Cooperative Mechanism against DDoS Attacks

Guangsen Zhang, Manish Parashar
The Applied Software Systems Laboratory
Department of Electrical and Computer Engineering
Rutgers University
{gszhang,parashar}@caip.rutgers.edu
Voice:(732) 445-5388, Fax:(732) 445-0593

Abstract

Distributed denial of service (DDoS) attacks on the Internet have become an immediate problem. As DDoS streams do not have common characteristics, currently available intrusion detection systems (IDS) can not detect them accurately. In this paper, we propose a distributed approach to detect distributed denial of service attacks by coordinating across the Internet. Unlike traditional IDS, we detect DDoS at the intermediate network. Our scheme uses a nonparametric point detection method to improve the detection accuracy at each individual node. Then, a gossip based multicast mechanism is used to exchange information between the individual nodes to further improve the detection accuracy. To provide reliable, rapid and widespread dissemination of attack information, the system is built as an overlay network on top of the internet. Initial results using simulation illustrate that the proposed approach is both efficient and feasible.

Keywords: Security, DDoS, Gossip, Correlation, Overlay Network.

1 Introduction

A Distributed Denial of Service (DDoS) attack is a large-scale, coordinated attack on the availability of services at a victim system or network resource. The DDoS attack is launched by sending an extremely large volume of packets to a target machine through the simultaneous cooperation of a large number of hosts that are distributed throughout the Internet. The attack traffic consumes the bandwidth resources of the network or the computing resource at the target host, so that legitimate requests will be discarded. The impact of these attacks can vary from minor inconvenience to the users of a web site, to serious financial losses to companies that rely on their on-line availability to do business [18, 20].

DDoS attack is likely to become an increasing threat to the Internet due to the easy availability of user-friendly attack tools, which help to coordinate and execute a large scale DDoS attack. Even an unsophisticated individual can launch a devastating attack with the help of these tools. Available tools include Trinoo, TFN, TFN2K, Shaft, and Stacheldraht and have been used in DDoS attacks against well-known commercial web-sites, such as Yahoo, Amazon, Ebay [4].

A key problem when addressing DDoS attacks is attack detection. Several DDoS detection systems have been proposed. Most of them can be classified as either signature based or abnormal behavior based mechanism. As DDoS attacks have no common attack signatures and a sophisticated or experienced attacker may change the attack pattern frequently, detecting attacks by performing pattern matching against a database of known attack signatures is not feasible. However, DDoS traffic generated by today's tools often has characteristics that make it possible to distinguish it from normal traffic using statistical measurement [23, 24, 1]. This abnormal behavior can be used to define methods to improve the detection accuracy at each individual node. Generally it is easy to detect the abnormal behavior of attack near the victim. However, it is also often too late to detect the DDoS attack at the victim network. The attack should ideally be stopped as close to the sources as possible, saving network resources and reducing congestion. However, there are no common characteristics of DDoS streams that can be used to detect the attacks near the source [3]. To balance this tradeoff, we try to detect the DDoS attacks in the intermediate network.

As the traffic is not aggregated enough in the intermediate network, current single deployment detection systems can not detect DDoS attacks with high accuracy. To improve the detection accuracy, we need a paradigm shift. Instead of building detection systems that operate in isolation, we need to build a distributed framework of detection nodes where heterogeneous systems can plug in and cooperate to achieve a better overall detection. The detection nodes are core routers with augmented functionality, which can monitor traffic and cooperate with other nodes as an overlay network. Traditional Intrusion Detection Systems (IDS) result in high false alarms when used to detect DDoS attacks. By cooperation, we can improve the accuracy of DDoS detection. However, given the large number of nodes in the Internet, we need a scalable and efficient self organizing architecture to share the information among the individual detection systems.

The primary contribution of this paper is to propose a global detection infrastructure by building an overlay network on top of the internet. This infrastructure provides reliable, rapid and widespread cooperation among individual detection nodes to improve the accuracy of DDoS detection in the intermediate network. Given the large scale of the internet and crucial purpose of this infrastructure, we need reliable and scalable communication mechanism to exchange the attack information. We design directional gossip mechanisms to fulfill this purpose while reducing the overhead of information sharing. To improve each individual node detection accuracy, we adapt the detection scheme proposed by Wang et al. [23], which is based on an advanced non-parametric change detection scheme, CUSUM. Initial results using the simulation illustrate that the proposed approach is both efficient and feasible.

The rest of the paper is organized as follows. Section 2 gives an overview of DDoS. Section 3 explains our approach. Section 4 presents an experimental evaluation. Section 5 discusses related work and Section 6 concludes the paper.

