UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

Caracterización de un Firewall Perimetral implementado mediante software libre OpenBSD y CentOS

TESIS

PARA OPTAR POR EL TÍTULO DE

MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

P R E S E N T A

FRANCISCO RUIZ SALA

DIRECTOR DE TESIS

DR. JOSÉ JAIME CAMACHO ESCOTO

INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAA APLICADAS EN SISTEMAS

CIUDAD UNIVERSITARIA, CIUDAD DE MÉXICO JUNIO DE 2024

Dedicatoria

Barbara Pichardo:

Gracias por creer en mí, como profesionista, y creer que era capaz de seguir adelante en mi preparación académica, y a pesar de que no estás con nosotros, este trabajo no sería posible si no me hubieras alentado a seguir estudiando, a pesar del tiempo, edad, pandemia y todas las adversidades.

“A veces la vida nos pone de rodillas, el secreto es tener la fuerza para seguir levantándonos y seguir adelante, y si nosotros no podemos, dejar un legado que se levante y siga adelante.” <<Francisco Ruiz>>

“Qué triste fueron esos años, tener el deseo y la necesidad de vivir, pero no tener la habilidad.” <<Charles Bukowski>>

Agradecimientos:

DR. JAVIER GÓMEZ, como tutor inicial y por confiar en mí para este proyecto y el gran apoyo en el posgrado.

Dr. Antonio Peimbert, Dr. Alan Watson, Dr. Jesús González, Instituto de Astronomía UNAM, por el apoyo, tiempo y manejo de la infraestructura de red, todo esto en el Instituto de Astronomía de la Universidad Nacional Autónoma de México.

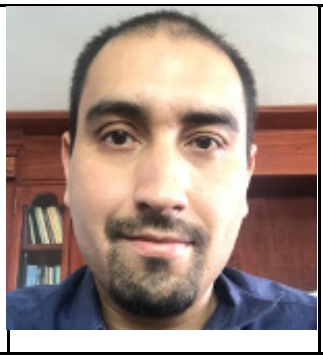
Dr. Diego López-Cámara, que gracias a él avancé para poder seguir con este trabajo.

Julieta Fierro, siempre fuiste quien me dio ánimos y una razón por la cual las cosas se deben comenzar, hacer, y terminar, no importa las circunstancias en la cual estés.

Ficha de los autores:

AUTOR DE TESIS

L.I Francisco Ruiz Sala
Licenciado en Informática
Especialista en Cómputo de alto Rendimiento
fruiz@astro.unam.mx



DIRECTOR DE TESIS

Dr. José Jaime Camacho Escoto
Professor member of Cátedras CONACyT's program
Facultad de Ingeniería
Universidad Nacional Autónoma de México
jcamachoe@unam.mx

Resumen

Hoy en día las redes y el acceso a internet son indispensables para el trabajo cotidiano, por lo cual se requiere tener una buena seguridad perimetral para proteger a los equipos que se encuentran en la red[1] LAN2[33] (Local Area Network). Este sistema de seguridad permite garantizar la confidencialidad de la información que pasa a través de ella. Además, protege a los equipos y otros recursos que conforman a la red. Uno de los mecanismos comúnmente utilizados para llevar a cabo esta tarea son los firewall[3] perimetrales. En la práctica, estos equipos a nivel comercial son muy costosos y no todas las instituciones pueden adquirirlos. El poder tener una solución de bajo costo tanto de hardware como de software representa una excelente opción si no se cuentan con recursos económicos suficientes. Una de las grandes desventajas de utilizar software libre es que muchas veces las herramientas suelen tener errores y/o no estar optimizadas por completo. Esto ocasiona que las prestaciones de muchas herramientas de software libre no ofrezcan el mismo rendimiento que las herramientas propietarias y de pago y que este rendimiento no siempre se encuentre correctamente caracterizado.

En este trabajo, se caracteriza el rendimiento de dos firewalls perimetrales de software libre activos e instalados en el Instituto de Astronomía (IA), uno con topología transparente[4] y el segundo con topología NAT[5]. Ambos firewalls corren en OpenBSD[6] 7.2 y utilizan Packet Filter[7]. Ambos firewalls operan en una red ethernet UTP de 10Gb/s. Para medir el rendimiento de los dispositivos se realizaron mediciones de throughuput[8], latencia[9] y jitter[10]. Adicionalmente y debido a los resultados encontrados, se realizó una propuesta con CentOS 7.2 que utiliza como firewall iptables[11]. Como los equipos reales se encuentran en producción, se diseñó una maqueta de pruebas que igualara las mismas condiciones de topología y hardware. La maqueta de pruebas consta de cuatro equipos: el cliente, el switch[12] de comunicación, el firewall y el equipo destino. Se utilizaron las herramientas iperf3 para medir el throughput, y ping para medir la latencia. A partir de varias repeticiones de las pruebas de ping, se calculó el jitter.

---

[1] ***Seguridad Perimetral de red de datos*:** Es una topología de red de datos la cual nadie fuera de ella puede acceder a los datos y recursos dentro de la misma, únicamente pueden acceder los miembros**.**

[2] ***LAN***: *Local Area Network (Red de Área Local)* Son definidas por una organización, las cuales comparten información entre sus usuarios, con direccionamiento y/o infraestructura aislada lógica o físicamente.

[3] ***Firewall (Pared de Fuego o contrafiuego)***: Equipos diseñado mediante hardware y/o software que se encarga mediante reglas específicas definidas por el usuario para realizar filtrado del tráfico de la red, este puede estar instalado en el mismo equipo o puede ser perimetral para proteger una red *LAN (Local Area Network)*

[4] ***Firewall transparente o puente (bridge):*** Es un equipo instalado entre 2 redes, que opera en la capa de Capa 2 del modelo OSI(Open Systems Interconnection). El firewall transparente no solo reenvía los paquetes entre 2 redes como lo haría un enrutador (router), si no además es capas de filtrar el tráfico de las redes interconectadas entre si, esto de acuerdo con las reglas definidas por el usuario. La ventaja de este firewall es que no altera la topología ni el direccionamiento de internet

[5] ***Firewall NAT (Networ Adress Translation):*** Es un equipo instalado entre 2 redes, permite el tráfico de las 2 redes para que pase a través de la puerta de enlace si un dispositivo de la red que lo solicite. Además, filtra los paquetes que no están en el filtro del firewall y descarta todas las solicitudes de direcciones que no corresponde a las redes especificadas.

[6] ***OpenBSD:*** Software de sistema operativo de código abierto de la familia de Unix, propiedad de Berkeley Software Distribution (BSD).

[7] ***Packet Filter (pf):*** Software libre que esta predeterminado en el sistema operativo OpenBSD (Berkeley Software Distribution), propiedad de la Universidad de California en Berkley, USA. Este software se encarga de realizar filtrado de tráfico de red.

[8] **Throughput**: Velocidad de transferencia de información también como la velocidad real de transporte de datos a través de una red de datos. La unidad de medida es de bits por segundo, (b/s), y los múltiplos Megabit por segundo (Mb/s), Gigabit por segundo (Gb/s). [21]

[9] ***Latencia (retardo)***: Es el tiempo de respuesta que tarda un equipo en red en responder una petición especifica, en TCP/IP el protocolo ICMP (*Internet Control Message Protocol)* nos da un indicador en tiempo (generalmente milisegundos) el tiempo que tarda en recibirse el mensaje de un equipo a otro

[10] ***Jitter***: Es la fluctuación o variación de la de la comunicación de transmisión de datos, esto es muy utilizado para verificar la calidad de servicios sincrónico o P2P donde generalmente se usa para voz o video conocidos como streaming, por tanto, determina la calidad de la comunicación de estos servicios y puede medirse de esta manera. Por tanto, se puede conocer la medida de la fluctuación de la comunicación.

11 **Iptables:** Programa que funciona como firewall disponible en todas las distribuciones del núcleo o kernel del Sistema Operativo Linux. Iptables es implementado mediante diferentes niveles de "netfilter". Como todo filtro de seguridad tipo firewall las reglas se almacenan en memoria.

[12] ***Switch (conmutador de paquetes):***Se dice de un equipo de comunicación que permite comunicar diferentes dispositivos de red a un red de datos, es el equipo que conecta las redes de internet por medio del protocolo TCP/IP, y funciona desde la capa 1 a la 5 del modelo OSI.

En las pruebas realizadas, se encontró que el firewall que usaba CentOS disminuyó la velocidad de transferencia del canal de aproximadamente 75%. Además, se encontró que no era posible cargar una cantidad grande de reglas a Packet Filter bajo el sistema operativo en cuestión. Como consecuencia, se propuso el uso de CentOS, ya que no tiene las limitaciones encontradas en OpenBSD y permite utilizar una velocidad de transferencia más cercana a la máxima permitida para las tarjetas de red utilizadas (10Gb/s).

Se conoce por otros trabajos como el reportado en [1] que uno de los elementos que afecta a la velocidad de transferencia percibida por los usuarios es el número de reglas de filtrado que se apliquen en el firewall perimetral. Para caracterizar el comportamiento de la red con diferentes cantidades de reglas, fue necesario crear reglas de forma aleatoria (para evitar alguna optimización que pudiera generar el software). Se realizaron mediciones variando el número de reglas desde 100 hasta 5000. Estas pruebas se utilizaron para las topologías Bridge y NAT, y las pruebas se realizaron para los sistemas operativos CentOS y OpenBSD. Los resultados mostraron que no había cambios significativos para tal cantidad de reglas. Por ello, se realizó una prueba adicional, en la cual se aumentaron las reglas hasta llegar a 1,000,000 para CentOS con iptables obteniendo al final una disminución de aproximadamente 45% en la velocidad de transferencia.

Introducción

Antecedentes

En el Instituto de Astronomía de la Universidad Nacional Autónoma de México (IA-UNAM) se requería la instalación de un firewall perimetral (firewall instalado en la red perimetral LAN [33]), debido a las crecientes intrusiones no desaseadas desde internet, lo que representaba un gran problema para los equipos y servidores que se alojan en dicha red LAN del IA. La red del IA-UNAM ya estaba instalada y configurada, por lo cual se requería poner un firewall, sin afectar la configuración de red ya establecida.

En este sentido, se planteó instalar un firewall perimetral tipo bridge usando el Sistema Operativo OpenBSD, que solucionaba el problema de accesos no deseados, y además la configuración de la red perimetral permanecería intacta.

En un principio se consideró instalar un firewall transparente entre la red perimetral del IA-UNAM y la salida a internet. Posteriormente, y con el crecimiento de los equipos que requerían conectarse a internet, se agregó un firewall adicional instalado también con OpenBSD. El firewall transparente funcionó desde 2006 y hasta el 2014. Por otro lado, el firewall NAT funcionó desde 2007 y sigue en operación. En el año 2023, este firewall representa parte fundamental de la red de datos del IA-UNAM porque realiza otras tareas como la de NAT (Network Address Translation) tanto de entrada como de salida, y da servicio de DHCP (Dynamic Host Configuration Protocol), para poder configurar automáticamente las direcciones ip de manera privada en la red LAN (teléfonos, portátiles equipos móviles, computadoras nuevas etc.).

Otra tarea que realiza este firewall es el port forwarding[13] [49] (reenvió de puertos), el cual permite identificar todas y cada una de las direcciones IP privadas identificadas con su dirección física MACAdress (Media Access Control Adress) [14] [50], para el acceso externo a la red privada.

[13] ***Port forwarding (Renvio de puertos):*** Es la forma de redirigir conexiones publicas a través de tu router (ruteador) para acceder a servicios y equipos en direcciones ip privadas. Para el caso de acceder a equipos y servicios en redes privadas, se identifican en el router por medio de un puerto TCP/IP, al llegar la petición a ese puerto, el router se encarga de redirigir la petición al servicio que tenga asignado ese puerto e ip privada.

[14] ***MAC Adress:*** La dirección MAC o (dirección física) viene del acrónimo en ingles ***M****edia* ***A****ccess* ***C****ontrol y* se define en el estándar IEEE 802, es un identificador de 48 bits que es único para cada dispositivo de red y funciona en la capa 2 del modelo OSI, este identificador es para establecer el enlace de red.

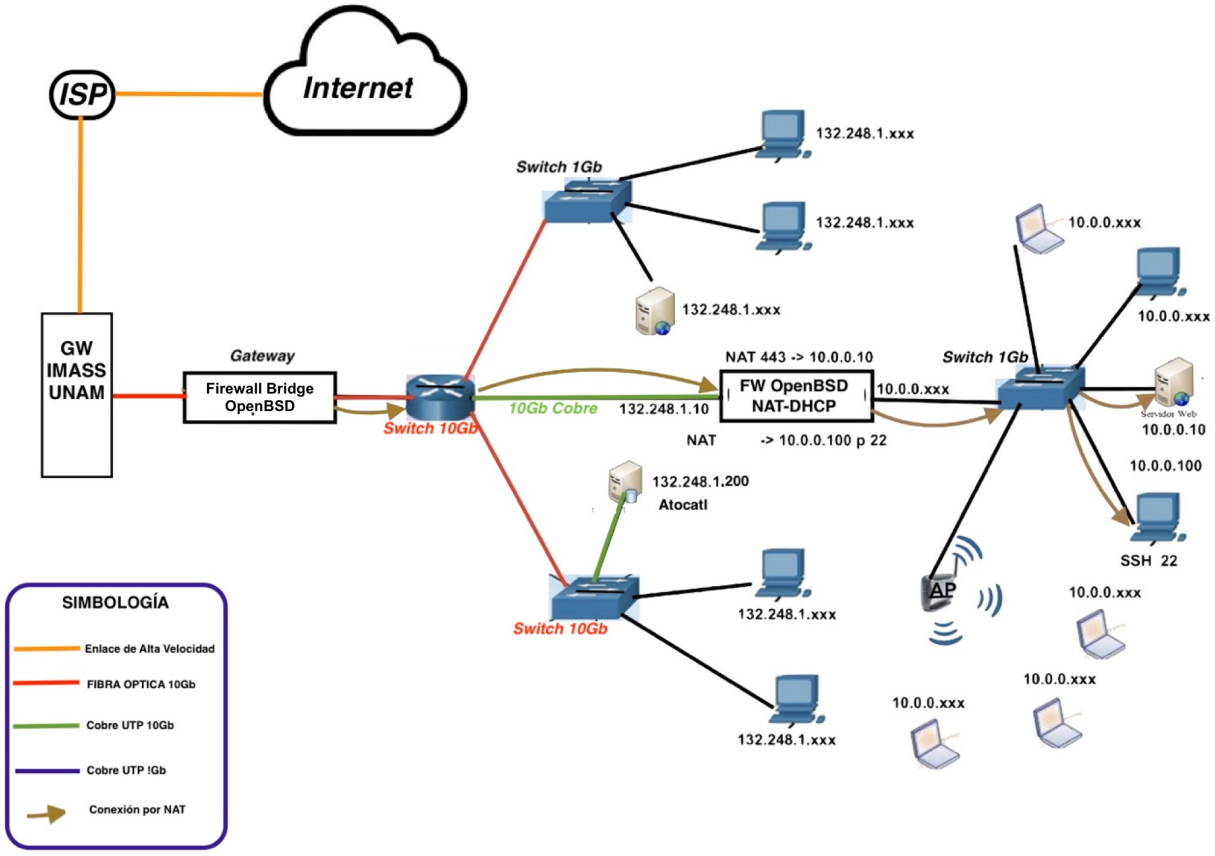


Figura 1. Topología de la red de datos del 2007 con los Firewalls Fortinet® y OpenBSD

Como se muestra en la figura 1, el firewall bridge se encuentra en el equipo de comunicaciones con acceso a la red principal para el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas de la Universidad Nacional Autónoma de México (IIMAS-UNAM) e ISP (Internet Service Provider).

Definición del Problema

El presente trabajo pretende conocer como es afectado el trouhgtput[21] en una la red LAN del IA-UNAM con un ancho de banda[15] de 10 Gb/s en un canal UTP (*Unshielded Twisted Pair*)[47] y con un firewall perimetral instalado con software libre con un Sistema Opertativo OpenBSD versión 7.2 y PF como software de filtrado tanto en topologías NAT como bridge, del mismo modo compararlo con un firewall perimetral topología NAT con Sistema Operativo CentOS versión 7.2. e iptables como software de filtrado.

Se observa en otros trabajos que hablan del mismo tema [1] que uno de los parámetros importantes es el número de reglas de filtrado. Por ello, se utilizarán dichos parámetros como referencia durante desarrollo de esta tesis.

[15] ***Ancho de banda:*** Es la cantidad de datos que se pueden transmitir en un período de tiempo fijo, este ancho de banda depende del medio físico, el cual puede ser de cobre, óptico, o inalámbrico del canal de comunicación,

Objetivo general

Caracterizar el throughput, latencia y jitter ofrecidos a los usuarios con el uso de un firewall instalado en la red perimetral del Instituto de Astronomía UNAM, implementado en OpenBSD 7.2 y comparar estas prestaciones con las que podría ofrecer un firewall bajo CentOS 7.2.

Objetivos específicos

- Medir el throughput y latencia del canal de comunicación en la red LAN UTP 10Gb/s del IA-UNAM.
- Medir el throughput y latencia para un firewall perimetral en la red LAN del IA-UNAM instalado en OpenBSD versión 7.2 con topología NAT.
- Medir el throughput y latencia para un firewall perimetral instalado en la red LAN del IA-UNAM OpenBSD con topología transparente.
- Medir el throughput y latencia para un firewall perimetral instalado en la red LAN del IA-UNAM CentOS versión 7.2 instalado en una red LAN con topología NAT.

Metas

Conocer y especificar la velocidad de throughput y latencia en un firewall OpenBSD en forma de bridge y NAT para cuantificar cuanto afecta el throughput de la red LAN del IA-UNAM.
Realizar una comparación de throughput con sistemas operativos OpenBSD y CentOS como firewall perimetral.

Justificación

La compra de equipos comerciales especializados como firewalls perimetrales es muy onerosa, y su tiempo útil de vida va de 1 a 5 años, sin tener la necesidad de actualizarlo, el poder tener una opción libre y de bajo costo, representa una ventaja para las organizaciones que no cuentan con recursos económicos para adquirir las soluciones comerciales. Una solución libre como OpenBSD o CentOS minimiza el gasto y depende de la habilidad y conocimiento del personal especializado que, con el tiempo, es mejor inversión que el outsourcing, ya que el comprar una solución comercial implica un gasto aproximado de más de 20,000 dólares americanos o lo equivalente a 400,000 pesos mexicanos (año 2023), para poder adquirir un producto profesional y configurable, además se deberá realizar en un par años una actualización con costos equivalentes, ya que la tecnología se vuelve obsoleta, y los promovedores dejan de ofrecer mantenimiento y actualizaciones. La importancia de conocer el throughput en un equipo instalado con OpenBSD como firewall perimetral permite conocer el aprovechamiento de la velocidad de transferencia de una red de datos. En casos en que se ha invertido tecnológicamente para que esta red sea de alta velocidad, hay que asegurar que se aprovecha totalmente. En este escenario, es necesario saber que tanto se desaprovecha el throughput con un sistema de seguridad perimetral firewall instalado con un equipo genérico con OpenBSD o CentOS. El poder realizar una instalación de una computadora como firewall perimetral nos permite poder hacerlo en cualquier organización y red, sin embargo, el poder conocer qué características de throughput, representa una información muy importante para poder utilizar estas soluciones en vez de las comerciales.
En el Instituto de Astronomía, existe el Laboratorio de Modelos y Datos Científicos (LAMOD) [54]. En este laboratorio hay equipos de alto rendimiento del Instituto de Astronomía (Atocatl), Instituto de Química, Instituto de Ciencias Nucleares, los cuales requieren protección perimetral. Además, estos equipos requieren transferir grandes volúmenes de información en un canal de 10Gb/s. Por

ello, es crítico tener herramientas que no afecten sustancialmente la velocidad de transferencia del canal y brinden los servicios de protección perimetral requeridos.
Este trabajo permitirá conocer a detalle las prestaciones ofrecidas en el IA-UNAM al tener un equipo con OpenBSD 7.2 como firewall, y también se propondrá el uso de otro firewall instalado bajo CentOS 7.2.

Metodología

La metodología es el procedimiento que utilizaremos para poder determinar el throughput y la latencia, teniendo en cuenta que la red, las tarjetas de red, los equipos de comunicación tienen variaciones con respecto al tiempo, por lo cual se tiene que establecer un procedimiento que permita acotar esas variables intrínsecas de los sistemas de comunicación y de esta manera poder asegurar que el ancho de banda que estamos midiendo está dentro de intervalos de confiabilidad estableciendo valores promedio y así poder determinar la velocidad de transferencia real.
El proceso que se sigue para realizar las mediciones, así como las herramientas utilizadas, se describen en la figura 14. En este diagrama de flujo se explica el procedimiento que se utilizó para los dos sistemas operativos, usados como firewall, es decir, OpenBSD y CentOS.

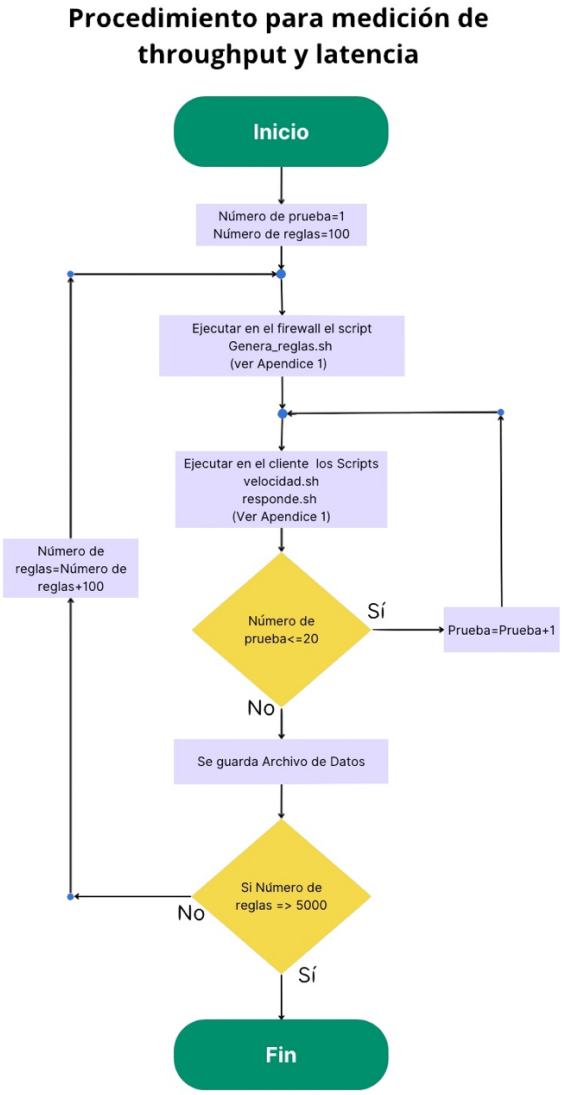


Figura 14. Procedimiento para realización de la medición de throughput y latencia.

