# Denial of Service Attack Detection using Extended Analog Computers

Craig Shue, Brian Kopecky, Chris Weilemann
*Computer Science Department, Indiana University*
*Bloomington, IN, U.S.A.*
{cshue, bkopecky, cweilema}@cs.indiana.edu

***Abstract*— Denial of Service (DoS) attacks, a damaging assault on computer networking infrastructure, have been extensively examined by the digital computing community. However, no work has been done to examine the ability of Extended Analog Computers (EAC) to detect DoS attacks. In this paper, we discuss how EACs could be used in DoS detection.**

## I. Introduction

Denial of service attacks, a problem on the Internet, facilitate cyber-terrorism and unlawful extortion attacks. In early 2000, several high-profile attacks cost providers millions of dollars in damages [1]. Unfortunately, automated tools make these attacks increasingly easier to execute [2].

Given the spread and proliferation of modern Internet worms and viruses, higher numbers of compromised machines have resulted. These machines can be commandeered and used to launch damaging attacks on e-commerce sites. Often, such attackers will extort money in order to cease their attacks.

Denial of service attacks attempt to overwhelm victim machines, preventing them from being able to provide their resources to legitimate users. Some denial of service attacks target the bandwidth capabilities of targeted machines while others target the machine's computational state.

Firewalls are often used by networks and end-hosts to prevent malicious traffic from damaging a victim. Network firewalls can prevent traffic from ever reaching the end-host, saving the intended victim from both the computation and network fallout of the attack.

With firewalls, destination networks must still mitigate the network overheads generated from a distributed DoS attack. Research has been conducted to evaluate the effectiveness of "push-back" mechanisms [3], where a destination under attack can seek support from network routers to filter attack packets before they reach the destination network. While a viable approach, push-back mechanisms require that a system recognize it is under attack and determine the attacker before it can be used.

While substantial work has been done on determining denial of service attacks in digital systems [4], [5], [6], [7], no work has been conducted to evaluate the effectiveness of extended analog computers (EAC) in denial of service detection.

In this paper, we provide a starting point for using extended analog computers to detect denial of service activity.

In our first approach, we propose using a hash of incoming IP addresses to bin addresses into current sources on the EAC. We then periodically sample the EAC board to determine if a particular source is abnormally concentrated, indicating a high traffic volume from a particular set of hosts.

In our second approach, we propose arranging the sources in a circle with an outer ring of sinks drawing off normalized current. An Lukasiewicz logic array (LLA) sink is situated in the center of the source ring and performs the exclusive or operation. If the current for any of the source points suddenly increases, the LLA's exclusive OR function will be triggered, alerting the system that there could possibly be an on-going denial of service attack.

While we do not examine distributed denial of service attacks in our model, we should note that they are much more difficult to distinguish from "flash crowds" of legitimate users accessing content. This difficulty complicates analysis and there currently does not appear to be a straight-forward analog approach to detecting these problems.

The remainder of our paper is arranged as follows. In section II, we describe the Extended Analog Computer. In section III, we describe our methodology for our experiment. We follow with our results in section IV and conclude with discussion in section V.

## II. The Extended Analog Computer (EAC)

The Extended Analog Computer (EAC) is a device which computes using semi-conductive material, voltage gradients, and logic arrays. The computational medium can be any semi-conductive material, such as black conductive foam, Jello brand gelatin, or silicon. Voltage gradients are induced on the conductive medium through the use of a series of sources and sinks. Sources transmit electrical current to the medium by producing an excess of electrons. These electrons, once on the medium, tend to travel towards a sink. A sink provides a path of least resistance for the electrons to exit the board. This is either a ground or a negative voltage source.

As electrons flow across the medium from sources to sinks, a voltage gradient is produced. This gradient can then be sampled across the medium to produce a graph of the electrical flows occurring on the conductive medium. Logic arrays are used in order to modify the electrical currents being output to the medium.

The EAC boards produced by Mills and Himebaugh use black conductive foam as the computational medium and make use of an array of Lukasiewicz logic units. These logic units

modify the output of the conductive sheet by performing piecewise-linear functions. These boards also contain an Ethernet card which allows them to be controlled over the Internet; however, reconfiguration of the sources and sinks must still be done by hand while sitting in front of the devices.

Currently, a set of Perl utilities can be used which provide mechanisms for connecting remotely to the EAC boards. Once connected, Lukasiewicz functions can be selected as well as desired voltage levels. After instructing the EAC to place current on the foam, the voltage gradient can be read from a two dimensional array of voltmeters connected to the underside of the foam.

