Collaborative Change Detection of DDoS Attacks on Community and ISP Networks*

Yu Chen and **Kai Hwang**

University of Southern California, Los Angeles, CA 90089, USA
{cheny, kaihwang}@usc.edu

ABSTRACT

A community network often operates within the same ISP (Internet Service Provider) domain or administered by a virtual organization spanning across multiple network domains with an established trust relationship. To counter DDoS (distributed denial-of-service) attacks in such a federated network environment, the routers can work cooperatively to raise early warning to avoid catastrophic damages. This paper proposes a collaborative architecture to detect DDoS flooding attacks. The scheme appeals, in particular, to protect networked resource centers that work as a collaboration Grid.

By monitoring the distribution of suspicious traffic changes over a number of attack-transit routers, we developed a new Change-Aggregation Tree (CAT) *mechanism to enable early detection of DDoS attacks on community networks. We want to detect flooding attacks as early as possible. Here, we report preliminary NS-2 simulation results on a single-domain ISP core network to prove the effectiveness of the new collaborative CAT architecture for DDoS defense. The simulated system achieved a detection rate as high as 95% with less than 1% of false positive alarms. Extensions of this architecture to cross-domain DDoS defense are discussed with further research challenges identified.*

Keywords: Network Security, DDoS Attacks, Collaborative Change-Point Detection, Internet Infrastructure, Collaboration Grids, Community Networks, Peer-to-Peer Systems, and Internet Service Provider.

* Manuscript submitted on March 10, 2006 to the IEEE *International Symposium on Collaborative Technologies and Systems* (CTS 2006), Special Session on Collaboration Grids and Community Networks, Las Vegas, NV. May 15-17, 2006. The research work reported here was supported by a NSF ITR Grant 0325409. Corresponding author: Kai Hwang, USC Internet and Grid Research Lab, EEB 212, Los Angeles, CA 90089. E-mail: kaihwang@usc.edu, Tel.: (213) 740-4470.

1. INTRODUCTION

Community networks and Grid systems can be large or small, ranging from local-area to wide-area networks. They form the backbone infrastructure for building multi-site computing clusters, collaboration Grids, P2P systems, web services, enterprise Grids, or any ISP-based core networks for community services. Community networks and collaboration Grids are often formed under a federation of IT administrators. Cooperative computing and high degree of resource sharing are expected in such networked systems [3][11][12].

The basic requirement in network-based resource sharing is to provide reliable and trusted access of local and remote resources in distributed applications [11]. *Distributed Denial of Services* (DDoS) attacks have been identified as the most damaging threat to such hot-spot resource centers [8][9][14][21][26]. To defend against DDoS attacks effectively, the key idea is to achieve real-time detection of the network anomalies.

A satisfactory solution must detect the flooding anomaly as soon as the attacks are launched. Today's DDoS defense schemes are most based on detecting sustained congestion on communication links [18], run out of half-open SYN queue, or imbalance between incoming and outgoing traffic volume on routers [6]. Unfortunately, the time overhead to reveal these anomalous conditions is too long, making the detection scheme ineffective to fence off the DDoS attacks timely.

Consequently, it is highly desired to detect the early launching of DDoS attacks instead of waiting for the flooding to become widespread. We propose a collaborative change-detection scheme to solve this problem. Using the NS-2 simulator, we carried out intensive experiments to verify the effectiveness of our new DDoS defense system.

Under different type of flooding attacks with variant flooding rate, our scheme is capable of detecting the start of DDoS attack quickly with

high accuracy. Another impressive advantage is the small false positive alarm rate experienced.

Treating Internet traffic as stochastic process, sequential change-point detection technique was developed to detect the start of flooding DDoS attacks [2]. The typical change point detection methodologies are hindered by lack of accurate statistical model to describe the pre-change and post-change traffic distributions. The nonparametric CUSUM algorithm was adopted for its simplicity and low computational complexity [2][24][27].

Wang et al. [27] adopted the nonparametric CUSUM method to detect TCP SYN flooding attacks. This is a well-designed change detection technique at the gateway level. Unfortunately, it is not a distributed solution and will not work if the edge network has more than one gateway routers.

Peng et al. [24] took a similar approach to monitoring the source IP addresses. Due to IP address spoofing, there are a lot of fake IP addresses used in DDoS attack. It requires an offline database to keep track of IP addresses appeared in normal cases. Recently, Soule et al. [25] implemented a variant of the classical likelihood ratio test. However, their methods suffer from long detection delays.

It is more efficient to perform detection at victim end and filtering or rate limiting at source side [13][20]. For instance, COSSACK [23] and DefCOM [19] chose to deploy detector at the victim side and send alert to filter or rate limiter located at the source side. Their premise is that edge networks are willing to act cooperatively. However, no one can guarantee this cooperativeness among routers owned by competing ISPs or different organizations.

The rest of this paper is organized as follows: Section 2 describes the system architecture. Section 3 characterizes DDoS attack patterns. Section 4 presents the principle of change detection method. Section 5 discusses the *Change Aggregation Tree* (CAT). Section 6 presents the collaborative CAT detection algorithm. Section 7 reports the NS-2 simulation setups and performance results. Section 8 elaborates on cross-domain DDoS defense. Finally, we conclude with a summary of contributions and discuss further work needed.

