Distributed Change-Point Detection of DDoS Attacks: Experimental Results on DETER Testbed

Yu Chen
Department of Electrical and Computer Engineering
SUNY – Binghamton
Binghamton, NY 13902

Kai Hwang
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089

Wei-Shinn Ku
Dept. of Computer Science and Software Engineering
Auburn University
Auburn, AL 36849

Abstract

It is highly desired to detect the DDoS flooding attacks at an early stage in order to launch effective countermeasures timely. We have developed a distributed change-point detection scheme to detect flooding type DDoS attacks over multiple network domains. The approach is to monitor the spatiotemporal pattern of the attack traffic. We have simulated the new defense system on the DETER testbed. The new scheme is proven scalable to cover hundreds of ISP-controlled network domains. With 4 network domains working collaboratively, we achieved on the DETER testbed a 98% detection rate with less than 1% false alarms.

1 Introduction

Timely detection is an essential to minimize the damage of *distributed denial of services* (DDoS) attacks. However, most of today's detection schemes are built on detecting the consequences rather than the causes of the flooding traffic [13, 20]. Unfortunately, damages have already been caused when consequences are observed. Thus, it is highly desirable to detect DDoS attacks at the earliest possible time, instead of waiting for the flood to become widespread [5].

At an early stage of a DDoS attack, the traffic changes are difficult to detect because low traffic fluctuations are not observable. Monitoring the Internet traffic at individual flow level is cost prohibitive to cover all possible flows. In addition, the global traffic in wide area network is too large to perform real-time detection of network anomalies effectively.

To be cost-effective, we propose to monitor the traffic at a superflow level. A superflow contains all packets destined for the same network domain from all possible source IP addresses and applies various protocols such as TCP or UDP, etc. This detection level lies between the level of large-scale aggregate traffic and individual traffic flows. All packets of a superflow have the same prefix in the destination IP addresses that indicate the network address of the destination domain.

In an earlier paper [7], we have proposed a *Distributed Change-point Detection* (DCD) architecture using a new mechanism, called *Change Aggregation Tree* (CAT). The CAT mechanism is designed to observe spatiotemporal distribution of changes in traffic volumes. When a DDoS attack is launched, the routers detect abrupt changes in traffic flows. The domain server uses the router-reported traffic change information to construct the CAT tree. Usually, these changes in traffic flows present a directional homing towards the victim system. Random fluctuations incurred with legitimate traffic flows do not present the homing effects.

It is critical to verify the effectiveness of a DDoS defense scheme in a systematical approach. Previous works suggested to designing the attack-defense experiments over five orthogonal dimensions [12]. On the DETER testbed, we have evaluated our DCD scheme through intensive experiments with various network topologies, background traffic and using real life DDoS attack tools. The major performance evaluation metrics include the detection accuracy and system overhead. To understand the scalability of the system, we implemented the detection scheme from 4 to 16 domains.

The rest of this paper is organized as follows: Section 2 briefly reviews related work. Section 3 presents the principle of change detection methods and explains the CAT tree construction across

* Manuscript submitted on July 16, 2007 to *DETER Community Workshop on Cyber Security Experimentation and Test,* in conjunction with *USENIX Security Symposium*, Boston, MA. August 6-7, 2007. The research work reported here was supported by a NSF ITR Grant 0325409. Corresponding author: Yu Chen, Dept. of Electrical & Computer Eng., SUNY–Binghamton, Binghamton, NY 13902. E-mail: ychen@binghamton.edu, Tel.: (607) 777-6133, Fax: (607) 777-4464.

multiple domains. Section 4 reports the DETER experiments setups and performance results. Section 5 summarizes our work and discusses further research.

2 Related Work

A plethora of DDoS defense and response mechanisms have been suggested in the past, including IP traceback [1], packet filtering [15], and flood pushback [14]. More sophisticated intrusion detection systems [21] and DDoS defense schemes [8, 20, 26] have been recently proposed. Researchers have attempted to combat repeated DDoS attacks [11]. Others use overlay networks [28] and DDoS-resilient scheduling [24] to establish the trust in distributed systems.