These EAC boards can be used for many different computational tasks. One particularly useful way these boards can be employed is in performing pattern recognition. As explained by Mills [8], the EAC can be trained to pick out certain patterns. Mills described a mechanism for recognizing characters of the alphabet. In this case, simple rules were used in order to determine which Lukasiewicz functions would properly detect different letters from a small set.

Similar techniques extend to the task of detecting DoS attacks. Regular, desirable traffic can be characterized be a set of patterns. Traffic produced from an attack, will generally not behave the same way as legitimate traffic. Because of these differences, a traffic pattern recognition system could be produced in order to detect an attack. By quickly discovering an attack in progress, it would then be possible to take evasive measures to limit its effectiveness or stop it all together.

One possible method for combating an ongoing attack would be to implement connection push-back [3] in core routers, preventing the attack from reaching the intended victim.

## III. Methodology

In this paper, we examine two approaches to using the EAC to detect denial of service attacks. In the first method, we arrange the sources and sinks linearly at opposing ends of the board. In the ring method, we arrange the sources and sinks in a circular arrangement with an LLA in the center of the circles.

When configuring the EAC for detecting denial of service, there are several factors that must be considered. The board's dimensions, the number of sink and source lines available, and the board's shape each play a role in the result.

The board's length and width determine the amount of computation that can be performed on the board. Likewise, the number of data sample points bound the precision that can be achieved from our experiments.

The EAC has a limited number of source and sink leads. This limits the number of source and sink points that can be used in the configuration. In our experimentation, this results in reduced accuracy.

The shape of the board is also an important consideration. The EACs available at Indiana University are rectangular in shape with data sampling points distributed in an equally-spaced grid. This creates a trade-off in our linear approach but does not affect our ring approach.

In order to evaluate both of our board configurations, we used an IP address distribution to determine the amount of current to apply to the source and sink points on the board. Ideally, the source IP addresses from a packet capture were then hashed into one of eight bins. After 50 source addresses were placed into the bin, current proportional to the number of entries in each bin were written to corresponding source and sink points on the board.

The packet capture we originally intended to use was for the Abilene backbone network [9]. Unfortunately, this capture was not representative of normal traffic due to its high degree of private IP addresses. Instead, we modeled the normal traffic patterns by selecting random numbers between 0 and 7 inclusively and distributed these into their respective hash bins.

To simulate a denial of service attack, we biased the random selection to have our fifth bin populated an additional 10% of the time. This would be equivalent to seeing the same IP address a greater number of times, in which case it would be hashed to the same bin, resulting in a higher bin count for that bin.

The key difference between the two approaches is the amount of digital processing required to evaluate the EAC's output. While the linear method does not require the use of an LLA to perform the computation, it requires substantial digital analysis of the EAC's output. The ring method, on the other hand, does not require this digital computation because the LLA will handle the determination of whether or not a denial of service attack has been detected.

In the linear method, we placed the source and sink points on opposing ends of the EAC's conductive foam. Each source point has a corresponding sink point on the other side of the board. We used a single sink point to remove a small amount of current from the board. This configuration is illustrated in figure 1.

As previously mentioned, the board's shape creates a trade-off between two different goals. If the sources and sinks are stationed the short edge of the board, a greater history of past could be recorded. If the they were instead stationed along the long side of the board, a greater number of source and sink points could be used without interfering with each other. Because we are unable to run the simulation in real time (due to the digital to analog communication bottleneck) and because we are already limited by the number of sink and source lines, there was no particular advantage to either arrangement. We arbitrarily selected to place the sources and sinks along the shorter edge of the board.

Using the linear method, normal Internet traffic would appear as a steady gradient from a source to its corresponding sink at the opposite end of the board. However, in the presence of a denial of service attack, a disproportionate amount of current will flow out of a given source. This will result in a gradient with local maxima and minima. The gradients are then processed by a digital system to determine if the gradient

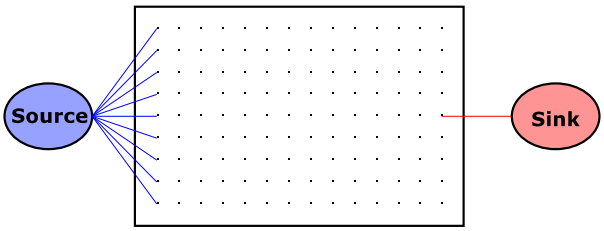


Fig. 1. Diagram of board setup for linear approach.

has maxima and minima or if the gradient is relatively smooth.

