계층적 오버레이를 이용한 DDoS 공격 감내 네트워크

김 미 희[†] · 채 기 준[††]

요 약

가장 위협적인 공격의 한 형태인 DDoS(Distributed Denial of Service) 공격은 다수의 공격 에이전트가 한꺼번에 많은 공격 트래픽을 특정 네트워크 또는 중요한 노드를 공격하는 특성을 갖고 있어 이로 인한 피해 지역 및 정도가 크다는 문제점이 있다. 이에 대한 기존의 많은 연구들은 탐지, 필터링, 추적 등에 집중되어 있고, 특히 피해 네트워크가 계층적인 구조를 갖고 있는 경우 특정 노드의 마비로 인해 하위 노드들의 정상 트래픽 전송이 어려워질 뿐 아니라, 탐지에 대한 다른 노드에 공지 및 추적을 위한 제어 트래픽 전송 또한 어려워 질 수 있다. 이에 본 논문에서는 계층적인 네트워크에서 이에 맞는 계층적인 오버레이를 구성하여, 공격 탐지 시 공지 및 추적을 위한 제어 트래픽을 오버레이를 이용해 전달하며, 공격 에이전트를 완전히 제거하기 전까지 정상적인 트래픽을 우회할 수 있는 DDoS 공격 감내 네트워크 구조를 제안한다. 또한 제안된 방법에서 오버레이 구성에 따른 오버헤드 분석과 공격 탐지 시 빠른 공격 차단 전달의 가능성과 신속성 및 정상 트래픽의 전송의 정도를 시뮬레이션을 통해 분석한다.

키워드 : 계층적 네트워크, DDoS 공격, DDoS 공격 감내 네트워크 구조, 계층적인 오버레이

DDoS Attack Tolerant Network using Hierarchical Overlay

Mihui Kim[†] · Kijoon Chae[††]

ABSTRACT

As one of the most threatening attacks, DDoS attack makes distributed multiple agents consume some critical resources at the target within the short time, thus the extent and scope of damage is serious. Against the problems, the existing defenses focus on detection, traceback (identification), and filtering. Especially, in the hierarchical networks, the traffic congestion of a specific node could incur the normal traffic congestion of overall lower nodes, and also block the control traffic for notifying the attack detection and identifying the attack agents. In this paper, we introduce a DDoS attack tolerant network structure using a hierarchical overlay for hierarchical networks, which can convey the control traffic for defense such as the notification for attack detection and identification, and detour the normal traffic before getting rid of attack agents. Lastly, we analyze the overhead of overlay construction, the possibility of speedy detection notification, and the extent of normal traffic transmission in the attack case through simulation.

Key Words : Hierarchical network, DDoS attacks, DDoS attack tolerant network structure, Hierarchical overlay

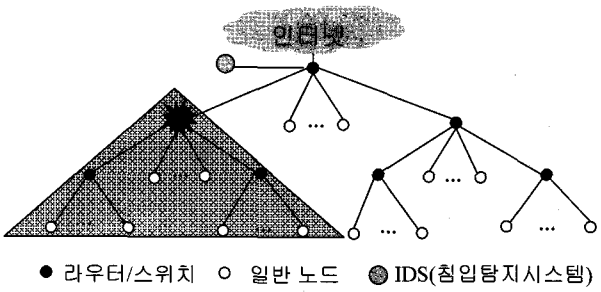
1. 서 론

DDoS 공격은 다수의 공격 에이전트를 분산 설치해 두고 동시에 공격함으로써 하나의 시스템 자원뿐 아니라 네트워크 자원까지도 고갈시킬 수 있는 간단하면서도 매우 강력한 공격이다. 실제로 웜 바이러스와 함께 DDoS 공격으로 인한 대량의 이상 트래픽으로 인해 인터넷 사용에 있어서 연결 실패나 속도 저하 등의 문제를 일으키는 사례가 증가하고 있으며, 이로 인한 피해는 점점 더 심각해지고 있는 실정이다.

특히 많은 근거리 통신 네트워크(LAN)가 그림 1과 같이 트리와 같은 계층적인 네트워크 구조를 띄고 있는데, 이러한 경우 특정 라우터가 공격에 의해 마비되면 그 하위 네트워크 또한 인터넷으로의 연결을 잃게 되어 통신이 두절될 수 있어 그 피해 지역은 더욱 커질 수 있다.

이처럼 DDoS 공격에 대한 대응책의 필요성이 대두되면서, 최근 DoS(Denial of Service) 공격 혹은 DDoS 공격에 대한 다각적인 연구들이 진행되어 왔다. 그러나 현재까지 제안된 보안 메커니즘들은 이에 대한 탐지, 공격 트래픽의 필터링, 그리고 공격 에이전트의 추적 등에 집중되어 있고, 특히 그림 1과 같은 계층적인 구조에서는 분산 공격에 대한 협동 대응을 위한 제어 트래픽 전송도 원활하지 못할 수 있다는 문제가 있다. 또한 그러한 네트워크 구조에서는 필터링 기능을 수행한다고 하더라도 일반적인 라우팅으로는 완전히 공격 에이전트를 제거하기 전까지 공격 트래픽에 의해 다른

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA 2006 C1090 0603 0028)
† 정 회 원 : 이화여자대학교 컴퓨터학과 박사
†† 종신회원 : 이화여자대학교 컴퓨터학과 교수
논문접수 : 2006년 10월 25일, 심사완료 : 2006년 12월 28일



(그림 1) 계층적인 네트워크 구성도

노드들의 트래픽 전송에 영향을 줄 수 있기 때문에 전체적인 피해가 커지게 된다.

본 논문에서 고려하는 DDoS 공격은 특정 서버를 공격해 서버의 서비스를 마비시키고 합법적인 다른 사용자의 서비스를 거부시키는 협의(narrow sense)의 DDoS 공격이라기 보다 대량의 많은 이상 트래픽 발생으로 특정 라우터 혹은 특정 네트워크를 마비시켜 그 피해를 증가시키는 광의(wide sense)의 DDoS 공격이다. 이러한 공격에 대해 빠른 대응을 가능하게 하고 피해를 최소화하고자 하는 목적을 갖고 있다.

이에 본 논문에서는 계층적인 네트워크 구조에 맞춰 계층적인 오버레이 네트워크를 형성함으로써 DDoS 공격 발생 시 이에 대응하기 위한 제어 트래픽을 피해 지역을 우회하여 신속하게 전송하여 빠른 대응을 가능하게 할 뿐만 아니라, 공격을 완전히 처리하기 전까지 정상 트래픽을 우회시켜 전체적인 피해를 최소화하는 DDoS 공격 감내 네트워크 구조를 제안하고자 한다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구로서 DDoS 공격에 대한 기존의 대표적인 연구와 문제점, 그리고 논문에서 사용하는 Chord 오버레이 네트워크[1]에 대해 간단히 설명하고자 한다. 3장에서는 본 논문에서 제안한 오버레이를 이용한 감내 네트워크에 대해 설명하고, 4장에서는 3장에서 기술한 감내 구조를 다양한 측면에서 분석, 실험한 내용과 그 결과를 기술하고자 한다. 마지막으로 5장에서는 결론으로써 본 논문을 마치고자 한다.

