



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

PROGRAMA DE MAESTRIA Y DOCTORADO EN  
INGENIERIA

FACULTAD DE INGENIERÍA

NOMBRE DE LA TESIS

**T E S I S**

QUE PARA OPTAR POR EL GRADO DE:

**MAESTRO EN INGENIERIA**

INGENIERÍA ELECTRICA – TELECOMUNICACIONES

P R E S E N T A:

**ING. CUAUHEMOC CARLON CARLON**

TUTOR:

**Dr. ENRIQUE DALTABUIT GODAS**

2009





Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## **JURADO ASIGNADO:**

Presidente: Dr. Javier Gómez Castellanos

Secretario: Dr. Víctor Rangel Licea

Vocal: Dr. Enrique Daltabuit Godas

1er. Suplente: Dr. Ramón Gutiérrez Castrejón

2do. Suplente: Dr. Gerardo Vega Hernandez

Lugar o lugares donde se realizó la tesis:

Laboratorio de Redes Edificio Valdez Vallejo 3er piso Facultad  
de Ingeniería CU

## **TUTOR DE TESIS:**

---

**FIRMA**

## Agradecimientos

A María Cristina Susana Carlon Vargas

Mi madre, que como fiel guardián a estado siempre al pendiente de cada paso que doy, gracias por nunca estar demasiado cansada, con tal de brindarme todo tu apoyo de manera incondicional, pues jamás me has dejado caer, no importando que tan duros o difíciles se hayan visto y sentido los tiempos, a ti, que si no fuera por tus desvelos, jamás hubiese siquiera soñado que algún día llegaría a hacer una Maestría y menos alcanzar los estudios Doctorales, gracias te doy madre, pues sólo gracias a ti, ha llegado este día. A ti madre que en los momentos propicios has sido mas que madre una gran amiga con la que siempre he contado en todo momento, a ti ...

GRACIAS, MUCHAS GRACIAS

A Yessica Gisela Arredondo Guzmán

A ti, te agradezco, mi dulce y fiel compañera, porque entiendes lo que persigo, porque desde que estás conmigo, te encargaste de hacerme la vida maravillosa, siempre pendiente de lo que me hace falta, te doy las gracias por tus palabras de aliento, por tu compañía, por tu amor y cariño.

Para siempre juntos, pues te querré siempre...

A mis maestros

A todos y a cada uno de ellos por dedicarme su tiempo e instruirme con tanta dedicación y paciencia, les agradezco por haberme formado y dado más que conocimientos, de corazón mis mas sinceros agradecimientos.

A la UNAM

Mi agradecimiento mas profundo por que a ella llegué sediento de conocimiento y ella tan generosa, no sólo me lleno de conocimiento, sino de todas mis virtudes, mi agradecimiento eterno porque a pesar de que el tiempo pase, yo seré orgullosamente universitario toda mi vida.

Por mi Raza Hablará el Espíritu...

## Índice de tablas

4.1. Serie de Tiempo	30
4.2. Serie de Tiempo Suavizada	31
4.3. Serie Residual	32
4.4. Variación Estacional	35
4.5. Predicción Lineal	36
4.6. Serie de Tiempo	39
4.7. Serie de Tiempo Suavizada	39
4.8. Serie Residual	39
4.9. Variación Estacional	40
4.10. Predicción Lineal	40
4.11. Promedios de pendientes negativas y (PPn)	43
4.12. Rangos de Calificación	45
4.13. Predicción Lineal y Error de Predicción	47
4.14. Rangos de Calificación	51
1. Parámetros del Cálculo de Tendencia y Serie Residual	93
2. Operaciones de la tabla $W(h)$	94
3. Parámetros	94
4. Cantidad de operaciones	95
5. Cantidad de operaciones y Comparaciones	99
6. Total de operaciones y Comparaciones	99
7. Tipos de Operaciones	99

## Índice de figuras

2.1. Una Típica Red DDoS	9
2.2. Flujo de Información en el Escenario DDoS	9
2.3. Modelos de los Ataques DDoS	10
2.4. Jerarquía del Ataque DDoS	12
2.5. DoS VS DDoS	13
2.6. Creación de una Botnet para Renta	14
3.1. Configuración de un Ataque DDoS	20
3.2. Analogía de la Transmisión-Recepción con el Área de un Cuadrado	21
3.3. DDoS Sobre un Cliente TCP	23
3.4. Cliente 2 UDP ante el Ataque DDoS	24
3.5. Cliente 2 TCP ante el Ataque DDoS	25
4.1. Clientes Comunicándose con un Servidor sin Ataque DDoS	29
4.2. Serie de Tiempo del Servidor sin Ataque DDoS	30
4.3. Serie de Tiempo Suavizada sin Ataque DDoS	32
4.4. Gráfica de la Recta de la Tendencia	34
4.5. Predicción VS Segundo 19	37
4.6. Predicción Lineal	41
4.7. Predicción Lineal bajo DDoS	42
4.8. División en Cuartos para Obtener (P Pnp)	44
4.9. Gráfica PPs	48
4.10. Predicción sobre TCP flooding	49
5.1. Posición Estratégica del Horizonte de Sucesos	56
5.2. Bifurcación del Flujo	57
5.3. Clasificador del flujo	57
5.4. Diagrama de Flujo del Clasificador	58
5.5. Estructura del Despachador	59
5.6. Diagrama de Flujo del Despachador UDP	60
5.7. Estructura de la Tabla 1 y la Tabla 3	61
5.8. Estructura de la Tabla 2	63
5.9. Diagrama de Flujo del Despachador TCP	65
5.10. Esquema Completo del Horizonte de Sucesos	66
5.11. Búsqueda con Apuntadores Adaptables	71
1. Serie de Tiempo en Excel	101
2. Suavizado de la Serie de Tiempo en Excel	102
3. Calculo del Promedio en Excel	103
4. Calculo de las demás Operaciones en Excel	104
5. Calculo de los Parámetros a y b en Excel	104
6. Construcción de la Tabla Residual en Excel	105
7. Promedios de cada Fila de la Tabla Residual	105
8. Calculo de los Parámetros de Error de Predicción	106

## Índice general

1. Introducción	1
1.1. Definición del Problema	1
1.2. Objetivo	1
1.3. Contribución	1
1.4. Descripción del contenido	2
2. El Ataque DoS y sus Derivaciones	3
2.1. Introducción	3
2.2. Definición del Ataque de DoS	4
2.3. Taxonomía del Ataque DoS	4
2.4. Definición del Ataque DDoS	6
2.5. Tres Etapas para un Ataque DDoS	8
2.6. Síntomas de un Ataque DDoS	11
2.7. Defensas Contra el Ataque de DDoS	12
2.8. Recomendaciones	13
2.9. Objetivos de los Atacantes	14
2.10. Resumen	15
2.11. Conclusiones del Capítulo	16
3. Modelado del Ataque DDoS	17
3.1. Introducción	17
3.2. Modelado del Ataque DDoS Bajo NS-2	18
3.2.1. Simulación de Redes de Datos	18
3.2.2. Escenario a Simular en NS-2	19
3.2.3. Topología a Nivel Transporte de la Simulación	20
3.3. Conclusiones del Capítulo	26
4. Aplicación de la Predicción Lineal	27
4.1. Introducción	27
4.2. Análisis del Muestreo	29
4.3. Predicción sin DDoS	31
4.4. Predicción bajo UDPflooding	39
4.5. Predicción bajo TCPflooding	47
4.6. Conclusiones del Capítulo	53
5. Minimización del ataque DDoS	54
5.1. Introducción	54
5.2. Descripción del la Propuesta de Tesis	55
5.3. Diseño de un Horizonte de Sucesos	67
5.4. Optimización de Búsqueda en las Tablas	70
5.5. Conclusiones del Capítulo	74
6. Conclusiones de la Propuesta	75
Apéndice A Código TCL del UDP-flooding	78
Apéndice B Código TCL del TCP-flooding	91
Apéndice C Tiempo de Ejecución	92
Apéndice D Uso de Excel	100
Glosario	107
Bibliografía	115

## Resumen

Existen otros trabajos donde se aplican las técnicas de Predicción Lineal en el área de Seguridad Informática. La Predicción Lineal a encontrado aplicación un muchas disciplinas y entre ellas, la de la seguridad en redes.

La Predicción es una pieza angular en este trabajo ya que sin ella no se tendría la flexibilidad de la detección oportuna del comienzo de un ataque DDoS.

Al tener un aviso de comienzo de ataque, justo a unos milisegundos de haber comenzado, es prudente tener un mecanismo de como minimizarlo. Es por ello que en este trabajo se habla de como diseñar un mecanismo de minimización del ataque DDoS en las variantes TCP flooding y UDP flooding.

El mecanismo de minimización explota la fortaleza del ataque mismo (el ser distribuido), para lograr contrarrestar el daño que pudiera ocasionar dicho ataque sobre las dos premisas más importantes:

- Conservar intactas las conexiones activas justo antes y en momentos donde el ataque este presente.
- La NO denegación del servicio cuando el ataque DDoS este en curso.

Al lograr conservar estas dos premisas, el ataque pierde efecto, al no poder alcanzar su objetivo de denegación del servicio y con ello se ve frustrado el intento del atacante por perjudicar a una determinada organización.



## Summary

Inside of security network, the Linear Prediction has taken a paper very important.

In this proposal, we use the Linear Prediction to detect a DDoS attack.

The Linear Prediction is the main piece, when is necessary to detect a DdoS attack just when it's beginning.

Alarm up allows activate a mechanism that minimizes DDoS attack, in this paper describes how to design a mechanism to make light of effects from this attack.

When a server is under DDoS attack in any of two variants: TCP flooding and UDP flooding. The most important is to protect two premises:

- The active connections most to survive during DDoS attack.
- The service offered by the server must be not refused when the server is under DDoS attack.

The minimization Mechanism exploit the strength from DDoS attack (to be Distributed) in order to counteract the damage than could cause this attack.

When these premises are achieved, the attack loss it's effectiveness, because it can't cause than the server denials it's services to any client.

# Capítulo 1

## Introducción

### Definición del Problema

En este trabajo se aborda una descripción general de las variantes del ataques DoS. Sin embargo, la variante de interés, es la del ataque DDoS, la cual será el tema central de este trabajo. Se utilizará el concepto de Predicción Lineal, para detectar dicha variante y posteriormente se describirá un método para minimizar el mencionado ataque; también a manera de ejemplo se llevará a cabo un diseño del método empleado para observar sus componentes más importantes.

### Objetivo

El objetivo de este trabajo es diseñar un mecanismo que permita minimizar los efectos del ataque DDoS sobre servidores de comercio electrónico en Internet. Sin embargo, este mecanismo no se debe de tener funcionando siempre, debido a un gasto de recursos innecesario, es por ello, que se utilizará la Predicción Lineal para detectar el comienzo de un ataque de esta naturaleza.

### Contribución

Una solida propuesta que minimice los efectos del ataque DDoS de manera puntual, dejando a un lado los tratamientos tradicionales a este problema, pues ninguna de estas soluciones da un tratamiento efectivo a los efectos de dicho ataque.

El diseño del dispositivo propuesto, es relativamente simple y además económico, así cualquier fabricante de dispositivos de interconexión de redes puede implantarlo.

Con esto se pretende reducir las pérdidas económicas y otras que pudieran generarse debido a los efectos del ataque DDoS.

### **Descripción del contenido**

En el capítulo 2, se habla de los diferentes tipos de ataque DoS y sus principales características, así como de las motivaciones más importantes para llevar a cabo un ataque de esta naturaleza y los daños que estos provocan.

En el capítulo 3, se abordará el modelado del ataque DDoS bajo el simulador NS-2, con la idea de conocer sus características más importantes, como la estructura jerárquica del ataque DDoS, que o quienes intervienen en esta estructura y los efectos adversos y cuantitativos en relación al funcionamiento de un determinado servidor bajo el mencionado ataque.

En el capítulo 4, se aplicarán las técnicas de la Predicción Lineal con la idea de NO predecir un ataque de esta naturaleza, sino con el objetivo de saber a la brevedad posible, cuando empieza dicho ataque, con el fin de ejecutar algoritmos que activen una alarma y ésta a su vez, al ser activada, que active el mecanismo de minimización del ataque DDoS en las dos variantes tratadas de este trabajo.

En el capítulo 5, se describe el diseño del mecanismo para minimizar el ataque DDoS en combinación con las técnicas de Predicción Lineal, así ambos tanto el mecanismo de minimización, como las técnicas de Predicción Lineal trabajan en conjunto, para disminuir los daños ocasionados por este ataque.

Por último, en el capítulo 6 se habla de las conclusiones a las que se llegó a lo largo de este trabajo, dejando como antecedente que lo expuesto aquí, es un sistema y como tal se debe de someter a un tratamiento, cada vez que se que el ataque mute, para conservar su efectividad.

## Capítulo 2

### El Ataque DoS y sus Derivaciones

#### Introducción

Los ataques Distribuidos de Denegación de Servicio, son un desarrollo relativamente reciente, y constituyen una gran amenaza para la seguridad de Internet en nuestros días.

Algunas de las grandes compañías, que proporcionan servicios a través de Internet, han sido víctimas de ataques de este tipo. Por nombrar algunas: Yahoo, Amazon, Buy.com, eBay y CNN no fueron accesibles durante horas, lo que ocasionó grandes pérdidas económicas.

El ataque de denegación de servicios distribuido DDoS, consistente en la realización de un ataque conjunto y coordinado entre varios equipos (que pueden ser cientos o miles) hacia un servidor víctima. La particularidad de este ataque, es el hecho de que el ataque proviene de diferentes partes del mundo, haciendo imposible cerrar la ruta de donde proviene el mismo, ya que no sólo es una, son varias, dejando entre las únicas opciones, desconectar el Servidor de la Red y esperar a que el ataque cese.

Normalmente los ataques se llevan a cabo por varias oleadas. Pueden durar un par de minutos o incluso días, como ha sucedido en casos verídicos. Esto es posible gracias a ciertos tipos de malware que permiten obtener el control de un gran número de máquinas en todo Internet, en donde un atacante ha instalado previamente en ellas, dicho malware, ya sea por intrusión directa o mediante algún gusano. Los DDoS consiguen su objetivo gracias a que agotan el ancho de banda de la víctima y sobrepasan la capacidad de procesamiento del Servidor o de los Routers, consiguiendo que los servicios ofrecidos por la máquina atacada no puedan ser prestados.

A consecuencia de esto, el servidor es puesto fuera de línea voluntariamente por los administradores y proveedores del servicio de colocación de servidores, debido al alto gasto que genera el Ancho de Banda con el ISP (Proveedor de Servicios de Internet), cuyo costo de dinero es considerado por las organizaciones.

## Definición del Ataque de DoS

Se pueden definir los ataques DoS (Denegation of Service) como la apropiación exclusiva de un recurso (ej. CPU, Ancho de Banda) o servicio (ej. información del servidor, etc.) con la intención de evitar cualquier acceso de terceros. También se incluyen en esta definición los ataques destinados a colapsar un recurso o sistema con la intención de destruir el servicio o recurso.

Los ataques DoS nacen como una consecuencia natural de la propia arquitectura de Internet. No es necesario tener grandes conocimientos para realizar este tipo de ataques y no es tan arriesgado como realizar un ataque directo contra un servidor.

Una característica de este tipo de ataques es la utilización de un equipo intermedio para luego poder borrar el rastro del atacante.

## Ejemplo del Ataque DoS:

Si un servidor tiene un ancho de banda de 1 Mbps y un usuario tiene un ancho de banda de 30 Mbps, este usuario podría denegar el servicio del servidor haciéndole muchas peticiones de algún servicio y agotando su ancho de banda. Así cualquier intento por parte de un usuario legítimo de acceder a algún tipo de servicio del servidor, queda totalmente imposibilitado.

## Taxonomía del Ataque DoS

### Mail bombing:

Se trata del primer ataque de esta naturaleza. El colapso de servidores WWW, mediante el envío masivo de tramas a una máquina, hasta saturar el servicio objetivo de la denegación.

### Los Tipos son:

Mail bombing: El colapso se produce enviando multitud de correos electrónicos a la vez.

Log bombing: El evento se produce con multitud de conexiones a la vez, de manera que produzca un overflow en el servidor.

### Smurfing:

Este sistema de ataque se basa en transmitir a la red una trama ICMP correspondiente a una petición de ping. Esta trama lleva como dirección de origen la dirección IP de la víctima (usando IP Spoofing) y como dirección de destino la dirección broadcast de la red atacada. De esta forma todos los equipos de la red contestan a la víctima de tal modo que pueden llegar a saturar su ancho de banda, es por ello que también se le conoce como un ataque de desbordamiento del buffer, está diseñado para agobiar al `\em software \rm` que corre en el sistema objetivo.

### SYN Flood:

El sistema atacante utiliza una IP inexistente y envía multitud de tramas SYN de sincronización a la víctima. Como la víctima no puede contestar al peticionario (porque su IP es inexistente)

las peticiones llenan la cola de tal manera que las solicitudes reales no puedan ser atendidas. Este ataque de sincronización (SYN) explota el \$handshake\$ de tres pasos del protocolo TCP.

DoS por Ruteo:

La mayoría de los protocolos de enrutamiento como RIP (Routing Information Protocol) o BGP (Border Gateway Protocol) carecen de autenticación, o tienen una muy sencilla.

Se trata por tanto de un escenario perfecto para que un atacante pueda alterar las rutas correctas y falsificando su IP origen, crear una condición DoS. Las víctimas de estos ataques verán como su tráfico se dirige, por ejemplo, hacia un agujero negro: a una red que no existe.

DoS Sobre DNS's:

Los ataques DoS sobre servidores de nombres de dominios (DNS) intentan convencer al servidor DNS, por ejemplo, para almacenar direcciones falsas: cuando un servidor DNS realiza una búsqueda, el atacante puede re direccionar el tráfico a su propio servidor o bien a un agujero negro.

Denegación del Servicio Distribuido DDoS:

Consistente en la realización de un ataque conjunto y coordinado entre varios equipos (que pueden ser cientos o miles ubicados en todas partes del mundo) hacia un servidor víctima u objetivo.

Los ataques DDoS (Distributed Denial of Service) consiguen su objetivo gracias a que agotan el ancho de banda de la víctima y sobrepasan la capacidad de procesamiento ya sea del servidor o de los Routers, consiguiendo que los servicios ofrecidos por la máquina atacada no puedan ser prestados.

Debido a su complejidad con respecto a las otras variantes del DoS, esta variante será el objeto de estudio de este trabajo.

En este trabajo se propondrá la forma de como detectar el ataque DDoS justo cuando comienza, utilizando técnicas basadas en la Predicción Lineal, y posteriormente se hablará de una solución sólida y robusta, para poder dar tratamiento o al menos minimizar lo más que se pueda, los efectos de este problema.

En la siguiente sección se describe a detalle el ataque DDoS y se hace una comparación con respecto al DoS original.

Definición del Ataque DDoS

Un ataque DDoS o ataque de denegación de servicios distribuido, es una variante del ataque DoS, consiste en un ataque simultáneo de una red de malware, donde cada malware corre sobre un equipo diferente, de esta forma todos los equipos con malware corriendo dentro ellos atacan a un servidor víctima.

La diferencia de este ataque con respecto al DoS es que el ataque proviene de diferentes partes del mundo, haciendo imposible cerrar las rutas del atacante ya que no sólo es una, dejando como única opción desconectar el servidor de la red y esperar a que el ataque cese.\

Normalmente los ataques DDoS, se llevan a cabo en oleadas, esto es posible gracias a las características del tipo de malware empleado para dicho fin, el cual permite obtener el control directo de los equipos.

Los atacantes DDoS consiguen su objetivo gracias al agotamiento del ancho de banda de la víctima y sobrepasando la capacidad de procesamiento de los Routers.

Las máquinas infectadas por el malware mencionado anteriormente se les conocen como máquinas Zombi y al conjunto de todas esas máquinas que están a disposición de un atacante se le conoce como BOTNET.

Existen multitud de herramientas de DDoS conocidas, algunas de las más importantes son:

Trinoo: Es la primera herramienta de ataque distribuido conocida. Constituye la base de otras herramientas más modernas y sofisticadas, diferenciándose unas de otras en la forma de

comunicación entre agentes y ataques desplegados. Se trata de un malware que convierte al equipo anfitrión en un cliente DDoS.

TFN y TFN2K: Estas dos herramientas son la evolución natural de la herramienta Trinoo, mejorando aspectos como la robustez, comunicación, etc. Esto conlleva una difícil detección de las nuevas herramientas, para combatir este ataque.

Stacheldraht: Nombre que proviene del alemán alambre de espino es una herramienta de ataque distribuido basada en el código fuente de TFN y que combina características de Trinoo y el TFN original. Añade cifrado en la comunicación entre el atacante y los maestros y un modelo de actualización de agentes esclavos bajo demanda.

Shaft: Su característica más distintiva es la capacidad de cambiar los servidores y puertos de comunicación en tiempo de ejecución. Hace especial hincapié en las estadísticas de los paquetes.

Los ataques posibles con las herramientas anteriores son: SYN flooding, UDP flooding, ICMP flooding y Smurf.

Antes de que el agresor pueda atacar al blanco definitivo, una flota de máquinas zombis (generalmente equipos no asegurados conectados permanentemente a Internet) debe coordinarse para el ataque.



## Tres Etapas para un Ataque DDoS

### 1.- Recopilación de información:

- a) Detección de la máquina destino.
- b) Detección de los servicios que se ejecutan en la máquina.
- c) Conocimiento de la topología de la red.
- d) Detección del sistema operativo de la máquina.

### 2.- Explotación:

La información recopilada en la fase anterior es utilizada para obtener acceso a la o las máquinas destino. Para ello se suelen aprovechar errores del sistema operativo, servicios mal configurados, malas políticas de asignación de claves, etc.

Esta búsqueda, se hace con la idea de infectar al mayor número de víctimas (lo que significa más equipos remotos) posible, esto se consigue mediante worms, troyanos, y otros tipos de malware que principalmente buscan en el sistema una puerta por la cual acceder.

### 3.- Metástasis:

Se divide en dos etapas:

**Consolidación:** Una vez que se han infectado varios sistemas, el atacante instala en las máquinas remotas, una aplicación la cual le va a permitir realizar este tipo de ataques. Este software también le garantizará el acceso en futuras ocasiones.

Una vez que se tiene acceso a una determinada máquina, el atacante intenta borrar las pistas que ha dejado durante la fase de explotación. Es posible, incluso, que intente eliminar las evidencias generadas durante la fase de recolección de información.

**Continuación:** Cuando el atacante tiene acceso a una máquina de la red, éste utiliza técnicas pasivas (pe: password sniffing) y activas para aumentar el número de máquinas comprometidas de la red. En esta fase el atacante se suele aprovechar de las relaciones de confianza entre diferentes máquinas.

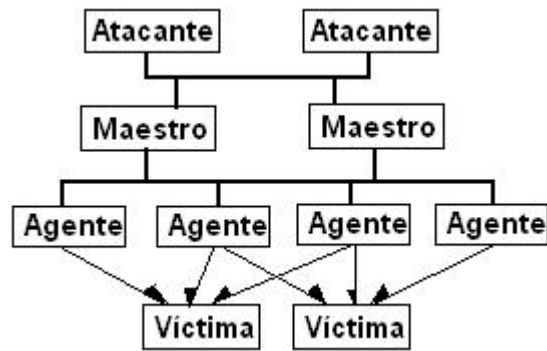


Figura 2.1: Una Típica Red DDoS

En la figura 2.1, se muestra la configuración de una red típica para llevar a cabo un ataque DDoS.

En la cima del diagrama se encuentra el Atacante o Atacantes, justo abajo de ellos, se disponen los Maestros, los cuales son los encargados de comunicarse con los Agentes para comenzar un ataque sobre la o las víctimas, por ordenes del Atacante o Atacantes.



Figura 2.2: Flujo de Información en el Escenario DDoS

En la figura 2.2 se puede ver un diagrama que muestra la comunicación de tráfico entre el Atacante y los Maestros, los Maestros y los Agentes, así como los Agentes y la Víctima. La comunicación se da de la siguiente manera, para la coordinación del ataque.

El Atacante o intruso, coordina a través de software a los Maestros que a su vez coordinan a los distintos Agentes para que se lleve a cabo un ataque DDoS.

La comunicación es unidireccional por el lado del Atacante -- Maestros; mientras que la comunicación Maestros -- Agente, es bidireccional, Por otro lado tenemos que la comunicación Agente -- Víctima vuelve a ser unidireccional.

Una característica importante del paradigma de ataque tradicional consiste en la realización de las acciones descritas anteriormente usando el modelo uno a uno (figura 2.3a), o uno a muchos (figura 2.3b), es decir, un atacante realiza sus acciones contra una o varias máquinas.

Los nuevos ataques distribuidos, por lo general, utilizan una aproximación basada en agentes. Estos agentes son programas instalados por el atacante, una vez que ya ha obtenido acceso, durante la fase de consolidación en varias máquinas. Los agentes, que son controlados por el atacante remotamente, se ejecutan continuamente como procesos servidores en estas máquinas. De esta forma un único atacante puede ordenar (controlar) a todos sus agentes para dirigir un ataque coordinado contra su víctima.

Los ataques distribuidos utilizan el modelo muchos a uno (figura 2.3c), o bien, muchos a muchos (figura 2.3d). Utilizando estos modelos, el atacante dificulta la tarea de descubrir cuál es el origen real del ataque, ya que las máquinas donde residen los agentes pueden pertenecer a redes completamente diferentes (incluso distar miles de kilómetros).

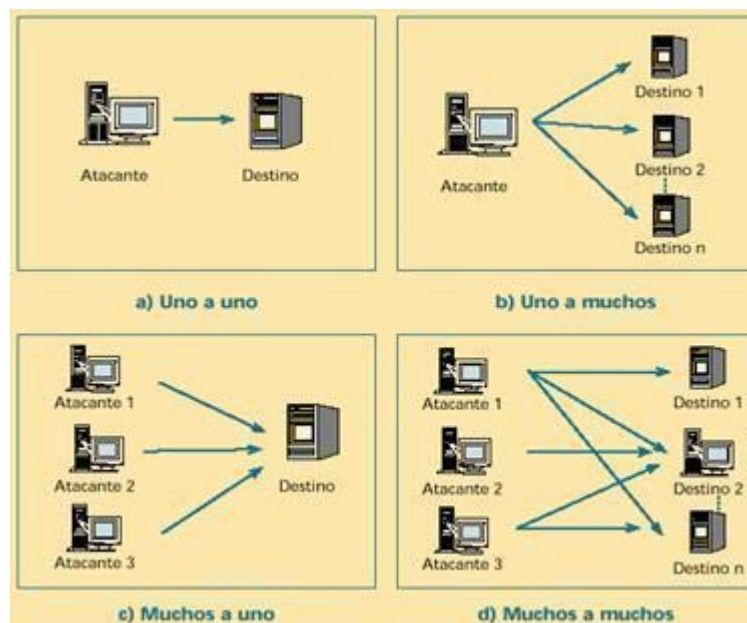


Figura 2.3: Modelos de los Ataques DDoS

## Síntomas de un Ataque DDoS

Se debe revisar el número de conexiones por dirección IP y el servicio a los que se conectan estas conexiones dentro del servidor examinado, de este modo se podrán quitar las dudas de si realmente se trata de un ataque DDoS.

En el escenario de un ataque y teniendo como ejemplo los siguientes datos mostrados abajo, se obtendría un listado parecido al siguiente: IP Servidor: 192.168.0.3, IP Atacante: 192.168.0.5

```
tcp 0 0 192.168.0.3:80 192.168.0.5:60808 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60761 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60876 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60946 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60763 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60955 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60765 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60961 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:60923 SYN-RECV
tcp 0 0 192.168.0.3:80 192.168.0.5:61336 SYN-RECV
```

### Descubrimiento del Ataque DDoS

Esta lista muestra la recepción en el servidor de segmentos de sincronía por parte de la dirección IP 192.168.0.5, que lo que está haciendo es solicitar conexiones con el servidor; La tasa de recepción de solicitudes es mucho más alta de lo habitual y no se detiene como lo haría una conexión normal.

Esta lista muestra un claro ejemplo del denominado SYN Attack al servidor Apache.

El problema es que cuando el número de conexiones Reading llena el MaxClients del Servidor Apache, no acepta nuevas peticiones, por lo que los nuevos clientes, aunque sean legítimos, no serán atendidos.

Podemos aumentar el valor del MaxClients, para que no se llene la cola de peticiones y acepte a todos los clientes, sean atacantes o no. Sin embargo, si la Botnet es muy grande, ésta será una solución temporal.

Otra medida es bajar el valor del Timeout del Servidor Apache para que las peticiones Reading sean ELIMINADAS rápidamente, antes que pueda llenarse el MaxClients a su tope. Sin embargo, también se eliminarán a los clientes legítimos.

Nota: La lista fue obtenida con el comando netstat con sus diferentes opciones.

