UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

"Implementación de Gestión Proactiva End-to-End para Servicios IP"

Trabajo de graduación presentado por:
Alvaro Xavier Arriola Díaz
para optar por el grado académico de
Licenciado en Ingeniería Electrónica

Guatemala
2013

Implementación de Gestión Proactiva End-to-End para Servicios IP

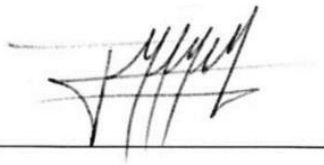
UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería
Departamento de Ingeniería Electrónica

"Implementación de Gestión Proactiva End-to-End para Servicios IP"

Trabajo de graduación presentado por:
Alvaro Xavier Arriola Díaz
para optar por el grado académico de
Licenciado en Ingeniería Electrónica

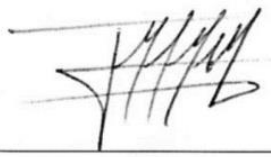
Guatemala
2013

Vo. Bo. :

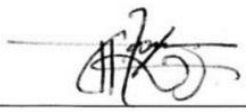
(f) 

Ing. José Manuel Morales Morales

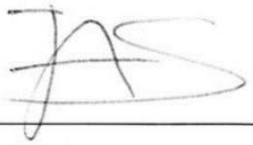
Tribunal Examinador:

(f) 

Ing. José Manuel Morales Morales

(f) 

Ing. Carlos Alberto Esquit Hernández

(f) 

Ing. José Augusto Sánchez Villanueva

Fecha de aprobación: Guatemala 5 de diciembre de 2013.

ÍNDICE

Página

LISTA DE FIGURAS

Página

LISTA DE CUADROS

Página

RESUMEN

El objetivo principal de este proyecto se centró en la implementación de un monitoreo para enlaces IP en la red de un carrier internacional. Dicha aplicación permite la proactividad ante los incidentes y refuerza el control sobre los eventos históricos en la red. El resultado que se obtuvo es un sistema capaz de alertar sobre un incidente a los operadores del centro de gestiones de la red y almacenar información sobre los enlaces para varios usos, entre ellos el monitoreo del rendimiento de ciertos parámetros particulares del servicio. Se realizaron configuraciones en equipo de ruteo del núcleo de la red de dicho proveedor mencionado para el monitoreo de los servicios IP-VPN seleccionados, asímismo programas en diferentes lenguajes para el sondeo de los diferentes datos de interés desde un servidor y desde el ordenador del operador. Al finalizar la realización de este trabajo se concluyó que la implementación del sistema agrega valor a los servicios, de esta manera aumentando la satisfacción del cliente. Se aplicó este monitoreo a más de 50 servicios y se desarrolló el sistema de manera que se facilita la adición de más servicios al mismo. También se destaca la importancia del hecho de que este resultado se concibió sin utilización de recursos económicos.

I. INTRODUCCIÓN

El siguiente trabajo tiene como objetivo demostrar la implementación de un sistema capaz de gestionar el estado de los servicios IP críticos en la red de un carrier, permitiendo la proactividad a los operadores en un Centro de Operaciones de Red, o NOC por sus siglas en inglés, ante un incidente, y aumentando la disponibilidad de la red. Este proyecto es capaz de garantizar la conectividad "End to End" de un enlace, en otras palabras, desde el punto inicial, donde es recibido por el cliente, hasta el final, donde se entrega, para cada servicio que le es configurado.

Estos enlaces, a los cuales se les implementó el monitoreo, son servicios IP-VPN. La detección del inconveniente en este tipo de servicios se logra por medio de la constante realización de pruebas sobre el mismo y generación de una alerta al detectar desconexión. También se deseó visualizar la información presente e histórica gráficamente como un detalle adicional.

Inicialmente se realizó investigación sobre los temas que se abarcan en el trabajo, con la finalidad de obtener una mejor comprensión del funcionamiento de los servicios, el monitoreo, la generación de alarmas y demás temas que se encontrarán detallados en la sección de Marco Teórico. A continuación se realizaron configuraciones de prueba para la observación del comportamiento de la red y el equipo del carrier con respecto a esta implementación. Luego se definió la organización, distribución y nomenclatura para las configuraciones. Se procedió con las configuraciones de los servicios y luego con la implementación del sistema servidor que recibiría las alarmas.

Es importante que el lector comprenda la importancia de la proactividad de una compañía con su cliente. El hecho de poder indicarle que existe un inconveniente, pero que se está al tanto de la situación, que el problema ya está siendo abordado. De esta manera el cliente puede sentirse seguro y confiar en su proveedor. Al mencionar estos puntos se puede notar la importancia y las ventajas de la implementación de un sistema de este tipo.

Al finalizar el trabajo fue posible confirmar que la utilización de un sistema con estas características resulta en una ventaja para una compañía proveedora de servicios de redes. Asimismo se puede concluir que esta afirmación no se limita a este monitoreo en específico, sino a cualquier implementación de gestión proactiva de servicios.

II. OBJETIVOS

A. Generales

Implementar un sistema de monitoreo end-to-end para servicios IP críticos que alerte a los operadores en un Centro de Operaciones de Red, o NOC por sus siglas en inglés, ante un incidente. De esta manera facilitando la proactividad ante los inconvenientes, aumentando la disponibilidad de la red y la satisfacción del cliente.

B. Específicos

1. Configuración de Real Time Performance Monitoring (RPM) en equipo Juniper de la red core del carrier para el monitoreo de los enlaces deseados.

2. Generación y envío de mensajes syslog relacionados con el monitoreo RPM, desde el equipo de la red llevando a cabo las pruebas de monitoreo hacia un ordenador.

3. Desarrollo de un programa servidor capaz de escuchar e interpretar mensajes enviados desde el equipo de la red, generando una alerta al operador debido a eventos de caídas en los servicios con dicho monitoreo, proporcionando una manera más eficiente e inmediata de reaccionar ante el inconveniente.

4. Integración de los resultados recolectados por el RPM a un servidor capaz de generar gráficas con los datos históricos y presentes para control de los servicios.

5. Demostración de la importancia y ventajas que ofrece la implementación de un sistema con estas características.

III. JUSTIFICACIÓN

El presente trabajo pretende permitir la proactividad del NOC ante un incidente en la red de un carrier, el cual presta servicios a compañías de alto prestigio en el mundo, y demostrar la importancia y ventajas de esto. El hecho de que estas empresas consideren a este carrier como proveedor en Latinoamérica, siendo algunos de estos Tier 1 ISP´s y existiendo una cantidad de empresas cada vez más alta en este negocio, nos indica que se debe aspirar a alcanzar un mayor nivel como compañía en estos negocios.

Al mantener el monitoreo de un enlace, el sistema puede estar atento al momento en el que se presenta una caída, y reaccionar inmediatamente, evitando que sea el cliente quien deba reportar el problema. De esta manera bajando drásticamente el MTTR, aumentando la disponibilidad de servicio que se proporciona y sobrepasar los acuerdos plasmados en el SLA respectivo. También permite comprobar la correcta operación del tramo por el cual se es responsable en el enlace de una manera inmediata. Adicionalmente, al ser un sistema desarrollado e implementado por la compañía misma, utilizando y aplicando características existentes en los equipos y plataformas, no se demuestra un gasto de recursos.

Al finalizar este trabajo se obtiene un sistema con la capacidad de indicar una falla al operador con una alerta, evitando la necesidad del monitoreo constante del personal. Es un sistema con alta escalabilidad, y se mantiene este documento como una referencia de la modalidad en la que funciona el mismo y la configuración para un servicio respectivo. Esto permite la fácil integración de monitoreo a un enlace nuevo en la red.

IV. MARCO TEÓRICO

A. MODELO OSI

Con la finalidad de entender como se logra la comunicación entre dos ordenadores a través del internet se debe comprender el funcionamiento del modelo de interconexión de sistemas abiertos, o modelo OSI. Este modelo tiene siete capas, y su finalidad es delegar ciertos aspectos de la comunicación a cada una de las capas, de esta manera el resto pueden desentenderse de esos detalles y concentrarse en su responsabilidad. Se podría decir que cada capa provee un servicio a su capa superior, hasta llegar a la capa de aplicación, que es con la cual el humano desea interactuar. En esta sección del Marco Teórico se describe y explica cada capa, comenzando con el 'inferior' en la jerarquía (físico) y proceder a la 'mejor' (la aplicación). Las capas se ordenan de esta forma:

- Física
- Enlace de datos
- Red
- Transporte
- Sesión
- Presentación
- Aplicación

1. Capa física. La capa física, la capa inferior del modelo OSI, se ocupa la transmisión y recepción de la secuencia de bits sin formato no estructurado a través de un medio físico. Describe las interfaces eléctricas ópticas, mecánicas y funcionales en el medio físico y lleva a cabo las señales para todos los niveles superiores (Blank, 2004). Proporciona los siguientes detalles:

a. La codificación de datos. Se modifica el modelo simple de señal digital (unos y ceros) utilizado por el equipo para adaptarse mejor a las características del medio físico y para ayudar en la sincronización de bits y el marco.

b. Determina estado de señal representa un binario 1. Como es que la estación receptora sabe cuando se inicia un "tiempo de bit", y cómo la estación receptora delimita un marco para el mismo.

c. Técnicas de transmisión para señales eléctricas u ópticas. Cuantos voltios/dB se deben utilizar para representar un determinado estado de la señal.

2. Capa de enlace de datos. Esta capa proporciona una transferencia sin errores de tramas de datos de un nodo a otro a través de la capa física, lo que permite a las capas por encima de él para asumirse

prácticamente libre de errores de transmisión sobre el vínculo. En esta capa se utilizan las direcciones Media Access Control, o MAC, también llamadas direcciones de hardware, las cuales están formadas por 48 bits que se suelen representar mediante dígitos hexadecimales que se agrupan en seis parejas, cada pareja se separa de otra mediante dos puntos ":" o mediante guiones "-", los primeros 24 bits identifican el fabricante de la tarjeta, y los siguientes diferencian cada una de las tarjetas. Por ejemplo, una dirección MAC podría ser F0:E1:D2:C3:B4:A5. En esta capa se encuentran protocolos tales como Ethernet y ATM. La capa de enlace de datos proporciona lo siguiente:

a. Vincula el establecimiento y terminación: Se establece y se termina vínculo lógico entre dos nodos.

b. Marco de control de tráfico: indica el nodo que transmite para "da marcha atrás" cuando no hay memorias intermedias de trama está disponible.

c. Secuencia de tramas: transmite y recibe tramas secuencialmente.

d. Confirmación de trama: proporciona y espera las confirmaciones de trama. Detecta y recupera de los errores que se producen en la física capa retransmitir marcos no reconocido y controlando Duplicar recibo de marco (Blank, 2004).

e. Delimitación de la trama: crea y reconoce los límites de la trama.

f. Comprobación de errores de trama: comprueba la integridad de las tramas recibidas.

g. Administración de acceso al medio: determina cuándo el nodo "tiene el derecho" a utilizar el medio físico.

3. Capa de red. La capa de red controla el funcionamiento de la subred, decidir qué los datos deben tomar basándose en las condiciones de red, la ruta de acceso física prioridad de servicio y otros factores. Esta capa utiliza, en su gran mayoría, el protocolo IP, el cual es explicado en mayor detalle más adelante. Esta capa proporciona:

a. Enrutamiento: dirige los paquetes entre redes.

b. Control de tráfico de subred: enrutadores (de las capas de red que intermedio sistemas) pueden indicar a una estación de envía "controlar" transmisión de su marco cuando se llena el búfer del enrutador.

c. Fragmentación del marco: si se determina que un enrutador de nivel inferior tamaño máximo de transmisión (MTU) de la unidad es menor que el tamaño de trama un enrutador puede fragmentar un paquete para la transmisión y volver a montarlas en la estación de destino.

d. Asignación de direcciones lógico / físico: traduce direcciones lógicas, o nombres en direcciones físicas.

e. Cuentas de uso de la subred: dispone de funciones de contabilidad para mantener pista de tramas reenviadas por sistemas intermedios de subred, para generar información de facturación.

El software de la capa de red debe crear encabezados para que la red software de la capa que residen en los sistemas intermedios de subred puede reconocerlos y utilizarlos para enrutar los datos a la dirección de destino (Blank, 2004).

Esta capa alivia las capas superiores de la necesidad de saber nada tecnologías de conmutación de acerca de la transmisión de datos e intermedio se utiliza para conectar los sistemas. Establece, mantiene y termina conexiones a través de las instalaciones de comunicaciones intermedios (uno o varios sistemas intermedios en la subred de comunicación).

En la capa de red y las capas inferiores, existen protocolos de igual entre un nodo y su vecino inmediato, pero el vecino puede ser un nodo a través del cual se enrutan los datos, no en la estación de destino. La las estaciones de origen y destino pueden estar separadas por medio de muchos sistemas, y por esta razón estos estos protocolos permiten la comunicación encapsulando los datos provenientes de tal manera que al ser recibidos por la contraparte se desencapsulen y sean entregados tal y como fueron enviados al siguiente sistema.

4. Capa de transporte. Esta capa garantiza que los mensajes se entregan sin errores, en la secuencia y sin pérdidas o duplicaciones. Alivia a los protocolos de nivel superior de cualquier problema con la transferencia de datos entre ellos y sus colegas. El tamaño y la complejidad de un protocolo de transporte depende del tipo de servicio que se requiere. Para un servicio confiable se utiliza el Transmission Control Protocol, o TCP, y para un servicio menos confiable, pero más rápido, se utiliza el User Datagram Protocol, o UDP. La capa de transporte proporciona:

a. La segmentación del mensaje: acepta un mensaje de la capa (sesión) por encima de él, se divide el mensaje en unidades más pequeñas (si no está pequeño suficiente) y pasa las unidades más pequeñas hacia abajo hasta la red capa. Permite volver a montar en la capa de transporte en la estación de destino el mensaje.

b. Mensaje de confirmación: proporciona mensajes end-to-end confiable entrega con las confirmaciones.

c. Control de tráfico de mensajes: indica a la estación de transmisión "da marcha atrás" cuando no hay búferes de mensajes está disponible.

d. Multiplexación de sesión: Multiplexa varias secuencias de mensajes, o las sesiones en un vínculo lógico y mantiene un seguimiento de los mensajes pertenecen a las sesiones, más adelante se explica la capa de sesión.

La información de encabezado de la capa de transporte, a continuación, debe incluir el control de la información, como inicio de mensaje y marcas de fin de mensaje, para permitir a la capa de transporte en el

otro extremo a reconocer los límites de mensajes. Además, si las capas inferiores no mantienen la secuencia, el encabezado de transporte debe contener información de secuencia para habilitar la capa de transporte en el extremo receptor para obtener de nuevo las piezas en el orden correcto antes de entregar el mensaje recibido a la siguiente capa (Blank, 2004).

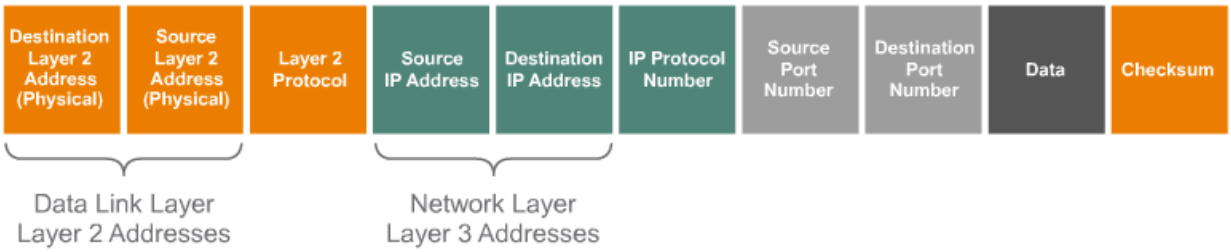
En esta capa se utilizan los puertos como origen y destino. Un puerto de red es una interfaz para comunicarse con un programa a través de una red. Un puerto suele estar numerado. La implementación del protocolo en el destino utilizará ese número para decidir a qué programa entregará los datos recibidos. Esta asignación de puertos permite que una máquina establecer simultáneamente diversas conexiones con máquinas distintas, ya que todos los paquetes que se reciben tienen la misma dirección, pero van dirigidos a puertos diferentes. Los números de puerto se indican mediante una palabra, o sea 2 bytes, por lo que existen del 0 al 65535 (Blank, 2004). Podemos usar cualquiera de ellos para cualquier protocolo, no obstante existe una organización Internet Assigned Numbers Authority, o IANA, encargado de la asignación de los mismos, el cual creó tres categorías (Internet Assigned Numbers Authority, 2013):

a. Los puertos inferiores al 1024. Estos son puertos reservados para el sistema operativo y usados por protocolos conocidos, si queremos usar uno de estos puertos tendremos que deshabilitar el servicio que los use, teniendo permisos de administrador.

b. Los comprendidos entre 1024 (0400 en hexadecimal) y 49151 (BFFF en hexadecimal). Estos son denominados como registrados y pueden ser usados por cualquier aplicación, existe una lista pública en la web del IANA donde ver que protocolo usa cada uno de ellos.

c. Los comprendidos entre los números 49152 (C000 en hexadecimal) y 65535 (FFFF en hexadecimal). Estos son denominados dinámicos o privados, porque son los usados por el sistema operativo cuando una aplicación tiene que conectarse a un servidor y por tanto necesita un puerto por donde salir.

Figura 1. Estructura de encapsulación de datos y encabezados de cada capa.



Fuente: Juniper Networks. (2010). Networking Fundamentals. Recuperado en enero de 2013, de Juniper Learning Portal: https://learningportal.juniper.net/juniper/user_activity_info.aspx?id=769

5. Capa de sesión. La capa de sesión permite el establecimiento de la sesión entre procesos que se ejecutan en diferentes estaciones. Proporciona:

a. Establecimiento de la sesión, mantenimiento y terminación. Permite dos procesos de aplicación en diferentes equipos para establecer, utilizar y terminar una conexión, llamada sesión.
b. Compatibilidad con la sesión: realiza las funciones que permiten a estos a comunicarse a través de la red, seguridad, de realizar los procesos el reconocimiento de nombre, el registro y así sucesivamente.

6. Capa de presentación. La capa de presentación da el formato a los datos que deberán presentarse a la capa de aplicación. Se puede ver como el traductor de la red. Esta capa puede traducir los datos de un formato utilizado por la capa de aplicación en un formato común en la estación emisora y después trasladar el formato común en un formato que la capa de aplicación en la estación receptora logrará comprender. La capa de presentación proporciona lo siguiente (Blank, 2004):

a. Traducción de los códigos de caracteres: por ejemplo, ASCII a EBCDIC.
b. Conversión de datos: bits de orden, punto CR-CR/LF, flotante entero, y así sucesivamente.
c. Compresión de datos: reduce el número de bits que es necesario transmitir en la red.
d. Cifrado de datos: cifrar los datos por motivos de seguridad. Por ejemplo, cifrado de contraseña.

7. Capa de aplicación. Esta capa describe como hacen su trabajo los programas de aplicación (navegadores, clientes de correo, terminales remotos, transferencia de ficheros, etc.). Esta capa implementa la operación con ficheros del sistema. Por un lado interactúan con la capa de presentación y por otro representan la interfaz con el usuario, entregándole la información y recibiendo los comandos que dirigen la comunicación. Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y servidor de ficheros (FTP). Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece sin parar.

Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente. Por ejemplo, un usuario no manda una petición "HTTP/1.0 GET index.html" para conseguir una página en html, ni lee directamente el código html/xml.

Al juntar todas las capas obtenemos la comunicación de ordenador a ordenador deseada, se puede ver en la Figura 1 como sería este proceso.

Figura 2. Comunicación entre dos ordenadores a través de una red utilizando el modelo OSI.

Fuente: Juniper Networks. (2010). Networking Fundamentals. Recuperado en enero de 2013, de Juniper Learning Portal: https://learningportal.juniper.net/juniper/user_activity_info.aspx?id=769

B. INTERNET PROTOCOL

El Protocolo de Internet (IP) es el método, o protocolo, por el cual se envían datos de un ordenador a otro a través de Internet. Cada equipo (conocido como host) en Internet tiene al menos una dirección IP pública que identifica de forma exclusiva de todos los demás equipos de la Internet. Hoy en día generalmente se utiliza la versión 4 de IP. Una está formada por 4 bytes, que se muestran en forma decimal, cada uno separado por un punto, por ejemplo, la IP 169.254.1.2. También cuenta con una máscara de red, la cual es un conjunto de bits cuya responsabilidad es delimitar que bits de la dirección IP forman parte de la red y cuales indican el host en específico en esa red.

La máscara también está formada por 32 bits y se puede expresar de diferentes maneras. Se puede expresar de la misma manera que una IP, por ejemplo, 255.255.255.0, o se puede indicar el número de bits de la máscara en forma decimal anteponiendo una diagonal: /24. Por lo tanto nuestra IP de ejemplo puede expresarse como 169.254.1.2 /24, o alternativamente como 169.254.1.2 255.255.255.0. La sección de la IP perteneciente a la red se obtiene realizando un AND lógico entre la IP y la máscara, esto nos da como resultado la red, y la sección eliminada nos indicaría el host.

Figura 3. Ejemplo de cálculo de la Red a partir de la IP y la máscara.

IP	192.168. 1. 8	11000000 10101000 00000001 00001000
Máscara	255.255.255. 0	11111111 11111111 11111111 00000000
Red	192.168. 1. 0	11000000 10101000 00000001 00000000

Fuente: Juniper Networks. (2010). Networking Fundamentals. Recuperado en enero de 2013, de Juniper Learning Portal: https://learningportal.juniper.net/juniper/user_activity_info.aspx?id=769

Anteriormente se mencionó que la dirección IP de un host es pública, se hizo énfasis en esto ya que también existen direcciones IP privadas. Se definen algunos bloques de direcciones en el RFC 1918, con la intención de ser utilizadas por entidades para sus redes privadas con la única intención de comunicarse entre sus miembros. Estos bloques son los siguientes: 10.0.0.0 – 10.255.255.255, 192.168.0.0 – 192.168.255.255, 172.16.0.0 – 172.31.255.255 y 169.254.0.0 – 169.254.255.255.

Estas direcciones privadas son utilizadas en empresas internacionales con cientos de equipos, así como también en un hogar con algunos cuantos hosts. Debe mencionarse que con la rapidez de crecimiento del internet, se agotaron rápidamente las direcciones IP, y es complicado conseguir una gran cantidad de IP públicas para uso propio. El RFC 1918 apoyo como una solución temporal a este problema. La solución a largo plazo sería IP versión 6, donde cada IP cuenta con 16 bytes, este tema está fuera del alcance de este documento al no estar relacionado con este trabajo.

Cuando se envía o recibe datos (por ejemplo, una nota de correo electrónico o una página web), el mensaje se divide en pequeños trozos llamados paquetes. Cada uno de estos paquetes contiene tanto la dirección de Internet del remitente y la dirección del receptor. Cabe mencionar que la acción de direccionar paquetes, desde su origen hasta su destino, basándose en direcciones IP, es realizado por ruteadores, en inglés routers, los cuales trabajan en la capa 3 del modelo OSI, discutido anteriormente. Cualquier paquete se envía primero a un ordenador de puerta de enlace, o Gateway, que comprende una pequeña parte del Internet. El sistema lee la dirección de destino y envía el paquete a una puerta de enlace adyacente, que a su vez lee la dirección de destino, y así sucesivamente a través de Internet hasta que una puerta de enlace reconoce el paquete como perteneciente a un ordenador dentro de su vecindad inmediata o dominio. Esa puerta de enlace reenvía el paquete directamente al ordenador cuya dirección se especifica.