2. 관련 연구

2.1 기존 공격 차단 방법 및 문제점

DDoS 공격의 피해를 줄이고자 제안된 기존 방법들은 크게 세가지 분류로 나누어 볼 수 있다. 선대응에 해당하는 탐지방법, 공격 트래픽의 필터링 방법, 후대응 방법인 공격자에 대한 추적 방법이 그것이다.

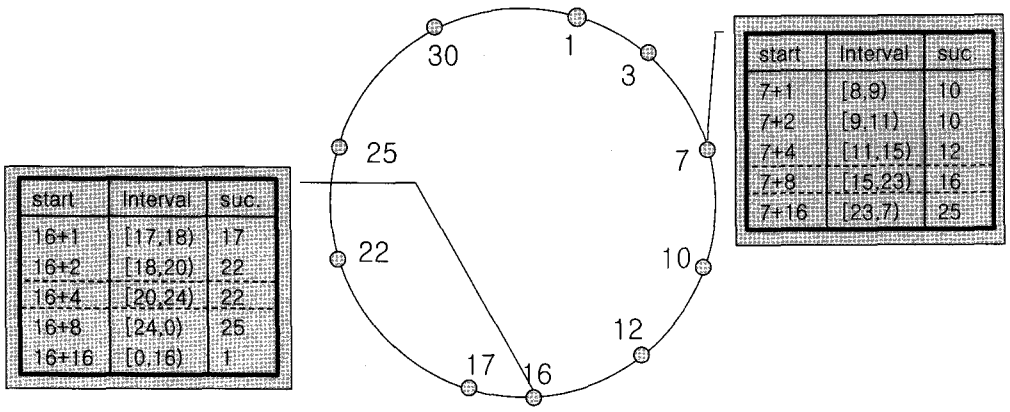
대표적인 공격 탐지 기법으로 통계적 방법을 이용한 연구에서는 공격 도구에 의해 생성된 공격 트래픽들은 정상 트래픽과 구별되는 특징을 갖고 있으며, 통계적인 기준을 이용하여 중심 라우터에서 정상과 공격 트래픽을 구별할 수 있다고 가정하고 통계적인 방법에 의한 공격 탐지 방법을 제안하였다[2]. 또한 tcpdump 네트워크 트래픽을 이용해 데이터 마이닝 기법에 의한 DDoS 공격 탐지 방법도 제안되었다[3]. 그러나 이들 방법에서는 통계적 기준에 의해 트래픽을 모니터링 하는 라우터와 데이터 마이닝에 의해 모델링된 탐지 모델을 가지고 탐지를 수행하는 중심 라우터가 공격에 의해 일부 혹은 전체 기능이 마비될 수 있고, 넓은 지역의 다수 라우터들간의 협동 대응이 필요한 경우 일반적인 라우팅으로는 공격 트래픽에 의해 제어 트래픽 전달 성공률이 떨어질 수 있다는 문제점이 있다.

또한 이러한 DDoS 공격이 탐지되면, 공격 근원지를 찾아 그에 대응하기 위하여 다양한 역추적 기법이 제안되었다[4]. 오늘날 가장 기본적인 역추적 기법으로서 사용되는 홉 바이 홉 IP 역추적 기법은 많은 연속적인 공격 패킷이 전송될 때 홉 바이 홉으로 공격 패킷의 근원지 라우터를 추적하는 방법이다. 그러나 이 방법은 모든 네트워크 도메인의 협동이 제공되지 않으면 수행 불가능하며, 공격 패킷에 의해 근원지를 찾아가는 제어 트래픽 전송이 어려울 수 있다는 문제점이 역시 존재한다.

마지막으로 대부분의 DDoS 공격이 IP 패킷의 소스 주소를 스푸핑하기 때문에 이러한 스푸핑된 패킷을 라우터에서 필터링하기 위한 다양한 방법이 제안되었다. 대표적인 방법으로 네트워크 입출력 단에서 유효하지 않은 소스 주소를 갖는 패킷을 필터링하는 입력단/출력단 필터링(Ingress/Egress Filtering)[5,6], 라우터에서 라우팅 테이블을 기반으로 잘못된 인터페이스로 수신되는 패킷을 필터링하는 패킷 기반의 필터링 기법[7] 등이 제안되었다.

위협적인 DDoS 공격에 대응하기 위해 기존에 제안되었던 탐지, 역추적, 필터링 기법들은 중요한 요소이다. 그러나 DDoS 공격 특성상 이들 기능이 수행되는 시스템들 간의 협동 대응은 필수적이며 이러한 협동 대응에 필요한 제어 트래픽은 공격이 수행되는 중에도 원활히 전달될 수 있어야 하고, 완전히 공격의 근원을 차단하여 해결하기 전까지 정상적인 트래픽의 전송을 보장할 수 있어야 한다. 이에 본 논문에서는 이러한 요소를 해결해 줄 수 있는 DDoS 공격 감내 구조를 제안하고자 한다.

본 논문에서 제안하는 구조는 오버레이 네트워크를 이용한 구조이다. 오버레이 네트워크는 P2P(Peer to Peer) 네트워크에서 데이터 공유를 위한 네트워크 구조[1,8]로 제안되었고, 이후에 멀티캐스팅을 위해 혹은 공격 차단을 위한 구조[9,10,11]로도 사용되었다. 오버레이를 이용해 DDoS 공격 차단에 관련된 연구로서 특정 서버에 대한 DDoS 공격 피해를 완화하는 목적으로 오버레이 라우팅을 이용해 정상 서비스 트래픽 포워딩의 랜덤화와 익명성을 높인 SOS 구조[10], 계층적인 서비스 구조를 갖고 있는 DNS(Domain Name System), LDAP(Lightweight Directory Access Protocol), PKI(Public Key Infrastructure) 등의 서비스에 대해 DoS 공격에 대한 저항성을 높이기 위한 계층적인 오버레이를 이용한 서비스 트래픽 우회 방법이 있다[11]. 전자는 특정 서버에 대한 DDoS 공격 보호의 목적을, 후자는 일반 트래픽이 아닌 특정 서비스 구조를 갖고 있는 서버들 간의 트래픽에 대한 DDoS 공



(그림 2) Chord 기반 오버레이 라우팅의 예 (m=5)

격 대응이라는 점에서 본 논문에서 추구하는 목적인 DDoS 공격에 대한 일반 정상 트래픽과 제어 트래픽 전송 보호라는 점이 상이하다.

