A Coordinated Detection and Response Scheme for Distributed Denial-of-Service Attacks

Ho-Yu Lam, Chi-Pan Li, Samuel T. Chanson and Dit-Yan Yeung
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{ivanlam,cstli,chanson,dyyeung}@cs.ust.hk

***Abstract*— Distributed denial-of-service (DDoS) attacks present serious threats to servers in the Internet. They can exhaust critical resources at a target host with the help of a large number of compromised Internet hosts and hence deny services to legitimate clients. This paper studies some existing schemes for the detection and defense against TCP-based DDoS attacks. We propose a distributed scheme that can mitigate the damage caused by DDoS through a coordinated detection and response framework. This proposed scheme composes of a number of heterogeneous defense systems which cooperate with each other in protecting Internet servers. We have set up a network testbed for carrying out extensive experiments using real server machines, routers and software attack tools. Experimental results show that, compared to existing schemes, our proposed scheme can greatly improve the throughput of legitimate traffic and reduce the attack traffic during DDoS attacks. To investigate the scale-up behavior of our scheme, we have also developed a software simulator for larger-scale experiments. Simulation results show that our scheme performs consistently well even in networks with more than 3000 nodes and under high traffic load.**

I. Introduction

Nowadays, many attacks are based on the so-called *distributed denial-of-service* (DDoS) attacks in which a large volume of malicious packets are sent to exhaust the resources of a victim server. Thus, services are rendered unavailable to legitimate users. A survey published in September 2005 by Arbor Networks [1] shows that DDoS remains the most concerning threat faced by network operators. The main approach to attack detection is still through manual customer reports.

There has been no effective way so far to defend against such attacks. In this study, we propose a distributed scheme to detect and respond to a large subset of DDoS attacks. Specifically, we will focus on protecting TCP-based services only in this work, although the framework is rather general and can be extended to cover other types of DDoS attacks as well. We choose to work on TCP in this study because it is the most commonly used protocol in the current Internet [2]. Unlike many existing studies that use software network simulators only to simulate and evaluate their defense schemes, we have set up a reasonably large, real network testbed for experimentation. This allows us to observe and monitor some lower-level behaviors that are not possible with simulation experiments. This testbed consists of 46 machines connected into a hierarchical structure. Extensive experiments have been performed on the testbed using real attack tools and real servers. To see whether our scheme still works in much larger networks, we have also used a software simulator for performing larger-scale experiments. An advantage of our scheme is that it still performs reasonably well even when it is only partially deployed.

II. Related Work

Existing research on DDoS falls into three main approaches: traceback, proactive, and reactive. The *traceback approach* traces the source of the attack to the zombies (i.e., the compromised machines) [3]–[6]. While this approach is useful in the forensic sense to prevent further attacks from the same zombies, it cannot protect the victim during the DDoS attack. On the other hand, the *proactive approach* aims at preventing DDoS attacks from happening [7]–[9]. However, proactive schemes typically require modification to the existing Internet protocols or assume full adoption of the scheme to be effective.

The *reactive approach* detects the attack first and then carries out some response actions accordingly. One popular type is to filter out the attack packets [10]–[13]. Packet filtering schemes have the advantage that only the victim is required to deploy the defense system. However, a victim-side system is susceptible to high-volume attacks.

Another variation of the reactive approach takes a distributed approach. Since DDoS attacks are highly distributed in nature, we believe a distributed defense scheme is the best approach to use if possible. The scheme proposed in this paper falls into this category. To our knowledge, there are several existing proposals that use different distributed strategies [14]–[19]. In this study, we include Pushback [14], Level-k [15] and GDI [16] for comparison with our scheme. The schemes described in [17] and [18] are not included because they do not contain sufficient details for realizing them for comparison. Also, the work described in [19] focuses on a different type of DDoS attacks that may not be appropriate to be compared.

Pushback [14] attempts to solve the DDoS attack problem from a congestion control perspective. The first-hop router of the victim server detects attacks by monitoring the packet dropping history. Rate limits on traffic destined for the victim are then enforced at upstream routers hop-by-hop until a desired hop count is reached.