MULTOPS [10] and D-WARD [16] suggested filtering or rate limiting on suspicious flows at the source end. The security managers often focus on protecting their own networks and choose a local detection approaches [5]. For instance, the COSSACK [22] and DefCOM [17] deploy detectors at the victim side and send alerts to filter or to rate limiter located at the source side. Chen and Song [6] proposed a perimeter-based scheme for ISP to enable anti-DDoS services to their customers. Their scheme relies on edge routers to identify the sources of flood.

Researchers use change-point detection theory to detect abnormal Internet traffic caused by DDoS attacks [4, 23]. Lacking accurate statistics to describe the pre-change and post-change traffic distributions, a nonparametric CUSUM scheme was developed for its low computational complexity [4]. The scheme monitors the short-term behavior shifting from a long-term behavior. Once the cumulative difference reaches certain threshold, an attack alert is raised. Wang et al. [27] have suggested a centralized DDoS defense scheme to monitor the change points at the gateway level. Peng *et al*. [23] took a similar approach to monitoring the source IP addresses.

Our DCD approach is unique and offers the very first attempt to explore distributed change-point detection over multiple collaborative network domains. In addition, it does not need efforts from end-points. This implies that the DCD scheme is promising to be integrated into the core network as part of the network infrastructure protection mechanism.

3 Distributed Change-Point Detection

The DCD scheme detects DDoS flooding attacks by monitoring the propagation of abrupt traffic changes inside the network. If a CAT tree is constructed sufficiently large and the tree size exceeds a preset threshold, an attack is declared.

3.1 Change-Point Detection Principle

In change-point detection problems, if pre-change and post-change distributions are known, several statistic methods have been suggested to solve the problem [4]. In DDoS attack detection, we adopt the non-parametric CUSUM approach for its simplicity and due to the lack of precise statistic model to describe the distribution of pre-change or post change network traffic.

Let $t_1, t_2, \ldots, t_m$ be discrete time instants and $x(t_m, i)$ be the number of packets received by a router during time slot m at port *i.* The historical estimate average number of packets is evaluated using weighted running average. The deviation of input traffic from the average indicates the differences between current traffic volume and history average. While a DDoS flooding attack is being launched, the cumulative deviation is noticeably higher than the random fluctuations. Furthermore, such an abnormal traffic surge will propagate in the network and converge towards the victim.

3.2 Traffic Surge Detection at Routers

All packets of an attacking superflow must be homing towards the same destination network. Before entering the destination domain, the flow paths present a converging homing-tree pattern. Only at the destination domain, the superflow scatters packets towards a particular edge network specified by the destination IP address.

Each router monitors traffic variation and counts the packet number within a monitory window at each I/O port. We use the term *traffic pattern* to refer to the combination of traffic surges at all I/O ports of a router. Figure 1 illustrates the four possible patterns of how traffic surge goes through a router may observe. The height of the black boxes in Fig. 1 signifies the magnitude of traffic volume at I/O links. The raised block height indicates a surge detected and the lower boxes represent normal traffic.

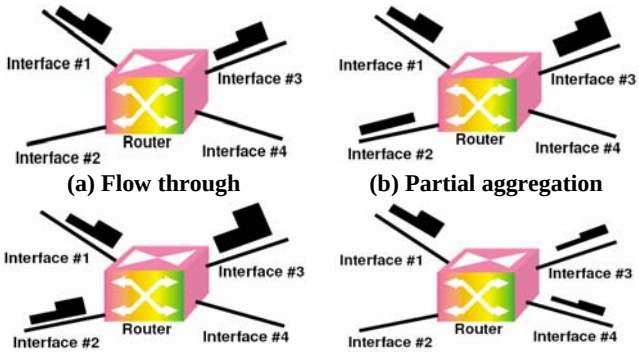


Fig.1. Four traffic changes detected at the I/O ports of a 2 by 2 router.