2.2 Chord 오버레이 네트워크

Chord 프로토콜은 P2P 네트워크에서 데이터가 저장된 노드를 찾기 위한 대표적인 분산 라우팅 프로토콜로 일부 노드에 대한 정보, 즉 N 개의 노드로 구성된 네트워크에서 각 노드는 단지 O(logN)의 다른 노드 정보만 저장하고 있으면 되며, 목적지에 도달하기 위해 O(logN)의 메시지 전송이 요구되고, 노드 조인/탈퇴 시 $O(\log^2 N)$ 메시지를 전송하면 모든 노드의 라우팅이 재구성 되는 특징을 갖고 있다. 이 프로토콜에 의해 기존 Consistent Hashing 방법의 범위성(Scalability) 문제를 해결하였고, 이 논문은 오버레이 네트워크 구조를 이용한 다양한 분야에서 1000건 이상 논문의 참고 논문으로 사용된 대표적 논문이다.

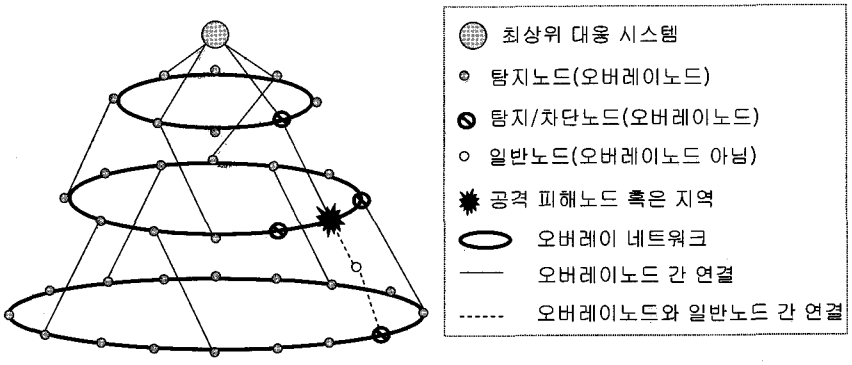
Chord 라우팅에 대해 조금 더 자세히 설명하면, 각 노드는 $[0, 2^m-1]$ 범위에서 해쉬 함수에 의해 노드 아이디를 할당 받고, 각 노드들은 개념적으로 노드 아이디에 의해 그림 2에서처럼 한 사이클을 구성하고 있다. 오버레이 네트워크에 조인된 각 노드 x는 핑커 테이블이라는 m개의 다른 노드에 대한 라우팅 정보를 저장하고 있고, 각 라우팅의 i번째 정보는 노드 아이디 $x+2^{i-1}(\mathrm{mod}\ 2^m)$ 대한 가장 가까운 다음 오버레이노드에 대한 정보이다. 예를 들어 그림 2의 7번 오버레이노드의 핑거 테이블을 설명하면, Interval [8,9) 즉 8번 노드를 책임지는 노드는 successor 노드인 10번 노드가 되고, Interval [9,11) 즉 9,10번 노드를 책임지는 노드도 역시 successor 노드인 10번 노드이므로 이 Interval에 포함되는 노드로의 패킷은 10번 노드에게 전송된다. 패킷 라우팅 방법은 노드 x가 노드 y로 향하는 패킷을 수신하면, y가 포함되는 Interval 값을 갖는 라우팅 엔트리의 후위자(successor) 노드로 전송하면 된다. 후위자 노드란 그 패킷을 책임지고 있는 대표 노드를 말한다. 예를 들어 그림 2에서 노드 7이 20으로 향하는 패킷을 수신했을 때 20은 Interval [15,23)에 포함되므로 노드 16으로 전송하고, 이를 수신한 노드 16에서는 패킷 20으로 향하는 패킷은 Interval [20,24)에 포함되므로 노드 22에게 전송한다. 노드 22는 20이 자신의 ID 번호보다 작으므로 자신이 이 패킷을 책임져야 할 목적지임을 알게 된다.

3. 제안하는 DDoS 공격 감내 구조

제안하는 DDoS 공격 차단 구조는 공격 탐지 후 공격 피해 지역을 파악하여, 공격 피해 지역으로의 또는 공격 피해 지역으로부터의 트래픽을 차단하며, 다른 노드의 정상 트래픽을 우회시켜 신속히 정상 작동시키고자 그림 3의 예와 같은 계층적 오버레이 구조를 사용한다. 그림 3은 3개의 오버레이 네트워크가 오버레이 멤버간 연결에 의해 연결되어 있으며, 최상위 대응 시스템은 가장 높은 오버레이 네트워크에 연결되어 있는 계층적인 오버레이 구성의 예이다. 최상위 대응 시스템의 위치 및 개수는 변할 수 있으며, 본 논문의 차단 구조에서 탐지노드가 자신이 탐지 정보를 전달해야 할 대응 시스템에 대한 계층 정보 및 위치 정보 등을 갖고 있다면 예시와 다른 네트워크에서도 적용 가능하다. 또한 최상위 대응 시스템과 모든 오버레이노드 간의 통신은 안전하다고 가정한다. 이하 본 논문에서는 그림 3의 예시를 사용하여 감내 구조 및 대응 방안을 설명한다. 본 논문에서 제시하는 구조는 다음과 같은 장점을 제공하고 있다.

- 공격 탐지 후, 최상위 대응 시스템에 탐지 정보의 빠른 전달
- 공격 피해 지역의 판단 후, 차단노드를 선정하여 다른 노드의 정상 트래픽 및 제어 트래픽을 빠르게 우회시킴으로써 피해를 최소화
- 계층적인 오버레이 구조의 사용으로 관리 비용의 최소화 및 확장성(Scalability) 제공

이에 대해 1)계층 오버레이 구성, 2)공격 탐지 전달, 3)공격 피해노드 및 차단노드 결정, 4)트래픽 우회 공지로 나누어 설명하고자 한다.



(그림 3) 계층적인 오버레이 네트워크를 이용한 차단 구조와 노드 구성의 예

3.1. 계층 오버레이 구성

본 논문에서 가정하는 감내 구조는 다수의 Chord 오버레이 네트워크[1]를 이용하여 최상위 대응 시스템으로의 공지 및 정상 트래픽의 우회를 유도하여 공격의 피해를 최소화하는 구조이다. 이를 위해 각 노드의 처리능력(capacity) 및 연결 구조에 따라 적당한 계층의 오버레이 네트워크에 조인하여 계층적인 오버레이 네트워크를 구성한다.

노드가 오버레이에 조인하여 오버레이 네트워크의 멤버로 구성되기 위한 방법은 두 가지로 관리자의 설정에 의한 멤버 조인과 메시지에 의한 멤버 조인으로 수행될 수 있다. 오버레이 구성 초기에는 각 노드의 오버레이 계층 및 해당 오버레이 계층이 포함하는 처리능력 범위 등을 설정하여 멤버로 조인하는 관리자 설정에 의한 조인 방법을 사용한다. 이 방법은 관리자가 해당 정보를 설정한 후 멤버로 조인되는 방법은 Chord 방법을 사용한다. 또한 초기 오버레이 네트워크 구성 후, 추가 노드 배치 및 재배치에 의한 동적인 네트워크 환경에서는 다음에서 설명하는 조인 탐사 메시지 및 조인 응답 메시지에 의해서 멤버 조인이 동적으로 수행된다.