2 DDoS Background

Distributed denial of service attacks (DDoS) pose a great threat to the Internet. A recent DDoS attack occurred on October 20, 2002 against the 13 root servers that provide the Domain Name System (DNS) service to Internet users around the world. Although the attack only lasted for an hour and the effects were hardly noticeable to the average Internet user, it caused 7 of the 13 root servers to shut down demonstrating the vulnerability of the Internet to DDoS attacks [7]. Distributed denial of service attacks occur when numerous subverted machines (zombies) generate a large volume of coordinated traffic toward a target, overwhelming its resources. DDoS attacks are advanced methods of attacking a network system to make it unavailable to legitimate network users. These attacks are likely to become an increasing threat to the Internet due to the convenience offered by many freely available user-friendly attack tools. Furthermore, attackers need not fear punishment, as it is extremely difficult to trace back the attack and locate even the agent machines, let alone the culprits who infected them.

2.1 DDoS Attack Classification

There are two main classes of DDoS attacks: bandwidth depletion and resource depletion attacks. A bandwidth depletion attack is designed to flood the victim network with unwanted traffic that prevent legitimate traffic from reaching the victim system. A resource depletion attack is an attack that is designed to tie up the resources of a victim system. This type of attack targets a server or process at the victim making it unable to legitimate requests for service [11].

2.2 DDoS Characteristics

There are several features of DDoS attacks that hinder their successful detection and defense:

- DDoS attacks generate a large volume flow to overwhelm the target host. The victim can not protect itself even if it detects this event. So the detection and defense of DDoS should ideally be near the source of the attack or somewhere in the network.

- It is difficult to distinguish attack packets from legitimate packets. Attack packets can be identical to legitimate packets, since the attacker only needs volume, not content, to inflict damage. Furthermore, the volume of packets from individual sources can be low enough to escape notice by local administrators. Thus, a detection system based on single site will have either high positive or high negative rates.

- Most DDoS attacks use spoofed IP source addresses. This is done primarily to disguise agent machines, but it can also be used as a means to perpetrate reflector attacks. Due to the large scale of a DDoS attack, it is impossible to locate agent machines [3].

- DDoS traffic generated by available tools often has identifying characteristics, making the detection based on statistics analysis possible. However, given the inherently busty nature of Internet, detecting DDoS attacks is error prone.

2.3 A Taxonomy of the DDoS Detection and Defense

Based on the underlying strategies used, we can categorize current DDoS detection and defense approaches into three categories: Proactive Mechanisms, Reactive Mechanisms and Post Attack Analysis.

Proactive defense mechanisms. [13] The motivation for these approaches is based on the observation that it is hard to detect DDoS attacks. So instead of detecting the attacks by using signatures (attack pattern) or anomaly behavior, these approaches try to improve the reliability of the global Internet infrastructure by adding extra functionality to Internet components to prevent attacks and vulnerability exploitation. The primary goal is to make the infrastructure immune to the attacks and to continue provide service to normal users under extreme conditions.

Reactive defense mechanisms using available IDS. [12, 6] These mechanisms typically deploy third-party Intrusion Detection Systems (IDS) to obtain attack information and take action based on this information. Consequently their usefulness depends on the capability of

Top

the IDS systems. Different strategies are used based on the assumptions made by the IDS systems. If the IDS system can detect the DDoS attack packets accurately, filtering mechanism are used, which can filter out the attack stream completely, even at the source network. If the IDS can not detect the attack stream accurately, rate limiting is used. This mechanism imposes a rate limit on the stream that is characterized as malicious by the IDS.

Post attack analysis. [22] The purpose of post attack analysis is to either look for attack patterns that will be used by IDS or identify attackers using packet tracing. The goal of packet tracing is to trace Internet traffic back to the true source (not spoofed IP address). As attackers change their strategy frequently, analyzing huge amount of traffic logs is time consuming and useless in detecting new attacks. Trace back mechanism can help to identify zombies in some situations, however, it is impractical to defend against DDoS attacks for the following reasons. First, during a DDoS attack, the attacker will control thousands of zombies (numbers will increase in the future) to launch an attack. As a result, identifying these zombies is expensive and infeasible. Second, since different network administrators control different section of the global Internet, it would be difficult to determine who would be responsible for providing trace back information.

3 Cooperative DDoS Detection Approach

As the distributed denial of service (DDoS) attack traffic transmits across the Internet towards the victim, the victim can easily detect the attacks by observing degraded services. However, it is too late to defend against DDoS attacks near the victim as the victim resources would be heavily loaded and would not be able to react to the DDoS attacks. The attacks should ideally be stopped as close to the sources as possible, saving network resources and reducing congestion. However, there are no common characteristics of DDoS streams that can be used to detect and filter them near the source. Our strategy is to defend the DDoS attacks in the intermediate network. We make the assumption that in the intermediate network, the aggregated attack flows toward the victim consume more bandwidth than aggregated normal flows to the victim. As the aggregate does not cause congestion in the network, and it is hard to detect the DDoS attacks in a single domain, we propose that by sharing information across domains distributed in the network, we can detect the DDoS attacks early.

