Chapter 10

# A SOURCE-END DEFENSE SYSTEM AGAINST DDOS ATTACKS

Fu-Yuan Lee
*Department of Computer Science and Information Engineering,*
*National Chiao Tung University, Hsinchu, Taiwan 300*

Shiuhpyng Shieh
*Department of Computer Science and Information Engineering,*
*National Chiao Tung University, Hsinchu, Taiwan 300*
*URL: http://dsns.csie.nctu.edu.tw/ssp*
*E-mail:ssp@csie.nctu.edu.tw*

Jui-Ting Shieh
*Department of Computer Science and Information Engineering,*
*National Chiao Tung University, Hsinchu, Taiwan 300*

Sheng-Hsuan Wang
*Department of Computer Science and Information Engineering,*
*National Chiao Tung University, Hsinchu, Taiwan 300*

**Abstract** In this paper, a DDoS defense scheme is proposed to deploy in routers serving as the default gateways of sub-networks. Each router is configured with the set of IP addresses belonging to monitored sub-networks. By monitoring two-way connections between the policed set of IP addresses and the rest of the Internet, our approach can effectively identify malicious network flows constituting DDoS attacks, and consequently restrict attack traffics with rate-limiting techniques. Current source-end DDoS defense scheme cannot accurately distinguish between network congestion caused by a DDoS attack and that caused by regular events. Under some circumstances, both false positive and false negative can be high, and this reduces the effectiveness of the defense mechanism. To improve the effectiveness, new DDoS detection algorithms are presented in this paper to complement, rather than replace existing source-end DDoS defense systems. The design of the proposed detection algorithm is based on three essential characteristics of DDoS attacks: *distribution*, *congestion*, and *continuity*. With the three characteristics, the proposed detection algorithm significantly improves detection accuracy, and at the same time reduces both false positive and false negative against DDoS attacks.

**Keywords:** information warfare , DoS/DDoS attacks , source-end defense

## 1. Introduction

Current Internet infrastructure is vulnerable to network attacks, and particularly, many security incidents have shown that the Internet is weak against distributed denial-of-service (DDoS) attacks. In general, a DDoS attack is accomplished by persistently overloading critical resources of the target Internet service so as to completely disable or degrade the service over an extended period of time. Such resource overloading can be achieved in several ways. First, an Internet service can be overloaded by a large number of service requests issued in a short period of time. As a result, legitimate service requests may be dropped due to insufficient resource such as computation power or memory space. Second, attackers can overload a network link near the target, and consequently, all flows traverse through the link will experience significant degrade of service quality .

To generate a great amount of traffic or service requests, attackers may first compromise a large number of computer systems. This can be easily accomplished due to the large number of insecure computer systems and the set of easily acquired and deployed exploit programs, such as Tribal Flood Network (TFN), TFN2K and Trinoo . On the other hand, detecting or preventing a DDoS attack is relatively much harder. The lack of explicit attack signatures/patterns makes it extremely difficult to distinguish attacks from legitimate traffic. Furthermore, the anonymous nature of IP protocol allows the attackers to disguise the attack origins, and thus makes it hard to detect the sources of DDoS attacks. These difficulties make the construction of an effective DDoS defense mechanism become a very challenging problem.

Issues for defending DDoS attacks have been extensively investigated in recent years, and several defense mechanisms have been presented in the literature. The deployment of these schemes can be categorized into three classes. The first class of schemes [Shaprio and Hardy, 2002, T. Aura and Leiwo, 2001, Mirkovic et al., 2002a, Juels and Brainard, 1999, Wang and Reiter, 2003, Leiwo et al., 2000, Mann et al., 2000, Feinstein et al., 2003, NFR, , Net, , Roesch, 1999] involve detecting and preventing a DDoS attack at the victim network. In this context, the term *victim network* indicates that the installed DDoS defense systems are used to protect a limited set of computers. These defense systems are generally deployed at end host systems or at routers which are able to examine and control communications between protected hosts/networks and the rest of the Internet. Placing defense mechanisms at the victim networks can be easier for detecting DDoS attacks. Since the DDoS traffic is aggregated toward the victim, a burst of network traffic would be the signal of a DDoS attack. However, from the network's perspective, filtering out DDoS attack packets at the victim side is ineffective because the attack flows may cause network congestion and waste valuable computation power of the routers along the path they traversed.

To improve the effectiveness of packet filtering, schemes in the second class attempt to construct DDoS defense lines toward attack sources. To achieve this objective, several schemes have been proposed, and these schemes can be further divided into two types. First, DDoS attacks are detected by DDoS defense systems installed in victim networks , and subsequently Internet core routers in the attack paths are requested to filter out attack traffic according to filtering criteria specified by downstream routers or DDoS defense systems [Ferguson, 1998, Park and Lee, 2001, Sung and X, 2002, Ioannidis and Bellovin, 2002, man, , Mahajan et al., 2002]. Second, traceback techniques [Savage et al., 2001, Savage et al., 2000, Dean et al., 2002, Song and Perrig, 2001] are utilized to identify attack sources and then legal sanctions can be performed to deter DDoS attacks. Schemes in the second class can partially avoid attack flows blending with legitimate flows and consequently somewhat reducc thc complexity for distinguishing from attack traffic and legitimate traffic. Furthermore, it may also reduce to certain degree possible network congestion caused by attack flows. However, owing to the cooperative and distributed nature, these schemes heavily rely on cooperation among Internet core routers . This would generally incur high deployment cost. Routers need to be upgraded to support packet filtering in high speeds. Coordination among ISPs may also bring unpredictable difficulties. In addition to the deployment costs, the way that core routers drop packets according to the information passed from victim-end systems may implicitly bring other substantial cost and security breaches. For instance, an Internet-wide authentication framework is needed; otherwise, core routers may accept instructions from malicious attackers and drop legitimate traffic. Therefore, to secure and authenticate communications between core routers and victim-end systems in large networks may bring infeasible high overhead. Thus, schemes in the second class are generally inadequate to be deployed in large networks such as the Internet.

