S-RAY : Yen 알고리즘을 응용한 보안 네트워크 라우팅[1)]

정제성*, 신승원*, 김광조†

*한국과학기술원(KAIST) 정보보호학원 / † 전산학부

S-RAY : Secure Routing Algorithm based on Yen's algorithm

Je-Seong Jeong*, Seungwon Shin*, Kwangjo Kim†

* Graduate School of Information Security /

† Dept. of Computer Science, KAIST

요 약

현대는 인터넷을 통해서 각종 정보를 얻고 있다. 이는 네트워크의 발전이 있기 때문에 가능한 일이다. 하지만 네트워크의 발전에 따라서 서비스 거부 공격(Denial of Service, DoS)이라던가 중간자 공격(Man In The Middle, MITM) 같은 부작용도 발생이 되었다. 무선 네트워크에서는 이를 막기 위해 무작위 경로 선정(Random Route)을 사용 하지만 아직까지 유선 네트워크에서는 겹치지 않는 경로를 찾는 것을 찾는 것이 어려워 고정된 경로를 사용하고 있다. 물론 기존에 임의 경로 변경(Random Route Mutation) 관련하여 논문이 발표 되기는 하였지만 효율성의 문제로 현재 네트워크에서는 적용을 하고 있지 않다. 따라서 본 논문에서는 유선 네트워크에서도 사용이 가능하고 효율성이 보장 되는 Yen 알고리즘을 응용한 보안 네트워크 라우팅(Secure Routing Algorithm based on Yen's algorithm, S-RAY)를 제안한다.

I. 서론

1973년 인터넷이 최초 사용이 된 이후로 네트워크 기술은 눈부시게 발전되어 현재는 인터넷을 통해 각종 정보를 얻고 있다. 그리고 스마트폰의 보급률이 증가 할수록 이러한 현상이 지속 될 것이다. 하지만 네트워크의 발전과 인터넷의 사용이 증가함에 따라 부작용도 발생이 되어, 서비스 거부 공격(Distributed Denial of Service, DoS), 중간자 공격(Man In The Middle, MITM) 등을 받게 되었다. 무선 네트워크에서는 이러한 공격을 막기 위해서 무작위로 경로를 선택하여 전송하고 있지만 유선 네트워크에서는 Load balancing을 제외하고는 고정된 경로를 사용하고 있다. 물론 기존에 임의 경로 변경(Random Route Mutation)[1] 관련 논문이 발표되기는 하였지만 이는 효율성이 부족한 관계로 사용을 하고 있지 않다. 따라서 본 논문에서는 이러한 문제를 해결하고자 Yen 알고리즘을 응용한 보안 네트워크 라우팅을 제안한다. 이는 데이터 전송 경로를 임의로 선택하는 방식으로 고정식 경로 전송 네트워크에 비해 보안성을 향상 시켰으며 기존 임의 경로 변경(Random Route Mutation)[1] 보다 효율성을 향상 시킨 방안이다. 본 논문은 기존에 발표 하였던 논문[2]을 발전시킨 내용이다.

2장에서는 배경지식을 설명 하겠으며, 3장에서는 관련연구, 4장에서는 S-RAY, 5장에서는 평가, 6장에서는 결론을 말하겠다.

II. 배경지식

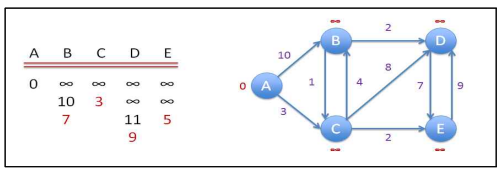
본 장에서는 다익스트라 알고리즘(Dijkstra algorithm)과 Yen 알고리즘(Yen's algorithm)에

1) 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2015R1A2A2A01006812).

대해서 설명하겠다.