Our detection mechanism includes two key stages. In the first phase, each local detection node detects traffic anomalies using profiles of normal traffic constructed using stream sampling algorithms. Due to the dynamic nature of the Internet, detections based on this mechanism alone will have high false positives. In the second phase, we enhance the accuracy of the detection by using gossip based multicast to share information among individual nodes.

To enhance the security and reliability of information sharing, our system is built on an overlay network composed of local detection nodes, which are routers with DDoS detection and attack packets filtering functionality. An overlay network is an isolated virtual network deployed over an existing network [2]. It is composed of routers, and tunnels. Tunnels are paths in the base network, and links in the overlay network. Individual components (routers) can participate in more than one overlay at a time or in a single overlay in multiple ways. As a result, wherever there is a physical path in the underlying network, there can exist a link in the overlay network. Having multiple available links increases the flexibility of the network, and a more flexible network is less likely to be susceptible to attacks. Furthermore, by building a large-scale, self-organizing and resilient overlay network on top of the Internet, the peer nodes of the overlay network can deliver attack information with speed and reliability [15].

We assume that the Internet is a set of Autonomous Systems (AS). Individual detection

Top

nodes are located at the egress routers of the Autonomous System, which collect some meaningful information and detect DDoS attacks locally. Then the system will use the overlay network to share the detection information using gossip protocol based on epidemic algorithm across the Internet. This is illustrated in the Figure 1.

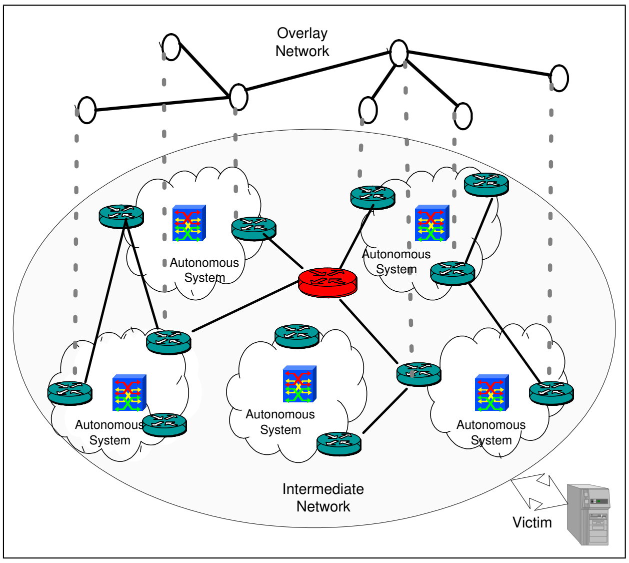

Figure 1: System architecture

The internals of an individual detection node can be fairly complex, but conceptually it can be structured into six pieces, as shown in the Figure 2. The traffic measurement module is responsible for measuring local traffic. Next, the local detection mechanism will use this data to detect any local anomaly. This local decision will be sent to the cooperative detection engine, which will combine this local decision with the decisions from neighboring nodes, using the message dissemination module, to make a global detection decisions. Finally, the detection decision module will inform the local response module to take action to defend against an attack.

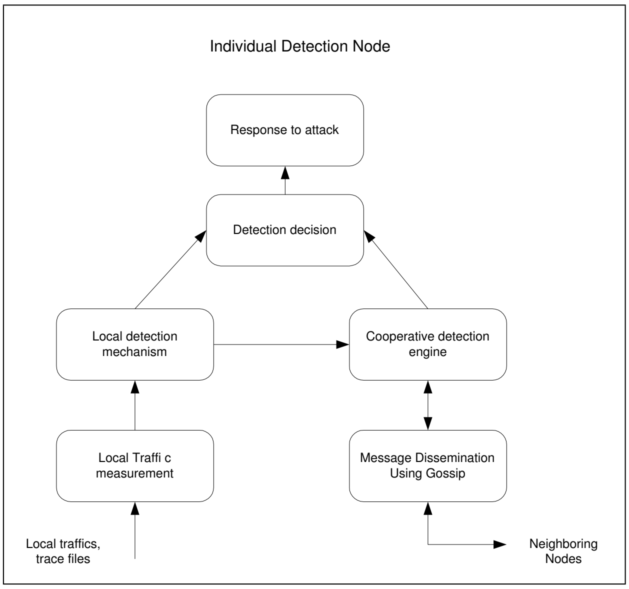

Figure 2: A conceptual architecture for an individual detection node

Top

3.1 Attack Detection Procedure

In our approach, the egress routers of the intermediate network coordinate with each other to provide the information necessary to detect and respond to the attack. The mechanism can improve the accuracy and speed of detecting the DDoS attacks. The operations on these egress routers are described below:

- The local detection node keeps traffic statistics for high-traffic destinations using sample-and-hold algorithms. It uses change detection algorithm, e.g., CUSUM [23] to locally detect abnormal behavior.
- When each individual node detects a possible DDoS attack, it will share this information with other nodes using a gossip mechanism. If each node collects enough information and knows that other nodes have detected the same attack, it will confirm the detection.
- Detection information is combined with a local timestamp. Individual node will check the timestamp and discard expired detection messages. Detail of this mechanism is discussed in [14].
- When an individual node confirms the DDoS attack, it will deploy countermeasures to prevent continuance of the attack.

