향상된 Approximated Vertex Cover(VC)을 이용한 AS망에서의 D-DoS 공격의 효율적 차단

Hoon-Jae Lee[0], Ju-Wook Jang
Dept of Elecronic Engineering Sogang University

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

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

요 약

Distributed Denial of Service(D-DoS) 공격을 차단하기 위해서는 AS(Autonomous System) 경계 라우터에 필터를 설치하는 것이 필요하다. 필터가 설치되는 라우터의 개수를 최소로 하는 Vertex Cover(VC)--모든 edge를 커버하는 Vertex의 모임--을 찾아내는 방법은 NP-complete 문제가 된다. 따라서 Vertex Cover(VC) 근사기법 중에서 Greedy 알고리즘과 Approximated VC 알고리즘에 대해 Vertex Cover(VC)을 찾아내는 방법을 적용하여 실험하였다. Vertex Cover(VC)를 찾을 경우 Worst case에서 이론상 VC수의 최대 2배의 Vertex Cover(VC)를 찾아낼 수 있는 Approximated VC 알고리즘의 장점과 적은 수의 Vertex Cover(VC)로 모든 edge를 커버할 수 있는 Greedy 알고리즘[7]의 장점을 모두 갖춘 Vertex Cover(VC) 근사 기법을 구현하였다. NS-2를 이용한 실험 결과, 제안된 알고리즘은 Greedy 알고리즘보다 Vertex Cover를 찾아가는 단계 수에 따른 커버되는 총 노드 수를 비교하였을 때, 전체 노드의 75%를 커버하는데 24번의 단계가 필요하여 Greedy 알고리즘의 40개 보다 40%의 단계의 수적인 감소가 일어났으며 전체노드의 90%를 커버하는데 38%의 단계 개수의 감소가 일어났다. 실험으로 제안된 알고리즘이 Vertex Cover(VC)를 찾아가는 단계 측면에서 좀더 빠르게 AS 경계 라우터에 필터를 설치하여 D-DoS에 효율적으로 대처해 나갈 수 있음을 확인할 수 있다.

1. 서론

Denial of Service (DoS) 공격이란 악의를 가지고 네트워크, 서버의 자원을 고갈시키는 것을 말한다. 이 중에서 Distributed Denial of Service(D-DoS)은 자신의 IP를 숨기고 다른 IP를 도용하여 수많은 TCP 커넥션을 맺어 특정 사이트의 자원을 고갈시켜 인터넷 장애를 일으키는 것을 말한다. 이를 막기 위해, AS 경계 라우터에서 지나가는 패킷의 source IP와 패킷이 지나가는 경로를 비교하여 임의로 도용된 패킷을 걸러내는 필터를 AS(Autonomous System) 경계 라우터에 설치하는 방법을 고안하였다[2].

D-DoS를 막기 위해, 네트워크의 AS 경계 라우터에서 라우팅 정보와 패킷의 출발과 도착 주소를 비교하여 지나가는 패킷이 임의로 도용된 패킷이라고 판단될 경우 필터링을 하게 된다. 이 논문에서는 AS(Autonomous System)망 레벨에서 D-DoS를 막고자 했으며, AS 경계 라우터에 필터를 설치하고자 한다. 필터를 설치할 라우터의 수를 최소화하기 위해서 AS border 라우터로 구성된 전체 토폴로지에서 Vertex Cover (VC)를 찾아 그 라우터에 필터를 설치하는 방법을 모색하였다.[7]

Greedy 알고리즘을 Vertex Cover (VC)에 적용한 방법을 적용하여 기존의 Approximated VC 알고리즘을 이용해 Vertex Cover(VC)를 찾는 방법보다 Vertex Cover(VC)의 수를 줄일 수 있었다.[10] 하지만 Vertex Cover(VC)의 개수뿐만 아니라, Vertex Cover를 좀더 빠르게 찾아서 D-DoS에 좀더 빠르게 대처할 수 있는 알고리즘의 필요성을 생각해 보게 되었다. 최악의 경우에 Approximated VC 알고리즘은 2n (단 n은 이론상 최소 Vertex Cover (VC)의 개수)의 Vertex Cover (VC)를 보장하며 Greedy 알고리즘은 log n개의 Vertex Cover를 보장하고 Greedy 알고리즘은 좀 더 적은 수의 Vertex Cover(VC)로 모든 노드를 커버하였다[1]. 따라서 Vertex Cover를 찾아가는 worst case인 경우 이론상 VC수의 최대 2배의 VC를 찾아낼 수 있는 장점을 갖는 Approximated VC 알고리즘과 적은 수의 Vertex Cover (VC)로 모든 edge를 커버할 수 있는 Greedy 알고리즘[7]의 장점을 모두 갖는 알고리즘을 제안하고자 한다.

