+

Sudan University of Science and Technology
College of Graduate Studies

Evaluation of Denial of Service Attacks in Neighbor Discovery Protocol over IPv6

تقويم هجمات حجب الخدمة في بروتوكول اكتشاف الجوار على الإصدار السادس من بروتوكول الإنترنت

A Thesis Submitted in Partial fulfillment for the Requirements of the Degree of M.Sc. in Electronics Engineering (Computer and Networks Engineering)

Prepared By:

Muna Mustafa Mohamed Rahmtalla

Supervised By:

Dr. Fath Elrahman Ismael Khalifa Ahmed

July 2021

Sudan University of Science & Technology

College of Graduate Studies

بسم الله الرحمن الرحيم

SUST

كلية الدراسات العليا

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

Ref: SUST/ CGS/A11

Approval Page

(To be completed after the college council approval)

Name of Candidate: Muna Mustafa Mohamed Rahmtalla

Thesis title: Evaluation of Denial of Service Attacks in Neighbor Discovery Protocol over IPV6

تقييم هجوم حجب الخدمة في برتكول إكتشاف الجوار

على الإصدار السادس من برتكول الإنترنت

Degree Examined for: Master of Science in Electronics Engineering (Computer and Networks)

Approved by:

1. External Examiner

Name: Elsadig Saeed Gebreel

Signature: .... Date: 1/8/2021

2. Internal Examiner

Name: Dr. Ahmed Abdalla Mohamed Ali

Signature: Dr. .... Date: 1/8/2021

3. Supervisor

Name: Dr. Fath Elrahman Ismael Khalifa

Signature: .... Date: 1/8/2021

ACKNOWLEDGMENT

First of all, all thanks, praise and sincere gratitude be to God who facilitates all steps towards success.

A special thanks to my parents for their endless love which gives me power to face all obstacles in life without fear.

To my Supervisor, Dr. Fath Elrahman Ismael, thank you for your huge efforts and continued support. His guidance helped me in all the time. Without his encouragement and guidance, this thesis would not have materialized.

I would like to thank the following people who have helped me undertake this research: my sibling, my friends and my colleague in the university and work.

Thanks for all without whom I would not have been able to complete this research, and without whom I would not have made it through my master's degree!

DEDICATION

I dedicate this work to my loving parents Nadia and Mustafa, to my sibling to my friends and to all who has a role to be who as I am now.

Thanks all for supporting me spiritually throughout my life
May you also be motivated and encouraged to reach your dreams

ABSTRACT

The Internet Protocol version 6 (IPv6) was developed and would gradually replace the Internet Protocol version 4 (IPv4). The Neighbor Discovery Protocol (NDP) one of the main protocols in the IPv6 suite providing many basic functions for the normal operations of IPv6 in a Local Area Network (LAN). However, NDP has several vulnerabilities that can be used by malicious nodes to launch attacks, because NDP assumes connections between nodes are safe. Hence, NDP messages are easily spoofed. In this research Denial of Service (DOS) attack is deployed by attacker computer in small virtual IPv6 network with two computers with different types of operating systems Windows and Linux as victims to evaluate the performance of the network before and during DoS attack using three network metrics throughput, delay and resources consumption. They are measured between monitoring computer and victim computers under flooding attack. Overall, the results had shown that both operating system Windows and Linux had been affected by the DOS attack. The performance of Linux was better than Windows in delivering low percentage of the sending packets.

التجريدة

تم تطوير الإصدار السادس من بروتوكول الإنترنت الذي سيحل محل الإصدار الرابع لبروتكول الإنترنت تدريجيا. بروتوكول اكتشاف الجوار هو أحد البروتوكولات الرئيسية في مجموعة بروتوكول الانترنت السادس، يقوم بتوفير العديد من الوظائف الأساسية لعمليات الإصدار السادس من بروتوكول الانترنت في الشبكة المحلية. بالرغم من ذلك يحتوي بروتوكول اكتشاف الجوار على العديد من نقاط الضعف التي يمكن استغلاها من قبل النقاط الخبيثة لشن الهجمات، وذلك لأن بروتوكول اكتشاف الجوار يفترض ان الاتصال بين النقاط امن، فبالتالي يمكن انتحال رسائل بروتوكول اكتشاف الجوار بسهولة. في هذا البحث سيتم تطبيق هجوم حجب الخدمة في شبكة افتراضية صغيرة لبروتوكول الانترنت الإصدار السادس تحتوي على نوعين مختلفين من أنظمة التشغيل لينكس و ويندوز كأجهزة سيتم الهجوم عليها لتقييم أداء الشبكة قبل وأثناء هجوم حجب الخدمة، تقييم الأداء سيكون بين جهاز المراقبة والأجهزة التي سيتم الهجوم عليها عن طريق ثلاثة معايير هم الإنتاجية والتأخير و استهلاك الموارد اثناء الهجوم. بشكل عام أظهرت النتائج ان نظامي التشغيل ويندوز و لينكس قد تأثروا بهجمات حجب الخدمة. نظام التشغيل لينكس كان أداءه أفضل من ويندوز خلال الهجوم.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AR	Address Resolution
ARP	Address Resolution Protocol
CERNET	China Education and Research Network
CPU	Central Processor Utilization
DAD	Duplicate Address Detection
DDoS	Distributed Denial of Service Attack
DoS	Denial of Service Attack
EUI-64	Extended Unique Identifier
IANA	Internet Control Message Protocol
IBM	International Business Machine
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol version 6
ID	Identification
IETF	lnternet Engineering Task Force
IPERF	Internet Performance Working Group
IPv6	Internet Protocol Version 6
IPv4	Internet Protocol Version 4
ISP	Internet Service Provider
IT	Information Technology
LAN	Local Area Network
MAC	Media Access Control
MBps	Mega Bit Per Second
MTU	Maximum Transfer Unit
NA	Neighbor Advertisement

NAT	Network Address Translation
ND	Neighbor Discovery
NDP	Neighbor Discovery Protocol
NDPSR	Represents the NDP Process Success Rate
NS	Neighbor Solicitation
NUD	Neighbor Unreachability Detection
OS	Operating System
RA	Router Advertisement
RD	Router Discovery
RS	Router Solicitation
RTT	Round Trip Time
SEND	Secure Neighbor Discovery Protocol
TCP	Transmission Control Protocol
THC-IPv6	The Hacker Choice-Internet Protocol Version 6
TV	Television
VM	Virtual Machine
VMSTAT	Virtual memory statistics
VoIP	Voice over Internet Protocol

CHAPTER ONE
INTRODUCTION

CHAPTER ONE

1. INTRODUCTION

This chapter presents an overview of Internet Protocol version 6 (IPv6) network, problem, proposed solution, objectives, methodology, and the contents of the research.

1.1 Preface

IPv6 is a protocol that was developed to succeed the Internet Protocol version 4 (IPv4) protocol. It aimed to solve the issues in today's Internet that IPv4 had to deal with, such as IP address space limitation and scalability and security. The Neighbor Discovery Protocol (NDP) is one of the major protocols found in the IPv6 suite. It is made up of IPv6 Stateless Address Auto configuration (SLAAC) and Neighbor Discovery for IPv6 [1].

NDP is known as the stateless protocol as it is utilized by the IPv6 nodes to determine joined hosts as well as routers in an IPv6 network without the need of dynamic host configuration protocol server[2]. IPv6 Neighbor Discovery (ND) is essentially a mechanism that determines how neighboring nodes relate to each other. It provides a method for a node in the local link to discover the link local address. With the use of NDP, nodes can discover other nodes on the link, determine their link-layer addresses to find routers, and maintain reachability information about the paths to active neighbors. ND was constituted as a replacement for the limited functionality of IPv4. It works along with IPv6 and replaces Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP) router discovery and the ICMP Redirect message used in IPv4 [3]. There are many critical functions of NDP including identifying physical addresses, detecting duplicate addresses, discovering nodes that are found within the same subnet, providing active neighbors with reachability information about paths, and discovering routers [4]. For the normal operation of IPv6, NDP also provides other functions including router/prefix/parameter discovery, address resolution, next-hop determination, Neighbor Unreachability Detection (NUD), Duplicate Address Detection (DAD) and redirection. All of these functions are based on the transmission of NDP messages, which are encapsulated in Internet Control Message Protocol Version 6 (ICMPv6) packets. NDP messages are confined to a link and only transmitted in the scope of a LAN. This means attached routers will not forward NDP messages from one network to another [5]. NDP is vulnerable to network attacks as it allows malicious nodes to impersonate other legitimate nodes or routers by forging ND protocol message [6] and it will be susceptible to different attacks that can be classified as spoofing, Replay, DoS, Redirect, or Rogue routing informationattack[1].

Denial of service (DoS) attack is one of the major security threats to the IPv4 and IPv6 networks. In DoS attacks, a victim host(s) can be denied from the services by wasting its resources and disrupt its communication with other neighboring hosts on same link. A targeted device is unable to process such large amount of network traffic and becomes unavailable or out of service. Moreover, when DoS attack is being attempted from large networks or systems then it is known as Distributed Denial of Service (DDoS) attacks [7].

NDP has several vulnerabilities that can be used by malicious nodes to launch attacks, because NDP messages are easily spoofed. Surrounding this problem many solutions have been proposed for securing NDP but these solutions either proposed new protocols that need to be supported by all nodes or built mechanisms that require the cooperation of all nodes [5].

1.2 Problems Statement

NDP one of the main protocols in IPv6 suit. The processes in NDP are done by exchanging NDP messages between nodes. NDP has several vulnerabilities that can be used by malicious nodes to launch attacks, because NDP assumes the connection between nodes is safe and NDP messages are easily to be spoofed. The network is seriously affected by DoS on NDP. However, regardless the kind of Operating System (OS) running in the network, the impact is happened but with different affecting.

1.3 Proposed Solution

The proposed solution is to evaluate the Denial of service (DoS) attack through NDP messages in small IPv6 network with different operating system.

1.4 Aim and Objectives

The aim of this thesis is to evaluate how far the DoS impacts the NDP and to find out the variety of attack's influence on different operating system and to determine the problems and vulnerabilities in NDP to improve the performance of IPv6. Three performance metrics will be used TCP Throughput, Round Trip Time (RTT) delay and CPU utilization to evaluate the impacts of DoS attacks over NDP and the response of the network various according to the OS.

1.5 Methodology

A small test network consists of monitoring computer, two victim's computers and one attacking computer is used to implement DoS attacks against IPv6 network.

This network is implemented virtually in laptop core i7-7700 HQ, RAM 16G and HD 1T. The test consists of one attacking node (Kali Linux 3.20.2) Kali Linux will be used to launch attacks with IPv6 address, the victim nodes Linux based (Ubuntu 16.04) and windows 7 will be used to test their behaviors and performance before and during attacks using monitoring computer Linux based (Ubuntu 16.04).

Three performances the Transfer Control Protocol (TCP) throughput, Round Trip Time (RTT) and CPU utilization are measured before and during the attacks. These computers are represented in VirtualBox 6.0 all with same specification one processor, RAM 3G and HD 100 GB, each of them connected to each other through nat-network adapter with the same name to provide virtual connectivity between computers in the level of layer two device as it is switch.

1.6 Thesis Organization

This thesis is composed of five chapters their outlines are as follow Chapter One includes introduction, problem statement, proposed solutions and methodology. Chapter Two contains a general overview of IPv6, NDP, Denial of Service Attacks (DOS). In addition, some of previous studies of the impacts of DoS over NDP on IPv6 link-local communication. Chapter Three contains a brief definition of the tools and software used in the methodology and detailed explanation of all stages to implement the proposed scenario. Chapter Four explains the result, Describes and discusses the results from the design. Chapter Five shows the conclusion of the thesis work and recommendation for future work.

CHAPTER TWO
LITERATURE REVIEW

Chapter Two

Literature Review

This chapter contains a general overview of IPv6, Neighbor Discovery Protocol (NDP), Denial of Service Attacks (DOS). In addition, some of previous studies of DOS over NDP.

2.1 Background

The current version of Internet Protocol, Internet Protocol version 4 (IPv4), According to the CERNET (China Education and Research Network), there is no IPv4 address to allocate in Asia, Europe, Latin America and Northern America. Internet Protocol version 6 (lPv6) is introduced as the next generation Internet Protocol, which is designed by IETF (lnternet Engineering Task Force) to replace IPv4. Exhaustion of IPv4 address space and security considerations are the major impetuses of introduction of IPv6. Compared with IPv4, IPv6 increases the IP address size from 32 bits to 128 bits, simplifies header format, improves support for extensions and options, enables the labeling of packets, and supports authentication, data integrity and mobility. IPv6 defines a new type of address called any cast address to send packets to any one of a group of nodes. IPv6 simplifies necessary header fields and introduces extensions and options to support authentication, data integrity, and other functions. [8]

2.2 Types and Categories of IPv6 Addresses

IPv6 addresses are divided into three main types: unicast address, multicast address and anycast address. Unicast Address is used to identify an interface. If a packet is sent to unicast address, it is only sent to a unique interface. There are three different unicast addresses: unicast global address, unicast link-local address and unicast site-local address. Unicast global addresses, also known as aggregatable global unicast addresses, are assigned by ISP to the sites which need to connect to Internet. They are similar with IPv4 public addresses [9]. Figure 2.1 showed The global routing prefix identifies the address range allocated to a site. This part of the address is assigned by the international registry services and the Internet service providers (ISP) and has a hierarchical structure. The subnet ID identifies a link within a site. A link can be assigned multiple subnet IDs. A local administrator of a site assigns this part of the address. The interface ID identifies an interface on a subnet and must be unique within that subnet[10].

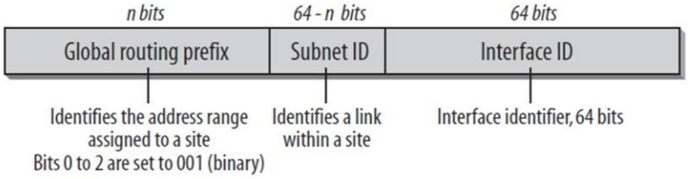

n bits
64 - n bits
64 bits
Global routing prefix
Subnet ID
Interface ID
Identifies the address range assigned to a site Bits 0 to 2 are set to 001 (binary)
Identifies a link within a site
Interface identifier, 64 bits

Figure 2.1 IPv6 unicast global address [10]

Unicast link-local addresses (the prefix is FE80::/64) are used by hosts when they want to communicate with other hosts in same local network. They are similar with IPv4 APIPA addresses, used by computers running Microsoft Windows. All IPv6 interfaces have link-local addresses and automatically configure these addresses to communicate with each other. Figure 2.2 shows the structure of link-local address[9].

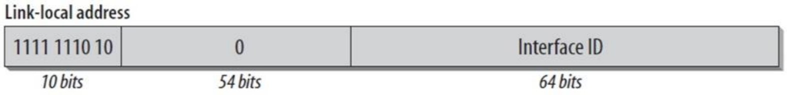


Figure 2.2. IPv6 unicast link-local address [10]

From Figure 2.2, it can be concluded that: first 10 bits are fixed values: 1111 1110 10; next 54 bits are 0 and last 64 bits are Interface address. A link-local addresses always has prefix is FE80::/64. Because unicast link-local address only be used in the same link, so router cannot transfer any packet with source or destination address is link-local address.

Unicast site-local addresses are not used in Internet. Normally they are used in an organization or a company. They are similar with IPv4 Private Address (10.X.X.X, 172.16.X.X, 192.168.X.X). First 10 bits are fixed: 1111 1110 11; next 38 bits are 0, next 16 bits are subnet ID and last 64 bits are interface ID. A site-local address always has the prefix FEC0::/48. The structure of site-local addresses is showed in Figure 2.4 [9]

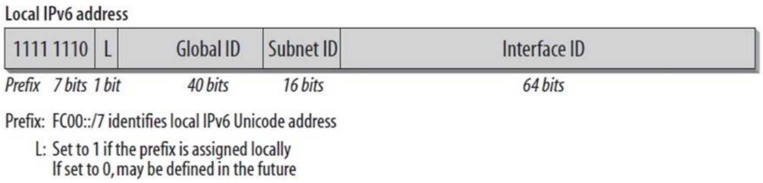

Local IPv6 address
1111 1110
L
Global ID
Subnet ID
Interface ID
Prefix
7 bits
1 bit
40 bits
16 bits
64 bits
Prefix: FC00::/7 identifies local IPv6 Unicode address
L: Set to 1 if the prefix is assigned locally
If set to 0, may be defined in the future

Figure 2.4. IPv6 Unicast Site-local Address [10]

Multicast Address is similar with multicast address in IPv4 and is used to identify a group of interfaces. IPv6 uses the address block with the prefix ff00::/8 for multicast applications. A message sent to a multicast IP address will be sent to all members of this group. It is often employed for streaming media applications on the Internet and private networks. In IPv6, broadcast address is removed. It is replaced and undertaken by multicast address. According to Figure 2.5, at the first octet, IPv6 multicast address has prefix is FF::/8. IPv6 addresses from FF00:: to FF0F:: are used for multicast purpose, defined by IANA [9] [11].

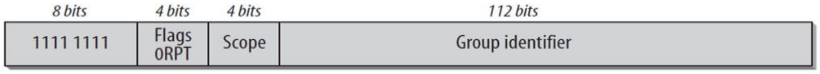


Figure 2.5. IPv6 Multicast Address [10]

The last bit of the Flag field indicates whether this address is permanently assigned one of the well-known multicast addresses assigned by the IANA—or a temporary multicast address. If this field is set to zero, the address is a permanently assigned multicast address. In contrast, if set to 1, it identifies a transient address. The scope field is used to identify the purpose of the multicast traffic, such as interface-local, link-local, site-local, organization-local, or global scope. The purpose of Group ID field is identifying the multicast group [10].