2. ISP CORE NETWORK - THE FIRST TESTING ENVIRONMENT

Essentially, the Grids and community networks are *virtual Organizations* (VO) atop the physical Internet [11]. VO members share resources based on application-specific requirements. The users have no control over physical networks applied. In this case, the networks used are most likely administrated by different ISPs. This adds to the complexity in performing collaborative work. Previous research suggests that a total solution to DDoS attack demands a global-scale defense system over the entire Internet [23].

Inside a single ISP core network, it is feasible to demand the routers to cooperate with each other in combating the DDoS attacks, collectively as illustrated in Fig.1. We propose a new distributed change detection scheme using *change aggregation tree* (CAT). The CAT is based on fast recognition of a traffic flow pattern directed towards the victim machine.

The root is the last-hop router to the edge network where the victim machine is attached. Each tree node corresponds to an *attack-transit router* (ATR). Each tree edge corresponds to the link between the ATRs. The system administrator ensures all routers work cooperatively. The CAT server in Fig.1 knows the topology of the network. Legitimate traffic patterns are not featured with the directionality and convergence characteristics. Therefore, once a CAT pattern is recognized beyond certain threshold, we have detected the very early-stage of a DDoS flooding attack.

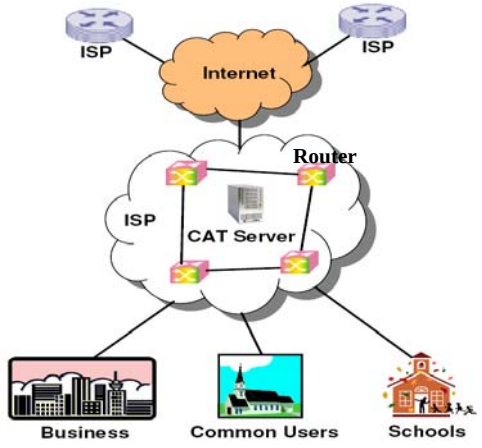


Figure 1. Collaborative change detection of DDoS flooding attacks by routers within a single ISP domain. The attack-transit routers work together to collect traffic anomaly information. A central CAT server aggregates the alerts and performs risk analysis to detect DDoS attacks in real time.

The whole DDoS detection scheme is designed to perform collection, dissemination, and fusion of attack information. The ATRs are in charge of information collection, whereas the CAT server

processes information collected. Periodically, routers compare the short-term behaviors with historical averages. In case a current traffic volume far exceeds the average, an *alert packet* is sent to the CAT server.

3. FLOODING PATTERNS OF DDOS ATTACKS

A DDoS attack deploys multiple attacking entities to deny legitimate application from obtaining a service. The DDoS attacks overwhelm the target host and associated network links with extraordinary huge amount of packets that the victims are incapable to handle. Legitimate traffic is simply blocked. Such brute force attacks do not rely on particular network protocols or system weakness.

As shown in Fig. 2, the attacker simply exploits the huge resource asymmetry between the Internet and the victim. The magnitude of the increased traffic is large enough to crash the victim machine by resource exhaustion, or jam its Internet connection by bandwidth exhaustion, or both Therefore, DDoS attacks can effectively take the victim off the Internet. To avoid being caught by trace back techniques, attackers launch attacks using spoofed IP addresses form innocent victims.

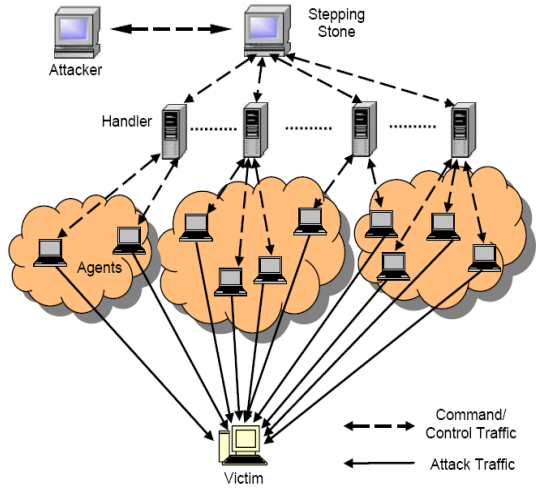


Figure 2. Traffic pattern of a typical DDoS attack.

To overwhelm the victim, DDoS flows converge toward the victim host. Therefore, we can observe abnormal traffic volume changes on routers along the paths of aggregation. The spatio-temporal traffic pattern tends to form a tree rooted the last- hop router to the edge network where the victim resides. By recognizing such tree-like attack patterns at each end router, we can detect the DDoS attacks.

At the early stage of DDoS attack, the abnormal changes are not obvious at each router due to the huge data rate in the core network. Meanwhile, routers cannot afford to monitor traffic on flow or packet level. We define a traffic *flow* by a set of packets satisfying a 5-tuple qualifier: {*source IP address, destination IP address, source port, destination port, protocol applied*} during a given observation window. Thus, such a flow is observable by the router.

We monitor the traffic at a level above the flow level. We define a term *super flow* to cover all packets sharing the same *n* bit prefix in their destination IP address. In addition, the CAT detection result does not need to specify any threshold in advance. The duty of individual router is to monitor the short-term deviation from long-term average behavior. Once certain abnormal change in propagation and aggregation pattern is recognized, the local pattern is sent to a CAT server where the statistic fusion is performed.

4. PRINCIPLES OF CHANGE DETECTION

Routers monitor all flows at each interface and count the incoming and outgoing packet number per time slot. If there is abnormal increase of incoming rate on a flow, the router will check the pattern of change propagation. We define the abnormality of a traffic increase using a *deviation from the average* (DFA) to differentiate abnormal short-term behavior from normal long-term behavior. We adopt weighted running average to describe the long-term behavior.