Level-k [15] models DDoS defense as a resource allocation problem. A notion called level-k max-min fairness was

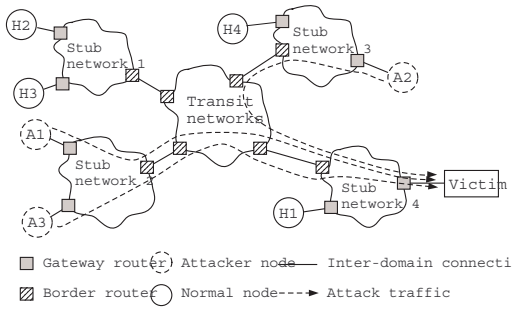


Fig. 1. Network model on which our scheme is based.

proposed to control the traffic admission rates of the routers k hops away from the victim using a max-min fairness approach.

GDI [16] employs a distributed intrusion detection system architecture. It consists of detection and traffic filter systems installed in transit network routers and only minimal traffic filter modules in stub network routers. A traffic volume anomaly detection algorithm is used to detect attack traffic and the network interfaces that contribute to the attack traffic are identified using a traffic thresholding algorithm.

III. Our DDoS Detection and Defense Scheme

Fig. 1 shows the transit-stub network model [20] on which our scheme is based. Every domain can be classified as a stub network or a transit network. A stub network is usually operated by a local ISP and mainly carries packets to and from its client hosts. A transit network interconnects stub networks. Very often the packets have to travel through several networks before getting to the destination.

An ideal DDoS defense measure is to drop all attack packets while allowing only the legitimate ones to pass through. Unfortunately, this can hardly be achieved in practice. The attackers can spoof source IP addresses to render packet filtering by source addresses ineffective. They can also generate seemingly legitimate packets to make content based filtering fruitless. The large volume of DDoS traffic that aggregates near the victim makes filtering difficult there. So, it is more desirable to do the filtering upstream instead. However, given the large number of source stub networks, it is difficult to achieve full deployment.

Therefore, we propose to set up a second line of defense in the transit networks. The number of transit networks is far less than the number of stub networks. Furthermore, transit networks are usually operated by large companies and they should have more incentive to deploy defense systems to protect their clients. The only concern is that the sheer volume of traffic flowing through the transit networks would make traffic analysis difficult, if not impossible. So, only lightweight tasks such as traffic filtering can be performed there.

A. A Coordinated Detection and Response Scheme

We propose a distributed approach based on a coordinated detection and response (CDR) scheme. The CDR scheme has two types of agents: *stub agent* (SA) and *transit agent* (TA). The SAs are located at the border routers of the stub networks for detection and response to attack flows originated from the networks. The TAs are deployed in the transit networks and perform only lightweight traffic filtering tasks. The agents are

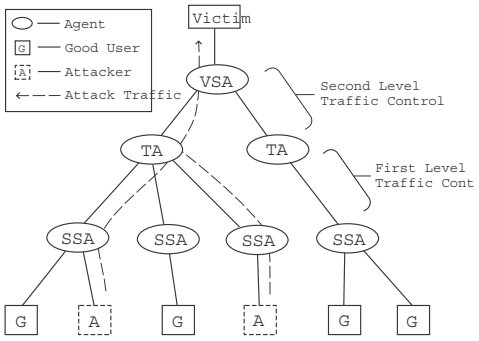


Fig. 2. Agent collaboration during an attack.

connected in a hierarchical manner. Fig. 2 shows conceptually how these agents collaborate with each other during an attack. Attack traffic and genuine user traffic originated from a stub network pass through the hierarchy of source-side SA (SSA), the TAs, and finally the victim-side SA (VSA) before reaching the victim. Details of the communication protocol used between agents are not presented in this paper since they are not our main focus here.

B. Detection Phase

1) Traffic monitoring: Attack detection is mainly performed by the stub agents (SAs). The SA is assumed to have the ability to read and modify every packet passing through the router. During normal operations, the SA observes the network traffic flowing through the router. If the SA detects a suspected victim, it will send an alert message to the nearest connected TA to start the response phase. As the traffic load between a stub network and the rest of the Internet is light compared to the traffic between transit networks, richer and more detailed analysis on network traffic can be carried out in the SAs.

The attack detection module of an SA operates in two modes: the *zombie mode* and the *victim mode*. The SA tries to detect attack traffic leaving the network in zombie mode and identify the victim inside the network in victim mode. Each SA maintains a large hash table to store traffic statistics for every distinct destination address seen in the network traffic. Each table entry consists of a number of packet counters that are updated on every packet arrival based on the information in the packet header. An entry will be deleted if no packet is sent to the associated IP address after a timeout period.