Below is a brief description of the four typical patterns illustrated by Fig. 1.

a. Flow-through pattern:

This traffic pattern is shown in Fig. 1(a). The router forwards the traffic flow from an input port to a selected output port without subdividing or diverting the traffic to other outgoing port.

b. Partial aggregation pattern:

All the incoming flows are merged at one outgoing port, not all incoming flows contain traffic surges as shown in Fig. 1(b).

c. Full aggregation pattern:

The outgoing flow merges multiple incoming flows, all containing traffic surges exceeding the threshold. This router is considered a merge point on the attacking path (Fig.1(c)).

d. Scatter pattern:

The incoming flow scatters at this router. This pattern is observed in the destination domain or on the path of multicast traffic. This is not part of a DDoS attack (Fig. 1(d)).

In patterns (a), (b) and (c), the income traffic surge does not scatter into different directions after going through the router. This implies that all the abruptly increased traffic goes to the same destination network. Therefore, it is suspicious that they are part of a DDoS flooding attack.

3.3 Distributed Change-Point Detection

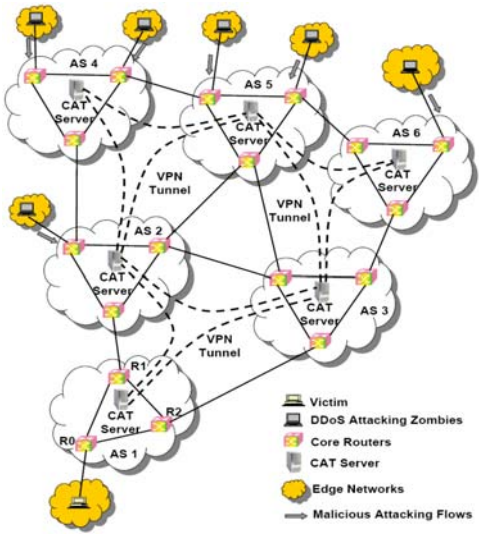
Figure 2 presents the system architecture of the DCD scheme. The system is deployed over multiple AS domains. There is a central CAT server in each domain. The system detects traffic changes, checks flow propagation patterns, aggregates suspicious alerts, and merges CAT subtrees from collaborative servers into a global CAT tree. The root of the global CAT tree is at the victim end. Each tree node corresponds to an *attack-transit router* (ATR), which is on the path by that attack traffic propagates to the victim. Each edge of the tree corresponds to a path link between the ATRs.

Individual router functions as a sensor to monitor local traffic fluctuations. A change-point detection scheme is executed on each router. A router raises an alert and reports an anomalous traffic pattern to the CAT server. The CAT server constructs a CAT subtree according to collected alerts. The subtree displays a spatiotemporal vision of the attack superflow in the domain. Then, the CAT servers at different domains form an overlay network or communicate with each other through *virtual private network* (VPN) channels.

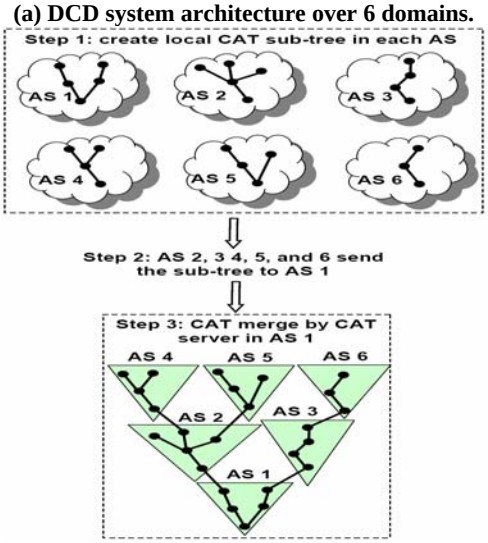
Fig.2. Distributed change detection of DDoS attacks over multiple AS domains.