Debido a que un mensaje se divide en una serie de paquetes, cada paquete puede, si es necesario, ser enviada por una ruta diferente a través de Internet. Los paquetes pueden llegar en un orden diferente que el orden en que fueron enviados, el Protocolo de Internet sólo les ofrece el transporte de origen a destino. Todo depende de otro protocolo, el Protocolo de Control de Transmisión (TCP), el cual trabaja en la capa

superior inmediata, para volver a ponerlos en el orden correcto y asegurar que se reciban todos los paquetes.

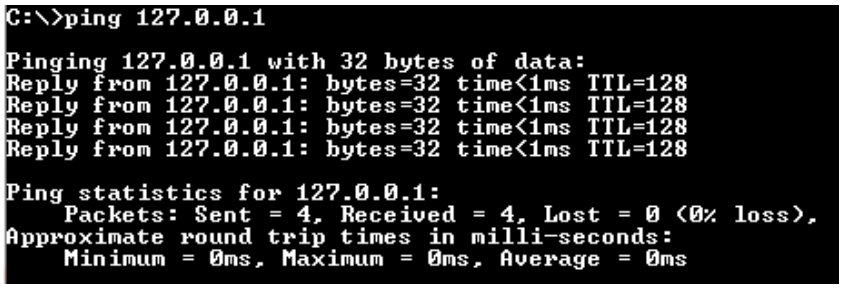
C. PRUEBAS DE CONECTIVIDAD IP

Todos los datos enviados a través de Internet se envían en paquetes. Los temas de IP fueron discutidos anteriormente. La pruebas de ping y trazado, o traceroute en inglés, son dos herramientas que se pueden utilizar para enviar paquetes de información a equipos remotos con el fin de recuperar la información. Estos programas son útiles para probar su conexión a Internet. Por ejemplo, podemos utilizar ping y traceroute desde la consola del sistema en Windows.

1. PING. Ping se puede utilizar para probar la velocidad de su conexión, la distancia al objetivo, y si su conexión está aún en funcionamiento. Nos indica cuánto tiempo un paquete de datos tarda en ir desde su ordenador a un servidor específico y viceversa. En este caso, el paquete es de 32 bytes de tamaño. Para utilizar Ping, se escribe ping seguido de un host de destino, puede ser el nombre del servidor, tales como www.google.com, o una dirección IP de host, por ejemplo, 209.166.161.121.

Al realizar el comando se debe recibir cuatro respuestas similares a las líneas de abajo. Esta prueba de ping comprueba el funcionamiento de la base de pila TCP/IP. Si TCP/IP funciona correctamente, no habrá problemas con el ping. Si recibe un mensaje de tiempo de espera o el error, hay un problema con TCP/IP, en cuyo caso puede que tenga que desinstalar y reinstalar TCP/IP (Expedient Holding, 2004). Al hacer una prueba de ping a la ip 127.0.0.1 comprobamos la conectividad de la computadora local.

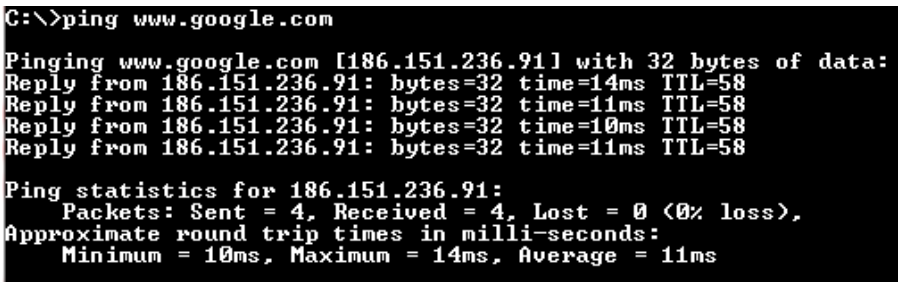
Figura 4. Resultado de prueba de ping desde la consola de windows.



También se puede realizar ping hacia una IP en internet para comprobar que la conexión se encuentre en buen estado. A continuación, también podemos introducir ping www.google.com u otro nombre de servidor. Esta prueba comprueba que el equipo es capaz de traducir las direcciones nombre, como www.google.com y www.valenciacf.com a las ip 186.151.236.91 y 212.111.96.96, la resolución de DNS.

Se debería recibir cuatro respuestas similares a las líneas de abajo. Si no se recibe respuesta, se debe comprobar la configuración del DNS (Expedient Holding, 2004).

Figura 5. Resultados de ping a www.google.com.



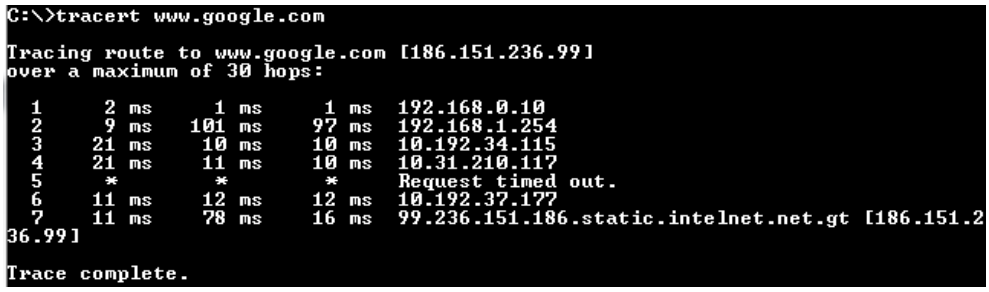
La siguiente línea muestra el nombre completo del huésped, tal como se encuentra por el programa ping. El número 186.151.236.91 es la dirección IP de la máquina. Las siguientes cuatro líneas debe mostrar o no si el host respondió, con la cantidad de bytes (tamaño del paquete), el tiempo de ida y vuelta (en milisegundos), y el time-to-live. Las últimas líneas muestran las estadísticas de ping al host. Estos incluyen el número de paquetes enviados, recibidos y perdidos. También se muestran los tiempos de ida y vuelta y promedios.

Si los 4 paquetes enviados se reciben, a continuación, la conexión está funcionando bien. Cualquier pérdida de paquetes puede indicar problemas de conexión lenta. Si no se reciben paquetes, verás como el que dicen algo así como "Host de destino inaccesible" o "Tiempo de espera agotado". Esto puede indicar que la conexión no es de enrutamiento correctamente. Por razones de seguridad, algunos servidores no permiten que usted los ping - obtendrá "Tiempo de espera agotado" los errores. Se deberían intentar varios servidores si se prueba su conexión.

Si no se puede realizar ping a un nombre de dominio, es decir, www.google.com, se debe intentar hacia la dirección IP, es decir 186.151.236.91. Si un ping a una dirección IP genera respuestas, pero el nombre de dominio no, entonces es probable que haya un problema con el DNS (Expedient Holding, 2004).

2. Traceroute. Es el trazado de saltos de la ruta que sigue un paquete desde su computadora a una dirección de destino. Un traceroute también muestra cómo muchas veces los paquetes están siendo retransmitidos por otros servidores hasta que llegue a su destino final. En Windows el comando es tracert.

Figura 6. Rastreo los saltos desde un ordenador a www.google.com.



La línea muestra el programa Ruta de seguimiento de la adquisición de la dirección IP del dominio. "Máximo de 30 saltos", es como muchos routers el paquete pasará a través de antes de renunciar a tratar de encontrar el host. Las siguientes líneas muestran cada servidor los paquetes recorrido para llegar al destino yahoo.com. Estos muestran tanto la dirección IP y el nombre de dominio de los servidores reales que los paquetes pasan a través.

Las trazas de ruta le permiten ver el camino los paquetes toman a través de Internet. A veces, también permitirán observar cómo la información viaja alrededor del mundo. Las trazas pueden mostrar dónde se produce una ruptura en su relación. Esto permite determinar exactamente donde los paquetes se están perdiendo, los paquetes descartados o perdidos en un traceroute por lo general se verán como asteriscos. Al igual que ping, algunos servidores no permiten ruta de seguimiento de todo el camino a ellos.

Ping y Traceroute le permiten diagnosticar problemas con su conexión a Internet. Estas utilidades le permiten determinar si un problema está en el equipo, fuera de la red, o en el servidor que está intentando alcanzar.

D. REAL-TIME PERFORMANCE MONITORING EN JUNIPER

El Real-time Performance Monitoring le permite supervisar el rendimiento de la red en tiempo real para evaluar y analizar la eficiencia de la red. Por lo general, el rendimiento de la red se evalúa en tiempo real basado en el jitter, retardo y pérdida de paquetes experimentado en la red. RPM es un servicio disponible en Junos OS que permite a un router medir parámetros tales como los retrasos de ida y vuelta y las peticiones de eco sin respuesta. Para lograr esto, RPM realiza intercambios de un conjunto de sondas con otros hosts IP en la red para fines de seguimiento y vigilancia de la red. Estas sondas son enviadas desde un nodo de origen a otros dispositivos de destino en la red que requieren seguimiento (Juniper Networks, 2010a).

Los datos tales como retardo de tránsito y la fluctuación de fase se pueden recoger a partir de estas sondas, y estos datos pueden ser utilizados para proporcionar una aproximación del retardo y jitter experimentado por el tráfico en tiempo real en la red. Los diferentes indicadores de tráfico en vivo como el tiempo de ida y vuelta, round-trip-time, o RTT, jitter salida positiva, jitter salida negativa, jitter entrada positiva, jitter entrada negativa, jitter de ida y vuelta positiva y negativa fluctuación de ida y vuelta se puede extraer de los resultados. RPM calcula, desviación estándar de pico a pico, y la suma de los cálculos de mínimo, máximo, promedio, para cada una de estas medidas. RPM sondas también se pueden utilizar para verificar la ruta de acceso entre los vecinos BGP.

RPM en su forma más simple puede ser descrito como tener a un cliente, o fuente, que envía consultas de sonda y un servidor, o destino, el cual responde. Durante una sonda, el dispositivo cliente envía un paquete a través de un servidor remoto, que a su vez devuelve el paquete con un acuse de recibo al remitente. Tanto el tipo y el contenido de las consultas enviadas desde el cliente son configurables por el usuario. Tanto los nodos de origen y de destino que ejecutan servicios de RPM son conscientes de que los paquetes en la sonda se utilizan para calcular la información, tales como RTT y la fluctuación de retardo.

Para cada sonda, RPM hace varias mediciones, como el por RTT. Para cada tipo de medida, RPM calcula el mínimo, máximo, promedio, pico a pico, la desviación estándar, y la suma global durante varias colecciones diferentes de mediciones. Por ejemplo, algunas de estas colecciones incluyen la corriente de prueba, la prueba más reciente, un "media móvil" de un número n de la mayoría de las sondas de los últimos, así como todas las pruebas realizadas (incluyendo el que está en progreso). La Figura 7 muestra un paquete de sondeo enviado desde el cliente y una respuesta correspondiente desde el servidor. La consulta de la sonda y las respuestas se producen entre fuente específica definida por el usuario y las direcciones de destino. El servicio de RPM es un proceso operativo Junos, rmopd, que se ejecuta en el routing engine (Juniper Networks, 2010a).

Figura 7. Operación cliente-servidor del RPM.

Fuente: Juniper Networks. (2010a). Real-time performance monitoring on juniper network devices. California: Juniper Networks, Inc.

Los diferentes tipos de paquetes que se pueden utilizar dentro de la sonda incluyen: Protocolo de mensajes de control de Internet (ICMP) eco ICMP de marca de tiempo, HTTP GET (no disponible para los servicios de BGP RPM), UDP eco, conexión TCP, UDP marca de tiempo. Icmp-ping es el tipo de sonda por defecto en los dispositivos con sistema operativo Junos.

En el caso de sondas de ICMP, uno de los escenarios implica el uso de sondas de RPM con un router no Juniper como un servidor. En este caso, el router no Juniper responde a las sondas ICMP RPM sin la adición de las marcas de tiempo del servidor y solamente el retardo RTT está calculado. Los principales elementos que se requieren para el servicio de RPM para que funcione correctamente son: La hora, la estructura RPM, configuración y comunicación de los resultados, la interfaz de destino (Juniper Networks, 2010a). Estas sondas son las que nos interesan en este trabajo, ya que el router en el punto final es propiedad del cliente.

Típicamente, un conjunto homogéneo de sondas, de ahora en adelante se referirán como probes en este documento, colectivamente llamado una prueba, de ahora en adelante test, se configura para cada dispositivo en la red, y la información obtenida de estas pruebas se almacena en las management information bases, o MIB, de ping y la MIB RPM. Se debe tener en cuenta que no se puede tener varios tipos de probes como parte de una sola prueba (Juniper Networks, 2010a). El objetivo sondeado se especifica mediante su dirección IPv4, y cada objetivo sondeado se controla durante el transcurso de un test que contiene múltiples probes. Durante un test, se generan probes y las respuestas recogidas a un nivel definido por el intervalo de la probe. Un probe se considera perdido si el intervalo de la probe expira antes de que se reciba una respuesta. El propietario del probe, o probe owner, y el nombre del test se exponen en forma conjunta en cada instancia de configuración RPM. Un test consiste en una gama de probes sobre el cual se calculan las métricas de rendimiento. El término test tiene dos contextos diferentes dentro de RPM. La primera se refiere al nombre de configuración en el que se combina con un nombre de propietario de identificar una instancia de configuración. En el segundo caso, denota un conjunto de enviado probes.

Los probes pueden ser uno de varios tipos, en este caso nos interesan las probes icmp-ping, las cuales envían requerimientos de echo icmp-ping al servidor. Una medición de RPM puede constar de varios tests, cada uno compuesto de un tipo de probe diferente e intercambiado entre un par dirección IP de origen-destino en particular. El intervalo entre los probes y los tests son configurables por el usuario, así como el contenido de la parte de los datos de la sonda. El usuario también puede controlar el número de probes que pertenecen a un test. Un algoritmo en el proceso de software determina el número de probes que se envían simultáneamente de un origen a un destino. Se pueden configurar umbrales, o thresholds en inglés, aceptables de los diferentes retrasos y la latencia, y es posible activar un fallo o alarma cada vez que este umbral se supera. Se puede generar un syslog o trap SNMP.

Para configurar RPM en un dispositivo Juniper se debe navegar a [edit services rpm] en la jerarquía. Se muestran los parámetros configurables para un test en la Figura 8. Para las pruebas icmp-ping nos interesa: data-size, probe-count, probe-interval, probe-type, routing-instance, target address, test-interval, thersholds y traps (Juniper Networks, 2010a).

Figura 8. Parámetros de configuración para un test y probe RPM.

```
owner owner {
    test test-name {
        data-fill data;
        data-size size;
        destination-interface interface-name;
        destination-port port;
        dscp-code-point dscp-bits;
        hardware-timestamp;
        history-size size;
        moving-average-size number;
        one-way-hardware-timestamp;
        probe-count count;
        probe-interval seconds;
        probe-type type;
        routing-instance instance-name;
        source-address address;
        target (url url | address address);
        test-interval interval;
        thresholds thresholds;
        traps traps;
    }
}
```

Fuente: Juniper Networks. (2013e). Configuring RPM Probes. Obtenido de Juniper technical documentation: http://www.juniper.net/techpubs/en_US/junos10.4/topics/task/configuration/rpm-probes-configuring.html

En la Figura DC podemos observar que tenemos la dirección IP destino 10.2.6.2, que es a la cual se enviarán los icmp-ping, al ser un tipo de probe icmp-ping. La configuración nos dice que se enviará un ping cada 5 segundos (probe-interval) hasta llegar a 10 pings enviados (probe-count) por cada test. Luego se esperará 60 segundos (test-interval) hasta iniciar un nuevo test. Los pings se enviarán desde el routing-instance R2. Si se llega al umbral de 10 pings fallidos se generará un trap de acuerdo.

Figura 9. Ejemplo de configuración RPM para prueba icmp-ping.

```
[edit services rpm probe prueba test test1]
aarriola@R3-SRX210HM# show
probe-type icmp-ping;
target address 192.168.0.1;
probe-count 10;
probe-interval 5;
test-interval 60;
routing-instance R2;
thresholds {
    total-loss 10;
}
traps test-failure;
```

Los resultados los podemos observar de dos maneras, utilizando dos comandos diferentes. Podemos observar un resumen de los resultados como en la Figura 10, o podemos observar los resultados por probe

como en la Figura 11, para manipular la cantidad de resultados de probes en este segundo podemos configurarlo con el parámetro history-size (Juniper Networks, 2010a).

Figura 10. Salida del comando show services rpm probe-results para el test en específico.

```
aarriola@R3-SRX210HM> show services rpm probe-results owner prueba test test1
    Owner: prueba, Test: test1
    Target address: 2.2.2.2, Probe type: icmp-ping
    Routing Instance Name: R2
    Test size: 10 probes
    Probe results:
      Response received, Sun Sep 15 00:51:02 2013, No hardware timestamps
      Rtt: 255 usec
    Results over current test:
      Probes sent: 10, Probes received: 10, Loss percentage: 0
      Measurement: Round trip time
        Samples: 10, Minimum: 241 usec, Maximum: 259 usec, Average: 253 usec, Peak to peak: 18 usec, Stddev: 6 usec,
        Sum: 2531 usec
    Results over last test:
      Probes sent: 10, Probes received: 10, Loss percentage: 0
      Test completed on Sun Sep 15 00:51:02 2013
      Measurement: Round trip time
        Samples: 10, Minimum: 241 usec, Maximum: 259 usec, Average: 253 usec, Peak to peak: 18 usec, Stddev: 6 usec,
        Sum: 2531 usec
    Results over all tests:
      Probes sent: 20, Probes received: 20, Loss percentage: 0
      Measurement: Round trip time
        Samples: 20, Minimum: 241 usec, Maximum: 312 usec, Average: 259 usec, Peak to peak: 71 usec, Stddev: 18 usec,
        Sum: 5185 usec
```

Figura 11. Salida del comando show services rpm history-results para el test en específico.

```
aarriola@R3-SRX210HM> show services rpm history-results owner prueba test test1
    Owner, Test                 Probe received              Round trip time
    prueba, test1               Sun Sep 15 00:48:30 2013             312 usec
    prueba, test1               Sun Sep 15 00:48:35 2013             309 usec
    prueba, test1               Sun Sep 15 00:48:40 2013             252 usec
    prueba, test1               Sun Sep 15 00:48:46 2013             243 usec
    prueba, test1               Sun Sep 15 00:48:51 2013             254 usec
    prueba, test1               Sun Sep 15 00:48:56 2013             253 usec
    prueba, test1               Sun Sep 15 00:49:01 2013             256 usec
    prueba, test1               Sun Sep 15 00:49:06 2013             257 usec
    prueba, test1               Sun Sep 15 00:49:11 2013             257 usec
    prueba, test1               Sun Sep 15 00:49:16 2013             261 usec
    prueba, test1               Sun Sep 15 00:50:16 2013             246 usec
    prueba, test1               Sun Sep 15 00:50:21 2013             253 usec
    prueba, test1               Sun Sep 15 00:50:26 2013             241 usec
    prueba, test1               Sun Sep 15 00:50:32 2013             250 usec
    prueba, test1               Sun Sep 15 00:50:37 2013             258 usec
    prueba, test1               Sun Sep 15 00:50:42 2013             256 usec
    prueba, test1               Sun Sep 15 00:50:47 2013             259 usec
    prueba, test1               Sun Sep 15 00:50:52 2013             259 usec
    prueba, test1               Sun Sep 15 00:50:57 2013             254 usec
    prueba, test1               Sun Sep 15 00:51:02 2013             255 usec
```

También se generan syslogs al completar un test exitosamente, al fallar un probe y al fallar un test. En el caso de la Figura se generó una configuración de un archivo en el que se guardaran los syslogs con respecto a las pruebas rpm, el log se le llamo rpm_ping. El tema de syslog se discutirá a detalle más adelante.

Figura 12. Muestra de syslogs generados al completar un test exitosamente.

```
aarriola@R3-SRX210HM> show log rpm_ping | match prueba,
Sep 15 00:49:16  R3-SRX210HM rmopd[1000]: PING_TEST_COMPLETED: pingCtlOwnerIndex = prueba, pingCtlTestName = test1
Sep 15 00:51:02  R3-SRX210HM rmopd[1000]: PING_TEST_COMPLETED: pingCtlOwnerIndex = prueba, pingCtlTestName = test1
Sep 15 00:52:48  R3-SRX210HM rmopd[1000]: PING_TEST_COMPLETED: pingCtlOwnerIndex = prueba, pingCtlTestName = test1
Sep 15 00:54:34  R3-SRX210HM rmopd[1000]: PING_TEST_COMPLETED: pingCtlOwnerIndex = prueba, pingCtlTestName = test1
```

E. SYSLOG

Syslog es un estándar de facto para el envío de mensajes de registro en una red informática IP. Por syslog se conoce tanto al protocolo de red como a la aplicación o biblioteca que envía los mensajes de registro. Un mensaje de registro suele tener información sobre la seguridad del sistema, aunque puede contener cualquier información. Junto con cada mensaje se incluye la fecha y hora del envío. A continuación se describen algunos detalles de este protocolo (Clemm, 2006).

1. Usos. Es útil registrar, por ejemplo: Un intento de acceso con contraseña equivocada, un acceso correcto al sistema. Anomalías como variaciones en el funcionamiento normal del sistema. Alertas cuando ocurre alguna condición especial Información sobre las actividades del sistema operativo. Errores del hardware o el software. También es posible registrar el funcionamiento normal de los programas, por ejemplo, guardar cada acceso que se hace a un servidor web, aunque esto suele estar separado del resto de alertas.

2. Protocolo. El protocolo syslog es muy sencillo: existe un ordenador servidor ejecutando el servidor de syslog, conocido como syslogd (demonio de syslog). El cliente envía un pequeño mensaje de texto (de menos de 1024 bytes). Los mensajes de syslog se suelen enviar vía UDP, por el puerto 514, en formato de texto plano. Su sencillez ha hecho que muchos dispositivos lo implementen, tanto para enviar como para recibir. Eso hace posible integrar mensajes de varios tipos de sistemas en un solo repositorio central.

3. Estructura del mensaje. El mensaje enviado se compone de tres campos: Prioridad, cabecera y texto. Entre todos no han de sumar más de 1024 bytes, pero no hay longitud mínima.

4. Prioridad. La prioridad es un número de 8 bits que indica tanto el recurso (tipo de dispositivo que ha generado el mensaje) como la severidad (importancia del mensaje), números de 5 y 3 bits respectivamente. Los códigos de recurso y severidad los decide libremente la aplicación, pero se suele seguir una convención para que clientes y servidores se entiendan.

5. Códigos de recurso. Los códigos de recurso se pueden observar en el Cuadro 1.

Cuadro 1. Códigos observados en varios sistemas.

0	Mensajes del kernel	12	Subsistema de NTP
1	Mensajes del nivel de usuario	13	Inspección del registro
2	Sistema de correo	14	Alerta sobre el registro
3	Demonios de sistema	15	Demonio de reloj
4	Seguridad/Autorización	16	Uso local 0
5	Mensajes generados internamente por syslogd	17	Uso local 1
6	Subsistema de impresión	18	Uso local 2
7	Subsistema de noticias sobre la red	19	Uso local 3
8	Subsistema UUCP	20	Uso local 4
9	Demonio de reloj	21	Uso local 5
10	Seguridad/Autorización	22	Uso local 6
11	Demonio de FTP	23	Uso local 7

Fuente: RFC 3164

6. Códigos de severidad. Los 3 bits menos significativos del campo prioridad dan 8 posibles grados.

Cuadro 2. Diferentes códigos de severidad.