For a given super flow, let $x(t, i)$ be the number of packets during time slot t coming in by port i and $\overline{X}(t,i)$ be the average number of packets, then the DFA and the historical average is computed by:

$$DFA_{in}(t,i) = x(t,i) / \overline{X}(t,i) \quad \textbf{(1)}$$

$$\overline{X}(t,i) = (1-\alpha) \cdot \overline{X}(t-1,i) + \alpha \cdot x(t,i) \quad \textbf{(2)}$$

Where $0 < \alpha < 1$. This shows how sensitive is the long-term average to current variations. DFA_{in} is defined as abnormality in incoming packet number. While a DDoS flooding attacks start, the current deviation should be noticeably larger than normal random fluctuations. If the abnormality level exceeds a threshold (e.g. 2.0), it is considered suspicious. Similarly, the *DFA* of outgoing traffic is calculated by:

$$DFA_{out} = y(t,i) / \overline{Y}(t,i) \quad \textbf{(3)}$$

$$\overline{Y}(t,i) = (1-\alpha) \cdot \overline{Y}(t-1,i) + \alpha \cdot y(t,i) \quad \textbf{(4)}$$

Where, $y(t, i)$ be the number of packets in time slot t leaving by interface i and $\overline{Y}(t,i)$ be the long-term average number of packets. DFA_{out} is defined as abnormality level of outgoing packet number. With routing table, routers know which port the super flow goes. Therefore, once a DFA_{in} at port i_{in} is considered suspicious, the outgoing port i_{out} is easily identified.

Attack pattern is characterized by the *Deviation Ratio* (DR) and *Offset Ratio* (OR) between the DFAs at the input and output ports of each router. DR specifies the deviation from the average of a super flow at input port i_{in} and output port i_{out}. OR describes the ratio of absolute volume of abnormal changes passed through the router from i_{in} to i_{out}.

$$DR(i_{in}, i_{out}) = DFA_{out}(i_{out}) / DFA_{in}(i_{in}) \quad \textbf{(5)}$$

$$OR(i_{in}, i_{out}) = \frac{y(t, i_{out}) - \overline{Y}(t, i_{out})}{x(t, i_{in}) - \overline{X}(t, i_{in})} \quad \textbf{(6)}$$

Different combinations of DR and OR indicate different patterns of anomaly propagation and aggregations. Figure 3 illustrates the scenarios of how abrupt changes propagate through a router and the aggregation patterns may be looks like. Three of them characterize suspicious traffic flow patterns resulted from DDoS flooding attacks.

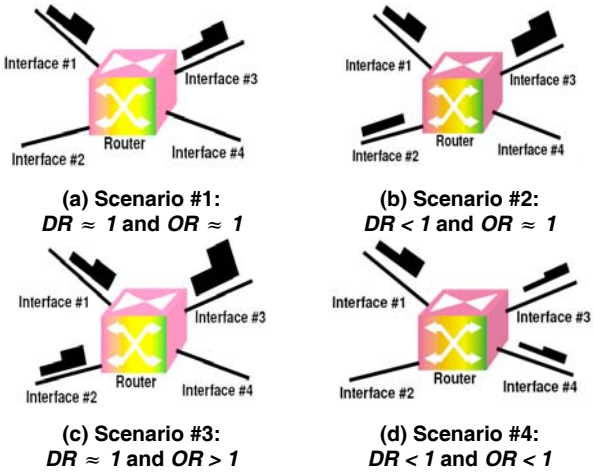


Figure 3. Scenarios of changes in traffic aggregation at attack-transit routers

a. ***DR ≈ 1 and OR ≈ 1***: The flow cuts through the router. The router essentially forwards all increased traffic as shown in Fig. 3(a).

b. ***DR < 1 and OR ≈ 1***: The outgoing flow merges multiple incoming flows, but not all incoming flows contain abnormally increased packets. As all of them are forwarded out through port i_{out}, this is a partial aggregation pattern (Fig. 2(b));

c. ***DR ≈ 1 and OR > 1***: The outgoing flow merges multiple incoming flows, each incoming flow contains abnormal increases with same deviation rate and they aim at the same destination. The router is a merge point on the attacking path and it is a full aggregation pattern (Fig. 2(c));

d. ***DR < 1 and OR < 1***: The changes are scattered, so it is not part of a DDoS attack (Fig. 2(d)).

Scenarios i, ii, or iii indicate possible starting of a DDoS flooding attack. Similar works are carried out in parallel for other flows. The pseudo code of the local attack pattern detection is given in Algorithm 1 below. However, the detection cannot be decided with a few incidences. We need aggregate all related traffic information from all nearby routers to raise accurate alerts timely. All incoming and outgoing packets are identified by the time instants and port numbers. The output of this algorithm is the alert packets to be sent to the central CAT server.

Algorithm 1: Attack Pattern Recognition

Input: x(*t,i*): Incoming packet in time slot *t* at port *i*
y(*t,i*): Outgoing packet in time slot *t* at port *i*
$\overline{X}(t-1,i)$: Average of packet arrivals up to time *t-1* at port *i*
$\overline{Y}(t-1,i)$: Average of outgoing packets up to time *t-1* at port *i*

Output: Alert packets sent to central CAT server.

