of $\mathbf{S}(m)$ during Period-5 (Case II)

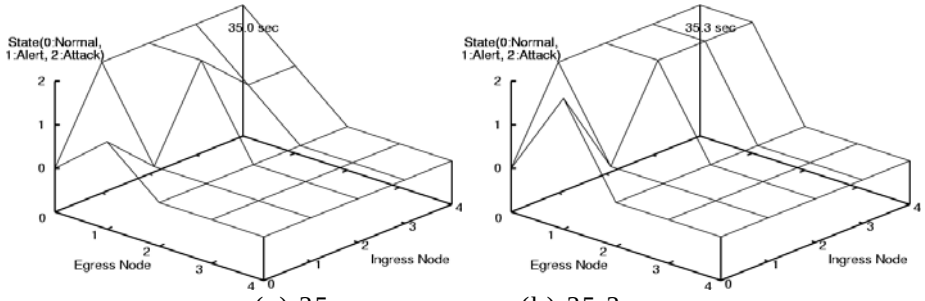
Experimental Results: **Fig. 14** shows the variation of $\mathbf{S}(m)$ in Period-2, in which only AS_0 is infected, hence, only dispersed attack packets from AD_0 were evenly distributed to all other edge routers via E_0, while concentrated attack packets from AC0 were sent only to E_2. Accordingly, first, $s^{0,2}$ changed to ATTACK via ALERT. Then, $s^{0,1}$, $s^{0,3}$, and $s^{0,4}$ also became ATTACK. This is because both concentrated and dispersed attacks were directed towards E_2, $s^{0,2}$ became ATTACK, first. And, since only dispersed attacks to E_1, E_3 and E_4 from E_0 were performed, $s^{0,1}$, $s^{0,3}$, and $s^{0,4}$ were changed to ATTACK, later. As time advanced via Period-3, Period-4 and Period-5, based on the pattern of variation of $\mathbf{S}(m)$ shown in **Fig. 15**, **Fig. 16**, and **Fig. 17**, it is clear that attacks are dispersed over the entire network according to the given time frame shown in **Fig. 10**.



(a) 10 sec (b) 10.2 sec

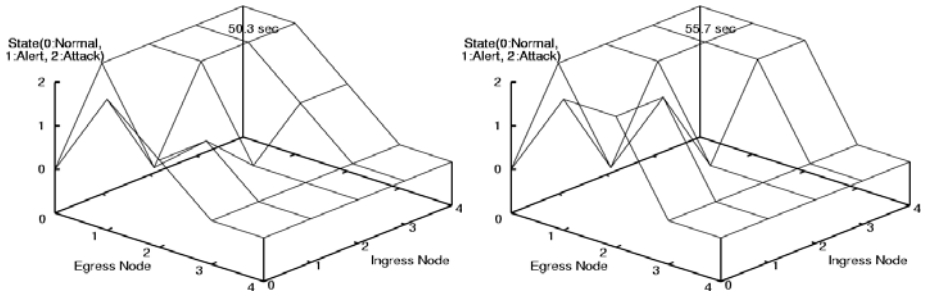
(c) 15 sec (d) 15.9 sec

Fig. 14. Variation of $\mathbf{S}(m)$ during Period-2 (Case III)



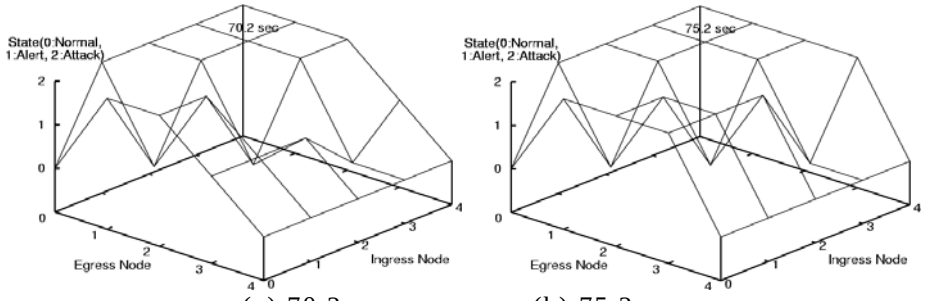
(a) 35 sec (b) 35.3 sec

Fig. 15. Variation of $\mathbf{S}(m)$ during Period-3 (Case III)



(a) 50.3 sec (b) 55.7 sec

Fig. 16. Variation of $\mathbf{S}(m)$ during Period-4 (Case III)



(a) 70.2 sec (b) 75.2 sec

Fig. 17. Variation of $\mathbf{S}(m)$ during Period-5 (Case III)

