Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Alvaro Garcia

Distributed Denial of Service Attack and OpenFlow:

Improving defense mechanism of DDoS with OpenFlow

Master's Thesis
Espoo, March 15, 2014

DRAFT! — February 9, 2014 — DRAFT!

Supervisors: Aapo Kalliola M.Sc. (Tech.)
Advisor: Aapo Kalliola M.Sc. (Tech.)

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Alvaro Garcia		
Title:	Distributed Denial of Service Attack and OpenFlow: Improving defense mechanism of DDoS with OpenFlow		
Date:	March 15, 2014	**Pages:**	16
Major:	Computer Science and Engineering	**Code:**	T
Supervisors:	Aapo Kalliola M.Sc. (Tech.)		
Advisor:	Aapo Kalliola M.Sc. (Tech.)		

Since DoS attacks are becoming more commons and emerging new technologies to separate the control plane and the data plane from the network devices (SDN), throughout this survey we will investigate how OpenFlow can help to prevent and locate these kinds of attacks. We will study the different DoS attacks and the current mitigation techniques. We will discuss as well, which of these techniques could be improved with OpenFlow and how develop them. In the end, we will implement and test some identified mitigation techniques and we will study their behaviour.

Keywords:	DoS, DDoS, SDN, OpenFlow
Language:	English

Acknowledgements

Espoo, March 15, 2014

Alvaro Garcia

Abbreviations and Acronyms

IP	Internet Protocol
SDN	Software Defined Networking

Contents

List of Figures

Chapter 1

Introduction

The original aim of the Internet was to provide an open and scalable network among research and educational communities, where billions of users are served through a global system of interconnected computer networks.

Unfortunately, with the rapid growth of the Internet over the last two decades, the number of attacks on the Internet has also increased rapidly. One of this attacks consists in disrupt the service provided by a network or server, either crashing the system sending some packets that exploit a software vulnerability or sending a large number of useless traffic to collapse the resources of the service. This kind of attack is known as Denial of Service (DoS) attack, or Distributed Denial of Service attack if it is launched by multiple hosts.

There are some design principles of the Internet that facility these kinds of attacks [5]:

Resource sharing: in IP networks, due to the packet-switched service, users share all the resources, and one user's service can be disturbed by other user's behaviour, so bandwidth attacks can disrupt services for legitimate users.

Simple Core and Complex Edge: One of the principles of the Internet is that the core network should be simple and pushes all the complexity into the end hosts. That means that the core of the networks is not able to integrate complex applications, such as authentication, security. Due to this simplification, when an attacker sends packets into the network and the victim receives them, it is almost impossible to recognize the real owner of the packets.

Fast Core Networks and Slow Edge Networks: The core networks needs to have high capacity due to the heavy traffic that has to support from many sources to many destinations. In contrast, an edge network needs less capacity because it only needs to support its end users. A disadvantage is

that traffic from high-capacity core can crush the slow-capacity edge.

Taking advantage of these principles and their vulnerabilities, have been arising a large number of different DoS and DDoS attacks and, as a result, a parallel growing of defense mechanisms to avoid these attacks. We might consider it like a constant battle between both sides, in which technological improvements are taking an important role on it.

In the current network architecture, the network devices (particularly routers) are bundle with a specialized control plane and various features. This vertical integration essentially binds you to whatever software and features are shipped with those particular devices. Software Defined Networking (SDN) effectively breaks these pieces apart.

SDN is a type of network architecture that separates the network data plane (network devices that forwarding traffic) from the control plane (software logic that controls ultimately how traffic is flowing through the network). OpenFlow [4] is a standard interface defined between the control and forwarding layers of an SDN structure.

One of the reasons to separate the control plane and the data plane is that the software control of the network can evolve independently of the hardware.

A second reason is it allows the network to be controlled from a single high-level software program. This software, used to control the network (in our case, POX), even though taking in count that works with a high-level programming language, has a lower layer abstraction and the difficulty for the Network Programmers is increased. Due to this inconvenient, it is time to speak about *Frenetic* [2].*Frenetic* is a Network Programming language which gives a high-level abstraction from POX, allowing them direct control over the network. *Pyretic (Python + Frenetic)* is one of the Frenetic family programming languages which provide a domain specific sub-language for specifying dataplane packet processing.