2.1 다익스트라 알고리즘(Dijkstra algorithm)

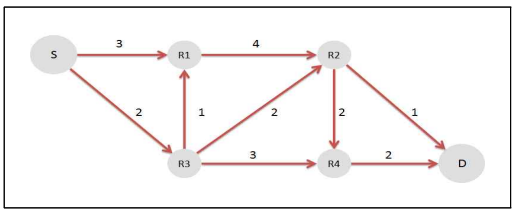
다익스트라 알고리즘(Dijkstra algorithm)은 어떤 변도 음수 값을 갖지 않는 유향 그래프에서 주어진 출발점과 도착점 사이의 최단 경로를 푸는 알고리즘이다[3]. [그림 1]은 다익스트라 알고리즘의 예시이다. 출발점을 A라 하고 나머지 꼭짓점을 목적지로 하여 거리 값을 구한다. 방법은 최초 A는 0 값으로 나머지는 ∞ 값으로 저장한다. 이후 A와 인접한 B, C 값 10과 3일 각 입력 후에 다음으로 B와 인접한 C, D와 C와 인접한 B, D, E 값을 구한 후에 A로부터의 B, C 로의 경로 값과 합한다. 이후 각 지점에서의 값을 비교해서 최소 값을 구한다.



[그림 1] 다익스트라 알고리즘(Dijkstra algorithm) 예[4]

2.2 Yen 알고리즘(Yen's algorithm)

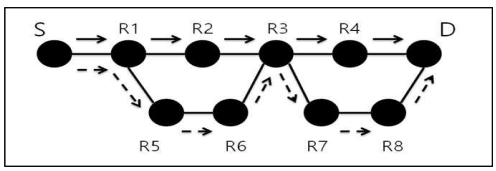
Yen 알고리즘은(Yen's algorithm)은 어떤 변도 음수 값을 갖지 않는 그래프에서 고리를 형성하지 않는 K 개의 최소 값들을 가지는 경로를 구하는 알고리즘이다[5]. Yen 알고리즘(Yen's algorithm)은 두 부분으로 나뉘어진다. 첫 번째는 검증 된 최단 경로들의 집합(A 집합)이고 두 번째는 후보 최단 경로 집합(B 집합)이다. [그림 2]는 Yen 알고리즘(Yen's algorithm) 예이다. 최초 다익스트라 알고리즘(Dijkstra algorithm)을 이용하여 최적 경로 A1(S - R3 - R2 - D, Cost : 5)를 구한 후 이를 A 집합에 저장을 한다. 이후 S - R3 구간을 ∞ 값으로 저장을 해서 출발점 S에서 R1으로 경로를 이동하게 하여 목적지 D 까지의 경로를(S - R1 - R2 - D, Cost : 8) 구한다. 이후 R3 - R2의 구간을 ∞ 값으로 놓고 D까지의 다른 경로들을 구한 후 이를 B 집합에 저장한다. 이후, B 집합 중 최소 값을 가지는 경로를 A^2(S - R3 - R4 - D, Cost : 7)로 하고 그 다음 값을 A^3(S - R1 - R2 - D, Cost : 8)로 한다.



[그림 2] Yen 알고리즘 예[6]

III. 관련연구

임의 경로 변화(Random Route Mutation, RRM)는 최대한 겹치지 않는 경로로 데이터 패킷을 전송하며, 사전 경로를 지정하는 방식이다. [그림 3] 은 임의 경로 변화 예시이다. 먼저 실선화살표(S → R1 → R2 → R3 → R4 → D)의 경로를 구한 후에 점선화살표(S → R1 → R5 → R6 → R3 → R7 → R8 → D)의 경로를 구한다. 이후 데이터 패킷을 전송을 할 때마다 둘 중 하나의 경로를 무작위로 선택을 해서 전송을 한다.



[그림 3] 임의 경로 변화(Random Route Mutation) 예[1]

IV. S-RAY

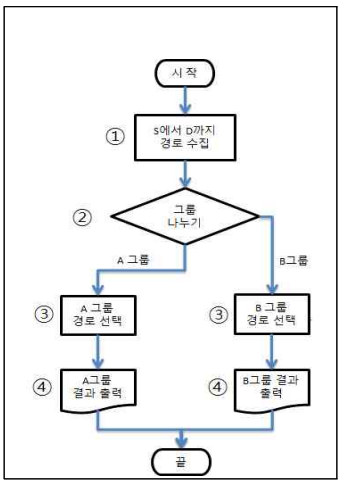
본 장에서는 Yen 알고리즘을 활용한 보안 알고리즘(Secure Routing Algorithm based on Yen's algorithm, S-RAY)의 전체적인 구조와 제안 알고리즘 Ⅰ, Ⅱ, Ⅲ을 설명하겠다.