The popular technique of Bloom filters [21] can be used here to achieve space-time trade-off in hash coding. It is not unreasonable to assume that modern routers have enough resources for implementing a single hash table with $2^{24} = 16 \times 2^{20}$ entries. The entire table consumes 32MB of memory if each table entry uses two bytes to implement a counter. A simple analysis shows that a Bloom filter using $2^3 = 8$ hash tables can represent $2^{21 \times 8} = 2^{168}$ different codes, which should be more than enough for our purpose.

2) Anomaly Detection: The phenomenon of disproportionate TCP packet rates to and from an IP address is employed to detect TCP-based attacks. This idea was first proposed by Gil [22]. The guaranteed-delivery nature of the TCP protocol requires the exchange of acknowledgments (ACK) between senders and receivers. Therefore, for normal TCP communi-

cations, the number of packets sent to and received from a host should be balanced. On the contrary, a zombie that floods a victim will hardly receive any proper ACK packet.

We have adopted the nonparametric version of the sequential change-point detection algorithm known as CUSUM (cumulative sum) [23], [24] to detect TCP rate anomaly. This algorithm is a statistical tool that is based on finding the time of switching from one state (normal) to another state (attack) in a time series. Without going into details that are covered in [25], let us define the ratio $X_n^D = toTCP^D/fromTCP^D$ where $toTCP^D$ and $fromTCP^D$ denote the numbers of TCP packets destined to and sent from the IP address D during the monitoring time interval Δ_n, respectively. Note that the $fromTCP^D$ counter does not count SYN/ACK and RST packets because there are relatively few of them in normal TCP flows but are numerous in abnormal flows. Next, we define $Z_n^D = X_n^D - \beta$ where β is an upper bound of X_n^D in normal network conditions, which in most situation is 3. Finally, the decision function H^D is defined to determine whether an attack targeting victim D has occurred:

$$y_n^D = S_n^D - \min_{1 \le k \le n} S_k^D \tag{1}$$

$$H_N^D(y_n^D) = \begin{cases} 0 & \text{if } y_n^D \le N; \\ 1 & \text{if } y_n^D > N, \end{cases} \tag{2}$$

where $S_k^D = \sum_{i=1}^{k} Z_i^D$, with $S_0^D = 0$; and N is the threshold for attack detection. The decision result '1' indicates an attack while '0' indicates a normal condition. Intuitively, using a large value for N requires a longer detection time. However, using a small value for N may give rise to a high false alarm rate. A suggested setting for N is 3β [26].

C. Response Phase

The response phase is activated when a DDoS attack is detected. In the response phase, coordination among agents takes on two levels as shown in Fig. 2.

1) First-Level Traffic Control: When an SA detects an attack, it notifies its nearest TA the identity of the victim. The TA then broadcasts the victim's identity and a *good mark* (as a bit pattern) to all SAs. All SSAs will then identify attack packets destined to the victim and stamp good packets, in either the 16-bit IPID field or the IP option field, using the good mark. The stamp is vital in avoiding good packets from being affected by the second-level traffic control.

To identify and rate-limit attack packets, SAs will monitor all subsequent TCP streams that are destined to the victim. IP addresses that send disproportionate TCP traffic to the victim will be marked as bad. For each monitoring period Δ_m, TCP streams that are destined to the victim from bad addresses will have their rate-limits tightened by a constant, whereas those streams from good addresses will have their rate-limits relaxed by γ/i, where γ is a constant parameter and i is the number of increases in the rate-limit. This multiplicative decrease in relaxation rate ensures that repeatedly unresponsive TCP streams will eventually enjoy no relaxation.

Unfortunately, if an attacker applies source IP address spoofing, it is impossible for an SA to identify the real bad source. To tackle this problem, SAs also keep a counter on the number of new IP addresses that have sent packets to the victim within a time window. If this counter exceeds a certain threshold, the SA will suppress traffic from new IP addresses. In the long term, proactive measures should be taken to prevent packets with spoofed IP address from entering the Internet.