Our approach can be combined with available mitigating or rate limit technologies to eliminate the attack before it has done much damage.

3.2 Local Abnormal Behavior Detection

To enable local abnormal behavior detection, we need to define the data that routers will collect using a statistical measurement method. As the high-traffic destinations are most likely to be under attack, it is reasonable to keep traffic statistics only for those high traffic flows that have the same destination IP addresses. We can use a *sample-and-hold* [5, 1] algorithm to let the egress routers keep track of destinations whose traffic occupies greater than a fraction *r* of the capacity *C* of the outgoing link. We call these destinations popular and destinations not in this list as unpopular.

Traffic profiles at each router are essentially a set of metrics M_i for the traffic to popular destinations. An effective choice of such metrics is key to characterizing traffic streams. However, computing arbitrary fingerprints might require excessive memory and computation. Several metrics have been proposed by research community. Some of them are:

- The fraction of new source IP addresses.
- The ratio of traffic between the two directions.
- An approximation of the flow-length distribution of traffic to the destination.

Based on these metrics, there are different abnormal behavior based detection approaches, such as those described in [10, 8]. In this paper, we use CUSUM to detect abnormal behavior [23]. Let X_n represent one of these metrics during time interval Δ_n. The main idea is that, during an attack, for the random sequence X_n, there is a step change in the mean value $E(X_n)$. The non-parametric CUSUM is asymptotically optimal for such Change Point Detection problems. This general approach is based on the model presented in Wang et al. [23] for attack detection using CUSUM. One of the assumption for the nonparametric CUSUM algorithm is that the mean value of the random sequence is negative during normal conditions, and becomes positive when a change occurs. In general, $E(X_n) = c \ll 1$. We choose a parameter γ that is the upper bound of c, i.e., $\gamma > c$. Thus without loss of any statistical feature, X_n is transformed into another random sequence Y_n with negative mean b during normal operation, i.e., $Y_n = X_n - \gamma$. When an attack happens, Y_n will suddenly become large and positive. Suppose, during an attack, the increase in the mean of $E(Y_n)$ can be lower bounded by h. Our change detection is based on the observation of $h \gg c$.

We use the recursive version of non-parametric CUSUM algorithm [23] which is shown as follows:

$$\begin{aligned} z_n &= (z_{n-1} + Y_n)^+, \\ z_0 &= 0, \end{aligned} \tag{1}$$

where $(z_{n-1}+Y_n)^+$ is equal to $(z_{n-1}+Y_n)$ if $(z_{n-1}+Y_n) > 0$ and 0 otherwise. z_n represents the continuous increment of Y_n. A large z_n is a strong indication of an attack.

Let $d_N(.)$ be the decision at time n: '0' for normal operation and '1' for attack. The decision function can be described as follows:

$$d_N(z_n) = \begin{cases} 0 & \text{if } z_n \leq N; \\ 1 & \text{if } z_n \geq N. \end{cases}$$

Here *N* represent the threshold for local attack detection. Let *conf* denote the confidence with which the individual detection node suspects an attack. We set $conf = \sum_i \delta(M_i) * d_N(M_i)$. δ assigns "weights" to a metric, depending on the extent to which the metric contributes to errors (false positive or negatives): $\delta(M_i) \propto \frac{1}{err(M_i)}$ where $err(M_i)$ is the sum of the false positive and negative rates for M_i. The appropriate δ can be configured from measurements.

When a local detection node detects an attack, it will sends the $(conf, dest)$ pair to its neighbor nodes in the overlay network infrastructure for correlation purpose.

