An Integrated Framework for Proactive Mitigation, Characterization and Traceback of DDoS Attacks

Bhavana Gandhi and R. C. Joshi,

Indian Institute of Technology Roorkee, Roorkee, Uttaranchal, India

Summary
Denial of Service (DoS) attacks pose a severe security threat to the steady functioning of any network. These attacks aim at depleting the resources of a server or an administrative network by overwhelming it with enormous and useless traffic. The outcome of this is the fact that legitimate users are denied service. Though an array of schemes has been proposed for the detection of the presence of these attacks, characterizing of the flows as a normal flow or a malicious one, identifying the sources of the attacks and mitigating the effects of the attacks once they have been detected, there is still a dearth of complete frameworks that encompass multiple stages of the process of defense against DoS attacks. In this paper, we propose a novel framework which deals with proactively mitigating the influence of the attack , characterization of the TCP flows as attack or legitimate, and identification of the path traversed by the flow once it has been characterized as an attack flow. Generation of copies of TCP/IP headers by predefined intermediate routers provides for the dual functionality of proactive mitigation and traceback. The characterization of the flows has been achieved by an innovative Exactly Periodic Subspace Decomposition (EPSD) based approach. We validate the effectiveness of the approach with simulation in ns-2, integrated with Matlab, on a Linux platform.

Key words:
Distributed Denial of Service (DDoS), EPSD, characterization, mitigation, traceback.

1. Introduction

The Internet (originally known as ARPANET) was created in 1969 to provide an open network for researchers [1]. Unfortunately, with the growth of the Internet, the attacks to the Internet have also increased incredibly fast. The widespread need and ability to connect machines across the Internet has caused the network to be more vulnerable to intrusions and has facilitated break-ins of a variety of types. According to [1], a mere 171 vulnerabilities were reported in 1995 which boomed to 8064 in the year of 2006. Apart from these, a large number of vulnerabilities go unreported each year. A **Denial of Service (DoS)** attack is commonly characterized as an event in which a legitimate user or organization is deprived of certain services, like e-mail or network connectivity, that they would normally expect to have. DoS attacks [2, 3] inject maliciously-designed packets into the network to deplete some or all of these resources.

The attack power of a Distributed DoS (DDoS) attack [4] is based on the massive number of attack sources instead of the vulnerabilities of one particular protocol. DDoS attacks, which aim at overwhelming a target server with an immense volume of useless traffic from distributed and coordinated attack sources, are a major threat to the stability of the Internet. The number and assortment of both the attacks as well as the defense mechanisms is monstrous. Though an array of schemes has been proposed for the detection of the presence of these attacks, characterizing of the flows as a normal flow or a malicious one, identifying the sources of the attacks and mitigating the effects of the attacks once they have been detected, there is still a dearth of complete frameworks that encompass multiple stages of the process of defense against DoS attacks.

In this paper, we shall propose an integrated framework for defense against flooding- based DDoS attacks. By "integrated", we mean, the framework will provide for the following activities in defense against DDoS attacks:

1. Effective *characterization* of the flows as attack or legitimate flows,
2. Proactive *mitigation* of the effect of the attack on the victim node or network,
3. Accurate *traceback* of the source(s) of the attack flows.

The rest of the paper is organized as follows. Section 2 gives a brief overview of some of the existing techniques to facilitate characterization, mitigation and traceback of DDoS attacks along with some of their limitations. Section 3 gives a gist of our proposed integrated framework. The efficiency of our mitigation scheme is charted in Section 4, whereas Sections 5 and 6 explain in detail our EPSD based characterization technique. Section 7 deals with the traceback facility of our framework. Section 8 gives the Experimental Design which is inclusive of the simulation testbed and the obtained results. We conclude our work and provide pointers to possible future work in Section 9.

Manuscript received March 24, 2007
Manuscript revised March 28, 2007

2. Related Work

This section charts out the different work done in the areas of characterization, mitigation of the effect of DDoS attacks and tracing back the sources of the attack.

Characterization: Several schemes have been suggested to characterize attack flows. It was proposed in [5] a simple statistics-based mechanism to detect TCP SYN flood attacks. The idea is to detect deviation from an expected balanced SYN/FIN packet ratio using a non-parametric, cumulative sum method. However, such a simple technique is not foolproof as the attackers can mix their SYN and FIN packets.