2) Second-Level Traffic Control: The VSA forms a second line of defense by collaborating with upstream TAs to rate-limit attack traffic through a feedback control mechanism. During the response phase, every TA marks the packets passing through it by its unique ID. This mark can even be dependent on the ingress interface so that higher differentiability can be achieved. Based on statistics of the marked packets, the VSA informs the TAs of the admission rates.

TCP connection tracking techniques, which are widely used in firewall systems, are used to identify bad connections. Using a large hash table, the VSA monitors all connections related to the victim. Bad connections are those that cannot complete the three-way handshake process. All packets belonging to a bad connection are classified as bad packets. The corresponding admission rate for a TA's interface IF is derived based on the policy, $\max(R_{\min}, R_{\max} \times g/T)$, where $R_{\max}$ and $R_{\min}$ are the maximum and minimum admission rates configured by the VSA, g is the number of good packets coming from IF, and T is the total number of packets coming from IF. It is clear that if many bad packets are coming from IF, its admission rate will drop to $R_{\min}$ quickly. A possible way to set $R_{\max}$ is based on the total desired inbound bandwidth divided by the total number of TA interfaces. Note that rate-limiting is not applied to the good packets marked by SSAs. Thus, good users of an SA-equipped network are not affected. However, for those good users of a network without SA, the chance of collateral damages is high if their packets share some paths with the attack traffic.

IV. Experiments on Testbed and Simulator

We have performed experiments based on both a physical testbed and the software simulator ns2 [27]. The setup details of the testbed and simulator can be found in [25]. A brief description of the setup is presented below.

A reasonably large network testbed consisting of 46 machines as shown in Fig. 3 has been set up for this study. Each software router (R1-R15) is connected with two end hosts to form a small network domain. These routers are interconnected to each other in a hierarchical structure with a Cisco hardware router as backbone to simulate the Internet. Network traffic generators for the three dominant TCP-based services: Web, FTP and E-Mail, which together account for the majority of the TCP traffic in the Internet [28], were developed to simulate background traffic in the testbed.

As most victims in DDoS incidents reported are Web servers, the Apache Web Server has been installed in the victim machine to serve 10 web pages and all their embedded

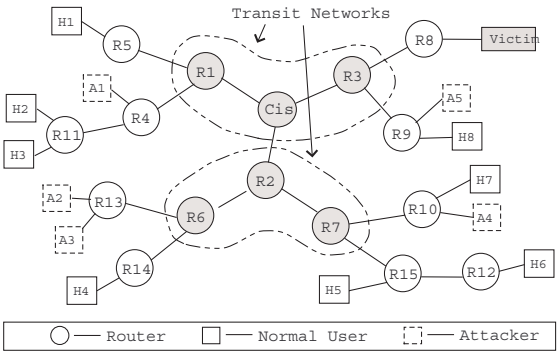


Fig. 3. Network topology of our testbed. "Cis" denotes the Cisco router.

objects downloaded from several well-known web sites. The ingress link bandwidth of the victim is set to 10Mbps.

Each good user host in Fig. 3 runs 10 simple Web client programs in parallel that randomly retrieve a page and its embedded objects from the victim server every few seconds. A customized attack tool based on the freely available attack tools [29] is used as attackers.

Each attack experiment was run for three minutes. The good users started accessing the victim immediately after the experiment commenced. Then, after one minute, the attackers sent attack packets to the victim in the remaining two minutes.

V. Testbed-Based Experimental Results

To compare the behavior and performance of CDR with other existing schemes (i.e., GDI, Level-k and Pushback), we tested all four schemes in our testbed. A 14Mbps bandwidth attack with source IP address spoofed was generated to test the four schemes.

Our first set of experiments was designed to investigate the effectiveness of the schemes under full deployment.

Fig. 4 shows the general response behavior of every scheme over time (attack started at time 60s). It can be seen clearly that the CDR scheme performed significantly better than the other three schemes. By successfully filtering the attack traffic in the source side, the CDR scheme rescued most of the legitimate traffic while some was filtered due to collateral damages in stub networks R9 and R10 (see Fig. 3), where both good users and attackers resided in the same subnet.

In the next set of experiments, we studied the impact of deployment level on the effectiveness of DDoS defense schemes. Two performance metrics were considered in these experiments:

- *Good throughput percentage* (GTP): The percentage of the average good inbound throughput during attack divided by that before attack.
- *Attack throughput percentage* (ATP): The percentage of the average attack traffic throughput divided by the victim inbound bandwidth (10Mbps in our testbed).