The ring method is less sensitive to the dimensions of the EAC board. To configure the board, we started by choosing a point near the center of the EAC and connecting that point to a Lukasiewicz logic array (LLA). We then form a circle with this LLA point as its foci. We then form an outer ring of sinks which are equally spaced from their matching source ring points. This layout is illustrated in figure 2.

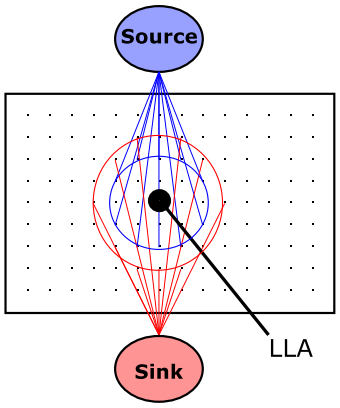


Fig. 2. Diagram of board setup for ring approach.

We again used hashing to determine the current which will flow to the EAC from the source points. Likewise, all sinks in the ring method are arranged to draw an average of the source currents off the board. Under this configuration, normal Internet traffic would result in the sinks completely absorbing the source point's current. However, when faced with a denial of service attack, a higher amount of current would come from a given source and would not be completely absorped by its corresponding sink. This increased amount of current would then be detected by the LLA at the center of the ring. This would indicate the presence of an attack to a digital computer.

The ring method is much more advanced due to a bulk of the post-gradient evaluation being done by the EAC, reducing the dependence on a digital computer.

## IV. Results

In order to examine the binning results, we distributed 100 points into our hash bins and then evaluated the bin distribution. This proceedure resulted in approximately 415 sample points. Figure 3 demonstrates the distribution for normal traffic patterns. This figure demonstrated that each bin had roughly the same number of elements.

To simulate a denial of service attack, we introduced a 10% bias on the fifth bin. This resulted in this bin having roughly double the number of elements present in the other bins. This distribution is illustrated in figure 4.

We then took samples from the normal traffic and the attack traffic distributions and wrote proportional currents to the EAC. In both cases, we multiplied the bin count by five to determine the respective current for each source point.

For the linear approach, we determined that using a sink current of 15 uA created gradients that best demonstrated the differences in source currents. From this, we were able to produce a gradient, shown in figure 5 of normal traffic patterns. When we included our simulated denial of service attack, figure 6, a noticable spike was present in the graph, reflecting the abnormal traffic for that source point.

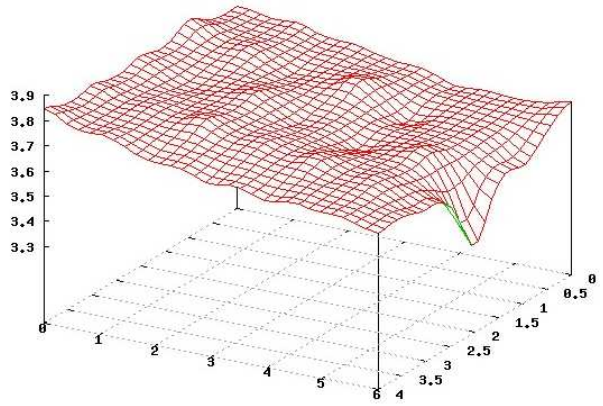


Fig. 5. This gradient shows normal traffic patterns from our linear approach.

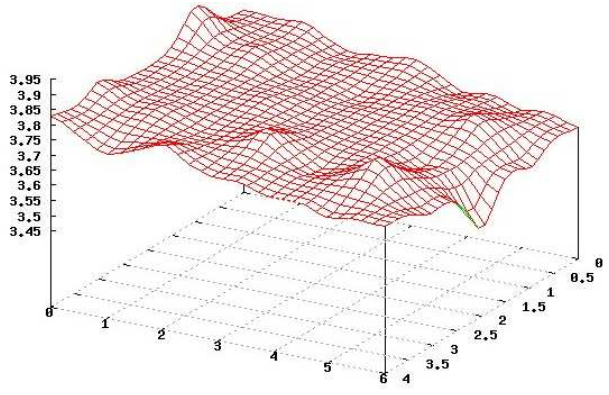


Fig. 6. This gradient shows attack traffic patterns from our linear approach.

For the ring approach, we repeated the source current calculations for the linear approach. However, we also arranged sinks next to these source points. Each sink drew 62.5 uA of current, which was the average of the source currents. The normal traffic, depicted in figure 7, shows each of the source points as peaks with the center of the circle appearing flat.