## Defensas Contra el Ataque de DDoS

Soluciones tradicionales:

- \* Desconectar el servidor de la red, pero con esto, el atacante logra su objetivo de denegar el servicio, pues el servidor desaparece de Internet.
- \* Mover al servidor de una red a otra, sin embargo, el ataque seguirá al servidor a donde quiera que éste se mueva en Internet.
- \* Empezar la defensa en los principales ISP. Se puede soportar el ataque de varias IP's (proveniente de varios equipos remotos), pero si el Botnet es muy grande, por ejemplo del orden de 5000 zombis o más, se estaría indefenso y no se podría realizar ninguna acción de defensa.
- \* Reporte de Recargas de WEB, si la IP, recarga 100 páginas a menos de 2 páginas por segundo, está será reportada desde el servidor, deteniendo el flujo que provenía por parte de esa dirección.
- \* No permitir el tráfico broadcast desde fuera de la red: De esta forma se evita que la red propietario se emplee como multiplicador durante un ataque smurf.
- \* Filtrar el tráfico IP spoof: Esto es algo que todo ISP debería hacer, ya que permitiría localizar y solucionar el problema con una gran rapidez. En pocas palabras, estos filtros evitan que un atacante se pueda hacer pasar por algún usuario legal.

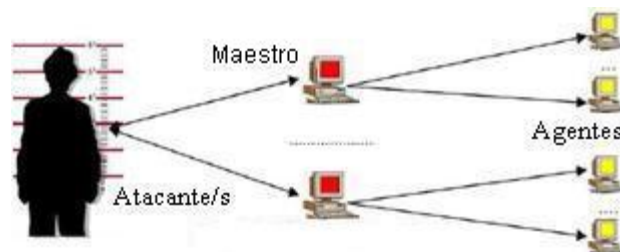


Figura 2.4: Jerarquía del Ataque DDoS

La ventaja para el agresor, en el uso de este ataque, es que le permite mantener su anonimato, ya que analiza el tráfico de los nodos agente y cuando detecta que el tráfico está siendo analizado en los agentes, cierran la conexión y posteriormente limpian las pruebas en

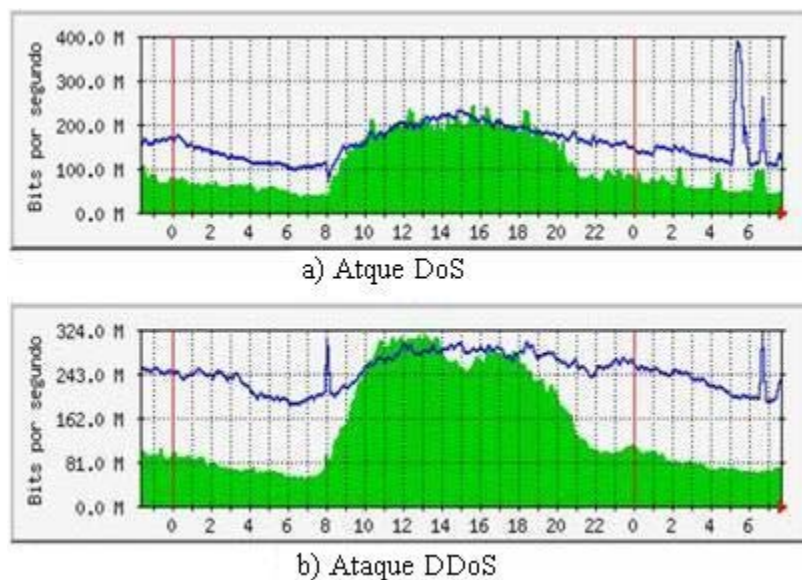
Maestros y finalmente cierran su conexión con éste. La figura 1.4 muestra la estructura.

### Recomendaciones

No tener proxies abiertos a todo Internet: Algunos administradores tienen los proxies, wingates, open sesame, SOCKS, SQUIDs, etc, abiertos a todo el mundo, sin ser conscientes de ello. Esto permite que cualquier usuario de Internet pueda atacar cualquier sistema responsabilizando a esa red intermedia mal administrada.

- a) Afinamiento en TCP/IP.
- b) Aumentar backlog.
- c) Disminuir el timeout.
- d) Instalar software con capacidades de detectar clientes DDoS (si los encuentra).

La figura 1.5 muestra gráficamente la diferencia en cuanto a severidad del ataque DoS y el ataque DDoS, viéndose que el ataque DDoS es mucho más robusto que el DoS.



La figura 1.5, muestra que el ataque DoS alcanza su punto máximo aproximadamente en los 200 Mbps; mientras que el DDoS alcanza su punto máximo en los 324 Mbps, casi 3/4 más de severidad en el ataque que el DoS, con la característica adicional de que es casi imposible de detectar pues las peticiones provienen de todas partes del mundo, a diferencia del DoS que generalmente las peticiones son realizadas por un sólo equipo.

## Objetivos de los Atacantes

- \* Los ataques de DDoS son importantes porque algunos criminales extorsionan a los administradores de sitios en la red pidiendo dinero para evitar un ataque.
- \* Sabotaje externo (competencia) o interno (personal mal intencionado).
- \* Subir o bajar el valor de las acciones de una determinada empresa. La pérdida de credibilidad produce daños perdurables sobre la reputación de un negocio.
- \* Renta de Zombis para el ataque DDoS. Como el Botnet es controlado de forma remota, habitualmente a través de uno o varios servidores IRC (Internet Relay Chat). Esta flota de Zombis puede ser rentada por alguna entidad para llevar a cabo un ataque con fines destructivos/económicos u otro en particular.

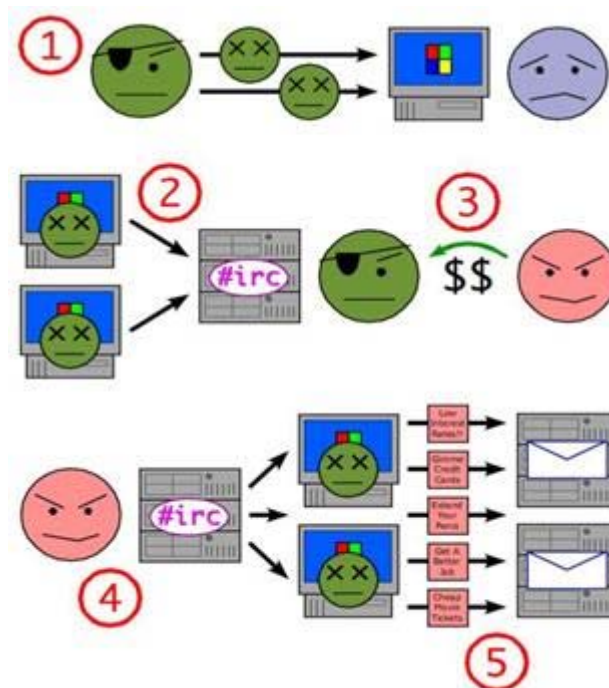


Figura 2.6: Creación de una Botnet para Renta

- \* El atacante lo puede hacer simplemente por probarse a si mismo que lo puede llevar a cabo.

## Resumen

Las variaciones del ataque DoS hacen muy peligrosa la evolución de éste, en el ataque DDoS, por ejemplo, se deben de tomar medidas en la prevención y hasta donde sea posible la minimización del ataque, se sabe que la arquitectura básica TCP/IP de Internet, no cambiará de una manera drástica en los próximos años, es por ello que seguirán existiendo las debilidades en la torre de protocolos TCP/IP, así que la solución o minimización, se debe de buscar por otros horizontes, antes de pensar en cambiar la arquitectura de Internet, ya que hacer cambios en su arquitectura, es mucho más complicado, que proteger un servidor de múltiples ataques.

Es importante hacer notar las características más destacadas del ataque DDoS:

- El atacante aprovecha la falta de seguridad de los zombis.
- El atacante irrumpe en el sistema directamente o mediante un virus de e-mail, por ejemplo.
- El objetivo de la irrupción o virus es instalar software para convertir a un sistema en un zombi.
- El atacante usa los zombis para lanzar un ataque DDoS sobre un blanco generalmente estudiado con anterioridad por diversos motivos.
- Las técnicas de implantación del DDoS pueden ser manuales o automatizadas a través de gusanos, troyanos, spyware, etc.

Los administradores de redes deben de hacer todo lo posible por adoptar políticas de seguridad para no dejar a la deriva los equipos de red.

Los Ataques DDoS están impactando a todos los negocios principales de Finanzas, Cuidados de la Salud, Gobierno Federal, Manufactura, Comercio Electrónico, Retail, Temporada de vacaciones, eventos deportivos importantes para los medios, juegos en línea, etc.



## Conclusiones del Capítulo

Al analizar el funcionamiento de los DDoS no fue difícil darse cuenta que no existen soluciones 100% fiables contra ellos.

La literatura asegura que la forma de defenderse de los efectos de un ataque distribuido como el DDoS es adoptando los siguientes 5 puntos básicos:

1. Una solución distribuida para un problema distribuido.
2. La solución no debe penalizar el tráfico de usuarios legítimos.
3. Solución robusta y universal (amenazas internas y externas).
4. El sistema debe ser viable en su aplicación.
5. Debe ser una solución incremental.

Las soluciones actuales se basan en firewalls clásicos y sistemas de detección de intrusos. Organismos como CISCO recomiendan soluciones sencillas como modificar el tamaño de la pila de TCP o disminuir el tiempo de espera de establecimiento de las conexiones.

Es evidente que toda solución va enfocada a la prevención del ataque DDoS, configuración de software detector de clientes DDoS, configuración de la seguridad de equipos de cómputo, etc.

No obstante nada soluciona el problema, una vez que éste llega, pues las medidas de configuración del servidor que está sufriendo el ataque sólo las alivian por momentos.

En este trabajo se intentará lograr detectar un ataque DDoS, justo unos instantes antes de que suceda y una vez detectado éste, aplicar un mecanismo de minimización de efectos para no denegar el servicio a los clientes legítimos.

Es evidente que la única forma de detener el ataque DDoS, es teniendo acceso a la máquina del atacante, pero llegar a ella, en el momento que está comandando el ataque, es virtualmente imposible y cambiar la arquitectura de Internet, es arriesgarse a que la Red de Redes no funcione, haciendo el ataque DDoS súper potente, en ese sentido estricto, ya que nadie sabe como afectaría un cambio de esa magnitud al Internet. Esas son razones suficientes para darle una solución al DDoS por otros medios.

## Capítulo 3

### Modelado del Ataque DDoS

#### Introducción

El punto central de este trabajo está en la minimización de los daños causados por el ataque DDoS, sin embargo, mirar un ataque de esta naturaleza, en todo su esplendor no es nada fácil, ya que para verlo se tendría que llevar a cabo en Internet, lo cual es inminentemente ilegal.

Una forma de llevarlo a cabo, sería tomar algunas máquinas de un laboratorio e intentar ejecutar un DDoS, pero los datos recopilados estarían demasiado lejos de la realidad, debido al bajo número de máquinas disponibles.

Otro acercamiento sería el uso de un simulador, ya que éste permitiría crear Botnes de varias decenas o incluso cientos de nodos, lo cual sería muy laborioso de hacer dentro de una universidad con equipos reales.

La opción del simulador ofrecerá un panorama similar al que se tendría con un servidor atendiendo a unos cuantos clientes en un determinado lapso de tiempo y los efectos de un ataque DDoS, en parte, de dicho lapso de tiempo.

En este capítulo se utilizará un simulador, para construir una red Botnet de 100 nodos, los cuales son suficientes para ver los efectos negativos que sufren, tanto las conexiones existentes en un servidor, así como el intento de conexión de nuevos clientes.

## **Modelado del Ataque DDoS Bajo NS-2**

### Simulación de Redes de Datos

Debido al ámbito de nuestro estudio, nos encontramos con que nuestro campo de trabajo es todo Internet. Obviamente resulta del todo imposible el hecho de plantearse realizar las distintas pruebas en el dominio real de nuestro estudio. Es más, incluso nos resulta muy difícil el intentar realizar pruebas reales dentro de una universidad.

Al igual que ha pasado en otros campos de la ciencia, la creación de modelos de comportamiento de las redes de datos y el aumento de velocidad de las computadoras, ha permitido la creación de programas de simulación.

Igualmente y aún suponiendo perfectos los programas de simulación, resulta del todo impensable modelar Internet para la realización de cualquier tipo de experimento.

Por otro lado en la actualidad nadie conoce exactamente la topología existente de Internet, ya que varía diariamente con el número de nodos conectados a ésta. Además las conexiones inalámbricas complican aún más el conocimiento de la distribución real de Internet.

Sin embargo, la simulación de pequeñas partes puede ser de gran utilidad, ya que el hecho de que Internet sea una red IP puede permitirnos verla como la suma de millones de subredes IP más pequeñas.

Aún, así, debido a que monitorear un ataque DDoS sobre Internet en tiempo real, para recopilar datos, es sumamente complicado, ya que se tendría que saber donde y cuando se llevaría a cabo un ataque.

Una vez explicados los motivos precedentes, se ha visto la necesidad de usar un simulador de redes, para representar un ataque DDoS a la escala máxima que el simulador y el equipo de cómputo a utilizar, lo permitan.

La herramienta de software que se incluirá en el estudio será NS2 (Network Simulator 2).

El simulador NS-2\cite{dw-NS2} es parte del código escrito para el simulador REAL con el objetivo de crear un simulador discreto de redes TCP. Incorpora las capacidades de encaminamiento y multicast en redes guiadas, inalámbricas y móviles.

Este simulador de redes de eventos discretos es utilizado principalmente en ambientes académicos debido a que está escrito en código abierto. Se pueden simular tanto protocolos unicast como multicast y se utiliza intensamente en la investigación de redes. Puede simular una amplia gama de protocolos tanto para redes cableadas o redes wireless, así como mixtas.

El proyecto NS-2 está patrocinado por DARPA, Xerox y el instituto de ciencias de la información (ISI) de la Universidad del Sur de California (USA).

Este simulador permite la modelación de las estructuras deseadas y actualmente ofrece un cierto soporte al usuario.

### **Escenario a Simular en NS-2**

El escenario a simular, es el mostrado con anterioridad en la figura 1.2, con las siguientes características adicionales:

- a) Un Atacante.
- b) Dos Maestros.
- c) Una Botnet de 100 Zombis (50 nodos por Maestro).

La figura 2.1 muestra el arreglo con los elementos descritos, para la configuración de un ataque DDoS.

La red Botnet de la figura 2.1 está formada por un atacante, representado por el cuadrado 0 de color rojo, los Maestros están representados por los cuadrados 1 y 2 respectivamente, color azul; mientras los Zombis están representados por los círculos negros y van del nodo 3 al 52 para el Maestro 1 y del nodo 54 al nodo 103, para el Maestro 2.

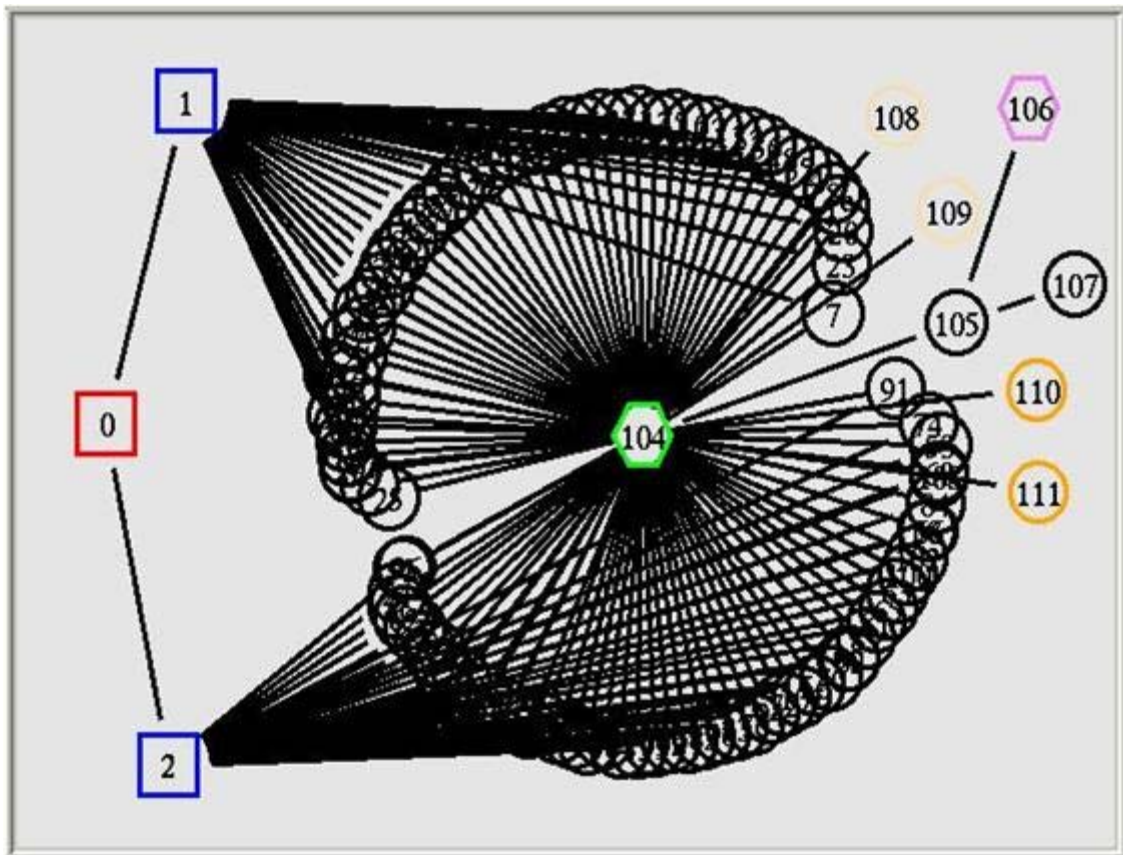


Figura 3.1: Configuración de un Ataque DDoS

Por el otro lado se encuentra el nodo 104 de color verde, el cual representa al Router intermediario entre la red del servidor víctima (nodo 106 con color violeta) y una sección de Internet. El nodo 105, simula el funcionamiento de un Switch y el nodo 107 simula la intranet de una determinada corporación a la cual pertenece el servidor víctima, al que se le someterá al ataque DDoS.

#### Topología a Nivel Transporte de la Simulación

Antes de que inicie el ataque DDoS, el servidor víctima tiene un funcionamiento normal, es decir, hay un par de clientes conectados a éste, el cliente 1 en el nodo 108 con un servicio UDP y el cliente 2 en el nodo 109 con un servicio TCP.

A continuación se describirá el efecto del ataque DDoS sobre el cliente 1 con servicio UDP. La figura 2.2 muestra dos cuadrados, el cuadrado rojo representa la transmisión de tráfico UDP así como su  $\$Esperanza\$$  de recepción en el nodo 106.

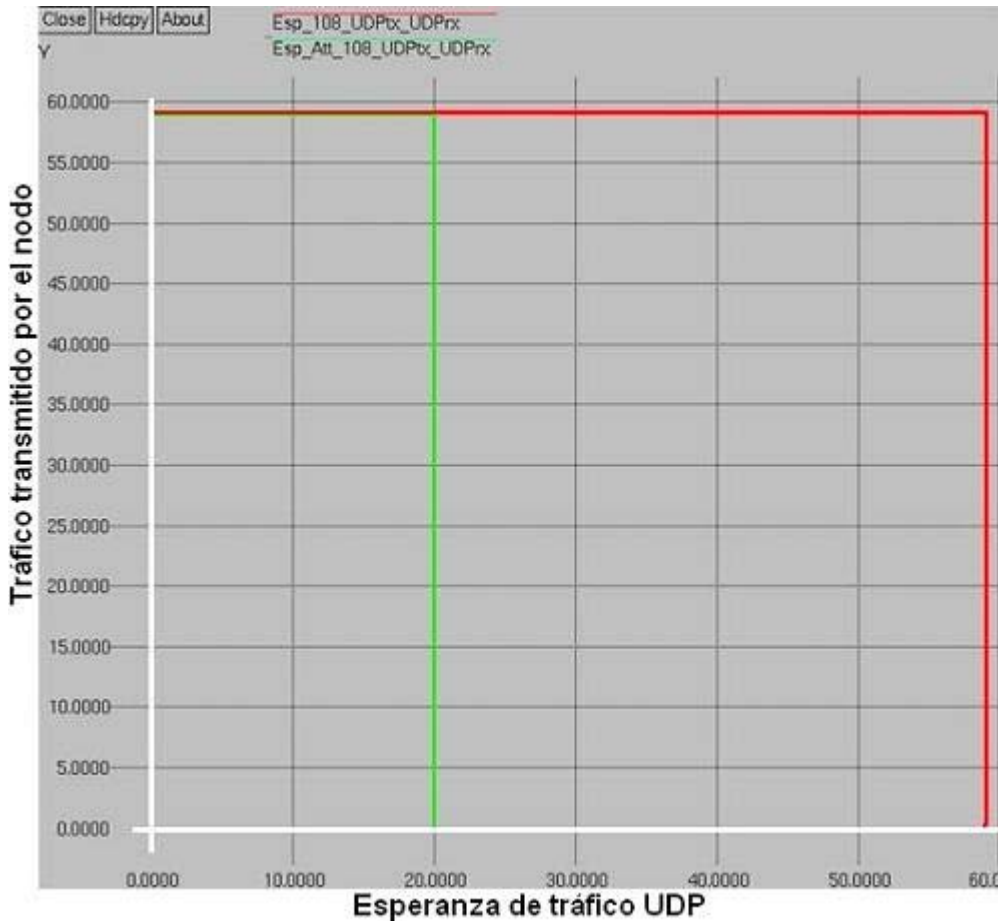


Figura 3.2: Analogía de la Transmisión-Recepción con el Área de un Cuadrado

En la figura 3.2, el eje de las abscisas representa la Esperanza del tráfico UDP que se recibiría en el nodo destino (nodo 106), sin la intervención de un ataque DDoS. Por otro lado el eje de las ordenadas representa el tráfico transmitido por el nodo origen (nodo 108).

Una vez explicado lo anterior y con la intención de aclararlo un poco más, hago la analogía de ver la cantidad de información Transmitida-Recibida, como la cantidad de área de un cuadrado. Entonces en una transmisión UDP en condiciones favorables, donde no se realiza un ataque DDoS, se podría tener un escenario como el que describe el cuadrado rojo de la figura 2.2, pues cómo la transmisión UDP tiene una tasa de transferencia constante (CBR) se ESPERA recibir lo mismo que se transmite.

Es importante hacer notar al lector que la transmisión entre el cliente 1 y el servidor, no comienzan al mismo tiempo que el ataque DDoS aparece en el escenario, es decir, la transmisión UDP entre el cliente 1 y el servidor, comienza a los 0.02 segundos, mientras que el ataque DDoS comienza para este ejemplo a los 0.13 segundos, con una duración total de 2.86 segundos.

Una vez iniciado el ataque, el cuadro rojo se ve severamente disminuido en el eje de las abscisas, este eje representa la cantidad de tráfico recibido en el nodo servidor por parte del cliente 1. Debido al ataque se comienza con una pérdida de paquetes, lo cual se traduce en la disminución del área del cuadrado rojo, convirtiéndose en el rectángulo de color verde de la figura 2.2, donde se aprecia que los paquetes mandados del cliente 1 al servidor, simplemente no llegan debido a que se sobrepasa la capacidad de procesamiento del Router gracias al tráfico generado por parte de los Zombis de la Botnet.

Es de importancia destacar que el tráfico entregado al servidor, por parte del cliente 1, representado por el rectángulo verde, es del 30%, sin embargo, la mayor parte de este 30%, se entregó antes de que comenzara el ataque DDoS, así que durante el ataque, los paquetes entregados fueron unos cuantos del total, quedando así el servicio denegado, cumpliéndose con el objetivo del atacante.

Por ejemplo, si se hubiese estado llevando a cabo una llamada de VoIP, ésta hubiese sido interrumpida de manera definitiva bajo el ataque DDoS.

Ahora se va describir el efecto del ataque DDoS sobre una conexión TCP. De igual forma que para el caso anterior, la conexión TCP se estableció antes de que se iniciara dicho ataque.

En la figura 2.3, se interpreta de la siguiente manera, el eje de las abscisas corresponde al tiempo, éste fue dividido en múltiplos de 0.05 segundos, el eje de las ordenadas representa la transmisión de segmentos TCP. La conexión TCP se inició en el segundo 0.04, de aquí en adelante el crecimiento de la ventana será el doble de la ventana actual, esto es, el flujo de segmentos será 1Win, el siguiente 2Win, el próximo 4Win y así sucesivamente, hasta que algún paquete TCP no sea confirmado, lo cual hará que la ventana descienda, por ejemplo, hasta 1Win y vuelva a empezar su crecimiento. El crecimiento de la ventana después de un segmento NO confirmado dependerá de la versión de TCP. Para este experimento la versión de TCP fue TCP-newRENO.

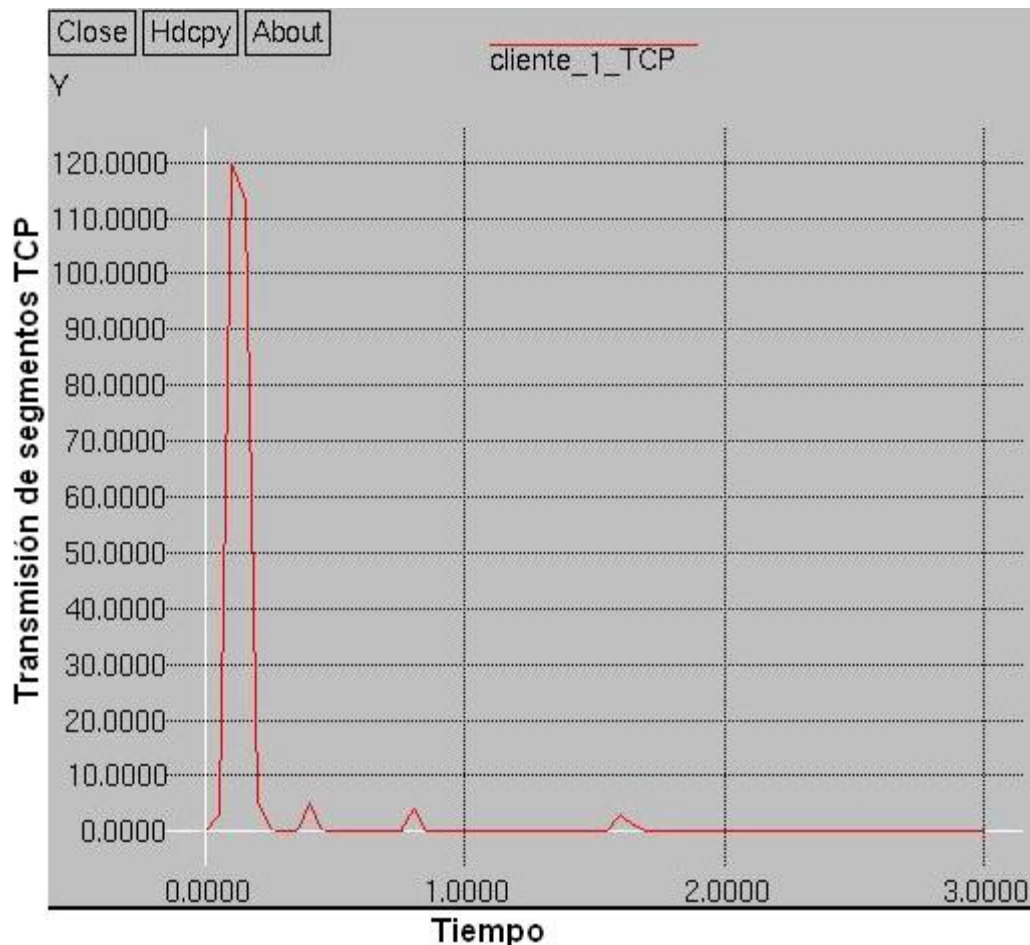


Figura 3.3: DDoS Sobre un Cliente TCP

Llegado a este punto, es buena práctica comprender el escenario donde en pleno ataque DDoS, un cliente UDP, trata de llevar a cabo una transmisión con el servidor.

Para este caso, el cliente 2, de tipo UDP, es el nodo 110. Este nodo tratará de hacer una transmisión al servidor en el segundo 0.14, justo cuando el ataque ya está en marcha, pues el DDoS comenzó el ataque al Router en el segundo 0.13.

En la figura 3.4 se muestra la recepción de información por parte del servidor. De nueva cuenta se tiene el cuadrado rojo como la \$Esperanza\$ de Transmisión-Recepción. El rectángulo verde muestra la severidad del ataque al dejar pasar apenas a una pequeña parte de los paquetes transmitidos. Es importante hacer ver que los flujos UDP, no soportan retardos muy grandes y mucho menos la pérdida de la mayor parte de los paquetes de información, ya que el emisor NO retransmitirá la información debido a la naturaleza propia de UDP.