All four schemes were tested with decreasing deployment levels. For the CDR and GDI schemes, the deployment level was divided into seven degrees d, from 0 to 6. The defense agents of $d \times 2$ randomly selected nodes were turned off. Degree 0 is full deployment. One exception was that the CDR scheme was ensured to have at least one TA and the VSA (R8) was deployed in all deployment levels as required by the second-level traffic control. The deployment levels for Level-k and Pushback are divided into five degrees from 0 to 4. Degree 0 means full deployment for Pushback and $k = 5$ for Level-k, respectively. Their deployment was decreased by one hop for each deployment degree.

Attack name	Packet type	Rate (packets/sec)	Bandwidth (bits/sec)	Spoof
High-BW-S	random TCP	15000pps	14Mbps	Yes
Low-BW-S	random TCP	5000pps	5Mbps	Yes
High-BW	random TCP	15000pps	14Mbps	No
Low-SYN	TCP SYN	5000pps	4.5Mbps	Yes
High-SYN	TCP SYN	15000pps	1.5Mbps	Yes

TABLE I

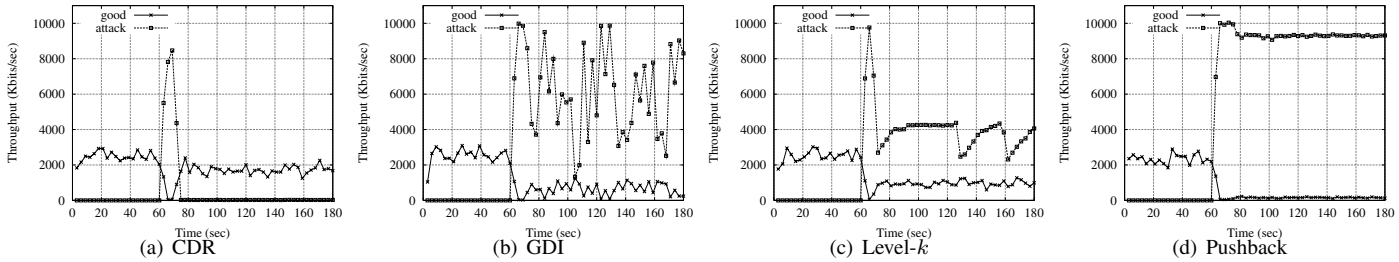
Attacks performed in our experimental studies.

Results shown in Fig. 5 are the average results over five sets of experiments. From Fig. 5(a), we observe that the GTP performance of the CDR scheme only degrades slowly with deployment level. Even when the deployment level is decreased to degree 3, the GTP only decreases slightly to 70% compared to 80% in degree 0. In addition, about 55% of good traffic is saved in degree 4, which may not be possible without the coordination strategy of the CDR scheme. This outcome demonstrates that the CDR scheme successfully attains the objective of good DDoS defense capability even under partial deployment. On the other hand, as the deployment level further decreases, the GTP drops quickly. This is because very few SAs are installed so fewer good packets are stamped with good marks and more attack packets are only throttled in the transit networks. The combined effect results in more severe collateral damages in the second-level traffic control stage. Readers may refer to [25] for analysis of performance of the other schemes.

It can be seen that all schemes generally show degraded performance in protecting good user traffic as the deployment level is reduced. It is hard to conclude which scheme is more suitable for partial adoption as each scheme has very different deployment requirements. For example, GDI and CDR deploy defense systems distributively in the transit and stub networks; Level-k requires all routers k hops away to participate in the defense; and Pushback requires full deployment in consecutive routers along all possible paths leading from the victim. However, we argue that our proposed CDR scheme and the GDI scheme are more flexible for deployment in the Internet. The rigid deployment criteria of Pushback and Level-k are not easily achievable in practice.

