A New Approximated VC Scheme to Prevent D-DoS Attack against Networks of ASes

Hoon-Jae Lee[0], Han-Soo Kim, Ju-Wook Jang
Dept. of Electronic Engineering Sogang University

Sinsu-dong 1, Mapo-gu,
121742 Seoul, Korea

{steak[0], kutestar, jjang}@eeca1.sogang.ac.kr

Abstract

Distributed denial of service (D-DoS) attacks continue to threaten the reliability of networking systems. An efficient way to prevent D-DoS Attack is to install filters at AS (Autonomous System) border routers. This requires to find minimum number of filters to cover the network or an optimal VC(Vertex Cover). It is an NP-complete problem. A heuristic algorithm called approximated VC scheme[1] has appeared in the literature. However, the sizes of the VCs resulting from the scheme tend to be much larger than those of the optimal VCs. We propose a new approximated VC scheme which improves the scheme[1] in the sizes of resulting VCs. The basic idea behind our scheme is to sort the nodes in non increasing order of the degree(number of links connecting the node) and install the filters in that order until the whole network is covered. Our simulation shows that the proposed scheme is able to prevent D-DoS Attack with the size of VC 26% less than the approximated VC algorithm.[1].

1 Introduction

Denial of Service (DoS) attacks consume resources in networks, server cluster and hosts, with the malicious objective of preventing or degrading service to user.[8] Resources that are typically consumed in such attacks include network bandwidth, server or router CPU cycles, server interrupt processing capacity and specific protocol data structures. In denial of service (DoS) attack, a malicious user exploits the connectivity of the internet to cripple the services offered by victim site, often simply by flooding a victim with many requests. A DoS attack can be either a single-source attack, originating at only one host, or a multi-source, where multiple hosts coordinate to flood the victim with a barrage of attack packet. The latter is called a distributed denial od Service(D-DoS) attack. distributed Denial of Service (D-DoS) attack uses numerous spoofed IP address to make malicious TCP/IP connections. D-DoS attack aims to make Internet site resource depletion and clogging.[9]

To prevent distributed DoS(D-DoS) attack, we use route based distributed packet filtering. Route-based distributed packet filtering (DPF) uses routing information to determine if a packet arriving at a router is valid with respect to its inscribed source/destination address. The way to prevent D-DoS is filtering out spoofed packet flows at AS border router by identifying source IP address.[1,8,9] To prevent D-DoS attack, one of the connected two nodes should be installed packet filter.

In this paper, we want to minimize the number of router to install filter, first of all, we need to find Vertex Cover(VC) in network topology in AS border router level and then to install filter to VC of topology.

The rest of the paper is organized as follows. In the next section, we give a summary of related works. Section 3 describes characteristics of networks of ASes. And in section 4, discusses the method to find a new approximated VC scheme. Section 5 presents simulation details, section 6 presents performance results based on inet topology, and Section 6 concludes with a discussion of our results.

2 Related Work

In this paper, we describe and evaluate route-based distributed packet filtering (DPF), an approach to distributed DoS attack prevention. Route-based distributed packet filtering (DPF) uses routing information to determine if a packet arriving at a router is valid with respect to its inscribed source, destination addresses. In Route-based DPF, the fraction of AS's from which spoofed IP flows can reach other As's is a small subset (less than 12%) which makes harnessing attack sites when engaging in D-DoS attack more difficult for an attacker.[1]

To install the minimum filter in network topology of ASes is to find VC node in network graph. So we study about vertex cover (VC) and then focus on finding VC to apply this approach to networks of ASes.

Vertex Cover (VC) is vertex set which covers all edge of graph, finding minimum VC is NP-complete problem. In this case, it is impossible to implement minimum VC algorithm.

2.1 Approximated VC

Since finding minimum VC is NP-problem, we want the heuristic method that has more than minimal VC nodes, but it has less time-complexity. General approximated VC[1] usually chooses edge randomly, deletes all edge connected to chosen edge, and then repeats this procedure. This method allows us to have twice VC nodes than minimum VC nodes.