3.3 Attack Information Sharing

A key requirement of the anomaly detection model is low false positive rates, calculated as the percentage of normalcy variations detected as anomalies, and high positive rate, calculated as the percentage of anomalies detected. In our approach, there are two factors which will affect the system performance: the overhead of the information sharing mechanism, and the delay for the decision making. If we disseminate information by multicast or broadcast, a large number of messages are propagated unnecessarily, as not all of the routers will be on the path to the attack destination. When a DDoS attack occurs, the network will be heavily loaded. If we share the local detection information by multicast or broadcast, it will further degrade the network. Communication bandwidth is often a scarce resource during the DDoS attack, so the attack information sharing should involve only small messages. In particular, any protocol collecting all local data at a single node will create communication bottlenecks, or a message implosion at that node. Recently, gossip-based protocols have been developed to reduce control message overhead while still providing high reliability and scalability of message delivery [9]. Gossip protocols are scalable because they don't require as much synchronization as traditional reliable multicast protocols. In gossip-based protocols, each node contacts one or a few nodes in each round (usually chosen at random), and exchanges information with these nodes. The dynamics of information spread bears a resemblance to the spread of an epidemic, and leads to high fault tolerance. Gossip-based protocols usually do not require error recovery mechanisms, and thus enjoy a large advantage in simplicity, while often incurring only moderate overhead compared to optimal deterministic protocols. The gossip protocol running at each node *n* has the structure as in Figure 3.

```
when ( node n builds a new (conf, dest) pair)
{
        while ( node n believes that not enough of its
        neighbors have received (const, dest) pair)
        {
        m =a neighbor node of p;
        send (conf, dest) pair to m;
        }
}
```

Figure 3: Gossip protocol for Our Approach

Compared with multicast or broadcast protocols, the gossip protocol has a smaller overhead. However, it requires a longer time for each node get the message. While reducing message dissemination overhead, we still want maintain the speedy information delivery provided by multicast or broadcast. A possible variant is directional gossip [16]. Directional gossip is primarily aimed at reducing the communication overhead of traditional gossip protocols. In our approach, we use a modified directional gossip strategy. We assume that the individual node knows its immediate neighbors in the network. Our gossiping protocol is described as the following: An individual node sends the $(conf, dest)$ pair to the node on its path to the destination target node with probability 1. It forwards the $(conf, dest)$ pair to all other nodes at random.

At anytime t, each node i maintains a list of $(conf_k, dest_k)$ pairs. The algorithm we use to share the DDoS attacks information is described as follows:

1. *Let $(conf_{r,k}, dest_{r,k})$ be all pairs sent to node i in round t-1.*
2. *Let $d_{i,k} = \Sigma_r conf_{r,k}$.*
3. *Compare $d_{i,k}$ with $Threshold_{i,k}$. If $d_{i,k} > Threshold_{i,k}$, then $dest_k$ is under attack.*
4. *Query the routing table, find out the next hop to $dest_{i,k}$, send the pair $(conf_{i,k}, dest_{i,k})$ to that node with probability 1. Send the pair to other neighbors with probability p.*

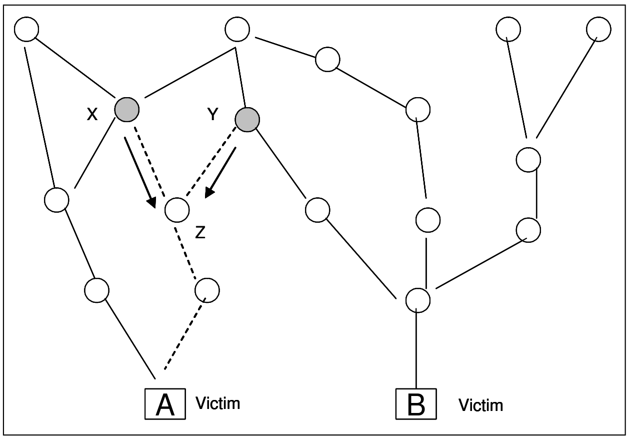

Figure 4: Gossip strategy in our approach

Top

The advantage of the strategy is illustrated in Figure 4. Suppose node X and node Y suspect that the destination host A is under attack, and both of them use node Z to forward packets to destination A. Obviously, it is better to send the $(conf, dest)$ pair with a higher priority to Z than to other neighbors. The rationale behind this scheme is as follows. Sending the detection information with higher probability to critical nodes allows them to make decision early, allowing the DDoS attacks to be mitigated earlier.

For each destination with $conf > 0$, each individual node in the overlay network sends the $(conf, dest)$ pair to its neighbors. On receiving such a message, the neighbors discard duplicates, compute the aggregate (*Aggr*) of the *conf* values received per destination, and forward non-duplicate values to their neighbors. If, for any destination, *Aggr* exceeds a pre-defined threshold, the individual node concludes that the destination is under attack. This cooperation stage helps reduce the errors in identification of attacks.

4 Simulation Results and Analysis

The objective of the simulation is to illustrate that our approach can improve detection accuracy while incoming reasonable overhead. We use Network Simulator version 2 (ns2 [1]) for our simulation. There are different approaches to build global network topology. We use Georgia Tech Internetwork topology models. Figure 5 shows the topology of the simulated network with 100 nodes.