Similar to the victim-end approaches, the third class of schemes involve deploying DDoS defense mechanisms at default gateways. The major difference is that, DDoS defense mechanisms in the third class are used to police hosts in the monitored networks from participating in DDoS attacks rather than protecting them. This approach can ideally prevent attack traffic from entering the Internet. In other words, DDoS attack flows are contained in their sources.

It subsequently avoids attack flows blending with legitimate flows, and as a result network congestion can be significantly reduced. Furthermore, since the degree of flow aggregation is relatively low and routers closer to source networks are likely to relay less traffic than core routers, it is possible to use sophisticated detection strategies which may require more computation power and system resources.

Although the idea of defending DDoS at sources is attractive, detecting the occurrence of a DDoS attack at the attack sources is very difficult [Chang, 2002]. The main difficulty arises from the insignificant aggregate of attack traffic which can be observed in attack sources. Other criteria for identifying DDoS attacks must be discovered. For instance, in the D-WARD system proposed by *Mirkovic et al* [Mirkovic et al., 2002b], network congestion measured by the ratio of incoming and outgoing packets of network connections is used to judge whether the monitored flow is part of a DDoS attack or not. By monitoring the changes of the ratio, D-WARD would be able to detect a DDoS attack that has already disable the victim. However, it is hard for D-WARD to distinguish a DDoS attack from network congestion caused by other events. On one hand, D-WARD can mis-classified a flow if the ratio of flow is high in its normal operation. On the other hand, D-WARD is weak in detecting low-rate attacks . In other words, a well-designed attack script can avoid being detected by D-WARD by carefully control the congestion caused by the attack.

To address the weakness of D-WARD, in this paper, we propose a source-end DDoS detection algorithm and an attack response mechanism, where the former can accurately identify an ongoing DDoS attack and the latter can effectively limit attack traffic in source networks. The proposed detection and response algorithms are built upon the system architecture originally proposed in D-WARD. Our proposal focuses on reducing both false positive and false negative on detecting two-way connections. That is, the proposed scheme attempts to complement, rather than replace the D-WARD system.

The design of proposed scheme is based on the observation of three essential characteristics of a DDoS attack: *distribution*, *congestion*, and *continuity*. *Distribution* refers to the spreading of attack traffic from a large number of compromised hosts. *Congestion* refers to the inherent consequence of a DDoS attack. That is, an increasing packet loss rate observed in a monitored network flow would generally represent a signal of a DDoS attack . Third, *continuity* directs to the observation that network congestion caused by DDoS attacks usually lasts for an extended period of time. Combining the above three criteria allows us to differentiate a DDoS attack from a typical network congestion caused by other events. Based on the three characteristics, a new DDoS defense mechanism is proposed. Since the proposed mechanism is built upon D-WARD architecture, the proposed DDoS defense mechanism is also deployed at routers serving as the default gateways. Online traffic statistics, in terms of distribution, congestion, and continuity, are gathered and compared against previous statistics derived from normal traffic. In this way, malicious network flows are identified and rate-limited. Rate limits are dynamically adjusted according to the behavior of malicious network flows. On one hand, dynamic adjustment allows a misclassified network flow to regain network bandwidth when the flow shows compliance to legitimate flow model. On the other hand, since attack scripts has no way to distinguish the effect of rate-limiting from that of a successful DDoS attack, dynamic adjustment helps restrain malicious flows.

This paper is organized as follows. Section 2 gives an review of the D-WARD system. The proposed source-end DDoS defense scheme is presented in Section 3, including its detection and rate-limiting mechanism. Section 4 describes an implementation of the proposed scheme and presents several experiments on estimating the effectiveness. Subsequently, we summarize and conclude our findings in Section 5.

## 2. Review of D-WARD

In this section, we briefly review D-WARD system, including system architecture, detection algorithm, and attack response algorithm.

### 2.1 System Architecture

From the architectural point of view, D-WARD consists of a *observation component* and a *throttling component.* The observation component examines all communications between the set of IP addresses in the monitored network and the external IP addresses, and then computes on-line traffic statistics. Note that, in D-WARD, time are divided into a set of uniform intervals, called *observation period* , which serves as a unit time frame to compute traffic statistics. In each observation period, new traffic statistics are compared against past statistics derived from normal traffic. Network flows are classified according to the comparison results. Moreover, the statistics and comparison results are then passed to the throttling component which generates rate-limiting rules based on the behavior of the monitored network flows.

Fig. 10.1 shows a possible deployment of D-WARD. As depicted in the figure, D-WARD is a separate unit that acquires traffic from the default gateway and feeds the gateway with rate-limiting rules.

### 2.2 Attack Detection

In D-WARD, the aggregate traffic between monitored addresses and a correspondent host is defined as a *flow.* A flow is considered two-way if its data flow comprises packets originating from the sender and corresponding reply from the peer. TCP connections and several types of ICMP messages, such as

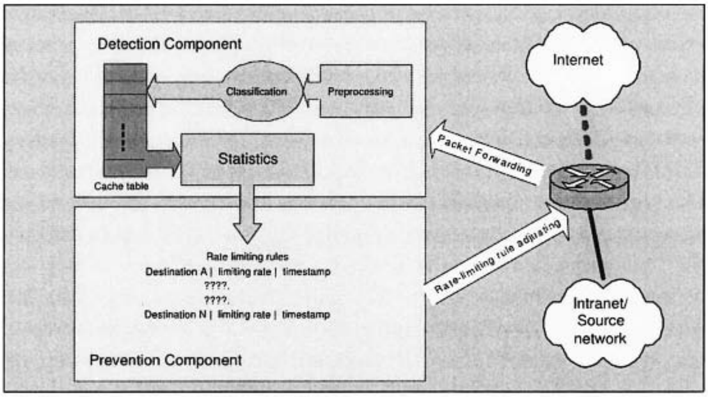

*Figure 10.1.* An example of the deployment of D-WARD

"timestamp" and "echo", are typical examples of two-way flows. On the other hand, a flow is considered one-way (or uni-directional) if its data flow does not require reply messages in its normal operation. Traffics based on UDP protocol are examples of one-way flows.