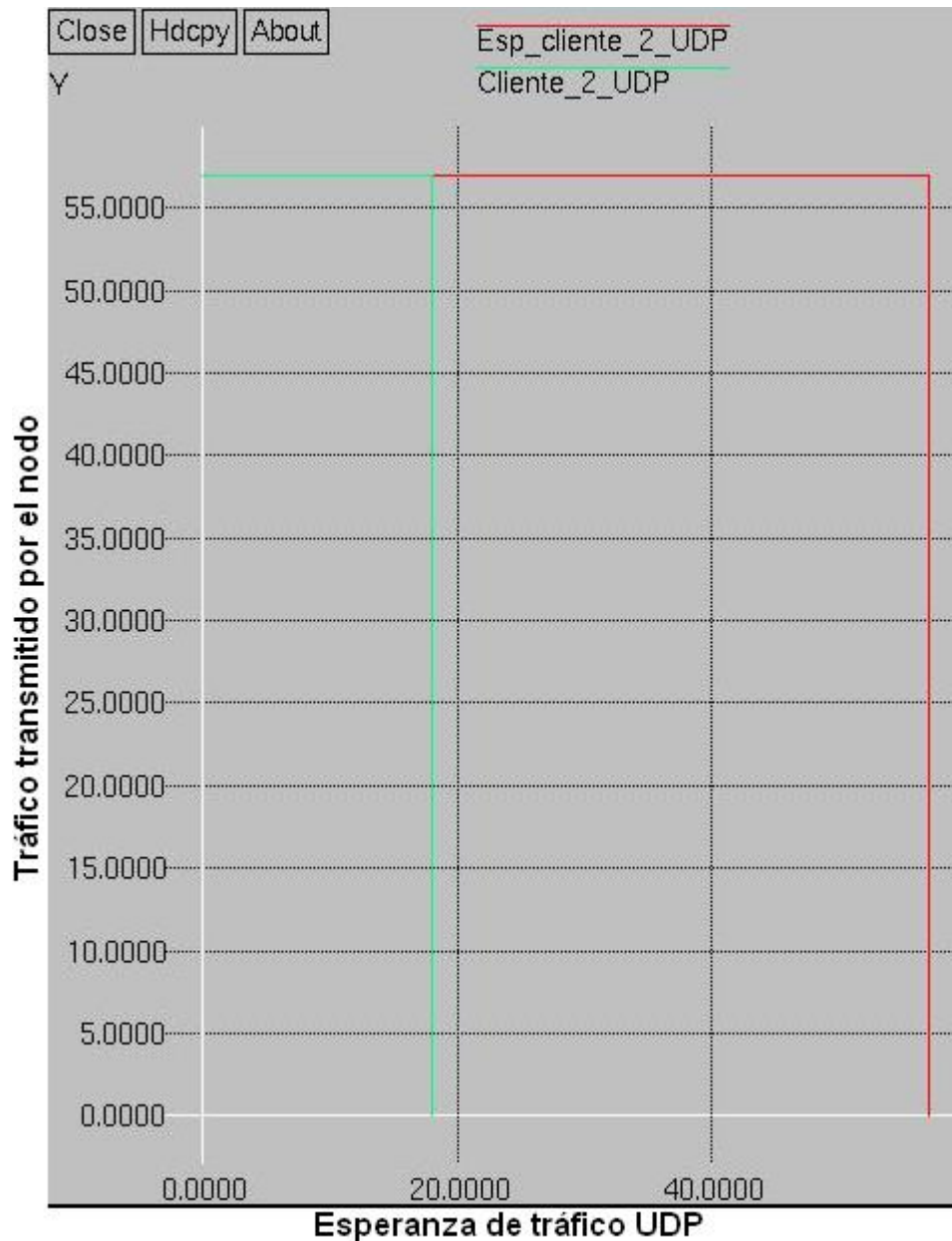


Figura 3.4: Cliente 2 UDP ante el Ataque DDoS

A manera de ejemplo supóngase que el cliente 2 quiso enviar paquetes de video en tiempo real al servidor, pero con la pérdida de la mayoría de los paquetes, aunque hayan llegado 18 de estos, no sirve de nada. El servicio no se pudo concretar, cayendo en la denegación, de esta forma el atacante vuelve a tener éxito.

Lo último que se hará en este capítulo será tratar de establecer una conexión TCP durante el ataque DDoS, para conocer sus efectos.

La figura 2.5 muestra como TCP envía segmentos de una manera muy discreta, debido a que estos segmentos nunca son confirmados, la ventana máxima que alcanza TCP por unos instantes de 10 segmentos.

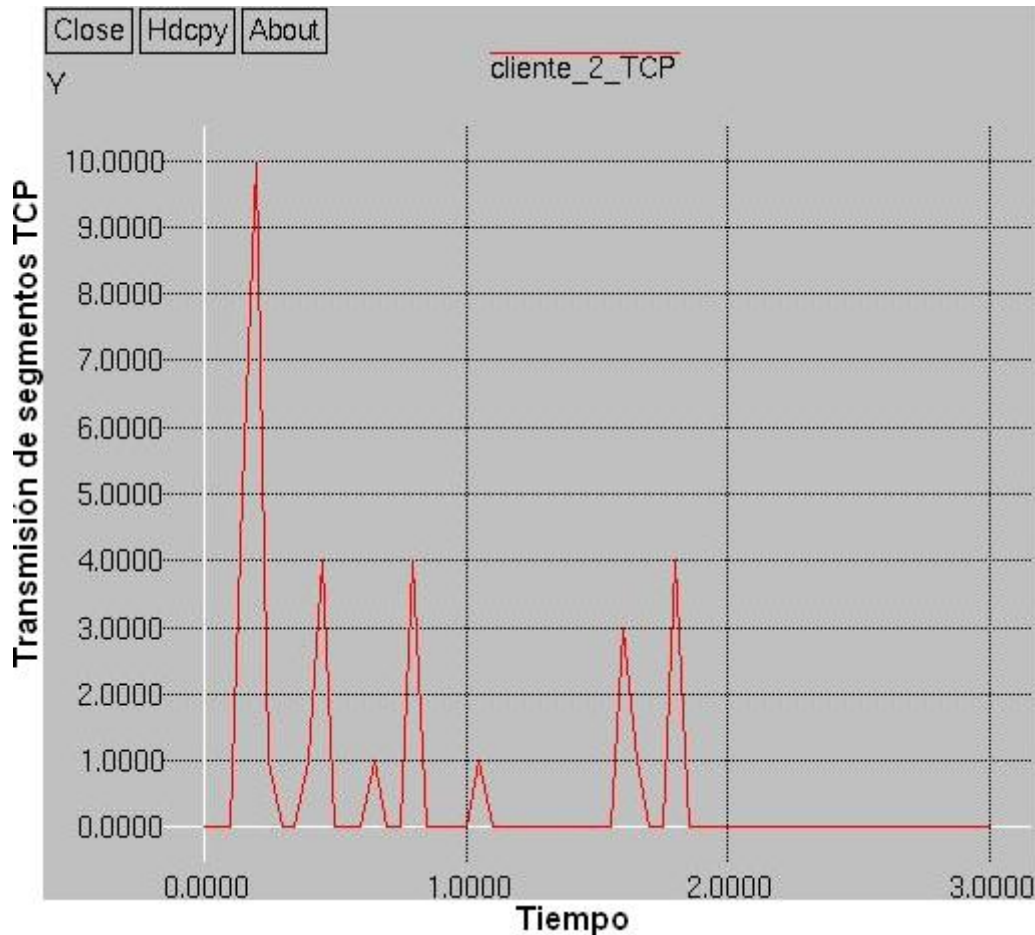


Figura 3.5: Cliente 2 TCP ante el Ataque DDoS

El nodo 111 es el cliente 2 TCP, el cual intenta establecer una conexión, para algunas transacciones típicas de un cliente. Sin embargo, al momento de que éste intenta la conexión con el servidor, éste se encuentra bajo un ataque DDoS, así, por tanto, el desempeño de la conexión por parte del cliente 2, es muy bajo.

Haciendo la comparación con el cliente 1 TCP, el cual logró una ventana de 120 segmentos entre 0.04 a 0.14 segundos ya que su conexión se llevó a cabo bajo condiciones normales; mientras que el cliente 2 TCP intentó hacer lo mismo bajo el ataque DDoS, sin ningún éxito.

Ante esta situación el cliente 2 TCP nunca logró su cometido, es decir, hubo denegación del servicio por sobrepasar la capacidad del procesamiento del Router, puerta de entrada a la red del servidor.

## Conclusiones del Capítulo

Este capítulo tuvo como objetivo afianzar el entendimiento del capítulo 2, a cerca del ataque DDoS y ver la simulación de sus efectos, a la escala de los recursos que se tuvieron disponibles a la hora de realizar el experimento, cómo fueron.

- a) Un Procesador 3.7 GHz
- b) Memoria RAM 1GB.
- c) NS-2 versión 2.33.

La existencia de algo tan grande como Internet y que además se encuentra en expansión perpetua, lo hace imposible de modelar y obviamente de medir, sin embargo, la extracción de una parte de Internet es algo posible. En este capítulo se modeló, sí así se desea ver, una célula de Internet, así como, un parásito que le afecta a la misma.

La herramienta de modelado, usada para este capítulo, fue de mucha ayuda a la hora de ver los efectos de los daños del ataque DDOS de manera gráfica. Esta forma de ver los efectos es mucho más entendible que un conjunto de datos en una tabla.

En este capítulo se observó la configuración de un Botnet de 100 nodos y sus efectos contra un servidor víctima.

No obstante, se debe de tener en cuenta que el ataque DDoS no sólo deniega el servicio a los clientes nuevos que intentan establecer una conexión legítima con un determinado servidor, si no que también aniquilan las conexiones ya existentes en dicho servidor.

La simulación de este ejercicio dejó ver múltiples facetas del ataque DDoS, sin embargo, en este trabajo sólo se abarcó una configuración básica del DDoS, sobre la cual se hará una propuesta para tratar de minimizar lo más que se pueda, los efectos vistos en este capítulo.

## Capítulo 4

### Aplicación de la Predicción Lineal

#### Introducción

En este capítulo no se pretende explicar la teoría de la Predicción Lineal, ya que para ello existen escritos bastantes y muy buenos libros como. Las pretensiones en este trabajo son las de aplicar la teoría de Predicción Lineal para hallar el momento en el cual, está comenzando un ataque DDoS.

Para hallar donde comienza un ataque DDoS en su modalidad UDP flooding o TCP flooding, se diseñarán dos algoritmos basados en las técnicas de Predicción Lineal, estos algoritmos serán usados como sensores en el servidor que se desea proteger, los cuales ayudarán al cálculo de un Diferencial.

Este Diferencial será el encargado de dar aviso al mecanismo de defensa, para que éste comience a hacer su trabajo, el cual será la minimización de los ataques ya mencionados.

Una vez más se utilizará el simulador NS-2 como herramienta, para simular los dos escenarios que se necesitan, para extraer los datos y poder aplicar las técnicas de Predicción Lineal a las Serie de Tiempo extraídas, según el monitoreo realizado en el simulador NS-2.

La virtud de utilizar la simulación recae en la ventaja de poder llevar a cabo tantos ataques como sean necesarios, en un servidor que se encuentre atendiendo a una cartera de clientes, de tal suerte que se puede plantear una excelente replica de la realidad. En lo consecuente se explicará el modelado y sus características así como la forma de organización del tráfico monitoreado.

## Análisis del Muestreo

En este capítulo nos abocaremos a monitorear el servidor que se desea proteger con la idea de saber cual es su comportamiento habitual, así como, saber la cantidad promedio de tráfico generado por los clientes de dicho servidor.

Hay muchas maneras de monitorear un servidor, sin embargo, por cuestiones prácticas, utilizaré de nueva cuenta el simulador NS-2, para obtener datos muy semejantes al monitoreo real, a fin de cuentas, el tratamiento de los datos es el mismo.

Supóngase que se tiene un servidor, el cual atiende 50 clientes, como máximo, en sus horas pico. Hago hincapié en que no importa que fuesen 500 clientes o más, el método sería semejante.

La figura 3.1 muestra a un servidor atendiendo a 50 clientes, que tiene el mencionado servidor, como máximo en sus horas de mayor estrés.

Cada hexágono en verde, hace referencia a un Router a través del cual se comunica un grupo de clientes con el Router etiquetado con el número 57, que conduce al servidor (circulo amarillo, número 60).

Los cuadrados en azul son Switches, el Switch etiquetado con el número 59, es el intermediario entre el Router, etiquetado con el número 57 y el servidor víctima.

La Botnet que en estos momentos sólo está latente, es decir, sin atacar. Está constituida por un atacante etiquetado con el número 0, así como 3 maestros 1, 2 y 3 respectivamente, que al igual que el atacante se pueden ver como 4 círculos rojos en la figura 3.1.

Los flujos en morado de la figura 3.1, muestran el tráfico TCP que generan los clientes de manera ALEATORIA, es decir, un cliente comienza a generar una petición al servidor, con tráfico TCP en un tiempo aleatorio y termina las solicitudes en otro tiempo que también es aleatorio, este proceso lo realizan los 50 clientes para este escenario de simulación. De esta forma se pretende simular el comportamiento aleatorio de los clientes del servidor, tomando en cuenta que el tiempo de duración de una conexión es diferente para cada cliente en particular.

El código TCL (Tool Command Language) que da origen a esta configuración y a los datos que se analizarán posteriormente, se encuentra en el apéndice A.

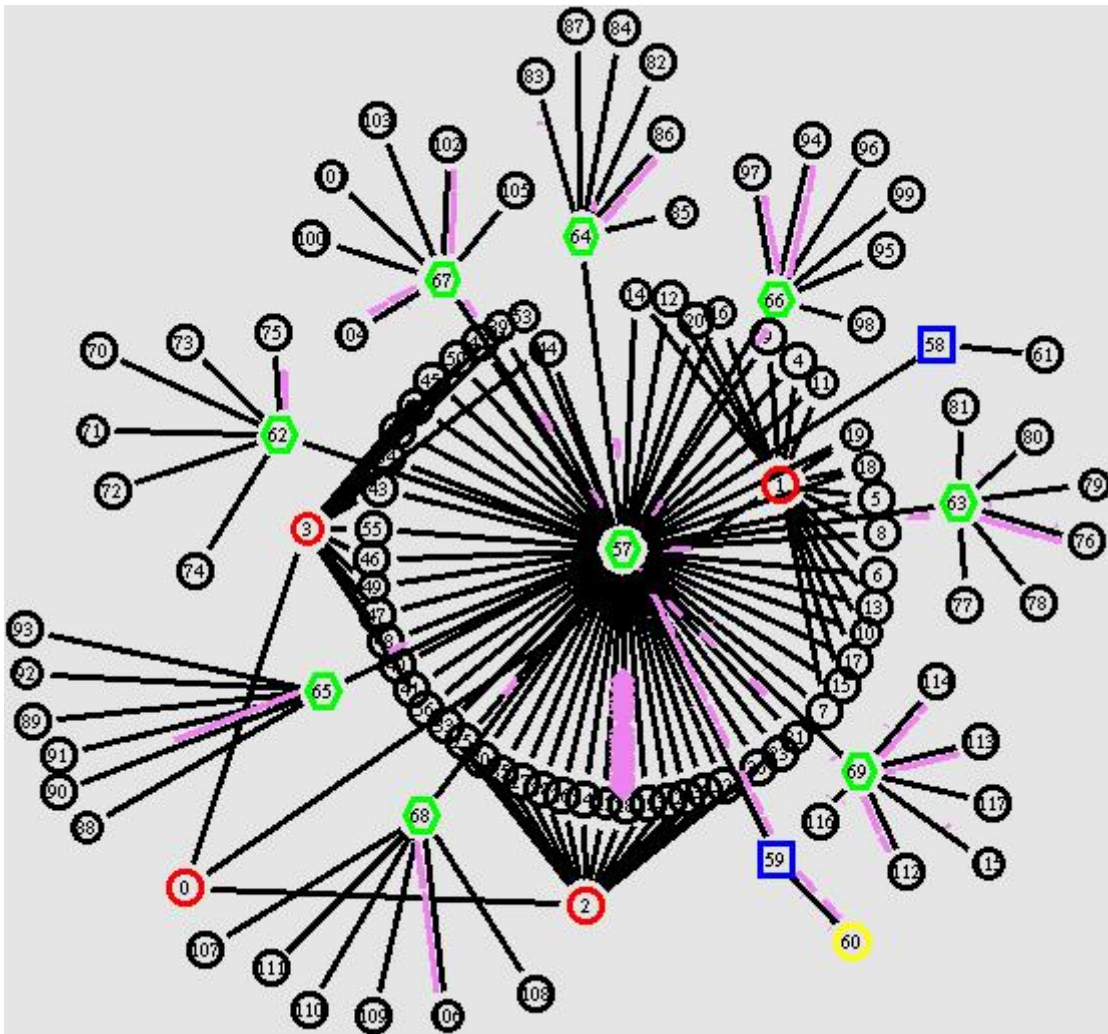


Figura 4.1: Clientes Comunicándose con un Servidor sin Ataque DDoS

En este capítulo se abordará la forma de como tratar de detectar en su inicio un ataque DDoS, utilizando las técnicas de Predicción Lineal, para ello se observará el comportamiento de la Predicción para los cuartos del segundo a predecir en base a una Serie de Tiempo cuando el ataque DDoS está ausente. Posteriormente se analizará el caso donde cada cuarto del siguiente segundo a predecir está bajo ataque DDoS, con la finalidad de encontrar el \$Diferencial\$ que nos pueda llevar a la conclusión de que se está comenzando un ataque DDoS o en su defecto, las condiciones son normales. Lo anterior, con la intención de minimizar el número de falsas alarmas.

## Predicción sin DDoS

Para este caso tomaremos esos 50 clientes, los cuales se monitorearán por un lapso de 40 segundos, de estos 40 segundos, se tomará una Serie de Tiempo de 5 segundos, que será del segundo 14 al segundo 18 y cada segundo a su vez será dividido en cuartos de 250 milisegundos (ms) y esos datos se someterán a las técnicas de la Predicción Lineal. La tabla 3.1 muestra la disposición de los datos, para el procesado bajo las técnicas de la Predicción Lineal.

Cuartos	Segundo 14	Segundo 15	Segundo 16	Segundo 17	Segundo 18
249 ms	2920	2920	2917	2940	2868
499 ms	2908	2891	2929	2948	2914
799 ms	2882	2921	2933	2887	2902
999 ms	2956	2864	2890	2872	2874

Tabla 4.1: Serie de Tiempo



Figura 4.2: Serie de Tiempo del Servidor sin Ataque DDoS

Es de observarse en la figura 3.2, que el eje de las ordenadas, por su parte hace referencia a la cantidad de segmentos registrados por cuarto de segundo monitoreado.

Es importante hacer notar al lector que el eje de las abscisas no representa al tiempo en unidades de un segundo, sino a los cuartos de segundo, es decir, el valor 1 del eje de las abscisas representa los primeros 250 ms, transcurridos en el monitoreo.

Una vez recopilados los datos hasta el segundo 18, se tratará de predecir el comportamiento del tráfico en los cuartos del segundo 19.

El primer paso para lograr esto, es tomar la Serie de Tiempo de la tabla 3.1 y convertirla en una gráfica en dos dimensiones. Esto nos permite detectar las componentes esenciales de la serie. La figura 3.2 ilustra esta transformación.

De esta forma, se está trabajando con 58,136 segmentos TCP de 1024 bytes cada uno, procesados por el servidor en 5 segundos.

Una vez graficada la Serie de Tiempo  $X(t)$ , se debe de tratar de detectar Outlier o puntos de la serie que se escapan de lo normal. Pues estos pueden reflejar un comportamiento anormal del fenómeno, quizás con incidencias futuras o un error de medición. Para este ejemplo en particular no se encontró ningún Outlier.

El siguiente paso es obtener a partir de la serie observada una nueva serie suavizada  $Z(t)$ , que suaviza los efectos ajenos a la tendencia (estacionalidad, efectos aleatorios), de manera que podamos determinar la dirección de la tendencia. Para ello se debe de obtener una serie suavizada  $Z(t)$ , con esta finalidad se utilizará el Promedio Móvil centrado de orden 3. El resultado del suavizamiento de la Serie de Tiempo lo muestra la tabla 3.2.

Cuartos	Segundo 14	Segundo 15	Segundo 16	Segundo 17	Segundo 18
249 ms		2919,333	2900,667	2921	2875,667
499 ms		2922,333	2903,333	2926	2884,667
799 ms	2903,333	2910,667	2926,333	2925	
999 ms	2915,333	2892	2917,333	2902,333	

Tabla 4.2: Serie de Tiempo Suavizada

Un ejemplo sería obtener el primer y segundo valor del Promedio móvil:  $Pm = \frac{2920+2908+2882}{3} = 2903,333$ . El siguiente valor se calcularía  $Pm = \frac{2908+2882+2956}{3} = 2915,333$ . El procedimiento sigue hasta llegar al término  $t_{n-3}$ .

La serie suavizada, también puede observarse en una gráfica, la figura 3.3, es la representación gráfica de la suavización de la Serie de Tiempo (en rojo).





Figura 4.3: Serie de Tiempo Suavizada sin Ataque DDoS

Una vez suavizada la serie, la Serie Residual  $R(t)$  con el objeto de eliminar la estacionalidad dentro del modelo y saber por medio de un análisis tabular de los residuos si el modelo es Aditivo o Mixto.

El paso siguiente es la obtención de la Serie Residual  $R(t)$ :

$$R(t) = \frac{X(t)}{Z(t)}$$

Cuartos	Segundo 14	Segundo 15	Segundo 16	Segundo 17	Segundo 18
249 ms		1.000228	1.005630	1.006504	0.997333
499 ms		0.989277	1.008840	1.007518	1.010168
799 ms	0.992652	1.003550	1.002278	0.987008	
999 ms	1.013949	0.990318	0.990630	0.989548	

Tabla 4.3: Serie Residual

La tabla 3.3 tiene los resultados del cálculo de la Serie Residual  $R(t)$ .

El siguiente paso es calcular la tendencia  $T(t)$ , para calcular esta tendencia se tienen varios modelos, pero el que en mi opinión se ajusta más a las características de este trabajo es el de Regresión Lineal RL, el cual tiene la siguiente relación matemática:

$$T(t) = at + b$$

El cálculo del parámetro  $a$  se efectúa con la relación:

$$a = \frac{\sum_{i=\text{orden3}}^{N-\text{orden}} Z(t_i)(t_i - \bar{t})}{\sum_{i=\text{orden3}}^{N-\text{orden}} (t_i - \bar{t})^2}$$

El cálculo del parámetro  $b$  se efectúa con la relación:

$$b = \bar{Z}(t) - a\bar{t}$$

Es importante destacar que estos cálculos se deben de hacer con los datos de la serie suavizada:  $t$  y  $Z(t)$ . Para este caso en particular los valores de estos parámetros son:

$$a = -0,931372549$$

$$b = 2918,862745$$

Así la tendencia queda definida, para este ejemplo como:

$$T(t) = -0,931372549t + 2918,862745$$

En la figura 3.4 se puede observar la gráfica donde se encuentra la recta de la tendencia  $T(t)$ , como se puede observar allí mismo, la tendencia es negativa debido a que la pendiente de la recta es negativa.

Que la pendiente sea negativa, sólo indica que de manera normal o natural, el sistema comienza a procesar menos tráfico, como este efecto, es parte del mismo sistema, no es motivo de alarma alguna.

También podría suceder que la pendiente fuese positiva, lo cual sería indicativo de que el tráfico a procesar por el servidor va en aumento, lo cual puede indicar que el número de clientes está aumentando en un determinado lapso de tiempo.



Figura 4.4: Gráfica de la Recta de la Tendencia

La Tendencia de la serie  $T(t)$ , representa la dirección predominante de la serie, es decir, su comportamiento promedio.

Llegado a este punto, es momento de calcular la Variación Estacional  $E(t)$ , la cual se caracteriza por tratarse de períodos o ciclos de la serie.

El proceso para calcular la Estimación de la Variación Estacional es el siguiente:

Haciendo uso de la tabla 3.3, la cual tiene a la Serie Residual  $R(t)$ , se obtienen una nueva tabla, obteniendo el promedio de cada fila de la tabla 3.3.

De esta forma una vez obtenido cada promedio de cada fila, se procede a obtener otro promedio el cual está compuesto de los promedios obtenidos, es decir, ahora se promedia  $\bar{w}(h)$  para obtener a  $\bar{W}(H)$ .

Denotaremos a cada fila de la tabla 3.3 como  $\bar{w}(h)$ .

$\bar{w}(h)$	$\bar{w}(h)$	$\bar{W}(H)$
$\bar{w}(1)$	1,002424	0,999714
$\bar{w}(2)$	1,003951	
$\bar{w}(3)$	0,996372	
$\bar{w}(4)$	0,996111	

Tabla 4.4: Variación Estacional

La tabla 3.4 tienen los cálculos precedentes. Una vez hecho lo anterior se procede a calcular la Estimación de la Estacionalidad  $E(t)$ . Para ello utilizaremos el modelo Mixto debido a que la literatura recomienda que en caso de Series de Índices se utilice el Modelos Mixtos.

El Modelo Mixto tiene la relación matemática siguiente:

$$\hat{E}(h) = w(h) - (\bar{W}(H) - 1)$$

Con la aplicación de este modelo se obtienen los valores  $\hat{E}(h)$  de la primera columna de la tabla 3.5, los cuales representan la estacionalidad. De aquí, cómo se esperaba, por el uso del Modelo Mixto se tiene que:

$$\bar{E}(h) = \frac{\sum_{i=1}^N E(h)}{N} = 1$$

Después de calcular la estacionalidad, se está listo para calcular la *Predicción*  $\hat{X}_n$ , para los siguientes cuartos, donde en este ejemplo, sería el cálculo del cuarto 21, 22, 23 y 24.

Para llevar a cabo el cálculo de la *Predicción*, primero se tienen que realizar el cálculo de la tendencia  $\hat{T}(t_i)$  para los cuarto antes mencionados.

Recordando que la relación de la tendencia para este ejemplo está definida por:

$$T(t) = -0,931372549t + 2918,862745$$

Al evaluar cada uno de los cuartos de segundo en la expresión de la tendencia  $T(t)$ , se arrojan los resultados de la segunda columna de la tabla 3.5.

$\hat{E}(h)$	$\hat{T}(t_i)$	Predicción $\hat{X}_n$	Segundo 19	E-Predicción
1,002709	2899,303922	2907,159564	2951	43,84043559
1,004236	2898,372549	2910,651532	2873	-37,65153196
0,996657	2897,441176	2887,755857	2954	66,24414315
0,996396	2896,509804	2886,072847	2928	41,92715337

Tabla 4.5: Predicción Lineal

Una vez calculada la tendencia futura  $\hat{T}(t_i)$ , se puede realizar el cálculo de la *Predicción Lineal*  $\hat{X}_n$  a través de la siguiente relación para el caso Mixto:

$$\hat{X}_n = \hat{T}(n+k) * \hat{E}(n+k)$$

Donde  $k$  es el horizonte de predicción o número de pasos adelante que se está prediciendo y  $n$  el origen de la Predicción.

Al sustituir los valores en la expresión de Predicción, se obtienen los valores de la tercera columna de la tabla 3.5.

Una vez obtenida la *Predicción*  $\hat{X}_n$ , se obtienen el *Error de Predicción*, el cual se calcula a partir del monitoreo de los respectivos cuartos del segundo 19, estos valores se encuentran en la cuarta columna de la tabla 3.5.

La relación matemática que describe el *Error de Predicción* [9] es:

$$e_n(k) = x(n+k) - \hat{X}_n(n+k)$$

Lo cual se resume en restarle el primer cuarto *real* del segundo 19 al primer cuarto de la *Predicción* del segundo 19. El *Error de Predicción* para cada cuarto se encuentra en la quinta columna (E-Predicción) de la tabla 3.5.

La sumatoria del *Error* es de 114.36 segmentos, esto se puede considerar como normal, sin embargo, lo ideal sería que este número tendiera a cero.

Lo siguiente es comparar los cuartos de la *Predicción* con la realidad para sacar conjeturas, con la finalidad de poder ver las diferencias importantes.