4.1 디자인

Yen 알고리즘을 활용하여 사용자가 지정한 출발점과 도착점 사이의 필요한 만큼의 경로를 얻은 후 이를 이용하여 그룹별 필요한 경로들을 무작위로 선정하여 사용한다.

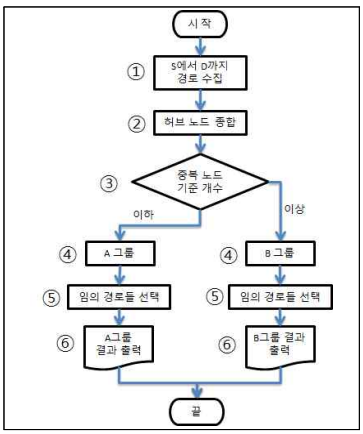
4.1 제안 알고리즘

[그림 4]은 제안 알고리즘 Ⅰ이다. ①에서 출발점에서 도착점까지의 필요한 경로를 구한 후 ②에서 경로를 몇 개의 그룹으로 만들지 판단 후에 ③에서 그룹별 필요 경로를 무작위로 선택 후 ④에서 경로를 얻어 사용하면 된다.



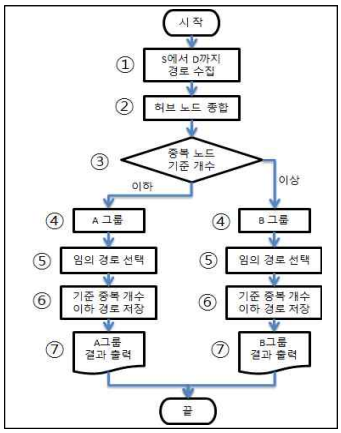
[그림 4] 알고리즘 Ⅰ

[그림 5]는 알고리즘 Ⅱ이다. 알고리즘 Ⅰ과 다른 점은 ②에서 사용자가 필요로 하는 만큼의 경로가 모두 지나는 허브 노드를 구한 후에 이를 기준으로 ③에서 다익스트라 알고리즘(Dijkstra algorithm)으로 구한 경로와 중복이 허용되는 개수를 정한 후 ④에서 그룹별 경로를 저장하여 ⑤에서 임의의 경로들을 선택하여 ⑥에서 경로를 얻어서 사용을 한다.



[그림 5] 알고리즘 Ⅱ

[그림 6]은 알고리즘 Ⅲ이다. 알고리즘 Ⅱ와 다른 점은 ⑤에서 각 그룹별 기준이 되는 경로를 선택 후에 ⑥에서 기준 경로와 중복이 몇 개까지 가능한지를 정한 후 이하인 경로들을 ⑦에서 얻어 사용 한다.



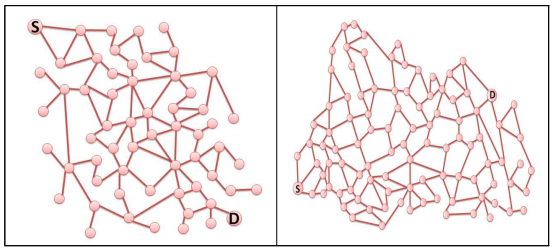
[그림 6] 알고리즘 Ⅲ

Ⅴ. 평가

본 장에서는 일반 라우팅과 4장에서 언급한 알고리즘을 실행 하였으며, 실험 환경은 인텔 i7 CPU와 우분투 14 버전에서 하였다.

5.1 네트워크 모형

[그림 7]은 실험에서 사용 된 60노드와 100노드 네트워크 모형이며 출발점은 S, 도착점은 D로 표시 하였다.



[그림 7] 60노드(좌) 100노드(우) 네트워크

5.2 일반 라우팅

60 노드와 100 노드에서 최단 경로 우선 프로토콜(Open Shortest Path First, OSPF) 방식으로 라우팅 경로를 찾을 시 소요 시간은 60 노드가 0.80초 100 노드가 5.35초 이다.