0	***Emergencia*****:** el sistema está inutilizable
1	***Alerta*****:** se debe actuar inmediatamente
2	***Crítico*****:** condiciones críticas
3	***Error*****:** condiciones de error
4	***Peligro*****:** condiciones de peligro
5	***Aviso*****:** normal, pero condiciones notables
6	***Información*****:** mensajes informativos
7	***Depuración*****:** mensajes de bajo nivel

Fuente: RFC 3164.

7. Cálculo de la prioridad. Para conocer la prioridad final de un mensaje, se aplica la siguiente fórmula: Prioridad = Recurso * 8 + Severidad. Por ejemplo, un mensaje de kernel (Recurso=0) con Severidad=0 (emergencia), tendría Prioridad igual a 0*8+0 = 0. Uno de FTP (11) de tipo información (6) tendría 11*8+6=94. Como se puede ver, valores más bajos indican mayor prioridad.

8. Cabecera. El segundo campo de un mensaje syslog, la cabecera, indica tanto el tiempo como el nombre del ordenador que emite el mensaje. Esto se escribe en codificación ASCII (7 bits), por tanto es texto legible. El primer campo, tiempo, se escribe en formato Mmm dd hh:mm:ss, donde Mmm son las iniciales del nombre del mes en inglés, dd, es el día del mes, y el resto es la hora. No se indica el año. Justo después viene el nombre de ordenador, hostname, o la dirección IP si no se conoce el nombre. No puede contener espacios, ya que este campo acaba cuando se encuentra el siguiente espacio.

9. Texto. Lo que queda de paquete syslog al llenar la prioridad y la cabecera es el propio texto del mensaje. Éste incluirá información sobre el proceso que ha generado el aviso, normalmente al principio (en los primeros 32 caracteres) y acabado por un carácter no alfanumérico (como un espacio, ":" o "["). Después, viene el contenido real del mensaje, sin ningún carácter especial para marcar el final.

F. PROTOCOLOS DE RUTEO

El propósito de los protocolos de enrutamiento es aprender las rutas disponibles que existen en la red de la empresa, construir tablas de enrutamiento y hacer decisiones de enrutamiento. Algunos de los protocolos de enrutamiento más comunes incluyen OSPF, IS-IS y BGP. Hay dos tipos de protocolos de enrutamiento primarios aunque existen muchos protocolos de enrutamiento diferentes definidos a esos dos tipos. Protocolos de estado de enlace, en inglés link state, y de vector de distancia, en inglés distance vector, son los tipos principales.

Los protocolos de vector de distancia anuncian su tabla de enrutamiento para todos los vecinos directamente conectados a intervalos frecuentes y regulares utilizando una gran cantidad del ancho de banda y son lentos para converger (Juniper Networks, 2009). Cuando una ruta está disponible, todas las tablas de ruteo deben actualizarse con la nueva información. El problema yace en que cada router debe tener que anunciar la nueva información a sus vecinos, y entonces se necesita mucho tiempo para que todos los routers logren una visión exacta actual de la red. Los protocolos de vector de distancia utilizan máscaras de subred de longitud fija que no son escalables. Un ejemplo sería el Routing Information Protocol, o RIP, el cual se utilizaría en redes muy pequeñas.

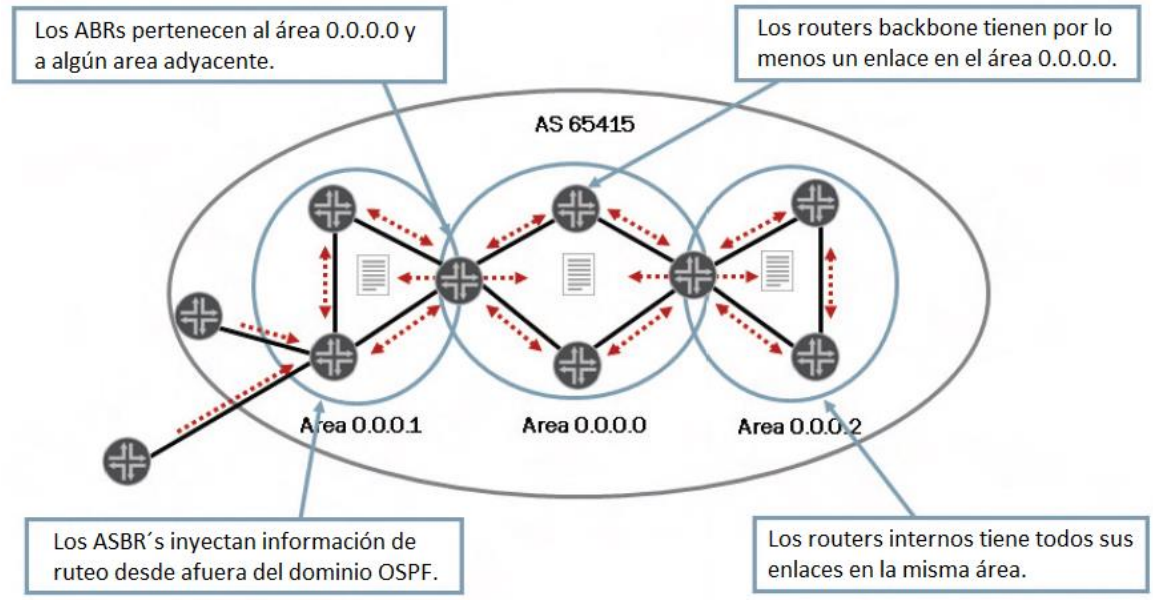
Por otro lado, los protocolos de estado de enlace anuncian actualizaciones de enrutamiento sólo cuando se producen, así utilizan el ancho de banda de una manera más eficiente. Los enrutadores no anuncian la tabla de enrutamiento entera, esto hace que se dé una convergencia más rápida. El protocolo de enrutamiento inundará la red con anuncios de estado de enlace a todos los routers vecinos por área en un intento de converger la red con la nueva información de ruta. El cambio incremental es todo lo que se anuncia a todos los routers como Link State Advertisements, o LSA´s, utilizados en OSPF. Usan máscaras de subred de longitud variable, que son escalables y utilizan el direccionamiento de manera más eficiente (Juniper Networks, 2009). Al ser el objetivo de este trabajo un implementación en un Internet Service Provider, o ISP, solamente nos concentraremos en los protocolos más utilizados en este ámbito escalables. A continuación se discuten brevemente dos de los protocolos más utilizados: OSPF y BGP.

G. OPEN SHORTEST PATH FIRST

El protocolo OSPF es un protocolo de estado de enlace, en inglés link state, desarrollado como un estándar abierto para el encaminamiento de IP a través de las grandes redes de múltiples proveedores. Un protocolo de estado de enlace enviará anuncios de estado de enlace a todos los vecinos conectados de la misma zona para comunicar información de la ruteo (Soricelli, JNCIA study guide, 2006).

1. Áreas. OSPF utiliza una jerarquía de áreas asignadas que se conectan a una red troncal principal de routers. Cada área se define por uno o más enrutadores que han establecido las adyacencias. Se tiene definida el área 0, como a principal, también llamada backbone, ya que todas las rutas deben de transitar por ella. Todas las áreas poseen una conexión directa al backbone. Los routers que se encuentran en el área 0 se les llama backbone routers. Los routers que poseen una o más interfaces en el backbone, y adicionalmente una o más interfaces en otra área se las llama Area Border Routers, o ABR. A los routers que contienen todas sus interfaces en una sola área se les denomina routers internos. También existen ruteadores que conectan hacia otro dominio que no utiliza OSPF e inyecta información de ruteo externa, a estos se les llama Autonomous System Border Routers, o ASBR. También existen los vínculos virtuales, los cuales se utilizan para agregar un área que no está directamente conectada al backbone (Soricelli, JNCIA study guide, 2006). En la Figura 2 se puede observar un ejemplo de conexión entre áreas y los respectivos tipos de routers.

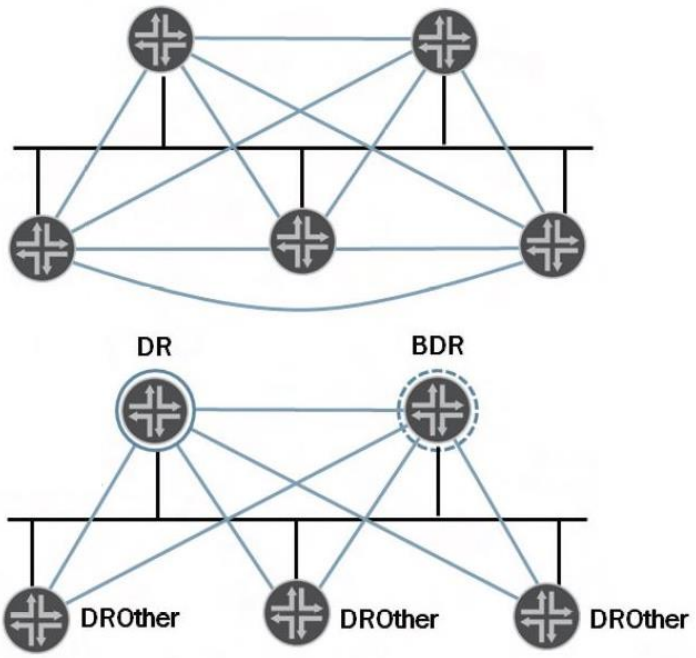
Figura 13. Ejemplo de interconexión entre áreas OSPF y los tipos de routers.



Fuente: Juniper Networks. (2013b). Open shortest path first. En JNCIS-SP study guide part 1 (págs. 51-82). California: Juniper Networks, Inc.

2. Adyacencias. Cada router habilitado para OSPF, al inicializarse, enviará paquetes de saludo a todos los routers OSPF conectados directamente, también llamados paquetes hello. Los paquetes hello contienen información como temporizadores de router, el ID del router y la máscara de subred. Si los routers están de acuerdo con la información, se convertirán en vecinos OSPF. Una vez que los routers se convierten en vecinos establecen adyacencias mediante el intercambio de bases de datos de estado de enlace, en inglés link state databases. Los routers con enlaces de punto a punto y enlaces punto a multipunto establecen adyacencias automáticamente. En un segmento de red se pueden llegar a tener muchas adyacencias OSPF, y resultaría impráctico mantener una relación por cada pareja de ruteadores. Por esta razón se establece un router designado, en inglés designated router (DR), para cada segmento de red, este se escoge en base a atributos de preferencia establecidos. De esta forma cada router en el segmento de red mantiene relación solamente con el DR, también existe un backup designated router, en caso que el principal falle. Cualquier router OSPF que no sea DR o BDR se designa como DRother (Soricelli, JNCIA study guide, 2006). Un diagrama de las adyacencias se puede observar en la Figura 3.

Figura 14. Comparación de adyacencias entre routers OSPF en un segmento de red.



Fuente: Juniper Networks. (2013b). Open shortest path first. En JNCIS-SP study guide part 1 (págs. 51-82). California: Juniper Networks, Inc.

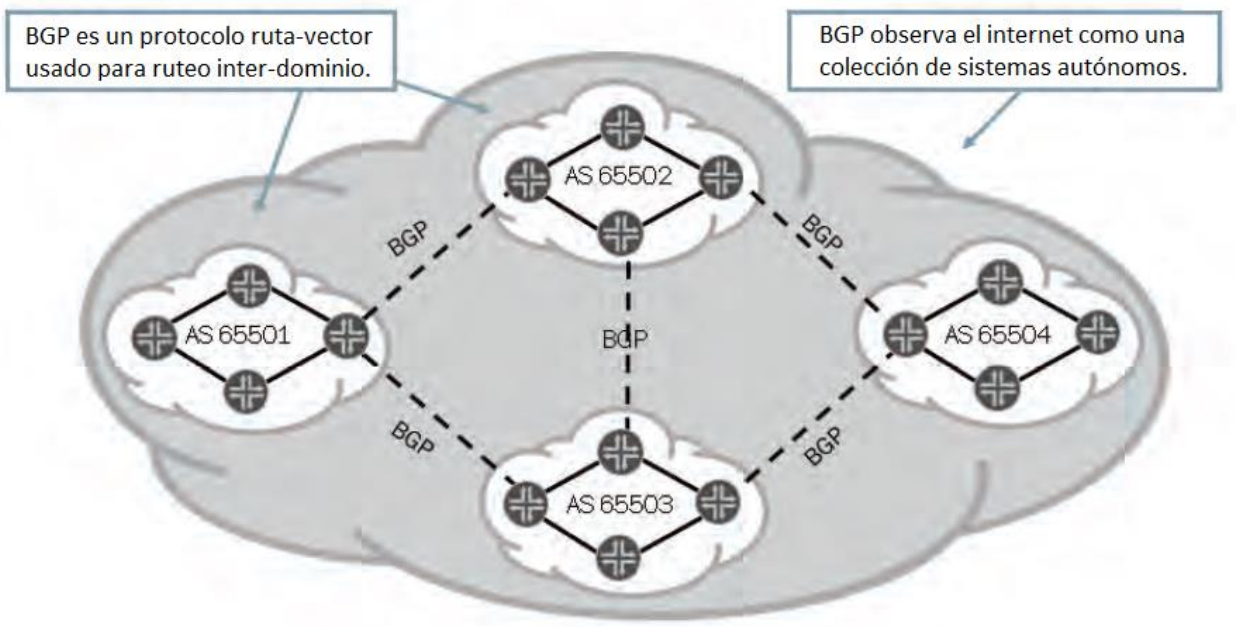
3. Link-state advertisements. Los routers OSPF utilizan anuncios de estado de enlace (LSA) para inundar fiable información acerca de sus enlaces de red y el estado de los vínculos con sus vecinos. Como la información de estado de enlace es compartido entre los routers OSPF, cada router OSPF crea y mantiene un estado de enlace (o topológica) base de datos. Los routers OSPF utilizan la información facilitada en los LSA como entrada para el algoritmo de ruta más corta, SPF por sus siglas en inglés, para calcular la mejor ruta para cada prefijo de destino. Los LSA's solamente se envían al suceder un cambio en la red. Adicionalmente se anuncian periódicamente para evitar que las rutas sean desechadas por los demás ruteadores, pero esto ocurre cada 30 a 45 minutos aproximadamente, según los parámetros establecidos (Soricelli, JNCIA study guide, 2006). Existen diferentes tipos de LSA's según el origen de la ruta, el router que la anuncia y el área en la que se inunda, este tema se encuentra fuera del alcance de este documento.

4. Convergencia. La rápida convergencia se logra utilizando el algoritmo SPF (Dijkstra) que determina un camino más corto desde el origen al destino. La tabla de enrutamiento se construye la ejecución de SPF que determina todas las rutas de los enrutadores vecinos. Dado que cada router OSPF tiene una copia de la base de datos de topología y la tabla de enrutamiento para su área en particular, cualquier cambio de ruta se detecta más rápido que con los protocolos de vector de distancia y también se determinan rutas alternas (Soricelli, JNCIA study guide, 2006).

H. BORDER GATEWAY PROTOCOL

Border Gateway Protocol, o BGP, es un protocolo de enrutamiento entre sistemas autónomos, AS por sus siglas en inglés y se refiere a veces como protocolo de enrutamiento ruta-vector, en inglés path-vector, ya que utiliza un atributo llamado AS path, usado como un vector, para evitar bucles de enrutamiento entre dominios. El término path-vector, en relación con BGP, significa que la información de enrutamiento BGP incluye una serie de números de AS, la cual indica el camino que una ruta lleva a través de la red. Aunque BGP se utiliza principalmente para enrutamiento inter-AS, BGP se utiliza también en grandes redes para VPN's basadas en MPLS, discutidas más adelante, y se utiliza para separar grandes dominios de OSPF, explicado anteriormente. BGP es mucho más escalable y ofrece una mayor cantidad de control a través de políticas que un IGP, la cual es una de las mayores razones de su uso en un ISP. Este protocolo intercambia información de enrutamiento entre AS's. Un AS es un conjunto de routers que funcionan bajo la misma administración. La información de ruteo de BGP incluye la ruta completa a cada destino. BGP utiliza la información de enrutamiento para mantener una base de información de la capa de información de accesibilidad de la red, NLRI por sus siglas en inglés, que intercambia con otros sistemas BGP (Soricelli, JNCIA study guide, 2006).

Figura 15. Ejemplo de topología de red que utiliza BGP.



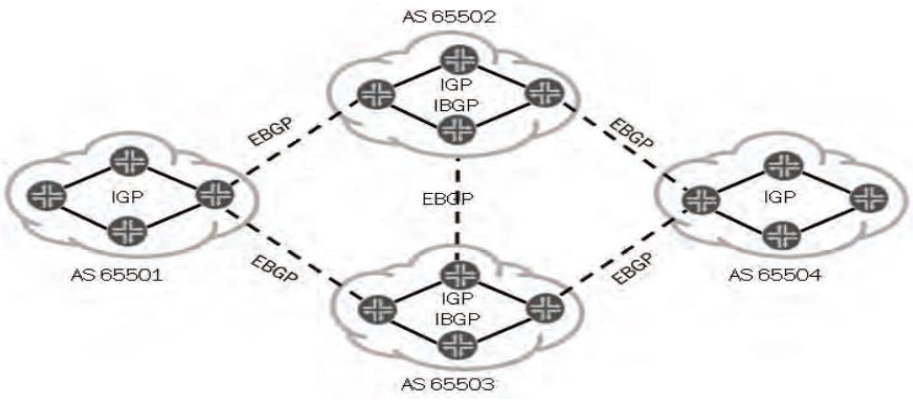
Fuente: Juniper Networks. (2013a). Border gateway protocol. En JNCIS-SP study guide part 1 (págs. 83-110). California: Juniper Networks, Inc.

1. Sesión BGP. Las sesiones BGP utilizan el protocolo TCP, discutido anteriormente, para la comunicación. Esto lo vuelve más confiable. BGP es compatible con dos tipos diferentes de intercambio de información de enrutamiento. Los intercambios entre AS se conocen como BGP externo, o sesiones EBGP y manejan enrutamiento inter-AS. Los intercambios dentro de un AS se conoce como BGP interno, o sesiones IBGP, y manejan enrutamiento intra-AS.

a. BGP externo. Una conexión EBGP existe entre un dispositivo en un AS y otro dispositivo en un AS distinto. La conexión entre los dos AS consiste en una conexión física y una conexión BGP. La conexión física es una de enlace de datos compartida entre los dos AS's. La conexión BGP existe entre dispositivos que hablan BGP en cada uno de los AS. Esta sesión puede comunicar destinos que se puede llegar a través de la publicidad AS. Por defecto se establece la conexión EBGP entre los dispositivos conectados inmediatamente ubicados en dos AS's diferentes, porque el time to live, o TTL por sus siglas en inglés, de los paquetes EBGP es igual a 1 (Soricelli, JNCIA study guide, 2006). Si un interlocutor EBGP se encuentra a más de un salto de distancia, se debe establecer una sesión BGP multi-hop, este tema está fuera del alcance de este trabajo.

b. BGP interno. Una conexión IBGP se establece normalmente entre las interfaces loopback de los routers, ya que no necesariamente están conectados adyacentemente, depende de la topología de la AS. BGP utiliza las interfaces loopback para la estabilidad por la razón de que estas interfaces siempre están arriba, a menos que el propio router muera. Debido a la conexión IBGP normalmente existe entre los routers conectados remotamente, existe la exigencia de que exista un IGP en el AS, el cual provee las rutas necesarias para alcanzar cada loopback, ya que la sesión TCP del BGP se establece mediante tablas de enrutamiento regulares (Soricelli, JNCIA study guide, 2006). Se puede considerar que BGP se configura por encima de un protocolo IGP, como OSPF, por ejemplo.

Figura 16. Topología BGP, indicando donde se utiliza BGP interno y externo.



Fuente: Juniper Networks. (2013a). Border gateway protocol. En JNCIS-SP study guide part 1 (págs. 83-110). California: Juniper Networks, Inc.

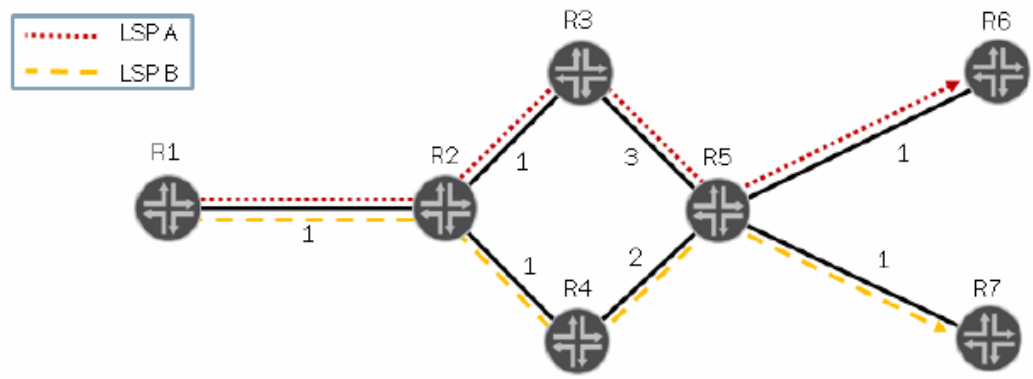
I. MULTI-PROTOCOL LABEL SWITCHING

Multi-Protocol Label Switching (MPLS) fue presentado originalmente como una forma de mejorar la velocidad de envío de los routers, pero ahora se está convirtiendo en una tecnología estándar crucial que ofrece nuevas capacidades para las redes de IP a gran escala. La ingeniería de tráfico, la capacidad de los operadores de red para dictar el camino que toma el tráfico a través de su red, y la red privada virtual de apoyo son dos ejemplos de las aplicaciones clave en las que MPLS es superior a cualquier tecnología IP disponibles en la actualidad (Soricelli, Tutorial: introduction to MPLS, 2003).

Aunque MPLS fue concebido como independiente de la capa 2, gran parte de la emoción generada por MPLS gira en torno a su promesa de proporcionar un medio más eficaz de la implementación de redes IP a través de redes troncales WAN basadas en ATM. MPLS es visto por algunos como uno de los desarrollos de las redes más importantes de la década de 1990.

La esencia de MPLS es la generación de una etiqueta corta de longitud fija que actúa como una representación abreviada de cabecera de un paquete IP. Se trata de la misma manera como un código postal es la abreviatura de la casa, la calle y la ciudad en una dirección postal, y se utiliza esa etiqueta para tomar decisiones sobre el reenvío de paquetes. Paquetes IP tienen un campo en su cabecera que contiene la dirección a la que el paquete se va a enrutar. Las redes enrutadas tradicionales procesan esta información en todos los routers en la ruta del paquete a través de la red, también llamado hop by hop routing. Estos caminos por los cuales se enrutan los paquetes son llamados label switched paths, o LSP's (Soricelli, Tutorial: introduction to MPLS, 2003).