All CAT servers send their CAT subtrees to the edge server in the destination domain, where the victim is attached. By merging CAT subtrees, the destination server has a global picture of the attack. The CAT detection scheme does not need to specify an absolute threshold on traffic volume. The detection is done by checking the number of routers raising the alerts from the CAT subtree.

Figure 3 shows a network environment involving six cooperating AS domains. The victim system is located in the AS1 domain. Zombies are scattered widely on the Internet outside the illustrated domains. By detecting abnormal traffic changes in each domain, the CAT server creates a CAT subtree locally at each domain. Figure 3(b) shows three steps taken to merge the 6 subtrees generated by 6 CAT servers of 6 AS domains.



(a) DCD system architecture over 6 domains.



(b) Merging 6 CAT subtrees to yield a global CAT tree

Figure 3. The construction of an example 6-domain global CAT tree for DDoS attacks.

4 Experiments on The DETER Testbed

We verified the performance of our DDoS detection scheme with network attack experiments on the DETER testbed [3] at USC Information Sciences Institute. The experimental settings and performance results are reported as follows.

4.1 Experiment Settings and Components Implementation

To evaluate the performance of the CAT-based DDoS detection system, we allow changes in three dimensions: network topology, attack scenario, and background traffic. We adopt the real-world ISP topologies downloaded from the Rocketfuel Project at the University of Washington [2]. Figure 4 presents one of the network configurations used in our experiments.

During the studies over 4 domains, typically we used network topologies which have about 30 routers. For the scalability studies, which involves from 4 to 16 domains, we adopted the smallest topology (about 12 routers) in the Rocketfuel data set due to the limitation of available machines on the testbed. The link bandwidth among the network domains was set at 100 MB/s.

To generate the background traffic closer to reality, we use the OC48 trace dataset from the CAIDA project [19] to regenerate Internet traces using the Harpoon traffic generator [25]. To generate DDoS attacks, we use the toolkit Stacheldraht (version 4.0) [9].

Stacheldraht generates the ICMP, UDP, TCP SYN flooding and Smurf attacks. The UDP and ICMP flooding packet rate (number of packet / second) for each individual zombie is adjustable through setting different UDP and ICMP packet size in number of bytes. The larger the packet is, the lower the packet rate is. The TCP SYN flooding uses fixed packet-size of 64 bytes and in turn, the fixed packet rate. The maximum UDP and ICMP packet size is limited to 1024 bytes in the Stacheldraht.

Table 1. Stacheldraht Packet Size vs. Packet Rate

Packet Size	128 bytes	512 bytes	1024 bytes
UDP	66k pkt/s	21k pkt/s	12k pkt/s
ICMP	60k pkt/s	20k pkt/s	12k pkt/s
TCP SYN	Fixed 64 bytes packet size, 62k pkt/s		

In our experiment, we use the low packet rate to simulate the highly distributed attacks. Table 1 lists the packet rate of different flooding types. Due to their similar packet rate, we observed similar detection rate for TCP SYN flooding and UDP/ICMP flooding with packet size 128 bytes. Also, we achieved similar detection rate for the UDP and ICMP flooding with same packet sizes.

We implemented our DCD architecture and CAT mechanism with Java. The experimental components were then uploaded and installed on each testing node inside the DETER testbed. Each node plays the role as an independent router, which encapsulates all the related experimental parameters.