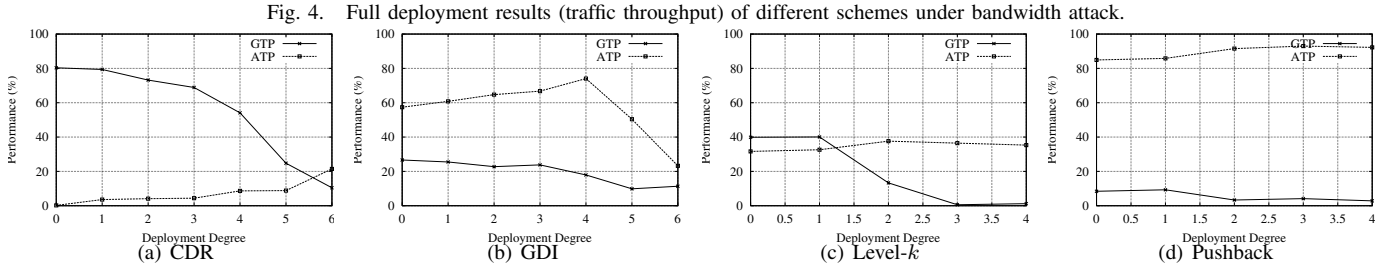
In the third set of experiments, five popular types of TCP-based attacks were tested and they are summarized in Table I. We studied the performance of the CDR scheme at a deployment degree of 3 and other schemes at their *full* deployment. The experiments were repeated five times and the average results are given in Table II. Note that some schemes simply cannot detect the occurrence of certain types of attacks (labeled 'N/A' in the table).

From the results, none of the three schemes was able to perform better than even the degree 3 deployment results of the CDR scheme. However, our primary objective is to show



(a) CDR (b) GDI (c) Level-k (d) Pushback

Fig. 4. Full deployment results (traffic throughput) of different schemes under bandwidth attack.



(a) CDR (b) GDI (c) Level-k (d) Pushback

Fig. 5. Partial deployment results for different schemes.

that non-bandwidth type attacks cannot simply be countered by traffic volume detection mechanisms. For instance, the low bandwidth consumption of Low-BW-S, Low-SYN and High-SYN attacks caused no link congestion, thus failing to trigger the Pushback scheme. The GDI and Level-k schemes both failed to detect the Low-SYN attack as the aggregated attack traffic is only 1.5Mbps (1000pps) which is even less than the bandwidth consumed by the good users (about 2.5Mbps). In contrast, all attacks were detected by the CDR scheme in all the experiments performed.

VI. Simulator-Based Experimental Results

The effectiveness of full deployment of the CDR scheme was tested on a hierarchical network with 3060 traffic generating nodes distributed over 12 stub networks.

The simulated good users started sending data to the victim as the simulation began while the simulated attackers had random start times between 85 and 95 seconds. The large number of legitimate users generated enough traffic that together consumed most of the victim's bandwidth, while attackers generated more than 1.3Gbps of unresponsive attack traffic to congest the 100Mbps victim link.

Fig. 6(a) shows that under large volume of attack traffic without IP address spoofing, the CDR scheme can effectively filter bad traffic and restore good traffic within 20 seconds after the attack starts. The peaks in the bad traffic throughput after the scheme has started to filter are caused by the relaxation mechanism in the SAs as described in Section III-C.1. The peaks appear less frequently over time and eventually vanish as expected, although they are not shown here due to space limitation.

Fig. 6(b) shows the full deployment effectiveness of the CDR scheme when attackers use spoofed source IP addresses. 95% of the good traffic remains unaffected. It remains consistently well compared to the testbed experiments. Interested readers are referred to [25] for more analysis on factors that may affect the performance of the CDR scheme under high volume of attack traffic.

The second set of simulation experiments is to study the partial deployment effectiveness. A topology with 30 transit networks and 90 stub networks were used to allow enough variations in deployment configurations. Ten traffic generating nodes were connected to each stub network with 20Mbps duplex links, resulting in a topology with 1020 nodes. Bandwidth attacks of 100Mbps with IP spoofing were performed. For each deployment degree d, $(d \times 10)\%$ of the defense agents were randomly selected to be turned off. Other requirements are the same as those in the testbed partial deployment experiments.

As can be seen in Fig. 7(a), the general trend of performance degradation is consistent with the testbed experimental results. Attack traffic that could have consumed all of the victim's bandwidth can be very effectively suppressed to a low level in all deployment degrees. With the concerted effort of SAs and TAs, in almost all cases, the GTP remains higher than the ATP. This once again demonstrates that the CDR scheme can still work quite satisfactorily even under partial deployment.