When we used our attack traffic, shown in figure 8, the bin with the highest traffic had a peak higher than the other bins. We note the scale on this graph indicated a substantial increase

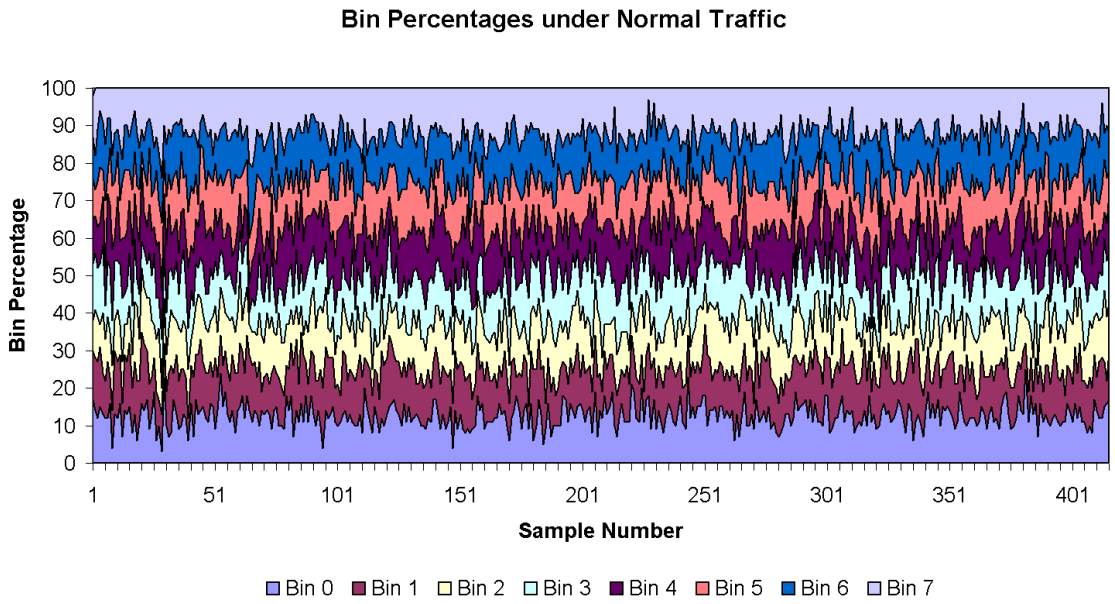


Fig. 3. This figure depicts the bin distribution of normal traffic. Each bin contains roughly the same number of elements.

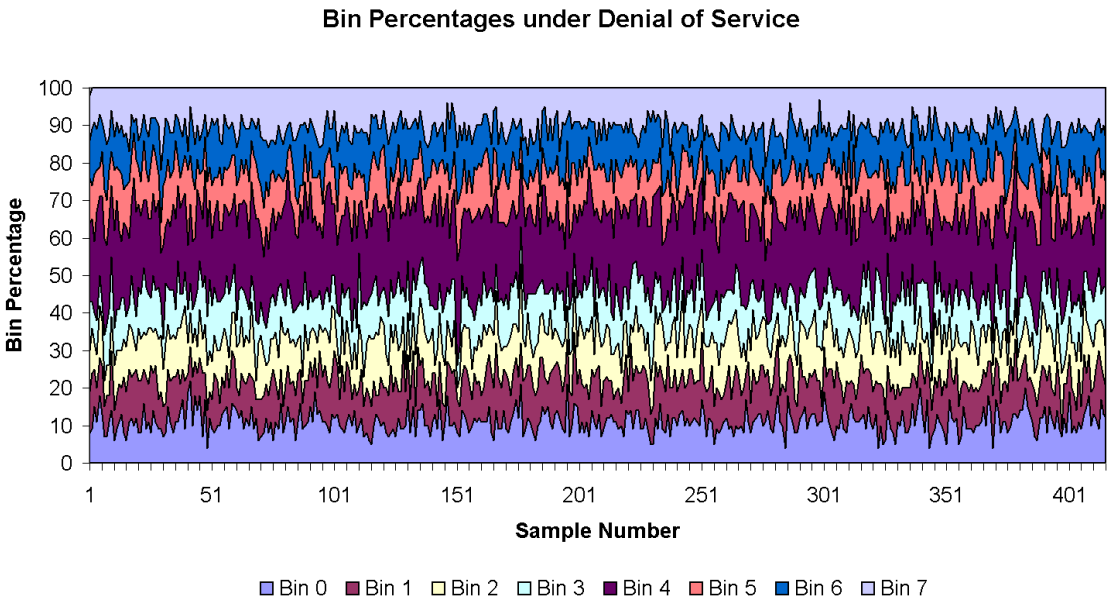


Fig. 4. In the case of denial of service activity, one bin will have more elements than the others. In this case, an attacker was hashed to the fourth bin.

in traffic, even though the current flow was normalized for the average case. These results seem to indicate that an LLA positioned in the center of the circle would be able to distinguish normal and attack traffic from each other.

