Universidad Nacional Autónoma de México

Facultad de Ingeniería

"CASO PRÁCTICO DE LA CONFIGURACIÓN DE FLUJO DE DATOS A TRAVÉS DE UNA RED"

TESIS PROFESIONAL

Que para obtener el título de:

Ingeniero en Computación

Presenta:

Emmanuel Eduardo Cuevas Aparicio

Directora de Tesis:

M.C. Cintia Quezada Reyes

Ciudad Universitaria, 14 de noviembre de 2014

Agradecimientos

A mi abuela.

Quiero dar gracias a mi abuela, que aunque ya no se encuentra presente entre nosotros ha sido esa fuerza invisible que me ha empujado a conquistar todas mis metas, que me ha inspirado a continuar en los momentos difíciles. Ella fue y seguirá siendo un ejemplo para mí pues su espíritu, fortaleza y humildad me han dado las herramientas para sortear los obstáculos que la vida me ha presentado.

Desde donde te encuentres abue, agradezco tus bendiciones.

A mis padres.

Ellos fueron ese gran apoyo moral y les doy gracias porque pueden estar seguros de que han hecho bien su trabajo educándonos bien a mí y mis hermanas, ya que nos hemos hecho personas de bien y con los valores bien inculcados, podemos tener siempre la frente en alto porque la mejor educación la hemos recibido en casa.

Gracias por todas esas horas de sudor y todos los años de aguantar mis travesuras, arranques y bromas que siempre les hice aun y cuando ya no tenía la edad para esos momentos tan divertidos. Gracias por esas oportunidades de oro que nos brindaron para tener un futuro asegurado como profesionales egresados de la mejor universidad del país. Gracias por habernos inculcado esa inocencia, humildad y nobleza para mantener siempre los pies en la tierra.

Al Ing. Carlos Saucedo Maciel.

Que ha sido un amigo incondicional y no conforme con eso ha sido y será un maestro en muchos ámbitos de mi vida. Tengo la firme convicción de que nuestros caminos se entrelazaron para que él fuera mi mentor a lo largo de todos estos años y yo le diera ese empuje y juventud a los suyos.

“Porque nunca dejaría que perdiera esas canicas que ha apostado por una persona como yo.”

A la vida.

A la vida, le agradezco que ha puesto a las personas correctas en mi camino (Iván, Vania, Angélica, Omar, Daniel, Fernanda, entre muchos más) para conocer miles de sentimientos como, tristeza, amor, desilusión, orgullo, cariño entre otros. Hay muchas personas a las que quiero agradecer porque ellas han formado parte de todas esas experiencias que llenan mis días y a las cuales les estaré agradeciendo en persona.

También quiero dar gracias a aquellas personas que aunque ya no están entre nosotros, se fueron dando una lección más de vida y que si fue fugaz o no su existencia me ha inyectado esas ganas de seguir adelante y darme cuenta de que un día podemos estar felices y de la noche a la mañana podemos faltar a nuestros allegados.

Si hay un buen momento de reconocer que amo vivir es aquí a través de estas palabras y dar fe que hay miles de experiencias que son una vez más insuperables.

Creo que la parte más difícil de haber concluido este proyecto de tesis son los agradecimientos, porque son tantas personas a las que quisiera colocar que las páginas de este anexo superaría el de la tesis misma. Además es mucho más difícil recordar muchos momentos y no sentir espasmos *esofágicos.*

No quisiera cerrar estas palabras sin agradecer a la M.C. Cintia Quezada Reyes que confió en mí y me apoyó para sacar adelante un pendiente en mi vida que yo creía perdido, posiblemente muchos chicos más le han agradecido más enérgicamente en comparación mía, pero quiero decirle de corazón que después de ser madre de familia, esposa y maestra me brindó su tiempo para que esto fuera posible.

Gracias.

Índice

Capítulo 1:
Red de Datos

"Locura: hacer lo mismo una y otra vez y esperar resultados diferentes."

-Albert Einstein-

1.1 Definición

Una red de datos es un sistema entre dos o más puntos entrelazados entre sí a través de un medio de comunicación o transmisión, ya sea éste o inalámbrico, con el fin de enviar o recibir un determinado flujo de información.

Los objetivos principales de una red de datos se podrían definir como los siguientes:

- Compartir información, recursos, inclusive software de manera remota ya sea localmente o de manera foránea dentro de un área geográfica.
- Brindar seguridad, confiabilidad y disponibilidad a la información empleando las herramientas necesarias para que ésta se pueda almacenar.
- Obtener una relación costo/beneficio.
- Transmitir información entre usuarios finales de manera rápida y eficiente.

Dentro de las redes de datos se definen dos principales tipos de transmisión (transferencia física de un flujo de bits a través de un canal de comunicación, ya sea físico o no físico):

a) **Redes de difusión:** este tipo de redes predomina dentro de la estructura del internet de hoy en día, básicamente es la conexión múltiple entre distintos equipos dentro de la misma red o de distintas redes entre sí.
   Un paquete que sea transmitido por este tipo de redes será enviado por un host o computadora y será recibido por múltiples hosts, cabe constatar que también influye el tipo de topología que se esté utilizando dentro de la compañía.

b) **Redes punto a punto:** este tipo de conexión es menos utilizado, ya que es una conexión de host a host para enviar mensajes o datos entre ellos, sin que sea necesario que dicha información tenga que pasar por varios dispositivos intermedios.

Una topología de red es una red de nodos interconectados entre ellos, en donde todos los nodos conectados intercambian datos dependiendo del tipo de topología que se haya configurado. Existen dos tipos de topologías:

a) **Topología física:** diseño del cableado de red.
b) **Topología lógica:** se define como la manera en la cual los hosts acceden a los medios.

La diferencia entre ambos tipos de topologías es que la topología física depende de cómo se interconectan todos los componentes de la red físicamente, la configuración para poder conectar los hosts a los conmutadores y éstos a los enrutadores. De esta manera se pueden evitar cuellos de botella y se configuran enlaces redundantes entre las tres capas que conforman una red correctamente estructurada; capa de núcleo, capa de distribución y capa de acceso.

Mientras que en la topología lógica se encuentra el protocolo configurado en el sistema para poder acceder a los datos, dentro de las dos topologías más conocidas están el token ring y Ethernet.

A continuación se describen las topologías físicas más utilizadas:

a) **Anillo**.- Dentro de una topología de anillo la información se transfiere de host a host hasta que ésta llegue al host final. Esto es posible gracias a una técnica denominada paso de tokens (serie especial de bits que viajan por la red, la cual permite conocer que host dentro de la red tiene el siguiente turno para enviar datos a través del medio). Cada vez que la información pasa por cada uno de los hosts, éste verifica las cabeceras del paquete examinando la dirección para verificar si dicho paquete va dirigido hacia ese host; si eso no ocurre el host devuelve el paquete a la red manteniendo intactas las cabeceras de destino. Finalmente dicho paquete es enviado al siguiente host en la línea (Figura 1.1).

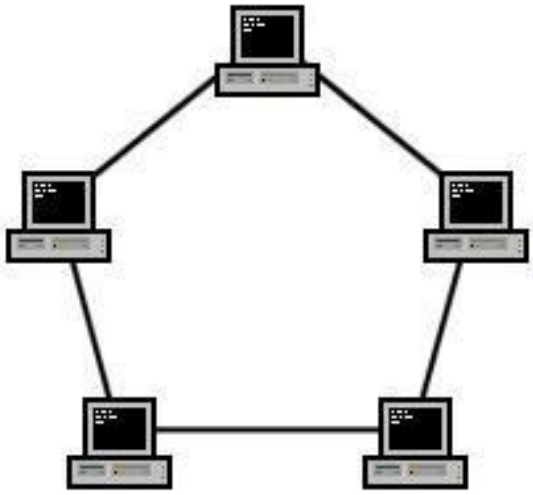


Figura 1.1. Topología de Anillo.

b) **Bus**.- Este tipo de topología pasiva (debido a que no regenera la señal de host a host a comparación de la topología de anillo) se caracteriza por que los hosts se conectan entre ellos a través de un bus por medio de segmentos de cable (Figura 1.2). Los datos pueden colocarse en el medio por parte de un host debido a una tecnología denominada control de acceso al medio.

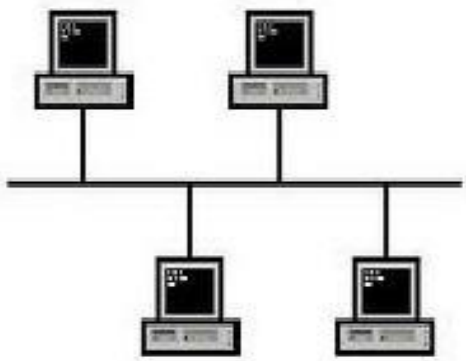


Figura 1.2. Topología de Bus.

c) **Estrella**.- La topología en estrella y estrella extendida son las más utilizadas en las conexiones de redes. En este tipo de topología los hosts se conectan a un dispositivo central el cual distribuirá los paquetes de datos a todos aquellos hosts que estén conectados a él (Figura 1.3). Anteriormente se utilizaban concentradores para este tipo de topología, pero debido a que este dispositivo causaba un gran tráfico de paquetes en la red hoy en día se utilizan conmutadores.

   Una topología en estrella es fácil de diseñar e instalar y gracias a que es escalable se pueden agregar más dispositivos a la red ya implementada anteriormente.

Figura 1.3. Topología de Estrella.

d) **Jerárquica.-** Este tipo de topología impone un orden en la red por medio del agrupamiento de equipos que se encuentran en ella, así como las dependencias de éstos, basándose en la ubicación física del equipo en la red (Figura 1.4).

Figura 1.4. Topología Jerárquica.

e) **Anillo Doble.-** Las características de esta topología son las mismas que las de la topología de anillo simple, una de las diferencias es que la anterior tiene el beneficio de proporcionar rutas predecibles para la recuperación de errores (Figura 1.5). También ofrece mayor confiabilidad que la topología de anillo, cuenta con dos rutas para que el tráfico de la red fluya.

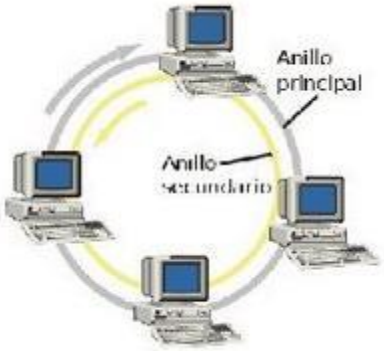


Figura 1.5. Topología de Doble Anillo.

f) **Malla.-** Las características de una topología en malla proporcionan redundancia en una red conectando un host con cada uno de los demás hosts de una red (Figura 1.6). Esta solución es muy costosa y se implementa en ambientes cuando no se pueden interrumpir las comunicaciones es muy confiada pero muy compleja. Por lo general, este tipo de topologías se implementan en los backbones de las empresas, debido a que los enlaces entre los equipos son más confiables y seguros.

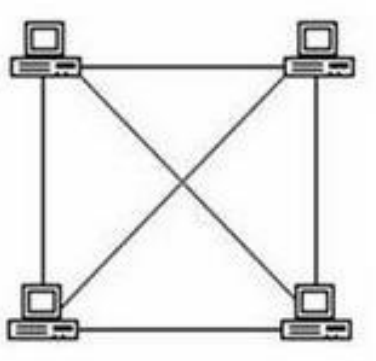


Figura 1.6. Topología de Malla.

De forma muy independiente de la tecnología así como de los métodos utilizados, las redes de datos pueden ser clasificadas según su alcance geográfico o tamaño en:

1. **PAN (Personal Area Network – Red de Área Personal)**

   Las redes PAN tienen un alcance muy limitado (no más de 5 m), se utilizan para interconectar dispositivos (PDA's, celulares, impresoras) vía infrarrojos, bluetooth y cables Ethernet de distancia corta. En casos muy recientes la nueva tecnología NFC (Near field communication - Campo de comunicación cercana) la cual solo logra comunicaciones a través de tecnología inalámbrica.

2. **LAN (Local Area Network – Red de Área Local)**

   Las redes LAN tienen un alcance más grande que la PAN pero no logran alcanzar una gran cantidad de equipos finales. Por lo general, son redes privadas que son instaladas dentro de un mismo edificio, oficina o campus universitario. Las velocidades de transmisión pueden llegar a ser de hasta 1000 Mb/s, encontrando en este tipo de redes topologías de tipo bus, estrella o anillo.

3. **WAN (Wide Area Network –Red de Área Amplia)**

   Las redes WAN se extienden a través de una amplia zona geográfica, la cual eventualmente agrupa a su vez otras subredes de tipo LAN, o ésta puede estar conformada por la interconexión de varias redes LAN. Las topologías utilizadas dentro de las redes WAN, al igual que las LAN predominan del tipo estrella, anillo, jerárquica y malla.

4. **GAN (Global Area Network –Red de Área Global)**

   Las redes GAN podrían denominarse como una sola red, es la interconexión de distintas WAN, dando como resultado a lo que hoy se conoce como internet. Este tipo de redes engloban a todos los servidores, computadoras y dispositivos interconectados en una misma red.

Debido a los enormes avances tecnológicos y a la gran demanda que las empresas demandan por la parte de servicios y aplicaciones a la industria de TI (Tecnologías de Información), se pueden encontrar otras dos nuevas clasificaciones de redes.

5. **SAN (Storage Area Network –Red de Área de Almacenamiento)**

   Una red SAN es una red concebida para interconectar servidores, matrices de discos y librerías de soporte dentro de una empresa. Este tipo de red basa su tecnología en un canal de fibra o en iSCSI (Internet Small Computer System Interface – Sistema de Interfaz de Internet de Pequeño Cómputo) con el fin de que la conexión entre dichos dispositivos sea rápida, segura y confiable.

6. **VLAN (Virtual LAN – Red de Área Local Virtual)**

   Es un grupo de computadoras conectadas entre sí de manera virtual, esto es gracias a tecnologías aplicadas tales como túneles para crear accesos a la red interna de una compañía por un medio seguro.

1.2 Medios de Transmisión

Una parte muy importante dentro de las redes informáticas son las tecnologías aplicadas para la interconexión de miles de dispositivos, los cuales ayudan a que una red de datos funcione de manera correcta.

A los medios de transmisión que se utilizan para el envío de información se clasifican en medios guiados y medios no guiados. Los medios guiados proporcionan un camino físico por el cual se puede enviar y recibir información entre los dispositivos de una forma directa y de cierta forma controlada. Mientras que los medios no guiados utilizan una antena para transmitir la información a través del aire, el vacío o inclusive el agua, éstos de una forma menos controlada, ya que no se puede tener el completo control de la información que viaje de esta manera.

Las características y calidad de la transmisión están determinadas por el tipo de señal, así como por las características del medio. En el caso de los medios guiados, el medio en sí mismo es lo más importante en la determinación de las limitaciones de transmisión.

En los sistemas de transmisión de datos, el medio de transmisión es el camino físico por el cual viajará la información entre el transmisor y el receptor en una red de datos, ya que en el diseño de sistemas de

transmisión es deseable que la distancia y la velocidad sean factores muy importantes a ser tomados en cuenta en el diseño de dichas redes.

1.2.1 Guiados

Se clasifica a los medios guiados como:

1. **Fibra óptica**

Hoy las empresas están implementando enlaces de fibra óptica, esto es debido a la gran confiabilidad y velocidad que se maneja en una conexión de este tipo, combinado con la enorme cantidad de información que viaja a través de la red de una empresa y a la gran demanda en cuestión al tiempo que una empresa llega a necesitar. Las empresas realizan inversiones millonarias para crear una conexión entre sus oficinas, ya sea que estén estas dentro del mismo país o en otra región del planeta.

Un cable de fibra óptica consta de filamentos de vidrio flexibles, estos filamentos pueden llegar a ser tan delgados como un simple cabello. El filamento de una fibra óptica está fabricado a base de silicio fundido, el cual conduce la luz, éste está compuesto por cinco partes importantes: el núcleo, el manto, el recubrimiento, los tensores y la chaqueta (Figura 1.7).

Las fibras ópticas son utilizadas para enlazar distintas redes LAN de alguna empresa, gracias a su gran capacidad de transmisión y a la enorme distancia que la señal puede viajar a través de la fibra. Convencionalmente dentro de las transmisiones utilizadas en los cables de fibra óptica un pulso de luz indica un bit, así como la ausencia de luz dentro del canal, en este caso el cable de fibra óptica indica un bit 0. Un punto muy importante a destacar es la aplicación de la física en este medio, debido a que la luz tiene la capacidad de refractarse cuando viaja a través de un medio, es decir, que se dobla entre las fronteras de los medios.

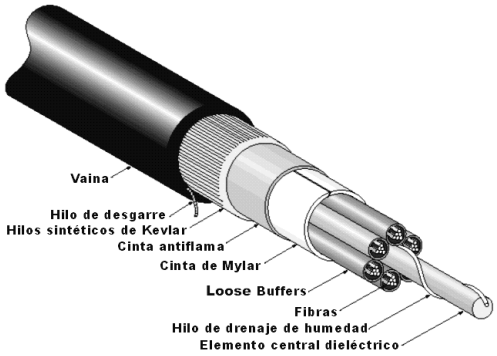


Figura 1.7. Ejemplo de fibra óptica.

Debido a la capacidad de transmisión de un cable de fibra, éste se clasifica en dos tipos:

a) Monomodo

La fibra Monomodo es un tipo de fibra la cual solo permite un modo de transmisión de la luz que se emite en el canal. Esto se puede lograr gracias a que se reduce el diámetro del núcleo de la fibra con un valor de 8.3 a 10 micrones. Gracias a esta importante característica, permite que las distancias de transmisión sean mayores a las fibras multimodo con valores que oscilan entre los 2.3 km hasta un

rango con valor de 100 km como máximo enviando por el medio grandes tasas de bits; esto es posible gracias a que solo se envía un único haz de luz con mayor intensidad por el medio.

b) Multimodo

La fibra multimodo es el tipo de fibra óptica el cual permite que más de un haz de luz viaje a través del cable, esto debido a que es un conjunto de fibras denominados comúnmente modos. Este tipo de cables de fibra óptica son utilizados en la industria para transmitir datos a alta velocidad, debido a que este tipo de cables de fibra son afectados comúnmente por los ángulos de refracción de la luz; los cuales hacen que dichos haces de luz no tengan la capacidad de viajar a grandes distancias como lo podría hacer un haz de luz dentro de un cable de fibra monomodo.

2. Cable Coaxial

El cable coaxial es otro tipo de medio de transmisión muy utilizado hoy en día, gracias a los beneficios que brinda y al bajo costo que presenta en comparación con otros medios de transmisión como lo puede ser la fibra óptica.

Se inventó alrededor de 1930 con el propósito de transportar señales eléctricas de alta frecuencia, esto debido a que en esos años se necesitaba manejar con señales de frecuencias cada vez más altas y con la reciente digitalización de servicios y comunicaciones; el cable coaxial se convirtió en la solución a muchos de los problemas en la actualidad.

El cable coaxial, tiene dos conductores para trabajar con señales de frecuencias más altas en comparación con un cable de par trenzado. Consta de un núcleo de hilo de cobre, el cual está rodeado por un aislante, que regularmente está fabricado de un material dieléctrico. Después del material dieléctrico se encuentra una malla de hilo trenzada que actúa como aislante del ruido eléctrico externo, misma que ayuda a reducir la afectación en las señales que viajan a través del núcleo. Finalmente está la cubierta exterior la cual se encuentra hecha de goma o plástico para evitar posibles descargas eléctricas por el medio.

En un cable coaxial el núcleo y la malla de hilo de cobre deben estar aisladas una de la otra, en dado caso de que ambas llegaran a tocarse entre ellas, llegaría a causar un corto circuito provocando que el ruido o las señales que se encuentren en el medio en ese instante, atravesaran el hilo de cobre.

Debido al blindaje de un cable coaxial o en otras palabras, a la disposición concéntrica de los conductores, el cable coaxial es mucho menos susceptible a interferencias eléctricas que un cable de par trenzado. Esto ayuda a que con un cable coaxial se puedan cubrir áreas más extensas, así como conectar un mayor número de estaciones o terminaciones en una línea compartida (Figura 1.8).

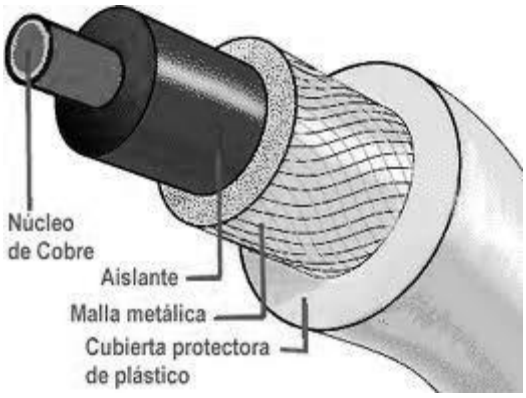


Figura 1.8. Ejemplo de Cable Coaxial.

3. Cable Par Trenzado

El cable de par trenzado fue inventado por Alexander Graham Bell, esto con el motivo de los tendidos de cable que se realizaban en las ciudades de Estados Unidos influenciado por el enorme crecimiento de las compañías telefónicas a finales del siglo XIX y principios del siglo XX.

Debido a que los primeros tendidos telefónicos eran un par de cables los cuales eran colocados a la altura de los cables de energía eléctrica y el alumbrado público de las ciudades, aunado con el crecimiento de la demanda de tendidos para el cableado telefónico, esto trajo un crecimiento en la interferencia en los tendidos de cableado telefónico por lo cual las compañías telefónicas se vieron en la necesidad de la aplicación de la torsión en los cables. Realizando los estudios necesarios, se llegó a la conclusión de aplicar una tasa de torsión de 4 vueltas por kilómetro.

Hoy en día el cable de par trenzado está hecho de pares de cables, los cuales están fabricados de cobre, éstos se encuentran trenzados uno alrededor del otro de tal manera que el entrelazado ayude a reducir la interferencia y la inducción electromagnética que se puede llegar a producir en el par de cables, estos dos efectos son causados por la transferencia de señales electromagnéticas a través de este medio físico. Cada entrelazado de cables lleva señales iguales así como de envío y recepción, mientras que el dispositivo de destino es el que se encarga de detectar las diferencias entre cada una de las señales, conociendo este método como modo diferencial de transmisión. Esto es posible debido a que el ruido que se introduce al enviar una señal por el medio físico se cancela con el ruido de la señal de regreso.

Existen tres categorías de cable de par trenzado:

a) Cable UTP (unshield twisted pair – par trenzado sin blindaje)

Dentro de la industria de las telecomunicaciones el cable de categoría UTP (unshield twisted pair – par trenzado sin blindaje) es el más demandado debido a las grandes ventajas que ofrece, entre ellas el bajo costo que tiene en el mercado, poniendo esta ventaja como la principal característica en comparación con los demás tipos de par trenzado que existen.

El cable de tipo UTP está fabricado generalmente a partir de 25 pares de acuerdo con un estándar desarrollado por un grupo de compañías. Uno de los subgrupos de los colores es el que se conoce comúnmente y el cual aparece en la mayoría de los cables UTP. Dicho subgrupo está conformado por los pares de colores:

- Azul – Azul/Blanco
- Verde – Verde/Blanco
- Naranja – Naranja/Blanco
- Café – Café/Blanco

Los pares de cables tienen una tasa de giro determinada para eliminar la interferencia entre ellos (Figura 1.9).

Figura 1.9. Ejemplo de Cable UTP categoría E.

Debido a que el estándar de redes de datos más común utiliza cables UTP, éstos son utilizados más a menudo en redes para conexiones en redes de área local por el bajo costo y el alto rendimiento que demuestran en estas condiciones.

Instituciones como la TIA (Telecommunications Industry Association – Asociación de Industrias de Telecomunicaciones)/EIA (Electronic Industries Alliance – Alianza de Industrias Eléctricas) han establecido normas de UTP y ha logrado clasificar seis categorías de cable UTP, algunas de las normas que se encuentran en la industria de las telecomunicaciones se describen en la tabla 1.1.

Categoría	**Velocidad**	**Características**
1	1 Mbps	Voz (Red Telefónica)
2	4 Mbps	Telefonía
3	16 Mbps	10BASET Ethernet
4	20 Mbps	Token Ring

5	100 Mbps (2 pares) 1000 Mbps (4 pares)	Gigabit Ethernet
5e	1000 Mbps	Gigabit Ethernet
6	10000 Mbps	Gigabit Ethernet

Tabla 1.1. Tabla de características para cable UTP.

b) Cable STP (shield twisted pair – par trenzado con blindaje)

El cable de par trenzado blindado es otro tipo de cable utilizado en redes informáticas con mejoras que el cable de par trenzado no-blindado no tiene. Una de ellas y la más importante es que utiliza dos pares de alambre, los cuales se envuelven en una malla de cobre con el fin de brindar un mejor rendimiento a interferencias electromagnéticas. Esto hace que suba su costo considerablemente en comparación con el cable sin blindaje (Figura 1.10).

Este factor ha sido el causante para que el cable STP (shield twisted pair – par trenzado con blindaje) no sea el elegido para los entornos normales de diseño para una red de datos, por lo que este tipo de cable se utilice solo donde los factores ambientales así como las interferencias electromagnéticas tengan un gran impacto dentro de la infraestructura de la red, brindando una mayor funcionalidad y disponibilidad en ésta.

Una desventaja muy importante y la cual debe de ser tomada en cuenta por los arquitectos de una red de datos, es que si una infraestructura de datos basada en cableado STP (shield twisted pair – par trenzado con blindaje) es combinada inadecuadamente, la señal puede llegar a degradarse viéndose afectado el rendimiento de red, por lo que para que una estructura de datos basada en STP (shield twisted pair – par trenzado con blindaje) sea eficaz del todo, toda la red debe ser blindada.

La longitud máxima que se llega a manejar en los estándares para una infraestructura STP (shield twisted pair – par trenzado con blindaje) es de 100 metros, llegando a tener tasas de transferencia de 10 a 100 Mbps.

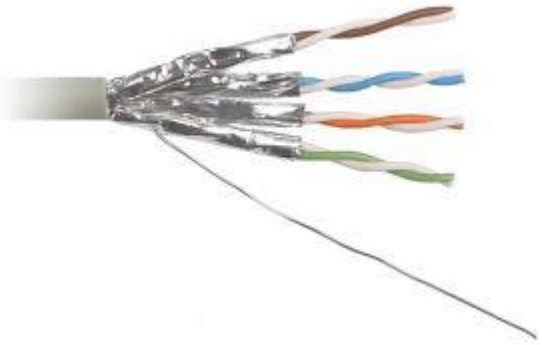


Figura 1.10. Ejemplo Físico de Cable STP.

c) Cable FTP (foiled twisted pair – par trenzado con lámina)

El cable de blindado global al igual que el cable STP (shield twisted pair – par trenzado con blindaje) está diseñado para la transmisión de datos de alta velocidad dentro de las redes de área local, con un blindaje para reducir las interferencias electromagnéticas en un rango mayor que el cable UTP (Figura 1.11).

Este cable es fabricado con pares conductores hechos de cobre entrelazados entre sí con una protección fabricada a base de una cinta de aluminio. A diferencia de un cable STP en el que el recubrimiento de la cinta de aluminio solo cubre un par trenzado de los que conforman el grueso del cable, FTP engloba y recubre todos los pares trenzados que forman el cable, siendo esta característica una de las razones por la cual su costo sea mucho mayor que un cable sin blindaje UTP y menor que un cable STP.

Este cable está fabricado para aplicaciones que requieren un aislamiento adicional en la señal, por lo cual cuenta con el blindaje de aluminio flexible y un hilo de cobre adicional para facilitar su conexión a tierra, siendo ideal para instalaciones que estén sujetas a una gran interferencia electromagnética externa.

Figura 1.11. Ejemplo Físico de Cable FTP.

1.2.2 NO GUIADOS

Los medios de transmisión no guiados son aquellos que transmiten señales electromagnéticas a través del ambiente, este tipo de medios, la información es enviada a través de medios como el vacío o el propio aire.

A partir de los años sesenta con el auge de la guerra fría y más aún por la conquista de la carrera espacial, los medios no guiados surgieron como la forma de comunicación del mundo moderno. Aunque los medios no guiados ya existían debido a las transmisiones radiofónicas durante la segunda guerra mundial, su explosión tecnológica se dio gracias a la utilización de satélites aeroespaciales y tecnologías de envío y recepción de datos a distancias totalmente enormes para esa época.

Hasta el día de hoy se siguen utilizando satélites, pero su capacidad ha sido mejorada debido a que transmiten una cantidad enorme de datos y de diferentes tipos de información tanto video, audio y

texto. También, este enorme crecimiento se ha dado gracias a que los medios de comunicación se han transformado al paso de los años, hoy en día debido a la gran necesidad de movilidad y comunicación los dispositivos también se han transformado de computadoras de escritorio a laptops y así hasta llegar a los celulares y tabletas inteligentes. Esto ha generado que la infraestructura también cambie a través de los años en los que relativamente se ha dado ese cambio, en un principio eran torres de radiotransmisión, continuando con los satélites y las antenas satelitales, microondas, infrarrojos, bluetooth y la tecnología NFC. Pero también se han transformado los métodos de decodificación y los estándares de transmisión pasando de las frecuencias moduladas hasta llegar a la tecnología wireless, 3G, 4G (englobando tecnologías como LTE, WiMax y HSPA+) llegando a ser ésta, la transmisión de datos extremadamente rápida.

La configuración para las transmisiones inalámbricas o no guiadas puede ser de las siguientes dos maneras; direccional y omnidireccional.

Durante el proceso de transmisión de datos direccional, la antena que transmite la información emite energía electromagnética concentrándola en un solo haz o canal de transmisión, por lo que la antena de recepción de datos debe estar alineada con la antena de transmisión para que la comunicación sea exitosa.

En el caso contrario, la transmisión de datos omnidireccional envía los datos de manera dispersa, siendo la señal emitida capaz de ser captada por diversas antenas las cuales se encuentren en el radio de transmisión de la antena transmisora. Algo similar a lo que ocurre con las transmisiones radiofónicas en las ciudades. Por lo tanto mientras mayor sea la frecuencia de transmisión de una señal a través de un medio no guiado, es mucho más factible que la señal converja en un solo haz de dirección.

Según la ITU (International Telecommunication Union – Unión Internacional de Telecomunicaciones) define a la comunicación entre dos puntos por medios no guiados como ondas electromagnéticas que se propagan por el espacio sin guía artificial y cuyo límite de frecuencia se fija convencionalmente a 300GHz.

Esto es gracias a que el espectro de frecuencias está dividido en bandas según se explica en la tabla 1.2.

Símbolo	**Nombre**	**Frecuencia**
ELF	Extremely Low Frequency	3- -30 Hz
SLF	Super Low Frequency	30 – 300 Hz
ULF	Ultra Low Frequency	300 – 3000 Hz
VLF	Very Low Frequency	3 – 30 kHz
LF	Low Frequency	30 – 300 Hz
MF	Medium Frequency	300 – 3MHz
HF	High Frequency	3 – 30 MHz

VHF	Very High Frequency	30 – 300 MHz
UHF	Ultra High Frequency	300 MHz – 3 GHz
SHF	Super High Frequency	3 GHz – 30GHz
EHF	Extremely High Frequency	30 – 300 GHz

Tabla 1.2. Tabla de frecuencias existentes y sus rangos.

Y básicamente se emplean tres tipos de ondas del espectro electromagnético (Figura 1.12) para las telecomunicaciones, que son:

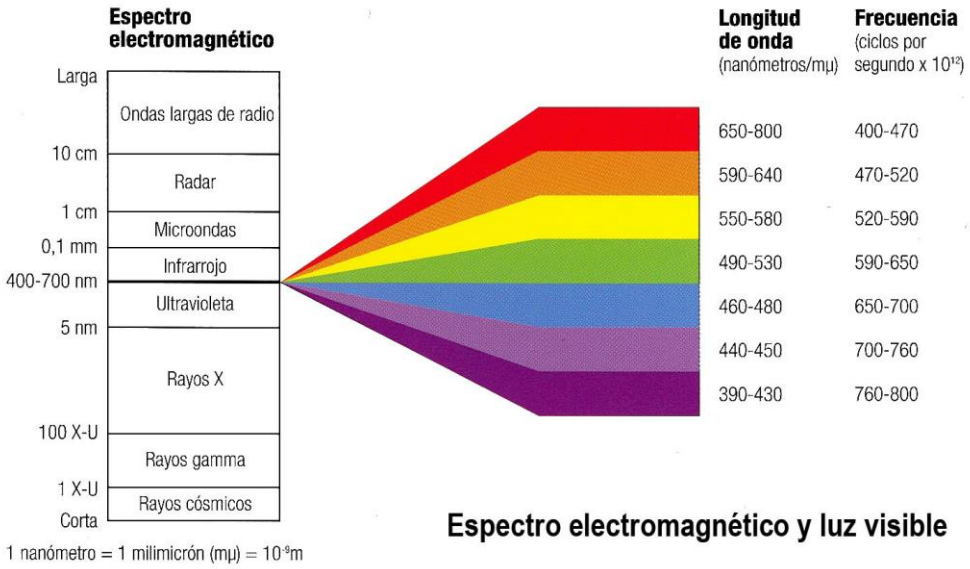


Figura 1.12. Imagen del espectro de rayos.

1. Infrarrojos

Descubiertos en el siglo XIX por William Herschel, las ondas infrarrojas son utilizadas en las telecomunicaciones para conexiones punto a punto de corta distancia, este tipo de ondas se encuentran en el espectro de banda de 300GHz a 400THz. Debido a que este espectro de ondas se encuentra más a la derecha de la luz visible por el ojo humano, el tipo de ondas puede llegar a ser detectado por el cuerpo en forma de calor y ya no en forma de luz visible. Es por eso que mientras más se acerquen las ondas a los extremos, ya sea izquierdo o derecho del espectro de frecuencias, es menos posible que se puedan detectar a simple vista o en alguna otra variante con los sentidos.

Es por eso que los rayos infrarrojos son un haz que emite calor. Este tipo de ondas fueron punta de lanza para el descubrimiento del espectro de ondas y sus características esenciales.

Los rayos infrarrojos se llegan a utilizar en la industria para controles remotos de televisores, conexiones punto a punto entre celulares y muy raras veces se llegan a usar en enlaces de larga distancia. Para que esto suceda, debe existir una alineación entre los dispositivos receptor y transmisor que ayude a la refracción del rayo entre ambos.

Una de las ventajas que tienen los rayos infrarrojos en comparación con los demás tipos de ondas de radiofrecuencia es que estas emisiones no necesitan una regulación por parte de alguna organización dentro del país, además de que necesitan menos cantidad de energía para que se produzca un rayo infrarrojo y que éste pueda ser emitido hacia un receptor.

Por eso, cuando evolucionó esta tecnología, fue una de las primeras que se implementó en la telefonía celular para establecer conexiones punto a punto y así compartir una gran variedad de información que iba desde una imagen, un contacto, hasta una canción. El único problema que presentaba era que los aparatos debían de estar perfectamente alineados para que la conexión fuera exitosa (Figura 1.13).

Con los años se lograron crear diferentes tecnologías que podían ser un híbrido de esta tecnología, como el bluetooth que establece conexiones multipuntos sin la necesidad de una alineación entre los dispositivos, pero siempre utilizando un espectro de rayos que va desde los rayos infrarrojos hasta las ondas de radio.

Figura 1.13. Conexión entre celulares punto a punto por medio de infrarrojos.

2. Microondas

Las señales microondas son uno de varios tipos de ondas electromagnéticas que se comenzaron a utilizar a finales de los 40 y comienzos de los años 50, debido a las mejoras que presentaba en comparación con las ondas de radio que eran utilizadas para el envío de información durante la Guerra Mundial.

Uno de los factores que ayudó a utilizar las microondas como forma de transmisión fue el auge de la Guerra Fría y la conquista de la era aeroespacial, debido a que las ondas de radio frecuencias eran

omnidireccionales y cualquier antena podría captar la señal y por medio de ésta obtener la información. Al ser direccionales las ondas de microondas la información se enviaba a los satélites que se comenzaban a poner en órbita a mediados del siglo XX.

Por lo tanto se utilizan microondas para enlaces punto a punto, ya que las altas frecuencias a las que son emitidas, hace posible la transmisión de información en largas distancias. Por lo tanto, las longitudes de onda corta crean ondas de alta frecuencia, mientras que las longitudes de onda larga crean ondas de baja frecuencia (Figura 1.14).

Su rango de frecuencia oscila generalmente entre los 300MHz y los 300 GHz, mientras que el periodo de oscilación está determinado por 3 nanosegundos y su longitud de onda oscila en un rango entre 1m y 1 mm.

Figura 1.14. Ejemplo de conexión por medio de microondas.

Su uso se limita a transmisión de microondas terrestres y satelitales. Y entre sus aplicaciones se pueden encontrar el envío de señales de televisión satelital, telefonía de larga distancia, mientras que en México se usa para conectar regiones apartadas para brindarles el servicio de internet.

Es por eso que al ser un método de transmisión de datos es necesario hacer una evaluación de las necesidades que se desea que cumplan, hay factores que benefician y afectan la transmisión de datos por este método. La señal es afectada debido a la atenuación que sufre cuando un evento meteorológico -como la lluvia- sucede, mientras que por el lado contrario, es mucho más costoso realizar un tendido de cable que la colocación de una antena en regiones apartadas.

3. Ondas de Radio

Las ondas de radio, así como la radiotransmisión fueron la primera forma de envío y recepción de información de modo inalámbrica, sus inicios datan a inicios del siglo XX, aunque las teorías de la transmisión de forma inalámbrica a través de ondas de radio databan desde veinte años antes. Guillermo Marconi fue el primero que logró realizar la transmisión trasatlántica radioeléctrica. De ahí en adelante se utilizaría la tecnología para el sector comercial a través de las transmisiones de radio, televisión y telefonía. El espectro de radiofrecuencia se encuentra más allá del electromagnético situándola entre los 3KHz y los 300 GHz, utilizando solamente corriente alterna para la transmisión de datos (voz y video).

Este tipo de señales son omnidireccionales, por lo que cualquier antena que se encuentre dentro del rango de señal activa, puede detectar la señal y codificarla dependiendo del tipo de información que haya sido enviada por el transmisor (Figura 1.15).

Dentro de las frecuencias de radio se encuentran dos tipos de señales que son las que se utilizan con mayor porcentaje, las ondas de amplitud modulada y las ondas de frecuencia modulada.

Un factor muy importante en las ondas de radio, son las llamadas interferencias por multi trayectorias. Debido a la superficie terrestre, objetos y otros factores en la geografía del medio se pueden crear diferentes ondas o multitrayectorias, las cuales al ser detectadas por el aparato receptor hace que la información o imagen se distorsione o se retrase según sea al caso. Esto se puede observar cuando al recibir una transmisión de televisión aparece con múltiples imágenes encimadas en la pantalla o interferencias.

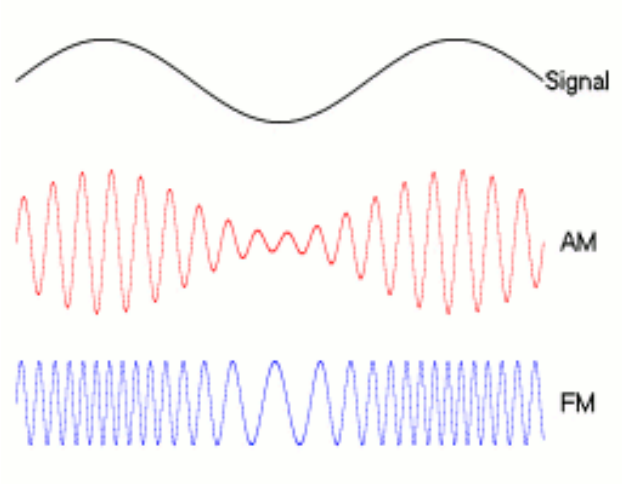


Figura 1.15. Tipos de radiofrecuencia de onda.

1.3 METODOS DE CODIFICACIÓN

Para que la transmisión de datos sea posible entre un emisor y un receptor, son necesarios métodos de codificación, los cuales sirven para convertir un flujo de datos analógicos a digitales y viceversa; con el fin de que tanto el emisor como el receptor puedan entender el mismo mensaje.

El proceso de codificación digital se establece a través de métodos de codificación ya establecidos, los cuales transforman la señal digital de un sistema y la codifican para que sea transportada por una antena emisora a través del aire en forma análoga. Del mismo modo la antena decodificadora recibe la señal de manera análoga y la transforma a una señal digital para que el sistema receptor pueda entender la señal que detectará en forma de bits.

Algunos parámetros importantes para la codificación y decodificación de la señal son:

a) **Número de canales**, según el tipo se definen en monoaural (transferencia de datos por un solo canal), biaural (transferencia de datos por dos canal) y multicanal (transferencia de datos por múltiples canales).
b) **Frecuencia de muestreo**, se define en la cantidad de muestras que se tienen de una señal por unidad de tiempo y se mide en Hz (ciclos por segundo).
c) **Resolución o número de bits**, determina la precisión con la que se reproduce la señal original. Ésta puede ser medida en diferentes muestras de 8, 10, 16 ó 24 bits.
d) **Bit rate**, es la velocidad o tasa de transferencia de datos y ésta se puede definir en bits por segundo.
e) **Pérdida**, está denominada por la cantidad de bits que se llegan a eliminar dentro de una transmisión de datos inalámbrica, mientras mayor sea la tasa de pérdida, menor es la credibilidad que tiene el codificador de los datos elegidos. Durante la trasmisión inalámbrica de información siempre habrá pérdida de bits, el aire es un medio no controlado; por lo que se han ideado algunos métodos para corroborar que la información que se transmite tenga un porcentaje aceptable de pérdida por el medio.

Los métodos de codificación de datos varían dependiendo de las necesidades del usuario y las capacidades del hardware en uso. Dentro de los métodos más conocidos se pueden clasificar por categorías; aquellos en los cuales su escala consta de dos niveles y en la cual solo se puede obtener un valor absoluto ya sea positivo o negativo. Mientras que existen los métodos que pueden tener un valor estrictamente positivo, negativo o nulo dentro de la escala de valores (de tres niveles).

Durante décadas los ingenieros eléctricos comenzaron a idear maneras de crear diferentes tipos de métodos de codificación, los cuales hicieran posible enviar un mensaje a través del uso de las señales electromagnéticas. Con el tiempo fueron descubriendo métodos de codificación que hacían posible transmitir cadenas de bits a mayor velocidad y con mayor confiabilidad con una menor tasa de pérdida de información. Algunos de los métodos de codificación son:

a) Codificación NRZ (no return to zero – sin retorno a cero)

El método de codificación No Return to Zero, es el más simple de todos los existentes, dentro de la escala de valores que existen, el valor absoluto del bit nunca regresa a cero entre los bits consecutivos. Por lo tanto, la codificación se define como una codificación bipolar, porque el receptor puede determinar el valor, si la señal está presente o no en el medio (Figura 1.16).

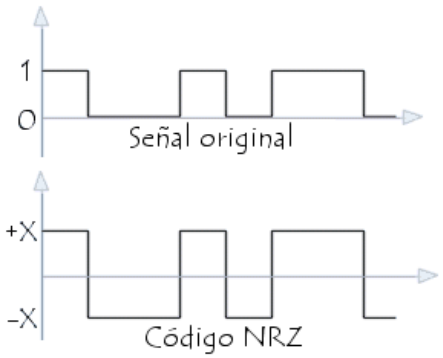


Figura 1.16. Gráfica representativa de los valores de código NRZ.

Algunas de las desventajas de este método, son su susceptibilidad a las interferencias y la continua sincronización que debe existir entre emisor y receptor.

b) Codificación NRZI (no return to zero inverted – no retorno a cero invertido)

El código NRZI también es un código bipolar, a diferencia del código NRZ, en este tipo de codificación la señal cambia cuando el valor de la misma es 1, mientras que mantiene la codificación cuando su valor es igual a 0 (Figura 1.17).

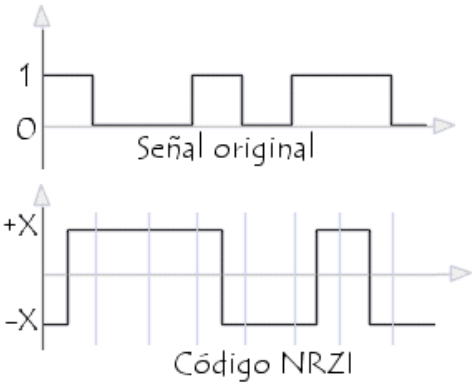


Figura 1.17. Gráfica representativa de gráfica de codificación NRZI

Una ventaja de este tipo de codificación es la detección de la señal a través del medio. Otro factor importante es la necesidad de una corriente de transmisión de baja señal, ésta puede llegar a ser un problema, ya que la presencia de una corriente continua durante una secuencia de ceros, llega a perturbar la sincronización entre el receptor y el transmisor.

c) Codificación MANCHESTER

La codificación Manchester también llamada a dos fases, es otro modelo de codificación de importancia para este tema, debido a que los enrutadores utilizan este tipo de codificación para una conexión física entre ellos.

Esta codificación se denomina auto sincronizada, ya que no es esencial utilizar una señal de reloj para sincronizar la transmisión entre el transmisor y el receptor, haciendo posible que en cada bit se obtenga la señal de reloj, obteniendo así su sincronización precisa durante el flujo de datos.

Para transmitir la señal a través del canal, el código Manchester hace posible un cambio de fase entre cada intervalo de la señal de reloj. Debido a este cambio de fase cuenta con un tiempo menor al periodo del receptor, le es más fácil detectar la cadena de bits sin la necesidad de sincronizar la señal de reloj con el transmisor. También la cadena se ve menos afectada por interferencias electromagnéticas. Por lo que una transición de un valor positivo a uno negativo representa un 0, mientras que la transición de un valor negativo a uno positivo representa en la cadena el valor de 1 (Figura 1.18).

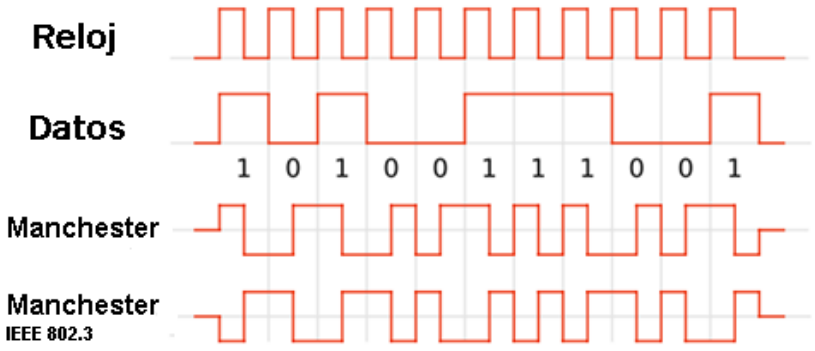


Figura 1.18. Ejemplo de codificación manchester.

Fue necesario incluir algunos métodos de codificación, para que el lector tuviera en cuenta que éstos son fundamentales para el envío y recepción de datos, así como conocer cuáles son los métodos que se utilizan para que la capa física -ya sea del modelo OSI (Open Systems Interconnection – Sistema de Interconexión Abierta) o del modelo TCP/IP (Transmission Control Protocol / Internet Protocol – Protocolo de Control de Transmisión / Protocolo de Internet)- puedan realizar las tareas necesarias para que las cadenas de bits sean colocadas en el medio en forma de bits (1´s y 0´s). Sin embargo, ésa es la manera en la que la capa física trabaja sin incluir tecnologías como las que hoy en día se utilizan en los medios de comunicación, como lo son los protocolos de transmisión de datos. Debe tenerse en cuenta que un protocolo y un método de codificación no es lo mismo, ni siquiera trabajan en la misma capa del modelo en uso.

Los protocolos utilizan los métodos de codificación para poder enviar la información a su destino final, sin embargo, éstos agregan cabeceras e información que ayudan a que el paquete llegue más rápido o que los datos que son encapsulados a través de los protocolos pierdan la menor cantidad de bits.

Muchos de los protocolos están pensados para que trabajen en tecnologías aplicadas a redes inalámbricas, debido a la enorme transformación de la industria de las telecomunicaciones en las que se necesitan mayor velocidad y confiabilidad en la transmisión de los datos de manera inalámbrica.

Los protocolos que se utilizan en la industria, dependen de factores como la velocidad, el ancho de banda y hasta el alcance de la señal. Dichos protocolos hicieron su aparición desde la transmisión de ondas de radio durante la Segunda Guerra Mundial, y de ahí en adelante se ha visto cómo la tecnología ha avanzado a través de los años. Desde los teléfonos celulares a inicios de los años 90, hasta el día de hoy que se encuentran servicios de internet de alta velocidad.

Capítulo 2:
Cableado Estructurado

"El éxito no es definitivo, el fracaso no es fatal: es el coraje para continuar lo que cuenta."

- Winston Churchill –

El buen funcionamiento de una red de datos es en buena medida producto del diseño de la red, no es suficiente el uso del hardware y del software si no hay de fondo un diseño capaz de lograr que la infraestructura sea pueda enviar y recibir datos de todos y cada uno de los dispositivos que forman parte de ella, haciendo uso de protocolos y medios físicos para que la comunicación sea posible.

Desde hace años existen organismos encargados de definir las normas y los estándares, los cuales ayudarán a definir la forma en la que los medios se comunicarán entre ellos. El cableado estructurado es un conjunto de normas y buenas prácticas, las cuales ayudan a los ingenieros y especialistas en diseño de redes a conocer cómo se interconectarán todos los equipos dentro de la red. Su objetivo principal es brindar una guía, para ayudar a diseñar una red que logre cubrir las necesidades de los usuarios durante la vida útil de cualquier diseño, haciendo posible que el éste permita una escalabilidad para futuras adiciones a la estructura de la red.

2.1 Subsistemas del Cableado Estructurado

Para realizar un buen diseño de una red, la estructura del cableado estructurado debe contar con los siguientes elementos o subsistemas:

1. Cableado Vertical o backbone
2. Cuarto de Equipos
3. Cuarto de Telecomunicaciones
4. Cableado Horizontal
5. Área de Trabajo
6. Entrada de Servicios

Cada uno de estos subsistemas cuenta con su propia gama de normas, las cuales son reguladas por organismos no gubernamentales e internacionales.

Una de estas organizaciones es la **ANSI**, el cual es el encargado de administrar y coordinar la normalización voluntaria y las actividades relacionadas a la evaluación de conformidad dentro de los Estados Unidos de Norteamérica.

También es la encargada de mejorar la competitividad entre las empresas promoviendo la utilización de normas y protegiendo la integridad de éstas a base evaluaciones de conformidad dentro de sus propios estándares.

Otro organismo encargado de vigilar las buenas prácticas es la **EIA**, la cual es la encargada de desarrollar las normas y publicaciones sobre las principales áreas técnicas: componentes electrónicos, electrónica para el consumidor, así como todo aquello que tenga que ver con la información sobre especificaciones técnicas de algún producto que sea fabricado en los Estados Unidos.

La **TIA**, así como la EIA son industrias que se dedican a desarrollar y publicar estándares para el cableado estructurado, éstas han sido acreditadas por la ANSI para desarrollar estándares voluntarios para la industria de las telecomunicaciones, es por este motivo que muchos de los estándares que se manejan en el cableado estructurado son clasificados como ANSI/TIA/EIA.

Otros organismos que son de gran importancia para este tema y los cuales no solo se dedican al desarrollo y publicación de normas y estándares dentro de los Estados Unidos son el **ISO** y el **IEEE**. El IEEE es el responsable de las especificaciones más técnicas dentro la industria de la electrónica, ya que "*Fomenta el conocimiento y los avances científicos y tecnológicos, los cuales, son transformados en productos prácticos y seguros, y en procedimientos que engrandecen la calidad de vida*", [1]tal como lo dicta su página oficial. Además tiene como tarea desarrollar estándares, tecnología y servir como un foro en el cual los ingenieros eléctricos discuten, promoven y desarrollan proyectos de trabajo los cuales fomentan una mejor vida.

Por el contrario, el ISO es una federación que se dedica a promover el desarrollo de estándares a nivel mundial y así facilitar el intercambio de bienes y servicios alrededor de todos los países por los cuales se encuentra conformada. Todas las actividades que allí se realizan, así como los acuerdos, son emitidos a nivel internacional y éstos se denominan como estándares internacionales, los cuales están protegidos por derechos de autor y para poder acceder a ellos es necesario realizar la compra de cada documento.

Todas estas organizaciones se encargan de emitir y verificar que las normas que cada una emite, sean acatadas por los fabricantes, desarrolladores y constructores de la industria de las telecomunicaciones, con el fin de que la interconexión entre las redes se lleve a cabo.

2.1.1 Cableado Vertical o backbone

El cableado vertical alberga las conexiones principales de toda la red, por lo cual algunas de las características que debe satisfacer son la alta velocidad y el gran ancho de banda.

Para cualquier compañía que cuente con diferente en un área geográfica es necesario que exista una conexión eficaz y confiable entre cada uno de los sitios que conforman la red, llamados enlaces.

Por otro lado, este tipo de enlaces suelen ser muy costosos debido a la distancia y por la cantidad de fibra óptica que se requeriría para enlazar cada una de las sucursales a la red de la empresa (Figura 2.1). Es por eso que distintas compañías emplean los servicios de un **ISP** o Proveedor de Servicios de Internet. Éstos ISP's se encargan de realizar las acciones necesarias para el tendido de fibra óptica o el levantamiento de antenas para que los enlaces estén funcionando correctamente. El proveedor de servicios de internet evaluará las necesidades de su cliente y él será el que determinará los aspectos indispensables con los cuales contarán cada uno de los enlaces para el envío y recepción de una enorme cantidad de datos.

[1] Misión del IEEE - http://www.ieee.org.mx/IEEE/IEEE_-_Mision.html

Figura 2.1. Gráfico representando un enlace backbone a nivel nacional.

Durante la evaluación que realiza el ISP se toman cuenta qué tipo de encapsulación se utiliza entre cada uno de los enlaces y el tipo de datos que viaja a través de estos. Dentro de estas características existen protocolos de encapsulación de alta velocidad, ya sea **PPP** (Point to Point protocol – Protocolo de punto a punto), **ATM** (Asynchronous transfer mode – Modo de transferencia asíncrona) o Frame Relay.

El cliente solo se tiene que preocupar porque su proveedor de servicios le entregue los enlaces con la conexión entre cada una de las sucursales.

Estos protocolos mencionados utilizan celdas donde se encapsula la información y se envía a través de la red, sin embargo, el protocolo debe estar configurado en ambos dispositivos que estén compartiendo la información para que una vez recibida, se pueda desencapsular y el dispositivo pueda entender la trama de datos recibida.

a. PPP

Uno de estos protocolos de comunicación es el conocido como PPP. El protocolo de punto a punto es un conjunto de estándares que permiten la interacción de software de acceso remoto de diversos proveedores de servicios (Figura 2.2). Este protocolo admite características avanzadas de interconexión de datos y puede utilizar protocolos tales como **TCP/IP** o **IPX** (Internetwork Packet Exchange – Paquete de Intercambio de red) como protocolo de red.

El proceso de conexión se establece a base de sesiones, las cuales primero realizan una negociación entre los dispositivos que vayan a realizar la transferencia, una vez establecido el punto de negociación se procede a establecer la calidad que tendrá el enlace durante la sesión. Finalmente se negocia el protocolo de red que se utiliza para concluir el proceso de establecer la conexión entre los dos puntos.