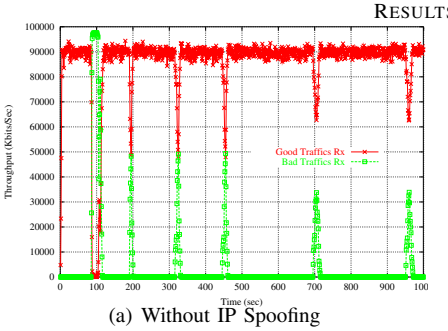
VII. Conclusion

DDoS attacks can cause severe disruption to the stability of the Internet. In this paper, we have presented a distributed scheme to detect and defend against TCP-based DDoS attacks. Through a two-level traffic control architecture, our scheme effectively thwarts DDoS attack traffic even under partial deployment. The coordination between SSA, TA and VSA proposed in this paper reduces collateral damages and facilitates effective suppression of attack traffic while requiring TAs to do only lightweight filtering tasks. A real network testbed has been used for realistic performance comparison with three other schemes. In addition, larger-scale experiments based on a software simulator have been used for studying the scalability of our scheme. Testbed experimental results show that our proposed scheme outperforms existing schemes in protecting TCP-based servers while minimizing collateral damages to legitimate traffic. Simulation results show that the scheme can still perform consistently well on much larger networks and under higher traffic load.

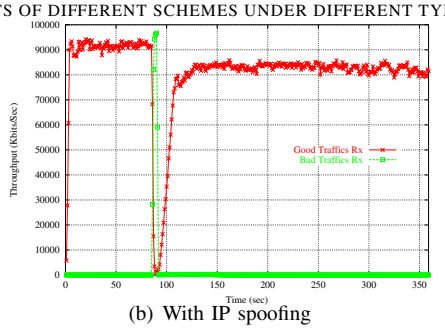
	GTP (%)					ATP (%)				
	No Defense	*CDR*	*GDI*	*Level-k*	*Pushback*	*No Defense*	*CDR*	*GDI*	*Level-k*	*Pushback*
High-BW-S	3.08	**62.01**	27.42	38.68	8.08	91.91	**6.76**	63	34.45	86.55
Low-BW-S	41.79	**75.12**	56.26	35.27	**N/A**	41.74	**4.69**	26.77	33.25	**N/A**
High-BW	4.01	**75.01**	62.64	47.82	10.32	92.34	**18.59**	44.06	31.42	86.08
Low-SYN	40.54	**61.85**	**N/A**	**N/A**	**N/A**	12.48	**2.38**	**N/A**	**N/A**	**N/A**
High-SYN	4.90	**46.30**	4.60	2.63	**N/A**	39.99	**10.05**	31.17	38.69	**N/A**

TABLE II

RESULTS OF DIFFERENT SCHEMES UNDER DIFFERENT TYPES OF ATTACKS



(a) Without IP Spoofing



(b) With IP spoofing

Fig. 6. Full deployment effectiveness

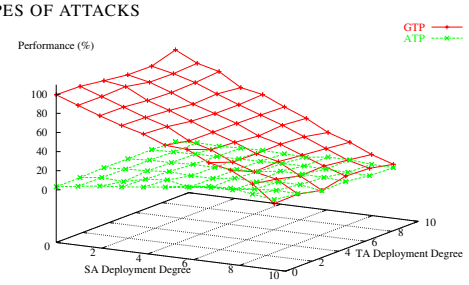


Fig. 7. Partial deployment effectiveness

ACKNOWLEDGMENT

This research has been supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China under the Area of Excellence scheme (Project No. AoE/E-01/99).

REFERENCES

[1] Arbor Networks, "Worldwide ISP security report," Sept. 2005. [Online]. Available: http://www.arbor.net/downloads/Arbor_Worldwide_ISP_Security_Report.pdf

[2] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proc. of the ACM SIGCOMM '03*, Karlsruhe, Germany, Aug 2003.

[3] S. Savage, D. Wetherall, A. R. Karlin, and T. Anderson, "Network support for IP traceback," *ACM/IEEE Transactions on Networking*, vol. 9, no. 3, pp. 226–237, June 2001.

[4] D. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proc. of the IEEE INFOCOM '01*, vol. 2, April 2001, pp. 878–886.

[5] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer, "Single-packet IP traceback," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 721–734, December 2002.

[6] J. Li, M. Sung, J. Xu, and L. Li, "Large-scale IP traceback in high-speed internet: practical techniques and theoretical foundation," in *Proc. of the 2004 IEEE Symp. on Security and Privacy*, May 2004, pp. 115–129.