Anycast is used to identify multiple interfaces and be used primarily by large ISPs. Microsoft TechNet said "IPv6 delivers packets addressed to an anycast address to the nearest interface that the address identifies. In contrast to a multicast address, where delivery is from one to many, an anycast address delivery is from one to one-of-many. Currently, anycast addresses are assigned only to routers and are used only as destination addresses" [9].

2.2.1 IPv4 and IPv6 comparison

IPv6 addresses are 128 bits long instead of 32 bits. This expands the address space from around 4 billion addresses to, well, an astronomic number (over 300 trillion trillion trillion addresses). IPv6 address size was expanded so much was to allows address to be hierarchically divided to provide a large number of each of many classes of addresses [12]. There is a big difference between fixed header of IPv4 and fixed header of IPv6. IPv6 header has 40 octets (or 40 bytes), different with 20 octets in IPv4. However, numbers of field in IPv6 is less than

IPv4, so less time needed to spend to process headers, and because of that, it is faster and more flexible. Address field of IPv6 fixed header is 4 times bigger than address field of IPv4 fixed header [9]. The Header Checksum field provides a checksum on the IPv4 header only. The size of this field is 16 bits. The IPv4 payload is not included in the checksum calculation because the IPv4 payload usually contains its own checksum. Each IPv4 node that receives IPv4 packets verifies the IPv4 header checksum and silently discards the IPv4 packet if checksum verification fails. When a router forwards an IPv4 packet, it must decrement the TTL. Therefore, the Header Checksum value is recomputed at each hop between source and destination[13]. There is also no packet segmentation in IPv6. In IPv4, when a packet is too big, router can segment it, however, this can make overhead for packet. In IPv6, only source host can segment a packet following by suitable value depend on Maximum Transmission Unit (MTU) it can find. So, to supporting for source host, IPv6 have a field to help finding MTU from source to destination [9]. Figure 2.6 shows IPv4 and IPv6 fixed headers.

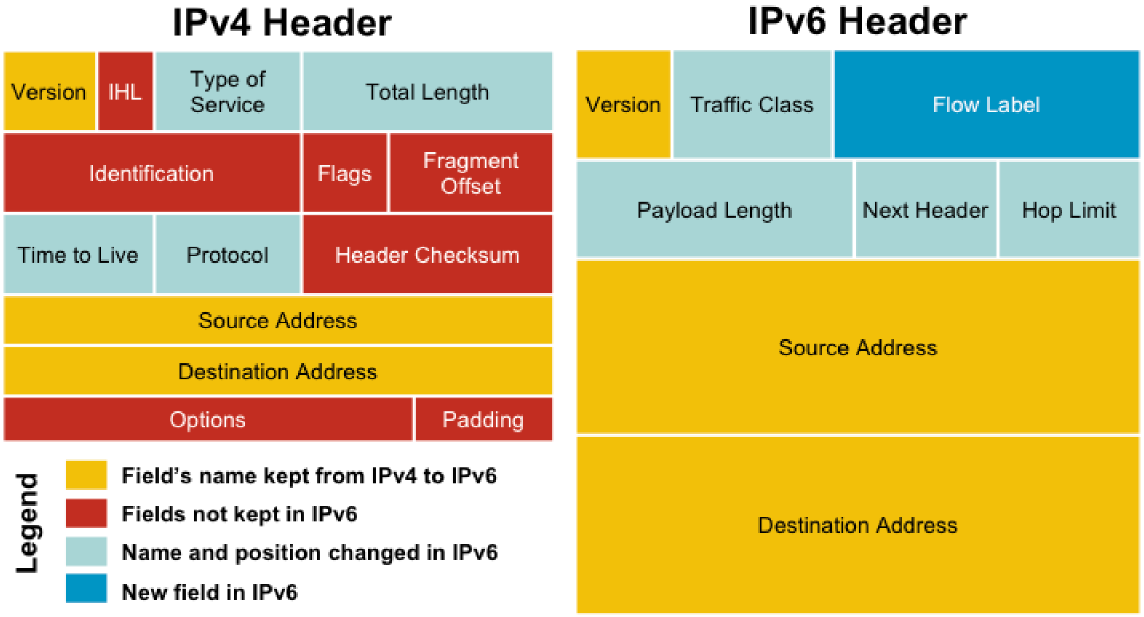


Figure 2.6 IPv4 and IPv6 fixed headers [9]

IPv6 uses 64 bits for Host-ID. A technique called EUI-64 is used to simply assign an address for a host comparing to IPv4. There are seven main advantages of IPv6 comparing with IPv4. They are: auto-configuration, high performance, mobility support, high security, simple header, route aggregation and renumbering IPv6 devices. Auto-configuration is used to simplify setting up host devices. IPv6 supports both stateful and stateless auto-configuration. Stateful auto-configuration requires manual configuration from administrators for IPv6 range on DHCPv6. With stateful auto-configuration, DHCPv6 takes charge of assign and administrate IP address for nodes over a network. DHCPv6 server will have a list of nodes and information about their state to know the availability of each IP address. In contrast to stateful auto-configuration, the hosts in network which uses stateless auto-configuration will connect with router and get the Network-ID. Even if there is no router, hosts in same network can determine their address from contents of received user advertisements. Stateless auto-configuration is normally suitable for individuals, small companies and organizations.

The transmission is in higher performance because IPv6 has enough IP addresses, so no need for private addresses, NAT or some other techniques. From that point of view, it can reduce the time to process packet’s header, reduce overhead because of address transformation. Using IPv6 can also reduce the routing time. Because many IPv4 ranges are allocated for users but cannot be summarized, it will increase amount of entries in routing table and overhead when routing. Different from that, IPv6 addresses are allocated through ISP, so reduce overhead and entry in routing table. IPv4 uses a lot of broadcast such as ARP request, when IPv6 uses Neighbor Discovery Protocol (NDP) to do auto-configuration function without using broadcast. Moreover, multicast limitation scope addresses such as global, organization-local, site-local, link-local or node-local are used to limit the multicast packets.

IP Security (IPsec) is an IETF standard protocol using for IP network security on both IPv4 and IPv6. Although basically, the functions of IPsec are the same on IPv4 and IPv6 environments, in IPv6, IPsec is compulsory and ready to be used. It makes IPv6 network safer. Header of IPv6 is simpler and reasonable than IPv4. IPv6 only has 6 fields if comparing with 10 fields in IPv4. So IPv6 packets will transfer faster, from that, increase network speed.

Route aggregation is a technique which is the same with route summarize in IPv4. ISP will summarize IP addresses with the same prefix and send that prefix to other routers for advertising purposes. By this way, routers can make routing tables smaller and increase routing scalability, leads to the network functions expansion such as optimizing bandwidth and increasing throughput used to connect more devices and service on Internet such as VoIP, TV on demand, high resolution video, real-time applications, game online, study or meeting online.

Renumbering IPv4 devices is a stressful issue for IT administrator. It affects network operation and consumes much manpower to re-configure IPv4 address for all devices in network. IPv6 is designed to renumber address easier. An IPv6 address which was assigned to nodes in two states: preferred and deprecated, depend on lifetime of that address. Lifetime can be configured manually on interface when configuring IP address or add to values used for auto-configuration on routers. Because of that, all nodes on IPv6 network can be renumbered by changing lifetime for a prefix on routers which provide this value. After that, routers can notice a new prefix and all nodes can renumber IP Address. In fact, node can maintain using old address for a period of time before deleting it totally [9].

2.2.2 Denial of Service Attack

One of the major concerns in interconnected networks of the current era is the network security. In this attack, the attacker aims to prevent a network node from acquiring a network address by generating the DoS [4] .

2.2.2.1 Overview of Denial of Service Attack

Network traffic can be disrupted by attack on one node which could severely affect the other nodes in a network. A network server may encounter various kinds of attacks, time to time, which results in the degrading of the performance of server in the network. A DoS which is considered to be a really troublesome problem to handle is one example of these attacks. A DoS attack takes place by preventing the victim node, by a malicious node, from communicate with other nodes on the network, as per Figure 2.8. Consequently, the victim node won't be able to process requests received from all other nodes. And because of this, the services needed by the authentic users could not get provided to them. Due to this, the inspection of the network traffic is essential to find the malicious or infected packets. And it should be done in such a way that the malicious packets are isolated from the uninfected ones thereby delivering services to the authentic users or clients smoothly. A small amount of resources and bandwidth are essential for the attackers to execute DoS attack. The attacks can take place in several ways, one way in which software vulnerabilities present in the victim node are exploited by an attacker and another way wherein an attacker produces a huge number of malicious packets[14].

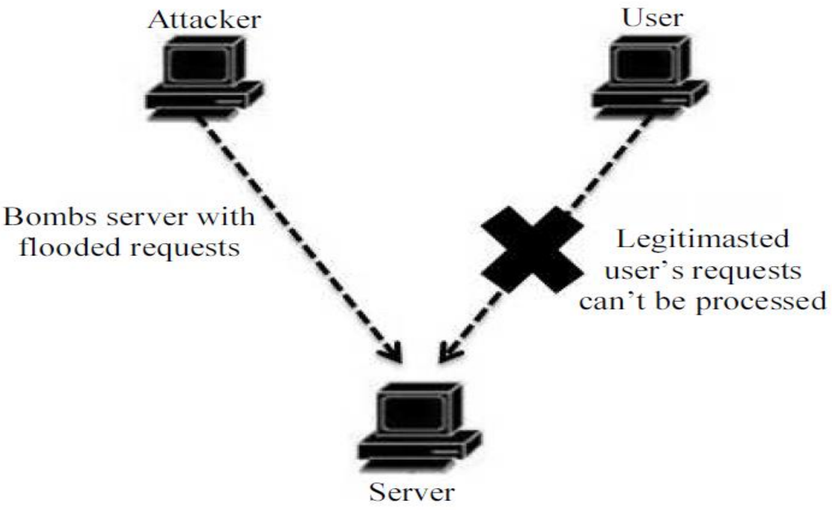


Figure 2.8. Denial of Service Attack [14]

A web server can be crashed by these types of attacks no matter what hardware capabilities it possesses. The first major DoS attack, recognized as email worm, was executed in Europe in the year 1987 by an IBM employee. The attack gathered quite some attention because IBM's shared network became overloaded and crashed in both continents Europe and USA. As a result of system downtime and recovery [15], a significant damage is still being caused to the productivity and revenues of corporates networks by these types of attacks. IPv6, which was created by the Internet Engineering Task Force (IETF) in order to address the limitations of IPv4, is exposed to DoS attacks. Legitimate nodes are prevented from acquiring access to network resources as a result of DoS attacks. Stealing of information is not included in a DoS attack instead the security of a network is violated and tends to discontinuing network connections. As these types of attacks are designed for the IP network, they can target any system regardless of its operating system. Therefore, any operating system using IPv4 or IPv6 can encounter these attacks [16]. Even though they are frequently aimed at IP network services, DoS attacks can also threat VoIP and other real-time services. The source of the DoS attack can be hidden by the attackers by means of spoofing, i.e., IP address spoofing or MAC address spoofing.

2.2.2.2 Classification of Denial of service Attack

A single computer is used in launching of a DoS attack, while Distributed Denial of Service (DDoS) attack is more complex than a DoS attack. A DDoS attack involves a number of compromised computers, known as zombies, which are all used at the same time [17]. Accordingly, flooding-based attacks could be initiated from one source in case of DoS attack or multiple sources in case of DDoS attack.

- **Software Exploits**

A low-rate DoS attack which, in order to remain hidden, keeps a low profile is referred to as software exploit. For the purpose of making use of the system vulnerabilities, to prevent authentic users from acquiring access to services and available resources, the attacker utilizes malicious nodes in a software exploits attacks.

- **Flooding**

In this type of DoS attack, the attacker sends a nonstoppable massive amount of packets to the victim's node to dissipate resources that can be earned by legitimate users. Due to this, the victim node freezes as the processing of the flood of malicious packets consumed all available resources. Traffic may be transferred from other nodes to the victim mode by the attacker during flooding attack [18]. Resulting in causing network congestion and consume the resources of the victim node like Central Processing Unit (CPU), memory or bandwidth.

Consequently, network communication amongst the victim and other nodes is prevented by this type of attack [19].

2.2.3 Neighbor Discovery Protocol

Consisting of a set of processes and messages as defined by [RFC 4861], IPv6 Neighbor Discovery (ND) is essentially a mechanism that determines how neighboring nodes relate to each other. ND was constituted as a replacement for the limited functionality of IPv4. It works along with IPv6 and replaces Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP) router discovery and the ICMP Redirect message used in IPv4. Nodes employ ND as a tool to perform a range of tasks. These tasks include non-router or host specific tasks, as well as router specific tasks. Among its general tasks are resolving problems associated with the neighboring node in regards to the link-layer address to which the IPv6 packet is being forwarded. In addition, it determines the reachability of a neighboring node along with its link-layer address. As for host specific tasks, ND is a tool to discover neighboring routers in addition to performing an automatic configuration of addresses, routes and prefixes among others parameters. As far as routers are concerned, ND seeks for router alternatives for improved next-hop performance to forward packets, in addition advertising router presence, configurations, routes and on-link prefixes [3].

2.2.3.1 Neighbor Discovery Message Format

There are five different types of ND messages, namely Router Solicitation (ICMPv6 type 133), Router Advertisement (ICMPv6 type 134), Neighbor Solicitation (ICMPv6 type 135), Neighbor Advertisement (ICMPv6 type 136) and Redirect (ICMPv6 type 137). All ND messages are formatted in a very specific way to operate within an ICMPv6 message structure. Messaging in ND consists of a message header, composed of an ICMPv6 header and ND message-specific data and zero or more ND options. Figure 2.8 shows the format of an ND message [13]. ND messages consist of several options that perform specific functions. These functions provide additional information, such as indicating MAC and IP addresses, on-link network prefixes, on-link MTU information, redirection data, mobility information and specific routes. All the messages that performs various functions pertaining to IPv6 ND specifically, five ICMPv6 messages are specified in [20], which are:

- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect

2.2.3.2 Router Solicitation

As a means to discover presence of IPv6 routers on the link, IPv6 hosts are used to send a multicast Router Solicitation message prompting an instant response from IPv6 routers as opposed to waiting for an unsolicited Router Advertisement message.

2.2.3.3 Router Advertisement

When multiple routers are advertised on a link, this can cause synchronization problems. To remedy this, unsolicited advertisements are sent at random intervals, which prompt a solicited response in the form of Router Advertisement messages, which contains various information demanded by hosts.

2.2.3.4 Neighbor Solicitation

IPv6 nodes send the Neighbor Solicitation message to discover the link-layer address of an onlinkIPv6 node or to confirm a previously determined link-layer address. It typically includes the link-layer address of the sender.

Typical Neighbor Solicitation messages are multicast for address resolution and unicast when the reachability of a neighboring node is being verified.

2.2.3.5 Neighbor Advertisement

In the event that a Neighbor Solicitation message is received, a Neighbor Advertisement message containing that information deemed necessary for nodes to determine the type of Neighbor Advertisement message and the sender's details is sent in return via the IPv6 node. At times, the same IPv6 node can send unsolicited Neighbor Advertisements as a means to track and inform neighboring nodes of changes in the role played by the nodes, i.e., in what pertains to link-layer addresses.

2.2.3.6 Redirect

The Redirect message is sent through an IPv6 router to acquire the details for an alternative (often better) first-hop address for a specific destination. Only routers can send this information, which is then relayed to the original host.

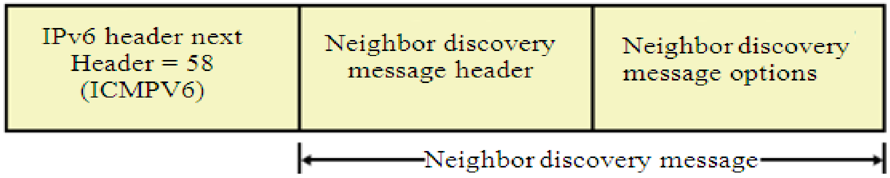

IPv6 header next Header = 58 (ICMPV6)
Neighbor discovery message header
Neighbor discovery message options
Neighbor discovery message

Figure 2.8. ND message format [13]

2.2.4 Neighbor Discovery Processes

There are several purposes behind message exchange within an ND protocol. These purposes include:

- Address resolution
- Duplicate Address Detection
- Neighbor unreachability detection
- Router discovery
- Redirect Function

2.2.4.1 Address Resolution

Resolving the problem of link-layer address of the on-link next-hop address for a given destination, requires the exchange between Neighbor Solicitation and Neighbor Advertisement messages. A multicast Neighbor Solicitation message is sent by the host which includes the link-layer address of the sending host in the Source Link-Layer Address option. Upon the target host receiving the message, the neighbor cache updates based on the source address and the link-layer address in the Source Link-Layer Address option. A Neighbor Advertisement consisting of the Target Link-Layer Address option is then sent by the target node to the Neighbor Solicitation sender. When the target nodes receive this, the neighbor cache of the sending host updates with an entry for the target after which it is possible to send unicast IPv6 traffic between the host and target.

2.2.4.2 Duplicate Address Detection

Duplicate address detection occurred when duplicate addresses on a local link is detected via means of Neighbor Solicitation messages, in which the

Target Address field is set to the IPv6 address for which duplication is being detected, as described in [21].