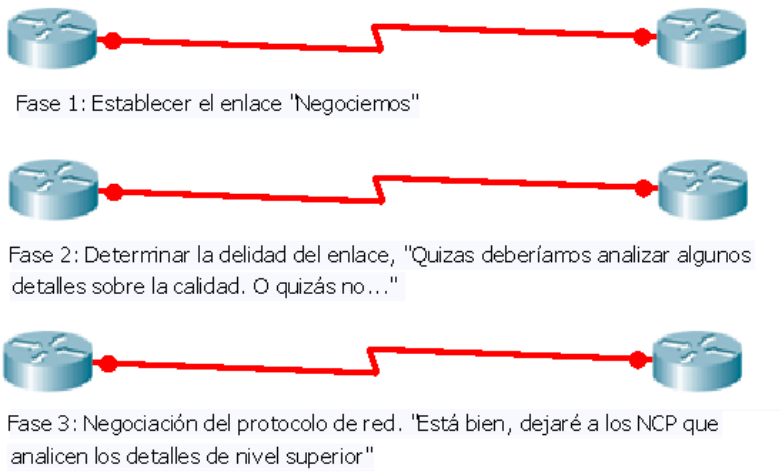


Figura 2.2. Establecimiento de una sesión PPP.

b. Frame Relay

Frame Relay es otro de los protocolos utilizados para la interconexión de dos puntos, éste evolucionó para ser un estándar que cubriera las necesidades del mercado en el sector de las telecomunicaciones.

Frame Relay ofrece una simplificación de los servicios bajo la nube, esto quiere decir que en vez de utilizar un enlace dedicado entre cada uno de los dispositivos de la red, éstos se entrelazan en un solo punto neurálgico que cual cuenta con los dispositivos necesarios para crear circuitos virtuales, éstos son necesarios para crear los enlaces entre los dispositivos de la red

Esto se traduce en un gran beneficio para las empresas que cuentan con enlaces de este tipo, en vez de estar alquilando un enlace dedicado a través de cada una de las sucursales, pueden alquilar solo un enlace de las características específicas a las necesidades de la empresa. Una ventaja entre una conexión Frame Relay en comparación con una PPP, es que no se aprovecha todo el ancho de banda disponible en un enlace directo, por lo que puede representar pérdidas económicas para una empresa (Figura 2.3).

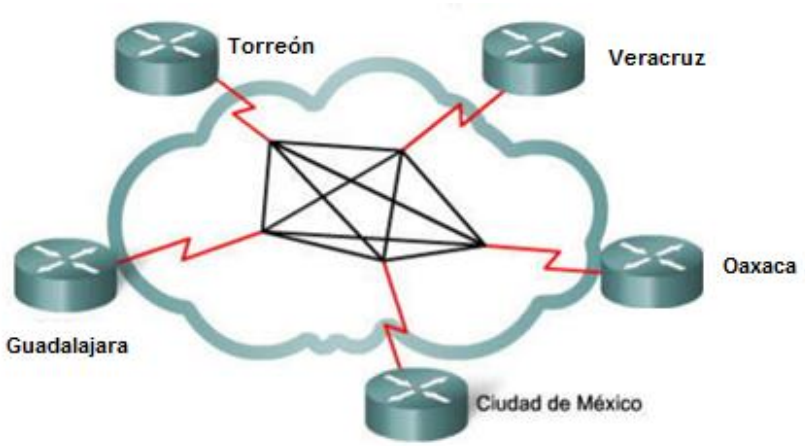


Figura 2.3. Ejemplo de Backbone bajo protocolo de enlace Frame Relay.

c. ATM

El protocolo de conexión ATM, está diseñado para soportar grandes cantidades de datos de la capa de enlace de datos. La versatilidad que presenta ATM para la conmutación de paquetes de longitud fija (denominadas celdas ATM) son una de las ventajas en comparación con otros protocolos de conexión de gran velocidad. Estas celdas son verificadas a través de conmutadores ATM los cuales aseguran que el gran tráfico de datos de la red sea conmutado de manera correcta a través de la red.

Al igual que Frame Relay, la conmutación de los paquetes dentro de la red funciona por medio de circuitos virtuales, los cuales contienen las celdas de información de ATM. Estas celdas tienen encabezados que contienen la información del dispositivo de origen y del dispositivo de destino.(Figura 2.4). La información es leída por el dispositivo enrutador para que pueda enviar el paquete a través del segmento de red correcto.

Este protocolo permite que los ISP's puedan establecer costos más bajos y que sus clientes solo paguen por la cantidad de celdas que fueron transportadas satisfactoriamente en vez de pagar por un servicio dedicado.

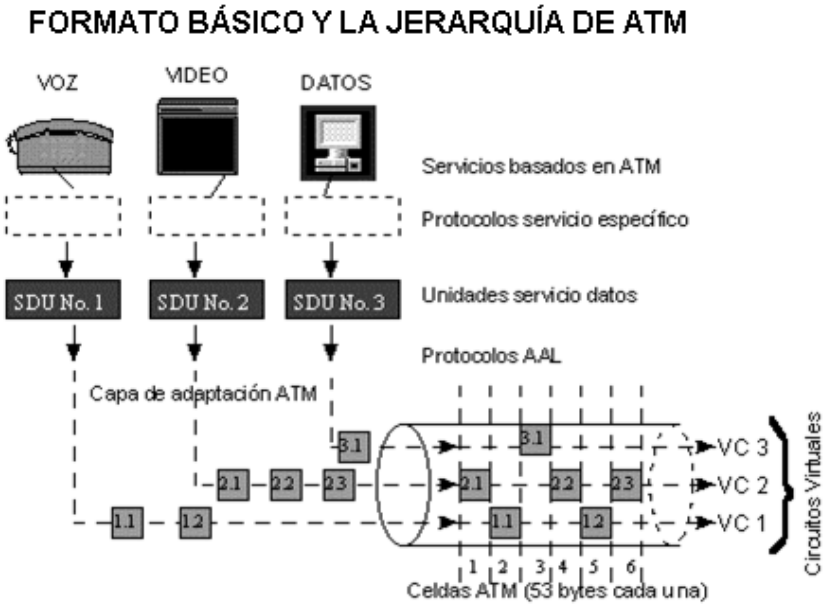


Figura 2.4. Ejemplo de multiplexación por medio de ATM.

2.1.2 Cuarto de Equipos

El cuarto de equipos es el lugar designado para los equipos de telecomunicaciones que trabajan en la capa de distribución, con el fin de enrutar la información entre los diferentes cuartos de telecomunicaciones de toda la red.

Se deben tomar en cuenta diferentes consideraciones para mantener el cuarto de telecomunicaciones en un ambiente de trabajo adecuado, la norma que estandariza estas consideraciones es la EIA/TIA 569.

Dentro de esta norma hay varias recomendaciones que se necesitan tomar en cuenta para que los equipos de telecomunicaciones trabajen en un ambiente ideal. Algunas de las recomendaciones son:

- El cuarto de equipos no debe de estar ubicado debajo de niveles de agua.
- Debe contar con sistemas de ventilación, enfriamiento y aire acondicionado.
- Debe estar ubicado lejos de fuentes de interferencias electromagnéticas (motores, transformadores eléctricos, generadores y equipos de rayos X).
- Debe contar con una altura mínima de 2.44 metros sin obstáculos.
- Se debe instalar un conducto de conexión directa a tierra.

Todas estas recomendaciones se hacen con el fin de procurar que los equipos de telecomunicaciones puedan trabajar bajo condiciones estándar, ya que el cuarto de equipos está directamente conectado al backbone de la red, por lo que si existe un problema en el cuarto de telecomunicaciones la red del edificio se verá seriamente afectada.

Los equipos de telecomunicaciones que se encuentran albergados en el cuarto de equipos son equipos de capa 2 (capa de distribución) tales como enrutadores conmutadores de distribución.

a) Enrutadores o routers

Estos dispositivos deben pertenecer al backbone de la **WAN** y no en la periferia. Para cumplir con esta función, el enrutador debe soportar varias interfaces de telecomunicaciones de la mayor velocidad que se utilice en el núcleo de la WAN y debe poder reenviar los paquetes IP a la velocidad máxima por todas esas interfaces (Figura 2.5). El enrutador también debe admitir los protocolos de enrutamiento que se configuren en el equipo como son vector distancia, estado enlace, **RIP** (Routing Information Protocol – Protocolo de información de ruta), **IGRP** (Interior Gateway Routing Protocol – Protocolo de enrutamiento de Gateway interno), **EIGRP** (Enhanced Interior Gateway Routing Protocol - Protocolo de enrutamiento de Gateway interno mejorado), **OSPF** (Open Shortest Path First – Primera ruta más corta abierta) y **BGP** (Border Gateway Protocol – Protocolo de Gateway de borde).

Figura 2.5. Enrutadores WAN marca Cisco.

b) Conmutadores o switches de distribución

Los conmutadores de capa de distribución son equipos que realizan mayores funciones de conmutación de paquetes, por esa razón deben tener un mayor rendimiento que un conmutador normal o de capa de distribución. Un conmutador de distribución debe configurarse en un punto focal dentro de la red debido al tráfico de VLAN (Virtual Local Area Network – Red de área local virtual) y normal que pasan a través de él, éste debe tener configuradas ciertas características las cuales permiten tener un mayor control sobre el tráfico de la red y que pueda funcionar como un enrutador (Figura 2.6).

Los conmutadores de distribución también se conocen como conmutadores multicapa, ya que son capaces de trabajar con los paquetes de la capa 2 así como de capa 3 del modelo OSI. Un beneficio de

los conmutadores multicapa es el precio en comparación con un enrutador, por lo que llega a ser más accesible colocar un conmutador multicapa y configurar redes virtuales en la red para la segmentación que se necesite.

Figura 2.6. Conmutadores de distribución marca CISCO.

2.1.3 Cuarto de telecomunicaciones

El cuarto de telecomunicaciones es donde se encuentran instalados todos los dispositivos que hacen posible que la red de datos funcione. Dentro de los cuartos de comunicaciones se colocan dispositivos que van desde enrutadores hasta conmutadores y son los encargados de distribuir todos los paquetes que viajan desde un punto A hasta el punto B dentro de la red de datos (Figura 2.7).

Figura 2.7. Cuarto de telecomunicaciones.

Algunos de los dispositivos que se colocan en este rack de telecomunicaciones son:

a) Módems

Estos dispositivos modulan una señal portadora analógica para codificar información digital y de-modula la señal portadora para decodificar la información transmitida. Un módem de banda de voz convierte las señales digitales producidas por una computadora en frecuencias de voz, las cuales se pueden transmitir a través de las líneas analógicas de la red de telefonía pública (Figura 2.8). En el otro extremo de la conexión, otro módem vuelve a convertir los sonidos en una señal digital para que ingrese a una computadora o a una conexión de red. Los módems más rápidos, por ejemplo los módems por cable y **DSL** (Digital Subscriber Line – Línea de Suscripción Digital), transmiten la información mediante el uso de frecuencias de banda ancha mayores.

Figura 2.8. Modem CISCO con 24 slots.

b) CSU/DSU

Los dispositivos **CSU/DSU** interconectan dos puntos sin importar la distancia que exista entre ellos, ya que necesitan una unidad de servicio de canal y una unidad de servicio de datos para hacer posible la interconexión entre ambos lugares. Éstas se pueden encontrar combinadas en una sola pieza del equipo, llamada CSU/DSU (Figura 2.9). La unidad CSU proporciona la terminación para la señal digital y garantiza la integridad de la conexión mediante la corrección de errores y la supervisión de la línea, mientras que la unidad DSU convierte las tramas de la línea Portadora T (designación de un sistema genérico para los sistemas digitales multiplexados) en tramas que la red LAN puede interpretar y viceversa.

Figura 2.9. Dispositivo CSU/DSU

c) Servidores de Acceso

Estos servidores concentran las comunicaciones de aquellos usuarios que utilizan servicios de acceso con marcación. Un servidor de acceso puede tener una mezcla de interfaces tanto analógicas como digitales y admitir a cientos de usuarios al mismo tiempo (Figura 2.10).

Figura 2.10. Servidores de acceso dedicado.

d) Conmutadores

Los conmutadores de red cuentan con varios puertos que se utilizan en redes portadoras. Estos dispositivos normalmente conmutan el tráfico (por ejemplo Frame Relay, ATM o X.25) y operan en la capa de enlace de datos del modelo OSI. Dentro de la nube también es posible utilizar conmutadores de red pública de telefonía conmutada (**PSTN**, Public Switched Telephone Network) para conexiones de

conmutación de circuitos, por ejemplo, red digital de servicios integrados (**ISDN**, Integrated Services Digital Network) o conexión telefónica analógica (Figura 2.11).

Figura 2.11. Conmutadores de acceso marca CISCO.

Como se puede observar, existe una gama interesante de dispositivos que ayudan a que la red trabaje de una manera más eficaz, utilizando las capas de núcleo, distribución y acceso, para que cada una de las estaciones de trabajo cuente con acceso a la red de forma rápida y segura (Figura 2.12). Estos dispositivos trabajan junto con los protocolos de red para colocar las tramas en los cables físicos.

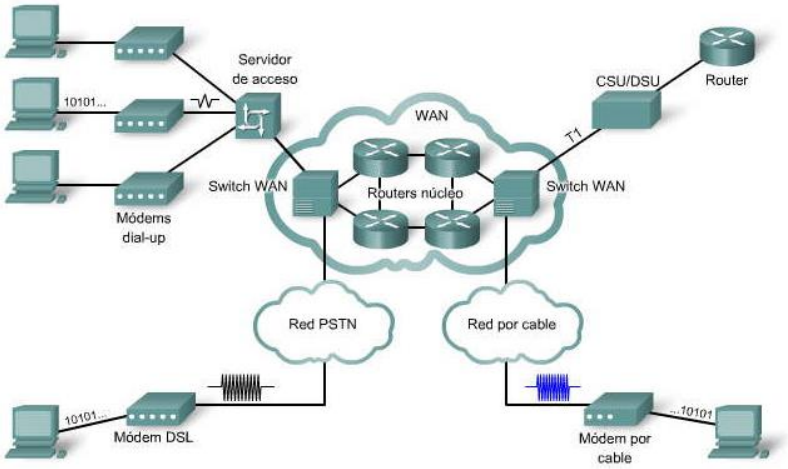


Figura 2.12. Dispositivos dentro de una red.

2.1.4 Cableado Horizontal

El cableado horizontal conecta los cuartos de telecomunicaciones con las estaciones de trabajo o dispositivos finales a través del tendido de cables entre ellos. Dentro del cuarto de telecomunicaciones se conecta al switch a través de la roseta final en un tendido de cable que va desde 95 metros, tal como lo recomienda el estándar ANSI/EIA/TIA 569 A (norma de recorridos y espacios de telecomunicaciones en edificios comerciales) (Figura 2.13).

Figura 2.13. Patch Panel

El recorrido del cableado comienza desde el patch panel, que es un panel que se conecta a un conmutador para poder organizar todas las conexiones de la red y así realizar cambios sin la necesidad de desconectar equipos de red directamente del conmutador o enrutador.

Un patch cord es un cable hecho de fibra óptica o cable UTP (unshielded twisted pair) con conectores de fibra o UTP por ambos lados (Figura 2.14). Éste sirve para interconectar dispositivos electrónicos en la red de datos. Son fabricados de distintos colores para facilitar la identificación de los dispositivos que están interconectados, su longitud varía de unos cuantos centímetros hasta 100 metros cumpliendo con los estándares de telecomunicaciones.

Figura 2.14. Patch cords hechos de cable UTP identificados por colores.

Algunas de las características esenciales que definen un cableado horizontal dentro de un edificio son las siguientes:

a) No se permiten puentes y derivaciones o empalmes a lo largo del trayecto del cableado.
b) Se debe considerar la proximidad con el cableado eléctrico, el cual genera altos niveles de interferencia electromagnética, tales como motores, transformadores o cualquier fuente de energía a base de inducción (ANSI/EIA/TIA 569, norma de construcción comercial para espacios y recorridos de telecomunicaciones).
c) La longitud máxima permitida para un patch cord es de 100 m, utilizando 95 m para el trayecto desde el patch panel hasta el área de trabajo y 5 m para el tramo final entre el equipo de trabajo y la roseta de conexión.
d) Utilización del estándar ANSI/EIA/TIA 568 para el cable de par trenzado y la fibra óptica:
   - UTP: Par trenzado sin blindaje – 100 Ohms, 22/24 AWG.
   - STP (shielded twisted pair): Par trenzado con blindaje – 150 Ohms, 22/24 AWG.
   - Fibra Óptica Multimodo, 62.5/125 y 50/125 µm (nanómetro) de 2 fibras.

2.1.5 Área de Trabajo

En lo general en el área de trabajo se siguen algunos consejos para que el usuario pueda llegar a conectar su computadora a una conexión en la pared y trabaje desde su escritorio o estación. Pero las necesidades de movilidad en la actualidad han llevado a convertir ciertos conceptos con los que se instalaban las áreas de trabajo para el acceso a la red.

Ahora es necesario la instalación de módems inalámbricos, los cuales permitan la conexión a través del wireless (comunicación inalámbrica que utiliza modulación de ondas electromagnéticas a través de un medio no físico) entre los dispositivos móviles y la red.

Algunas de las características que definen a un área de trabajo es la conexión entre el cableado horizontal y el equipo que está corriendo aplicaciones que utilizan la transacción de voz, video, datos o control (Figura 2.15).

Algunas recomendaciones para la instalación final de la estación de trabajo son:

- La roseta que se encuentra en el muro y es la conexión entre el cableado horizontal y el equipo final debe contener tres conectores: uno para voz, uno para video y datos y el último debe ser un respaldo para video y datos.
- Los conectores deben ser de rosetas tipo hembras para la conexión del patch cord que se conecta al equipo final.
  a. La longitud máxima del patch cord debe ser de 5 metros.
  b. La ubicación de la roseta de conexiones deberá estar alojada a la mitad del cuarto, por posibles reubicaciones futuras.

Estas cuatro recomendaciones mencionadas han sido integradas dentro de la norma de construcción comercial para espacios y recorridos de telecomunicaciones ANSI/EIA/TIA 569 A.

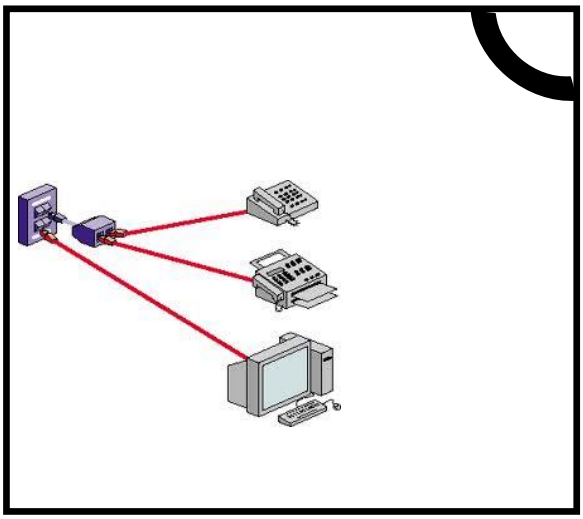


Figura 2.15. Muestra de un área de trabajo.

2.1.6 Entrada de servicios

Para que la red se conecte al internet, se hará por la denominada entrada de servicios que no es más que el Gateway (interfaz lógica/física capaz de dotar a todos los equipos conectados a esta de conexión con un dispositivo de red externo) de la organización o edificio hacia internet. El ISP o proveedor de servicios puede proveer a la compañía con un enlace el cual lo conecte a internet ya sea por sucursal o un enlace dedicado para toda la compañía. Pero eso será tomado en cuenta por los administradores de la red, ya que es necesario que se esté informado de cuál será el tráfico neto hacia el exterior, así como de las posibles amenazas que pueda correr la información que viaja a través de la red. Cuestiones como ésta son de gran importancia cuando se está realizando la instalación de un cableado estructurado para una red de datos.

2.2 Normas

Algunas de las normas que se utilizan en la industria para la instalación de un correcto cableado estructurado puede indagar más a fondo en los estándares emitidos como:

a. **ANSI/TIA/EIA 568 B:** Cableado de Telecomunicaciones en Edificios Comerciales. Compuesta por:
   - **TIA/EIA 568 B1:** Requerimientos Generales.
   - **TIA/EIA 568 B2:** Componentes de cableado mediante par trenzado balanceado.
   - **TIA/EIA 568 B3:** Componentes de Cableado, fibra óptica.

El propósito de esta norma es establecer un estándar de cableado genérico con el fin de apoyar un entorno en donde puedan competir múltiples proveedores de servicios, también ayuda a establecer la planificación e instalación de una estructura para edificios comerciales.

b. **ANSI/TIA/EIA 569 A:** Normas de recorridos y espacios de Telecomunicaciones en Edificios Comerciales.

Esta norma tiene como objetivo garantizar la operatividad, flexibilidad, administración y longevidad de los sistemas de telecomunicación, los cuales abarcan la transmisión de datos, voz y video. También describe los elementos de diseño arquitectónico de las especificaciones de las rutas y salas dedicadas a equipos de telecomunicaciones.

c. **ANSI/TIA/EIA 570 A:** Normas de Infraestructura Residencial de Telecomunicaciones.

d. **ANSI/TIA/EIA 606 A:** Normas de Administración de Infraestructura de sistemas de puesta a tierra de Telecomunicaciones en Edificios Comerciales.

e. **ANSI/TIA/EIA 607:** Requerimientos para instalaciones de sistemas de puesta a tierra en Telecomunicaciones en Edificios Comerciales.

El objetivo principal de este protocolo es proporcionar la orientación necesaria para la correcta conexión a tierra de la infraestructura de telecomunicaciones, con el fin de realizar una trayectoria eléctricamente conductora para asegurar la continuidad eléctrica y la capacidad de conducir con seguridad cualquier corriente que pudiera ser dañina para los equipos de telecomunicaciones.

f. **ANSI/TIA/EIA 758:** Norma Cliente – Propietario de cableado de Planta Externa de Telecomunicaciones.

Todas estas normas mencionadas deben tomarse en cuenta al momento de realizar la planeación para la instalación de una red. Muchas de las instalaciones de redes de datos no cuentan con una planeación ni estandarización correcta, por lo que todo esto eleva el riesgo de que haya fallas en el futuro y con ello un aumento en el costo final para una red de datos funcional.

2.3 Modelos de protocolos

Existen dos tipos básicos de modelos de red los cuales hacen posible la interacción entre los usuarios, la red y la información que viaja a través de ésta.

Estos modelos son conocidos como modelos de referencia debido a que ayudan a mantener una consistencia en todos los tipos de protocolos y los servicios de la red. Éstos no están diseñados para proporcionar un nivel de detalle suficiente para definir de forma precisa los servicios de la arquitectura de la red. Los modelos de referencia se subdividen para realizar funciones específicas dentro de cada una de sus capas. Los modelos más conocidos alrededor del mundo son el modelo OSI, el cual se utiliza para diseñar las redes de datos, especificaciones de funcionamiento y la resolución de problemas de la

red, que bien podría decirse que es un modelo el cual es visto más teóricamente, pues el modelo de referencia usado a nivel práctico en la mayoría de las redes alrededor del mundo es el modelo definido por TCP/IP (Figura 2.16).

Figura 2.16. Modelos de referencia, OSI y TCP/IP.

2.3.1 Modelo TCP/IP

El modelo TCP/IP define cuatro categorías de funciones que deben tener lugar para que las comunicaciones entre los dispositivos funcionen de manera correcta. Este conjunto de protocolos deben ser ejecutados en los hosts que implementan el circuito de comunicación (Figura 2.16). Un proceso de comunicación que se usa normalmente con el modelo TCP/IP, consta de una serie de pasos que se ejecutan tanto en el transmisor como en el receptor. Estos pasos ocurren de manera inversa cuando son ejecutados por el host receptor (Figura 2.17).

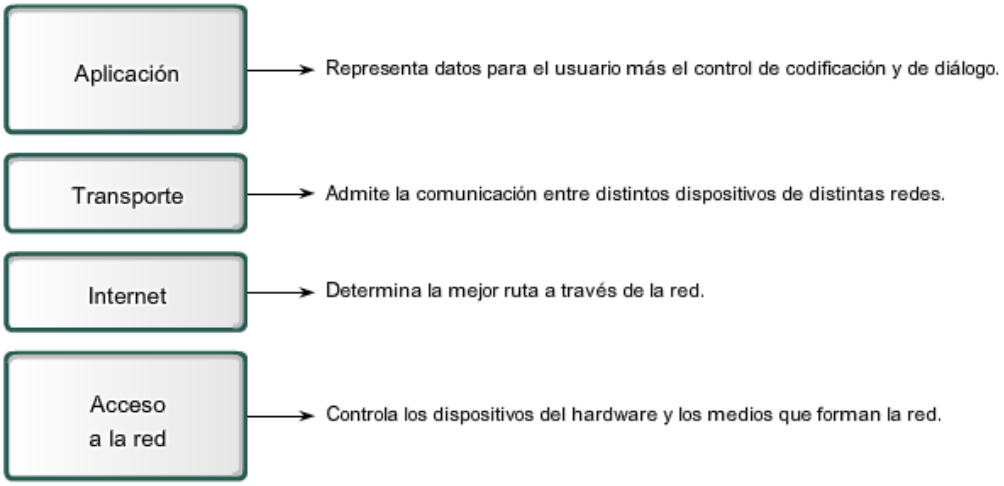


Figura 2.17. Modelo TCP/IP

Los pasos que se implementan en TCP/IP son (Figura 2.18):

1. Los datos se crean en la capa de aplicación del modelo, donde el usuario interactúa con el equipo y con los datos que éste crea o modifica, según los diferentes tipos de datos que viajen a través de la red.
2. La capa de transporte crea el circuito que existirá entre los diferentes dispositivos (emisor y receptor) que interactúan en el envío de la información a través de la red.
3. La capa de internet determina la mejor ruta por la cual viajan los datos a través de los distintos dispositivos que conforman la red de datos.
4. La capa de Acceso de Datos se encarga de controlar los distintos dispositivos de hardware que conforman la red.

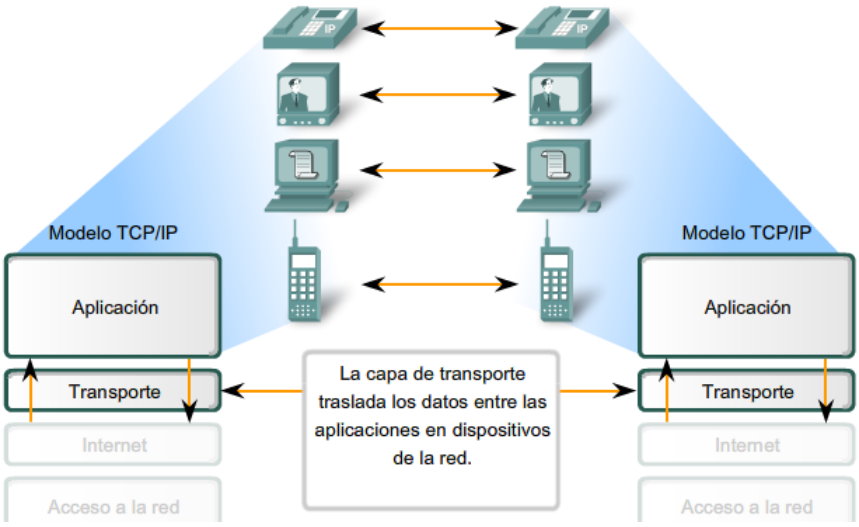


Figura 2.18. Interacción entre aplicaciones por medio de TCP/IP.

Durante el proceso de comunicación de TCP/IP la información se encapsula en cada una de estas capas agregando información necesaria para que cada una de ellas pueda realizar su tarea específica durante la interacción entre los dispositivos. A medida que la información pasa de una a capa a otra, ésta va cambiando de nombre, durante la interacción del usuario con las aplicaciones éstos se denominan datos cuando están en la capa de aplicación, cuando los datos pasan a la capa de transporte éstos se vuelven a encapsular y se le agrega información llamándolo segmento, cuando el segmento pasa a la capa de internet se le vuelve agregar información además de la que se incluyó en la capa anterior para llamarlo paquete, para que cuando al pasar el paquete a la capa de acceso a la red se le agregue más información de nueva cuenta para que a todo este conjunto de información se le denomine trama. En el momento en el que la trama pasa al medio físico se le conoce como bits. Todo este proceso que se acaba de describir se le conoce como "*transmisión del stack*".

Cuando el stack completo alcanza al host de destino, éste comienza a ser desempaquetado a través de las mismas capas, solamente que el stack comienza a ser desempaquetado por la capa de acceso a la red y así sucesivamente mientras pasa por cada una de las capas del protocolo inversamente a como se empaquetó en el proceso de transmisión.

Este modelo de referencia es el que se utiliza actualmente en todas las comunicaciones entre los dispositivos de la red, existiendo dispositivos específicos que trabajan en una capa para que la comunicación entre los dispositivos finales sea mucho más rápida.

2.3.2 Modelo OSI

El otro protocolo de referencia fue diseñado por la ISO, como un estándar que funcionara como un marco teórico sobre el cual las empresas dedicadas se basaran en una serie de protocolos =que no estuvieran bajo dependencia de ninguna organización privada.

El protocolo OSI cuenta con una serie de funciones separadas por capas que al igual que el protocolo TCP/IP realizan funciones específicas y agregan una serie de información para que viaje de un host a otro y que el proceso de comunicación sea eficaz y el mismo en ambos dispositivos. Debido a la velocidad con la que se implementó el protocolo TCP/IP en el internet y a las proporciones en las que fue implementado, el protocolo OSI, así como su desarrollo e implementación se fue rezagando, quedando solo como un marco teórico para el adiestramiento de profesionales en el campo de las redes.

Las capas que conforman este protocolo son:

a. Aplicación o capa 7

Esta capa proporciona la interfaz en la que el usuario interactúa con las aplicaciones que utiliza día a día. Los protocolos de la capa de aplicación utilizan datos entre los programas que son ejecutados en los hosts de origen y destino.

Algunas de las aplicaciones o protocolos que se utilizan comúnmente en esta capa son denominados servicios (Figura 2.19), estos servicios pueden ser páginas de internet (Protocolo HTTP), servicios de

nombres de domino (protocolo DNS), transferencia de datos o archivos (protocolo FTP) hasta envío de correos electrónicos (protocolo SMTP o POP). Estos protocolos son los más utilizados por los usuarios y cuentan con funciones específicas, los cuales los diferencian entre cada uno de ellos.

Figura 2.19. Capa de aplicaciones

b. Presentación o capa 6

La capa de presentación puede no ser utilizada dentro del stack, ya que sus implementaciones no se vinculan con los protocolos, por lo que la capa de presentación tiene tres funciones muy importantes:

- La capa de presentación codifica y convierte los datos de la capa de aplicación para garantizar que éstos puedan ser interpretados por el dispositivo de destino.
- Comprime los datos que provienen de la capa de aplicación para que pueden ser descomprimidos por el dispositivo de destino.
- Y por último cifra los datos que serán transmitidos, para que el dispositivo de destino los pueda descifrar.

Debido a las funciones antes mencionadas, es posible que tanto el dispositivo transmisor como receptor puedan entender los datos que se están enviando y así mostrarlos al usuario, dependiendo si es un correo, una imagen o inclusive un documento de texto. Es por eso que los dispositivos llegan a conocer el contenido de los datos y diferenciar entre imágenes de tipo .gif, .jpeg o .bmp.

c. Sesión o capa 5

Esta capa es la encargada de iniciar la comunicación con el dispositivo de destino, ya que comienza notificando al dispositivo de destino que hay datos y que están por enviarse, por lo cual necesita una respuesta para que el destino esté consciente de que se enviará dicha información y de qué tipo es. Por lo tanto, la capa de sesión es la encargada de manejar el intercambio de información para iniciar la transferencia y mantener el enlace activo, al igual que reiniciar la comunicación en caso de que se haya interrumpido o desactivado durante un periodo de tiempo prolongado.

d. Transporte o capa 4

La capa de transporte permite la segmentación de los datos que se están enviando a través del medio en paquetes más pequeños, esto con la finalidad de que la red no sufra una congestión mientras se envía la información por el medio y estos segmentos puedan ser ensamblados de nuevo en el dispositivo de destino.

La división de los datos en segmentos más pequeños, así como el envío de éstos desde el origen hasta el destino permiten que se pueda realizar una multiplexación del canal. Sin la segmentación que la capa de transporte realiza sería imposible llegar a visualizar correos, páginas web y video si es que los tres viajan a través de la red, debido a que todo el ancho de banda del canal sería utilizado por la transmisión del video.

Otras de las funciones importantes que debe realizar la capa de trasporte es el seguimiento de la comunicación entre las aplicaciones entre el dispositivo de origen y el de destino. Por lo que también es importante que la capa de transporte identifique las diferentes aplicaciones entre los dispositivos. Para eso la capa de transporte se ayuda de una serie de protocolos e información para poder llevar a cabo su tarea.

Para que la capa de trasporte tenga un control sobre las comunicaciones es necesario que ésta establezca una sesión entre las aplicaciones para que se comuniquen entre sí antes de que se transmitan los datos. También es necesario que se establezca una entrega confiable y asegurar que todos los segmentos en los que se dividió el paquete lleguen a su destino, en caso de que haga falta un segmento del paquete, éste se pueda volver a transmitir por si se perdió o se volvió corrupto en el lapso entre el dispositivo de origen y el destino.

Otra de las ventajas con las que cuenta la capa de transporte es el control de flujo de la red, ya que si ésta detecta que hay demasiados segmentos en la red o que los recursos de la red se encuentran sobrecargados, solicita a la aplicación que se encuentra enviando segmentos por el canal, reduzca el flujo de datos en lo que se estabiliza la cantidad de segmentos que viajan por el canal. Este control de flujo ayuda a prevenir que se pierdan segmentos en el canal y evitar que tengan que volver a transmitirse.

La capa de transporte se ayuda de dos protocolos para poder realizar todas las tareas que se mencionaron. Estos protocolos son:

1. **TCP** (Protocolo de Control de Transmisión)

Este protocolo es orientado a conexión, eso quiere decir que es necesario establecer la comunicación antes de que se envíen los datos entre los dispositivos de origen y destino, por lo tanto se tiene la necesidad de utilizar recursos adicionales para el agregado de funciones que ayudan a que se realice una entrega confiable y un buen control de flujo entre los dispositivos. Este protocolo necesita un acuse de recibo por parte del dispositivo de destino que le informe que todos los segmentos del paquete llegaron correctamente y de manera veraz para que la comunicación entre ambos pueda terminar (Figura 2.20). Se ayuda agregando encabezados al segmento de TCP incluyendo información como el

número de secuencia y el número de acuse de recibo, para que la entrega de los segmentos sea correcta y el dispositivo de destino pueda volver a integrar los segmentos en el paquete original gracias al número de secuencia con el que el dispositivo de origen los identificó.

Las aplicaciones que utilizan TCP son:

- Exploradores web
- Correos electrónicos
- Transferencia de archivos

Bit (0)	Bit (15)	Bit (16)	Bit (31)
Puerto de origen (16)		Puerto de destino (16)	
Número de secuencia (32)			
Número de acuse de recibo (32)			
Longitud del encabezado (4) Reservado (6) Bits de código (6)		Ventana (16)	
Checksum (16)		Urgente (16)	
Opciones (0 ó 32 si las hay)			
DATOS DE LA CAPA DE APLICACIÓN (el tamaño varía)			

Figura 2.20. Segmento TCP y UDP

2. **UDP** (Protocolo de Datagramas de Usuario)

UDP es un protocolo más simple que TCP, ya que no es orientado a conexión y cuenta con la ventaja de proveer gran cantidad de datos sin utilizar demasiados recursos. Debido a que es un protocolo no orientado a conexión, envía los datagramas denominados como mejor intento (Figura 2.21), por lo que el datagrama llega a ser mucho más pequeño que un segmento de TCP.

Algunas de las aplicaciones que utilizan UDP son:

- Sistemas de nombres de dominio
- Streaming (envío contínuo de datos dentro de una red de datos a través del protocolo UTP) de video
- Voz sobre IP

Bit (0)	Bit (15)	Bit (16)	Bit (31)
Puerto de origen (16)		Puerto de destino (16)	
Longitud (16)		Checksum (16)	
DATOS DE LA CAPA DE APLICACIÓN (el tamaño varía)			

Figura 2.21. Datagrama de UDP

Ambos protocolos se guían a través de encabezados que son agregados a los segmentos, además de los encabezados que cada protocolo agrega por su parte. Estos encabezados definen el puerto de origen así como el de destino y un checksum (es una función hash que tiene como propósito detectar cambios en una secuencia de bits, con el fin de proteger la integridad de los datos) que ayuda a verificar que el segmento no ha sido modificado. El encabezado que determina el puerto, ayuda a los dispositivos de entrada como de salida a identificar qué tipo de protocolo se está utilizando debido a que existe una serie de puertos determinados para cada tipo de servicio.

Estos puertos designados por la IANA (Internet Assigned Numbers Authority - Autoridad de números asignados de internet) y ayudan a que el modelo de direccionamiento sea más eficaz dentro de la red.

Existen distintos tipos de puertos:

- **Puertos bien conocidos:** estos puertos están designados en un rango que va de 0 hasta el 1023 y están reservados para servicios y aplicaciones del servidor, así las conexiones que provengan de aplicaciones del cliente son programadas para solicitar la conexión a un puerto en específico y el servicio asociado a éste.
- **Puertos registrados:** estos puertos se encuentran dentro del rango de 1024 a 49151, y se encuentran reservados para los procesos o aplicaciones que corren en el cliente. Estos puertos corren principalmente aplicaciones individuales que son instaladas en el dispositivo del usuario, las cuales al realizar una petición a un servicio lo harán por medio de un puerto bien conocido. Además de que estos puertos pueden ser utilizados de manera dinámica si son seleccionados por el usuario para realizar una petición a través de otro puerto.
- **Puertos dinámicos o privados:** el rango en el que se encuentran estos puertos va desde el 49152 hasta el 65535. Estos puertos suelen asignarse de manera dinámica entre aplicaciones que inician una conexión que sea punto a punto, ya que no es muy común que un cliente se conecte a un servicio utilizando uno de estos puertos.

La capa de transporte hace uso de todos los encabezados y ayuda a que los paquetes lleguen a su destino gracias a los protocolos que define, además de que logra identificar los servicios que se solicitan por medio del encabezado que definen los puertos de origen y destino. Sin todo este conjunto de protocolos y encabezados los paquetes no llegarían a su destino, y en caso de que llegaran no sería posible visualizar la información que se transmitió puesto que no se tendría idea de qué tipo de información es y por qué puerto entró y salió (Figura 2.22). Es por eso que la capa de transporte está definida en el modelo de referencia de OSI como en el de TCP/IP, mientras que capas superiores como la de sesión y presentación son englobadas en la capa de aplicación, ya que aunque tienen funciones definidas éstas pueden ser omitidas sin que perjudique la comunicación entre los dispositivos.

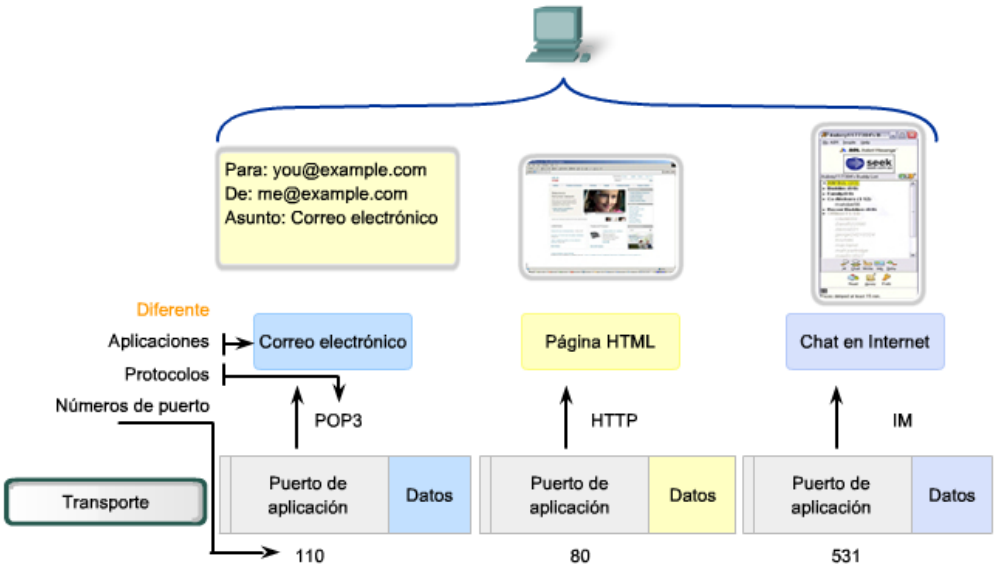


Figura 2.22. Capa de trasnporte

e. Red o capa 3

La capa de red provee servicios para transmitir los paquetes a través de la red entre los dispositivos finales identificados. Para que esto sea posible la capa de transporte provee un mecanismo para direccionar estos dispositivos finales. Ambos dispositivos cuentan con una dirección única, por lo que la capa 3 recibe el segmento proveniente de la capa 4 para agregarle un encabezado que debe contener la dirección del host de destino y otro para la dirección del host de origen. A este paso se le denomina encapsulación.

Estas direcciones son utilizadas por el enrutador que interconecta las redes para leer dichos encabezados y así enviar los paquetes por la ruta correcta hacia su destino.

Los protocolos de la capa de red son:

- IPv4
- IPv6
- IPX
- AppleTalk
- CLNS/DECNet

Para que la capa de red haga posibles todas sus funciones, agrega una serie de encabezados a la PDU del paquete que el enrutador identifica y utilizará para direccionar el paquete por la red correcta. Estos encabezados son:

1. Dirección IP

La dirección de origen como la de destino representa la dirección de host, la cual contiene un valor binario de 32 bits, divididos en 4 octetos.

2. TTL (Time To Live - tiempo de vida)

El tiempo de vida es un valor el cual representa el tiempo remanente de vida del paquete y éste se encuentra formado por un valor de 8 bits. Es de gran ayuda porque va restando su valor en uno cada vez que pasa por un enrutador, esto ayuda a evitar bucles dentro de la red al momento que el TTL de un paquete es igual a 0, éste simplemente es descartado de la red.

3. ToS (Tipe of Service–Tipo de servicio)

El tipo de servicio es utilizado para determinar la prioridad de cada paquete

4. Protocolo

El campo de protocolo determina qué protocolo de transporte se está utilizando. La capa de red se apoya de todos estos campos para que el enrutador conozca porque interfaz enviará el paquete. Esto es posible gracias a que el enrutador des encapsula y encapsula de nueva cuenta los paquetes para tener acceso a los campos que le ayudan a determinar la mejor ruta.

f. Enlace o capa 2

La capa de enlace de datos brinda una serie de servicios que hacen posible el intercambio de datos a través de medios locales comunes. Permite a las capas superiores acceder a los medios usando una serie de técnicas que controlan la forma en la que los datos son colocados en el medio, utilizando otras técnicas para la detección de errores.

En esta capa los datos son preparados agregando campos para transmitirlos al canal, los datos no necesitan saber por qué medio serán enviados, por lo tanto solo necesitan información de hacia dónde se dirigen, estos campos agregados serán identificados e interpretados por las NIC´s. La información que es agregada a las tramas de datos son:

- Datos: es el paquete que se creó en la capa de red.
- Encabezado: contiene información del control de direccionamiento.
- Trailer: contiene la información de control agregada al final del paquete.

Cuando la capa de enlace de datos coloca las tramas en el medio, lo hace en forma de bits esto implica un problema ya que un host, en este caso el host de destino, no sabe en qué momento termina una cierta trama y en qué momento empieza otra.

Para solucionar este problema los ingenieros agregaron a la trama una serie de campos que ayudan a que la NIC determine en qué momento termina un stream de datos y que otro comienza una trama durante un flujo continuo de información. Esto es posible gracias a los campos indicadores de comienzo y detección, los cuales marca el correcto comienzo y la finalización de una trama. Otro campo que ayuda a la NIC es el campo tipo, este campo determina el tipo de paquete que esta contenido dentro de la trama. Y por último el campo de calidad, el cual sirve como ayuda para el control de la trama. Todos estos campos son agregados al paquete que es encapsulado en la capa de transporte para preparar a las tramas y colocarlas en el medio.

Debido a la específica tarea de la capa de enlace de datos, se optó por subdividir las funciones que se realizan en este nivel. Estas subcapas realizan el control de los datos y el medio que hacen posible que los paquetes que provienen de la capa de red sean puestos en el medio físico como cadenas interminables de bits. Esta capa se subdividió en dos capas las cuales tienen funciones muy diferentes debido a las características con las que cuenta cada una.

Estas subcapas son:

- **Control de enlace lógico (LLC):** coloca la información en la trama que identifica el protocolo que se utiliza en la transmisión de datos con un identificador al inicio de la trama.
- **Control de Acceso al Medio (MAC):** proporciona a la capa de enlace de datos el direccionamiento necesario para las tramas a través de las tarjetas de red, además de una delimitación de las tramas que viajan a través de éstas.

Toda esta información es identificada por los dispositivos que trabajan con los campos de información de la capa de enlace de datos como conmutadores y tarjetas de red que direccionan las tramas por el medio con base en las direcciones MAC de origen y destino con las que fueron encapsuladas las tramas al momento de ser puestas en el medio. Esto se debe a que los dispositivos que trabajan en las capas más bajas del modelo OSI realizan las tareas con mayor velocidad y menos carga en el trabajo de encapsulación y des encapsulación de las tramas.

g. Física o capa 1

La física es el medio por el cual viajaran los datos en forma de bits, estas interminables cadenas de bits de 1's y 0's hacen uso de los medios físicos y no físicos que fueron descritos en el capítulo pasado. Estos bits no son más que una combinación de pulsos electromagnéticos donde un 1 representa un voltaje en el medio y un 0 representa un voltaje nulo en el medio.

Estas cadenas de unos y ceros pueden ser transmitidos a grandes distancias en forma de luz, un pulso eléctrico o una onda de radio.

Capítulo 3:
Routers

“Siempre ten en mente que tu propia resolución de triunfar es más importante que cualquier otra”

-Abraham Lincoln-

Un *enrutado*r (router) es un dispositivo físico capaz de encaminar enormes cantidades de paquetes de datos a través de una red. Este dispositivo cuenta con características específicas, las cuales lo llegan a diferenciar de una computadora común.

A lo largo de los años los ingenieros que administraban las redes de datos se dieron cuenta que los equipos de los cuartos de telecomunicaciones necesitaban ciertas especificaciones de hardware así como de software, las cuales ayudarían a tener un mayor rendimiento en cuanto a las necesidades de la red. Fue así como con el transcurrir de los años se fueron modificando los equipos de enrutamiento, pasaron de un enorme armario que podía estar en una sala ambientada hasta los enrutadores de alta velocidad, que tienen un diseño delgado y resistente a las características del ambiente donde se encuentren.

Los ingenieros no solo fueron modificando el hardware de los equipos de aquellos años, conforme la tecnología avanzaba a través de los años, éstos realizaban cada vez funciones más complejas, implementando software especial y funciones específicas, esto se fue dando en las modificaciones tanto en el software como en el hardware.

Los procesadores que son la parte neurálgica de cualquier dispositivo electrónico, han ido evolucionando casi a la par de todas las tecnologías, obviamente en cada una de las ramas las mejoras suceden dentro de un tiempo mucho más corto en comparación de los avances tecnológicos que se han registrado en años anteriores. Hoy es posible adquirir un procesador de 16 núcleos capaz de realizar millones de operaciones en solo una fracción de segundo, siendo ésta una de las características primordiales de un enrutador, que sea capaz de enrutar millones de paquetes que viajan por segundo por toda la red. Realizando operaciones básicas con todos los paquetes que son enrutados por cada una de sus interfaces, entregando los paquetes al destino correcto y de manera confiable.

3.1. Elementos de un enrutador

Un enrutador cuenta con 7 elementos importantes, éstos ayudan a que el dispositivo lea las cabeceras iniciales y conozca el origen y el destino del paquete, utilice diferentes protocolos de enrutamiento en las diversas interfaces que conforman el hardware y que contenga una tabla actualizada de todos los segmentos de red que están directamente conectados a sus interfaces de alta velocidad.

A través de este capítulo se explica porque cada uno de estos elementos son esenciales para las tareas que realiza un enrutador.

Los elementos que lo componen son:

a) CPU (Central Process Unit)

La unidad central de procesamiento está conformada por las mismas partes que se encuentran en una computadora, exceptuando algunas tarjetas o ranuras para su utilización en el enrutamiento de paquetes.

Uno de los componentes principales es el procesador, éste es el cerebro del enrutador, realiza todas las operaciones necesarias para encaminar, encapsular o des-encapsular los paquetes de datos, además de iniciar el sistema operativo que controla todos los elementos del enrutador. Los procesadores varían

según las capacidades y el desempeño de un enrutador, por lo que se ha optado por equipar a los enrutadores con una serie de procesadores específicos, dos de estos ejemplos son el Motorola 68030 y el Orion/R4600. Procesadores como estos dos ejemplos, requieren una arquitectura capaz de realizar operaciones con el fin de que el enrutador tome decisiones sobre enrutamiento y puenteo, además de mantener las tablas de enrutamiento y otras funciones de administración del sistema.

El Motorola 68030 es un microprocesador de arquitectura de 32 bits de la familia Motorola 68000, entre sus características incluye una memoria caché dividida en 256 bytes para las instrucciones de 256 bytes de datos. Su micro-arquitectura está formada básicamente por la memoria caché la cual no incrementa en mucho las características de sus antecesores.

Por otro lado, la arquitectura del Orion R4600 incluye arquitecturas de 32 y 64 bits, cuenta con una memoria caché de nivel 2, una versión mejorada de las que utilizaba Motorola en procesadores anteriores. Debido a su arquitectura, puede operar a velocidades de reloj de hasta 50MHz para obtener un ancho de banda máximo de 400MB/s.

Otras características con las que cuenta la unidad central de procesamiento son la memoria caché, esta memoria es de alta velocidad, mejorando la velocidad de lectura de la memoria RAM, por la que el equipo mejora su rendimiento al leer y escribir instrucciones en la memoria caché, en vez de acceder a la memoria RAM en busca de las instrucciones necesarias para la ejecución de las instrucciones.

Finalmente estos dos componentes son alojados en la tarjeta madre junto con los otros componentes del enrutador así como la memoria RAM, ranuras de expansión, la memoria caché y el BIOS.

b) ROM (Read Only Memory – Memoria de solo lectura)

La memoria ROM se encuentra dentro de la tarjeta madre, ésta es utilizada para almacenar información que es ejecutada por el sistema, esta información no será borrada del sistema cada vez que éste sea reiniciado, tenga alguna falla eléctrica ó deba volver a acceder al sistema. Toda la información que se guarde en una memoria ROM, podrá ser almacenada en unidades de disco de alta capacidad, hoy en día existen unidades de almacenamiento de hasta 1 TB. Toda la información almacenada en estas unidades de memoria no puede ser modificada, por lo cual esta característica es indispensable en el momento de iniciar el sistema operativo en los enrutadores.

La memoria ROM es un tipo de firmware que se encuentra dentro de un circuito programado con una serie de instrucciones específicas que ayudan a controlar los circuitos electrónicos. Estas instrucciones establecen una lógica de bajo nivel para que el enrutador pueda preparar un entorno capaz de cargar el sistema operativo.

Existen distintos tipos básicos de memoria ROM, cada una con sus características específicas. Aunque todas las memorias ROM cuentan con dos características en común.

1. Son memorias no volátiles, lo cual significa, que la información almacenada en ellas no se perderá, aún cuando se le haya eliminado la fuente de energía.
2. La otra característica más importante es que la información almacenada en estas memorias no puede modificarse, si se requiere modificar alguna información se sigue una serie de pasos

especiales para poder editar la información contenida en ellas. Esta característica es la que brinda la oportunidad de cargar un sistema operativo en el dispositivo, ya que las instrucciones al estar precargadas no hay necesidad de insertarlas nuevamente, esto ahorra tiempo y mejora el desempeño de los dispositivos eléctricos.

Los tipos de memoria ROM son:

1. Memoria ROM, está integrada por una serie de malla hecha por columnas y filas, esta malla contiene diodos en cada una de las intersecciones. Si la conexión está activa entre la columna y la fila, la memoria ROM tendrá el valor de uno en esa posición, mientras que si dicha conexión no se encuentra activa tendrá el valor de cero. A esta conexión entre filas y columnas se le denomina celda, por lo que si en el diodo existe una carga que pase por la celda hacia tierra, entonces esa carga será tomada como un valor positivo, en este caso el valor de uno en un sistema binario.

2. PROM (Programmable Read Only Memory – Memoria de solo lectura programable) cuenta con una malla y celdas, formando el circuito ideal para la lectura de instrucciones en un sistema hexadecimal. A diferencia de una memoria ROM común, las PROM no están fabricadas con diodos en sus celdas. Este tipo de memoria cuenta con fusibles en cada una de sus celdas, el fusible toma el valor binario de uno, por lo que el estado inicial de todas las celdas en la memoria PROM es uno. Para programar esta memoria sólo se necesita una cantidad de corriente que pase por la celda para que al pasar por ella rompa el fusible de la misma. Esto hace que la celda tome inmediatamente el valor de cero por donde la corriente desactiva la conexión quemando el fusible.
   Debido a que este proceso sólo se puede hacer una sola vez, las memorias PROM no se pueden volver a usar si la información contenida en ellas es dañada por algún cambio de corriente el cual haga que algún otro fusible se queme.

3. EPROM (Eraseable Programmable Read Only Memory – Memoria de sólo lectura programable/borrable) a diferencia de los dos tipos de memoria ROM antes mencionados, las memorias EPROM pueden borrar la información que se encuentra almacenada en ella con el fin de reprogramarla. Para poder realizar el proceso de borrar y escribir nuevamente, se necesita de una herramienta adicional, la cual debe emitir una luz ultravioleta a una cierta frecuencia. Dentro de la malla de una memoria EPROM, cada celda cuenta con dos transistores, los cuales están separados entre sí por una delgada capa de óxido.
   A estos transistores se les conoce como floating gate (puerta flotante) y control gate (puerta de control), al igual que una memoria PROM, los valores iniciales de cada una de las celdas es de uno, esto quiere decir que existe una conexión entre los transistores, para poder obtener el valor de cero en la celda, se ingenió un proceso denominado *Tunneling* (túneles). Este proceso es utilizado para modificar los electrones contenidos dentro del floating gate, el cual, cuando al hacer pasar un cierto voltaje por la celda, se busca que éste vaya por el camino más corto a tierra, haciendo que los transistores se separen entre ellos y así poder abrir el circuito.

Para borrar una memoria EPROM, se utiliza el dispositivo de luz ultravioleta (antes mencionado) que suministra un nivel de energía capaz de hacer que el componente floating gate regrese a su estado inicial. Dicha fuente de luz debe estar a una frecuencia y un determinado periodo de tiempo, específicos para que el proceso de borrado de la EPROM sea exitoso. En dado caso de que la memoria sea sobre-expuesta a la luz ultravioleta, ésta puede cargar las partículas del floating gate hasta el punto en que los electrones no puedan volver a tener la carga necesaria para realizar la conexión en la celda.