[7] K. Park and H. Lee, "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack," in *Proc. of the IEEE INFOCOM '01*, vol. 1, April 2001, pp. 338–347.

[8] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," in *Proc. of 10th IEEE International Conference on Network Protocols*, Nov 2002, pp. 312–321.

[9] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure overlay services," in *Proc. of the ACM SIGCOMM '02*, Pittsburgh, Pennsylvania, USA, Aug 2002, pp. 61–72.

[10] M. Sung and J. Xu, "IP traceback-based intelligent packet filtering: A novel technique for defending against Internet DDoS attacks," in *Proc. of 10th IEEE International Conference on Network Protocols*, Nov 2002, pp. 302–311.

[11] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against DDoS attacks," to appear in *Proc. of the IEEE Symposium on Security and Privacy*, May 2003.

[12] R. Thomas, B. Mark, T. Johnson, and J. Croall, "NetBouncer: Client-legitimacy-based high-performance DDoS filtering," in *Proc. of the DARPA Information Survivability Conference and Exposition*, vol. 1, April 2003, pp. 14–25.

[13] J. Kang, Z. Zhang, and J. bin Ju, "Protect e-commerce against DDoS attacks with improved D-WARD detection system," in *Proc. of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, March 2005, pp. 100–105.

[14] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *Technical Report, AT&T Center for Internet Research at ICSI*, July 2001.

[15] D. K. Y. Yau, J. C. S. Lui, and F. Liang, "Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles," in *Proc. of the IEEE International Workshop on Quality of Service*, May 2002, pp. 35–44.

[16] K. K. K. Wan and R. K. C. Chang, "Engineering of a global defense infrastructure for DDoS attacks," in *Proc. of the IEEE International Conference on Networks*, Aug 2002, pp. 419–427.

[17] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, "COSSACK: Coordinated suppression of simultaneous attacks," in *Proc. of the DARPA Information Survivability Conference and Exposition*, vol. 1, April 2003, pp. 2–13.

[18] J. Mirkovic, M. Robinson, and P. Reiher, "Alliance formation for DDoS defense," in *NSPW '03: Proc. of the 2003 workshop on New security paradigms*. New York, NY, USA: ACM Press, 2003, pp. 11–18.

[19] H. Sun, J. C. Lui, and D. K. Yau, "Defending against low-rate TCP attacks: dynamic detection and protection," in *Proc. of the 12th IEEE International Conference on Network Protocols*, Oct 2004, pp. 196–205.

[20] K. Calvert, M. Doar, and E. W. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, vol. 35, pp. 160–163, June 1997.

[21] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[22] T. Gil and M. Poleto, "MULTOPS: A data-structure for bandwidth attack detection," in *Proc. of the 10th Usenix Security Symposium*, August 2001, pp. 23–28.

[23] R. B. Blazek, 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," in *Proc. of the 2nd Annual IEEE Systems, Man, and Cybernetics Information Assurance Workshop*, West Point, NY, June 2001.

[24] B. E. Brodsky and B. S. Darkhovsky, "Nonparametric methods in change-point problems," *Kluwer Academic Publishers*, 1993.

[25] H. Lam, C. Li, S. Chanson, and D. Yeung, "A coordinated detection and response scheme for distributed denial-of-service attacks," Department of Computer Science, Hong Kong University of Science and Technology, Tech. Rep. Technical Report HKUST-CS06-01, March 2006, `ftp://ftp.cs.ust.hk/pub/techreport/06/tr06-01.pdf`.

[26] H. Wang, D. Zhang, and K. G. Shin, "SYN-dog: Sniffing SYN flooding sources," in *Proc. of the IEEE International Conference on Distributed Computing Systems' 2002*, Vienna, Austria, July 2002, pp. 421–428.

[27] L. N. R. Group. (1997, Sept.) UCB/LBNL/VINT network simulator–ns (version 2). [Online]. Available: http://www-mash.cs.berkeley.edu/ns/

[28] MAWI working group traffic archive. [Online]. Available: http://tracer.csl.sony.co.jp/mawi

[29] DDoS attack tools (TFN, TFN2k and Trinoo) download website. [Online]. Available: http://www.angelfire.com/rock/nsi/