2.2.4.3 Neighbor Unreachability Detection

Neighbor Unreachability Detection: How nodes determine that a neighbor is no longer reachable. For neighbors used as routers, alternate default routers can be tried. For both routers and hosts, address resolution can be performed again [22]. The issue of neighbor Unreachability is when failure occurs in the receipt and process of IPv6 packets sent to the neighboring node. However, it is not an absolute determination that the sent packets did not arrive the designated destination, as a neighboring node can function as both host and router. This implies that the neighboring node may not have been the targeted destination. This process seeks only to determine if the first hop to the destination is reachable. This can be determined via a unicast Neighbor Solicitation message and the receipt of a solicited Neighbor Advertisement message. The Neighbor Advertisement message must be solicited to prove reach ability. This form of verification only works from Neighbor Solicitation to Neighbor Advertisement messages and not vice versa. Among the methods of ascertaining reachability is determining the forward progress of communication via the next-hop address. This is determined when acknowledgement segments for sent data are received. In the case of TCP, first hop reach ability to the destination is communicated to the IPv6 in the form of TCP acknowledgments. In those protocols wherein forward progress of communication cannot be determined, reach ability is determined through the exchange of Neighbor Solicitation and Neighbor Advertisement messages [3].

2.2.4.4 Router Discovery

When nodes seek to determine the set of routers on the local link, this is called router discovery. In the IPv6 protocol, this process is similar to ICMP router discovery for IPv4, as described in [23]. The major difference between both methods of discovery is the mechanism employed by both processes to select a new default router when the previous default router is no longer available. In the IPv6 process, the time span for a default router is included in the Router Lifetime field contained with the Router Advertisement message. When the current default router is no longer available, neighbor Unreachability detection is used instead of the Router Lifetime field to immediately select a new router from the list of possible default routers. It should be noted that the IPv6 router discovery mechanism performs a number of configuration [3].

2.2.4.5 Redirect Function

We redirect routers for improved first-hop traffic processing. In normal usage, there are two common occasions wherein the redirect function is employed. Firstly, when there are multiple routers on a local link, the IP address closest to the targeted destination is identified and traffic is redirected through it. Secondly, when the prefix of the destination is not included in the prefix list of the host, this is necessary to match the prefix on the list. The IPv6 redirect process consists of several steps. It begins by sending a unicast packet to its default router, which then processes the packet on the basis that the originating host is a neighbor and that the host and next-hop address share the same link. A redirect message is the sent to the originating host. In this message is the Target Address field, which serves as the next-hop address where the packet and all subsequent packets should be sent. When the Redirect message is received, the cache of the originating host updates the destination address with the address in the Target Address field [3].

2.2.5 NDP Vulnerabilities

According to [24], NDP vulnerabilities have three common types. The redirect attacks are the first vulnerability type whereby the malicious nodes are to direct away the packets from the legitimated nodes. Hence, we cannot trace the packets from the last hop router. It is important to mention that other genuine receivers are directed to alternative nodes upon facing the redirect attacks. The DoS is believed to be the second category of NDP vulnerabilities. The preventions of information flow between the attacked nodes and all other nodes, performed by malicious nodes, are likely to describe this type of attack [25]. The communication is also disallowed between the attacked nodes and specific intended addresses. Thirdly, the NDP is encountered by the attack of Flooding DoS [1]. The malicious nodes direct the traffic of other hosts to the victim node in such attack. A scenario of flooded bogus traffic is created whereby the victim host is the target. Three sub sections are used to identify threats, of NDP, with regarding to routing process in the given below section. These are: Threats that are related to the routing data, router independent threats and threats that can be remotely manipulated. We used NDP trust models and threats in [24] to outline those categories of threats.

2.2.5.1 Non-Routing Based Threats

- **Neighbor Solicitation/Advertisement Spoofing**

In this type of attack, legitimated nodes will not receive their legitimated packets. Instead, the attacker will divert it to other node either by sending NA message with incorrect target link layer address or NS message with incorrect source link layer address, as in Figure. 2.9. Neighbor Unreachability Detection (NUD) Failure This attack success because the attacker send a fabricated NA message in response to the victim NS message during NUD process [26]. The victim will be cheated by receiving this fabricated NA message and thought the neighbor is still reachable, while it is not.

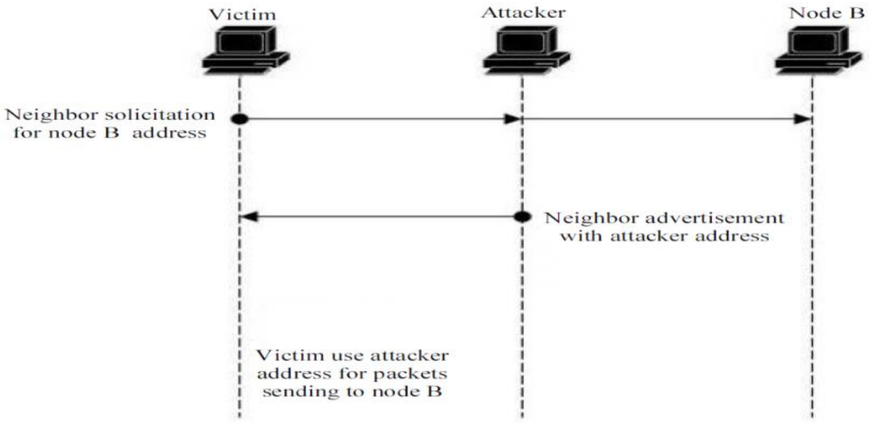

Victim
Attacker
Node B
Neighbor solicitation for node B address
Neighbor advertisement with attacker address
Victim use attacker address for packets sending to node B

Figure 2.9. Neighbor solicitation/advertisement spoofing attack

• Duplicate Address detection DOS Attack

When a new node joins an IPv6 link, it will make DAD check for the address that it trying to use. This is the nature of SLAAC mechanism within IPv6 communication link. As a response the attacker will replay to every single check for an IPv6 address that victim trying to use, claiming that he (attacker) already using this address [19]. This will prevent the victim from gaining a valid address and consequently denied access to the communication link, as in Figure. 2.10.

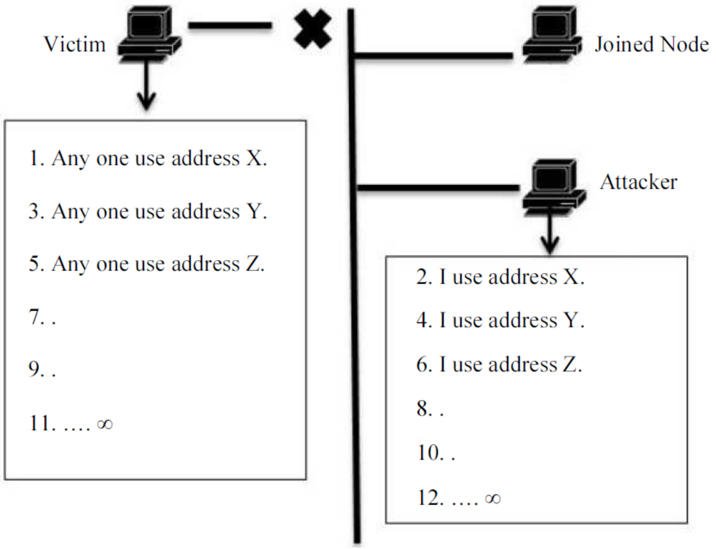


Figure 2.10. Duplicate address detection DoS attack [5]

2.2.5.2 Routing Based Threats

- **Malicious Last Hop Router**

Attacker in this type of attack pretending to act as last hop router by sending spoofed RA messages either as a response to RS message or in a routine base. This spoofed RA message, with the last hop router source address, has a short router life time. Followed by another RA message, has attacker source address, but with longer router life time [27]. Once the victim select attacker address as default router all traffic will be directed to the attacker’s host instead of the last hop router, as in Figure. 2.11.

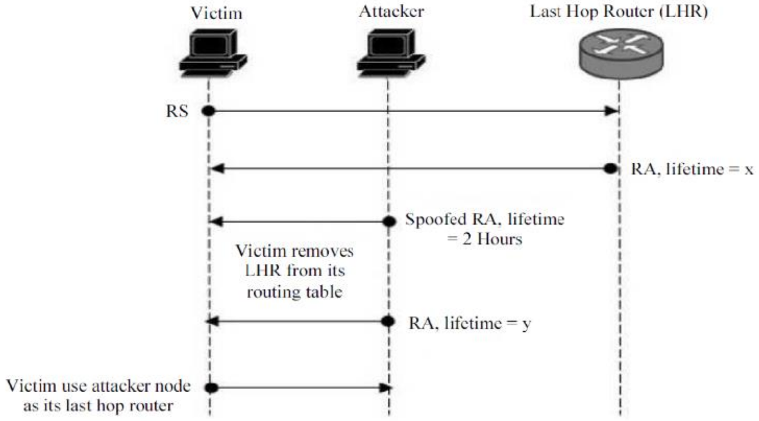

Victim
Attacker
Last Hop Router (LHR)
RS
RA, lifetime = x
Spoofed RA, lifetime = 2 Hours
Victim removes LHR from its routing table
RA, lifetime = y
Victim use attacker node as its last hop router

Figure 2.11. Malicious last hop router DoS attack [5]

• Default Router is Killed

In this type of attack, the victim assumes that all nodes are local. This is simply happened because attacker killed the default router, either by launching a DoS attack against the router or sends a spoofed RA message with zero life time and make default router list empty [28]. Consequently, and according to [20] victim will never send packets to the default router, as per Figure 2.12

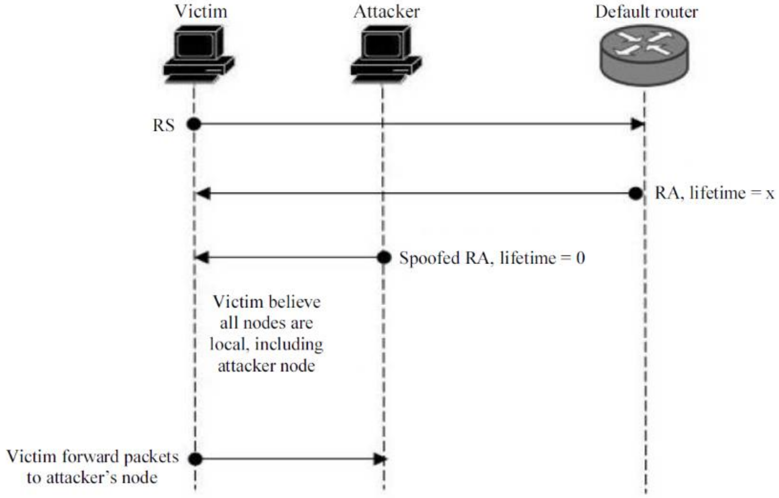

Victim
Attacker
Default router
RS
RA, lifetime = x
Spoofed RA, lifetime = 0
Victim believe all nodes are local, including attacker node
Victim forward packets to attacker's node

Figure 2.12. Default router is killed DoS attack [5]

- **Spoofed Redirect Message**

This attack used to redirect packets for a specific destination to another node attached to the local link. The attacker uses the current first hop router’s link-local address to send spoofed redirect message [29]. Packets will continue to flow to that specific destination as long as attacker replays to NUD messages.

- **Bogus in Link Prefix**

The attacker cheats the victim that some prefix is on-link by sending fabricated RA message. Accordingly, the victim will assume the nodes are on link and instead of send the packets to router it will send NS messages that will never be responded and lead to service denying to that node [30].

- **Bogus Address Configuration Prefix**

In this type of attack, the victim received a bogus RA message from attacker that identify wrong subnet prefix. Consequently, and according to SLACC procedure the victim will use this invalid prefix and construct invalid address. The victim will denied service as a result because nodes will replay using invalid source address of the victim when sending packets to victim’s host [30].

- **Parameter Spoofing**

As a part of SLAAC procedure the RA message contains some parameters that should be used by nodes in order to establish communication. The attacker executing this attack by sending RA messages that include incorrect parameters that may cause the communication between nodes to be interrupted [31].

2.2.5.3 Replay Threats

- **Replay Attacks**

The replay attacks are susceptible to all router discovery and neighbor discovery messages. The valid messages can also be captured by an attacker and he/she would replay them later, even if they were cryptographically secured so that one cannot falsify their contents. Hence, a secure mechanism must be established for protection against replay attacks [18].

- **Neighbor Discovery DOS Attack**

The addresses are fabricated with the subnet prefix and packets are continuously being sent to the victims in such type of attack. After sending neighbor solicitation packets, these addresses are resolved by the last hop router [18]. From the last hop router, the neighbor discovery service is not obtained by a legitimate host attempting to enter the network as it will be already busy with sending other solicitations. Since the attacker may be off-link, this DoS attack is different from the other attacks. In this attack, the conceptual neighbor cache is the resource being attacked, which will be occupied with attempts to resolve IPv6 addresses containing a valid prefix but invalid suffix [2].

2.2.6 Related Works

The authors in [5] proposed a complete test bed setup for examining IPv6 NDP related attacks, to report the impacts of DoS attacks over NDP. A research test bed was setup to implement several vulnerabilities that can be used by malicious nodes to launch attacks that prove these vulnerabilities are implemented on different types of operating systems Windows and Linux platforms.

The impacts of these attacks under different types of operating systems have been investigated, analyzed and evaluated using a real network before and during the different types of DoS attacks. The test bed consists of two operating system Windows-based and Linux-based with static IPv6 addresses, to test the Transfer Control Protocol (TCP) throughput, Round Trip Time (RTT) and CPU utilization before and during the attacks. Overall, the results had shown that performance of Linux based operating system is better than Windows based operating system. All attacks rely on the abusing or spoofing of the NDP message. According to IETF two types of solutions have been introduced to protect NDP, which are Internet Protocol Security (IPSec) Limitations NDP intended to use IPSec to protect itself through IP layer authentication and Secure Neighbor Discovery (SEND) but it has many limitations including computation, deployment and security.

Authors in [32] proposed the main processes of NDP and the security issues of these processes. the DoS attack is launched on each of the processes of NDP, including NUD, RD, DAD, etc. The experiments were conducted with the aim of measuring the NDP processes' performance during the DoS attack. The results revealed that the NDP processes are completely vulnerable to the DoS attack. The main problems with the current NDP processes include (i) the five NDP messages are unsecured by design, (ii) all the nodes that are located on a similar link (including the attacker) are capable of joining any NDP process, and (iii) there is no verification mechanism, which can detect the originality of these exchange NDP messages during these processes. A new prevention mechanism is, therefore, necessary to secure NDP messages, which are used during the processes. The proposed mechanism should be able to verify the incoming messages. It should distinguish between legitimate messages and illegitimate ones on both sides, i.e., the sender and the receiver.

The authors in [7] proposed the testbed was designed and implemented to analyze the impact of DoS on DAD attack and its outcome. The test was conducted on DoS-on-DAD attack in IPv6 networks and running the DoS attack on DAD process in IPv6 local link network called dos-new-ip6 to exploit the testbed setup environment during DAD process on Windows (Win7, Win Vista) and Linux based (Ubuntu, Fedora) hosts on deployed IPv6 testbed setup environment. Two experimental scenarios have been conducted the Normal Scenario and Attacking Scenario. In case of normal scenario address auto configuration process has been performed on hosts After successful DAD process hosts are able to configure their preferred IPv6 link local addresses. However, in Attacking Scenario it has been noticed that Windows-based hosts are unable to configure IPv6 link local addresses. While ongoing DoS attack, IPv6 hosts cannot obtain the IP addresse. Likewise, Linux Hosts are able to generate tentative IP address but fails to perform DAD process. Thus, due to the DAD process failure hosts are unable to verify the uniqueness of the generated (tentative) IP address. As a result, the new hosts are unable to communicate with their neighboring hosts on the same link. There are existing mechanisms and approaches that, to some length, address this issue but have drawbacks in terms of efficiency and complexity. Thus, a more effective security mechanism is required to secure DAD process during address auto-configuration in IPv6 link local network.

CHAPTER THREE
CONFIGURATION OF SYSTEM SCENARIOS

Chapter Three

Configuration of System Scenarios

This chapter contains a detailed explanation about the testbed of IPv6 network has been attacked by DoS attack. Also, a brief definition of the tools used to implement the experiment.

3.1 Simulation Test Case

By establishing the network in VirtualBox, the scenario of the network's connectivity will be as shown in Figure 3.1.