Procedure:

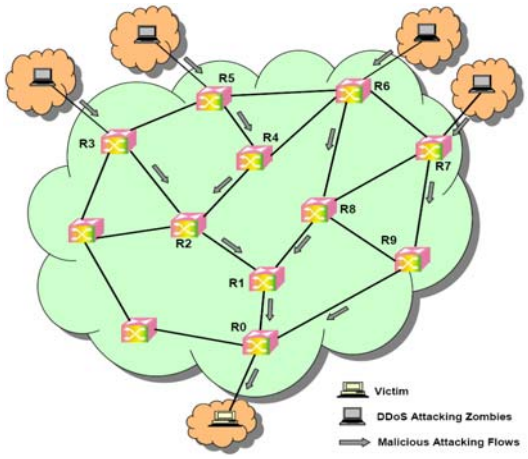
```
01: Update historical average of I/O packets in a flow
02: Calculate DFA_in and DFA_out using Eqs. (1) and (3)
03: If DFA_in > threshold Then
04:     Calculate DR and OR using Eqs. (5) and (6)
05:     If DR ≈ 1 Then
06:         If OR ≈ 1 Then
07:             Suspicious pattern detected, alert packet sent;
08:         Else if OR > 1 Then
09:             Suspicious pattern detected, alert packet sent;
10:         End If
11:     Else if DR < 1 AND OR ≈ 1 Then
12:         Suspicious pattern detected, alert packet sent;
13:     End If
14: End If
```

5. CHANGE AGGREGATION TREES

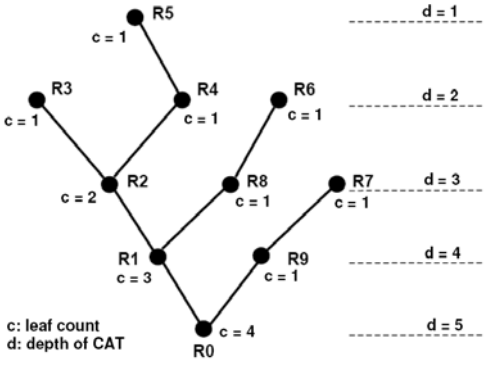
While the flooding traffic starts propagating towards the victim, routers along the path capture the suspicious patterns. Then each router generates an alert packet and sends it to the CAT construction server, where an alert will be raised once a CAT tree is formed. The alert packets report where the suspicious pattern are captured, from which port(s) abnormal traffic detected, and by which port the abnormal traffic is heading.

Figure 4 illustrates how a CAT tree is constructed by an example. While a flooding attack is

launched, a router may detect abnormal traffic volume increase. Then it generates an alert packet. The downstream routers merge the sub-trees. Iteratively, the last hop router would have the whole CAT constructed. In large ISP networks, once we obtained a large enough sub-tree, we can detect the start of DDoS flooding even before the attack flows reach the victim network.



(a) The scenario of 4 zombies launching DDoS attacks on a victim machine attached to router R0



(b) CAT tree obtained at router R0

Figure 4. Detecting DDoS flooding attacks using a CAT-based pattern recognition approach

The CAT-based detection scheme consists of two algorithms. One is Algorithm 1 presented in section 4 for attack pattern recognition at local routers and the other for network-wide attack information fusion at the CAT server to be specified in section 6.

The CAT scheme is deployed in the core network routers where high data rate and limited resources routers can share to perform complicated security functions. It is very difficult to set the threshold for lack of information of end applications.

These constraints imply that the collaborative distributed detection mechanism has to be lightweight with low complexity. Another critical requirement is that it has to be adaptive to changes in dynamic traffic properties. Our CAT detection scheme is adaptive in nature.

6. COLLABORATIVE CAT DETECTION

Before introducing the attack alert fusion algorithm, we specify the alert packets an ATR sends to the central CAT server. To indicate the location of a suspicious pattern, the router ID has to be sent. It is also mandatory to identify the flow in which abnormality was observed. The alert packet provides the upstream and downstream router IDs instead of the port numbers. Since all routers are under same authority and work cooperatively, each router knows their immediate neighbors. Summarized in Table 1, only few parameters are required in each alert packet

Table 1. Flow Parameters in An Alert Packet

Parameter	Brief Definition
nd_id	Node ID, where suspicious pattern was observed
fl_id	Suspicious flow ID
up_num	Number of upstream nodes
dn_num	Number of downstream nodes
up_id	Node ID of upstream node
dn_id	Node ID of downstream node

The CAT server maintains a graph of the network topology. Periodically it tries to construct CAT tree in the graph based on collected alert packets. Figure 5 presents the flow chart of CAT fusion algorithm executed on the CAT server. Starting from the node R_{min} with minimum ID, CAT server takes it as the root node. The server scans through upstream child nodes indicated by the *up_id* and the children of a child. This leaf search procedure is performed iteratively, until the leaf nodes are counted. Hence, a subtree rooted at R_{min} is completed

If there is a downstream router R_{dn} indicated with *dn_id*, we take router R_{dn} as the new root and router R_{min} becomes one of R_{dn}'s children. So the previous sub-tree is merged into the tree rooted at R_{dn}. Meanwhile, the leaf search procedure is repeated for all upstream routers of root R_{dn} except R_{min}. Then we check the downstream router of R_{dn} and repeat what we did on R_{dn} until the downstream router is out of the domain or is pointing to an edge network.