Por lo tanto, el enrutador utiliza la memoria ROM para almacenar en ella el bootstrap (negociación de arranque), el cual es invocado cada vez que el enrutador es encendido e inicia la secuencia de arranque. Una vez que el enrutador se enciende, se comienza a ejecutar el código almacenado en la dirección F000:FFF0, el cual pertenece al BIOS almacenado en la ROM, este proceso se lleva a cabo para realizar una serie de pruebas al sistema e iniciar la secuencia de arranque del mismo. Una vez que el BIOS termina de ejecutarse, éste carga el primer sector en la dirección 000:7C00 y comprueba que la imagen del sistema operativo sea válida.

c) Flash memory

La memoria Flash es una memoria de tipo ROM, los ingenieros la clasifican dentro de las memorias EEPROM (Electronically Erasable Programmable Read Only Memory – Memoria de Solo Lectura programable, borrable electrónicamente), por lo tanto, al ser una memoria de tipo ROM, cuenta con una malla formada por columnas y filas en las que cada intersección se le conoce como celda, como se ha definido anteriormente. A diferencia de las memorias EPROM, para modificar el contenido de la memoria no es necesario la utilización de herramientas adicionales como el emisor de luz ultravioleta, ya que este tipo de memoria utiliza la corriente que pasa a través de la celda para poder modificar la información que se encuentra almacenada en cada celda.

Del mismo modo que las memorias EPROM, la memoria flash cuenta con dos transistores separados uno del otro por una delgada capa de óxido. La memoria flash cuenta con dos transistores como control gate y floating gate (Figura 3.1). El estado inicial de los transistores es de uno, por lo tanto, se utiliza el mismo proceso de tunneling para cambiar el estado a cero.

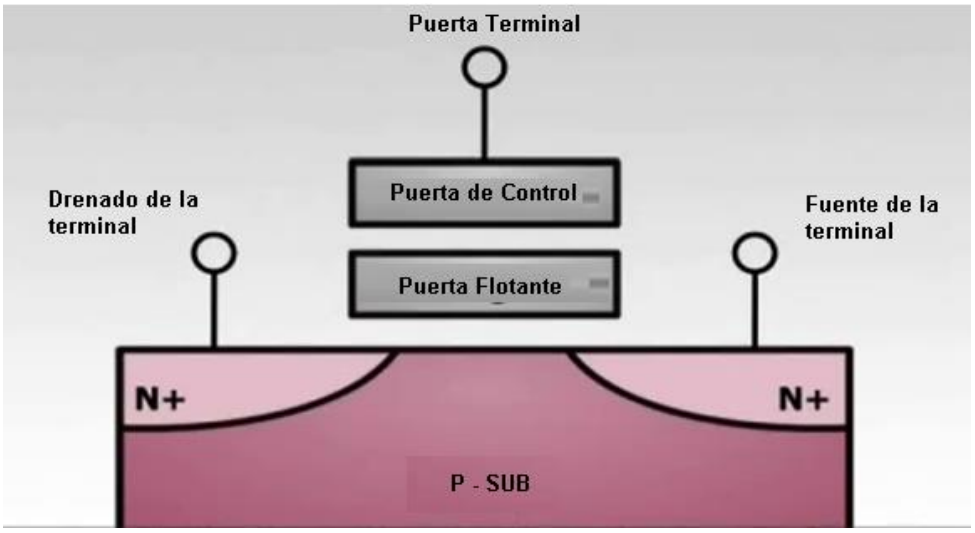


Figura 3.1. Celda de una Memoria Flash.

Para el cambio de valor en la celda se utilizan elementos como el tunneling para modificar el estado de los electrones en el floating gate. Pero a diferencia de utilizar la luz ultravioleta como en las memorias EPROM, aquí se utilizan voltajes, los cuales hacen que los electrones cambien de estado (Figura 3.2).

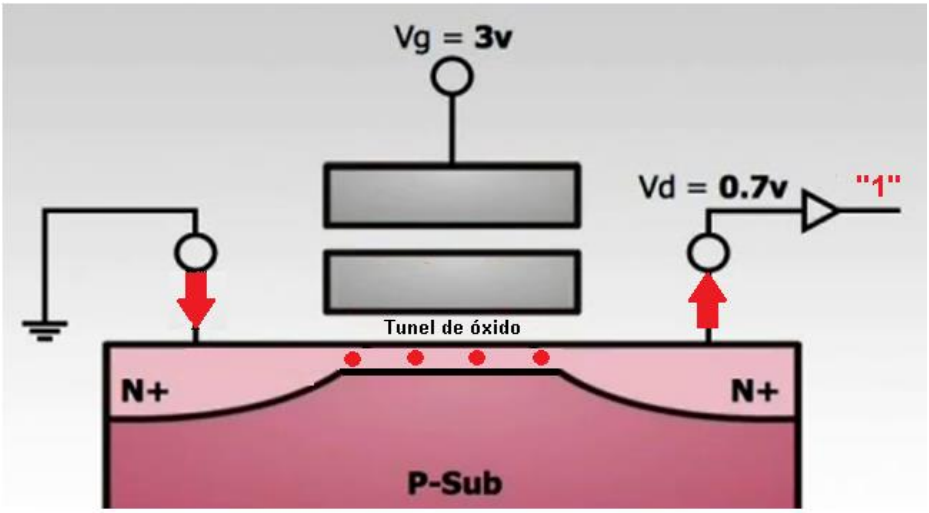


Figura 3.2. Celda con valor binario de 1.

Al hacer pasar un voltaje de 3 volts por el transistor llamado control gate, éste hace que el floating gate cambie de posición para que el voltaje fluya por el substrato que se encuentra en la celda. Por el lado contrario, para que el floating gate cambie de posición se aplica mayor voltaje al substrato (una cantidad de 0.7 volts), para que los electrones del floating gate vuelvan a hacer que modifique su posición (Figura 3.3).

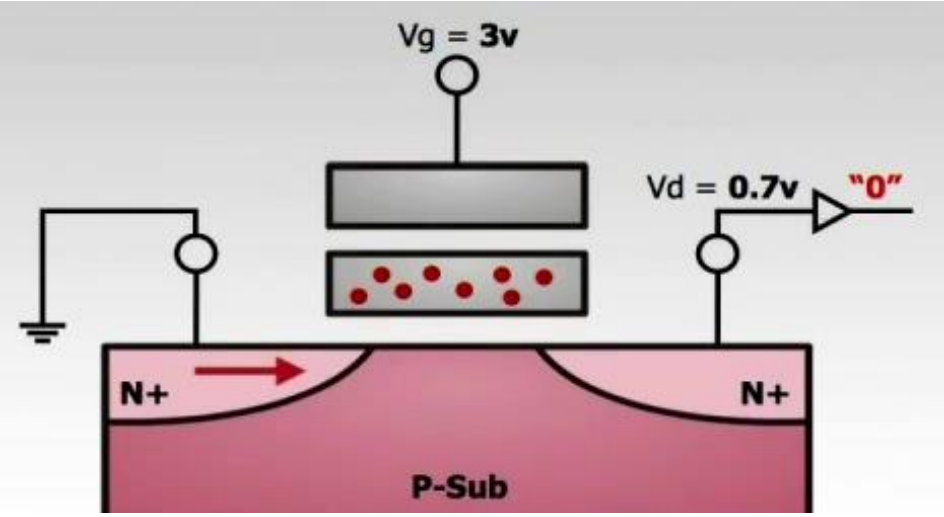


Figura 3.3. Celda con valor binario de 0.

Con este método resulta más fácil guardar información y volverla a borrar los valores de una memoria flash, ya que con sólo leer los valores de los voltajes de cada celda y aplicar voltaje a las celdas se realizan las acciones de leer, escribir y borrar dentro de una memoria EEPROM. Por otro lado, este método de Tunneling resulta ser más costoso por bit, que una unidad de disco duro, por lo que aún su precio es bastante desfavorecedor en comparación con los otros tipos de memorias en el mercado.

El enrutador utiliza la memoria flash para guardar en ella una o más imágenes del sistema operativo, por lo que una vez terminada la secuencia de arranque de la memoria ROM, todas las imágenes que se encuentren en la memoria flash (si es que hay más de una), serán verificadas para corroborar que no se encuentren dañadas y así iniciar la imagen del sistema operativo más reciente que se encuentre almacenada en ella.

d) RAM

La memoria RAM (Random Access Memory – Memoria de acceso aleatorio), es un tipo de memoria volátil, ésto quiere decir que si se le deja de suministrar corriente al circuito, perderá la información que se encuentre contenida en cada una de las celdas. Se denomina memoria de acceso aleatorio ya que con sólo saber la dirección de memoria conformada por el número de columna y fila, se accede a la celda y se lee la información contenida en ella.

Este circuito integrado está formado por millones de capacitores y transistores, en cada celda se encuentra un dueto formado por un transistor y un capacitor. El funcionamiento es un proceso sencillo, si no fuera por el constante suministro de energía que debe recibir el circuito integrado para mantener la información almacenada en la memoria RAM (Figura 3.4).

Figura 3.4. Circuitos Integrados de Memoria RAM.

El capacitor mantiene almacenado el bit de información, y esto es leído como un uno, mientras que el transistor actúa como un pequeño interruptor, el cual permite que los circuitos que controlan la memoria puedan leer y cambiar el estado del capacitor. El elemento de importancia es el capacitor, ya que si el controlador de la memoria no detecta carga en él, entonces éste asume que el valor en esa posición es de cero. Por el contrario, si el controlador detecta carga en esa posición, su valor será de uno.

Esto es posible gracias a que el capacitor sólo logra almacenar por unos instantes los electrones, por lo que en cuestión de milisegundos el capacitor se vacía. Para que el capacitor pueda guardar la carga almacenada en éste, el controlador de la memoria debe volver a recargar el capacitor antes de que éste se descargue. Por lo que el controlador realiza ciento de veces las operaciones de leer y escribir en las celdas de la memoria para no perder la información de ésta.

Las celdas de memoria contenidas en una RAM serían inútiles si no se tuviera un controlador capaz de leer su información y escribir en ellas, por lo que se buscó hacer posible una serie de funciones capaces de realizar mucho más allá que sólo leer y escribir.

Las funciones del controlador son:

a) Identificar cada fila y columna.
b) Hacer el seguimiento de la secuencia de actualización.
c) Leer y restablecer la señal de cada una de las celdas.
d) Y decirle a la celda cuándo debe cambiar de un valor a otro.

Viendo el funcionamiento de la memoria RAM, ahora es importante saber cómo es utilizada dentro de un enrutador.

El enrutador utiliza la memoria RAM para mantener algunas tablas del sistema y buffers, básicamente es utilizada para todos los requerimientos operacionales del sistema. Debido a que es mucho más rápido acceder y comparar las tablas de enrutamiento en los slots (ranuras) de RAM, que si éstas estuvieran almacenadas en la ROM. Debido a que el enrutador necesita realizar operaciones de enrutamiento,

accede a las tablas almacenadas en la RAM, compara la dirección de destino con sus tablas y enruta el paquete por la interfaz de salida. Hay otras funciones en las cuales el enrutador utiliza la memoria RAM, uno de esos casos es cuando éste se encuentra configurando con algún protocolo de enrutamiento dinámico y necesita encontrar adyacencias con vecinos y crear las tablas de enrutamiento, estas operaciones se deben realizar en el menor tiempo posible por cuestiones de configuración. Un factor que muy pocas veces es tomado en cuenta por los administradores de la red es que procesos tales como debugging incrementan el uso de la memoria RAM para realizar sus funciones, por lo que el rendimiento del enrutador se ve afectado cuando el proceso de debugging (depuración) se utiliza, por lo cual debe ser usado con cautela.

e) NVRAM

Las memorias NVRAM (Non Volatile Random Access Memory – Memoria de acceso aleatorio no volátil), son microcircuitos de memoria que trabajan de igual manera que los microcircuitos RAM. La única diferencia entre estas dos categorías de memoria está basada en que un microcircuito de memoria RAM pierde su información en el instante en que pierde su fuente de energía, como al apagar el enrutador, una NVRAM trabaja muy similar a una EEPROM, de hecho se considera que las memorias flash son un tipo de memoria NVRAM, ya que éstas trabajan de manera que su comportamiento simula a las memorias ROM, pero la forma en las que acceden a la información almacenada en ellas hace que se les denomine RAM. Por lo que una memoria NVRAM es no volátil debido a que tiene un flujo continuo de energía y sigue almacenando la información aun después de haber cortado la fuente de energía en ellas; pero sus procesos de lectura, escritura y borrado son aleatorios. Algunas otras memorias cuentan con un bloque que contiene una pequeña batería para que sigan teniendo un flujo de corriente continuo, por lo que también son clasificadas como NVRAM. Estos bloques pueden ser removibles para poder cambiar la batería que contengan (Figura 3.5).

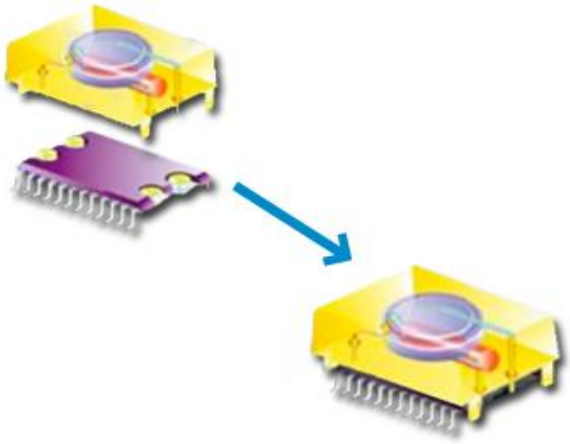


Figura 3.5. Memoria NVRAM, con batería.

La memoria NVRAM es útil porque el enrutador almacena en ésta la configuración de inicio o startup-configuration. Ésta es la configuración que va a leer el enrutador al cargar el sistema operativo para después cargar todas las instrucciones que se encuentren configuradas en ésta, tales como el nombre del enrutador, algunas rutas estáticas, algún protocolo de enrutamiento o alguna lista de acceso previamente configurada y almacenada en el registro de la configuración de inicio. El enrutador guarda la configuración de inicio en la memoria NVRAM porque es una memoria de alta velocidad y también es persistente a las pérdidas de corriente en el sistema, por lo que si éste sufre la pérdida de energía, la configuración quedará almacenada en la NVRAM.

Estos microcircuitos son una parte esencial del enrutador, ayudan a que el enrutador realice todas sus funciones a gran velocidad. Aunque también cuenta con interfaces para conectar el enrutador con otros elementos de la red, además del software que le brinda la funcionalidad necesaria.

f) Bios

La función primordial del BIOS (Basic Input/Output System – Sistema Básico de Entrada/Salida) es la de cargar el sistema operativo. Cuando se enciende una computadora el microprocesador trata de ejecutar la primera instrucción, la computadora realiza una serie de pasos para ejecutar las primeras instrucciones que cargan el sistema operativo, éste se encuentra almacenado en el disco duro del enrutador, pero las instrucciones que hacen que el enrutador cargue el sistema no pueden almacenarse en éste, ya que se haría muy lento el arranque del sistema. Por esa razón, se almacenaron las primeras instrucciones que debe ejecutar el enrutador para iniciar el sistema antes de empezar a cargar el sistema operativo. Todas estas instrucciones se almacenan en una memoria flash la cual es mucho más fácil de leer (Figura 3.6).

La secuencia que se ejecuta en el enrutador es:

1. Comprobar la configuración del CMOS (Complementary metal–oxide–semiconductor – Semiconductor complementario de óxido metálico).
2. Cargar el manejador de interrupciones y los controladores de los dispositivos.
3. Iniciar los registros del sistema.
4. Mostrar los ajustes del sistema.
5. Determinar los dispositivos que son booteables (arranque).
6. Iniciar la secuencia de bootstrap.

La primera acción que realiza el BIOS es verificar la información almacenada en la CMOS, la cual provee la información detallada al sistema para que se pueda alterar según las configuraciones que se le hayan hecho. Una vez que la BIOS haya comprobado el sistema de manera exitosa, se cargará el sistema a partir de la dirección en la cual se haya especificado.

Figura 3.6. Microcircuito BIOS, INTEL.

Un enrutador puede catalogarse como un equipo de cómputo común, pero su hardware y su software lo hacen de cierta manera tan peculiar, cuenta con la misma arquitectura que cualquier otro equipo de cómputo de uso personal. Pero las características de su procesador así como de todas sus interfaces, las cuales son de mayor velocidad que las de una computadora normal, hagan esa pequeña diferencia en un equipo de cómputo y un enrutador. También sus interfaces de conexión son mucho más rapidas que la conexión Ethernet con la que cuenta la computadora.

g) Interfaces de conexión

Por cuestiones prácticas se toman en cuenta las interfaces que conforman un equipo enrutador de la serie 2800 de Cisco, con las caracteristicas especiales con las que se estarán trabajando de aquí al final de este proyecto de tesis. Esto no es motivo para sólo enfocarse a las interfaces que son parte de este hardware, cada una de las series que conforman los equipos de enrutamiento cuentan con interfaces y puertos especiales para satisfacer las necesidades de los clientes según el rubro en el que se vaya a manejar cada equipo de enrutamiento.

Además de que los equipos cuentan con slots para agregar módulos de conexión de alta velocidad, ya sea de fibra o cable UTP. Los slots ayudan a que los técnicos que manipulan los equipos puedan agregar las interfaces que sean necesarias para la red de la empresa.

Los enrutadores cuentan con interfaces fisicas que se pueden diferenciar por grupos según las caracteríisticas de cada uno de éstos. Un enrutador de la serie 2900 cuenta con interfaces de administración, interfaces WAN e interfaces LAN (Figura 3.7).

En el modelo de enrutador Cisco Series 2900 se cuenta con dos puertos o interfaces de administración, uno de ellos es el puerto de consola, este puerto de consola sirve para poder conectar un equipo PC al enrutador y poder entrar al modo de configuración por medio de una conexión Telnet y SSH. Las

características para poder efectuar una conexión al enrutador por medio de una PC son prederterminadas por el sistema (Figura 3.8).

Figura 3.7. Interfaces de conexión de Enrutador Cisco Series 2900.

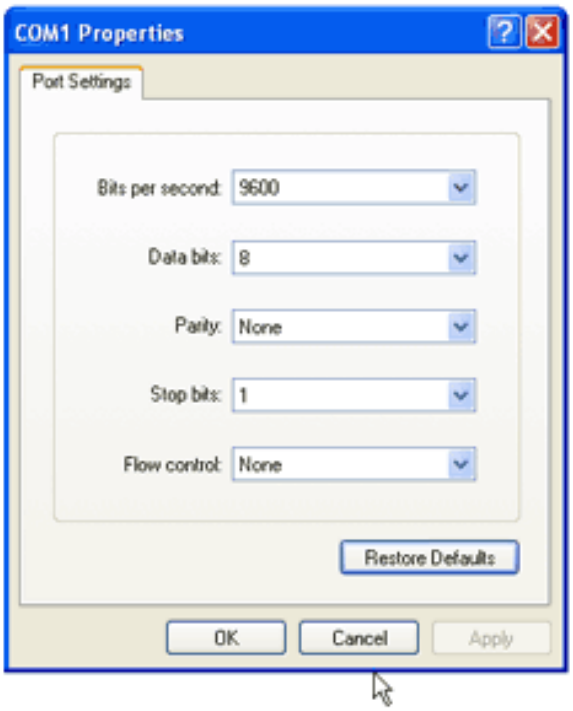


Figura 3.8. Valores Predeterminados para una conexión de Consola.

El otro puerto de administración con el que cuenta el enrutador es el AUX (auxiliar), éste tiene propósitos similares al del puerto de consola, puede tener una segunda conexión a consola o tambien conectar un modém al puerto, esto con el fin de mantener una conexión de marcación por petición de respaldo hacia otra ubicación en caso de que el enrutador tenga problemas de conexión. El puerto de consola se utiliza con regularidad para este propósito, esto se debe a que es un puerto serial asíncrono.

Tambien las interfaces de un enrutador pueden clasificarse de dos formas según el tipo de conexión que brinde.

1. Interfaces LAN

Las interfaces de conexión LAN son utilizadas para configurar las conexiones, ya sean Ethernet o FastEthernet, al realizar las configuraciones en las interfaces LAN se necesita información básica tal como la dirección IP y la máscara de red, además de esto hay características que el IOS permite estabecer conexiones troncales hacia un switch y agregar servicios a las interfaces. Además se puede modificar la velocidad de conexión que se desea, agregar listas de acceso que permitan tener un mejor control sobre el tráfico que entre y salga de la interfaz.

Adicionalmente se pueden configurar otras características de una interfaz de red dentro de un enrutador; como modo de ejemplo se cuenta con un enrutador Cisco 2900 con dos interfaces de conexión LAN, la fastethernet0/0 y la fastethernet0/1, a las cuales se accede vía el menu de configuración para demostrar las opciones que existen en ambas interfaces. Dentro de este menú se puede acceder a más opciones de configuración en cada una de los servicios (Figura 3.9). Para poder acceder a este menú en el IOS de Cisco, basta con presionar *"? + Enter"*.

Los menús que se encuentran dentro del IOS de Cisco (Figura 3.9), fueron pensados para que se accediera a ellos en forma de árbol, así al momento de configurar alguna opción en el caso de algún protocolo de enrutamiento, sólo será necesario acceder a la interfaz que sería afectada por la configuración realizada para dicho protocolo de enrutamiento.

```
Router(config)#int f0/0
Router(config-if)#?
  arp                 Set arp type (arpa, probe, snap) or timeout
  bandwidth           Set bandwidth informational parameter
  cdp                 CDP interface subcommands
  crypto              Encryption/Decryption commands
  custom-queue-list   Assign a custom queue list to an interface
  delay               Specify interface throughput delay
  description         Interface specific description
  duplex              Configure duplex operation.
  exit                Exit from interface configuration mode
  fair-queue          Enable Fair Queuing on an Interface
  hold-queue          Set hold queue depth
  ip                  Interface Internet Protocol config commands
  ipv6                IPv6 interface subcommands
  mac-address         Manually set interface MAC address
  mtu                 Set the interface Maximum Transmission Unit (MTU)
  no                  Negate a command or set its defaults
  pppoe               pppoe interface subcommands
  priority-group      Assign a priority group to an interface
  service-policy      Configure QoS Service Policy
  shutdown            Shutdown the selected interface
  speed               Configure speed operation.
  tx-ring-limit       Configure PA level transmit ring limit
  zone-member         Apply zone name
```

Figura 3.9. Menú de servicios de una interfaz LAN.

2. Interfaces WAN

Las interfaces WAN cuentan con un menú de opciones de configuración al igual que las interfaces LAN, la diferencia entre las interfaces de LAN y WAN es la velocidad de conexión que pueden soportar dichas

interfaces, ya que en las interfaces WAN se pueden configurar enlaces con encapsulación PPP, HDLC o Frame Relay. Estas interfaces se utilizan para conectar redes locales a otras redes a través de distancias más extensas. De igual modo, deben contar con una dirección IP, una máscara de subred y algún tipo de encapsulación. Una diferencia importante al momento de configurar un enlace entre dos enrutadores por medio de una conexión WAN, es determinar quién de los dos configurará el comando de *clock rate* (velocidad de reloj). Este comando es importante para que ambos enrutadores puedan sincronizarse entre ellos por medio de la utilización de un cable DCU/DSU.

Sin una frecuencia de reloj configurada, el enlace no quedará activo, ni se podrá efectuar comunicación entre ambos enrutadores, por lo que al verificar el estado del enlace por medio del comando *show interfaces* (Figura 3.10), en el modo privilegiado, éste aparecerá como *line protocol is down* (el protocolo está abajo).

```
Serial0/0/1 is administratively down, line protocol is down (disabled)
  Hardware is HD64570
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=down  DSR=down  DTR=down  RTS=down  CTS=down
```

Figura 3.10. Salida del comando Show Interfaces.

3.2. PROCESO DE ARRANQUE DE UN ROUTER

Una vez que se enciende el enrutador, éste inicia un proceso que arranca el sistema operativo con el fin de iniciar cada una de las interfaces de conexión del enrutador (Figura 3.11). Este proceso en teoría es muy fácil, pero la cantidad de subprocesos que se ejecutan deben tomarse en cuenta.

1. Primero el enrutador ejecuta la prueba de autocomprobación de encendido, esto quiere decir que verifica que todos los componentes de hardware que integran el equipo trabajen apropiadamente. El POST (Power on self test – autocomprobación de encendido) es ejecutado por la ROM, realizando pruebas a la CPU, a la RAM y a la NVRAM.

2. La ROM copia el programa bootstrap hacia la RAM para que el CPU pueda ejecutar las instrucciones necesarias en el bootstrap para localizar el IOS de CISCO y poder cargarlo hacia la memoria RAM.

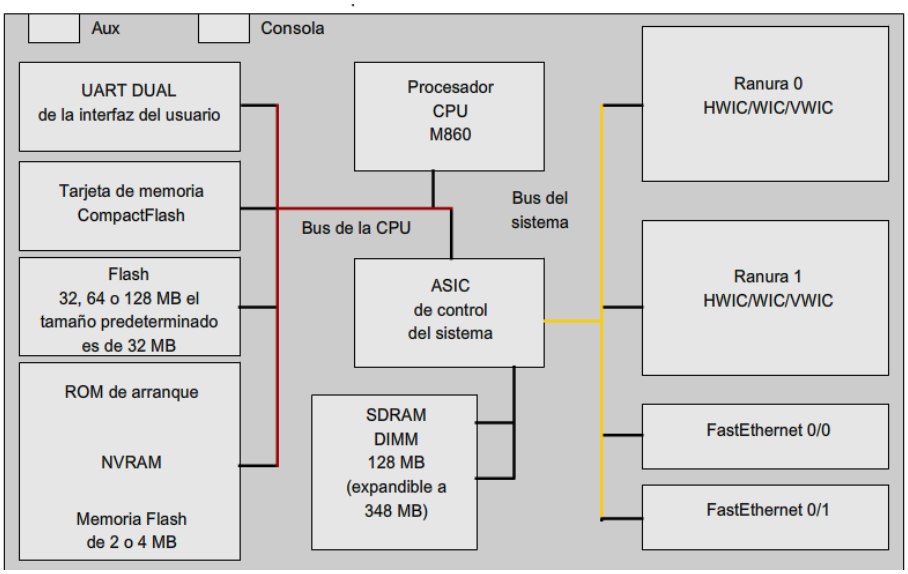


Figura 3.11. Diagrama Lógico de los componentes internos de un Enrutador Cisco 1841.

3. Una vez localizado el IOS de Cisco dentro de la memoria Flash o algún servidor TFTP y copiado a la memoria RAM aparecerá un mensaje en la consola con numerales, lo que significa que el sistema está siendo cargado.
4. Cuando se haya cargado por completo el sistema, el programa bootstrap que fue el encargado de buscar el sistema operativo en la memoria flash, realizará la búsqueda de la configuración de inicio llamada "*startup-config*" en la memoria NVRAM. En caso de que no haya localizado dicho archivo de configuración, el enrutador realizará una búsqueda a través de un servidor TFTP o enviará un broadcast a través de sus interfaces con enlace activo para poder obtener un archivo de configuración de algún enrutador vecino.
5. Una vez encontrado el "*startup-config*", el enrutador cargará la configuración de a una línea por vez y lo guardará en la memoria RAM como "*runnig-config*", ya que este archivo contiene los parámetros y los comandos necesarios para configurar las direcciones de interfaz, la información de los protocolos de enrutamiento, las contraseñas de los diferentes tipos de usuarios y cualquier otra configuración realizada por el administrador de la red.
6. Finalmente el sistema habrá iniciado y el usuario podrá realizar las configuraciones que necesite realizar en el equipo.

Como se ha comentado, un enrutador es un equipo que cuenta con características físicas y de software que lo hacen una herramienta indispensable en cualquier red de datos, ya que es el encargado de des-encapsular y encapsular todos los paquetes que lleguen a sus interfaces y verificar la información que le dirá hacia qué subred debe enviar el paquete, todo esto en cuestión de milisegundos.

Capítulo 4:
Balanceadores de Carga

"El único lugar donde el éxito viene antes que el trabajo, es en el diccionario"

– Donald M. Kendall -

Hoy en día la mayoría de las empresas tienen grandes cantidades de servidores para dar soporte a la gran demanda que pueden llegar a tener; siendo ésta la manera más económica y efectiva de mantener los servicios que brindan disponibles en todo momento.

Para que todo esto sea posible, las empresas de tecnología destinan grandes cantidades del presupuesto a la construcción de granjas de servidores para que los clientes tengan acceso a la diferente gama de servicios que éstas ofrecen. Esto se puede volver un gran problema si las empresas no llegan a medir el impacto que puede llegar a tener la falta de disponibilidad en el servicio; siendo éste perjudicial para el ecosistema de la empresa. Un servidor puede llegar a soportar un límite de peticiones dependiendo de las capacidades de hardware con las que cuente, pero cuando un servidor comienza a disminuir su capacidad de respuesta debido a la enorme cantidad de peticiones, la solución más obvia es que se incremente la capacidad de memoria, de almacenamiento, además de algunas mejoras al procesador. Esta opción no siempre es viable para el continuo crecimiento del tráfico en las redes de datos, quedando como una solución temporal al verdadero problema (Figura 4.1).

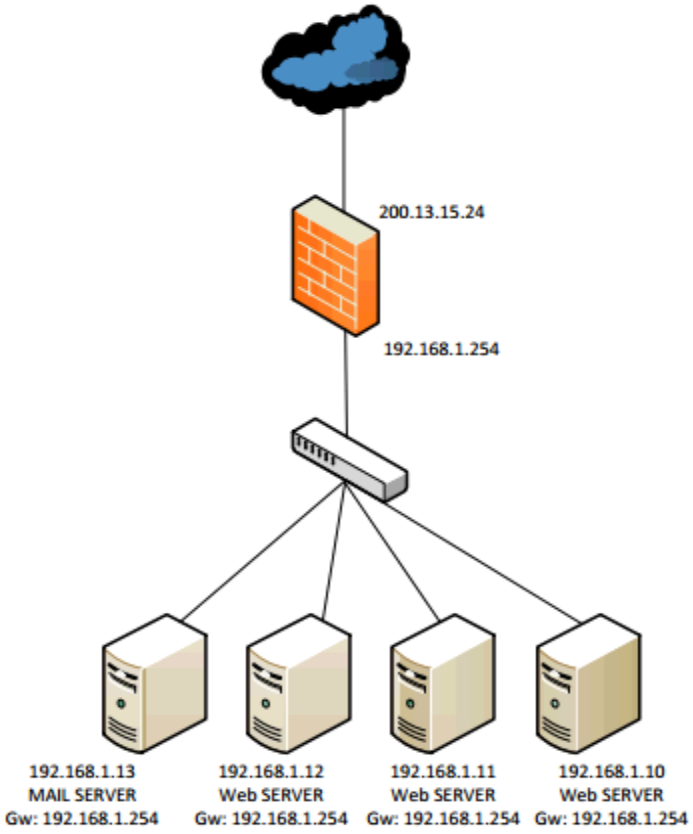


Figura 4.1. Diseño de red interna.

La opción más razonable a largo plazo es el balanceo de peticiones entre todos los servidores de la red con la finalidad de incrementar la velocidad de acceso del usuario al servidor, mejorando la confianza del cliente, la tolerancia a las fallas en la red y la opción de dar mantenimiento a los equipos sin que el servicio se vea afectado. De esta manera muchas de las empresas más conocidas en el ramo de los

servicios en TI como Google, pueden dar respuesta a la enorme cantidad de demanda con la que cuenta su servicio de búsqueda. Sin embargo, para que esto sea posible es necesario contar con balanceadores de carga los cuales dividirán todas las peticiones de conexión que reciban entre los servidores que estén conectados a éstos (Figura 4.2).

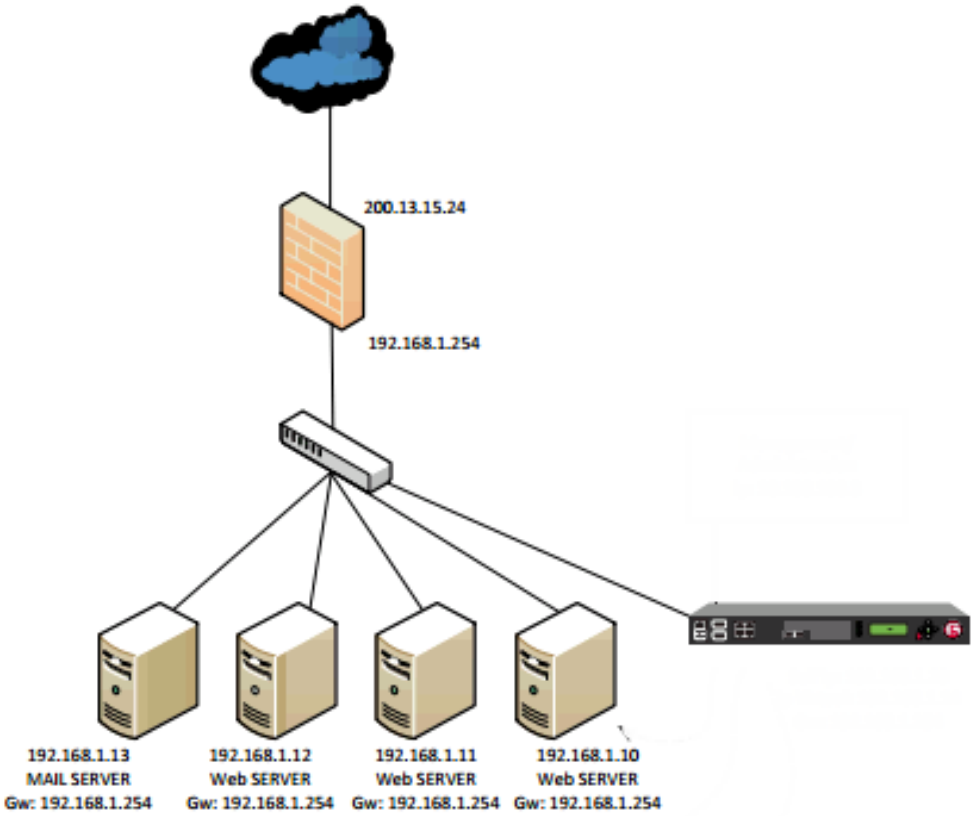


Figura 4.2. Diseño de red con el uso de un balanceador de red.

Los balanceadores de carga pueden ser una barrera de equipos de hardware y en otros casos software los cuales de manera estratégica se colocan antes de un pool de servidores para que estos repartan todas las peticiones entre los servidores que estén dentro de la red.

Actualmente los balanceadores de carga se han vuelto un dispositivo indispensable dentro del diseño y configuración de una buena red de datos, las empresas han fabricado dispositivos que son capaces de mantener un control minucioso sobre las conexiones que llegan a ellos y cómo éstos pueden actuar bajo diferentes ambientes de trabajo. Aun cuando hay enrutadores o conmutadores que cuentan con opción del balanceo de la carga de sus conexiones, éstos llegan a ser obsoletos en comparación con un equipo especializado para esta tarea.

Mientras un balanceador de carga está hecho específicamente para realizar tareas las cuales le permitan realizar un balance de qué servidores cuentan con menos conexiones en ese momento, un enrutador no

podría realizar todas aquellas operaciones las cuales son necesarias para monitorear la carga que tiene cada uno de los servidores que están conectados a éste y las tareas que comúnmente realiza.

De la misma manera el balanceo de cargas es la metodología que permite distribuir los flujos de trabajo a través de múltiples equipos con el fin de obtener la óptima utilización de los recursos y así maximizar el rendimiento y minimizar el tiempo de respuesta para prevenir la saturación del sistema.

4.1 Funcionamiento de un balanceador de carga

El balanceador está formado con los mismos componentes que un enrutador o un conmutador, la diferencia entre estos dos y los balanceadores de carga se encuentra en las funciones que realiza y las características de los protocolos IP que soporta. Es necesario que un balanceador de carga conozca el protocolo IP que contienen los paquetes que llegan a éste, para saber hacia qué servidor enviar las peticiones; es por eso que a los balanceadores de carga se les podría denominar como un conmutador de capa de aplicación.

El balanceador obtendrá el protocolo de la aplicación leyendo las cabeceras del paquete que llegue a éste, con el fin de determinar a qué servidor enviará la petición el balanceador tomará en cuenta factores de hardware como de red para enviar el paquete al servidor que cuente con menos conexiones activas. Esto es posible, ya que el balanceador cuenta con una serie algoritmos los cuales le permite conocer que servidores pueden atender las peticiones en puerta; estos algoritmos están basados en dos métodos de balanceo, el método de balanceo de **“Round Robin”** y el **“NLB” (Network Load Balancing – Balanceo de Carga de Red)**.

a) Round Robin

El método de balanceo de red de Round Robin, está basado en la configuración de un servidor de DNS (Domain Name Service – Servicio de Dominios de nombres), éste debe estar configurado para que re-direccione el tráfico a diferentes direcciones IP, con el fin de que el servicio tenga alta disponibilidad a la hora de recibir las peticiones. Esto ofrece un balanceo de carga muy sencillo, pero irregular e inconsistente, ya que nunca repartirá equitativamente las peticiones entre los servidores activos.

Otra peculiaridad que existe con este método al momento del balanceo, es que si llega a detenerse el servicio de alguno de los servidores o la dirección IP del servidor, deja de funcionar, no habrá manera de detener las peticiones que apunten al servidor que dejó de funcionar.

Lo que ocurre, es que una vez que el cliente envía una petición de servicio al DNS, el servidor busca en su base de datos y asigna una dirección IP de las que estén dentro de su configuración para que pueda atender la petición del cliente.

Se puede deducir, que este tipo de balanceo además de ser muy básico puede llegar a ser un problema al momento en el que la red reciba muchas peticiones de internet y éstas no sean direccionadas de manera inteligente.

b) Network Load Balancing

Este método se trata de una tecnología propia de la empresa Microsoft, la cual crea una red de servidores que mediante distintos mecanismos y algoritmos se comunican continuamente con los servidores que estén conectados directamente, con el fin de decidir qué servidor será el que recibirá la petición y si surge un problema en cualquiera de los servidores, éste será retirado automáticamente de la red para que no reciba peticiones hasta que se encuentre correctamente en funcionamiento.

Dentro de las características generales de balanceadores de distintos fabricantes se encuentra que pueden soportar protocolos de aplicación tales como; DNS, FTP (file transfer protocol – protocolo de transferencia de datos), HTTP (hipertext transfer protocol – protocolo de transferencia de hipertexto), HTTPS (hypertext transfer protocol safe – protocolo de transferencia de hipertexto segura), SSL (secure Shell – escudo seguro), radius (protocolo de autenticación), RDP (remote desktop protocol – protocolo de escritorio remoto) y TCP/UDP. Además de soportar una amplia gama de protocolos, un balanceador de carga ayuda a evitar ataques informáticos de DoS (Denial of Service – Denegación de Servicio), ya que cuenta con filtración de paquetes y DNSSEC (Domain Name System Security Extensions – extensiones de nombres de dominio de seguridad de sistema). Otra característica que tienen los balanceadores es la capacidad de conexiones en red, hay dispositivos que tienen conexiones LAN (Local Area Network – área de red local) de hasta 40 equipos con el mismo ancho de banda para todas sus conexiones activas.

Tomando en cuenta todas estas características, los fabricantes han hecho que un balanceador de carga sea un dispositivo mucho más complejo que una computadora que analiza el tráfico; dentro de todos los dispositivos de red se encontrarán mínimas diferencias entre ellos, son estas diferencias las que permiten que cada uno de ellos realice una tarea muy específica e indispensable dentro de la red.

Existen dos generaciones de balanceadores de carga y éstos se clasifican basándose en cómo miden el rendimiento de los servidores.

1. Primera Generación

Está generación se caracteriza por el hecho de cómo un balanceador de carga puede detectar el rendimiento de un servidor por medio de “passive polling (sondeo pasivo)”, basándose en el tiempo de respuesta del servidor. El balanceador puede tener una idea de cómo está funcionando el servidor dentro de la red y así determinar si es correcto enviarle más peticiones de conexión. Esta manera de evaluar el rendimiento de los servidores dentro de la red no es tan confiable, ya que sólo descubre que los servidores tienen un problema después de detectar los retrasos, o en el peor de los casos, hasta que los servidores ya no están en funcionamiento.

2. Segunda Generación

La segunda generación de balanceadores posee la capacidad de mensajería, dando un informe de todos aquellos servidores que se encuentran fuera de servicio y también el momento en el que estos han vuelto a ser puestos en línea. Además permiten que los servidores puedan ser desconectados o se les

pueda dar mantenimiento a través de un apagado progresivo; el cual no envía más peticiones al servidor, pero éste sigue en modo activo hasta que el tráfico de la red haya disminuido.

Para lograr estas funciones, el balanceador realiza peticiones continuamente a todos los servidores que estén conectados a él, con el fin de monitorear el uso de CPU (central process unit – unidad central de procesamiento), el uso de memoria y el número de conexiones abiertas, por lo que es posible detectar si el servicio que brinda éste, se encuentra activo.

Generalmente esta generación de balanceadores de carga se comercializa en parejas, con el fin de que uno de ellos funcione como respaldo del otro, de esta manera será posible configurar uno de los balanceadores de carga como el activo (que estará en funcionamiento realizando el balanceo correctamente) y el otro equipo funcionará como su respaldo o en modo stand-by (en espera). En caso de existir alguna falla o error humano el balanceador en espera entrará inmediatamente como activo; para que esto sea posible ambos equipos deberán estar configurados con la misma dirección IP, así como la misma MAC para que el tráfico no sea afectado al momento de que un cliente solicite una conexión a cualquiera de los servidores de la red.

4.2 Configuración básica de un balanceador de carga

Para poder realizar la correcta configuración de un balanceador de carga de la marca F5, es necesario conocer las características generales de los balanceadores así como los métodos de balanceo que se pueden configurar dentro del equipo; con el fin de poder realizar un balanceo de carga más efectivo entre los equipos que se encuentren conectados al F5 (Figura 4.3).

Figura 4.3. Balanceador de carga marca F5.

Todos los balanceadores cuentan con características específicas las cuales ayudan a que el objetivo dentro de la red se cumpla cabalmente, estas características son las siguientes:

1. **Carga asimétrica.-** una prioridad puede ser asignada manualmente a cada uno de los nodos que están siendo monitoreados, esto con el fin de que los servidores que tengan mayores recursos de hardware reciban más tráfico.

2. **Prioridad de activación.**- cuando el número de servidores disponibles es reducido dentro de la red, aquellos servidores que se encuentren en estado de stand-by (espera) podrán pasar a un modo activo de manera automática; esto para poder reducir la latencia de respuesta de los servidores.

3. **Aceleración y descarga de SSL.**- el balanceador puede concluir las conexiones de SSL activas, eliminando así la demanda del procesador por parte de los nodos balanceados.

4. **Prevención de DOS.**- los balanceadores proveen características como SYN cookies, limitación de conexiones por IP y "delayed binding" (retraso en la unión – los nodos balanceados no reciben tráfico del cliente hasta que el handshake TCP esté completo en el balanceador).

5. **Compresión de HTTP.**- el tráfico http se comprime en el balanceador y son enviados al cliente con el fin de minimizar el ancho de banda de la red.

6. **Caché HTTP.**- el balanceador cuenta con una memoria caché donde se almacenan los datos más solicitados por los clientes, con el fin de enviar la información al cliente sin la necesidad de requerir la información a los servidores.

7. **Monitoreo de servicios.**- el balanceador es capaz de monitorear y validar la funcionalidad de los equipos, así como de los servicios que proveen los nodos balanceados.

8. **Seguridad HTTP/IPS.**- debido a que todas las peticiones que van hacia los servidores pasan a través del balanceador, éste es capaz de validar el tráfico que sea HTTP y XML, y detener el código malicioso proveniente de la red externa.

9. **Autentificación de clientes.**- el balanceador también sirve como un punto central para la autentificación de clientes que accederán a los nodos.

10. **Priority queuing.**- el balanceador permite asignar diferentes tipos de prioridad a diferentes tipos de tráfico.

Por otra parte, es importante conocer los métodos de balanceo que pueden ser configurados en el balanceador, ya que al conocer cada una de las características que son configurables dentro del equipo; éste hace uso de los métodos de balanceo para lograr su objetivo dentro de la red, que es mantener un nivel de conexiones estables.

Los métodos de balanceo son los siguientes:

1. **Round robin.-** con este método el balanceador envía cada petición de conexión a cada uno de los servidores que se encuentran dentro del pool (comunidad), distribuyendo las conexiones de manera equitativa. Este método no es recomendable si dentro del pool de servidores existen equipos con diferentes características de hardware.
2. **Ratio.**- en este método el número de conexiones que cada servidor recibirá es proporcional a un peso promedio que se determina para cada uno de los equipos. Supóngase que existen cuatro servidores dentro del pool configurado en el balanceador y se le asignan las configuraciones 3, 2, 1, 1; el balanceador enviará tres paquetes al servidor a, dos al servidor b y un paquete a los servidores c y d respectivamente. Para este caso se deben tener en cuenta las características de hardware de cada uno de los servidores dentro del pool para conocer qué peso se le dará a la cantidad de peticiones que éste podrá responder sin que su rendimiento se vea afectado.
3. **Dinamic ratio.**- el balanceador asignará de manera automática los pesos de las conexiones que cada uno de los servidores dentro del pool, esto será posible realizando un análisis de las conexiones de cada uno de los servidores y de la latencia en cada una de sus conexiones.
4. **Fastest.**- el balanceador realizará un análisis de la latencia de cada uno de los servidores, de tal forma que elegirá el servidor que responda más rápido para asignarle la conexión en espera. Este método es útil cuando los servidores del pool tienen las mismas características de hardware, así como del servicio que proporcionan.
5. **Least connections.**- cuando el balanceador tenga configurado este método, seleccionará el servidor que tenga el menor número de conexiones.
6. **Observed.**- el balanceador utiliza una combinación de la lógica utilizada en los métodos de least connections y fastest, por lo que el servidor que tenga menor número de conexiones y mayor respuesta a sus peticiones será elegido para responder la siguiente petición en espera.
7. **Predictive.**- el balanceador utiliza la lógica del método observed, además de analizar la tendencia de carga durante un tiempo determinado, con el fin de analizar qué servidor dentro del pool cuenta con el mejor rendimiento y así asignarle la conexión en espera.
8. **Weighted least connections.-** este método utiliza la combinación del número de conexiones que cada nodo tiene, así como el límite de conexiones asignadas a cada servidor; con el fin de determinar el algoritmo que asignará un límite de conexiones a cada uno de los servidores del pool.
9. **Least sessions.**- este método asignará mayor número de conexiones al servidor que tenga menor número de entradas en su tabla de persistencia. Para que éste funcione de manera eficaz, se deberá asignar un perfil de persistencia para un servidor virtual.

En caso de que un servidor pertenezca a múltiples pools de servicios, las mediciones se realizarán con base en los requerimientos que cada pool realice (member), o a los tiempos de respuesta y peticiones

que el mismo servidor sea capaz de manejar (nodo). Por lo general se escogerá el método de balanceo por nodo y no por pool al que pertenezca el servidor que se está balanceando.

Una vez entendidos los parámetros anteriores se podrá conocer las características mínimas para realizar la configuración básica de un balanceador, para integrar un dispositivo como éste es necesario entender que se configuran tres diferentes direcciones IP dentro de la red (Figura 4.4).

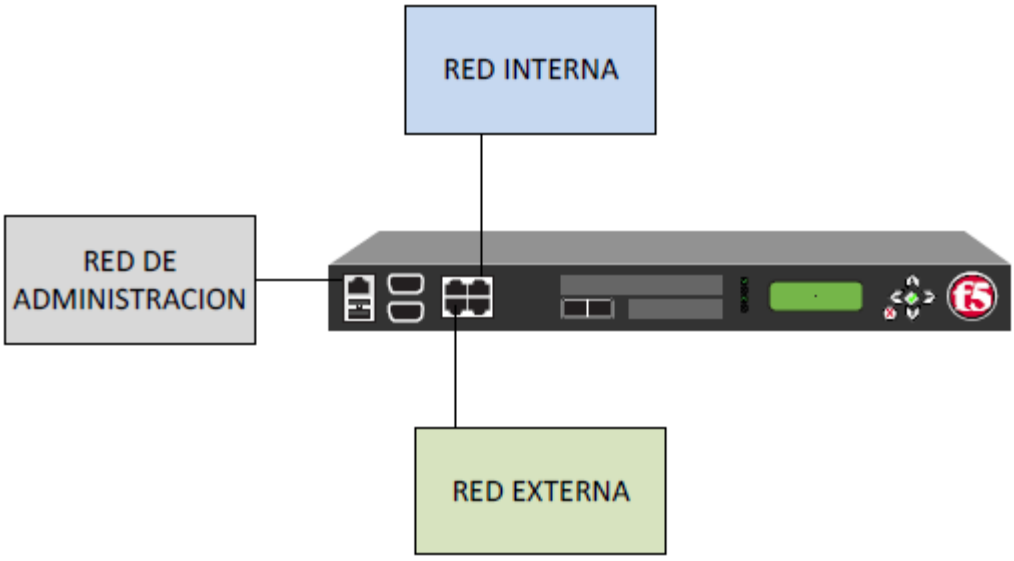


Figura 4.4. Características lógicas de un balanceador de carga.

El equipo deberá ser configurado con tres direcciones IP, las cuales pertenecerán a tres segmentos de red diferentes, además que serán de ayuda para la mejor administración del equipo; éstas se podrán configurar dentro de la interfaz gráfica del equipo o a través de una consola de administración por medio de una conexión telnet o SSH.

a) Dirección IP de red interna, esta dirección IP deberá estar conectada al pool de servidores de los cuales se estará balanceando la carga, es importante conocer la dirección IP exacta de esta interfaz, ya que estará configurada como Gateway del pool de servidores. (Figura 4.5).

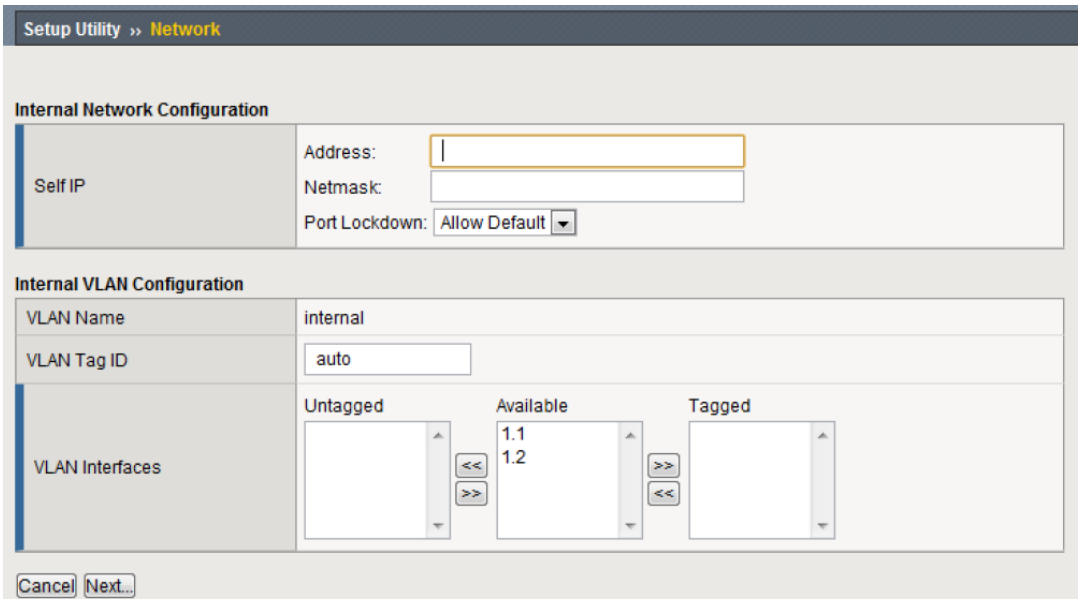


Figura 4.5. Configuración de la dirección IP de red interna.

b) Dirección IP de red externa, ésta deberá ser la dirección IP que se encuentre configurada con una dirección IP de la red externa y a la que llegarán las peticiones de conexión, estas serán balanceadas entre los servidores que se encuentran en red interna, al llegar a este punto la petición, se realizará el proceso de selección del método de balanceo para que el cliente pueda recibir una respuesta más rápida del servidor (Figura 4.6).

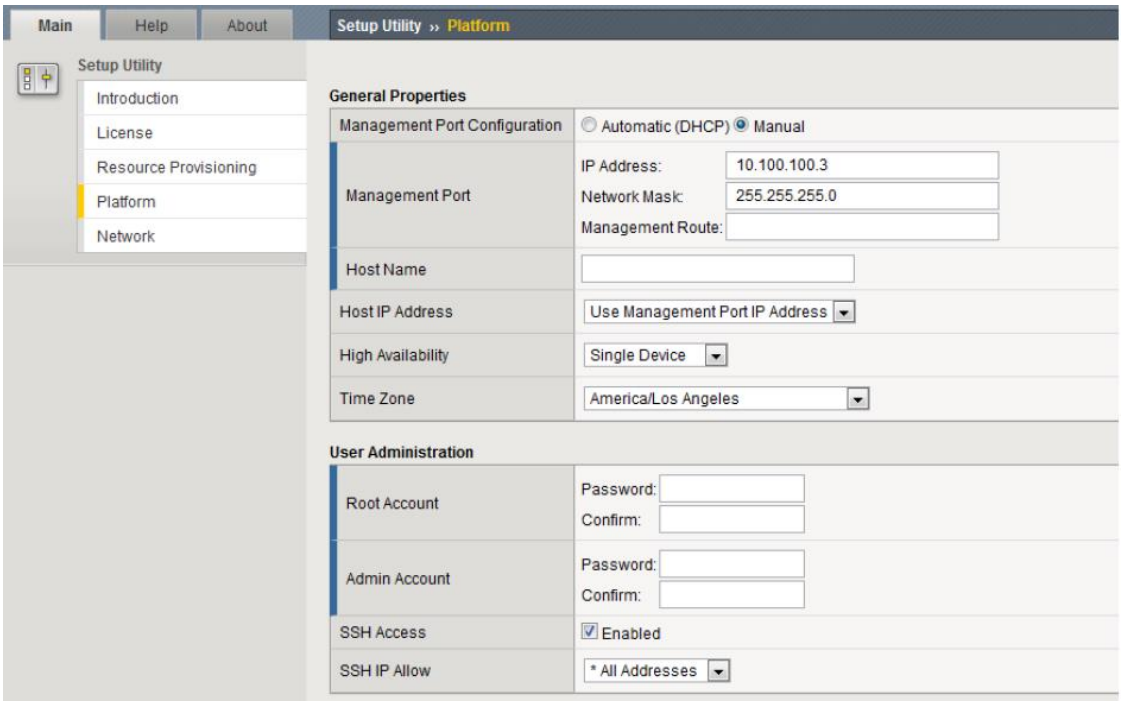


Figura 4.6. Configuración de la dirección IP de red de administración.

c) Dirección IP de red de administración, esta será la dirección IP de administración del equipo y por la que el administrador de la red se conectará a la interfaz de configuración y realizará los cambios necesarios al equipo. Cabe mencionar que esta dirección IP no debe pertenecer al mismo segmento que la dirección IP de red interna y tampoco de red externa, ya que de esta manera se podrá evitar cualquier amenaza de seguridad y comprometer la información que pasa por el balanceador de carga (Figura 4.7).