It was proposed in [6] a spectral analysis method to distinguish attack flows from the normal ones by determining the periodicity in the packet process as defined in this paper. But the method does so by using the Welch's modified periodogram, which has several disadvantages as compared to the EPSD technique used in this paper. The pros of the latter and the cons of the former have been highlighted in this paper.

Mitigation: In [7], Bohacek has suggested a mitigating approach that relies on routers filtering enough packets so that the server is not overwhelmed while ensuring that as little filtering as possible is performed. He has proposed a solution wherein packets should be filtered at routers through which the attack packets are passing. But, it is a reactive mitigation technique that also has the drawback that legitimate traffic packets may also be dropped en route to the destination.

In [8], Kalantari et al. have proposed a proactive method for mitigation of the effects of DDoS attacks wherein each router maintains a partition of active TCP flows into aggregates. Each aggregate is probed to estimate the proportion of attack traffic that it contains. Packets belonging to aggregates that contain significant amounts of attack traffic may be subject to aggressive drop policies to prevent attack at the intended victim. Again, in this case too, legitimate packets face the risk of being dropped. Also, proper definition of aggregates is a critical part of the approach. Moreover, aggregates have to be defined in advance of the attack so that their response measurements are taken to normal (non-attack) traffic in order to be compared later on with measurements under an attack, if any.

It is observed that most of the mitigation techniques in practice today, suffer from the following drawbacks:

1. They are reactive in nature.
2. They deploy packet dropping policies at the routers wherein even legitimate packets face the risk of being dropped.
3. In cases like [8], the topology of the network needs to be known in advance.

The mitigation technique used in the framework proposed in this paper does away with all these drawbacks as we shall see in Section 4.

Traceback: IP Marking [9] is traditionally used for IP Traceback. The basic idea of the IP marking approach is that routers probabilistically write some encoding of partial path information into the packets during forwarding, so that based on this information the destination server can reconstruct the path that was taken by the packets.

In [10], Song and Perrig have suggested Advanced and Authenticated Marking Schemes that encode the edge information in 16 bits of the packet to be marked. For this purpose, the 16-bit IP Identification field used for fragmentation in the IP header is overloaded, i.e. this field carries the encoding information instead of the regular packet fragmentation information.

The obvious drawback in the methods discussed for IP Traceback is that they do not work for packets that are fragmented as the IP Identification field is overloaded for edge information.

3. Proposed Solution

In this section, we shall discuss in great depth the various facets of our proposed framework for defense against DDoS attacks.

The proposed framework provides for proactive mitigation against the effect of DDoS attacks as described next. Whenever a packet arrives at a router to be forwarded to the server to be protected from a DDoS attack, instead of sending that packet on the outbound link, a copy of its header [11] is sent toward the server for characterization. This provides a *proactive* approach to mitigation against the attack as the bandwidth of the links involved will not be exhausted by the voluminous attack traffic as only the headers (that are small in size) will traverse on the links to the server.

The technique to be used in this framework for mitigation provides the *dual* functionality of IP Traceback as well. The 16-bit IP Identification field in the header of the original packet which was being used traditionally for traceback need not be used now. In the proposed technique, the IP Identification field of the original packet will not be used for traceback purposes. Instead, the IP Identification field in the *copy of the header* generated will be used to store the edge information.

The copies of headers generated represent the actual dynamics of the traffic flow to which they belong. These headers will be subject to the characterization test described next. For characterization, instead of the Welch's periodogram method used in [6], the Exactly Periodic Subspace Decomposition (EPSD) [12] technique (as discussed in detail in Section 5) will be used as part of

this framework. The EPSD technique does away with the disadvantages of the Welch's method by difference in the selection of time domain input elements that constitute the frequency domain output elements.

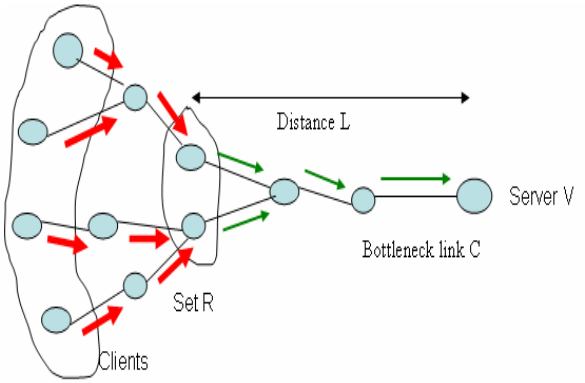

Fig. 1: Topology to illustrate proposed solution.