El script utilizado para generar las reglas se puede encontrar en el apéndice 1. Se tiene el script Genera_reglas.sh, el cual genera reglas específicas para packet filter de OpenBSD, así como para CentOS, esto porque la sintaxis es diferente para cada Sistema Operativo y sistema de manejo de reglas.
El diagrama de flujo de la figura 13 describe los siguientes pasos:

1. Se generan las reglas ejecutando el script en shell bash "Genera_reglas.sh", y se generan las reglas especificadas para cada conjunto de 100 pruebas.
2. Se carga ese archivo en el firewall en OpenBSD es el archivo (/etyc/pf.conf) y para el caso de CentOS es en (/etc/sysconfig/iptables). Esto se realiza por cada 100 reglas y se repite hasta llegar a 5000 reglas.
3. Una vez cargadas las reglas, se ejecuta en el cliente el script en bash "velocidad.sh" y "responde.sh" que son los que entregarán el throughput y la latencia, respectivamente, dichos archivos ejecutarán 20 en 20 medidas cada uno.
4. Se guarda la información obtenida en archivos de datos, uno para el throughput y otro para la latencia.
5. Se incrementa la cantidad de reglas en 100 y se repite el procedimiento desde el paso 2 hasta llegar a las 5000 reglas.

Las mediciones del throughput en el script en bash utiliza el programa iperf, y las mediciones de latencia están en el script "responde.sh" utilizan el programa ping. Los resultados obtenidos de las veinte ejecuciones de ambas pruebas se utilizarán para el análisis los promedios obtenidos. Es importante mencionar que después de la prueba del canal, la cual se realizará sin ningún firewall intermedio, se probarán 2 tipos de firewalls en este canal de comunicación. El primer firewall instalado sobre OpenBSD con Packet Filter, y el segundo con CentOS con Iptables. Para estas pruebas se utilizarán las aplicaciones date, iperf, ping y traceroute, que forman parte de un script que permite conocer la fecha y hora de inicio y término de las pruebas, así como comprobar la ruta del canal de comunicación. Estos scripts en Shell se detallan en el apéndice 2.
El hardware utilizado se describe en la tabla 5 y las figuras 10 y 11, así como las imágenes 1 y 2, que muestran las características de los equipos, e imágenes de los equipos y cómo están conectados.
Referente al cableado de conexión que utilizaremos, este es considerado de alta velocidad en redes locales LAN, y está descrito por el estándar IEEE 802.3an del año 2006 [28], corresponde a un cableado tipo Ethernet (10000-Base-T)de tipo par trenzado UTP (Unshielded Twisted Pair), clasificado por proveedores de cableado en la categoría 6a por su forma de armado y distancia máxima, tiene un ancho de banda de 500 MHz, y soporta una velocidad máxima de transferencia de datos de 10,000 (Diez mil) bits por segundo (10 Gb/s), a una distancia máxima de 100 metros.
En la tabla 5 se hace la descripción física y de software de los equipos de los cuales se harán las pruebas:

Uso/ Nombre	*Procesador*	*Memoria RAM y Disco duro*	*Tarjeta de RED*	*Memoria Caché*	*Sistema Operativo*	*Tipo de Firewall*
Equipo 1/ Cefalópodo	HP AMD Opteron Model 285 2.6GHz	8GB SSD 150G	Intel® Ethernet Converged Network Ethernet Controller X540-AT2 PCIe v2.1 (5.0GT/s)	1MB dual core	CentOS 7, 64 bits	Sin firewall
Equipo 2/ Atocatl	Intel(R) Xeon(R) CPU X5650	50 GB SATA 1 TB	Intel® Ethernet Converged Network Ethernet Controller X540-AT2 PCIe v2.1 (5.0GT/s)	12288 KB	CentOS 7, 64 bits	Cliente Iptables
Firewall	HP AMD Opteron Model 285 2.6GHz	8GB SSD 150G	IIntel® Ethernet Converged Network Ethernet Controller X540-AT2 PCIe v2.1 (5.0GT/s)	1MB dual core	CentOS 7, 64 bits / OpenBSD 7.3 AMD64	Firewall IpTables, PacketFilter

Tabla 5. Descripción del hardware utilizado para la topología de red.

Cabe mencionar que, aunque la red corresponde a tecnología actualizada del año 2022 y es de una velocidad de 10Gb/s, ethernet en cobre, el hardware de las computadoras no lo es, corresponde tecnología del 2007. Sin embargo, esta última característica no es determinante para la velocidad de transferencia de la información, debido a que las interfases de red "tarjetas de red" son de generación 2022 a una velocidad de 10Gb/s, y funcionan correctamente en estos equipos. En la figura 13 mostramos el diagrama de flujo correspondiente a la metodología de las mediciones de throughput, así como el procedimiento para la medición de la latencia.

Figura 14. Vista Frontal de los 2 equipos "Calamar", y el firewall.


Figura 15. Vista trasera de los equipos firewall y cliente conectados en las tarjetas de red 10Gb Ethernet.

En la figura 14 se ve el equipo de prueba que llamamos calamar y abajo se puede observar el equipo que será el firewall, el cual también hará el NAT de la red privada a la red pública y del mismo modo será el BRIDGE y se encuentra señalado como DHCP Prueba. En la figura 15 se muestran las conexiones de red, una para el caso de calamar, y 2 conexiones para el caso del equipo que será el firewall NAT y DHCP.

Marco Teórico

Conceptos Generales

Red de datos

Una red de datos [33] permite la comunicación y transferencia de información digital mediante un medio físico, como cable, fibra óptica o inalámbrica. Cuando uno o más dispositivos de comunicación están conectados entre sí a través de un medio, estos pueden intercambiar información. Se dice que es una red y se dice de datos cuando dicha red incluye no solo dispositivos de comunicación como switches y routers, sino también equipos de usuarios finales como computadoras e incluso algún dispositivo Internet of Things (IoT)[16]. Cuando estos equipos comparten información en el mismo canal de comunicación y mismos protocolos, es entonces que tenemos una red de transferencia de datos, en la cual, por definición de comunicación, se aplica el concepto de emisor-mensaje-receptor, que interactúan y se retroalimentan de los mensajes en ambos sentidos.

Red de computadoras

Una red de computadoras [33], como se muestra en la figura 2, se define como una conexión básica de una computadora a otra, también si se comunican con más computadoras a través de un canal y un protocolo de comunicación.
El medio y el canal de comunicación pueden ser mediante un cable con pulsos digitales eléctricos, o por pulsos electromagnéticos en el espacio, u ópticamente mediante una fibra óptica con pulsos de luz en, o cualquier otro.

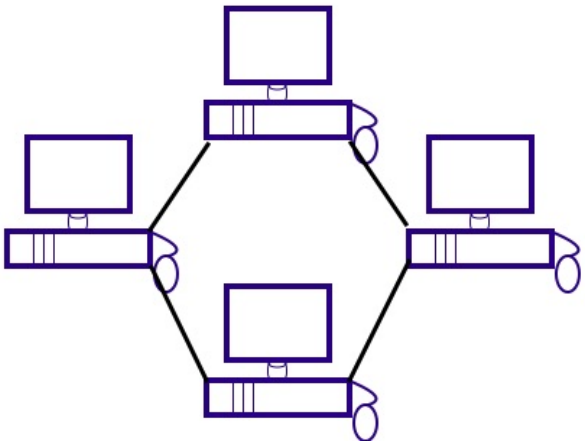


Figura 2. Ejemplo de red de 4 computadoras conectadas por cable mediante una topología de anillo [33].

Principales tipos de redes y topologías

Las redes de se dividen en varios tipos dependiendo de su naturaleza de conexiones y protocolos de comunicación [51], los tipos de redes son:

[16] ***El Internet de las cosas (IoT):***Son aquellos aparatos eléctricos, electrónicos y electrodomesticos(lavadoras, refrigeradores, televisiones, focos, etc.), que cuentan con acceso a internet mediante la instalación de una electrónica que permite dicha tarea, estos dispositivos pueden ser controlados por medio de accceso a el Internet, sus funciones van desde encendido apagado hasta seleccionar funciones mas complejas como controlar todas las funciones del dispositivo IoT.

- ***Redes Públicas:*** No tienen restricción en su uso y es accesible desde cualquier lugar, como el caso de la internet.
- ***Red Privada:*** (PN: Private Network) Están limitadas a un grupo restringido de usuarios por medio de protocolo, o direcciones privadas de internet.
-

De igual forma, se pueden dividir por su tamaño como:

- ***Redes de Área Personal:*** *(PAN: Personal Area Network)* Se usan por un solo usuario, en la actualidad se generan varios canales para que se usen de manera personal.
- ***Red de Área Local:*** *(LAN: Local Area Network)* Son definidas por una organización, las cuales comparten información entre sus usuarios.
- ***Redes de Área Metropolitana:*** *(MAN: Metropolitan Area Network)* Conectan a ciudades de varios tipos de redes entre sí, generalmente las redes metropolitanas están divididas por ciudades pequeñas, aunque hoy en día las redes metropolitanas se extienden por cientos o miles de kilómetros dependiendo la densidad de los usuarios.
- ***Redes de Área Amplia:*** *(WAN: Wide Area Network)* Conectan a las redes metropolitanas entre sí.

En la figura 3 se pueden ver los tipos de redes con sus acrónimos en inglés, y como van de abajo hacia arriba de la más pequeña a la más extensa.

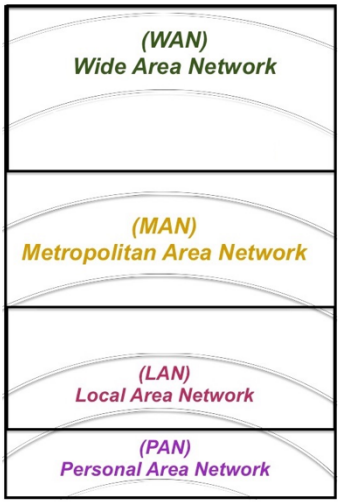


Figura 3. Clasificación de los tipos de redes de datos, de acuerdo con su extensión de cobertura y número de nodos: PAN, LAN, MAN, y WAN [51].

Topologías de red

Una topología de red [33] se define como la distribución física o lógica de la red de datos. Cada diseño de topología depende de la distancia, el tipo de conexión y distribución de todos los nodos, y esta configuración tiene ventajas y desventajas según el uso que se quiera dar a la red de datos.
La clasificación de la topología depende de su forma física o de la forma en que se distribuyen los datos lógicamente, también dependen del hardware y software manejados en cada nodo o de un equipo especializado que interconecte los nodos entre sí.
Las topologías físicas de una red de datos más comunes son: anillo, bus, árbol, malla, estrella, jerárquica e híbrida (combinación de 2 o más topologías).

La Topología lógica se refiere a la forma en que una red de datos transfiere la información entre los nodos. Esta disposición consta de conexiones lógicas entre los nodos de una red. Los protocolos en la parte de enlace de datos definen estas rutas en forma de señales lógicas.
Partiendo del hecho que una red se conforma de 2 o más computadoras o dispositivos de red conectadas entre sí, la primera en definirse es la más simple, la conexión entre 2 equipos, a esta la llamamos protocolo punto a punto o *point to point protocol(PPP)*.
A partir de que existe hardware diseñado para interconectar dos o más computadoras o dispositivos de red, definimos las siguientes topologías:
Bus: Tiene un único flujo que llamamos bus de comunicaciones o bus troncal (backbone) donde se conectan los diferentes dispositivos, por lo cual todos los dispositivos tienen el mismo canal.
Estrella: Los dispositivos de red (nodos o computadoras) están conectadas mediante un nodo central el cual se encarga de interconectar a todos los nodos, a este equipo normalmente se le llama concentrador hub (enlazador) o también puede ser un equipo llamado switch (conmutador).
Anillo (circular): La conexión de cada uno de los nodos depende del nodo contiguo, de tal forma que al cerrar los enlaces se forma una ruta única y continua, la cual forma un círculo cerrado, por lo cual debe su nombre de anillo, existe el anillo doble que tiene una conexión redundante que recorre el mismo anillo, pero por un canal diferente de tal forma que si el anillo principal falla, el segundo anillo establece la comunicación que el principal no puede.
Malla: Todos los nodos están interconectados entre sí. Una característica de esta topología de red es disminuir la distancia de conexión entre los nodos, además de que existe tal redundancia de conexión que representa una topología contra fallas de acceso.
Árbol o jerárquica: Se dice árbol porque cada nodo tiene una rama y de ahí nacen hojas que son otro grupo de nodos interconectados con la rama. Se puede decir que son topologías de estrellas interconectadas por una ruta específica, la cual vendría siendo la rama del árbol que físicamente sería un hub o un switch, existen switch especializados para interconectar redes lógicamente diferentes, a estos equipos los conocemos como router (enrutador).
Híbrida: Pueden interconectarse redes de diferentes topologías entre sí, la Internet está conformada por una topología híbrida donde conviven todas las tipologías.
En la figura 4 se muestra gráficamente cómo son las diferentes topologías que definimos anteriormente.

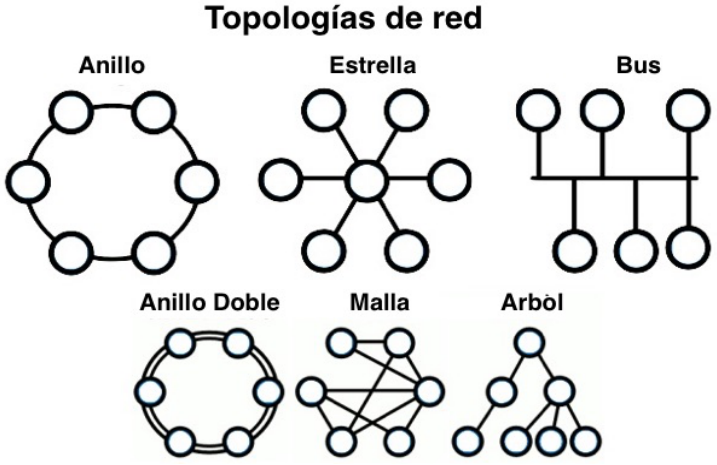


Figura 4.: Los 6 tipos de topologías de red Anillo, Estrella, bus, Anillo Doble, Malla y Árbol.

El modelo OSI de red

El modelo OSI [33] de su acrónimo en inglés **O**pen **S**ystems **I**nterconnection, es el modelo de referencia para los protocolos de la red, fue diseñado desde 1977 y publicado en 1983 por la Unión Internacional de Telecomunicaciones (UIT) y un año después por la Organización Internacional de Normalización (ISO). Es el estándar que se usa para interconectar sistemas de distinto diseño y origen, y de esta forma puedan intercambiar información, debido a que sus los protocolos que conforman el modelo que son para este fin, y es el estándar utilizado en la actualidad
En el modelo conceptual, el modelo OSI tiene 7 niveles de abstracción, donde cada nivel tiene una función específica y está definida por su conjunto de protocolos. En la Tabla 1 describimos cada una de las capas o niveles y la función de cada una.

Nivel/Capa OSI	Nombre	Función
7	Aplicación	Es la capa que permite al usuario establecer varias formas de comunicación mediante diferentes aplicaciones mediante diferentes protocolos de comunicación.
6	Presentación	Se encarga de convertir los datos binarios a representación de alto nivel, para que el usuario pueda entender los mensajes y enviarlos.
5	Sesión	Se encarga de establecer la comunicación de emisor-receptor, la concurrencia y verificar la validez de los paquetes.
4	Transporte	Se encarga de aceptar los datos de las capas superiores para pasarlos a la capa de red.
3	Red	Se encarga que los datos lleguen correctamente al destino.
2	Enlace de Datos	Se encarga de la topología de la red, es decir, qué tipo de conexiones como bus, estrella, árbol.
1	Física	Se encarga de las conexiones físicas electrónicas de los datos (Cableado, Fibra Óptica, Señal Electromagnética).

Tabla 1: Capas del modelo OSI [33].

Protocolo de comunicación en la red

Un protocolo de comunicación [33] lo definimos como el conjunto de reglas que rigen los mensajes que se intercambian entre 2 o más equipos de cómputo o de red de datos. Como observamos en el modelo OSI, los protocolos de comunicación se establecen por niveles y se llaman capas, y juntas, una pila de protocolos, donde todos interactúan entre sí y de manera jerárquica, desde el usuario la capa más alta hasta el medio físico de conexión, es decir, la capa más baja, las capas o niveles van desde la capa 1 la física hasta la capa 7 de aplicación.

Protocolo TCP/IP

El acrónimo TCP/IP [44] proviene de las siglas en inglés *Transmission Control Protocol/Internet Protocol* [25] *y* al igual que el modelo OSI, TCP/IP es un modelo jerarquizado en el cual existen 4 niveles o capas, se divide en:

1. Acceso al Medio y Física.
2. Red e Internet.
3. Transporte y Host a Host.
4. Aplicación.

El significado de TCP/IP proviene del acrónimo en inglés Transmission Control Protocol/Internet Protocol, su origen viene de la creación de la primera red de computadoras llamada ARPANET, la cual tenía como objetivo conectar a las computadoras de diferentes sistemas y fabricantes entre sí, TCP/IP fue en parte la evolución de ARPANET y fue desarrollado por Vinton Cerf y Robert E. Kahn en los años 70. *TCP/IP* es un protocolo que sirve para establecer y estandarizar la comunicación, y es el estándar del protocolo. TCP/IP en realidad son 2 tipos de protocolos que se reúnen para definir lo que hoy en día conocemos como el protocolo de Internet, ya que es el conjunto de protocolos que enlazan a la red de manera virtual en una sola red de alcance mundial.

Las reglas aplicadas al modelo OSI son normas que no distinguen al fabricante del equipo ni su sistema de comunicación de fábrica, ya que, si tiene el protocolo estándar, deberá establecer la comunicación en la red de datos, y actualmente es el conjunto de protocolos TCP/IP. Por ejemplo, para enviar un dato por medio de TCP/IP, debemos comenzar en la capa más alta que es a nivel Usuario, es decir, comenzamos en la capa 7 de OSI hacia abajo, en donde es hasta la capa 5 se ejecuta la aplicación, posteriormente de la capa 3 a la 1 son acciones de transporte de datos como se muestra en la tabla 2.

Capa Nivel OSI	Acción en OSI	Capa Nivel TCP/IP	Acción TCP/IP Ejemplo
7 6 5	Aplicación Presentación Sesión	4	Aplicación: NFS, NIS, DNS, LDAP, HTTP, HTTPS, SSH … y otros
4	Transporte	3	Protocolos TCP, UDP, SCTP
3	Red	2	Protocolo: Pv4, IPv6, ARP, ICMP [8]
2 1	Enlace Físico	1	Físico y Enlace: PPP, T1, Ethernet, Token Ring, PP, RS-232, FDDI

Tabla 2: Comparativo entre capas del Modelo OSI y TCP/IP [44].

Protocolo IP (Internet Protocol):

En el caso del modelo de protocolo TCP/IP [44], una de las tecnologías fundamentales para operar es la conmutación de paquetes, cada paquete se fragmenta en paquetes más pequeños, que se transmiten fragmentados en la red, y luego en el cliente se reconstruye el paquete original.

Los datos se generan desde la capa o nivel más alto que es la del usuario, según el RFC 791 [8] “INTERNET PROTOCOL, DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION”

publicado en septiembre del 1981, vemos la trayectoria de transferencia de datos, comienza en la Aplicación (programa) posteriormente pasa al módulo de internet (Internet Module), de allí pasa a la interfaz local de internet 1 (Local Network Interface 1) , para ir a la Red Local 1 (Local Network 1), posteriormente regresa al módulo de internet, donde avanza a LNI-2 y Local Network 2, regresa al LNI-2 de nuevo al módulo de internet para recibirse en el programa la aplicación del usuario como una respuesta a la comunicación. Para entenderlo mejor, podemos ver la figura 5, en donde se explica la fragmentación de paquetes que van ordenados de manera secuencial.

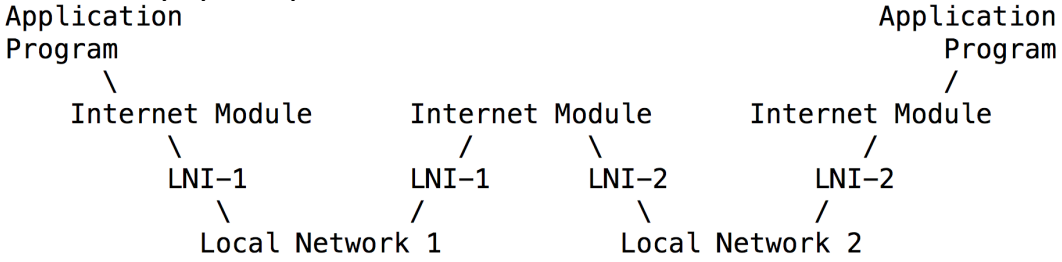


Figura 5: Ruta de transmission (Transmission Path), correspondiente al RFC 791 [8] *INTERNET PROTOCOL publicado en septiembre de 1981, por Defense Advanced Research Projects Agency Information Processing Techniques Office.* [8]

Las direcciones se separan en fragmentos de 8 bits, que para el caso de IPV4 son cuatro fragmentos o 32 bits, que se transmiten en sesiones identificadas con un número único que se establece durante la transmisión de los datos. En la figura 6 se observa el flujo de la información con las capas del modelo TCP IP. Las flechas hacia abajo representan el flujo en el transmisor y las flechas hacia arriba representan en flujo en el receptor.

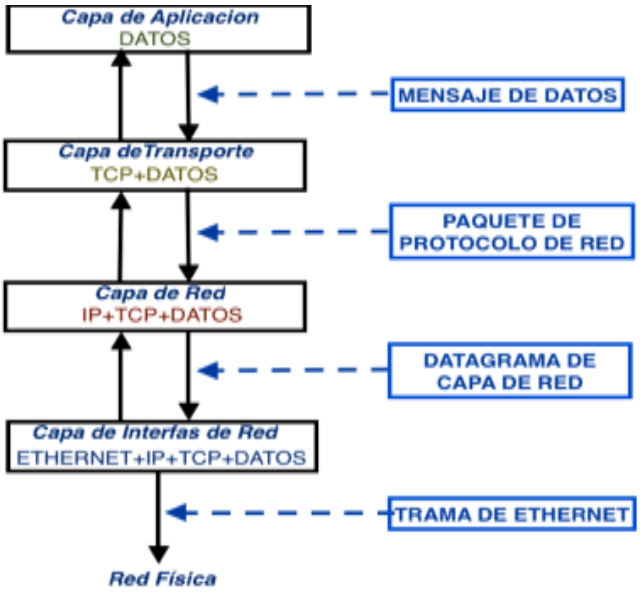


Figura 6. Cómo fluyen los datos en ambas capas de direcciones a través de TCP/IP [44].

En el caso de TCP/IP solo existen 4 capas o niveles conceptuales, los cuales los describimos con otros protocolos que forman parte de OSI y de TCP/IP.

Red Pública de internet o Internet Protocol versión 4 (IPv4):

El protocolo definido en el RFC 1180 [25] evolucionó hasta tener la versión 4 que llamamos IPv4, este está conformado por direcciones de 32 bits, en fracciones de 1 byte separados por un punto. Como estas direcciones se forman por 32 bits, estas permiten crear 4,294,967,296 direcciones diferentes. Para facilitar algunas funciones como la de enrutamiento, las diferentes direcciones se agruparon en cinco clases que se nombran con letras del abecedario desde la A hasta la E. Cada dirección se define en intervalos de la tabla 3.

Para poder utilizar una dirección IP (Internet Protocol) pública es necesario solicitarla a InterNIC (Public Information Regarding Internet Domain Name Registration Services), que es la organización mundial dedicada a asignar direccionamiento, con el fin de poder acceder a Internet.

Cuando se creó IPV4, los más de cuatro mil millones de direcciones eran suficientes para asignar cada computadora y dispositivo de red de datos del mundo, pero con el avance de la tecnología y la necesidad de conectar todo a internet, esta cantidad ya se rebasó. Ahora, solo se reciclan direcciones que se desocupan, y la mayoría del direccionamiento es privado y a veces son redes privadas sobre privadas y así sucesivamente. Para poder lograr esta configuración en IPV4, describimos a continuación cómo una red o conjunto de redes privadas puede salir a internet con una sola dirección pública, mediante un *Network Address Translation (NAT).*

Red Pública:

Clase	Inicio Direccionamiento Público	Término Direccionamiento Público
A	0.0.0.0	126.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255
E	240.0.0.0	254.255.255.255

Tabla 3. Red Privada o direccionamiento de internet IPv4 (Internet Protocol versión 4) [25].