2. 관련 연구

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, 그림1과 같이 실제 노드는 degree가 높은 쪽에 적은 수의 노드가 몰리고 degree가 적은 쪽에 많은 수의 노드가 몰리는 Power-law 형태의 분포를 가지게 된다[2,3,4].

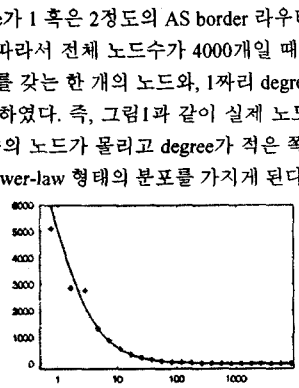

그림 1 Power Law 분포 곡선

표 1는 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 degree	590	745	979
average. degree	3.42	3.65	3.76

표 1 실제 AS망의 maximum degree와 average degree[2]

따라서 위와 같은 degree가 큰 수개의 노드들이 존재하는 AS 토폴로지에 좀더 효과적인, degree가 높은 쪽의 노드에 확실히 필터를 설치하고 좀 더 빠르게 노드들을 커버해 나가는 알고리즘이 필요했다.

2.2 기존 알고리즘

네트워크 망에서 필터를 설치할 최소의 AS 경계 라우터를 찾는 일은 그래프 이론에서 Vertex Cover(VC)를 찾는 것과 같다. VC(Vertex Cover)는 모든 edge를 커버하는 vertex의 모임을 가리키며, minimum VC를 찾는 일은 NP-complete problem이므로 이를 해결하기 위한 두 가지 근사 기법인 Greedy 알고리즘과 Approximated VC 알고리즘을 Vertex Cover(VC)를 찾는데 사용하기로 한다.

Approximated Vertex Cover(VC)

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

Greedy 알고리즘

Greedy 알고리즘은 각 상황에서 가장 최적의 선택을 하는 알고리즘이다. Greedy 알고리즘을 Vertex Cover를 찾는데 적용하는 방법은 모든 Vertex의 degree를 계산하여 그 순서대로 Vertex Cover로 만들고 이를 선택된 Vertex에 의해 모든 edge가 커버될 때까지 하면 된다. 이 방법으로 이론상 minimum VC보다 log n배의 VC 노드수를 가질 수 있다.[1]

3. 제안하는 알고리즘

제안하는 알고리즘은 worst case에서 Approximated VC 알고리즘이 VC 노드를 최대 2배까지 가질 수 있는 장점과 Greedy 알고리즘이 가진 효과적으로 VC를 찾아가는 장점을 모두 갖춘 VC 근사 기법을 제안하고자 한다. Greedy 알고리즘과 같은 방식대로 sorting을 하지만 Approximated VC 알고리즘과 같이 edge를 기반으로 한다. 즉, edge의 degree대로 올림차순으로 정렬한 뒤, 가장 많은 degree를 가지는 edge를 구성하는 두개의 Vertex를 모두 VC에 넣고 이 과정을 모든 edge가 커버될 때까지 한다. 제안하는 알고리즘의 Pseudocode는 표2와 같다.

```
Finding Approximated VC
Approx-Vertex-Cover(G) : G = (V, E)
  C = ∅
  edge-degree는 edge에 접한 두 노드의 degree의 합이다
  edge-degree을 내림차순으로 sorting하고 E라 한다
  E' = E
    while E'≠ϕ
      E'중에서 가장 큰 edge-degree를 갖는 edge를 (u,v)라 놓는다
      C = C U { u , v }
      E' = E' - { all edges incident on u and v }
    end while
  return C
```

표2. 제안하는 알고리즘의 Pseudocode

3.1 세 알고리즘의 성능 비교

NS-2 시뮬레이션을 통해 얻은 실제 AS 토폴로지의 특징을 갖는 4000개의 토폴로지를 이용해 세 알고리즘에 따른 VC를 찾는 알고리즘을 구현하여 보았다. 표1은 이를 정리한 결과이며 내용은 다음과 같다. 세 가지 알고리즘을 이용하여 VC를 찾은 결과로 각 알고리즘에 따라 모든 edge를 커버하는데 필요한 VC 개수를 실험 결과로 나타내었다. 또한 실험 결과를 좀더 비교하기 쉽게 하기 위해, VC를 찾는 효율성 측면의 지표인 coverage ratio Υ을 Υ= |VC| / n로 정의 하였다. 단, Approximated VC 알고리즘은 랜덤하게 edge를 선택해 나가는 방법을 사용하기 때문에, 실험 결과의 평균값을 표3에 나타내었다.

	Greedy	Approximated VC	제안된 알고리즘
# of VC	891/4000	1591/4000	1356/4000
(|VC|/n	0.22	0.39	0.34

표3. 세 가지 알고리즘에 따른 VC 개수 비교

실험 결과 Greedy 알고리즘이 가장 적은 수의 Vertex를 가지고 모든 edge를 커버하였으며, 제안된 알고리즘은 Approximated VC 알고리즘과 Greedy 알고리즘 사이의 값을 가진다. 위의 실험 결과를 통해, Greedy 알고리즘의 성능이 찾아낸 Vertex Cover의 개수 측면에 있어서는 가장 좋다는 것을 알 수 있다.