메시지에 의한 조인 과정은 새로이 조인하고자 하는 노드 n_{new}가 자신의 용량($Capacity_{new}$)을 담은 *조인 탐사 메시지(Join Inquiry Message, JIM)*를 주변 노드들에 전송한다. 주변 노드라함은 자신과 직접 연결된 상, 하위 노드들과 이더넷과 같은 매체 공유 네트워크에 연결된 경우, 브로드케스트로 전달될 수 있는 이웃 노드들을 칭한다. 이 메시지를 수신한 이미 오버레이 네트워크 노드 n_i들은 자신의 용량($Capacity_i$)과 소속된 오버레이 계층($Level_i$)을 담은 *조인 응답 메시지(Join Response Message, JRepM)*를 전송한다. 조인 탐사 메시지를 보낸 노드는 일정 시간을 기다렸다가 수신된 조인 응답 메시지를 참고하여 자신의 오버레이 계층을 결정하고, 자신과 동일한 오버레이에 소속된 노드에게 조인 요청 메시지를 전송한다. 만약 일정 시간 내에 조인 응답 메시지를 수신하지 못하면, 조인 탐사 메시지를 최상위 대응 시스템에게 직접 전송한다.

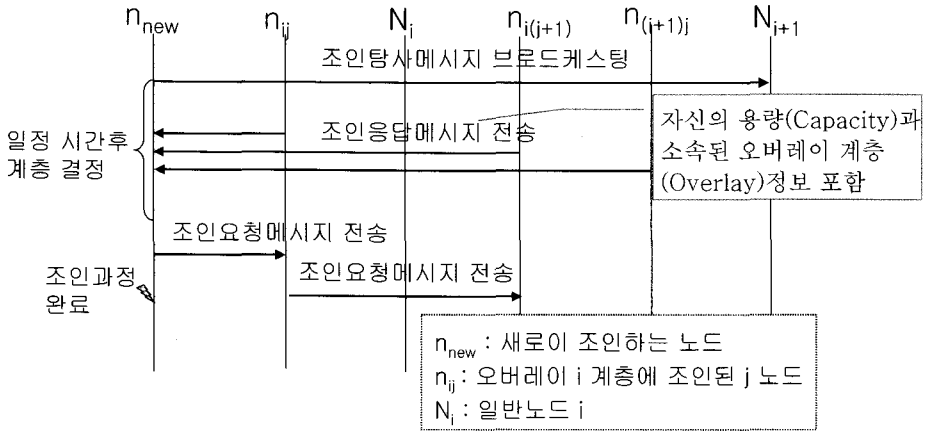
수신된 조인 응답 메시지의 용량에 따라 자신의 계층을 표 1과 같이 결정한다. 즉 수신된 조인 응답 메시지의 용량을 정렬시켜 자신의 용량이 포함되는 구간을 결정하고, 자신의 용량이 포함되는 구간이 존재하는 경우, 그 구간이 자신의 용량과 같거나 작은 용량의 값을 보낸 조인 메시지의 계층 값을 자신의 계층으로 결정한다. 또는 포함되는 구간이 없고 수신된 메시지의 모든 용량보다 자신의 용량이 작다면 최소의 용량 값을 보낸 조인 응답 메시지의 계층 값을, 수신된 메시지의 모든 용량보다 자신의 용량이 크다면 최대의 용량 값을 보낸 조인 응답 메시지의 계층 값을 자신의 계층 값으로 결정한다.

조인할 계층을 결정한 다음, 결정된 계층의 오버레이노드에게 직접 *조인 요청 메시지(Join Request Message, JReqM)*를 전송하여, 네트워크 조인 처리를 수행한다. 이때 수행되는 처리 과정은 Chord의 조인 처리를 그대로 따르되, 자신에 직접 연결된 상(high), 하위(low) 오버레이노드가 존재한다면, 조인 요청 메시지에 그 노드들의 연결 정보 및 계층

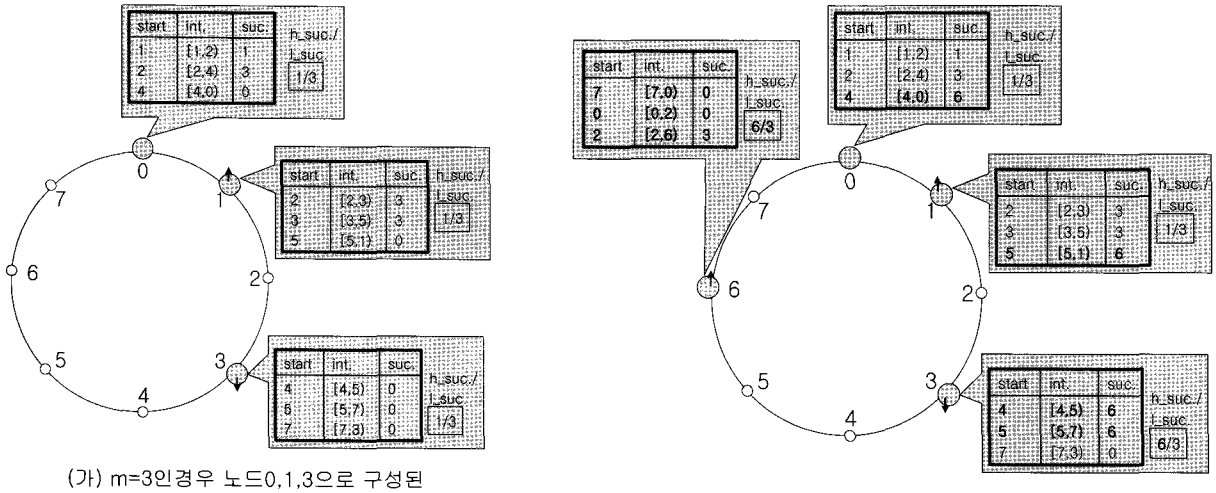
〈표 1〉 새로운 노드 조인 시 조인할 계층 결정 방법

if ($Capacity_i \le Capacity_{new} < Capacity_{(i+1)}$)	$Level_{new} = Level_i$
else if ($Capacity_1 \ge Capacity_{new}$)	$Level_{new} = Level_1$
else if ($Capacity_K \le Capacity_{new}$)	$Level_{new} = Level_K$

- n_i ($1 \le i \le K$, K : *JRepM*을 보낸 노드 개수)
- $Capacity_i$: 노드 n_i의 용량 (용량으로 오름차순 정렬하여 $Capacity_i \le Capacity_{(i+1)}$ 임)
- $Level_i$: 노드 n_i의 오버레이 계층
- $Capacity_{new}$: 새로 조인하는 노드의 용량
- $Level_{new}$: 새로 조인하는 노드의 결정된 레벨