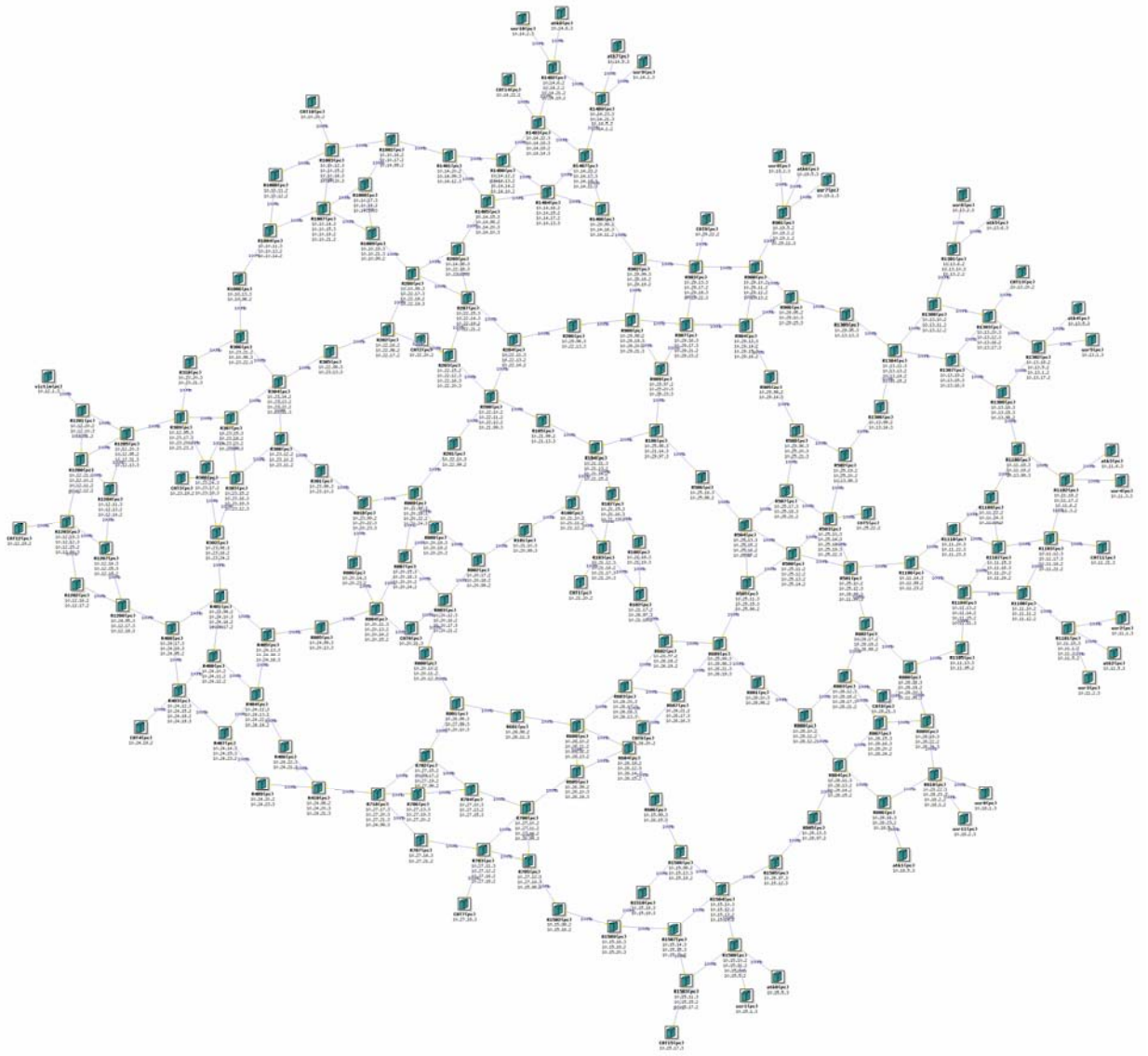


Figure 4. An experimental network topology.

4.2 Performance Evaluation Metrics

The performance of our DCD detection scheme is evaluated with two metrics: *detection rate* and *false-positive alarms*. All the metrics are measured under different DDoS attacks using TCP, UDP, and ICMP protocols. The *detection rate* R_d of DDoS attacks is defined by the following ratio:

$$R_d = a / n \qquad (1)$$

where a is the number of DDoS attacks detected in the simulation experiments and n is the total number of attacks generated by the Stacheldraht toolkit during the experiments.