The aim of this survey is how might SDN help us to improve current DDoS defense mechanism. Throughout this project, we will review the main DDoS defense and attack mechanisms and further we will go through some algorithms already developed and then, we will explain how could be improve them with OpenFlow. We will test these algorithms on virtual scenarios-through Mininet.

This thesis is structure as follows: The next chapter we will explain the background related with this survey. We talk about the current situation of DDoS attacks and defenses and how OpenFlow works and its structure. In the chapter 3,

Chapter 2

Background

2.1 DDoS attack and defense mechanisms

A denial-of-service attack is characterized by an explicit attempt by attackers to prevent the legitimate users of a service from using that service [1] provided by a network or server. There are two manner to launch this kind of attack. The first approach is overwhelm the network and occupy all the resources of a service sending massive volumes of useless traffic

2.1.1 Protocol Attacks

2.1.2 Bandwidth Attacks

2.1.3 Logic Attacks

2.2 OpenFlow

The explosion of mobile devices, server virtualization, security problems and advent of cloud service are among the reasons because the networking industry is beginning to question the traditional network architecture. OpenFlow is intended to solve the problem of assigning resources to users in a easy-way giving them the control plane of the network without disturbing the traffic flows.

In traditional routers and switches, both control plane (high level routing decisions) and data plane (packet forwarding) are embedded in the same device. An OpenFlow Switch separates these two functions (Figure 2.1). The data plane function still resides on the switch, while the control plane is moved to a separate device called Controller (see 2.2.2) that manages the

switch and communicates to each other over the Secure Channel (see 2.2.1.2) via the OpenFlow protocol (see 2.2.1.3).

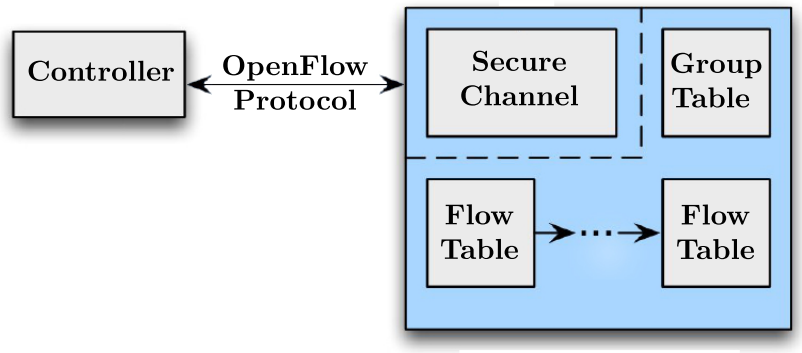


Figure 2.1: OpenFlow Switch components

The switch contains *flow tables* 2.2.1.1, which are updated through OpenFlow protocol adding, updating and deleting their *flow entries*. When the flow traffic arrives to the switch, it checks if the arrived packets match in the flow table, if so, the action defined in the flow entry is executed. Otherwise, the packet is either sent it to the Controller or dropped.

Throughout this section, we will explain in detail the main parts of the OpenFlow Switch, as well as the Controller and how they work together. The first version of the OpenFlow (1.1) protocol was released on 2011, one year later, in February 2012, the ONF approved and published the version 1.2. Nowadays, the current version of the protocol and the one that will be used in this project is the 1.4 [6].

2.2.1 Switch Components

2.2.1.1 Flow Table

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

Table 2.1: Main components of a flow entry

Ingress Port	Ether source	Ether dst	Ether type	VLAN id	VLAN priority	IP src	IP dst	IP proto	IP ToS bits	src port	dst port