(그림 4) 메시지에 의한 동적인 멤버 조인의 예

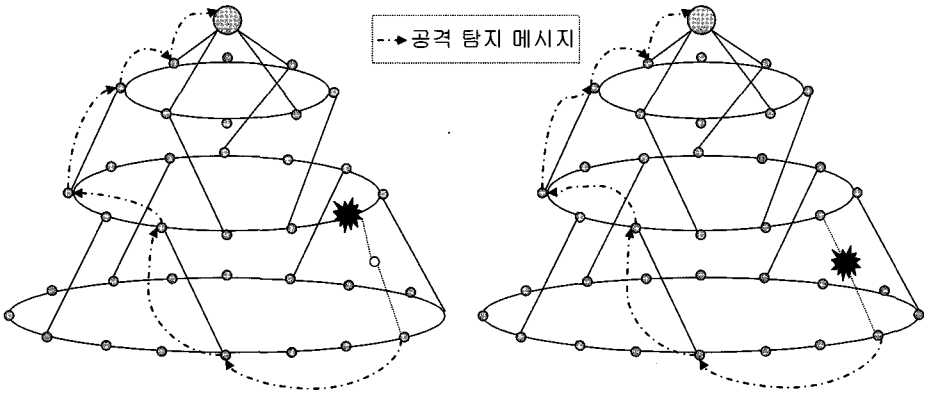


(그림 5) 노드 0,1,3을 포함하고 있는 오버레이 네트워크 예시와 노드 6이 조인한 경우 변화 예

정보(high_level, low_level)를 함께 전송한다. 이미 오버레이에 조인된 기존 노드들은 조인 과정 시 해당 부속 정보(상하위에 연결된 노드 정보)를 갱신하여, 추후 공격 탐지 전달에 사용하도록 한다. 각 노드가 갖고 있는 정보는 1)오버레이 네트워크에서 이전 노드(predecessor)에 대한 정보(오버레이 구성 시 사용됨)와 2)핑거 테이블이라는 라우팅 테이블과 3)상위 계층의 노드에 직접 연결이 존재하는 동일 계층의 다음 노드(high_successor), 4)하위 계층의 노드에 직접 연결이 존재하는 동일 계층의 다음 노드(low_successor)이다. high_successor/low_successor 노드는 자신을 포함하여 링을 시계방향으로 돌아갈 때 상위/하위 오버레이노드와 연결이 존재하는 첫 번째 노드이다. 그림 4는 이러한 메시지에 의한 동적인 멤버 조인 과정을 예시하였다.

Chord의 기본 조인 과정에 기반하여 변형된 조인 과정을 간략히 설명하면 다음과 같다. 조인 과정은 1)새로이 조인하고자 하는 노드 n_{new}의 이전 노드(predecessor)와 핑거 테이블 및 상/하위 계층에 연결된 다음 노드를 구성하고, 2)노드 n_{new}의 추가로 인해 영향을 받는 기존 노드들의 이전 노드와 핑거테이블, high_successor/low_successor를 조정한다. 이 때, 이전 노드와 핑거 테이블을 구성하는 방법은 Chord의 기법을 그대로 사용하고, 계층적인 오버레이 구성을 위해 추가적으로 관리되는 상/하위 계층의 노드에 직접 연결이 존재하는 동일 계층의 다음 노드를 구성하는 방법은 다음과 같다.

새로이 조인하고자 하는 노드 n_{new}에서 송신한 조인 요청 메시지에 포함된 상/하위 계층에 연결된 노드 정보(high_level/low_level)가 있는 경우, 기존 노드 n_{*x}(임의 계층의 x 노드)들은 자신이 갖고 있는 상/하위 다음 노드 정보(high_successor$_x$/low_successor$_x$)와 비교하여 자신을 기준으로 시계방향 순번에서 먼저 있는 경우, 즉 high_successor$_x$ > new > x 또는 low_successor$_x$ > new > x인 경우, 자신의 정보를 갱신한다.



(가) 희생자노드가 오버레이노드인 경우 (나) 희생자노드가 일반노드인 경우

(그림 6) 탐지노드가 희생자노드를 탐지한 경우, 최상위 대응 시스템에 공격 탐지 메시지를 전달하는 예시

그림 5(가)는 m이 3이고, 노드 0,1,3을 포함하고 있는 오버레이 네트워크에서 각 노드가 갖고 있는 정보의 예이고, 그림 5(나)는 노드 6이 조인한 경우 변화하는 각 노드의 정보의 예이다. 이 때, 노드 3의 high_successor가 갱신 되었다.

3.2. 공격 탐지 전달

본 논문에서의 공격 탐지 방법에 관해서는 기존에 제안되었던 분산적인 탐지 방법을 이용한다고 가정한다. 예를 들어 중간 노드들이 네트워크 구성정보를 이용해 패킷의 송수신 IP 주소를 검사하거나 송신자의 IP 주소와 MAC 주소의 변화율을 감시하는 방법[12], 데이터 마이닝 방식을 이용하여 정상적인 트래픽 패턴을 구성하고 이를 이용해 트래픽 패턴을 검사하는 방법[13] 등을 이용할 수 있다. 이외에도 네트워크의 중간 노드들에서 분산 공격 탐지를 수행하는 다른 방법을 활용할 수도 있다.

이들 방법에 의해 특정 노드 혹은 지역에서 공격 발생이 판단되면, 공격 탐지노드는 자신이 속한 오버레이 계층에서 상위 계층에 연결된 다음 노드(high_successor)에게 자신의 핑거 테이블을 이용하여 *공격 탐지 메시지(Attack Detection Message, ADM)*를 전달한다. 공격 탐지 메시지에는 1)탐지노드 정보(IP 주소, 오버레이 계층), 2)공격에 피해를 받은 희생자(victim)노드 정보(IP 주소, 있다면 오버레이 계층), 3) 탐지노드/희생자노드와의 관계(상/하, 하/상, 동일계층) 정보, 4)탐지노드의 주변노드 정보(상, 하 혹은 동일계층으로 연결된 노드) 등이 담겨 있다. 이를 전달 받은 high_successor 노드는 자신에 연결된 상위 계층 노드에게 수신한 메시지를 전달하고, 이를 수신한 노드도 마찬가지로 자신의 high_successor 노드에게 해당 메시지를 전달하게 된다. 이 과정을 반복하여 최상위 대응 시스템에게 해당 공격 탐지 메시지를 전달한다.

공격 탐지노드는 제안된 오버레이 구조에서 오버레이노드로 가정하고, 희생자(victim) 노드는 오버레이노드일 수도 아닐 수도 있다. 첫 번째 희생자노드가 오버레이노드인 경우, 탐지노드는 자신이 만든 공격 탐지 메시지를 상위 계층에 연결된 다음 노드(high_successor)로 공격 탐지 메시지를 전달하여 전 단락에 설명된 메커니즘 대로 계층 오버레이를 거쳐 최상위 대응 시스템에게 공격 탐지 메시지를 전달한다. 두 번째 희생자노드가 오버레이노드가 아닌 경우, 자신이 감지한 희생자노드의 정보(IP 주소)와 함께 공격 탐지노드의 정보를 담아 공격 탐지 메시지를 계층 오버레이를 거쳐 최상위 공격 차단노드에게 전송한다. 그림 6은 탐지노드가 희생자노드를 탐지한 경우 최상위 대응 시스템에 공격 탐지 메시지를 전달하는 경로를 예시하였다.