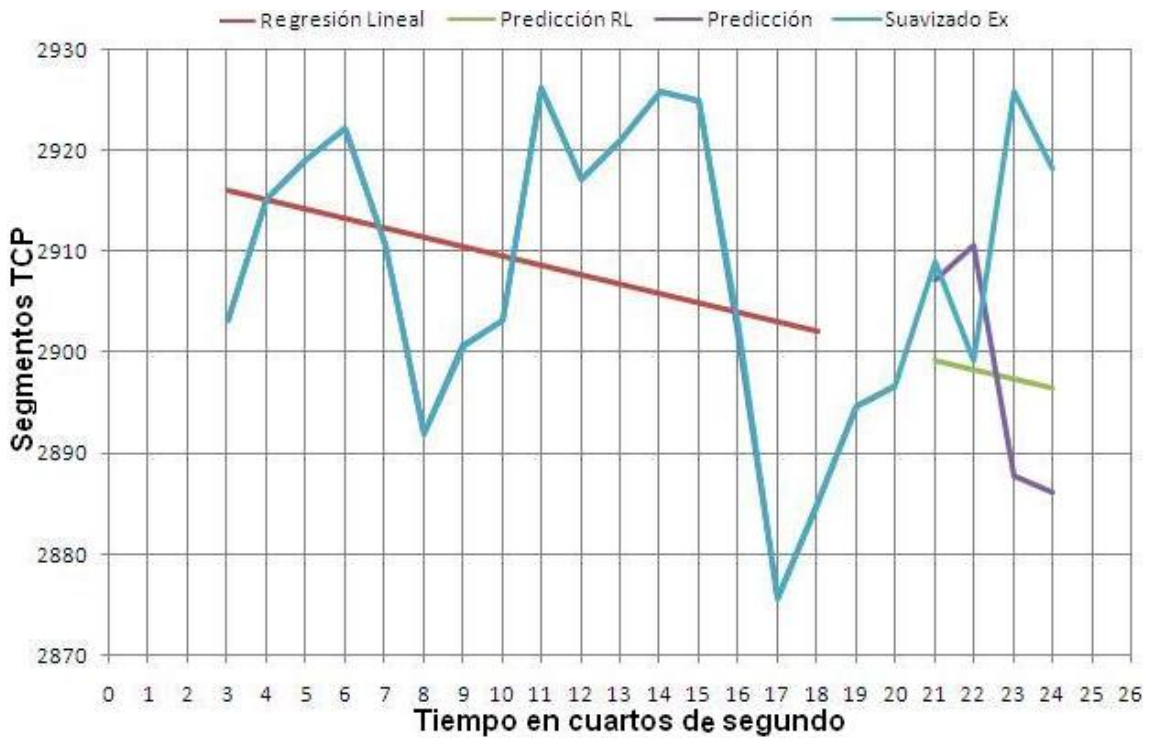


Figura 4.5: Predicción VS Segundo 19

La gráfica de la figura 3.5, muestra en color azul, el suavizado extendido (Suavizado Ex) el cual ha tomado en cuenta los cuartos del segundo 19. Por su parte la tendencia sigue fiel a su trayectoria en la recta verde.

Por otro lado la Predicción (en morado), parece que no es muy parecida a lo monitoreado en el segundo 19, no obstante, ésta, sigue de manera particular a la tendencia predicha, mientras que el tráfico aleatorio medido, no guarda una relación matemática con la tendencia como lo hace la Predicción.

En la siguiente sección se analizará el tráfico del siguiente segundo, cuando éste, se encuentre dentro del intervalo de tiempo del ataque DDoS.

Para lograr estos cálculos, así como las gráficas de esta sección, se utilizó la herramienta de Excel. El apéndice D, muestra los pasos seguidos en la hoja de cálculo mencionada.

## Predicción bajo UDP flooding

Básicamente este ataque consiste en generar grandes cantidades de paquetes UDP contra la víctima elegida. Debido a la naturaleza sin conexión del protocolo UDP, este tipo de ataques suele venir acompañado de IP spoofing.

UDP flooding envía una tormenta de datagramas UDP con el objetivo de saturar la capacidad de procesamiento del destino, que puede ser por ejemplo, un dispositivo de VoIP (Voz on IP).

Para utilizar este ataque, es preciso especificar la dirección IP de origen, dirección IP de destino, el puerto de origen y el de destino, y el número de datagramas a enviar.

La capacidad de resistencia a este ataque dependerá de las características de hardware de los dispositivos a atacar.

Es usual dirigir este ataque contra máquinas que ejecutan el servicio echo de forma que se generen mensajes echo de un elevado tamaño.

En esta sección se inicializará un ataque UDP flooding en el segundo 20, el cual nos ayudará a encontrar el punto donde activar una alarma para poner en marcha un mecanismo, que ayude a minimizar el mencionado ataque.

Para este apartado se monitorea el mismo esquema que en la sección anterior a lo largo de 5 segundos, tiempo durante el cual el servidor procesó segmentos TCP. Cada uno de los segundos monitoreados, fue dividido en cuartos de 250 ms de la misma forma en que se explica en la sección previa.

Los datos a partir del segundo 15 son los mismo, pues como el intervalo a evaluar es de 5 segundos, esto significa que el segundo 14 es removido y se lleva a cabo, un desplazamiento de una columna hacia la izquierda y en el hueco que se genera, se introduce la nueva medición, la cual está compuesta de los cuartos del segundo 19. El objetivo ahora es Predecir los cuartos del segundo 20, sin embargo, en el primer cuarto de éste segundo se dará el ataque DDoS y con esto, se verán diferencias importantes para llegar al \$Diferencial\$ buscado en este capítulo, para accionar el mecanismo de defensa contra el ataque DDoS que será discutido en el siguiente capítulo.

Cuartos	Segundo 15	Segundo 16	Segundo 17	Segundo 18	Segundo 19
249ms	2920	2917	2940	2868	2951
499ms	2891	2929	2948	2914	2873
749ms	2921	2933	2887	2902	2954
999ms	2864	2890	2872	2874	2928

Tabla 4.6: Serie de Tiempo

La primera tabla que representa el histórico estadístico o la Serie de Tiempo con relación a lo expuesto, es la tabla 3.6.

La siguiente tabla a obtener es la del suavizamiento de la Serie de Tiempo, los resultados del suavizamiento están en la tabla 3.7.

Cuartos	Segundo 15	Segundo 16	Segundo 17	Segundo 18	Segundo 19
200ms		2900,667	2921	2875,667	2909
400ms		2903,333	2926	2884,667	2899,333333
600ms	2910,667	2926,333	2925	2894,667	
800ms	2892	2917,333	2902,333	2896,667	

Tabla 4.7: Serie de Tiempo Suavizada

En seguida se calculará la Serie Residual, según lo expuesto en la sección anterior.

Cuartos	Segundo 15	Segundo 16	Segundo 17	Segundo 18	Segundo 19
200ms		1,005630889	1,006504622	0,997333952	1,014437951
400ms		1,008840413	1,007518797	1,010168708	0,990917452
600ms	1,00355016	1,002278164	0,987008547	1,002533395	
800ms	0,990318119	0,990630713	0,989548639	0,992174914	

Tabla 4.8: Serie Residual

Usando la serie suavizada:  $t$  y  $Z(t)$ . Para este caso en particular los valores de los parámetros  $a$  y  $b$  son:

$$a = -0,923529412$$

$$b = 2914,988725$$



Así la tendencia queda definida como:

$$T(t) = -0,923529412t + 2914,988725$$

De allí se obtienen la tabla  $W(H)$  la cual se muestra en la tabla 3.9.

$\bar{w}(h)$	$\bar{w}(h)$	$W(H)$
$\bar{w}(1)$	1,005976853	0,999962215
$\bar{w}(2)$	1,004361343	
$\bar{w}(3)$	0,998842566	
$\bar{w}(4)$	0,990668096	

Tabla 4.9: Variación Estacional

Una vez obtenida la tendencia se sigue con el cálculo de los demás parámetros. Todos y cada uno de ellos se calculan como se vio en la sección pasada.

Los resultados de estos cálculos son concentrados en la tabla 3.10.

$\hat{E}(h)$	$\hat{T}(t_i)$	Predicción $\hat{X}_n$	Segundo 20	E-Predicción
1,006014639	2895,594608	2913,010563	1608	-1305,010563
1,004399128	2894,671078	2907,405107	1425	-1482,405107
0,998880352	2893,747549	2890,50757	1449	-1441,50757
0,990705881	2892,82402	2865,93777	1271	-1594,93777

Tabla 4.10: Predicción Lineal

Como puede verse en la columna de la Predicción, se esperaban cuartos del orden de los 2900's sin embargo, debido al ataque en el primer cuarto del segundo 20, la cantidad de segmentos procesados por el servidor fue de aproximadamente la mitad o menos por cuarto.

Lo primero que destaca es el Error de Predicción, pues su sumatoria es de -5823,86101 segmentos TCP contra la sumatoria de 114.36 del cálculo de la sección pasada.

La figura 4.6 ilustra la gráfica de la Predicción Lineal, en este caso de color rojo. La Predicción parece ser la continuación de la gráfica del suavizamiento (en azul); también se observa que la Predicción corta en un punto a la recta de la Tendencia  $T(t_i)$  Predicha en color morado.



Figura 4.6: Predicción Lineal

Esta Predicción trata de decir como sería la cantidad de segmentos TCP recibidos en cada cuarto del segundo 20. Al observar la gráfica 3.6, podría asumirse que se podría recibir algo semejante a lo mostrado por la gráfica de la Predicción, sin embargo, como vimos en la sección pasada, esta idea es muy cuestionable, no obstante, aunque las gráficas no sean coincidentes, este método servirá para encontrar el Diferencial buscado, pues si observamos la gráfica 4.7 en los cuartos del segundo 20 monitoreados, nos damos cuenta de una caída muy prolongada cuando se está dando, el ataque DDoS.

Esta caída es ocasionada por el sobre-procesamiento del cual es objeto el Router (nodo 57 de la figura 3.1) con esto se consigue llenar el búfer de éste y comenzar a tirar los segmentos TCP de los clientes, los cuales obedeciendo el funcionamiento del protocolo TCP, disminuyendo su ventana y debido a que es muy complicado que uno de los segmentos TCP de un cliente llegue al servidor, la ventana de cualquier cliente se mantiene en el valor más bajo y esto deriva en varias situaciones, una de ellas es que el tiempo de la conexión espere en el lado del servidor y con ello se obligue al cliente a estar solicitando una nueva conexión a través de segmentos SYN. Un cliente que está en una situación semejante, ha perdido todo servicio proporcionado por el servidor.

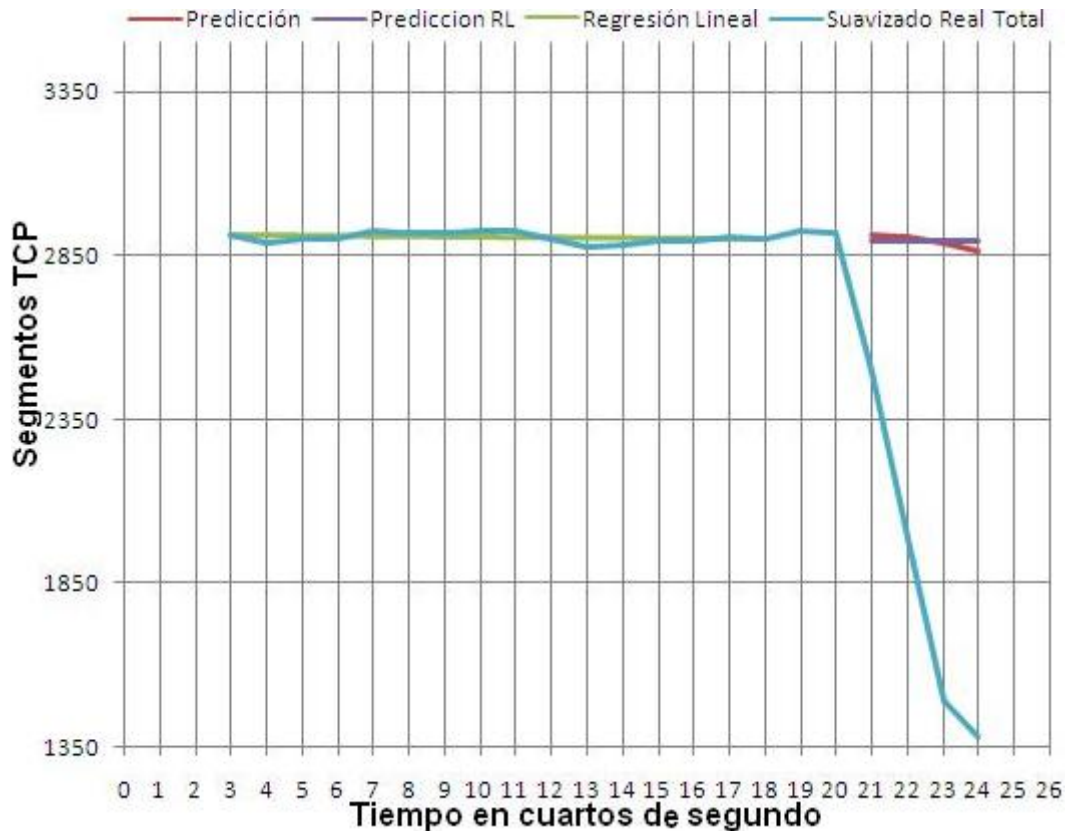


Figura 4.7: Predicción Lineal bajo DDoS

Por el lado del atacante, los zombies controlados no sufrirán las mismas consecuencias que cualquier cliente que utilice TCP. Debido a que el ataque es del tipo UDP flooding y éste tiene una tasa de transmisión constante. En este escenario la probabilidad de que la mayoría de los frames procesados por el servidor, pertenezcan a las máquinas zombi es muy elevada, es decir, si hablamos de una Botnet de 100 zombies enviando datagramas UDP, contra lo que podría ser, para el ejemplo tratado aquí, de a lo más 50 clientes en los tiempos de mayor estrés. Matemáticamente hablando la probabilidad de que un segmento TCP de un cliente llegue al servidor es de apenas del 33%, contra la probabilidad del 67% de que un datagrama UDP procesado por el servidor pertenezca a una máquina zombi. Para hacer disminuir este 33% de probabilidad de éxito de conexión por parte de un cliente legítimo, el atacante se da a la tarea de reclutar más zombies para agrandar su Botnet y hacer más letales los ataques DDoS.

Con la finalidad de encontrar un Diferencial entre los dos casos vistos en este capítulo, y activar un mecanismo de minimización de los efectos del ataque DDoS, se propone el siguiente criterio:

- 1.- Obtener del servidor, los primeros 5 segundos de tráfico y dividir cada segundo en cuartos.

- 2.- Grácar los cuartos de cada segundo y obtener su promedio de pendientes negativas ( $P_p n$ ) en segmentos TCP por segundo.
- 3.- Calcular el Promedio de los Promedios de las pendientes negativas ( $PP_p n$ ) de los 5 segundos.
- 4.- Calcular la *Predicción* para el siguiente segundo, primer cuarto.
- 5.- Obtener los valores REALES del primer cuarto del siguiente segundo.
- 6.- Calcular el *Error de Predicción* .
- 7.- Calcular cuantas veces cabe el ( $PP_p n$ ) en el *Error de Predicción* .
- 8.- Darle una calificación al cálculo previo.
- 9.- Aplicar el Criterio de RANGO, para discernir de una condición normal o de un ataque DDoS.

A manera de ejemplo se aplicará el Criterio de RANGO, en el segundo caso visto en este capítulo.

El primer paso habla de obtener los primeros 5 segundos de tráfico con la división propia en cuartos de cada uno, estos valores fueron mostrados en la tabla 4.6.

En el primer segundo (cuartos del 4-8), se observa sólo una pendiente negativa encerrada en un ovalo verde, la cual representa una caída de 9 segmentos TCP. Por su parte el siguiente segundo (cuartos 8-12) reporta dos pendientes negativas, una de 3 segmentos y otra de 20 segmentos. El tercer segundo (cuartos 12-16) tiene sólo una pendiente negativa de 28 segmentos TCP. El último segundo (cuarto 12-16) tiene dos pendientes negativas, una de 9 segmentos y otra de 8 segmentos respectivamente, tal como se observa en la figura 4.8.

El cálculo del promedio de cada segundo  $\frac{\sum_{i=1}^{N_p n} \text{segmentosTCP}(\text{pendienteNegativa})}{\text{NumeroPendientesNegativas}N_p n}$ , así cómo, el cálculo del ( $PP_p n$ ) se encuentra en la tabla 3.11

Segundos	Promedios	( $PP_p n$ )
Segundo 16	9	14.25
Segundo 17	11.5	
Segundo 18	28	
Segundo 19	8.5	

Tabla 4.11: Promedios de pendientes negativas y ( $PP_p n$ )



Figura 4.8: División en Cuartos para Obtener ( $PP_p n$ )

Lo siguiente es obtener el valor de la *Predicción*  $\hat{X}_n$  para el primer cuarto del siguiente segundo (cuarto 1 del segundo 20). Este y los demás valores que se tratarán para la obtención del *Diferencial*, están disponibles en la tabla 3.10. En particular  $\hat{X}_n = 2913,010$ .

Una vez obtenida la *Predicción* se debe de calcular el *Error de Predicción*, utilizando el valor REAL de los segmentos TCP procesados en el primer cuarto del siguiente segundo (1608 segmentos TCP), restándole el valor obtenido en la *Predicción*  $\hat{X}_n$  (2913,010563 segmentos TCP). Así, con lo anterior se obtiene un *Error de Predicción* de -1305,010563 segmentos TCP no procesados, para el primer cuarto del segundo 20. En este momento ya es posible construir el *Diferencial* (en valor absoluto), esto se hace tomando el *Error de Predicción* y dividirlo entre el ( $PP_p n$ ).

$$Diferencial = \left| \frac{-1305,010563}{14,25} \right|.$$

$$Diferencial = 92,58.$$

Para calificar a este *Diferencial*, se sugiere el siguiente esquema:

Los RANGOS de calificación (*Cal*), van del 1 al 20, según la tabla 3.12, en caso de que el *Diferencial* salga del rango, se le asignará la calificación de  $\infty$ , lo cual implica la activación inminente del mecanismo de defensa.

Rango	Calificación	Parámetro $K_j$
0 - 4	1	
5 - 9	2	
10 - 14	3	
15 - 19	4	
20 - 24	5	
25 - 29	6	
30 - 34	7	$\leftarrow K_7$
35 - 39	8	
40 - 44	9	
45 - 49	10	
50 - 54	11	
55 - 59	12	
60 - 64	13	
65 - 69	14	
70 - 74	15	
75 - 79	16	
80 - 84	17	
85 - 89	18	
90 - 94	19	
95 - 99	20	

Tabla 4.12: Rangos de Calificación

Obtenido el  $\$Diferencial\$, se procede a encontrar el Rango al que pertenece según la tabla 4.12. Una vez encontrado dicho Rango, se le asigna la calificación que le corresponde.$

Posteriormente esa calificación, adquiere un valor *booleano* (verdadero o falso), la cual se compara con la calificación apuntada por el apuntador  $K_j$ .

**SI**  $Cal < K_j$ .

Hacer Nada: Condiciones Normales

**SI NO**

Activar el mecanismo de protección

El mecanismo de protección será objeto del próximo capítulo. Es importante hacerle ver al lector que la calificación apuntada por el apuntador K<sub>j</sub>, así como los Rangos deben ser parámetros configurables para el administrador de redes, pues cada servidor podrá presentar condiciones diferentes, sin embargo, la metodología sería la misma.

Para el ejemplo de este trabajo, el rango máximo soportado es el calificado con \$6\$ y esto nos lleva soportar una pérdida máxima de  $413.25 ((PPpn) * 29)$  segmentos TCP, antes de activar un mecanismo de defensa.

Explicado lo anterior, se cae en la cuenta de que un ataque DDoS será detectado en los primeros 250 ms. Si se quisiera detectar antes, basta con reducir las porciones en las cuales está dividido cada segundo, es decir, en lugar de calcular cuartos podríamos calcular octavos con esto se podría detectar un ataque DDoS en 125 ms, de esta forma se podría configurar el tiempo de detección del ataque DDoS por parte de un administrador de redes.

Como consecuencia de la evaluación previa, si el Diferencial fuese calificado igual o mayor a K<sub>j</sub>, el servidor debe de enviar un mensaje al mecanismo de defensa para que éste se active a petición del servidor, usando algún mecanismo de autenticación, sin embargo, no se indagará a cerca de este mecanismo ya que sale de los límites de este trabajo.

#### Predicción bajo TCP flooding

En esta sección se analizará de nueva cuenta el caso de la sección previa, pero sobre los efectos que causaría ahora un ataque DDoS en su modalidad TCP flooding.

En el ataque DDoS en su modalidad TCP flooding, el atacante manda segmentos TCP con la bandera SYN = 1 y ACK = 0, haciendo con esto, una solicitud de conexión de tal manera que estas solicitudes se envíen mucho más rápido de lo que la víctima pueda procesarlas.

Para hacer lo anterior el atacante crea direcciones IP origen de manera aleatoria para ponerlas en cada paquete IP. En la parte de cada segmento TCP de cada paquete IP generado, la bandera SYN es puesta a uno y la bandera ACK es puesta a cero, con la finalidad de abrir una nueva conexión con el servidor víctima, hacia la dirección IP falsa.

La víctima responde a la IP falsa y posteriormente espera la confirmación que nunca llega, el tiempo de espera es del orden de 3 minutos, por dirección IP falsa.

Esto hace que la tabla de conexiones en el servidor víctima sea llenada; mientras éste se encuentra aguardando la respuesta de las IP falsas.

Entonces una vez que la tabla es llenada, toda nueva conexión es ignorada, si entre estas conexiones se encuentran usuarios legítimos, éstos también son ignorados y consecuentemente no pueden tener acceso al servicio prestado por el servidor.

Una vez que el atacante detiene el TCP flooding, el servidor vuelve a trabajar de manera normal.

Una solución es aumentar el número de conexiones soportadas por la aplicación o sistema operativo, con la finalidad de hacer más difícil el desbordamiento de las tablas de conexión de clientes, sin embargo, la vulnerabilidad persiste.

Después de estos antecedentes, es hora de tratar de encontrar en este ataque un Diferencial que permita activar un mecanismo de defensa, del cual se hablará en el siguiente capítulo.

La metodología para llevar a cabo este cometido, será estrictamente la misma de la sección anterior, con la salvedad de que ahora se obtendrá el \$Diferencial\$ cuando el ataque es con segmentos TCP.

Igual que en la sección pasada, se obtendrá la medición de cinco segundos previos, los cuales pasaran por los procesos ya descritos hasta obtener el Error de Predicción y las gráficas correspondientes.

Al hacer esto se obtienen la tabla 4.13

$\hat{E}(h)$	$\hat{T}(t_i)$	Predicción $\hat{X}_n$	Segundo 20	E-Predicción
1,006014639	2895,594608	2913,010563	3154	240,989437
1,004399128	2894,671078	2907,405107	3099	191,594893
0,998880352	2893,747549	2890,50757	3062	171,49243
0,990705881	2892,82402	2865,93777	3059	193,06223

Tabla 4.13: Predicción Lineal y Error de Predicción



Una vez hallados estos valores, se procede a encontrar el \$Diferencial\$, para ello se necesitara la ayuda de la gráfica suavizada de los 5 segundos monitoreados. La gráfica correspondiente se muestra en la figurara 3.9, de aquí se debe de hacer el cálculo de lo que denominaremos PPpp o promedio de promedios de pendientes positivas.

Calcular el PPpp es muy simple se hace igual que el cálculo PPpn de la sección anterior, solo que está vez en lugar de contar las pendientes negativas, se cuentan las pendientes positivas.

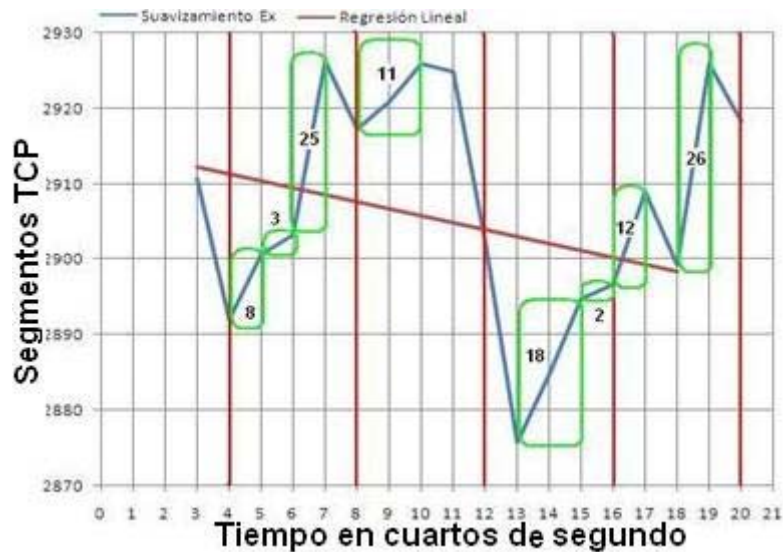


Figura 4.9: Gráfica PPs

La gráfica 4.9, ilustra los 5 segundos de monitoreo para el cálculo del PPpp que se utilizará para la obtención del Diferencial.

La gráfica 4.10 por su parte muestra la Predicción Lineal para el segundo 20, como se esperaba, ésta, se encuentra interceptando a la tendencia, sin embargo, el monitoreo real, eleva a la función azul, la cual representa al suavizamiento total de lo monitoreado debido al ataque TCP flooding.

Para este ejemplo el PPpp = 13. No obstante, la metodología para encontrar el Diferencial es diferente a como se hizo para UDP flooding, ya que en este caso, aparte de cuidar el Ancho de Banda, ahora se debe de cuidar el número máximo de conexiones que se pueden hacer con la aplicación, aunado a esto, las características de hardware disponibles en el servidor.



Figura 4.10: Predicción sobre TCP flooding

La idea fundamental para la obtención del \$Diferencial\$ se lista continuación:

- 1.- Calcular el  $PP_{pp}$ .
- 2.- Obtener el número Máximo de Conexiones (MaxCon) que puede soportar el servidor.
- 3.- Obtener el número de Clientes Conectados (ClientesCon) en el momento que se va a realizar el cálculo.
- 4.- Calcular las Conexiones Disponibles (ConDisp) en ese determinado momento, como la diferencia de  $MaxCon - ClientesCon$ .
- 5.- Obtener el valor de una Variable Porcentual (VarPor).
- 6.- Calcular las Conexiones en Contienda (ConC) como el producto de  $ConDisp * VarPor$ .
- 7.- Obtener el valor de  $K$  como el cociente de  $\frac{ConC}{PP_{pp}}$  y colocarlo como apuntador de un Rango.
- 8.- Calcular la Calificación (Cal) como el cociente de  $\frac{ErrordePrediccion}{PP_{pp}}$ .
- 9.- Comparar a  $Cal$  con lo apuntado por  $K$ .
- 10.- Aplicar el *Diferencial*.

Con la idea de dejar clara esta sección se calculará el \$Diferencial\$ para este caso en particular. El  $PP_{pp}$  ya se obtuvo, entonces según la lista precedente se debe de obtener al MaxCon, este parámetro se obtienen directo del servidor dependiendo de la aplicación, supongamos para este ejemplo que el MaxCon = 400.

A continuación se obtendrá a ClientesCon, este valor representa las conexiones activas en el quinto segundo, supóngase que para este ejemplo ClientesCon = 120.

En seguida se obtienen las conexiones disponibles:

$$\text{ConDisp} = 400 - 120$$

$$\text{ConDisp} = 280$$

Lo que prosigue es obtener a VarPor, este valor, es un valor que se encuentra en el intervalo [0,1] y se deriva del monitoreo propio de cada servidor, pues sirve para indicar que porcentaje de las conexiones por cuarto se deben conceder, por lo tanto este parámetro debe de poder ser configurado por el administrador de redes.

Para este ejemplo le daré el valor de VarPor = 0.5.

Para el siguiente paso, una vez ya conocido el valor de VarPor, se procede a calcular las conexiones en contienda, es decir, cuantas conexiones serán liberadas para los clientes por cuarto de segundo.

$$\text{ConC} = 280 * 0.5$$

ConC = 140 conexiones se liberarán de las 280 disponibles.

Lo siguiente es calcular el valor del apuntador  $K$  para conocer su posición en la escala de Rangos:

$$K = \frac{140}{13} \quad K = 10,76$$

Los RANGOS de calificación van del 1 al 19, según la tabla 3.14.

Rango	Calificación	Parámetro $K_j$
0 - 1	1	
2 - 3	2	
4 - 5	3	
6 - 7	4	
8 - 9	5	
10 - 11	6	$\leftarrow K_6$
12 - 13	7	
14 - 15	8	
16 - 17	9	
18 - 19	10	

Tabla 4.14: Rangos de Calificación

Entonces K es movido a la posición 6 en el Rango (10-11). K representa el límite máximo de conexiones a otorgar, según el Ppp de los últimos 5 segundos de monitoreo, pero en el Rango de calificaciones.

Llegado a este punto se calcula la Calificación:

$$Cal = \frac{240,989437}{13} \quad Cal = 18,5376$$

En este punto, ya se está listo para calcular el *Diferencial*, este se obtiene de la siguiente manera:

El *Diferencial* está dado por un valor *booleano*, obtenido al comparar el apuntador  $K_j$  con  $Cal$ , es decir:

SI  $Cal > K_j$

Activar Mecanismo de minimización de ataque *TCPFlooding*

SI NO

Hacer Nada: Actividad Normal

Como siempre se recomienda que los Rangos de Calificación sean configurables. Es de importancia recomendar que si la calificación obtenida es mayor las propuestas o configuradas, se le asigne el valor de  $\infty$  y se active de manera inminente el mecanismo de defensa contra el ataque *TCPflooding*.