Network Address Translation (NAT):

Cuando se observó el crecimiento de internet, se decidió realizar distintas estrategias provisionales:

- Reservar algunos bloques de direcciones (direcciones privadas).
- Uso de direccionamiento sin clase (CIDR).
- NAT (traducción de dirección de red) [26].

El protocolo de IPv4 permitía a los nodos u hosts conectarse a internet (haciendo uso de IP públicas). Cada empresa u organización acudía al RIR correspondiente y este asignaba un bloque de direcciones de acuerdo con sus necesidades, así ya podían tener conexión. El término de dirección IP pública se refiere a las enrutables en internet y no a que cualquier persona u organización pueda usarlas públicamente sin contrato previo.

También se usaban direcciones IP privadas, que no son enrutables en internet y por ello, no se necesita pedir permiso a ninguna entidad para poder usarlas. Ocurre lo contrario que con las públicas, ya que es necesario contratarlas y hacer una solicitud formal a un ISP o un RIR dependiendo del tamaño de la organización/empresa/hogar que vaya a usarla.

En el direccionamiento privado están las redes privadas definidas en el RFC 1918 [26], para usarse en una LAN de manera local, es decir, que no se puede enrutar ni acceder desde redes públicas, ni a internet. Las organizaciones usan las redes privadas para crear redes locales administradas por la misma organización y permiten solucionar el problema de falta de direccionamiento público. A diferencia del direccionamiento público, el direccionamiento de la red privada no es asignado por la InterNIC. En la tabla 4 se muestran las tres clases que se puede utilizar para el direccionamiento privado. Estas clases varían en tamaño, o sea, en el número de clientes que pueden albergar cada una.

Clase	Inicia	Termina
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

Tabla 4. Direccionamiento de protocolo IP (Internet Protocol) privado [33].

El acrónimo NAT proviene del inglés *Network Address Translation*, consiste en la traducción de direcciones IP privadas a públicas. El trabajo consiste en convertir una dirección IP privada y traducirla a una dirección IP pública o viceversa. Se usa cuando se requiere que los dispositivos en la red (con IP privadas) se comuniquen a internet. Esto se creó para que las organizaciones tuvieran direccionamiento IP sin necesidad de solicitar direcciones públicas, además permite que salgan todos los conectados en la organización con direccionamiento IPV4 con una dirección pública de internet.

En 2023, se solucionó el problema de falta de direccionamiento público, porque el crecimiento de internet llevó al IPV4 a su límite, es decir, se terminaron las direcciones públicas. El direccionamiento privado fue una solución a la falta de direccionamiento público del IPV4, pero desde 2012 ya se aplica el nuevo protocolo IPV6. Sistemas operativos, equipos de comunicación y servidores lo fueron adoptando poco a poco, pero aún, muchos equipos no están listos para el cambio, por ser complicados en su manejo de direccionamiento y por falta de actualización, configuración e implementación a nivel mundial.

En la figura 7 mostramos que un dispositivo con una dirección IP privada local (por ejemplo: 10.0.0.10) proporcionada por un DHCP (Dynamic Host Connection Protocol) enviará al dispositivo en cargado de la NAT todas las solicitudes a Internet y este a su vez traducirá a la IP pública 132.248.1.15 y así saldrá la dirección IP 10.0.0.10 a Internet. La solicitud de algún equipo de Internet devolverá el paquete a la IP 132.248.1.15 que es la IP pública, entonces la NAT va a traducir esa IP pública a la privada y la va a enviar al equipo para que reciba el paquete de vuelta. Todo esto sucede con los demás equipos con direccionamiento privado, lo que implica que los destinos de internet solo verán las múltiples peticiones de una sola dirección pública, pero estas las hacen múltiples equipos de la red privada.

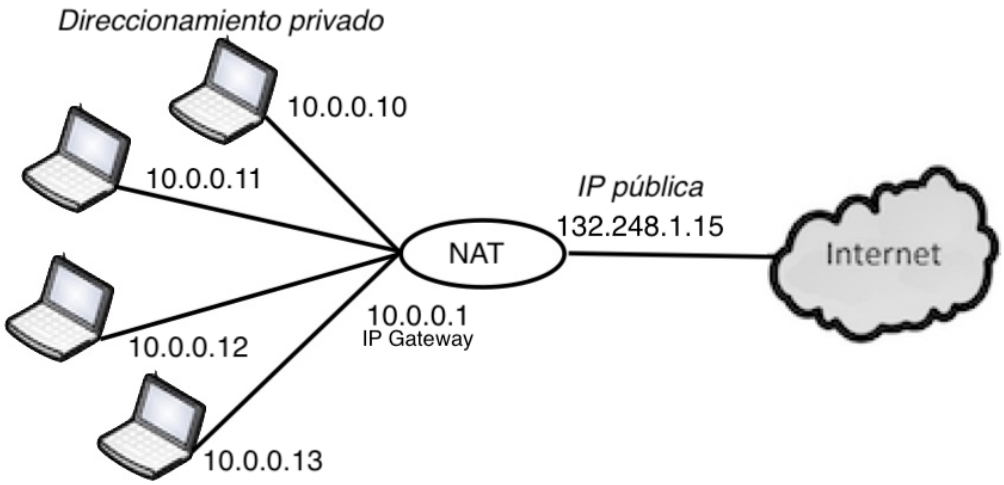


Figura 7. Ejemplo de la topología de NAT (Network Address Translation), para el caso de direccionamiento privado a público [26].

Existen varios tipos de NAT, como son NAT estática, NAT dinámica, NAT con sobrecarga, PAT, que mencionamos únicamente como referencia, ya que en este trabajo utilizaremos NAT estática.

Una NAT permite utilizar una sola dirección IPv4 para muchos equipos; además, su configuración es de forma sencilla y no requiere mantenimiento constante. Al estar los dispositivos aislados, ofrece seguridad perimetral, ya que solo depende de la configuración del NAT. Por último, ofrece conectividad entre los miembros de direcciones IP privadas y es compatible con los protocolos TCP y UDP.

El usar una NAT también trae desventajas, por ejemplo, el equipo NAT requiere de mucho mayor capacidad de procesamiento para atender todas las solicitudes de los clientes en direccionamiento IPV4 privado. Además, es incompatible con algunos protocolos como ejemplo el ICM y se pierde el seguimiento IP de extremo a extremo. También es importante destacar que no es una solución óptima para resolver problemas de manera remota, ya que la velocidad de transferencia de los dispositivos dentro de la NAT depende de la capacidad del NAT y del enlace principal.

Throughput (Velocidad de transferencia de datos)

El throughput [21] o "velocidad de transferencia de datos" generalmente se confunde con el ancho de banda.

La diferencia entre ambos es muy simple, el ancho de banda se refiere al canal físico de conexión en una red de datos, es decir, el máximo rendimiento de transferencia de información con respecto al tiempo y está limitado por las características del canal físico de comunicación. Esto se refleja como el límite físico si es Fibra Óptica o es cable o es conexión de alta velocidad o si es monomodo o multimodo, para lo cual decimos el ancho de banda y nos referimos a la cantidad de datos máxima que el canal puede transmitir en un período de tiempo fijo.

La velocidad de transferencia que en adelante llamaremos *throughput* no solo depende del canal físico de comunicación. Además, depende de la saturación del canal, de los equipos terminal y origen, así como de las conexiones de red, sistemas operativos, interferencias, saturación de los dispositivos de red conectados, capacidad de procesamiento de los equipos que se conectan entre sí, etc. A esto llamamos el rendimiento, el cual se mide en bits por segundo (BPS). Generalmente, se puede expresar en megabits por segundo (Mbps), o gigabits por segundo (Gbps).

Es claro que esta velocidad varía todo el tiempo, lo que implica que no es fija, por lo cual hablamos de la "*Tasa de conexión"* la cual es un promedio de velocidad a la que se establece una conexión

entre dos dispositivos. Esta se utiliza para medir la velocidad a la que el adaptador de red se comunica con otro. Esta medida incluye información de cabecera, encriptación, datos que se transfieren, etc.
En este trabajo veremos que la velocidad de throughput depende también del Sistema Operativo que se utiliza, y de hecho al colocar un equipo entre dos puntos de comunicación el throughput se ve disminuido.

Tiempo de respuesta o Latencia

El tiempo de respuesta en una red de datos o *Latencia* [21] es el tiempo que tarda un equipo en responder a una solicitud específica de la red de un dispositivo de red a otro, esto independientemente de la velocidad de transferencia, ya que los dispositivos de comunicación tardan un tiempo en procesar las peticiones de conexión, y este retardo corresponde al tiempo de respuesta que tarda en contestar la petición de enlace.
Este tiempo aumentará según los equipos de comunicación en el camino entre el transmisor y el receptor, o también depende de la cantidad de peticiones que atender. Con la conmutación de paquetes, la latencia disminuye sensiblemente, pero sigue dependiendo de la capacidad de respuesta de las peticiones de los equipos y dispositivos de red. La latencia también se afecta cuando el enlace es débil o existe ruido en la red que impida que se establezca la conexión. Esto sucede mucho en las redes inalámbricas, ya que la calidad de la conexión es directamente proporcional a la señal entre los equipos de la red.
La respuesta de los equipos enlazados en la red también depende de las peticiones que tengan que atender, esto no solo afecta la latencia, también afecta la velocidad de transferencia de la información. Por esta razón, en este trabajo se mide la velocidad de transferencia, así como la latencia el mismo tiempo para ver el desempeño real de los equipos a caracterizar.

Puente de comunicación o Bridge

Los puentes de comunicación en una red de datos sirven para varias razones. Una es recuperar la señal física, la cual se puede ver afectada por la distancia o por el ruido en el canal de comunicación. La diferencia con un NAT es que el Bridge no cambia la configuración TCP/IP; sin embargo, sí efectúa cambios en la capa 2 de enlace de datos, donde la dirección física o MAC address es sustituida por la del puente o Bridge. A diferencia de un hub (multi conector de red) o un switch (conmutador de datos), el bridge (puente de datos) tiene solamente una entrada y una salida.
Existen bridges que incluso cambian la interfase de comunicación y hasta la velocidad del ancho de banda, también son capaces de filtrar protocolos en capa 3, 4 y 5 del modelo OSI, aunque ya con esas características no son considerados de un bridge, también se utilizan para vigilar el tráfico de la red, en cuyo caso son equipos especializados para realizar dicha tarea.
En el caso de los firewalls implementados mediante Sistemas Operativos como CentOS u OpenBSD, la traducción de las IP se realiza en el software “iptables” o “Packet Filter” respectivamente para cada uno de estos Sistemas Operativos, y no es así en los “bridges” o puentes, ya que allí el enrutamiento puede ser transparente, es decir sin reglas y el tráfico pasara con la afectación que implica el acceso directo a la tarjeta de red por medio del Sistema Operativo.

Seguridad informática

Una definición muy simple de seguridad es la ausencia de peligro o algo que afecte nuestra propiedad, integridad y en este caso, la información, así como los equipos que la transmiten y la

contienen. Para poder entender este concepto se requiere explicar qué es lo que puede poner en peligro la información y todo lo relacionado con ella a nivel informático.
Se define la ***amenaza*** a la seguridad como un fenómeno o proceso natural o causado por el ser humano que puede poner en peligro la información, ya sea porque las afecte directa o indirectamente por sus efectos sobre el hardware o software [27]. En un sentido más específico, las amenazas van desde que se dañe donde se aloja la información hasta que sea afectada de manera intencional para hacer daño a una persona u organización. En esta parte están incluidos desastres naturales como inundaciones, terremotos, incendios, fallas de corriente eléctrica, por mencionar algunos ejemplos. También se incluyen causas humanas como personas o programa (virus informático), que pueden causar daño o pérdida de la información.

Riesgo

El riesgo es la probabilidad de que suceda algo que cause un daño a la información y a los equipos que la contengan, al ser una probabilidad es posible cuantificar, y en general es lo que las compañías de seguros evalúan para cobrar un seguro.

Vulnerabilidad

Una vulnerabilidad se le conoce como a la incapacidad de resistencia cuando se presenta un evento que dañe la información, también se define como la incapacidad para reponerse de un daño o un siniestro.
La seguridad informática se conoce como ciberseguridad, pertenece al área de la informática encargada de proteger el hardware y software y todo lo relacionado con la informática, y no solo proteger el hardware, sino principalmente la información contenida en este. Esto lo logramos gracias a poder determinar las amenazas, riesgos y vulnerabilidades que existen en un sistema informático, principalmente en el software y la información contenida en el hardware.
Para hablar de seguridad tenemos que definir que contiene un sistema seguro, en esta parte se conjugan tres definiciones fundamentales que definen seguridad.

- Confidencialidad: Consiste en el acceso a la información solamente por aquellos que tienen permiso para ello
- Integridad: Esta característica nos habla el cómo se guarda la información, es decir, que no se dañe en donde está contenida la información, ni tampoco que se pierda parte de esta información.
- Disponibilidad: Esta característica implica que la información está completa y por alguna razón (hardware o software) no se puede acceder a ella. Un ejemplo muy común es el no tener u olvidar la contraseña del correo electrónico, otro ejemplo es cuando tenemos la información en un medio o en un formato irreconocible y por la cual no podemos acceder a ella.

En la figura 8 se muestran las tres definiciones, las cuales describen la seguridad informática, y a esta figura se le conoce como el *"Triángulo de la Seguridad Informática" [27].*

Figura 8: Triángulo de la seguridad [27].

No hay un orden en estas definiciones, es decir, las tres están en el mismo nivel de importancia y sin una de ellas, se ha perdido la seguridad informática.

Firewall

Un firewall (pared de fuego) en informática es un equipo o programa capaz de analizar el tráfico de la red y filtrar o bloquearlo según reglas predefinidas. Los firewalls funcionan mediante reglas de salida y entrada. La mayoría de los firewalls relacionan servicios de nivel 6 en la capa OSI con ciertas reglas de filtrado en nivel 3 de OSI. Un firewall perimetral de nivel 3 en OSI, es aquel que permite salida y entrada mediante reglas de filtrado. Estas reglas nos especifican un servicio (un puerto) relacionado con la capa 3 del modelo OSI. El filtrado se va construyendo con reglas simples, como si se permitirá la entrada o salida de datos de la dirección de IP o el servicio especificado o ambos. Así se construyen las reglas de filtrado para todas las direcciones IP y para todos los servicios, como SSH, HTTP, HTTPS, NTP, etc. La idea de estas reglas es especificar que direcciones tienen que servicios, para así salir y entrar solo lo necesario por la organización.

Hay dos definiciones de filtrado de firewalls: los permisivos que permiten salida y entrada, excepto las especificadas en las reglas, y los restrictivos que no permiten ni salida ni entrada de datos, excepto lo definido en las reglas especificadas. Los firewalls perimetrales están en la salida a Internet (ISP), pero también existen los que funcionan en el Sistema Operativo de los equipos de trabajo. Los firewalls pueden estar localizados también en dispositivos de red, como switches o puntos de acceso inalámbricos. El filtrado de datos que realiza un firewall está dividido en varias secciones del modelo OSI, y puede ir desde la capa física, hasta la capa de aplicación.

Si es un dispositivo, se parece a cualquier equipo de comunicación como un switch, con la diferencia de un hardware y un Sistema Operativo dedicado a él filtrado de datos, si es un programa, se implementa dentro de Sistema Operativo. Todos los Sistemas Operativos tanto de GNU como los propietarios tienen la opción de tener esa característica de configuración de firewall y de hecho ya vienen con reglas predefinidas cuando se instalan, y algunos pueden instalarse como firewalls perimetrales. Existen equipos de comunicación especializados como firewalls, solo mencionaremos algunas marcas como Cisco®, Fortinet®, Palo Alto®, etc.

Los sistemas operativos como el OpenBSD en sus diferentes versiones permiten diseñar e implementar firewalls perimetrales con un software especializado embebido dentro del Sistema Operativo llamado PF (Packet Filter). Otro Sistema Operativo que se puede utilizar como firewall perimetral es CentOS en sus diferentes versiones. En el caso de CentOS, mencionaremos que su sistema de seguridad está basado en Iptables. Los sistemas de firewall embebidos en los diferentes sistemas operativos mencionados se configuran en software para tener una topología adecuada para el filtrado y limitación de la red LAN tanto física como lógicamente. Los equipos comerciales de marca están diseñados para tener la funcionalidad de firewalls, y tienen el hardware y software desarrollados para ese fin específico y por la misma marca.

En este trabajo nos enfocaremos en proteger la red LAN del IA-UNAM, por lo que describiremos solo la topología de un firewall perimetral, que consiste en tener 2 interfaces, una de entrada y otra de salida.
La interfaz que llamamos de entrada es aquella que está conviviendo en nuestra red de datos local, y la interfaz de salida es aquella que está conectada directamente al Internet Service Provider (ISP), como mostramos en la figura 9. En la topología se muestran dos redes, la primera donde están las "computadoras" en verde, las cuales pertenecen a la red LAN, y la segunda en donde está la "nube" donde está definida la red WAN que corresponde al acceso a internet, dentro del firewall en naranja es donde se filtra el tráfico de internet es decir donde se aplican las reglas de entrada y salida de la red LAN hacia Internet y viceversa.

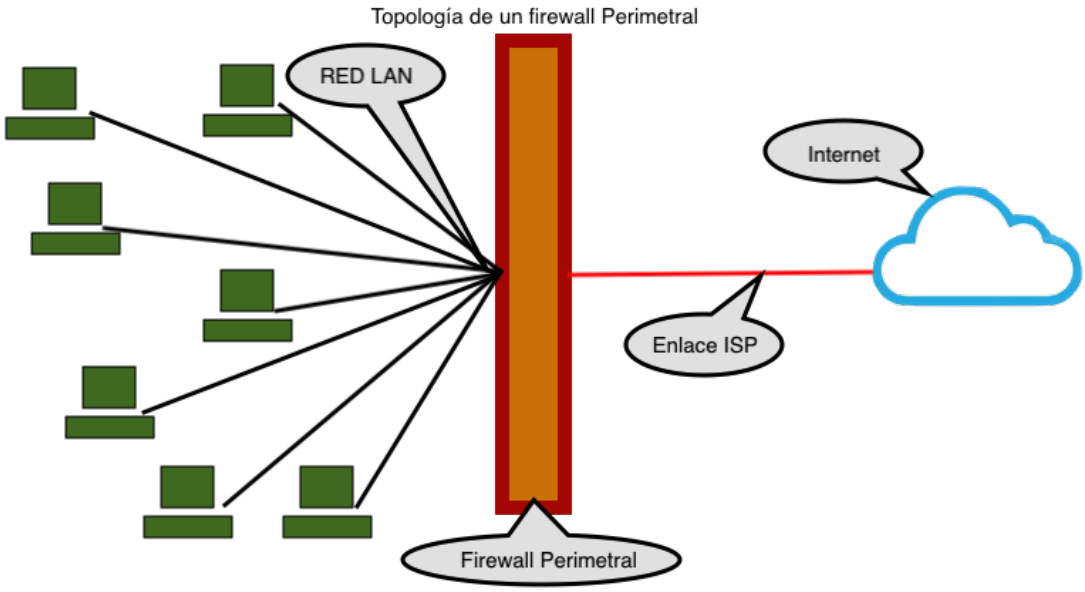


Figura 9. Topología de un firewall perimetral.

Los firewalls funcionan según su ubicación en la topología de la red, según el filtrado de datos que realizan y según el nivel OSI en el que filtran los datos.
Para el año 2023 existen 5 tipos principales de firewalls [48]:

1. **Firewall proxy:** Es el que se encarga de Filtrar con una puerta de salida (Gateway) los paquetes y su contenido, Firewall de inspección activa que hace una inspección activa, el más común, pues permite o bloquea el tráfico según el estado, el puerto y el protocolo. y Monitorea toda la actividad de la red, en general este tipo de firewall está en la Capa 3 del Modelo OSI.
2. **Firewall de administración unificada de amenazas (UTM):** Esta combina de forma flexible las funciones de un firewall de inspección activa con prevención de intrusiones y antivirus.
3. **Firewall de filtrado de paquetes:** Este tipo de firewall filtra los paquetes de datos en el Nivel capa 3 del modelo OSI por lo cual filtra las direcciones de red, protocolos o puertos.
4. **Firewall de aplicación:** Este tipo de firewall no utiliza direccionamiento ip ni puertos de acceso de la capa 3 del OSI, utiliza los encabezados de los paquetes, los cuales llevan la característica de la aplicación utilizada, y por medio de esta información deja pasar o bloquea la comunicación.
5. **Firewall de próxima generación:** Un firewall de próxima generación *NGFW (Next Generation Firewall),* funciona por medio de la inspección profunda de paquetes, la cual

permite bloquear aplicaciones e instrucciones que pudieran ser maliciosas, es decir, no solo opera con servicios ni con protocolos ni con puertos de acceso, funciona analizando los paquetes y de manera heurística y por aprendizaje bloquea instrucciones y peticiones maliciosas.

Sistema Operativo como firewall

Es posible utilizar una computadora como firewall perimetral instalado con algún sistema operativo que lo permita. En el caso del Sistema Operativo OpenBSD, que nace del proyecto tipo UNIX gratuito y multiplataforma basado en 4.4 BSD [52].

La política de OpenBSD se esfuerza por proporcionar código que cualquier persona pueda usar, copiar, modificar y distribuir libremente y para cualquier propósito. Esto mantiene el espíritu de Berkeley Software Distribución original. La redacción preferida de una licencia que se aplicará al código nuevo se puede encontrar en la plantilla de licencia. OpenBSD se mantiene por Computer Systems Research Group de Berkeley y otros, el Unix "Berkeley" se distribuye de manera gratuita. Los precursores del proyecto dan portabilidad, estandarización, corrección, y la seguridad proactiva y la criptografía integrada. Entonces surge OpenBSD, el popular software OpenSSH que proviene de OpenBSD, y está disponible en forma gratuita en su sitio https://www.openbsd.org/, el 10 de abril de 2023 se publicó la versión OpenBSD 7.3 el cual está desarrollado por voluntarios. El financiamiento del proyecto es a través de contribuciones recaudadas por la Fundación OpenBSD.

Debido al manejo de Packet Filter en su Kernel (Núcleo del Sistema Operativo), done en el caso del OpenBSD permite instalar y configurar un firewall perimetral, esto porque el OpenBSD tiene un desarrollo de alto nivel de algoritmos de cifrado de datos que tiene el Kernel y ayuda al manejo de la seguridad haciéndolo uno de los Sistemas Operativos más seguros distribuidos como software libre. Por tanto, un firewall (pared de fuego) es el encargado en la red de filtrar paquetes de información, por medio de reglas preestablecidas, por tanto, un firewall (pared de Fuego) nos permite limitar, cifrar y hasta de ser necesario decodificar el tráfico de comunicaciones entre una computadora en una red local o internet, deteniendo los accesos o transferencia de información que no esté definido dentro de las reglas establecidas por la política de seguridad informática permitida, en términos más generales nos permite protegernos de accesos no válidos a nivel usuario, con un firewall implementado con el Sistema Operativo en una computadora, y a nivel red LAN permite proteger toda la red con políticas generales de acceso y salida.

Existen varios tipos de firewall, se dividen en 2 grupos, **los definidos por su forma de instalación y por su forma de protección**.