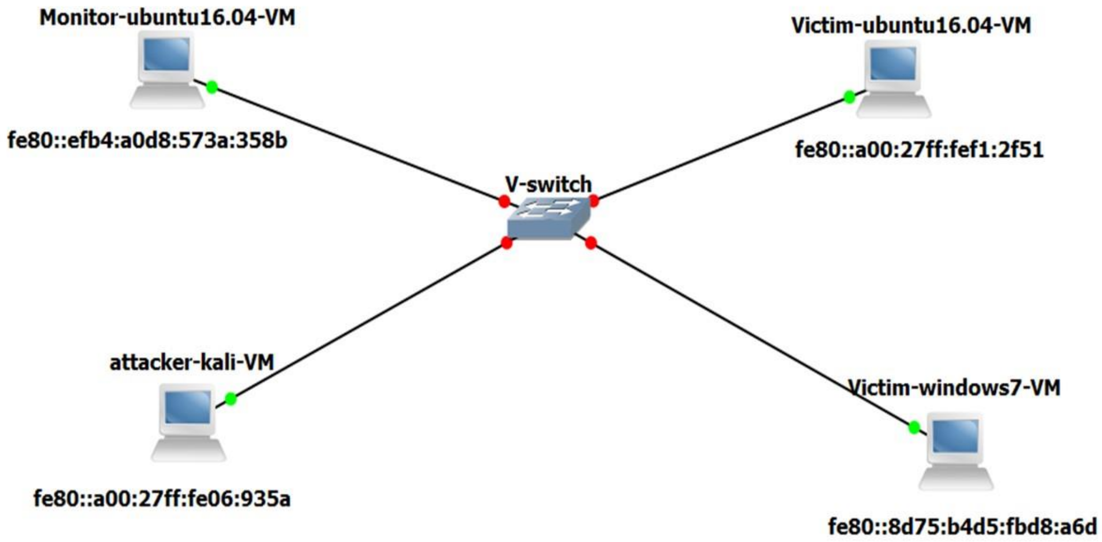


Figure 3.1 Network Topology

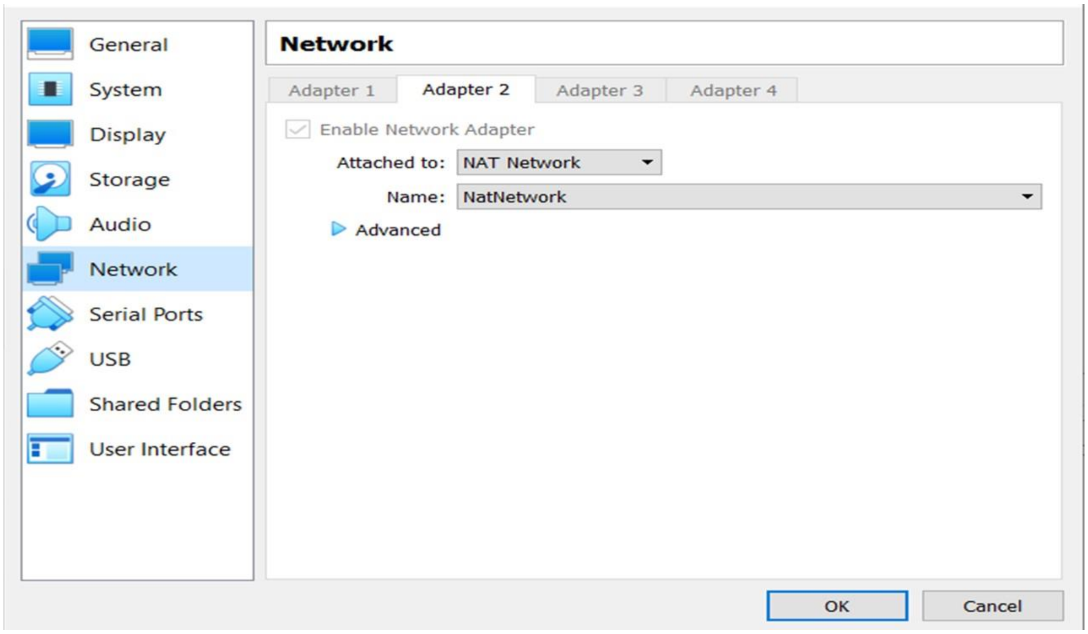
3.2 Configuration

In this section configuration is detailed as follows:

- **Configuring the Network**

The network consists of one monitoring computer, one attacking computer and two victim’s computers. As shown in Fig.3.1 Linux based computer with link-local address fe80::efb4:a0d8:573a:358b was set up as monitoring computer. Two victims' computers Windows and Linux based, which have link-local address fe80::8d75:b4d5:fbd8:a6d and fe80::a00:27ff:fef1:2f51 respectively, were used to test their behaviors and performance before and during attacks. Kali Linux was used to launch attacks with link-local address fe80::a00:27ff:fe06:935a.

All these devices are created as virtual machines connected to each other by nat-network adapters were attached for each VM which provide internal network that allows outbound connections is shown in Figure 3.2.



☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork

Figure 3.2. Nat-Network Adapter Configuration

3.3 System Tools

This section describes the system tools which are used to implement the simulation network as the follows:

- **VM VirtualBox**

Oracle VM VirtualBox is a cross-platform virtualization product that enables you to run multiple operating systems on your Mac OS, Windows, Linux, or Oracle Solaris systems.

- **Kali Linux**

Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali contains several hundred tools which are geared towards various information security tasks, such as Penetration Testing, Security research, Computer Forensics and Reverse Engineering.
Kali is a Linux-based open source system; it has built-in THC-IPv6 attacking toolkit support.

- **Iperf**

It is a network tool that measures TCP or User Datagram Protocol (UDP) bandwidth. Iperf can measure the maximum amount of data transmitted between any two hosts at any given time.

- **Ping**

Operates by sending ICMP/ICMPv6 echo request messages to the target node and waiting for an ICMP/ICMPv6 echo reply messages. The Ping utility program reports errors, packet loss and a statistical summary of the packets journey. It reports errors, packet loss and a statistical summary of the packets journey. Typically including the minimum, maximum, the mean round-trip times and standard deviation of the mean for the packets sent.

- **Virtual Memory Statistics**

Virtual memory statistics (vmstat) is a computer system monitoring tool that collects and displays summary information about operating system memory, processes, interrupts, paging and block I/O. Users of vmstat can specify a sampling interval which permits observing system activity in near-real time.

- **THC-IPv6**

  The hacker choice's IPv6 (THC-IPv6) is a complete tool set to attack the inherent protocol weakness of IPv6 and ICMP6 and includes an easy to use packet factory library.

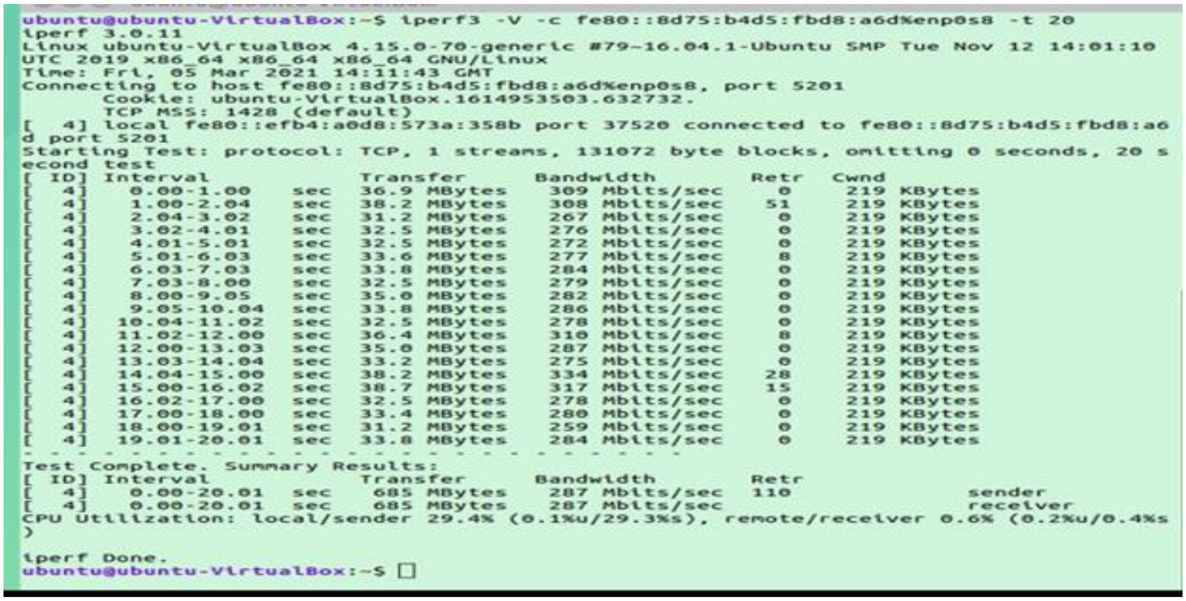
3.4 Testing Scenario

In this network topology the test has been applied over the network before and during different types of DoS attacks which they are router solicitation (RS), router advertisement (RA), neighbor solicitation (NS) and neighbor advertisement (NA) to evaluate the impacts of DoS attacks over NDP. Impacts of DoS over a network could be measured using a parameter such as:

- **TCP Throughput**

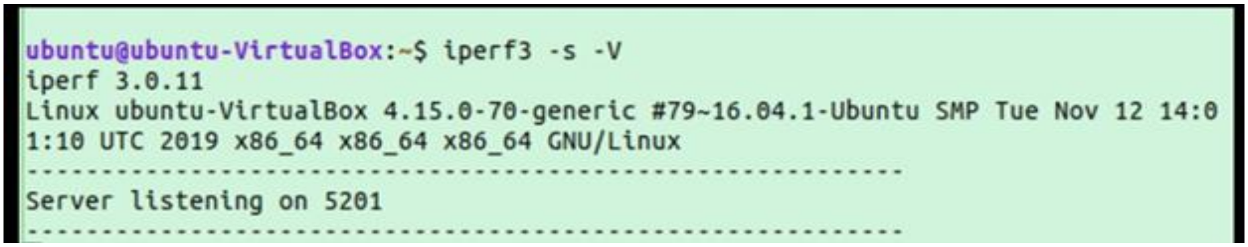
  Network throughput defined as the average number of bytes received successfully by the intended receiver at a given time. Impacts of DoS over a network could be measured using a parameter such as TCP Throughput. Throughput is important for TCP based traffic, as it may lower the ratio at which it sends packets in case of network congestion occurred. TCP Throughput was measured on Windows 7 and UBUNT 16.04 clients using Iperf. By default, Iperf uses port 5001 and 10 sec tests time periods. In this research 20 sec were used to test time periods for more consistency. Iperf can measure the maximum amount of data transmitted between any two hosts at any given time. For Iperf to work correctly it needs to be installed on two hosts one act as Iperf client and the other act as Iperf server.

  Iperf was installed on victims and monitoring computers. On monitoring computer Ubuntu 16.04 is defined as Iperf client as shown in Figure 3.3 and victim's computers are defined as Iperf server in Ubuntu 16.04 as shown in Figure 3.4. and in Windows 7 as shown in Figure 3.5. TCP throughput was measured between the monitoring computer Ubuntu16.04 and victim's computer Windows 7 and then it was measured between the monitoring computer and victim's computer Ubuntu 16.04. Thus, TCP Throughput was measured in Mega Bytes per second (MBps).



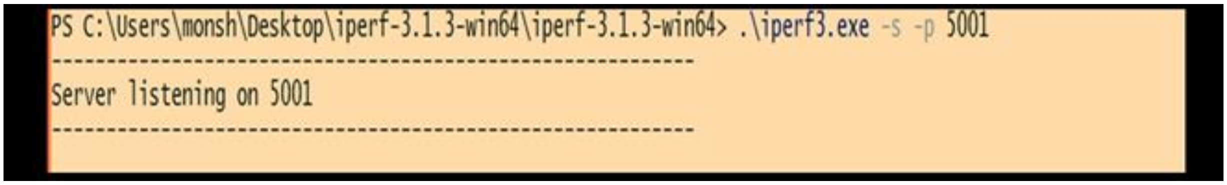
```
ubuntu@ubuntu-VirtualBox:~$ iperf3 -V -c fe80::8d75:b4d5:fbd8:a6d%enp0s8 -t 20
iperf 3.0.11
Linux ubuntu-VirtualBox 4.15.0-70-generic #79~16.04.1-Ubuntu SMP Tue Nov 12 14:01:10
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
Time: Fri, 05 Mar 2021 14:11:43 GMT
Connecting to host fe80::8d75:b4d5:fbd8:a6d%enp0s8, port 5201
      Cookie: ubuntu-VirtualBox.1614953503.632732.
      TCP MSS: 1428 (default)
[  4] local fe80::efb4:a0d8:573a:358b port 37520 connected to fe80::8d75:b4d5:fbd8:a6
d port 5201
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks, omitting 0 seconds, 20 s
econd test
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  36.9 MBytes   309 Mbits/sec    0    219 KBytes
[  4]   1.00-2.04   sec  38.2 MBytes   308 Mbits/sec   51    219 KBytes
[  4]   2.04-3.02   sec  31.2 MBytes   267 Mbits/sec    0    219 KBytes
[  4]   3.02-4.01   sec  32.5 MBytes   276 Mbits/sec    0    219 KBytes
[  4]   4.01-5.01   sec  32.5 MBytes   272 Mbits/sec    0    219 KBytes
[  4]   5.01-6.03   sec  33.6 MBytes   277 Mbits/sec    8    219 KBytes
[  4]   6.03-7.03   sec  33.8 MBytes   284 Mbits/sec    0    219 KBytes
[  4]   7.03-8.00   sec  32.5 MBytes   279 Mbits/sec    0    219 KBytes
[  4]   8.00-9.05   sec  35.0 MBytes   282 Mbits/sec    0    219 KBytes
[  4]   9.05-10.04  sec  33.8 MBytes   286 Mbits/sec    0    219 KBytes
[  4]  10.04-11.02  sec  32.5 MBytes   278 Mbits/sec    0    219 KBytes
[  4]  11.02-12.00  sec  36.4 MBytes   310 Mbits/sec    8    219 KBytes
[  4]  12.00-13.03  sec  35.0 MBytes   287 Mbits/sec    0    219 KBytes
[  4]  13.03-14.04  sec  33.2 MBytes   275 Mbits/sec    0    219 KBytes
[  4]  14.04-15.00  sec  38.2 MBytes   334 Mbits/sec   28    219 KBytes
[  4]  15.00-16.02  sec  38.7 MBytes   317 Mbits/sec   15    219 KBytes
[  4]  16.02-17.00  sec  32.5 MBytes   278 Mbits/sec    0    219 KBytes
[  4]  17.00-18.00  sec  33.4 MBytes   280 Mbits/sec    0    219 KBytes
[  4]  18.00-19.01  sec  31.2 MBytes   259 Mbits/sec    0    219 KBytes
[  4]  19.01-20.01  sec  33.8 MBytes   284 Mbits/sec    0    219 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-20.01  sec   685 MBytes   287 Mbits/sec  110             sender
[  4]   0.00-20.01  sec   685 MBytes   287 Mbits/sec                  receiver
CPU Utilization: local/sender 29.4% (0.1%u/29.3%s), remote/receiver 0.6% (0.2%u/0.4%s
)

iperf Done.
ubuntu@ubuntu-VirtualBox:~$ 
```

Figure 3.3. Executing Iperf in monitoring computer



```
ubuntu@ubuntu-VirtualBox:~$ iperf3 -s -V
iperf 3.0.11
Linux ubuntu-VirtualBox 4.15.0-70-generic #79~16.04.1-Ubuntu SMP Tue Nov 12 14:0
1:10 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

Figure 3.4. Executing Iperf in victim's computer Ubuntu 16.04.

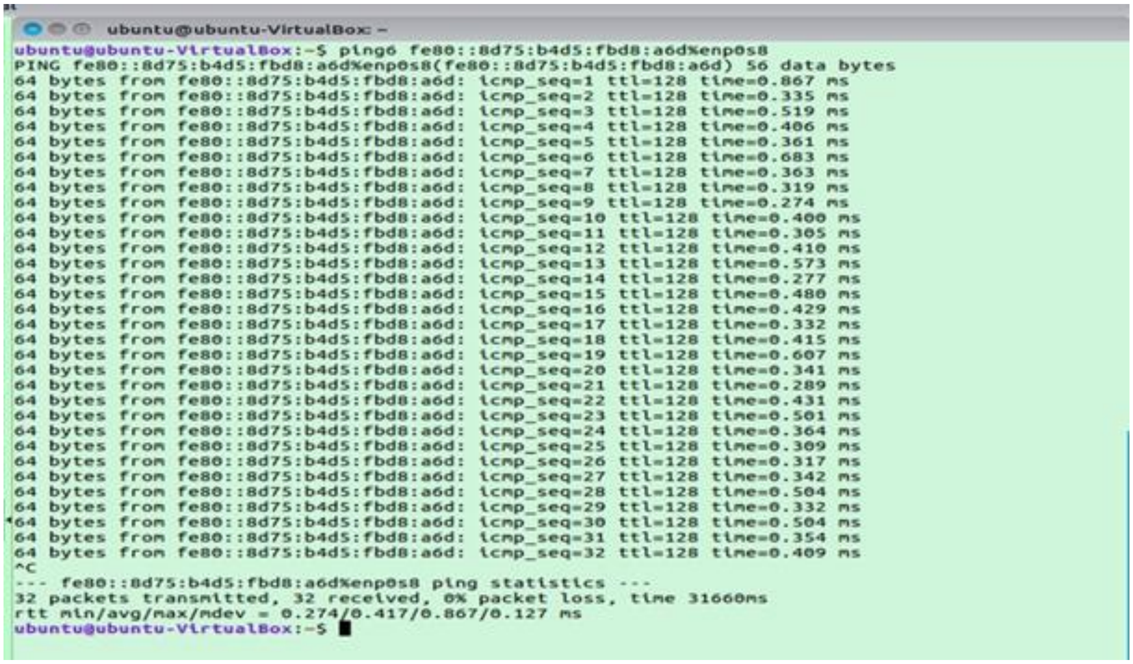


```
PS C:\Users\monsh\Desktop\iperf-3.1.3-win64\iperf-3.1.3-win64> .\iperf3.exe -s -p 5001
-----------------------------------------------------------
Server listening on 5001
-----------------------------------------------------------
```

Figure 3.5. Executing Iperf in victim's computer Windows 7

- **Round Trip Time**

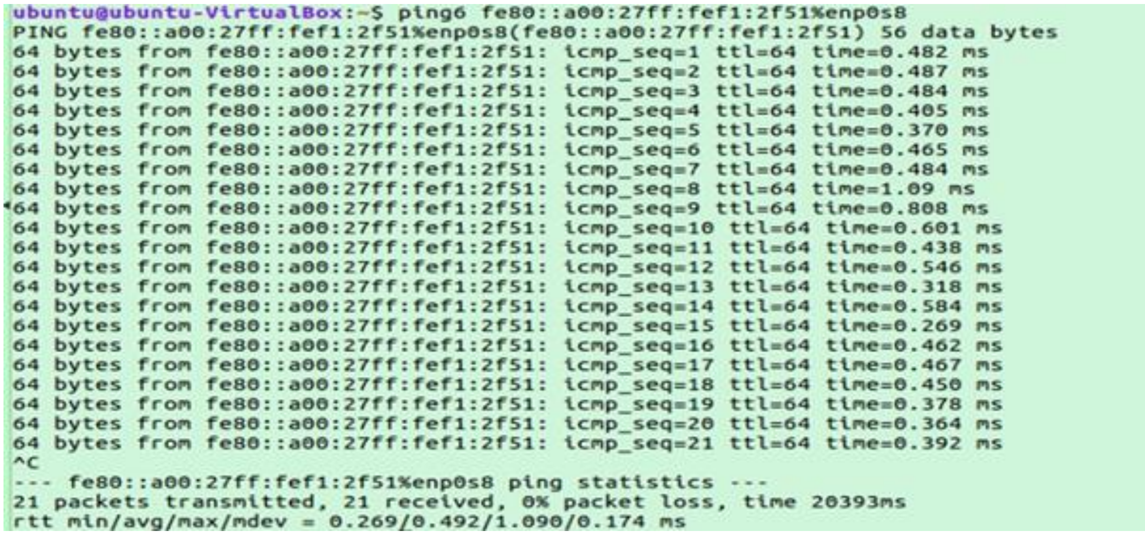
RTT is calculated by subtracting the time at which a network packet was sent from the time at which acknowledged, for this packet, is received. RTT is significant because it used for measuring delay within computers networks. A packet considered lost if it is going beyond its predefined RTT, that's why during DoS attack retransmissions always occurred. RTT delay was measured on Windows 7 and Ubuntu 16.04 using ping utility executing in monitor computer to the victims as shown in Figure 3.6 and 3.7 as samples of measuring the RTT.