To get a better understanding of the proposed model, consider a sample topology shown in Figure 1. The topology considered is similar to the one used traditionally to depict a typical client-server scenario in the Internet for simulation purposes [6]. The clients (attack and legitimate) send their requests to the server V (indicated by thick arrows). The routers (set R) en route from the clients to the server will proactively generate copies of these packets and save the original packets with them. These routers will also stamp their identity in the Identification field of the copy of the IP header thus generated and send them to V (indicated by thin arrows). The other routers through which these header copies will traverse before reaching V will also append their edge information in the same Identification field. Once these header copies reach the bottleneck link C, they will undergo the EPSD test for periodicity and thus the flows will be characterized as attack or legitimate. If a flow is characterized as a legitimate flow, only then will the routers belonging to set R be instructed to forward the stored packets to the server. If a flow is characterized as an attack flow, then the encoding information in the generated copies of the headers will be used to construct the attack graph for IP Traceback [9] and the routers (set R) will be asked to drop the corresponding original attack packets. A flowchart depicting the solution is illustrated in Figure 2.

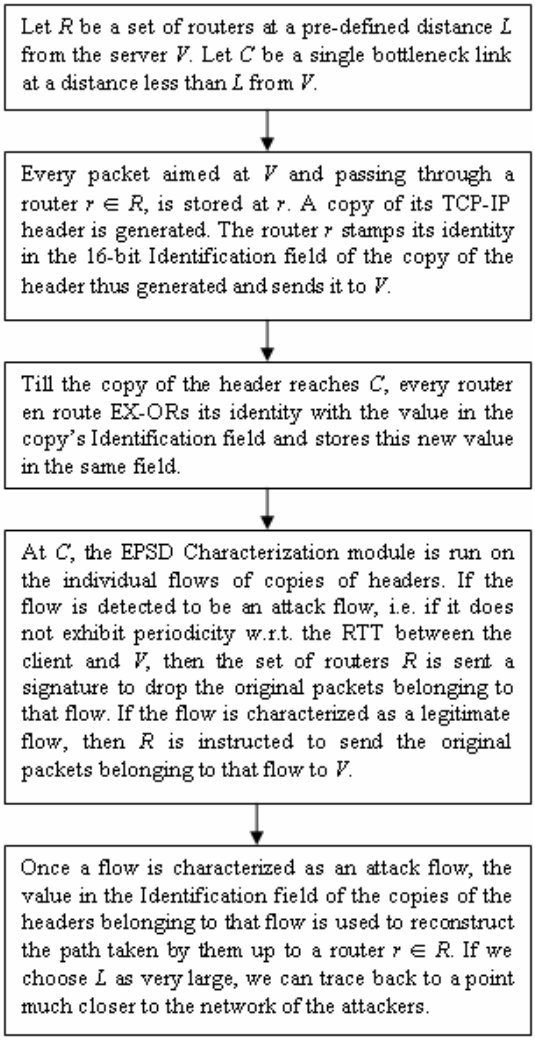

Fig. 2: Flowchart depicting details.

4. Efficiency of the Mitigation Technique

The main aim of the attackers implementing flooding-based DDoS attack on a server is to overwhelm the bottleneck links of the victim by bombarding it with useless and maliciously designed packets. As discussed above, the mitigation technique that forms part of our framework is a proactive one, i.e. it comes into action even before the attack is actually detected. Instead of sending the packets, attack or legitimate, to the server, only the copies of their headers are forwarded to the server. This reduces the load on the bottleneck links to a large extent. The savings on the bandwidth of the link can be calculated as:

Ratio of Bandwidth of the bottleneck links in use =

$$\frac{\text{Size of TCP header} + \text{Size of IP header}}{\text{Size of TCP header} + \text{Size of IP header} + \text{MTU of Ethernet}} \quad (1)$$

$$= \frac{20+20}{20+20+1500}$$ each value in bytes (considering IPv4) [13, 14]

$$= \frac{40}{1540}$$

$$= 0.026$$

= 2.6% in use due to the proactive mitigation technique, i.e. 97.4% bandwidth saved (only when an attack flow is considered).

As is clear, the saving on the bandwidth of the bottleneck links is gargantuan.

Our approach toward mitigating the effects of a DDoS attack has the following advantages over most of the existing approaches (discussed in section 2) for the same:

1. It is a proactive technique, i.e. it does not wait for an attack to be detected to get activated.
2. It does not need knowledge of the network topology in advance.
3. The packet dropping policy only aims at dropping attack packets.