For TCP flows, D-WARD defines a threshold that specifies the maximum allowed ratio of the number of packet sent to and received from in a flow. Notice that in the following context in this paper, the ratio of the number of packet sent and received in a flow is refereed to as the *O/I* of the flow. Then, for TCP flows, whenever the O/I value of a flow breaches a pre-defined threshold, $TCP_{rto}$, the flow is classified as a DDoS attack flow. Similarly, for ICMP-based two-way flows, $ICMP_{rto}$ is used to define the maximum O/I value of an ICMP flow. In D-WARD's experimental settings, $TCP_{rto}$ is set to 3 and $ICMP_{rto}$ is set to 1.1.

For a one-way flow, D-WARD defines three thresholds: an upper bound on the number of allowed monitored hosts issuing one-way connections to a correspondent host, a lower bound on the number of allowed packets in each one-way connection, and a maximum allowed sending rate. Whenever any of the three threshold is breached, the flow is considered attack. In D-WARD's experiment, the number of host in the same UDP flow must be smaller than 100. There must be at least one packet in each connection, and the maximum allowed sending rate is 10MBps.

In D-WARD, a flow is classified as *normal*, *suspicious* or *attack* according to the comparison on the statistics derived from normal flows and the currently gathered statistics. If the statistics of a monitored flow does not consistent with normal model defined by thresholds mentioned above, the flow is classified as attack. If a flow that was ever classified as attack and the current comparison indicates compliance with normal flow model, it is classified as suspicious. Finally, if a flow is always compliant with normal flow model, it is classified as normal.

## 2.3 Attack Response

According to the comparison result passed from the observation component, the throttling component specifies allowed sending rates for monitored flows. D-WARD utilizes a flow control mechanism which is similar to the congestion control mechanism of TCP protocol. The sending rate is exponentially decreased in the first phase of attack response. Then, if further comparison indicates compliance with the normal flow model, a rate-limited flow can regain its bandwidth after the slow recovery and fast recovery process. On the other hand, a rate-limited flow can be more severely restrained if it does not comply with the rate limit and attempts to persistently rebel against the limited sending rate.

## 3. Proposed System

In this section, we first show that there are normal TCP flows with its $O/I$ value which is greater than the threshold defined by D-WARD. This indicates that D-WARD would classify these TCP flows as attack while they are in their normal operations. This problem cannot be solved by using a sufficient large threshold since it will increase the false negative. Specifically, low rate attacks will not be detected. To cope with the problem, a new algorithm for detecting and limiting TCP-based DDoS attacks are presented herein.

It is worthy to notice that although DDoS attacks may take many different forms, it is reported [Chang, 2002, Mahajan et al., 2002, Moore et al., 2001] that over 94% of DDoS attacks use TCP. Thus, the scheme presented in this paper may help defend against a majority of DDoS attacks. As to the detection of DDoS attacks based on of one-way flows, we suggest using the algorithm presented in D-WARD at current stage, but further enhancement is possible for the future work.

## 3.1 Basic idea of the proposed scheme

As mentioned above, D-WARD classifies a TCP flow as an attack flow if the O/I value of the flow is greater than $TCP_{rto}$. (Recall that, in D-WARD, this threshold is set to 3.) This approach suffers from the difficulty in determining an appropriate value for $TCP_{rto}$. It is because the O/I value of a TCP flow heavily depends on the implementation of TCP/IP protocol stack of the peers, and other factors such as round trip time and network congestion . This would result in a wide range of O/I values. For instance, Fig. 10.2 shows the average O/I values of TCP flows in a typical network consisting of 30 personal computers. Operating systems installed in these computers include Windows 2000, Windows XP, FreeBSD, and Linux. As shown in the figure, there are flows with O/I values which are greater than 3. (The highest average O/I value is 3.68. It is observed in a flow consisting of only one FTP data connection.)

The observation motivates a new algorithm for detecting TCP-based DDoS attacks. The proposed algorithm exploits three essential characteristics of DDoS attacks, namely distribution, congestion and continuity, to detect the presence of DDoS attacks. First, distribution refers to the observation that DDoS attack scripts will normally infect as many insecure computer systems as possible so as to amplify the power of the DDoS attack. Therefore, in the monitored networks, if there is an increasing number of hosts attempting to send traffic to a destination host, a DDoS attack may just have been started. The statistics on the number of hosts sending packets to the same target will provide a valuable criterion for judging whether there is a DDoS attack or not. Second, DDoS attack usually lead to high packet loss rate toward the victim. Since monitoring packet loss rates of individual TCP flows would incur infeasible high cost, similar to D-WARD, the packet loss rate of a flow is abstractly represented as the O/I value of the flow. Third, continuity reflects to the observation that a DDoS attack usually lasts for an extended period of time. As we shall see later, this makes us be able to distinguish network congestion caused by DDoS attacks and other network events.

By taking advantage of the three DDoS characteristics, the proposed detection algorithm can classify TCP flows more precisely. In the proposed scheme, there are two phases: initialization phase and detection phase. In the initialization phase, the proposed scheme constructs initial profiles for TCP flows according to the past traffic in the flows. In the profile database, each profile specifically represents the legitimate flow model of a TCP flow. Then, in the detection phase, traffic statistics are then compared with profiles. Profiles are

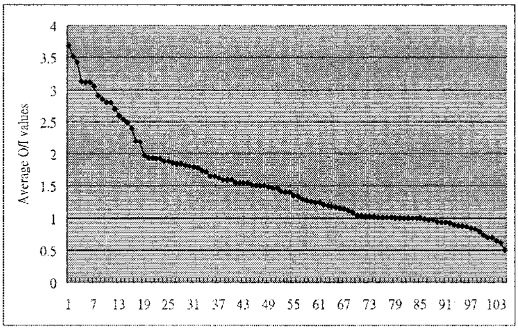

*Figure 10.2.* Average $O/I$ values

dynamically adjusted to reflect the current behavior of monitored flows In this way, different thresholds can be used to classify different TCP flows, and thus the efficiency of the detection algorithm can be effectively improved.

## 3.2 Detection Phase

In the proposed scheme, characteristics and thresholds of a flow are derived from the past traffic of the monitored flow. For each TCP flow, its traffic statistics computed from the current observation period are compared against the legitimate flow model defined by the profile of the flow so as to determine whether it is malicious or not. To better understand the proposed legitimate flow model, some notations are introduced as follows.