CentOS 7 como firewall perimetral

Primero definiremos el proyecto del Sistema Operativo CentOS de su acrónimo en inglés “Community ENTerprise Operating System”, el cual es una distribución Linux y consiste en una división a nivel binario de la distribución “GNU/Linux Red Hat Enterprise Linux RHEL”, es decir una versión libre y que es compilado por voluntarios a partir del código fuente publicado por Red Hat.

El Sistema Operativo es de código abierto, basado en Red Hat Enterprise Linux, pero muy similar, su objetivo es ofrecer un software gratuito a toda la comunidad, es robusto, estable y a nivel usuario fácil de instalar y utilizar. Cada actualización recibe soporte durante diez años aproximadamente, y en 2022 se tiene la versión 7 que recibirá actualizaciones de seguridad hasta el 30 de junio de 2024. En el año 2022 se notificó que ya no estará disponible el proyecto CentOS, y sus creadores de Cloud Linux OS, anunciaron la creación de su continuación llamada Alma Linux.5 CentOS es un Sistema

Operativo Robusto usado principalmente para Servidores, y es muy utilizado para implementación de páginas web, así como clústeres, ya que comparte casi el 95% de las características de la Red Hat Enterprise Linux RHEL de IBM,

Forma de Instalación

La instalación está definida **por Software**, es decir, implementado mediante un programa especializado que se ejecuta como una aplicación independiente, p embebido dentro del Sistema Operativo de una computadora, y el segundo son los equipos dedicados que cuentan con características de un equipo de red especializado y además permite el filtrado de datos que pasan por este equipo. Es muy importante para este trabajo mencionar que, al hacer un filtrado y análisis de datos de la red, afecta la velocidad de transferencia de información, y, por tanto, el throughput (velocidad de transferencia) se ve afectado por estos sistemas de filtrado de datos en la red. Los firewalls por hardware suelen ser comerciales, ofrecidos como equipos de red de datos especializados para proteger redes LAN de organizaciones y sirven como equipos de comunicación.
Los firewalls implementados por medio de software son aquellos que están integrados en equipos personales, los cuales mediante un programa o embebidos dentro del Sistema Operativo realizarán la función de filtrado de datos, también se puede implementar mediante una computadora dedicada exclusivamente a filtrar datos de la red, esta computadora puede ser una computadora con características básicas e interfaces o conexiones de red adicionales que se pueden agregar a una computadora de usuario o mediante un servidor, con el cual nos permite mediante el Sistema Operativo el filtrado de datos en una red LAN.
Es importante mencionar que al estar integrado al Sistema Operativo del equipo a utilizar el filtrado de datos es más eficiente.

Herramientas para la medición del throughput y latencia

Iperf

Una de las herramientas a utilizar para medir la velocidad de transferencia es *Iperf*, el cual se utiliza para hacer la medición de la velocidad de transferencia de datos en una red en producción, es decir. que esté operando en condiciones comunes. Su funcionamiento consiste en crear flujos de datos TCP y UDP y de esa forma poder medir el rendimiento de la red.
Iperf fue creado por Distributed Applications Support Team (DAST) en el National Laboratory for Applied Network Research (NLANR), está escrito en lenguaje de programación C++. En Iperf, se pueden ajustar varios parámetros y esto permite hacer diferentes pruebas. Iperf funciona como cliente o como servidor y mide el rendimiento entre los dos extremos en donde se encuentre instalado; funciona unidireccional o bidireccionalmente.
Este software nos permitirá conocer el rendimiento de nuestra conexión, y nos podría permitir hacer el cálculo del throughput para la caracterización de nuestro canal y nuestros equipos con los dos sistemas operativos a utilizar.

Ping

El programa ping utiliza el protocolo ICMP de TCP/IP para la medición de la latencia y nos da un tiempo que incluye la tardanza de transmisión del canal, que puede considerarse despreciable, ya que el canal electromagnético u óptico transmite a la velocidad de la luz. La respuesta va desde que

se envía el pequeño paquete y lo procesa el equipo remoto, indicando el tiempo tardado en ir y regresar el paquete del emisor al receptor. Lo que mide es el tiempo de respuesta sin importar la velocidad del canal. El tiempo que se tarda en responder el equipo remoto generalmente se mide en milisegundos. El comando ping utiliza el protocolo ICMP o Protocolo de mensajes de control de Internet por su acrónimo en Inglés *Internet Control Message Protocol,* es uno de los protocolos de TCP/IP que pertenece a la capa de transporte de datos y está definido en el RFC 792 [29]. Los mensajes ICMP son generados como una respuesta para encontrar errores en los datagramas del protocolo IP definido en la especificación RFC 1122 y son para diagnóstico y ruteo. La versión que se usa de ICMP de IPv4 se le conoce como ICMPv4.

Traceroute

El comando traceroute es un programa que también utiliza el protocolo ICMP que entrega, además, el detalle de los equipos de comunicación por los que está pasando el paquete y que retardo o latencia tiene cada uno de ellos.
En este trabajo se utilizan estos comandos con sus protocolos correspondientes para conocer si el firewall NAT o el transparente están enrutando la comunicación como es debido, y nos arrojará del mismo modo la latencia y el throughput que nos permitirá conocer si el número de reglas aplicadas en el firewall también afecta esta respuesta.

Trabajo Relacionado

Entre las publicaciones citadas, la que está más relacionada con esta tesis y contiene mucha información del rendimiento de la red mediante un firewall instalado con OpenBSD, se titula "Design and Performance of the OpenBSD Stateful Packet Filter (pf)" del año 2002 [1], en este trabajo se describe el diseño y la implementación de firewalls de estado los cuales pueden ser instalados con el sistema operativo OpenBSD y PF (Packet Filter) versión 3.0 , y se compara su escalabilidad y rendimiento con las existentes implementaciones de sistema de código abierto para firewalls de filtrado de paquetes por direcciones ip, además se describe el diseño y la implementación de pf y compara su escalabilidad y rendimiento con las existentes implementaciones de Linux con filtro de paquetes similares, las pruebas se realizaron con computadoras CPU Intel® i386 Pentium a 166 MHz y 64 MB de RAM; todos los dispositivos tienen interfaces de red idénticas (la del cliente, las 2 del firewall, y el servidor), son tarjetas NetGear PCI son idénticas.
La forma de medir la transferencia de datos está hecha con software el software libnet y libcap, software que ya no existe en la actualidad y los resultados del troughtput se dan en paquetes/segundo. Entre los resultados relevantes destacan el rendimiento, la latencia y pérdida dependiendo de la tasa de envío. Para un conjunto de 100 reglas, utilizando paquetes de 256 bytes, Iptables supera a pf e IPFilter [55]. En esta prueba en Iptables se tiene un rendimiento más alto y menor latencia en comparación con los otros dos filtros de paquetes, a diferencia de esta tesis, los cambios son sustanciales, la tecnología es del año 2002, que corresponde a un procesador con cuatro núcleos AMD Opteron 64 Model 285 2.6GHz, 8 GB de memoria RAM, y las interfaces de red son Intel® Ethernet Converged Network 10Gb/s, e igualmente las 4 interfaces de red son iguales (la del cliente, las 2 del firewall, y el servidor), y en cuanto al software, se tiene un OpenBSD y CentOS versiones 7.2, donde el sistema operativo permite distribuir los procesos entre los 4 procesadores, lo que en el artículo relacionado que estamos mencionando no es posible, porque es un solo procesador para los equipos involucrados, al comparar el hardware, revisaremos el rendimiento de los equipos realizando la prueba del throughput y latencia para los 2 sistemas operativos con topología NAT y verificar el rendimiento que tienen los sistemas operativos, así como sus firewalls con los 2 sistemas operativos OpenBSD, así como CentOS con pf e Iptables respectivamente.
Los resultados no son comparables con los resultados de esta tesis, sin embargo, es una gran referencia para saber cómo funcionan los sistemas operativos empleados como firewalls, así como la referencia de cuántas pruebas realizar para poder caracterizar el OpenBSD y CentOS, ambos en la versión 7.2.
Otro trabajo relacionado, fue presentado en la conferencia "The Technical BSD Conference 2019" titulado "Measuring Performance on OpenBSD" [56](https://av.tib.eu/media/45174), donde Alexander Bluhm, nos describe la forma de implementar un sistema por medio de máquinas virtuales para instalar y probar OpenBSD en diferentes versiones, en particular nuestro interés está en la versión 7.2, en la parte del rendimiento de red, porque utiliza interfaces de red virtuales de 10Gb/s y también utiliza las topologías NAT y Bridge para obtener el throughput.
Para medir utiliza la herramienta iperf, la cual nos permite medir el throughput del canal de comunicación, y nos lo entrega en bits por segundo. Megabits por Segundo y Gigabits por segundo. En los resultados se puede observar que el throughput obtenido es de 3487407407.40741bits por segundo, que equivale a un aproximado de 3.5 Gbits por segundo, esto lo podemos verificar en los resultados que muestra en la página de resultados "OpenBSD perform 7.2 release test results" [12] donde se encuentra el detalle de las mediciones. Para poder realizar todas estas mediciones de rendimiento y funcionamiento, utilizan máquinas virtuales con una arquitectura Intel(R) Xeon(R) E-2236 CPU con 3.40GHz, memoria RAM de 32522MB, e interfaces de red de 10Gb/s.

A pesar de que se muestra ya el resultado de las pruebas, solo se prueba el rendimiento del sistema operativo, donde solo instala las reglas predefinidas con pf como firewall, y no involucra algún número de reglas de filtrado, simplemente mide el rendimiento de la red con esta arquitectura implementada con máquinas virtuales.
En el trabajo "Performance Evaluation and Comparative Analysis of Network Firewalls" publicado en 2011 [6], muestra resultados de rendimiento para 3 tipos de firewall, entre los cuales se encuentran Cisco© y OpenBSD, ambos con PF como software para filtrado, y podemos ver que el firewall de 2024 Cisco© modelo ASA [58] proporciona mejor rendimiento en el punto de control (Checkpoint SPLAT), además proporciona una mejor funcionalidad. Otra observación que reportan es que OpenBSD PF demostró ser la mejor solución de código abierto, porque el costo del punto de control da una mejor gestión del firewall con políticas centralizada y mejor interfaz de usuario que Cisco© PF. Los resultados indicaron que los firewalls que comparan demostraron una detección básica de intrusiones, capacidad y transmisión bloqueada contra ataques típicos.
En el comparativo de este trabajo vemos que la solución OpenBSD resulta, según el comparativo, la más segura que se puede encontrar de código abierto, y representa una solución de bajo costo, y que es de la cual se analiza su troughtput de esta tesis.

El siguiente trabajo que analizamos es "A firewall performance test" [14] del año 2005, concluye la eficiencia de Linux para poder procesar paquetes de manera más eficiente que OpenBSD. Sin embargo, iptables filtra paquetes más rápidamente que PF y se explica por qué sucede esto, sin embargo, el PF, si es capaz de filtrar y evitar ataques de UDP y para este caso suele ser más efectivo, y pierde menos tiempo de procesamiento porque realiza una optimización automática del conjunto de reglas, procesándolo en múltiples listas enlazadas. Esta conclusión es muy importante para esta tesis, debido a que lo que nos interesa saber es la eficiencia en el copiado de información desde y hacia los equipos de alto rendimiento.

En la publicación "Building Firewalls with OpenBSD and PF" del año 2003 [57], su autor Jacek Artymiak, nos explica cómo construir, configurar y administrar firewalls para filtrar direccionamiento IP utilizando hardware básico, el sistema operativo OpenBSD y el software de firewall pf de Daniel Hartmeier. Todo esto es esencial para la instalación de firewalls, aunque han cambiado las versiones de OpenBSD, la mayor parte del material que presenta es esencial para realizar la instalación y configuración con diferentes topologías que usamos en la tesis aquí presentada, nos muestra desde cómo realizar la instalación del Sistema Operativo hasta cómo configurar de manera detallada y con ejemplos el pf.

En la publicación "Installation de pare-feu redondants avec OpenBSD", del 2022, el autor Matthieu Herrb nos presenta la instalación y configuración de un firewall utilizando el filtro de paquetes PF en OpenBSD. Este tutorial a diferencia del libro "Building Firewalls with OpenBSD and PF" nos explica con información actualizada del hardware y versiones de OpenBSD. El tutorial nos actualiza las versiones de OpenBSD, así como de arquitecturas de hardware de la familia Intel/AMD (64 y 32 bits), que es la que utilizamos en esta tesis. También nos explica que cada 6 meses aparece una nueva versión de OpenBSD, nos explica las topologías de red y también el detalle de cómo realizar las configuraciones del pf, del mismo modo nos explica cómo identificar las interfaces de red, las cuales que en nuestro caso representan las tarjetas Ethernet Intel X5x0 10Gb/s que se reconocen en el Sistema Operativo con el prefijo ix y que son parte fundamental de esta tesis. Nos explica las topologías NAT y Bridge y cómo realizar las configuraciones para poder instalar un firewall.

En la publicación “Testing iptables” [16] del 2003, los autores Daniel Hoffman, Durga Prabhakar, Paul Strooper presentan los resultados de pruebas de rendimiento y el retraso en función del tamaño de la base de reglas de iptables. Estas mediciones se realizaron a través de enlaces Ethernet que funcionan a 10, 100 y 1000 Mbps. Aún no realizan pruebas a 10 Gbps, porque la tecnología aún no estaba disponible, sin embargo, las pruebas de rendimiento que realizan a 10 Mbps y aumentan 100 Mbps concluyen que para iptables no hay problemas de rendimiento, pero si demasiada latencia. Para 100 Mbps las tramas son más cortas, pero a 1000 Mbs el retraso es muy grande y parece ser independiente de la longitud de la trama. Esta última parte es algo que nos da información, ya que en esta tesis probamos el rendimiento, y podemos ver que en 10 Gbps y 1,000,000 reglas con iptables, conocemos el troughput y latencia llegamos a tener.

DESARROLLO

Las pruebas por realizar serán con las siguientes tres topologías*:*

1. **Topología tipo transparente o “Bridge”:** Es un firewall perimetral, no filtra los paquetes, dejando pasar todo el tráfico. En este tipo de topología, el firewall funciona como un equipo de comunicación tipo switch que reenvía los datos del emisor al receptor.
2. **Topología tipo “NAT:** Es un firewall que convierte muchas direcciones privadas en una sola pública. Ambas topologías pueden ser utilizadas no solo para un firewall tipo filtrado de paquetes, sino que además permite realizar *port forwarding (reenvío de puertos)* [49], el cual permite desde una dirección IP pública acceder a una dirección privada por medio de algún puerto TCP/UDP. Más adelante describiremos ambas topologías.

Para probar la caracterización del canal de comunicación, se utilizó la topología mostrada en la figura 10. Se puede observar la conexión con un switch 10Gb/s entre uno de los dos equipos para caracterizar el canal de comunicación que hay entre el firewall OpenBSD y CentOS.

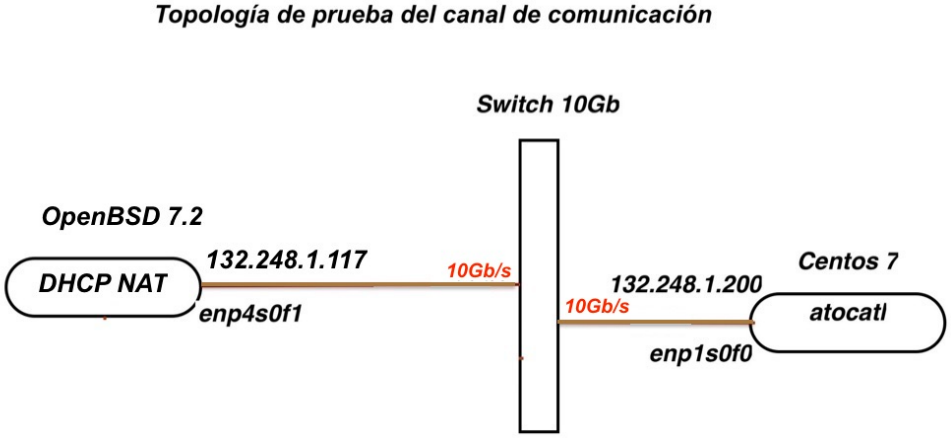


Figura 10. Topología para la caracterización del canal 10Gb/s.

Al caracterizar el canal, se espera que el throughput sea cercano al reportado por el ancho de banda del canal de 10Gb/s, que es el reportado por los fabricantes de los equipos y tarjetas de comunicación. En la Figura 13 de lado izquierdo se tiene un equipo OpenBSD 7 como cliente y el nombre de la interfaz de red detectada por el Sistema Operativo, en la parte de en medio se tiene el switch de comunicación Ethernet en cobre el cual proporciona un ancho de banda de 10Gb/s, y en el lado izquierdo se tiene el equipo servidor Atocatl, el cual también tiene una interfaz de red de 10Gb/s. Los equipos son conectados con un cable Ethernet UTP [47] con cable de cobre categoría

6, que es el utilizado para anchos de banda de 10Gb/s. El direccionamiento IP corresponde al mismo segmento, y ambos equipos no tienen reglas de filtrado en su Sistema Operativo para esta prueba.

Topología de red tipo transparente

En la figura 11 se muestra la topología transparente utilizada para el firewall con Sistema Operativo OpenBSD, el cual se pondrá en medio de las comunicaciones entrando por la interfaz de red detectada como ix0 y sale por la interfaz ix1, usando el mismo segmento de red de configuración de direccionamiento IP, el cual permitirá caracterizar las comunicaciones cuando los mensajes pasan por el firewall.
Como se observa en la figura 11, el firewall con OpenBSD se encuentra entre el switch y el equipo que llamamos Cefalopodo.

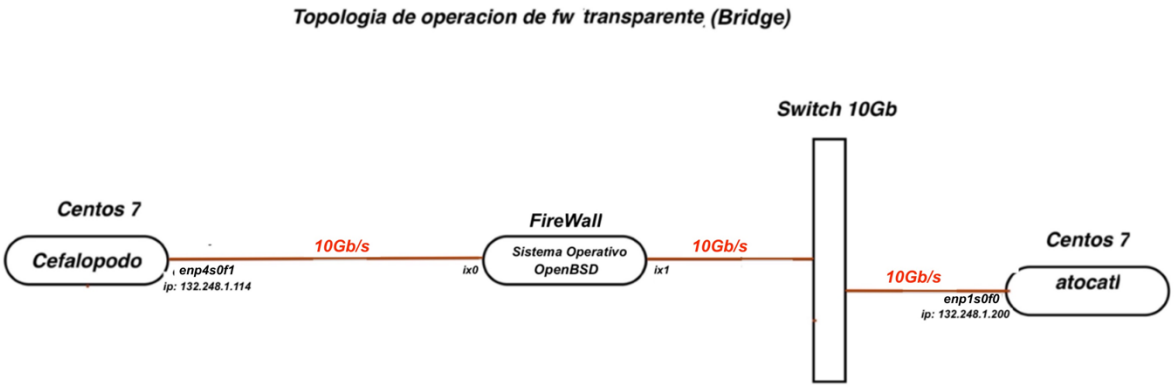


Figura 11. Topología para la caracterización con un firewall OpenBSD filtrando los datos entre los dos equipos por medio de Packet Filter entre Cefalopodo y Atocatl.

Topología de red tipo NAT

En la figura 12 se muestra la topología tipo NAT en donde el equipo Cefalópodo de lado izquierdo con dirección ocupa la dirección IP 10.0.0.100 y llega por el canal de comunicación 10Gb/s en cobre directo a la interfaz.

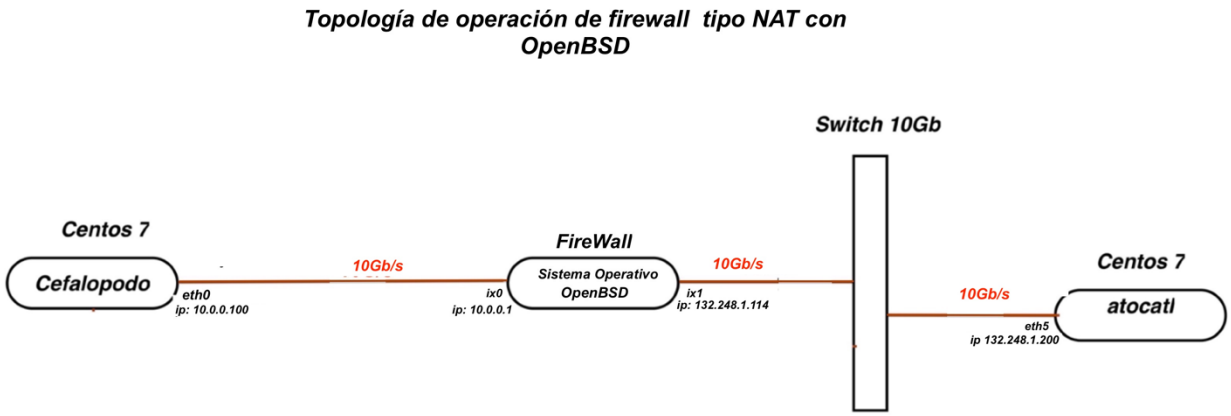


Figura 12. Topología para la caracterización con un firewall OpenBSD filtrando los datos por medio de Packet Filter en modo NAT entre los dos equipos llamados Cefalopodo y Atocatl.

En el centro de la figura 12 se ve el equipo instalado como firewall OpenBSD, al cual se le aplicarán de 0 a 5000 reglas y en cada 100 reglas se verificará el throughput, con la metodología descrita anteriormente.

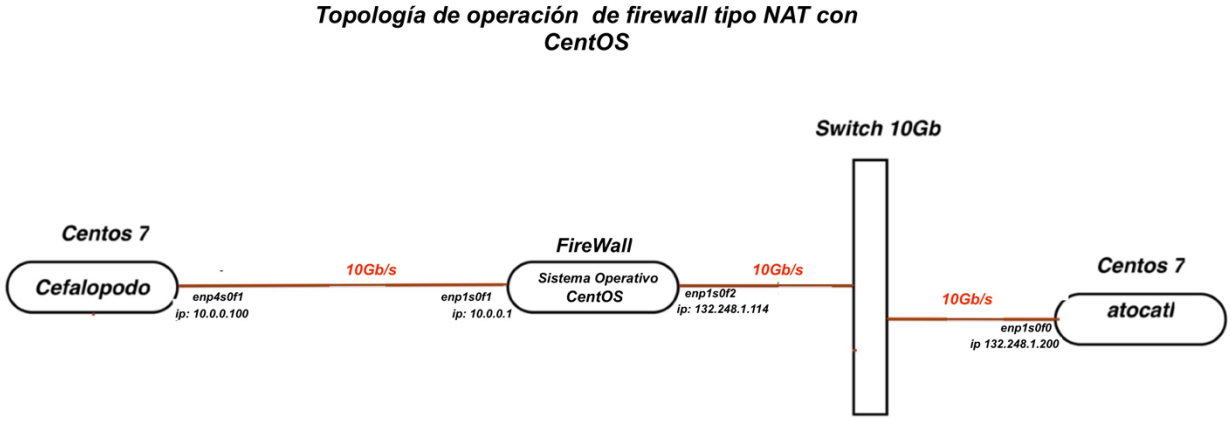


Figura 13. Topología para la caracterización con un firewall OpenBSD filtrando los datos por medio de Packet Filter en modo NAT entre los dos equipos llamados Cefalopodo y Atocatl.

En la figura 13 mostramos la topología del firewall con CentOS e Iptables como sistema de filtrado de firewall. La forma de configurar los firewalls con las diferentes topologías se está descrita en el anexo 1 de este trabajo.

