향상된 VC 근사기법을 이용한 AS망에서의 D-DoS 공격의 효율적 차단

김한수 이훈재 장주욱
서강대학교 전자공학과
{ kutestar, steak, jjang}@eeca1.sogang.ac.kr

D-DoS Attack Prevention Using Improved 'Approximated VC' in AS Network Topology

Han-Soo Kim Hoon-Jae Lee Ju-Wook Jang
Dept. of Electronic Engineering, Sogang Univ.

Abstract

The prevention of D-DoS Attack requires to install filters at AS border routers. This follows that finding minimum number of filters - VC(Vertex Cover), which is NP-complete problem. So, We propose improved 'Approximated VC' which is more efficient to real AS topology using topology property. Simulation shows that our algorithm, improved 'Approximated VC' enables us to reduce 26% VC nodes in comparison with 'Approximated VC'.

요 약

D-DoS 공격을 차단하기 위해서는 AS 경계 라우터에 필터 설치가 필요하며, 이는 최소한의 라우터에 필터를 설치하기 위해 VC(Vertex Cover)를 찾아내는 NP-complete 문제로 귀결된다. 따라서 실제 AS 망구성의 특성을 이용해 이에 적합한 VC 근사기법을 찾아내는 알고리즘을 제안한다. 실험 결과, 제안된 알고리즘(Improved 'Approximated VC')은 기존의 'Approximated VC'에 의해 필요한 노드수의 26%를 줄였다.

1. 서론

Distributed Denial of Service (D-Dos) 공격이란 자신의 IP를 숨기고 다른 임의의 IP를 도용하여 수많은 TCP 커넥션을 맺어 특정 site의 resource를 고갈시켜 인터넷 장애를 일으키는 것을 말한다. 이를 막기 위해, AS의 Border Gateway 라우터에 source IP를 보고 도용된 IP를 가진 패킷을 걸러내는 필터를 설치하였다[1].

D-DoS를 막으려면 연결된 두 노드 중 한쪽에는 반드시 필터가 설치되어야 하고, 필터를 설치할 라우터의 수를 최소화하기 위해서 AS border 라우터로 구성된 전체 토폴로지에서 Vertex Cover(VC)를 찾아 그 라우터에만 필터를 설치하고자 한다.

1.1 VC(Vertex Cover)

VC(Vertex Cover)는 모든 edge를 커버하는 vertex의 모임, 즉 연결된 두 노드 중 적어도 어느 한 쪽을 선택하였을 때의 노드들의 모임을 지칭하며, minimum VC를 찾는 일은 NP-complete problem이다. 따라서 minimum VC를 찾는 알고리즘의 구현은 불가하다.

1.2 Approximated VC

minimum VC를 찾는 일은 NP-Problem이므로, 이에 근사하고 낮은 시간 복잡도를 갖는 Approximated VC를 실현해야 한다. 통상적인 Approximated VC는 edge를 랜덤하게 선택하고 그 edge에 연결된 두개의 노드에 근접한 edge를 그래프 상에서 모두 제거하여, 이를 반복하는 방법이다. 이 방법으로는 이론상의 minimal VC보다 2배의 VC 노드수를 가질 수 있다.

2. 알고리즘에의 접근

논문 [1]에서는 Approximated VC를 적용하여 필터를 설치하였으나 실제 AS 토폴로지에 좀더 적합하게 minimum VC를 찾고자 한다.

2.1 실제 AS Topology에 대한 특성 분석

현재 AS 토폴로지는 과거 97-99년 까지 AS 계층에서 backbone 역할을 하는 full-mesh 구조의 degree가 매우 큰 노드들이 생겨났고, 그 이후에는 degree가 1 혹은 2정도의 AS border 라우터들이 붙여진 형태를 가진다.

따라서 전체 노드수가 4000개일 때, 856개의 degree, 즉 연결된 노드를 갖는 한 개의 노드와, 1짜리 degree를 갖는 1199개의 노드가 존재하였다. 즉, 그림2.1과 같이 실제 노드는 degree가 높은 쪽에 적은 수의 노드가 몰리고 degree가 적은 쪽에 많은 수의 노드가 몰리는 Power-law 형태의 분포를 가지게 된다[3].