4. 실험 결과 및 분석

VC 개수를 찾는 실험으로 찾은 Vertex Cover의 개수 측면에서 Greedy 알고리즘이 가장 우수하다는 결론을 내릴 수 있었다. 하지만 좀 더 다각적인 성능 평가를 위해 두 가지 실험을 더 하기로 하였다.

한 가지는 하나의 VC가 증가됨에 따른 커버되는 총 노드 수를 비교하는 실험이고 다른 한 가지는 VC를 찾아가는 단계별 커버되는 총 노드 수를 비교하는 실험으로 두 가지 실험으로 각 알고리즘의 성능을 분석하고자 한다.

4.1 VC 증가에 따른 커버되는 총 노드 수

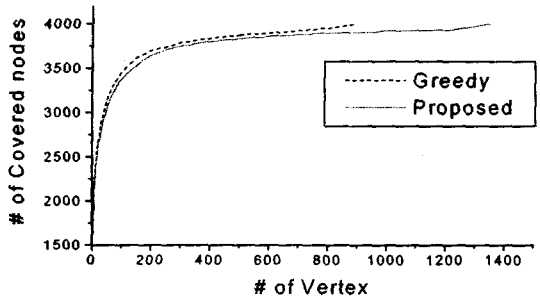

그림 2. VC 증가에 따른 커버되는 총 노드 수

그림 2는 VC가 하나씩 찾아질 때, 그에 따라 커버되는 총 노드 수를 나타낸 그래프이다. Greedy 알고리즘은 알고리즘 상, 한번에 하나의 vertex를 추가하는 반면에 제안하는 알고리즘은 edge에 접한 두 개의 vertex를 선택하기에 커버되는 노드 수가 항상 제안되는 알고리즘에 비해 높았다.

4.2 VC를 찾아가는 단계 개수에 따른 커버되는 총 노드 수

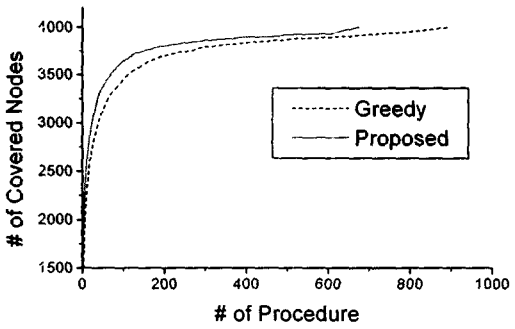

그림3. VC를 찾아가는 단계 개수에 따른 커버되는 총 노드 수

그림 3은 VC를 찾아가는 단계(Procedure)가 계속될 때, 그에 따라 커버되는 총 노드 수를 나타낸 그래프이다. 제안하는 알고리즘에서 매 단계에 두 개의 Vertex가 추가되게 된다. 위의 실험 결과에서 커버되는 노드 수가 전체 노드 수의 75%(3000개) 진입시의 VC는 Greedy가 40개, 제안되는 알고리즘은 24개로 40%의 적은 VC 찾는 단계를 거쳐서 제안되는 알고리즘이 커버를 했다.

4.3 결과 분석

실험 결과, Vertex 증가에 따른 커버되는 총 노드 수는 Greedy 알고리즘이 언제나 제안하는 알고리즘보다 좋았다. 이는 제안하는 알고리즘이 언제나 edge에 연결된 Vertex 두개를 모두 Vertex Cover로 만들기 때문이며 이를 보정한 VC를 찾아가는 단계 개수에 따른 커버되는 총 노드수를 비교하면 제안하는 알고리즘이 좀 더 빠르게 노드들을 커버해 나감을 알 수 있다.

	3000개(75%) 커버할 때의 VC 개수	3600개(90%) 커버할 때의 VC 개수
Greedy 알고리즘	40	146
제안하는 알고리즘	24	90
Improvement	40%	38%

표4. VC를 찾아가는 단계 개수에 따른 커버되는 총 노드 수 비교

5. 결론 및 추후과제

실험 결과로 제안된 알고리즘이 VC를 찾을 때 최악의 경우 이론상 2배의 VC를 가지며, 찾아가는 단계 측면에서 좀더 빠르게 노드들을 커버해 나가는 것을 볼 수 있었다. Vertex Cover가 DoS 공격을 막기 위한 필터의 설치에 있어서 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].

6. 참고문헌

[1] Steven S. Skiena, "The Algorithm Design Manual", Springer-Verlag, 1997

[2] 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

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

[7] Hoon-Jae Lee, Han-Soo Kim, Ju-Wook Jang, *A New Approximated VC Scheme to Prevent D-DoS Attack against networks of ASes*, JCCI 2004