5. Exactly Periodic Subspace Decomposition for Characterization

5.1 EPSD in brief

Muresan et al. [12] proposed the EPSD technique to identify different frequency components in noise prone data. In this paper, we use EPSD based technique to demonstrate the two methodologies of online and offline detection of DoS attacks.

Definition: A signal S is of exactly period P if S is in $R(\psi^P)$, and the projection of S onto $R(\psi^{P'})$ is zero for all P'< P (where $R(\psi^{P'})$ is the subspace of signal of period P') [12].

With the above definition, a signal of exactly period P is not exactly period of 2P, 3P, etc. In addition, not every periodic signal is exactly periodic, but every exactly periodic signal is periodic. For example, an exactly periodic 4 signal is

R = [1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1]

The EPSD technique finds the subspace corresponding to the signal of exactly periodic P and shows that these subspaces are orthogonal to each other.

5.2 Advantage of EPSD over Welch's modified periodogram used in [6]

The Welch method, based on Barlett's procedure, splits a set of data into smaller sets of data and calculates the modified periodogram (the power spectrum) of each set. The modified periodogram is calculated by applying a window function to the time-domain data, computing the Discrete Fourier Transform (DFT), using Fast Fourier Transform (FFT), and then computing the magnitude square of the result. Then the frequency domain coefficients arising from calculating the modified periodograms are averaged over the frequency components of each data set to reduce the variance. FFT has many restrictions itself [15]. If we use FFT, there are several conditions that must be satisfied. These conditions are:

1. Since spectrum only records the components at scales, all components of signal must be integral-times of frequency resolution.
2. The number of sample data N must be equal to 2r.

When FFT is unsatisfied due to these two conditions, it can cause serious errors. The picket-fence effect and the leakage effect result from condition (1) being unsatisfied. The former makes components only display at frequency scales, so real frequency cannot be shown; the latter makes the energy of a harmonic diffuse to nearby scales and not concentrate on a specific scale.

Thus, the two effects result in frequency and amplitude errors individually. When condition (2) is unsatisfied, spectrum analysis cannot be calculated by FFT, which will increase computation quantity significantly.

The disadvantages [16] of the Welch's modified periodogram include frequency resolution limitations, possibility of side lobes masking signal components due to the necessity of windowing (which implicitly assumes that all samples outside the window are zero), and the inability to accurately represent very short data segments.

Just like the Fourier decomposition, the EPSD decomposes the signal into orthogonal components [12]. But unlike the Fourier transform, the EPSD is obtained by taking projections onto exactly periodic orthogonal multidimensional subspaces of periods that divide, whereas the Discrete Fourier Transform is obtained by taking orthogonal projections onto one-dimensional (1-D) complex exponentials $e^{j((2\pi)/N)k}$ with frequencies (k/N), k=0,1,…,N-1.

The EPS is spanned by a collection of Fourier exponentials, which is dictated by the period. When searching for periodic components, this is the main advantage of the EPSD. By having subspaces of dimensions larger than one, the EPS can better capture, in

one coefficient, the periodic energy than can the Fourier transform.
Basically DFT is computed as follows:
Let x_0, ..., x_{N-1} be complex numbers. The DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk} \qquad k = 0, \ldots, N-1. \tag{2}$$

FFT is an algorithm to compute the same result more efficiently. The problem with this is that each number x_n of the input signal contributes towards each element X_k of the DFT or FFT.
The EPSD technique resolves this issue. The way this is done is elucidated in simplifying terms as follows: The given signal is evaluated for one frequency at a time starting with the lowest ones. When the signal is evaluated for energy at frequency 2, the signal components that contribute towards this energy are eliminated from the original signal so that these components do not again contribute to energy while evaluating for frequency 4, 6, 8, and so on, which they would, unless eliminated.