In addition, we are interested in the performance of our DCD scheme under normal traffic without DDoS attacks. An alert is called a *false-positive alarm*, if an attack is detected out of normal traffic without attacks. Let p be the number of false positive alarms raised by the CAT server and m is the total number of normal traffic flow events checked by the simulator. Therefore, the ratio p/m defines the *false positive alarm rate*:

$$R_{fp} = p / m \qquad (2)$$

The ROC (*receiver operating characteristic*) curve shows the tradeoff between the detection rate and false-positive rate. Next subsection reports the detection accuracy measured under different detection thresholds. Another critical issue is the time overhead to detect the launch of

DDoS attacks. The average detection time measures from the start of a DDoS attack to the time of raising an alarm. The monitoring window should be chosen greater than this detection time.

4.3. Experimental Results

Figure 5 illustrates the variances of the detection rate with respect to different server detection thresholds (θ). Here the θ is the number of ATRs the global CAT includes. The TCP SYN attack has the highest detection rate which is close to 100% with $\theta \leq 12$.

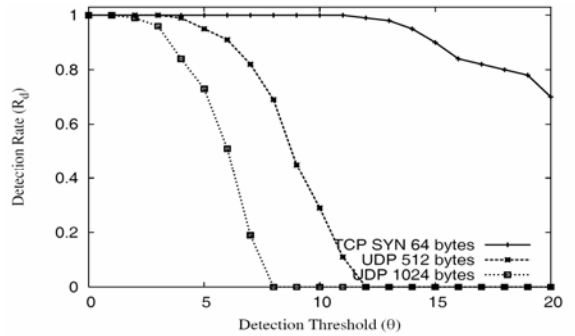


Figure 5. Effects of server threshold on the detection rate of 3 DDoS attack types.

For UDP attacks of 512-byte packets, the detection rate is still above 80% with $\theta \leq 9$. When the packet size increases to 1024 bytes, the detection rate drops to zero with $\theta \geq 7$. These results demonstrates that in order to maintain high detection rate on TCP and UDP SYN attacks, we need to set θ with a small value, such as $\theta = 5$ and adjust the packet size to 1024 bytes.

Figure 6 shows the false positive alarm rate against the CAT server threshold θ. The number of alert generated by random fluctuation in normal traffic is small and negligible. With a server detection threshold $\theta = 4$, the false positive rate drops to less than 1%.

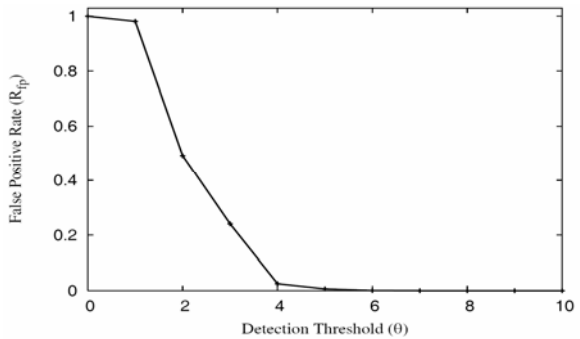


Figure 6. Effects of the threshold on false-positive rate in detecting TCP SYN attacks.

However, the real challenge lies in the fact that highly distributed attacks may use low packet rates to avoid from being detected [20]. Only after sufficient attack flows are merged, the deviation is detected by the routers. Hence, a small detection threshold value is required to achieve high detection accuracy with a low false positive rate.

The ROC curve in Fig.7 explains the tradeoff between the detection rate and false positive rate under various attacks. Our detection scheme achieves a detection rate as high as 99% with less than 1% false positive rate for high-rate DDoS attacks. Even for low-rate UDP attacks, our choice of low CAT threshold ($\theta = 3$) accomplishes a detection rate of 91% at a false-positive rate of 23%. This result proves the effectiveness of the DCD detection mechanism.