```
ubuntu@ubuntu-VirtualBox:~$ ping6 fe80::8d75:b4d5:fbd8:a6d%enp0s8
PING fe80::8d75:b4d5:fbd8:a6d%enp0s8(fe80::8d75:b4d5:fbd8:a6d) 56 data bytes
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=1 ttl=128 time=0.867 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=2 ttl=128 time=0.335 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=3 ttl=128 time=0.519 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=4 ttl=128 time=0.406 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=5 ttl=128 time=0.361 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=6 ttl=128 time=0.683 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=7 ttl=128 time=0.363 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=8 ttl=128 time=0.319 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=9 ttl=128 time=0.274 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=10 ttl=128 time=0.400 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=11 ttl=128 time=0.305 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=12 ttl=128 time=0.410 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=13 ttl=128 time=0.573 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=14 ttl=128 time=0.277 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=15 ttl=128 time=0.480 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=16 ttl=128 time=0.429 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=17 ttl=128 time=0.332 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=18 ttl=128 time=0.415 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=19 ttl=128 time=0.607 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=20 ttl=128 time=0.341 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=21 ttl=128 time=0.289 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=22 ttl=128 time=0.431 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=23 ttl=128 time=0.501 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=24 ttl=128 time=0.364 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=25 ttl=128 time=0.309 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=26 ttl=128 time=0.317 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=27 ttl=128 time=0.342 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=28 ttl=128 time=0.504 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=29 ttl=128 time=0.332 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=30 ttl=128 time=0.504 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=31 ttl=128 time=0.354 ms
64 bytes from fe80::8d75:b4d5:fbd8:a6d: icmp_seq=32 ttl=128 time=0.409 ms
^C
--- fe80::8d75:b4d5:fbd8:a6d%enp0s8 ping statistics ---
32 packets transmitted, 32 received, 0% packet loss, time 31660ms
rtt min/avg/max/mdev = 0.274/0.417/0.867/0.127 ms
ubuntu@ubuntu-VirtualBox:~$ 
```

Figure 3.6. Measuring RTT using ping computer for windows 7in monitoring

Ping is a network utility used to test the reachability of a node within IP networks. It measures the RTT for packets sent from a source node to destination node. Ping operates by sending ICMP/ICMPv6 echo request messages to the target node and waiting for an ICMP/ICMPv6 echo reply messages. The Ping utility program reports errors, packet loss and a statistical summary of the packets journey.



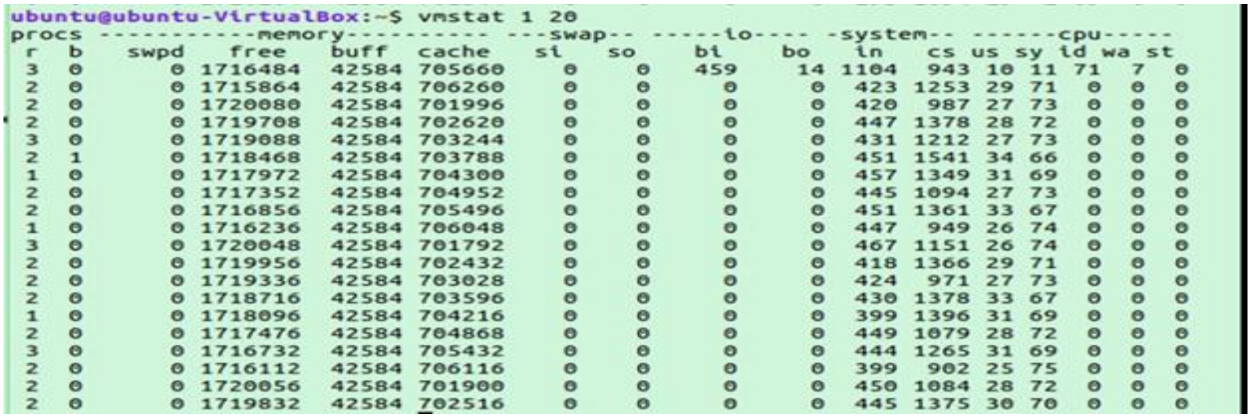
```
ubuntu@ubuntu-VirtualBox:~$ ping6 fe80::a00:27ff:fef1:2f51%enp0s8
PING fe80::a00:27ff:fef1:2f51%enp0s8(fe80::a00:27ff:fef1:2f51) 56 data bytes
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=1 ttl=64 time=0.482 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=2 ttl=64 time=0.487 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=3 ttl=64 time=0.484 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=4 ttl=64 time=0.405 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=5 ttl=64 time=0.370 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=6 ttl=64 time=0.465 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=7 ttl=64 time=0.484 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=8 ttl=64 time=1.09 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=9 ttl=64 time=0.808 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=10 ttl=64 time=0.601 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=11 ttl=64 time=0.438 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=12 ttl=64 time=0.546 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=13 ttl=64 time=0.318 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=14 ttl=64 time=0.584 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=15 ttl=64 time=0.269 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=16 ttl=64 time=0.462 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=17 ttl=64 time=0.467 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=18 ttl=64 time=0.450 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=19 ttl=64 time=0.378 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=20 ttl=64 time=0.364 ms
64 bytes from fe80::a00:27ff:fef1:2f51: icmp_seq=21 ttl=64 time=0.392 ms
^C
--- fe80::a00:27ff:fef1:2f51%enp0s8 ping statistics ---
21 packets transmitted, 21 received, 0% packet loss, time 20393ms
rtt min/avg/max/mdev = 0.269/0.492/1.090/0.174 ms
```

Figure 3.7. Measuring RTT using ping in monitoring computer for Ubuntu 16.0

Typically including the minimum, maximum, the mean round-trip times and standard deviation of the mean for the packets sent. In our experiment, Ping measured RTT between monitoring computer and victims' computers. Ping was installed by default on Windows 7 and Ubuntu 16.04. We test the RTT 20 times between the monitor computer and victims' computers for more consistency. It was measured in milliseconds.

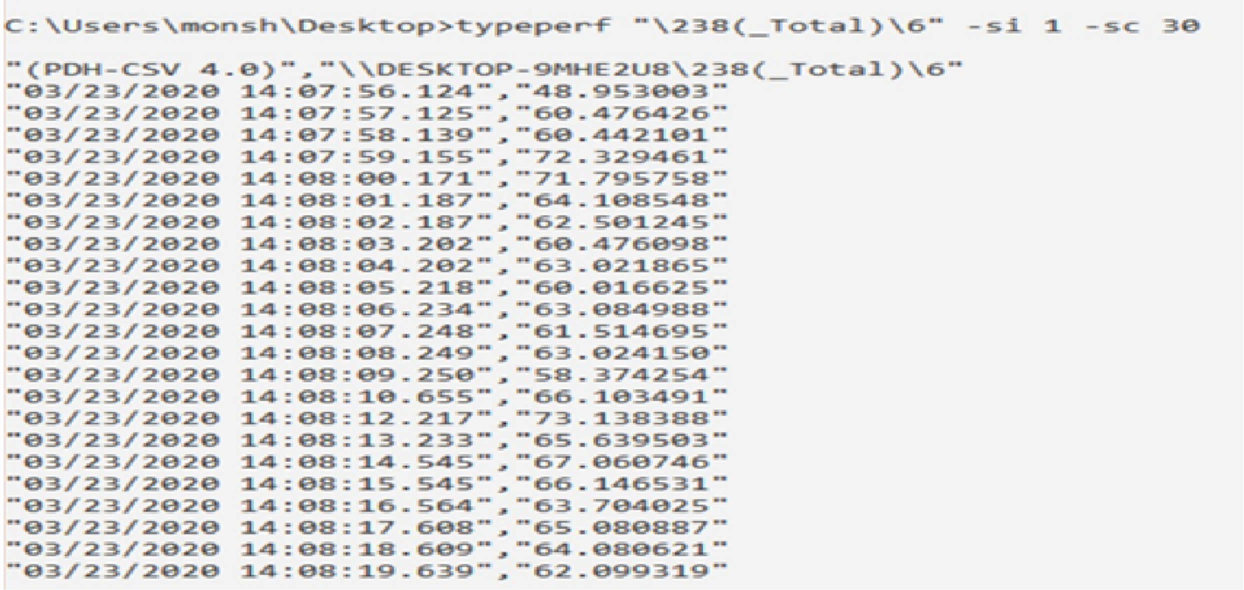
- **CPU Utilization**

During DoS based attacks packet transmission exhaust the processor, which in turn reduce the host's performance. To see processor utilization, hard disk, network and memory usage. For Linux based systems a tool did exist is named vmstat. In our experiment, **vmstat** is used to monitor the computer's processor usage on Ubuntu 16.04 Victim for a period of 20 seconds, taking the ideal value subtract it from 100 to gain the utilized value of cpu, a sample of cpu utilization's result is shown in Figure 3.8. While in Windows 7 the processor utilization shown by **typeperf** command which writes performance data to the command window. CPU utilization was measured as percentage. A sample of cpu utilization's result is shown in Figure 3.9



```
ubuntu@ubuntu-VirtualBox:~$ vmstat 1 20
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3  0      0 1716484  42584 705660    0    0   459    14 1104  943 10 11 71  7  0
 2  0      0 1715864  42584 706260    0    0     0     0  423 1253 29 71  0  0  0
 2  0      0 1720080  42584 701996    0    0     0     0  420  987 27 73  0  0  0
 2  0      0 1719708  42584 702620    0    0     0     0  447 1378 28 72  0  0  0
 3  0      0 1719088  42584 703244    0    0     0     0  431 1212 27 73  0  0  0
 2  1      0 1718468  42584 703788    0    0     0     0  451 1541 34 66  0  0  0
 1  0      0 1717972  42584 704300    0    0     0     0  457 1349 31 69  0  0  0
 2  0      0 1717352  42584 704952    0    0     0     0  445 1094 27 73  0  0  0
 2  0      0 1716856  42584 705496    0    0     0     0  451 1361 33 67  0  0  0
 1  0      0 1716236  42584 706048    0    0     0     0  447  949 26 74  0  0  0
 3  0      0 1720048  42584 701792    0    0     0     0  467 1151 26 74  0  0  0
 2  0      0 1719956  42584 702432    0    0     0     0  418 1366 29 71  0  0  0
 2  0      0 1719336  42584 703028    0    0     0     0  424  971 27 73  0  0  0
 2  0      0 1718716  42584 703596    0    0     0     0  430 1378 33 67  0  0  0
 1  0      0 1718096  42584 704216    0    0     0     0  399 1396 31 69  0  0  0
 2  0      0 1717476  42584 704868    0    0     0     0  449 1079 28 72  0  0  0
 3  0      0 1716732  42584 705432    0    0     0     0  444 1265 31 69  0  0  0
 2  0      0 1716112  42584 706116    0    0     0     0  399  902 25 75  0  0  0
 2  0      0 1720056  42584 701900    0    0     0     0  450 1084 28 72  0  0  0
 2  0      0 1719832  42584 702516    0    0     0     0  445 1375 30 70  0  0  0
```

Figure 3.8. Executing vmstat