Finding approximated VC Algorithm[1] is as follows.

```
Pseudo Code of Finding Approximated VC
  Approx-Vertex-Cover(G) ; G = (V, E)
     C = {}
     E' = E
       while E' {}
          (u,v) = arbitrary edge from E'
          C = C {u,v}
          remove from E' every edge incident on u or v
       return C
```

Table 2.1 Pseudo Code of Finding approximated VC

3 Network Characteristics of ASes

We want to use approximated VC to install packet filter, but we want to propose a more appropriate method to real AS topology. [1]

During 1997~99, AS topology has generated high-degree nodes as a part of AS backbone which has full-mesh form. Since then, low-degree nodes have been attached to this full-mesh structure. Therefore, when the total nodes are 4,000, the highest node has 856 degrees and 1,199 nodes have only 1 degree. According to figure 3.1, AS network nodes show Power-Law distribution that is consisted of few high-degree nodes and many low-degree nodes. [3]

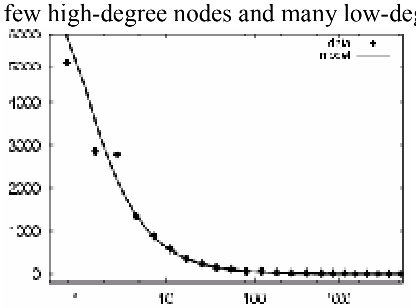

Figure 3.1 Power Law Distribution

Power Law can well describe AS (Autonomous System) level internet topology. Table 3.1 represents maximum degree nodes and average degree of real AS network

	int-11-97	int-04-98	int-12-98
nodes	3015	3530	4389
edges	5156	6432	8256
maximum degree	590	745	979
average. degree	3.42	3.65	3.76

Table 3.1 Maximum Degree and Average Degree of real AS Network[4]

3.1 Approximated VC Scheme using Power Law

To find approximated VC can be solved by classifying both sides of bipartite graph. Topology can be divided into two node-sets as to network characteristics, especially the degree of links. Consequently we need more efficient scheme that easily filters out high-degree nodes.

4 Proposed Approximated VC Scheme

We want to explain the procedures of a proposed scheme appropriate to real AS network. Firstly, choose maximum degree node. Do not install filter to nodes that are adjacent to filter-installed node. Secondly, among these nodes install the filters in the node with decreasing degree. In this procedure, if connection formed loop, move node to undetermined state. Repeat these two steps until all nodes are determined to be installed or not.

1. Install filter to maximum high-degree node
2. **Do not install the filter which is adjacent to filter-installed nodes**
3. **Install filter to nodes adjacent to filter-uninstalled nodes**
   - **If filter has already installed to adjacent node, stop the procedure.**
4. repeat step 2 and 3, until all nodes are determined to install or not

Table 4.1 Proposed approximated VC scheme

As seen in Table 4.1, we can come to a conclusion that one step chooses higher-degree node and install filter and then the other chooses lower degree and not to install filter and do these two steps repeatedly. Finally we can classify all nodes to high degree nodes and low degree nodes like Figure 4.1.

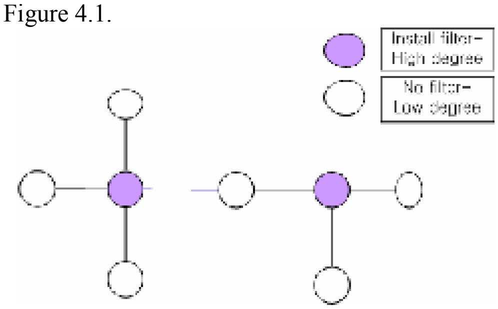

Figure 4.1 Overview of proposed approximated VC scheme

```
- Initialize -
filter-install array = [ no nodes are determined ]
first array = [ 0 ] (choose maximum-degree node)
- Iteration - ( repeat until all nodes' determination)
{
    this array = [ sorting in order of low-degree ]
    next array = [ ]
    (repeat until all nodes of this array)
    {
        determine installation or not of all nodes of this array
        add install-determined nodes to next array
    }
    copy next array to this array
    delete next array
}
- result -
filter-install array = [list of filter-install determination ]
```

Table 4.2 Pseudo Code of Filter Installer

5 Implementation and Simulation

We use Inet 3.0 to generate AS networks which is similar to real networks of ASes. And then we make Inet Parser in order to use Inet data. Inet Parser is made to provide each node's number of degree and decide filter installation using proposed approximated VC scheme.

5.1 Inet 3.0 : Topology Generator

Inet 3.0 is AS level Internet topology generator. It generates random networks with characteristics similar to those of Internet from 1997 to 2002.[5] The format of Inet 3.0 is as follows :

- ■ total number of links and nodes
- ■ location of nodes
- ■ list of links in topology
- ■ weight of each links

5.2 Inet Parser

As described above, inet topology generator sorts nodes in order of low-degree. In other words, node id 0 has maximum degree and node id 3999 has degree 1. As to proposed approximated VC, we want to determine whether we install the filter or not. Thus we implement inet parser that analyzes inet topology and filter installer that determines to install filter or not.

inet parser creates degree table using inet data table, and filter table records each nodes' filter installation or non-installation using degree table and filter table. In this procedure, two arrays are used for recursive calculation.