## V. Discussion

Our results indicate that denial of service activity can be depicted on an extended analog computer. Additionally, the gradients obtained from the EACs can be digitally analyzed to automatically detect such activity.

Our ring approach yielded results which seem to support using an LLA at the center of the graph to determine the presence of a denial of service attack. Further, the EACs are still being developed and have great potential for increased functionality. If LLA functions were created to be able to easily determine the presence of abnormal current saturation, denial of service detection could be performed on the analog system itself, reducing the demands on digital systems.

Future extended analog computers could be designed to have much greater dimensions and more numerous source and sink leads, resulting in greater accuracy from our approaches. Additionally, EACs can be arranged in multiple-board architectures, allowing for more complex analysis without requiring additional digital to analog communication. A multiple board configuration has the potential to greatly reduce the EAC's

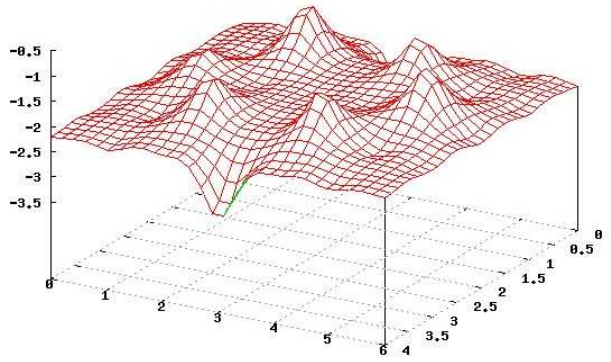


Fig. 7. This gradient shows normal traffic patterns from our ring approach.

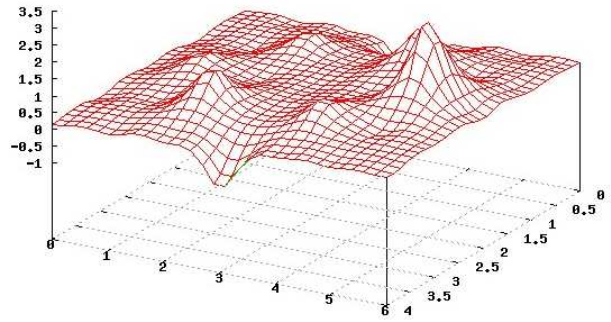


Fig. 8. This gradient shows attack traffic patterns from our ring approach.

dependence on digital systems for post-processing.

### A. Future Work

In our configuration, we cannot do real-time writing and sampling of data because of a communication bottleneck between the digital and analog systems. Future work may make it possible to overcome these limitations and provide an advantage to using the shorter side of the board. In this manner, current could constantly be varied across the board and data points read quickly enough to capture a history of traffic patterns on each column.

Another approach would be to hook up additional boards to each row of the original EAC. These boards could use these currents as a source and examine whether the currents peaked over a period of time, indicating an increase in the amount of traffic in a particular bin. This could be used for flash-crowd detection or even a DoS attack, if the results were communicated with the other boards.

## ACKNOWLEDGMENTS

We would like to thank Bryce Himebaugh for his insight on configuring the EACs as well as for providing a simulator for testing our configurations.

## REFERENCES

[1] L. Garber, "Denial-of-service attacks rip the internet," *IEEE Computer*, 2000.

[2] S. Staniford, V. Paxson, and N. Weaver, "How to own the internet in your spare time," USENIX Security Symposium, 2002.

[3] J. Ioannidis and S. Bellovin, "Implementing pushback: Router-based defense against ddos attacks," Network and Distributed System Security Symposium, 2002.

[4] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM SIGCOMM Computer Communication Review*, 2001.

[5] D. Moore, G. Voelker, and S. Savage, "Inferring internet denial-of-service activity," USENIX Security Symposium, 2001.

[6] A. Akella, A. Bharambe, M. Reiter, and S. Seshan, "Detecting DDoS attacks on ISP networks," Workshop on Management and Processing of Data Streams, 2003.

[7] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to DDoS attack detection and response," DARPA Information Survivability Conference and Exposition, April 2003.

[8] J. W. Mills, "The continuous retina: Image processing with a single-sensor artificial neural field network," IEEE International Conference on Neural Networks, 1996.

[9] N. L. for Applied Networking Research, "Abilene-iii trace data," June 2004, http://pma.nlanr.net/Special/ipls3.html.