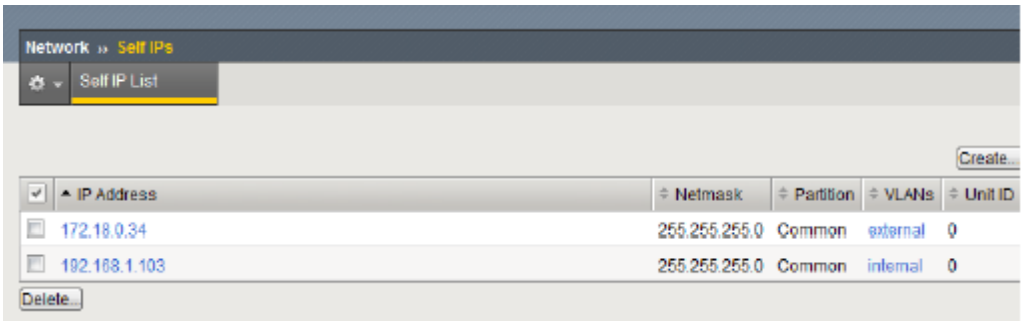


Figura 4.7. Interfaces de red configuradas en el sistema.

Una vez que se han configurado las direcciones IP de red de administración, interna y externa, se podrá observar que estas se encuentran ya configuradas dentro del sistema; esto es necesario para poder acceder a visualizar el estado del equipo tal como memoria, carga del procesador y su capacidad de almacenamiento (Figura 4.8). Todo esto, con el objetivo de que el administrador de la red pueda visualizar las características físicas del equipo y así determinar los métodos de balanceo que configurará en el equipo para un buen funcionamiento del balanceador en la red.

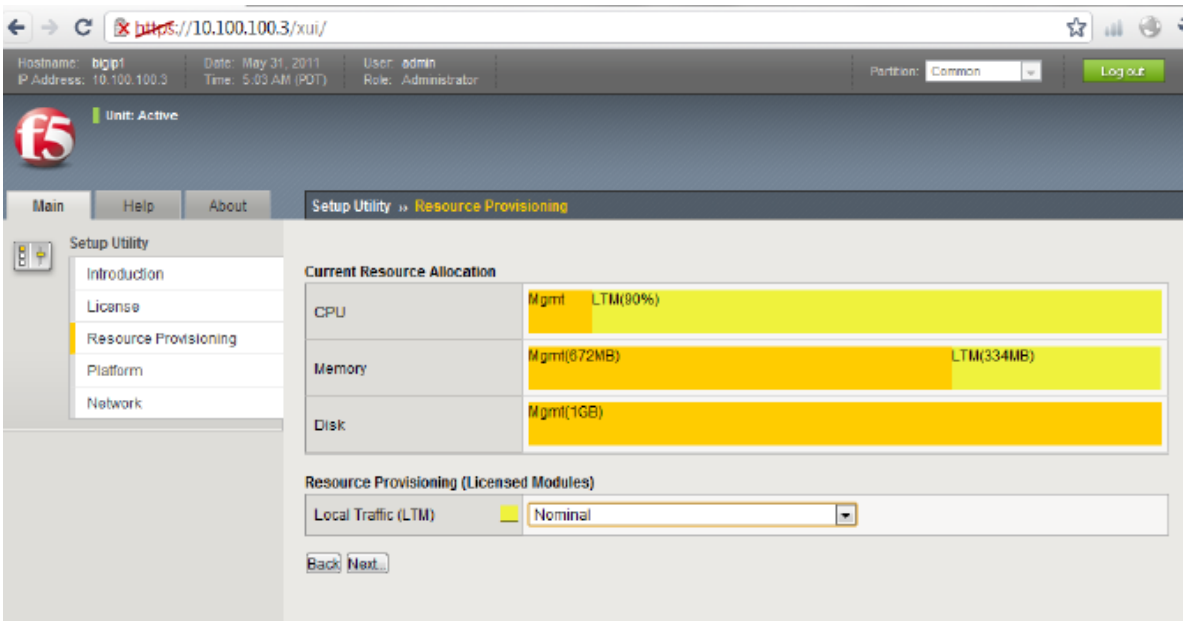


Figura 4.8. Monitor de los recursos del sistema.

Una vez que se han configurado las tres interfaces de red, hay dos maneras de realizar la configuración del método de balanceo; una de ellas es por interfaz web y la otra es a través de consola. Antes que nada, es importante que se configure el pool de nodos que se estarán balanceando en el equipo, para posteriormente configurar los servicios que se estarán monitoreando al momento de que el equipo realice el balanceo.

Para crear un pool de balanceo, primero hay que crear los nodos, los cuales estarán siendo las interfaces lógicas a balancear; por lo tanto es necesario seguir los siguientes pasos:

1. Dentro de la página principal del balanceador se selecciona en el menú del lado izquierdo la pestaña "*Local Traffic*".
2. En el siguiente sub – menú seleccionar la pestaña *"Nodes".*
3. En el menú siguiente habrá una fila con la leyenda – No records to display -, ahí se selecciona el botón *"Create".*
4. Se llenan las casillas con los datos de red de los nodos (la dirección IP del servidor) y se selecciona el radio que se le estará dando al nodo al momento de realizar el balanceo (Figura 4.9).

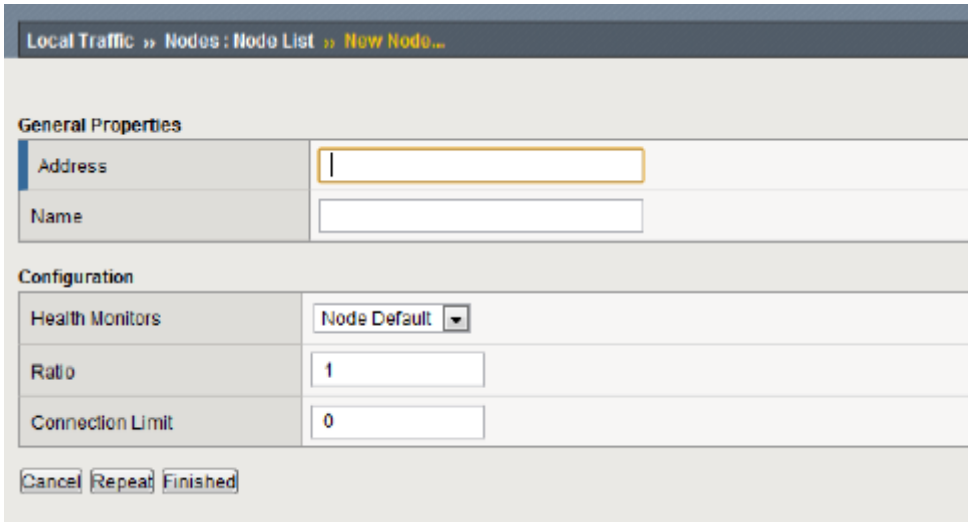


Figura 4.9. Pantalla de configuración de nodos en el balanceador de carga.

5. Se coloca la dirección IP que corresponda al servidor que se está configurando, un pseudónimo con el cual se le identificará dentro de la red y además se configuran otras características como el monitor que se desea colocar al nodo, el peso que se utilizará al momento del balanceo y el número de conexiones que se desea que acepte el servidor (Figura 4.10).

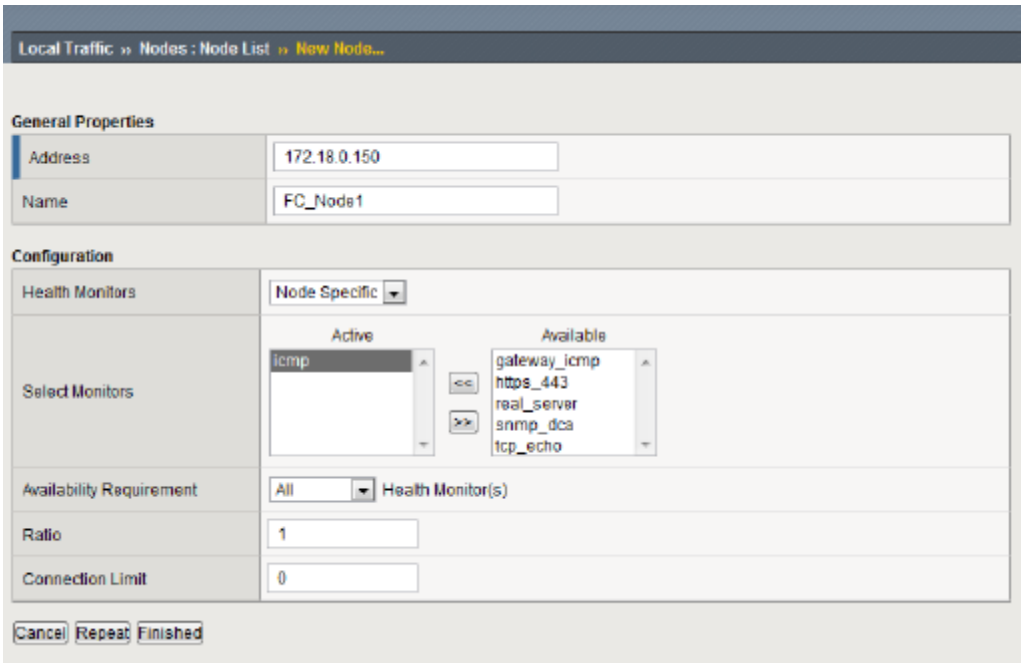


Figura 4.10. Ejemplo de configuración de nodo de balanceo.

6. Una vez que se de click en finalizar, éste aparecerá en la lista de nodos configurados dentro de la lista de los nodos (Figura 4.11).

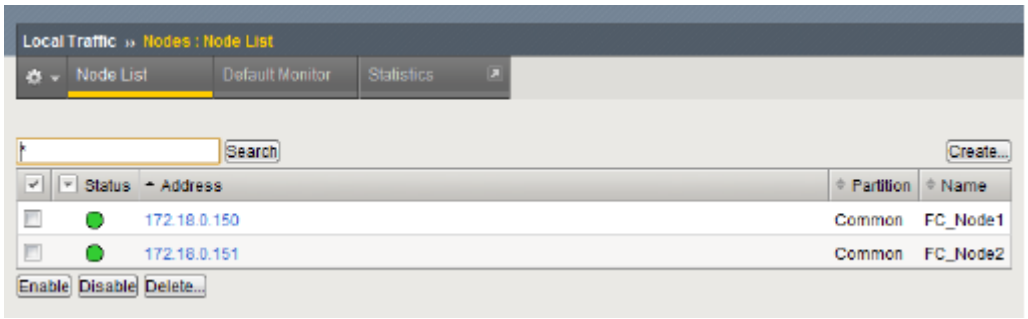


Figura 4.11. Monitor de nodos en balanceo.

Una vez creados todos los nodos que corresponden a los servidores que están dentro de la red, es necesario crear un pool.

7. Seleccionar la pestaña "*Pools*", aparecerá una pantalla parecida al momento de crear los nodos, dar click en el botón *"Create"* para crear un pool de balanceo.
8. Se debe de especificar un nombre para el pool de balanceo, así como el método que se utilizará para balancear los nodos que estarán dentro del pool. Los balanceadores cuentan con monitores de salud, los cuales ayudan a verificar que el servicio está siendo monitoreado además de la cantidad de conexiones que está recibiendo el nodo (Figura 4.12).

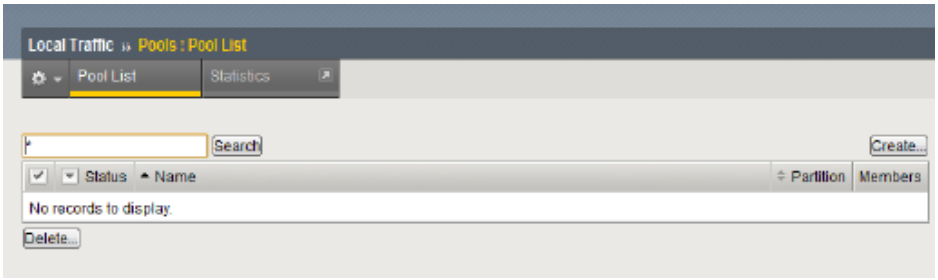


Figura 4.12. Pantalla de configuración de pool de balanceo.

9. Agregar los nodos que serán balanceados por el pool (Figura 4.13).

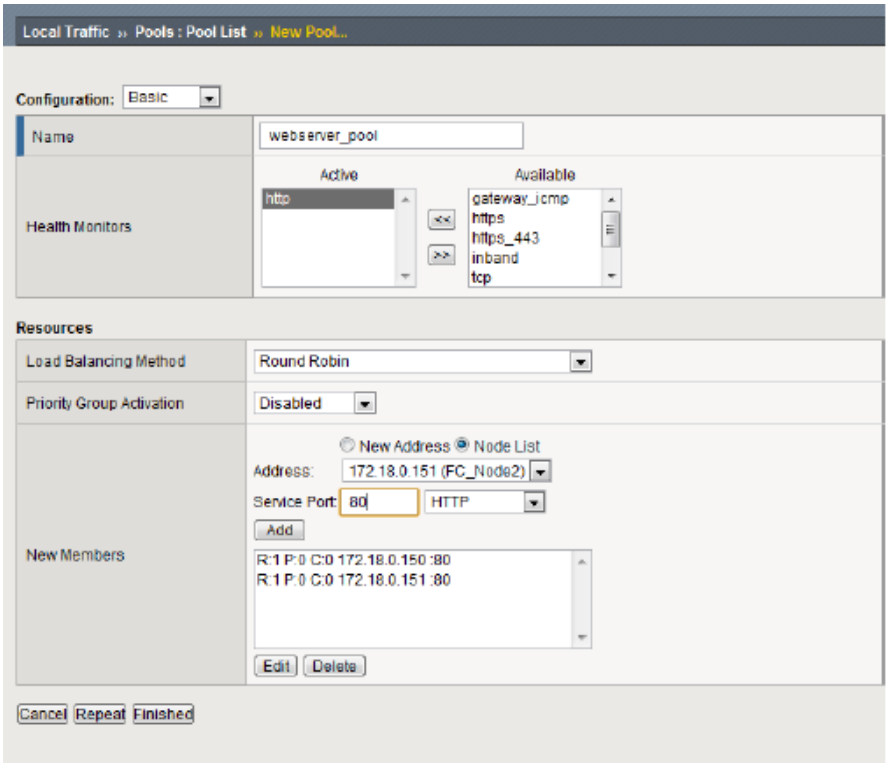


Figura 4.13. Ejemplo de configuración de pool de balanceo.

10. Dar click en finished (Figura 4.14).

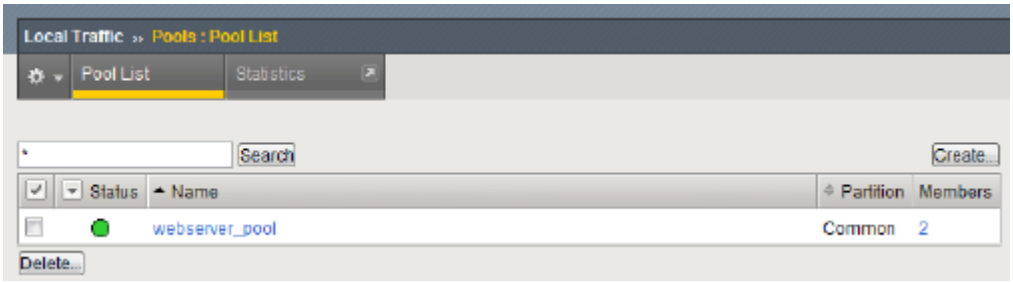


Figura 4.14. Monitor de pool de balanceo.

Otra ventaja que brindan los balanceadores es que se pueden monitorear las conexiones de los servidores a través de un servicio en específico (http, https, dns, etc) o por protocolo de conexión (tcp/udp). Esto beneficia al momento de tener varios pools dentro de la configuración del balanceador, ya que se podrá contar con un mejor monitoreo en caso de que los servidores cuenten con contenido más específico como video, contenido html e imágenes. Realmente un balanceador es muy útil dentro de una red de datos, porque puede ayudar a prevenir ataques de desbordamiento, así como detectar posibles fallas en la red monitoreando el servicio que están brindando los servidores a los clientes. La mayoría de las industrias deben ser capaces de conocer sus vulnerabilidades y conocer muy bien los activos de la compañía; esto con el fin de determinar si es necesario invertir en equipos como éstos o en su caso implementar otras soluciones.

Capítulo 5:
Firewalls (Cortafuegos)

"Hay tres tipos de personas, los que hacen que las cosas pasen; los que miran las cosas que pasan y los que preguntan qué paso."

- Nicholas Murray Butler –

5.1 Definición

Los firewalls (cortafuegos) son otro tipo de dispositivos que brindan seguridad en una red de datos. Estos dispositivos ayudan a mantener un nivel de seguridad de tal modo que se pueda prohibir el acceso a cualquier persona que intente burlar la seguridad dentro de la red.

Todas las computadoras que estén conectadas a internet pueden ser directamente accesibles para cualquier persona en la red a través de una conexión de FTP o Telnet, donde otra persona o atacante puede tomar el control del equipo y explotar cualquier vulnerabilidad que pudiera existir en la red. Es por eso que se colocan los firewalls en las redes de datos, para que exista un control en el tráfico de la red y sea más difícil que algún perpetrador pueda obtener el control de alguna computadora que se encuentre dentro de la red. Un firewall es un sistema al igual que otros componentes que puede ser software o hardware dependiendo de las necesidades y características de la red.

Los firewalls pueden ser utilizados de diferentes formas para agregar seguridad a una red de datos, ya sea una red corporativa en donde se maneja información importante para la empresa o en un pequeño negocio el cual solo cuenta con información necesaria para sus operaciones diarias. Toda la información puede estar protegida gracias a los beneficios que brindan los firewalls.

Un firewall permitirá que los administradores configuren sentencias con las cuales puedan determinar qué tipo de tráfico o paquetes se les permitirá el acceso a la red. El firewall será el primer punto de contacto entre la red interna e internet, es por eso que es muy importante tener el conocimiento del tipo de protocolos que serán permitidos dentro de una red y también de aquellos que inmediatamente serán descartados por el firewall.

El equipo es configurado con sentencias llamadas Access List (Listas de Acceso), las cuales son reglas que deberán de estar configuradas en el equipo para que una vez que un paquete solicite acceso a la red interna, éste será descifrado y será comparado con las listas de acceso configuradas en el firewall; si el tipo de protocolo está dentro de una lista de acceso que le permita acceder a la red, el firewall dejará pasar la información del paquete al host destino, en caso contrario el paquete será descartado por el firewall.

El objetivo principal de los firewalls en la red es mejorar la seguridad de ésta basándose en los siguientes puntos:

- Proteger y aislar las aplicaciones, servicios y equipos de la red interna de tráfico peligroso y no identificado de la red externa.
- Limitar o deshabilitar el acceso a dispositivos de la red interna del internet público.
- Soportar el protocolo NAT (Network Address Translation – Traducción de direcciones de red), el cual permite a un segmento privado de la red interna, conectarse a internet a través de una sola dirección IP pública.

Todo esto es posible debido a que un firewall puede trabajar a partir del conjunto de reglas configuradas de dos distintas formas, *exclusiva* o *inclusiva*. Un firewall que trabaje de manera *exclusiva*, permitirá el acceso a todo el tráfico excepto a aquel que concuerde con las reglas configuradas. Mientras que cuando el equipo se encuentra configurado para que trabaje de manera *inclusiva* solo permitirá el

acceso al tráfico que concuerde con las reglas configuradas y desechará todos los paquetes que provengan de internet.

Un firewall inclusivo permite un mejor control sobre el tráfico externo, siendo éste la mejor opción para aquellos sistemas que ofrecen servicios en el internet público, además de hacerlos más seguros, ya que reduce el riesgo al no permitir el acceso a tráfico no deseado.

a) Firewall (Hardware)

Los firewalls pueden adquirirse como un producto independiente, pero recientemente se pueden encontrar firewalls integrados dentro de los enrutadores de banda ancha.

Este tipo de firewalls pueden ser efectivos al momento de brindar seguridad en la red con poca o ninguna configuración de filtrado de paquetes y pueden proteger cada una de las máquinas de la red local. La mayoría de los firewalls llegan a tener como mínimo cuatro puertos de conexión de red.

Este tipo de firewalls debe cumplir con las funciones básicas de un equipo de estas características, donde se utiliza el filtrado de paquetes para examinar las cabeceras de cada uno de éstos y así determinar su dirección de origen y destino, así como el tipo de protocolo que contiene el paquete. Esta información es comparada con un conjunto de reglas predefinidas o configuradas para determinar si al paquete se le permitirá el acceso a la red o será descartado por el equipo.

Para asegurar que el firewall está configurado de manera exitosa es necesario determinar las reglas de filtrado que serán configuradas en la red y así evitar el tráfico no deseado en la red.

b) Firewall (Software)

Un firewall de tipo software es usado con más regularidad por usuarios caseros que desean mejorar la seguridad de su red privada. De esta manera el usuario debe instalar el software del firewall deseado como cualquier otro software en su computadora y configurarlo de la manera que más le convenga. Este método puede ser muy útil para asegurarse de que nadie tome el control de su equipo de cómputo y pueda lucrar con la información que considera como importante o inclusive utilizar el equipo de cómputo en un ataque de denegación de servicio.

Existen algunas compañías en las que sus firewalls proporcionan seguridad en contra de troyanos, gusanos y virus. Muchos firewalls también cuentan con controles definidos de usuarios para configurar el uso de impresoras en red, de archivos seguros y también evitar que se ejecuten aplicaciones maliciosas, además pueden incorporar controles de privacidad, filtrado web y más.

Hay que tener en cuenta que un firewall en software consume recursos de procesamiento del equipo en el que es instalado, que a comparación con un firewall en hardware, éste está completamente dedicado al filtrado de los paquetes, aún y cuando hay firewalls que pueden ejecutarse en modo "*background" (segundo plano),* éstos no llegan a ser tan efectivos como un firewall de tipo hardware porque no están dedicados completamente al filtrado de paquetes.

5.2 Funcionamiento de un Firewall

El funcionamiento de un firewall es muy sencillo, este dispositivo se encarga de revisar cada paquete o petición que llegue a él y permitirle o negarle el paso basado en las reglas que hayan sido configuradas previamente. Usualmente se usan las funciones de *accept* o *drop* (permitir o tirar) y muy pocas la función *deny* (denegación), debido a que denegar el acceso implica un gasto en el ancho de banda para dar respuesta al host que envió la petición. Por obvias razones es mucho más fácil tirar un paquete el cual no se le permita el acceso, porque de esta manera si el host no recibe respuesta por parte del sistema remoto, éste entenderá que el paquete ha sido descartado e intentará de nuevo o se dará por vencido finalmente.

Las reglas que se configuren en el firewall deberán estar regidas por alguna política de seguridad que haya sido aprobada y adoptada por la organización. Por lo general se utilizan dos tipos de políticas de seguridad, las cuales son:

- Todo lo que no está permitido, queda explícitamente prohibido (Prohibitiva).
- Todo lo que no está prohibido, queda explícitamente permitido (Permisiva).

Hay que tener mucho cuidado al utilizar estas dos políticas de seguridad dentro de las configuraciones de un firewall, porque muchas veces no se llega a hacer un verdadero juicio de las necesidades de seguridad de la red, por lo tanto no se elige una política de seguridad acorde con lo que la organización requiere.

También es posible realizar sus propias políticas de seguridad para que éstas sean implementadas, pero hay que tener en cuenta cuando ésta se tiene que turnar permisiva o prohibitiva y muchas veces esto es pasado por alto al momento que se está elaborando una política de seguridad al interior de la organización.

Ahora bien, se ha hablado de manera muy general el funcionamiento del firewall y de cómo su objetivo es permitir o no el paso de los paquetes a través de éste para mejorar la seguridad de la red. Pero ¿cómo utiliza las sentencias que le dictan al firewall qué paquetes descartar?, ¿cuál es la estructura de las sentencias?, ¿cómo se configuran en el firewall?.

Básicamente el firewall analiza el encabezado de cada paquete que se ha intercambiado entre un host de la red interna y uno de la red externa, de esta manera todo el paquete es des-encapsulado analizando los siguientes elementos:

a) La dirección IP del host que envía los paquetes.
b) La dirección IP del host que recibirá los paquetes.
c) El tipo de protocolo del paquete (UTP, TCP, etcétera.).
d) El puerto de destino el cual ayudará a detectar el servicio que es solicitado.

Una vez que se haya recabado la suficiente información por el firewall, éste comparará la información con las sentencias configuradas en el equipo y aplicará la función que exista para un determinado tipo de paquete (aceptar o denegar).

5.2.1 Access Lists (Listas de Acceso)

Las access-lists (listas de acceso) son un elemento importante de los firewalls, si éstas no se encuentran bien configuradas puede existir un riesgo latente dentro de la red. Una vez determinada la política de seguridad será importante conocer la estructura de las reglas que se configurarán en el equipo.

Existen diferentes tipos de sentencias dependiendo del tipo de firewall que se vaya a utilizar, se pueden configurar en un entorno Linux, IOS de cisco o hasta en Windows. Los elementos que son esenciales en todas estas plataformas son las siguientes:

1. Acción; la cual se ejecutará una vez que se des-encapsule el paquete que llega al firewall y se compare con la regla.
2. Dirección IP origen; la mayoría de las veces es importante conocer la dirección IP origen para dar mayor seguridad, en caso de que se encuentre en una red de enlace directo se podrá eliminar un paquete que tenga como origen una dirección IP diferente a la que se tenga configurada en la sentencia.
3. Dirección IP destino; es importante conocer la dirección IP destino, para que en dado caso, se eliminen los paquetes que van dirigidos a ese host final.
4. Protocolo; en combinación con el puerto de destino se puede llegar a filtrar la mayoría del tráfico maligno que no desee que ingrese a la red interna.
5. Puerto destino; se puede filtrar la mayoría de paquetes malignos con el simple hecho de conocer el puerto destino de un paquete.

Una vez que se comprende una sentencia o regla del firewall, éste comenzará a ponerla en práctica des-encapsulando los paquetes que lleguen al equipo y descartando o permitiendo el acceso a la red. Sin duda alguna las reglas del firewall son la columna vertebral de este dispositivo. Apoyadas de una buena política de seguridad que haya sido adoptada por la organización, la seguridad se verá incrementada de una manera significativa y los activos de la organización tendrán un mínimo riesgo de ser el objetivo de los atacantes de la red.

Otro elemento de una red de datos es el proxy, el cual es un servidor que brinda un servicio de manera transparente al usuario, lo que hace es recibir peticiones de usuarios y redirigirlas a internet. Normalmente en la red un proxy funciona a su vez como un servidor caché, esto porque puede almacenar las páginas web que tienen mayor tráfico desde red interna hasta el internet; de esta manera el tiempo de latencia llega a ser menor y el tráfico de la red se reduce sustancialmente. Por lo general un proxy se configura antes del firewall si el sentido es de red interna a red externa, por lo que un proxy ayuda a que el firewall que se encuentre entre la red externa y la red interna tenga menor tráfico y éste se sature.

En los últimos años se ha podido añadir funciones de filtrado a los proxies, con lo que pueden permitir a un usuario determinado acceder a ciertas páginas de internet o por el lado contrario, negarle el acceso basado en las reglas que se pueden configurar en el equipo.

Un servidor proxy y un firewall como se acaba de plantear, son totalmente diferentes, pero en realidad se complementan dentro de la estructura de la red. El proxy se utiliza para redirigir todas aquellas

peticiones de los usuarios de manera transparente, mientras que el firewall es el método de protección para la red local, dispositivo que permite el acceso a determinado número de puertos.

Además los servidores proxy tienen la posibilidad de monitorear el tráfico a través de capturas, esto ayuda a que el firewall no se sature de procesos y es utilizado por los administradores de redes frecuentemente para realizar un monitoreo de los puertos que están siendo utilizados durante un período de tiempo, esto con el fin de eliminar ataques cibernéticos o detectar qué usuario puede estar entrando a páginas web prohibidas.

Para que se puedan realizar capturas en el proxy es indispensable que el servicio del squid esté configurado en el sistema Linux. El archivo *squid.conf* se encuentra localizado en la carpeta */usr/local/squid/etc/*. Dentro del archivo es necesario configurar las acl (listas de acceso) que estarán siendo utilizadas al momento de las capturas.

A continuación se mostrará un archivo básico *squid.conf* (Figura 5.1) en donde se configuran las listas de acceso para un sistema Linux. Esto le servirá al lector para que pueda observar que se puede configurar la lista de acceso con base en un segmento de red, a una dirección IP específica e inclusive por puertos conocidos que se pueden permitir.

```
#archivo de configuración para listas de acceso

http_port 3128

refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320
acl localnet src 10.0.0.0/8     # RFC 1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC 1918 possible internal network
acl localnet src fc00::/7       # RFC 4193 local private network range
acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged)
machines
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost
http
```

Figura 5.1 Archivo de configuración *squid.conf* configurado en un sistema Linux.

Como se puede ver en la configuración del archivo *squid.conf* es necesario indicar el puerto que se utilizará para el monitoreo del tráfico http (3128); la configuración del segmento de red al que pertenece el proxy y determinar todos los puertos que se consideran como seguros.

Esto ayuda a que se mejore la seguridad de red interna y se restrinjan todos aquellos portales a los cuales no estará permitido acceder desde la red interna.

Para configurar el archivo *squid* antes del firewall se utilizarán las siguientes sentencias.

El *squid* deberá estar conectado al servidor de DNS, de esta forma se especifican cuáles serán las peticiones que se redirigirán al caché y cuáles otras podrán redirigirse a la red externa a través del firewall (Figura 5.2).

```
acl INSIDE dstdomain .mydomain.com
always_direct allow INSIDE
never_direct allow all
```

Figura 5.2 Configuración de un servidor de DNS en el archivo squid.conf.

También se pueden especificar los servidores de red interna por medio de su dirección IP (Figura 5.3).

```
acl INSIDE_IP dst 1.2.3.0/24
always_direct allow INSIDE_IP
never_direct allow all
```

Figura 5.3 Configuración de la dirección IP de cualquier servidor en red en el archivo de *squid.conf.*

Finalmente si se utilizan varias memorias cachés para la red interna, se tendrá que configurar una como la memoria predeterminada a la que accederá el *squid* en caso de que éste no sepa cuál de todas las memorias caché utilizar (Figura 5.4).

```
cache_peer xyz.mydomain.com parent 3128 0 no-query default
```

Figura 5.4 Sentencia de configuración de caché en el archivo *squid.conf.*

También hay que configurar todos los sitios a los que se permitirá acceder desde la red interna (Figura 5.5).

```
acl dstdomain .tvazteca.com
acl dstdomain .unam.mx
acl dstdomain .cnn.com

http_access allow dstdomain
```

Figura 5.5 Configuración de sitios web permitidos por usuarios de red interna en el archivo *squid.conf.*

Para que el *squid* cargue la configuración en el sistema habrá que guardar los cambios en el archivo *squid.conf* y corroborarlos a través de introducir las siguientes sentencias en la consola (Figura 5.6).

```
squid –k parse
squid –k reconfigure
```

Figura 5.6 Comandos para corroborar sentencias y reconfiguración del comando *squid.conf.*

Una vez configurado el *squid* en el proxy, se podrá hacer el monitoreo de los puertos y por consiguiente ayudará a monitorear el tráfico en tiempo real, así como las respuestas que reciben los usuarios por medio de las banderas del protocolo de transmisión (**SYN, SYN-ACK, ACK**).

5.2.2 Monitoreo de puertos

El monitoreo del tráfico en la red, será posible a través de *log-files*, esto podrá realizarse de dos maneras diferentes utilizando los siguientes comandos: *tail* y *grep*.

Con el comando *tail* se puede monitorear todo el tráfico de un usuario en específico, todo esto en tiempo real. Así se puede ver todas las páginas que el usuario visite, las imágenes que se carguen en su ordenador, inclusive el tráfico que generan algunos programas que estén instalados en su equipo y realicen solicitudes a la red externa. El comando se puede ejecutar en un sistema Linux (Figura 5.7).

```
#tail –f /var/log/squid/access.log
```

Figura 5.7 Sentencia de uso del comando *tail.*

Con el comando *grep* se puede obtener información más detallada dentro del tráfico que está siendo analizado y buscar alguna página web específica dentro de la enorme cantidad de información que pasa por el firewall, por ejemplo *facebook*. Con este comando será necesario indicar a través de apóstrofes el texto que se buscará entre todo el tráfico de red que se genere en el servidor del usuario que está siendo analizado (Figura 5.8).

```
#grep 'facebook' /var/log/squid/access.log
```

Figura 5.8 Sentencia del comando grep buscando tráfico que contenga la palabra *facebook.*

Con este comando se observará todo el tráfico entre el sitio de *Facebook* y red interna (Figura 5.9).

```
872739961.631 1566 10.0.0.21 ERR_CLIENT_ABORT/304 83 GET
https://www.facebook.com/ - DEFAULT_PARENT/localhost.home.nl -
872739962.976 1266 10.0.0.21 TCP_CLIENT_REFRESH/304 88 GET
https://www.facebook.com/photo.php?fbid=579423112120149&set=a.10264472313
1326.3756.100001573958973&type=1&relevant_count=1&ref=nf -
DEFAULT_PARENT/localhost.home.nl -
872739963.007 1299 10.0.0.21 ERR_CLIENT_ABORT/304 83 GET
https://www.facebook.com/?ref=tn_tnmn - DEFAULT_PARENT/localhost.home.nl
-
```

Figura 5.9 Resultado del uso del comando *grep* usado en la figura 5.8.

Un servidor proxy puede configurarse para que monitoree o escuche el tráfico de la red en distintos puertos, además del puerto configurado por default (3128), para ello se emplea la configuración que a continuación se muestra (Figura 5.10).

```
http_posdrt 3128 8080 9000
```

Figura 5.10 Sentencia para la configuración monitoreo por puertos.

Una vez realizada la configuración el *squid* podrá monitorear todo el tráfico a través de los puertos 3128, 8080 y 9000.

5.2.3 Analizador de tráfico

Otra manera de monitorear el tráfico de la red de datos, es a través de un analizador de tráfico. Este software tiene la posibilidad de segmentar el tráfico por protocolo, dirección IP de origen y dirección IP de destino. Una vez que se ha configurado el servidor proxy como un dispositivo antes del firewall es posible instalar un software de análisis de tráfico como lo pueden ser *Wireshark, Capsa free, Zenoss Core* y *NetworkMiner;* en este caso se utilizará W*ireshark* debido a que su interfaz es de fácil uso e interacción con el usuario (Figura 5.11).

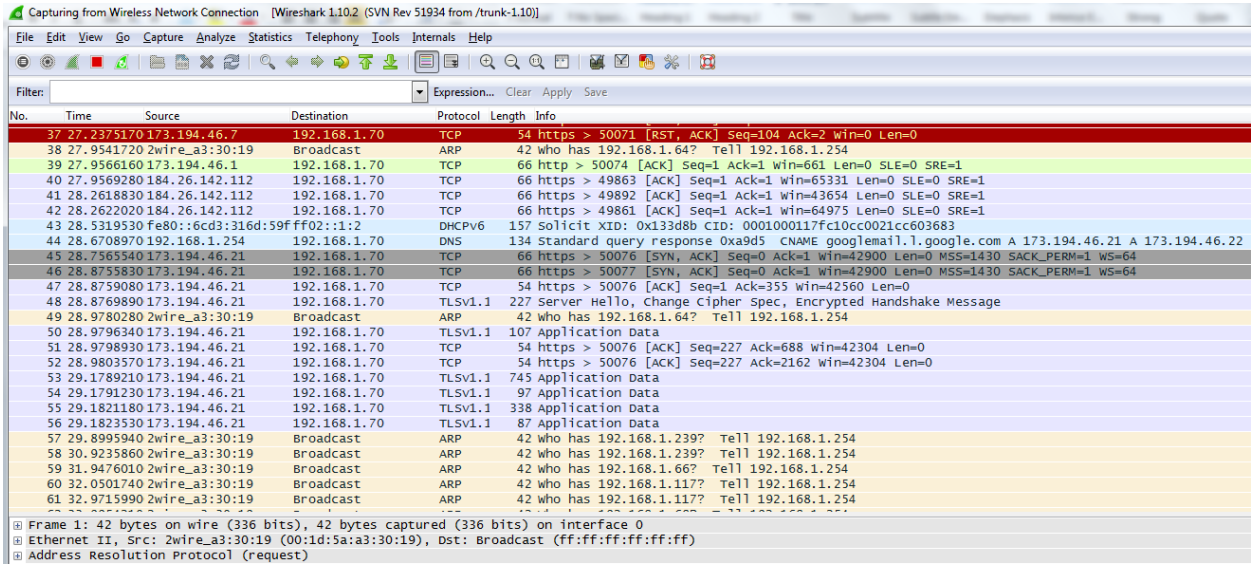


Figura 5.11. Captura de paquetes realizada a través de Wireshark.

Wireshark detectará el tráfico del puerto de red seleccionado y así brindará información más detallada tanto del protocolo que se está utilizando, como de las direcciones de origen y destino; además mostrará las banderas del protocolo de comunicación e información adicional que podrá ser de mucha ayuda en momentos en donde no se puede detectar a simple vista el problema por el cual la comunicación no se ha establecido correctamente.

Se puede obtener información más detallada con el uso de analizadores de tráfico o con las capturas que se pueden realizar en los squid, para ello hay que entender y conocer los tipos de banderas que el

protocolo de TCP mostrará en el mismo instante que se realice el análisis del tráfico. Además hay que identificar cada una de las acciones que se estarán realizando con el simple hecho de ver la bandera.

5.2.4 Capturas

Finalmente hay que realizar el análisis del tráfico capturado; esto se podrá realizar conociendo las banderas de TCP, las cuales vienen dentro de las cabeceras de los paquetes y son de 1 bit de longitud. Éstas además pueden tener un valor de 0 o 1 dependiendo si se encuentran activadas o desactivadas. Todo esto es a través del protocolo TCP/IP el cual es encargado de establecer la comunicación entre los dos hosts que intercambiarán la información y también de transportar los paquetes de datos del host origen al host destino.

Las banderas de TCP que se encontrarán dentro de una captura de datos TCP son las siguientes:

a) **SYN (synchronize – sincronizar):** Esta conexión se utiliza para iniciar una conexión TCP. Una vez que se envía un paquete con esta bandera activa un puerto X de cualquier host de la red, existen tres posibilidades de respuesta.

1. Si el puerto está abierto, la máquina enviará como respuesta dos banderas SYN y ACK, con el fin de que el host responda con una bandera ACK y se complete el protocolo *three-way handshake*.
2. Si el puerto está cerrado, el host receptor responderá con un paquete que tenga activadas las banderas RST y ACK.
3. En caso de que el puerto esté filtrado, no se recibirá nada.

En el ejemplo se puede observar cómo se envía un paquete con la bandera SYN activada al puerto 21 del host 192.168.1.9 y éste mismo responde el paquete con las banderas SYN y ACK (Figura 5.12).

```
# hping2 -c 1 -S -p 21 192.168.1.9
HPING 192.168.1.9 (en0 192.168.1.9): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.9 ttl=64 DF id=37670 sport=21 flags=SA seq=0
win=32768 rtt=0.4 ms
--- 192.168.1.9 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.4/0.4 ms
```

Figura 5.12 Captura realizada en sistema Linux al host 192.168.1.9 con las banderas SA activas.

En este ejemplo se puede identificar cómo se envía un paquete con la bandera SYN en activo al puerto 80 del host 192.168.1.9 y éste responde con otro paquete con las banderas RESET y SYN activas, lo que demuestra que el puerto 80 del host destino se encuentra cerrado o inactivo(Figura 5.13).

```
# hping2 -c 1 -S -p 80 192.168.1.9
HPING 192.168.1.9 (en0 192.168.1.9): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.9 ttl=64 id=38005 sport=80 flags=RA seq=0 win=0
rtt=0.6 ms
--- 192.168.1.9 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.6/0.6/0.6 ms
```

Figura 5.13 Captura realizada en sistema Linux al host 192.168.1.9 con las banderas RA activas.

En este ejemplo se puede observar un paquete con la bandera SYN activa al puerto 3306 del host 192.168.1.9 al cual no se obtuvo respuesta alguna, dejando demostrado que el servicio no estaba activo o bien que el puerto está filtrado (Figura 5.14).

```
# hping2 -c 1 -S -p 3306 192.168.1.9
HPING 192.168.1.9 (en0 192.168.1.9): S set, 40 headers + 0 data bytes
--- 192.168.1.9 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figura 5.14 Captura realizada en sistema Linux al puerto 3306 host 192.168.1.9 sin obtener respuesta.

b) **ACK (acknowledgement – reconocimiento):** Esta bandera se utilizará con el fin de enviar notificaciones al host emisor o receptor, dependiendo del sentido que tenga la comunicación. Si se envía un paquete con bandera ACK "no solicitado" a un puerto *x* de cualquier host, éste debería de ser respondido con otro paquete con bandera RST, sea cual fuere el estado del puerto.

   En el ejemplo que se muestra se observa que se envía un paquete con la bandera de ACK activa, pero al ser un paquete no solicitado la respuesta del host destino devuelve el paquete con la bandera RST. Esto ocurre cuando la comunicación no se inicia con un paquete con bandera SYN (Figura 5.15).

```
# hping2 -c 1 -A -p 21 192.168.1.9
HPING 192.168.1.9 (en0 192.168.1.9): A set, 40 headers + 0 data bytes
len=46 ip=192.168.1.9 ttl=64 id=38407 sport=21 flags=R seq=0 win=0
rtt=0.7 ms
--- 192.168.1.9 hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.7/0.7/0.7 ms
```

Figura 5.15 Captura realizada en sistema Linux al host 192.168.1.9 con la bandera R activa.

c) **RST (reset – reinicializar):** Esta bandera solamente se enviará para reinicializar la conexión debido a paquetes corruptos o a paquetes con la bandera SYN duplicados, inclusive retardados. Un paquete que contenga la bandera RST activa y que no esté siendo solicitado simplemente será ignorado.

Cuando el host destino recibe un paquete con la bandera RST activa el cual no haya solicitado, simplemente lo ignorará (Figura 5.16).

```
# hping2 -c 1 -R -p 21 192.168.1.9
HPING 192.168.1.9 (en0 192.168.1.9): R set, 40 headers + 0 data bytes
--- 192.168.1.9 hping statistic ---
1 packets tramitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figura 5.16 Captura realizada en sistema Linux al host 192.168.1.9 sin obtener respuesta de regreso.

d) **PSH (push – presionar):** Esta bandera únicamente se utilizará para forzar el envío inmediato de datos tan pronto como sea posible. Si el host emisor envía un paquete con esta bandera activa, el host receptor sabrá que tiene que entregar los datos inmediatamente a la aplicación receptora sin la necesidad de colocarlos en un buffer y esperar a que lleguen más datos.

e) **URG (urgent – urgente):** Esta bandera servirá para identificar un paquete de datos como "urgente". El campo de *urgent pointer* de la cabecera es un puntero que señala dónde terminan estos datos catalogados como urgentes; de esta manera se puede marcar el bloque de datos como urgente.

f) **FIN (finalize – finalizar):** Esta bandera sirve para finalizar una conexión activa entre dos hosts.

El protocolo TCP tomará los datos y los fragmentará en paquetes no mayores a 64 bits, los etiquetará con las banderas antes vistas para que el protocolo IP sea el encargado de transportarlos a su destino a través de la red. Finalmente el protocolo TCP se encargará nuevamente de ensamblarlos en el host destino para formar los datos originales. Existen otras cabeceras del protocolo IP que son importantes al momento de analizar los datos que se observan durante la captura. Estos campos son:

a) **Campo VERS (versión – versión):** este campo especifica la versión del protocolo al que pertenece el datagrama. En lenguaje binario será (0100) para IPv4 y (0110) para IPv6; el tamaño de la cabecera será de cuatro bits (Figura 5.17).

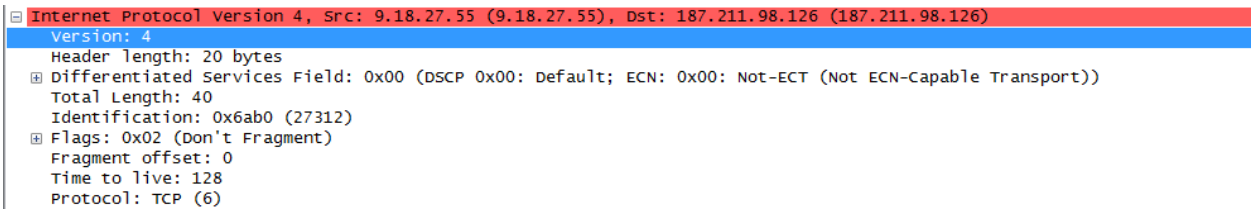


Figura 5.17 Captura obtenida del host 187.211.98.126 con el campo versión seleccionado.

b) **Campo HLEN (header length – longitud de cabecera):** este campo especificará la longitud de la cabecera en palabras de 32 bits, de forma que si este campo tiene el valor de 7 (00111) significa que la longitud de la cabecera es de 7 palabras de 32 bits, por lo tanto tendrá una longitud total de 224 bits. El valor máximo que podrá tener este campo podrá ser de 31 bits (11111) (Figura 5.18).

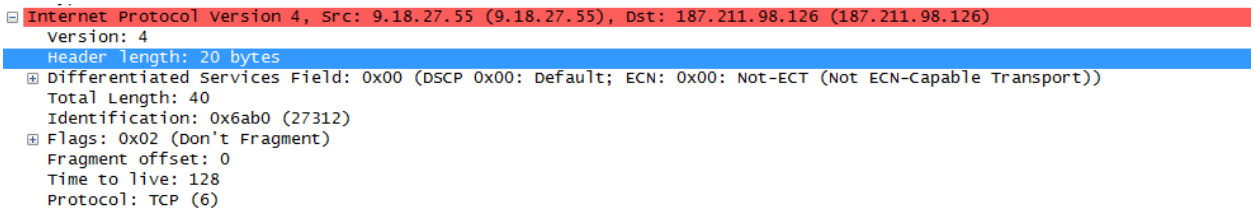


Figura 5.18 Captura obtenida del host 187.211.98.126 con el campo *header length* seleccionado.

c) **Campo TOS (type of Service – tipo de servicio):** este campo especifica el tipo de servicio con respecto a la seguridad, velocidad, fiabilidad y el tamaño de longitud del mismo será de 8 bits.

- Los primeros 3 bits (00000**XXX**) especifican la *preferencia*, el tipo de tráfico que es prioritario para el tipo de servicio que se desea.
- El cuarto bit (0000**X**000) está reservado para el *delay* (retardo), si el bit está activo el paquete solicita un bajo *retraso* en la transferencia.
- El quinto bit (000**X**0000) está reservado para el *throughput (rendimiento),* si este bit se encuentra activo el paquete será clasificado como con *alto-rendimiento* durante la transferencia.
- El sexto bit (00**X**00000) está reservado para la *reliability (confiabilidad)*, al estar este bit en activo el paquete solicitará alta fiabilidad en la transferencia.
- Finalmente los últimos dos bits (**XX**00000) son clasificados como reservados.

d) **Campo Total Length (Longitud total):** este campo especifica la longitud total de todo el paquete, incluyendo los datos. La longitud máxima que podrá tener el paquete será de 65,535 bytes; mientras que el tamaño del campo será de 16 bits (Figura 5.19).

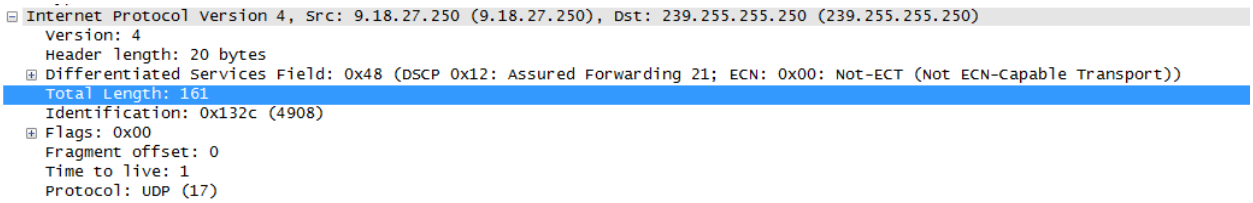


Figura 5.19 Captura obtenida del host 239.255.255.250 con el campo *total length* seleccionado.

e) **Campo ID (identificación):** este campo permite colocar un identificador único a cada datagrama con el objetivo de que se pueda volver a ensamblar al ser dividido el paquete original en fragmentos más pequeños por el protocolo TCP. El tamaño de este campo será de 16 bits (Figura 5.20).

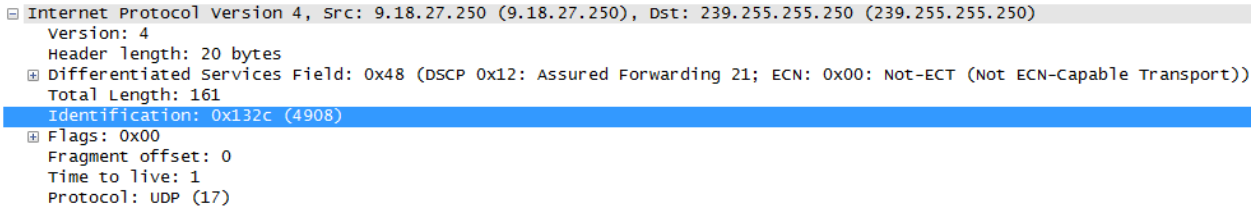


Figura 5.20 Captura obtenida del host 239.255.255.250 con el campo *indentification* seleccionado.

f) **Campo Fragmentation Offset (compensación de fragmentación):** este campo especifica la posición a la que pertenece el fragmento del datagrama. El tamaño del campo es de 13 bits y se especifica en unidades de 8 bytes, ésta es la unidad básica de fragmentación (Figura 5.21).

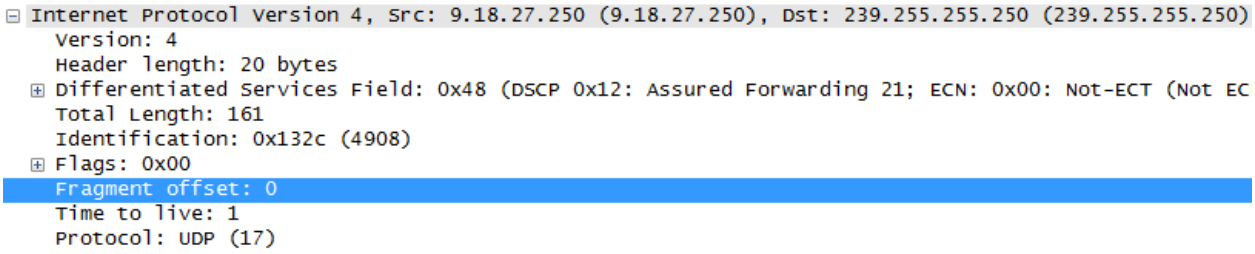


Figura 5.21 Captura obtenida del host 239.255.255.250 con el campo *fragment offset* seleccionado.

g) **Campo TTL (time to live – tiempo de vida):** este campo sirve para especificar el número máximo de saltos que da el paquete entre enrutadores. Cada vez que el paquete pasa por un enrutador el valor del campo decrementará en 1 su valor hasta llegar al host destino. Si este valor llega a 0 el paquete será descartado por el próximo enrutador que reciba el paquete (Figura 5.22).

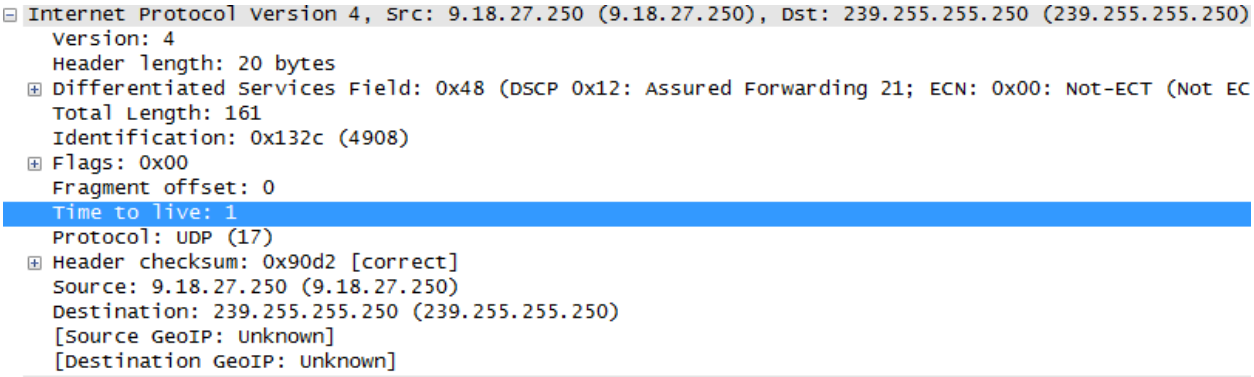


Figura 5.22 Captura obtenida del host 239.255.255.250 con el campo *time to live* seleccionado.

h) **Campo de protocolo:** este campo especifica el protocolo de transmisión que se está utilizando. El tamaño de este campo será de 8 bits; como ejemplo el protocolo TCP se identifica con el valor 6 (00000110), mientras que el protocolo ICMP con el valor 1 (00000001) (Figura 5.23).

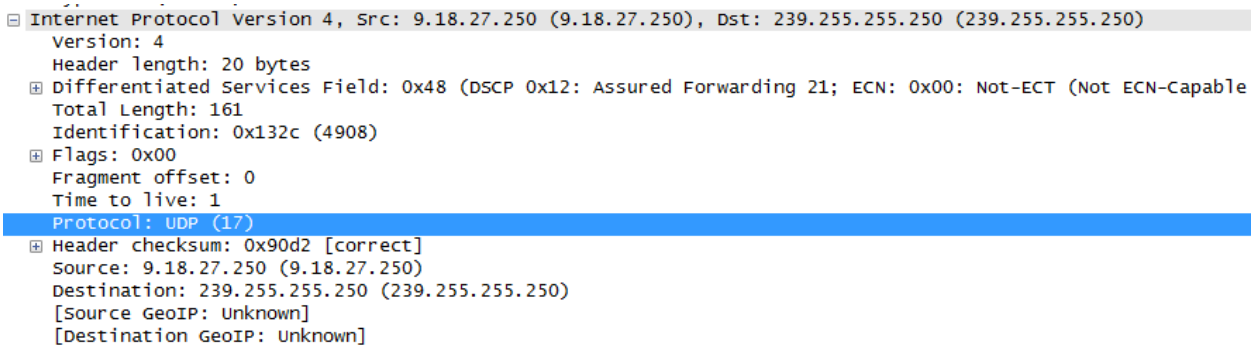


Figura 5.23 Captura obtenida del host 239.255.255.250 con el campo *protocol* seleccionado indicando que el protocolo del paquete es UDP.

i) **Campo Header-Checksum (cabecera de suma de verificación):** este campo sirve para ~~la~~ controlar errores en los paquetes, esto derivado a que se tiene que calcular nuevamente cada vez que el paquete pasa a través de un enrutador a otro. El tamaño de este campo es de 16 bits (Figura 5.24).