6. Defining the Packet Process

TCP is a sliding-window and acknowledgement (ACK) based transport protocol. The window size of a TCP flow limits the number of in-flight packets it can have in the network. The window size is determined by the advertised window size of the receiver and the estimated congestion level of the network. TCP is a window based, and Acknowledgement (ACK) - based transport protocol widely used in the Internet. Every data packet arriving at the receiver can permit the receiver to transmit an ACK packet to the sender [6]. Similarly, every ACK packet arriving at the sender allows it to place a new data packet on the network. Thus, if we monitor the network at any point between the sender and the receiver and if we observe a certain number of packets belonging to a particular flow, then it is quite probable that the same number of packets belonging to that flow will be visible after one round – trip time between the sender and the receiver. This introduces periodicity in a normal TCP flow. In an attack flow, the attackers overwhelm the server by not obeying the TCP policy of waiting for ACK packets before the outstanding data packets can be sent. Thus, there is the loss of periodicity in such flows.
We consider a random process [6] $\{X(t), t = n\delta, n \in N\}$, where δ is a constant time interval, N is the set of positive integers, and for each t, X(t) is a random variable. Here X(t) represents the number of packet arrivals for a TCP flow in $(t-\delta t]$. We refer to this random process as the *packet process* in the rest of this paper. To study the periodicity embedded in the packet process, we use the EPSD technique because of its advantages over other conventional techniques, as elucidated in the previous section.

7. The Traceback Scheme

The IP header encoding as described in [9] has several practical limitations. It negatively impacts users that require fragmented IP datagrams. Though recent studies [17] show that a very small percentage of packets are fragmented, our proposal works for fragmented data grams as well. This is possible as our technique does not use the 16-bit Identification field of the original packet for traceback purposes, but instead utilizes this field of the copy of the headers generated. As suggested by [9], for IPv6, where the Identification field is not available, another field in the IP header like the 24-bit *flow label* field can be used for the same purpose.
The distance L plays an important role in determining the efficiency of the Traceback scheme. The larger the value of L is, the closer we can get to the attackers. In [9], the routers right next to the clients (legitimate or attackers) start marking packets passing through them. But, it may be noted that routers that are so close to the attackers can easily be subverted by them as they may belong to the same network as the attacker.

8. Experimental Design

8.1 Simulation Topology

The simulation is carried out in Network Simulator ns-2 [18] integrated with Matlab [19] in which the functionality for EPSD [12] is coded. The topology used for the simulation purposes is similar to the one shown in Figure 1. The legitimate clients are TCP agents that request files of size 1 Mbps each with request inter-arrival times drawn from a Poisson distribution. The attackers are modeled by UDP agents. The rate is kept very high (8Mbps) which is very typical of an attack flow. A UDP connection is used instead of a TCP one because in a practical attack flow, the attackers would normally never follow the basic rules of TCP, i.e. waiting for ACK packets before the next window of outstanding packets can be sent, etc. As discussed in depth, the generation of copies of TCP/IP headers of the packets sent to the server is done at the routers belonging to set R. Suitable marking of the Identification field in these copies is done by routers en route to the server.
The specifications for the Characterization module at the bottleneck link C follow next. Let T_{sample} be the time interval after which the flow statistics (packet arrivals) are

monitored continuously per flow. Let $N_{current}$ be the number of packets arrived till the sample instant from the time the flow was active minus the number of packets arrived till the last sample instant. Let small_stats be an array of length N_{stats} which stores the value of $N_{current}$ for the last N_{stats} instants. Once the flow is past its slow start phase, for every T_{EPSD} seconds, the EPSD functionality is invoked online for the small_stats array, i.e. for the latest N_{stats} samples. This is done as even a legitimate flow lacks periodicity in its slow start phase. We should delay the decision of tagging the flow as attack or legitimate till EPSD has been called for cnt_thresh times. For each time that the periodicity is found to be missing from the array small_stats, another counter bad_flow is incremented. If bad_flow is greater than a pre-defined threshold bad_thresh, then the flow is tagged as an attack flow and further packets from the flow are discarded. The detailed steps are shown in the form of a flowchart in Figure 3 (next page).

The value of N_{stats} should be chosen such that it should neither be too large to cause a great overhead in terms of storage and processing requirements, and at the same time it should be large enough to indicate at least two complete cycles with respect to the RTT from the source node to the server in order to safely judge the periodicity, i.e.

$$N_{stats} > 2*RTT/ \text{Sample Period} \quad (3)$$

8.2 Results

The proactive mitigation approach used in our framework provides for up to 50 % reduction in the utilization of the bandwidth of the bottleneck link C (as shown in Figure 1) when tested for a varying number of legitimate flows and attackers.

For characterization, the domain of analysis includes the observation of the Exactly Periodic Subspace (EPS) energy vs. the period at which it occurs. The period, other than 1 (as period of 1 denotes the dc component of the energy), at which the significant positive EPS energy is observed denotes the exact period of the packet process.

Legitimate Flow

For T_{sample} = 10 ms. a sample result is as shown next. The trace files are analyzed to determine the round trip time between any one source and the server, which comes out to be approx. 90 ms. Figure 4 illustrates the packet process for a legitimate flow. The packets observed at the sample instants are plotted against the corresponding sample number. Here, only 30 samples are shown.