First, a two-way flow $f$ is a collection of connections, and each connection is associated with a pair of IP addresses – an IP address in the set of monitored addresses and an IP address of the correspondent hosts. The former is referred to as *initial address* and the latter is *terminal addresses*. The number of distinct initial addresses in a flow $f$ is denoted as $S_f$. For a connection $c$, $n_c$ denotes the ratio of the number of packets originated from the initial address and received from the terminal address in one observation period in connection $c$. Then, $n_f$ represents the average of the O/I value of all connections in flow $f$.

Furthermore, there are two threshold values, $N_f$ and $T_f$, which help determine the malicious level of a monitored flow. $N_f$ represents the mini threshold of a flow $f$. If $n_f \leq N_f$, then $f$ is considered as a normal flow. $T_f$ denotes the maximum allowed $n_f$. If $n_f \geq T_f$, then $f$ is classified as an attack flow. If $N_f \leq n_f \leq T_f$, then further traffic statistics must be examined to determine the malicious level of the flow.

Then, the level of congestion and distribution can be quantified. Consider a flow $f$ with $N_f \leq n_f \leq T_f$, the level of congestion of $f$ refers to $(n_f - N_f)/(T_f - N_f)$. In this expression, we can clearly see that if the packet loss rate of the flow approaches $T_f$, the value of the expression will approach 1. On the other hand, if $n_f$ approaches $N_f$, the value will approach 0. Next, the level of distribution is quantified as $S_f/C$, where $C$ denotes a configuration parameter obtained from the past behavior of the monitored network (We will describe how to obtain this parameter later). Then, the level of congestion and distribution are combined and used to generate a value representing the malicious level of a monitored flow. Herein, the malicious level is denoted $\alpha$ and computed as follows. (In Eq. 10.1, $\lambda$ is a number between 0 and 1, that is, $0 < \lambda < 1$. It is used to restrict the saturation of $\alpha$ between 0 and 1.)

$$\alpha = \frac{1-\lambda}{\lambda} * \sum_{i=1}^{\lfloor S_f/C \rfloor} (\lambda * \frac{n_f - N_f}{T_f - N_f})^i \qquad (10.1)$$

It is worthy to note that $\alpha$ has two important characteristics. First, it is clear that $\alpha$ increases as $n_f$ increases. In other words, if the packet loss rate of a monitored flow $f$ gets higher, $n_f$ will increase and consequently $\alpha$ increases. Second, $\alpha$ increases along with $S_f$ even if $n_f$ remains the same. This feature is especially useful in detecting DDoS attacks launched by attack programs which spoof source IP addresses. The $\alpha$ value will close to 0 when the monitored flow is in its normal operation. On the other hand, it will increase significantly when both the level of congestion and the level of distribution increases.

Although a surge of $\alpha$ value may indicate an DDoS attack that results in an abnormal increase in the packet loss rate or in the number of initial addresses in a flow, the $\alpha$ value can also go up due to regular network congestion. Nevertheless, the period of time the $\alpha$ value arises becomes a significant difference between the two causes. That is, normal network applications will stop sending more packets to a highly congested destination host after several attempts while DDoS attack scripts continually flush the victim for an extended period of time. With this observation, we can effectively distinguish a DDoS attack from a conventional network congestion by examining the length of time that DDoS attack signal lasts. This concept is implemented as follows. Consider a TCP flow $f$. $\alpha_f$ is a threshold that represents the maximum allowed $\alpha$ derived from the current network traffic. Once the threshold $\alpha_f$ is breached consecutively for $t_f$ observation periods, $f$ is considered a DDoS attack flow.

According to the proposed DDoS detection strategy, a network flow $f$ can be classified into four types: *normal*, *suspicious*, *attack*, and *transient*. The transition of these types are depicted in Fig. 10.3. In brief, $f$ is classified as a suspicious flow if $\alpha \geq \alpha_f$, where $\alpha$ is derived from the traffic in the current observation period. If $\alpha_f$ is breached for consecutive $t_f$ observation periods, $f$ is classified as an attack flow, and rate limiting techniques are applied to $f$. Once the traffic statistics of $f$ shows compliance with legitimate flow model, i.e. $\alpha \leq \alpha_f$, for consecutive *PenaltyPeriod* observation periods, $f$ is then classified as transient. For transient flows, rate limiting rules are carefully removed. When the allowed bandwidth of $f$ reaches *MaxRate*, $f$ is classified as a normal flow. Algorithm 1 shows pseudo code of the proposed detection algorithm.

In addition to the determination of the malicious level of monitored flows, it is desirable to update the thresholds for classifying network flows. This allows our scheme to learn the changing behavior of normal traffic, and dynamically adjust the thresholds according the current traffic statistics of monitored flows. For the adjustment of thresholds, attack traffic will be filtered out, and only traffic of a normal flow will be used to update thresholds. In this way, thresholds will not be polluted by attack traffic.

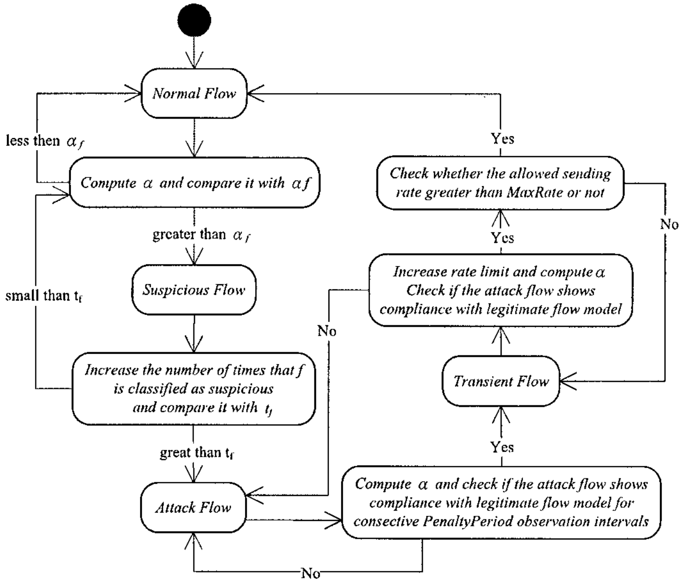
Normal Flow
less then $\alpha_f$
Compute $\alpha$ and compare it with $\alpha f$
greater than $\alpha_f$
Suspicious Flow
small than $t_f$
Increase the number of times that f is classified as suspicious and compare it with $t_f$
great than $t_f$
Attack Flow
Yes
Check whether the allowed sending rate greater than MaxRate or not
No
Yes
Increase rate limit and compute $\alpha$ Check if the attack flow shows compliance with legitimate flow model
No
Transient Flow
Yes
Compute $\alpha$ and check if the attack flow shows compliance with legitimate flow model for consective PenaltyPeriod observation intervals
No