Debido a que se está trabajando con tiempos muy pequeños, se hace de importancia conocer el tiempo de calculo, para todas y cada una de las operaciones precedentes, con la finalidad de poder activar una alarma, justamente pasando el cuarto de segundo, que se está analizando, pues ya sería mucho el daño, si se advirtiera de un ataque, uno o dos cuartos de segundo después de haberse detectado dicho ataque.

Está es la razón de porque justamente terminando el lapso de segundo analizado, se requiere saber si se está bajo ataque DDoS, de manera casi instantánea.

El apéndice C, describe los cálculos para hallar las cifras de todas las operaciones realizadas, así como su tiempo de ejecución, de tal forma que se tiene un cálculo de 7,4525 microsegundos, para TODAS las operaciones que se necesitan, para advertir de un ataque, justo terminando el lapso estudiado. Es de hacerse notar que el cálculo hecho no consume, ni siquiera la centésima parte de un milisegundo, entonces se podría dividir el segundo en porciones mucho más pequeñas que un cuarto con la finalidad de tener lo antes posible un aviso de ataque DDoS.

## Conclusiones del Capítulo

La adopción del sistema que tiene por objetivo detectar al UDP flooding, así como el sistema que hará lo mismo para el TCP flooding, son los dos brazos del mecanismo de defensa, los cuales estarán corriendo en el servidor a proteger, como un par de sensores, que tendrán la misión de dar aviso a dicho mecanismo para minimizar los efectos del DDoS, sobre el servidor mencionado.

La Predicción Lineal, es la herramienta fundamental para la obtención del  $\$Diferencial\$,$  pues en ambos sistemas es primordial conocer el Error de Predicción, el cual nos deja ver de manera clara en una gráfica, las condiciones normales, así como las diferencias que se presentan bajo un ataque DDoS.

La forma de obtener al Diferencial aparentemente parece algo complicada, pero al analizar que no existe forma o manera de diferenciar a un segmento TCP en cuanto a su estructura de un cliente legítimo de un segmento proveniente de una máquina zombi, vale la pena pagar el precio de hacer estos cálculos si con estos se logra activar un mecanismo para minimizar el daño causado por un ataque de esta naturaleza.

En el sentido de la obtención de los datos a través del simulador, se haría lo mismo con los datos obtenidos en un monitoreo de un servidor real, pues al final dicho monitoreo llegaría a una tabla de Series de Tiempo, de allí a una gráfica, sin importar que el servidor atendiera a 400 clientes o a 4000 clientes, el tratamiento de los datos recabados sería el mismo, de allí la importancia de dejar ciertos parámetros configurables en ambos sistemas, que cada administrador según sus mediciones debe de ajustar.

## Capítulo 5

### Minimización del ataque DDoS

#### Introducción

Una vez entendida la problemática descrita en el capítulo 2 y 3, que me llevo a la necesidad de tener un mecanismo que me advirtiera a cerca del inicio de un ataque, del cual, ya hable de su importancia en el capítulo 4, es momento de mostrar un diseño que de una solución solida para tratar al ataque DDoS en sus variantes TCP flooding y UDP flooding.

En estás variantes del ataque DDoS, tendremos como principales objetivos, dos importantes cuestiones, la primera es defender las conexiones activas, es decir, no permitir que el ataque destruya las conexiones existentes entre los clientes y el servidor.

La segunda por su parte, es hacer posibles las conexiones de clientes legítimos durante el ataque DDoS. Venciendo así, la suposición o creencia popular de que es imposible hacer una o varias conexiones legítimas, bajo el ataque DDoS, debido a que el servidor no estará disponible, pues es, de está forma que el atacante consigue su objetivo, al obligar al servidor a denegar los servicios ofrecidos.

Este diseño consistirá en la clasificación del tráfico, ya sea UDP o TCP. Una vez clasificado éste, se someterá a diferentes algoritmos tanto para tráfico TCP como UDP.

Los algoritmos para discernir las conexión legítimas o ilegítimas serán descritos a lo largo de este capítulo, así como, los detalles de diseño de la solución propuesta a través de un ejemplo en particular, que servirá para tener una mejor comprensión de la propuesta hecha en este trabajo.

## Descripción del la Propuesta de Tesis

El objetivo fundamental de este trabajo, es minimizar los efectos del ataque DDoS, respondiendo a las siguientes dos premisas:

- 1.- Se debe de garantizar, que las conexiones en curso no sea aniquiladas por el ataque DDoS.
- 2.- Se debe de tratar de aceptar, al mayor número de clientes legítimos durante el ataque DDoS.

Si tenemos estas dos premisas, como lo más importante a la hora de darle una solución a los momentos críticos del ataque DDoS, entonces, la solución no se puede esperar, que sea muy trivial, sin embargo, tampoco versará en la complejidad.

Una vez comprendida, la naturaleza del ataque DDoS, así como los efectos sobre sus víctimas, que al final realmente el servidor no es la víctima, sino los usuarios de dicho servidor, pues la denegación de sus solicitudes, los hace víctimas indirectas del ataque.

La propuesta de tesis, cuidará las dos premisas mencionadas, a continuación se describirá una solución para salvaguardar a estas dos premisas durante el ataque DDoS.

Como se mencionó en el capítulo 2, no es posible detener el ataque DDoS, en cuanto al su flujo de paquetes, sin embargo, aprovecharemos a nuestro favor su principal característica, la cual es el ser distribuido.

Se comenzará describiendo el esquema de la figura 5.1. En la parte más alta se encuentra Internet, la cual es la fuente del ataque DDoS y la puerta de entrada de los clientes del servidor, refiriéndonos con esto, a los clientes externos de la intranet exclusivamente.

Se colocará de manera estratégica, una Caja Negra entre Internet y el Router de la organización.

Esta caja se encargará de hacer una clasificación de tráfico UDP y TCP, para pasarlo, verificarlo y posteriormente a la verificación, se decidirá si continua hacia el Router o se encamina hacia el Hoyo Negro correspondiente.

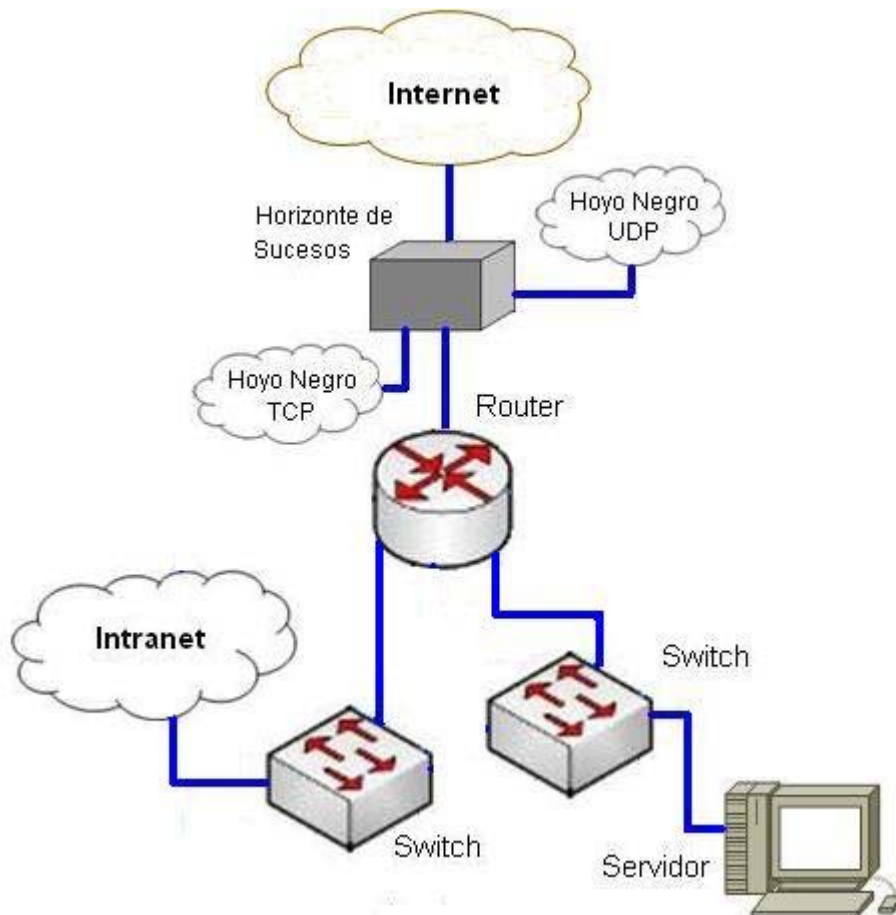


Figura 5.1: Posición Estratégica del Horizonte de Sucesos

En adelante veremos el funcionamiento de manera detallada de como trabaja el Horizonte de Sucesos. La primera idea fundamental se deriva de una frase clásica que dice: 'Divide y Vencerás'. La aplicación de esta idea se lleva cabo entendiendo que a un medio de comunicación como por ejemplo un cable UTP, que transporta frames Ethernet a su máxima capacidad, puede verse como un circuito eléctrico, el cual está formado por alambre del lado izquierdo, en medio una resistencia  $R_1$  y otro alambre del lado derecho. Este circuito transporta una corriente  $I$  a su máxima Intensidad soportada, como se ve en la figura 5.2 región A.

Entonces, si se pusiera una bifurcación en el extremo derecho del alambre, se podría hacer la pregunta: ¿Qué sucedería con la máxima Intensidad  $I$  a la que estaba sujeto dicho circuito?, la respuesta sería muy obvia, pues la Intensidad  $I$  se dividiría formando 2 Intensidades  $I_1$  e  $I_2$ .

La figura 5.2 región B, ilustra esta idea, viéndose en esta, cómo la  $I$  disminuye en la bifurcación, debido a que ahora la  $I$  tiene dos alambres en lugar de uno.



El efecto, entonces es, la disminución de la  $I$  en cada alambre de la bifurcación. La siguiente expresión modela el fenómeno de disminución de Intensidad,  $I = I_1 + I_2$ , siendo que  $R_1$  es diferente a  $R_2$  y diferente a  $R_3$ , con la finalidad de lograr diferentes  $I_i$  en cada brazo de la bifurcación.

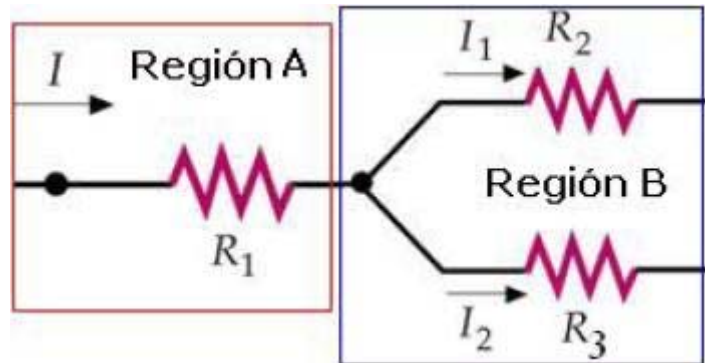


Figura 5.2: Bifurcación del Flujo

La aplicación de esta idea en el Horizonte de Sucesos es la siguiente: a la entrada de éste, se encuentra un clasificador, el cual dividirá el flujo de frames en datagramas UDP y segmentos TCP.

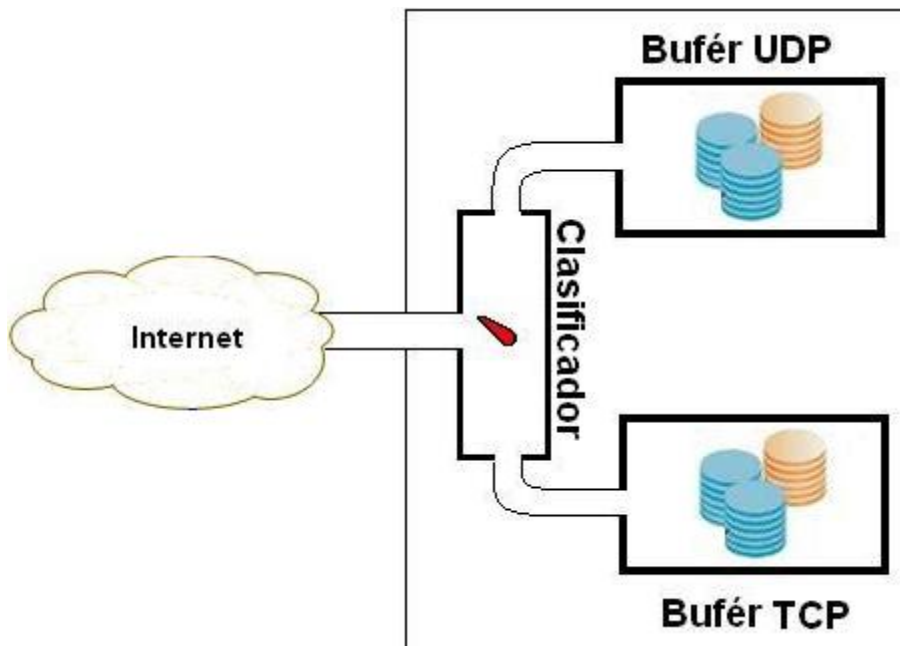


Figura 5.3: Clasificador del flujo

Cada frame será puesto en un búfer separado para su procesamiento. Con esto queda aplicada la idea precedente. La figura 5.3 muestra al clasificador conectado directamente a Internet.

Éste se encarga de mandar cada segmento TCP o datagramas UDP a su respectivo Búfer. Ya ahí, se seguirá otro procedimiento, para darle tratamiento a los datagramas y segmentos.

El diagrama de flujo siguiente, describe la clasificación y envío de datagramas y segmentos a sus respectivos buffers, para su almacenamiento.

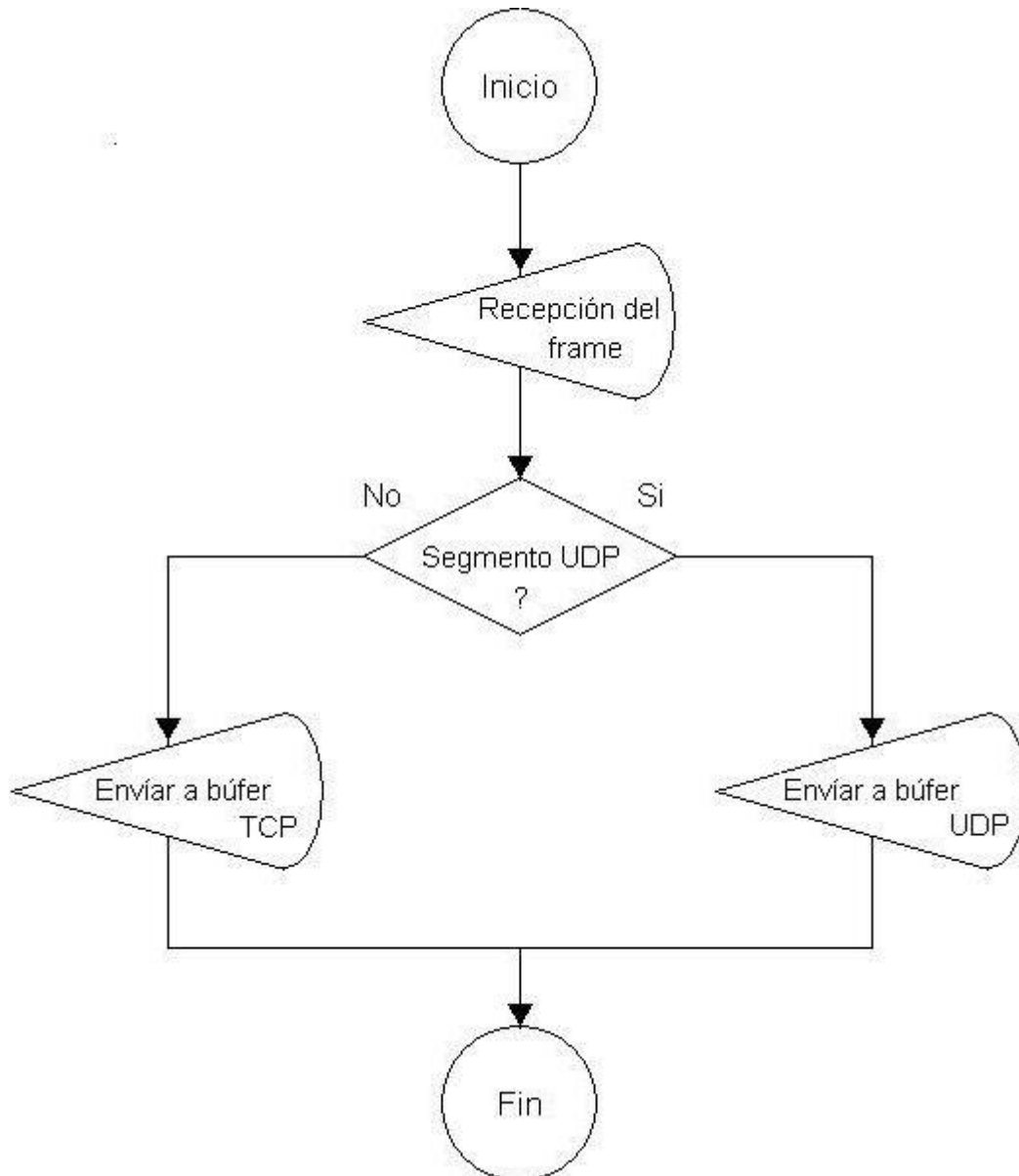


Figura 5.4: Diagrama de Flujo del Clasificador

El diagrama de flujo de la figura 4.4 desempaqueta el contenido dentro del frame Ethernet, para averiguar si se trata de un datagramas UDP o de un segmento TCP, así dependiendo de lo que sea éste, lo envía al búfer correspondiente.

El siguiente bloque será el del Despachador, éste tendrá la función de ir a uno de los dos buffers y extraer el primer frame de éste, para procesarlo de acuerdo a los criterios seleccionados para cada tipo de datagramas UDP o segmento TCP.

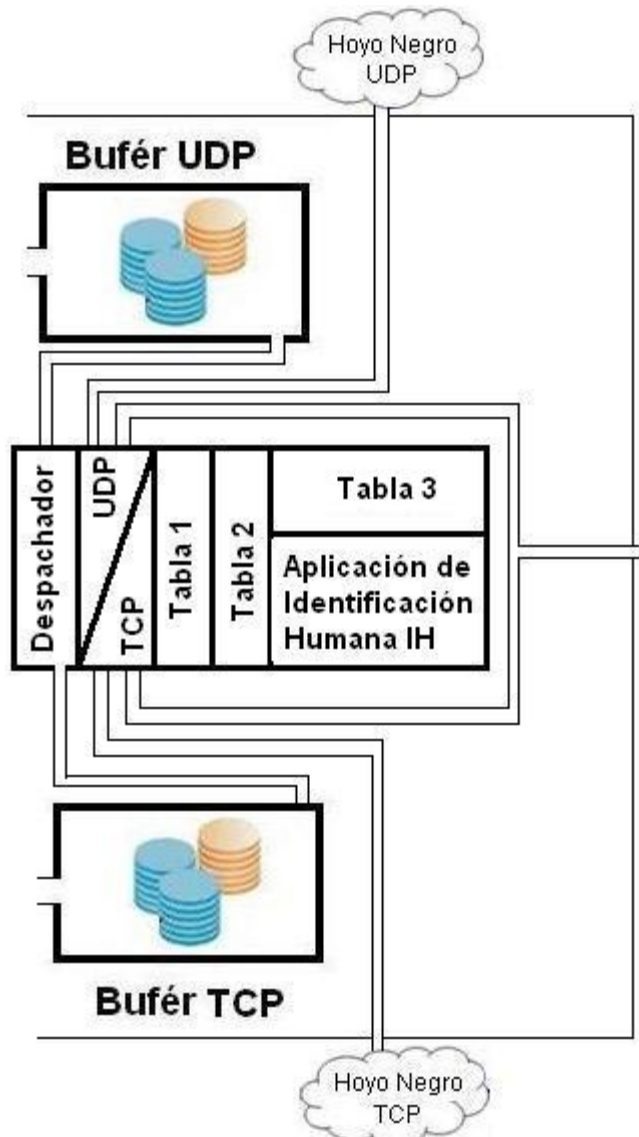


Figura 5.5: Estructura del Despachador

A continuación se describirá la explicación del algoritmo empleado para despachar los datagramas UDP. Debido a que este trabajo está dirigido a la protección de servidores de comercio electrónico, los cuales no requieren flujos de datagramas UDP desde Internet hasta éstos. Entonces, es por ello que los datagramas UDP tendrán una política de denegado, siendo entonces el tráfico UDP aceptado y enviado al Router, única y exclusivamente, si su destino NO es el servidor que se intenta proteger.

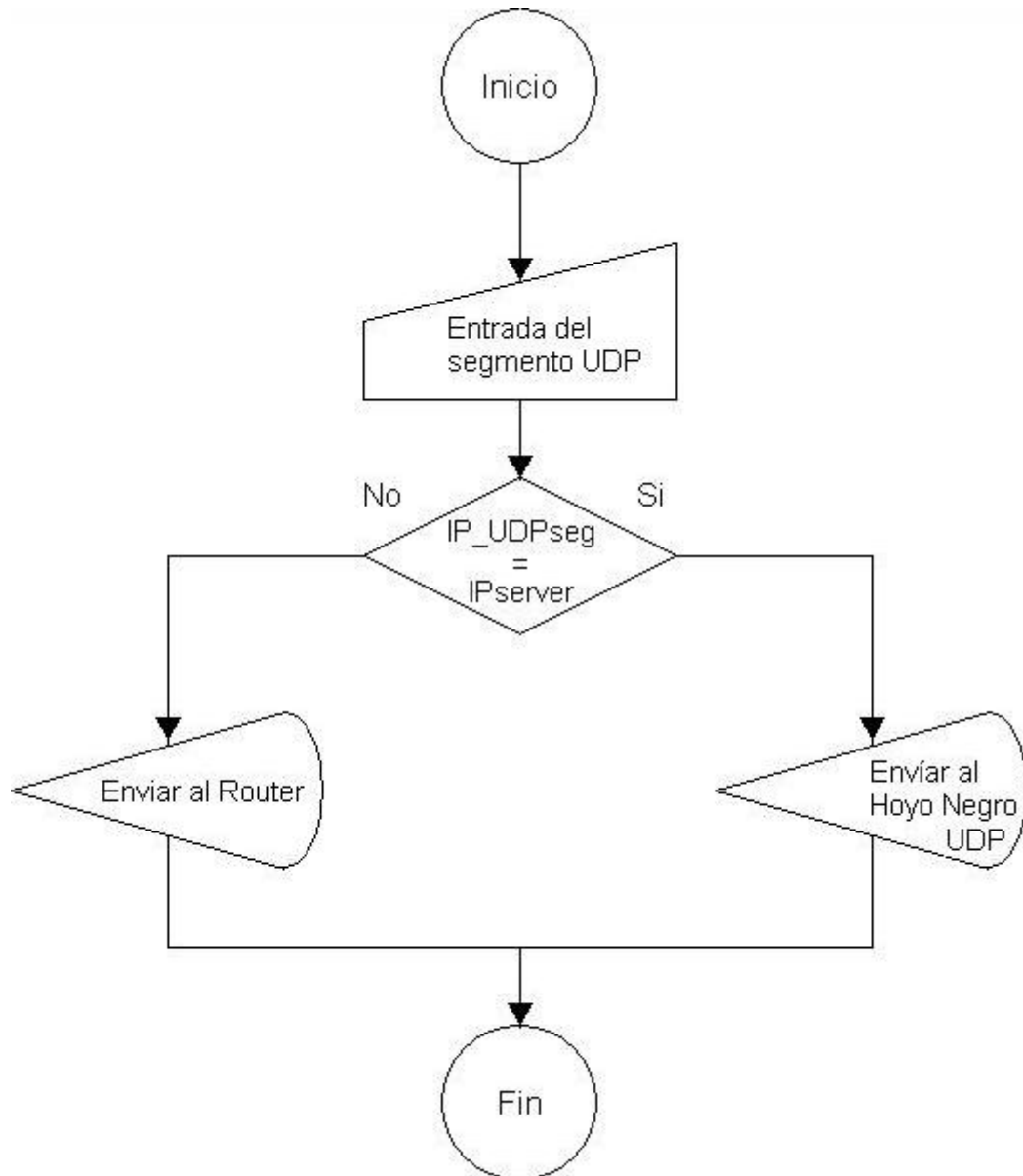


Figura 5.6: Diagrama de Flujo del Despachador UDP

La figura 4.6 describe el proceder de un diagrama de flujo para un datagrama UDP, es decir, si el datagrama UDP analizado, lleva en su área de datos, la dirección IP del servidor, éste será encaminado a un Hoyo Negro exclusivo para la destrucción de tráfico UDP ilegal que intenta ir hacia el servidor, este tráfico nunca llegará al Router y mucho menos al servidor, con esto evitamos el ataque DDoS del tipo UDP flooding.

Para el caso de la intranet que si acepta tráfico UDP, la regla es simplemente dejarlo pasar al Router, para que éste haga su trabajo.



Para solucionar este problema, haremos que la fortaleza del ataque DDoS, que recae sobre la distribución de zombies por todo el mundo, sea su principal debilidad.

La forma de hacer esto, es la siguiente:

Todo segmento que no este registrado en la Tabla 1, por NO tener como destino al servidor a proteger, será enviado al outer, sin más tramite, para que éste lo encamine a su destino.

Los segmentos TCP a examinar, son todos aquellos que tengan como dirección IP destino, la del servidor a proteger.

Los segmentos TCP, que no estén registrados en la Tabla 1, serán buscados en la Tabla 3, si la dirección IP origen del segmento TCP, se encuentra registrada en está tabla, será remitido al Hoyo Negro.

La Tabla 3 tiene registrados a todos aquellos segmentos que tienen su dirección IP sancionada y por consecuencia, estos frames son enviados al Hoyo Negro para su destrucción.

Todo segmento TCP con la bandera de SYN = 1 y la bandera ACK = 0, que tengan dirección IP del servidor como destino, deberán ser analizado antes de otorgarle el grado de cliente humano, para ello se hará uso de la debilidad ya mencionada del ataque, siendo está, la de que el atacante no puede estar en dos lugares al mismo tiempo y mucho menos en dos ciudades o continentes.

El primer paso del proceso de legitimidad de un cliente, es el vaciado de datos en la Tabla 2. Está tabla contendrá los datos de un posible cliente de manera temporal.

El formato de la Tabla 2 incluye, a demás de lo mencionado en la Tabla 1, dos campos más, el de la bandera SYN y el de la bandera ACK. La combinación de los valores de estos dos campos, son únicos, cuando un cliente desea hacer una conexión.

De está forma sabemos cuando el tráfico TCP es para establecer una nueva conexión o simplemente es la continuación de una conexión ya en existente.



Con lo anterior se logra destruir a los segmentos TCP enviados por los Zombis y la autenticación IH no le resulta molesta al cliente, pues para él todo se lleva cabo en unos cuantos segundos, siendo el proceso totalmente transparente, así de esta forma tratamos de captar la mayor cantidad de clientes para no caer en la denegación de servicio, aprovechando la naturaleza del ataque y explotando la flexibilidad del retraso sobre TCP.

Es importante hacerle ver al lector, que el clasificador sólo entra en funcionamiento cuando el modelo de Predicción Lineal le indica que está comenzando un ataque.

Así, mientras el modelo de Predicción Lineal NO señale el comienzo de un ataque, el Horizonte de Sucesos podría verse como una entidad inactiva sin causas de retraso y totalmente transparente al flujo de información, con la excepción del llenado de la Tabla 1.

La figura 4.9 muestra el diagrama de flujo del procesamiento de un segmento TCP. En este diagrama se ve más claro el flujo de procesos que se llevan a cabo.

En seguida se dará una breve explicación de la nomenclatura que contiene dicho diagrama de flujo:

1.- IPO = IPSERV: Compara la dirección IP del segmento TCP entrante con la dirección IP del servidor a proteger.

2.- IPO = RegT1: Se busca la dirección IP del segmento entrante en cada registro de la Tabla 1, para ver si se encuentra registrada como conexión activa.

3.- IPO = RegT3: Se busca la dirección IP del segmento entrante en cada registro de la Tabla 3, para ver si se encuentra registrada como IP sancionada.

4.- Información a T2: Esto implica que se llenarán los campos de la Tabla 2 con la información del segmento TCP que se está analizando.

5.- Registro T1: Quiere decir que al cliente se le da el calificativo de humano y se registra en la tabla de conexiones activas.

6.- Registro T3: Quiere decir que al cliente se le da el calificativo de NO humano y se registra en la tabla de direcciones IP sancionadas.



En principio se podría pensar que se necesitarán tablas muy grandes para llevar a cabo estos registros, además de una gran velocidad en el proceso de cada uno de los segmentos, sin embargo, en la siguiente sección donde se discutirá el diseño de un Horizonte de Sucesos, se caerá en la cuenta de que los requerimientos para la construcción de éste, no son para nada extraordinarios sino todo lo contrario.