Resultados

En esta sección se presentarán los resultados y se evalúa el desempeño tanto en función del throughput como de latencia, del mismo modo se calcula el Jitter para observar a más detalle el tiempo de respuesta. Se realizaron 170 pruebas, descritas en esta sección.

CentOS 7 sin firewall

La primera prueba realizada es la caracterización del canal de comunicación, con las tarjetas de red a 10Gb/s. Esta prueba hace uso de iperf3 para caracterizar el throughput. En la figura 10 de la metodología se describe con detalle la topología utilizada para esta prueba.

No. De Prueba	Gb/s
1	9.18
2	8.52
3	8.89
4	8.47
5	9.29
6	9.22
7	9.36
8	9.23
9	8.92
10	9.21
11	9.23
12	9.38
13	9.37
14	9.38
15	9.36
16	9.32
17	8.95
18	9.38
19	9.34
20	9.36

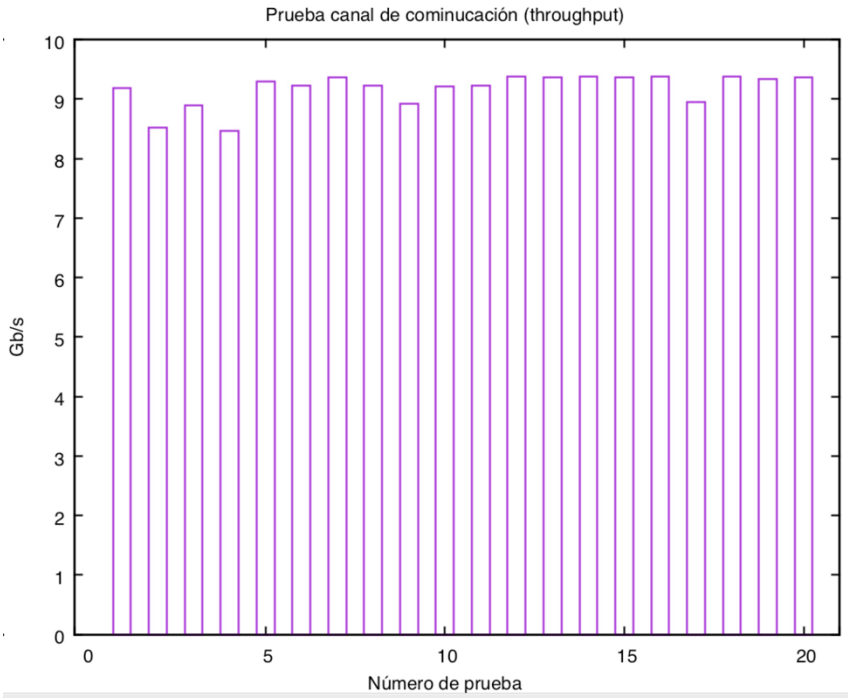


Figura 16. Throughput del canal de comunicación en CentOS.

La figura 16 muestra el resultado de la ejecución de iperf3 en veinte pruebas. Se observa que el throughput para el canal oscila entre los 8.47 Gb/s y 9.38Gb/s con una media promedio de 9.267Gb/s. El intervalo de confianza es de 0.121Gb/s. Los resultados tienen variaciones debido a condiciones instantáneas del canal, pero, al realizar varias mediciones, se obtiene una media bastante estable. De esta prueba se obtiene que, dada la topología probada, en promedio se aprovecha aproximadamente el 92.6% de la velocidad de la reportada por el fabricante de los equipos de comunicación.

Después, se llevaron a cabo las pruebas para medir la latencia con la herramienta ping. La figura 17 muestra los resultados de las pruebas realizadas.

No. de prueba	milisegundos
1	0.851
2	0.815
3	0.872
4	0.902
5	0.906
6	0.868
7	0.632
8	0.578
9	0.867
10	0.886
11	0.955
12	0.862
13	0.926
14	0.857
15	0.835
16	0.982
17	0.783
18	0.901
19	0.931
20	0.903

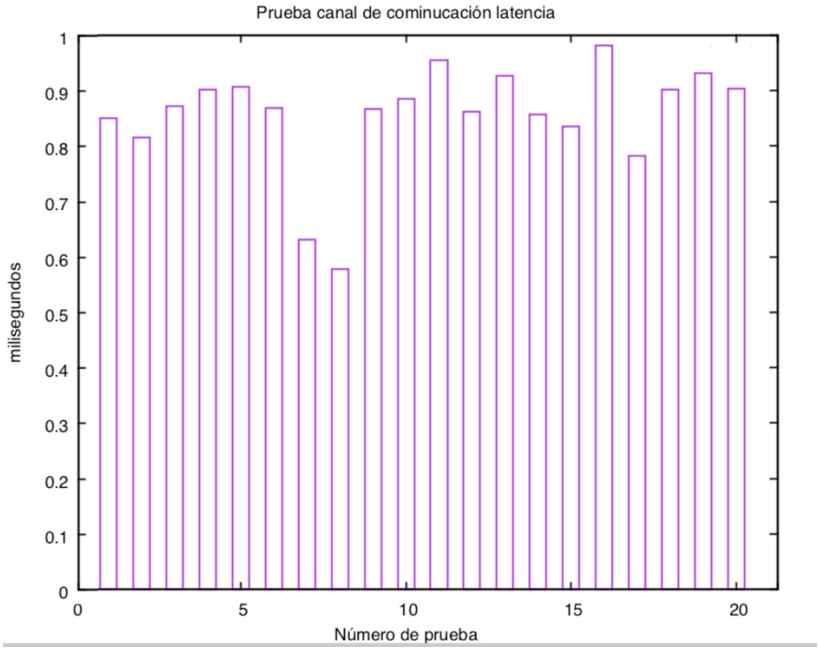


Figura 17. Latencia del canal de comunicación utilizando CentOS.

De la figura 17, se observan valores que van desde los 0.578 hasta los 0.982 milisegundos con una latencia promedio de 0.855 ms (milisegundos). El intervalo de confianza para esta prueba es de 0.0428 ms. El Jitter observado de las mediciones es de 94.62ms, debido en gran parte a los dos valores de las pruebas 7 y 8 en los que se observaron cambios muy grandes en las mediciones.
Las siguientes pruebas se realizarán ya con un firewall de por medio en topologías NAT y transparente (Bridge), las cuales permitirán conocer el desempeño de los firewalls implementados con OpenBSD y CentOS hacia el servidor Atocatl..

Topología transparente con OpenBSD 7.2 como firewall

La segunda prueba realizada es colocar el firewall perimetral entre el equipo Cefalopodo y Atocatl, conectados mediante un switch de comunicación, usando las mismas tarjetas de red a 10Gb/s, al igual que en la prueba del canal de comunicación. Esta prueba también hace uso de iperf3 para caracterizar el throughput. En la figura 11 de la metodología se describe con detalle la topología utilizada para esta prueba. La tabla 6 muestra los resultados de las 50 pruebas que se realizaron variando la cantidad de reglas. Es importante mencionar que para cada cantidad de reglas se realizaron 20 experimentos y la figura muestra los valores promedio, así como los intervalos de confianza al 95%.

Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]
0	2.119	**1100**	1.62	**2200**	1.5915	**3100**	1.558	**4200**	1.561
100	1.581	**1200**	1.6195	**2300**	1.566	**3200**	1.585	**4300**	1.574
200	1.622	**1300**	1.598	**2400**	1.558	**3300**	1.603	**4400**	1.5805
300	1.635	**1400**	1.5865	**2500**	1.614	**3400**	1.596	**4500**	1.5985
400	1.606	**1500**	1.6235	**2600**	1.575	**3500**	1.572	**4600**	1.591
500	1.6135	**1600**	1.589	**2700**	1.5855	**3600**	1.598	**4700**	1.5805
600	1.581	**1700**	1.5835	**2800**	1.6175	**3700**	1.557	**4800**	1.596
700	1.614	**1800**	1.588	**2900**	1.578	**3800**	1.596	**4900**	1.539
800	1.608	**1900**	1.57	**3000**	1.5755	**3900**	1.576	**5000**	1.549
900	1.5975	**2000**	1.6055	**3100**	1.558	**4000**	1.575		
1000	1.532	**2100**	1.5945	**3200**	1.585	**4100**	1.573		

Tabla 6. Throughput de 0 a 5000 reglas con firewall OpenBSD utilizando Packet Filter y topología transparente.

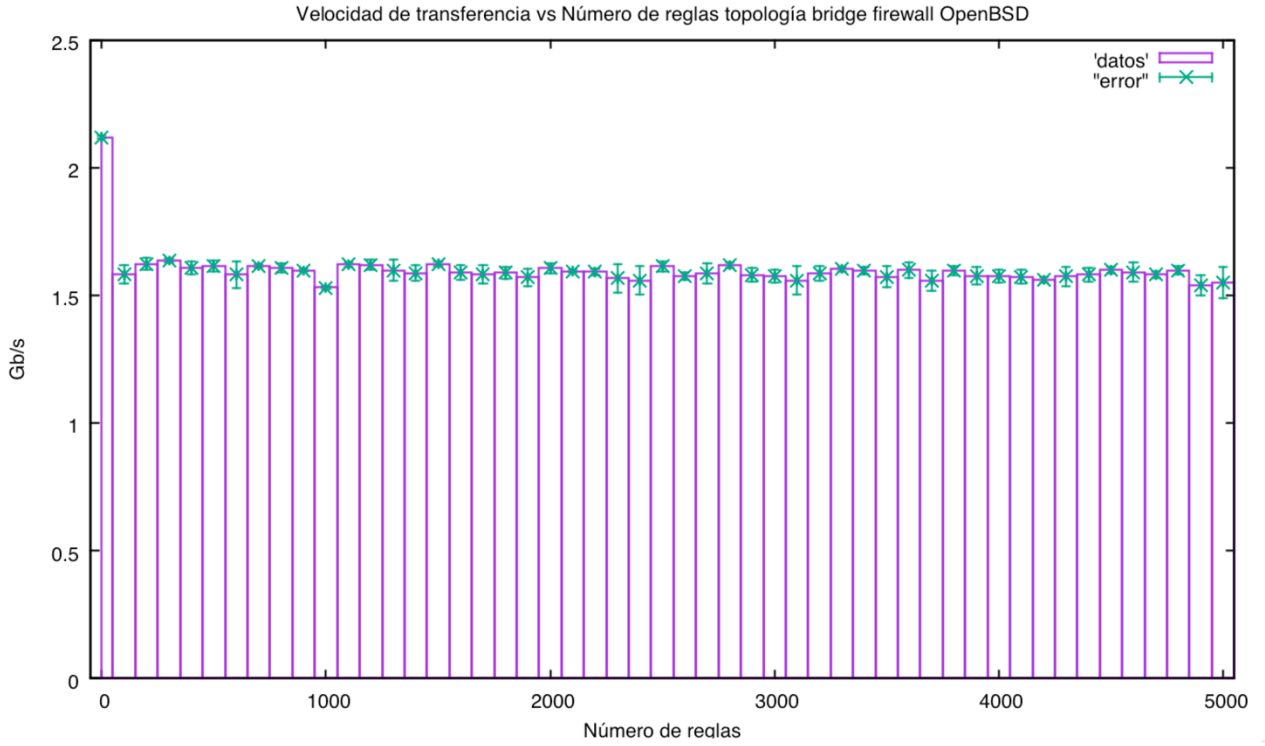


Figura 18. Throughput de 0 a 5000 reglas con firewall OpenBSD utilizando Packet Filter y topología transparente.

En la tabla 6 y figura 18 se muestran los resultados del throughput en topología transparente con OpenBSD como firewall. Se observa que el throughput está alrededor de 1.58 Gb/s, este valor se mantiene muy similar hasta las 5000 reglas. Las variaciones que se encontraron en las diferentes pruebas son muy pequeñas, lo que sugiere que el throughput tiene un comportamiento casi constante. Sin embargo, el valor que corresponde a cero reglas es de 2.119 Gb/s, esto se debe a que, al no existir

ninguna regla de filtrado, el throughput, no se requiere revisar cada paquete y esto se realiza más rápido.

Reglas	Retardo [ms]	Reglas	Retardo [ms]	Reglas	Retardo [ms]	Reglas	Retardo [ms]	Reglas	Retardo [ms]
0	3.7692	1100	4.67315	2100	4.36455	3100	4.7264	4100	5.69245
100	4.61565	1200	4.4082	2200	4.42715	3200	4.51995	4200	5.3368
200	4.6086	1300	4.56655	2300	4.7035	3300	4.6177	4300	4.74265
300	4.6175	1400	4.541	2400	4.914	3400	4.74135	4400	4.62515
400	4.88135	1500	4.8929	2500	4.73365	3500	4.3386	4500	4.59905
500	4.82085	1600	4.872	2600	4.8192	3600	4.5881	4600	4.51575
600	4.96815	1700	4.80225	2700	4.75915	3700	4.40075	4700	4.25615
700	4.29355	1800	4.34205	2800	#DIV/0!	3800	4.7028	4800	4.25615
800	4.79355	1900	4.43825	2900	4.81275	3900	5.1159	4900	5.72585
900	4.33255	2000	4.36455	3000	4.5112	4000	4.66135	5000	4.7324
1000	4.61565								

Tabla 7. Latencia de 0 a 5000 reglas con firewall OpenBSD utilizando Packet Filter y topología transparente.

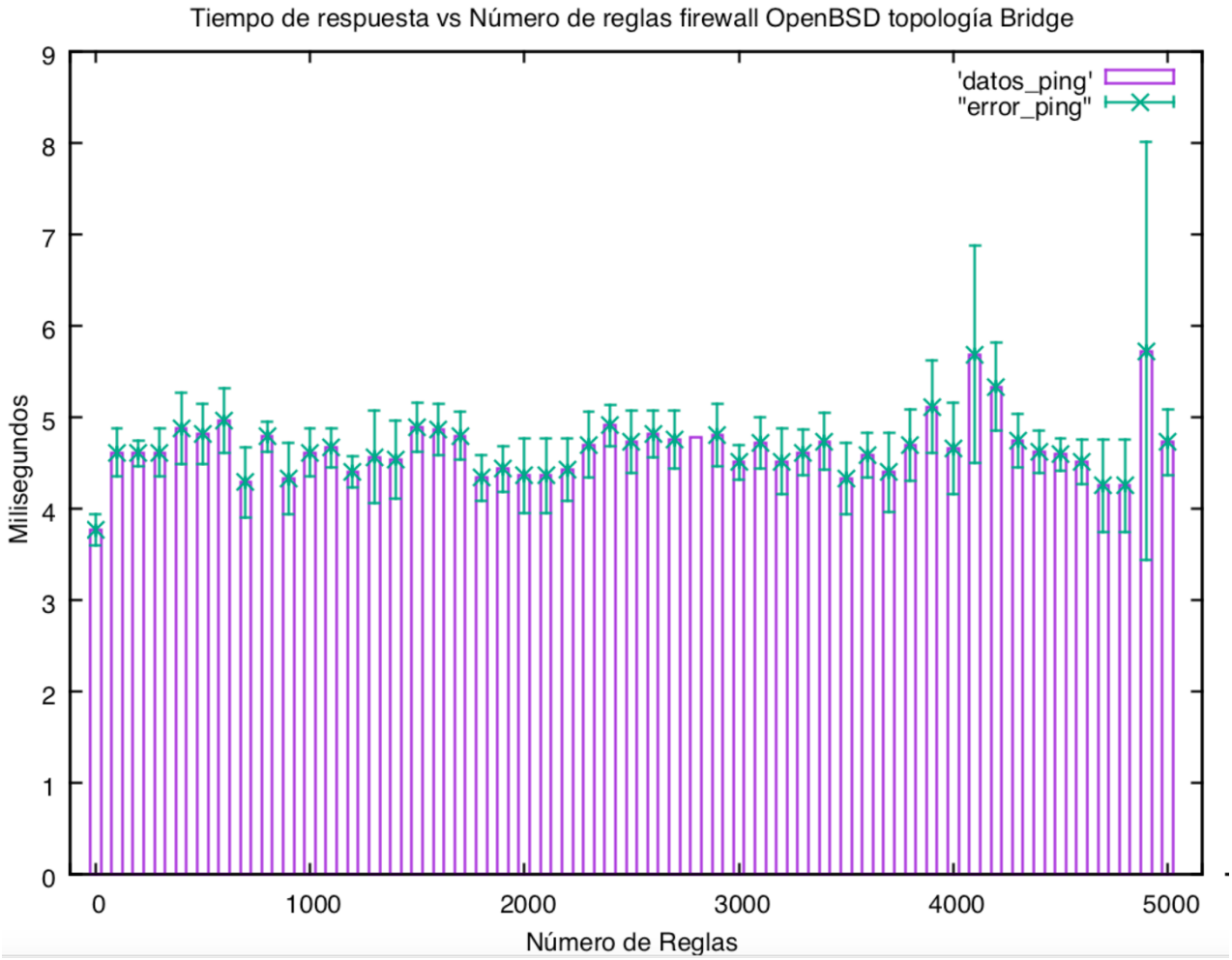


Figura 19. Latencia de 0 a 5000 reglas con firewall OpenBSD utilizando Packet Filter y topología transparente.

En la tabla 7 y figura 19 se muestran los resultados de las de latencia, con un promedio de 4.65ms, con un intervalo de confianza promedio de 0.431ms, excepto un par de valores cercanos a las cinco mil reglas, en donde el intervalo de confianza es muy grande, por lo cual esas mediciones no sean confiables. Para el caso en 0 reglas, tenemos una latencia de 3.7692ms, por lo que suponemos que, al no existir reglas de filtrado, no se revisan los paquetes, dando como resultado, una latencia muy baja, comparada con el resto de las mediciones.

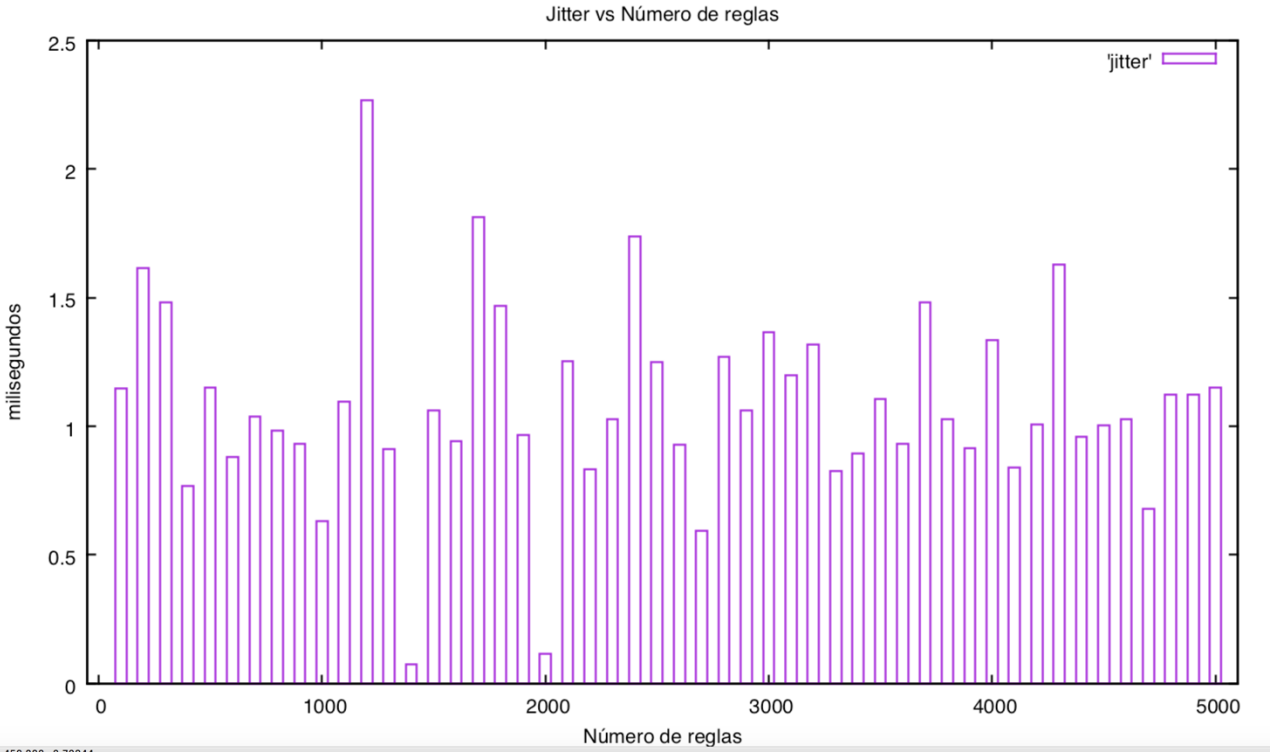


Figura 20. Jitter de 0 a 5000 reglas con firewall OpenBSD utilizando Packet Filter y topología transparente.

En la figura 20 se muestra la variación del tiempo de respuesta conocida como Jitter [30], este cálculo de medición basado en la latencia permite ver la diferencia en variaciones que juega un papel importante en algunas aplicaciones. Los valores que presentan un Jitter muy bajo (de menos de 0.5ms) son un par que se encuentran en las 4700 y 4800 reglas. En general, el Jitter va desde los 0.75ms hasta los 1.75ms con una diferencia entre muy pequeña y generalmente menor a 1ms para las reglas probadas. En promedio, se tiene un Jitter de 1.2ms.

Topología NAT con firewall OpenBSD aplicadas con Packet Filter.

En la tercera prueba realizada, aplicamos la topología de la figura 13 de la metodología, pero en este caso utilizaremos CentOS como firewall. Del mismo modo, en esta prueba usaremos el iperf3 para caracterizar el throughput.

Número de regla	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]
100	**2.6385**	**1100**	**2.6135**	**2100**	**2.6115**	**3100**	2.6425	**4100**	2.617
200	2.6345	**1200**	2.628	**2200**	2.6125	**3200**	2.632	**4200**	2.631
300	2.624	**1300**	2.6025	**2300**	2.6245	**3300**	2.6375	**4300**	2.6315
400	2.609	**1400**	2.627	**2400**	2.632	**3400**	2.63	**4400**	2.6285
500	2.6385	**1500**	2.631	**2500**	2.6295	**3500**	2.624	**4500**	2.6275
600	2.6205	**1600**	2.6245	**2600**	2.626	**3600**	2.608	**4600**	2.62
700	2.637	**1700**	2.63	**2700**	2.637	**3700**	2.6395	**4700**	2.633
800	2.61	**1800**	2.616	**2800**	2.6315	**3800**	2.618	**4800**	2.6225
900	2.6195	**1900**	2.604	**2900**	2.64	**3900**	2.6325	**4900**	2.6225
1000	2.6415	**2000**	2.614	**3000**	2.6425	**4000**	2.62	**5000**	2.457

Tabla 8. Throughput, 0 a 5000 reglas para firewall OpenBSD utilizando Packet Filter con topología NAT.