*Figure 10.3.* Classification of Traffic Flow

Updating a threshold is accomplished by feeding back the current traffic statistics. As we shall see shortly in this section, given a volume of historical traffic of the monitored flow $f$, we can derive $T_f$, and $N_f$ for the monitored flow. In fact, by the same procedure, we can compute traffic statistics for each observation period. Assume that flow $f$ is classified as normal in period $i$. Let $\widehat{T_{f,i}}$ denote the maximum allowed O/I value derived from the traffic volume in observation period $i$, and $T_{f,i}$ denote the same threshold used to classify flow $f$ in period $i$ (i.e., $T_f$ of period $i$). Then, $T_{f,i+1}$ of the next observation period $(i+1)$ can be computed in the same way:

$$T_{f,i+1} = \beta * \widehat{T_{f,i}} + (1 - \beta) * T_{f,i}$$

In the similar fashion, $N_f$ can be updated as follows.

$$N_{f,i+1} = \beta * \widehat{N_{f,i}} + (1 - \beta) * N_{f,i}$$

where $\beta$ is a configurable parameter ranging from zero to one. At present, we suggest $\beta$ to be $1/2$. However, its best value for a particular type of networks heavily relates to the variation of monitored network traffic, and may need further investigation.

## 3.3 Initialization Phase

It is clear that the settings of thresholds play an important role for the classification. As mentioned above, the learning phase is to compute these thresholds according to normal traffic of the monitored flows. It is worthy to note that the traffic used in the learning phase must be carefully examined and cannot contain attack traffic. Otherwise, the statistics derived will be incorrect and cannot be used to detect attacks. Due to this concern, Currently in the proposed scheme we perform off-line learning. That is, after all the thresholds are determined according to the historical traffic, the learning phase halts. This helps prevent our scheme from being polluted by on-line attack traffic. Next, the configurations of the threshold are described.

Consider parameters used in Eq. 10.1, i.e. $T_f$, $N_f$ and $C$. Recall that $T_f$ stands for the maximum allowed threshold of the $O/I$ value of a monitored flow and $N_f$ represents a mini threshold of the $O/I$ value. Assume that the trail of historical network traffic is available which does not contain attacks. The traffic is partitioned into volumes in terms of observation periods, and then, thresholds are derived from the partitioned traffic volumes. In our scheme, the $O/I$ value of each observation is measured. Then, we set $T_f = 2 * OI_{f,max}$ and $N_f = OI_{f,avg}$, where $OI_{f,max}$ denotes the maximum observed $O/I$ value of flow $f$ derived from historical traffic data and $OI_{f,avg}$ denotes the average $O/I$ value. Next, Let $C$ be the maximum number of distinct initial addresses in a flow during one observation period.

**Algorithm 1** Detection Procedure

```
loop
  Collect IP packets received in one observation period.
  for each packet originating from monitored IP addresses do
    if Protocol = TCP then
      Classify the packet into a flow according to the destination IP address.
    end if
  end for
  calculate the O/I values of monitored TCP connections.
  for each flow (let the current flow be denoted as $f$) do
    if $N_f \leq n_f \leq T_f$ then
      compute $\alpha$ for flow $f$.
      if $\alpha \geq \alpha_f$ then
        increase the number of time that $f$ is classified as suspicious.
        if the number of times that $f$ is classified as suspicious $\geq t_f$ then
          generate a DDoS attack alert and classify the flow as attack.
          perform rate-limiting.
        end if
      else
        reset the number of times that $f$ is classified as suspicious.
      end if
    else if $n_f \geq T_f$ then
      Set $\alpha$ to 1, generate a DDoS attack alert and classify the flow as attack.
      perform rate-limiting
    end if
  end for
end loop
```

After $T_f$, $N_f$ and $C$ are configured, we can then compute a set of $\alpha$ values, one for each observation period. Then, we can set $\alpha_f$ to the average of the set of $\alpha$ values and subsequently set $t_f$ to be the maximum consecutive number of times that $\alpha_f$ is breached in the set.

## 3.4 Rate Limiting

In addition to detecting DDoS attacks, rate limiting is another component of the proposed scheme. In our approach, if a flow $f$ is classified as attack, rate limiting technique will be applied to the flow in order to limit malicious traffic to a manageable level. One important design principle of our rate limiting strategy is that the rate limit applied to a malicious flow must reflect to current behavior of the flow. In this way, we can further restrict an ill-behaviored flow when it continually violates the legitimate flow model. From this point of view, the $\alpha$ value, which represents the malicious level of the monitored flow, serves as a rate limiting parameter. For the first time a flow is classified as an attack flow, the correspondent rate limit is:

$$rl = R * (1 - \alpha) \tag{10.2}$$

In Eq. 10.2, $rl$ denotes the rate limit and $R$ denotes the sending rate of the monitored for in the current observation interval. In the following observation periods, if the malicious flow does not show compliance to the legitimate flow model, it will be restrict further, according to the following formula:

$$rl_{new} = \min(rl_{old}, R) * (1 - \alpha) * \frac{P_s}{P_s + P_{drop}} \tag{10.3}$$

In Eq. 10.3, $rl_{new}$ denotes a new rate limit to be applied on the malicious flow. $rl_{old}$ denotes the rate-limit applied on the flow in previous observation interval. $R$ represents the realized sending rate in previous observation interval. $P_s$ is the total number of packets sent in the flow and $P_{drop}$ is the total number of packets dropped because of the imposed rate limit.