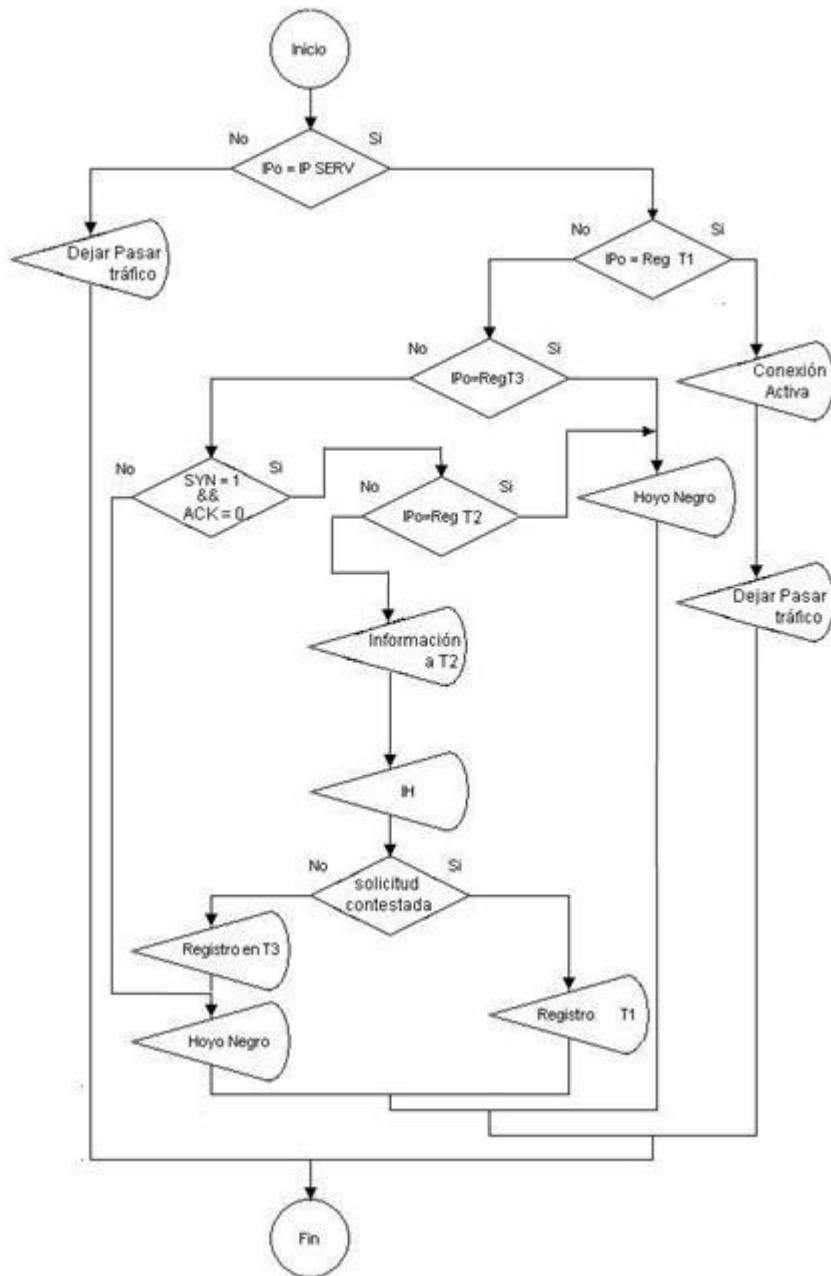


Figura 5.9: Diagrama de Flujo del Despachador TCP

La figura 5.10 muestra el esquema completo del Horizonte de Sucesos, con todos los bloques que lo componen y en el sentido que recibe los flujos.

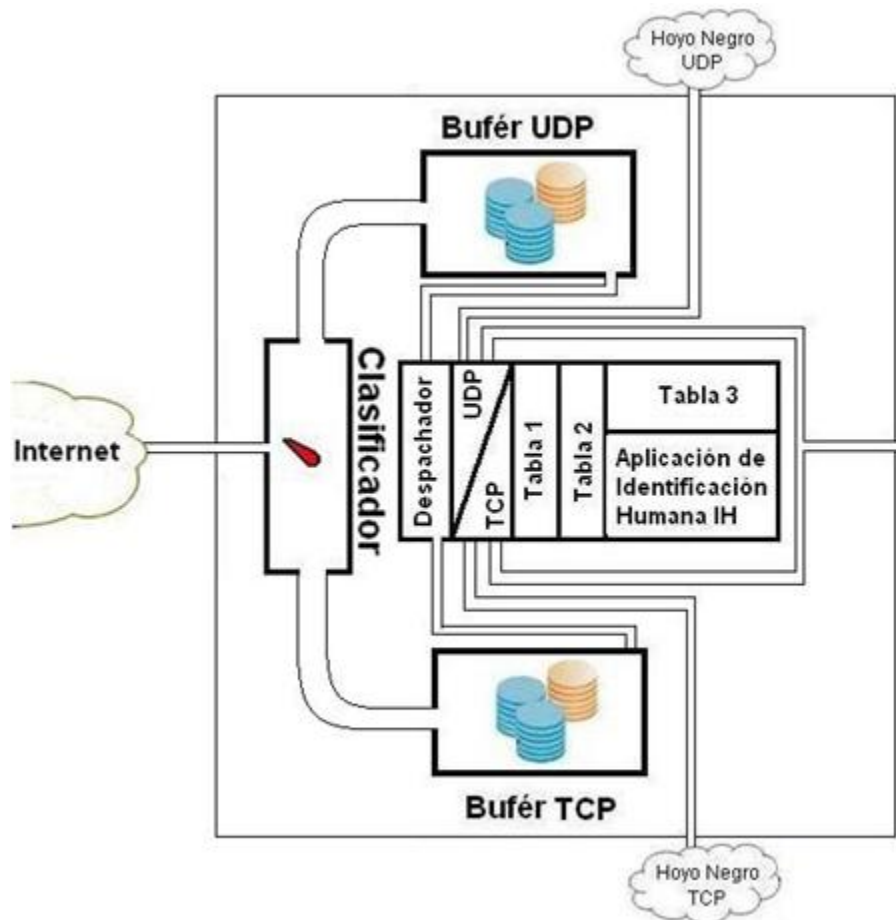


Figura 5.10: Esquema Completo del Horizonte de Sucesos

Un detalle importante a considerar, sería a cerca de cuantos datagramas UDP atender por uno segmento de TCP ( $M \times N$ ), es decir, un ejemplo sería, atender 3 datagramas UDP por un segmento TCP ( $1 \times 3$ ), este procedimiento buscaría minimizar el retardo sobre datagramas UDP, haciendo al Horizonte de Sucesos aún más eficiente.

El Horizonte de Sucesos se integraría como elemento activo dentro de una Red de Comunicaciones para una determinada organización y éste encontraría un nicho, al integrarse a la Instalación de Entrada del conjunto de estándares del Cableado Estructurado.

## Diseño de un Horizonte de Sucesos

En esta sección se explicará cómo diseñar un Horizonte de Sucesos. En este diseño se atenderá un problema en particular.

El problema a resolver, es la protección de un servidor del ataque DDoS en sus variantes SYN flooding, UDP flooding.

Las mediciones en el monitoreo de un determinado servidor, arrojan que un servidor puede atender hasta 50 clientes con gasto en memoria RAM de 128 MB, en las aplicaciones particulares requeridas por la organización para la cual se está haciendo el diseño. El monitoreo también arrojó que en las horas pico se han registrado hasta 390 conexiones simultáneas en este tipo de servidores en particular, ello implica que el servidor debe contar con una memoria RAM de al menos 998.4 MB, pero como la memoria RAM se compra generalmente en múltiplos de 2, seguramente se le asignará al servidor memoria RAM por 1024 MB, sólo para la atención de clientes.

Los servicios de telecomunicaciones llegan a través de un cable UTP categoría 5, el cual tiene una tasa de 100 Mbps como máximo al menos para este ejemplo, usando como protocolo de Acceso al Medio a Ethernet.

Para el diseño del primer módulo, o sea, el Clasificador, tenemos que, como cada frame Ethernet tiene a lo más 1518 bytes, pero mínimo 64 bytes, entonces con la idea de siempre trabajar con el peor de los casos, se tomará la longitud menor de un frame.

Si el medio soporta 100 Mbps entonces tendremos que soporta 12,500,000 bytes, ello implica que se tendrán como máximo 195,312 frames de 64 bytes por segundo, entonces la primera impresión señala que el clasificador debe de ser capaz de realizar esta cantidad de operaciones. Ahora si vemos que el clasificador, sólo decide, si el frame trae un segmento TCP o datagramas UDP y suponiendo que en eso se gastaran 2 ciclos de reloj (uno para la lectura y otro para la indagación), entonces el clasificador tendría que hacer 390,525 operaciones en un segundo.

Esta cantidad podría parecer elevada en un principio, pero si hacemos la analogía con un procesador de generaciones anteriores, digamos uno de 200 MHz, este procesador tiene la capacidad de hacer 200 millones de operaciones por segundo.

En estos términos, puede verse que a nivel procesamiento, la cantidad de operaciones pedidas al clasificador son realmente pocas y esto en el peor de los casos, es decir, que llegasen sólo frames de 64 bytes en un segundo, lo cual no es muy probable en las comunicaciones cotidianas.

Siguiendo con el diseño, lo siguiente sería calcular el tamaño de cada uno de los buffers. Para cualquiera de los dos, ya sea TCP o UDP, el peor de los casos sería que en un segundo todos los frames fueran ya sea UDP o TCP.

Bajo este argumento, la recomendación sería poner la capacidad de almacenamiento de cada búfer en función del ancho de banda del enlace, es decir, si tenemos en medio de transmisión de 100 Mbps, el tamaño de cada búfer se recomienda de 100 MB, ya que el despachador se encargará de vaciar el búfer mientras el clasificador lo está llenando, la siguiente sección se explicará cómo se llevará a cabo esta tarea. Lo explicado deja ver que el tamaño de cada búfer sería muy pequeño. No tendría sentido poner un tamaño mayor de búfer ya que nunca podrían entrar más flujos TCP o UDP por segundo a través del medio UTP.

Una vez explicados los diseños del clasificador y el tamaño de los buffers, se proseguirá con el despachador.

El despachador tendrá la tarea de ir al inicio de un búfer y extraer un frame que contiene un segmento TCP o datagramas UDP. Al extraer el frame, por ejemplo UDP, este datagrama será pasado a través del algoritmo ya explicado en la figura 4.6. La cual deja ver dos operaciones leer la IP destino y decidir si el paquete es enviado al Hoyo Negro a se deja pasar.

Por otro lado el Despachador tendrá una tarea más compleja cuando se trate de segmentos TCP, pues el diagrama de flujo mostrado en la figura 4.9, así lo señala.

Una vez que el despachador toma un segmento TCP de la base del búfer, lo primero que debe de hacer es verificar que el segmento va dirigido al servidor, si es así, lo que prosigue es compararlo con la Tabla 1, entonces se debe diseñar la Tabla 1, y la pregunta es ¿Cuál debe de ser la longitud de la Tala 1?, eso es fácil de contestar, si las mediciones en el monitoreo se tomaron a conciencia.

La longitud de la Tabla 1 en registros, será igual a la capacidad de atención a clientes en memoria RAM.

Un ejemplo de esto sería, si a un servidor se le asignará la cantidad 1024 MB en memoria RAM, para atender conexiones y según las mediciones del monitoreo cada usuario usa a lo más 2.56 MB, entonces, éste servidor podrá atender hasta 400 conexiones, he allí la longitud de la Tabla 1 en registros, es decir, se asignará un registro por conexión soportada por el servidor.

Como ya se explicó, la comparación de los datos del segmento TCP entrante, con los registros de la Tabla 1, servirá para diferenciar entre conexiones establecidas con el servidor e intentos de una nueva conexión con el servidor. Recordar que si el segmento TCP tiene una dirección IP diferente a la del servidor, se dejará pasar brincando todo proceso dentro del Horizonte de Sucesos.

Si el segmento se encuentra dentro de las conexiones establecidas de la Tabla 1, no hay más, que dejarlo pasar, en caso contrario, se debe de ir a buscar a la Tabla 3, la cual como ya se mencionó, se recomienda que sea idéntica a la Tabla 1. Si la consulta en está tabla da positivo, quiere decir que es una IP sancionada, por lo tanto el segmento será dirigido al Hoyo Negro. En caso de que la consulta haya salido negativa, se examinará el segmento para ver si tiene activada la bandera SYN y desactivada la bandera ACK, como ya se dijo.

Para el caso que se confirme como paquete de sincronía, se deben de llenar los campos de la Tabla 2. La estructura de está tabla ya fue mostrada en la figura 4.8, pero ¿qué tan larga en términos de registros debe ser?, cómo se hace evidente, los segmentos que se procesen con la Tabla 2 serán sometidos al IH, todos aquellos segmentos que no cumplan con el IH serán direccionados a la Tabla 3.

Una recomendación sería que la Tabla 2 tuviese el doble de longitud en registros que la Tabla 1, ya que así crece la posibilidad de capturar a más clientes legítimos, como ya se mencionó un cliente legitimo se identificara como humano bajo el IH y éste pasaría a la Tabla 1, mientras que los que no lo hagan pasarían a la Tabla 3.

Entonces es inevitable preguntar por la longitud de la Tabla 3. Está se recomendaría de 8 veces el tamaño de la Tabla 1, así las direcciones IP sancionadas tardarían mas en ser eliminadas por las nuevas IP sancionadas que entrasen a está tabla y en definitiva la Botnet que estuviera detrás del ataque ya tendría que ser de un tamaño considerable, mayor a la capacidad de almacenamiento de la Tabla 3, puesto que si no lo fuese justamente aquí se debilitaría considerablemente dicho ataque.

En resumen, las tablas tendrían una longitud recomendada de:

Tabla 1 = Número de conexiones Máximas soportadas por el servidor.

Tabla 2 = 2 veces la longitud de la Tabla 1 en registros.

Tabla 3 = 8 veces la longitud de la Tabla 1 en registros.

La longitud de las tablas pudiera indicar que la búsqueda de un segmento TCP con la dirección IP asociada será laboriosa y sobre todo tardada, pero realmente no será así, en la siguiente sección se describirá un algoritmo para la optimización de estas búsquedas sobre las tablas.

#### Optimización de Búsqueda en las Tablas

La eficiencia del Horizonte de Sucesos se puede mejorar muchísimo, si lo dotamos de un mecanismo de búsqueda sobre los tres tipos de tablas.

Comenzaremos describiendo la búsqueda de una dirección IP en la Tabla 1, para ver si pertenece a una conexión activa.

La búsqueda optimizada se llevará a cabo dividiendo la Tabla 1 en tres regiones adaptables, usando para ello 6 apuntadores de regiones de tablas y uno más que será auxiliar.

Las regiones en que será dividida la Tabla 1, guardan una estrecha relación con las clases de direcciones IP, es decir, la primera región será la A, la cual hace alusión a la clase A de direcciones IP, por consecuencia la segunda región será la que sirva para alojar a las direcciones IP de clase B y por último tendremos la tercera región reservada para las direcciones IP de clase C.

Los 6 apuntadores de regiones, son distribuidos de manera uniforme entre las tres regiones, esto nos da dos apuntadores por región. En esencia, estos apuntadores son los que hacen posible que cada una de estas regiones sea adaptable.

En este trabajo denominaremos adaptabilidad a la capacidad que tiene una región de cambiar su dimensión.

En la figura 4.10, se muestra a la Tabla 1, dividida en las tres regiones antes mencionadas, aunado a esto, se puede ver en el primer registro de la región A, hay un apuntador denominado A-Star, que hace referencia a que él es el apuntador que apunta al primer registro de la región A. Al final de la región A se puede ver a otro apuntador denominado A-End, el cual hace referencia al final de la región A.



Una vez obtenida la dimensión de cada región, se procede a entender la adaptabilidad, supóngase que la región A, B y C, se encuentran vacías y llegan los primeros tres segmentos TCP con dirección IP de clase A.

En este momento el apuntador A-End se debe de mover tres posiciones hacia abajo para apuntar al último registro de la región A; mientras los apuntadores de la región B y C, permanecen sin cambio alguno.

Ahora vamos a suponer que todos los flujos de los segmentos que llegan en un momento determinado, traen consigo direcciones IP de clase A, hasta agotar la mencionada región, y así como la gota que derrama un vaso con agua, llega el siguiente segmento con una dirección IP de clase A. La primera impresión sería que ya no existe un registro donde colocar a dicho segmento, pero debido a la característica de la adaptabilidad, no sucederá eso. Para evitar el desbordamiento en una región se hará uso de un apuntador auxiliar (este apuntador será creado sólo cuando se necesite), el cual tiene la tarea de ir a la siguiente región, directo a la posición del apuntador B-End de la región B, para este ejemplo, una vez allí, se verificará si los dos siguientes registros están vacíos, si es así, en el segundo registro vacío se guarda temporalmente la información del segmento en proceso de acomodar. Una vez guardados los datos del segmento, se recorre el apuntador B-End al registro inmediato inferior que ya se verificó que está vacío, posteriormente a esto, se recorre de manera horizontal hacia abajo un registro toda la pila de la región B hasta llegar al apuntador B-Start el cual al recorrerse una posición hacia abajo, genera el hueco para poder poner los datos del segmento en la región A que le corresponde para ello se copian los datos que se encuentran apuntados por el apuntador auxiliar y una vez hecho esto, el registro anfitrión es inicializado para su reuso.

En el caso de que la región B no hubiese tenido dos registros libres, el apuntador auxiliar, se dirigiría a la siguiente región, para este caso en particular, sería la región C, en busca de los dos registros necesarios para realizar el corrimiento tanto de la región C y posteriormente de la región B, para acomodar al segmento de la región A que llegó. Desde luego con esto se examina siempre el peor de los casos.

La recomendación para usar este tipo de búsqueda es aumentar la Tabla 1, en 6 registros adicionales y asignar dos registros más por región, con la idea de que siempre existan los dos registros buscados por el apuntador auxiliar. Algo más que se debe de incluir, es un contador del máximo de conexiones, para detener todo el proceso debido a que el servidor ha alcanzado su máximo número de clientes y éste ya no podrá atender a un cliente más, estuviese o no bajo ataque. La idea aquí es, poder acomodar las 400 conexiones, sin importar a que clase de direcciones IP, éstas pertenezcan con la ventaja de que se aumentaría en más de un 66% la velocidad de las búsquedas.

Debido a que cada Tabla, se dividirá en tres regiones, según la clase de dirección IP. Cuando llegue un segmento se preguntará a que clase pertenece, de esta forma si pertenece a la clase C, se iría directamente a el apuntador C-Start, de manera directa, sin tener que comparar todos y cada uno de los registros de la región A y la región B, con esto la búsqueda, se reduciría a un tercio del tiempo esperado.

Otra característica importante es el hecho de que si la región A cuenta con 134 registros inicialmente y de ellos sólo hay 20 ocupados, el apuntador A-End se encontrará en el último registro ocupado NO en el último registro de la región A. Esto ayuda muchísimo en lo siguiente: Supóngase que llega un segmento TCP con dirección IP de clase A y supóngase también que el segmento no es una conexión activa por lo tanto no se encuentra registrado en la región A, pero para que el sistema lo averigüe tienen que compararlo contra cada registro



de la región mencionada, sin embargo, cómo se dotó a dicha región con el apuntador A-End, la búsqueda se detendrá en el registro 20 y no en el registro 134, esta característica aumenta todavía más la eficiencia de la búsqueda que era del 66.6% obtenido con la división de la tabla en tres regiones.

Para terminar esta sección, pretendo quitar la posible falsa impresión de que las tablas podrían consumir demasiada memoria al almacenar tal cantidad de información, pues bien, esto definitivamente tampoco es verdad.

- Para almacenar una dirección IPv4, se necesitan 4 bytes (a.b.c.d).
- Para almacenar un puerto, se necesitan 2 bytes (0 - 65535).

Entonces un registro de la Tabla 1, ocuparía 12 bytes, si la longitud de la Tabla 1 es para 400 registros, esto da la cantidad de 4800 bytes o 4.8 KB de memoria consumidos por la Tabla 1.

En el caso de la longitud de la Tabla 2, se tienen que incluir dos bits más (SYN y ACK), por ello la longitud de un registro sería de 98 bits, pero como su longitud en registros es del doble de la Tabla 1, tendríamos que la memoria que ocuparía la longitud en registros de la Tabla 2 sería de 9.8 KB.

Por su parte la Tabla 3, que tendrá una longitud en registros de 8 veces lo de la longitud de la Tabla 1, entonces sería de 38.4 KB, recordando que la Tabla 1 y 3 son idénticas en su formato. Así la memoria total empleada apenas llegaría a los 53 KB.

Las búsquedas en las Tablas 2 y 3 se harían de la misma manera que para la Tabla 1.

## Conclusiones del Capítulo

Después del análisis de lo que sería esta caja negra, denominada Horizonte de Sucesos, de manera natural se cae en el caso clásico de los antibióticos, es decir, cuando se tiene una bacteria, se desarrolla un antibiótico para tratar a dicha bacteria, sin embargo, ésta, puede presentar resistencia con el tiempo al antibiótico, llevando a cabo una mutación. En el caso del ataque DDoS sucede exactamente lo mismo, pues éste sería la bacteria y el Horizonte de Sucesos sería el antibiótico.

Entonces, en definitiva, aunque el Horizonte de Sucesos consiga un buen desempeño a la hora de tratar los dos tipos de Ataque DDoS mencionados en este capítulo, el ataque DDoS puede mutar de tal forma que se convierta una nueva variante de dicho ataque, al grado de que el Horizonte de Sucesos, no pudiera hacer nada al respecto.

Sin embargo, igual que los investigadores en antibióticos, se tendría que hacer la investigación pertinente con la finalidad de poder desarrollar un Paquete de Servicio que fortalezca al Horizonte de Sucesos, con finalidad de atacar la nueva variante del ataque DDoS.

Explicado lo anterior, se tienen que a fin de cuentas el Horizonte de Sucesos es un sistema y como cualquier sistema, requiere mantenimiento. Es por ello que no se puede esperar que la solución planteada aquí sea definitiva.

Siempre que sea posible desarrollar una nueva variante del ataque DDoS, nunca habrá una solución general para dicho ataque, no obstante, se pueden ir atendiendo cada una de las mutaciones del ataque DDoS como se vayan generando. En este trabajo nos enfocamos a dos de las principales y dañinas variantes.

## Conclusiones de la Propuesta

En definitiva el ataque DoS con todas sus diferentes variantes, incluida la variante DDoS, es sumamente peligroso para los servidores que se dedican al comercio electrónico, debido a las pérdidas que se registran por efectos de las distintas variantes de este ataque. Por otro lado, las soluciones convencionales o las recomendaciones mencionadas por la literatura, no hace frente de manera sólida a la amenaza y debido a esto, este ataque es mucho más peligroso que un malware común que se pudiera esconder un equipo de oficina.

Entendida su naturaleza del ataque DDoS no se puede esperar resolver, con algún software antivirus o algo semejante, pues no es un módulo de software que se encuentre corriendo en el servidor con alguna firma fácil de detectar, sino que es una enorme oleada de tráfico, la cual hay que contener de alguna manera.

La solución propuesta en este trabajo, no versa en la sencillez, de un simple algoritmo, sin embargo, no es lo bastante complicada o compleja, para no implantarse, y minimizar los efectos del mencionado ataque.

Como se explico en el capítulo 5, los requerimientos tecnológicos son muy pocos, los módulos de software a diseñar son muy sencillos, las búsquedas en las tablas del Horizonte de Sucesos, fueron optimizadas y siendo un poco diestro en el lenguaje de programación que se deseará emplear, todavía se podría mejorar aún más.

Hablando del esquema visto en los ejemplos de:

Internet --- Horizonte de Sucesos --- Router --- Switch --- Servidor

Éste es opcional, pues el Horizonte de Sucesos podría adaptarse a uno o más esquemas tan complicados o tan simples como:

Internet --- Horizonte de Sucesos --- Servidor

No importa en realidad el arreglo o disposición donde poner al Horizonte de Sucesos mientras que sea antes del servidor.

La idea de dejar en la implantación del sistema varios parámetros a configurar por el administrador de redes, dota a este sistema de flexibilidad para un mejor desempeño en la obtención de resultados, entendiendo que diferentes escenarios requieren diferentes parámetros de funcionamiento, no obstante, la forma de tratar al problema sería la misma, al menos para las variantes aquí presentadas, pues como ya se dijo si el ataque sufre una mutación, de manera natural la solución también tendrá que sufrir un cambio.

De lo expuesto en los capítulos previos, cabe señalar que se debe de contar con una serie de políticas estrictas del funcionamiento y configuración del servidor a proteger, así como un mecanismo de monitoreo constante. Todo esto con la finalidad de tener un control total del comportamiento del servidor, objeto de múltiples eventualidades, como el ataque DDoS, a medida que se tenga bien medido el comportamiento del servidor, se podrán tener los mejores ajustes en los sistemas de sensores que correrán sobre éste, para tratar de evitar en la medida de lo posible las falsas alarmas al Horizonte de Sucesos.

Es evidente que la solución propuesta aquí se divide en dos partes la primera, son los dos sensores, tanto para UDP flooding como para TCP flooding que se encuentran corriendo sobre el servidor a proteger y la segunda, el mecanismo denominado Horizonte de Sucesos, dispuesto como primera frontera de Internet, sin embargo, las dos partes, cada una en su lugar, se encontrarán trabajando en conjunto para un mismo fin.

## Resumen

Existen otros trabajos donde se aplican las técnicas de Predicción Lineal en el área de Seguridad Informática. La Predicción Lineal a encontrado aplicación un muchas disciplinas y entre ellas, la de la seguridad en redes.

La Predicción es una pieza angular en este trabajo ya que sin ella no se tendría la flexibilidad de la detección oportuna del comienzo de un ataque DDoS.

Al tener un aviso de comienzo de ataque, justo a unos milisegundos de haber comenzado, es prudente tener un mecanismo de como minimizarlo. Es por ello que en este trabajo se habla de como diseñar un mecanismo de minimización del ataque DDoS en las variantes TCP flooding y UDP flooding.

El mecanismo de minimización explota la fortaleza del ataque mismo (el ser distribuido), para lograr contrarrestar el daño que pudiera ocasionar dicho ataque sobre las dos premisas más importantes:

- Conservar intactas las conexiones activas justo antes y en momentos donde el ataque este presente.
- La NO denegación del servicio cuando el ataque DDoS este en curso.

Al lograr conservar estas dos premisas, el ataque pierde efecto, al no poder alcanzar su objetivo de denegación del servicio y con ello se ve frustrado el intento del atacante por perjudicar a una determinada organización.

## Summary

Inside of security network, the Linear Prediction has taken a paper very important.

In this proposal, we use the Linear Prediction to detect a DDoS attack.

The Linear Prediction is the main piece, when is necessary to detect a DdoS attack just when it's beginning.

Alarm up allows activate a mechanism that minimizes DDoS attack, in this paper describes how to design a mechanism to make light of effects from this attack.

When a server is under DDoS attack in any of two variants: TCP flooding and UDP flooding. The most important is to protect two premises:

- The active connections most to survive during DDoS attack.
- The service offered by the server must be not refused when the server is under DDoS attack.

The minimization Mechanism exploit the strength from DDoS attack (to be Distributed) in order to counteract the damage than could cause this attack.

When these premises are achieved, the attack loss it's effectiveness, because it can't cause than the server denials it's services to any client.