```
C:\Users\monsh\Desktop>typeperf "\238(_Total)\6" -si 1 -sc 30

"(PDH-CSV 4.0)","\\DESKTOP-9MHE2U8\238(_Total)\6"
"03/23/2020 14:07:56.124","48.953003"
"03/23/2020 14:07:57.125","60.476426"
"03/23/2020 14:07:58.139","60.442101"
"03/23/2020 14:07:59.155","72.329461"
"03/23/2020 14:08:00.171","71.795758"
"03/23/2020 14:08:01.187","64.108548"
"03/23/2020 14:08:02.187","62.501245"
"03/23/2020 14:08:03.202","60.476098"
"03/23/2020 14:08:04.202","63.021865"
"03/23/2020 14:08:05.218","60.016625"
"03/23/2020 14:08:06.234","63.084988"
"03/23/2020 14:08:07.248","61.514695"
"03/23/2020 14:08:08.249","63.024150"
"03/23/2020 14:08:09.250","58.374254"
"03/23/2020 14:08:10.655","66.103491"
"03/23/2020 14:08:12.217","73.138388"
"03/23/2020 14:08:13.233","65.639503"
"03/23/2020 14:08:14.545","67.060746"
"03/23/2020 14:08:15.545","66.146531"
"03/23/2020 14:08:16.564","63.704025"
"03/23/2020 14:08:17.608","65.080887"
"03/23/2020 14:08:18.609","64.080621"
"03/23/2020 14:08:19.639","62.099319"
```

Figure 3.9. Executing typeperf command

Table 3.1 shows the NDP attacks and corresponding commands to execute it in the attacking computer. Table 3.2 illustrates the software and hardware specifications of the joint nodes and the role of each node as well.

Table 3.1: Attacks commands

Attack name	Command

RS Flooding	atk6-flood_rs6 [-sS] interface [target]
RA Flooding	atk6-flood_router6 <interface>
NS Flooding	atk6-flood_solicitate6 <interface> [target-ip]
NA Flooding	atk6-flood_advertise6 <interface> [target-ip]
NS/NA Spoofing	atk6-parasite6 <interface> [fake-mac]

Table 3.2: Computers roles, software and hardware specifications

Node role	Operating system	MAC address	Software
Monitoring	Ubuntu 16.04	08:00:27:0c:81:a7	Iperf.
Attacker	Kali Linux 3.20.2	08:00:27:06:93:5a	THC-IPV6.
Victim	Ubuntu 16.04	08:00:27:f1:2f:51	Iperf/vmstat
Victim	Windows 7	08-00-27-45-69-CC	Iperf/typeperf

CHAPTER FOUR
RESULT AND DISCUSSION

Chapter Four

Result and Discussion

In this chapter, the attacks will be deployed over IPv6 network and all the results will be captured and discussed.

4.1 System Simulation

In this experiment, to evaluate the impacts of DoS attacks over NDP of victim's computers on IPv6 network using three performance metrics before and during attacks, the attacks have been applied by attacker computer and the results has been shown by monitor computer which is used to monitor the taken for a period of 20 sec.

Table 4.1: Experiment legends

Legend	Description
LBA	Linux Before Attack.
LDA	Linux During Attack.
WBA	Windows Before Attack.
WDA	Windows During Attack.

Refer to Figure 4.1 the TCP throughput in windows before attack was fluctuating slightly between (727 – 1126.4) Mbps. While under RS flooding attack the TCP throughput declined and was varying modestly between (259-334) Mbps. The percentage of decreasing was 69.99%.

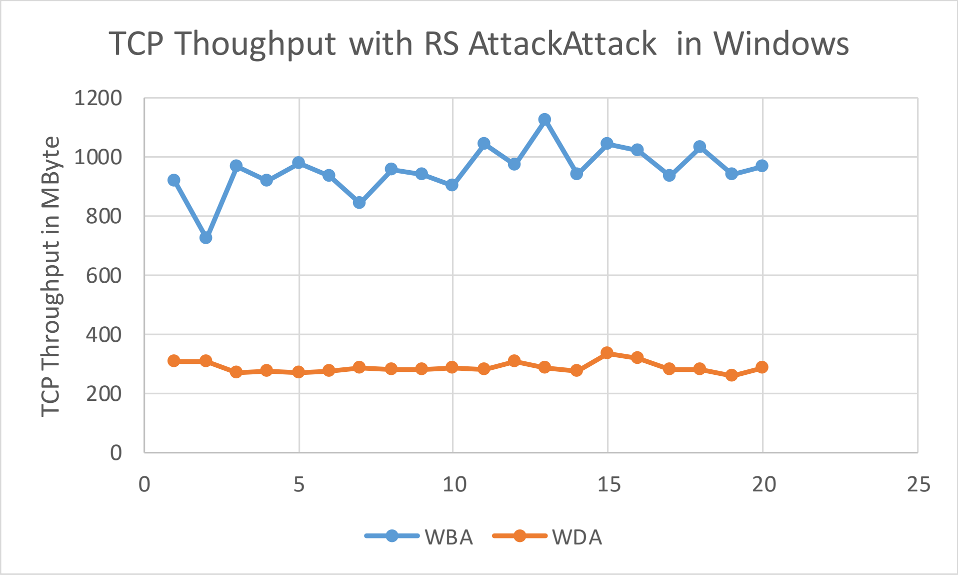


Figure 4.1. TCP Throughput with RS Attack in windows

Refer to Figure 4.2, 4.3 and 4.4 the TCP throughput in windows under RA, NS and NA flooding attack respectively plummeted to zero Mbps. The percentage of decreasing was 100%.

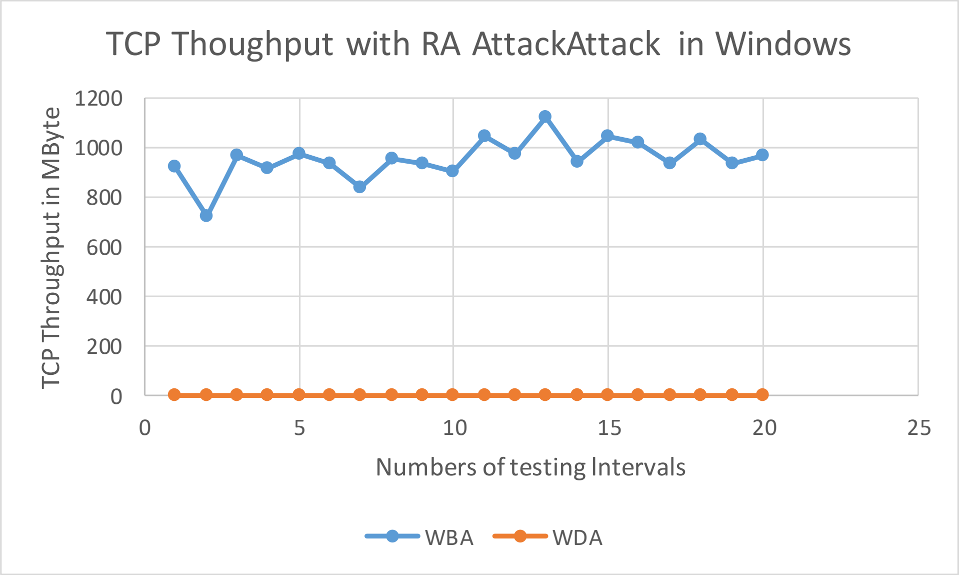


Figure 4.2. TCP Throughput with RA Attack in windows

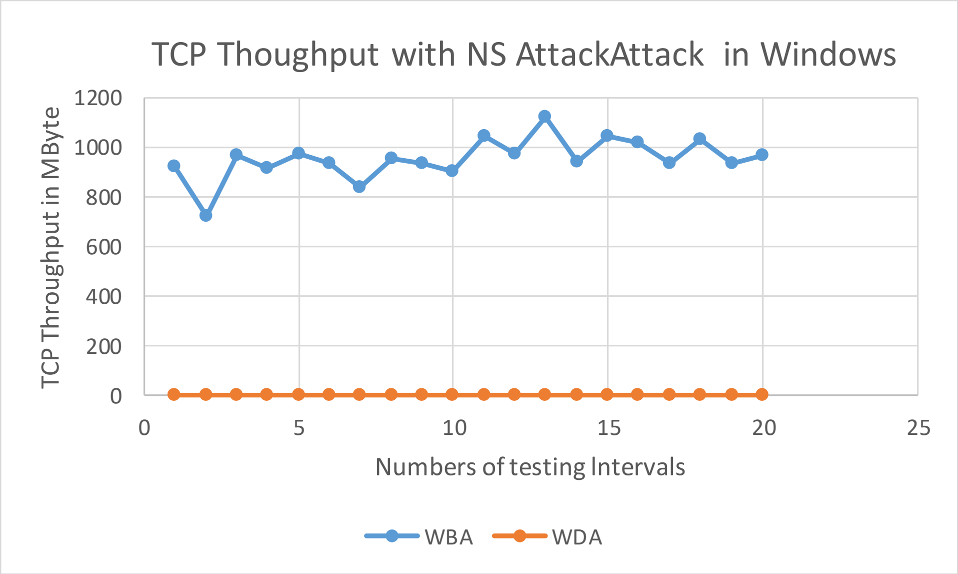


Figure 4.3. TCP Throughput with NS Attack in Windows

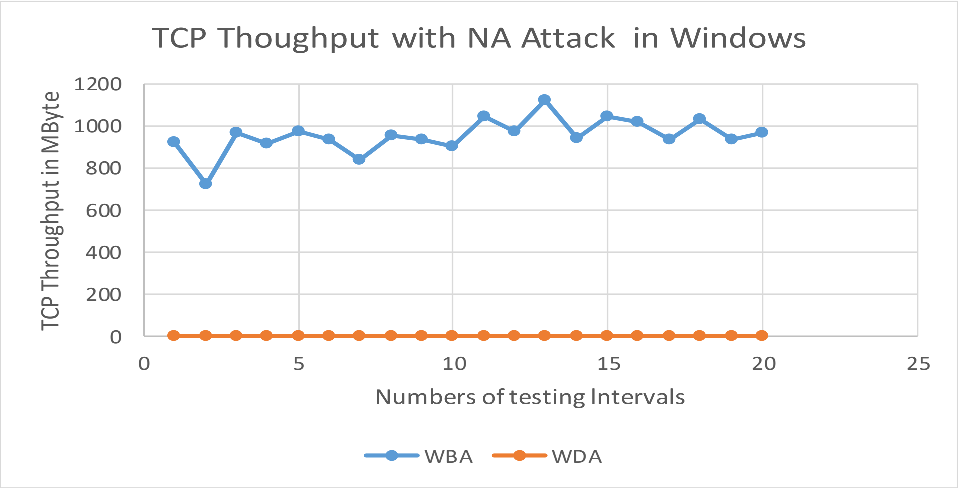


Figure 4.4. TCP Throughput with NA Attack in Windows

Refer to Figure 4.5 the TCP throughput in Linux before attack was fluctuating marginally between (1110 - 1400) Mbps. While the TCP throughput under RS flooding attack decreased and was fluctuating gradually between (566-864) Mbps. The percentage of decreasing was 43.1%.

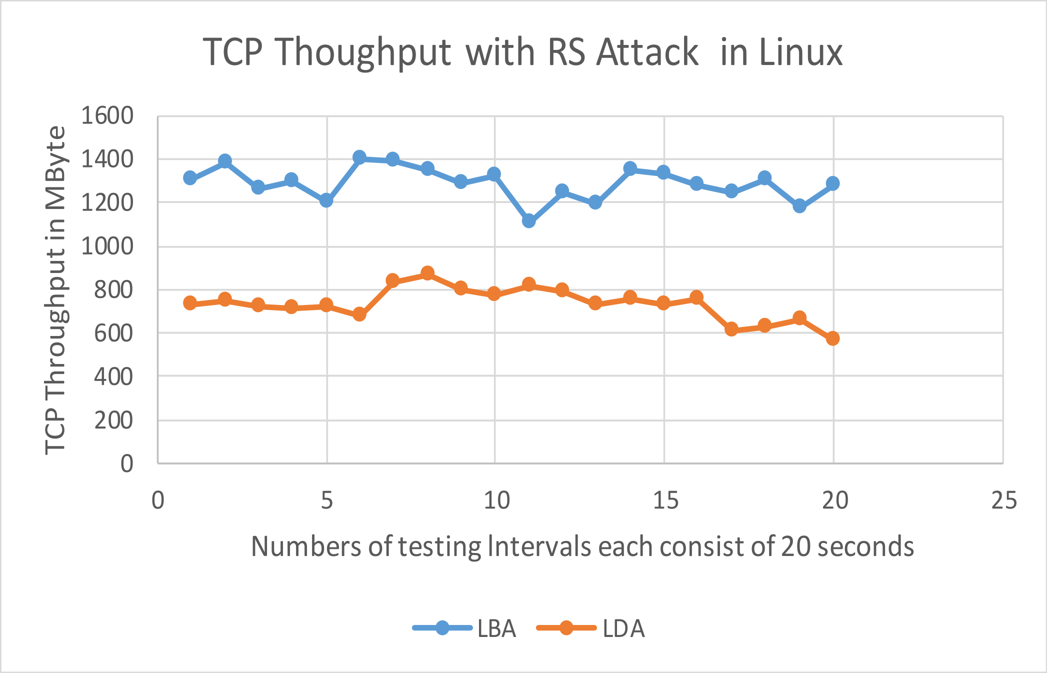


Figure 4.5. TCP Throughput with RS Attack in Linux

Refer to Figure 4.6 the TCP throughput under RA flooding attack decreased remarkably and was fluctuating gradually between (723 - 1010) Mbps. The percentage of decreasing was 33.68%.

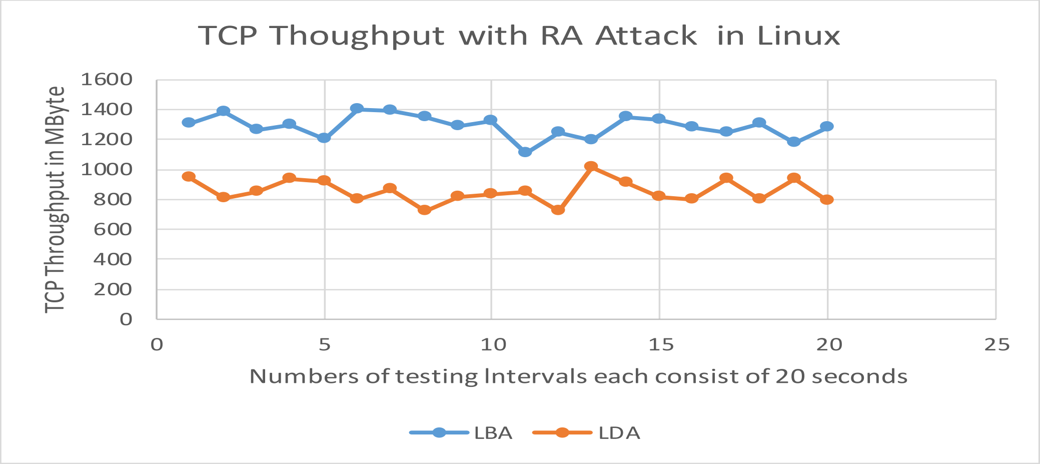


Figure 4.6. TCP Throughput with RA attack in Linux

Refer to Figure 4.7 the TCP throughput under NS flooding attack decreased sharply and was fluctuating gradually between (247 - 554) Mbps. The percentage of decreasing was 77.39%.

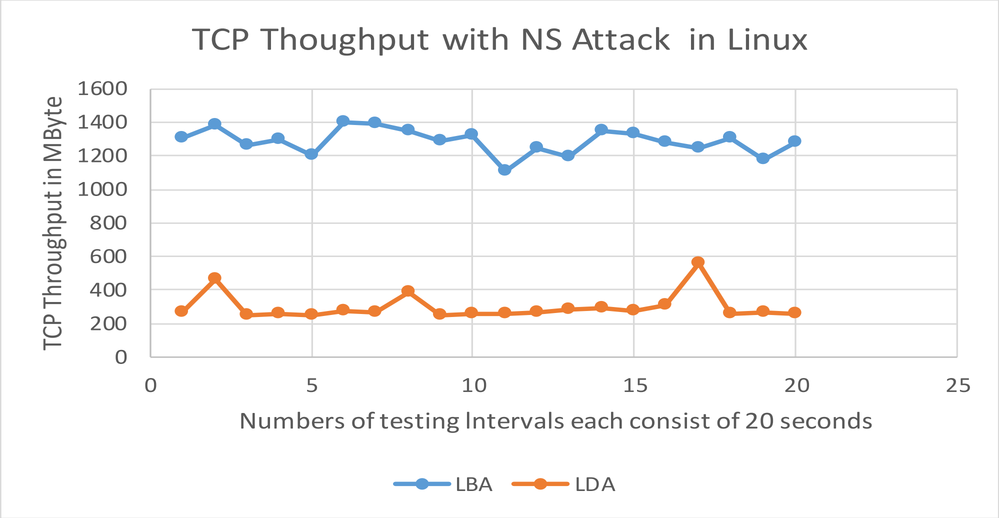


Figure 4.7. TCP Throughput with NS Attack in Linux

Refer to Figure 4.8 the TCP throughput under NA flooding attack decreased and was fluctuating modestly between (819 - 1010) Mbps. The percentage of decreasing was 29.58%.

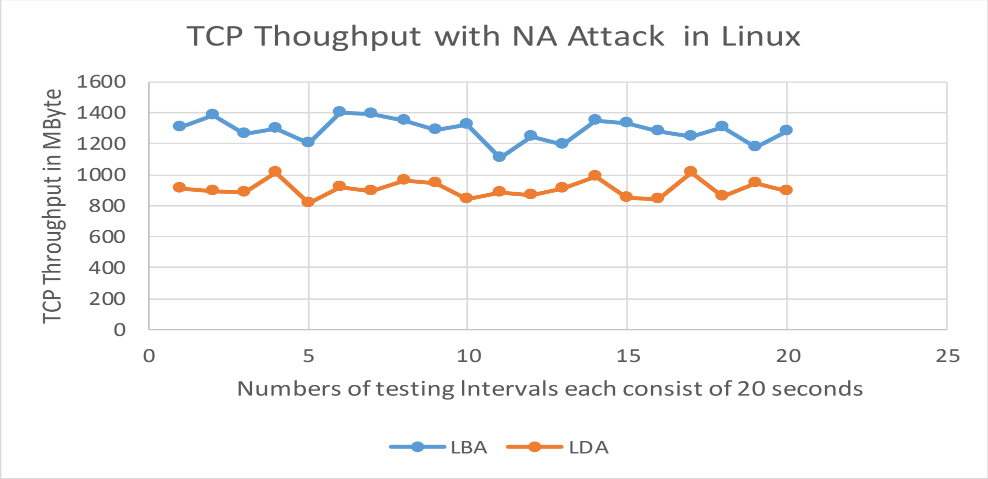


Figure 4.8. TCP Throughput with NA Attack in Linux

Refer to Figure 4.9 the CPU utilization in Windows before attack was fluctuating slightly between (23.43 – 59.37) Mbps. While the CPU utilization under RS flooding attack increased gradually until the 4th second then it fluctuated modestly and was fluctuating gradually between (60.93 – 75.38) %. The percentage of increasing was 10.02%.

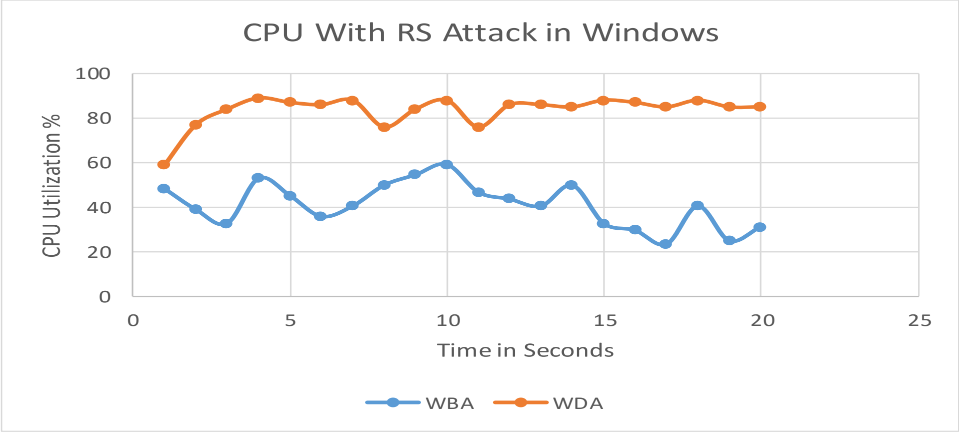


Figure 4.9. CPU with RS Attack in Windows

Refer to Figure 4.10, 4.11 and 4.12 the CPU utilization in Windows under RA, NS and NA flooding attacks respectively spiked to 100%. The percentage of increasing was 40.63%.

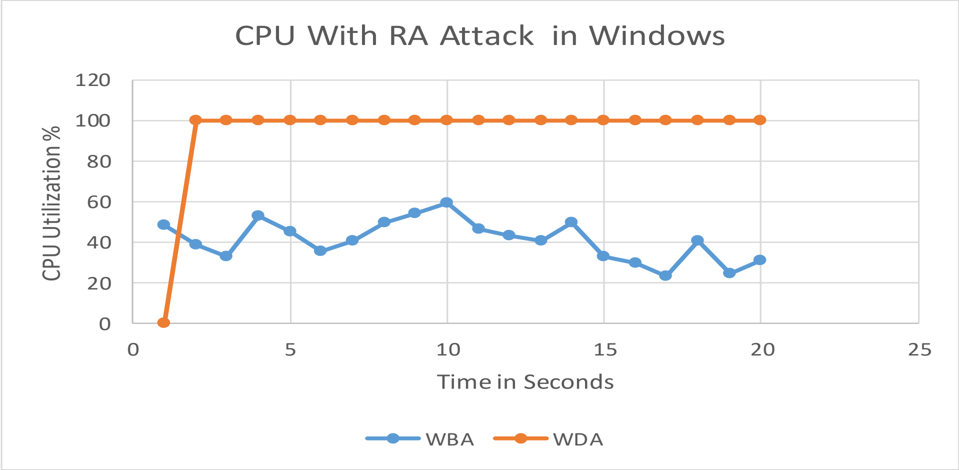


Figure 4.10. CPU with RA Attack in Windows

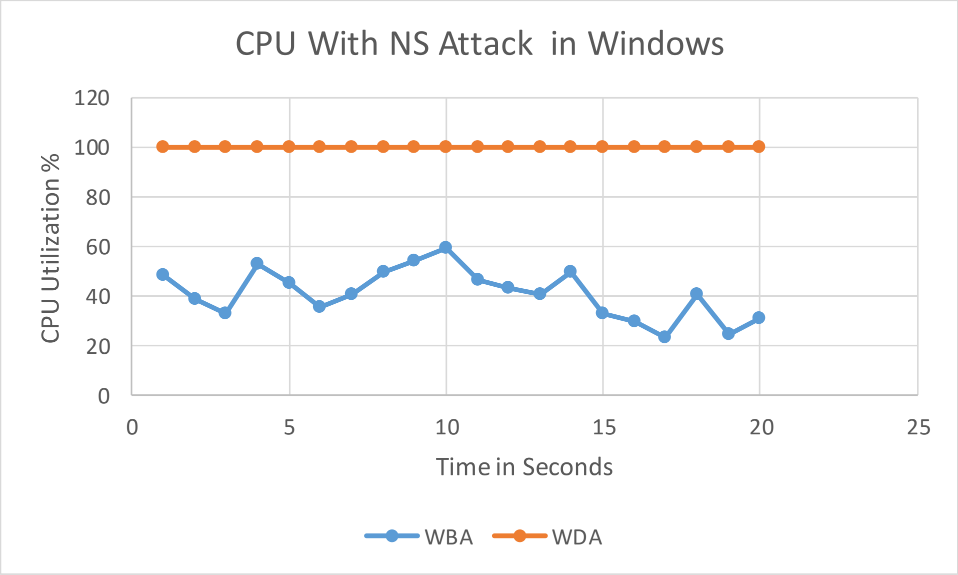


Figure 4.11. CPU with NS Attack in Windows

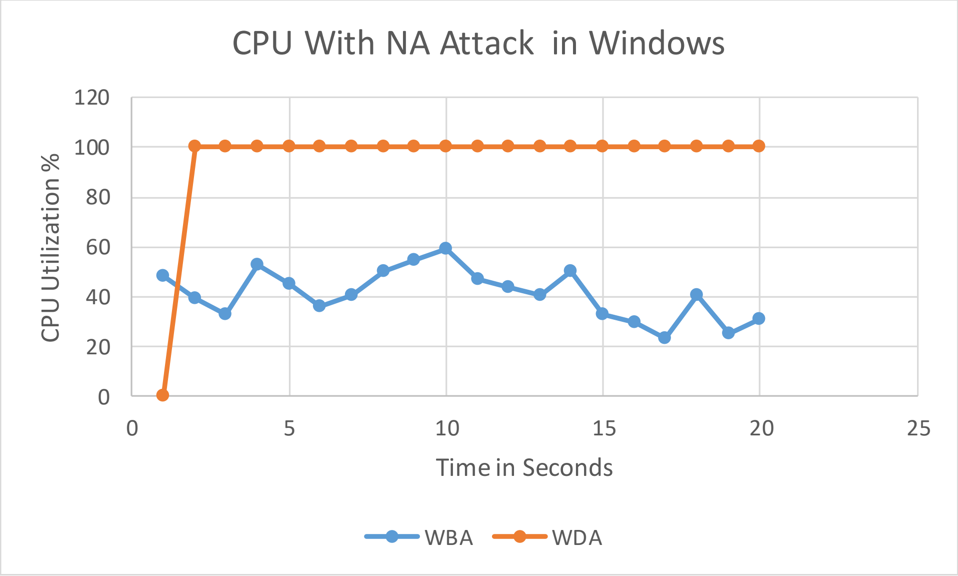


Figure 4.12. CPU with NA Attack in Windows

Refer to Figure 4.13 the CPU utilization in Linux before attack fluctuated slightly between (16 - 32) %. While the CPU utilization during RS flooding attack increased sharply reached 92% then it fluctuated marginally between (83 - 94) %. The percentage of increasing was 61.7%.