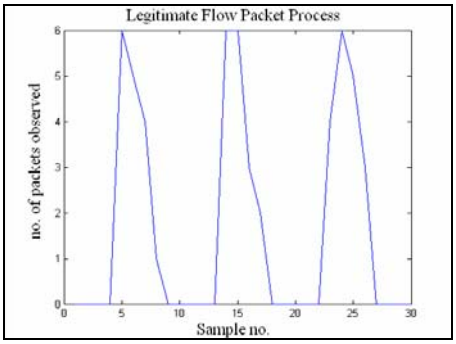

Fig. 4: A few sample observations from the legitimate flow packet process.

To determine the exact periodicity, the EPSD technique is applied to these samples. The resulting EPSD graph is shown in Figure 5.

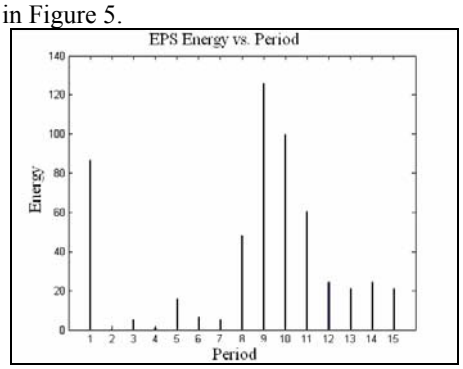

Fig. 5: EPS Energy vs. Period for legitimate TCP connection.

It is visible that the period of the packet process is 9, i.e. the signal due to the packet process is periodic with a period of 90 ms (as the samples are taken at 10ms intervals), which is the RTT between the source and V.

Attack Flow

The packet process of an attack flow is illustrated in Figure 6.

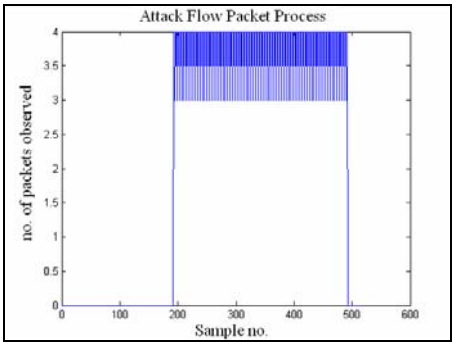

Fig. 6: Packet process for an attack flow.

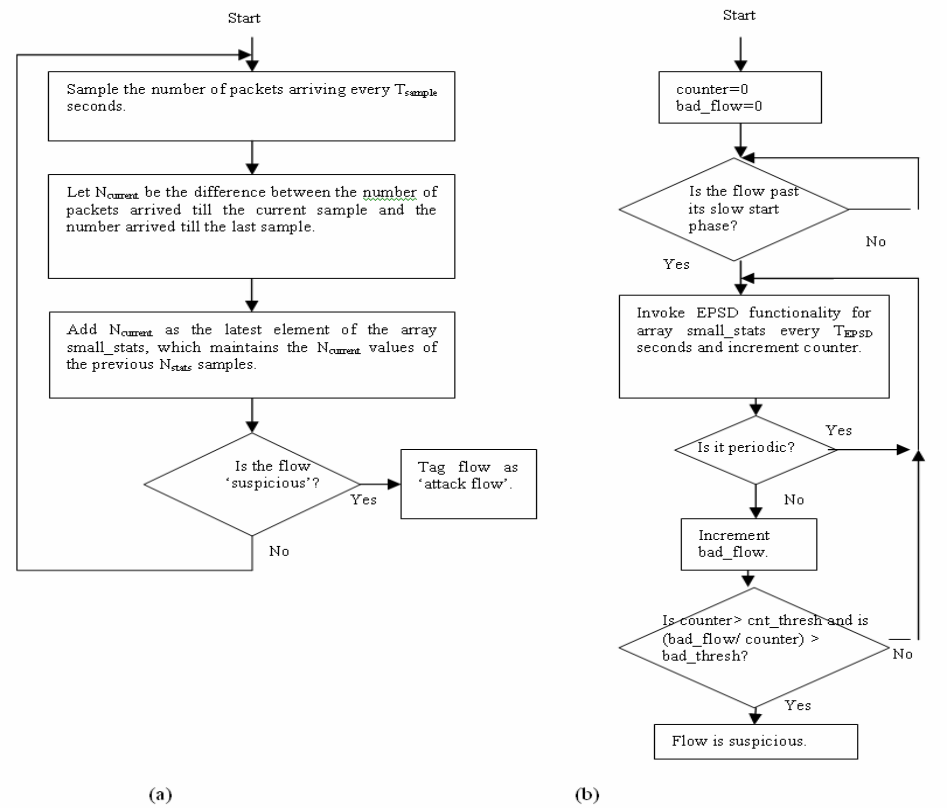

Fig. 3: (a) Flowchart for sampling the number of packets per flow. (b) Flowchart for invoking the EPSD functionality.