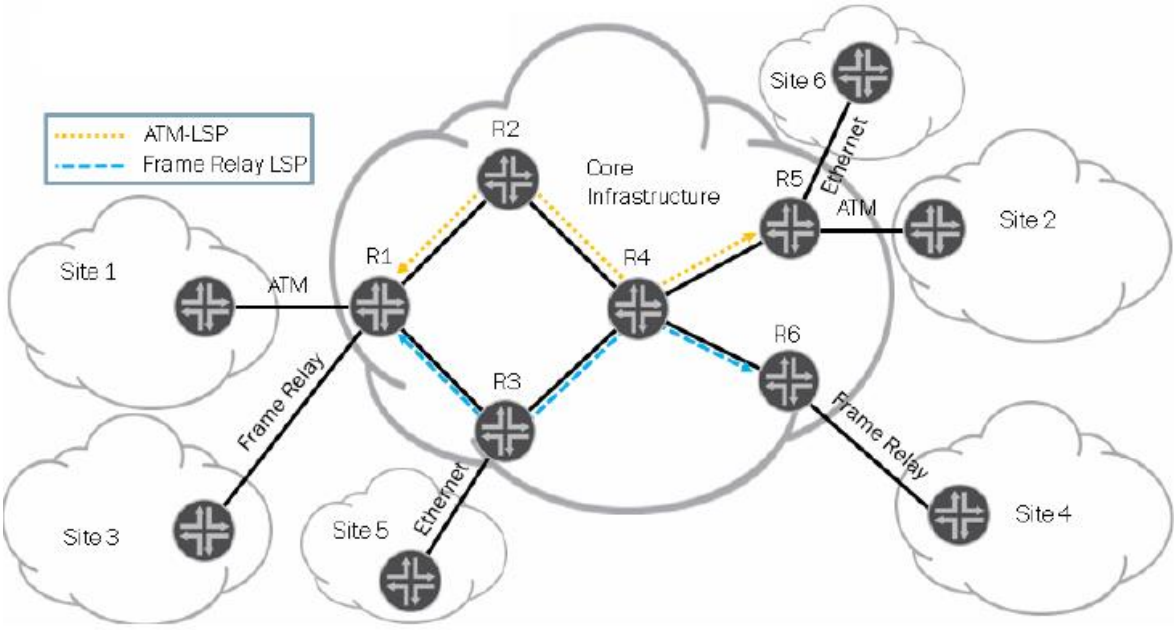
Figura 17. Ejemplo de trayectorias de LSP´s en una red MPLS.



Fuente: Juniper Networks. (2013d). MPLS fundamentals. En JNCIS-SP study guide part 3 (págs. 9-38). California: Juniper Networks, Inc.

En la terminología de MPLS, los nodos de manipulación de paquetes o routers se llaman Label Switched Routers, o LSR's. La derivación de la expresión debe ser obvia, los enrutadores MPLS reenvían paquetes tomando decisiones de conmutación basados en la etiqueta MPLS. Esto pone de manifiesto otro de los conceptos clave en MPLS. Los routers IP convencionales contienen tablas de enrutamiento en las que se realiza una búsqueda, utilizando con la cabecera IP de un paquete, para decidir cómo enviar ese paquete. Estas tablas son construidas por los protocolos de enrutamiento IP, como OSPF por ejemplo, que llevan información de accesibilidad IP en forma de direcciones IP. En la práctica, nos encontramos con que la búsqueda en la cabecera IP y el plano de control, osea generación de las tablas de enrutamiento, están estrechamente unidos. Ya que el reenvío de MPLS se basa en las etiquetas, es posible separar el plano de reenvío basado en etiquetas del plano de control de protocolo de enrutamiento. Al separar los dos, cada uno se puede modificar de forma independiente. Con esta separación, no es necesario cambiar el mecanismo de reenvío, por ejemplo, para la migración de una nueva estrategia de enrutamiento a la red. Se pueden transportar varios servicios utilizando la misma infraestructura de reenvío. (Soricelli, Tutorial: introduction to MPLS, 2003)

Figura 18. Diferentes tecnologías sobre la misma infraestructura MPLS.



Fuente: Juniper Networks. (2013d). MPLS fundamentals. En JNCIS-SP study guide part 3 (págs. 9-38). California: Juniper Networks, Inc.

El reenvío de paquetes en MPLS funciona de la siguiente manera. En el borde de la red se requiere clasificadores de paquetes de alto rendimiento que puedan aplicar y eliminar, las etiquetas necesarias: llamamos a estos routers de borde MPLS, label edge routers, o LER. En MPLS, los paquetes IP se

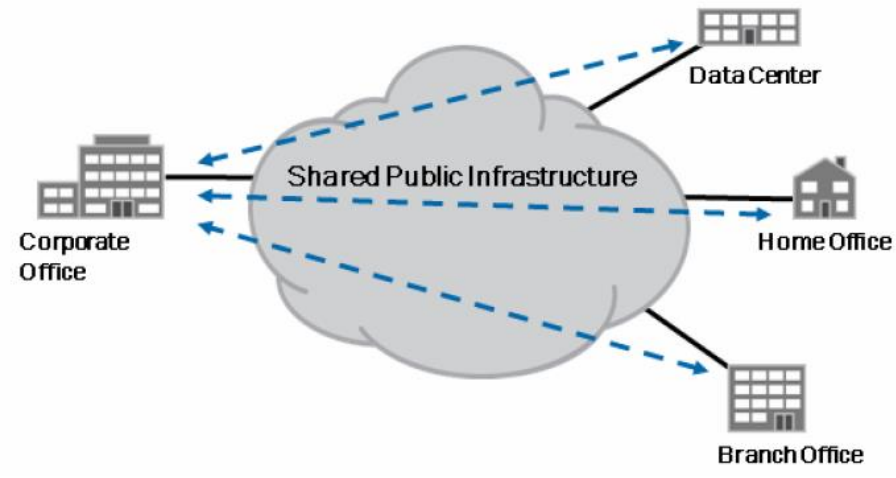
encapsulan con estas etiquetas por el LER, el cual analiza el contenido de la cabecera IP y selecciona una etiqueta apropiada con la que se encapsulará el paquete. Los LSR's en el core de la red MPLS tienen que ser capaces de procesar los paquetes etiquetados a anchos de banda extremadamente altos, a esos se les llama routers de tránsito. Parte de la gran potencia de MPLS proviene del hecho de que, en contraste con el encaminamiento IP convencional, este análisis puede basarse en algo más que la dirección de destino transportada en la cabecera IP. En todos los nodos posteriores dentro de la red la etiqueta MPLS, y no la cabecera IP, se utiliza para hacer la decisión de reenvío para el paquete. Por último, cuando los paquetes etiquetados dejan la red MPLS, otro router de borde elimina las etiquetas. En este caso no necesariamente debe ser el último router en remover la etiqueta, también se puede delegar este trabajo al penúltimo router en el LSP, a esto se le llama penultimate-hop popping, o PHP (Soricelli, Tutorial: introduction to MPLS, 2003).

J. VIRTUAL PRIVATE NETWORK

Las redes privadas virtuales, o VPN por sus siglas en inglés, son redes privadas que utilizan una red pública para conectar dos o más sitios remotos. En lugar de conexiones dedicadas entre redes, las VPN's utilizan conexiones virtuales enrutadas, o sea túneles, a través de redes públicas que suelen ser redes de proveedores de servicios. Las VPN's son una alternativa rentable a las líneas dedicadas, las cuales resultan en un mayor gasto de recursos. El tipo de VPN está determinado por las conexiones que utiliza, y si la red del cliente o la red del proveedor realiza el túnel virtual (Juniper Networks, 2010b).

Una red privada virtual se compone de dos áreas: topológicas de la red del proveedor y la red del cliente. Red del cliente se encuentra comúnmente en varios sitios físicos y también es privada. Un sitio del cliente consistirá típicamente en un grupo de enrutadores u otros equipos de red situado en una sola ubicación física. La red del proveedor, que se ejecuta a través de la infraestructura de Internet, consiste en routers que proporcionan servicios de VPN a la red del cliente, así como los routers que proporcionan otros servicios. La red del proveedor comunica las diferentes instalaciones de los clientes (Juniper Networks, 2010b).

Figura 19. Diagrama ejemplo de una VPN.



Fuente: Juniper Networks. (2013). VPN review. En JNCIS-SP study guide part 3 (págs. 157-170). California: Juniper Networks, Inc.

Para asegurarse de que se mantienen las VPN's privadas y aisladas de otras VPNs y del internet público, la red del proveedor mantiene políticas que mantienen la información de enrutamiento de las diferentes VPN's para cada uno de sus clientes separadas. Un proveedor puede dar servicio a múltiples VPN's, siempre y cuando sus políticas siguen rutas de diferentes VPN's separadas. Del mismo modo, el sitio del cliente puede pertenecer a múltiples VPN's, siempre y cuando mantenga las rutas de las diferentes VPN's separadas. Existen distintos tipos de implementación VPN's, existen de capa 3, capa 2, utilizando MPLS, vpls, etc. En este trabajo nos interesan las VPN's Capa 3 que utilizan MPLS y BGP, las cuales se detallan a continuación (Juniper Networks, 2010b).

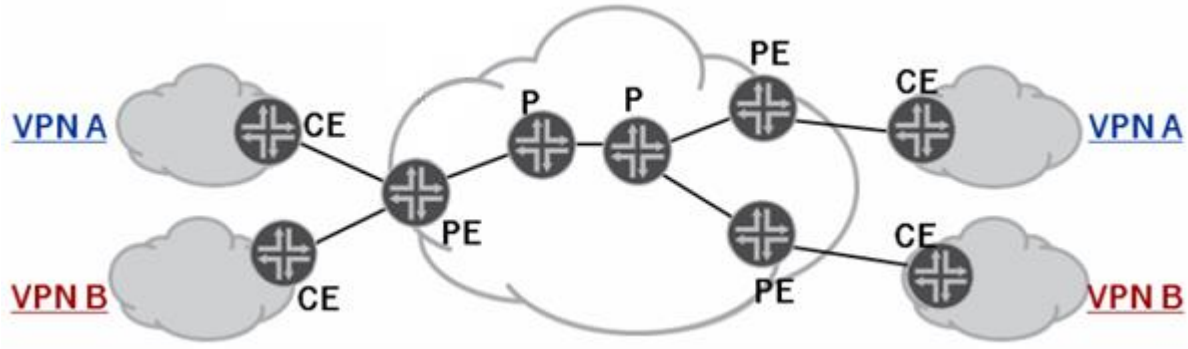
K. VPN BASADA EN BGP/MPLS.

Una VPN MPLS opera en la capa 3 del modelo OSI. Se basa en el RFC 4364. La VPN se compone de un conjunto de sitios que están conectados a través de una red pública de un ISP. Los sitios comparten información de enrutamiento común y la conectividad de los sitios está controlada por un conjunto de políticas (Juniper Networks, 2013c).

En una VPN de este tipo, el enrutamiento se produce en los routers del proveedor de servicios. Los routers del proveedor enrutan y reenvían el tráfico de la VPN hacia adelante en los puntos de entrada y salida de la red de tránsito. La red de proveedores de servicios deben aprender las direcciones IP de los dispositivos que envían tráfico a través de la VPN y las rutas deben ser anunciadas y filtradas en toda la red del proveedor. Como resultado, la VPN de capa 3 requieren información acerca de rutas de los clientes y una configuración más extensa de enrutamiento de VPN routing and forwarding, o VRF, a comparación de

una VPN capa 2. Un VRF es una instancia de ruteo, o routing instance en inglés, el cual es una colección de interfaces, tablas de ruteo y parámetros de protocolos de ruteo. El conjunto de interfaces pertenece a las tablas de ruteo, y los parámetros de los protocolos controlan la información en dichas tablas. Pueden existir múltiples tablas de ruteo para cada routing instance (Juniper Networks, 2013c).

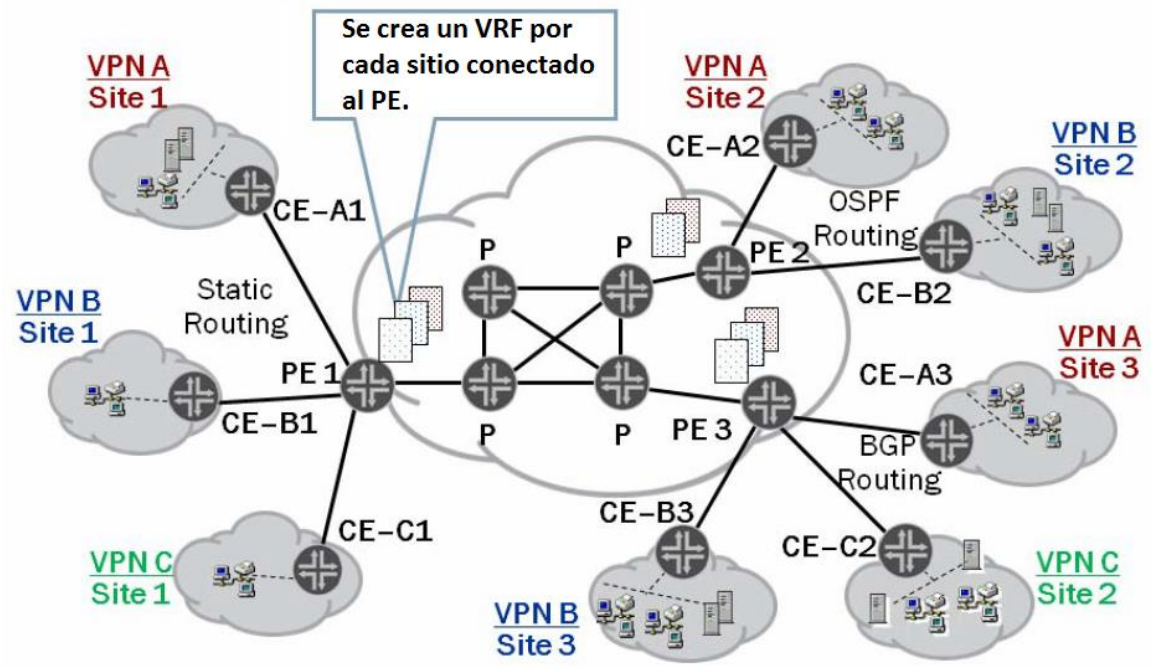
Figura 20. Esquema simplificado de una VPN.



Fuente: Juniper Networks. (2013c). Layer 3 VPNs. En JNCIS-SP study guide part 3 (págs. 171-192). California: Juniper Networks, Inc.

Los routers situados en el borde la red del proveedor se les denomina routers PE, o provider edge. Estos se interconectan por un lado a los routers del lado del cliente, denominados CE, o customer edge, y del otro lado hacia los routers del núcleo del proveedor, denomiados P. Los routers PE mantienen un VRF por cada sitio al que se interconectan, esto permite la utilización de redes IP privadas sin que se traslapen con las utilizadas por otros clientes. Los routers PE y CE funcionan como compañeros de enrutamiento, con el router PE dando por concluido el intercambio de enrutamiento entre el sitio del cliente y el núcleo del proveedor, en esta conexión se puede utilizar cualquier protocolo de ruteo. Las rutas aprendidas de los routers de la CE, se almacena en la tabla VRF del PE router y se envían a los routers PE remotos utilizando Multiprotocol Border Gateway Protocol, o MP-BGP. Los routers PE utilizan LSP's para reenviar el tráfico VPN entre los sitios de los clientes (Juniper Networks, 2010b).

Figura 21. Ejemplo de esquema completo de servicios VPN capa 3.



Fuente: Juniper Networks. (2013c). Layer 3 VPNs. En JNCIS-SP study guide part 3 (págs. 171-192). California: Juniper Networks, Inc.

Debido a que una red de tránsito típico está configurado para manejar más de una VPN, los PE pueden tener varias instancias VRF configuradas. Como resultado, dependiendo del origen del tráfico y cualquier regla de filtrado aplicada al tráfico, las tablas de enrutamiento BGP pueden contener múltiples rutas para una dirección de destino en particular. Debido a que BGP requiere exactamente una ruta BGP por destino puede importar a la tabla de reenvío, BGP debe de tener alguna forma de distinguir en su NLRI los mensajes recibidos desde diferentes VPNs (Juniper Networks, 2010b).

Un distinguidor de ruta, o route distinguisher en inglés, es un número único que identifica localmente toda la información de las rutas de una VPN particular. Identificadores numéricos únicos BGP permiten distinguir entre rutas que por lo demás serían idénticas. Cada VRF que se configura en un router PE debe tener un route distinguisher único. Existen dos formatos posibles (Juniper Networks, 2010b):

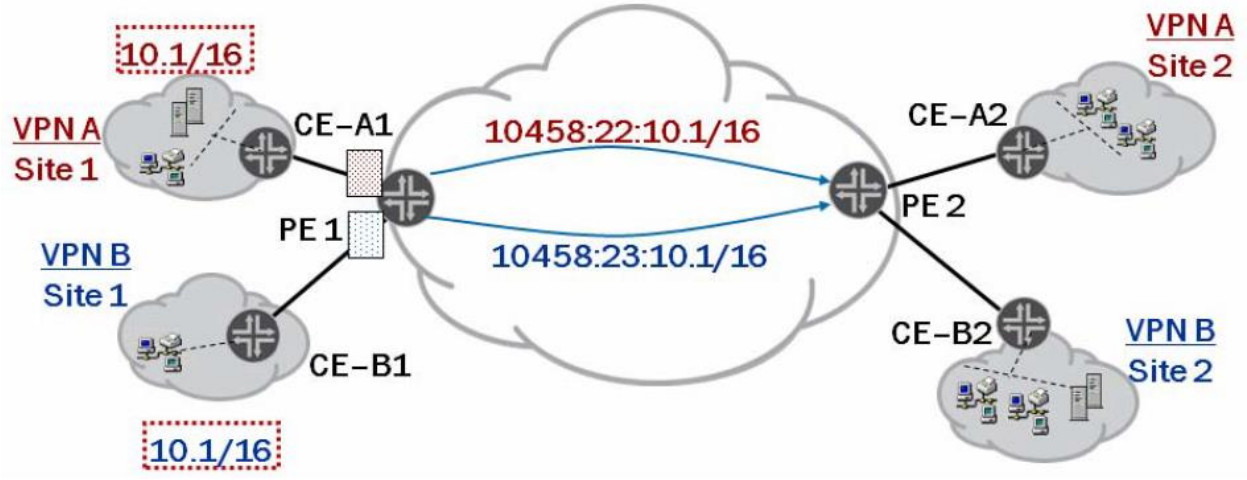
a. Número de AS:número, donde el número de AS es un sistema autónomo, discutido anteriormente, y el número es cualquier valor de 4 bytes. Se recomienda utilizar un numero autorizado por la IANA, y no privados, preferiblemente el del ISP o el AS del cliente.

b. Dirección IP:número, donde es una dirección IP normal, y el número es cualquier valor de 2 bytes. La dirección IP puede ser cualquier dirección unicast global único. Le recomendamos que utilice la

dirección que se configure en la cuenta de router-id, que es una dirección IP pública en su rango de prefijo asignado.

La destino de ruta, o route target en inglés, define qué ruta es parte de una VPN. Un route target único ayuda a distinguir entre los diferentes servicios de VPN en el mismo router. Cada VPN también tiene una política que define cómo se importan las rutas en la tabla VRF en el router. Una VPN capa 3 utiliza un route target única para distinguir entre las rutas de VPN. El router PE entonces exporta la ruta en sesiones IBGP a los otros routers del proveedor. La exportación de ruta se rige por cualquier política de ruteo que se ha aplicado a la tabla VRF en particular. Para propagar las rutas a través de la red de proveedores, el enrutador PE también debe convertir la ruta al formato de VPN, que incluye al route distinguisher, a este formato se le llama dirección VPN-IPv4 (Juniper Networks, 2010b).

Figura 22. Rutas idénticas en A y B de PE1 al PE2 no se traslapan por el route distinguisher.



Fuente: Juniper Networks. (2013c). Layer 3 VPNs. En JNCIS-SP study guide part 3 (págs. 171-192). California: Juniper Networks, Inc.

Cuando el router PE 2, observado en la Figura 22, recibe la ruta, despoja el route distinguisher de la ruta y anuncia la ruta al router CE conectado, normalmente a través de anuncios ruta estándar BGP IPv4.

L. LA IMPORTANCIA DE LA PROACTIVIDAD CON EL CLIENTE

La proactividad es una actitud en la que el sujeto u organización asume el pleno control de su conducta de modo activo, lo que implica la toma de iniciativa en el desarrollo de acciones creativas y audaces para generar mejoras, haciendo prevalecer la libertad de elección sobre las circunstancias del contexto. La proactividad no significa sólo tomar la iniciativa, sino asumir la responsabilidad de hacer que las cosas sucedan, decidir en cada momento lo que queremos hacer y cómo lo vamos a hacer.

El concepto opuesto es el de reactividad o tomar una actitud pasiva y ser sujeto de las circunstancias y por ende, de los problemas. A nivel empresarial tiene mucha importancia, es adelantarse y ofrecer no solo lo que te pide el cliente, sino ofrecer más, ir más allá utilizando los recursos, y experiencias profesionales para darles a los clientes el valor añadido de que no tengan ellos que pensar por uno, sino que se les de ya lo que buscan o quieren.

Si un cliente tiene una crisis por ejemplo en lo que a ventas se refiere, no has de esperar a que el cliente, por ejemplo, si trabajas en una consultoría de comunicación indique que necesita que se posicionen sus productos en los medios porque las ventas están bajando, sino que se debe adelantar y enviar un plan de comunicación con distintas proposiciones para que las ventas aumenten. Esta proactividad se convierte en un valor añadido para el consultor, justo lo que el cliente necesita.

Es cierto, que ser proactivos tanto a la hora de aplicar este valor a nivel personal como profesional no es fácil. Es algo que se ha de ir interiorizando, que se ha de ir gestando y que poco a poco la persona que así lo hace, consigue ser proactiva de manera casi automática y sobre todo eficaz.

La proactividad a nivel profesional, se considera ser más difícil que la proactividad en la vida personal, dado que aunque se tenga esa aptitud, la de ser proactivo. Pero el secreto de conseguir la proactividad a nivel profesional, está en siempre querer dar lo mejor al cliente. Quizás no se conoce, como en el ejemplo anteriormente mencionado, que las ventas están cayendo, pero sí que la competencia está sacando productos que ponen en peligro las ventas de tu cliente, y entonces se debe poner en marcha un plan de cómo revalorizar o poder dar un valor añadido a los productos que vende el cliente. En este caso, por el interés y seguimiento que se hace del cliente o de su entorno, competencia, al final se está siendo proactivo de igual modo, se buscan soluciones para que el cliente se posicione siempre en lo más alto, en nuestro caso, brindar el mejor servicio posible, manteniendo el tiempo promedio para una reparación bajo, y de esta manera manteniendo la disponibilidad de su enlace alta (Jiménez-Zarco & Torrent-Sellens, 2009).

M. IMPORTANCIA DEL MONITOREO DE UNA RED

El monitoreo de la red es la función de recoger información de la gestión de red. Las aplicaciones de monitorización de una red se crean para recopilar datos para las aplicaciones de gestión de red. El propósito del monitoreo de la red es la recogida de información útil a partir de varias partes de la red para que la red puede ser gestionada y controlada utilizando la información recolectada. La mayoría de los dispositivos de red están ubicados en lugares remotos. Estos dispositivos por lo general no se conectan directamente a terminales de manera que la aplicación de gestión de red pueda controlar sus estados fácilmente. Por lo

tanto, se han desarrollado técnicas de monitoreo de la red para permitir que las aplicaciones de gestión de red puedan comprobar los estados de sus dispositivos de red. Como se usan cada vez más dispositivos de red para construir redes más grandes, las técnicas de monitoreo de red se han ampliado a las redes de monitoreo en conjunto (Stallings, 1996).

A medida que más personas se comunican mediante redes, las redes se han convertido más grandes y más complejas. La utilización del Internet ha aumentado al ritmo de expansión de la red. En esta era de redes grandes y complejas, las aplicaciones del monitoreo de la red deben utilizar medios eficaces para verificar el estado de sus redes para que las aplicaciones de gestión de red pueden controlar plenamente sus redes y proporcionar servicios de redes económicas y de alta calidad a los usuarios. Es muy importante saber cuáles son los objetivos a alcanzar en el monitoreo de la red. Al conocer los objetivos del monitoreo de la red , la aplicación de monitorización puede elegir entre las técnicas de monitoreo de red que mejor ayudarán a monitorear las redes (Stallings, 1996).