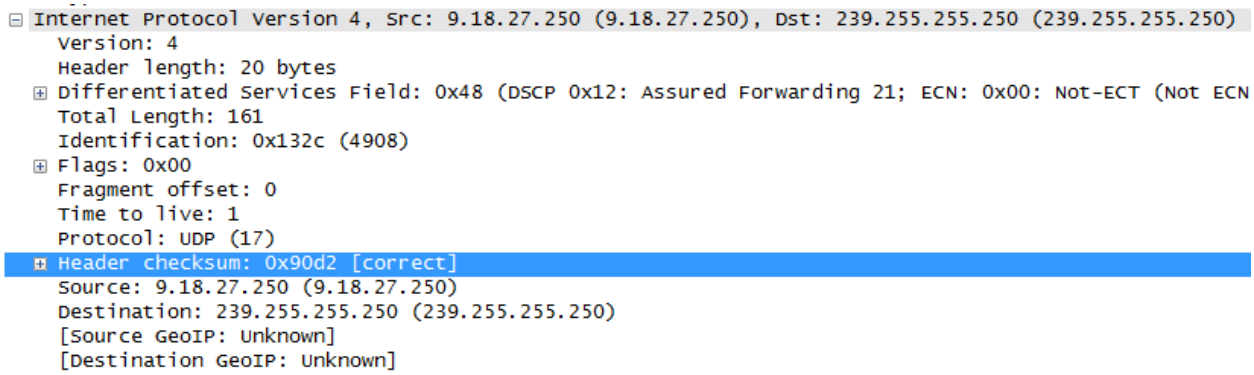


Figura 5.24 Captura obtenida del host 239.255.255.250 con el campo *checksum* seleccionado, indicando que el paquete se envió correctamente y que el *checksum* es correcto.

j) **Campo de dirección de origen y destino**: este campo indica la dirección IP del host emisor y de destino. El tamaño de cada campo será de 32 bits (Figura 5.25).

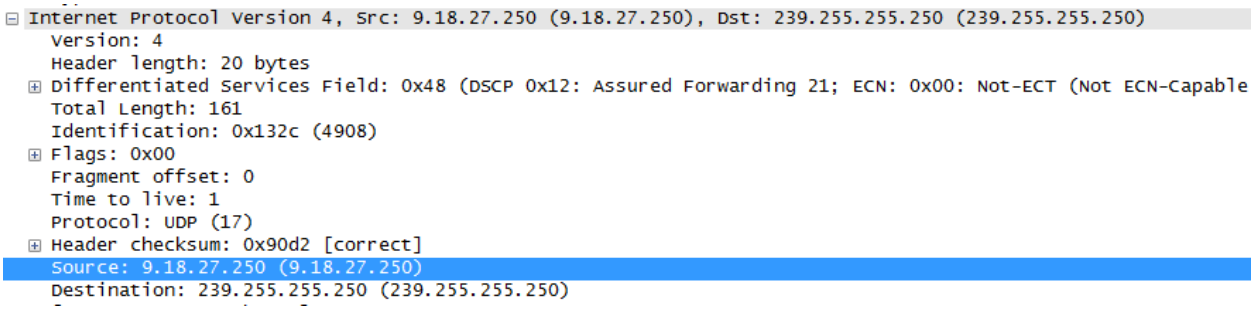


Figura 5.25 Captura obtenida del host 239.255.255.250 con el campo *source* seleccionado; la dirección IP indica cual es el host que inició la comunicación.

5.3 Configuración básica de un Firewall (Cortafuegos)

Ahora se mostrará la configuración básica de un firewall en modo software y también de uno en modo de hardware.

Se tomará un firewall del modelo PIX de CISCO para el ejemplo en hardware y para el otro caso se mostrará la configuración de un firewall en sistema Linux.

Comenzando con el firewall de tipo hardware se utiliza un modelo de CISCO debido a que este firewall permite la configuración de tres diversas zonas de seguridad: *inside, dmz y outside;* además cada una de ellas tiene asignado un valor dependiendo de los niveles de seguridad a los que estén asociadas, 100, 50 y 0 respectivamente.

Esto se debe a que el equipo cuenta con una configuración predefinida la cual dice que "*no se puede pasar de un nivel de seguridad de uno menor a uno mayor".* En caso de que se configuren listas de acceso en el equipo, éste ignorará los niveles de seguridad preestablecidos entre las zonas de seguridad y comprobará el tráfico a través de las listas de acceso.

Una ventaja con la que cuenta este equipo es que las listas de acceso pueden configurarse de manera transparente para el usuario final, porque implementa un mecanismo automático para permitir tráfico de entrada cuando se genera tráfico de salida, es decir, si se genera tráfico *http* de salida, no se podrá tener tráfico *ftp* de entrada, puesto que ya se habrá configurado la lista de acceso para el tráfico de entrada que tendrá que ser *http* para este caso en específico. Este mecanismo se conoce como CBACs (Context Based Access – Acceso Basado en contexto) funciona a través de conexiones TCP/UDP y es propiedad de Cisco Systems.

Una vez que se tengan los segmentos de red y el equipo conectado tal como vaya a quedar en funcionamiento, es necesario introducir los comandos con los que quedará configurado el equipo en la red. Para este caso práctico se describe la siguiente configuración mostrada en la Figura 5.26.

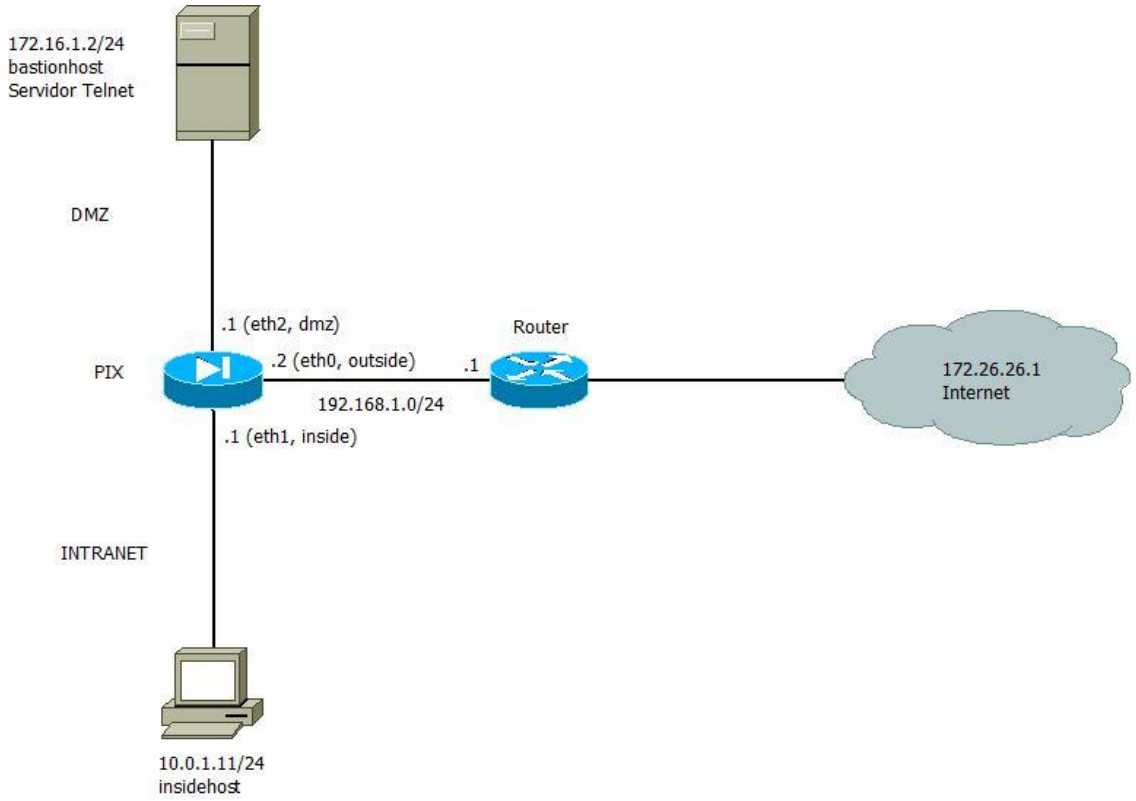


Figura 5.26. Simulación de conexión de un firewall en red.

El PIX será muy fácil de configurar, cuenta con la misma interfaz de configuración que cualquier equipo de Cisco Systems por lo que si se ha tenido que configurar cualquier equipo de la misma marca con anterioridad el proceso será mucho más rápido.

Al inicializar el equipo aparecerá un mensaje predeterminado de arranque, el cual indica si se desea entrar al modo de configuración básica.

Al entrar en el modo de configuración es necesario ingresar las siguientes sentencias para configurar el equipo (Figura 5.27).

```
Pixfirewall> enable
Pixfirewall# configure terminal
Pixfirewall(config)#
```

Figura 5.27 Comandos de configuración para un firewall PIX de CISCO.

Si se desea ver la configuración que tiene el equipo de forma predeterminada y los valores asignados se teclea la sentencia show run (Figura 5.28).

```
Pix# show run

nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 intf2 security 10

hostname Pix

names

interface ethernet0 auto shutdown
interface ethernet1 auto shutdown
interface ethernet2 auto shutdown

ip address outside 127.0.0.1 255.255.255.255
ip address inside 127.0.0.1 255.255.255.255
ip address intf2 127.0.0.1 255.255.255.255
```

Figura 5.28 Comando *show run* mostrando la configuración que está ejecutando el firewall.

Se configuran las interfaces asignando el nombre y nivel de seguridad dependiendo del perfil que se esté configurando y se comprueba (Figura 5.30).

```
Pix(config)#nameif e2 dmz security50
Pix(config)#show nameif

nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security 50
```

Figura 5.30 Configuración del perfil de *dmz* dentro del firewall PIX.

Se configura la resolución de nombres para que el equipo no consulte ningún servidor DNS externo (Figura 5.31).

```
Pixfirewall(config)# hostname Pix
Pix(config)# names
Pix(config)#name 172.16.1.2 bastionhost
Pix(config)#name 10.0.1.11 insidehost
```

Figura 5.31 Configuración de nombres de hosts permitidos dentro del firewall PIX.

Se habilitan las interfaces físicas del PIX y se corrobora que las interfaces hayan sido correctamente configuradas (Figura 5.32).

```
Pix(config)#interface e0 100full
Pix(config)#interface e1 100full
Pix(config)#interface e2 100full
Pix(config)#sh interface
interface ethernet0 "outside" is up, line protocol is up
Hardware is i82558 ethernet, address is 0090.2724.fd0f
IP address 127.0.0.1, subnet mask 255.255.255.255
MTU 1500 bytes, BW 100000 Kbit full duplex
…
interface ethernet1 "inside" is up, line protocol is up
Hardware is i82558 ethernet, address is 0090.2716.43dd
IP address 127.0.0.1, subnet mask 255.255.255.255
MTU 1500 bytes, BW 100000 Kbit full duplex
…
interface ethernet2 "dmz" is up, line protocol is up
Hardware is i82558 ethernet, address is 0090.2725.060d
IP address 127.0.0.1, subnet mask 255.255.255.255
MTU 1500 bytes, BW 100000 Kbit full duplex
```

Figura 5.32 Activación de las interfaces de *inside, outside* y *dmz*.

El siguiente paso será configurar las direcciones IP de las tres interfaces (Figura 5.33).

```
Pix(config)# ip address inside 10.1.1.1 255.255.255.0
Pix(config)# ip address dmz 172.16.1.1 255.255.255.0
Pix(config)# ip address outside 192.168.1.2 255.255.255.0
```

Figura 5.33 Configuración de las direcciones IP de cada uno de los perfiles (*inside, outside* y *dmz*) que se configuraron en el firewall.

Finalmente se tendrá que configurar una ruta por defecto en el firewall para que el tráfico pueda dirigirse a la red externa y se configuran las listas de acceso que estarán trabajando en el equipo (Figura 5.34).

```
Pix(config)# route outside 0.0.0.0 0.0.0.0 192.168.1.1 1
Pix(config)# access-list outside_access_in deny icmp any any
Pix(config)# access-list outside_access_in permit http any any
```

Figura 5.34 Configuración de las políticas de seguridad que se utilizará en el firewall.

Si se introduce el comando para comprobar la configuración básica del equipo se podrá ver en la pantalla una configuración muy parecida a la siguiente (Figura 5.35).

```
Pix(config)# write terminal

nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
…
hostname PIX
…
names
name 172.16.1.2 bastionhost
name 10.0.1.11 insidehost
…
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
…
ip address outside 192.168.1.2 255.255.255.0
ip address inside 10.0.1.1 255.255.255.0
ip address dmz 172.16.1.1 255.255.255.0
…
global (outside) 1 192.168.1.200-192.168.1.254 netmask 255.255.255.0
route outside 0.0.0.0 0.0.0.0 192.168.1.1 1
```

Figura 5.35 Comando para corroborar la configuración realizada en el firewall.

De esta manera se habrá configurado el firewall de manera básica para que se le puedan ingresar las listas de acceso basadas en la política de seguridad que se haya determinado para la red.

Por otra parte, si no se cuenta con el presupuesto necesario para la compra de un firewall con tales características como el PIX de CISCO, se puede colocar en la red un firewall de tipo software basado en un sistema de Linux.

Una de las desventajas entre los firewalls de software y hardware es que los firewalls basados en software utilizan demasiados recursos de memoria y de red, siendo éstos dos factores, una limitación en comparación a un equipo totalmente dedicado a la tarea del filtrado de paquetes.

Para demostrar la configuración básica de un firewall de tipo software, se tomará como base el servicio de *iptables* el cual se puede configurar en cualquier sistema Linux.

Lo primero que se debe hacer es instalar el servicio de *iptables* a través del comando *yum,* el cual es un administrador de software que se utiliza en las diversas distribuciones de Linux (Figura 5.36).

```
yum –y install iptables
```

Figura 5.36 Comando para la instalación de *iptables* a través del comando *yum.*

Con este comando se instalará o actualizará la última versión de *iptables* sin la necesidad de pedir autorización al administrador del equipo.

Dentro de la paquetería instalada hay reglas de destino las cuales determinan cuál será la acción a tomar para todo el tráfico que pase a través del firewall. Estas reglas de destino pueden aceptar conexiones **(ACCEPT)**, descartar conexiones **(DROP)**, rechazar conexiones **(REJECT)**, encaminado posterior de paquetes **(POSTROUTING)**, encaminado previo de paquetes **(PREROUTING)**, así como **SNAT** y **NAT**, entre otras.

Una vez instalada la paquetería de *iptables* las posibilidades de configurar el firewall pueden llegar a ser infinitas, se pueden realizar configuraciones complejas, con el fin de que el firewall analice determinados paquetes que entren y salgan de la red.

Lo siguiente será editar el archivo de las reglas que se estarán usando en la paquetería de *iptables*, para abrir el archivo en donde se encuentran las reglas hay que teclear el comando (Figura 5.37).

```
etc/iptables/iptables.rules
```

Figura 5.37 Ruta de ubicación del archivo de configuración de *iptables.*

En este archivo se encuentran almacenadas por defecto las siguientes reglas (Figura 5.38):

```
iptables –p INPUT ACCEPT
iptables –p FORWARD ACCEPT
iptables –p OUTPUT ACCEPT
```

Figura 5.38 Reglas por defecto configuradas en el archivo de *iptables.*

Por lo que será necesario configurar las reglas que se estarán ejecutando en el firewall (Figura 5.39).

```
iptables –A FORWARD –i eht1 –o eth0 –j ACCEPT
iptables -A INPUT -i eth0 -s 192.168.0.0/24 -j DROP
iptables -A POSTROUTING -t nat -s 192.168.0.0/24 -o eth0 -j SNAT 201.87.65.1
iptables -A OUTPUT -d 10.0.10.234 -s 192.168.0.0/24 -j REJECT
```

Figura 5.39 Reglas configuradas en el archivo de *iptables*.

Finalmente una vez que se hayan incluido todas las reglas que estará ejecutando el firewall dentro del archivo de *iptables,* se deberá utilizar el siguiente comando para guardar las modificaciones (Figura 5.40).

```
service iptables save
```

Figura 5.40 Comando para salvar los cambios en el archivo *iptables.*

Para ejecutar por primera vez el servicio de *iptables*, debe utilizarse el siguiente comando (Figura 5.41).

```
service iptables start
```

Figura 5.41 Comando para iniciar el servicio de *iptables.*

Y para hacer que las modificaciones en el archivo de las reglas que ejecutará el firewall surtan efecto, se ingresa el siguiente comando (Figura 5.42).

```
service iptables restart
```

Figura 5.42 Comando para reiniciar el servicio de *iptables.*

Si se desea detener el servicio por el motivo que sea, el comando siguiente ayudará a que el firewall deje de filtrar los paquetes (Figura 5.43).

```
service iptables stop
```

Figura 5.43 Comando para detener el servicio de *iptables.*

Es importante recalcar que existen reglas específicas para la creación de una regla de *iptables,* las opciones más comunes son:

a) –A; añade una cadena a la sentencia creada.
b) –i; define la interfaz de entrada en la que se estará filtrando el tráfico.
c) –o; define la interfaz de salida.
d) –j; establece una regla de destino del tráfico, que puede ser **ACCEPT, DROP** o **REJECT.**
e) **–**m; define que se aplica la regla si hay una coincidencia específica.
f) **- -**state; define una lista de los estados de las conexiones, éstas deben estar separadas por comas (**INVALID, ESTABLISHED, NEW, RELATED**).
g) **- -** to- source- port; define que la dirección IP a la que se envía un reporte del tráfico externo.
h) **–**s; define el tráfico de origen.
i) **–**d; define el tráfico de destino.
j) **- -**source-port; define el puerto desde el cual se origina el tráfico.
k) **- -**destination-port; define el puerto desde el cual se dirige el tráfico.
l) **–**t; define la tabla a utilizar, la cual puede ser *nat, filter, mangler* o *raw.*

Cuando finalice la configuración del servicio de *iptables* en el equipo que se haya instalado, éste comenzará a funcionar como un firewall. Lo importante es tener en cuenta que todo depende de las reglas que se configuren dentro del mismo firewall, éstas serán las que brinden la seguridad a la red de todo el tráfico malicioso que llegue de red externa.

5.4 Cisco ASA (Adaptive Security Appliance – Equipo de Seguridad Adaptiva)

Los firewalls de la familia ASA son equipos que se adaptan a las necesidades del cliente, hoy en día las redes corporativas están cambiando de una manera nunca antes visto en la historia, por lo que sus usuarios necesitan tener acceso a la intranet desde cualquier lugar y sin importar el tipo de dispositivo que se utilice.

Los equipos ASA de CISCO ofrecen un control sobre aplicaciones, así como de las capacidades de identidad de usuario con el fin de mejorar la visibilidad y el control del tráfico de la red en comparación con firewalls de sus competidores directos. Además de estas características los equipos ASA permiten a los administradores:

- Tener el control de comportamientos específicos dentro de aplicaciones permitidas.
- Restringir el acceso a ciertas páginas de internet, así como de aplicaciones web basándose en la reputación del sitio.
- Protección de manera proactiva contra amenazas de internet.
- Políticas diferenciadas en función del usuario, dispositivo, función, tipo de aplicación y perfil del riesgo.

El equipo ASA de Cisco cuenta con el Cisco Prime Security Manager el cual se trata de una solución de gestión integral que permite tener una mayor visibilidad de la red, además de proporcionar una aplicación detallada dependiendo del usuario, el comportamiento, las políticas y el control de los dispositivos utilizados dentro de la red. También proporciona a los administradores de la seguridad en la red la visibilidad de extremo a extremo a través de la red. Este equipo proporciona una mayor visibilidad en el tráfico que circula por toda la red, proporciona información sobre el tipo y la ubicación de un dispositivo móvil antes de que éste pueda acceder a la red.

Con éstas y otras tecnologías de seguridad en toda la red, los equipos ASA de CISCO brindan una serie de servicios para un mayor control de la seguridad:

- Autenticación Robusta:
  Además de los métodos de autenticación pasiva con aplicativos de Windows, cuenta con Lightweight Directory Access Protocol (Protocolo Ligero de Acceso a Directorios), Kerberos y Windows NT LAN manager, que se utiliza para proporcionar una autenticación activa para el acceso a la red.

- Información del dispositivo:
  Los sistemas ASA cuentan con *Cisco AnyConnect* que proporciona información sobre los tipos de dispositivos y usuarios que intentan acceder a la red, así como todo aquel dispositivo que se encuentre de forma local o remota, permitiendo a los administradores de la red proteger y controlar el acceso a los activos de la red interna.

- Reputación basada en defensa contra amenazas:
  Los equipos cuentan con inteligencia contra amenazas, la cual funciona a través del CISCO SIO para analizar un tercio del tráfico de internet y de correo electrónico para determinar vectores de amenazas web. Estos vectores son utilizados por Cisco IPS para ayudar a reducir el riesgo y la exposición a amenazas con protección en tiempo real contra amenazas conocidas o de día cero.

Todas estas características hacen que los sistemas ASA brinden una mayor seguridad en zonas desmilitarizadas y así mantener los activos lejos de las amenazas potenciales como robo de información, gusanos, virus y demás amenazas. Esto es debido a la arquitectura con la que cuentan los sistemas ASA de CISCO incluyen soporte a las capas 3 y 4 soportando traducción de direcciones de sistema de red, control de acceso, así como mantener políticas de firewall de inspección de estado para cumplir las reglas configuradas en el firewall. En la siguiente tabla (Tabla 5.1) se muestran las características y beneficios de los sistemas ASA.

Característica	**Beneficios**
Conciencia de aplicación	Hace cumplir la política de acceso de red basándose en más de 1,200 aplicaciones de uso común y 150,000 aplicaciones micro, proporciona un control de acceso basado en el comportamiento de la aplicación para aplicar un control de usuario, controles de puerto y el protocolo de salto de todas aquellas aplicaciones que pueden evadir los controles de seguridad básicos.
Seguridad basada en la identidad	Proporciona un control de acceso diferenciado en función del usuario y el rol de usuario, además de apoyar los mecanismos de identidad como Windows Active Directory, LDAP, Kerberos y Windows NT LAN Manager.
Aplicación basada en el tipo de dispositivo	Utiliza el cliente Cisco AnyConnect para identificar el tipo de dispositivo que está intentando acceder a la red (incluyendo dispositivos móviles y tabletas), y controla cuáles son los dispositivos que tienen acceso a la red y cuáles tienen restricción de navegar por ella.
Filtrado de URL	Permite un control preciso del tráfico de internet a través de filtrado URL.
Prevención de Intrusiones	Detecta y bloquea amenazas nativas de internet que tienen como objetivo a usuarios finales, así como sus dispositivos personales. Protegen el borde final de internet y reduce la complejidad a través de políticas simplificadas las cuales se encuentran integradas en el sistema ASA.
Inteligencia contra amenazas globales	Utiliza el sistema de huella global de implementaciones de seguridad de Cisco para una protección de la red de una manera más completa. Permite protección contra fuentes de amenazas en tiempo real y contra malware de día cero.
Capacidades de inspección de estado	Realiza el seguimiento de las conexiones de red (TCP o UDP) que viajan a través del firewall y distingue todos los paquetes que son legítimos para los diferentes tipos de conexiones. Todos aquellos paquetes que coincidan con una conexión activa conocida se les permitirán el paso a través del firewall; todas las demás conexiones

	serán rechazadas.
Solución de gestión intuitiva	Los sistemas ASA incluyen Cisco Prime Security Manager, una solución de gestión de gran alcance e intuitiva que simplifica la administración de servidores de seguridad.

Tabla 5.1 Características y beneficios de los sistemas ASA.

Capítulo 6:
La Zona Desmilitarizada (DMZ)

Una tesis es un trabajo de perseverancia,y para concluirla son necesarias cuatro cosas; la primera es un objetivo definido; la segunda es mentalizar un plan para concluirla; la tercera es una mente blindada contra las influencias negativas y desalentadoras; y la última es una amistad con personas que te animen a seguir adelante.

-Emmanuel Cuevas-

6.1 Antecedentes[2]

DevApps es una compañía dedicada al desarrollo de aplicaciones para móviles, fue creada en el 2009 con apenas 5 personas, hoy en día cuenta con más de 400 empleados en 7 países. Debido al extenso portafolio de aplicaciones con las que cuentan y la enorme información que generan sus sistemas, han decidido dejar de rentar un servicio de *cloud* a un tercero para implementar su propia red e incluir una DMZ *in-house* con el fin de tener mayor control en la información que manejan de sus clientes.

El CIO de DevApps se ha dado cuenta que tiene más de 200,000 visitas a su servidor por día, que por cierto rentan, y este número llega a aumentar a 300,000 en horas pico. Durante el último mes su servidor se ha caído hasta 7 veces por semana debido a la alta demanda de descargas y visitas que reciben. El servicio que renta a un proveedor externo es un servicio de 80% de disponibilidad y en el contrato que firmó quedó estipulado que es un servicio *nice-to-have.* No pueden demandar a la compañía por el tiempo que su servicio este abajo.

Es por eso que se evaluó la situación y llegó a la conclusión de que paga $70,000.00 dólares al mes a la compañía que le da el servicio de *outsourcing*. Durante el 2009 el mayor número de usuarios que entraban a sus sistemas era de 1,000 visitantes al día; se ha dado cuenta de que en 5 años las visitas han crecido un porcentaje considerable y esperan llegar a los cinco millones de visitas en menos de un año.

Además de todo lo anterior, la compañía cuenta con aplicaciones críticas las cuales brinda servicio a instituciones bancarias y que al estar en un sistema en *cloud*, tienen un nivel de criticidad muy elevado comparado con otras aplicaciones de su portafolio de ofrecimientos.

Dentro de sus ofrecimientos tienen aplicaciones bancarias, de redes sociales y juegos, todas éstas adaptadas para los usuarios de los países en los que tienen presencia.

Debido a esto al tomar la decisión de implementar su propia red de DMZ, se ha definido que se instalen tres servidores con los que puedan brindar los servicios más importantes que los usuarios solicitan y que tendrán una escalabilidad de 40% al año.

El CFO de la compañía ha destinado como presupuesto inicial para el proyecto $500,000.00 dólares poniendo como un presupuesto tope $1,000,000.00 de dólares. Él mismo realizó la viabilidad del proyecto y vio que la inversión inicial podía recuperarla en los próximos 6 meses si se ahorraba la renta del servicio del tercero. El proyecto estaría funcionando en tres meses, por lo que tendría que pagar $210,000.00 dólares de la renta de servicios; dinero que ya está incluido en el monto inicial del proyecto.

[2] Los antecedentes aquí descritos fueron propuestos por un arquitecto en redes de datos, Ing. Alfonso Alejandro Reyes Jiménez, que cuenta con la especialidad en arquitectura para zonas desmilitarizadas desarrolladas para industrias de alta disponibilidad. Al iniciar el capítulo con este apartado, podrá ser más fácil un ejemplo práctico para aplicar todo el conocimiento adquirido durante todo el tiempo invertido para desarrollar este proyecto de tesis.
Con información proporcionada de forma estadística, ésta se pudo tomar como ejemplo para utilizarla en este proyecto con fines demostrativos.

Uno de los tres servidores está destinado para su sitio web, en el que reciben miles de visitas para hacer descargas de sus aplicaciones; también cuentan con una conexión directa a la base de datos de una importante compañía de celulares con la cual actualizan las versiones que se encuentran en los servidores de la compañía de celulares.

El segundo servidor tiene como objetivo almacenar las bases de datos, con el cual los usuarios que descarguen sus aplicaciones tendrán que darse de alta en sus servidores y podrán descubrir quién de sus amigos tiene la misma aplicación.

Finalmente el último servidor cubre la demanda para el servicio de DNS para toda la intranet que se conecta a la red externa y reducir la latencia para acceder a ciertas páginas web, además de ayudar a reducir el tráfico en la red hacia internet. Por lo que a través de este capítulo se demostrará la instalación de la DMZ y la configuración de la topología de manera física y lógica.

6.2 Introducción

Una Zona Desmilitarizada (DMZ) funciona como una zona aislada entre la red interna e internet con el fin de mantener seguros aquellos activos que son importantes para la compañía u organización. Esto es posible, gracias a que se colocan servidores y servicios que pueden ser accesibles desde internet dentro de este segmento de red, mientras que se mantiene aislada la red interna para que se mantenga un nivel de seguridad dentro de la red.

En capítulos anteriores se mencionó que es importante contar con políticas de seguridad, las cuales marcan la línea base en la que todos los usuarios de la red se guían para un mejor uso de los activos. Estas políticas de seguridad se utilizarán como base para la configuración de los equipos de la red como, routers, switches y firewalls, entre otros.

Algunas de las políticas de seguridad que se manejan de manera pre-determinada dentro de una DMZ son:

a. El tráfico de internet a la DMZ está permitido.
b. El tráfico de internet a red interna está prohibido.
c. El tráfico de red interna a DMZ está permitido.
d. El tráfico de red interna a internet está permitido.
e. El tráfico de DMZ a red interna está prohibido.
f. El tráfico de DMZ a internet está denegado.

Además de las políticas de seguridad que serán utilizadas para el filtrado de paquetes, la zona desmilitarizada debe contar con un mapeo bidireccional. Esto es necesario cuando se tiene una gran cantidad de equipos que se necesita se conecten a internet; el sistema NAT permite que todas las computadoras de un determinado segmento de red privado se conecten a internet a través de una sola dirección IP pública, de esta forma se tendrá un mayor control en el acceso a internet y se reducirán costos al contar con solo una dirección IP pública como acceso a internet.

Sin una política de seguridad bien definida, el diseño o implementación de una DMZ será ineficaz y no será rentable, porque tendrá que reconfigurarse y adaptarse a las necesidades de seguridad de la organización, lo cual costará tiempo y dinero.

Se deben tener en cuenta ciertas consideraciones especiales mientras se está trabajando en el diseño e implementación de la DMZ, a continuación se listarán algunas de ellas:

1. El uso de FTP en general que no esté restringido podría producir un incumplimiento de seguridad. Todo el servicio de FTP deberá estar prohibido desde la red interna.
2. El diseño de la DMZ permite por sí mismo el control de servicios innecesarios que podrían estar en la red externa. Por ejemplo, se puede incorporar el bloqueo de los puertos de salida a servicios que faciliten la mensajería instantánea, redes no empresariales, entre otras restricciones según sea el sistema.
3. Los puertos conocidos deberán ser bloqueados desde la red interna.

El diseño sencillo de la DMZ permitirá realizar una perfecta asociación entre la configuración lógica de la DMZ y comparándola con el diseño físico que se realizó previamente. Esto ayuda a que los administradores de la red puedan conocer y detectar aquellos puntos débiles o fallas que podrían ser causadas por un flujo de datos mal configurado.

A continuación se presenta el diseño de la DMZ (Figura 6.1) el cual será explicado brevemente y será causa de estudio para su implementación de manera lógica.

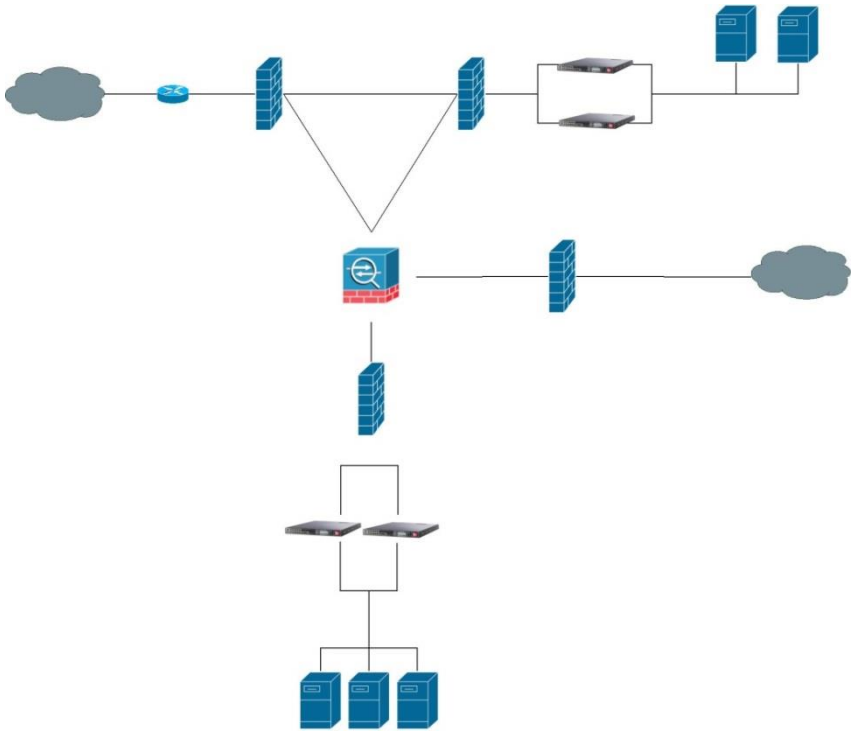


Figura 6.1 Diseño físico de la DMZ.

Como puede observarse en la figura 6.1, la DMZ cuenta con 4 firewalls, 4 balanceadores de carga que estarán ayudando a mantener los servicios activos, 5 servidores (Figura 6.2), los cuales estarán brindando diferentes servicios dentro de la red, un router que estará en el borde de la red y estará conectado directamente al ISP y finalmente un Firewall ASA que ayudará a enrutar y analizar todos los flujos de datos que lleguen a configurarse en el futuro.

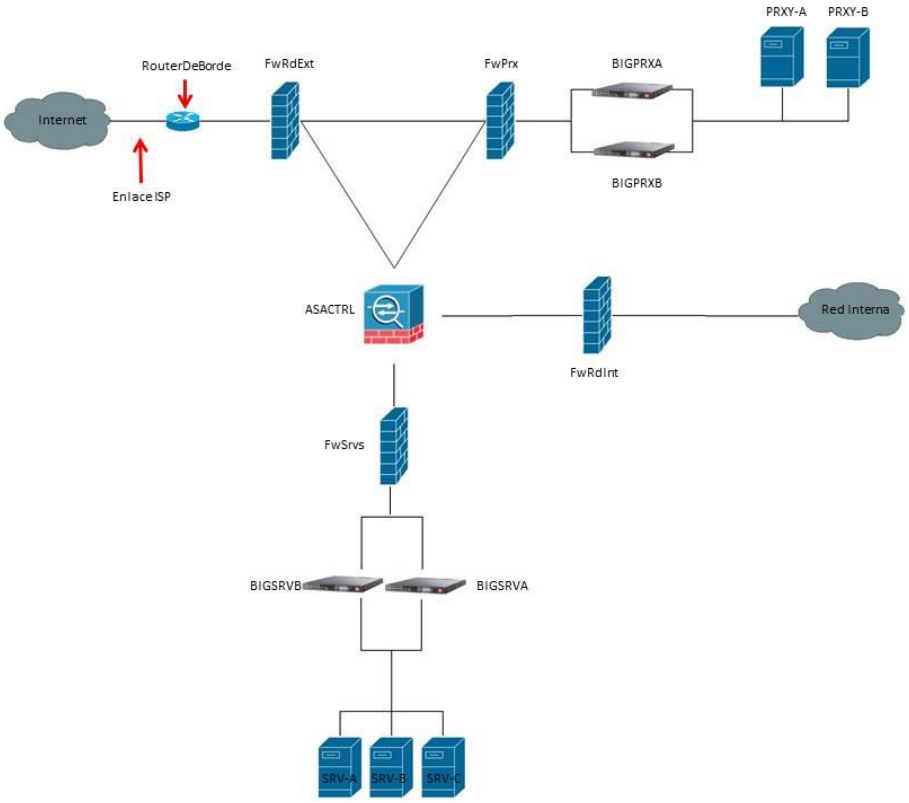


Figura 6.2 Nomenclatura de los equipos dentro de la DMZ.

Este diseño propuesto permitirá mantener la página web de la corporación en uno de los servidores que tendrá un servicio web, otro de los servidores estará configurado para brindar un servicio de base de datos que se encuentre ligado con el portal web y finalmente un servidor albergará un servicio de DNS. Por otra parte, existirán dos servidores más, los cuales estarán configurados para brindar el servicio de proxy, con ello se podrá tener un mejor control para que los usuarios de red interna puedan acceder a páginas en internet.

El diseño mostrado en la figura 6.3, muestra el flujo del tráfico que estarían llevando a cabo los datos dentro de la DMZ. Este flujo de datos se puede lograr a través de firewalls que se encuentran en la red además de la funcionalidad del NAT que será provista a través de un ISP que se conectará en el router de borde y el cual estará brindando el acceso a internet.

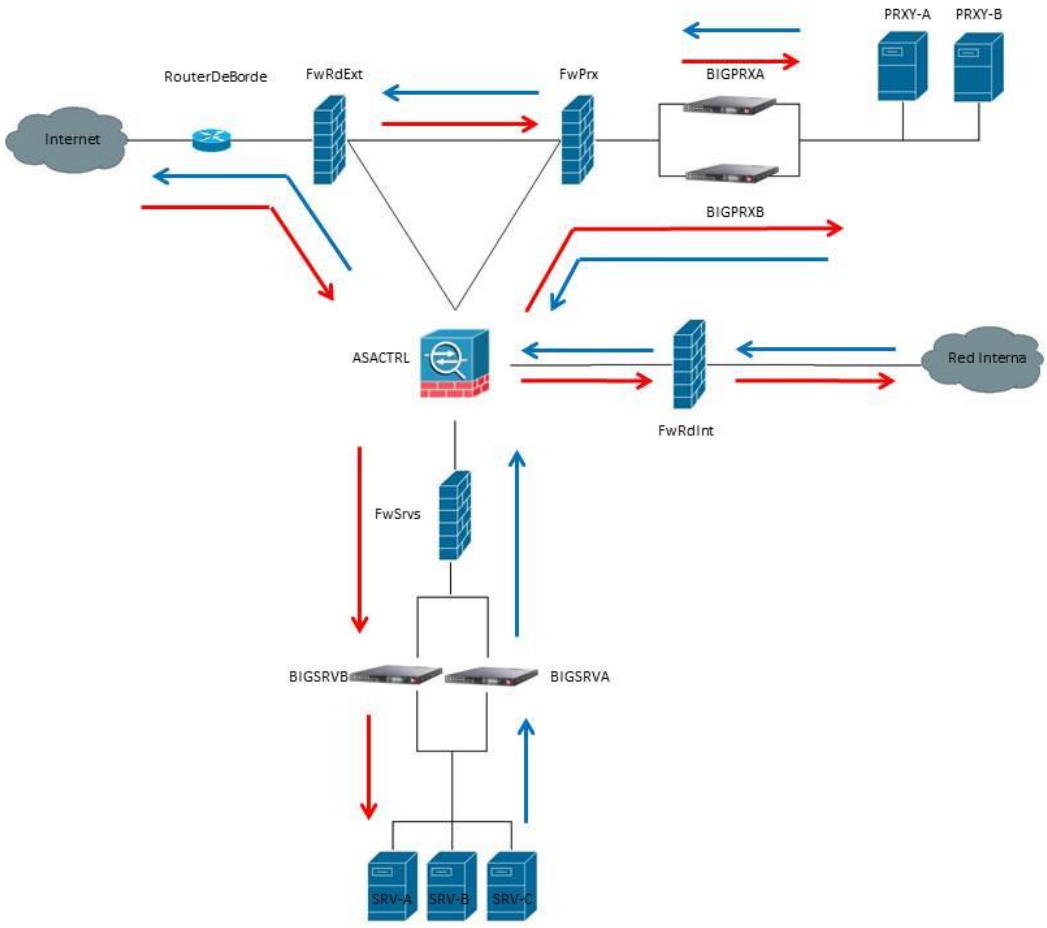


Figura 6.3 Flujo del tráfico de la DMZ

Nótese que los usuarios de red interna siempre están buscando información en internet, por lo que primero se les dirige a cualquiera de los servidores proxies, en los cuales podrán corroborar sus credenciales para tener acceso a internet. Una vez que se hayan autenticado en cualquiera de los servidores podrán tener acceso a la red externa. Todo el tráfico que éstos generen será dirigido a través de los proxies nuevamente, por lo que se podrá tener un mayor control acerca de las páginas que visitan en la red externa.

Por otra parte, se tiene contemplado que todo el tráfico que proviene de la red externa será dirigido a los servidores de servicios, en los que se puede colocar un servidor de HTML, de resolución de nombres o de una base de datos. Una vez hecha la petición a los servidores regresará a la red externa; ésta no podrá dirigirse a la red interna como se muestra en la Figura 6.4.

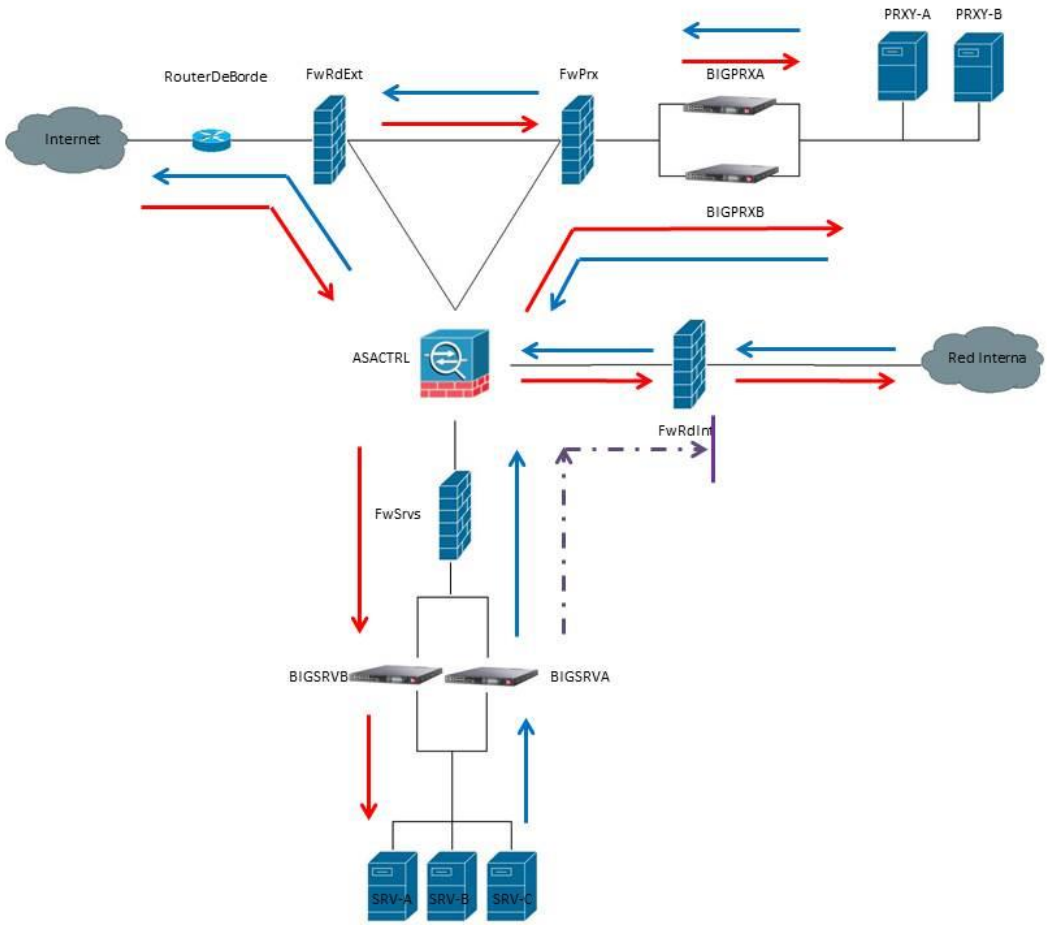


Figura 6.4 Flujo de tráfico de la red externa denegado a la red interna.

El siguiente paso será determinar los segmentos de red asignados a las diferentes terminales de cada uno de los equipos, además de tomar en cuenta que hay ciertos dispositivos que deberán contar con tres segmentos de red configurados dentro del mismo equipo; como puede ser el caso de los balanceadores de red.

Para establecer las subredes que se utilizarán es necesario tomar en cuenta los segmentos de direcciones IP para redes privadas, los cuales son los siguientes:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

Para este caso práctico se utilizarán 9 segmentos diferentes, los cuales serán tomados de cada una de las clases existentes para determinar la topología lógica que se configurará y que permitirá que la DMZ funcione de manera correcta.

Para iniciar, se determinarán los segmentos de red en los que se configurarán cada uno de los servidores de la DMZ. Hay que recordar que deben ser dos segmentos de red diferentes que albergarán servicios

distintos el uno del otro. Tomando en cuenta la posibilidad de escalar la DMZ en un futuro, hay que determinar un máximo de 254 direcciones disponibles para cada uno de los segmentos de red en los que se configurarán los servidores, utilizando una máscara de 24 bits que se representa como 255.255.255.0.

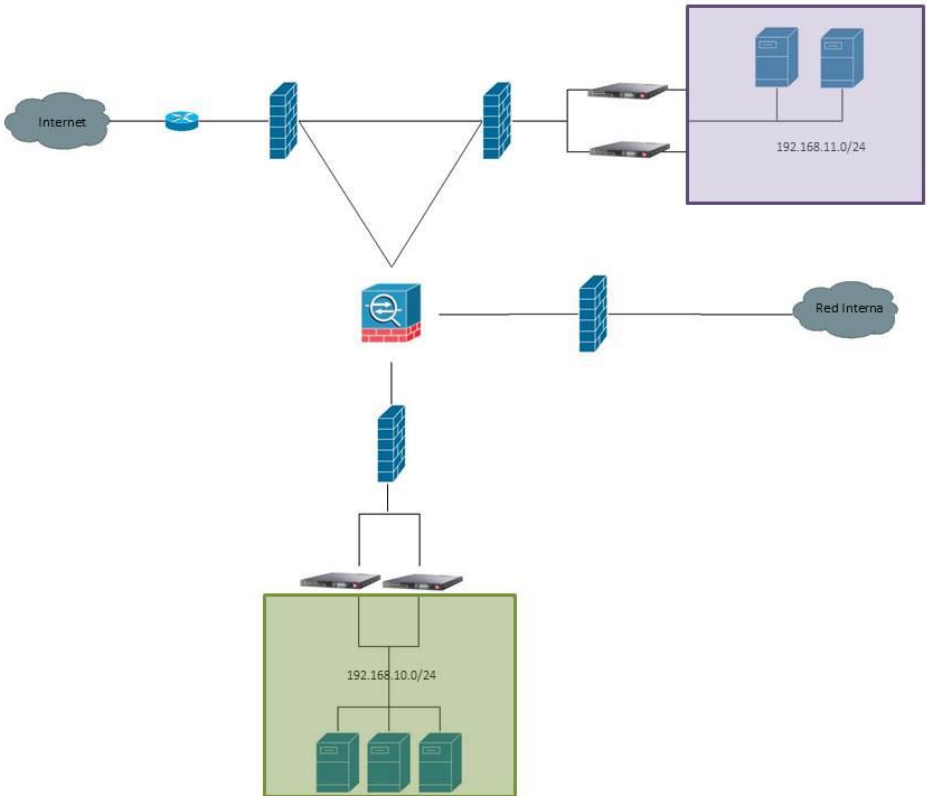


Figura 6.5 Segmentación lógica de la red.

A continuación se demuestra el proceso que se realiza para segmentar las redes elegidas en el diseño lógico y que se utilizan para albergar a los dispositivos previamente pronosticados en cada segmento de red que se obtiene al final de este sencillo proceso.

a) Seleccionando las redes 192.168.10.0 y 192.168.11.0 (Figura 6.5) de manera aleatoria, se podrá iniciar el proceso para la segmentación de manera correcta haciendo uso de la máscara de red previamente determinada. Se toma el segmento de red seleccionado y se convierte a su representación en número binario.

   192.168.10.0 = 11000000.10101000.00001010.00000000

b) Luego se toma la máscara de red y también se convierte a su representación en número binario.

   255.255.255.0 = 11111111.11111111.11111111.00000000

c) Una vez realizadas ambas conversiones, se aplica una operación **OR** a ambas direcciones en su representación binaria con el fin de determinar cuáles serán los bits que permanecerán encendidos y así determinar el número total de hosts disponibles en el segmento de red.

11000000.10101000.00001010.00000000 OR 11111111.11111111.11111111.00000000 =

11111111.11111111.11111111.00000000

d) Finalmente se conocerá que los únicos bits que podrán encenderse y apagarse para determinar las diferentes direcciones de red, son los que se encuentran en el último octeto.

11111111.11111111.11111111.**00000000**

e) Estos bits podrán tomar cualquier dirección entre el rango de 00000000 y 11111111. Encendiendo y apagándose de manera secuencial para tomar valores decimales entre 0 y 255.

f) Como último paso se toma la dirección más baja dentro del rango para determinar el que será el segmento de red. Y también se toma la dirección más alta dentro del segmento para ser utilizada como dirección de broadcast dentro del segmento de red creado.

192.168.10.0 – Segmento

192.168.10.1 a 192.168.10.254 – Rango de direcciones disponibles

192.168.10.255 – Dirección de Broadcast

Para el segmento de red 192.168.11.0 se realiza el mismo proceso para determinar el número de direcciones disponibles dentro de los segmentos recién creados.

Una vez realizada la segmentación para las sub-redes donde se configurarán los servidores se podrá continuar con el mismo proceso para las conexiones entre los equipos de red (Figura 6.6). Esto es de mucha ayuda, ya que debido a que la mayoría de las conexiones son punto a punto, la cantidad de direcciones disponibles se reduce de manera considerable.

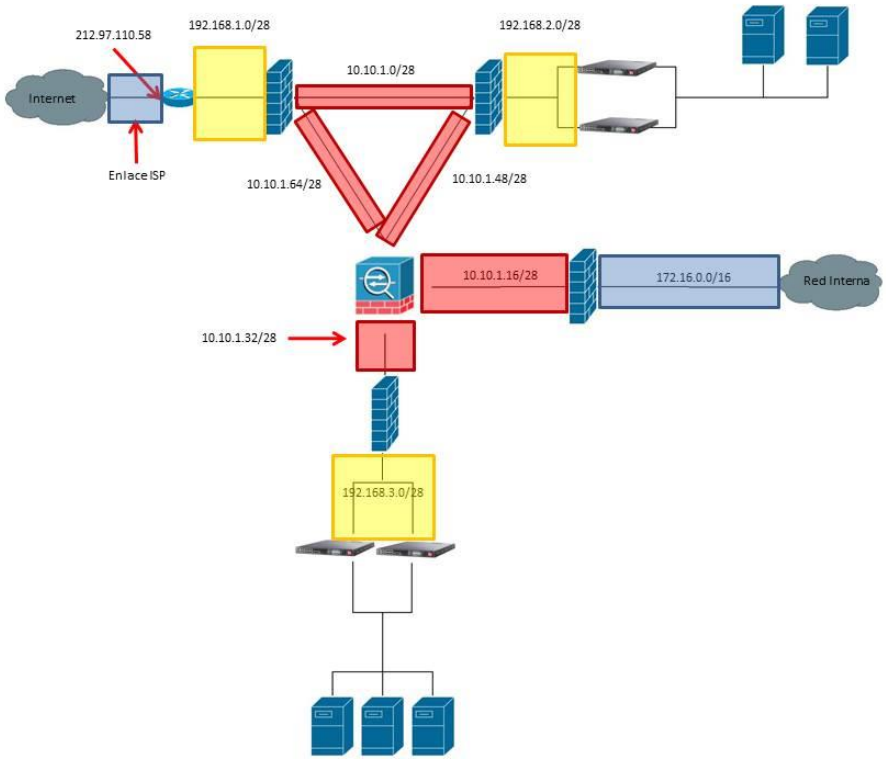


Figura 6.6. Segmentos de red seleccionados para la conexión de equipos de la DMZ.

Para este caso se toma como ejemplo el segmento de red 10.10.0.1/28 (Figura 6.7), para la creación de las subredes y poder determinar la cantidad de direcciones disponibles para interconectar los firewalls con el ASA.

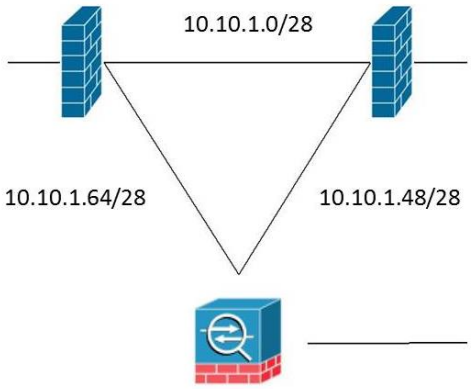


Figura 6.7. Segmento de red 10.10.1.0

Nuevamente se realizan los seis pasos que se usaron previamente para determinar las subredes que se obtienen dependiendo de la máscara de red que se determinó.

a) 10.10.1.0 = 00001010. 00001010.00000001.00000000
b) /28 = 11111111.11111111.11111111.11110000
c) 00001010.00001010.00000001.00000000 OR 11111111.11111111.11111111.11110000 = 11111111.11111111.11111111.11110000
d) El rango disponible para esta subred va desde 0000 hasta 1111 dando como resultado hasta 16 direcciones de red disponibles.
e) **Subred**
   Segmento -> 10.10.1.0
   Rango de direcciones disponibles -> 10.10.1.1 – 10.10.1.14
   Broadcast -> 10.10.1.15

Si el proceso se realiza de manera continua se podrá encontrar a todas las sub-redes que se obtienen con la misma máscara de red utilizada en el paso anterior, a continuación se realiza el proceso para obtener las primeras 4 subredes.

Subred 1
Segmento -> 10.10.1.0
Rango de direcciones disponibles -> 10.10.1.1 – 10.10.1.14
Broadcast -> 10.10.1.15

Subred 2
Segmento -> 10.10.1.16
Rango de direcciones disponibles -> 10.10.1.17 – 10.10.1.30
Broadcast -> 10.10.1.31

Subred 3
Segmento -> 10.10.1.32
Rango de direcciones disponibles -> 10.10.1.33 – 10.10.1.46
Broadcast -> 10.10.1.47

Subred 4
Segmento -> 10.10.1.48
Rango de direcciones disponibles -> 10.10.1.49 – 10.10.1.62
Broadcast -> 10.10.1.63

Subred 5
Segmento -> 10.10.1.64
Rango de direcciones disponibles -> 10.10.1.65 – 10.10.1.78

Broadcast -> 10.10.1.79

En la configuración se tomarán las primeras tres subredes para configurar los segmentos que se encuentran directamente conectados con el ASACTRL.

Se realiza el mismo proceso para los segmentos que están sombreados en color amarillo y que se muestran en la topología como *segmentos de red no críticos.*

- 192.168.1.0
- 192.168.2.0
- 192.168.3.0

Al tener todos los segmentos la misma máscara de red, se realizará el proceso para el segmento 192.168.1.0 y se realizará el mismo proceso para los otros dos segmentos de red.

a) 192.168.1.0 = 11000000.10101000.00000001.00000000
b) /28 = 11111111.11111111.11111111.11110000
c) 11000000.10101000.00000001.00000000 OR 11111111.11111111.11111111.11110000 = 11111111.11111111.11111111.11110000
d) Rango de direcciones disponibles 0000 – 1111 (0 – 15)
e) Subred

192.168.1.0 – Segmento de red

192.168.1.1 – 192.168.1.14 – Direcciones disponibles

192.168.1.15 – Dirección de Broadcast

Finalmente se realiza el proceso para determinar el segmento de red a utilizarse en la conexión entre red interna y el firewall de red interna (FwRdInt). Al terminar el proceso la subred quedará de la siguiente manera.