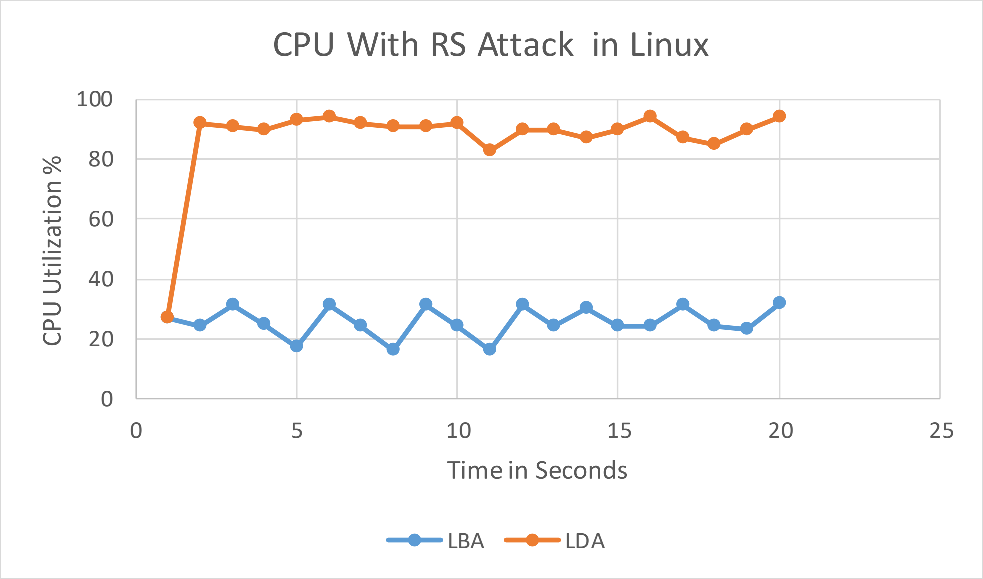


Figure 4.13. CPU with RS Attack in Linux

Refer to Figure 4.14 the CPU utilization in Linux during RA flooding attack leapt to 100%. The percentage of increasing was 70.95%.

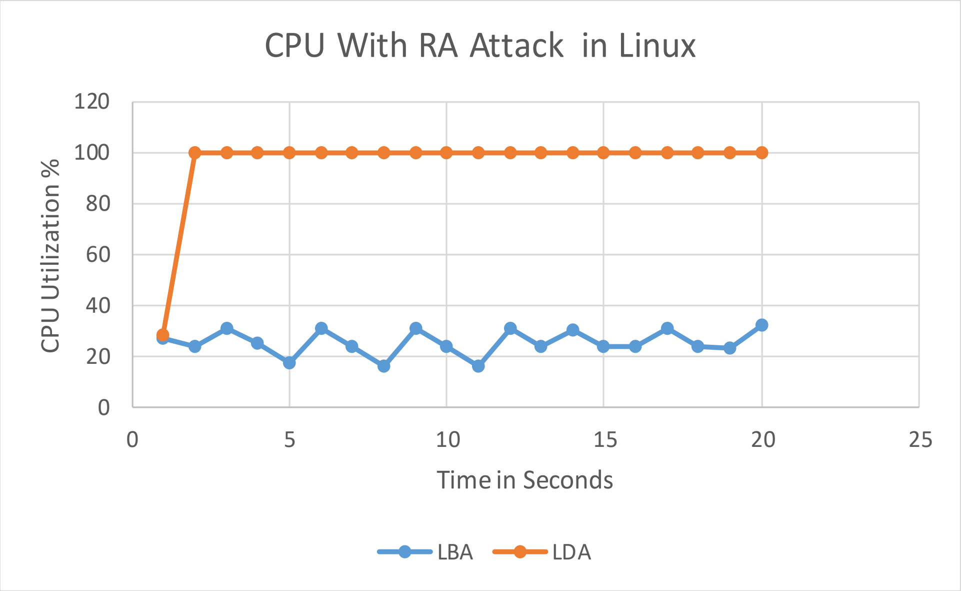


Figure 4.14. CPU with RA Attack in Linux

Refer to Figure 4.15 the CPU utilization in Linux during NS flooding attack leapt to 100%. The percentage of increasing was 71%.

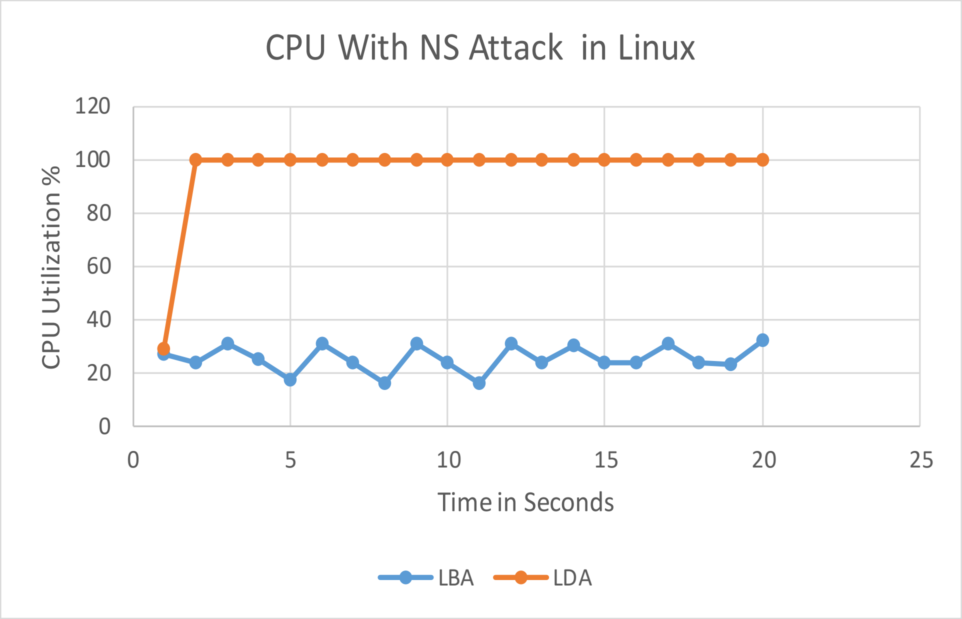


Figure 4.15. CPU with NS Attack in Linux

Refer to Figure 4.16 the CPU utilization in Linux during NA flooding attack increased sharply reached 91% then it fluctuated marginally between (86 - 97) %. The percentage of increasing was 61.7%.

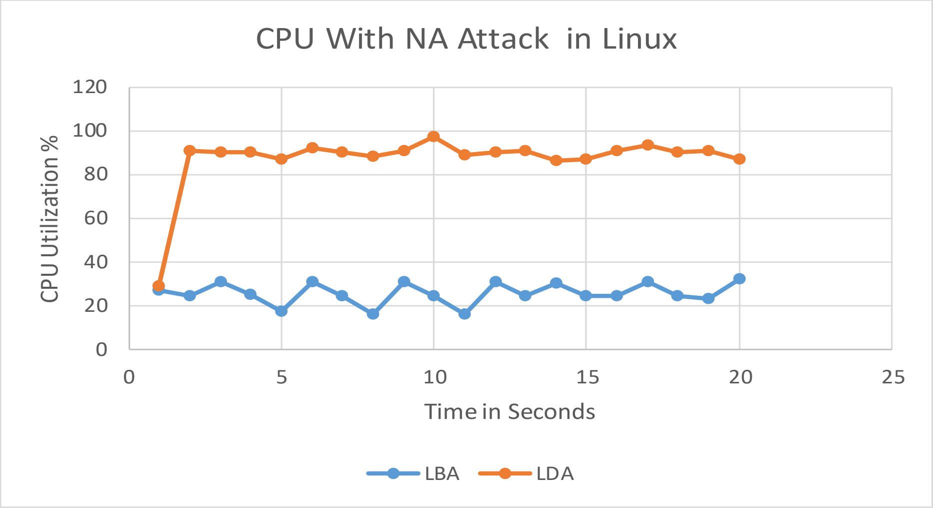


Figure 4.16. CPU with NA Attack in Linux

Refer to Figure 4.17 the RTT in Windows before attack was 0.87 ms as the firs value then fluctuated slightly between (0.398 – 0.875) ms. While during RS flooding attack the RTT increased remarkably and fluctuated slightly between (0.674 – 0.983) ms. The increasing percentage from the normal state was 52.12 %.

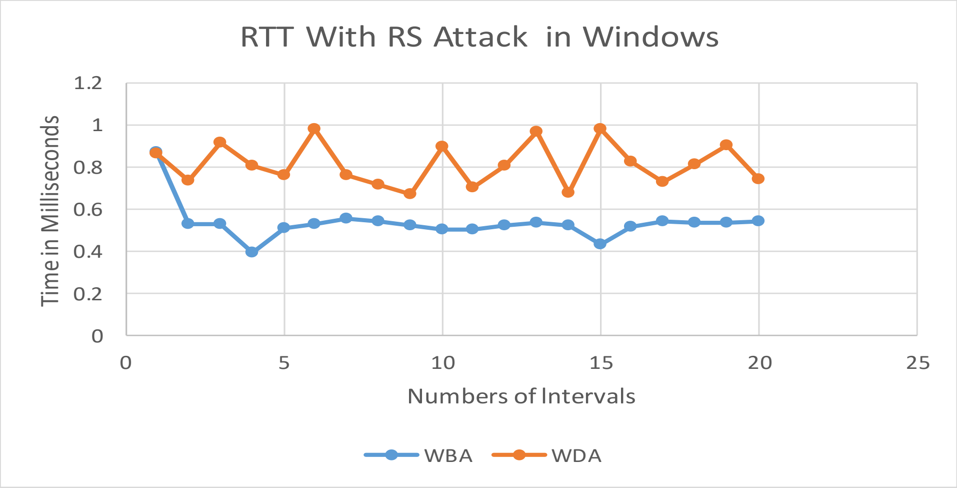


Figure 4.17. RTT with RS Attack in Windows

Refer to Figure 4.18, 4.19 and 4.20 showed the RTT in Windows under RA, NS and NA flooding attack respectively result with no buffer space available.

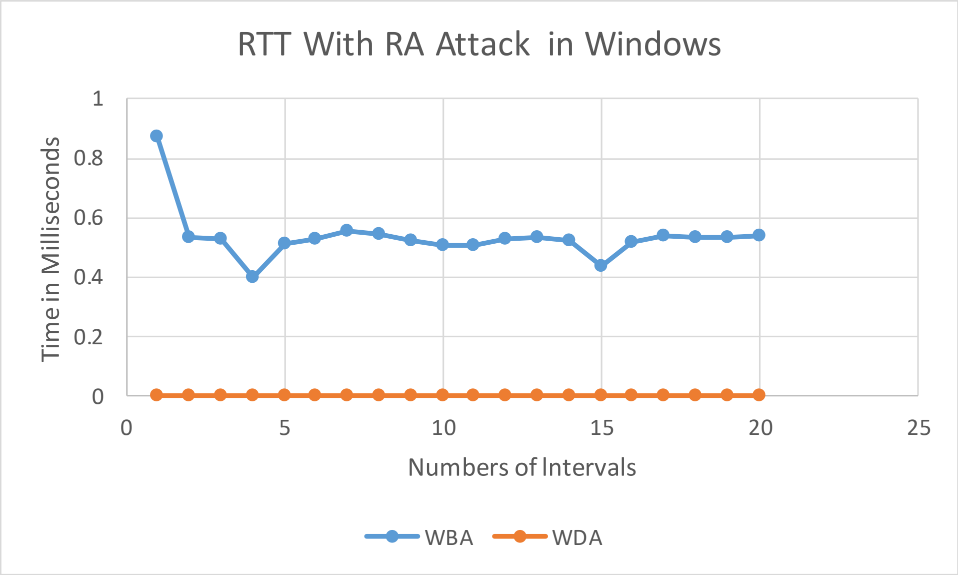


Figure 4.18. RTT with RA Attack in Windows

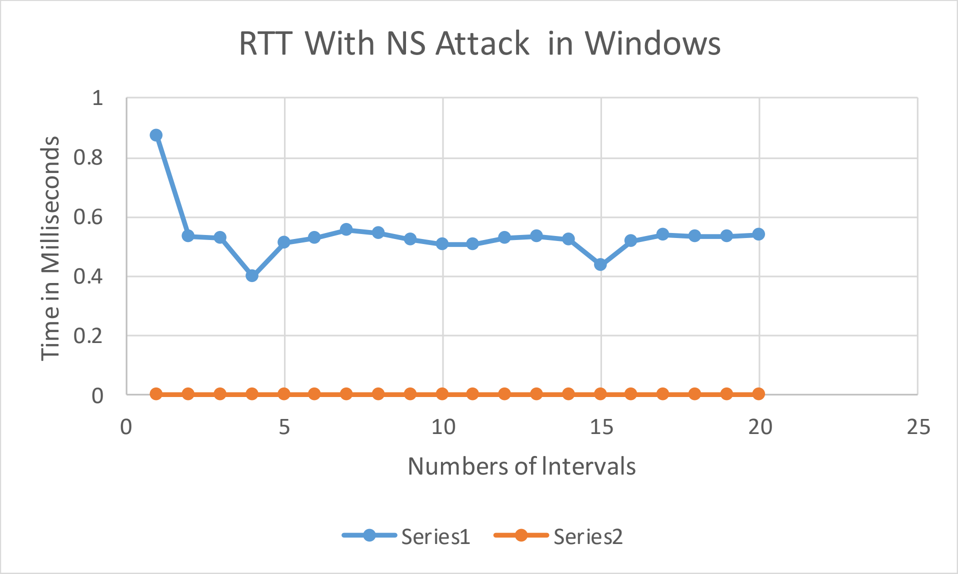


Figure 4.19. RTT with NS Attack in Windows

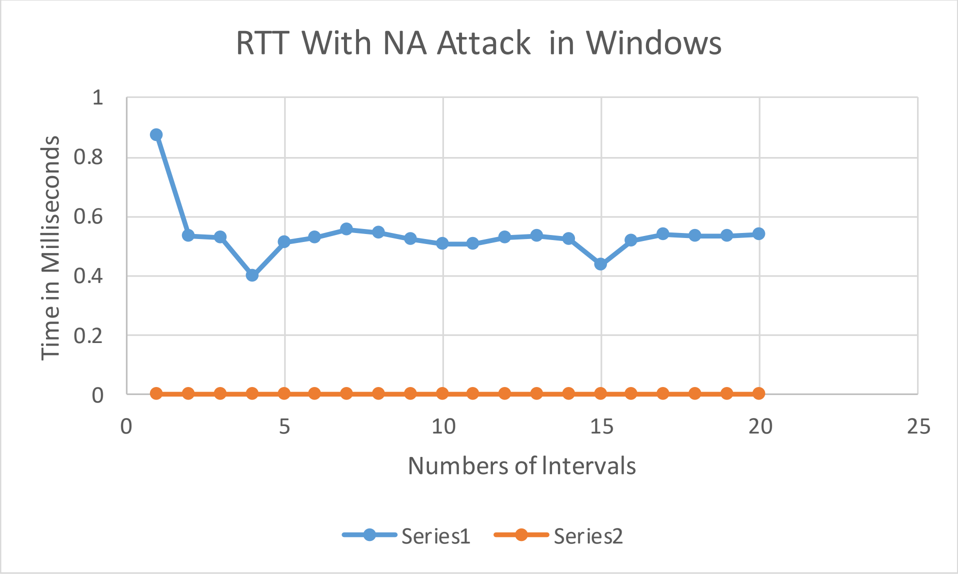


Figure 4.20. RTT with NA Attack in Windows

Refer to Figure 4.21 the RTT in Linux before attack fluctuated marginally between (0.32 – 0.41) ms. While during RS flooding attack the RTT did not show significant change just in 6th and 7th second the values increased slightly. It fluctuated slightly between (0.18 – 0.46) ms.

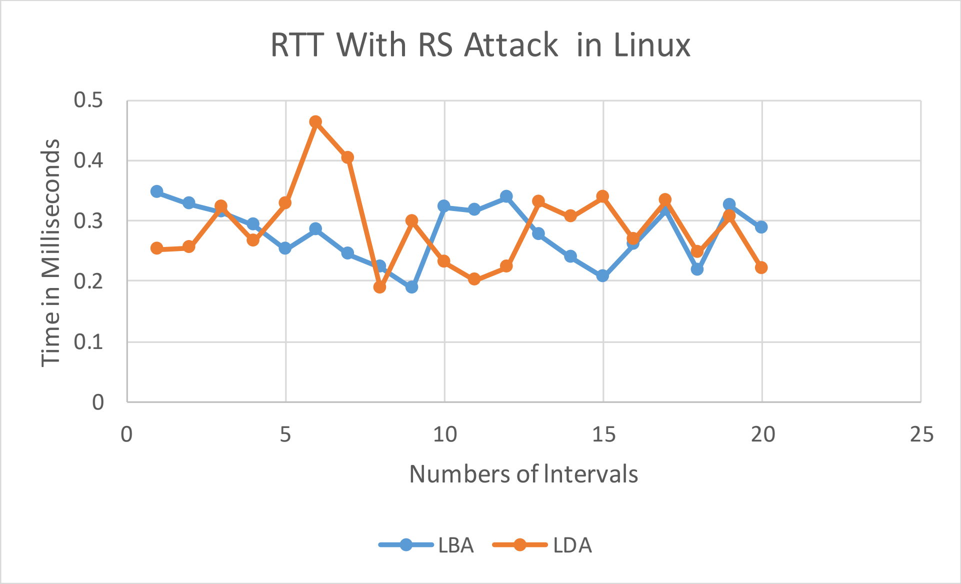


Figure 4.21. RTT with RS Attack in Linux

Refer to Figure 4.22 the RTT in Linux under RA flooding attack leapt to 1816 and fluctuated marginally between (1687 - 1865). The percentage of increasing was very high exceeded 100%, it multiplied 4549.34 times.

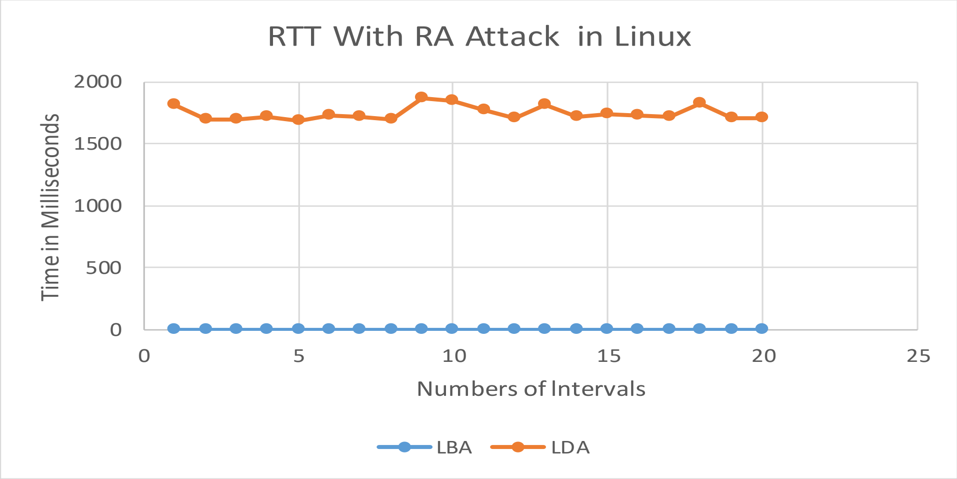


Figure 4.22. RTT with RA Attack in Linux

Refer to Figure 4.23 the RTT in Linux under NS flooding attack leapt to 1490 and fluctuated marginally between (1687 - 1856). The percentage of increasing was very high exceeded 100%, it multiplied 5256.77 times.

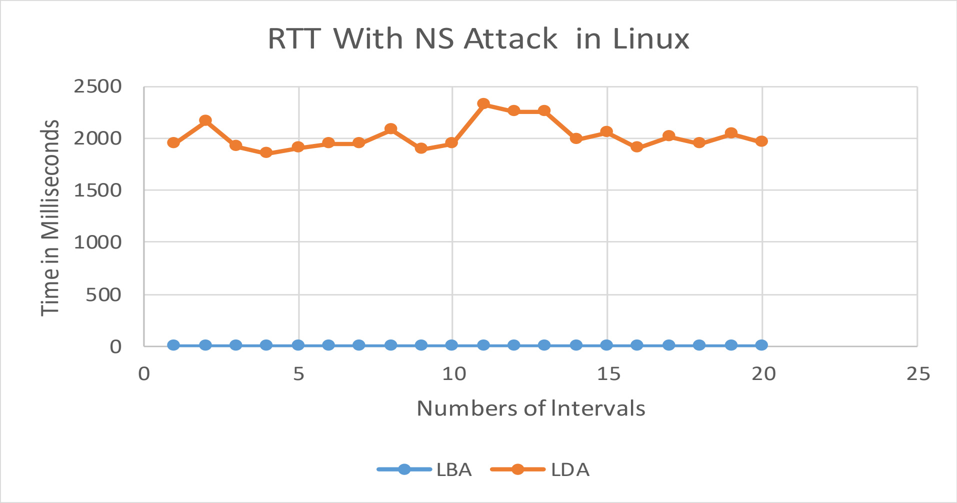


Figure 4.23. RTT with NS Attack in Linux

Refer to Figure 4.24 the RTT in Linux under NS flooding attack did not show significant change just in 16th and 19th second the values increased remarkably. It fluctuated slightly between (0.17 – 0.35) ms.

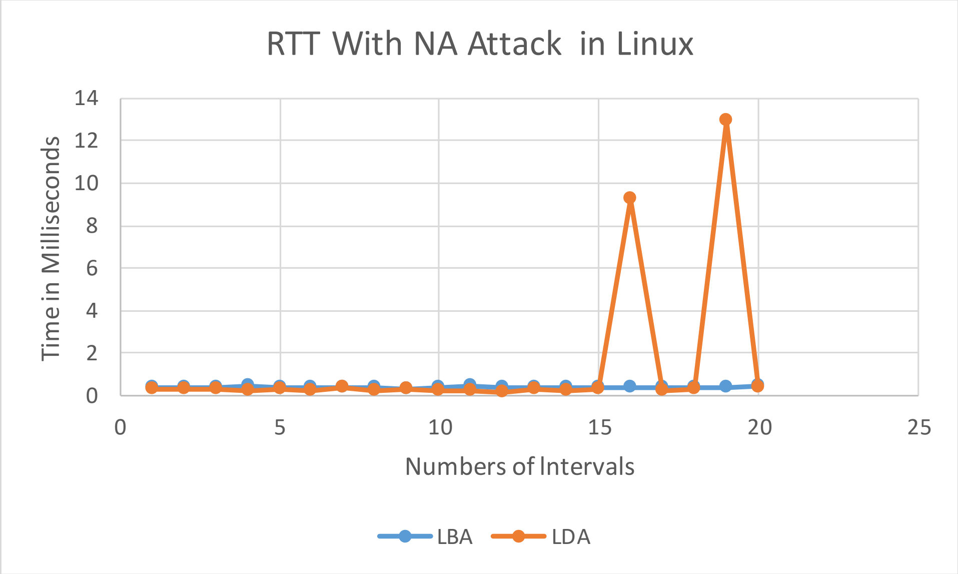


Figure 4.24. RTT with NA Attack in Linux

TCP Throughput

TCP throughput decreasing significantly in windows. During RS flooding attack, legitimate packets could be transmitted at lower rates the average of TCP throughput reached to 287.1 Mbps, while during RA, NS and NA flooding attacks the Legitimate packets could not be transmitted. While for Linux legitimate packets could be transmitted at lower rates, the average of the TCP throughput reached to 731.95, 853.1, 290.8 and 905.9 during RS, RA, NS and NA flooding attack respectively.

Round Trip Time (RTT)

RTT in windows increased 52.11% in RS flooding attack compare to the RTT before attack, while in the rest of attacks the RTT result was no buffer space available. On contrary in Linux RTT showed no change during RS and NA flooding attacks, while during RA and NS flooding attacks the increasing multiplied more than 4000 times compared to RTT before attack.

CPU Utilization

CPU utilization in widows during RS flooding attack increased 10% compare to the CPU utilization before attack and spiked to 100% in the rest of flooding attacks. In Linux CPU utilization in Linux increased in RS and NA flooding attack reached to 78% and 61.55% respectively and spiked to 100% in the rest of the flooding attacks.

Chapter Five
Conclusion and Recommendations

Chapter Five
Conclusion and Recommendations

5.1 Conclusion

NDP is the core protocol of IPv6 suite. When NDP was developed there is an assumption that mutual hosts within a subnet will trust each other. This trust formed vulnerabilities which make attackers join the network then exploit this trust and the lack of NDP security to attack the network. This may lead to a total system crash. A simulated network setup and corresponding configurations to evaluate the impacts of DoS attacks over NDP on Windows and Linux based operating systems were provided in this research. The impacts of each DoS attack were evaluated using TCP Throughput, RTT and CPU utilization metrics between monitoring and victims' computers before and during attacks. Overall, the results have shown that NDP is susceptible to DoS attack. The metrics which were chosen to evaluate the performance of the network showed reduction in the throughput, the latency in the network and resources were consumed. Both operating systems have been affected by DoS attack but the performance of Linux was better than Windows, it was mainly because of pre-allocation of fixed-sized memory buffers to avoid the overhead.

5.2 Recommendations