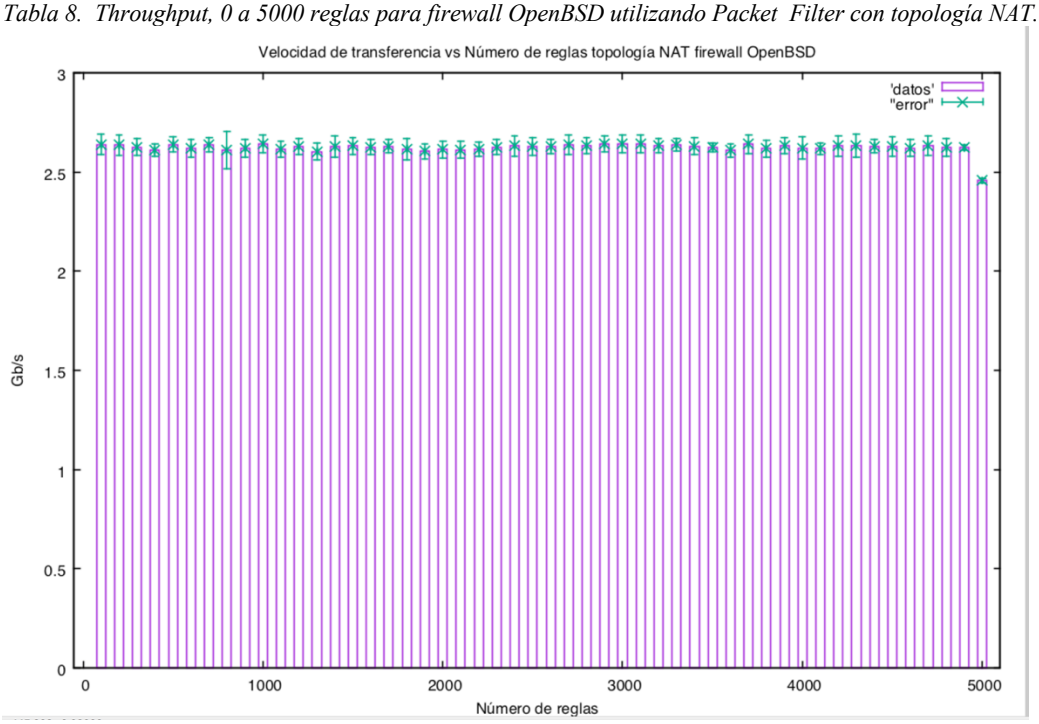


Figura 21. Throughput aplicando 0 a 5000 reglas para firewall OpenBSD utilizando Packet Filter con topología NAT.

En la tabla 8 y figura 21, se muestran los resultados del throughput en topología NAT con OpenBSD como firewall, se observa que el throughput es en promedio 2.457Gb/s, este valor se mantiene prácticamente constante hasta las 5000 reglas. No existen variaciones significativas, excepto en la última medición encontramos un valor menor a 2.5Gb/s, sin embargo, es la única medición fuera del intervalo.

Número de Reglas	Retardo [ms]		Reglas	Retardo [ms]		Reglas	Retardo [ms]		Reglas	Retardo [ms]		Reglas	Retardo [ms]
100	5.0085		**1100**	4.3304		**2100**	4.40645		**3100**	4.945		**4100**	4.208
200	5.0891		**1200**	4.5846		**2200**	4.08315		**3200**	4.5731		**4200**	4.426
300	4.596		**1300**	4.2244		**2300**	4.3813		**3300**	4.7045		**4300**	4.624
400	4.4779		**1400**	3.835		**2400**	4.4908		**3400**	4.2587		**4400**	4.280
500	4.4170		**1500**	4.3018		**2500**	4.26585		**3500**	4.3769		**4500**	4.236
600	4.1814		**1600**	4.5796		**2600**	4.2705		**3600**	4.5836		**4600**	4.270
700	4.404		**1700**	4.3446		**2700**	4.57635		**3700**	4.3676		**4700**	4.339
800	4.6176		**1800**	4.3023		**2800**	4.7919		**3800**	4.3883		**4800**	4.799
900	4.6361		**1900**	4.1661		**2900**	4.8687		**3900**	4.3371		**4900**	4.799
1000	4.6914		**2000**	4.5686		**3000**	4.7463		**4000**	4.392		**5000**	4.617

Tabla 9. Latencia aplicando de 0 a 5000 reglas para firewall OpenBSD utilizando Packet Filter con topología NAT.

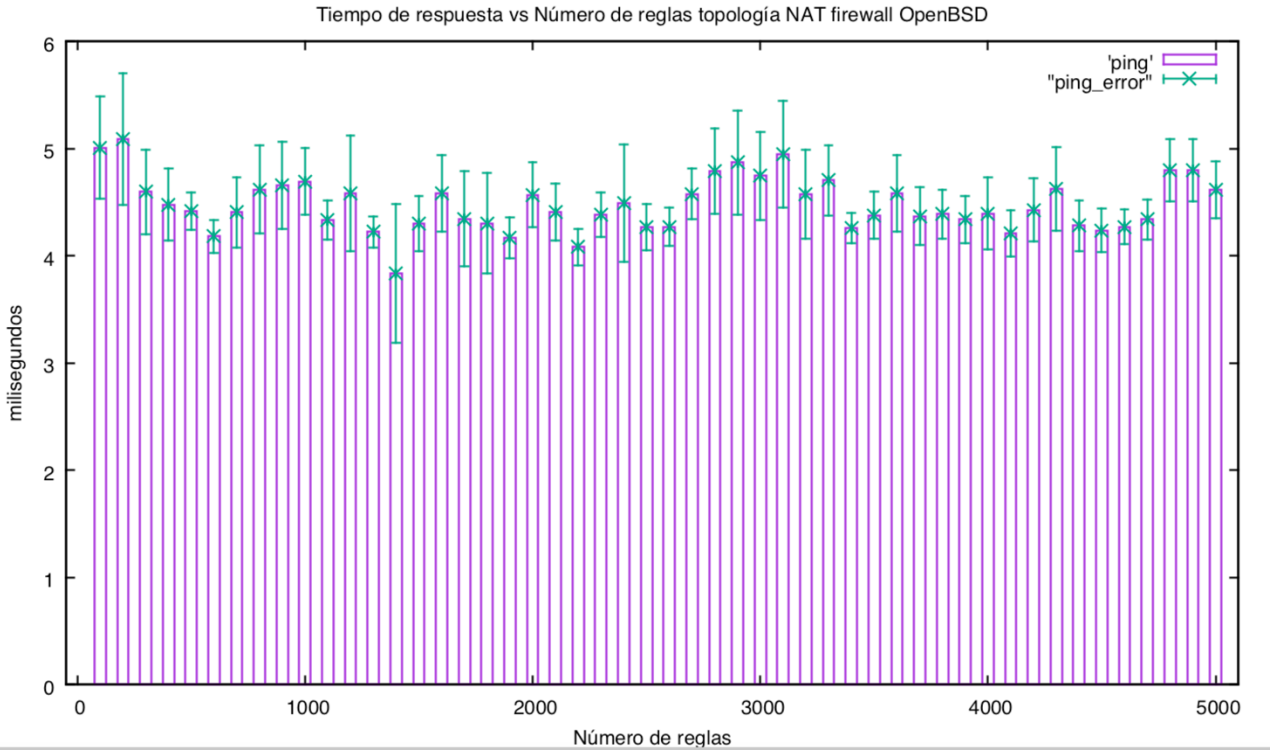


Figura 22. Latencia aplicando de 0 a 5000 reglas para firewall OpenBSD utilizando Packet Filter con topología NAT.

En la tabla 9 y figura 22 se muestran los resultados de la de latencia, con un promedio de 4.799ms, un intervalo de confianza promedio de 0.315ms, hay mucha oscilación en los valores que van desde 3.83ms hasta 5.08ms.

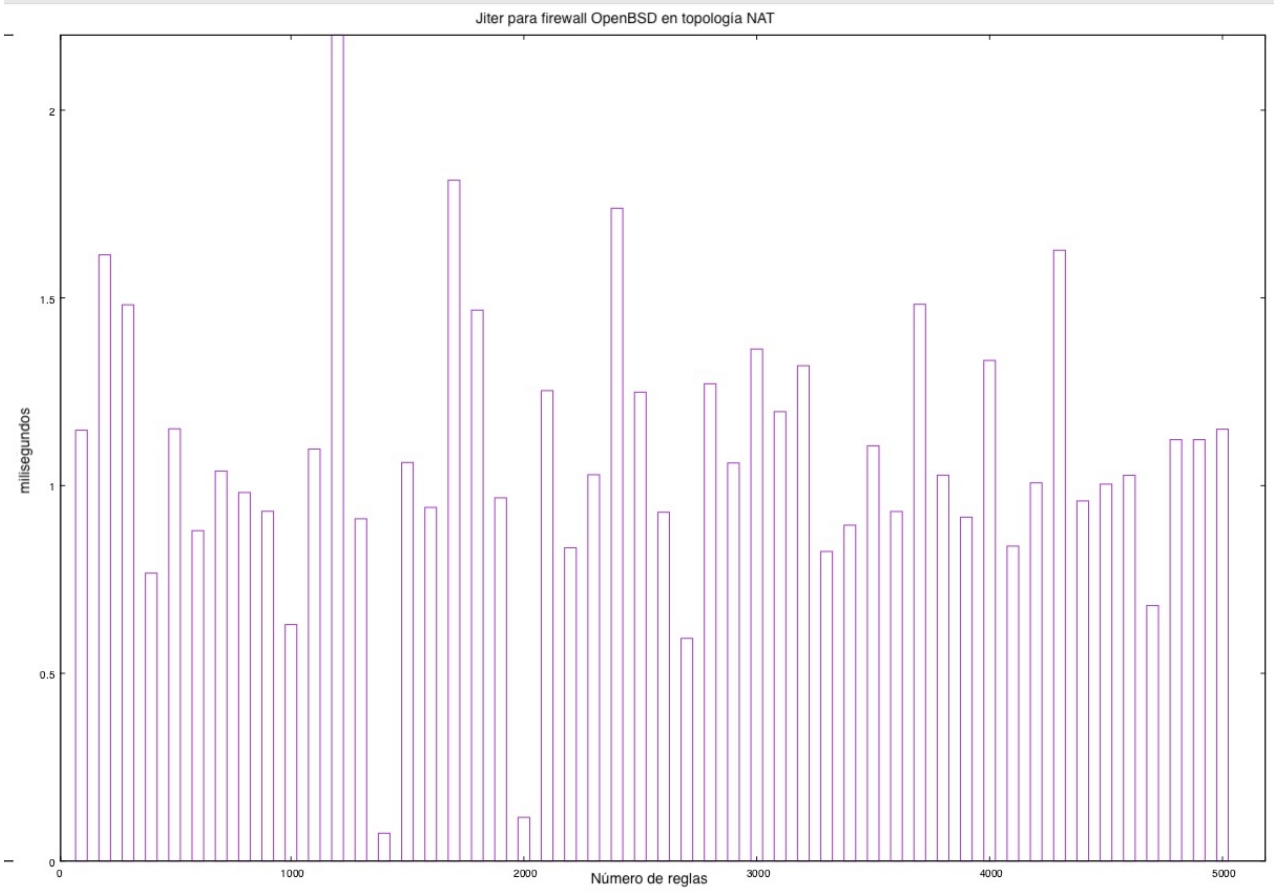


Figura 23. Jitter de 0 a 5000 reglas para firewall OpenBSD utilizando Packet Filter con topología NAT.

En la figura 23 muestra el Jitter para este experimento. En este caso podemos observar que, de igual forma, la oscilación de los valores continúa, resaltando los valores 1400 y 2000 reglas en donde el Jiterr es cercano a 0.1ms. En su mayoría, el Jitter va desde los 0.6ms hasta los 1.6ms, mostrando variaciones similares a las presentadas en la prueba anterior.

Topología NAT con firewall CentOS 7.2

En la cuarta prueba utilizamos la topología NAT con CentOS como firewall como se muestra en la figura 13 de la metodología. En esta prueba se coloca el firewall perimetral entre el equipo Cefalopodo y Atocatl, conectados mediante un switch de comunicación y utilizando las mismas tarjetas de red a 10Gb/s. Del mismo modo, esta prueba hace uso de iperf3 para caracterizar el throughput. La tabla 10 muestra los resultados de las 50 pruebas con diferentes cantidades de reglas.

Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]
100	8.882	**1100**	8.8795	**2100**	8.84	**3100**	8.842	**4100**	8.717
200	8.9285	**1200**	8.539	**2200**	8.789	**3200**	8.892	**4200**	8.944
300	9.3942	**1300**	8.724	**2300**	8.792	**3300**	8.885	**4300**	8.867
400	8.9195	**1400**	8.525	**2400**	8.920	**3400**	8.9185	**4400**	8.9755
500	8.906	**1500**	8.823	**2500**	8.93	**3500**	8.934	**4500**	8.957
600	8.9075	**1600**	8.609	**2600**	8.914	**3600**	8.9455	**4600**	8.9815
700	8.902	**1700**	8.648	**2700**	8.9	**3700**	8.407	**4700**	8.911
800	8.9345	**1800**	8.5115	**2800**	8.954	**3800**	8.9275	**4800**	8.96062
900	8.9385	**1900**	8.1095	**2900**	9.41	**3900**	8.9355	**4900**	8.94526
1000	8.873	**2000**	8.935	**3000**	8.876	**4000**	8.945	**5000**	8.846

Tabla 10. Throughput, desde 0 hasta 5000 reglas aplicadas en el firewall CentOS 7.2 utilizando IPTables en topología NAT.

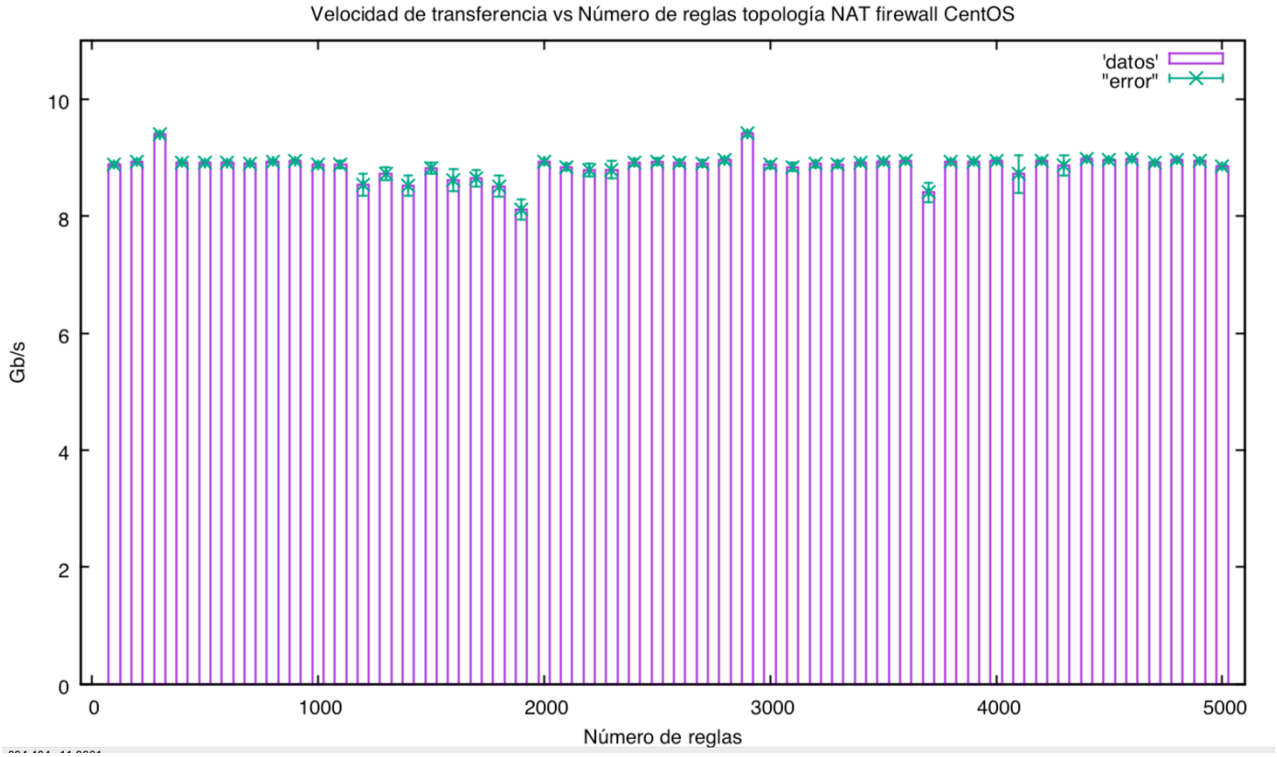


Figura 24. Throughput aplicando de 0 a 5000 reglas del firewall CentOS 7.2 utilizando IPTables con topología NAT.

En la tabla 10 y figura 24, se muestran los resultados del throughput en topología NAT con CentOS como firewall. Se observa que el throughput en promedio es de 8.84Gb/s, y se tiene un promedio de intervalo de confianza de 0.0702, nuevamente, estos valores se mantienen muy similares hasta las 5000 reglas. Podemos observar que el intervalo de confianza es cercano a 0.1Gb/s, lo que sugiere una buena estabilidad cuando se utiliza CentOS.

Número de Reglas	Retardo [ms]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]	Número de reglas	Throughput [Gb/s]
100	0.5375	**1100**	0.47395	**2100**	0.50515	**3100**	0.51695	**4100**	0.4483
200	0.52155	**1200**	0.39605	**2200**	0.50695	**3200**	0.50985	**4200**	0.4623
300	0.51375	**1300**	0.426	**2300**	0.51325	**3300**	0.51385	**4300**	0.4538
400	0.52515	**1400**	0.4064	**2400**	0.47505	**3400**	0.5373	**4400**	0.5424
500	0.52405	**1500**	0.48315	**2500**	0.5156	**3500**	0.47805	**4500**	0.53895
600	0.51835	**1600**	0.4806	**2600**	0.5121	**3600**	0.46975	**4600**	0.5689
700	0.5196	**1700**	0.47345	**2700**	0.49775	**3700**	0.4828	**4700**	0.4946
800	0.52965	**1800**	0.46245	**2800**	0.5226	**3800**	0.5313	**4800**	0.5412
900	0.4973	**1900**	0.38105	**2900**	0.50525	**3900**	0.4968	**4900**	0.5175
1000	0.48015	**2000**	0.5253	**3000**	0.52725	**4000**	0.4961	**5000**	0.5586

Tabla 11. Latencia desde 0 a 5000 reglas para firewall CentOS con IPtables con topología NAT.

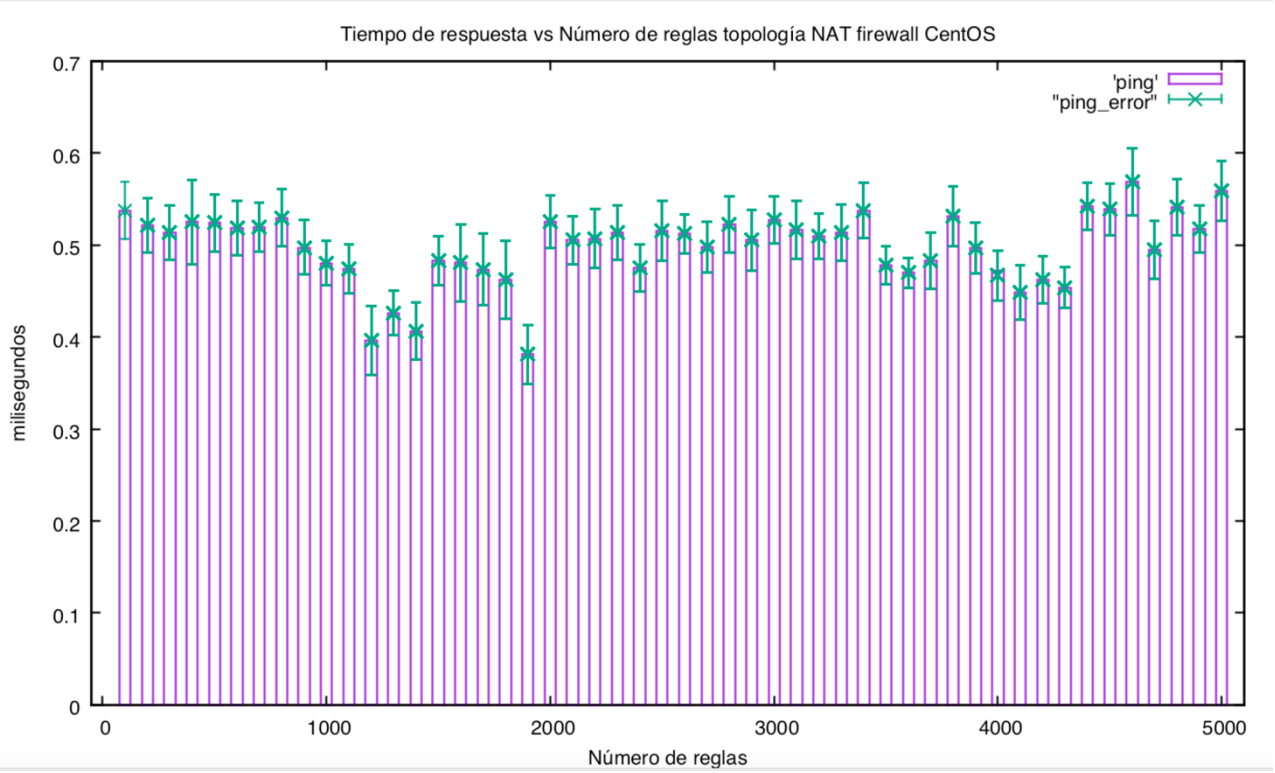


Figura 25. Latencia desde 0 a 5000 reglas para firewall CentOS con IPtables con topología NAT.

En la tabla 11 y figura 25 se muestran los resultados de las de latencia, con un promedio de 0.4983ms. como tiempo de respuesta, con un intervalo de confianza promedio de 0.0295ms. En esta prueba se observa una oscilación donde los datos que más resaltan son de 0.39605ms. para 1200 reglas y de 0.38105ms. a 1900 reglas.

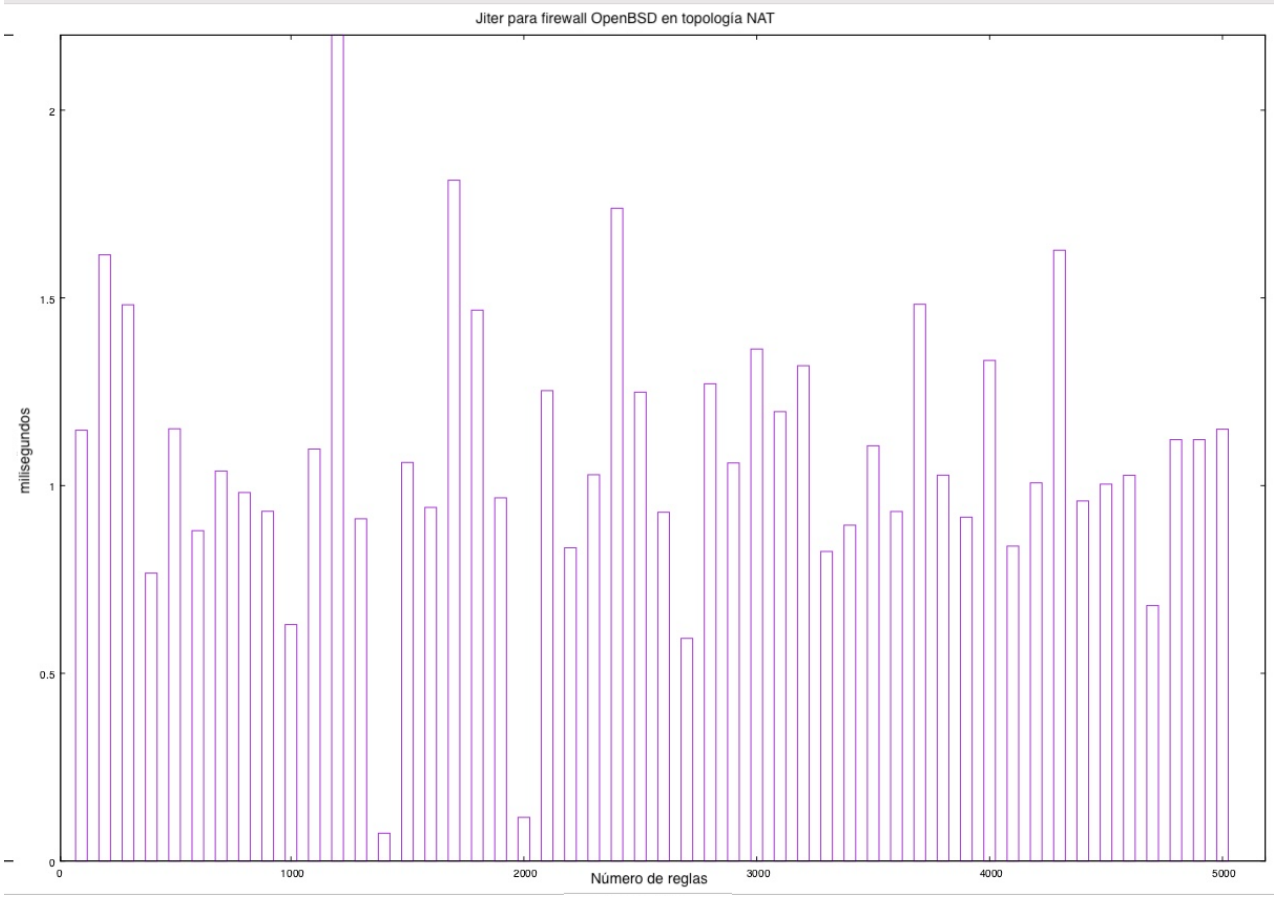


Figura 26. Jitter de 0 a 5000 reglas del firewall CentOS con IPtables en topología NAT.