En general, existen tres objetivos básicos para la supervisión de la red: monitoreo del rendimiento, el monitoreo de fallos y el monitoreo de cuenta (Stallings, 1996).

Estos objetivos son tres de las cinco áreas funcionales de gestión de red propuestos por OSI, Open Systems Interconnect. Las otras dos áreas funcionales no están relacionadas con la supervisión de la red. Ellos son la gestión de la configuración y gestión de la seguridad.

La supervisión del rendimiento trata de medir el rendimiento de la red. Hay tres aspectos importantes en el monitoreo del rendimiento. En primer lugar, la información de seguimiento de los resultados se suele utilizar para planificar la expansión futura de la red y localizar los problemas actuales de su uso. En segundo lugar, el marco de tiempo de seguimiento de los resultados debe ser suficiente para establecer un modelo de comportamiento de la red. En tercer lugar, la elección de qué medida es importante. Hay demasiadas variables medibles en una red. Pero la lista de elementos a medir debe ser significativo y rentable. Esta lista de elementos a medir se llama indicadores de red, ya que indican los atributos de la misma (Stallings, 1996).

Cuadro 3. Descripciones de indicadores de red.

Indicadores de red	**Descripción**
Disponibilidad de circuito	El tiempo verdadero que un usuario puede utilizar una red y la conexión de la red está disponible para el usuario.
Disponibilidad de nodo	El tiempo verdadero que un usuario puede usar nodos de la red, multiplexores y routers sin tener errores.
Factor de bloqueo	El número de usuarios que no puede accesar la red por estar ocupada.
Tiempo de respuesta	El tiempo para trasmitir una señal y recibir una respuesta.

Fuente: Stallings, W. (1996). SNMP, SNMPv2, and RMON: practical network management (2da ed.). Michigan: Addison-Wesley Pub. Co.

El monitoreo de fallos se ocupa de la medición de los problemas en la red. Hay dos cuestiones importantes en la supervisión de fallas. En primer lugar, el monitoreo de fallos trata con las diversas capas de la red. Cuando se produce un problema, puede ser en diferentes capas de la red. Por ello, es importante saber en qué capa se está teniendo problemas. En segundo lugar, la supervisión de fallos requiere el establecimiento de una serie de características normales de la red en un período prolongado de tiempo (Stallings, 1996).

Siempre hay errores en la red, pero cuando los hay, esto no significa que la red está teniendo problemas persistentes. Algunos de estos errores se espera que se produzca. Por ejempl , el ruido en un enlace de red puede causar errores de transmisión. La red sólo tiene un problema cuando el número de errores de repente se ha incrementado por encima de su comportamiento normal. Por lo tanto, un registro de la conducta normal es importante (Stallings, 1996).

El monitoreo de cuenta ocupa cómo los usuarios utilizan la red. La red mantiene un registro de lo que los dispositivos de la red son utilizados por los usuarios y la frecuencia con que se utilizan. Este tipo de información se utiliza para la facturación de usuario para el uso de la red, y, lo más importante, para predecir el futuro uso de la red. A continuación se indican brevemente algunos puntos que demuestran la importancia del monitoreo de una red (Downing, 2013).

1. Fiabilidad. El monitoreo de la red hace un seguimiento de los aparatos de misión crítica y de software, y notifica al administrador de red antes de cuestiones se vuelven fáciles de impactar problemas. Por ejemplo, el monitoreo de red le permite saber si un servidor falla, si el servicio deja de responder o si usted está en peligro de quedarse sin espacio en disco. Esto garantiza un enfoque proactivo para hacer frente a los problemas, en lugar de esperar a que los usuarios finales se encuentren con un problema.

2. Conocimiento. los sistemas de monitoreo de red avisará al administrador de TI de los problemas de rendimiento y eventos de fallo mediante el envío de una serie de alertas a las computadoras, localizadores o dispositivos móviles. Esto permite que el administrador de TI ser consciente de los problemas, independientemente de dónde se encuentre.

3. Capacidad. Tener un conocimiento profundo de cómo se están utilizando sus dispositivos le permite identificar de manera proactiva las áreas que requieren más espacio en disco y desplegar capacidad adicional de una manera controlada.

4. Solución de problemas. Con supervisión de la red se puede identificar rápidamente el dispositivo que está causando el problema, lo que limita el tiempo de inactividad y la pérdida de tiempo tratar de diagnosticar el problema. En lugar de esperar a que un usuario final informe de un problema y la solución del mismo, el control de red permite al equipo de apoyo la detección, diagnóstico y solución del problema antes de que los usuarios sean conscientes de ello.

5. Seguimiento de las tendencias. Los problemas que se producen intermitentemente o en determinadas horas punta pueden ser complicados de encontrar, pero los informes de supervisión de red en curso permitirán comprender las principales tendencias en el rendimiento y la salud general de la red.

6. Plan de mejoras y cambios. Si un dispositivo está funcionando constantemente cerca de su límite, este nos puede indicar que es el momento de hacer un cambio. Las aplicaciones de monitoreo de red permiten realizar un seguimiento de este tipo de datos y planificar con anticipación para hacer los cambios necesarios con facilidad.

7. Mostrar lo que está sucediendo. Informes y estadísticas sobre la salud y la actividad de la red son una gran herramienta para probar la adhesión a un acuerdo de nivel de servicio o demostrar el por qué a las necesidades específicas del dispositivo de fijación o sustitución.

8. Saber cuándo aplicar las soluciones de recuperación de desastres. Sin supervisión de la red, los principales problemas pueden pasar desapercibidos. Con la notificación adicional proporcionada por un control adecuado, se pueden implementar protocolos de recuperación de desastres a tiempo para evitar el tiempo de inactividad y/o fallos del sistema. Ejemplos de esto incluyen baterías de alimentación ininterrumpida durante un corte de energía, el estado de realización de copia de seguridad y la capacidad de predecir el fallo del disco duro.

9. Asegurar que los sistemas de seguridad funcionan correctamente. Sin supervisión de la red, no hay forma de garantizar que los sistemas de seguridad de alta prioridad están

desempeñando sus funciones. Un ejemplo es el firewall, que protege los datos y permite una conectividad segura a Internet.

10. Ahorrar dinero. Reducir el tiempo de inactividad y el tiempo de investigación, ya que menos horas de trabajo significa menos dinero gastado cuando se producen problemas y una mayor productividad en toda la organización.

11. Aumento los beneficios. Evitar las pérdidas financieras causadas por fallas en el sistema sin ser detectado es el resultado final de ser capaces de señalar de manera proactiva y hacer frente a problemas de la red. Todos sus servicios de misión crítica se ejecutarán sin problemas con un servicio de vigilancia, lo que permite más tiempo para hacer funcionar a la empresa.

V. DISEÑO EXPERIMENTAL

A. CONFIGURACIONES RPM DE PRUEBA

Para iniciar se realizaron configuraciones en el equipo Juniper que sería desde el cual se mantendrá este monitoreo. Este equipo se encuentra en el Network Access Point, o NAP, del carrier. Un NAP es un punto de interconexión de Internet importante que permite a los proveedores de acceso a Internet y las compañías de intercambio de tráfico y los servicios entre ellos, de la misma manera que un gran aeropuerto internacional permite a los operadores el cambio de pasajeros y de carga para que puedan llegar a su destino final.

Específicamente se trata del NAP de las Américas, NAP of the Americas en inglés, de Terremark, actualmente propiedad de Verizon. La instalación Nivel IV fue el primer punto de acceso a la red independiente del operador especialmente construido, y es la única instalación de su tipo diseñada específicamente para enlazar a América Latina con el resto del mundo.

El propósito de las pruebas fue observar el comportamiento del equipo, específicamente se deseaba observar la existencia de un aumento en consumo de CPU, o algún tipo de comportamiento no deseado.

Al principio se procedió con la configuración de dos interfaces de prueba en equipos adyacentes, de manera que no se involucrara algún equipo ajeno. Se configuraron dos interfaces lógicas que formaban una WAN. En la Figura se pueden observar las configuraciones realizadas en cada una de las interfaces. Luego se estableció en el equipo principal el test RPM respectivo, cuya configuración se observa en la Figura 23.

Figura 23. Ejemplo configuraciones de interfaces logicas y configuración RPM de prueba.

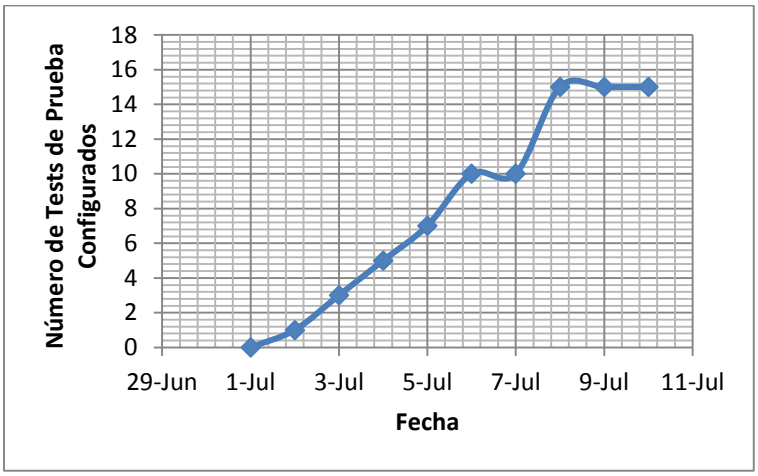
```
unit 0 {
    family inet {
        address 10.2.6.1/30;
    }
}
```

```
unit 192 {
    family inet {
        address 10.2.6.2/30;
    }
}
```

```
probe PRUEBA-INICIAL {
    test TEST-INICIAL {
        probe-type icmp-ping;
        target address 10.2.6.2;
        probe-count 5;
        probe-interval 5;
        test-interval 10;
    }
}
```

Siendo el objetivo de estas configuraciones el monitoreo del comportamiento del equipo. Obviamente esto no nos podría indicar que el equipo podría soportar ser la base de este monitoreo, ya que se habla de decenas de tests que debería manejar. De esta manera se procedió a añadir más pruebas de una manera gradual como se observa en la Figura 24.

Figura 24. Gráfica de la fecha versus el número de tests de prueba en el equipo.



B. DEFINICIÓN Y ORGANIZACIÓN DE PARÁMETROS DE TESTS

Durante esta sección del diseño experimental se concentró en la manera en la que se llegaría a manejar el orden de estas pruebas, específicamente la nomenclatura y organización de tests. Es bastante obvio para el lector de que no sería óptimo el hecho de monitorear un servicio y alarmar a el personal encargado del mismo, si no se tiene una homogeneidad en los monitoreos y sus avisos. Esto entorpecería la proactividad deseada. El objetivo fue idear como relacionar las instalaciones con las configuraciones de los monitoreos, de tal manera que el personal en el NOC sea capaz de ubicar el servicio con inconvenientes de una manera inmediata.

Con respecto a la nomenclatura, se tomaron en cuenta ciertas características del servicio que debían ser identificadas:

- Nombre del cliente.
- Número identificador de la instalación.
- Nombre del routing-instance.
- Opcionalmente el nombre del cliente final.

A la vez se determinó que los únicos identificadores decididos por el usuario en la configuración del RPM serían el nombre del probe y del test, ya que el resto son parámetros específicos para el servicio. Así que en resumen para los RPM se escogería lo siguiente:

- Nombre del probe (owner).
- Nombre del test.

Otro aspecto importante era la distribución de estos RPM. En este caso se analizó la estructura de estos servicios VPN. En otras palabras se debía identificar:

- Número de equipos IP en el NAP involucrados en estos servicios.
- Distribución de los servicios en dichos equipos.
- Ubicación de las configuraciones principales y de las redundancias.

C. CONFIGURACIONES DE LOS SERVICIOS

En esta sección se debieron considerar aspectos tales como:

- Identificación de la IP del CE del servicio en el país respectivo y asegurarse que es el último punto del enlace.
- Identificar el routing-instance en el equipo del NAP que corresponde a la VPN.
- Comprobación de conectividad desde el NAP hacia el cliente.
- Parámetros de la configuración.
- Modalidad de configuración.

1. Identificación de la IP del CE. En este caso se procedió a realizar los siguientes pasos generales para lograr identificar la IP del cliente.

a. Ingreso al PE en el país correspondiente. Usualmente este PE se encuentra en la red core, de esta manera se ingresaba a este equipo inicialmente.

b. Búsqueda del ID de instalación. Se buscaba el ID de la instalación documentado en la descripción de una interfaz. Esto se realizaba con comandos de despliegue de la configuración completa y filtros de despliegue tales como include o match. Como ejemplo se podría tomar el ID de instalación

1234, para su búsqueda se ingresaba el comando show configuration | match 1234 | display set para encontrarlo rápidamente. Un ejemplo de este comando se observa en la Figura 25.

Figura 25. Ejemplo de búsqueda del ID del servicio en la configuración.

```
aarriola@R3-SRX210HM> show configuration | match 1234 | display set
set interfaces fe-0/0/3 unit 0 description "Id de instalacion: 1234"
```

c. Búsqueda del routing-instance. Al obtener la interfaz correspondiente se buscaba el routing-instace al cual pertenecía la misma. Esto también se realizaba utilizando los comandos y filtros anteriormente mencionados. Se observa un ejemplo de esta búsqueda en la Figura 26.

Figura 26. Búsqueda de la instancia correspondiente a la interfaz ubicada.

```
aarriola@R3-SRX210HM> show configuration routing-instances | match fe-0/0/3.0 | display set
set routing-instances instancia-cliente interface fe-0/0/3.0
```

d. Detección de la IP del CE. En la mayoría de los casos se maneja BGP entre PE y CE, en estos casos se observaba la IP del neighbor la cual, idealmente, debía ser del cliente, en algunos pasos se tenía una sesión BGP con un equipo del mismo carrier, se debía ingresar a este equipo, realizar la misma búsqueda, y entonces se encontraría la IP del CE. Se podía detectar rápidamente si el neighbor BGP correspondía al equipo del cliente o equipo de la misma red observando el número del sistema autónomo, en otras palabras, se observaba si la sesión era interna o externa. En los casos que se utilizarán rutas estáticas, usualmente el next-hop de estas indicaban la IP del cliente. Como último paso se realizaba un telnet hacia el equipo, y se observaba en la advertencia que el equipo correspondía al cliente. En la Figura se observa un ejemplo.

2. Identificación del routing-instance en el NAP. Luego de obtener la IP del cliente, se procedía a ubicar el routing-instance correspondiente en el NAP. Esto se lograba con el parámetro vrf-target, ambos debían coincidir. Se realizaba utilizando los comandos y filtros mencionados anteriormente.

Figura 27. Ejemplo de búsqueda del routing-instance correspondiente a la VPN.

```
aarriola@R3-SRX210HM> show configuration routing-instances | match target:2222:456 | display set
set routing-instances instancia-cliente vrf-target target:2222:456
```

Figura 28. Importancia de utilización del comando display set en la búsqueda.

```
aarriola@R3-SRX210HM> show configuration routing-instances | match target:2222:456
    vrf-target target:2222:456;

aarriola@R3-SRX210HM> show configuration routing-instances | match target:2222:456 | display set
set routing-instances instancia-cliente vrf-target target:2222:456
```

3. Comprobación de conectividad desde el NAP hacia el cliente. Al tener completamente ubicada la estructura del servicio y la IP del cliente se procedía a probar conectividad desde el NAP hacia el cliente final realizando pruebas de ping. Esto con el fin de comprobar que se llegaba a la IP y que el equipo del cliente permitía recepción de ICMP, y que respondía a los mismos, esto ya que en algunos casos los equipos tienen bloqueadas estas funciones por seguridad.

4. Parámetros de la configuración. Luego de cumplir los pasos anteriores ya se podía proceder a configurar el RPM respectivo para el servicio, de esta manera monitoreando la conexión punto a punto del mismo. Los parámetros que debían tomarse en cuenta serían los siguientes: probe-count, probe-interval, test-interval y threshold. Se debía determinar una combinación de estos que nos garantice el hecho que al fallar el ping test el servicio este actualmente caído, y que no sea solamente una coincidencia o inestabilidad. Se probaron variaciones de los mismos.

5. Modalidad de configuración. Al compartir estos tests muchas líneas de configuración en común se deseó agrupar estas en una sola parte de la configuración y heredar la misma a cada test. De esta manera se facilita el cambio de los parámetros, ya que solo se debe cambiar una vez por todos los test RPM, en lugar de una vez por cada test. En este punto se procedió con la investigación de los grupos de configuración en Juniper para observar la mejor manera de implementar la herencia de configuración a cada test RPM.

D. SERVIDOR PARA GRÁFICAS DEL MONITOREO

Se planteó entre los objetivos del proyecto tener gráficas de estos servicios para referencia histórica. Para este fin se utilizó Cacti. Esta es una solución completa para la generación de gráficos en red. Esta herramienta, desarrollada en PHP, provee un poller ágil, plantillas de gráficos avanzadas, múltiples métodos para la recopilación de datos, y manejo de usuarios. Tiene una interfaz de usuario fácil de usar, que resulta conveniente para nuestro propósito. Existe en el NAP un servidor al cual posee este sistema.

La MIB de RPM posee una gran variedad de OID’s interesantes, los cuales podría sondear el sistema y realizar las gráficas correspondientes. En este caso nos interesaron el RTT y el Jitter, con tal de no sobrecargar la gráfica. El hecho de que no existan datos en la gráfica significa que el servicio está caído, ya que el RPM no ha podido obtener estos datos.

Para agregar la gráfica al Cacti se debe obtener el OID, crear la gráfica estableciendo el nombre y las características como, por ejemplo, los colores. Esto OID se obtuvieron consultando en el equipo como se observa en la Figura 29 y 30, cada test tiene su propio OID por resultado deseado, como se mencionó, nos

interesan el round-trip-time y el jitter, y estos son los que el servidor debe sondear para obtener el valor deseado para la gráfica. Estos OID se pueden convertir de ASCII a decimal como se observa en la Figura 31, esto nos sirve para confirmar el OID en decimal del servicio, de esta manera podemos ver el nombre del test y del probe correspondiente a un OID.

Figura 29. Ejemplo de OID's en ASCII que sondeara el servidor.

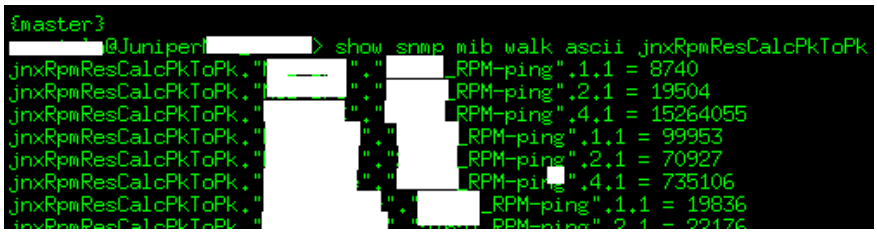


Figura 30. Ejemplo de OID's en decimal que sondeara el servidor.

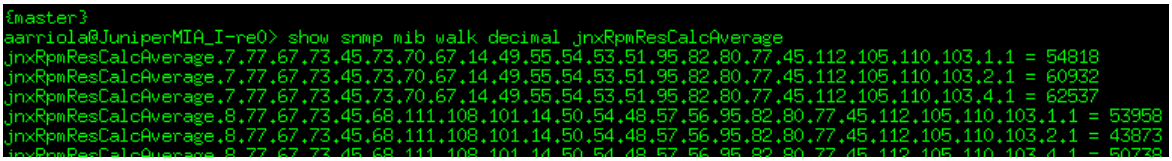
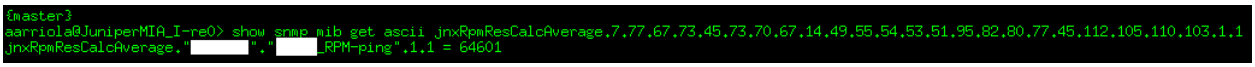


Figura 31. Ejemplo conversion OID de decimal a ASCII.



Una de las intenciones de este trabajo es facilitar la adición de más enlaces al monitoreo, por esta razón se encontró una manera para añadir las gráficas de una manera mucho más sencilla. Para este propósito se tuvo un script en el lenguaje Perl, el cual facilita al Cacti realizar un query de los OID indicados anteriormente, detectando los existentes. De esta manera simplemente se procede a seleccionar el cual se desea crear una gráfica, al tener una plantilla de gráfica específica vinculada a este sondeo de datos.

Figura 32. Parámetros de la plantilla de gráfica para los RPM.

Graph Template Items [edit: Juniper RPM]

Graph Item	Data Source	Graph Item Type	CF Type	Item Color
Item # 1	**(avaragertt): RTT Ave**	**AREA**	**AVERAGE**	**C4FD3D**
Item # 2	(avaragertt): Current:	GPRINT	LAST	
Item # 3	(avaragertt): Average:	GPRINT	AVERAGE	
Item # 4	(avaragertt): Maximum:<HR>	GPRINT	MAX	
Item # 5	**(jitter): Jitter**	**LINE2**	**AVERAGE**	**FF3932**
Item # 6	(jitter): Current:	GPRINT	LAST	
Item # 7	(jitter): Average:	GPRINT	AVERAGE	
Item # 8	(jitter): Maximum:<HR>	GPRINT	MAX	

El script en Perl hace el query, encuentra el nombre del owner y del test, el cual pasa a ser el nombre de la gráfica más adelante. Debe convertir el OID a formato ASCII y luego de regreso dependiendo de lo que se necesita, si hacer el query o enviar el nombre. El script se encuentra en la sección de Anexos debido a que es muy extenso.

E. SERVIDOR SYSLOG

Para la generación de los avisos hacia el operador se decidió utilizar syslog debido a que completa los requerimientos necesarios y funciona de una manera simple, haciéndolo la mejor opción. En esta sección se manejaron los siguientes puntos con respecto a los syslog:

- Generación.
- Envío.
- Recepción.
- Manejo de la información.
- Alerta al operador.

Con respecto a la generacíon de los syslog, en este punto se deseaba lograr generar un syslog a partir de los eventos de los test RPM, por ejemplo, cada vez que se completaba exitosamente un test, cuando fallaba un ping y, más importante, cuando fallaba un test, de esta manera, según los parámetros elegidos para la prueba RPM, indicándonos la caída del servicio. Para lograr esto se realizaron pruebas para observar cuando ocurría la generación de syslogs, y cuales eran estos mensajes.

Para lograr observar estos resultados se realizaron pruebas de laboratorio. Se configuró una interfaz que actuaría como servidor cuya configuración se puede observar en la Figura 33, se configuró un test RPM en otro router para actuar como cliente, esta configuración se observa en la Figura 34. También se debió configurar una interfaz en el router cliente para lograr la conectividad, esto se ve en la Figura 35. Para obtener los syslogs se dejó correr las pruebas y luego se desconectó la conexión, de esta manera forzando la falla.

Figura 33. Configuración de la interfaz del servidor RPM para pruebas syslog.

```
[edit interfaces]
aarriola@R1-SRX100H# show fe-0/0/3
unit 0 {
    family inet {
        address 10.2.6.2/30;
    }
}
```

Figura 34. Configuración del test RPM en el router cliente para pruebas syslog.

```
[edit services rpm]
aarriola@R3-SRX210HM# show
probe PRUEBA1 {
    test test1_PRUEBA1 {
        probe-type icmp-ping;
        target address 10.2.6.2;
        probe-count 4;
        probe-interval 5;
        test-interval 30;
        routing-instance R2;
        thresholds {
            total-loss 5;
        }
        traps test-failure;
    }
}
```

Figura 35. Configuración de interfaz en router cliente para syslog.

```
[edit interfaces]
aarriola@R3-SRX210HM# show fe-0/0/3
unit 0 {
    family inet {
        address 10.2.6.1/30;
    }
}
```

Adicionalmente se crearon dos archivos para guardar logs, a el archivo rpm_ping, cuya configuración se observa en la Figura 36, se le enviaba cualquier syslog tipo info y que tuviera en su contenido rmopd, que es el daemon responsable de estas tareas. Al archivo failed_ping_tests se le enviaba cualquier syslog tipo información, pero que contuviera PING_TEST_FAILED, el cual es uno de los tres syslog que nos interesan, la explicación de estos syslogs se puede observar en la Figura 38, y la configuración del archivo se puede observar en la Figura 37. La información de la Figura 38 fue importante para generar las configuraciones de los archivos, al conocer la severidad y el tipo de los syslogs.