The Neighbor Discovery Protocol (NDP) vulnerabilities is the emerging area of research today. Overall, there are many security issues related to NDP that can be used by attackers to impact the legitimate communication of users. Although the NDP defined many rules for the nodes to send or receive NDP messages legitimately, there is no compulsive method to guarantee the node behaves normally. Therefore, malicious nodes can launch attacks illegally using NDP messages. To provide more evaluation, a special attention should be put in DoS attacks under IPv6. One of the vulnerabilities discovered allows Denial of Service Attack using the Duplicate Address Detection mechanism. In order to study and analyze this attack, an IPv6 security testbed should be designed and implemented.

References:

[1] A. S. Ahmed, R. Hassan, and N. E. Othman, “Improving security for IPv6 neighbor discovery,” *Proc. - 5th Int. Conf. Electr. Eng. Informatics Bridg. Knowl. between Acad. Ind. Community, ICEEI 2015*, pp. 271–274, 2015.

[2] A. S. A. Mohamed Sid Ahmed, R. Hassan, and N. E. Othman, “IPv6 Neighbor Discovery Protocol Specifications, Threats and Countermeasures: A Survey,” *IEEE Access*, vol. 5, pp. 18187–18210, 2017.

[3] R. Hassan, A. S. Ahmed, and N. E. Osman, “Enhancing security for IPv6 neighbor discovery protocol using cryptography,” *Am. J. Appl. Sci.*, vol. 11, no. 9, pp. 1472–1479, 2014.

[4] A. S. Ahmed, R. Hassan, and N. E. Othman, “Denial of service attack over secure neighbor discovery (SeND),” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 5, pp. 1897–1904, 2018.

[5] A. S. Ahmed, R. Hassan, N. E. Othman, N. I. Ahmad, and Y. Kenish, “Impacts evaluation of DoS attacks over IPv6 neighbor discovery protocol,” *J. Comput. Sci.*, vol. 15, no. 5, pp. 702–727, 2019.

[6] M. Huang, J. Liu, and Y. Zhou, “An improved send protocol against DoS attacks in mobile IPv6 environment,” *Proc. 2009 IEEE Int. Conf. Netw. Infrastruct. Digit. Content, IEEE IC-NIDC2009*, pp. 232–235, 2009.

[7] S. Ul and S. Manickam, “Denial of Service Attack in IPv6 Duplicate Address Detection Process,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 6, pp. 232–238, 2016.

[8] T. Zhang and Z. Wang, “Research on IPv6 Neighbor Discovery Protocol (NDP) security,” *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc.*, pp. 2032–2035, 2017.

[9] T. D. Hoang, “Deployment Ipv6 Over Ipv4 Network Infrastructure Deployment Ipv6 Over Ipv4 Network Infrastructure.”

[10] Silvia Hagen, *book ipv6 essentials*, 3rd ed. United States of America: O’Reilly Media, Inc., 2006.

[11] K. Batiha, “I MPROVING IP V 6 A DDRESSING T YPES AND S IZE,” vol. 5, no. 4, pp. 41–51, 2013.

[12] K. Bhamidipati, “A Comparative study of IPv6 Statistical Approach,” no. December, 2014.

[13] JOSEPH DAVIES, *Understanding IPv6*, 3rd ed. Microsoft Corporation, 2012.

[14] S. U. Rehman and S. Manickam, "Novel mechanism to prevent Denial of Service (DoS) attacks in IPv6 duplicate address detection process," *Int. J. Secur. its Appl.*, vol. 10, no. 4, pp. 143–154, 2016.

[15] S. Ul Rehman and S. Manickam, "Integrated framework to detect and mitigate denial of service (DoS) attacks on duplicate address detection process in IPv6 link local communication," *Int. J. Secur. its Appl.*, vol. 9, no. 11, pp. 77–86, 2015.

[16] S. U. Rehman and S. Manickam, "Rule-based mechanism to detect Denial of Service (DoS) attacks on Duplicate Address Detection process in IPv6 link local communication," *2015 4th Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2015*, no. September, 2015.

[17] R. C. Baishya, N. Hoque, and D. K. Bhattacharyya, "DDoS attack detection using unique source IP deviation," *Int. J. Netw. Secur.*, vol. 19, no. 6, pp. 929–939, 2017.

[18] F. Najjar, M. Kadhum, and H. El-taj, "Neighbor Discovery Protocol Anomaly Detection Using Finite State Machine and Strict Anomaly Detection."

[19] S. U. Rehman and S. Manickam, "Significance of duplicate address detection mechanism in IPv6 and its security issues: A survey," *Indian J. Sci. Technol.*, vol. 8, no. 30, pp. 1–8, 2015.

[20] E. S. Han and A. goleman, daniel; boyatzis, Richard; Mckee, "RFC 2461," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[21] Dr. Juliansyah Noor, "RFC 4862," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[22] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," *Req. Comments*, vol. 6, no. March, pp. 1–93, 1998.

[23] "RFC 1256," no. September, pp. 1–19, 1991.

[24] Dr. Juliansyah Noor, "RFC 3756," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[25] A. S. Ahmed, N. H. A. Ismail, R. Hassan, and N. E. Othman, "Balancing performance and security for IPv6 neighbor discovery," *Int. J. Appl. Eng. Res.*, vol. 10, no. 19, pp. 40191–40196, 2015.

[26] S. Praptodiyono, I. H. Hasbullah, M. Anbar, and R. K. Murugesan,

"Improvement of Address Resolution Security in IPv6 Local Network using Improvement of Address Resolution Security in IPv6 Local Network using Trust-ND," no. September, 2015.

[27] G. J. Song and Z. Z. Ji, "Novel duplicate address detection with hash function," *PLoS One*, vol. 11, no. 3, pp. 1–19, 2016.

[28] S. Praptodiyono, I. H. Hasbullah, M. M. Kadhum, R. K. Murugesan, C. Y. Wey, and A. Osman, "Improving Security of Duplicate Address Detection on IPv6 Local Network in Public Area," *Proc. - AMS 2015 Asia Model. Symp. 2015 - Asia 9th Int. Conf. Math. Model. Comput. Simul.*, pp. 123–128, 2016.

[29] K. Perumal and M. J. P. Jeya Priya, "Trust based security enhancement mechanism for neighbor discovery protocol in IPV6," *Int. J. Appl. Eng. Res.*, vol. 11, no. 7, pp. 4787–4796, 2016.

[30] J. L. Shah, "A novel approach for securing IPv6 link local communication," *Inf. Secur. J.*, vol. 25, no. 1–3, pp. 136–150, 2016.

[31] J. L. Shah and J. Parvez, "Optimizing Security and Address Configuration in IPv6 SLAAC," *Procedia Comput. Sci.*, vol. 54, pp. 177–185, 2015.

[32] A. K. Al-Ani, G. Mousa, and F. Qays Kamal, "Denial of Service Attack on Neighbor Discovery Protocol Processes in the Network of IPv6 Link-Local," *Int. J. Electr. Electron. Eng. Telecommun.*, pp. 1–5, 2019.