The packet process highlights the fact that it was a bursty DoS attack. The process does not appear to be periodic with respect to the RTT.

When the EPSD technique is applied on the time window of any 30 samples, the generated result is shown in Figure 7.

To qualify this flow as a normal flow, there should have been significant energy at period 9, but there appears to be none according to the generated EPSD graph, thus characterizing the flow as an attack flow.

Once a flow of the header copies is characterized as a legitimate one, the routers R are sent a message to send original packets belonging to the corresponding flow to the server V. If the header copies of a certain flow do not satisfy the RTT property of our characterization module, then routers R are sent a signature to drop packets belonging to that flow. At the same time the Identification fields in those header copies are used by the Traceback module to find out which of the routers R generated them and thus determine the location from where the original attack packets were generated. Details of the formation of the *attack graph* and Traceback can be found in [9].

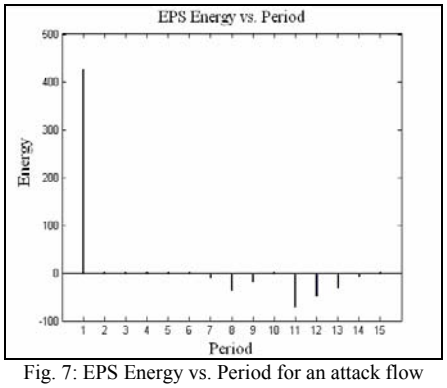

Fig. 7: EPS Energy vs. Period for an attack flow

A comparative study of our proposed integrated framework with the existing defense techniques is shown in Table 1.

9. Conclusions and Future Work

The framework proposed in this paper provides an end-to-end solution for defense against flooding-based DDoS attacks. Generation of copies of headers at pre-defined

intermediate routers provides for proactive mitigation against DDoS attacks as well as facilitates for IP Traceback without having to compromise the technique for fragmented packets. For characterization, we are using the EPSD technique to distinguish legitimate TCP flows from the attack ones by analyzing the flow of these same copies of headers. We illustrate the effectiveness of this approach by appropriate simulation testbed.

Another advantage of our framework is that instead of deploying the intelligence of generating copies of headers at all routers, we are doing so only at routers that are at a certain distance L from the server. By doing this, we reduce the risk that is possible by compromising the routers where this functionality is deployed. With less number of such routers, it is easy to monitor them and protect them from vulnerabilities.

The various parameters defined for characterization, like N_{stats}, T_{sample}, T_{EPSD}, cnt_thresh, and bad_thresh can be calculated by heuristic methods by keeping in mind the specific network under consideration.

Stages	Features	
	Existing Technique	**Proposed Integrated Framework**
Mitigation	Optimal Filtering - Bohacek [7]	
	reactive	proactive
	packet dropping policy drops legitimate packets too.	packet dropping policy does not drop legitimate packets.
	Partitioning TCP flows into aggregates - Kalantari et. al [8]	
	topology to be known in advance	topology need not be known in advance
	packet dropping policy drops legitimate packets too.	packet dropping policy does not drop legitimate packets.
Characterization	Spectral Analysis - Cheng et. al [6]	
	Welch's periodogram method used.	EPSD technique used (advantages over Welch's periodogram discussed in detail in section 5).
Traceback	Advanced and Authenticated Marking Schemes - Song and Perrig [10]	
	does not work for fragmented datagrams.	works for fragmented datagrams as well.

Table 1: A comparison with the existing techniques.

The model for calculating these parameters are left for future work.

There is a network and router overhead with the generation and transmission of copies of headers, with some routers having to temporarily store the original packets. But with the dual functionality of *mitigation* and *traceback* provided by it, it proves to be an attractive technique. The packet marking procedure required by the technique is well within the capability of conventional routers [20].

References

[1] CERT Statistics, URL http://www.cert.org/stats/certstats.html

[2] CERT. Denial of Service Attacks. http://www.cert.org/techtips/denial of service.html, 1997.