Figura 36. Configuración archivo para logs rpm_ping para pruebas.

```
[edit system syslog]
aarriola@R3-SRX210HM# show file rpm_ping
any info;
match rmopd;
archive size 1m files 1;
```

Figura 37. Configuración archivo para logs failed_ping_tests para pruebas.

```
[edit system syslog]
aarriola@R3-SRX210HM# show file failed_ping_tests
any info;
match PING_TEST_FAILED;
archive size 5m files 5;
```

Figura 38. Detalle de los syslogs a observar durante la prueba.

```
aarriola@R3-SRX210HM> help syslog PING_TEST_COMPLETED
Name:          PING_TEST_COMPLETED
Message:       pingCtlOwnerIndex = <test-owner>, pingCtlTestName = <test-name>
Help:          ping test completed
Description:   All probes were sent and the number of failed probes was less
               than the pingCtlTrapTestFailureFilter threshold.
Type:          Event: This message reports an event, not an error
Severity:      info

aarriola@R3-SRX210HM> help syslog PING_TEST_FAILED
Name:          PING_TEST_FAILED
Message:       pingCtlOwnerIndex = <test-owner>, pingCtlTestName = <test-name>
Help:          ping test failed
Description:   All probes were sent but the number of failed probes equaled or
               exceeded the pingCtlTrapTestFailureFilter threshold.
Type:          Event: This message reports an event, not an error
Severity:      info

aarriola@R3-SRX210HM> help syslog PING_PROBE_FAILED
Name:          PING_PROBE_FAILED
Message:       pingCtlOwnerIndex = <test-owner>, pingCtlTestName = <test-name>
Help:          Number of ping probe failures exceeded threshold
Description:   The number of successive probe failures exceeded the
               pingCtlTrapProbeFailureFilter threshold.
Type:          Event: This message reports an event, not an error
Severity:      info
```

Enseguida se procedió a configurar el envío de estos syslogs a un host. Esto se dividió en dos partes: las pruebas y la configuración real. Para las pruebas se configuró un host en el router cliente, hacia donde se enviarían estos syslog. Para esto se configuró una interfaz para comunicación con una computadora. La configuración en el router se puede observar en la Figura 39, la colocación de la respectiva IP en la computadora en la Figura 40. Y la configuración del host en la Figura 41. Se procedió a utilizar wireshark para observar lo recibido en la tarjeta de la computadora y confirmar la presencia de los syslog.

Figura 39. Configuración de interfaz hacia computadora para pruebas syslog.

```
[edit interfaces]
aarriola@R3-SRX210HM# show vlan
unit 7 {
    family inet {
        address 192.168.7.211/24;
    }
}
```

Figura 40. Colocación de IP en computadora para pruebas de envío syslog.

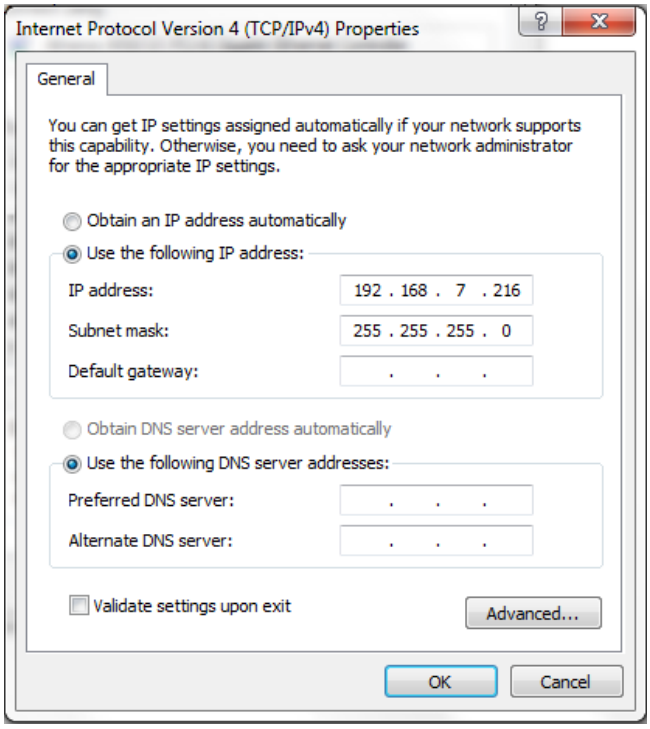


Figura 41. Configuración para el host syslog en el router cliente para pruebas de envío.

```
[edit system syslog]
aarriola@R3-SRX210HM# show
archive size 100k files 3;
user * {
    any emergency;
}
host 192.168.7.216 {
    any info;
    match PING;
    log-prefix SRX210H;
}
```

Luego de la parte de prueba se procedió a la configuración que sería utilizada para el envío de los syslog en el equipo en el NAP. Inicialmente se enviaron los syslog a un host el cual mantendría un host syslog, llamado Syslog Watcher, el cual es freeware y es compatible con Windows, esta configuración se puede observar en la Figura 42, con el objetivo de observar los syslog siendo recibidos por un host syslog. Luego se procedió a enviar esta información hacia una computadora personal, esta configuración se observa en la Figura 43, en esta configuración solamente interesaba enviar los syslog correspondientes a PING_TEST_FAILED y PING_PROBE_FAILED. Cabe mencionar que la configuración de la Figura 43 es la utilizada para el funcionamiento del proyecto.

Figura 42. Configuración de host hacia host con Syslog Watcher.

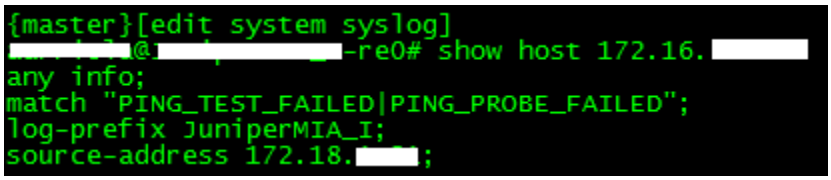
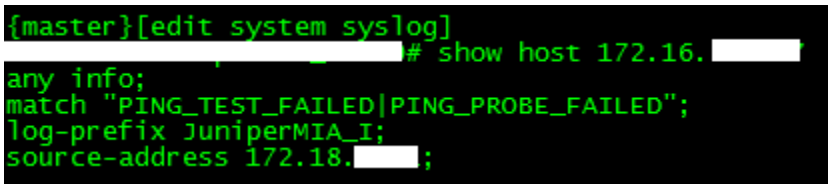


Figura 43. Configuración de host syslog utilizada.



A continuación se trató con la recepción de los syslog. Como se mencionó anteriormente, se configuró un host para el Syslog Watcher en una computadora, y otro para la computadora personal. En el caso de la computadora personal se desarrolló un programa en Python el cual actuara como servidor de syslog. Para esto debía ser un programa que tuviera un socket escuchando en el puerto 514 por mensajes UDP. Este código se puede observar en la Figura 44.

Figura 44. Código en Python para un servidor Syslog

```
HOST, PORT = "0.0.0.0", 514
import SocketServer
class SyslogUDPHandler(SocketServer.BaseRequestHandler):
    def handle(self):
        global actual_log
        data = bytes.decode(self.request[0].strip())
        socket = self.request[1]
        actual_log=str(data)
        print(actual_log)
if __name__ == "__main__":
    try:
        server = SocketServer.UDPServer((HOST,PORT), SyslogUDPHandler)
        server.serve_forever(poll_interval=0.5)
    except (IOError, SystemExit):
        raise
```

Al tener este código se deseaba proceder con analizar cada syslog recibido para identificar si correspondía a un servicio caído, y en este caso se deseó alertar al usuario de la computadora de alguna manera. Se decidió generar un pop-up en el centro de la pantalla, que contuviera el nombre del owner del test, y el número identificador de la instalación. Para esto se tuvo que realizar un análisis de la información recibida y si lo indicaba desplegar el pop-up con la información deseada. También se decidió agregar una manera de archivar todos los logs recibidos para revisión. Se agregaron dichos requerimientos en el código, para el cual fue necesario utilizar diferentes hilos de ejecución, uno de ellos escuchando los syslog y el otro analizando la información y alertando al usuario, los logs se guardaban al archivo pinglogfile.log. Se debe

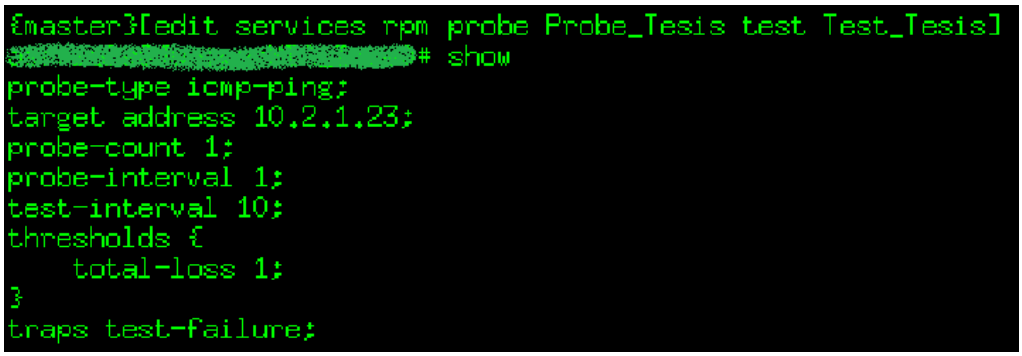
mencionar que el código mostrado en la Figura 45 es el utilizado en el proyecto. La utilización de excepciones fue vital para evitar que el programa se detuviera al tener un error, se atrapa la excepción y el programa debe continuar funcionando.

Figura 45. Código en Python para el servidor syslog encargado de alertar al usuario.

```
#!/usr/bin/env python
LOG_FILE = 'pinglogfile.log'
HOST, PORT = "0.0.0.0", 514
import logging
import SocketServer
import threading
import time
import ctypes
import win32ui

actual_log = ""
logging.basicConfig(level=logging.INFO, format='%(message)s', datefmt='',
                    filename=LOG_FILE, filemode='a')

class SyslogUDPHandler(SocketServer.BaseRequestHandler):
    def handle(self):
        global actual_log
        data = bytes.decode(self.request[0].strip())
        socket = self.request[1]
        actual_log=str(data)
        print(actual_log)
        logging.info(actual_log)
#Hilo de ejecucion que escucha los syslog
def poll_syslog():
    try:
        server = SocketServer.UDPServer((HOST,PORT), SyslogUDPHandler)
        server.serve_forever(poll_interval=0.5)
    except (IOError, SystemExit):
        raise
#Hilo de ejecucion que analiza la informacion y alerta al usuario
def parse_syslog():
    global log_in_array
    while 1:
        time.sleep(3)
        log_in_array = actual_log.split()
        #Si el servicio esta caido:
        try:
            if log_in_array[5] == 'PING_TEST_FAILED:':
                instalacion = log_in_array[len(log_in_array)-1].split('_')[0]
                cliente_vrf = log_in_array[8]
                win32ui.MessageBox(cliente_vrf+"\n"+instalacion,
                                   "PING_TEST_FAILED",4096)
        except:
            pass #en caso de index out of bounds exception

t1 = threading.Thread(target=poll_syslog)
t2 = threading.Thread(target=parse_syslog)
try:
    t1.start()
    t2.start()
except (IOError, SystemExit):
    raise
```

Se configuró un test de prueba para poder mostrar en este trabajo el resultado de la alerta sin comprometer la confidencialidad de la información de los clientes y datos de los enlaces del carrier. Esta configuración se puede observar en la Figura 46, esta tenía una dirección IP aleatoria para asegurar que fallaría.

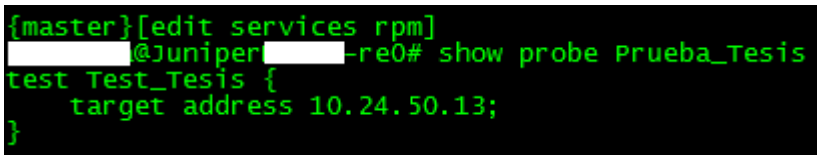
Figura 46. Configuración RPM de prueba (Se censura por confidencialidad).



F. PRUEBA DE FUNCIONAMIENTO DE LA ALERTA AL OPERADOR

Con el fin de probar en tiempo real el correcto accionamiento de la alerta se configuró el test Test_Tesis como se puede observar en la Figura 47, dicho test hereda todos sus parámetros de un grupo de configuración discutido en la sección de resultados. Por este momento solamente se indica la IP a la que se enviarán los probes, la cual es una ping que se conoce de antemano que no contestará. Esto deberá accionar la alerta al operador al alcanzar el umbral e indicándole al operador que dicho servicio se encuentra caído.

Figura 47. Configuración test RPM para prueba de alerta al operador.

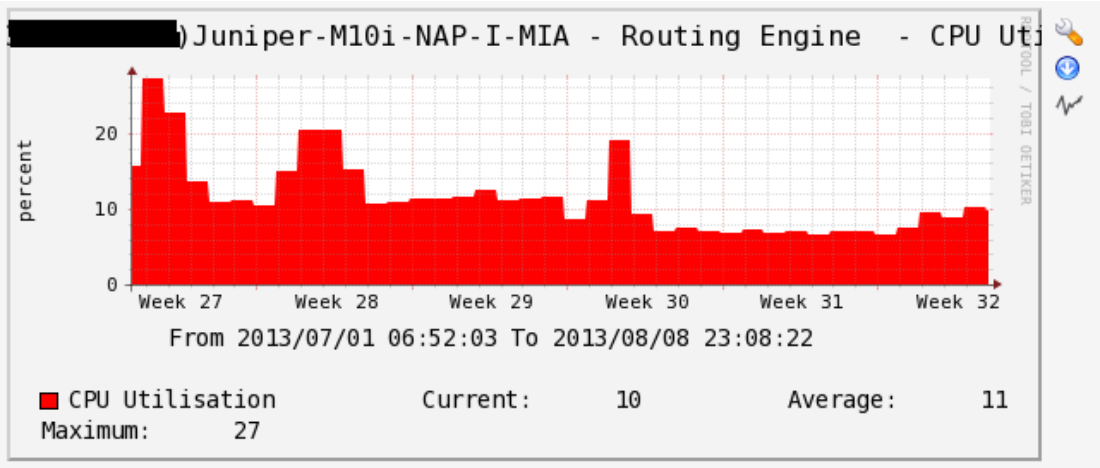


VI. RESULTADOS

A. RESULTADOS DE CONFIGURACIONES RPM DE PRUEBA

El propósito de las pruebas fue observar el comportamiento del equipo, específicamente se deseaba observar la existencia de un aumento en consumo de CPU, o algún tipo de comportamiento no deseado. Como se observa en la Figura 48, no existió un aumento en el consumo durante el periodo de pruebas, los picos observados son esporádicos y pudieron haber existido por algún tipo de evento especial, como un aumento considerable de tráfico, o un cambio sustancial en la configuración. Interesa observar de la semana 30 en adelante, donde se ve un consumo normal y estable del 10%.

Figura 48. Consumo de CPU durante el periodo de pruebas del 1 de julio al 14 de julio.



B. RESULTADOS DE DEFINICIÓN Y ORGANIZACIÓN DE TESTS

Se vinculó el nombre del routing-instance con el nombre del probe (owner), ya que en este se encuentra en nombre del cliente y del cliente final. Para el nombre del test, se decidió colocar el número identificador del enlace continuado por _RPM-ping, la razón de añadir este último identificador fue debido al hecho que se planea escalar estos monitoreos RPM a otros tipos, de esta manera se lograría identificar el tipo de prueba con este añadido. En la Figura 49 se puede observar un ejemplo de la estructura que tendría una configuración RPM. Se debe mencionar que algunos servicios estarían bajo el mismo owner (probe).

Figura 49. Ejemplo de nomenclatura en configuración de un probe RPM.

```
probe NombreRouting-Instance {
    test IdInstalacion_RPM-ping {
        probe-type icmp-ping;
        target address 10.0.0.1;
        probe-count 10;
        probe-interval 10;
        test-interval 10;
    }
}
```

Con respecto a la distribución de los mismos, se identificó que las configuraciones principales se manejan desde un mismo equipo, así como las secundarias desde otro. De esta manera se facilitó la distribución de las configuraciones RPM, ya que se encontrarían todas en un mismo equipo.

C. RESULTADOS DE LAS CONFIGURACIONES DE LOS SERVICIOS

Con respecto a las configuraciones de los servicios, se utilizaron variaciones de los parámetros hasta encontrar una combinación que acertará al indicar un incidente con un servicio. Entre las combinaciones de parámetros se pueden observar algunas en el Cuadro 4. Algunos problemas identificados pueden ser, por ejemplo, en la combinación 1, el tiempo de la prueba es de 2 minutos, y la cantidad de pings es 4, esto indicaba a veces servicios caídos, los cuales no necesariamente se encontraban con un inconveniente, sino que era sencillo que se perdieran 4 sondas. Los parámetros finalmente escogidos son los observados en el Cuadro 5.

Cuadro 4. Combinaciones de parámetros para test RPM probadas.

Parámetro	**Combinación 1**	**Combinación 2**	**Combinación 3**
Probe-count	4	6	8
Probe-interval (s)	30	30	20
Test-interval (s)	30	30	20
Data-size (bytes)	128	128	128
Thresholds	Total-loss 4	Total-loss 6	Total-loss 8

Cuadro 5. Parámetros elegidos para los tests.

Parámetro	Valor
Probe-count	9
Probe-interval	20 segundos
Test-interval	30 segundos
Data-size	128
Thresholds	Total-loss 9

Con respecto a las líneas de configuración en común y la agrupación de las mismas para herencia se puede observar el resultado en la Figura 50, como se puede notar se le llamo al grupo RPM-standard-pingtest-config para que su propósito sea obvio, así se podrán crear más grupos de configuración para diferentes tipos de pruebas en el futuro. Esta configuración se aplicó bajo la jerarquía services rpm, para que entonces pasará a heredarse a cualquier probe, y cualquier test, el ejemplo de la colocación de esta línea de configuración se observa en la Figura 51.

Figura 50. Grupo de configuración para los test (se censura por confidencialidad).

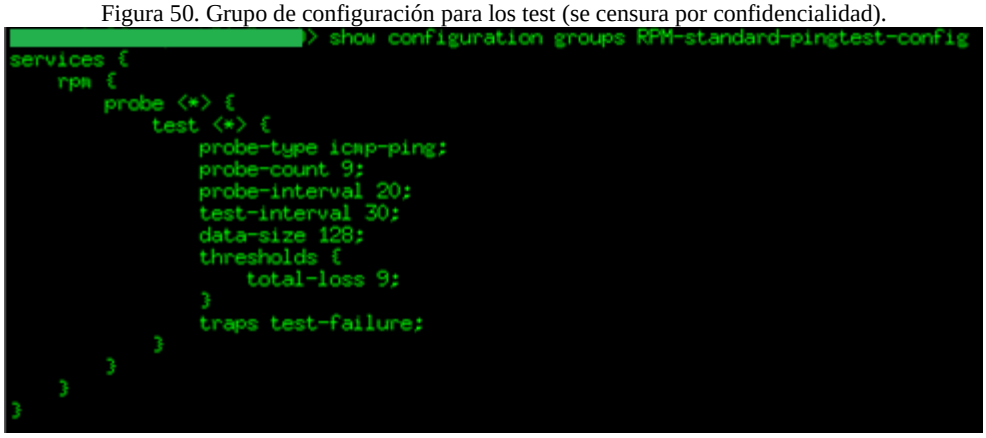
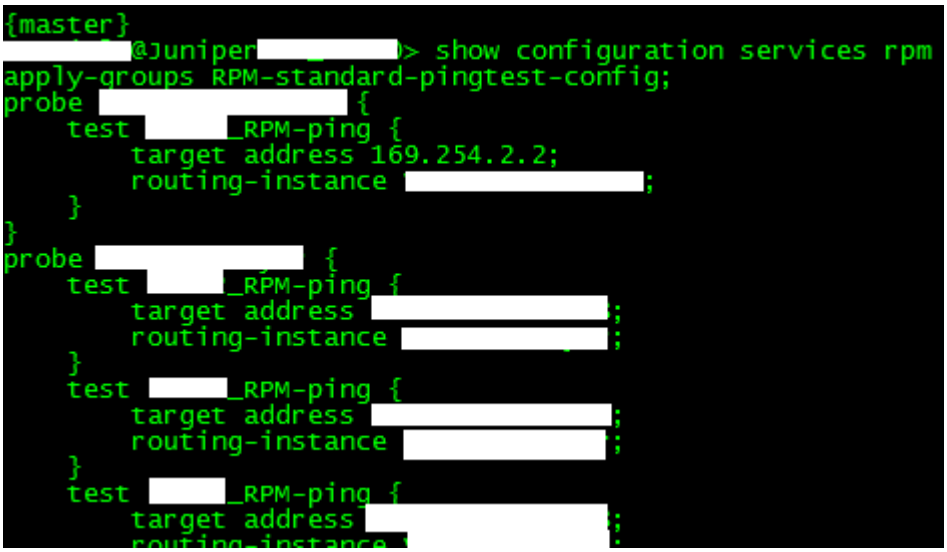


Figura 51. Aplicación del grupo de configuración a los test (se censura por confidencialidad).



Fueron inicialmente configurados 50 servicios de una manera distribuida durante 1 semana para monitorear el comportamiento del equipo. Se puede observar en la Figura 52 la distribución de las fechas y el número de configuraciones, y se puede observar en la Figura 53 el consumo de CPU del equipo durante estas fechas, y como se puede notar, existen algunos picos en el consumo, estos, como fue mencionado anteriormente, pueden ser relacionados a algún evento especial, pero se puede confirmar que las sondas RPM no son responsables debido a la estabilidad en el consumo observada luego del periodo de configuraciones.

Figura 52. Gráfica de la fecha contra el número de configuraciones en el equipo.