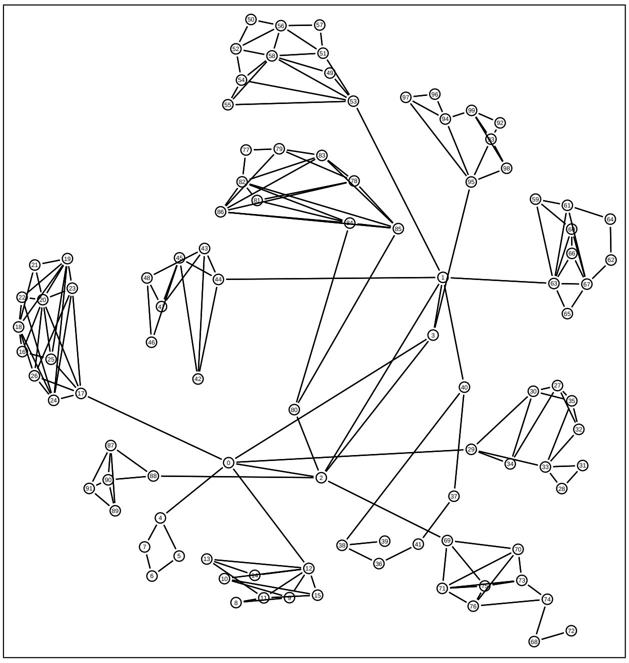

Figure 5: Simulated network topology

The attack is simulated using a given number of compromised nodes in different sub networks. Detection agents are deployed at selected nodes and execute the algorithm described. The communication agents use gossip to share information. We have modified the Multicast routing module of ns2 to support the gossip mechanism. We simplify the attack model by assuming that the attack traffic rate to be constant. In the case of UDP attack, we setup one CBR traffic generator on each attack node. We use the fixed size of 1000 bit for each packet. The

[1] http://www.isi.edu/nsnam/ns

Top

attack traffic rate at each source is set to 300Kbps which is a little bit higher than the legitimate traffic.

Let *p* represent the probability that each detection node in the detection overlay network sends the $(conf; dest)$ pair to its neighbor nodes. We vary the Gossip probability *p* among 0.2, 0.4, 0.6, 0.8, 1.0. The performance of the approach with different gossip probability *p* used are shown in Table 1

Table 1: Detection performance

Gossip Probability	Accuracy	Delay
0.2	94%	65.2s
0.4	97%	60.0s
0.6	98%	54.0s
0.8	99%	50.0s
1.0	99.5%	49.0s

As we can see from the simulation results, our algorithm can detect DDoS attacks with high accuracy. With $p = 0.4$ we can detect DDoS attack within 60.0 seconds with 97% accuracy. In the case of $p = 1.0$, the detection nodes share information by broadcasting to every neighbor. The detection accuracy and delay improves slightly, however, the cost is increased much higher. We measure the cost of information sharing in term of the overhead introduced to the network. Figure 6 shows the per-node overhead with different number of nodes in the system.

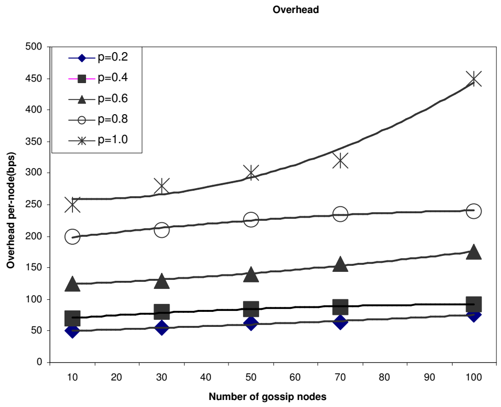

Figure 6: Information sharing overhead

We have evaluated the approach on a set of simulated distributed denial of service attacks. In these tests we evaluate the ability of our approach to detect the DDoS attacks. Each test run lasts for 30 minutes. We generate several TCP connections between legitimate clients and the victim and interleave them with attack traffic. The attack traffic that we simulated is the UDP flooding attack. Figure 7 gives the result of one test with gossip probability $p = 0.6$. The attack is started at 8^{th} minute and lasts until 30^{th} minute. The solid line represents the attack bandwidth measured at one individual detection node, and the dotted line represents the normal traffic bandwidth passed the detection node.