3.3. 공격 피해노드 및 차단노드 결정

최상위 대응 시스템은 하나 이상의 공격 탐지 메시지를 수신하면, 이 메시지에 담긴 탐지 정보를 근거로 희생자노드 및 공격 유형, 실제 공격 트래픽을 차단하고 정상 트래픽을 우회시킬 차단노드 집합을 선정한다. 희생자노드 또는 지역에 대한 정보를 수집한 최상위 대응 시스템은 희생자노드 및 지역을 판단한다(희생자노드 집합 = {x | 희생자노드 $x \in \forall ADM$ }). 이때, 희생자노드는 공격 트래픽 방향(상향/하향)에 상관없이 아예 라우터의 기능을 멈춘 경우(강한 공격)와 특정 공격 트래픽 방향에 의해 희생자노드의 특정 인터페이스만이 원활한 전송을 할 수 없는 경우(약한 공격)로 나누어 볼 수 있다. 표 2와 같이 전자의 경우, 탐지노드의 위치는 희생자노드를 중심으로 상위/하위/동일 계층일 것이며, 후자의 경우에는 상위/동일 계층 또는 하위/동일 계층에서의 탐지노드가 공격 탐지 메시지를 전달할 것이다. 이러한 탐지노드와 희생자노드와의 관계 분석에 의해 전자의 경우에는 희생자노드 혹은 지역을 전체적으로 우회할 수 있도록 상/하향 트래픽을 우회할 차단노드 집합을 결정하고, 후자의 경우에는 특정 방향(상향 혹은 하향)의 트래픽을 우회할 수 있도록 차단노드 집합을 결정한다.

공격 탐지 메시지들 정보에 의해 희생자노드 집합과 공격 유형이 결정되면, 최상위 대응 시스템은 표 3의 원칙으로 차단노드 집합을 결정한다

〈표 2〉 공격 탐지 메시지 정보에 의해 공격유형 결정 방법

```
if ((minD_layer ≥ maxV_layer ) or (DV_rel ⊆ {UD, S}))
    AttackType = Weak_Down
else if ((maxD_layer ≤ minV_layer ) or (DV_rel ⊆ {DU, S}))
    AttackType = Weak_Up
else
    AttackType = Strong
```

- $minD_{layer}$/$maxD_{layer}$: 탐지노드 집합의 오버레이 계층 최소값/최대값
- $minV_{layer}$ /$maxV_{layer}$: 희생자노드 집합의 오버레이 계층 최소값/최대값
- DV_{rel} : 탐지노드와 희생자노드의 연결 관계 집합 (UD: 상하, DU: 하상, S: 동일)
- AttackType : 공격유형 (Strong: 강한 공격,
  Weak_Down : 하향 공격 트래픽에 의한 약한 공격
  Weak_Up: 상향 공격 트래픽에 의한 약한 공격)

〈표 3〉 차단노드 집합 결정 원칙

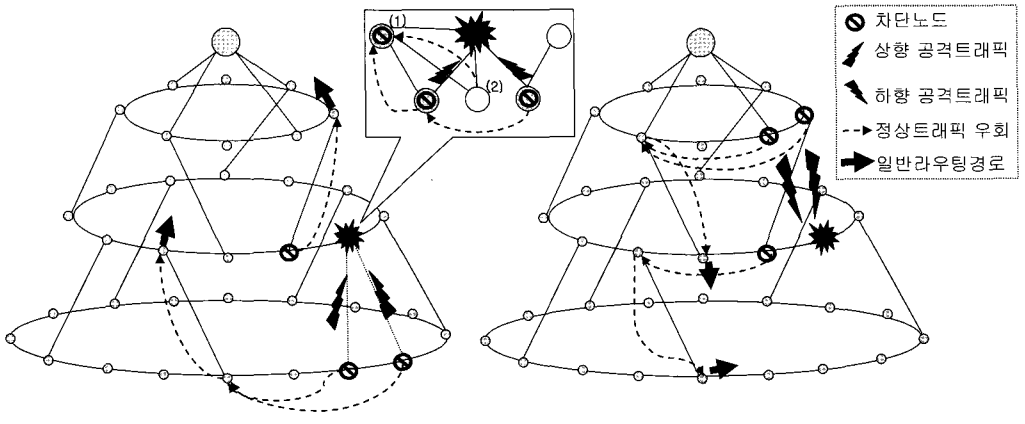
차단노드 집합 = { 탐지노드들, 공격 트래픽 방향으로 희생자노드에 연결된 노드들, 희생자노드와 같은 레벨의 노드들 }

3.4. 트래픽 우회 공지

차단노드 집합을 결정한 후, 최상위 대응 시스템은 계층적인 오버레이 네트워크를 통해 차단노드 구성 정보와 희생자노드 집합의 정보(IP 주소, 계층 정보 등)를 포함한 *차단 공지 메시지(Defense Notification Message, DNM)*를 차단노드 집합에게 전송한다. 이를 수신한 차단노드 집합은 정상 트래픽으로 판단되는 트래픽을 그림 7에서처럼 오버레이 네트워크를 이용하여 공격 피해 지역을 우회한다. 즉, 상향공격인 경우 피해 지역을 피해 오버레이의 한 계층 위로 패킷을 우회시킨 뒤, 그 뒤의 경로는 일반 라우팅을 이용하도록 하고, 마찬가지로 하향공격인 경우에도 피해 지역을 피해 오버레이의 한 계층 아래로 패킷을 우회시킨 뒤 그 뒤의 경로는 일반 라우팅을 이용하도록 한다. 특히 그림 7에서의 노드(1)처럼 희생자노드와 같은 레벨의 차단노드들은 자신에게 연결된 일반노드(노드(2))들의 트래픽을 자신을 통해 우회할 수 있도록 일반 라우팅 프로토콜을 사용하여 트래픽을 우회시킨다. 그림 7과 달리 공격유형이 Strong인 경우에도 비슷한 원칙대로 차단노드 집합을 결정하여 상/하향 정상 트래픽을 오버레이 네트워크를 통해 우회시킬 수 있다.

4. 성능 분석

본 장에서는 제안된 오버레이 네트워크를 이용한 차단 구조에 대해 오버헤드 분석과 공격 탐지 시 빠른 공격 탐지 전달의 가능성 및 신속성을 분석하고, 마지막으로 오버레이를 이용한 정상 트래픽 우회에 대한 전송률 분석을 통해 본 구조의 성능을 평가하고자 한다.