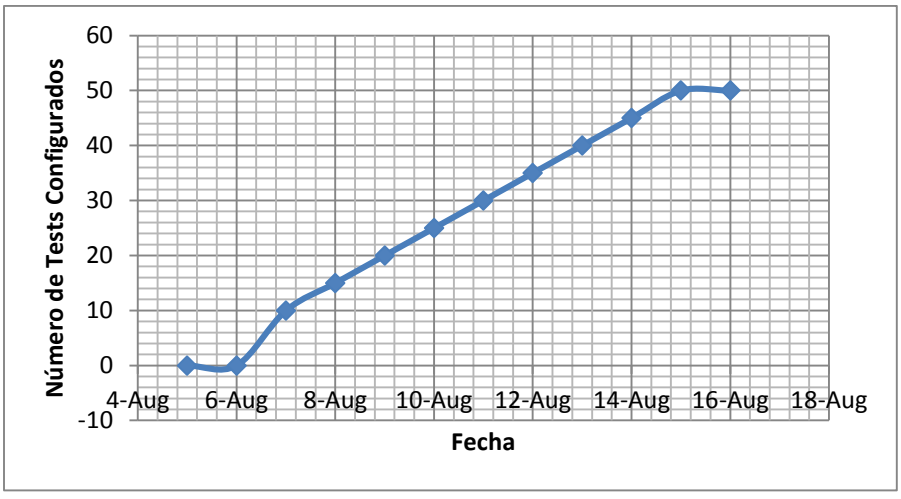
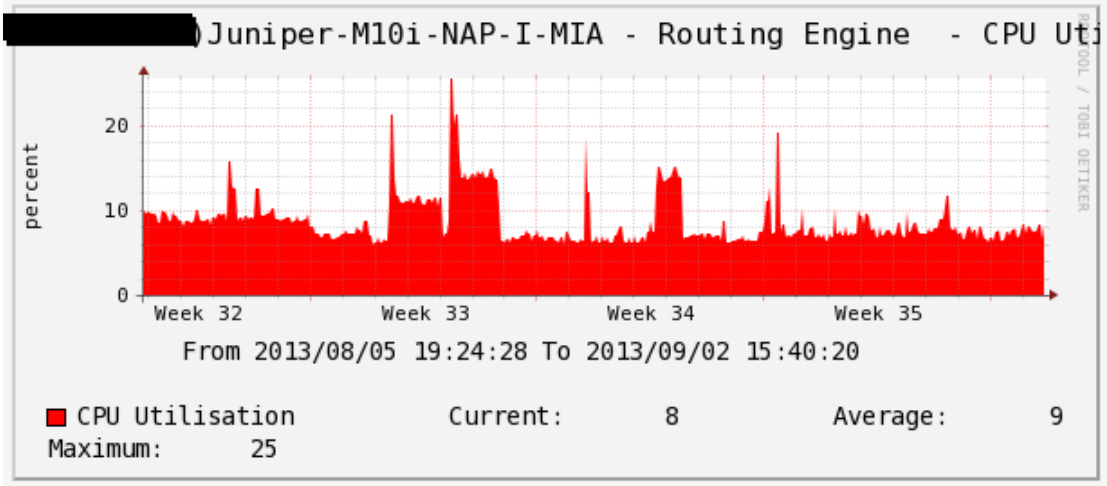


Figura 53. Gráfica del consumo de CPU del equipo durante el periodo de configuraciones.



Debido a razones de confidencialidad no se pueden mostrar todas las configuraciones hechas, ya que estas contienen los nombres de los clientes del carrier, los números identificadores de las instalaciones y, en muchos casos, direcciones IP públicas, toda esta información no puede ser revelada.

D. RESULTADOS DEL SERVIDOR PARA GRÁFICAS DEL MONITOREO

En esta sección se muestran algunas de las gráficas para los servicios. Debe ser mencionado que se censuran los datos confidenciales del carrier en las imágenes. Podemos observar en la Figura 54 una gráfica que abarca el día actual, en la Figura 55 se observa una gráfica que abarca la semana y en la Figura 56 para el mes actual, también se cuenta con la gráfica para el año. En la Figura 57 podemos observar el buscador de gráficas del Cacti, simplemente ingresando el número de instalación en donde se indica con el cuadro verde se encuentra la gráfica del servicio. En la Figura 58 podemos ver un ejemplo de un servicio que estuvo caído por un momento, de esta manera ya se puede dar cuenta el lector de lo conveniente que es esta herramienta como complemento.

Figura 54. Ejemplo de gráfica del día actual para uno de los enlaces en el servidor Cacti.

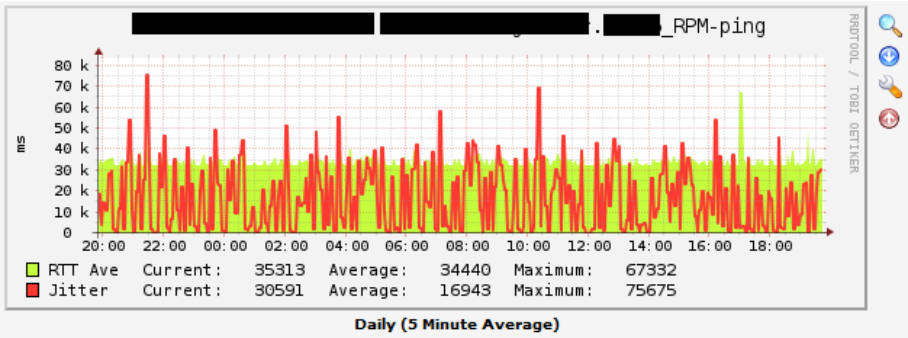


Figura 55. Ejemplo de gráfica de la semana para enlace en el servidor Cacti.

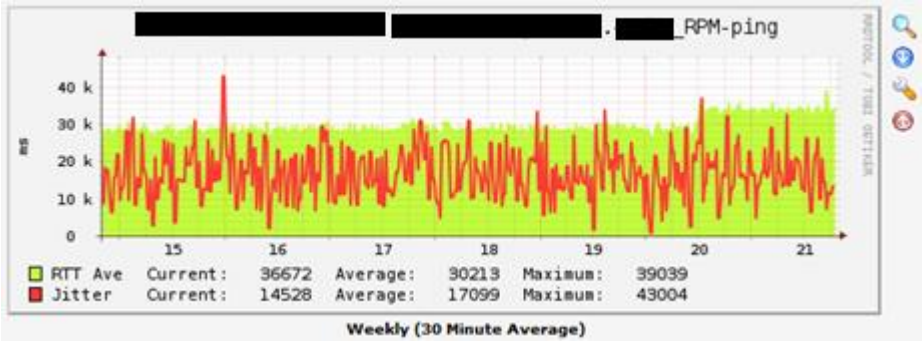


Figura 56. Ejemplo de gráfica del mes actual para uno de los enlaces en el servidor Cacti.

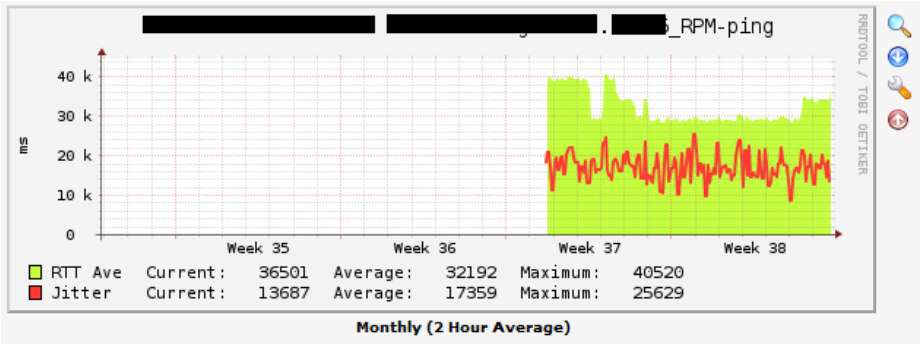
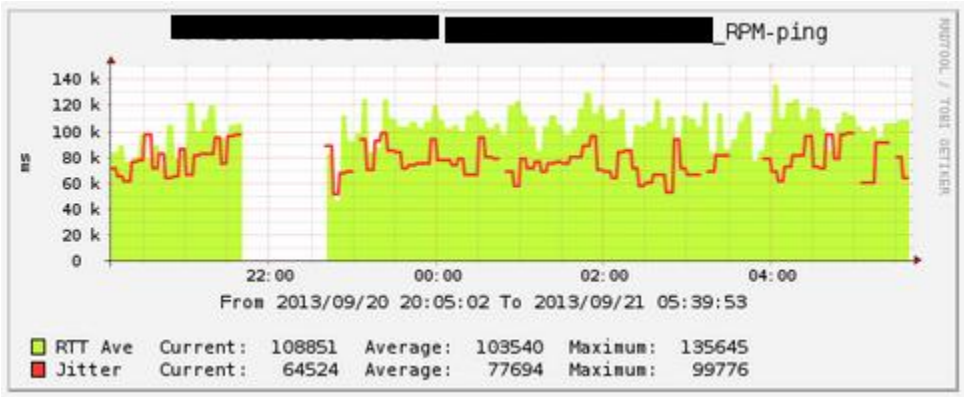


Figura 57. Buscador de gráficas en Cacti.

Figura 58. Gráfica donde se aprecia la caída momentánea de un servicio.



E. RESULTADOS DEL SERVIDOR SYSLOG

Como se mencionó en la sección de diseño experimental, el tema de syslog se dividió en las siguientes tareas:

- Generación.
- Envío.
- Recepción.
- Manejo de la información.
- Alerta al operador.

1. Generación. Podemos observar en la Figura 59 los resultados históricos de la prueba realizada, se observa que luego de completar 4 probes y completar el test se desconectó para observar 4 probes sin ruta al objetivo y fallar el test. En la Figura 60 se observa el resumen de los resultados de los probes para la prueba. Nuestro objetivo era observar la generación de syslogs en los archivos para logs creados en el diseño experimental, y efectivamente podemos observar en la Figura 61 el contenido de ambos logs, a los cuales se les borró cualquier información antes de iniciar el test como se puede observar en el log. Efectivamente podemos observar que concuerda la generación del syslog PING_TEST_FAILED en la Figura 61, a las 10:48, luego de 4 probes no exitosos y fallar el test en la Figura G1. De esta manera se confirmó que los syslog deseados se estaban generando.

Figura 59. Resultados históricos de la prueba RPM realizada para generación de syslogs.

```
aarriola@R3-SRX210HM> show services rpm history-results
    Owner, Test                 Probe received              Round trip time
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:47:19 2013            1996 usec
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:47:24 2013            1821 usec
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:47:29 2013            1532 usec
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:47:34 2013            1687 usec
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:48:05 2013  No route to target
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:48:10 2013  No route to target
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:48:15 2013  No route to target
    PRUEBA1, test1_PRUEBA1      Sat Aug 17 10:48:20 2013  No route to target
```

Figura 60. Resumen de resultados de la prueba RPM realizada para generación de syslogs.

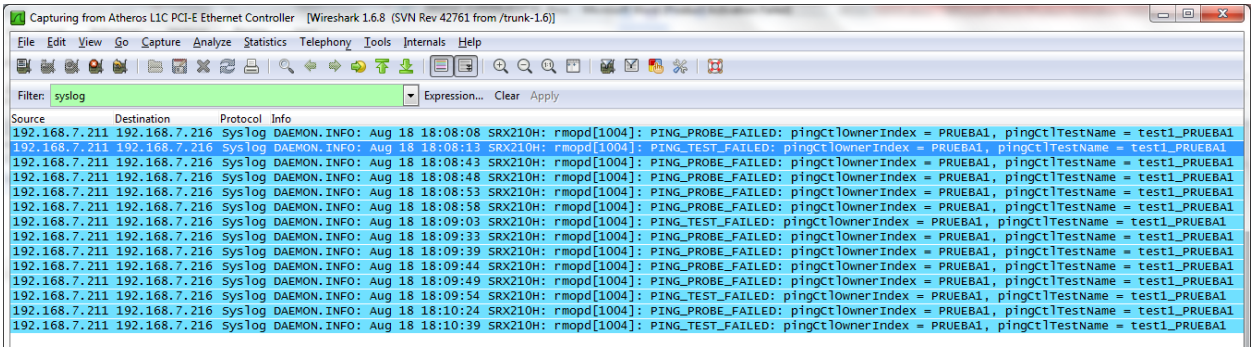
```
aarriola@R3-SRX210HM> show services rpm probe-results
    Owner: PRUEBA1, Test: test1_PRUEBA1
    Target address: 10.2.6.2, Probe type: icmp-ping
    Routing Instance Name: R2
    Test size: 4 probes
    Probe results:
      No route to target, Sat Aug 17 10:48:20 2013
    Results over current test:
      Probes sent: 4, Probes received: 0, Loss percentage: 100
    Results over last test:
      Probes sent: 4, Probes received: 0, Loss percentage: 100
    Results over all tests:
      Probes sent: 8, Probes received: 4, Loss percentage: 50
      Measurement: Round trip time
        Samples: 4, Minimum: 1532 usec, Maximum: 1996 usec, Average: 1759 usec,
        Peak to peak: 464 usec, Stddev: 171 usec, Sum: 7036 usec
```

Figura 61. Contenido de ambos logs creados para los syslogs deseados.

```
aarriola@R3-SRX210HM> show log rpm_ping
Aug 17 10:47:05 R3-SRX210HM clear-log[3175]: logfile cleared
Aug 17 10:47:12  R3-SRX210HM mgd[1043]: UI_CHILD_START: Starting child '/usr/sbin/rmopd'
Aug 17 10:47:12  R3-SRX210HM mgd[1043]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rmopd', PID 3182, status 0
Aug 17 10:47:19  R3-SRX210HM mgd[1043]: UI_COMMIT_PROGRESS: Commit operation in progress:  notifying rmopd(24)
Aug 17 10:47:34  R3-SRX210HM rmopd[1004]: PING_TEST_COMPLETED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:05  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:05  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:10  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:10  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:15  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:15  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:20  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:20  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:25  R3-SRX210HM rmopd[1004]: PING_TEST_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1

aarriola@R3-SRX210HM> show log rpm_ping
Aug 17 10:47:05 R3-SRX210HM clear-log[3175]: logfile cleared
Aug 17 10:47:12  R3-SRX210HM mgd[1043]: UI_CHILD_START: Starting child '/usr/sbin/rmopd'
Aug 17 10:47:12  R3-SRX210HM mgd[1043]: UI_CHILD_STATUS: Cleanup child '/usr/sbin/rmopd', PID 3182, status 0
Aug 17 10:47:19  R3-SRX210HM mgd[1043]: UI_COMMIT_PROGRESS: Commit operation in progress:  notifying rmopd(24)
Aug 17 10:47:34  R3-SRX210HM rmopd[1004]: PING_TEST_COMPLETED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:05  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:05  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:10  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:10  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:15  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:15  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:20  R3-SRX210HM rmopd[1004]: RMOPD_ICMP_SENDMSG_FAILURE: sendmsg(ICMP): No route to host
Aug 17 10:48:20  R3-SRX210HM rmopd[1004]: PING_PROBE_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
Aug 17 10:48:25  R3-SRX210HM rmopd[1004]: PING_TEST_FAILED: pingCtlOwnerIndex = PRUEBA1, pingCtlTestName = test1_PRUEBA1
```

2. Envío. Luego de confirmar que los syslog deseados se estaban generando, se procedió a confirmar que se enviaban. Primero, las pruebas utilizando wireshark, como fue mencionado en la sección de Diseño Experimental, y se puede observar en la Figura 62 el resultado positivo.

Figura 62. Captura de syslog en computadora host utilizando wireshark.



3. Recepción. Como se menciona en el diseño experimental, se recibieron los syslog en Syslog Watcher como se observa en la Figura 63 y en el servidor realizado con Python. La recepción de los syslog en Python se puede observar funcionando en la Figura 64.

Figura 63. Recepción de syslog con Syslog Watcher.

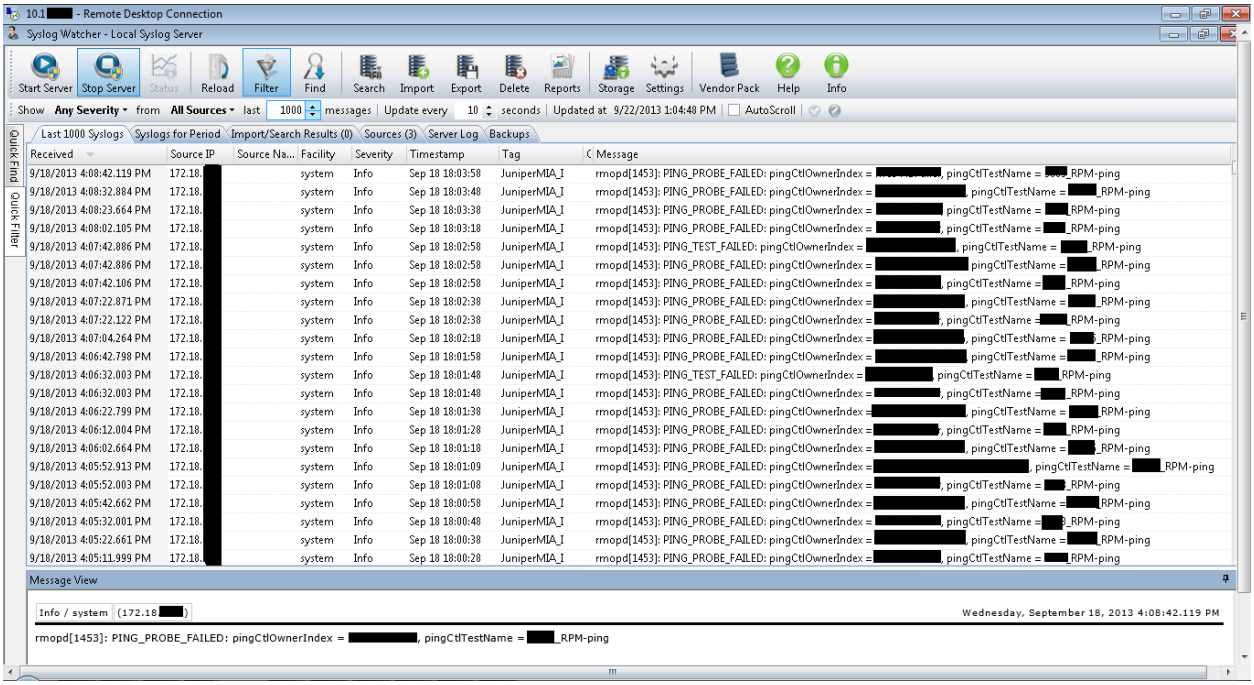
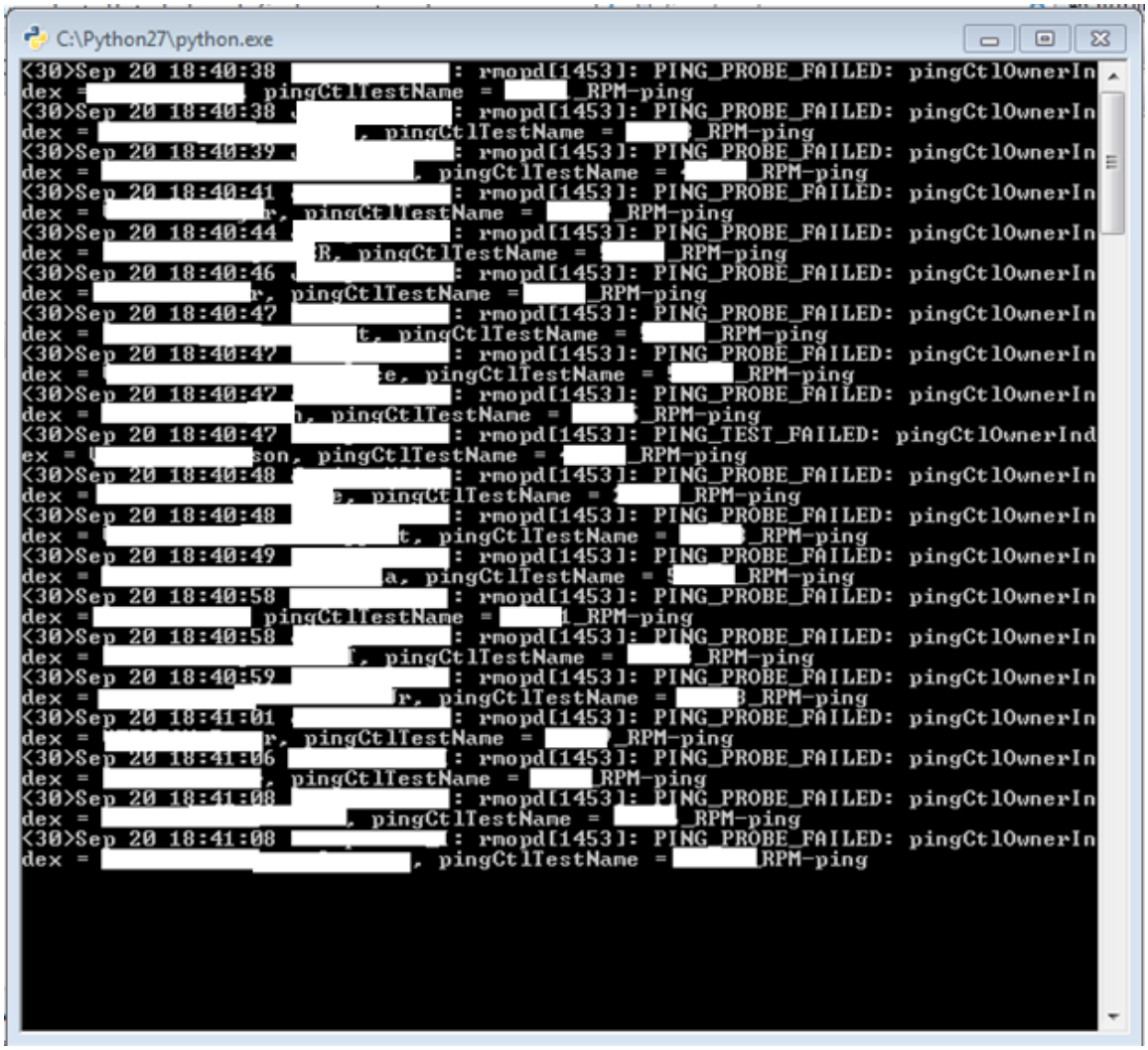


Figura 64. Recepción en tiempo real de syslog en el servidor realizado con Python.



4. Manejo de la información. El manejo de la información se utiliza para establecer un vínculo entre la recepción y la alerta al operador. No hay resultados que mostrar.

5. Alerta al operador. La alerta al usuario se puede observar en la Figura 65, se nota un pop-up en el centro de la pantalla, el cual nunca dejará de estar en el centro y encima de toda las ventanas hasta cerrarlo, ya sea con OK o la X. Para fines de ilustración se observa en la Figura 66 la alerta generada por la configuración RPM de prueba mencionada en el diseño experimental. Se debe mencionar que las Figuras 63 y 65 observan muchas instalaciones con problemas debido a que se tomaron durante una falla general en cierto país, este no es el comportamiento normal de este sistema de monitoreo.

Figura 65. Alerta del servidor syslog en Python al usuario (se censura por confidencialidad).