In this test, some of the individual nodes in the detection system can detect the attacks near 10^{th} minute. We can see from the simulation results, after some individual nodes detected the attack and respond to it, the target victim was still under the attack. This is because in our approach, every detection node make decision and counteract against the attack traffic independently. There is a delay for all the individual detection nodes to take effect. Our preliminary

Top

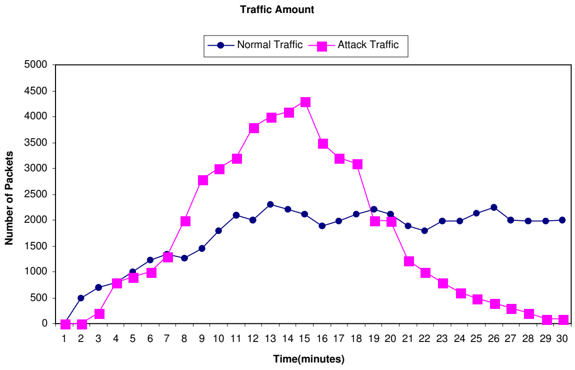

Figure 7: Traffic pattern at an individual detection node

experiments have shown that this approach does produce good results, although we have not had enough experience to determine the best way of making adjustments to the simulation parameters. we leave this for future work.

The approach of detecting DDoS attacks distributively based on traffic anomalies has its own limitations. On one hand, there are a set of theoretical issues related to the detection algorithms, such as the choices of local and global thresholds, traffic modeling, and admitting multilevel local detection results. One the other hand, since the large scale distributed cooperative mechanism induces a certain amount of delay to reach a global detection decision, this defense infrastructure is not very useful for DDoS attacks of very short durations. For example, the infrastructure should target to handle DDoS attacks longer than 5 min, which is around 75 percent of all the attacks measured in a recent study [19].

5 Related Work

Tao Peng et al. [21] observed that during DDoS attacks, the IP addresses of the attack packets seldom appeared before in normal situation. They proposed a mechanism called History-based IP filtering (HIF) for the edge router to admit the incoming packets according to a pre-built IP address database, which is built based on the previous connection history. The key point is to build an accurate IP address database (IAD), and use it to filter the incoming packets according to this IAD. In this approach, the legitimate requests whose IP addresses are not in IAD will be refused and the mechanism itself is brittle to DDoS attack.

Attacking DDoS at the source is proposed by J. Mirkovic et al [18]. The proposed system is located at the source network router (either LAN or border router) that autonomously detects and suppresses DDoS flows originating at this network. This system observes the outgoing and incoming traffic and gathers lightweight statistics on the flows, classified by destination. These statistics, along with built-in traffic models, define legitimate traffic patterns. Any discrepancy between observed traffic and a legitimate traffic pattern for a given destination is considered to be the signal of a potential DDoS attack. For example, TCP traffic is monitored and compared to an equational approximation of the TCP congestion control model. A TCP stream that is observed violating the behavior of the model is marked as an attack and is subsequently throttled back by the edge network's egress router. The amount of throttling is proportional to the flows deviation from it's expected behavior. Similar approach is applied to other transport protocols, the source router decides to throttle all traffic to the suspected target of the attack and at the same time attempts to separate attacking flows from legitimate flows and identify the attack-

Top

ing machines. This approach has the benefit of preventing malicious flows from entering the network and consuming resources.

MULTOPS [8] proposes a heuristic and data structure that network devices can use to detect DDoS attacks. It based on a assumption that during normal operations on the Internet, the packet rate of traffic going in one direction is proportional to the packet rate of traffic going in the opposite direction. So a significant disproportional difference between the packets in these two directions indicates an attack. Each network device maintains a multi-level tree, monitoring certain traffic characteristics and storing data in nodes corresponding to subnet prefixes. The tree expands and contracts within a fixed memory budget. The attack is detected by abnormal packet ratio values, and offending flows are rate limited. MULTOPS uses only the aggregate packet ratio to model normal flows. Non-TCP flows in a system using MULTOPS can either be misclassified as attack flows, or recognized as special and rate limited to a fixed value. In the first approach, harm is done to a legitimate flow, while in the second approach, sufficiently distributed attacks can successfully make use of the allowed transfer rate.

6 Conclusion and Future Work

Distributed denial of service is a major threat that cannot be addressed through isolated actions of sparsely deployed defense nodes. Instead, various defense systems must organize into a framework and inter-operate, exchanging information and service, and acting together, against the threat [17, 20]. In this paper we proposed a global detection infrastructure by building an overlay network on top of the internet. A gossip-based scheme is used to detect distributed denial of service attacks by information sharing. Compared to the existing solutions, our contribution is to provide a distributed proactive DDoS detection and defense mechanism. Our approach continuously monitors the network. When an attack begins, we have a high probability of detecting the attacks and stop the attacks before they do damage to the target victim. By correlating the detection information of each individual nodes, our scheme can greatly improve the accuracy of the detection.

Future work will fold in more topology information and vulnerability information gleamed from automated scanning and mapping tools. When the nodes know more topology information of the global Internet, it can utilize more intelligent gossip strategy to reduce the information sharing overhead while trying to detect attacks with speed. Armed with these more sophisticated methods, our approach can detect attacks more efficiently. We are investigating several important questions that still need to be addressed. These include the consensus algorithm and optimal gossip period. We also plan to validate this scheme by running them on real attack data sets.