In this way, flows that are part of DDoS attacks would be quickly restricted to a very low rate since the attack scripts would persistently send attack packets to the victim. Consequently, the fraction $(P_s)/(P_s+P_{drop})$ would become very low quickly.

Next, consider the case that the rate limited flow is mis-classified. In this case, TCP-based network applications will stop sending packets when the network is highly congested. Since the TCP/IP protocol will actively slow down the sending rate, the flow will show compliance with the legitimate flow model. In our approach, whenever an attack flow is compliant with the normal flow model for consecutive *PenaltyPeriod* observation periods, the flow is considered a transient flow and the recovery process begins. In the recovery process, rate limit are carefully removed according to the following equation:

$$rl_{new} = rl_{old} * \frac{1}{\alpha} * \frac{P_s}{P_s + P_{drop}} \tag{10.4}$$

In Eq. 10.4, it is clear that the speed of recovery is controlled by $\alpha$ and $P_s/(P_s + P_{drop})$. Both reflect the current behavior of the monitored flow. When the rate limit reaches *MaxRate*, a transient flow is classified as a normal flow, and rate limit is completely removed.

## 4. Performance Evaluation

To evaluate the performance of the proposed scheme, we implemented both prototypes of D-WARD and our approach on a machine which runs the FreeBSD operating system. In our experiment, two types of DDoS attacks are conducted: TCP SYN flooding attack and link overloading attack. In the TCP SYN flooding attack, each attack agent floods the victim with TCP SYN packet at the maximum rate of 100KBps. In this experiment, we will show that attacks detected by D-WARD can also be detected by our approach. Even further, our scheme can detect the attacks earlier than D-WARD. Next, In the link overloading attack, each agents sends the victim at the maximum rate of 100KBps. The link bandwidth of the victim is restricted to 500KBps. This is accomplished by using Dummynet [Rizzo, 1997]. (there are in total 10 agents) In this experiment, we will show that our approach can detect attacks which cannot detected by D-WARD. For both types of attacks, we replicate the four attack scenarios tested in D-WARD. That is, constant rate attack , pulsing attack , increasing rate attack and gradual pulsing attack.

### 4.1 Experimental Results

Fig. 10.4, 10.5, 10.6, and 10.7 show the experimental results of TCP SYN attack. The x-axis denotes time measured in second and the y-axis stands for attack bandwidth measured in KB per second. The line with "x" symbols represents the attack bandwidth generated by attack agents. The line with triangle symbols represents attack bandwidth going through D-WARD, and the line with square symbols denotes the attack bandwidth passing by the proposed scheme. According to the figure, our scheme can detect the attack earlier than D-WARD. This makes our scheme more effective than D-WARD mainly because the thresholds used in our scheme are continually adjusted and derived from the past behavior of the monitored flows.

Next, we examine the experimental results of link overloading attacks. In this experiment, by controlling the attack sending rate, the O/I value of the attack flow only reaches 2, smaller than threshold value 3 used in D-WARD.

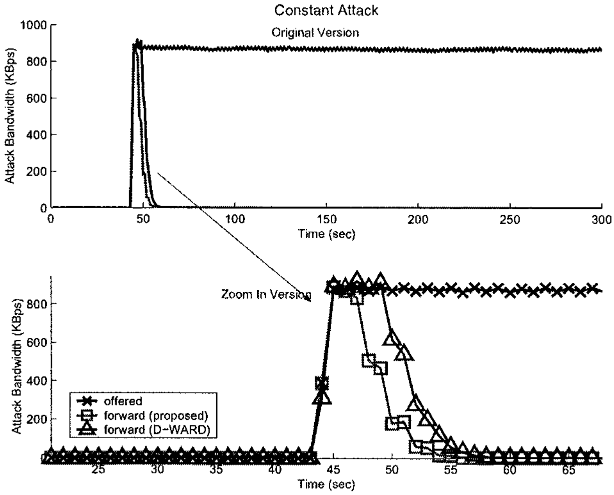

*Figure 10.4.* Constant SYNC attack.

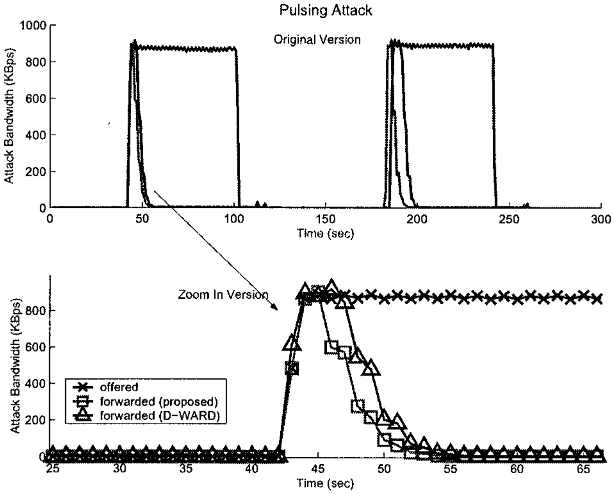

*Figure 10.5.* Pulsing SYNC attack.

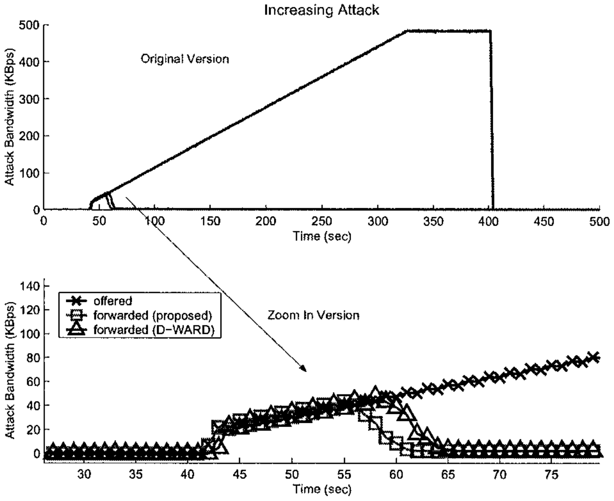

*Figure 10.6.* Increasing SYNC attack.