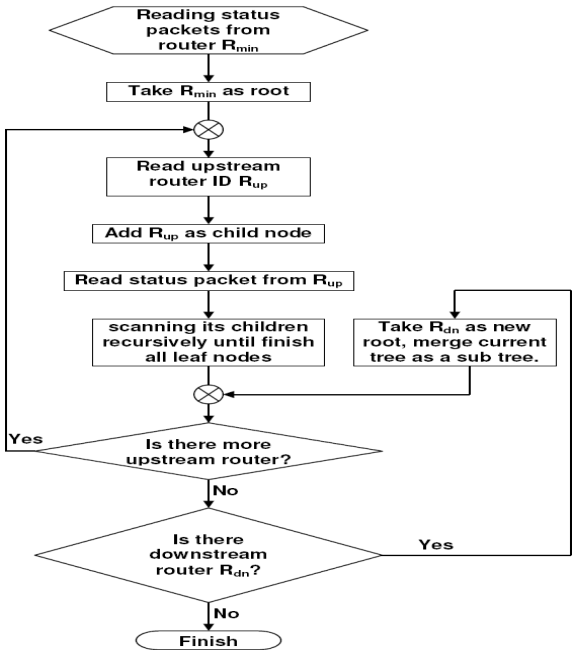


Figure 5. Flow chart for on-line CAT construction based on fusion of all alert packets received

7. NS-2 SIMULATION RESULTS

We verified our DDoS detection scheme with results from NS-2 simulation experiments [22]. This section introduces the experimental setup and reports preliminary performance results.

7.1 Experimental Setup

To evaluate the performance of our CAT detection scheme, we allow variations in three dimensions: *test topology, background legitimate traffic,* and *attack characteristics*. We adopted real ISP topology downloaded from the Rocketfuel project website in University of Washington [1]. Figure 6 presents one typical topology simulated in our experiments. The delays (RTTs) between legitimate hosts and victim nodes are uniformly distributed in the range of [40 ms, 200 ms], and the bandwidth is set as 100MB.

The background traffic is generated according to statistical parameters obtained by analyzing the real OC48 trace dataset from the CAIDA project [4].

To provide DDoS attacks flooding flows for our research, we studied real-world DDoS attack tool Stacheldraht V4 [10]. It is one representative of the DDoS attack toolkits that emerged in early 2000. Although new toolkits provide more sophisticated monitoring and control capabilities, the flood traffic generated by Stacheldraht serves our purpose. Stacheldraht generates ICMP, UDP, TCP SYN, and Smurf attacks. In our simulations, we generated flooding attack traffic patterns according to Stacheldraht's behavior.

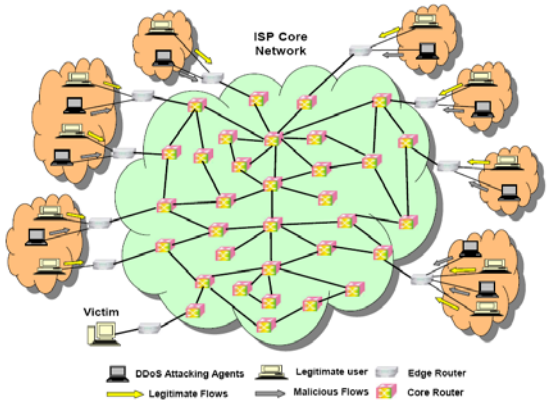


Figure 6. Topology in NS-2 experiments.

7.2 Performance Evaluation Results

Before evaluating the performance of our collaborative DDoS attack detection scheme, we define the basic performance metrics used. Then we will discuss the plotted results from experiments.

A. Performance Metrics

The performance can be evaluated with three metrics: *detection delay, detection accuracy, and false positive rate.* All metrics are measured under different attacks (TCP flooding, UDP flooding, ICMP flooding) with varied attacking traffic rates. The *average detection time* τ measures the time interval between the start of a DDoS attack and the time the CAT server raise the alarm of attack.

The detection accuracy is evaluated using three metrics, the *detection rate* R_d, *false alarm rate* R_{fp}, and *receiver operating characteristic* (ROC). The detection rate is formally defined by:

$$R_d = a / n \qquad (7)$$

where a is the number of detected attacks and n is the total number of actual attacks. The false positive rate measures the ratio of normal traffic being wrongly detected as attacks. The formal definition is:

$$R_{fp} = p / m \qquad (8)$$

where p is the total number of false positive alarms and m is the total number of normal traffic events. The ROC curve is adopted to describe the tradeoff between the detection rate and false positive rate.

B. Simulated Performance Results

The CAT scheme detects the start of DDoS flooding by monitoring the variance in traffic volume, collecting all individual suspicious patterns, and constructing CAT tree periodically. Here we need to decide whether the constructed CAT tree is resulted from an actual DDoS attack or merely from random traffic fluctuation.

As discussed in section 4, the initial attack pattern of a flooding DDoS offense and random traffic fluctuations may be confused at local router level. The difference lies in the observation that the random fluctuations do not propagate a long distance. They do not show the flow directionality and homing convergence properties in the aggregation process. We observed that random fluctuations only lead to a smaller CAT tree with a short height and limited number of leaf nodes.

Hence, we need to specify a threshold tree size that implies a true flooding attack. Simply put, we use the sum of leaf count and tree height to assess the size of a CAT tree. This size sets a *detection threshold* of the DDOS attacks. The threshold value is determined by the network topology and training experiences. The higher is the threshold, the more accuracy is expected, which means more difficult to achieve higher detection rate. Formally, we define the *CAT detection threshold* by:

$$\theta = c + d \qquad (9)$$

where c is the leaf count and d is the tree height.