Table 2.2: Main components of the Match Field

2.2.1.2 Secure Channel

2.2.1.3 OpenFLow Protocol

2.2.2 Controller

2.3 POX

Chapter 3

Theory

3.1 Protocol Attacks: TCP SYN Flooding

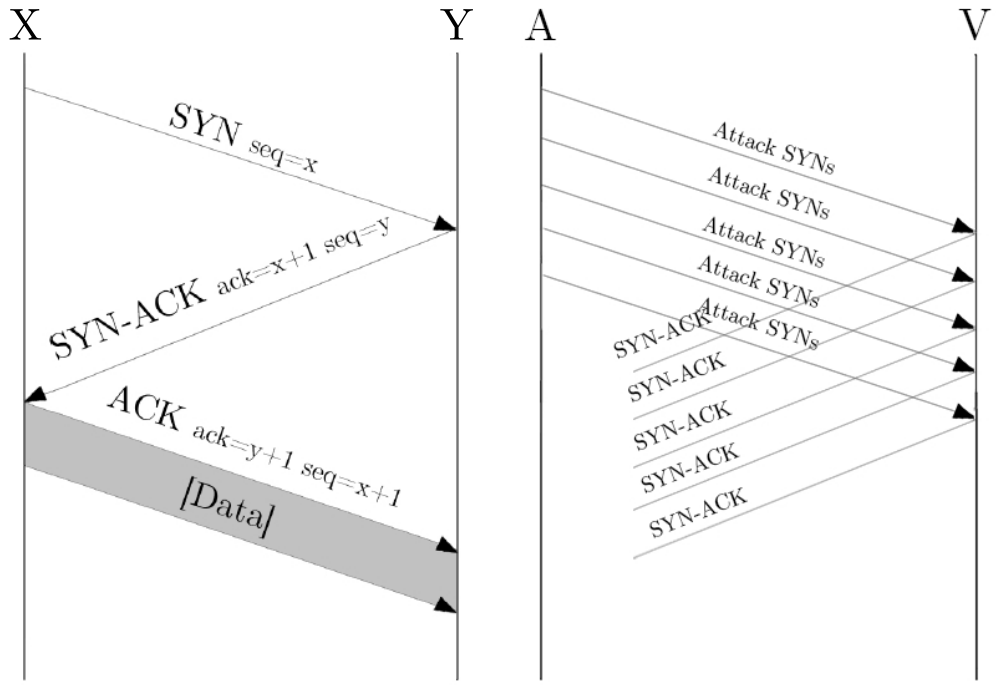
TCP (Transmission Control Protocol) is one of the most common protocols within the transport layer, and one of the core of the Internet Protocol suite (IP). TCP provides reliable, ordered, error-checked of stream of packets between two hosts. In addition to these characteristics, TCP is a connection-oriented protocol, that is, before starting the exchange of information a prior connection between both parties is necessary. This process is known as TCP three-way handshake (Figure3.1(a)).

Suppose X as a client that wants to carry out a friendly TCP connection with the server Y. First of all, X requests by sending a synchronize *(SYN)* message to Y. The server receives the request and responses by sending an acknowledge *(SYN-ACK)* back to the client. Once the client receives the SYN-ACK, it responses with an ACK, an the connection is established.

While the server waits the *SYN/ACK's* response, it keeps the connection in a half open state and maintains a backlog queue for the information about the connections. Once the server receives the *ACK*, it changes the state to *established* and frees up memory of the queue.

3.1.1 Methods of Attacks

The attack can be categorized depending on how the attacker carries out the attack over the victim: Direct Attack, Spoofed-based Attack and Distributed Attack [3].



(a) TCP Tree-way handshake (b) SYN Flooding

Figure 3.1: TCP Tree-way handshake and SYN Flooding

3.1.1.1 Direct Attack

3.1.1.2 Sppofed-based Attack

3.1.1.3 Distributed Attack

3.1.2 Prevention and Response

3.1.3 Tercer apartado

Bibliography

[1] CC., C. Denial of service attack. `http://www.cert.org/tech_tips/denial_of_service.html`.

[2] Frenetic. Frenetic official webpage. http://frenetic-lang.org/overview.php.

[3] Inc., C. Defenses Against TCP SYN Flooding Attacks. http://www.cisco.com/web/about/ac123/ac147/archived$_issues/ipj_9$ − 4/$syn_flooding_attacks.html$.

[4] OpenFlow. Open Flow standard. http://archive.openflow.org/.

[5] Peng, T., Leckie, C., and Ramamohanarao, K. Survey of network-based defense mechanisms countering the dos and ddos problems. *ACM Computing Surveys (CSUR) 39*, 1 (2007), 3.

[6] Specification, O. S. Version 1.4.0. *Open Networking Foundation* (2013).

Appendix A

First appendix

This is the first appendix. You could put some test images or verbose data in an appendix, if there is too much data to fit in the actual text nicely.

For now, the Aalto logo variants are shown in Figure A.1.

(a) In English

(b) Suomeksi

(c) På svenska

Figure A.1: Aalto logo variants