Inet Parser creates three array and two table. Each array and table functions are as follows.

■ Next array
add searched nodes to array

■ Current array
add Next array to adjacent nodes in order of low-degree

■ Filter - Installed array
write filter installation or not

■ inet data table
list of connected nodes' id

■ degree of each node table
list of each nodes' degree.

6 Simulation Results and Analysis

As a result, we can have two tables. one is each node's degree table and the other is filter installation table. node's degree table indicates number of degree and filter installation table informs each nodes' filter installation or not. So in order to compare proposed approximated VC to general approximated VC[1], we look into the difference of the number of VC nodes and time-complexity.

6.1 Number of Degree

When Our total simulation nodes are 4,000, the highest node has 856 degrees and 1,199 nodes have only 1 degree.

Node id	Number of degree	Node id	Number of degree
0	856	...	...
1	468	3981	1
2	329	3982	1
3	232	3983	1
4	182	3984	1
5	151	3985	1
6	130	3986	1
7	114	3987	1
8	102	3988	1
9	92	3999	1
10	85	3990	1
11	78	3991	1
12	73	3992	1
13	68	3993	1
14	64	3994	1
15	60	3995	1
16	57	3996	1
17	54	3997	1
18	52	3998	1
...	...	3999	1

Table 6.1 each Nodes' number of degree

6.2 Number of VC nodes to Install Filter

Simulation results as follows.

node id	filter	node id	filter	node id	filter
0	O	1051	X	3581	X
1	O	1052	O	3582	X
2	O	1053	O	3583	O
3	O	1054	X	3584	X
4	O	1055	O	3585	O
5	O	1056	O	3586	O
6	O	1057	O	3587	X
7	O	1058	O	3588	O
8	O	1059	O	3589	X
9	O	1060	O	3590	O
10	O	1061	O	3591	X
11	O	1062	O	3592	O
12	O	1063	O	3593	X
13	O	1064	O	3594	X
14	O	1065	O	3595	X
...	...	...	...	...	...

Table 6.2 Output of Filter Installation Array

As shown in table 6.2 filter installation array determines installation or non-installation. Most of filters were installed to high-degree nodes. In other words, as we're expected, lower node id was rarely installed to lower-degree nodes. As a result, filter-installed nodes are 2990 nodes and uninstalled nodes are 1010 among the total nodes of 4,000.

Now compare proposed approximated VC scheme with Approximated VC that was cited in paper [1]. Coverage ratio(|VC|/n), the number of VC nodes out of total nodes is 0.2525(1010/4000), which is 26% improved than general Approximated VC 0.34(1360/4000).

	approximated VC[1]	proposed approximated VC
# of VC	1360/4000	1010/4000
(|VC|/n	0.34	0.25
Improvement	(1360-1010)/1360=26%	

6.3 Time Complexity

In comparing time-complexity with general approximated VC, proposed approximated VC time-complexity is only n^2 increased. It is complexity that consider n nodes' n nodes connection so as to add adjacent nodes in order of low-degree. Consequently, it is shown that the number of nodes is deceased and polynomial time-complexity is increased.

7 Future Works

Nowadays many simulators and mathematical methods are developed to make more exact network topology generator, thus in-depth study of this research will be needed.

We can find that VC is the most appropriate application to prevent D-DoS attack from our simulation. However we need to study the deeper meaning of VC in network topology and its application to networks topology.

8 References

[1] Ki-hong park and Hee-jo Lee, "*On The Effectiveness of Route-based Packet Filtering for Distributed DoS attack Prevention in Power-law Internets*", SIGCOMM, 2001

[2] Eran Halperin, "*Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs*", SODA 2000

[3] Mikko Vapa, "*Power-Laws in Distributed Systems*", 2003, http://tisu.it.jyu.fi/embedded/TIE370/TIE370.htm

[4] Michalis Faloutsos, Petros Faloutsos and Christos Faloutsos, "*on Power-Law Relationships of the Internet Topology*", 1999

[5] Jared Winick and Sugih Jamin, "*Inet-3.0 : Internet Topology Generator*", 2001

[6] Tian Bu and Don Towsley, "*On Distinguishing between Internet Power Law Topology Generators*", IEEE 2002

[7] Qian Chen, Hyunseok chang, Ramesh Govindan, Sugih Jamin, Scott J. Shenker, Walter Willinger, "*The Origin of Power Laws in Internet Topologies Revisited*" IEEE 2002

[8] Kevin J. Houle, George M. Weaver, "*Trends in Denial of Service Attack Technology*"

[9] Alefiya Hussain, John Heidemann, Christos Papadopoulos, "*A Framework for Classifying Denial of Service Attacks*", SIGCOMM 2003