Figure 7 plots the variances of the *detection rate* (Eq.7) and the *false positive rate* (Eq.8) with respect to different CAT tree sizes as a detection threshold. The tree size is obviously sensitive to the topology of the network. The message being conveyed here is that in most cases the CAT tree incurred with traffic fluctuation is restricted by a threshold less than 5.

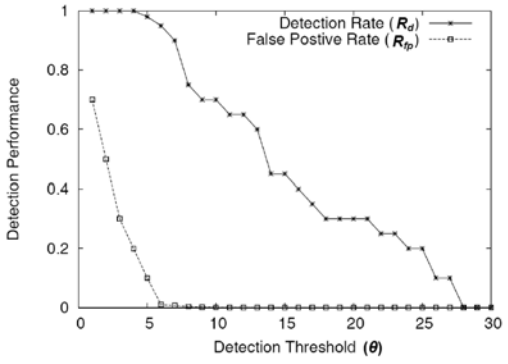


Figure 7. Detection performance plotted against the detection threshold (CAT tree size)

Under a threshold of 5, the detection rate R_d is almost 100% for sure but accompanied by some false alarms. Since random fluctuation cannot sustain the test to form a large CAT tree, when a high R_d of 95% is maintained at threshold 5 the false alarm rate R_{fp} drops quickly from 70% to zero. As the threshold increase to a higher value of 10, the detection rate can be still maintained at 75% level.

In our simulation, we have studied the relationship between the detection threshold and the traffic rate experienced by ATRs. In a highly distributed DDoS flooding attack, where traffic rate of experienced by individual router is pretty low. The flooding situation is not detected until multiple streams aggregate and cause noticeable changes.

Figure 8 plots effects on the threshold value by variation traffic rate experienced by the ATRs. The level of flooding is directly related to this measure. When the traffic rate is low, a small CAT tree may suffice to distinguish the attacking traffic from the regular traffic patterns. As the traffic rate increases or when the flooding level mounts, the threshold chosen must also increase steadily.

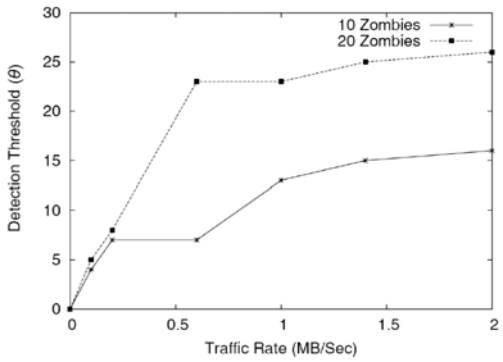


Figure 8. Variation of detection threshold with respect to changes in traffic rate.

However, eventually the threshold value becomes saturated. Further, the more zombies are involved, the higher is the threshold value chose. Generally, after traffic rate of greater than 1 MB/s, both threshold curves become saturated. That means we have a more stable detection threshold to use, when the DDoS flooding reaches sufficiently high level as seen in Fig.8.

Surely we want to capture attacks at the highest achievable accuracy, but the false positive alert may also increase with the detection rate. The false alarms will incur additional system overhead, since false alarms trigger unnecessary but costly countermeasures. The ROC curve shown in Fig.9

reveals that our CAT based detection scheme can reach a detection rate as high as 95% with less than 1% of false positive rate. This result is very encouraging in proving the effectiveness of the collaborative CAT defense system.

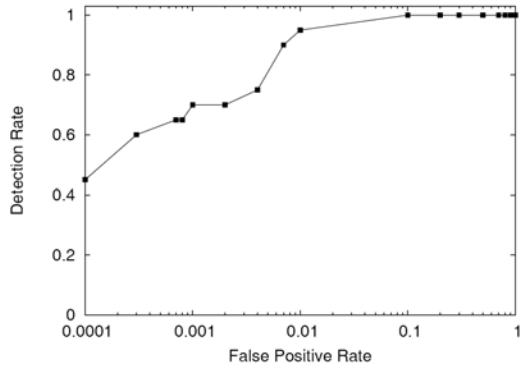


Figure 9. ROC curves showing the tradeoff between detection rate and false positive rate.

Another critical issue is how quickly can we detect the launch of DDoS attacks from large number of zombies. Since we update the CAT tree at all ATRs, periodically, the time delay may incur with the updating of the CAT server with frequent local changes detected by individual ATRs. The CAT server may also need some processing time, if large number of servers are involved. We estimate that updating in every half second is needed. Any delay longer than 0.5 second may not be tolerable.

8. MULTI-DOMAIN DDOS ATTACK DETECTION

We need to extend the scale of CAT-based DDoS detection to multiple network domains. Inter-domain communication is thus needed in the alert aggregation process. The CAT aggregation algorithm must be extended to perform wide-area network anomaly detection. We must reach agreement to resolve conflicts between security policies applied in different domains.

The routers at various domains exchange alert packets under agreed terms. The idea of cross-domain DDoS defense is illustrated in Fig.10. Multiple CAT servers at different AS domains must be protected by dedicated VPN channels among them. The alert packets generated by ISP routers from different domains may follow different policies and data formats. Multiple CAT servers must work together to resolve the conflicts.

In follow-up research, we have to reveal the messaging overheads involved in cross-domain communications. One can establish VPN links or structured overlay networks to support cross-domain alert packet exchanges. This demands special policies to share security information.