[3] K.J. Houle, G. M. Weaver, N. Long, and R. Thomas. "Trends in denial of service attack technology". *Technical Report Version 1.0*, CERT Coordination Center, Carnegie Mellon University (2001). http://www.cert.org/archive/pdf/DoS trends.pdf.

[4] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovi. "Distributed denial of service Attacks," *In Proceedings of IEEE International Conference on Systems and Cybernetics*, vol. 3, pp. 2275-2280, 2000.

[5] H. Wang, D. Zhang, and K. G. Shin. "Detecting SYN flooding attacks," *In Proceedings of IEEE INFOCOM 2002*, pp. 1530 – 1539, June 2002.

[6] Chen-Mou Cheng, H. T. Kung, and Koan-Sin Tan, "Use of Spectral Analysis in Defense Against DoS Attacks," *In the Proceedings of Global Telecommunications Conference, 2002, GLOBECOM '02. IEEE,* Vol. 3, pp: 2143 – 2148, Nov. 2002.

[7] B. Stephan, "Optimal filtering for denial of service mitigation," *Proceedings of the 41st IEEE Conference on Decision and Control, 2002,* Vol. 2, pp: 1428 – 1433, Dec. 2002.

[8] M. Kalantari, K. Gallicchio and M. A. Shayman, "Using transient behavior of TCP in mitigation of distributed denial of service attacks," *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, Vol. 2, pp: 1422 – 1427, Dec. 2002.

[9] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson, "Practical network support for ip traceback," in *Proceedings of the 2000 ACM SIGCOMM Conference*, Aug. 2000.

[10] Dawn Xiaodong Song and Perrig A., "Advanced and authenticated marking schemes for IP traceback," *Proceedings. IEEE INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2, pp: 878 – 886, Apr. 2001.

[11] S. Bellovin. ICMP traceback messages, March 2000. Internet Draft: http://www.cs.columbia.edu/~smb/papers/draft-bellovin-itrace-00.txt

[12] D. Darian Muresan, and Thomas W. Parks, "Orthogonal, Exactly Periodic Subspace Decomposition," *IEEE Trans. on Signal Processing*, Vol. 51, No. 9, pp: 2270-2279, Sep. 2003.

[13] Andrew S. Tanenbaum, Computer Networks - Fourth Edition, pearson Education, 2003.

[14] Wikipedia-The free Encyclopedia, http://en.wikipedia.org/wiki/Maximum_transmission_unit

[15] Rong-Ching Wu, and Ta-Peng Tsao, "Theorem and Application of Adjustable Spectrum," *IEEE Trans. on Power Delivery*, Vol. 18, No. 2, pp: 372-376, April 2003.

[16] P.L. Feibig, D.M. Etter, and S.D. Stearns, "A Software Tool for comparing Spectral Estimation Techniques," *Twenty-Third Asilomar Conference on Signals, Systems and Computers, 1989*. Vol. 1, pp: 371–375, 1989.

[17] Ion Stoica and Hui Zhang, "Providing guaranteed services without per flow management," in *SIGCOMM'99*, pp. 81–94, 1999.

[18] NS-2 Network Simulator, 2006. http://www.isi.edu/nsnam/ns/

[19] The Mathworks- MATLAB- www.mathworks.com/products/matlab.

[20] R. Chen and J-M Park, "Attack diagnosis: throttling distributed denial-of-service attacks close to the attack sources," *Proceedings of the 14th International Conference on Computer Communications and Networks, 2005*, pp: 275 – 280, Oct. 2005.

Bhavana Gandhi received the B.E. degree in Information Technology from K. J. Somaiya College of Engineering, University of Mumbai in 2004. During 2004-2005, she worked as a Software Engineer with Infosys Technologies Ltd. She is currently in her final year of M.Tech. in Information Technology being pursued at Indian Institute of Technology Roorkee. Her areas of interest include Network Security, Operating Systems and Design and Analysis of Algorithms.

R. C. Joshi received the B.E. degree in Electrical Engineering from Allahabad University in 1967. He received his M.E. and Ph.D. in Electronics and Computer Engineering from University of Roorkee in 1970 and 1980, respectively. He is currently a Professor at Indian Institute of Technology Roorkee. He has guided several Ph.D. theses, M.E./M.Tech. dissertations and completed various projects. His areas of interest include Network Security, Parallel and Distributed Processing, AI and Databases.