## Apéndice A

### Código TCL del UDP flooding

```
#Crear el objeto simulador
set ddos [new Simulator]
set val(nn) 118
set rng [new RNG]
$rng seed 1.5
# Definiendo color de tráfico
$ddos color 0 yellow
$ddos color 1 orange
$ddos color 2 red
$ddos color 3 green
$ddos color 4 darkseagreen
$ddos color 5 wheat
$ddos color 7 violet
$ddos color 10 blue
#Archivo de trazado
set trazo [open ddostrv4.tr w]
$ddos trace-all $trazo
#Archivo de animación
set anime [open ddosnamv4.nam w]
$ddos namtrace-all $anime
#Creación de nodos
for {set i 0} {$i < $val(nn)} {incr i} {
    set ws($i) [$ddos node]
}
#Conexión Atacante -- Maestros
for {set i 1} {$i < 4} {incr i} {
    #$ddos duplex-link $ws($i) $ws([expr ($i+1) %7]) 1Mb 10ms DropTail
    $ddos duplex-link $ws($i) $ws(0) 10Mb 1ms DropTail
}
```

```

#Conexión maestro 1 -- Agentes
for {set i 4} {$i < 21} {incr i} {
    #$ddos duplex-link $ws($i) $ws([expr ($i+1) %7]) 1Mb 10ms DropTail
    $ddos duplex-link $ws($i) $ws(1) 10Mb 1ms DropTail
}
#Conexión maestro 2 -- Agentes
for {set i 21} {$i < 39} {incr i} {
    #$ddos duplex-link $ws($i) $ws([expr ($i+1) %7]) 1Mb 10ms DropTail
    $ddos duplex-link $ws($i) $ws(2) 10Mb 1ms DropTail
}
#Conexión maestro 3 -- Agentes
for {set i 39} {$i < 57} {incr i} {
    $ddos duplex-link $ws($i) $ws(3) 10Mb 1ms DropTail
}
#Conexión de los agentes -- Router
for {set i 4} {$i < 21} {incr i} {
    $ddos duplex-link $ws($i) $ws(57) 10Mb 1ms DropTail
}
for {set i 21} {$i < 39} {incr i} {
    $ddos duplex-link $ws($i) $ws(57) 10Mb 1ms DropTail
}
for {set i 39} {$i < 57} {incr i} {
    $ddos duplex-link $ws($i) $ws(57) 10Mb 1ms DropTail
}
#Conexión Router - Switch-LAN
$ddos duplex-link $ws(57) $ws(58) 10Mb 1ms DropTail
#Conexión Router - Switch - Sever
$ddos duplex-link $ws(57) $ws(59) 100Mb 1ms DropTail
#Conexión Switch -Server - Server
$ddos duplex-link $ws(59) $ws(60) 100Mb 1ms DropTail
#Conexión Switch - Intranet
$ddos duplex-link $ws(58) $ws(61) 100Mb 1ms DropTail
# Conexión de Routers Clientes- Router Server
for {set i 62} {$i < 70} {incr i} {
    $ddos duplex-link $ws($i) $ws(57) 100Mb 1ms DropTail
}
# Route 1 - 6 clientes 62. 70-75
for {set i 70} {$i < 76} {incr i} {
    $ddos duplex-link $ws($i) $ws(62) 10Mb 1ms DropTail
}
# Route 2 - 6 clientes 63: 76-81
for {set i 76} {$i < 82} {incr i} {
    $ddos duplex-link $ws($i) $ws(63) 10Mb 1ms DropTail
}

```



```

# Route 3 - 6 clientes 64: 82-87
for {set i 82} {$i < 88} {incr i} {
    $ddos duplex-link $ws($i) $ws(64) 10Mb 1ms DropTail
}
# Route 4 - 6 clientes 65: 88-93
for {set i 88} {$i < 94} {incr i} {
    $ddos duplex-link $ws($i) $ws(65) 10Mb 1ms DropTail
}
# Route 5 - 6 clientes 66: 94-99
for {set i 94} {$i < 100} {incr i} {
    $ddos duplex-link $ws($i) $ws(66) 10Mb 1ms DropTail
}
# Route 6 - 6 clientes 67: 100-105
for {set i 100} {$i < 106} {incr i} {
    $ddos duplex-link $ws($i) $ws(67) 10Mb 1ms DropTail
}
# Route 7 - 6 clientes 68: 106-111
for {set i 106} {$i < 112} {incr i} {
    $ddos duplex-link $ws($i) $ws(68) 10Mb 1ms DropTail
}
# Route 7 - 6 clientes 69: 111-116
for {set i 112} {$i < 118} {incr i} {
    $ddos duplex-link $ws($i) $ws(69) 10Mb 1ms DropTail
}
#-Tráfico de Clientes
# Router 1 - 6 clientes
for {set i 70} {$i < 76} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp($i)
    $tcp($i) set class_ 7
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
}
$ddos at 24.84 "$ftp(70) start"
$ddos at 14.56 "$ftp(71) start"
$ddos at 2.32 "$ftp(72) start"
$ddos at 2.04 "$ftp(73) start"
$ddos at 7.24 "$ftp(74) start"
$ddos at 2.28 "$ftp(75) start"
#Receptor
set sink70 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink70
$ddos connect $tcp(70) $sink70
$ddos at 25.96 "$ddos detach-agent $ws(70) $tcp(70);

```

```

$ddos detach-agent $ws(60) $sink70"
set sink71 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink71
$ddos connect $tcp(71) $sink71
$ddos at 21.0 "$ddos detach-agent $ws(71) $tcp(71);
$ddos detach-agent $ws(60) $sink71"
set sink72 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink72
$ddos connect $tcp(72) $sink72
$ddos at 12.04 "$ddos detach-agent $ws(72) $tcp(72)
;$ddos detach-agent $ws(60) $sink72"
set sink73 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink73
$ddos connect $tcp(73) $sink73
$ddos at 17.56 "$ddos detach-agent $ws(73) $tcp(73)
;$ddos detach-agent $ws(60) $sink73"
set sink74 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink74
$ddos connect $tcp(74) $sink74
$ddos at 26.92 "$ddos detach-agent $ws(74) $tcp(74)
;$ddos detach-agent $ws(60) $sink74"
set sink75 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink75
$ddos connect $tcp(75) $sink75
$ddos at 16.6 "$ddos detach-agent $ws(75) $tcp(75)
;$ddos detach-agent $ws(60) $sink75"
# Router 2 - 6 clientes
for {set i 76} {$i < 82} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp($i)
    $tcp($i) set class_ 7
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
}
$ddos at 16.0 "$ftp(76) start"
$ddos at 26.68 "$ftp(77) start"
$ddos at 19.8 "$ftp(78) start"
$ddos at 6.68 "$ftp(79) start"
$ddos at 0.24 "$ftp(80) start"
$ddos at 2.68 "$ftp(81) start"
#Receptor
set sink76 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink76

```

```

$ddos connect $tcp(76) $sink76
$ddos at 21.2 "$ddos detach-agent $ws(76) $tcp(76);
$ddos detach-agent $ws(60) $sink76"
set sink77 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink77
$ddos connect $tcp(77) $sink77
$ddos at 30.4 "$ddos detach-agent $ws(77) $tcp(77);
$ddos detach-agent $ws(60) $sink77"
set sink78 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink78
$ddos connect $tcp(78) $sink78
$ddos at 36.92 "$ddos detach-agent $ws(78) $tcp(78)
;$ddos detach-agent $ws(60) $sink78"
set sink79 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink79
$ddos connect $tcp(79) $sink79
$ddos at 26.36 "$ddos detach-agent $ws(79) $tcp(79);
$ddos detach-agent $ws(60) $sink79"
set sink80 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink80
$ddos connect $tcp(80) $sink80
$ddos at 3.4 "$ddos detach-agent $ws(80) $tcp(80)
;$ddos detach-agent $ws(60) $sink80"
set sink81 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink81
$ddos connect $tcp(81) $sink81
$ddos at 33.92 "$ddos detach-agent $ws(81) $tcp(81)
;$ddos detach-agent $ws(60) $sink81"
# Router 3 - 6 clientes
for {set i 82} {$i < 88} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp($i)
    $tcp($i) set class_ 7
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
}
$ddos at 24.24 "$ftp(82) start"
$ddos at 4.48 "$ftp(83) start"
$ddos at 21.08 "$ftp(84) start"
$ddos at 13.06 "$ftp(85) start"
$ddos at 5.92 "$ftp(86) start"
$ddos at 15.24 "$ftp(87) start"

```

```

#Receptor
set sink82 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink82
$ddos connect $tcp(82) $sink82
$ddos at 26.32 "$ddos detach-agent $ws(82) $tcp(82)
;$ddos detach-agent $ws(60) $sink82"
set sink83 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink83
$ddos connect $tcp(83) $sink83
$ddos at 15.48 "$ddos detach-agent $ws(83) $tcp(83)
;$ddos detach-agent $ws(60) $sink83"
set sink84 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink84
$ddos connect $tcp(84) $sink84
$ddos at 27.12 "$ddos detach-agent $ws(84) $tcp(84)
;$ddos detach-agent $ws(60) $sink84"
set sink85 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink85
$ddos connect $tcp(85) $sink85
$ddos at 22.88 "$ddos detach-agent $ws(85) $tcp(85)
;$ddos detach-agent $ws(60) $sink85"
set sink86 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink86
$ddos connect $tcp(86) $sink86
$ddos at 10.44 "$ddos detach-agent $ws(86) $tcp(86)
;$ddos detach-agent $ws(60) $sink86"
set sink87 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink87
$ddos connect $tcp(87) $sink87
$ddos at 26.64 "$ddos detach-agent $ws(87) $tcp(87)
;$ddos detach-agent $ws(60) $sink87"
# Router 4 - 6 clientes
for {set i 88} {$i < 94} {incr i} {
set tcp($i) [new Agent/TCP]
$ddos attach-agent $ws($i) $tcp($i)
$tcp($i) set class_ 7
set ftp($i) [new Application/FTP]
$ftp($i) attach-agent $tcp($i)
}
$ddos at 0.68 "$ftp(88) start"
$ddos at 1.08 "$ftp(89) start"
$ddos at 14.44 "$ftp(90) start"
$ddos at 31.32 "$ftp(91) start"
$ddos at 13.96 "$ftp(92) start"

```

```

$ddos at 3.8 "$ftp(93) start"
#Receptor
set sink88 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink88
$ddos connect $tcp(88) $sink88
$ddos at 8.8 "$ddos detach-agent $ws(88) $tcp(88)
;$ddos detach-agent $ws(60) $sink88"
set sink89 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink89
$ddos connect $tcp(89) $sink89
$ddos at 39.36 "$ddos detach-agent $ws(89) $tcp(89)
;$ddos detach-agent $ws(60) $sink89"
set sink90 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink90
$ddos connect $tcp(90) $sink90
$ddos at 26.90 "$ddos detach-agent $ws(90) $tcp(90)
;$ddos detach-agent $ws(60) $sink90"
set sink91 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink91
$ddos connect $tcp(91) $sink91
$ddos at 35.16 "$ddos detach-agent $ws(91) $tcp(91)
;$ddos detach-agent $ws(60) $sink91"
set sink92 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink92
$ddos connect $tcp(92) $sink92
$ddos at 31.64 "$ddos detach-agent $ws(92) $tcp(92)
;$ddos detach-agent $ws(60) $sink92"
set sink93 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink93
$ddos connect $tcp(93) $sink93
$ddos at 7.88 "$ddos detach-agent $ws(93) $tcp(93)
;$ddos detach-agent $ws(60) $sink93"
# Router 5 - 6 clientes
for {set i 94} {$i < 100} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp($i)
    $tcp($i) set class_ 7
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
}

```

```

$ddos at 4.44 "$ftp(94) start"
$ddos at 1.92 "$ftp(95) start"
$ddos at 15.36 "$ftp(96) start"
$ddos at 9.84 "$ftp(97) start"
$ddos at 3.88 "$ftp(98) start"
$ddos at 26.96 "$ftp(99) start"
#Receptor
set sink94 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink94
$ddos connect $tcp(94) $sink94
$ddos at 9.0 "$ddos detach-agent $ws(94) $tcp(94)
;$ddos detach-agent $ws(60) $sink94"
set sink95 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink95
$ddos connect $tcp(95) $sink95
$ddos at 5.16 "$ddos detach-agent $ws(95) $tcp(95)
;$ddos detach-agent $ws(60) $sink95"
set sink96 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink96
$ddos connect $tcp(96) $sink96
$ddos at 20.92 "$ddos detach-agent $ws(96) $tcp(96)
;$ddos detach-agent $ws(60) $sink96"
set sink97 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink97
$ddos connect $tcp(97) $sink97
$ddos at 37.44 "$ddos detach-agent $ws(97) $tcp(97)
;$ddos detach-agent $ws(60) $sink97"
set sink98 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink98
$ddos connect $tcp(98) $sink98
$ddos at 12.64 "$ddos detach-agent $ws(98) $tcp(98)
;$ddos detach-agent $ws(60) $sink98"
set sink99 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink99
$ddos connect $tcp(99) $sink99
$ddos at 37.0 "$ddos detach-agent $ws(99) $tcp(99)
;$ddos detach-agent $ws(60) $sink99"
# Router 6 - 6 clientes
for {set i 100} {$i < 106} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp($i)
    $tcp($i) set class_ 7
}

```

```

set ftp($i) [new Application/FTP]
$ftp($i) attach-agent $tcp($i)
}
$ddos at 6.4 "$ftp(100) start"
$ddos at 34.2 "$ftp(101) start"
$ddos at 16.32 "$ftp(102) start"
$ddos at 4.8 "$ftp(103) start"
$ddos at 25.56 "$ftp(104) start"
$ddos at 13.24 "$ftp(105) start"
#Receptor
set sink100 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink100
$ddos connect $tcp(100) $sink100
$ddos at 19.6 "$ddos detach-agent $ws(100) $tcp(100)
;$ddos detach-agent $ws(60) $sink100"
set sink101 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink101
$ddos connect $tcp(101) $sink101
$ddos at 35.36 "$ddos detach-agent $ws(101) $tcp(101)
;$ddos detach-agent $ws(60) $sink101"
set sink102 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink102
$ddos connect $tcp(102) $sink102
$ddos at 34.2 "$ddos detach-agent $ws(102) $tcp(102)
;$ddos detach-agent $ws(60) $sink102"
set sink103 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink103
$ddos connect $tcp(103) $sink103
$ddos at 17.16 "$ddos detach-agent $ws(103) $tcp(103)
;$ddos detach-agent $ws(60) $sink103"
set sink104 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink104
$ddos connect $tcp(104) $sink104
$ddos at 25.96 "$ddos detach-agent $ws(104) $tcp(104)
;$ddos detach-agent $ws(60) $sink104"
set sink105 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink105
$ddos connect $tcp(105) $sink105
$ddos at 22.68 "$ddos detach-agent $ws(105) $tcp(105)
$ddos detach-agent $ws(60) $sink105"

```

```

# Router 7 - 6 clientes
for {set i 106} {$i < 112} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp($i)
    $tcp($i) set class_ 7
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
}
$ddos at 6.04 "$ftp(106) start"
$ddos at 31.6 "$ftp(107) start"
$ddos at 14.08 "$ftp(108) start"
$ddos at 11.04 "$ftp(109) start"
$ddos at 16.08 "$ftp(110) start"
$ddos at 10.72 "$ftp(111) start"
#Receptor
set sink106 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink106
$ddos connect $tcp(106) $sink106
$ddos at 34.52 "$ddos detach-agent $ws(106) $tcp(106)
;$ddos detach-agent $ws(60) $sink106"
set sink107 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink107
$ddos connect $tcp(107) $sink107
$ddos at 35.6 "$ddos detach-agent $ws(107) $tcp(107)
;$ddos detach-agent $ws(60) $sink107"
set sink108 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink108
$ddos connect $tcp(108) $sink108
$ddos at 37.96 "$ddos detach-agent $ws(108) $tcp(108)
;$ddos detach-agent $ws(60) $sink108"
set sink109 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink109
$ddos connect $tcp(109) $sink109
$ddos at 25.92 "$ddos detach-agent $ws(109) $tcp(109)
;$ddos detach-agent $ws(60) $sink109"
set sink110 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink110
$ddos connect $tcp(110) $sink110
$ddos at 21.98 "$ddos detach-agent $ws(110) $tcp(110)
;$ddos detach-agent $ws(60) $sink110"
set sink111 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink111

```



```

$ddos connect $tcp(111) $sink111
$ddos at 25.0 "$ddos detach-agent $ws(111) $tcp(111)
;$ddos detach-agent $ws(60) $sink111"
# Router 8 - 6 clientes
for {set i 112} {$i < 118} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp($i)
    $tcp($i) set class_ 7
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
}
$ddos at 10.72 "$ftp(112) start"
$ddos at 27.36 "$ftp(113) start"
$ddos at 20.44 "$ftp(114) start"
$ddos at 1.52 "$ftp(115) start"
$ddos at 10.72 "$ftp(116) start"
$ddos at 7.64 "$ftp(117) start"
#Receptor
set sink112 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink112
$ddos connect $tcp(112) $sink112
$ddos at 25.2 "$ddos detach-agent $ws(112) $tcp(112)
;$ddos detach-agent $ws(60) $sink112"
set sink113 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink113
$ddos connect $tcp(113) $sink113
$ddos at 32.08 "$ddos detach-agent $ws(113) $tcp(113)
;$ddos detach-agent $ws(60) $sink113"
set sink114 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink114
$ddos connect $tcp(114) $sink114
$ddos at 23.2 "$ddos detach-agent $ws(114) $tcp(114)
;$ddos detach-agent $ws(60) $sink114"
set sink115 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink115
$ddos connect $tcp(115) $sink115
$ddos at 6.64 "$ddos detach-agent $ws(115) $tcp(115)
;$ddos detach-agent $ws(60) $sink115"
set sink116 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink116
$ddos connect $tcp(116) $sink116

```

```

$ddos at 34.36 "$ddos detach-agent $ws(116) $tcp(116)
;$ddos detach-agent $ws(60) $sink116"
set sink117 [new Agent/TCPSink]
$ddos attach-agent $ws(60) $sink117
$ddos connect $tcp(117) $sink117
$ddos at 33.6 "$ddos detach-agent $ws(117) $tcp(117)
;$ddos detach-agent $ws(60) $sink117"
#-ATAQUE DDoS UDP flooding
#Conección del Atacante con Maestros NULL y UDP: 1-4
for {set i 1} {$i < 4} {incr i} {
    set udp($i) [new Agent/UDP]
    $ddos attach-agent $ws(0) $udp($i)
    $udp($i) set class_ 0
    set cbr($i) [new Application/Traffic/CBR]
    $cbr($i) set interval_ 0.05
    $cbr($i) attach-agent $udp($i)
    set null($i) [new Agent/Null]
    $ddos attach-agent $ws($i) $null($i)
    $ddos connect $udp($i) $null($i)
    $ddos at 20.0 "$cbr($i) start"
}
#Maestro 1 - Agentes TCP-SINK 4-21
for {set i 4} {$i < 21} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws(1) $tcp($i)
    $tcp($i) set class_ 1
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
    $ddos at 20.001 "$ftp($i) start"
    set sink($i) [new Agent/TCPSink]
    $ddos attach-agent $ws($i) $sink($i)
    $ddos connect $tcp($i) $sink($i)
    $ddos at 40.0 "$ddos detach-agent $ws(1) $tcp($i)
;$ddos detach-agent $ws($i) $sink($i)"
}
#Maestro 2 - Agentes TCP-SINK 21-39
for {set i 21} {$i < 39} {incr i} {
    set tcp($i) [new Agent/TCP]
    $ddos attach-agent $ws(2) $tcp($i)
    $tcp($i) set class_ 1
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
    $ddos at 20.001 "$ftp($i) start"
    set sink($i) [new Agent/TCPSink]
    $ddos attach-agent $ws($i) $sink($i)
    $ddos connect $tcp($i) $sink($i)
    $ddos at 40.0 "$ddos detach-agent $ws(1) $tcp($i)
;$ddos detach-agent $ws($i) $sink($i)"
}
#Maestro 3 - Agentes TCP-SINK 39-57
for {set i 39} {$i < 57} {incr i} {
    set tcp($i) [new Agent/TCP]

```

```

$ddos attach-agent $ws(3) $tcp($i)
$tcp($i) set class_ 1
set ftp($i) [new Application/FTP]
$ftp($i) attach-agent $tcp($i)
$ddos at 20.001 "$ftp($i) start"
set sink($i) [new Agent/TCPSink]
$ddos attach-agent $ws($i) $sink($i)
$ddos connect $tcp($i) $sink($i)
$ddos at 40.0 "$ddos detach-agent $ws(1) $tcp($i)
$ddos detach-agent $ws($i) $sink($i)"
}
#Agentes - Víctima
for {set i 4} {$i < 57} {incr i} {
  set udp($i) [new Agent/UDP]
  $ddos attach-agent $ws($i) $udp($i)
  $udp($i) set class_ 2
  set cbr($i) [new Application/Traffic/CBR]
  $cbr($i) set interval_ 0.0005
  $cbr($i) attach-agent $udp($i)
  set null($i) [new Agent/Null]
  $ddos attach-agent $ws(60) $null($i)
  $ddos connect $udp($i) $null($i)
  $ddos at 20.002 "$cbr($i) start"
}
puts "Comenzar la Simulación del Ataque DDoS con una BOTNET
de 51 nodos sobre el server con 50 clietes..."
$ddos at 40.0 "finish"
proc finish {} {
  global ddos trazo anime
  $ddos flush-trace
  close $trazo
  close $anime
  puts "Comenzar la Animación del Ataque DDoS con una BOTNET de 51 nodos..."
  exec nam ddosnamv4.nam &
  exit 0
}
$ddos run

```

## Apéndice B

### Código TCL del TCP flooding

Del código del Apéndice A, la única parte que se tienen que cambiar, para conseguir el efecto de ataque DDoS en su versión TCP flooding, es la siguiente:

```
#Agentes - Víctima
for {set i 4} {$i < 57} {incr i} {
    set tcp1($i) [new Agent/TCP]
    $ddos attach-agent $ws($i) $tcp1($i)
    $tcp1($i) set class_ 1
    set ftp1($i) [new Application/FTP]
    $ftp1($i) attach-agent $tcp1($i)
    $ddos at 20.001 "$ftp1($i) start"
    set sink($i) [new Agent/TCPSink]
    $ddos attach-agent $ws(60) $sink1($i)
    $ddos connect $tcp1($i) $sink1($i)
    $ddos at 20.002 "$ddos detach-agent $ws($i) $tcp1($i)
; $ddos detach-agent $ws(60) $sink1($i)"
}
```

## Apéndice C

### Tiempo de Ejecución

En este apéndice se calculará el tiempo de ejecución consumido por las operaciones que se llevan a cabo, para encontrar el Error de Predicción, así como, las que se necesitan, con el fin de encontrar el Diferencial, para UDPflooding y TCPflooding.

Se comenzará por contar el número de operaciones aritméticas (sumas, restas, multiplicaciones y divisiones), sin olvidar las operaciones lógicas (AND, OR, NOT).

Para calcular el Error de Predicción se requiere las operaciones mostradas en la tabla C.1.

$t$	$Z(t)$	$\bar{t}$	$(t_i - \bar{t})$	$(t_i - \bar{t})^2$	$Z(t)(t_i - \bar{t})^2$	$Z(t)$	$R(t)$
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Tabla 1: Parámetros del Cálculo de Tendencia y Serie Residual

En el caso del suavizado que sigue la relación  $Z(t) = \frac{t_i + t_{(i+1)} + t_{(i+2)}}{\text{orden}3}$  para  $t = 1, 2, 3, \dots, 16$ , nos deja con un total de 32 sumas y 16 divisiones.

El promedio del tiempo  $\bar{t}$ , se calcula con 16 sumas y una división.

En el caso de la diferencia entre  $(t_i - \bar{t})$ , se tienen 16 restas.

Para la columna  $(t_i - \bar{t})^2$  se calcula haciendo el producto  $(t_i - \bar{t}) * (t_i - \bar{t})$ , dando un total de 16 multiplicaciones.

La columna  $Z(t)(t_i - \bar{t})^2$ , se calcula empleando 16 multiplicaciones.

El promedio del suavizado  $z(t)$  se calcula con 16 sumas y una división.

La serie  $R(t) = \frac{X(t)}{Z(t)}$ , se calcula con 16 divisiones.

El parámetro a

$a = \frac{\sum_{i=\text{orden3}}^{N_{\text{orden}}} Z(t_i)(t_i - \bar{t})}{\text{sum}_{i=\text{orden3}}^{N_{\text{orden}}} (t_i - \bar{t})^2}$  requiere de 32 sumas y una división.

El cálculo del parámetro b

$b = Z(\bar{t}) - \bar{t}$ , requiere de una resta y una multiplicación.

$\bar{w}(h)$	$\bar{w}(h)$	$\bar{W}(H)$
$\bar{w}(1)$	.	.
$\bar{w}(2)$	.	.
$\bar{w}(3)$	.	.
$\bar{w}(4)$	.	.

Tabla 2: Operaciones de la tabla W(h)

La columna  $\bar{w}(h)$  de la tabla C.2, requiere 16 sumas y 4 divisiones.

La columna  $\bar{W}(H)$ , requiere 4 sumas y 1 división.

$\hat{E}(h)$	$\hat{T}(t_i)$	$\hat{X}_n$	$e_n$
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Tabla 3: Parámetros

En el caso de la Estimación de la Estacionalidad.

$\hat{E}(h) = w(h) - (\bar{W}(H) - 1)$ , el número de operaciones 8 restas y 4 multiplicaciones.

La tendencia futura\

$\hat{T}(t) = at + b$ , se realizan 4 sumas y 4 multiplicaciones, para  $t = 21, 22, 23, 24$ .

La Predicción Lineal

$\hat{X}_n = T(n + k) * E(n + k)$ , utiliza 4 multiplicaciones.

El Error de la Predicción

$e_n(k) = x(n + k) - \hat{X}_n(n + k)$ , ejecuta 4 restas.

De esto, se tiene el resumen, expuesto en la tabla C.4.

<i>Sumas</i>	<i>Restas</i>	<i>Multiplicaciones</i>	<i>Divisiones</i>	<i>TOTAL</i>
120	29	45	41	235

Tabla 4: Cantidad de operaciones

Por otro lado se tiene que la cantidad de operaciones que se deben realizar para calcular el UDPflooding están dadas de la siguiente manera:

Lo primero es saber, si en un cuarto se refleja una pendiente positiva o una pendiente negativa, para esto, se toma el valor de la ordenada justo al empezar un cuarto y luego se sustrae el valor de la ordenada al término de dicho cuarto.

$$C = Y_t - Y(t + 250ms)$$

1. SI  $C \leq 0$

2. fue una Pendiente Positiva

3. SI NO

4. fue una Pendiente Negativa

Para esto se requieren 16 restas, 16 sumas y 16 comparaciones para los 5 segundos.

Para el promedio de pendientes negativas  $P_p n$ :

$P_b = \frac{\sum_{i=1}^4 \text{segmentosTCP}(\text{pendientenegativa})_i}{N_p n}$ , para lo cual se requiere de N sumas y una división por segundo.

Para el ejemplo de la tabla 3.11 fueron en total por los 5 segundos: 2 sumas y 4 divisiones.

Para el cálculo de  $PP_p n$ :

$PP_p n = \frac{\sum_{i=1}^4 P_p n_i}{4}$ , para lo cual se requiere de 4 sumas y una división.

En el caso del Diferencial:

$Diferencial = \left\lfloor \frac{e_n}{PP_p n} \right\rfloor$ , sólo se necesita una división.

Para buscar el *Diferencial* en la tabla de calificaciones (tabla 3.12), se debe de ejecutar el siguiente pseudocódigo:

```

1. cal = 1, flag = 0;
2. limiteInferior = 0, limiteSuperior = 4
3. while(flag != 1)
4. {
5.   if(Diferencial ≥ limiteInferior AND Diferencial ≤ limiteSuperior)
6.   {
7.     cal = cal;
8.     flag = 1;
9.     if(cal ≥ K)
10.    {
11.      Activarmecanismo de proteccion
12.    }
13.   else
14.   {
15.     NOmecanismo de proteccion
16.   } // fin del if - else interno
17. }
18. else
19. {
20.   limiteInferior + = 5;
21.   limiteSuperior + = 5;
22.   cal + = 1;
23. } // fin del if - else
24. } // fin del while

```



Del pseudocódigo anterior se obtienen 81 comparaciones y 60 sumas.

Por otro lado se tiene que la cantidad de operaciones que se deben realizar para calcular el TCPflooding están dadas de la siguiente manera:

El cálculo de una pendiente positiva o de una pendiente negativa se efectúa igual que en la sección pasada.

$$C = Y_t - Y(t + 250ms)$$

1.  $SIC \leq 0$
2. fue una pendiente positiva
3. SINO
4. fue una Pendientes Negativa

Para esto se requieren 16 restas, 16 sumas y 16 comparaciones para los 5 segundos.

Para el promedio de pendientes positivas  $P_p n$ :

$P_p n = \frac{\sum_{i=1}^4 \text{segmentosTCP(pendientepositivas)}_i}{N_{pp}}$ , para lo cual se requiere de N sumas y una división por segundo.

Para el ejemplo de la gráfica 3.9 fueron en total por los 5 segundos: 7 sumas y 4 divisiones.

Para el cálculo de  $PP_{pp}$ :

$PP_{pp} := \frac{\sum_{i=1}^4 P_{b_i}}{4}$ , para lo cual se requiere de 4 sumas y una división.

El cálculo de ConDisp:

ConDisp = MaxCon - ClientesCon, para lo cual se requiere de una resta.

En el caso del cálculo de ConC:

ConC = ConDisp \* VarPor, para lo cual se requiere de una multiplicación.

Para obtener el valor de  $K$  se hace el cálculo:

$$K = \frac{ConC}{PP_p}, \text{ para lo cual se requiere de una división.}$$

En este punto se lleva al parámetro  $K$ , a su posición en la tabla 3.14, con el siguiente pseudocódigo:

```
1. flag = 0, calificacion = 1
2. limiteInferior = 0, limiteSuperior = 1
3. while(flag != 1)
4. {
5.   if ( $K \geq$  limiteInferior AND  $K \leq$  limiteSuperior)
6.   {
7.     K = calificacion
8.     flag = 1
9.   }
10.  else
11.  {
12.    limiteInferior + = 2;
13.    limiteSuperior + = 2;
14.    calificacion + = 1;
15.  } // findelif - else
16. } // findelwhile
```

Este pseudocódigo consume a lo más 40 comparaciones y 30 sumas.