(가) 공격유형이 Weak_Up인 경우 (나) 공격유형이 Weak_Down인 경우

(그림 7) 공격 탐지 시, 차단노드 집합에 의한 정상 트래픽 우회 예시

4.1 오버헤드 분석

본 논문에서는 공격 탐지 시, 공격 차단에 필요한 제어 트래픽 우회를 통해 빠른 처리를 가능하게 해 주며 정상 트래픽 우회를 통해 DDoS 공격으로 인한 피해를 최소화 시키고자 오버레이 네트워크를 구성한다. 이 오버레이 네트워크는 노드의 이질성(heterogeneity) 및 많은 노드 수 지원(scalability)을 위하여 각 노드의 용량(capacity)에 따라 계층적인 오버레이 네트워크를 이용한다. 이 오버레이는 기존의 Chord 오버레이 기본 구성방법을 따르므로 전체 노드 수가 N이라고 했을 때, 멤버 조인 및 탈퇴에 $O(\log^2 N)$이 의 메시지 전송이 필요하며, 이는 실용적인 옵티마이징에 의해 $O(\log N)$으로 구성 오버헤드를 줄일 수 있다[1]. 본 논문에서는 계층적인 다른 노드 수에 의한 오버레이 네트워크 구조를 가정했으므로 그 구성 오버헤드는 표 4와 같은 관계식에 의해 하나의 오버레이로 구성하는 경우보다 더 작은 오버헤드를 요구함을 알 수 있고, 여러 개의 오버레이로 구성되는 경우에는 같은 수의 노드로 구성하는 것이 최소의 오버헤드를 요구함을 알 수 있다. 표 4에서 α_i는 β개의 각 오버레이의 노드 수이다. 또한 그림 8은 이러한 관계식에 의해 N의 증가에 따른 오버헤드 비교 그림으로 이는 N이 커질수록 계층적인 오버레이의 오버헤드와 하나의 오버레이로 구성될 때의 오버헤드 차이가 증가함을 알 수 있다. 이 그림에서 계층적인 오버레이는 3계층으로 구성되어 있고, 1000개의 노드로 구성된 계층적인 오버레이(1:10:100)인 경우에는 10개의 노드, 100개의 노드, 890개의 노드가 각각의 오버레이로 구성되었을 때의 오버헤드를 도시하였다.

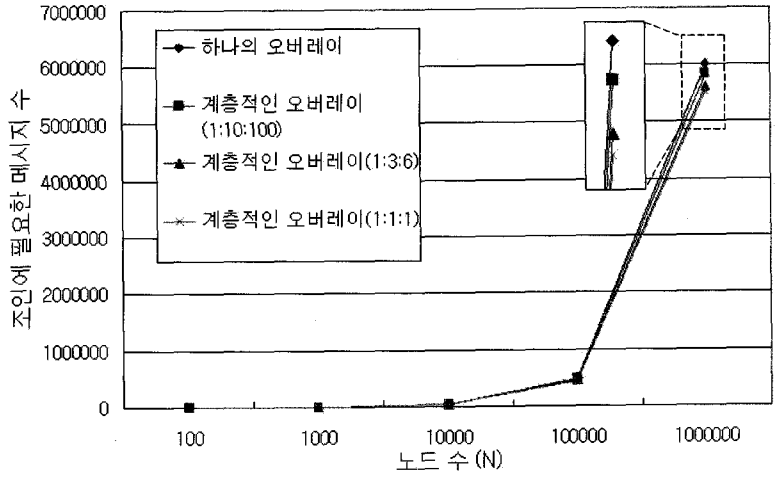
4.2 공격 탐지 전달의 신속성

공격 탐지 시, 제어 패킷의 신속한 전달 가능성을 시뮬레이션 하기 위해 유무선 네트워크 시스템에서 광범위한 네트워크 시뮬레이션을 가능하게 해 주는 Glomosim[14] 시뮬레이터를 사용하여 그림 9와 같은 네트워크를 구성하였다. 이는 3개의 계층으로 구성된 오버레이 네트워크를 구성하고 있고, 최상위 대응 시스템은 오버레이 계층1에 연결되어 있다. 그림 9에는 주 라우팅 경로는 실선으로 표시했으며, 한 오버레이 계층에 소속된 노드들 사이에는 간접적인 연결이 존재하고, 오버레이노드가 공격 탐지 시 공격 탐지 메시지를 1초 단위로 5개까지 최상위 대응 노드에게 전달한다고 가정한다. 표 5는 공격 시나리오 및 정상 트래픽에 대한 내용을 설명하고 있다.

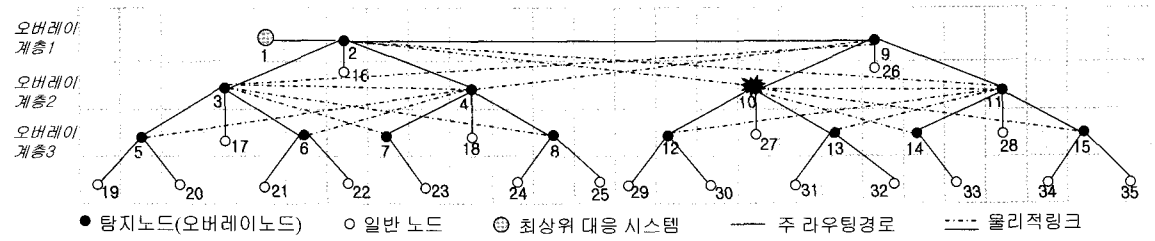
공격 트래픽, 정상 트래픽이 모두 노드 10의 라우터를 경유하고, 노드 10의 링크 용량이 트래픽에 비해 작다고 설정하여, 노드 10에서 트래픽 과부하로 인한 병목현상이 발생하고 이를 노드 12,13이 탐지하여 공격 탐지 메시지를 오버레이를 통해 최상위 대응 시스템에 전달하게 된다. 예를 들어 12번 노드는 high_successor 노드가 자신이 되므로, 자신에 연결된 상위 오버레이노드 11에게 전달이 되고, 이는 13번 노드도 마찬가지여서 해당 공격 탐지 메시지를 노드 11을 통해 노드 2, 결국 최상위 탐지 시스템인 노드 1에게 전달이 된다. 오버레이 감내 구조를 사용하지 않은 경우와 사용하는 경우에 대하여 공격 탐지 트래픽 전달시간 및 전송률의 결과를 비교하면 표 6과 같다. 실험 결과, 오버레이 감내 구조를 사용하지 않은 경우 공격 탐지 메시지가 희생자

〈표 4〉 하나의 오버레이 구성과 계층 오버레이 구성의 오버헤드 비교

하나의 오버레이 구성 ≥ β개의 서로 다른 수의 노드로 구성된 오버레이 구성 ≥ β개의 서로 같은 수의 노드로 구성된 오버레이 구성
$$N \cdot \log N \geq \sum_{i=1}^{\beta} \alpha_i \cdot \log \alpha_i \geq \beta \cdot N/\beta \cdot \log(N/\beta)$$