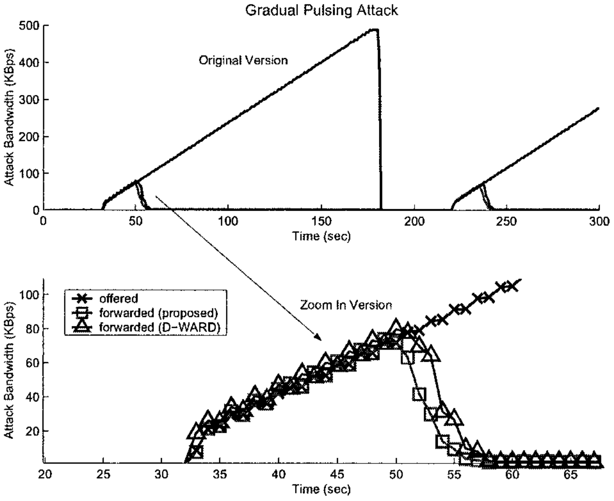

*Figure 10.7.* Gradual SYNC attack.

Thus, D-WARD is unable to detect the presence of the attack. On the other hand, the proposed scheme can identify the attack and perform subsequent rate limiting. Fig. 10.8, 10.9, 10.10, and 10.11 show the experimental result. Similarly, the x-axis denotes time measured in second and the y-axis stands for attack bandwidth measured in KB per second. The line with "x" symbols represents the attack bandwidth generated by attack agents. The line with triangle symbols represents attack bandwidth passing by D-WARD, and the line with square symbols denotes the attack bandwidth going through the proposed scheme.

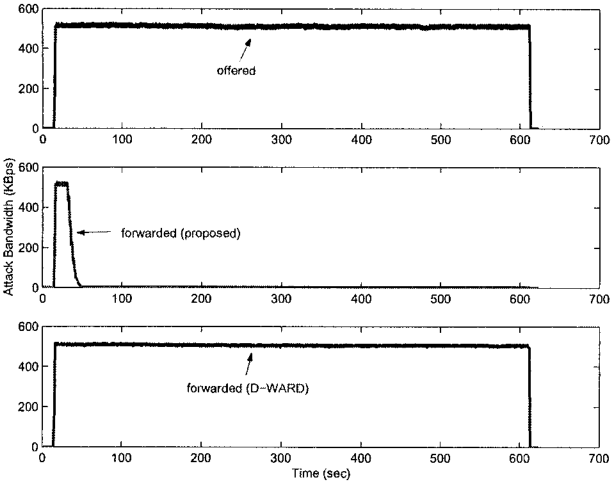

*Figure 10.8.* Constant bandwidth overloading attack.

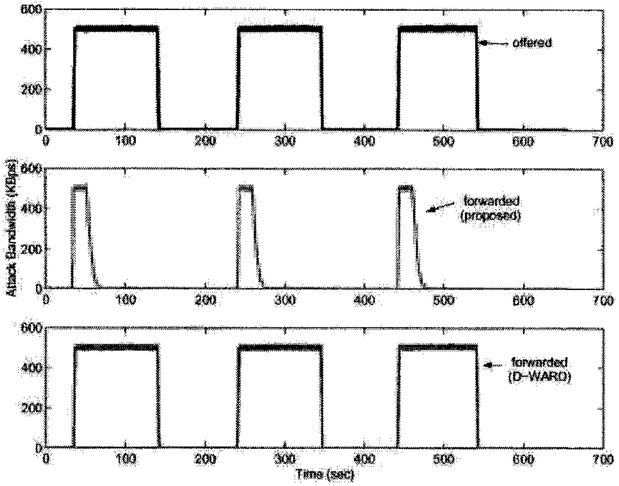

*Figure 10.9.* Pulsing bandwidth overloading attack.

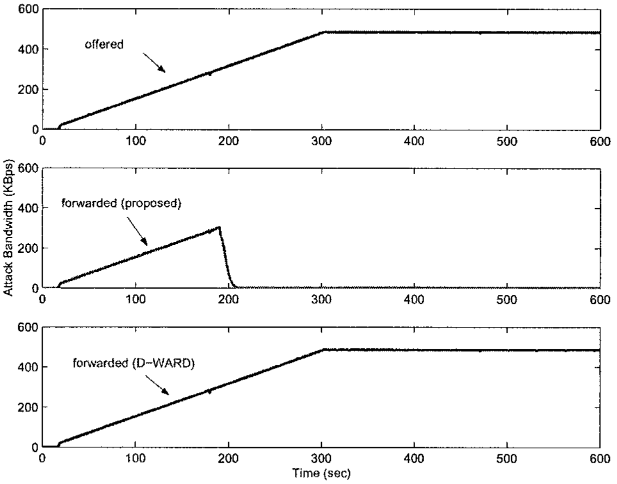

*Figure 10.10.* Increasing bandwidth overloading attack.

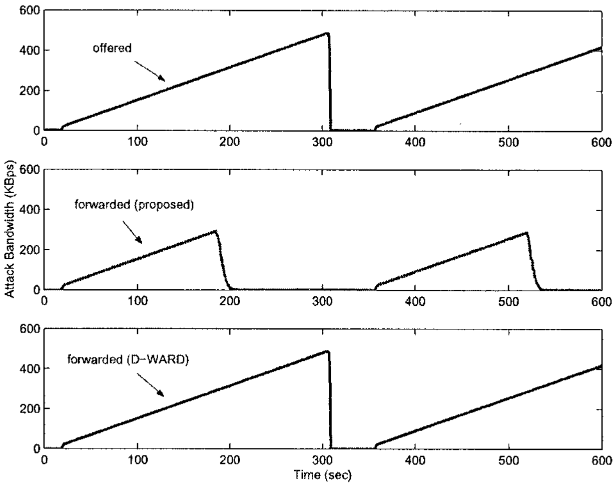

*Figure 10.11.* Gradual bandwidth overloading attack.