Subred

172.16.0.0 – Segmento de Red

172.16.0.1 – 172.16.255.254 – Direcciones disponibles

172.16.255.255 – Dirección de broadcast

Una vez que se haya realizado el subneteo correcto y se hayan asignado las direcciones a cada uno de los nodos de todos los equipos en la DMZ, la topología lógica queda de la siguiente manera (Figura 6.8).

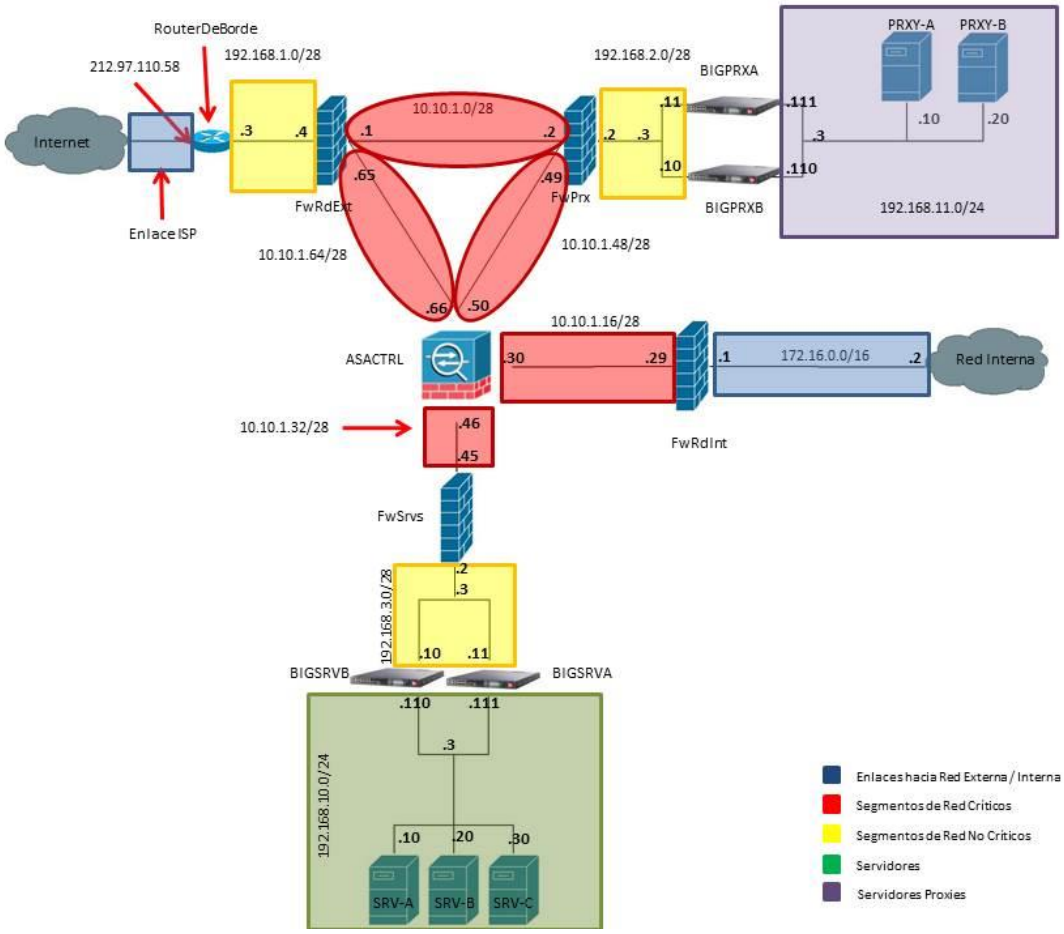


Figura 6.8 Diseño Lógico de la DMZ

La dirección que se configura como enlace al ISP en el RouterDeBorde es proporcionada por el proveedor del enlace a la red, motivo por el cual no hay necesidad de entrar a mayor detalle sobre este punto en específico.

El siguiente paso es configurar cada uno de los equipos de tal manera que tengan las direcciones de red asignadas a las interfaces que corresponden; subsecuente del paso anterior hay que configurar los protocolos de balanceo en los *balanceadores de red*, las listas de acceso dentro de los firewalls y finalmente configurar el ASACTRL para poder tener un buen control sobre la DMZ.

6.3 Configuración de los equipos de la DMZ

Lo primero que se hace es configurar todos los equipos por niveles; los primeros en configurar son los servidores y proxies. A éstos le siguen los balanceadores, los firewalls, el ASA y finalmente se continúa con los equipos de borde; router y firewall para red externa y red interna respectivamente.

Hay que tomar en cuenta que la DMZ será para un entorno empresarial sin importar el tamaño de la compañía a la que le será instalado. Con base en lo anterior se solicita la compra de 5 licencias del sistema operativo de *Red Hat Enterprise Linux Server* que son necesarias para la instalación de los demonios de cada servicio en cada uno de los equipos que se utilizan.

El proveedor es responsable de instalar cada una de las licencias en los servidores que están dentro de la DMZ, motivo por el cual la instalación del sistema operativo en los equipos queda fuera del alcance de este proyecto de tesis.

A) Servidor DNS

Para configurar el servicio de DNS hay que tener en cuenta dos cosas importantes: configurar el servicio y configurar la interfaz de red. Hay que poner atención a estos dos puntos, ya que el primer paso permite que se pueda contar con el servicio de DNS en el equipo, y después poder configurar la interfaz de red que le corresponde al equipo dentro de la topología lógica.

La instalación del servicio de DNS se realiza completando los siguientes pasos:

a) Instalar la utilería de names por medio del comando *yum.*
b) Configurar el archivo de named.conf con los parámetros necesarios para activar el servicio.
c) Realizar pruebas de conexión al servicio de DNS para confirmar que está funcionando correctamente.

Algunos de los archivos que son importantes tomar en cuenta para la configuración del servicio de DNS son:

1. named.conf[3]

   El archivo es un demonio que implementa el servicio de DNS dentro de un sistema Linux. Este archivo es parte de la paquetería BIND (Berkeley Internet Name Domain) que es una implementación de protocolos del sistema de nombres de dominio. En el fichero se indica el directorio donde se almacenan los ficheros de zonas, el tipo de resolución que el servidor va a realizar.

   La organización está formada por bloques; el primer bloque es el de *options*, donde se configura el directorio de zonas y la recursión. También se tienen bloques por cada zona que se configure

[3] https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/3/html/Reference_Guide/s1-bind-namedconf.html, Configuración y características de named.conf.

y en donde se especifican las restricciones o reglas que se cumplirán en la zona en cuestión. Todas las zonas especificadas en el fichero, a excepción de la raíz de tipo *hint*, son zonas en las que el servidor DNS será configurado de manera autoritativa.

El servidor se puede configurar de forma que no exista ninguna zona dentro del archivo named.conf, de esta manera, todas las peticiones de los clientes serán dirigidas a otro equipo de resolución de nombres, o en donde el servidor solicite a otros equipos remotos las listas de resolución de nombres de cada uno y éste las guarde en memoria caché.

2. resolv.conf[4]

El archivo resolv.conf tiene como propósito contener la información que ayuda a determinar los parámetros de funcionamiento de la resolución de direcciones a nombres. Este archivo permite correr aplicaciones en el sistema operativo para traducir nombres de dominios en direcciones IP numéricas, las cuales son requeridas para acceder a recursos que se encuentran en internet. El archivo resolv.conf es un conjunto de rutinas que proporcionan acceso al DNS. Éste contiene información que es leída por las rutinas del archivo, una vez que se invoca por primera vez un proceso. Si este archivo no estuviera en el sistema, únicamente el nombre del servidor DNS podría ser consultado, ya que el nombre de dominio se determina a partir del nombre del host y la ruta de búsqueda de dominio se construye a partir del dominio a buscar.

Una vez que se instale toda la paquetería de names para el servicio de DNS, es necesario realizar la configuración de los archivos de names.conf y resolv.conf, que fueron descritos previamente. Para iniciar es necesario arrancar el servicio de names y configurar las zonas que estarán siendo utilizadas en la configuración. Es importante recordar que después de cada configuración realizada se deben dar los permisos necesarios para que los procesos del servicio DNS puedan realizar las tareas de resolución de nombres, modificación de archivos de nombres, entre otras tareas.

Otros puntos importantes a ser tomados en cuenta son:

a. Configuración de la tarjeta de red con la dirección IP asignada al servidor de manera lógica.
b. Configuración de las listas de acceso necesarias para permitir peticiones al servidor de forma segura.
c. Configuración de los temporizadores para mantener la lista de nombres lo más actualizada.

Si toda la configuración fue correcta al momento de realizarla, hay un par de pruebas que son necesarias hacer para corroborar que todo esté funcionando correctamente.

a) Realizar un ping al servidor de DNS.
b) Realizar una consulta por medio del comando *dig* a cualquier página de internet.

La configuración detallada se observa en el Anexo B.

[4] http://man7.org/linux/man-pages/man5/resolv.conf.5.html

B) **Servidor Base de datos**

Otro de los servidores que se debe configurar en el entorno diseñado, es el que contendrá una base de datos. El uso de este servidor es básicamente para que los clientes que instalen una aplicación de Devapps en su celular, puedan reconocer cuáles de sus contactos o amigos tienen instalada la misma aplicación y agregarlos[5] de una manera más fácil.
Para que esto sea posible es necesario que cada usuario cree un perfil con los siguientes campos:

- Nombre
- Correo
- Teléfono Celular

Estos campos son los que formarán parte de la base de datos.

Básicamente un servidor de base de datos ayuda a almacenar, recuperar y administrar los datos de una base de datos por medio de un lenguaje de SQL. El servidor será encargado de gestionar toda la información que se encuentre dentro del administrador de bases de datos y podrá hacer consultas a la información que se requiera por medio de *querys*.

Las actividades que necesitan realizarse para culminar la configuración del servidor de base de datos son:

a) Descargar e instalar el demonio de mysql, que será utilizado para el servicio de base de datos.
b) Configurar el servicio de mysql.conf con los parámetros necesarios para la base de datos.
c) Crear las bases de datos
d) Crear las tablas con los elementos dentro de la base de datos.

El primer paso es la configuración de la base de datos que estará en el servidor. Para ello hay que entrar al servidor y firmarse como administrador para tener privilegios de instalación de cualquier paquetería. Existen sistemas de administración de bases de datos que proveen funcionalidades para los servidores, mientras que hay otros que solo permiten la construcción y el acceso a la base de datos.

Los objetivos de un Sistema de Gestión de Base de Datos [6]o Database Management System (DBMS) son:

- Definir una base de datos especificando los tipos de datos, estructuras y restricciones para los datos que se almacenarán en el servidor.
- Construir una base de datos dentro de algún medio de almacenamiento definido.
- Manipular las bases de datos por medio de funciones que permitan consultar, actualizar la información contenida dentro de la base de datos.

[5] La programación de la API que conecta la aplicación con la base de datos queda fuera del alcance de este tema de tesis.
[6] http://www.alegsa.com.ar/Dic/SGBD.php

Una vez que se haya instalado el sistema de administración de base de datos en el servidor, es necesario configurar los permisos, usuarios y contraseñas para poder acceder al DBMS y crear la base de datos que contendrá la información de los usuarios.

Paso seguido se necesita crear la tabla o tablas donde se guardará la información dependiendo de los campos que se hayan configurado; en este caso serán nombre, correo electrónico y teléfono celular.

Una vez que el sistema de administración de la base de datos haya sido configurado de manera correcta, se obtendrá la información por medio de query’s, las cuales son cadenas de consulta que dependiendo de la sintaxis que se utilice en la cadena, se podrá obtener información de la base de datos.

Existen cinco comandos que se utilizan para obtener información de forma ordenada al momento de realizar una búsqueda en la base de datos. Los comandos son los siguientes:

- SELECT; con este comando se seleccionan los datos que serán buscados en las tablas de las bases de datos.
- FROM; con el comando FROM se selecciona la tabla en la que serán buscados los datos.
- WHERE; con el comando WHERE se hace una búsqueda con un condicionante en la tabla seleccionada.
- GROUP BY; con este comando la búsqueda resultado será agrupada a través de un discriminante.
- ORDER BY; con el comando ORDER la información será ordenada ya sea de forma ascendente o descendente.

Aunque existen más comandos para ser utilizados en la búsqueda de información en la base de datos, por medio de una query; con los cinco comandos anteriores se puede generar una búsqueda y obtener información de manera ordenada y precisa.

Finalmente hay que tener en cuenta el tipo de campo que se utilizará en la tabla, ya que difieren mucho al momento de que la base de datos ya se encuentra configurada y funcional. En este caso solo se utilizarán dos tipos de campos en la base de datos: int y varchar.

a) En el caso de int (*integer*), es un campo de tipo numérico y pertenece a un número entero con o sin signo. El rango de valores va de -2147483648 a 2147483647.
b) Para el campo de tipo varchar, es un campo de tipo de cadena y almacena una cadena de longitud variable, la cual puede contener desde 0 hasta 255 caracteres.

Para revisar la configuración detallada del servidor de base de datos, dirigirse al Anexo B.

C) Servidor web

Un servidor web dentro de la DMZ es indispensable para los objetivos por los cuales se propuso el proyecto. Teniendo el servidor web en la DMZ, los usuarios accederán a la página web de la empresa sin tener el riesgo de poner información de red interna para que sea robada o alterada.

Para la configuración del servidor web hay que realizar una serie de pasos sencillos, con los cuales se podrá brindar el servicio:

1. Instalar la paquetería de httpd.
2. Configurar e iniciar el servicio de apache/httpd.
3. Realizar pruebas de conexión al servidor web.

El servidor con apache brindará características que son altamente confiables, ya que está desarrollado dentro del proyecto HTTP Server (httpd) de la Fundación de Software Apache. Además cuenta con autenticación de base de datos, negociado de contenidos y carece de una interfaz gráfica para su configuración.

El sistema de administración de Apache ha aumentado el número de confiabilidad, por lo que en 2005 alcanzó el 70% de sitios web que utilizan este software para albergar las páginas web empresariales. Aunque en el mercado existen otros sistemas de administración de servidores web la mayoría se concentran en cinco grandes incluyendo Apache, los otros sistemas de administración son:

- Nginx
- Internet Information Services (IIS)
- Cherokee
- Tomcat

Alguna de las ventajas por las que se utiliza más el sistema de administración de Apache, es porque brinda la posibilidad de correr aplicaciones basadas en Linux, MySQL, PHP, Phyton, Perl y Ruby. Debido a que es un sistema robusto puede cubrir las necesidades de la industria. Los servidores con el sistema de administración Apache, cuentan con su propio sistema de administración de bases de datos. Éste nos será utilizado, debido a que se tiene pensado que las bases de datos del sistema sean mayores a lo que el servidor tiene la capacidad de soportar.

Si se siguen los tres pasos mencionados previamente, se podrá conseguir que el servidor esté funcionando correctamente. En realidad la configuración de un servidor web es relativamente sencilla y puede ser realizada en un tiempo relativamente corto. Para administrar la página web es necesario contratar a un web master el cual estará a cargo de la gestión de archivos, imágenes y funciones que estén ejecutándose en el servidor.

La instalación detallada se observa en el Anexo B.

D) Proxys

Un servidor proxy es un equipo que actúa como intermediario entre el explorador web e internet. Éste ayuda a mejorar el rendimiento de los usuarios en internet, ya que almacena una copia de las páginas web más utilizadas en la memoria caché del servidor. Además el servidor proxy ayuda a mejorar la seguridad en la red, debido a que filtra ciertos contenidos web y software con base en reglas configuradas en el archivo del squid.

Algunas ventajas de los servidores proxys configurados en la red son:

- Se tiene un mejor control dentro de la red, ya que el servidor hace todo el trabajo y se puede limitar y restringir los derechos de los usuarios, así como dar permisos solo al mismo servidor.
- Hay un mejor rendimiento en la velocidad de la red, ya que si varios usuarios van a solicitar el mismo recurso de internet, el servidor guarda la información en su memoria caché con el fin de no volver a solicitar la información a red externa.
- El proxy puede hacer tareas de filtrado y negar algunas peticiones, si éste detecta que están prohibidas.

Hay algunas organizaciones que implementan proxies transparentes en la red, con el fin de interceptar y desviar las conexiones hacia el servidor proxy sin necesidad de configurar nada en el cliente.

Los pasos para configurar un servidor proxy son los siguientes:

1. Instalar y configurar el servicio de squid.
2. Configurar el servicio de ip-tables.
3. Correr los scripts y hacer pruebas de funcionamiento.
4. Configurar el navegador web para que todas las peticiones se realicen por medio del servidor proxy.

Los puntos importantes para poner atención al momento de configurar el servidor proxy son:

- El puerto por el que escuchará el servidor proxy será el 80 y todas las peticiones que lleguen a este puerto serán re-enviadas al puerto 3128 para la autenticación de cada usuario.
- Es necesario que las listas de acceso se configuren para que todas las peticiones que lleguen al puerto 80 sean reenviadas al puerto 3128.
- Será necesario configurar en cada uno de los navegadores web el puerto por el cual tendrán que hacer peticiones al servidor proxy y autentificarse para acceder a internet.

Para configurar el proxy en el navegador web se realizan los siguientes pasos.

1. En el navegador en el que se vaya a configurar, es necesario ir a herramientas y después al menú de opciones generales o configuración general (Figura 6.9).

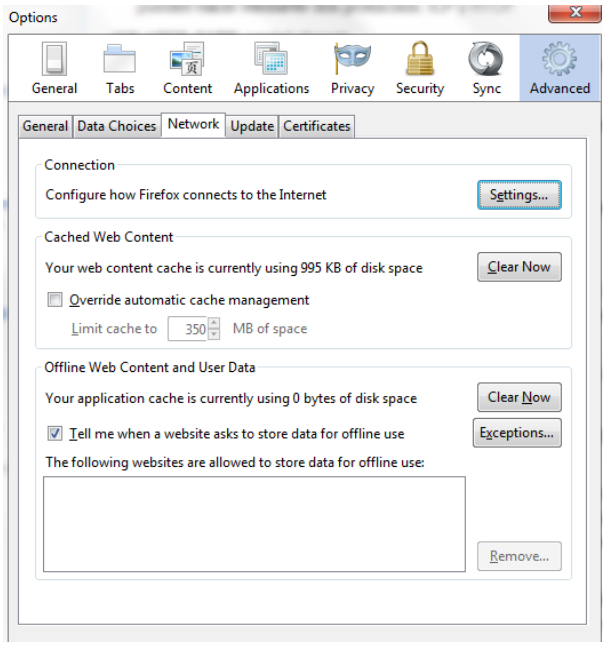


Figura 6.9 Menú de Opciones de navegador web mozilla Firefox.

2. Ir a configuración de red (Figura 6.10).

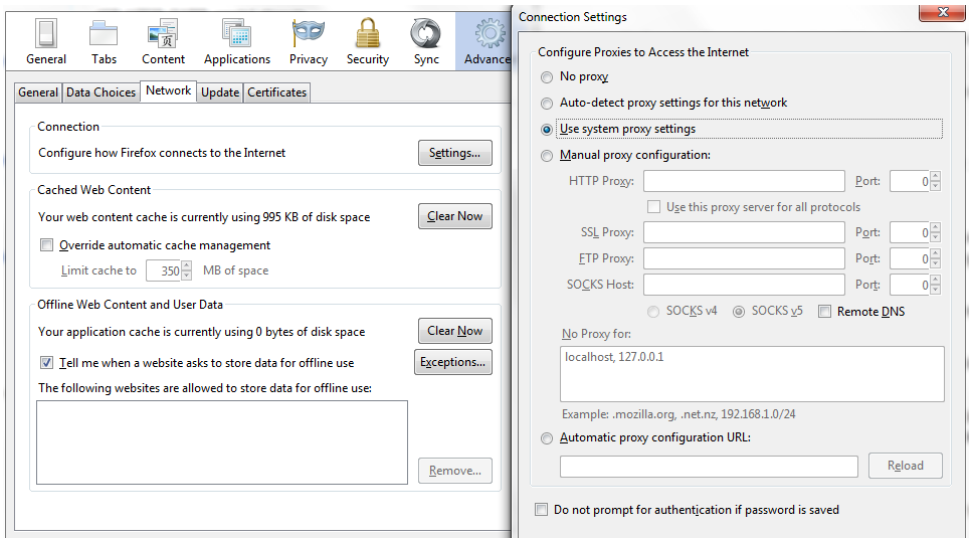


Figura 6.10 Configuración de conexiones de red.

3. Llenar la información de la dirección IP y el puerto del servidor proxy (Figura 6.11).

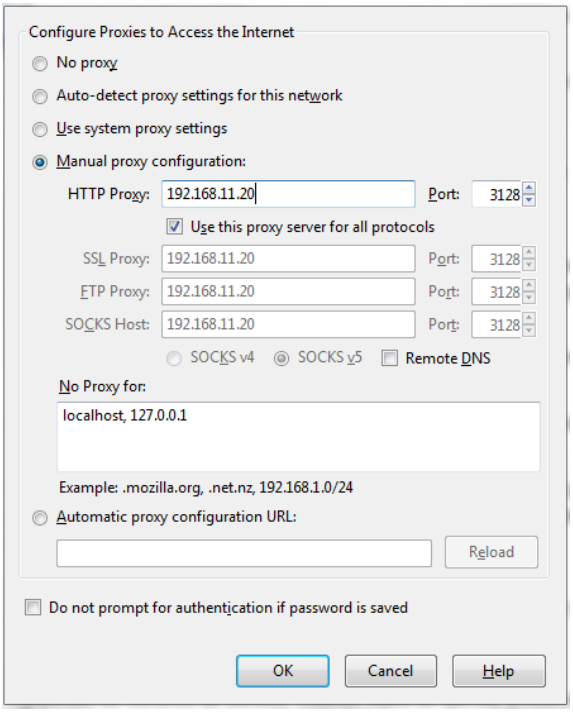


Figura 6.11 Configuración del servidor de proxy.

Una vez que se ha configurado el servidor web en la DMZ y todos los usuarios de red interna han configurado los navegadores web, habrá mayor control en la información que entra y sale, desde y hacia internet de todos los usuarios de red interna.

La configuración detallada de los servidores proxies puede consultarse en el Anexo B.

E) Balanceadores de carga

Los balanceadores de carga de la red tendrán como función mantener un equilibrio entre los servidores y las peticiones que lleguen a éstos dependiendo del pool de balanceo al que se dirijan. Un punto importante es que los balanceadores se deben configurar en pares o parejas y cada par tendrá la misma configuración, incluyendo la dirección IP a la que llegarán las peticiones provenientes del firewall. Esto se hace con el fin de tener un balanceador en modo activo y el otro en pasivo.

Si por alguna razón el balanceador que está en modo activo deja de funcionar, automáticamente el balanceador que esté en modo pasivo entrará como respaldo de manera invisible y las peticiones seguirán siendo enviadas a los servidores.

Para iniciar con la configuración hay que recordar que se debe tener en cuenta el método que se utilizará para el balanceo de los servidores que se encuentren configurados dentro de un mismo pool.

Es importante tomar en cuenta los siguientes pasos para una configuración básica de los balanceadores dentro de la red.

Paso 1. Configurar las direcciones IP de administración, red interna y red externa del equipo.

En el capítulo 4 se comentó que un balanceador debe ser configurado con 3 direcciones IP diferentes para que pueda trabajar correctamente en la red, todas ellas deberían estar en tres segmentos de red diferentes para que la seguridad del equipo no estuviera comprometida.

Nota: Es recomendable que la configuración de los equipos se realice por medio del método de consola, aunque es importante dejar claro que los balanceadores cuentan con la opción de configuración de manera gráfica por medio del protocolo https.

Paso 2. Configurar los pools de balanceo donde se agregarán los servidores basándose en el servicio que ofrece cada servidor.

Dentro del equipo existe una sentencia que se puede tomar como base para configurar el pool de balanceo en el equipo (Figura 6.12).

```
b pool <nombre_pool> {lb_method <metodo_balanceo> member
<ip_dispositivo:puerto>… <ip_dispositivo:puerto>…<ip_dispositivo:puerto>… }
```

Figura 6.12 Sintaxis del comando para la creación de un pool de balanceo.

Paso 3. Configurar el método de balanceo que se utilizará en los equipos.

En la figura 6.12 se muestra la sintaxis del comando que se utiliza para configurar un pool de balanceo dentro de la configuración del equipo balanceador*, lb_method* es utilizado para determinar el método de balanceo que se utilizará dentro de un pool específico. En el capítulo 4 se habló acerca de todos los métodos de balanceo que se pueden utilizar dentro de un balanceador de carga y las características de cada uno de ellos.

En este caso, se utilizarán dos métodos diferentes dependiendo de los equipos que estén balanceando los F5. Se utilizará el método de least connections para el pool de balanceo de los proxies y los métodos de fastest y least connections para el pool de los servidores de la DMZ.

Paso 4. Configurar el equipo como activo o standby para la opción de failover.

Para configurar los balanceadores con redundancia hay que tener en cuenta la dirección IP del mismo equipo *self_ip*, la dirección IP del segundo balanceador que se configurará como backup *peer* y una tercera que se determinará como dirección *float_ip.* Esta tercera dirección IP es la que recibirá las peticiones que lleguen del firewall *FwSrvs* y las enviará al balanceador que se encuentre como activo (Tabla 6.1 y 6.2).

BIGSRVA		BIGSRVB	
External VLAN			
Self	192.168.3.10	**Self**	192.168.3.11
Float	192.168.3.3	**Float**	192.168.3.3
Peer	192.168.3.11	**Peer**	192.168.3.10

Tabla 6.1 Configuración de red externa de los balanceadores de la DMZ.

BIGSRVA		BIGSRVB	
Internal VLAN			
Self	192.168.10.110	**Self**	192.168.10.111
Float	192.168.10.3	**Float**	192.168.10.3
Peer	192.168.3.111	**Peer**	192.168.3.110

Tabla 6.2 Configuración de red interna de los balanceadores de la DMZ.

Una vez determinadas las direcciones IP, los balanceadores trabajan de la siguiente manera (Figura 6.62).

Cada uno de los equipos cuenta con su propia dirección IP, con la única particularidad de que la dirección *float_ip* será configurada en ambos equipos para que puedan compartirla dentro del segmento de red a la que esté conectada. El equipo que se configure como activo estará balanceando a los servidores que estarán configurados dentro del pool de balanceo, como se puede observar en la figura 6.13.

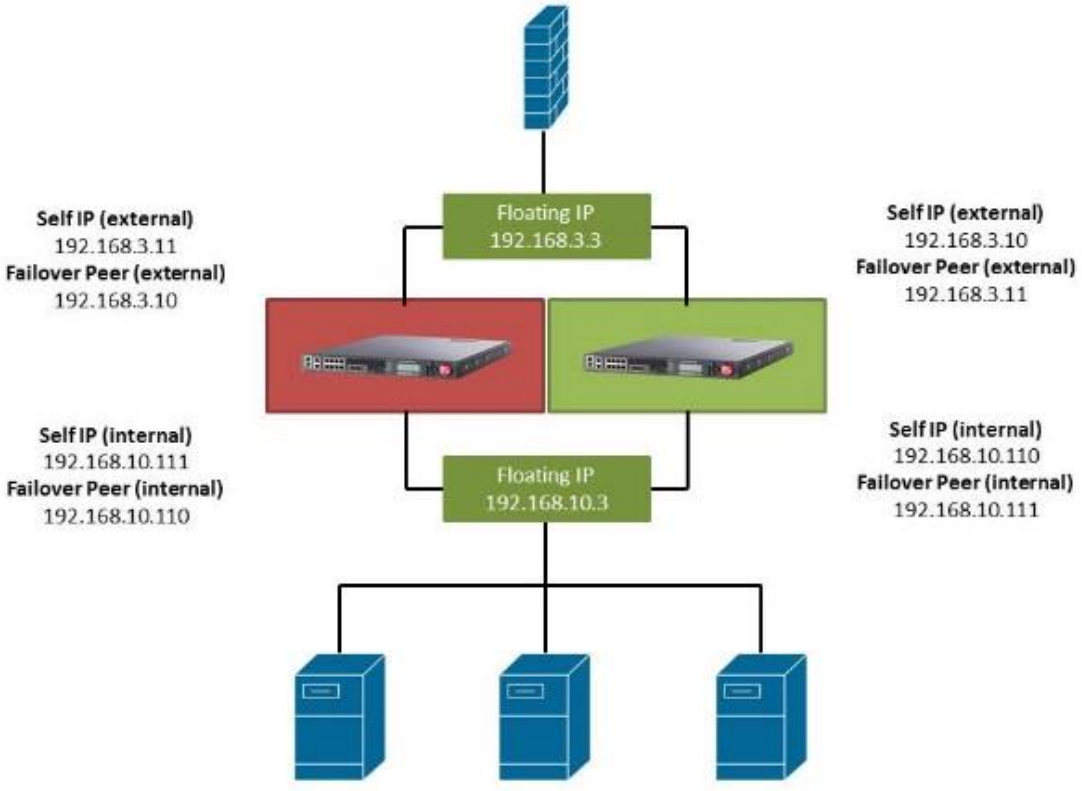


Figura 6.13 Diagrama de la configuración de direcciones IP en los balanceadores de la DMZ.

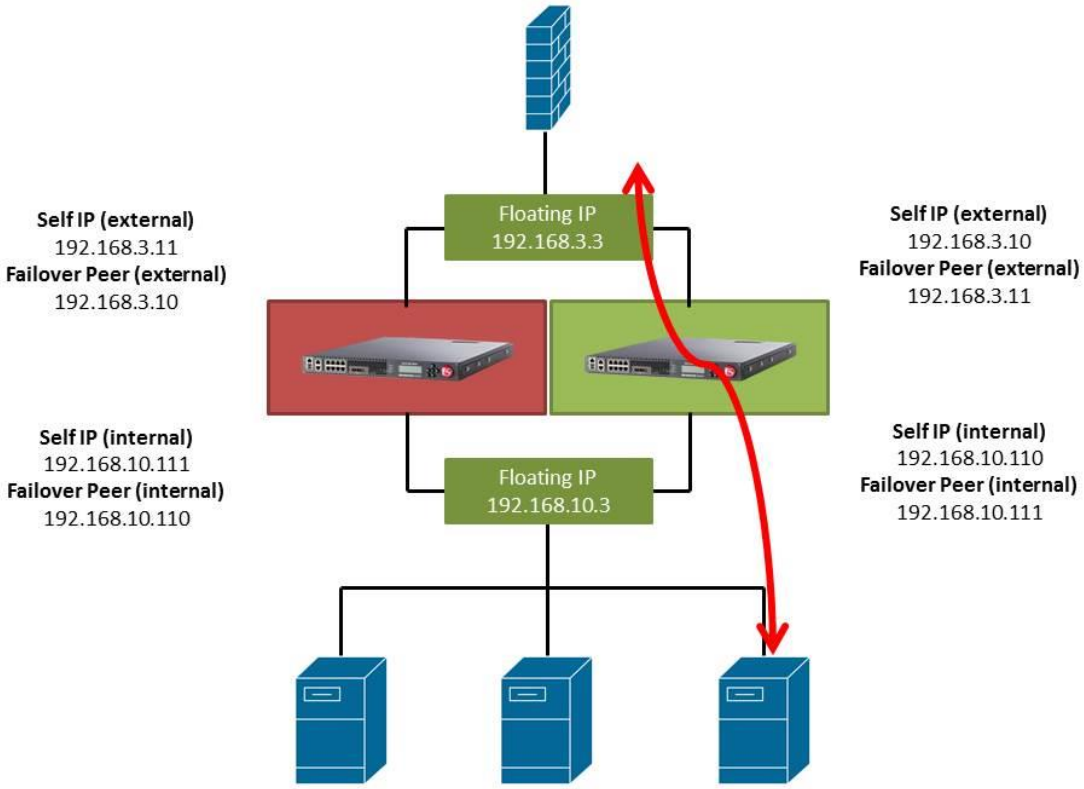


Figura 6.14 Diagrama del flujo que siguen los paquetes de datos dentro de la configuración realizada en los balanceadores de la DMZ.

Finalmente el equipo que se encuentre en modo activo cambia a *standby* por medio del *failover*. Esto puede suceder solo en ocasiones donde la energía del balanceador de carga se corte por alguna razón externa, que se le vaya a dar algún tipo de mantenimiento al equipo o se haya des configurado por error humano., en todos estos casos el equipo se tendrá que poner en modo activo al BIGSRVB. La configuración será la misma por lo que las peticiones no se verán afectadas al momento de que el cambio de *failover* se haya realizado (Figura 6.15).

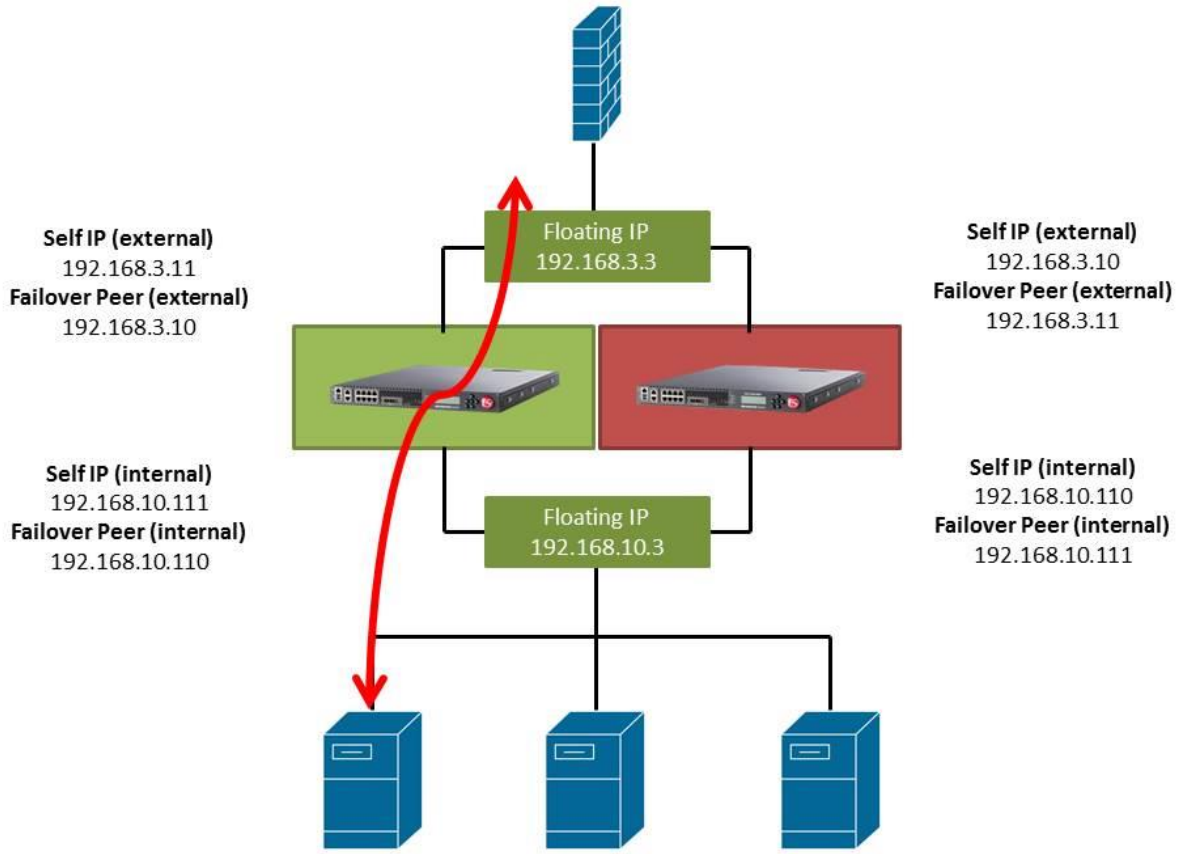


Figura 6.15 Diagrama de flujo que siguen los paquetes de datos cuando hay un failover en los balanceadores de carga.

Es importante aclarar que para que el *failover* se haga de manera transparente es necesario que los dos equipos se estén sincronizando de manera continua entre ambos y éstos siempre tengan la misma configuración, además de que el archivo de configuración se haya ingresado la dirección IP del *peer* con el que se estará sincronizando de forma automática.

La configuración detalla se puede observar en el Anexo B.

F) Firewalls

Los firewalls de la red son equipos Cisco los cuales fueron seleccionados para utilizarse dentro de la red, en este caso se comienza por la configuración básica del equipo como nombres, rutas y protocolos de enrutamiento, para seguir con las listas de acceso que son las que harán que la DMZ funcione de manera correcta permitiendo o bloqueando los paquetes según la configuración que a los administradores de la red les sea más conveniente.

Paso 1. Configuración básica del firewall.

La configuración básica del equipo consta de configurar los siguientes puntos:

- Nombre
- Líneas de consola y conexión remota.
- Contraseñas de usuario maestro y de líneas de conexión.

Una vez realizado este paso se podrá continuar con el proceso de configuración.

Paso 2. Configuración de las interfaces de red del firewall.

El siguiente paso es darle un nombre a las interfaces, para que se puedan identificar de manera más sencilla para configurar, administrar o modificar la información que se haya configurado en cada una de las interfaces (Figura 6.90).

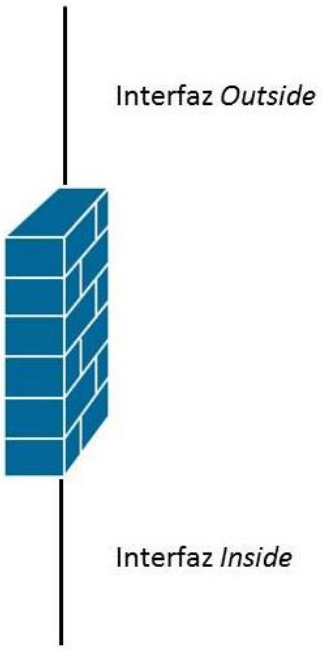


Figura 6.16 Diagrama del firewall indicando el nombre de las interfaces.

La configuración del nombre se realiza a través del comando **nameif**, el firewall tiene dos interfaces ya configuradas con nombre por defecto *inside* y *outside* y también están configuradas con un nivel de seguridad por defecto. La interfaz *outside* toma los valores por defecto de la interfaz Ethernet0 y un nivel de seguridad de 0 (cero). En cambio la interfaz *inside* toma los valores por defecto de la interfaz Ethernet1 y un valor de seguridad de 100.

También hay que configurar la interfaz para que trabaje a una velocidad de 100 Mbps y en modo full dúplex.

Además de configurar ambas interfaces con modo full duplex, el comando anterior ayuda a activar la interfaz sin la necesidad de utilizar el comando *no shutdown.*

El siguiente paso es configurar la dirección IP de las interfaces que estarán siendo utilizadas en el firewall. En comparación con la configuración de un router o switch, no es necesario entrar a la interfaz para asignar la dirección IP con la que estará trabajando.

La sintaxis que se utiliza es la siguiente **ip address** *interface_name ip_address subnet_mask.*

Después hay que configurar el NAT en el firewall para que no permita que los usuarios que envían solicitudes a los servidores de la DMZ puedan ver la dirección IP real del dispositivo. Para configurar una regla de NAT existen dos comandos para que se pueda configurar el servicio, ya sea de forma estática o dinámica.

La primera forma de configurar un método NAT es el modo estático, donde podemos ver la sintaxis de la regla a continuación.

static (*internal_if_name, external_if_name*) *global_ip local_ip* **netmask** *mask*

En donde los comandos están definidos de la siguiente manera:

- *internal_if_name* es el nombre de la interfaz con mayor nivel de seguridad donde el tráfico desea ir.
- *external_if_name* es el nombre de la interfaz con menor nivel de seguridad de donde el tráfico proviene.
- *global_ip* es la dirección IP registrada que es accesible desde red externa.
- *local_ip* es la dirección IP no registrada que es el destino final.
- *mask* es la máscara de subred o la máscara de host.

La segunda forma para configurar un NAT es a través de dos sintaxis que deben ser configuradas en paralelo para que estas funcionen correctamente a través del firewall.

La primera sintaxis es:

global (*external_if_name*) *ID global_ip/global_ip_range* **netmask** *mask*

Donde los comandos tienen el siguiente significado:

- *external_if_name* es el nombre de la interfaz de red externa.
- *ID* es un identificador con el que se discriminará al pool asignado de los demás que se configuren en el equipo.
- *global_ip/global_ip_range* es la dirección IP o el rango de direcciones que usarán todas las direcciones provenientes de un nivel de seguridad mayor con el mismo ID declarado en la sentencia.
- *mask* es la máscara de subred o la máscara de host.

La segunda sintaxis es:

nat (*inside_if_name*) *ID inside_ip/inside_ip_range mask 0 0*

Donde los comandos tienen el siguiente significado:

- *inside_if_name* es el nombre de la interfaz de red interna.
- *ID* es un identificador con el que se discriminará al pool asignado de los demás que se configuren en el equipo.
- *inside_ip/inside_ip_range* es el rango de direcciones IP que provengan de red interna.
- *mask* es la máscara de subred o la máscara de host.
- *0 0* se refieren al máximo número de conexiones activas y el límite de conexiones embriónicas[7]; esto quiere decir todas aquellas conexiones que aún no están completas. Los valores de 0 significan ilimitadas.

Ambas sintaxis ***global*** y ***nat*** van de la mano; deben usarse en conjunto cuando se va de una interfaz de seguridad alta a una interfaz de seguridad baja, como es el caso de todas aquellas conexiones que van de red interna a red externa.

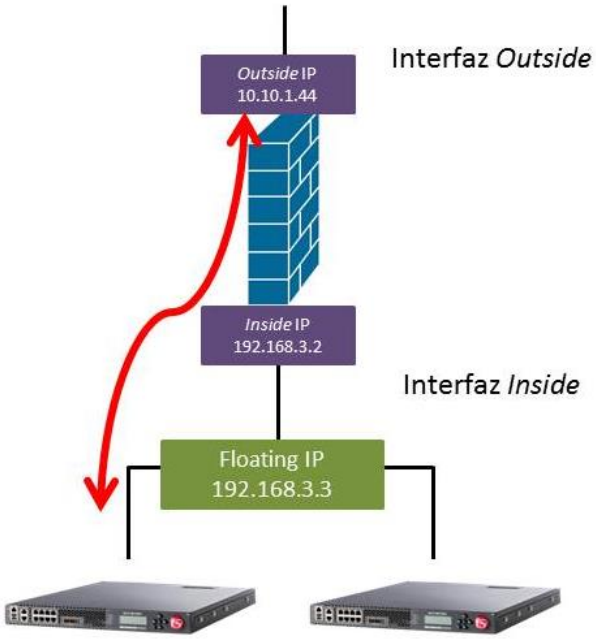


Figura 6.17 Diagrama del flujo que siguen los paquetes de datos a través del firewall.

Paso 3. Configuración del protocolo de enrutamiento.

[7] Toda conexión es embriónica hasta el momento que se completa.

Para cada red conectada a un router es necesario configurar un protocolo de enrutamiento con el fin de que el equipo conozca todas las redes que están directamente al dispositivo. Para los firewalls se determina que se configure el protocolo de enrutamiento de OSPF.

Debido a que el firewall cuenta con el mismo sistema operativo que los routers de Cisco se puede configurar una ruta por defecto que sea la única ruta que se configuré en el equipo, ésta ayudará para todos aquellos paquetes que el firewall no sepa porque interfaz enviar, el equipo los enviará por la ruta por defecto.

Paso 4. Configuración de las listas de acceso.

Finalmente se configuran las listas de acceso que son todas aquellas sentencias que permitirán decir que paquetes pasarán a través del firewall y que otros serán descartados por el equipo cuando sean comparados con las listas de acceso.

Es necesario conocer las políticas que se configurarán para cada uno de los firewalls que trabajarán en la red. En el caso específico del firewall FwSrvs, tendrá solo dos interfaces activas, *inside* y *outside;* con los servidores conectados en la interfaz *outside.* Las políticas de seguridad que se utilizarán y configurarán dentro del equipo son las siguientes:

- Toda solicitud de un equipo que este dentro de la DMZ puede acceder al servidor de DNS.
- Toda solicitud de un equipo de red externa puede acceder al servidor de WWW.
- Cualquier petición telnet está prohibida a los servidores de la interfaz *inside.*
- Cualquier petición ping está prohibida a los servidores de la interfaz *inside.*

Es necesario entender la sintaxis de las listas de acceso, ya que cabe la posibilidad de que se bloquee todo el tráfico hacia los servidores. En este caso en específico la sintaxis de la sentencia utilizada es la siguiente:

access-list *access_list_name* [permit|deny] tcp *source destination*

Los comandos representan lo siguiente:

- access-list es el comando con el que se ingresa la lista de acceso al equipo.
- *access_list_name,* es el nombre que se le asignará a la lista de acceso con el fin de identificarla de manera más rápida y sencilla entre todas las listas de acceso que se configuran dentro del equipo.
- [permit|deny] es la acción que realiza el firewall una vez que identifique la lista de acceso que se aplica al paquete entrante.
- tcp es el tipo de protocolo que utiliza la lista de acceso.
- *source* es la dirección IP origen del paquete.
- *destination* es la dirección IP destino del paquete.

Finalmente con los pasos anteriores se definió la configuración básica para todos los firewalls de la red.

G) ASA CISCO

El haber utilizado la mayoría de equipos cisco para la implementación de la red planteada al inicio del capítulo, ayuda a que muchos problemas de compatibilidad sean menores. Esto es importante, ya que la configuración de los equipos es similar en la mayoría de ellos.

Se tomará como base la configuración de los equipos Cisco que existen en la red para basarse en la configuración básica del equipo ASA. La configuración se realizará con base en la topología física mostrada con anterioridad.

La sintaxis de las sentencias son las siguientes:

- *router ospf <process_id>*
- *network <ip_address> <mask_area> <area_id>*

Donde:

- *process_id* es un identificador interno para el proceso de enrutamiento y puede ser un valor positivo o negativo.
- *ip_address* es el segmento de red que se encuentra conectado directamente conectado al equipo.
- *mask_area* es la máscara de red del segmento de red.
- *area_id* es un identificador lógico de enlaces OSPF que engloba redes y conexiones.

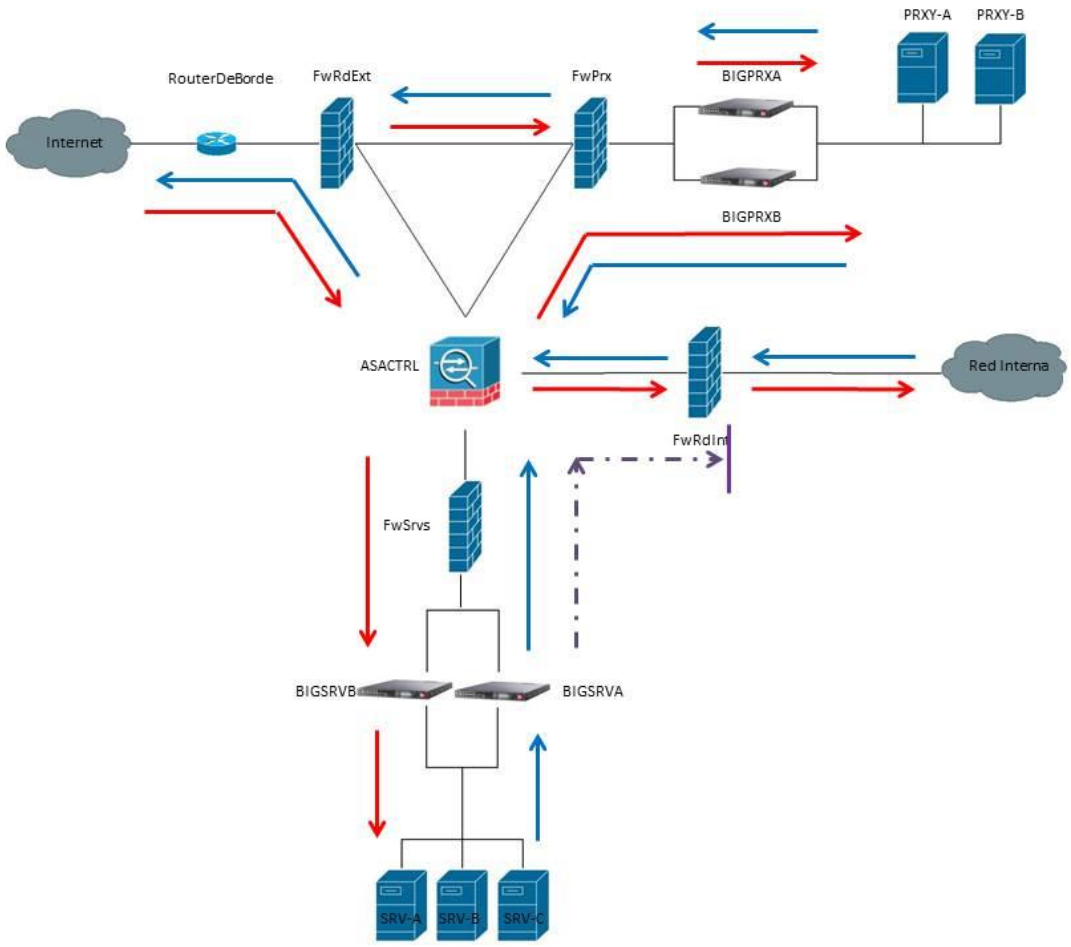


Figura 6.18 Diagrama del flujo de datos configurado en la red.

Los pasos para la configuración de los equipos son los siguientes:

Paso 1. Configuración básica del equipo (nombre del equipo, direcciones IP de las interfaces que estarán activas, nombre de las interfaces.

Paso 2. Configurar el Protocolo de enrutamiento del equipo.

Paso 3. Configurar las listas de acceso del equipo.

La configuración detallada se muestra en el Anexo B.

H) ROUTER de GW

El último equipo a configurar será el router de borde, que en realidad solamente es una conexión punto a punto con el ISP que da salida a internet. Hay que tener en cuenta que solamente se debe configurar

el nombre, activar las interfaces y las rutas, para que puedan entrar todos los paquetes que se dirigirán hacia los servidores que están dentro de la DMZ.

La configuración del router se realiza con la ayuda de los siguientes pasos:

Paso 1. Configuración básica del router.

Paso 2. Configuración de las interfaces del router.

La configuración de las interfaces del equipo es muy sencillo, por una parte el ISP ha indicado cual será la dirección IP que asigno al router de borde; por tal motivo solo es cuestión de configurar la interfaz serial con la dirección IP dada.

Por otra parte la otra interfaz que debe ser configurada es la que está directamente conectada al firewall FwRdExt, el primer firewall que estará filtrando todos los paquetes que lleguen de red externa. De esta manera la configuración será muy rápida en este equipo.

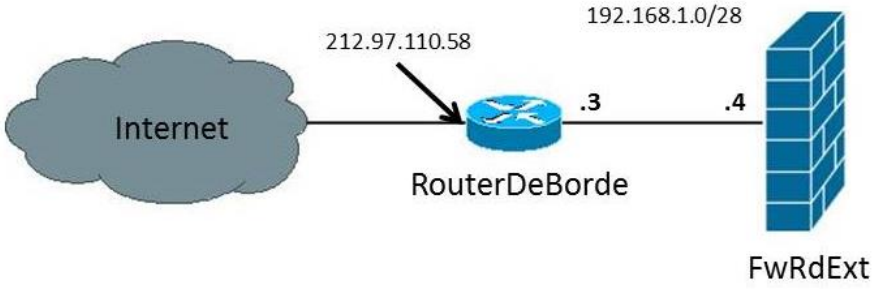


Figura 6.19 Diagrama de conexión del router RouterDeBorde.

Paso 3. Configuración del protocolo de enrutamiento.

Hay que configurar el nombre, contraseñas e interfaces de conexión, tal como se ha hecho en todos los equipos Cisco que conforman el diseño de la red.

Paso 2. Configuración de las interfaces del router.

La configuración de las interfaces del equipo es muy sencillo, por una parte el ISP ha indicado cual será la dirección IP que asigno al router de borde; por tal motivo solo es cuestión de configurar la interfaz serial con la dirección IP dada.

Por otra parte la otra interfaz que debe ser configurada es la que está directamente conectada al firewall FwRdExt, el primer firewall que estará filtrando todos los paquetes que lleguen de red externa. De esta manera la configuración será muy rápida en este equipo.

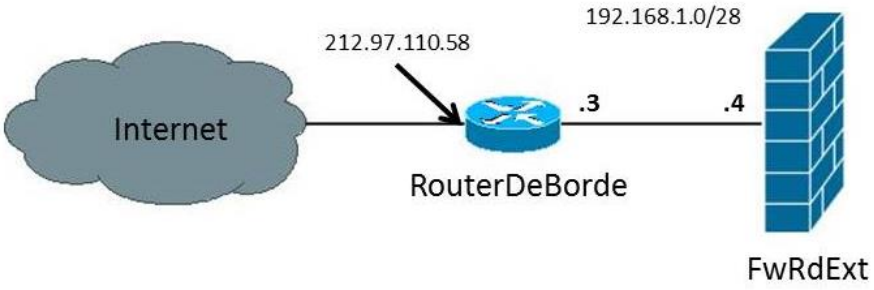


Figura 6.19 Diagrama de conexión del router RouterDeBorde.

Paso 3. Configuración del protocolo de enrutamiento.

El protocolo que se configura en el equipo es OSPF, con el fin de sea más sencillo que el equipo identifiqué todos los segmentos de red que se encuentran conectados directamente a éste.

Además de la configuración del protocolo OSPF, es necesario configurar la ruta por defecto con el fin de que el equipo tenga una ruta por la que envíe todos los paquetes donde no identifique el segmento de red al cual se dirige el paquete de datos.

Con esto se finaliza la configuración de todos los equipos que se encuentran dentro de la red descrita al principio del capítulo. Tomándose como hipótesis que este trabajo de configuración está configurado en cada equipo la conexión debe ser correcta entre todos los dispositivos y los paquetes que lleguen de red interna saldrán a internet a través de los proxies. Y también que todas las peticiones que lleguen de red externa podrán acceder a los servidores web y de bases de datos que se encuentran en la DMZ.

La configuración detallada del router se demuestra en el Anexo B.

Conclusiones

Conclusiones

Hoy en día las redes de datos tienen una enorme importancia en el día a día, a tal grado que las empresas invierten grandes cantidades de dinero e incrementan el porcentaje de inversión en este rubro año con año. Esto ha hecho que la industria de TI haya crecido a pasos agigantados en la última década.

Debido a lo anterior los requerimientos y las necesidades de todos los clientes, han hecho que cualquier red de datos cumpla con los objetivos teóricos básicos que a continuación se muestran:

- Disponibilidad
- Confiabilidad
- Eficiencia
- Seguridad

Durante cualquier curso de redes de datos se llegan a conocer los conceptos básicos de los componentes de las redes de datos. No es sino hasta que los estudiantes tienen experiencia práctica respecto a una red de datos que llegan a conocer la mayoría de los elementos que deben ser utilizados para que la red cumpla con los cuatro objetivos antes mencionados; así como la configuración de cada uno de los equipos que funcionan dentro de ella.

Con base en esto, se planteó un proyecto de tesis que propusiera de manera teórica todos los elementos que no son explicados dentro de un curso de redes de datos con la profundidad que ayude al alumno al momento de salir al campo laboral.

Por tal motivo esta tesis tiene como objetivo proponer el diseño y ser la base teórica para la instalación de una DMZ con los servicios básicos que se pueden encontrar en una red de esta naturaleza.