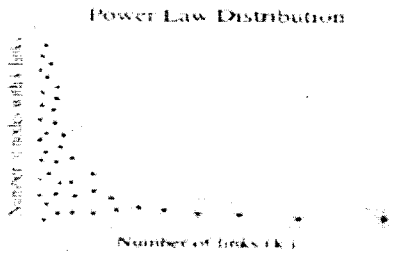

그림 2.1 Power Law 분포 곡선

표 2.2는 실제 AS 망에서 가장 큰 degree를 가지는 노드와, 각 노드의 평균 degree를 나타낸다.

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

표 2.2 실제 AS망의 max degree와 avg degree[4]

따라서 위와 같은 degree가 큰 수개의 노드들이 존재하는 AS 토폴로지에 좀더 효과적인, degree가 높은 쪽의 노드에 확실히 필터를 설치하기 위한 알고리즘이 필요했다.

2.2 알고리즘에의 접근 방식

'Approximated VC'를 찾는 문제는 bipartite graph에서 양단을 어떤 특성을 기준으로 분류해 낼 수 있느냐의 문제로 볼 수 있다. 토폴로지를 간단히 두 노드의 연결의 집합이라고 생각하고 두 노드를 한개는 왼쪽, 한개는 오른쪽으로 분류할 때 두 집합 중 한 쪽에 필터를 설치하는 방법을 적용한다.

3. 제안된 알고리즘

먼저, degree가 가장 큰 노드를 선택하여 필터를 설치한다.

필터가 설치된 노드에 연결된 모든 노드에 필터를 설치하지 않는다. 이 노드들 중 degree가 적은 노드부터 순서대로 인접한 모든 노드에 필터를 설치한다. (토폴로지에는 연결이 loop를 이루는 경우가 발생하므로, 이런 경우에 설치/미설치의 단계를 이동한다.)

모든 노드에 필터 설치/미설치의 여부가 결정 될 때까지 위의 두 과정을 반복한다.

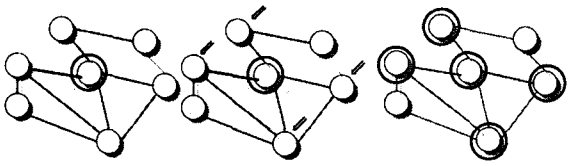

그림 3.1 제안된 알고리즘의 순서도 a), b), c)

1. 가장 degree가 큰 노드에 필터 설치<그림 3.1 a) >
2. 필터가 설치된 노드에 연결된 노드에는 필터 미설치 <그림 3.1 b) >
3. 필터가 미설치된 노드 중 degree가 적은 노드부터, 인접한 모든 노드에 필터 설치<(그림 3.1 c) >
   ■ 필터 설치된 노드 만나면 End
4. 모든 노드가 필터 설치 여부가 결정될 때까지 2번과 3번 과정을 반복

그림 3.2 알고리즘의 적용단계

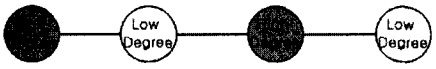

그림 3.3 알고리즘의 적용순서

결국 그림3.1, 3.2, 3.3과 같이, degree가 높은 노드를 선택하여 필터를 설치하고, degree 낮은 노드를 선택해 필터 설치를 하지 않는 과정들이 반복적으로 이뤄지게 된다.

3.1 filter installer의 pseudo code

```
 - initialize -
filter-install array = [ 모든 노드가 미결정 ]
first array = [ 0 1 2 ] (가장 degree가 높은 노드들)

 - iteration -
(모든 노드에 설치여부가 결정될 때 까지 반복)
{
    this array = [ degree 순서대로 정렬 ]
    next array = [ ]

    (this array가 끝날 때까지 반복)
    {
        this array의 노드와 인접한 노드에, 필터 설치 여부를
결정
        설치가 결정된 노드는 next array에 추가함
    }
    next array를 this array로 복사함
    next array를 비움
}

 - result -
filter-install array = [ 모든 노드가 필터 설치 또는 미설]
```

3.2 inet 3.0과 알고리즘

inet 3.0은 현재 구성되어 있는 실제 AS 토폴로지의 노드 및 링크의 비율에 맞게, 원하는 수만큼의 노드로 구성된

토폴로지를 제공한다.[5] 12000여개의 실제 토폴로지는 실험하는 데 어려움이 있으므로, 실제와 같은 비율로 축소한 4000개의 노드를 가진 토폴로지를 제공하는데, 그 토폴로지의 정보는 다음과 같다.