Ahora para realizar el cálculo de la Calificación Cal:

$$Cal = \left\lfloor \frac{e_n}{PP_p} \right\rfloor, \text{ sólo se necesita una división.}$$

A continuación se requiere conocer el valor booleano de  $K$  y  $Cal$ , consume una comparación.

```
1. if( $Cal > K$ )
2. {
3.   Activarmecanismo de protección
4. }
5. else
6. {
7.   NOactivarmecánismo, CondicionesNormales
8. } // findelif - else
```

De esto se tiene el siguiente resumen, expuesto en la tabla C.5.

<i>Sumas</i>	<i>Restas</i>	<i>Multiplicaciones</i>	<i>Divisiones</i>	<i>TOTAL</i>	<i>Comparaciones</i>
139	33	1	13	186	154

Tabla 5: Cantidad de operaciones y Comparaciones

Entonces el total de operaciones y comparaciones que se deben realizar, para llevar a cabo todos los cálculos y obtener los Diferenciales, está dado en la tabla C.6.

<i>Sumas</i>	<i>Restas</i>	<i>Multiplicaciones</i>	<i>Divisiones</i>	<i>TOTAL</i>	<i>Comparaciones</i>
259	62	46	54	421	154

Tabla 6: Total de operaciones y Comparaciones

Para conocer el tiempo que tardan estos cálculos en realizarse y ver hasta donde es posible disminuir la ventana de Predicción, con la finalidad de tener una alarma de ataque DDoS en el menor tiempo posible, se consideraría a un Microprocesador a manera de ejemplo, con las siguientes características:

Procesador Pentium I con una velocidad de reloj de 133MHZ:

- Realiza 201 MIPS.
- Procesa 63 MFLOPS.

Se tienen entonces que, existen dos tipos de operaciones: Aritmeticas y lógicas, la tabla C.7, contiene dicha clasificación:

<i>OperacinesAritmeticas</i>	<i>OperacinesLogicas</i>
Suma	OR
Resta	AND
Multiplicación	NOT
División	

Tabla 7: Tipos de Operaciones

Si consideramos perfecto al compilador empleado, es decir, que traduzca cada instrucción a una instrucción de lenguaje máquina.

Una operación aritmética de la clase Punto-Flotante se realiza en  $0,0159\mu s$ . Esto quiere decir, que 421 operaciones son realizadas en  $6,6825\mu s$ .

Por otro lado, las operaciones lógicas (mayor que, menor que, diferente a, AND, OR y NOT), sí se considera que consumen una Instrucción, entonces si una Instrucción se realiza en  $0,005\mu s$ . Por lo tanto 154 operaciones lógicas se realizan en  $0,77\mu s$ .

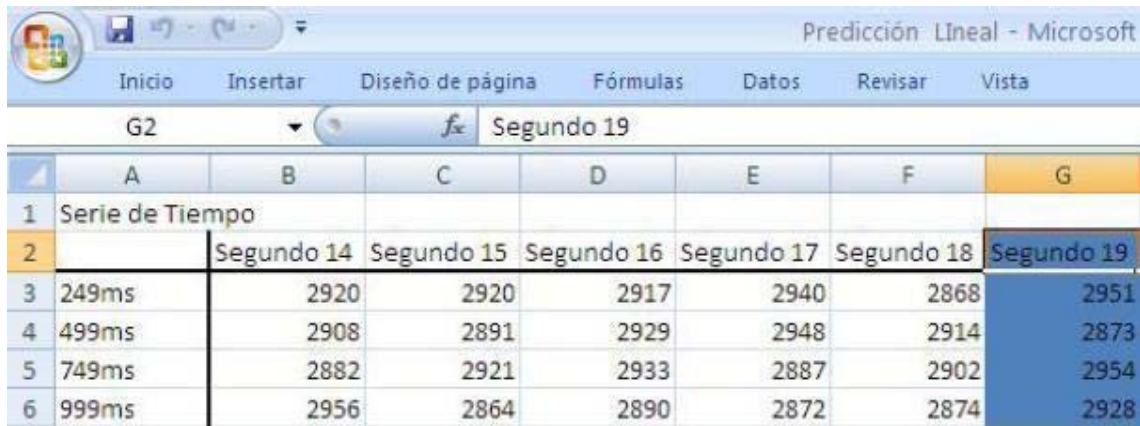
De está forma se llega al total del tiempo consumido como la suma de  $6,6825\mu s + 0,77\mu s$ , que da un total de  $7,4525\mu s$ . Ni siquiera la centésima parte de un milisegundo.

Esto nos deja ver que con un procesador como este, podríamos hacer cálculos con ventanas no sólo de 250 milisegundos, sino hasta, por ejemplo, de un milisegundo, pues el tiempo de cálculo es realmente muy pequeño.

## Apéndice D

### Uso de Excel

Al obtener los datos del monitoreo, lo primero que se hizo fue trasladarlos a la hoja de cálculo, la cual me permite hacer uso de funciones matemáticas para calcular la Predicción Lineal.\



	A	B	C	D	E	F	G
1	Serie de Tiempo						
2		Segundo 14	Segundo 15	Segundo 16	Segundo 17	Segundo 18	Segundo 19
3	249ms	2920	2920	2917	2940	2868	2951
4	499ms	2908	2891	2929	2948	2914	2873
5	749ms	2882	2921	2933	2887	2902	2954
6	999ms	2956	2864	2890	2872	2874	2928

Figura 1: Serie de Tiempo en Excel

La figura D.1, muestra la hoja de cálculo que contienen, los datos recabados en el monitoreo incluyendo los curatos del segundo a predecir (segundo 19).

Por su parte la figura D.2 ilustra la fórmula que se utilizó para obtener cada caso del promedio Móvil, es decir, para el primer valor del Promedio Móvil, se utilizaron las celdas: B9, B10 y B11, las cuales se sumaron y fueron divididas entre el orden 3, quedando la siguiente fórmula:  $(B9+B10+B11)/3$  y el resultado de esta operación fue puesto en la celda C11, para el siguiente valor del Promedio Móvil, se construye la fórmula  $(B10 + B11 + B12)/3$ , para poner el resultado en la celda C12.

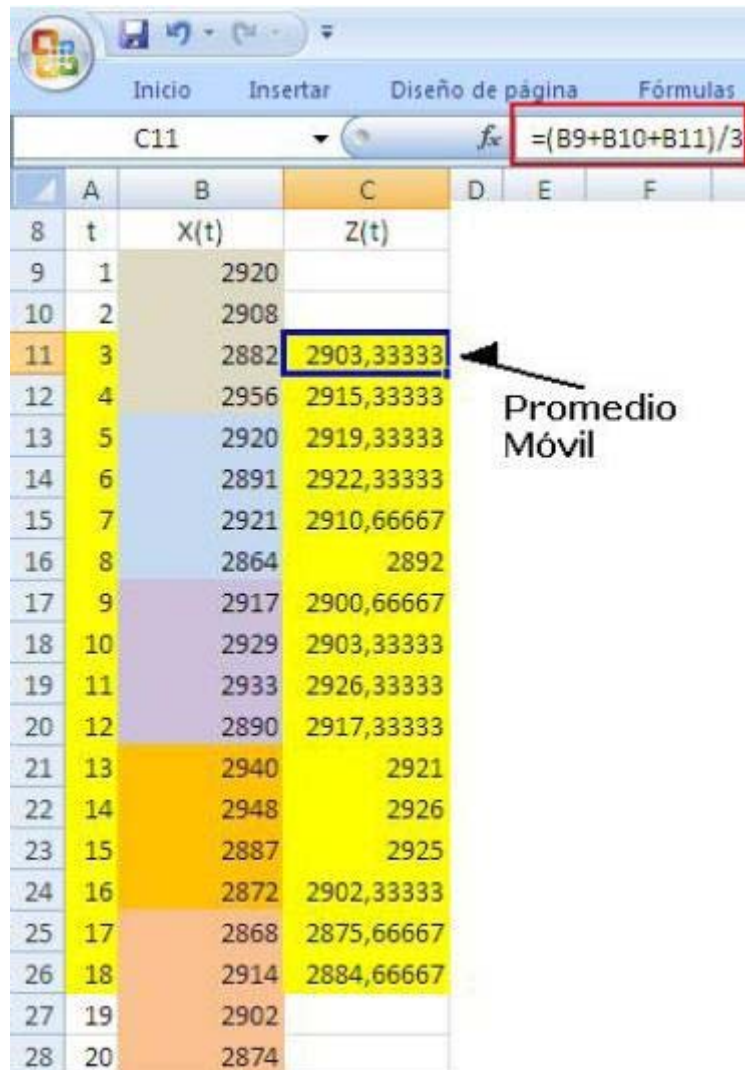


Figura 2: Suavizado de la Serie de Tiempo en Excel

La siguiente acción es calcular el Promedio del tiempo t, tomando en cuenta que NO serán incluidos para esto, todos los valores de t, es decir, sólo se deben de incluir en el calculo del Promedio a aquellos valores de t, que sirvan como abscisa para cada valor de Z(t), posteriormente se manda a llamar a la función de Excel que calcula el Promedio, como se ve en la figura D.3.

Para este caso, si observamos a la figura D.2, columna A, nos damos cuenta de que a partir de la celda A11 hasta la celda A26, se cumple con lo antes mencionado. Al computar estas celdas bajo la función del Promedio de Excel, se obtienen que éste, tienen el valor de 11. Este valor ocupa la columna D, fila 11.

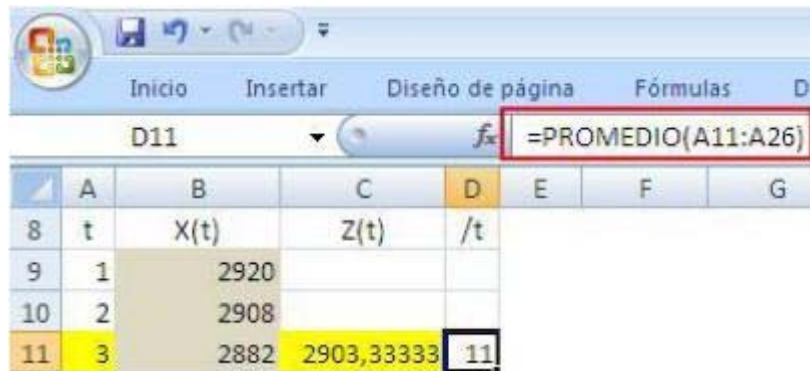


Figura 3: Calculo del Promedio en Excel

La siguiente operación a realizar es  $(t_i - \bar{t})$ , la cual se realiza con la fórmula  $A_{11} - D_{11}$ ,  $A_{12} - D_{12}$ , ...,  $A_{26} - D_{26}$ . Estos resultados fueron dispuestos en las celdas E11 a la E26 respectivamente.

Posteriormente, la operación  $(t_i - \bar{t})^2$ , se calculo haciendo el producto de  $E_{11} * E_{11}$ ,  $E_{12} * E_{12}$ , ...,  $E_{26} * E_{26}$  y disponiendo estos resultados en las celdas F11 a la F26 respectivamente.

En este momento ya se puede obtener el producto  $Z(t)(t_i - \bar{t})$ , el cual se calcula haciendo el producto de la celda  $C_{11} * E_{11}$ ,  $C_{12} * E_{12}$ , ...,  $C_{26} * E_{26}$ , disponiendo los resultados a partir de la celda G11 a la G26.

La siguiente operación es calcular el promedio de  $Z(t)$ , para ello se manda a llamar a la función de Excel que calcula el Promedio, la cual se alimentará con el rango de celdas C11 a la C26. El resultado de la llamada de está función se coloca en la celda H11.

Para calcular los valores de  $W(t)$ , se lleva a cabo la división de las celdas  $B_{11}/C_{11}$ ,  $B_{12}/C_{12}$ , ...,  $B_{26}/C_{26}$ , los resultados se pondrán en el rango de celdas de la I11 a la I26.

Posteriormente se lleva a cabo la  $\sum (t_i - \bar{t})^2$ , y la  $\sum Z(t)(t_i - \bar{t})$ , poniendo los resultados en la celda F28 y G28 respectivamente.

Los cálculos previos han sido acomodados en columnas, las cuales se muestran en la figura D.4.

	A	B	C	D	E	F	G	H	I
8	t	X(t)	Z(t)	/t	(ti - /t)	(ti - /t)^2	Z(t)(ti-/t)	/Z(t)	W(t)
9	1	2920							
10	2	2908							
11	3	2882	2903,33333	11	-7,5	56,25	-21775	2909,08333	0,99265212
12	4	2956	2915,33333	11	-6,5	42,25	-18949,67	2909,08333	1,01394923
13	5	2920	2919,33333	11	-5,5	30,25	-16056,33	2909,08333	1,00022836
14	6	2891	2922,33333	11	-4,5	20,25	-13150,5	2909,08333	0,98927797
15	7	2921	2910,66667	11	-3,5	12,25	-10187,33	2909,08333	1,00355016
16	8	2864	2892	11	-2,5	6,25	-7230	2909,08333	0,99031812
17	9	2917	2900,66667	11	-1,5	2,25	-4351	2909,08333	1,00563089
18	10	2929	2903,33333	11	-0,5	0,25	-1451,667	2909,08333	1,00884041
19	11	2933	2926,33333	11	0,5	0,25	1463,1667	2909,08333	1,00227816
20	12	2890	2917,33333	11	1,5	2,25	4376	2909,08333	0,99063071
21	13	2940	2921	11	2,5	6,25	7302,5	2909,08333	1,00650462
22	14	2948	2926	11	3,5	12,25	10241	2909,08333	1,0075188
23	15	2887	2925	11	4,5	20,25	13162,5	2909,08333	0,98700855
24	16	2872	2902,33333	11	5,5	30,25	15962,833	2909,08333	0,98954864
25	17	2868	2875,66667	11	6,5	42,25	18691,833	2909,08333	0,99733395
26	18	2914	2884,66667	11	7,5	56,25	21635	2909,08333	1,01016871
27	19	2902							
28	20	2874				340	-316,6667		

Figura 4: Calculo de las demás Operaciones en Excel

Concluidos estos cálculos, se computa  $a = G28 / F28$  y  $b = H26 - J28 * D26$ , poniendo los resultados en las celdas J28 y J29 respectivamente. Ello se muestra en la figura D.5.

	I	J	K	L
28	a =	-0,93137255		
29	b =	2918,86275		

Figura 5: Calculo de los Parámetros a y b en Excel

Lo siguiente que se recomienda es construir una tabla, con los datos de la columna  $W(t)$ , de la figura D.4, para calcular los promedios de cada una de las filas. Esta nueva tabla se muestra en la figura D.6.



	P	Q	R	S	T	U
1	Serie Residual: Estimación de la variación Estacional					
2		Segundo 15	Segundo 16	Segundo 17	Segundo 18	Segundo 19
3	200ms		1,00022836	1,00563089	1,00650462	0,99733395
4	400ms		0,98927797	1,00884041	1,0075188	1,01016871
5	600ms	0,99265212	1,00355016	1,00227816	0,98700855	
6	800ms	1,01394923	0,99031812	0,99063071	0,98954864	

Figura 6: Construcción de la Tabla Residual en Excel

Los promedios de cada fila se muestran en la figura D.7, donde para sacar el promedio, se usó la función Promedio de Excel. Los resultados del cálculo del Promedio, fueron dispuestos en un rango de celdas, que van de la celda X3 a la celda X6, respectivamente.

	W	X	Y	Z
1	Promedio /w(h), para los 249ms, 499ms, 749ms y 999ms			
2		/w(h)	/W(w(h))	
3	/w(1)	1,00242446	0,99971496	
4	/w(2)	1,00395147		
5	/w(3)	0,99637225		
6	/w(4)	0,99611168		

Figura 7: Promedios de cada Fila de la Tabla Residual

El siguiente paso es calcular el Promedio de los Promedios, para ello, se calcula el promedio de la columna X, de la figura D.7, en el rango X3 a X6, el resultado de esta operación es puesto en la celda Y 3.

	AB	AC	AD	AE	AF
1					
2		E(h)	T*(t)	Predicción	E-Predicción
3	E(1)	1,00270949	2899,30392	2907,15956	43,8404356
4	E(2)	1,00423651	2898,37255	2910,65153	-37,651532
5	E(3)	0,99665729	2897,44118	2887,75586	66,2441432
6	E(4)	0,99639671	2896,5098	2886,07285	41,9271534

Figura 8: Calculo de los Parámetros de Error de Predicción

Lo siguiente es Estimar la Estacionalidad  $E(h)$ , para ello, se involucra a las celdas  $(X3) - (Y3 - 1)$ ,  $(X4) - (Y3 - 1)$ , ...,  $(X6) - (Y3 - 1)$ . Disponiendo los resultados en la celda AC3 hasta AC6, de manera respetiva, de la figura D.8.

Para el caso de la tendencia futura  $T^*(t)$ , la fórmula empleada es  $J28*21+J29$ ,  $J28*22+J29$ ,  $J28*23+J29$  y  $J28*24+J29$ , para  $t = 21, 22, 23, 24$  y los resultados se exponen en la columna AD de la figura D.8, de la fila 3 a la fila 6.

La Predicción se calcula como el producto de  $AC3*AD3$ ,  $AC4*AD4$ ,  $AC5*AD5$  y  $AC6*AD6$ . Los resultados son dispuestos en la columna AE en el rango AE3-AE6, de la figura D.8.

La última operación es calcular el Error de Predicción, el cual se calcula con las formulas:  $G3-AE3$ ,  $G4-AE4$ ,  $G5-AE5$  y  $G6-AE6$ . Los resultados son dispuestos en la columna AF en el rango AF3-AF6, de la figura D.8.

Con lo anterior, quedan explicados los cálculos realizados con el uso de Excel, para llevar a cabo el cómputo del Error de Predicción.

## Glosario

### A

**Agente:** Un agente es un módulo de software de administración de red que reside en un dispositivo administrado (zombi). Este agente posee un conocimiento local de información del sistema infectado. Entre sus características el agente también posee la capacidad de comunicarse con la entidad administradora (Maestros) de red.

**Amazon:** Es una compañía estadounidense de comercio electrónico con sede en Seattle, Washigton. Fue una de las primeras grandes compañías en vender libros a través de Internet.

**Apache:** Este servidor web es actualmente el más implantado entre los distintos servidores que ofertan servicios web en Internet. Además Apache, servidor originalmente pensado para el entorno Linux, dispone de versión para el entorno Windows.

**Atacante:** Es alguien que viola la seguridad de un sistema informático para realizar la intrusión con fines de beneficio personal o para hacer daño. Se considera que la actividad realizada por esta clase de entes o individuos es dañina e ilegal.

### B

**backlog:** Número máximo de conexiones que el kernel de TCP encolará para un socket.

**BGP:** Border Gateway Protocol es un protocolo mediante el cual se intercambia información de encaminamiento entre Sistemas Autónomos. Por ejemplo, los ISP registrados en Internet.

**Buy.com:** Una enorme empresa que dedica a la venta de múltiples productos a través de Internet.

**Botnet:** Una red de este tipo está compuesta de computadoras que han sido violadas y contaminadas con programas que pueden ser instruidos para lanzar ataques desde una computadora de control central.

Buffer: Es una ubicación de la memoria en una computadora reservada para el almacenamiento temporal de información digital, la cual está esperando ser procesada.

## C

Cifrado: Es el arte o ciencia de cifrar y descifrar información utilizando técnicas que hagan posible el intercambio de mensajes de manera segura que sólo puedan ser leídos por las personas a quienes van dirigidos.

Cisco: Es una empresa multinacional ubicada en San José California Estados Unidos, dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones.

CNN: Cable News Network (Cadena de Noticias por Cable), es una cadena de televisión estadounidense.

## D

DDoS: Distributed Denial of Service Attack.

Dirección IP: Es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP.

DNS: Domain Name System, es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet.

DoS: Denegation of Service.

## E

eBay: Es un sitio destinado a la subasta de productos a través de Internet. Es uno de los pioneros en este tipo de transacciones, puesto que su presencia en la comunidad en línea es de varios años.

e-mail: Es un servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente mediante sistemas de comunicación electrónicos.

H

handshake: Es el mecanismo mediante el cual un cliente establece una conexión con un servidor usando el protocolo TCP.

I

ICMP: Internet Control Message Protocol, es el subprotocolo de control y notificación de errores del Protocolo de Internet IP.

Internet: Es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

IP: Internet Protocol, es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

IP Spoofing: Suplantación de IP, consiste básicamente en sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar.

ISP: Proveedor de servicios de Internet.

IRC: Internet Relay Chat es una red de comunicación en tiempo real en la que puedes hablar con un grupo de usuarios al mismo tiempo.

M

Maestros: Herramienta de software corriendo en un equipo, que le sirve al atacante para administrar el ataque DDoS, además de brindarle un escondite y le permite NO estar en contacto directo con las maquinas zombi, las cuales son las que están llevando a cabo el ataque.

MaxClients: Es el número máximo de conexiones del servidor Apache, si se alcanza este número, Apache no acepta nuevas peticiones, por lo que los nuevos clientes, no serán aceptados.

Malware: Es un software que tiene como objetivo infiltrarse en o dañar un equipo sin el conocimiento de su dueño y con finalidades muy diversas ya que en esta categoría encontramos desde un troyano hasta un spyware.

Mbps: Mega bits por segundo

Metástasis: Es un término médico que se refiere a la propagación del cáncer desde su origen hacia otros lugares del cuerpo. Este término fue sugerido para la seguridad informática por la forma de diseminarse del malware.

N

netstat: Es una herramienta de línea de comandos que muestra un listado de las conexiones activas de un equipo de cómputo, tanto entrantes como salientes. Existen versiones de este comando en varios sistemas, como Unix/Linux, Mac OS X, Windows y BeOS..

O

Open sesame: Es un administrador de nombres de usuario y contraseñas, donde se puede mantener todas las cuentas de usuario y contraseñas guardadas en una sola ubicación, así sólo se necesitará recordar una contraseña de entrada. El programa tiene un sistema de ventana flotante que permite introducir la información. Los datos guardados son protegidos con un sistema de cifrado.

Outlier: Se refiere a conductas no esperadas de la Serie de Tiempo.

Overflow: Es un error de software que se produce cuando se copia una cantidad de datos sobre un área que no es lo suficientemente grande para contenerlos, sobrescribiendo de esta manera otras zonas de memoria.

## P

**Password:** Es una forma de autenticación que utiliza información secreta para controlar el acceso hacia algún recurso.

**Ping:** Packet Internet Grouper, se trata de una utilidad que comprueba el estado de la conexión con uno o varios equipos remotos por medio de los paquetes de solicitud de eco y de respuesta de eco, definidos en el protocolo de red ICMP, para determinar si un sistema IP específico es accesible en una red. Es útil para diagnosticar los errores en redes o enrutadores IP, pero también es usado para ataques DDoS.

**Predicción Lineal:** Predicción de valores futuros de un Serie de Tiempo, tratando de obtener en lo posible, límites de confianza.

**Proxies:** Hace referencia a un programa o dispositivo que realiza una acción en representación de otro. La finalidad más habitual es la de servidor proxy, que sirve para permitir el acceso a Internet a todos los equipos de una organización cuando sólo se puede disponer de un único equipo conectado, esto es, una única dirección IP.

## R

**Red:** Es un conjunto de equipos (computadoras y/o dispositivos) conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, que comparten información, recursos, servicios, etc.

**Reading:** Hace referencia al número de conexiones que se encuentra atendiendo el servidor Apache.

**RIP:** Routing Information Protocol, es un protocolo de puerta de enlace interna, utilizado por los routers.

**Router:** Es un dispositivo de hardware para la interconexión de redes de computadoras que opera en la capa de red. Este dispositivo permite asegurar el encaminamiento de paquetes entre las redes de datos.

## S

**Sniffing:** Es la acción de usar un programa que captura las tramas de red. Generalmente se práctica para administrar la red, aunque también puede ser utilizado con fines maliciosos.

**Socks:** Es un protocolo de Internet que permite a las aplicaciones Cliente-servidor usar de manera transparente los servicios de un firewall de red. SOCKS es una abreviación de SOCKeTS.

**Software:** Se refiere al equipamiento lógico de una computadora digital, tal componente lógicos incluyen, entre otras, aplicaciones informáticas tales como procesador de textos, que permite al usuario realizar todas las tareas concernientes a edición de textos, también provee una interface ante el usuario.

**Spyware:** Los programas espías son aplicaciones que recopilan información sobre una persona u organización sin su conocimiento. La función más común que tienen estos programas es la de recopilar información sobre el usuario y distribuirlo a empresas publicitarias u otras organizaciones interesadas.

**Squids:** Es un programa implementa un servidor proxy y un demonio para caché de páginas web. Tiene una amplia variedad de utilidades, desde acelerar un Servidor Web, guardando en caché peticiones repetidas a DNS y otras búsquedas para un grupo de gente que comparte recursos de la red, hasta caché de web, además de añadir seguridad filtrando el tráfico.

**SYN:** SYN es un bit de control dentro del segmento TCP, que se utiliza para sincronizar los números de secuencia iniciales de una conexión en el procedimiento de establecimiento de handshake.

## T

**TCL:** (Tool Command Language) Es un lenguaje de script muy potente. En este trabajo se utilizó para programar simulaciones de redes.



TCP: Transmission Control Protocol, Sirve para crear conexiones entre equipos a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.

Timeout: Es un contador que una vez expirado hace que sucedan eventos, como por ejemplo, la desconexión a un servidor.

troyano: Se denomina troyano (caballo de Troya) a un programa malicioso capaz de alojarse en computadoras y permitir el acceso a usuarios externos, a través de una red local o de Internet, con el fin de recabar información o controlar remotamente a la máquina anfitriona.

V

Víctima: Desde el punto de vista informático, una víctima es el sistema que sufre un daño o perjuicio, que es provocado por una acción, ya sea por la infección de uno o muchos malwares o individuos.

W

Web: World Wide Web, es un sistema de documentos de hipertexto e hipermedios enlazados y accesibles a través de Internet.

Win: Window-TCP, Hace referencia al número de segmentos que la ventana de transmisión de TCP transmite.

Wingates: es un potente servidor Proxy/Firewall que permite compartir una sola conexión a Internet entre todos los equipos de su red.

Worms: Es un virus informático que tiene la propiedad de duplicarse a sí mismo. Los gusanos utilizan las partes automáticas de un sistema operativo que generalmente son invisibles al usuario. A diferencia de un virus, un gusano o worm no precisa alterar los archivos de programas, sino que reside en la memoria para duplicarse a sí mismo. Los gusanos siempre dañan la red, aunque sea simplemente consumiendo ancho de banda.

## Y

Yahoo: Es una empresa global de medios con sede en Estados Unidos, cuya misión es ser el servicio global de Internet más esencial para consumidores y negocios.

## Z

Zombi: Es un equipo o sistema infectado con algún malware que ejecuta órdenes de un atacante.

## **Bibliografía**

[1] Internet Denial of Service: Attack and Defense Mechanisms, Jelena Mirckovih, Sven Dietrich, David Dittrich and Peter Reiher, Prentice Hall 2004.

[2] Internet Denial of Service: Attack and Defense Mechanisms, by Jelena Mirkovic, Sven Dietrich, David Dittrich and Peter Reiher, Prentice Hall PTR.

[3] Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks, 2007 - 2008 Cisco Systems, Inc. All rights reserved

[4] Threat Classification, Copyright 2004, Web Application Security Consortium. All rights reserved

[5] Tutorial for the network simulation NS - 2, by Marc Greis, <http://www.isi.edu/nsnam/ns/>

[6] Introduction to Network Simulator NS2, Issariyakul, Teerawat, Hossain, Ekram, Hardcover, 2008.

[7] Econometric Models and Econometric Forecasts, by R. Pindick and D. Rubinfeld, MacGraw Hill, 1998.

[8] The NS 2 Manual (formerly ns Notes and Documentation), by Kevin Fall, Editor Kannan, 2007.

### **Bibliografía Complementaría:**

[9] TRANSMISSION CONTROL PROTOCOL, DARPA INTERNET PROGRAM, Defense Advanced Research Projects Agency Information Processing Techniques Office 1400 Wilson Boulevard Arlington, Virginia 22209, by Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California 90291

[10] RFC768 - User Datagram Protocol RFC 768, by J. Postel ISI 28 August 1980

[11] INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION, September 1981, prepared for Defense Advanced Research Projects Agency Information Processing Techniques Office 1400 Wilson Boulevard Arlington, Virginia 22209, by Information Sciences Institute University

114 of Southern California 4676 Admiralty Way Marina del Rey, California 90291 September 1981

[12] Living with Systems in Production Avoiding Heartbreak in Long Term Relationships, by Michael T. Nygard ATI, Inc.

[13] Xgraph User's Manual

[14] Chao, Lincoln L. Estadística para ciencias sociales y administrativas. McGraw-Hill 1975.