En la figura 26 se observa el Jitter para la prueba actual. Los resultados muestran mucha oscilación en los valores. Resaltan un par de valores: en 1200 reglas, donde está el valor más alto de todo el experimento, y en 1400 reglas, donde se aprecia el valor más pequeño de todas las mediciones cercano a 0.1ms.

Para el caso del CentOS, y revisando todas pruebas realizadas, no se observa ningún cambio de 0 a 5,000 reglas, por lo cual, usando la misma topología y para ver sí en número de reglas afecta la medición del throughput, se trató de realizar pruebas con más de 100k reglas. Se cargaron hasta un millón de reglas de filtrado, pero en OpenBSD no fue posible, ya que no se pudieron cargar más de 15000 reglas. Además, para cargar 15000 reglas, el tiempo de carga de estas era de aproximadamente 35 minutos, y después de 15000 reglas OpenBSD comenzaba a enviar mensajes de “Kernel Panic” por lo cual ya era necesario reiniciar el equipo.

Una vez habilitadas las 100k reglas en CentOS, se realizaron 10 experimentos desde 100k reglas y hasta 1M con saltos de 100k reglas para verificar si más reglas afectaban el rendimiento de la red. Las figuras 27 y 28 muestran los resultados que muestran cómo se afecta el throughput y la latencia en las condiciones mencionadas.

Prueba con firewall CentOS en intervalos de 100,000 hasta llegar a 1,000,000 de reglas.

Número de reglas	Throughput [Gb/s]
100	8.9945
100k	8.603
200k	8.3629
300k	8.1223
400k	7.774
500k	7.4538
600k	6.2852
700k	6.1805
800k	6.6486
900k	4.9638
1M	5.0385

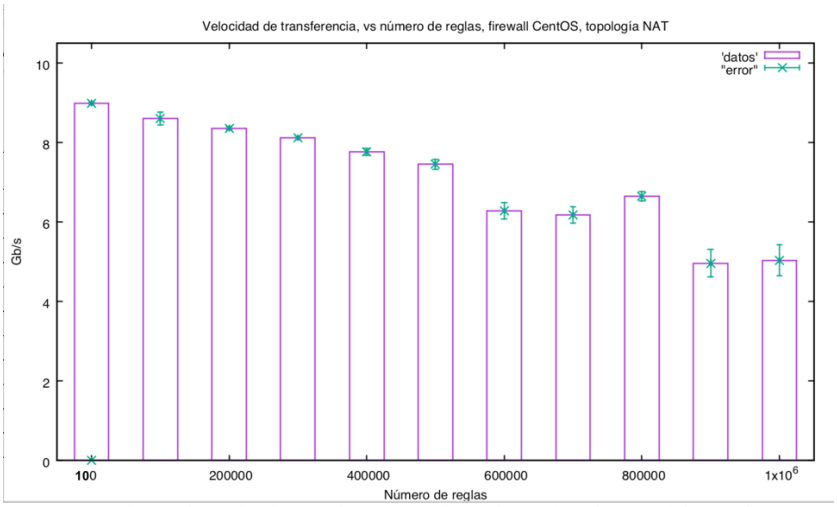


Figura 27. Throughput desde 100 hasta 1M de reglas aplicadas en el firewall CentOS con IPTables en topología NAT.

En la figura 27 y sus datos que aparecen al lado izquierdo se observa como al aplicar las primeras 100 reglas y tomándolas como valor base, el throughput comienza en 8.99 Gb/s, y disminuye hasta 5.03Gb/s cuando se tienen 1M de reglas. Se puede observar un decremento casi lineal en el throughput ocasionado por la cantidad de reglas que se tengan. Vale la pena mencionar que a pesar de que el throughput si se ve afectado, son pocos los sistemas que requerirían una cantidad de reglas tan grande.

Reglas	Retardo [ms]
100	0.513
100k	2.129
200k	3.987
300k	2.091
400k	12.325
500k	30.386
600k	30.381
700k	23.495
800k	19.856
900k	31.405
1M	159.208

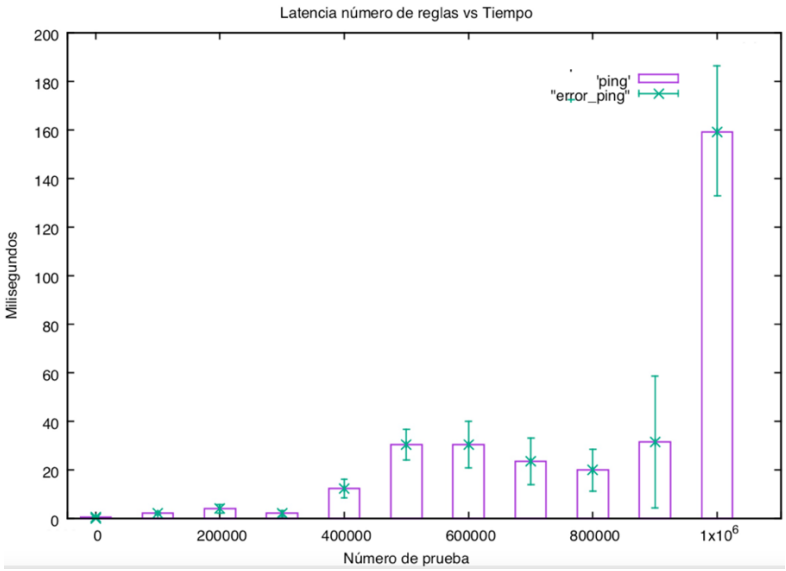


Figura 28. Latencia desde 100 hasta 1M de reglas aplicadas en firewall CentOS con IPTables con topología NAT.

Para el caso de la latencia podemos observar en la figura 28 y los datos que están a lado izquierdo que el tiempo de respuesta para 100 reglas corresponde a 0.513ms. Este tiempo va aumentando con una tendencia exponencial y cuando llegamos a 1M de reglas la latencia es de 159.208ms, lo cual multiplica drásticamente el retardo de la información.
Podemos observar que, para el caso de 900k, el intervalo de confianza es casi igual que la medida, lo que indica que puede que el valor real esté por encima del encontrado, siguiendo el comportamiento exponencial propuesto.

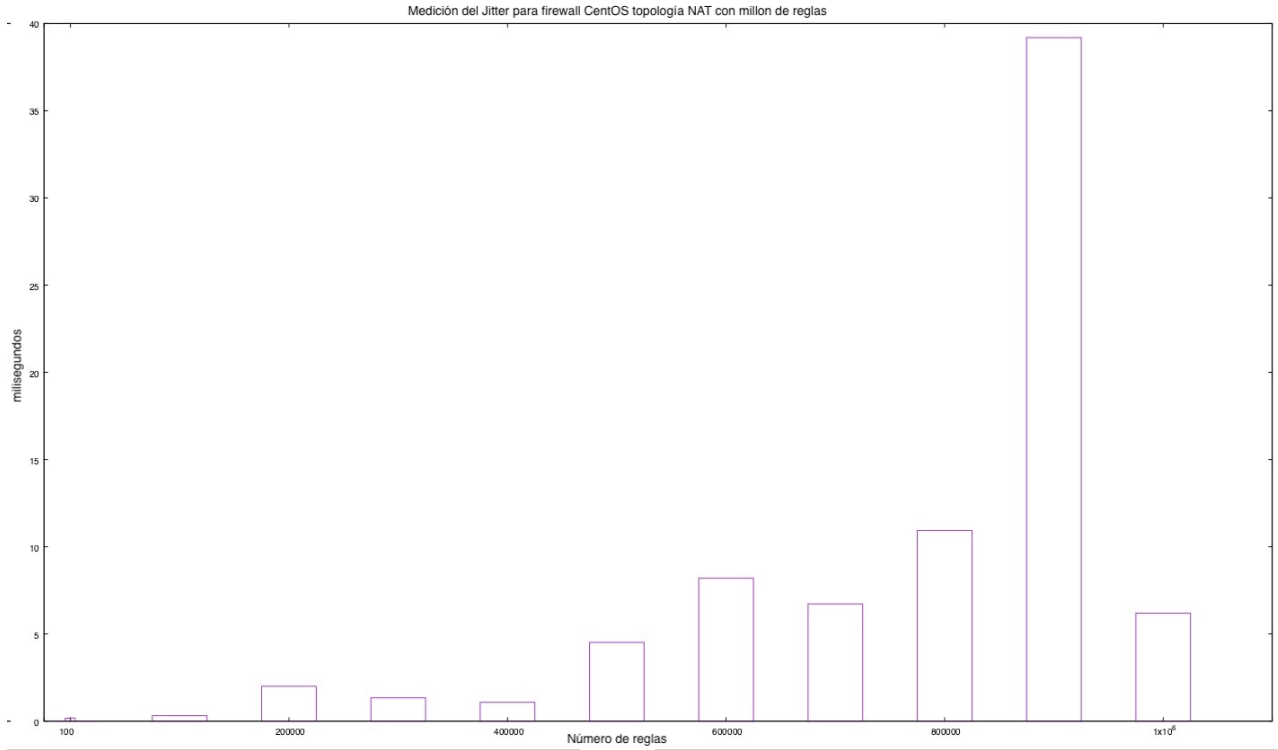


Figura 29. Jitter desde 1000 a 1M de reglas del firewall OpenBSD aplicadas con topología NAT.

En la figura 29 se muestra el Jitter. Se observa que esta métrica también se afecta por la cantidad de reglas y aumenta según el aumento en la cantidad de reglas. La figura muestra que el valor correspondiente a 900M reglas (el cual está muy desproporcionado al resto de los demás valores) lo cual nos confirma que la media de la latencia podría no ser la encontrada.

Análisis de Resultados

Para el caso de CentOS con iptables, se puede observar que al llegar a 1M de reglas el throughput es de 5.0385Gb/s la cual es aproximadamente 56.17% de lo que se tenía con 100 reglas. En este caso si se ve claramente que 1M de reglas si afecta el throughput. Sin embargo, y para el caso de OpenBSD con packetfilter, no es posible hacer una prueba similar debido al consumo excesivo de memoria, por lo cual, en la tabla 12 se muestra una comparativa entre OpenBSD y CentOS, ambos como firewall únicamente, para el caso máximo de 5000 reglas aplicadas con Packet Filter e Iptables, respectivamente.

Sistema Operativo	Sistema de filtrado de datos	Topología	Promedio Throughput [Gb/s]	Promedio Latencia en milisegundos
CentOS 7	----	Figura 11	9.267	0.1214
OpenBSD 7.2	Packet filter	Bridge, figura 12	1.549	4.7324
OpenBSD 7.2	Packet filter	NAT, figura 13	2.457	4.799
CentOS 7	Iptables	NAT, figura 14	8.846	0.541

Tabla 12. Comparativos promedios para 5000 reglas entre los sistemas operativos OpenBSD y CentOS.

La tabla 12 muestra un resumen comparativo en donde se ven los promedios de throughput y latencia, se puede observar la diferencia entre OpenBSD y CentOS. En el troughput para la topología NAT, la diferencia es de 6.38 Gb/s y en el caso de la latencia es de 4.258ms.

En la caracterización del canal de comunicación (segunda línea tabla 12), la velocidad de transferencia corresponde a un promedio de 9.267 Gb/s, que equivale a un aproximado del 90% de la velocidad de transferencia nominal reportada por los equipos y tarjetas de comunicación. Podemos observar entonces que CentOS con iptables como sistema de firewall perimetral con topología NAT, permite utilizar casi toda la velocidad de transferencia disponible porque al aplicar de 0 a 5000 reglas tenemos una pérdida en promedio de 0.421Gb/s, mientras que en OpenBSD la pérdida promedio en el troughput de 6.81Gb/s.

En el único caso en que CentOS se ve afectado en su troughput es cuando realizamos una prueba extra mostrada en la figura 28. En dicha prueba se puede concluir que la velocidad de transferencia disminuye casi a la mitad cuando se llega a un millón de reglas. Por otro lado, OpenBSD solo se pudo probar con 5000 reglas y la velocidad de transferencia promedio es de 1.5Gb/s para topología transparente y 2.47 Gb/s para topología NAT, aproximadamente del 15 % al 25 % de la velocidad de transferencia máxima del canal de comunicación de 10Gb/s. Es probable que un firewall bridge consume más memoria al cargar cualquier paquete con direccionamiento IP y realizar el filtrado.

El cálculo el Jitter indica que, dependiendo del retardo, hay en un envío de datos por un canal de comunicación o, en el caso de la reproducción de un video, dice qué tanto retraso hay con respecto a la fuente de despliegue. Como se observa en los resultados, en las tres pruebas realizadas el resultado es afectado por los errores de transmisión, pudiendo de esta manera ver si el envío de paquetes tiene muchos errores. Así, se puede inferir que esos datos pudieran tener muchos errores o incluso que falle la comunicación.

Los firewalls al filtrar el tráfico de la red afectan directamente el throughput e inducen un retardo. La cantidad de reglas que se apliquen en el firewall afecta directamente el rendimiento, de hecho, en otra prueba que no es parte de este trabajo, se encontraron resultados que demuestran esta afirmación. Para el caso de un equipo Fortinet-FortiGate-600C®se observó el bajo desempeño con un canal de comunicación de fibra óptica de 10Gb/s.

Un ejemplo de ello fue en pruebas realizadas en el año 2022, con un firewall marca Fortinet-FortiGate-600C®, también reportaba una velocidad de transferencia de 3.85Gb/s, dando un aproximado del 30% de rendimiento en la velocidad de transferencia.

```
ldg@atocatl ~]$  iperf3 -c 192.100.199.1
[ ID] Interval           Transfer     Bandwidth       Retr
[SUM]   0.00-60.00  sec  26.9 GBytes  3.85 Gbits/sec  21613 sender
[SUM]   0.00-60.00  sec  26.9 GBytes  3.85 Gbits/sec        receiver
```

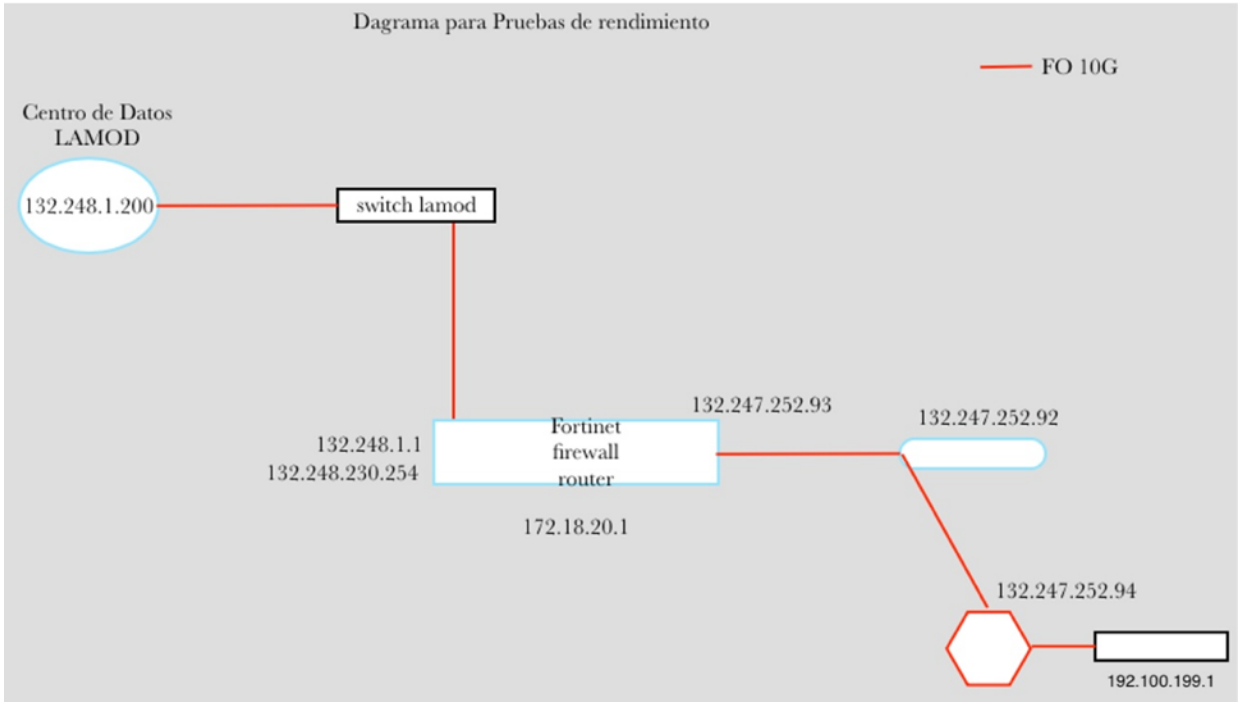


Figura 30. Topología de pruebas que se realizó entra Atocatl y un equipo fuera de la red LAN del Instituto de Astronomía, en la cual se probó para poder realizar transferencia de datos a velocidades cercanas a 10Gb/s. (estas pruebas se realizaron en abril del 2020)

En la figura 30 se muestra la topología para esta prueba, lo cual permite comparar el rendimiento descrito con el del firewall comercial de Fortinent®. En el resultado de esta prueba se muestra que, con un firewall Fortinet® FortiGate-600C® colocado en una red de fibra óptica con una velocidad máxima de transferencia de 10Gb/s, el rendimiento que se obtiene es aproximadamente de 3.85 Gb/s.

El equilibrio entre la disponibilidad y la confidencialidad es un aspecto muy sensible, y en seguridad al tener cada vez más confidencialidad, se afecta negativamente la disponibilidad. En este caso particular, al tener un firewall perimetral se reduce la velocidad de transferencia, porque el análisis de paquetes de datos provoca un aumento en la latencia y el troughput.

En 2024 la topología de red del Instituto de Astronomía tiene 2 redes como se muestra en la figura 1, estas redes son de diferentes velocidades, la red que llega a los usuarios es de 1Gb/s, y existe otra red de 10Gb/s, en la cual está conectado el dhcp-firewall y el ISP, lo que permite una trasferencia muy cercana a 1Gb/s para todos los usuarios.

Conclusiones

OpenBSD no es una opción recomendable para usar como un firewall perimetral porque reduce hasta un 70% la velocidad de transferencia máxima del canal de comunicación, ya que el rendimiento medido promedio es del 15% al 25% del máximo. Por otro lado, CentOS 7 como solución de firewall perimetral tiene mejor rendimiento porque el throughput promedio está arriba del 80%. OpenBSD está diseñado para fungir como firewall, pero no permite aprovechar toda la velocidad del canal de comunicación. Con un firewall en CentOS, como firewall perimetral, es mayor el aprovechamiento del ancho de banda del canal de comunicación, y solo se ve afectado por el número de reglas, como pudimos ver para el caso con un millón de reglas.

En el caso de OpenBSD, el throughput se mantiene constante al menos hasta 5,000 reglas y se puede afirmar que al usar OpenBSD como solución de firewall perimetral, tendremos un throughput aproximado de 2.74Gb/s en topología NAT, mientras que en CentOS el throughput es de 8.46Gb/s en promedio. De esta forma, se puede comprobar OpenBSD como solución de firewall de seguridad perimetral es muy ineficiente para transferencia de datos.

Una de las observaciones hechas en las pruebas fue que la distribución de la carga de trabajo en los procesadores para OpenBSD era muy grande para los cuatro núcleos, pero para CentOS era más eficiente y consumía menos recursos. Esto se puede explicar por la forma como funciona el sistema de filtrado Packet Filter de OpenBSD e Iptables de CentOS: mientras que CentOS carga las reglas por índice y conforme las va requiriendo las toma directamente de archivo de reglas, OpenBSD carga todas las reglas en memoria no importando si se requieren o no. Se observó que OpenBSD tardaba aproximadamente 10 minutos para cargar 5000 reglas, además, al tratar de cargar 20000 reglas tomaba más de 40 minutos, e incluso en muchas ocasiones fallaba y entregaba un mensaje de ”Kernel Panic”, razón por la cual no se pudieron realizar más de 5,000 pruebas en OpenBSD. En CentOS el tiempo aproximado de carga de las reglas era de 1 minuto, por lo cual se pudieron cargar 1M de reglas en 1 minuto. Estos resultados indican que iptables en CentOS es más eficiente en el manejo de reglas, pero es posible que no sea más seguro, debido a que no todas las reglas se cargan en memoria.

Trabajo Futuro

Una tarea significativa por realizar es un análisis del porqué el Kernel OpenBSD tiene limitaciones en la velocidad de transferencia con todas las interfases de red.

Otra tarea importante es cambiar el sistema de firewall y sustituir el OpenBSD con uno con la familia de CentOS con el fin de mejorar la throughput.

Un aspecto adicional por investigar, son las causas por las que el OpenBSD no es eficiente en el manejo de las velocidades de transferencia, esto probablemente sea provocado por cómo está configurado y diseñado su Kernel, y provoque un muy bajo rendimiento de throughput. El conocer esta razón y tratar de corregirla investigando a fondo el Kernel de OpenBSD, daría la oportunidad de poder utilizar un firewall eficiente y con alto rendimiento en velocidades de transferencia, pudiendo instalar un firewall de seguridad perimetral, muy seguro, eficiente y muy barato de implementar.

Otro aspecto que hay que medir el tiempo de OpenBSD en cargar las reglas, así como la eficiencia de estas, y compararlo con CentOS.

Respecto a los equipos instalados con sistemas operativos OpenBSD y CentOS como soluciones de firewalls perimetrales, se deberá analizar el uso de memoria al cargar las reglas y aplicarlas, uno de los resultados observados es que ninguno cambia su rendimiento a 5000 reglas, pero si se observó una disminución de throughput cuando se aplicaron 1,00,000 para el caso de CentOS.

Para ambos firewalls perimetrales no se probó el desempeño mientras tenía un ataque de fuerza bruta, y tampoco se midió el consumo de memoria y procesamiento, por lo que sería el trabajo pendiente para completar todo el análisis reportado en este trabajo, y ver el rendimiento del kenel, así como verificar las causas del porqué el OpenBSD tiene un bajo throughput.