- 전체 링크 및 노드의 수
- 노드의 위치(x,y)
- 각 노드의 id 및 연결된 노드 id
- 각 링크의 용량

4. 구현 및 실험

4.1 실험을 위해 구현한 topology generator 및 filter marker

위에서 설명했던 inet 토폴로지는, link의 수에 대해 내림차순으로 node가 정렬되어 있다. 즉, 0번 노드가 가장 많은 링크를 가지며, 3999번 노드는 1개의 링크만을 가진다. 따라서 이를 해석하여, 제안한 알고리즘대로 각 노드의 필터 설치 여부를 결정하려 하였다. 이를 위해, 먼저 inet 토폴로지를 해석하는 parser와, 이를 가지고 노드의 필터 설치 여부를 결정하는 filter installer를 구현하였다.

parser에서는 inet data table을 가지고 각 노드의 degree를 계산하는 degree table을 생성하며, filter installer에서는 이 두 *table을 이용하여 filter-installed array에* 각 노드의 필터 설치 여부를 기록한다. 이 과정에서, recursive한 계산을 위하여 두 개의 array가 사용된다.

각 array와 table의 기능은 다음과 같다.

- Next array

search된 노드를 array에 추가한다.

- Current array

degree가 낮은 노드부터 주위의 노드를 Next array에 추가한다.

- Filter - Installed array

노드에 filter가 설치되었는지의 여부를 기록한다.

- inet data table

inet에서 제공한, 각 노드에 연결된 노드들의 번호가 기록된 테이 블.

- degree of each node table

각 노드의 degree를 계산한 table.

5. 실험 결과 및 분석

5.1 VC 노드수 분석

위 과정에 의해 실험한 결과는 다음과 같다.

노드번호	설치여부	노드번호	설치여부	노드번호	설치여부
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
...	...	...	...	...	...

표 4.1 각 노드에 대한 필터 설치 여부(filter installation array)

표 4.1과 같이, *filter installation array*에 각 노드의 필터 설치 여부가 결정된다. 예상했던 결과와 같이, 링크가 많은 노드, 즉 노드번호가 작은 노드에는 대부분 필터가 설치되며, 링크가 적은 노드에는 미설치되는 수가 많아짐을 볼 수 있다. 총 4000개의 노드 중, 필터가 설치된 노드는 2990개, 미설치된 노드는 1010개임을 확인하였다.

Improved Approximated VC와 논문상에서 근거 알고리즘으로 제시한 기존의 Approximated VC를 비교해 보기로 하자. 전체 노드중 VC 노드의 개수(coverage ratio(|VC|/n))를 보면 0.2525(1010/4000)가 나왔으며 이는 기존의 Approximated VC의 0.34(1360/4000)보다 26%의 향상이 있었다.

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

5.2 시간 복잡도(Time-complexity) 분석

시간 복잡도의 비교를 하면, 기존의 Approximated VC보다 복잡도는 n^2만이 늘어난다. 이는 degree가 가장 낮은 노드부터 주변노드를 추가할 때 n개의 노드가 n개의 노드 연결여부를 계산하는 복잡도 증가이다.

즉, VC의 개수는 줄면서 VC를 찾는데 걸리는 시간 복잡도가 polynomial로 늘어나서 추가로 계산에 필요한 시간 증가가 크지 않았다는 것을 볼 수 있다.

6. 결론 및 추후과제

시뮬레이션을 위한 좀더 정확한 네트워크 토폴로지 구성을 위한 많은 툴들과 수학적 기법들이 개발되고 있으며 이에 대한 좀더 심화적인 연구가 필요하다.

VC가 DoS 공격을 막기 위한 필터의 설치에 최적의 응용분야임을 확인할 수 있었다. 하지만 VC가 네트워크 토폴로지 측면에서 갖는 의미를 찾아내는 연구들이 현재 활발히 이루어지고 있으며, 이들에 대한 적용 방법에 대한 연구가 필요하다[6][7].

7. 참고문헌

[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