5.3 Yen 알고리즘 라우팅

Yen 알고리즘을 활용하여 출발점부터 도착

점 까지의 경로 수에 따른 소요 시간은 [표 1]과 같다. 경로 수에 따라 60노드는 약 {0.80 + (경로 수 X 0.008)초가 걸리고 100노드는 약 {5.35 + (경로 수 X 0.016)}초가 걸리는 것을 알 수 있다.

[표 1] 경로 수에 따른 소요 시간 단위 : 초

60 노드						
경로 수	10	50	100	150	200	228
시간	0.87	1.16	1.5	1.85	2.3	2.68
100 노드						
경로 수	10	40	80	120	160	192
시간	5.56	6.01	6.51	6.87	7.32	8.09

5.4 S-RAY

4장에서 언급한 S-RAY 제안 알고리즘으로 구한 소요시간은 [표 3]과 같으며 경로가 그룹별 10개 이상 선택 되는 것을 기준으로 하였다. 단, 제안 알고리즘 Ⅱ, Ⅲ에서 사용되는 그룹별 경로 수는 [표 2]와 같다.

[표 2] 그룹 별 경로 수 단위 : 개

항 목	A그룹	B그룹
60노드	51	177
100노드	53	139

[표 3] 제안 알고리즘 별 소요 시간 단위 : 초

알고리즘	Ⅰ	Ⅱ	Ⅲ
60 노드	2.80	4.29	4.19
100 노드	8.04	8.78	8.25

제안 알고리즘 Ⅰ, Ⅱ, Ⅲ에서 걸리는 시간은 Yen 알고리즘을 이용해서 최대 경로를 구할 때의 시간과 비슷한 것으로 미뤄봐 중요한 것은 처음 알고리즘에 사용 할 경로 수를 어떻게 결정 하느냐에 따라 소요 시간이 정해지는 것을 알 수 있다. 또한, 경로수를 줄이면 보안성은 낮아지는 대신 소요 시간은 줄어드는 것을 알 수 있다.

VI. 결론

인터넷 사용이 더욱 증가 할 것으로 예상 됨에 따라 네트워크의 중요성도 더욱 중요해 질 것이다. 따라서, 네트워크가 공격자들로부터 피해를 입지 않도록 보안성을 강화 시키는 것은 매우 중요하다.

본 논문에서는 유선 네트워크에서의 보안성을 향상 시킬 수 있는 S-RAY를 제안 하였으며 이는 유선 뿐만이 아니라 무선에서도 사용이 가능 하겠다. 그리고 특히 정부와 군과 같은 보안성이 중요한 인트라넷에 사용에 적합하며 현재의 네트워크에서도 사용을 할 수가 있으며, 차후 네트워크라 불리는 소프트웨어 정의 네트워크(Software Defined Network, SDN)에서도 추가 비용이 없이 적용을 할 수가 있다.

[참고문헌]

[1] Qi Duan, Ehab Al-Shaer and Haadi Jafarian, “Efficient Random Route Mutation Considering Flow and Network Constraints”, Department of Software and Information Systems, University of North Carolina at Charlotte, Charlotte, NC, USA, 2013 IEEE

[2] 정제성, 신승원, 김광조, “Yen의 알고리즘을 응용한 암의 경로 선택 방식”, 카이스트 정보보호대학원, 전산학부, 2015년 충청지부 정보보호 학술대회

[3] E. W. Dijkstra: A note on two problems in connexion with graphs. In: Numerische Mathematik. 1 (1959), S. 269 - 271

[4] DIJKSTRA’S ALGORITHM By Laksman Veeravagu and Luis Barrera, http://www.cs.utexas.edu/~EWD/

[5] E.Yen, Jin Y. (1970). "An algorithm for finding shortest routes from all source nodes to a given destination in general networks". Quarterly of Applied Mathematics 27: 526 - 530.

[6] Yen’s algorithm, https://en.wikipedia.org/wiki/Yen%27s_algorithm