(그림 8) 하나의 오버레이 구성과 다수의 계층적인 오버레이 구성에서의 오버헤드 비교



(그림 9) 시뮬레이션을 위한 네트워크 구성도

〈표 5〉 공격 시나리오 및 정상 트래픽

공격 유형	트래픽
강한 공격 (strong)	- 5개 Attack Agent 공격트래픽(노드29→노드26), 1ms 주기, 공격시간 10~150s - 5개 Attack Agent 공격트래픽(노드31→노드26), 1ms 주기, 공격시간 10~150s - 정상 트래픽 노드30→노드16, 1ms 주기, 트래픽발생시간 0~150s 노드32→노드16, 1ms 주기, 트래픽발생시간 0~150s
약한 공격 (weak up)	- 1개 Attack Agent 공격트래픽(노드29→노드26), 1ms 주기, 공격시간 10~150s - 1개 Attack Agent 공격트래픽(노드31→노드26), 1ms 주기, 공격시간 10~150s - 정상 트래픽 : 노드30→노드16, 1ms 주기, 트래픽발생시간 0~150s 노드32→노드16, 1ms 주기, 트래픽발생시간 0~150s

〈표 6〉 공격 유형에 따른 공격 탐지 메시지 전송률과 전송시간

공격 유형	전송률		전송시간(Sec)	
	오버레이 감내 구조 없는 경우	오버레이 감내 구조 있는 경우	오버레이 감내 구조 없는 경우	오버레이 감내 구조 있는 경우
강한공격	20%	100%	4.139377	0.006121
약한공격	20%	100%	2.219606	0.006121

〈표 7〉 공격 유형에 따른 정상 트래픽 전송량(패킷수)

공격 유형	오버레이 감내 구조 없는 경우	오버레이 감내 구조 있는 경우
강한공격	5639	12364
약한공격	7771	12364

노드를 거쳐야 하므로 전송 손실 가능성을 보여 주었고, 오버레이 감내 구조를 사용하는 경우 공격 탐지 메시지 전달의 높은 가능성 및 속도 향상으로 인해 빠른 공격의 차단 가능성을 입증할 수 있었다.

4.3 정상 트래픽 전송량

DDoS 공격 시 정상 트래픽의 전송량을 비교한 결과는 표 7과 같다. 실험 결과 오버레이 차단 구조를 사용하는 경우 공격 강도에 큰 영향 없이 정상 트래픽 전송량이 제공되므로 공격 피해의 최소화를 가능하게 함을 알 수 있다.

5. 결 론

본 논문에서는 DDoS 공격 차단 과정에서 간과될 수 있는 기능인 협동을 위한 제어 트래픽의 빠르고 정상적인 전송 보장 및 공격 발생시 정상 트래픽의 피해 최소화를 위하여 계층적인 오버레이를 이용한 공격 감내 구조를 제안하였다. 감내 구조 구성하기 위해 오버레이 구성 시 대표적인 오버레이 네트워크 구성 방법인 Chord 방법을 사용하였고, 계층적인 네트워크 구조 및 다양한 처리능력의 노드가 존재함을 고려하여 다수의 계층적인 오버레이를 구성하여 구성 시 오

버헤드를 줄일 수 있었다. 제안된 구조의 성능을 평가하기 위해 Glomosim으로 네트워크를 구성하여 실험하였고, 실험 결과 빠른 탐지 메시지 전달의 가능성과 신속성 및 공격 강도에 큰 영향 없이 정상 트래픽의 전송량을 보장할 수 있음을 입증하였다.

참 고 문 헌

[1] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan, "Chord: A Scalable Peer to peer Lookup Service for Internet Applications," SIGCOMM'01, pp.149-160, Aug., 2001.

[2] Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, Darrell Kindred, "Statistical Approaches to DDoS Attack Detection and Response," DARPA Information Survivability Conference and Exposition, 2003.

[3] Wenke Lee, Salvatore J. Stolfo, "Data Mining Approaches for Intrusion Detection," Proc. of the 7th USENIX Security Symposium, pp.79-94, Jan., 1998.

[4] Howard F. Lipson, "Tracking and Tracing Cyber Attacks: Technical Challenges and Global Policy Issues," Special Report, CMU/SEI 2002 SR 009, Nov., 2002.

[5] P. Ferguson,D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," IETF RFC2827, May, 2000.

[6] Heather L. Flanagan, "Egress filtering - keeping the Internet safe from your systems," http://www.giac.org/practical/gsec/Heather_Flanagan_GSEC.pdf

[7] Kihong Park, Heego Lee, "On the Effectiveness of Route Based Packet Filtering for Distributed DoS Attack Prevention in Power Law Internets," ACM SGOMM, pp.15-26, 2001.

[8] M. Kaashoek, D. Karger, "Koorde: A Simple Degree Optimal Distributed Hash Table," Second Int'l Workshop Peer to Peer Systems, Feb., 2003.

[9] Zhan Zhang, Shigang Chen, Yibei Ling, and Randy Chow "Capacity Aware Multicast Algorithms on Heterogeneous Overlay Networks," IEEE Transactions on parallel and distributed systems, Vol.17, No.2, pp.135-147, Feb., 2006.

[10] Angelos Keromytis, Vishal Misra, Dan Rubenstein, "SOS: An Architecture for Mitigating DDoS Attacks," IEEE Journal on Selected Areas in Communications (JSAC), Vol.22, No.1, Jan., 2004.

[11] Hao Yang, Haiyun Luo, Yi Yang, Songwu Lu, Lixia Zhang, "HOURS: Achieving DoS Resilience in an Open Service Hierarchy," International Conference on Dependable Systems and Netowrks (DSN'04), pp.83-92, June, 2004.

[12] Mihui Kim, Kijoon Chae, "Detection and Identification Mechanism against Spoofed Traffic Using Distributed Agents," ICCSA2004, LNCS 3043, pp.673-682, May, 2004.

[13] Mihui Kim, Hyunjung Na, Kijoon Chae, Hyochan Bang, Jungchan Na, "A Combined Data Mining Approach for DDoS Attack Detection," ICOIN2004, LNCS 3090, pp.943-950, Feb., 2004

[14] GloMoSim (Global Mobile Information Systems Simulation Library), http://pcl.cs.ucla.edu/projects/glomosim/

김 미 희

e-mail : mihui@ewhain.net

1997년 이화여자대학교 전자계산학과(학사)

1999년 이화여자대학교 대학원 컴퓨터학과(석사)

1999년 (주)인티 연구원

1999년~2003년 한국전자통신연구원 연구원

2007년 이화여자대학교 컴퓨터학과(박사)

관심분야 : 네트워크 보안, NEMO(NEtwork MObility) 보안, 센서 네트워크 보안, 유비쿼터스 네트워크 보안

채 기 준

e-mail : kjchae@ewha.ac.kr

1982년 연세대학교 수학과(학사)

1984년 미국Syracuse University 컴퓨터학과(석사)

1990년 미국 North Carolina State University 컴퓨터공학과(박사)

1990년~1992년 미국 해군사관학교 컴퓨터학과 조교수

1992년~현재 이화여자대학교 컴퓨터학과 교수

관심분야 : 네트워크 보안, 인터넷/무선통신망/고속통신망 프로토콜 설계 및 성능분석, 센서 네트워크, 홈 네트워크, 유비쿼터스 컴퓨팅