A lo largo de la realización de este proyecto se encontraron diferentes puntos que son importantes remarcar y que puedan servir como temas de tesis para futuros ingenieros que deseen tomarlos como temas de titulación:

- Se encontró que no existe una plataforma web o programa que ayude a los ingenieros en computación (que se están formando en el campo de las redes) a realizar una simulación de una red de datos y que incluya diferentes elementos que la conforman.

- Mucha de la documentación de configuración, administración e interconexión entre equipos de diferentes fabricantes no se encuentra estructurada y la mayoría está en blogs de expertos que ponen casos prácticos y muy poco explicados.

- No existen programas de diseño de redes que sean funcionales y fáciles de usar para que el ingeniero en computación invierta mayor tiempo en el diseño estructural de la red de datos que está por implementar y no en conocer y entender el correcto uso del programa de diseño. Se encontraron distintos programas que no son intuitivos para el usuario final.

- No existe ningún portal web que logré simular máquinas virtuales para la configuración de servidores basados en Linux.

Tomando en cuenta los cuatro puntos anteriores se concluye que aún hacen falta muchas herramientas para los ingenieros en computación que se especializan en redes de datos, ya que no fue posible implementar todo el marco teórico detallado en este trabajo de tesis por las siguientes razones:

a) No se pudo implementar de forma física el marco teórico por el enorme costo que tienen los equipos que se detallaron en el trabajo de tesis.

   Durante la búsqueda para realizar la cotización y adquisición de equipos necesarios para la implementación de la DMZ se pudo conocer el elevado costo de los equipos, tal como se muestra en el anexo. Esto complicó demasiado la realización del trabajo, ya que queda fuera del presupuesto de un trabajo de tesis como se planteó desde el principio.
   Una alternativa fue solicitar apoyo de algún proveedor para la implementación de la red, lo cual también fue un rotundo fracaso porque la inversión que un proveedor de redes de datos hace para este rubro es tan grande que pensar en quitar los equipos de una función primordial para una red lo hace imposible. El proveedor desea que su inversión trabaje para que obtenga ganancias en el menor tiempo posible (ROI).

b) No se pudo concretar la configuración física del balanceador de carga debido a la falta de una licencia comercial por parte del fabricante que tiene un costo elevado.

   Durante la realización del capítulo 4 “Balanceadores”, se contaba con la facilidad de simular un equipo de balanceo a través de una máquina virtual. El problema se dio específicamente al momento de configurar los últimos parámetros para la configuración del equipo de manera virtual. Era necesario contar con un equipo F5 y una licencia por parte de la compañía que finalizaba el registro del software y hardware que serían utilizados en cuestión.
   Al encontrarse con esta problemática tuvo que recurrirse a solicitar prestado un balanceador F5 a través de una compañía especializada para finalizar el proceso de instalación y conocer los protocolos de balanceo que pueden ser utilizados en un equipo de la gama F5.

   De cierta manera tuvo sus puntos malos y además sus puntos buenos, los malos pueden ser listados de la siguiente manera: i) No se pudo llevar a cabo la simulación de la red como se tenía previsto al principio del trabajo de tesis; ii) No se configuró un sistema de balanceo de manera física para dos computadoras que tenían como objetivo simular los servidores que se encontraban en la DMZ; iii) No se contaba con el presupuesto para realizar la adquisición de un equipo con las características de un equipo F5.

Por otra parte, los puntos a favor fueron los siguientes: i) se pudo tener acceso a un equipo balanceador F5 para concluir el proceso de configuración básica, que fue necesario para este trabajo de tesis; ii) Se contó con la guía de un experto en la materia para aclarar algunas dudas y comentarios que surgieron durante la realización del capítulo 4; y iii) Se finalizaron los pasos necesarios para configurar un equipo F5 y crear un pool de balanceo para los diferentes servicios que pudieran estar contenidos dentro de una DMZ.

De esta manera, al final pudo completarse el objetivo del capítulo 4, que consistía en la realización de una guía básica para la configuración y balanceo de los servicios contenidos en una DMZ.

c) Se encontró que no hay una herramienta que ayude a diseñar, conectar e implementar una red de datos con equipos de distintos fabricantes; inclusive no es posible tener dispositivos del mismo fabricante.

En este punto en particular se encontró que existen diferentes herramientas para el diseño físico de una red de datos, durante el proceso de realización de la tesis se buscó software de diferentes fuentes o herramientas que ayudarán para la realización de diagramas, imágenes y un diseño físico como tal; encontrando las siguientes herramientas como las que a continuación se listan:

- Dia
- PacketTracer
- Weresc
- Creatly, entre otros.

El problema que se observó es que no hay una plataforma que incluya tecnologías de diferentes fabricantes, así como diferentes elementos que se utilizan dentro de las redes de datos.

De manera general se utilizaba el programa de Cisco, Packet Tracer para tratar de simular la conexión de un router para una nube de internet, pero aún es un programa que tiene mucho por mejorar, ya que no incluye mucha de la tecnología cisco que pudiera utilizarse dentro de una red de datos como pueden ser firewalls, balanceadores y switches de diferentes gamas.

En el caso de Dia, se utilizó para la generación de un diagrama, aunque no es posible agregar texto para delimitar o diferenciar un segmento de red por lo que en ese caso es muy básico para la utilización en el diseño de redes de datos.

Para el resto del software, fue casi imposible utilizarlos, ya que muchos de ellos solo sirven para el diseño de manera gráfica y otros más no pueden ser utilizados debido a una muy mala interfaz gráfica para el usuario.

Durante el proceso de ejecución del proyecto se contó con grandes dificultades para la implementación de la base teórica hecha en este proyecto de tesis y con base en la hipótesis planteada al principio se llegó a la conclusión de que este trabajo sirve como base teórica para la elaboración de una red de datos y otros puntos extra que pueden ser retomados para ser desarrollado, actualizados e implementados posteriormente, tal y como se planteó al inicio de las conclusiones.

Anexo A

Anexo A

Dentro de este anexo se describen los precios encontrados en diferentes páginas de internet de proveedores, así como de fabricantes para los diferentes equipos utilizados en la realización del proyecto de tesis.

Las características y precio del Balanceador de carga F5, puede observarse en la Tabla A.1. que está a continuación. En ella puede observarse que el precio de estos equipos es realmente alto debido a su especialidad en el mercado del hardware.

F5 BIG-IP LTM 8900 Load Balancer[8]			
		Precio	$89,995.00 USD
Procesador	2	Carga de Balanceo Global	Opcional
Software	32 bit	Factor de Forma	2 U
Puertos Gigabit	16	Rendimiento de Tráfico	12 Gbps
Puertos de Fibra	8/2	Conexión en capa 4	400,000
Almacenamiento	HDD	RPS en capa 7	1,2 Mil

Tabla A.1 Características de un balanceador de carga LTM 8900 marca F5.

Por otra parte, se puede observar que el precio de mercado para un router ha disminuido (Tabla A.2), debido a que la gran demanda de equipos como este ha hecho que el precio baje a un costo más competitivo.

CISCO 2901/K9 Integrated Services Router[9]			
		Precio	$12,461.00 MXN
Procesador	2	Características	Totalmente integrado de distribución de energía a los módulos de soporte 802.3af Power over Ethernet y cisco mejorada. Incrustado vpn cifrado acelerado por hardware para las comunicaciones seguras vpn de colaboración. Control integrado de amenazas utilizando Cisco IOS firewall, Cisco
Software	CISCO IOS		
Puertos Gigabit	2 (10/100/1000 puertos Ethernet)		
Puertos WAN	4		
Almacenamiento	HDD		

8 http://buyloadbalancers.com/f5biltm89lob.html

9 http://www.ebay.com/itm/New-Sealed-CISCO2901-K9-Cisco-2901-Router-Cisco-Router-2900-4-EHWIC-256MB-CF-/231099820601?pt=LH_DefaultDomain_15&hash=item35ce9f6a39

			IOS firewall basada en zonas, Cisco IOS IPS y Cisco IOS filtrado de contenido.

Tabla A.2 Características de un router CISCO 2901.

En el caso de los firewalls de ASA, el precio de un equipo como este puede variar dependiendo del modelo y las características (Tabla A.3), para este proyecto el precio en el que oscila el equipo necesario para que la red sea implementada es de $28.491.00 (veintiocho mil cuatrocientos noventa y un pesos), mientras que un firewall del modelo PIX tiene un precio que oscila entre los tres mil pesos (Tabla A.4).

CISCO ASA 5500 SSL VPN 25 Premium License[10]			
		Precio	$28,491.00 MXN
Rendimiento del Firewall	Hasta 1.2 Gbps	Interfaces	8 puertos e Gigabit Ethernet, 4 puertos de fibra SFP y un puerto de FAST-Ethernet
Rendimiento de VPN	Hasta 425 Gbps	Interfaces Virtuales	400
Sesiones Concurrentes	650,000	Escalabilidad	VPN clustering y balanceo de carga
Peers en IPsec VPN	5000	Alta Disponibilidad	Activo/Activo, Activo/Standby
Contextos de Seguridad	Hasta 100		

Tabla A.3 Características de un appliance CISCO 5500.

CISCO PIX 501 Firewall Security Appliance[11]			
		Precio	$3,253.24 MXN
Rendimiento del Firewall	Hasta 1.2 Gbps	Interfaces	8 puertos e Gigabit Ethernet, 4 puertos de fibra SFP y un puerto de FAST-Ethernet
Rendimiento de VPN	Hasta 425 Gbps	Interfaces Virtuales	400

10 http://www.ebay.com/itm/CISCO-ASA5500-SSL-25-ASA-5500-SSL-VPN-25-Premium-User-License-/321418137317?pt=US_Security_Cameras&hash=item4ad6038ee5

11 http://www.ebay.com/itm/NEW-CISCO-PIX-501-BUN-K9-FIREWALL-VPN-SECURITY-APPLIANCE-47-10539-01-/161263433256?pt=LH_DefaultDomain_0&hash=item258c0cb628

Sesiones Concurrentes	650,000	Escalabilidad	VPN clustering y balanceo de carga
Peers en IPsec VPN	5000	Alta Disponibilidad	Activo/Activo, Activo/Standby
Contextos de Seguridad	Hasta 100		

Tabla A.4 Características de un firewall PIX 501 de la marca CISCO.

Finalmente el precio de un servidor con las características necesarias para la implementación de la red tiene un costo aproximado de veintiséis mil pesos (Tabla A.5). En este caso el hardware no es tan caro en comparación con el resto de los equipos, lo que sube el precio de éstos, son las licencias del software para la administración del equipo.

IBM Server System x3630 Xeon E5620[12]			
		Precio	$26,486.28 MXN
Procesador	1 (2.4GHz)	Marca	IBM
Tipo de memoria	DDR3 SDRAM		
Memoria RAM	4 GB		
Modelo	System x3630 M3		

Tabla A.5 Características de un servidor x3630 marca IBM.

Con base en los montos mostrados dentro de este anexo se estima que el precio estimado para la compra de los todos los equipos necesarios para la implementación del proyecto es de **$4,866,136.36 MXN** haciendo la conversión a dólares la cantidad total es de $371,460.71 dólares**.**

Balanceador F5 - $89,995.00 (USD) x $13.00 (MXN) = $1,169,935.00 (x 4) = $4,679,740.00 MXN

Router - $12,461.00 MXN

Cisco Adaptive Security Appliance - $28,491.00 MXN

Cisco Firewall Pix - $3,253.24 (x 4) = $13,012.96 MXN

Servidor IBM - $26,486.28 (x 5) = $132,431.40 MXN

[12] http://www.ebay.com/itm/IBM-Server-System-x3630-M3-QC-Xeon-E5620-2-4GHz-4GB-NOB-/370863198062?pt=DE_Computing_Server&hash=item56592b4f6e

Anexo B

Anexo B – Configuraciones

1. Servidor de DNS

Paso 1. Instalar la paquetería de bind en el servidor.

Es necesario instalar la paquetería de bind (Figura B.1) que se necesita para dar soporte al servicio de DNS.

```
yum install bind bind-utils –y
yum update -y
```

Figura B.1 Sentencia para instalación de paquetería bind.

Paso 2. Configurar los archivos de named.conf y resolv.conf con los parámetros necesarios para activar el servicio.

El siguiente paso será crear o configurar cuatro archivos para iniciar el servicio de DNS:

- /etc/named.conf
- /etc/sysconfig/network-scripts/ifcfg-eth0
- /etc/resolv.conf
- /var/named/[domain].zone

Se inicia el servicio de DNS a través del comando mostrado en la figura B.2, para proceder a la configuración de los archivos comentados anteriormente.

```
#/etc/init.d/named start
```

Figura B.2 Comando para el arranque del servicio de DNS en un sistema Linux.

Se configura el archivo named.conf para que se configure y quede como se puede observar en la figura B.3.

```
# vi /var/named/chroot/etc/named.conf

acl devapp-DMZ{192.168.4.0/24; 127.0/8};

options {
        listen-on port 53 {192.168.1.10};
        directory "/var/named";
```

```
        dump-file "/var/named/data/cache_dumb.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query {devapp-DMZ;};
        recursion yes;

         forwarders { 212.97.110.58; };

        forward only;

        dnssec-enable no;
        dnssec-validation no;
        dnssec-lookaside auto;

         };

Logging {
        channel default debug{
        file "data/named.run";

        severity dynamic;

};

};

zone "devapp.lan" in {
        type master;
        file "devapp.lan.zone";
};



zone "4.168.192.in-addr.arpa" in {
        type master;
        file "192.168.4.zone";
};

zone "." in {
         type hint;
         file "named.ca";
};
```

Figura B.3 Archivo de configuración de DNS.

Una vez hechos los cambios se guardan y se verifica que la sintaxis de los comandos esté correcta a través del comando siguiente, como se puede observar en la figura B.4.

```
# named-checkconf /var/named/chroot/etc/named.conf
```

Figura B.4 Comando para revisar la sintaxis utilizada en el archivo de configuración named.

Si todo está correcto, el sistema no envía ningún error. El siguiente paso es dar los permisos necesarios para que se pueda leer el archivo named.conf (Figura B.5).

```
# chmod 644 /var/named/chroot/etc/named.conf
```

Figura B.5 Comando para otorgamiento de permisos dentro del archivo named.conf.

Posterior a lo ya realizado hay que configurar el archivo ifcfg-eth0 y colocar las direcciones de red que corresponden a la dirección IP del dispositivo, la dirección de Gateway y cambiar el valor al campo BOOTPROTO a *static* (Figura B.6).

```
DEVICE=”eth0”
NM_CONTROLLED=”yes”
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
IPADDR=192.168.1.10
PREFIX=24
GATEWAY=192.168.1.1
DNS1=192.168.1.10
```

Figura B.6 Configuración de la tarjeta de red del servidor de DNS.

Al guardar los cambios es necesario reiniciar la tarjeta de red para asegurarse de que los cambios fueron realizados de manera correcta y realizar el reinicio del servicio de red (Figura B.7).

```
# /etc/init.d/network restart
```

Figura B.7 Comando de reinicio del demonio de red.

Si todos los cambios son correctos enviará un mensaje de [OK].

Ahora hay que realizar los cambios al archivo de configuración resolv.conf, a través del comando empleado en la figura B.8.

```
# vi /etc/resolv.conf
---------------
GENERATED BY NETWORK MANAGER
nameserver 192.168.1.2
---------------
```

Figura B.8 Comando para la configuración del archivo resolv.conf

Ya que este servidor está dentro de una DMZ, es necesario configurar una lista de acceso que permita únicamente a los usuarios de red interna consultar las listas de resolución de nombres. Este paso se realiza por medio del comando mostrado en la figura B.9.

```
# iptables –A INPUT –s 192.168.1.0/24 –p udp –dport 53 –j ACCEPT
# /etc/init.d/iptables save
```

Figura B.9 Lista de acceso configurada dentro del archivo de configuración resolv.conf.

Finalmente se dan los permisos a los archivos devapp.zone y 192.168.1.zone y se agregan en la carpeta /var/named/… (Figura B.10)

```
//se hacen las modificaciones para agregar el archivo devapp.lan.zone a la
carpeta y se modifica en owner del archivo, así como los privilegios para que
se pueda modificar el archivo.
# touch devapp.lan.zone
# chown named:named devapp.lan.zone
# chmod 644 devapp.lan.zone

//se hace el mismo proceso para el archive 192.168.1.zone

# touch 192.168.4.zone
# chown named:named 192.168.4.zone
# chmod 644 192.168.4.zone
```

Figura B.10 Comandos para otorgar permisos para los grupos configurados por zona.

Como último paso para la configuración del servicio de DNS se configuran los archivos que se acaban de agregar a la carpeta /var/named/ (Figura B.11)

```
#vi devapp.lan.zone

//devapp.lan.zone

STTL 1D
devapp.lan. IN SOA devapp.lan. root.devapp.lan(
         201206170 ;serial
         2H         ;refresh slaves
         5M        ;retry
         1W        ;expire
         1M       ;negative TTL
)

@       IN      NS      devapp.lan.
@       IN      A       127.0.0.1
www     IN      A       192.168.1.10

:wq!
```

Figura B.11 Configuración del archivo devapp.lan.zone con los tiempos de actualización de las listas de resolución de nombres.

Se corrobora que el archivo no tenga ningún error de sintaxis (Figura B12).

```
# named-checkzone devapp.lan devapp.lan.zone
```

Figura B.12 Comando para revisar la sintaxis utilizada del archivo de configuración devapp.lan.zone.

Y como siguiente paso se procede a modificar el archivo 192.168.1.zone (Figura B.13).

```
# vi 192.168.1.zone

//192.168.1.zone

STTL 1D
devapp.lan. IN SOA devapp.lan. rootdevapp.lan(
         201206170 ;serial
         2H         ;refresh slaves
         5M        ;retry
         1W        ;expire
         1M       ;negative TTL
)

        IN      NS      devapp.lan.
www     IN      A       192.168.1.10
192.168.1.10     IN     PTR     root.devapp.lan.

:wq!
```

Figura B.13 Configuración del archivo 192.168.1.zone con los tiempos de actualización de las listas de resolución de nombres.

Nuevamente se corrobora que el archivo no tenga ningún error de sintaxis, por medio de la sentencia utilizada en la figura B.14.

```
# named-checkzone 4.168.192.in-addr.arpa 192.168.1.zone
```

Figura B.14 Comando para revisar la sintaxis utilizada del archivo de configuración 192.168.1.zone.

En ambos casos si todo está correcto el sistema envía un mensaje de OK.

Se reinicia el servicio de named para que los cambios realizados en la configuración tengan efecto (Figura B.15).

```
# /etc/init.d/named restart
```

Figura B.15 Comando de reinicio del daemon de DNS.

Paso 3. Realizar pruebas al servicio de DNS para confirmar que el servidor está funcionando.

Para finalizar la configuración se configura la dirección del servidor DNS en la configuración de red de un dispositivo que se encuentre en la misma subred y se realizan un par de pruebas para verificar que todo esté correcto.

Con el comando ping se revisa que haya conexión al servicio de DNS (Figura B.16).

```
# ping devapp.lan
```

Figura B.16 Comando ping para lograr revisar que el servicio de DNS esté funcionando.

Posterior a ello se hace una prueba con el comando *dig*[13] para revisar que el servicio de names esté resolviendo los nombres correctamente y guardando las direcciones IP que corresponden a la URL que se indique dentro del comando (Figura B.17).

```
# dig www.google.com
```

Figura B.17 Comando para la resolución del nombre de google.com.

Si ambas pruebas salen exitosas quiere decir que se ha configurado correctamente el servidor de DNS.

2. Servidor de Base de Datos

Configuración del servidor de base de datos.

Paso 1. Descargar e instalar el demonio de mysql en el servidor.

Hay que instalar la paquetería de la base de datos, en este caso se utiliza mysql para albergar la base de datos en el servidor (Figura B.18).

```
# yum install mysql-server mysql
```

Figura B.18 Comando para la instalación de la paquetería de base de datos.

Una vez que se introduzca el comando *yum* hay que esperar que toda la paquetería se instale para configurar la base de datos que albergará la información de cada uno de los usuarios.

[13] http://www.linux.com/learn/tutorials/442431-check-your-dns-records-with-dig

Seguido del punto anterior hay que iniciar el proceso de *mysql server.* Con los comandos de la figura B.19 se inicia el proceso de manera correcta.

```
# chkconfig mysql on
# /etc/init.d/mysqld start
```

Figura B.19 Comando para inicio del demonio de mysql.

El siguiente paso es configurar un password para el usuario *root* dentro del servicio de mysql (Figura B.20).

```
# myaqladmin –u root password PASSWORD
```

Figura B.20 Comando para configurar usuario y contraseña del servicio de mysql.

Finalmente hay que realizar una prueba de conexión al servidor de base de datos (Figura B.21).

```
# mysql –u root –p
```

Figura B.21 Comando para la configuración de usuario y contraseña del usuario del servicio de mysql.

Paso 2. Configurar el servicio de mysql.conf con los parámetros necesarios para la base de datos.

Es necesario configurar algunos campos importantes dependiendo de la base de datos que se vaya a utilizar como se observa en la figura B.22.

```
# vi /etc/mysql.conf


#En esta parte se configura el caché del query de mysql#


 query_cache_type             =1
 query_cache_limit            =1M
 query_cache_size             =32M



#También se puede configurar el tamaño del buffer del MyISAM
# MyISAM#

        key_buffer_siza        =24M
        myisam_recover         =FORCE, BACKUP

#Hay que configurar los logs para la detección de errores
#LOGGING#

        log_queries_not_using_indexes  =1
        slow_query_log                 =1
        slow_query_log_file            =/var/lib/mysql/mysql-slow-query.log
```

```
#También hay que establecer cachés y otros límites#

        tmp_table_size                  = 32M
        max heap table size             = 32M
        max_connections                 = 500
        thread_cache_size               = 50
        open_files_limit                = 65535
        table_definition_cache          = 4096
        table open cache                = 512

#se guardan los cambios y se cierra el archivo#

:wq!
```

Figura B.22 Archivo de configuración para el archivo *mysql.conf.*

Hecho esto se debe reiniciar el servicio de *mysql* (Figura B.23).

```
# /sbin/service mysqld restart
```

Figura B.23 Comando para reiniciar el demonio de mysql.

Paso 3. Crear la base de datos en donde se almacenarán los datos.

Una vez que se ha reiniciado el servicio de *mysql* hay que entrar como usuario *root* para poder crear la tabla. Para crear la tabla es necesario utilizar el comando CREATE DATABASE, para crear la tabla en la que se guardan los datos de todos los usuarios al crear un perfil (Figura B.24).

```
mysql> CREATE DATABASE user_profiles;

Query OK, 1 row affected (0.00 sec)
```

Figura B.24 Sentencia para la creación de una base de datos dentro de mysql.

Se verifica que la base de datos se haya creado correctamente por medio del comando *SHOW DATABASE* (Figura B.25).

```
mysql> SHOW DATABASES;

OUTPUT:

+---------------------+
| Database            |
+---------------------+
| information_schema  |
```

```
| mysql               |
| user_profiles       |
+---------------------+

3 rows in set (0.00 sec)
```

Figura B.25 Sentencia para mostrar las bases de datos creadas dentro de mysql.

Como se muestra en la figura anterior, a este punto la base de datos ya fue creada y ahora habrá que crear las tablas forman parte de la base de datos, mismas que contendrán cada uno de los campos de información que los usuarios nuevos llenarán para crear su perfil.

Paso 4. Crear las tablas con los elementos dentro de la base de datos.

Para poder crear la tabla es necesario determinar primero la cantidad de caracteres que forma cada campo. En este caso la tabla se configura de forma básico con el fin de no profundizar en el tema de bases de datos. Para mayor información al respecto se recomienda buscar textos especializados[14].

Se crea la tabla *user* con los siguientes campos:

- Nombre
- Apellido
- Teléfono Celular
- E-mail

Para ingresarlos al servidor de base de datos se modifican los nombres de los campos y se determina la longitud de cada uno, para que queden de la siguiente manera:

- FirstName, varchar(20)
- LastName, varchar(20)
- Phone int(15);
- Email, varchar(20)

Ahora el siguiente paso es introducir la sentencia para crear la tabla (Figura B.26).

```
mysql> use user_profiles;
mysql> create table user (FirstName char(20), LastName char(20), Phone
int(15), Email char(20));
mysql> show tables;

+-------------------------+
| Tables_in_user_profiles |
+-------------------------+
| user                    |
+-------------------------+
```

[14] Database Principles and Design, Colin Ritchie.

```
mysql> describe user;

+------------+----------+------+-----+---------+-------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+---------+-------+
| FirstName  | char(20) | YES  |     | NULL    |       |
| LastName   | char(20) | YES  |     | NULL    |       |
| Phone      | int(10)  | YES  |     | NULL    |       |
| Email      | char(20) | YES  |     | NULL    |       |
+------------+----------+------+-----+---------+-------+
4 rows in set (0.04 sec)
```

Figura B.26 Sentencia para la creación de tablas dentro de la base de datos y los campos que la conforman.

Finalmente se pueden ingresar los datos de los usuarios a la tabla *user* de la base de datos *user_profiles.*

3. Servidor Web

Configuración del servidor web.

Paso 1. Instalación de la paquetería de httpd en el servidor.

Para poder configurar el servidor web dentro de la DMZ es necesario instalar la paquetería de *httpd* a través del comando *yum*. Como ya se ha hecho anteriormente se entra a la consola de redhat con los privilegios de administrador y se introduce el comando como se muestra en la figura B.27.

```
# yum install httpd
```

Figura B.27 Comando para la instalación del servicio de http en un sistema Linux.

Paso 2. Configurar e iniciar el servicio de apache/httpd.

Una vez finalizado el proceso de instalación hay que iniciar el servicio de Apache/httpd. Para que esto sea posible se introduce el comando que reinicia el servicio (Figura B.28).

```
# chkconfig http on
# /etc/init.d/httpd start
```

Figura B.28 Comando para arrancar el *daemon* de http.

```
### Global Environment ##################################################
```

```
#
# The directives in this section affect the overall operation of Apache,
# such as the number of concurrent requests.

# run under this user/group id
Include /etc/apache2/uid.conf

# - how many server processes to start (server pool regulation)
# - usage of KeepAlive
Include /etc/apache2/server-tuning.conf

# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
ErrorLog /var/log/apache2/error_log

# generated from APACHE_MODULES in /etc/sysconfig/apache2
Include /etc/apache2/sysconfig.d/loadmodule.conf

# IP addresses / ports to listen on
Include /etc/apache2/listen.conf

# predefined logging formats
Include /etc/apache2/mod_log_config.conf

# generated from global settings in /etc/sysconfig/apache2
Include /etc/apache2/sysconfig.d/global.conf

# optional mod_status, mod_info
Include /etc/apache2/mod_status.conf
Include /etc/apache2/mod_info.conf

# optional cookie-based user tracking
# read the documentation before using it!!
Include /etc/apache2/mod_usertrack.conf

# configuration of server-generated directory listings
Include /etc/apache2/mod_autoindex-defaults.conf

# associate MIME types with filename extensions
TypesConfig /etc/apache2/mime.types
DefaultType text/plain
Include /etc/apache2/mod_mime-defaults.conf

# set up (customizable) error responses
Include /etc/apache2/errors.conf

# global (server-wide) SSL configuration, that is not specific to
# any virtual host
Include /etc/apache2/ssl-global.conf

# forbid access to the entire filesystem by default
<Directory />
```

```
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

# use .htaccess files for overriding,
AccessFileName .htaccess
# and never show them
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

# List of resources to look for when the client requests a directory
DirectoryIndex index.html index.html.var
```

Figura B.29 Archivo de configuración de httpd.

Hay que tener en cuenta que el archivo de configuración tiene la opción de adjuntar los archivos de la página web de manera automática, aunque también es posible que el web-master los vaya colocando en la carpeta que él desee y configurar la ruta en el archivo de configuración (Figura B.29).

Paso 3. Realizar pruebas de conexión al servidor web.

Para probar que el servicio está activo, se verifica que el Puerto 80 esté listo y disponible para escuchar las solicitudes de los clientes que visitarán la página web (Figura B.30).

```
# netstat –tulpn | grep :80
```

Figura B.30 Comando para revisar que el puerto 80 esté activo dentro de la configuración de la tarjeta de red.

Una vez que se hayan realizado las pruebas anteriores es necesario configurar los servicios. Para comenzar es necesario realizar los siguientes tres pasos:

1. Iniciar *system-config-services*
2. Iniciar los servicios de *httpd* y *mysqld*.
3. Verificar que el navegador web del sistema funcione correctamente y que al entrar al url *http://localhost/* muestre el mensaje de que el servidor apache está funcionando (Figura B.31).

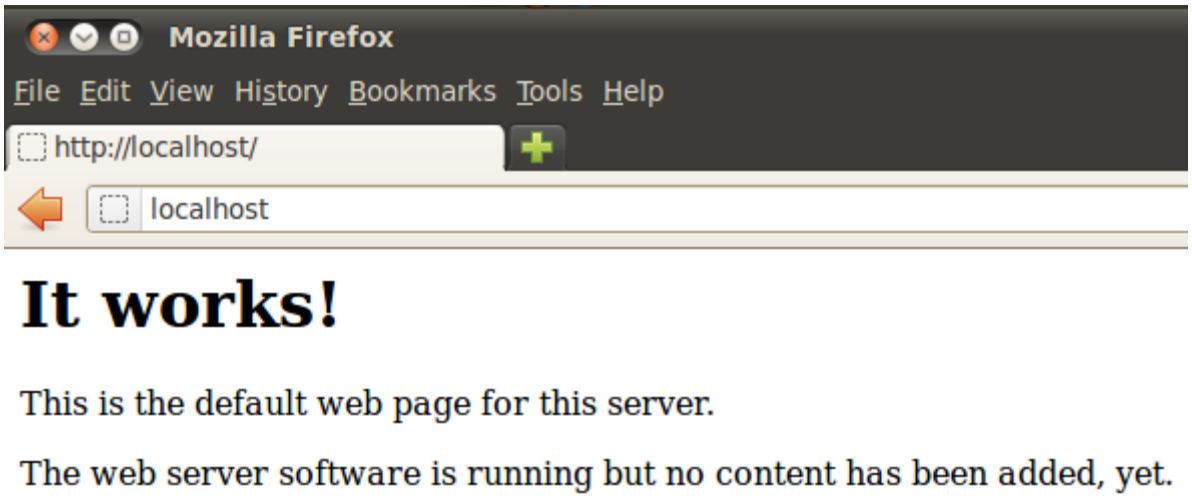


Figura B.31 Mensaje satisfactorio al realizar la prueba de que el servicio de http sirve correctamente.

Una vez que se haya finalizado de configurar el servicio de httpd, es momento de agregar los archivos html dentro de la carpeta ***/var/www/html*** con esto se asegura que los usuarios que envíen una solicitud al puerto 80 del servidor podrán ver el contenido de la página web.

4. Proxies

4.1. Proxy A

Para la configuración del servidor proxy es necesario realizar los siguientes pasos.

Paso 1. Instalar y configurar el servicio de squid para que funcione de manera transparente.

En este punto es necesario instalar el servicio de squid a través del comando yum, justo como se observa en la figura B.32.

```
# yum install squid
```

Figura B.32 Comando de instalación para el servicio squid en un sistema Linux.

Es probable que al ejecutar el comando yum se observe un resultado como se observa en la figura B.33.

```
Loading "installonlyn" plugin
Setting up Install Process
```

```
Setting up repositories
Reading repository metadata in from local files
Parsing package install arguments
Resolving Dependencies
--> Populating transaction set with selected packages. Please wait.
---> Package squid.i386 7:2.6.STABLE6-4.el5 set to be updated
--> Running transaction check
Dependencies Resolved
================================================================================
 Package                 Arch       Version           Repository        Size
================================================================================
Installing:
 squid                   i386       7:2.6.STABLE6-4.el5  updates
1.2 M
Transaction Summary
================================================================================
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)
Total download size: 1.2 M
Is this ok [y/N]: y
Downloading Packages:
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: squid                         ######################### [1/1]
Installed: squid.i386 7:2.6.STABLE6-4.el5
Complete!
```

Figura B.33 Pantalla de progreso al momento de instalación del sistema squid.

Una vez finalizada la instalación hay que configurar ciertos parámetros para que el servicio del squid funcione perfectamente al momento de recibir peticiones de conexión a internet.

Para entrar al archivo de configuración hay que modificar el archivo squid.conf (Figura B.34).

```
# vi /etc/squid/squid.conf
```

Figura B.34 Comando para la edición del archivo de configuración squid.conf.

Una vez dentro del archivo de configuración es necesario modificar los siguientes valores, dependiendo del diseño de la red (Figura B.35).

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
acl lan src 99.90.0.0/16
http_access allow localhost
http_access allow lan
```

Figura B.35 Configuración de parámetros web para el servicio squid.

Donde los comandos anteriores ayudan a:

- httpd_accel_host virtual, donde el squid funciona como un acelerador httpd.
- httpd_accel_port 80, indica que el puerto 80 se utilizará como proxy.
- httpd_accel_with_proxy on, el squid actuará de dos formas, como acelerador httpd y como proxy.
- http_accel_uses_host_header on, la cabecera está acticva que es el hostname del URL.
- acl lan src 99.90.0.0/16, es la lista de acceso que permite a las computadoras de un segmento de red usar el squid.
- http_access_allow localhost, el squid solamente utiliza las listas de acceso del localhost y del segmento de red permitido.
- http_access_allow lan, mismo funcionamiento que el comando anterior.

Paso 2. Configurar el servicio de IP-Tables.

a) Configurar IP Tables en el sistema.

Lo primero que se debe hacer es instalar el servicio de iptables a través del comando yum, el cual es un administrador de software que se utiliza en las diversas distribuciones de Linux (Figura B.36).

```
# yum install iptables
```

Figura B.36 Comando de instalación del servicio de iptables en un sistema Linux.

Lo siguiente será editar el archivo de las reglas que se estarán usando en la paquetería de iptables, para abrir el archivo en donde se encuentran las reglas hay que teclear el comando (Figura B.37).

```
# vi etc/iptables/iptables.rules
```

Figura B.37 Sentencia para editar el archivo de configuración iptables.rules.

Hay que configurar las siguientes listas de acceso que funcionarán con el proxy (Figura B.38).

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j DNAT --to
192.168.11.10:3128
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port
3128
```

Figura B.38 Sintaxis de las listas de acceso utilizadas en el servicio de squid.

Finalmente una vez que se hayan incluido todas las reglas que estará ejecutando el firewall dentro del archivo de iptables, se deberán guardar las modificaciones realizadas en el script.

Se reinicia el servicio y se comprueba que esté activo (Figura B.39).

```
# etc/init.d/iptables restart

Flushing firewall rules:                                   [  OK  ]
Setting chains to policy ACCEPT: filter                    [  OK  ]
Unloading iptables modules:                                [  OK  ]
Applying iptables firewall rules:                          [  OK  ]
Loading additional iptables modules: ip_conntrack_netbios_n[  OK  ]
```

Figura B.39 Mensaje muestra al momento de reiniciar el servicio de squid.

b) Reenviar todas las peticiones http al puerto 3128 (DNAT).

Una vez hecha la configuración básica del servicio de squid es necesario configurar las listas de acceso que reenvíen todas las solicitudes http que lleguen al puerto 80 hacia el puerto del servidor del squid (puerto 3128) como se observa en la figura B.40.

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j DNAT --to
192.168.11.10:3128
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port
3128
```

Figura B.40 Listas de acceso configuradas en el equipo.

El archivo de configuración del proxy tendría que quedar de la manera en la que se observa en la figura B.41.

```
#!/bin/sh
# squid server IP
SQUID_SERVER="192.168.11.10"
# Interface connected to Internet
INTERNET="eth0"
# Interface connected to LAN
LAN_IN="eth1"
# Squid port
SQUID_PORT="3128"
# DO NOT MODIFY BELOW
# Clean old firewall
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
# Load IPTABLES modules for NAT and IP conntrack support
modprobe ip_conntrack
modprobe ip_conntrack_ftp
# For win xp ftp client
#modprobe ip_nat_ftp
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
# Setting default filter policy
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
# Unlimited access to loop back
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# Allow UDP, DNS and Passive FTP
iptables -A INPUT -i $INTERNET -m state --state ESTABLISHED,RELATED -j ACCEPT
# set this system as a router for Rest of LAN
iptables --table nat --append POSTROUTING --out-interface $INTERNET -j
MASQUERADE
iptables --append FORWARD --in-interface $LAN_IN -j ACCEPT
# unlimited access to LAN
iptables -A INPUT -i $LAN_IN -j ACCEPT
iptables -A OUTPUT -o $LAN_IN -j ACCEPT
# DNAT port 80 request comming from LAN systems to squid 3128 ($SQUID_PORT)
aka transparent proxy
iptables -t nat -A PREROUTING -i $LAN_IN -p tcp --dport 80 -j DNAT --to
$SQUID_SERVER:$SQUID_PORT
# if it is same system
iptables -t nat -A PREROUTING -i $INTERNET -p tcp --dport 80 -j REDIRECT --
to-port $SQUID_PORT
# DROP everything and Log it
iptables -A INPUT -j LOG
iptables -A INPUT -j DROP
```

Figura B.41 Archivo de configuración para el servicio de squid.

Paso 3. Correr los scripts e iniciar el servicio de squid.

Finalmente es necesario salvar el script anterior como fw.proxy y darle los permisos para que actúe como un router (Figura B.42).

```
# chmod +x /etc/fw.proxy
# /etc/fw.proxy
# Service iptables sabe
# chkconfig squid on
```

Figura B.42 Sentencia de comandos para dar permisos al archive fw.proxy y revisar la sintaxis del archivo squid.

Se revisa que la sintaxis del script del squid esté correcta y se reinicia el servicio (Figura B.43).

```
# squid –k parse
# /etc/init.d/squid restart
# chkconfig squid on
```

Figura B.43 Comando para revisar la sintaxis del archivo de configuración que utiliza el squid al momento de estar en funcionamiento.

Paso 4. Configuración del proxy en el navegador web.

Para configurar el proxy en el navegador web se realizan los siguientes pasos.

1. En el navegador en el que se vaya a configurar, es necesario ir a herramientas y después al menú de opciones generales o configuración general (Figura B.44).

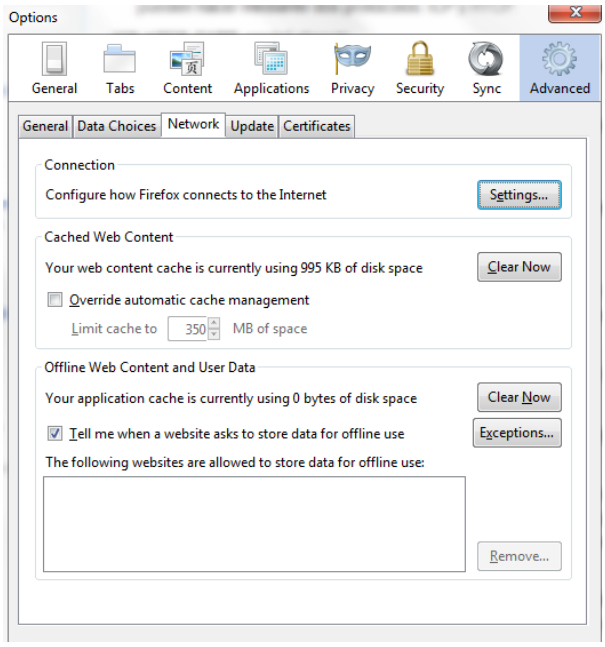


Figura B.44 Menú de Opciones de navegador web mozilla Firefox.

2. Ir a configuración de red (Figura B.45).

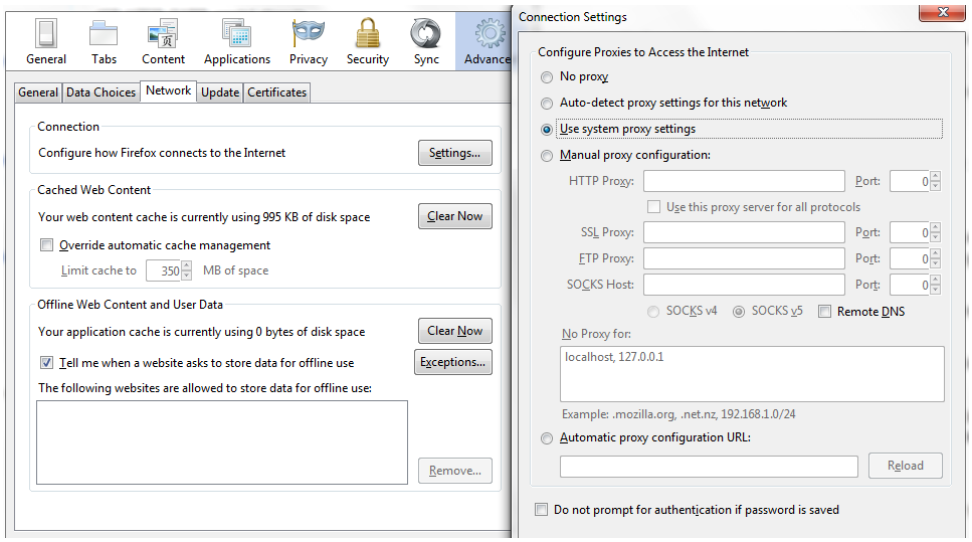


Figura B.45 Configuración de conexiones de red.

3. Llenar la información de la dirección IP y el puerto del servidor proxy (Figura B.46).

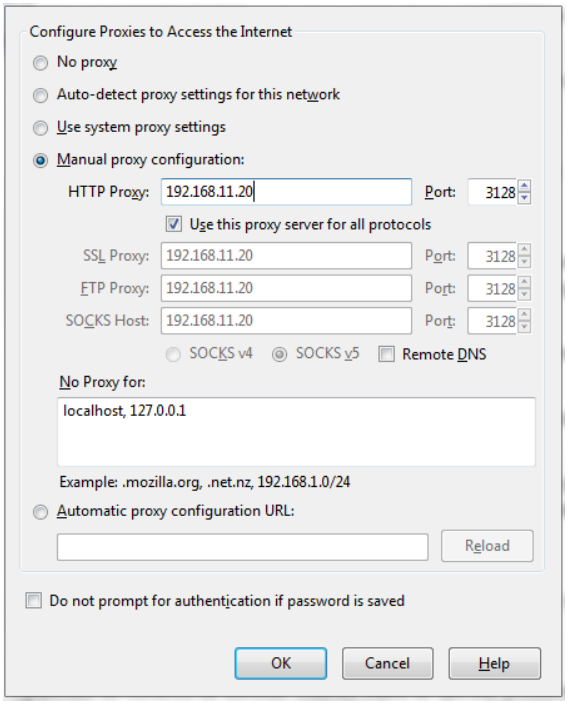


Figura B.46 Configuración del servidor de proxy.

Una vez que se ha configurado el servidor web en la DMZ y todos los usuarios de red interna han configurado los navegadores web, habrá mayor control en la información que entra y sale, desde y hacia internet de todos los usuarios de red interna.

4.2. Proxy B

Para el proxy B se debe realizar el mismo proceso de configuración anterior, tomando en cuenta que la dirección IP cambia para este equipo (figura B.47).

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j DNAT --to
192.168.11.20:3128
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port
3128
```

Figura B.47 Sintaxis de las reglas utilizadas como listas de acceso.

Script del archivo fw.proxy para el Proxy B (Figura B.48).

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

```
# squid server IP
SQUID_SERVER="192.168.11.20"
# Interface connected to Internet
INTERNET="eth0"
# Interface connected to LAN
LAN IN="eth1"
# Squid port
SQUID_PORT="3128"
# DO NOT MODIFY BELOW
# Clean old firewall
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
# Load IPTABLES modules for NAT and IP conntrack support
modprobe ip_conntrack
modprobe ip_conntrack_ftp
# For win xp ftp client
#modprobe ip_nat_ftp
echo 1 > /proc/sys/net/ipv4/ip_forward
# Setting default filter policy
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
# Unlimited access to loop back
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
# Allow UDP, DNS and Passive FTP
iptables -A INPUT -i $INTERNET -m state --state ESTABLISHED,RELATED -j ACCEPT
# set this system as a router for Rest of LAN
iptables --table nat --append POSTROUTING --out-interface $INTERNET -j
MASQUERADE
iptables --append FORWARD --in-interface $LAN_IN -j ACCEPT
# unlimited access to LAN
iptables -A INPUT -i $LAN_IN -j ACCEPT
iptables -A OUTPUT -o $LAN_IN -j ACCEPT
# DNAT port 80 request comming from LAN systems to squid 3128 ($SQUID_PORT)
aka transparent proxy
iptables -t nat -A PREROUTING -i $LAN_IN -p tcp --dport 80 -j DNAT --to
$SQUID_SERVER:$SQUID_PORT
# if it is same system
iptables -t nat -A PREROUTING -i $INTERNET -p tcp --dport 80 -j REDIRECT --
to-port $SQUID_PORT
# DROP everything and Log it
iptables -A INPUT -j LOG
iptables -A INPUT -j DROP
```

Figura B.48 Archivo de configuración para el servicio del squid.

El siguiente paso será configurar todos los equipos de red que interconectarán todos los segmentos de red entre ellos para finalmente haber concluido con el equipo ASA que será el punto neurálgico de la red.

5. Balanceadores de Carga

Pasos para la configuración de los balanceadores de carga dentro de la DMZ.

Paso 1. Configurar las direcciones IP de administración, red interna y red externa del balanceador.

Para configurar la dirección IP de administración del equipo se tomará el segmento de red 10.10.1.48. [15]

Se selecciona una dirección IP para el balanceador BIGSRVA y se configura de la manera en la que se observa en la figura B.49.

```
[root@BIGSRVA: Active] config# b mgmt 192.168.30.33 {netmask 255.255.255.0}
```

Figura B.49 Sintaxis para la configuración de la dirección IP de administración del equipo F5.

Se configuran las rutas de administración con la cual se podrá llegar al equipo (Figura B.50).

```
[root@BIGSRVA: Active] config# b mgmt route 192.168.30.33 netmask
255.255.255.0 '{gateway 192.168.30.1}'
```

Figura B.50 Sintaxis para la configuración de la ruta de administración del balanceador.

El balanceador se guía a través de VLAN's que permiten tener un mayor control de todos los equipos que se balancean y también esto ayuda a que si en un futuro es necesario configurar flujos de red a través del equipo, éste puede utilizar las VLAN's como si fuera un perfil para cada dirección IP.

Por lo tanto se configuran las VLAN para los segmentos de red externo e interno. Es necesario buscar en el archivo de configuración /config/bigip.conf las vlans y deben quedar configuradas como se muestra en la figura B.51.

```
vlan external {
tag 10
interfaces 1.1
}

vlan internal_phy{
tag 4093
interfaces 1.2
}

vlan internal virt{
tag 4092
interfaces 1.4
}
```

Figura B.51 Configuración de las interfaces físicas del balanceador.

[15] El segmento de red se obtuvo del trabajo de segmentación al inicio del capítulo de Balanceadores de carga.

Una vez hecho esto, se configuran los segmentos de red y se configuran las direcciones de red interna y externa como se observa en la figura B.52.

```
[root@BIGSRVA: Active] config# b self 192.168.3.11 '{netmask 255.255.255.240
vlan external allow tcp ssh tcp https tcp 4353}'
```

Figura B.52 Sintaxis para la configuración de dirección IP de red externa del balanceador.

Para configurar la red interna se hace como se puede observar en la figura B.53.

```
[root@BIGSRVA: Active] config# b self 192.168.10.111 '{netmask 255.255.255.0
vlan internal_virt allow tcp https}'
```

Figura B.53 Sintaxis para la configuración de dirección IP de red interna del balanceador.

Finalmente para concluir este paso es necesario abrir el puerto 4353 que es donde estarán llegando todas las peticiones y del cual se realizará el balanceo de las cargas para los servidores que estén en la red internal_virt (Figura B.54).

```
[root@BIGSRVA: Active] config# b self allow '{default udp snmp tcp ssh tcp
domain tcp snmp udp efs tcp 4353 udp domain udp 4353 tcp http proto ospf}'
```

Figura B.54 Sintaxis utilizada para configurar los protocolos que serán balanceados por el equipo.

Con esto el puerto podrá recibir todas aquellas peticiones en tramas udp y tcp que hayan sido declaradas en la sentencia anterior.

Paso 2. Configurar los pools donde se agregarán a los servidores basándose en el servicio que ofrece cada servidor.

Para agregar cada uno de los pools dentro de la configuración por medio de línea de comando se utiliza la sentencia mostrada en la figura B.55.

```
[root@BIGSRVA: Active] config# b pool <nombre_pool> {lb_method
<metodo_balanceo> member <ip_dispositivo:puerto>…
<ip_dispositivo:puerto>…<ip_dispositivo:puerto>… }
```

Figura B.55 Sintaxis del comando para la creación de un pool de balanceo.

Se configurará un pool para los servidores de https (Figura B.56).

```
[root@BIGSRVA: Active] config# b pool http{lb_method …
member 192.168.10.30:80
}
```

Figura B.56 Sintaxis utilizada para agregar nodos dentro de un pool de balanceo.

Con esto se agrega el pool para el servicio de http, pero tomando en cuenta que además se puede configurar el servicio de https, es recomendable configurar un pool especialmente para https (Figura B.57).

```
[root@BIGSRVA: Active] config# b pool https{lb_method …
member 192.168.10.30:443
}
```

Figura B.57 Nodo 192.168.10.30 dentro de un pool de balanceo.

Finalmente se configura el pool donde se agrega el servidor de DNS (Figura B.58).

```
[root@BIGSRVA: Active] config# b pool dns{lb_method …
member 192.168.10.30:443
}
```

Figura B.58 Nodo 192.168.10.30 dentro de un pool de balanceo.

Paso 3. Configurar el método de balanceo.

En la sentencia utilizada en el paso 2 para la creación de un pool se omitió la configuración del método de balanceo por motivos prácticos, pero la sentencia que se utiliza para este proceso se muestra en la figura B.59.

```
[root@BIGSRVA: Active] config# b pool …{lb_method <metodo_balanceo>
member …
}
```

Figura B.59 Sintaxis utilizada para configurar el método de balanceo que se utiliza en cada pool configurado dentro del balanceador.

La configuración queda de la siguiente manera como se observa en la figura B.60.

```
[root@BIGSRVA: Active] config# b pool http{
lb_method least conn
member 192.168.10.30:80
}


[root@BIGSRVA: Active] config# b pool https{
lb_method fastest
member 192.168.10.30:443
}

[root@BIGSRVA: Active] config# b pool dns{
lb_method fastest
member 192.168.10.10:53
}
```

Figura B.60 Pools de balanceo con los servicios de la DMZ configurados en el balanceador.

Así finalmente quedará configurado el método de balanceo en cada uno de los pools creados. Este paso se seguirá cada vez que se cree un pool sin importar el servicio que ofrezcan cada uno de los equipos que se encuentren dentro de la red interna del balanceador.

Paso 4. Configurar el equipo como activo o standby para la opción de failover.

A continuación se muestran los comandos de configuración del *failover* (Figura B.61).

```
[root@BIGSRVA: Active] config# b config sync all
```

Figura B.61 Comando utilizado para la sincronización entre los balanceadores.

Para configurar la dirección IP que trabajará como dirección *float_ip* se utilizarán los siguientes comandos.

Configuración de la dirección *float_ip* en red externa (Figura B.62).

```
[root@BIGSRVA: Active] config# b self 192.168.3.3 {
netmask 255.255.255.240
unit 2
mode
floating enable
vlan external
allow default
}
```

Figura B.62 Configuración de la dirección *float_ip* de red externa dentro del balanceador.

Configuración de dirección *float_ip* en red interna (Figura B.63).

```
[root@BIGSRVA: Active] config# b self 192.168.10.3 {
netmask 255.255.255.0
unit 2
mode
floating enable
vlan internal_virt
allow default
}
```

Figura B.63 Configuración de la dirección *float_ip* de red interna dentro del balanceador.

Finalmente hay que determinar cuál de los dos balanceadores quedará como activo y configurarlo. En este caso se determina que el BIGSRVA será el activo (Figura B.64).

```
[root@BIGSRVA: Active] config# b failover active
```

Figura B.64 Configuración del modo failover como active.

Como el BIGSRVA queda como activo, al momento de configurar el failover en el BIGSRVB se tendrá que configurar como standby.

A continuación de muestra la configuración que se tendrá en el equipo BIGSRVB

- Dirección IP de administración (Figura B.65).

```
[root@BIGSRVB: Active] config# b mgmt 192.168.30.34 {netmask 255.255.255.0}
```

Figura B.65 Sintaxis para la configuración de la dirección IP de administración del equipo F5.

- Ruta de administración (Figura B.66).

```
[root@BIGSRVB: Active] config# b mgmt route 192.168.30.34 netmask
255.255.255.0 '{gateway 192.168.30.1}'
```

Figura B.66 Sintaxis para la configuración de la ruta de administración del balanceador.

- Dirección IP de red interna (Figura B.67).

```
[root@BIGSRVB: Active] config# b self 192.168.10.110 '{netmask 255.255.255.0
vlan internal virt allow tcp https}'
```

Figura B.67 Sintaxis para la configuración de dirección IP de red interna del balanceador.

- Dirección IP de red externa (Figura B.68).

```
[root@BIGSRVB: Active] config# b self 192.168.3.10 '{netmask 255.255.255.240
vlan external allow tcp ssh tcp https tcp 4353}'
```

Figura B.68 Sintaxis para la configuración de dirección IP de red externa del balanceador.

- Pools de balanceo (Figura B.69).

```
[root@BIGSRVB: Active] config# b pool http{
lb_method least conn
member 192.168.10.30:80
}


[root@BIGSRVB: Active] config# b pool https{
lb_method fastest
member 192.168.10.30:443
}
```

```
[root@BIGSRVB: Active] config# b pool dns{
lb_method fastest
member 192.168.10.10:53
}
```

Figura B.69 Pools de balanceo con los servicios de la DMZ configurados en el balanceador.

- Dirección IP de *float_ip interna* (Figura B.70).

```
[root@BIGSRVB: Active] config# b self 192.168.10.3 {
netmask 255.255.255.0
unit 2
mode
floating enable
vlan internal_virt
allow default
}
```

Figura B.70 Configuración de la dirección *float_ip* de red interna dentro del balanceador.

- Dirección de *float_ip externa* (Figura B.71).

```
[root@BIGSRVB: Active] config# b self 192.168.3.3 {
netmask 255.255.255.240
unit 2
mode
floating enable
vlan external
allow default
}
```

Figura B.71 Configuración de la dirección *float_ip* de red externa dentro del balanceador.

- Failover (Figura B.72)

```
[root@BIGSRVB: Active] config# b failover standby
[root@BIGSRVB: Standby] config#
```

Figura B.72 Configuración del modo failover como standby.

Con esto se habrá configurado el balanceo de los servidores de la DMZ. Es importante indicar que el método de configuración será el mismo para los dos proxies que se encuentran en otro segmento de red.

Para fines prácticos se mostrará la configuración del balanceador BIGPRXA.

- Dirección IP de administración (Figura B.73).

```
[root@BIGPRXA: Active] config# b mgmt 192.168.30.35 {netmask 255.255.255.0}
```

Figura B.73 Sintaxis para la configuración de la dirección IP de administración del equipo F5.

- Ruta de administración (Figura B.74).

```
[root@BIGPRXA: Active] config# b mgmt route 192.168.30.35 netmask
255.255.255.0 '{gateway 192.168.30.1}'
```

Figura B.74 Sintaxis para la configuración de la ruta de administración del balanceador.