References

[1] A. Akella, A. Bharambe, M. Reiter, and S. Seshan. Detecting DDoS attacks on ISP networks. In *ACM SIGMOD Workshop on Management and Processing of Data Streams*, pages 20–23, San Diego, CA, 2003.

[2] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of 18th ACM Symposium on Operating Systems Principles*, pages 131–145, Banff, Canada, October 2001.

[3] R. K. C. Chang. Defending against flooding-based, distributed denial-of-service attacks: A tutorial. *IEEE Communications Magazine*, 40(10):42–51, 2002.

Top

[4] D. Dittrich. Distributed denial of service (DDoS) attacks/tools, 2004. http://staff.washington.edu/dittrich/misc/ddos/.

[5] C. Estan and G. Varghese. New directions in traffic measurement and accounting. In *Proceesings of SIGCOMM 2002*, pages 270–313, Pittsburgh, PA, USA, 2002.

[6] S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Manajan, and V. Paxson. Pushback messages for controlling aggregates in the network, 2003. http://www.icir.org/pushback/.

[7] A. P. for Fox News. Powerful attack cripples internet, 2002.

[8] T. M. Gil and M. Poleto. Multops: a data-structure for bandwidth attack detection. In *Proceedings of 10th Usenix Security Symposium*, pages 23–28, Washington, D.C., USA, August 2001.

[9] I. Gupta, K. P. Birman, and R. van Renesse. Fighting fire with fire: using randomized gossip to combat stochastic scalability limits. *Special Issue Journal Quality and Reliability Engineering International:Secure, Reliable Computer and Network Systems*, 18(3):165–184, May 2002.

[10] S. Hariri, T. Dharmagadda, M. Ramkishore, G. Qu, and C. Raghavendra. Vulnerability analysis of faults/attacks in network centric systems. In *Proceedings of Parallel and Distributed Computing Systems*, pages 256–261.

[11] Q. Huang, H. Kobayashi, and B. Liu. Analysis of a new form of distributed denial of service attack. In *Proceedings of CISS03, the 37th Annual Conference on Information Science and Systems*, Johns Hopkins University, Baltimore, Maryland, March 2003.

[12] J. Ioannidis and S. M. Bellovin. Implementing pushback: Router-based defense against DDoS attacks. In *Proceedings of Network and Distributed System Security Symposium, NDSS '02*, pages 100–108, Reston, VA, USA, February 2002.

[13] A. Keromytis, V. Misra, and D. Rubenstein. Using overlays to improve network security. In *Proceedings of the ITCom Conference, special track on Scalability and Traffic Control in IP Networks*, pages 245–254, Boston, MA, August 2002.

[14] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[15] J. Li, P. Reiher, and G. Popek. Resilient self-organizing overlay networks for security update delivery. *IEEE Journal on Selected Areas in Communications, special issue on Service Overlay Networks*, 22(1), January 2004.

[16] M. Lin and K. Marzullo. Directional gossip: gossip in a wide area network. In *Proceedings of Dependable Computing - Third European Dependable Computing Conference*, pages 364–379, Berlin, Germany, 1999.

[17] J. Mirkovic, G. Prier, and P. Reihe. Alliance formation for DDoS defense. In *Proceedings of the New Security Paradigms Workshop, ACM SIGSAC*, pages 11–18, Ascona, Switzerland, August 2003.

[18] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *Proceedings of ICNP 2002*, pages 312–321, Paris, France, November 2002.

Top

[19] D. Moore, G. Voelker, and S. Savage. Inferring internet denial of service activity. In *Proceedings of the USENIX Security Symposium*, pages 9–22, Washington, DC, USA, August 2001.

[20] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan. Cossack: Coordinated suppression of simultaneous attacks. In *DARPA Information Survivability Conference and Exposition*, volume 1, pages 2–13, Washington, DC, April 2003.

[21] T. Peng, C. Leckie, and R. Kotagiri. Protection from distributed denial of service attack using history-based IP filtering. In *Proceedings of IEEE International Conference on Communications*, volume 1, pages 482–486, Anchorage, Alaska, USA, May 2003.

[22] D. X. Song and A. Perrig. Advanced and authenticated marking schemes for IP traceback. In *Proceedings of IEEE Infocomm*, volume 2, pages 878–886, Anchorage, Alaska, USA, 2001.

[23] H. Wang, D. Zhang, and K. G. Shin. Detecting syn flooding attacks. In *Proceesings of IEEE Infocom*, volume 3, pages 1530–1539, New York City, NY, June 2002.

[24] S. Zhang and P. Dasgupta. Denying denial of service attacks: A router based solution. In *Proceedings of International Conference on Internet Computing*, pages 301–307, Las Vegas, Nevada, USA, June 2003.

Top