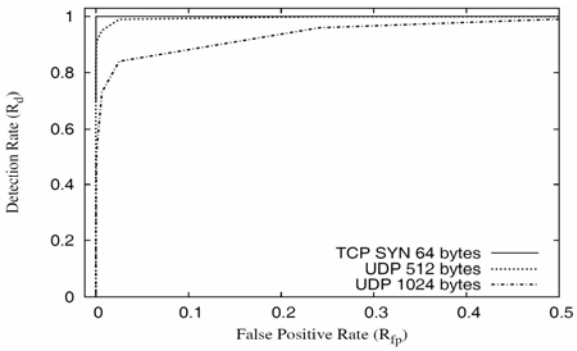


Figure 7. ROC curves showing the tradeoff between the detection rate and false-positive rate.

5 Discussions and Conclusions

In this paper, we propose a novel distributed aggregation scheme based on change-point detection across multiple network domains and verified the effectiveness of the scheme through intensive experiments on the DETER testbed. This novel scheme enables the building of an early warning system for DDoS defense across multiple ISP domains. Our DCD scheme is capable of tracing back automatically, once the detection is successfully carried out. The global CAT tree detects anomalies in real time.

Our experiment was fully tested on the DETER testbed, more than 180 nodes were used. Meanwhile, we have tried to carry out experiment in larger scale by using the testbed nodes located at UC Berkeley. Unfortunately, we failed to have all nodes working as expected. Some nodes were restarted for a couple of times and we could not synchronize them successfully.

The typical domestic Internet RTT is around 100 ms and the average global Internet RTT is 140 ms [29]. In our experiments, the monitory time window was set from 100 ms to 500 ms [7]. Therefore, the attacking flows from different agents could cross at most two monitory windows. Since the CUSUM algorithm uses the weighted running average to calculate the deviation of current traffic, it is insensitive to variant RTT times. For this reason, we did not set the delay time in the DETER experiment.

One concern regarding the CAT construction procedure is that whether subtree information can reach the CAT server of destination domain timely when the network is under high-rate attacks. Since our DCD scheme tries to detect the DDoS attacks at the earliest stage, we assumed that link bandwidth is still available by that time. This actually depends on the architecture and routing algorithms of the network. One suggested solution is to assign higher priority to CAT packets. We will study this question with more details in our future work.

Our ongoing efforts include new DDoS countermeasures and their implementation using FPGA devices. The recent advances in DDoS experiment methodologies and benchmarks [18] give us more options to design benchmark experiments closer to the reality. More objective criteria are expected to emerge to evaluate the performance of any distributed DDoS defense systems.

Acknowledgements

We would like to thank all reviewers for their valuable feedback, which not only helps us to improve this paper, but also provides us new directions in our ongoing work.

References

[1] H. Aljifri, "IP Traceback: A New Denial-of-Service Deterrent," *IEEE Security and Privacy*, May/June 2003, pp. 24-31.

[2] T. Anderson, et al, "Rocketfuel: An ISP Topology Mapping Engine," http://www.cs.washington.edu/research/networking/rocketfuel/, 2006.

[3] T. Benzel, et al, "Experience with DETER: A Testbed for Security Research", *Second IEEE Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities* (TridentCom2006).

[4] R. Blazek, et al, "A Novel Approach to Detection of DoS Attacks via Adaptive Sequential and Batch-sequential Change-Point Detection Methods," *Proc. of IEEE Workshop on Information Assurance and Security*, June 2001.

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

[6] S. Chen and Q. Song, "Perimeter-Based Defense against High Bandwidth DDoS Attacks," *IEEE Trans. on Parallel and Distributed Systems,* Vol. 16, No. 6, June 2005.