- Dirección IP de red interna (Figura B.75).

```
[root@BIGPRXA: Active] config# b self 192.168.11.111 '{netmask 255.255.255.0
vlan internal_virt allow tcp https}'
```

Figura B.75 Sintaxis para la configuración de dirección IP de red interna del balanceador.

- Dirección IP de red externa (Figura B.76).

```
[root@BIGPRXA: Active] config# b self 192.168.2.11 '{netmask 255.255.255.240
vlan external allow tcp ssh tcp https tcp 4353}'
```

Figura B.76 Sintaxis para la configuración de dirección IP de red externa del balanceador.

- Pools de balanceo (Figura B.77).

```
[root@ BIGPRXA: Active] config# b pool PROXIES{
lb_method least conn
member 192.168.11.10:3128
member 192.168.11.20:3128
}
```

Figura B.77 Nodos de los servidores proxies configurados dentro del pool de balanceo.

- Dirección IP de *float_ip interna* (Figura B.78).

```
[root@BIGPRXA: Active] config# b self 192.168.11.3 {
netmask 255.255.255.0
unit 2
mode
floating enable
vlan internal_virt
allow default
}
```

Figura B.78 Configuración de la dirección *float_ip* de red interna dentro del balanceador.

- Dirección IP de *float_ip externa* (Figura B.79).

```
[root@BIGPRXA: Active] config# b self 192.168.2.3 {
netmask 255.255.255.240
unit 2
mode
floating enable
vlan external
allow default
}
```

Figura B.79 Configuración de la dirección *float_ip* de red externa dentro del balanceador.

- Failover (Figura B.80).

```
[root@BIGPRXA: Active] config# b failover active
[root@BIGPRXA: Active] config#
```

Figura B.80 Configuración del modo failover como active.

A continuación se muestra la configuración que se tendrá en el equipo BIGPRXB.

- Dirección IP de administración (Figura B.81).

```
[root@ BIGPRXB: Active] config# b mgmt 192.168.30.34 {netmask 255.255.255.0}
```

Figura B.81 Sintaxis para la configuración de la dirección IP de administración del equipo F5.

- Ruta de administración (Figura B.82).

```
[root@ BIGPRXB: Active] config# b mgmt route 192.168.30.34 netmask
255.255.255.0 '{gateway 192.168.30.1}'
```

Figura B.82 Sintaxis para la configuración de la ruta de administración del balanceador.

- Dirección IP de red interna (Figura B.83).

```
[root@BIGPRXB: Active] config# b self 192.168.10.110 '{netmask 255.255.255.0
vlan internal_virt allow tcp https}'
```

Figura B.83 Sintaxis para la configuración de dirección IP de red interna del balanceador.

- Dirección IP de red externa (Figura B.84).

```
[root@BIGPRXB: Active] config# b self 192.168.3.10 '{netmask 255.255.255.240
vlan external allow tcp ssh tcp https tcp 4353}'
```

Figura B.84 Sintaxis para la configuración de dirección IP de red externa del balanceador.

- Pools de balanceo (Figura B.85).

```
[root@BIGPRXB: Active] config# b pool http{
lb_method least conn
member 192.168.10.30:80
}


[root@BIGPRXB: Active] config# b pool https{
lb_method fastest
member 192.168.10.30:443
}


[root@BIGPRXB: Active] config# b pool dns{
lb_method fastest
member 192.168.10.10:53
}
```

Figura B.85 Pools de balanceo con los servicios de la DMZ configurados en el balanceador.

- Dirección IP de *float_ip interna* (Figura B.86).

```
[root@BIGPRXB: Active] config# b self 192.168.10.3 {
netmask 255.255.255.0
unit 2
mode
floating enable
vlan internal_virt
allow default
}
```

Figura B.86 Configuración de la dirección *float_ip* de red interna dentro del balanceador.

- Dirección IP de *float_ip externa* (Figura B.87).

```
[root@BIGPRXB: Active] config# b self 192.168.3.3 {
netmask 255.255.255.240
unit 2
mode
floating enable
vlan external
allow default
}
```

Figura B.87 Configuración de la dirección *float_ip* de red externa dentro del balanceador.

- Failover (Figura B.88).
-

```
[root@BIGPRXB: Active] config# b failover standby
[root@BIGPRXB: Standby] config#
```

Figura B.88 Configuración del modo failover como standby.

Con esto se habrá configurado el balanceo de los servidores de la DMZ. Es importante indicar que el método de configuración será el mismo para los dos proxies que se encuentran en otro segmento de red.

Para fines prácticos se mostrará la configuración del balanceador BIGPRXA.

- Dirección IP de administración (Figura B.89).

```
[root@BIGPRXB: Active] config# b mgmt 192.168.30.35 {netmask 255.255.255.0}
```

Figura B.89 Sintaxis para la configuración de la dirección IP de administración del equipo F5.

- Ruta de administración (Figura B.90).

```
[root@BIGPRXB: Active] config# b mgmt route 192.168.30.35 netmask
255.255.255.0 '{gateway 192.168.30.1}'
```

Figura B.90 Sintaxis para la configuración de la ruta de administración del balanceador.

- Dirección IP de red interna (Figura B.91).

```
[root@BIGPRXB: Active] config# b self 192.168.11.111 '{netmask 255.255.255.0
vlan internal_virt allow tcp https}'
```

Figura B.91 Sintaxis para la configuración de dirección IP de red interna del balanceador.

- Dirección IP de red externa (Figura B.92)

```
[root@BIGPRXB: Active] config# b self 192.168.2.11 '{netmask 255.255.255.240
vlan external allow tcp ssh tcp https tcp 4353}'
```

Figura B.92 Sintaxis para la configuración de dirección IP de red externa del balanceador.

- Pools de balanceo (Figura B.93).

```
[root@BIGPRXB: Active] config# b pool PROXIES{
lb_method least conn
member 192.168.11.10:3128
member 192.168.11.20:3128
}
```

Figura B.93 Nodos de los servidores proxies configurados dentro del pool de balanceo.

- Dirección IP de *float_ip interna* (Figura B.94).

```
[root@BIGPRXB: Active] config# b self 192.168.11.3 {
netmask 255.255.255.0
unit 2
mode
floating enable
vlan internal_virt
allow default
}
```

Figura B.94 Configuración de la dirección *float_ip* de red interna dentro del balanceador.

- Dirección IP de *float_ip externa* (Figura B.95).

```
[root@BIGPRXB: Active] config# b self 192.168.2.3 {
netmask 255.255.255.240
unit 2
mode
floating enable
vlan external
allow default
}
```

Figura B.95 Configuración de la dirección *float_ip* de red externa dentro del balanceador.

- Failover (Figura B.96).

```
[root@BIGPRXB: Active] config# b failover active
[root@BIGPRXB: Active] config#
```

Figura B.96 Configuración del modo failover como active.

6. Firewalls

6.1. FwSrvrs

Paso 1. Configuración básica del firewall.

Para iniciar el proceso de configuración hay que conectar un cable de consola al firewall configurando una terminal con 9600 badios, 8 bits de datos, sin paridad, 1 bit de parada y sin flujo de control. Para crear la conexión se debe instalar un programa cliente para conexiones telnet o ssh (Figura B.97).

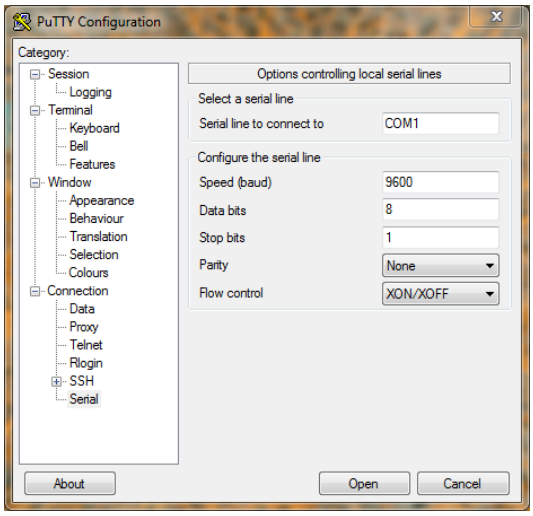


Figura B.97 Ventana de configuración de PuTTY para la conexión con el firewall.

Una vez que se realice la conexión al firewall aparecerá una pantalla como la que se muestra en la figura B.98.

```
C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(25r)FX, RELEASE SOFTWARE (fc4)
Cisco WS-C2960-24TT (RC32300) processor (revision C0) with 21039K bytes of memor
y.
2960-24TT starting...
Base ethernet MAC Address: 00D0.D3C3.A67D
Xmodem file system is available.
Initializing Flash...
flashfs[0]: 1 files, 0 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 64016384
flashfs[0]: Bytes used: 4414921
flashfs[0]: Bytes available: 59601463
flashfs[0]: flashfs fsck took 1 seconds.
...done Initializing Flash.

Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4


Loading "flash:/c2960-lanbase-mz.122-25.FX.bin"...
################################################################################ [OK]
              Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

           cisco Systems, Inc.
           170 West Tasman Drive
           San Jose, California 95134-1706



Cisco IOS Software, C2960 Software (C2960-LANBASE-M), Version 12.2(25)FX, RELEAS
E SOFTWARE (fc1)
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Wed 12-Oct-05 22:05 by pt_team
Image text-base: 0x80008098, data-base: 0x814129C4
```

Figura B.98 Mensaje de arranque del firewall.

Es necesario dar enter para que aparezca el *prompt* como se muestra en la figura B.99.

```
Firewall>
```

Figura B.99 Prompt del firewall.

Seguido de ellos hay que configurar el nombre al equipo y la seguridad como se muestra en las figuras B.100 y B.101.

Con el comando *hostname* se configura el nombre del equipo y esto ayuda a tener un mejor control sobre los dispositivos de red, ya que ayuda a conocer el dispositivo que se está configurando.

```
Firewall>enable
Firewall#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Firewall(config)#hostname FwSrvs
FwSrvs(config)#
```

Figura B.100 Sentencias de configuración para modificar el nombre del equipo.

Lo siguiente es configurar la seguridad de consola y telnet para que el equipo no pueda ser modificado por personas que no tengan las contraseñas de configuración.

```
FwSrvs(config)#line console 0
FwSrvs(config-line)#password cisco
FwSrvs(config-line)#login
FwSrvs(config-line)# exit
FwSrvs(config)#
FwSrvs(config)#line vty 0 15
FwSrvs(config-line)#password cisco
FwSrvs(config-line)#login
FwSrvs(config-line)#exit
FwSrvs(config)#
```

Figura B.101 Sentencias de configuración de las interfaces de consola y telnet del firewall.

Con los pasos anteriores se habrá configurado el firewall con las sentencias primordiales para poder continuar.

Paso 2. Configurar las interfaces de red del firewall.

Se configuran las interfaces de Ethernet del firewall que serán utilizadas de manera activa (Figura B.102).

```
FwSrvs(config)#interface ethernet0 100full
FwSrvs(config)#interface ethernet1 100full
```

Figura B.102 Sentencias de configuración para las interfaces físicas del firewall.

Se asigna la dirección IP desde el modo de configuración tal como se ve en la figura B.103.

```
FwSrvs(config)#ip address inside 192.168.3.2 255.255.255.240
FwSrvs(config)#ip address outside 10.10.1.45 255.255.255.240
```

Figura B.103 Sentencias de configuración de las direcciones IP de cada una de las interfaces físicas del firewall.

Para concluir este paso es necesario configurar un nat en las interfaces; un NAT (network address traslation – traducción de direcciones de red) es la manera de mapear un rango de direcciones de red a una sola dirección de red interna o de la DMZ. Esto permite ocultar la dirección IP real de los servidores para todas aquellas peticiones que llegan desde la red externa.

Para el caso del equipo FwSrvs en particular, se configuran dos nat estáticos para la conexión al servidor de DNS y el servidor web de la subred 192.168.10.0.

Las sentencias que se ingresan en el equipo quedan de la forma en la que se observa en la figura B.104.

```
FwSrvs(config)#static (inside,outside) 10.10.1.45 192.168.3.3 255.255.255.255
0 0
```

Figura B.104 Sentencia de configuración de nat en el firewall.

La explicación de los comandos es la siguiente:

Cualquier solicitud que provenga de la interfaz *outside* e intente conectarse a la dirección IP 10.10.1.45, en realidad se conecta a la dirección IP 192.168.3.3 de la interfaz *inside*, que es la dirección *float_ip* que está conectada a los pools de balanceo. Debido a que es una traducción uno a uno requiere una máscara de red de 255.255.255.255 (Figura B.104).

Paso 3. Configuración del protocolo de enrutamiento.

Comando de configuración de ruta por defecto del firewall FwSrvs (Figura B.105).

```
FwSrvs(config)#ip route outside 0.0.0.0 0.0.0.0 10.10.1.46 1
```

Figura B.105 Configuración de ruta por defecto en el firewall.

Con el comando anterior el firewall enviará a la ruta por defecto aquellos paquetes que no coincidan con las redes que tenga configuradas el firewall dentro de su tabla de ruteo. De esta manera envía todos los paquetes de regreso al ASA que estará trabajando como un router/firewall para que reenvíe el paquete que regresó el FwSrvs por la interfaz correcta o lo descarte de una vez por todas. El número 1 al final de la sentencia anterior, significa que la dirección IP a la que se envía el paquete está a un salto[16] de distancia.

Paso 4. Configuración de las listas de acceso.

Las listas de acceso configuradas en el firewall se pueden observar en la figura B.106.

```
FwSrvs(config)#access-list acl_inside permit tcp any host 192.168.3.3 eq 80
FwSrvs(config)#access-list acl_inside permit tcp any host 192.168.3.3 eq 443
```

[16] Un salto se refiere a los dispositivos intermedios a través de los cuales los datos deben pasar entre el origen y el destino.

```
FwSrvs(config)#access-list acl_inside permit tcp any host 192.168.3.3 eq 53
FwSrvs(config)#access-list acl_inside deny tcp any 192.168.3.3 eq telnel
FwSrvs(config)#access-list acl_inside deny icmp any 192.168.3.3
```

Figura B.106 Configuración de las listas de acceso en el firewall.

6.2. FwdRdInt

Una vez finalizada la configuración es necesario revisar la configuración completa de cada uno de los firewalls, en este caso hay que utilizar el comando *show run config* para que el equipo nos muestre la configuración del equipo. Para fines prácticos se muestra el resultado obtenido del FwdRd Int. (Figura B.107).

```
PIX Version 6.2
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password xxxxxxxxxxxxxx encrypted
passwd xxxxxxxxxxxxxxxx encrypted
hostname FwRdInt
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
no pager
logging on
no logging timestamp
no logging standby
no logging console
no logging monitor
no logging buffered
no logging trap
no logging history
logging facility 20
logging queue 512
interface ethernet0 100full
interface ethernet1 100full
mtu outside 1500
mtu inside 1500
mtu dmz 1500
ip address inside 172.16.0.1 255.255.255.252
ip address outside 10.10.1.29 255.255.255.240
ip audit info action alarm
ip audit attack action alarm
```

```
...
pdm logging informational 100
pdm history enable
arp timeout 14400
nat (inside) 1 172.16.0.0 255.255.255.252 0 0
global (outside) 1 10.10.1.16 netmask 255.255.255.240
static (inside,outside) 172.16.0.0 172.16.0.0 netmask 255.255.255.252 0 0
...
access-list acl_inside permit tcp 172.16.0.0 255.255.0.0 any eq www
access-list acl_inside permit tcp 172.16.0.0 255.255.0.0 any eq ftp
access-list acl_inside permit tcp 172.16.0.0 255.255.0.0 any eq 443
access-list acl_inside permit tcp 172.16.0.0 255.255.0.0 any eq 53
access-list acl_inside permit tcp 172.16.0.0 255.255.0.0 any eq 3128
...
route inside 172.16.0.0 255.255.255.0 172.16.0.2 1
route inside 172.20.0.0 255.255.255.0 172.16.0.2 1
route outside 0 0 10.10.1.30 1
timeout xlate 0:05:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip
0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
http server enable
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet 172.16.0.0 255.255.0.0 inside
telnet timeout 5
ssh timeout 5
terminal width 80
Cryptochecksum:6380fd60019789e34aac0d059a269
: end
```

Figura B.107 Archivo de configuración del firewall FwdRdInt.

6.3. FwdPrx

Del mismo modo se ingresa el comando que muestra la configuración que está siendo utilizada en el equipo FwdPrx (Figura B.108).

```
PIX Version 6.2
nameif ethernet0 outside security0
nameif ethernet2 DMZ security50
nameif ethernet1 inside security100
enable password xxxxxxxxxxxxxx encrypted
passwd xxxxxxxxxxxxxxxxx encrypted
```

```
hostname FwPrx
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
fixup protocol squid 3128
names
no pager
logging on
no logging timestamp
no logging standby
no logging console
no logging monitor
no logging buffered
no logging trap
no logging history
logging facility 20
logging queue 512
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
mtu outside 1500
mtu inside 1500
mtu dmz 1500
ip address dmz 192.168.2.2 255.255.255.240
ip address inside 10.10.1.49 255.255.255.240
ip address outside 10.10.1.2 255.255.255.240
ip audit info action alarm
ip audit attack action alarm
pdm logging informational 100
pdm history enable
arp timeout 14400
...
nat (inside) 10.10.1.48 255.255.255.240 0 0
nat (dmz) 1 192.168.2.0 255.255.255.240 0 0
global (outside) 1 10.10.1.15 netmask 255.255.255.0
...
static (inside,dmz) 10.10.1.12 192.168.2.3 netmask 255.255.255.255 0 0
static (dmz, outside) 10.10.1.11 192.168.2.3 netmask 255.255.255.255 0 0
...
access-list acl_inside permit tcp 10.10.1.48 255.255.255.240 any eq www
access-list acl_inside permit tcp 10.10.1.48 255.255.255.240 any eq ftp
access-list acl_inside permit tcp 10.10.1.48 255.255.255.240 any eq 443
access-list acl_inside permit tcp 10.10.1.48 255.255.255.240 any eq 3128
access-list acl_dmz permit tcp 192.168.2.0 255.255.255.240 any eq www
access-list acl_dmz permit tcp host 192.168.2.3 host 10.10.1.49 eq 3128
access-list acl_outside deny tcp any host 192.168.2.3 eq www
access-list acl_outside deny tcp any host 192.168.2.3 eq 443
...
route inside 10.10.1.48 255.255.255.240 10.10.1.50 1
```

```
route outside 0 0 10.10.1.1 1
...
timeout xlate 0:05:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip
0:30:00 sip media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
http server enable
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
terminal width 80
Cryptochecksum:6380fd60019789e34aac0d059a269
: end
[OK]
```

Figura B.108 Archivo de configuración del firewall FwPrx.

6.4. FwRdExt

Archivo de configuración del firewall FwRdExt (Figura B.109).

```
PIX Version 6.2
nameif ethernet0 outside security0
nameif ethernet2 DMZ security50
nameif ethernet1 inside security100
enable password xxxxxxxxxxxxxx encrypted
passwd xxxxxxxxxxxxxxxxx encrypted
hostname FwRdExt
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
no pager
logging on
no logging timestamp
no logging standby
no logging console
no logging monitor
no logging buffered
no logging trap
```

```
no logging history
logging facility 20
logging queue 512
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
mtu outside 1500
mtu inside 1500
mtu dmz 1500
ip address dmz 10.10.1.1 255.255.255.240
ip address inside 10.10.1.65 255.255.255.240
ip address outside 192.168.1.4 255.255.255.240
ip audit info action alarm
ip audit attack action alarm
pdm logging informational 100
pdm history enable
arp timeout 14400
...
nat (inside) 10.10.1.64 255.255.255.240 0 0
nat (dmz) 1 10.10.1.0 255.255.255.240 0 0
global (outside) 1 192.168.1.15 netmask 255.255.255.0
...
access-list acl_inside permit tcp 10.10.1.64 255.255.255.240 any eq www
access-list acl_inside permit tcp 10.10.1.64 255.255.255.240 any eq ftp
access-list acl_inside permit tcp 10.10.1.64 255.255.255.240 any eq 443
access-list acl_dmz permit tcp 10.10.1.0 255.255.255.240 any eq www
access-list acl_outside deny tcp any host 10.10.1.66 eq www
access-list acl_outside deny tcp any host 10.10.1.66 eq 443
...
route inside 10.10.1.64 255.255.255.240 10.10.1.66 1
route outside 0 0 192.168.1.3 1
...
timeout xlate 0:05:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip
0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
http server enable
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable
telnet timeout 5
ssh timeout 5
terminal width 80
Cryptochecksum:6380fd60019789e34aac0d059a269
: end
[OK]
```

Figura B.109 Archivo de configuración del firewall FwRdExt.

7. ASA CISCO

Pasos para la configuración del equipo ASA de Cisco.

Paso 1. Configuración básica del equipo (nombre del equipo, direcciones IP de las interfaces que estarán activas, nombre de las interfaces).

Para iniciar se configura de manera básica el ASA, nombrando cada una de las interfaces del equipo según el equipo que esté conectado al otro extremo (Figuras B.110, B111, B.112 y B.113).

```
ASACTRL(config)#interface ethernet 0/0
ASACTRL(config)#ip address 10.10.1.66 255.255.255.240
ASACTRL(config)#nameif fwrdext
ASACTRL(config)#no shutdown
```

Figura B.110 Comandos utilizados para la configuración de la interfaz de red 0/0.

```
ASACTRL(config)#interface ethernet 0/1
ASACTRL(config)#ip address 10.10.1.50 255.255.255.240
ASACTRL(config)#nameif fwprx
ASACTRL(config)#no shutdown
```

Figura B.111 Comandos utilizados para la configuración de la interfaz de red 0/1.

```
ASACTRL(config)#interface ethernet 0/2
ASACTRL(config)#ip address 10.10.1.30 255.255.255.240
ASACTRL(config)#nameif fwrdint
ASACTRL(config)#no shutdown
```

Figura B.112 Comandos utilizados para la configuración de la interfaz de red 0/2.

```
ASACTRL(config)#interface ethernet 0/3
ASACTRL(config)#ip address 10.10.1.46 255.255.255.240
ASACTRL(config)#nameif fwsrvs
ASACTRL(config)#no shutdown
```

Figura B.113 Comandos utilizados para la configuración de la interfaz de red 0/3.

Paso 2. Configurar el Protocolo de enrutamiento del equipo.

Debido a que el ASA será la parte central de la red y por donde todos los paquetes pasen para ser enviados a las interfaces correctas es recomendable configurar un protocolo de enrutamiento de alto rendimiento. Para tal objetivo se configura OSPFv2 en el equipo, agregando todas las interfaces que se encuentran directamente conectadas al equipo para que el propio equipo cree su tabla de enrutamiento de forma automática (Figura B.114).

```
ASACTRL(config)#router ospf 2
ASACTRL(config)#network 10.10.1.64 255.255.255.240 area 0
ASACTRL(config)#network 10.10.1.48 255.255.255.240 area 0
ASACTRL(config)#network 10.10.1.16 255.255.255.240 area 0
ASACTRL(config)#network 10.10.1.32 255.255.255.240 area 0
```

Figura B.114 Comandos utilizados para la configuración del protocolo de enrutamiento OSPF.

Paso 3. Configurar las listas de acceso del equipo.

El siguiente paso es configurar las listas de acceso en cada interfaz basándose en la topografía lógica que se explicó al inicio del capítulo. Las listas de acceso se configuran como se muestra en la figura B.115.

```
ASACTRL(config)# ASACTRL(config)#access-list acl_fwext permit tcp 10.10.1.64
255.255.255.240 any eq www
ASACTRL(config)#access-list acl_fwext permit tcp 10.10.1.64 255.255.255.240
any eq ftp
ASACTRL(config)#access-list acl_fwext permit tcp 10.10.1.64 255.255.255.240
any eq 443

ASACTRL(config)#access-list acl_fwprx permit tcp 10.10.1.48 255.255.255.240
any eq www
ASACTRL(config)#access-list acl_fwprx permit tcp 10.10.1.48 255.255.255.240
any eq ftp
ASACTRL(config)#access-list acl_fwprx permit tcp 10.10.1.48 255.255.255.240
any eq 443
ASACTRL(config)#access-list acl_fwprx permit tcp 10.10.1.48 255.255.255.240
any eq 3128

ASACTRL(config)#access-list acl_fwint permit tcp 172.16.0.0 255.255.0.0 any
eq www
ASACTRL(config)#access-list acl_fwint permit tcp 172.16.0.0 255.255.0.0 any
eq ftp
ASACTRL(config)#access-list acl_fwint permit tcp 172.16.0.0 255.255.0.0 any
eq 443
ASACTRL(config)#access-list acl_fwint permit tcp 172.16.0.0 255.255.0.0 any
eq 53
ASACTRL(config)#access-list acl_fwint permit tcp 172.16.0.0 255.255.0.0 any
eq 3128

ASACTRL(config)#access-list acl_dmz permit tcp 10.10.1.32 255.255.255.240 any
eq 80
ASACTRL(config)#access-list acl_dmz permit tcp 10.10.1.32 255.255.255.240 any
eq 443
ASACTRL(config)#access-list acl_dmz permit tcp 10.10.1.32 255.255.255.240 any
eq 53
```

Figura B.115 Listas de acceso configuradas en el equipo ASA.

Como consecuencia de haber implementado el protocolo de enrutamiento en las interfaces directamente conectadas al ASA, al momento de ingresar las listas de acceso el equipo hará un igualamiento con los segmentos de red que tiene en la tabla de enrutamiento.

Con esto se finaliza la configuración básica del ASA para la DMZ, brindando mayor seguridad a la red y detectando todos aquellos paquetes que pasen a través de éste.

8. Router de GW

Pasos para la configuración del router.

Paso 1. Configuración básica del router, paso en donde se realiza la configuración del nombre y seguridad de las líneas de conexión de consola y telnet (Figura B.116).

```
Router>enable
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname RouterDeBorde
RouterDeBorde (config)#
RouterDeBorde (config)#line console 0
RouterDeBorde (config-line)#password cisco
RouterDeBorde (config-line)#login
RouterDeBorde (config-line)# exit
RouterDeBorde (config)#
RouterDeBorde (config)#line vty 0 15
RouterDeBorde (config-line)#password cisco
RouterDeBorde (config-line)#login
RouterDeBorde (config-line)#exit
RouterDeBorde (config)#
```

Figura B.116 Comandos de configuración básica para el router RouterDeBorde.

Paso 2. Configuración de las interfaces del router.

El siguiente paso es configurar las interfaces del router, según la topología mostrada en la Figura B.108.

Las interfaces que se configuran son las FastEthernet0/0 y la FastEthernet0/1; la primera estará conectada directamente con el ISP y la segunda estará conectada al firewall FwRdExt. A continuación se muestran los comandos de configuración para cada una de las interfaces como se observa en las figuras B.117 y B.118.

```
RouterDeBorde (config)#interface fastEthernet 0/0
RouterDeBorde (config-if)#ip address 192.168.1.3 255.255.255.240
RouterDeBorde (config-if)#no shutdown
RouterDeBorde (config-if)#description line interface connected with DMZ
RouterDeBorde (config-if)#exit
RouterDeBorde (config)#
```

Figura B.117 Comandos utilizados para la configuración de la interfaz de red 0/0 del router.

```
RouterDeBorde (config)#interface serial 0/1/0
RouterDeBorde (config-if)#ip address 212.97.110.58 255.255.255.252
RouterDeBorde (config-if)#no shutdown
RouterDeBorde (config-if)#description line interface connected to the ISP
RouterDeBorde (config-if)#exit
RouterDeBorde (config)#
```

Figura B.118 Comandos utilizados para la configuración de la interfaz serial 0/1/0 del router.

Paso 3. Configuración del protocolo de enrutamiento.

Finalmente se configura el protocolo de enrutamiento y las rutas por defecto. Para el protocolo de enrutamiento se utiliza el protocolo OSPFv2 para que el router realice el proceso de identificar las sub-redes que se encuentren conectadas directamente al equipo y por otra parte se configuran dos rutas por defecto en caso de que el equipo no tenga una sub-red dentro de su tabla de enrutamiento.

Para la configuración se utilizan los comandos mostrados en la Figura B.119.

```
RouterDeBorde (config)#router ospf 2
RouterDeBorde (config)#network 192.168.1.0 255.255.255.240 area 0
RouterDeBorde (config)#network 212.97.110.56 255.255.255.252 area 0
RouterDeBorde (config)#
```

Figura B.119 Comandos utilizados para la configuración del protocolo de enrutamiento OSPF.

Ahora se configuran la ruta por defecto (Figura B.120).

```
RouterDeBorde (config)#ip route 0.0.0.0 0.0.0.0 212.97.110.57
```

Figura B.120 Comando utilizado para la configuración de la ruta por defecto del router.

Referencias

Referencias

- ¿Qué es balanceo de carga?. Recuperado en mayo de 2013 de: http://www.computerworld.es/tendencias/que-es-el-balanceo-de-carga

- ANSI. Recuperado en abril de 2012 de: http://www.ansi.org/about_ansi/overview/overview_sp.aspx?menuid=1

- Balanceador de servicios Web. Recuperado en abril de 2013 de: http://www.networkworld.es/archive/balanceador-de-servicios-web

- Balanceo de Red: Round Robin DNS + NLS. Recuperado en mayo de 2013 de: http://www.bujarra.com/pdfs/ProcedimientoBalanceoLAN.pdf

- Cable par trenzado. Recuperado en febrero de 2012 de: http://docente.ucol.mx/al972052/public_html/CABLE%20PAR%20TRENZADO.htm.

- Cableado estructurado. Recuperado en abril de 2012 de: http://materias.fi.uba.ar/6679/apuntes/CABLEADO_ESTRUC.pdf

- Cabling. Recuperado en febrero de 2012 de: http://fcit.usf.edu/network/chap4/chap4.htm.

- Características de conectores para fibra óptica. Recuperado en febrero de 2012 de: http://www.paratorpes.es/manuales/conectores%20opticos.pdf

- Cisco Networking Academy. (2011). CCNA Exploration 5.0, Módulo 1: Aspectos básicos de Networking. México. Pearson Prentice Hall.

- Cisco Router Architecture. Recuperado en marzo de 2013 de: http://www.cisco.com/networkers/nw99_pres/601.pdf.

- Cisco Router Hardware. Recuperado en marzo de 2013 de: http://www.skullbox.net/routers.php.

- Configuración básica de un router cisco. Recuperado en febrero de 2013 de: http://davidvegabonilla.com/routerbasico.html.

- EAP-TLS Deployment Guide for Wireless LAN networks. Recuperado en junio de 2012 de: http://www.cisco.com/en/US/tech/tk722/tk809/technologies_white_paper09186a008009256b.shtml

- Estándar IEEE 802.11. Recuperado en marzo de 2012 de: http://standards.ieee.org/getieee802/download/802.11-2012.pdf

- Estándar IEEE 802.16. Recuperado en marzo de 2012 de: http://standards.ieee.org/getieee802/download/802.16-2012.pdf

- Gráficas Métodos de Codificación (NRZ). Recuperado en Octubre de 2011 de: http://es.kioskea.net/contents/transmission/transnum.php3

- Guía rápida para routers de la serie Cisco 2800. Recuperado en marzo de 2013 de: http://www.cisco.com/application/pdf/en/us/guest/products/ps5854/c1616/ccmigration_09186a00802c35a2.pdf.

- Joskiwicz, J. (2008), Redes Corporativas, Uruguay, (pp. 4 – 10).

- Radiación térmica y radiación solar. Recuperado en marzo de 2012 de:
- http://www.blogdequk.com/2011/04/la-radiacion-termica-y-radiacion-solar.html

- Redes CDMA y GSM. Recuperado en abril del 2012 de: http://www.pac.com.ve/index.php?option=com_content&view=article&catid=68&Itemid=91&id=4906

- Redes de Datos LAN. Recuperado en Enero de 2012 de: http://www.uazuay.edu.ec/estudios/electronica/proyectos/redes_de_datos_lan.pdf

- Redes GSM. Recuperado en junio de 2012 de: www.dsi.fceia.unr.edu.ar/downloads/.../**RedesGSM**.pdf

- Schmied, W., Imperatore, D., Shinder. T., Shimonski. R. J., Chang, V., Simonis, D. (2003). Building DMZ's for Enterprise Networks. Estados Unidos. Syngress Publishing, Inc.

- Tabla Cable UTP categoría 3 – características. Recuperado en Octubre del 2011 de: http://www.ampnetconnect.com/documents/Cat_3_UTP_4-Pair_Cable_Cut_Sheet_(040430).pdf

- Tabla Cable UTP categoría 6. Recuperado en Octubre del 2011 de: http://media.extron.com/download/files/specs/UTP_CAT_6_cable_020402.pdf

- Tabla de características de los principales cables coaxiales. Recuperado en febrero de 2012 de: http://www.electronicafacil.net/tutoriales/Tabla-cable-coaxial.php

- Tabla fibra monomodo vs multimodo. Recuperado en Febrero de 2012 de: http://grupoorion.unex.es/fibras/ayudas/monovsmulti.htm

- Técnicas de balanceo de carga. Recuperado en mayo de 2013 de: http://julianrv.com/blog/2006/01/tecnicas-de-balanceo-de-carga-nlb-y-round-robin.html

- Telecomunicaciones a través de fibras ópticas. Recuperado en Febrero de 2012 de: http://www.slideshare.net/yussting/fibra-optica-1431491

- Topologías de red. Recuperado en Enero de 2012 de: http://alexalvarez0310.wordpress.com/topologias-de-red/

- Topologías de redes de datos. Recuperado en febrero de 2012 de: http://ocw.upm.es/...y.../**topografia**-ii/Tema_10_Teoria.pdf

- Zona Desmilitarizada. Recuperado en marzo de 2014 de: http://es.kioskea.net/contents/589-dmz-zona-desmilitarizada

GLOSARIO

Glosario:

A

- **µm (nanómetro)**: es la unidad de longitud que equivale a una mil millonésima parte de un metro (1nm = 10^{-9}).

- **ACK**: es un bit dentro del segmento TCP, se utiliza para enviar un mensaje para confirmar la recepción del mensaje inicial SYN, que fue enviado por el host origen para iniciar el protocolo de comunicación.

- **ACL (Access Control List – lista de control de acceso)**: es un concepto de seguridad informática usado para fomentar la separación de privilegios en las redes de datos. A través de las listas de acceso es posible determinar los permisos apropiados para un determinado usuario, dependiendo de ciertos aspectos del proceso que hace el pedido.

- **ANSI (American National Standards Institute - Instituto Nacional de Normalización Estadounidense**): es una organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos. También puede acreditar a organizaciones que realizan certificaciones de productos o personal de acuerdo con los requisitos definidos en los estándares internacionales.

- **ASA (Adaptive Security Appliance – equipo de seguridad adaptable)**: equipo de seguridad de Cisco Systems de la familia ASA, los cuales son normalmente utilizados para proteger redes corporativas de cualquier tamaño. Proporciona a los usuarios un acceso altamente seguro a los datos.

- **ASACTRL:** término utilizado en el proyecto de tésis para referirse al equipo ASA – CENTRAL; dispositivo que se encuentra en el centro lógica de la dmz desarrollada en este proyecto.

- **AWG (american wire gauge – calibre de cable americano)**: es una referencia de clasificación de diámetros. Cuanto más alto es este número, más delgado es el alambre. El alambre de mayor grosor es menos susceptible a la interferencia, posee menos resistencia interna y soporta mayores corrientes a distancias más grandes.

B

- **Backbone (columna vertebral)**: se refiere a las conexiones principales de internet o de una red. Se compone de un gran número de enrutadores comerciales, gubernamentales y universitarios interconectados y que llevan los datos a través de los países, continentes y océanos del mundo mediante cables de fibra óptica.

- **BGP (Border Gateway Protocol – Protocolo de Gateway de borde)**: protocolo de enrutamiento mediante el cual se intercambia información entre sistemas de diferentes fabricantes. Entre los sistemas autónomos de los ISP se intercambian sus tablas de rutas a través del protocolo BGP. Este intercambio de información de encaminamiento se hace entre los routers externos de cada sistema autónomo, los cuales deben soportar BGP. Se trata del protocolo más utilizado en redes con intención de configurar un Exterior Gateway Protocol.

- **BIOS (Basic Input/Output System – Sistema básico de entrada/salida)**: es un programa informático inscrito en componentes electrónicos de memoria flash existentes en la placa base. Este programa controla el funcionamiento de la placa base y de dichos componentes. Se encarga de realizar las funciones básicas de manejo y configuración de la computadora.

- **Booteables**: en informática es la capacidad de iniciar la secuencia de arranque desde un dispositivo periférico. El que el usuario enciende la computadora y el sistema se encarga de iniciar el sistema que se encuentra almacenado en un dispositivo de memoria ajeno a la computadora.

- **Buffer**: es un espacio de la memoria en un disco o en un instrumento digital que está reservada para el almacenamiento temporal de información digital. Mientras que está esperando a ser procesada.

C

- **CFO (*Chief Financial Officer):*** es el ejecutivo a cargo de la gestión financiera de una compañía u organización. Es el responsable de la planificación, ejecución e información financiera.

- **Checksum (suma de verificación)**: es una función *hash* que tiene la función de detectar cambios accidentales en una secuencia de datos para proteger la integridad de estos, verificando que no haya discrepancias entre los valores obtenidos al hacer una comprobación inicial y otra final tras la transmisión.

- **Clock Rate**: se refiere a la frecuencia a la que un CPU puede ejecutar las tareas asignadas. Esta unidad de medida se mide en Hertz.

- ***Cloud*:** se refiere a un nuevo concepto de prestación de servicios de negocio y tecnología bajo la abstracción de la infraestructura técnica que brinda un servicio a través de internet.

- **CMOS (Complementary Metal-oxide-semiconductor – Semiconductor complementario de óxido metálico)**: es una de las familias lógicas empleadas en la fabricación de circuitos integrados. Su principal característica consiste en la utilización conjunta de transistores de tipo pMOS y tipo nMOS configurados de tal forma que en reposos, el consumo de energía es únicamente el debido a las corrientes parásitas, colocando obviamente en la placa base.

- **CPU (Central Processing Unit – Unidad de procesamiento central)**: es el componente principal de una computadora y el cual interpreta las instrucciones contenidas en los programas y procesa los datos.

- **CSU/DSU (Channel Service Unit / Data Service Unit – Unidad de Servicio de Canal / Unidad de Servicio de Datos)**: una unidad de servicio de canal (CSU) es un dispositivo que conecta un terminal a una línea digital, mientras que una unidad de servicio de datos (DSU) es un dispositivo que lleva a cabo funciones de protección y de diagnóstico en una línea de telecomunicaciones. En general, los dos dispositivos se entregan formando una sola unidad, CSU/DSU. Una CSU/DSU es un módem muy potente y caro. Se requiere un dispositivo como este para cada extremo de una conexión T-1 o T-3; las unidades que están en los dos extremos deben ser del mismo fabricante.
  La unidad de servicio de canal (CSU) recibe y transmite señales desde y hacia la línea de WAN, proporcionando la barrera para la interferencia eléctrica de cada lado de la unidad. Por otra parte, la unidad de servicio de datos (DSU) gestiona el control de la línea y la convierte las tramas de entrada y salida entre los conectores RS-232C, RS-449 o tramas V.35 desde la red de área local y el multiplexado por división de tiempo (TDM).

D

- **DMZ (Demilitarized zone):** siglas de denominación para una zona desmilitarizada, la cual consta de una red local que se ubica entre red interna de una organización y una red externa (internet).

- **Debugging**: es el proceso de identificar y corregir errores de programación.

- **DNS (Domain Name System – Sistema de nombres de dominio)**: es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a internet o a una red privada. Su función principal, es traducir nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red.

- **DSL (Digital Subscriber Line – Línea de Suscripción Digital)**: tecnología que proporciona acceso a internet mediante la transmisión de datos digitales a través de los cables de una red telefónica local. El servicio DSL es entregado simultáneamente con el servicio telefónico por cable en la misma línea telefónica, haciendo posible que esta tecnología utilice bandas de frecuencia más altas para los datos.

E

- **EEPROM (Electrically Erasable Programmable read-only memory – memoria de solo lectura programmable y borrada eléctricamente)**: es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente, a diferencia de la EPROM que debe borrarse mediante un aparato que emite rayos ultravioleta. Las celdas de memoria de una EEPROM están constituidas por un transistor MOS, que tiene una compuerta flotante, la cual en su estado normal proporciona una salida lógica de 1.

- **EIA (Electronic Industries Alliance - Alianza de Industrias Electrónicas)**: es una organización formada por la asociación de las compañías electrónicas y de alta tecnología de los Estados Unidos, su misión es promover el desarrollo del mercado y la competitividad de la industria de alta tecnología a través de esfuerzos locales (dentro de los Estados Unidos) e internacionales. Cuenta con 1,300 empresas del sector como miembros y cuyos productos y servicios abarcan desde los componentes electrónicos, hasta los sistemas más complejos que son usados en defensa militar, el espacio y la industria.

- **EIGRP (Enhanced Interior Gateway Routing Protocol - Protocolo de enrutamiento de Gateway interno mejorado)**: es un protocolo de enrutamiento vector distancia avanzado propiedad de CISCO, que ofrece lo mejor de los algoritmos de vector de distancias y del estado de enlace. Sus características se basan en la mejora de las propiedades de convergencia y que opera con mayor eficiencia que IGRP, estas características son: (i) protocolo de transporte confiable (RTP); (ii) actualizaciones limitadas; (iii) algoritmo de actualización de difusión (DUAL); (iv) establecimiento por adyacencias y (v) tablas de vecinos y topología.

  Las rutas reciben un estado y se pueden rotular para proporcionar información adicional de utilidad, ya que mantiene tres tablas que le ayudan a cumplir esta función: tabla de vecinos, tabla de topología y tabla de encaminamiento.

- **EPROM (Erasable Programmable read-only memory – memoria de solo lectura programable borrable)**: es un tipo de memoria ROM no volátil formada por celdas de transistores de puerta flotante.

- **Ethernet**: es un estándar de redes de área local para computadoras con acceso al medio por detección de la onda portadora y con detección de colisiones (CSMA/CD). Este estándar define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

F

- **Fast-Ethernet**: es el nombre de una serie de estándares de IEEE de redes Ethernet de 100 Mbps. En su momento se le agrego el prefijo *fast* para diferenciarla de la versión original Ethernet de 10 Mbps.

- **Firewalls (cortafuegos)**: sistema que está diseñado para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas. Se puede tratar de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar o descifrar el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

- **Frame Relay**: es una técnica de comunicación mediante retransmisión de tramas para redes de circuito virtual, introducida por la ITU-T a partir de la recomendación I.122. Esta recomendación consiste en una forma simplificada de tecnología de conmutación de paquetes que transmite una variedad de tamaños de tramas para datos, perfecto para la transmisión de grandes cantidades de datos.

- **FTP (File Tranfer Protocol – Protocolo de Transferencia de Archivos)**: es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. El servicio FTP es ofrecido por la capa de aplicación utilizando el puerto de red 20 y 21.

H

- **HDLC (High-level Data Link Control – Control de enlace de datos de alto nivel)**: es un protocolo de comunicaciones de propósito general punto a punto y multipunto, que opera a nivel de enlace d datos. Proporciona recuperación de errores en caso de pérdida

de paquetes de datos, fallos de secuencia y otros, por lo que ofrece una comunicación confiable entre el transmisor y el receptor.

- **HTTP (Hyper Text Transfer Protocol – Protocolo de Transferencia de Hypertexto)**: protocolo usado en cada transacción de *World Wide Web*. Este protocolo está orientado a transacciones y que sigue el esquema petición-respuesta entre un cliente y un servidor. A la información transmitida se le llama recurso y se le identifica mediante un localizador uniforme de recursos (URL).

I

- **IANA (Internet Assigned Numbers Authority - Autoridad de números asignados de internet)**: es la que supervisa la asignación global de direcciones IP, sistemas autónomos, servidores raíz de nombres de dominio y otros recursos relativos a los protocolos de internet.

- **IEEE, (Institute of Electrical and Electronics Engineers - Instituto de Ingenieros Eléctricos y de Electrónica)**: es una asociación técnico-profesional mundial dedicada a la estandarización la cual se encuentra formada por ingenieros del ramo electrónico e informático. Su objetivo es promover la creatividad, el desarrollo y la integración, compartir y aplicar los avances en las tecnologías de la información, electrónica y ciencias en general para el beneficio de la humanidad y de los mismos profesionales.

- **IGRP (Interior Gateway Routing Protocol – Protocolo de enrutamiento de Gateway interno)**: es un protocolo desarrollado y patentado por CISCO que se emplea con el protocolo TCP/IP según el modelo de internet. Este protocolo se basa en la tecnología vector-distancia, el cual utiliza una métrica compuesta para determinar la mejor ruta basándose en el ancho de banda, el retardo, la confiabilidad y la carga del enlace. El concepto se basa en la regla de que cada router no necesita saber todas las relaciones de ruta/enlace para la red entera.

- **IOS (Internetwork Operating System)**: es el software utilizado en la mayoría de routers y switches de Cisco Systems. IOS es un paquete de funciones de enrutamiento, conmutamiento, trabajo de internet y telecomunicaciones que se integra estrechamente con un sistema operativo multitarea.

- **IP (Internet Protocol – Protocolo de internet)**: es un protocolo de comunicación de datos digitales clasificado funcionalmente en la capa de red. Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un

protocolo no orientado a conexión que transfiere paquetes conmutados a través de distintas redes físicas previamente enlazadas según la norma OSI de enlace de datos.

- **IPX (Internetwork Packet Exchange – Paquete de Intercambio de red)**: es un protocolo de la cpaa de red de ***Netware*** que se utiliza para transferir datos entre el servidor y los programas de las estaciones de trabajo. En este protocolo los datos se transmiten en datagramas, proveyendo servicios en las capas 3 y 4 del modelo OSI.

- **ISDN (Integrated Services Digital Network – Red Digital de servicios integrados)**: es una red que facilita conexiones digitales extremo a extremo para proporcionar una amplia gama de servicios, tanto de voz como de otros tipos, y a la que los usuarios acceden a través de un conjunto de interfaces normalizados.

- **ISO (International Organization for Standardization - Organización Internacional de Estandarización)**: organismo encargado de promover el desarrollo de normas internacionales de fabricación (productos y servicios), comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. El objetivo principal de esta organización es buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

- **ISP (Internet service provider – proveedor de servicios de internet)**: es una empresa que brinda conexión a internet a sus clientes. Un ISP conecta a sus usuarios a internet a través de diferentes tecnologías como DSL, Cablemódem, GSM y Dial-up.

L

- **LAN (Local Area Network – Red de área local)**: es una red de computadoras que conecta todos los dispositivos que se encuentran dentro de un área limitada. Las características que definen a una LAN es que incluyen un área geográfica pequeña y no es necesario la inclusión de líneas de telecomunicación arrendadas.

- **LDAP (Lightweight Directory Access Protocol – Protocolo ligero de acceso a directorios)**: es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

- **Log-files (archivo de registro)**: es un archivo que contiene una lista de eventos que han sido registrados por un ordenador. Los archivos de registro se generan a menudo durante las instalaciones de software, pero pueden utilizados para otros propósitos. La mayoría de los archivos de registro se guardan en texto plano, lo que minimiza el tamaño del archivo y permite que sean vistos en un editor de texto básico.

N

- **NAT (Network Address Translation – Traducción de dirección de red):** mecanismo utilizado por dispositivos de red para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles.

- **NCP (Network Control Protocol – Protocolo de control de red)**: es un protocolo de control del nivel de red que se ejecuta por encima de PPP. Se utiliza para negociar y configurar la red que va sobre PPP.

- **NVRAM (Non-volatile Random Access Memory – Memoria de acceso aleatorio no volatile)**: es un tipo de memoria de acceso aleatorio que no pierde la información almacenada al cortar la alimentación eléctrica.

O

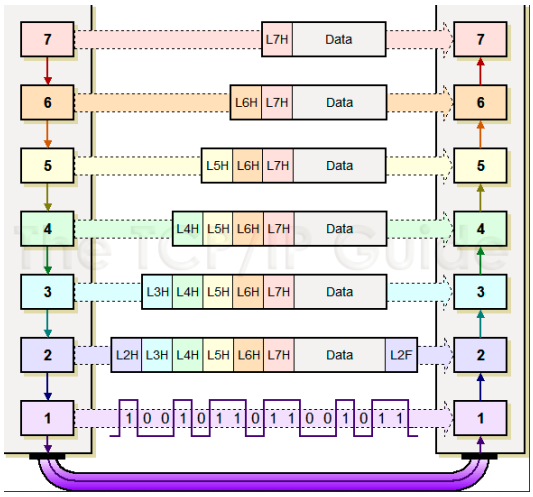
- **OSPF (Open Shortest Path First – Primera ruta más corta abierta)**: es un protocolo de enrutamiento jerárquico de pasarela interior que utiliza el algoritmo *SmoothWall Dijkstra* enlace-estado para calcular la ruta más corta posible, utilizando la métrica de menor costo. Además construye una base de datos enlace-estado idéntica en todos los enrutadores de la zona, puede operar con seguridad utilizando MD5 para autentificar a sus puntos antes de realizar nuevas rutas y antes de aceptar avisos de enlace-estado.

P

- **Passive polling**: término utilizado para la selección pasiva de un dispositivo dentro de un algoritmo de balanceo.

- **PC (Personal Computer – computadora personal)**: computadora diseñada para ser usada por una sola persona a la vez. Suele estar equipada para cumplir tareas comunes de la informática como navegar por internet, escribir textos y realizar trabajos de oficina entre otras.

- **PDU (Protocol Data Unit – Unidad de datos de protocolo)**: se utilizan para el intercambio de datos entre unidades disparejas, dentro de una capa de datos del modelo OSI.



PDU pasando por cada una de las capas del modelo OSI.

- **POST**: es uno de los métodos de los muchos métodos de petición que admite el protocolo HTTP. Este método está diseñado para solicitar que un servidor web acepte los datos adjuntos en el cuerpo del mensaje de solicitud de almacenamiento.

- **PPP (Point-to-point Protocol – Protocolo de punto a punto)**: es un protocolo de nivel de enlace estandarizado en el documento RFC 1661. Se utiliza comúnmente para establecer una conexión directa entre dos nodos de red.

- **PROM (Programmable read-only memory – memoria de solo lectura programable)**: Es una memoria digital donde el valor de cada bit depende del estado de un fusible, que puede ser quemado una vez. De esta manera la memoria puede ser programada una sola vez a través de un dispositivo especial.

- **Proxy**: se trata de un programa o dispositivo que realiza una acción en representación de otro. Su objetivo principal es interceptar las conexiones de red que un cliente hace a un servidor de destino, por motivos de seguridad y rendimiento.

- **PSTN (Public Switched Telephone Network – Red pública de telefonía conmutada)**:es un conjunto de elementos constituidos por todos los medios de transmisión y conmutación necesarios para enlazar a voluntad dos equipos terminales mediante un circuito físico que se establece específicamente para la comunicación y que desaparece una vez que se ha completado la misma. Se trata por tanto, de una red de telecomunicaciones conmutada.

R

- **RAM (Random Access Memory – Memoria de acceso aleatorio)**: memoria que es utilizada por la computadora como dispositivo de almacenamiento de trabajo para el sistema operativo, los programas y el software. Aquí es donde se cargan todas las instrucciones que ejecuta el procesador, además de otras unidades de cómputo. Se le denomina de *acceso aleatorio* porque se puede leer o escribir en una posición de memoria y no es necesario seguir un orden para acceder a la información de la manera más rápida posible.

- **ROM (Read-only memory – Memoria de solo lectura)**: es un medio de almacenamiento utilizado en ordenadores y dispositivos electrónicos, que permite sólo la lectura de la información y no de su escritura, independientemente de la presencia o no de una fuente de energía.

- **Router**: es un dispositivo que proporciona conectividad a nivel de red o nivel tres en el modelo OSI. Su función principal consiste en enviar o encaminar paquetes de datos de una red a otra.

- **RST**: es un bit que se encuentra en el campo del código en el protocolo TCP, y se utiliza para reiniciar la conexión.

S

- **Slot**: es un conector o puerto de expansión en la tarjeta madre de la computadora.

- **SSH (Secure Shell – intérpreta de órdenes segura)**: protocolo que permite el acceso a máquinas de manera remota a través de una red de forma segura. Permite el control completo de la terminal remota mediante un intérprete de comandos. Además de la conexión a otros dispositivos, permite copiar datos de forma segura, gestionar claves de RSA y transferir datos de forma segura a través de un canal seguro tunelizado.

- **Stand-by**: término utilizado para definir que un equipo electrónico se mantiene en espera.

- **STP (shielded twisted pair)**: cable de par trenzado similar al UTP con la diferencia de que cada par tiene una pantalla protectora, además de tener una lámina externa de aluminio de cobre trenzado diseñada para reducir la absorción de ruido eléctrico.

- **Streaming (lectura en continuo, difusión en flujo, descarga continua)**: es la distribución multimedia a través de una red de computadoras de manera que el usuario consume el producto, generalmente archivo de video o audio, en paralelo mientras se efectua su descarga. La palabra *streaming* se refiere a una corriente continua que fluye sin interrupción.

- **SYN**: es un bit de control dentro del segmento TCP, que se utiliza para sincronizar los números de secuencia iniciales ISN de una conexión en el procedimiento de establecimiento de tres fases (three way handshake).

T

- **TB (terabyte)**: es una unidad de almacenamiento de información cuyo valor equivale a 10^{12} bytes.

- **TCP/IP, (Transmission Control Protocol/Internet Protocol – Protocolo de control de transmisión/ Protocolo de internet)**: es un conjunto de protocolos de red en los que se basa internet y que permiten la transmisión de datos entre computadoras. El TCP/IP es la base de internet y sirve para enlazar las computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre rede de área local (LAN) y área extensa (WAN). Este protocolo se compone de los dos protocolos más importantes: (i) Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP), que fueron dos de los primeros protocolos en definirse y los más utilizados hoy en día.

- **Telnet**: es el nombre de un protocolo de red que nos permite viajar a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella. El puerto que utiliza este protocolo es el 23.

- **TFTP (Trivial File Transfer Protocol – Protocolo de transferencia de archivos trivial)**: es un protocolo de transferencia muy simple que se utiliza para transferir pequeños archivos entre ordenadores de una red.

- **TIA (Telecommunications Industry Association – Asociación de Industrias de Telecomunicaciones)**: es una asociación de comercio de los Estados Unicods que representa casi 600 empresas en el ramo de las telecomunicaciones.

- **TTL (Time to Live – Tiempo de vida)**: concepto usado en redes informáticas para indicar el número de enrutadores o nodos por los que pasó un paquete antes de ser descartado por un router o devuelto a su origen. Si un router recibe un paquete con un TTL igual a 1 o 0, éste no será enviado por ninguna de las interfaces del equipo, sino que enviará una notificación a la dirección IP de origen indicando que el destino se encuentra muy lejos y que se procederá a descartar el paquete.

U

- **UTP (unshielded twisted pair)**: cable de par trenzado que no se encuentra blindado y que se utiliza principalmente para comunicaciones.

W

- **WAN (World Area Network – Red de área mundial)**: es una red de computadoras que abarca varias ubicaciones físicas, proveyendo servicio a una zona, un país, o incluso varios continentes. Muchas WAN son construidas por organizaciones o empresas para le uso privado, algunas otras son instaladas por los proveedores de internet para dar servicio a sus clientes.

Trabajo realizado por

Emmanuel Eduardo Cuevas Aparicio