## 5. Conclusion and Future Work

Technology resisting DDoS attacks has drawn considerable attention in recent years. However, most existing approaches suffer from either low detection rate, high deployment cost, or lack of effective attack response mechanisms. In this paper, we present a DDoS defense approach which monitors two-way traffic between a set of monitored IP addresses and the rest of the Internet. Our approach can accurately identify DDoS attack flows and consequently apply rate-limiting to the malicious network flows. In this way, DDoS attack traffic can be contained in source networks, and consequently lower the effectiveness of the attack. To effectively stop DDoS attacks, our approach needs to be deployed in routers serving as default gateways. With cooperative routers, our approach provides an effective defense mechanism against DDoS attacks.

Although the scheme presented in this paper can effectively detect DDoS attacks based on two-way flows, several important issues need further investigation. For instance, one pressing problem not addressed in this paper is how to establish the profile of a new type of flow that did not appear in historical traffic data. As mentioned previously, historical traffic used in the learning phase must not have attack traffic; otherwise, characteristics of normal flow behavior may not be derived. To achieve this, the simplest way is to manually examine the collected traffic before it can be passed to learning process. However, it is clear that this approach is not efficient since it requires an extensive amount of time to examine the traffic manually. Furthermore, investigation for effective creation of new flow profiles is desirable. Additionally, an effective profile management system is important and critical to the overall performance of the DDoS defense system. With all the systems putting together, the source-end DDoS defense can be quite effective and consequently deter DDoS attacks.

## References

[man, ] MANAnet DDoS White Papers. http://www.cs3-inc.com/mananet.html.

[Net, ] NetRanger Overview. http://www.cisco.com/univercd/cc/td/doc /product/iaabu/csids/csids1/csidsug/overview.htm.

[NFR, ] NFR Network Intrusion Detection. http://www.nfs.com/products/NID/.

[Chang, 2002] Chang, K. C. (2002). Defending against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial. In *IEEE Communications Magazine*, volume 40, pages 42–51.

[Dean et al., 2002] Dean, Drew, Franklin, Matt, and Stubblefield, Adam (2002). An Algebraic Approach to IP Traceback. *ACM Transactions on Information and System Security*, (2):119–137.

[Feinstein et al., 2003] Feinstein, L., Schnackenberg, D., Balupari, R., and Kindred, D. (2003). Statistical Approaches to DDoS Attack Detection and Response. In *Proceedings of DARPA Information Survivability Conference and Exposition*, volume 1, pages 303–314.

[Ferguson, 1998] Ferguson, P. (1998). Network Ingress Filtering: Defending Denial of Service Attacks Which Employ IP Source Address Spoofing.

[Ioannidis and Bellovin, 2002] Ioannidis, J. and Bellovin, S. M. (2002). Implementing Pushback: Router-Based Defense Against DDoS Attacks. In *Proceedings of Networks and Distributed System Security Symposium*.

[Juels and Brainard, 1999] Juels, A. and Brainard, J. (1999). Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *Proceedings of Networks and Distributed System Security Symposium*, pages 151–165.

[Leiwo et al., 2000] Leiwo, J., Nikander, P., and Aura, T. (2000). Towards network denial of service resistant protocols. In *Proceedings of 15th International Information Security Conference*, pages 301–310.

[Mahajan et al., 2002] Mahajan, R., Bellovin, S., Floyd, S., Paxson, V., and Shenker, S. (2002). Controlling high bandwidth aggregates in the network. *ACM Computer Communications Review , 32(3)*, pages 62–73.

[Mann et al., 2000] Mann, G. R., Watson, D., Jahanian, F., and howell, P. (2000). Transport and Application Protocol Scrubbing. In *Proceedings of INFOCOM*, pages 1381–1390.

[Mirkovic et al., 2002a] Mirkovic, J., Martin, J., and Reiher, P. (2002a). Taxonomy of DDoS Attacks and DDoS Defense Mechanisms. Technical Report 020018, UCLA Technical.

[Mirkovic et al., 2002b] Mirkovic, J., Prier, G., and Reiher, P. (2002b). Attacking DDoS at the Source. In *Proceedings of International Conference on Network Protocols,* pages 312–321.

[Moore et al., 2001] Moore, D., Voelker, G., and Savage, S. (2001). Inferring internet denial-of-service activity. In *Proceedings of 10th USENIX Security Symposium.*

[Park and Lee, 2001] Park, K. and Lee, H. (2001). On the Effectiveness of Router-Based Packet Filtering for Distributed DoS Attack prevention in Power-Law Internets. In *Proceedings of ACM Sigcomm,* pages 15–26.

[Rizzo, 1997] Rizzo, Luigi (1997). Dummynet: a simple approach to the evaluation of network protocols. *ACM Computer Communication Review.*

[Roesch, 1999] Roesch, Martin (1999). Snort - Lightweight Intrusion Detection for Networks. In *Proceedings of LISA'99: 13th Systems Administration Conference,* pages 229–238.

[Savage et al., 2001] Savage, Stefan, Wetherall, David, Karlin, Anna, and Aderson, Tom (2001). Network Support for IP Traceback. *IEEE/ACM Transactions on Networking,* (3):226–237.

[Savage et al., 2000] Savage, Stefan, Wetherall, David, Karlin, Anna R., and Anderson, Tom (2000). Practical Network Support for IP Traceback. In *Proceedings of SIGCOMM Conference,* pages 295–306.

[Shaprio and Hardy, 2002] Shaprio, J. and Hardy, N. (2002). EROS: A principle-driven operating system from the ground up. *IEEE Software,* pages 26–33.

[Song and Perrig, 2001] Song, Dawn and Perrig, Adrian (2001). Advanced and Authenticated Marking Schemes for IP Traceback. In *Proceedings of IEEE INFOCOM Conference,* pages 878–886.

[Sung and X, 2002] Sung, M. and X, J. (2002). IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks. In *Proceedings of International Conference on Network Protocols,* pages 302–311.

[T. Aura and Leiwo, 2001] T. Aura, P. Nikander and Leiwo, J. (2001). DOS-Resistant Authentication with Client Puzzles. *Lecture Notes in Computer Science,* 2133.

[Wang and Reiter, 2003] Wang, X. and Reiter, M. (2003). Defending Against Denial-of-Service Attacks with Puzzle Auctions. In *Proceedings of IEEE Symposium on Security and Privacy,* pages 78–92.