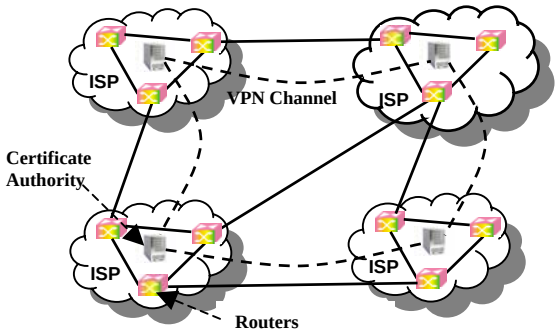


Figure 10. Multiple CAT servers in several ISP domains communicating with each other to resolve the conflicts in security policies

We will reveal the performance attributes tied to policy fusion methodology applied. It was suggested to approach the policy conflict problem through trust negation [5]. Our prior work [7] on collaborative detection and filtering of shrew DDoS attacks [17] can be extended to combat flooding DDoS attacks across multiple ISP administrative domains

9. CONCLUSIONS AND FURTHER WORK

The complexity of DDoS attack patterns grows fast, as new network vulnerability is identified and more sophisticated attack tools are available. There is no magic that can handle all types of DDoS attacks. The shared sources in collaboration Grids and community networks are especially prone to such attacks. One solution works well in a given network environment but may fail in other networks. In this section, we summarize our contributions and then discuss security assurance, system scalability, and limitations of our CAT DDoS detection system.

9.1 Concluding Remarks

This paper reports our work in detection of DDoS flooding attacks against Grid resource sites or hot-spot servers in community networks. It is essential to detect DDoS attacks sufficiently early before harms are done to legitimate applications. Our major contributions are summarized below:

(A). Early detection of DDoS Flooding Wave: Based on spatio-temporal pattern of an anomaly detected in an ISP network, our change aggregation tree can detect a DDoS flooding attack early at its outbreak. This novel approach captures the abrupt changes at the starting of a wave of DDoS flooding directed towards the victim.

(B). Deployment in ISP Core Networks: Our detection scheme is implementable in the routers used in an ISP core network under the same authority. Simulation results supported the claimed advantages. Issues and solutions to extend the CAT scheme across multiple ISP domains are discussed.

(C). Tradeoffs between Detection Rate and False Alarm Tolerance: We verified the effectiveness of our detection scheme through intensive simulation experiments. The results indicated that the system is capable of detecting flooding DDoS attacks swiftly with a high detection rate and low false positive rate. This accuracy is yet to be tested further with deployed prototype and benchmark experiments in the future.

9.2 Discussions and Further Work

Malicious attackers can hide local anomalies or send false attack pattern to the CAT server. These false alarms can break the CAT tree construction process. We can make the CAT construction more robust by introducing a verification function. Having the network topology, the CAT server is capable of making up the single lost point or rectifying the false patterns according to upstream and downstream patterns. Isolated attackers can be quarantined by this chained protection scheme.

If multiple compromised routers send false attack patterns collectively, the CAT server can be deceived to construct the wrong pattern. Attackers can exploit this weak point to launch DDOS attacks on the victim. Theoretically speaking, once an intruder gain full control over multiple routers, she can do whatever she wants. Our CAT based detection system does not create new weakness. This problem is wide open and not solved yet.

It is a considerable challenge to discriminate DDoS flooding attacks from sudden increases in legitimate traffic or flash events [6][16]. When flash crowd happens, the CAT server may create a similar tree. Actually, when there are no more changes to be detected, it becomes harder to segregate flash crowd flows from DDoS flooding flows. This is especially true in highly distributed attacks in which each individual flow occupies a normal bandwidth share.

For further effort, we suggest to separate flash crowd from DDoS flooding by other performance metrics to capture their differences. The candidate ideas include the use of source IP addresses counts or packet content matching. A more efficient technique is yet to be developed to cope with this situation to meet the real time detection requirement.

Furthermore, we suggested extending the CAT-based DDoS detection scheme across the network domain boundary. Efficient and effective techniques are needed to resolve policy conflicts applied at different domains [15]. Multiple CAT servers must work together to resolve the conflicts. In addition, a secure and reliable communication platform is necessary for routers at different domains to exchange alert packets under negotiated policy agreement. This opens up more research problems to solve.

REFERENCES:

[1] Anderson, T., R. Mahajan, N. Spring, and D. Wetherall, "Rocketfuel: An ISP Topology Mapping Engine," http://www.cs.washington.edu/research/networking/rocketfuel/, Feb. 2006

[2] Blazek, R., H. Kim, B. Rozovskii, and A. Tartakovsky, "A Novel Approach to Detection of Denial-of-Service Attacks via Adaptive Sequential and Batch-sequential Change-Point Detection Methods," *Proc. of the 2001 IEEE Workshop on Information Assurance and Security*, June 2001.

[3] Berman, F., G. Fox, and A. Hey (editors), *Grid Computing*, John Wiley, England, 2003.

[4] Monk, T. and K. Claffy, "Cooperation in Internet Data Acquisition and Analysis," Coordination and Administration of the Internet Workshop, Cambridge, MA., Sept. 8-10, 1996. (CAIDA Project, http://www.caida.org/)

[5] Cai, M., K. Hwang and Y. Chen, "Hybrid Intrusion and Anomaly Detection with Weighted Signature Generation", *IEEE Trans. On Dependable and Secure Computing,* revised Sept. 2005.

[6] Carl, G., G. Kesidis, R. Brooks, and S. Rai, "Denial-of-Service Attack Detection Techniques," *IEEE Internet Computing*, January 2006.

[7] Chen, Y. and K. Hwang, "Collaborative Detection and Filtering of Shrew DDoS Attacks with Spectral Analysis", *Journal of Parallel and Distributed Computing,* accepted to appear 2006.