Otro aspecto es la eficiencia computacional de los firewalls y su software de protección, ya que, según la sintaxis y su forma de aplicarse requieren diferentes parámetros, como puertos, direcciones, si son un grupo continuo secuencial o solo un grupo como los puertos, etc.

Otra característica que se pudo observar en estas pruebas es que, dependiendo del manejo de memoria y procesador de cada Sistema Operativo, marca la diferencia fundamental en el manejo de las reglas y el cómo se almacenan en memoria, así como el modo en que se aplican por los diferentes Kenel’s (núcleos) de cada Sistema Operativo.

Referencias

[1] Daniel Hartmeier Systor AG, “Design and Performance of the OpenBSD Stateful Packet Filter” (pf), [USENIX 2002 Technical Program Index] Pp. 171–180 of the Proceedings,

[2] Chirag Sheth, Rajesh Thakker, “Performance Evaluation and Comparison of Network Firewalls under DDoS Attack”, I. J. Computer Network and Information Security, 2013, 12, 60-67, Published Online October 2013 in MECS

[3] Michael A. Ihde, " EXPERIMENTAL EVALUATIONS OF EMBEDDED DISTRIBUTED FIREWALLS: PERFORMANCE AND POLICY ". THESIS, University of Illinois at Urbana-Champaign, June 2006

[4] OpenBSD Project, Manual Oficial OpenBSD (2022, mayo) [Online]. https://www.openbsd.org/

[5] Matthieu Herrb, “Installation de pare-feu redondants avec OpenBSD”, Laboratoire d’Analyse et d’Architecture des Systèmes, Jres Marseille 2022.

[6] Chirag Sheth, Rajesh Thakker, “Performance Evaluation and Comparative Analysis of Network Firewalls”, IEEE Xplore, Published in: 2011 International Conference on Devicesc and Communications (ICDeCom), Date of Conference: 24-25 February 2011

[7] Pilar Manzanares-Lopez, Juan Carlos Sanchez-Aarnoutse, Josemaria Malgosa-Sanahuja, Joan Garcia-Haro, “A Multicast Transport Protocol Design Methodology: Analysis, Implementation and Performance Evaluation”, Journal of Communications, vol.1, no. 5, Agosto 2006.

[8] Defense Advanced Research Projects Agency Information Processing Techniques Office, “Internet protocol Darpa internet program protocol specification”, RFC:791, septiembre 1981, pp 6.

[9] Product Specifications INTEL® (diciembre 1 2023) contenido en línea, Disponible en: https://ark.intel.com/content/www/us/en/ark.html

[18] Matthieu Herrb, Installation de pare-feu redondants avec OpenBSD, (Caos Conciencia, 2009), contenido en línea, Disponible en: https://www.openbsd.org/papers/jres2021-matthieu-tuto-pf-companion.pdf

[11] Alexander Bluhm ," Enviaste Visualization of Regression and Performance Know when something went wrong ", presentado en BSDCan 2019 ROM:DMS 1120, Junio 2019.

[12] OpenBSD perform 7.2 release test results (2019, 1 junio). [En línea]. Disponible en: http://bluhm.genua.de/perform/results/7.2/perform.html

[13] OpenBSD - Packet Filter/PF Rules Diagram. [En línea]. Disponible en: https://www.instructables.com/OpenBSD-PF-Filter-Rules/

[14] Massimiliano Adamo, Mauro Tabló, “A firewall performance test;,”, login: the usenix magazine, diciembre 2005, volume 30 number 6, Pp. 39-42.

[15] Ancho de banda vs rendimiento vs velocidad vs tasa de conexión, Intel® contenido en línea, Disponible en: https://www.intel.la/content/www/xl/es/support/articles/000026190/wireless.html

[16] Daniel Hoffman, Durga Prabhakar, Paul Strooper, ”Testing iptables”, Advancing Computing as a Science & Profession ACM Digital Library, CASCON '03: Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research October 2003, Pp. 80–91

[17] OpenBSD PF - Runtime Options, 2023 contenido en línea, Disponible en: https://www.openbsd.org/faq/pf/options.html

[18] Monitoring PF Firewalls For Health And Performance, Prefetch Technologies, 2023, contenido en línea, Disponible en: https://prefetch.net/articles/monitoringpf.html

[19] Protocolos TCP/IP, IBM®, 2023 contenido en línea, Disponible en: https://prefetch.net/articles/monitoringpf.html

[20] Modelo de arquitectura del protocolo TCP/IP Oracle®, 2023 contenido en línea, Disponible en: https://docs.oracle.com/cd/E19957-01/820-2981/ipov-10/

[21] G. Bianchi, "IEEE 802.11-saturation throughput analysis," in *IEEE Communications Letters*, vol. 2, no. 12, pp. 318-320, Dec. 1998.

[22] Mohammed M. Alani, “Guide to OSI and TCP/IP Models”, Springer, Cham, 04 March 2014

[23] Institute Sa Palomera, Control de acceso al medio Topologías, 2023, contenido en línea, Disponible en: https://www.sapalomera.cat/moodlecf/RS/1/course/module4/4.4.1.2/4.4.1.2.html

[24] Claudio Jeker, OpenBSD network stack internals, 2023, contenido en linea: https://www.openbsd.org/papers/asiabsdcon08-network.pdf

[25] T T.J. Socolofsky, C.J. Kale, "A TCP/IP Tutorial", Systems Limited, January 1991

[26] Y. Rekhter Cisco Systems B. Moskowitz Chrysler Corp. D. Karrenberg RIPE NCC, G. J. de Groot RIPE NCC E. Lear, Address Allocation for Private Internets, Silicon Graphics Inc., February 1996

[27] J.J. Stapleton, Security without Obscurity A Guide to Confidentiality, Authentication, and Integrity 1st ed.. Auerbach Publications, 2014

[28] IEEE, “802.3ae-2002 - IEEE Standard for Information technology - Local and metropolitan area networks - Part 3: CSMA/CD Access Method and Physical Layer Specifications

[29] Media Access Control (MAC) Parameters, Physical Layer, and Management Parameters for 10 Gb/s Operation”, IEEE Xplore, 26 August 2002

[30] IEEE, “Amendment: Physical Layer and Management Parameters for 10 Gb/s Operation, Type 10GBASE-T802.3an-2006 - IEEE Standard for Information technology”, August 2002

[31] Alyssa Lamberti, How to Measure Jitter & Keep Your Network Jitterbug Free Last pdated on Apr 9, 2023. [En línea]. Disponible en: https://obkio.com/blog/how-to-measure-jitter/#:~:text=Ping%3

[32] Systor AG Institute Sa Palomera, Daniel Hartmeier “Design and Performance of the OpenBSD Stateful Packet Filter (pf)” 2002, [En Línea] Disponible en: https://www.benzedrine.ch/pf-slides.pdf

[33] TANENBAUM, ANDREW S., “Redes de computadoras cuarta edición”, PEARSON EDUCACIÓN, México, 2003

[34] Claudia Kale , Theodore Socolofsky, “A TCP/IP Tutorial”, RFC 1180 IEEE, Enero 1991

[35] S. Bradner and J. McQuaid., “Benchmarking methodology for network interconnect devices”, Internet RFC 2544, Marzo 1999.

[36] Guido Van Rooij, “Real Stateful TCP Packet Filtering in IP Filter”, 10th USENIX Security Symposium (USENIX Security 01), Abril 2000.

[37] Malan, G.R., Watson, D. Jahanian Fateme, Howell, “Transport and application protocol scrubbing”. 1381 - 1390 vol.3. IEEE INFOCOM, Abril (2000).

[38] Kent Christopher, Mogul Jeffrey,”Fragmentation Considered Harmful”. ACM SIGCOMM Computer Communication Review, Febrero 2001

[39] C. Kreibich, M. Handley, V. Paxson., “Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics”, USENIX Security Symposium, Agosto 2001.

[40] Matthieu Herrb, “Installation de pare-feu redondants avec OpenBSD”, JRES 2022 - Marseille, Disponible en línea en: https://www.openbsd.org/papers/jres2021-matthieu-tuto-pf-companion.pdf

[41] Mark Handley and Vern Paxson and Christian Kreibich, “Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics”, 10th USENIX Security Symposium (USENIX Security 01) USENIX Association, Agosto 2001.

[42] ManKier Linux man pages, iperf3, (Abril 2 2024) [En Línea]. Disponible en:

https://www.mankier.com/1/iperf3#

[43] ESnet Fasterdata Knowledge Base iperf2 / iperf3, (Mayo 1 2024) [En Linea]. Disponible en: https://fasterdata.es.net/performance-testing/network-troubleshooting-tools/iperf/

[44] narkive NEWS GRUP ARCHIVE Maxim Khitrov, 10GbE (Intel X540) performance on OpenBSD 5.3, (Diciembre 15 2013) [En Linea]. Disponible en: https://mailing.openbsd.misc.narkive.com/qVVFzoPT/10gbe-intel-x540-performance-on-openbsd-5-3

[45] Javier Díaz, Luis Marrone, "Performance y Escalabilidad del Kernel Linux aplicado a Redes de Alta Velocidad", Tesis de grado Facultad de Informática, Universidad Nacional de La Plata,Argentina, Abril 2007.

[44] OpenBSD PF - Network Address Translation, (Marzo 2023) [En Línea]. Disponible en: https://www.openbsd.org/faq/pf/nat.html

[45] Protocolos TCP/IP IBM, (Abril 12 2021) [En Línea]. Disponible en: https://www.ibm.com/docs/es/aix/7.2?topic=protocol-tcpip-protocols

[46] Vinton Cerf, Yogen Dalal, "SPECIFICATION OF INTERNET TRANSMISSION CONTROL PROGRAM", IEEE Network Working Group, Diciembre de 1974

[47] R. A. de M. Valentim et al., "Medium access control: Multicycles Protocol for Hospital Automation over multicast with IEEE 802.3," 2008 34th Annual Conference of IEEE Industrial Electronics, pp. 1493-1498.

[48] Transparent Firewall Definition, Fortinet. (2022) [En Línea]. Disponible en: https://www.fortinet.com/lat/resources/cyberglossary/transparent-firewall

[49] Haoyuan Wang, Yue Xue, Xuan Feng, Chao Zhou, Xianghang Mi., " Port Forwarding Services Are Forwarding Security Risks" 10 Apr 2024.

[50] IEEE 802 MAC Addresses IEEE Standards Association (2023) [En Línea]. Disponible en: https://standards.ieee.org/products-programs/regauth/mac/

[51] KEVIN DOOLEY., "Network Topology The No-Sweat Guide", auvik Networks inc. 2015

[52] Robert L. Ziegler, Carl B. Constantine, "Linux Firewalls 2nd Edición", Prentice Hall, 2001

[53] Red Hat Enterprise Linux, ¿Qué es el Internet de las cosas (IoT)? (2023) [En Línea]. Disponible en: https://www.redhat.com/es/topics/internet-of-things/what-is-iot

[54] En marcha, nueva sala de supercómputo, Gaceta UNAM Dirección General de Comunicación Social, Dic 5, 2019 [En Línea]. Disponible en: https://www.gaceta.unam.mx/en-marcha-nueva-sala-de-supercomputo/

[55] Phil Dibowitz, "IP Filter FAQ", 2002 – 2007 [En Línea]. Disponible en: https://www.phildev.net/ipf/index.html

[56] Bluhm, Alexander, "Measuring Performance on OpenBSD", The Technical BSD Conference 2019 [En Línea]. Disponible en: https://av.tib.eu/media/45174

[57] Jacek Artymiak, “Building Firewalls with OpenBSD and PF”, Second Edition, devGuide.net Jacek Artymiak, 2003.

[58] ©Cisco Systems, Inc,“Software de Cisco Adaptive Security Appliance (ASA)” 2024 [En Línea]. Disponible en: https://www.cisco.com/c/es_es/products/security/adaptive-security-appliance-asa-software/index.html

Glosario de términos

Hercio o *hertz:* unidad de frecuencia del Sistema Internacional de Unidades; representa un ciclo por segundo en una señal. 1 MHz = 1 000 000 Hz.

Kilobit por segundo: se utiliza para expresar velocidades de transmisión de datos por segundo. Mb/s = 1 000 000 bits por segundo. Gb/s = 1 000 000 000 de bits por segundo especificadas el en Sistema Internacional de peseas y medidas. (SI)

***Ethernet*:** Esta definido en el Estándar IEEE 802.3 es una colección de estándares de la IEEE que definen la capa física y el control de acceso o "media access control" (MAC) de la capa de enlace de datos por cable. Los estándares son elaborados por la "Eléctricos y Electrónicos (IEEE)". Es una tecnología de red generalmente de área local (LAN) con algunas aplicaciones de red de área amplia (WAN).

Estas conexiones físicas se realizan entre equipos de red y/o computadoras por medio de equipos de red como hubs, switches y routers, mediante cableado de cobre o fibra óptica con arquitectura específica para este uso.

Throughput: Velocidad de transferencia de información también como la velocidad real de transporte de datos a través de una red de datos. La unidad de medida es de bits por segundo, (b/s), y los múltiplos Megabit por segundo (Mb/s), Gigabit por segundo (Gb/s).

Firewall (Pared de Fuego o contrafiuego): Equipos con software o software encargado en base a reglas específicas definidas por el usuario. Se encargan de filtrar el tráfico de la red, este puede estar instalado en un Sistema Operativo o puede ser perimetral para proteger una red *LAN (Local Area Network)*

Packet Filter (pf): Software libre que esta predeterminado en el Sistema Operativo OpenBSD (Berkeley Software Distribution), propiedad de la Universidad de California en Berkley, USA. Este software se encarga de realizar filtrado de tráfico de red.

Iptables: Programa que funciona como firewall disponible en todas las distribuciones del núcleo o kernel del Sistema Operativo Linux. Iptables es implementado mediante diferentes niveles de "netfilter". Como todo filtro de seguridad tipo firewall las reglas se almacenan en memoria.

Sistema Operativo (SO): Software especializado desarrollado en la tercera generación de computadoras, que permite administrar el hardware de una computadora de manera muy eficiente, traduciéndolo de código binario a lenguaje de alto nivel para que el usuario pueda desarrollar y programar aplicaciones y utilidades, el resultado es un software que facilita el uso de una computadora.

OpenBSD: Software de Sistema Operativo de código abierto de la familia de Unix, propiedad de Berkeley Software Distribution (BSD). OpenBSD se centra en la seguridad y está basado en NetBSD. Se enfoca en 5 características: libre, seguro, multiplataforma, estable y portable.

CentOS: Sistema Operativo del acrónimo en ingles Community ENTerprise Operating System, corresponde a una distribución Linux basado en GNU/Linux Red Hat Enterprise Linux RHEL, que es el código binario desarrollado por Red Hat, con la diferencia que es una distribución GNU de código abierto, y gratuito. La última actualización fue la versión 7 la cual se podrá actualizar hasta el 30 de junio de 2024.

Kernel: En informática es el "Núcleo" o la parte estructural básica de un Sistema Operativo de computadora, representa la arquitectura fundamental del Sistema Operativo.

Interfaz de red: Corresponde al hardware añadido a la computadora el cual por medio de una conexión tipo PCI o PCI-Express se agrega al hardware de la computadora con el fin de añadir a sus capacidades una conexión a la red de datos. El Sistema Operativo la identifica con algún alias de acuerdo como la reconozca para agregarla en el kernel del Sistema Operativo.

Bus de datos: En informática, el bus corresponde al canal digital con el que se transfiere información entre todos los componentes de la computadora, puede ser de manera paralela o serial.
Jitter: Se define como la fluctuación o variación de la de la comunicación de transmisión de datos, esto es muy utilizado para verificar la calidad de servicios sincrónico o P2P donde generalmente se usa para voz o video conocidos como streaming, por tanto, determina la calidad de la comunicación de estos servicios y puede medirse de esta manera. Por tanto, se puede conocer la medida de la fluctuación de la comunicación.
IEEE: Organización llamada por su acrónimo en inglés (IEEE, Institute of Electrical and Electronics Engineers) o Instituto de Ingenieros Eléctricos y Electrónicos. La IEEE cuenta con profesional técnico del mundo, agrupa a ingenieros, científicos, tecnólogos y profesionales gran parte del mundo y se dedican al avance en la innovación tecnológica. La organización es líder en el desarrollo de estándares para la industria, de los cuales destaca el estándar WiFi IEEE 802.11, que corresponde al estándar utilizado para el manejo de redes de datos, así como el estándar IEEE 802.3 utilizado ampliamente para redes cableadas.

Apéndices

Apéndice 1

Generador de Reglas para Packet Filter con OpenBSD
Archivo que contiene el programa en Linux shell tipo bash: Genera_reglas.sh

```
#!/bin/sh
rm pf.conf.n                        #---------->Borra las reglas anteriores si existiera
cp pf.conf.orig.b  pf.conf.n
echo Dame el Numero de Reglas:
read reglas
j=1
for ((n = 1; n <= $reglas; n++))
do
echo "block in quick on ix1 proto {tcp, udp} from
132.$(($RANDOM%254)).$(($RANDOM%254)).$(($RANDOM%254)) to 10.$(($RANDO
M%254)).$(($RANDOM%254)).$(($RANDOM%254)) port $(($RANDOM%65535)) #$n">>pf.conf.n
#echo $(($RANDOM%100))
done          # --------------------> de la n
```

Generador de Reglas para iptables Centos 7
Archivo que contiene programa en Linux shell tipo bash: Genera_reglas_iptables.sh

```
#!/bin/sh
rm iptables.n
cp iptables.orig  iptables.n
echo Dame el Numero de Reglas:
read reglas
for ((n = 1; n <= $reglas; n++))
do
echo -A INPUT -s 132.$(($RANDOM%254)).$(($RANDOM%254)).$(($RANDOM%254))/32 -d
10.0.$(($RANDOM%254)).$(($RANDOM%254))/32
-p tcp -m tcp --sport $(($RANDOM%65535)) --dport $(($RANDOM%65535)) -j REJECT -
-reject-with icmp-port-unreachable >>iptables.n
done
echo COMMIT>>iptables.n
```

Programa en shell tipo bash que genera las reglas en el cliente Calamar con CentOS 7.2, para medir la velocidad de transferencia por medio del programa iperf3, con un firewall CentOS 7 con iptables con topología NAT y con OpenBSD con pf (Packet Filter) con topologías NAT y transparente:

velocidad.sh

```
#!/bin/sh
```

```
date
traceroute 132.248.1.200
ifconfig enp4s0f1
for ((n = 1; n <= 20; n++))
do
echo Prueba numero: $n
echo ----------------------------------------------
iperf3 -l 1000 -V -c 132.248.1.200
done
date
```

Programa en Linux shell tipo bash, en el cliente para medir la velocidad de transferencia por medio de iperf3 para firewalls Centos 7 con iptables y OpenBSD con Packet Filter en topologías NAT y transparentes:
responde.sh

```
#!/bin/sh
date
traceroute 132.248.1.200
ifconfig enp4s0f1
for ((n = 1; n <= 20; n++))
do
echo Prueba numero: $n
echo
iperf3 -l 1000 -V -c 132.248.1.200
done
date
```

Apéndice 2

Configuración de la topología transparente (bridge) en OpenBSD
Instrucciones paso a paso para configurar el equipo que contendrá el firewall OpenBSD:

Se escribe un nuevo archivo de configuración llamado hostname.bridge0 y se agregan las interfases que formaran el puente (brigdge) y se le da el parámetro "up" (activo) para que inicie desde arranque: /etc/hostname.bridge0.

```
echo "up media autoselect" > /etc/hostname.ix0
echo "up media autoselect" > /etc/hostname.ix1
```

A continuación, se configura la interfaz puente (bridge) de acceso.
Esto hace que las interfases ix0 e ix1 se puedan ver como una sola y todo se enrute de ix0 a ix1 y viceversa, pero en el orden secuencial establecido, en este caso la ix0 corresponde a la red externa de la LAN, y la ix1 corresponde a la red interna de la LAN.

```
echo "inet 132.248.1.117 255.255.255.0 132.248.1.1" > /etc/hostname.bge0
```

Se reinicia el equipo para aplicar los cambios desde inicio de arranque del sistema, se usa el comando:

```
reboot.
```

Configuración de la topología NAT en OpenBSD:
Para la configuración NAT se deberán comunicar 2 redes, la privada y la pública.
En nuestra configuración es el equipo con dirección ip invalida
10.0.0.100 a la red 132.2481.114 como se muestra en esta topología.

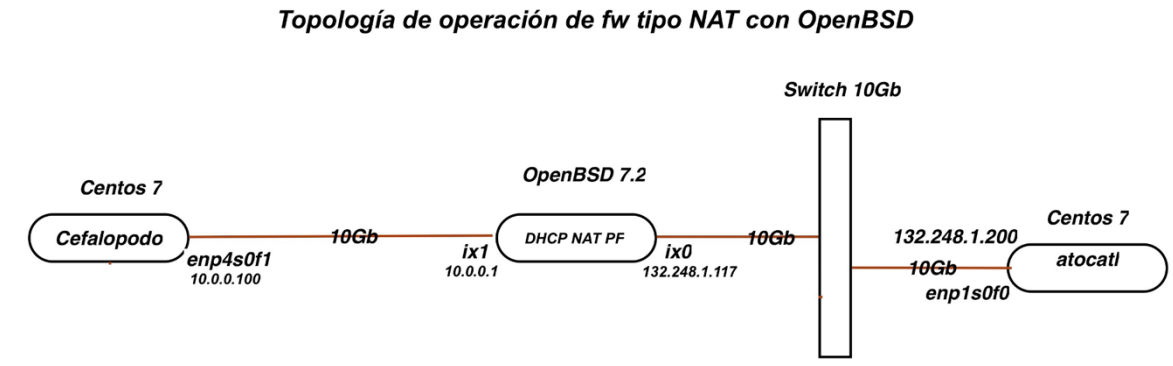


Figura 19. Topología NAT con firewall en OpenBSD

En la figura 19, definimos el Reenvío de IP, porque que NAT casi siempre se usa en ruteadores y puertas de enlace de red, se habilitara el reenvío de IP para que los paquetes puedan viajar entre interfaces de red en la máquina OpenBSD. El reenvío de IP se habilita mediante el sysct en el Kernel.

```
echo 'net.inet.ip.forwarding=1' >> /etc/sysctl.conf
```

Posteriormente definiremos las ip de cada interface para las que asignamos a la interfase llamada ix0 como la privada con dirección ip 10.0.0.100 y la ix1 con la dirección ip 132.248.1.117 como la interfaz pública
Con lo siguientes comando en el Shell :

```
echo "inet 132.248.1.117 255.255.255.0 132.248.1.1" > /etc/hostname.ix1
```

Y para ix0:

```
echo "inet 10.0.0.100 255.255.0.0 10.0.0.1" > /etc/hostname.ix0
```

Se define el gateway o puerta de salida con la dirección ip con el siguiente comando:

```
echo "1132.248.1.1" > /etc/mygate
```

Al existir un Packet Filter en el Kenel del Sistema Operativo OpenBSD 7.2 es necesario enrutar los paquetes que llegan al Packet Filter.

```
echo "match out log on ix0 from 10.0.0.0/8 to any received-on ix1 tag EGRESS nat-to (ix0:0)" >>/etc/pf.conf
```

Las siguientes lineas nos dan la opcion para definir la salida y entrada de tod excepto lo que se genere del generador de reglas

```
echo "pass in on ix0" >>/etc/pf.conf
echo "pass out on ix0" >>/etc/pf.conf
echo "pass in on ix1" >>/etc/pf.conf
echo "pass out on ix1" >>/etc/pf.conf
```

https://gnuplot.sourceforge.net/docs_4.2/node140.html

https://gnuplot.sourceforge.net/docs_4.2/node140.html