[7] Y. Chen, K. Hwang, and W.-S. Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Networks", *IEEE Transaction on Parallel and Distributed Systems*, accepted and to appear in 2007.

[8] Y. Chen and K. Hwang, "Collaborative Detection and Filtering of Shrew DDoS Attacks using Spectral Analysis," *Journal of Parallel and Distributed Computing*, Special Issue on Security in Grids and Distributed Systems, Sept. 2006, pp.1137-1151.

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

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

[11] A. Hussain, J. Heidemann, and C. Papadopoulos, "Identification of Repeated Denial of Service Attacks," *IEEE INFOCOM* 2006, Barcelona, Spain, April 23-29, 2006.

[12] A. Hussian, S. Schwab, R. Thomas, S. Fahmy, and J. Mirkovic, "DDoS Experiment Methodology," *Proc. of 2006 DETER Community Workshop*, June 15-16, 2006, Arlington, VA.

[13] K. Hwang, M. Cai, Y. Chen, and M. Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes", *IEEE Trans. on Dependable and Secure Computing,* Vol.4, No.1, Jan-March, 2007, pp.41-55.

[14] J. Ioannidis and S. M. Bellovin, "Implementing Pushback: Router-Based Defense against DDoS Attacks," *Network and Distributed System Security Symposium.* (NDSS), San Diego, CA. Feb. 6-8, 2002.

[15] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "PacketScore: Statistics-Based Overload Control Against Distributed Denial of-Service Attacks," *Proc. INFOCOM,* 2004.

[16] J. Mirkovic and P. Reiher, "D-WARD: A Source-End Defense Against Flooding DoS Attacks," *IEEE Trans. on Dependable and Secure Computing,* July 2005, pp. 216-232.

[17] J. Mirkovic, M. Robinson, P. Reiher and G. Oikonomou, "Distributed Defense Against DDOS Attacks," *Technical Report CIS-TR-2005-02*, CIS Department, University of Delaware, 2005.

[18] J. Mirkovic, E. Arikan, S. Wei, S. Fahmy, R. Thomas, and P. Reiher, "Benchmarks for DDoS Defense Evaluation," *Proc. of the Milcom 2006*, October, 2006.

[19] T. Monk 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/.

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

[21] P. Ning, S. Jajodia, and X. S. Wang, "Abstraction-based Intrusion Detection in Distributed Environment", *ACM Trans. On Information and System Security*, Nov. 2001, pp. 407-452.

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

[23] T. Peng, C. Leckie, and K. Ramamohanarao, "Detecting Distributed Denial of Service Attacks by Sharing Distributed Beliefs," *The Eighth Australasian Conference on Information Security and Privacy* (ACISP 2003), Australia, July 9-11, 2003.

[24] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, "DDoS-resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection," *IEEE INFOCOM* 2006, Barcelona, April 23-29, 2006.

[25] J. Sommers and P. Barford, "Self-Configuring Network Traffic Generation," in Proc. of ACM Internet Measurement Conference, Taormina, Sicily, Italy, Oct. 25-27, 2004.

[26] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker "DDoS Defense by Offense," *ACM SIGCOMM* 2006, Pisa, Italy, September 2006.

[27] H. Wang, D. Zhang, and K. Shin, "Change-Point Monitoring for the Detection of DoS Attacks," *IEEE Trans. on Dependable and Secure Computing,* Vol. 1, Oct.-Dec., 2004.

[28] X. Wang, S. Chellappan, P. Boyer, and D. Xuan, "On the Effectiveness of Secure Overlay Forwarding Systems under Intelligent Distributed DoS Attacks," *IEEE Trans. on Parallel and Distributed Systems,* Vol. 17, July 2006.

[29] B. Gibson, "TCP Limitations on File Transfer Performance Hamper the Global Internet," white paper, Sept. 2006, http://www.niwotnetworks.com/gbx/TCPLimitsFastFileTransfer.htm