4. Conclusion

In this paper, we proposed an efficient and fast method for detecting patterns as well as symptoms of a distributed global-scale-network attack. We showed the effectiveness of the proposed method in experiments. As mentioned previously, conventional schemes are based on individual packet or flow levels, thus, they have very high computational complexity, and cannot be directly applied to detection of global-scale network attacks on high-speed Internet backbone networks. On the other hand, since the proposed method can detect attack symptoms at the aggregate traffic level, it can be implemented on high speed backbone links. In addition, GDS gathers the attack information from all edge routers, the exact pattern of global-scale network attack can be detected. With the detected attack pattern, the reaction

against the attack can be easily done, which requires further study. Though the network model with 5 routers are used to show the effectiveness of the proposed scheme, even within backbone network with a large set of routers and a huge amount of background traffic, the proposed scheme can work well because the detection process is applied to each pair between ingress and egress nodes at aggregate traffic level. As we mentioned, when attack traffic is added, the aggregate traffic pattern will be dramatically changed compared to the normal traffic environment, in which only background-like traffic are appeared.

In addition, since each backbone network has its own particular network management systems for monitoring and maintaining itself, we postulate that it is easy to construct a global defense infrastructure against network attacks, with the capability of our proposed method to detect patterns of distributed global-scale network attacks. Details on the construction of the global defense infrastructure require further study.

Acknowledgement

This research was supported by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Advancement (IITA-2008-C1090-0801-0003).

References

[1] D. Moore, C. Shannon, J. Brown, "Code-Red: a Case Study on the Spread and Victims of an Internet Worm," *Proc.2nd ACM Internet Measurement Workshop*, 2002.
[2] D. Moore, V. Paxon, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "The Spread of the Sapphire/Slammer Worm," http://www.cs.berkeley.edu/~nweaver/sapphire.
[3] R. Chang, "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A tutorial," *IEEE Communications Magazine*, pp. 42-51, Oct. 2002.
[4] M. Carmo, J. Maia, G. Junior, R. Holanda, "Using Statistical Discriminators and Cluster Analysis to P2P and Attack Traffic Monitoring," *IEEE LANOMS* 2007, Sept. 2007.
[5] H. Kim, J. Kim, S. Bahk, and I. Kang, "Fast Classification, Calibration, and Visualization of Network Attacks on Backbone Links," *ICOIN 2004*, LNCS, Vol. 3090, Feb. 2004.
[6] A. Chakrabarti and G. Manimaran, "Internet Infrastructure Security: A Taxonomy," *IEEE Networks*, Vol. 16, No. 6, pp.13-21, Nov./Dec. 2002.
[7] B. Roh, and S. Yoo, "A Novel Detection Methodology of Network Attack Symptoms At Aggregate Traffic Level on Highspeed Internet Backbone Links," *ICT 2004*, LNCS, Vol. 3124, Aug. 2004.
[8] V. Paxon, "Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic," *ACM SIGCOMM Computer Communication Review*, Vol. 27, Issue 5, Oct. 1997.
[9] The network simulator version-2, ns-2, http://www.isi.edu/nsnam/ns/.
[10] Nicholas C. Weaver. Warhol worms: The potential for very fast in plagues. http://www.cs.berkeley.edu/nweaver/warhol.html.
[11] Y. Xiang et al, "Detecting DDOS attack based on network self-semilarity," *IEE Proc. Comm.*, Vol.157, No. 3, June 2004.
[12] R. Kompella, S. Singh, and G. Varghese, "On Scalable Attack Detection in the Network," *IEEE/ACM Trans. on Networking*, vol. 15, No. 1, pp. 14-25, 2007.
[13] A.Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim, "A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods," *IEEE Transactions on Signal Processing*, vol. 54, No 9, pp. 3372 – 3382, 2006.

Sun Ho Kim received B.S. and M.S. degrees in Computer Science from Ajou University, Suwon, Korea, in 2004 and 2006, respectively. He is currently with the IS Research Center, Daewoo Electronics Co. Ltd., working as an Engineer. His research interests include multimedia systems and networking, and network security.

Byeong-hee Roh received a B.S. degree in Electronics Engineering from Hanyang University, Seoul, Korea, in 1987, and M.S. and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1989 and 1998, respectively. From 1989 to 1994, he was with Telecommunication Networks Laboratory, Korea Telecom, as a researcher. From February 1998 to March 2000, he worked with Samsung Electronics Co., Ltd., Korea, as a Senior Engineer. Since March 2000, he has been with the Graduate School of Information and Communication, Ajou University, Suwon, Korea, where he is currently an associate professor. During 2005, he was a visiting associate professor at Dept. of Computer Science, State University of New York, at Stony Brook, New York, USA. His research interests include mobile multimedia networking, network QoS, wireless sensor networks, network security, and military communications.