[8] Chen, Y., M. Cai, and K. Hwang, "Defense against Internet Worms and DDoS Attacks with Dynamic Hardware and Network Processors", Chapter 13 in *Hardware-based Security*, edited by R. Lee and S. Smith, Morgan Kaufmann, to appear 2006

[9] Chakrabarti, A. and G. Manimaran, "Internet Infrastructure Security: A Taxonomy", *IEEE Network*, November 2002.

[10] Dittrich, D., "The 'Stacheldraft' Distributed Denial of Service Attack Tool," http://staff.washington.edu/dittrich/, 2000.

[11] Foster, I., C, Kesselman, and S. Tuecke, "The Anatomy of the Grid," *International Journal of Supercomputer Applications*, 15(3), 2001

[12] Fox, G., et al, "Peer-to-Peer Grids", Chapter 18 in *Grid Computing,* edited by Berman, Fox, and Hey, Wiley, 2003, pp.471-490.

[13] Gil, T. and M. Poletto, "MULTOPS: a Data-Structure for Bandwidth Attack Detection," *Proc. of 10th USENIX Security Symposium*, August 2001.

[14] Houle, K., G. Weaver, N. Long, and R. Thomas, "Trends in Denial of Service Attack Technology", CERT Coordination Center Document, 2001, www.cert.org/archive/pdf/.

[15] Hwang, K., Y.K. Kwok, S. Song, M. Cai. And Yu Chen, and Ying Chen, "Security Binding and Worm/DDoS Defense Infrastructure for Trusted Grid Computing", *International Journal on Critical Infrastructures*, Vol.2, No.4, 2005.

[16] Jung, J., B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial-of-Service Attacks: Characterization and Implications for CDNs and Web Sites," *Proc. of Int'l World Wide Web Conference*, ACM Press, 2002.

[17] Kuzmanovic, A. and E. W. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks—The Shrew vs. the Mice and Elephants," *Proc. of ACM SIGCOMM 2003*, Aug. 2003

[18] Mahajan, R., S. Floyd, and D. Wetherall, "Controlling high-bandwidth flows at the congested router," *Proc. of ACM 9th International Conf. on Network Protocols (ICNP)*, Nov. 2001

[19] Mirkovic, J., M. Robinson, P. Reiher, and G. Kuenning, "Alliance Formation for DDoS Defense," in *Proceedings of the New Security Paradigms Workshop, ACM SIGSAC*, August 2003.

[20] Mirkovic, J. and P. Reiher, "D-WARD: A Source-End Defense Against Flooding Denial-of-Service Attacks," *IEEE Trans. on Dependable and Secure Computing*, Vol. 2, No. 3, July 2005, pp. 216-232

[21] Moore, D., G. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *Proceedings of the 10th USENIX Security Symposium*, 2001.

[22] McCanne, S. and S. Floyd, NS-2 Network Simulator, http://www.isi.edu/ nsnam/ns/, 1997.

[23] Papadopoulos, C., R. Lindell, J. Mehringer, A. Hussain, R. Govindan, "COSSACK: Coordinated Suppression of Simultaneous Attacks," *in Proceedings of DISCEX III*, 2003, pp. 2—13.

[24] Peng, T., C. Leckie, and K. Ramamohanarao, "Detecting Distributed Denial of Service Attacks by Sharing Distributed Beliefs," *ACISP 2003*, LNCS 2727, pp.214-225, 2003.

[25] Soule, A., K. Salamatian, and N. Taft, "Combining Filtering and Statistical Methods for Anomaly Detection," *Proc. of ACM Internet Measurement Conf.* Berkeley, CA. Oct. 19-21, 2005.

[26] Specht, S. and R. B. Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures," *Proceedings of PDCS*, 2004.

[27] Wang, H., D. Zhang, and K. Shin, "Change-Point Monitoring for the Detection of DoS Attacks," *IEEE Trans. on Dependable and Secure Computing*, Vol. 1, No. 4, Oct.-Dec., 2004.

[28] Zhou, R. and K. Hwang, "Trust-Preserving Overlay Networks for Global Reputation Aggregation in Scalable P2P Systems ", *IEEE transaction on Parallel and Distributed Systems,* (TPDS), revised March 2006.

BIOGRAPHICAL SKETCHES:

Yu Chen received his B.S. and M.S. from Chongqing University, China in 1994 and 1997, respectively, and the M.S. in Computer Engineering from University of Southern California (USC) in 2002. He is currently a Ph.D. candidate in Computer Engineering and works at the USC Internet and Grid Computing Laboratory. His research interest includes Internet infrastructure security, DDoS attack detection and defense, Internet traffic analysis and distributed security infrastructure. He can be reached at cheny@usc.edu.

Kai Hwang is a Professor of Electrical Engineering and Computer Science and Director of Internet and Grid Research Laboratory at University of Southern California. He received the Ph.D. from the University of California, Berkeley. An IEEE Fellow, he specializes in computer architecture, parallel processing, Internet security, Grid and cluster computing, and distributed computing systems.

Presently, he leads a NSF-supported GridSec project at USC. The GridSec project develops security-binding techniques, defense schemes, and reputation systems against system intrusions, malicious peers, network worms, and DDoS attacks in designing and applications of large-scale Grid and P2P computing systems. Dr. Hwang can be reached via Email: kaihwang@usc.edu or through URL: http://GridSec.usc. edu/Hwang.html.