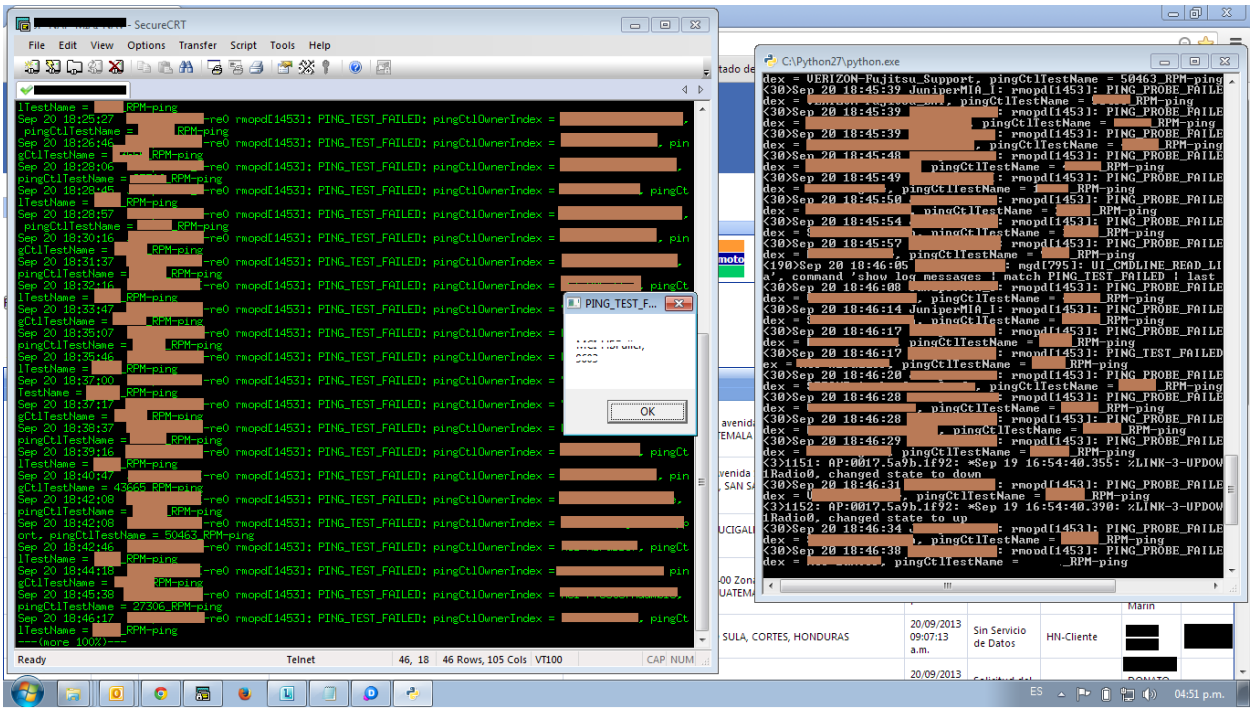
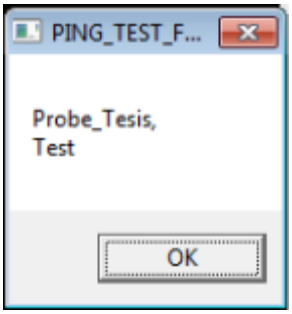


Figura 66. Alerta del servidor Python por RPM de prueba para fines de ilustración.

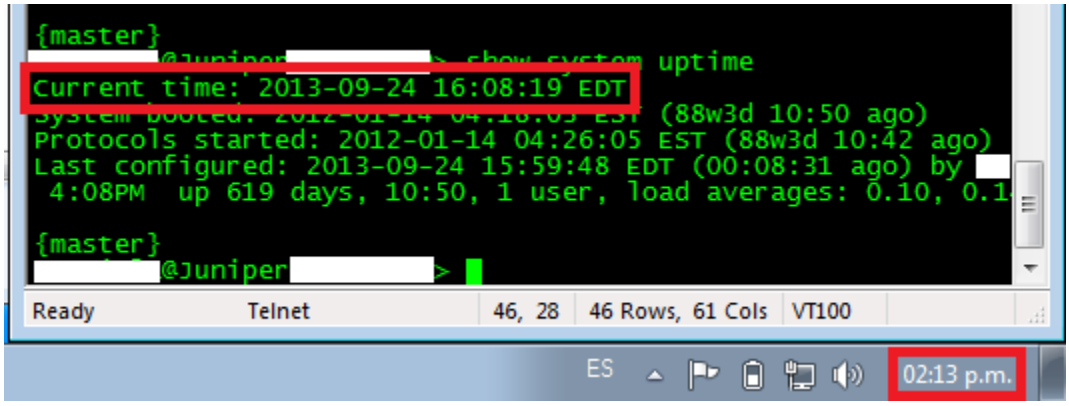


F. RESULTADOS DE FUNCIONAMIENTO DE LA ALERTA AL OPERADOR

Esta sección demuestra el funcionamiento correcto del propósito del proyecto, en tiempo real. Se configuró el test RPM de prueba discutido en el Diseño Experimental, dicho test RPM hereda los parámetros de la configuración observada en la Figura 50, en la sección que se discuten los resultados de las configuraciones. Más a detalle, este test enviará 9 pings ICMP a la IP target, cada ping se envía cada 20

segundos, lo que nos da un total de 3 minutos. En otras palabras, luego de 3 minutos, si existen 9 resultados no exitosos el RPM indicará que se falló el test generando un syslog PING_TEST_FAILED. Se debe mencionar que para esta prueba se utiliza la computadora del operador. Ya que es importante notar los tiempos, en la Figura 67 se observa la hora del equipo en Miami y la hora de la computadora personal cada una marcada por un cuadro rojo.

Figura 67. Hora del equipo en Miami y computadora personal de operador.



En la Figura 68 podemos observar los resultados históricos del test RPM, la línea roja observada separa los pings de un test realizado, como se puede notar son 9 las líneas arriba de la línea roja, y cada una sucede cada 20 segundos, así como está indicado en la configuración el parámetro probe-interval 20, cada uno de estos pings fallados debe generar un syslog PING_PROBE_FAILED como se determinó en una sección anterior del trabajo. Se debe tomar en cuenta que a las 16:02:48, hora del equipo en Miami, se falla el noveno ping, lo que debe causar el mensaje PING_TEST_FAILED. Este resultado esperado se puede observar en la Figura 69, por cada ping fallado se genera un syslog PING_PROBE_FAILED, y, efectivamente, a las 16:02:48 se genera el PING_TEST_FAILED esperado.

Figura 68. Resultados históricos del RPM para la prueba de alerta al operador.

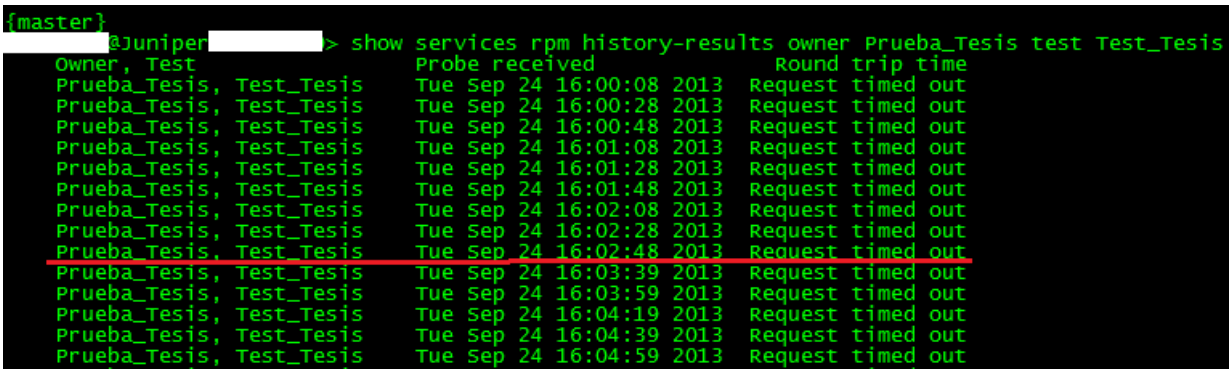
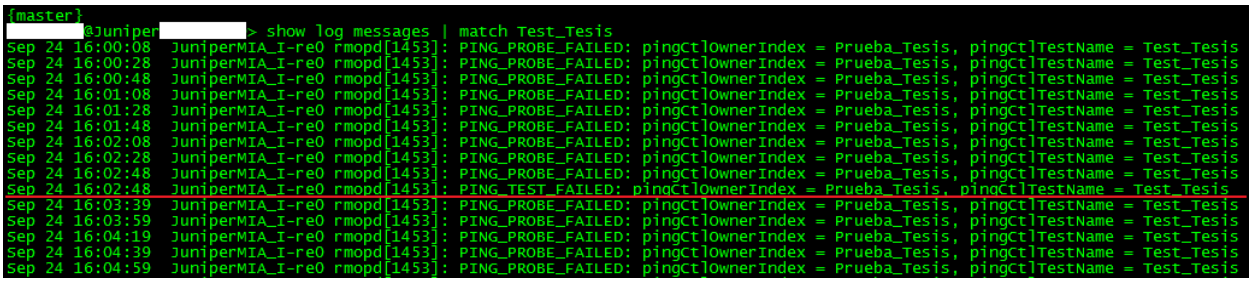


Figura 69. Log del equipo en Miami para la prueba de alerta al operador.



Durante esta prueba se mantuvo el servidor en Python corriendo en la computadora del operador, se puede observar la ventana de los syslog desplegados en la Figura 70, donde se encuentran los mismos logs que en la Figura 69. Nuevamente la línea roja nos muestra la separación de un test de otro, y podemos observar el syslog indicando PING_TEST_FAILED a las 16:02:48, hora del equipo. En la Figura 71 se muestra como la alerta aparece en el centro de la pantalla y por encima del resto de ventanas a las 02:08, hora de la computadora del operador, lo que coincide con las 16:02:48 del equipo. En la Figura 72 se observa como la alerta al operador permanece hasta cerrarla.

Figura 70. Syslog recibidos por servidor en Python durante prueba de alerta al operador.

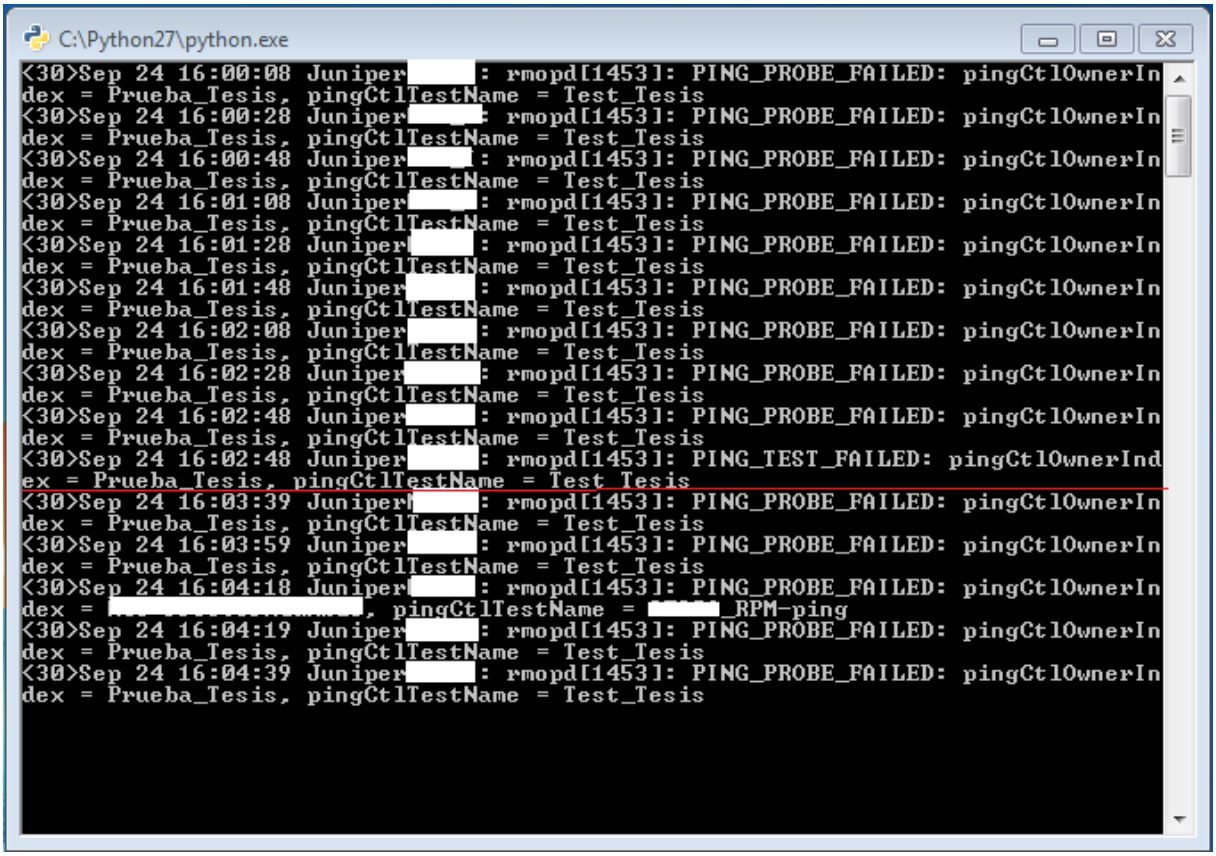


Figura 71. Pop-up resultado de la prueba de alerta al operador.

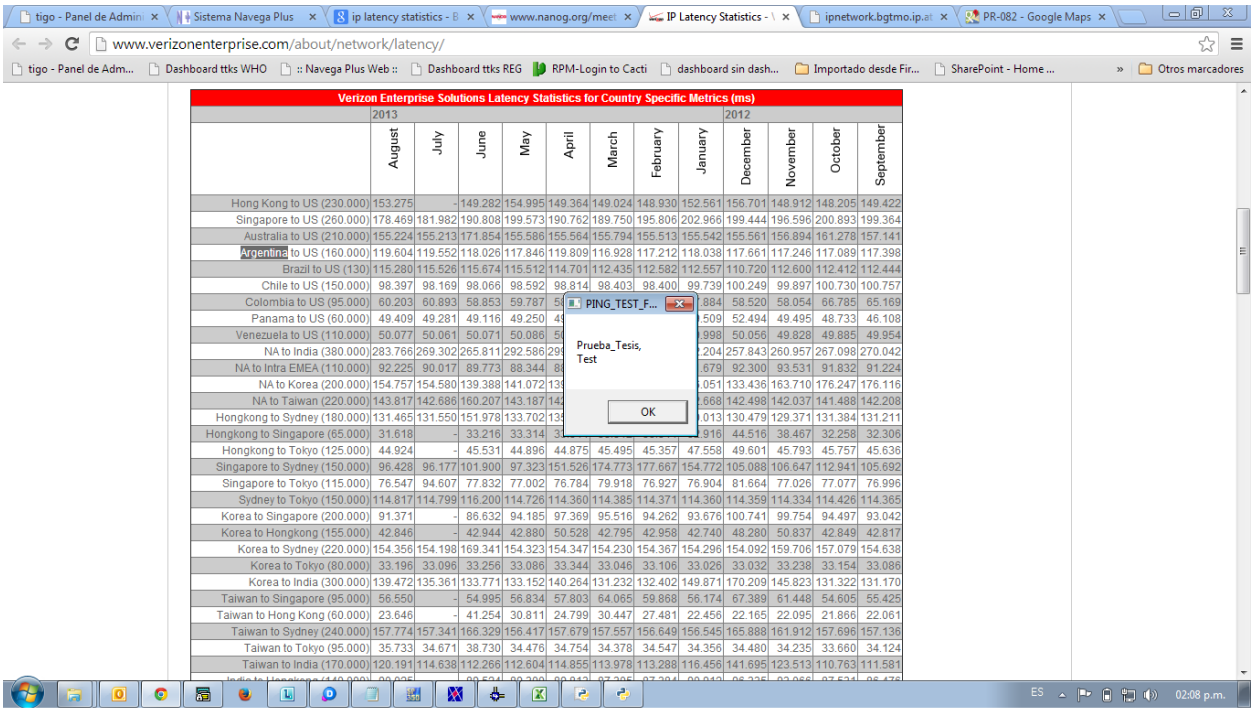
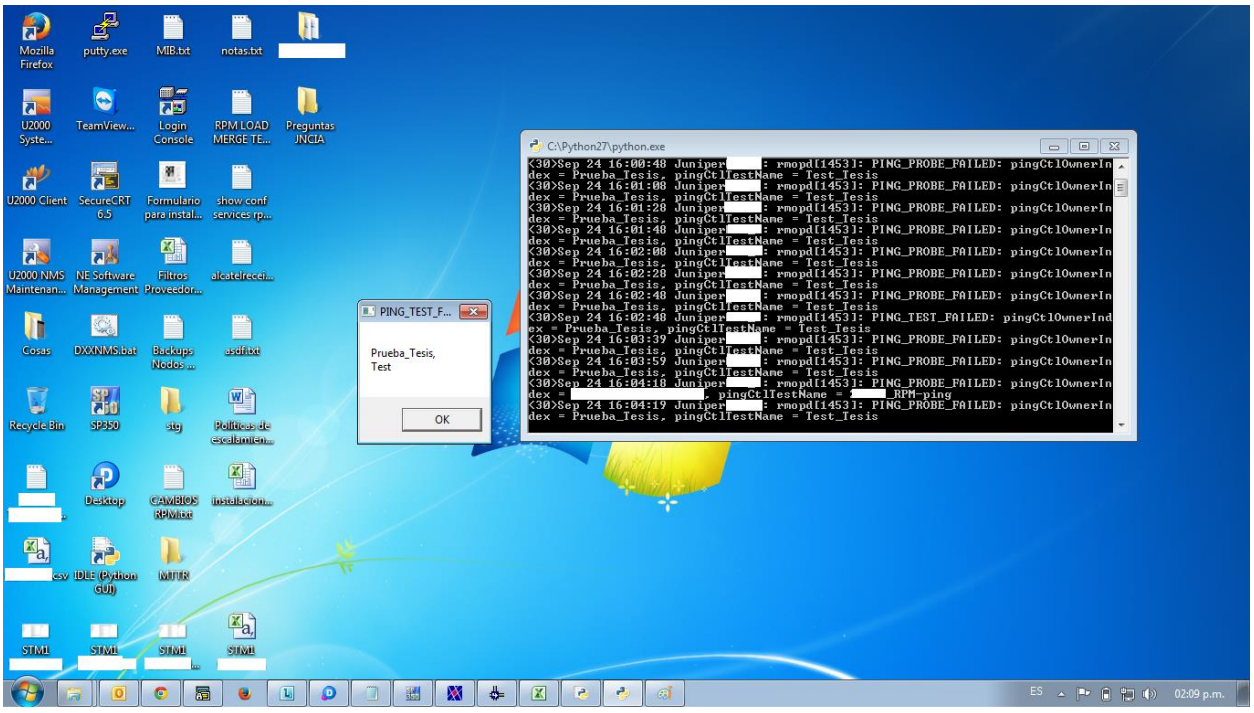


Figura 72. Segunda imagen del Pop-up resultado de la prueba de alerta al operador.



VII. DISCUSIÓN

El objetivo principal de este trabajo fue implementar un sistema con la capacidad de supervisar enlaces IP-VPN de manera que al presentarse un incidente en alguno de ellos, notificará al operador del Centro de Operaciones de Red sobre el evento. A continuación se discutirán más a detalle los resultados obtenidos al realizar este proyecto.

Antes de iniciar el desarrollo del sistema se tomaron en cuenta los recursos y la capacidad del equipo en la red, por ejemplo, en el inciso A de la sección de Resultados se logró identificar que la configuración del RPM en este equipo core no afectaba su desempeño. Este tipo de pruebas previas a la implementación son necesarias y, hasta cierto punto, obligatorias, debido al riesgo que se toma alterando un sistema de esta importancia, del cual dependen no solamente los servicios de clientes mayoristas, sino el resto de clientes y consumidores normales. Es importante también notar como se llevaron a cabo las configuraciones en el inciso C, se realizaron de tal manera que se habrían notado cambios en el desempeño del equipo en tal momento que la afectación habría sido mínima. Afortunadamente lo que se logró observar fue la nula alteración en el desempeño.

En el inciso B, de la sección de Resultados (de ahora en adelante la referencias serán únicamente a esa sección del trabajo), se determinaron dos aspectos vitales para el proyecto: los parámetros de configuración para los test y la nomenclatura, distribución y organización de los mismos. Conocemos que un test en RPM consta de un número específico de probes, y estos a su vez se envían cada cierto tiempo. Se logró determinar el valor de la cantidad y el tiempo de separación entre cada probe, de tal manera que al fallar el test podríamos estar seguros que el servicio se encontraba caído, a la vez que no se podía permitir que el test tomara mucho tiempo ni recursos de la red. Entonces la decisión fue 9 pings distribuidos normalmente en un lapso de 3 minutos, esto resultó positivo al no observar aumento en uso de recursos, y al fallar la prueba se garantiza que el servicio esta fuera. La herencia de configuración fue otro aspecto importante de esta sección, el cual tomó trabajo optimizar, hasta el punto de lograr una sola línea de configuración por lo que habrían sido cientos.

La decisión de la manera de generar una alerta al operador fue probablemente una de las más conflictivas, ya que la idea inicial fue cambiada algunas veces y no fue la finalmente implementada. Inicialmente se deseaba alertar al usuario haciendo uso de un trap del protocolo SNMP. Más adelante se fue considerando el hecho que existían opciones más simples y directas, tales como utilizar syslog. La decisión se tomó debido a que con el trap SNMP no se obtenía directamente el cliente y el identificador del servicio, se obtenía un OID y se debía realizar la obtención de los datos requeridos. Siendo una característica de esta aplicación la simplicidad, y el objetivo que se requería era únicamente indicar estos dos datos al operador, se desaprovechaban las ventajas que traería un trap SNMP y se optó por utilizar syslog, lo cual resultó en

una manera simple y directa de obtener los datos, y permitió tener un histórico detallado en un log. Finalmente se obtuvo un servidor de syslog capaz de: recibir mensajes syslog, guardarlos y alertar al usuario cuando se cae un enlace. Cabe mencionar que el proyecto realizado si genera traps de SNMP, pero estos son para un uso futuro en otro proyecto que se implementará.

Con respecto al servidor Cacti para las gráficas de datos presentes e históricos, cuyos resultados se encuentran en la sección D, la decisión fue sencilla debido al hecho de que es una plataforma muy completa, simple y adicionalmente es freeware. Puede ser curioso el hecho de desear tener un sistema que actúe en tiempo real y a la vez tener una gráfica que nos pueda mostrar el comportamiento histórico. La razón de esto es para no descuidar ninguno de los dos aspectos, en este tema es importante tanto el presente como el pasado, los clientes mantienen un monitoreo de sus enlaces, en muchos casos no les alerta de una caída en tiempo real, pero tienen acceso a la información histórica del circuito, y se debe poder corroborar si el inconveniente que el cliente observó fue a causa del tramo del enlace responsabilidad nuestra, o si el tramo permaneció operativo durante ese incidente.

El planteamiento del sistema de monitoreo optó por la simplicidad desde un principio, ya que este sistema realiza un ping desde un extremo del servicio hacia el equipo del cliente y de esta manera certifica la conectividad punto a punto. Esto conlleva una gran ventaja, ya que el enlace puede utilizar diferentes tecnologías de transporte, como SDH o DWDM, o enlaces capa 2 para conectar entre el CE y el PE, y el monitoreo especial para este tipo de circuitos es más específico y complicado, pero con este sistema se logra unificar todos estos tramos y certificar su operatividad con una misma prueba, ya que todo este transporte es transparente el enlace IP-VPN, siendo este un servicio capa 3.

Cabe mencionar que la finalidad de esta aplicación es el monitoreo de la conectividad, lo que nos indica que este sistema se encarga de una parte crucial para el enlace, y se puede considerar como el centro para un sistema de monitoreo con más detalles. Estos detalles podrían ser las sesiones BGP, las cuales pueden estar caída aún en el caso de tener completa conectividad, es obvio que no es un caso usual, es más, es una situación esporádica, pero se considera como una posibilidad. También debe mencionarse que estos tipos de monitoreo consumen más recursos del equipo, lo cual se evitó durante la realización de este trabajo.

Durante la redacción de este trabajo se tuvieron ciertas dificultades. Entre estas se encuentra el aspecto de la información confidencial manejada por el carrier, en este trabajo no se mencionan nombres ni números que puedan comprometer a las partes involucradas. Esto talvez pueda evitar que la importancia del trabajo destaque a gran escala, pero de igual manera se mantiene la visión de la importancia del sistema. Otra dificultad experimentada fue la representación de ejemplos en los cuales este sistema entró en acción y asistió en la proactividad para reparar una falla, esto viene de la mano con la censura de los datos

confidenciales, ya que en cualquiera de los casos ocurridos, no se pueden mencionar detalles de los mismos.

Una de las ventajas de la manera en la que se desarrolló la implementación fue que se consideró a futuro, se realizó de tal manera que se tiene opción a la escalabilidad, ya que agregar un nuevo servicio al monitoreo es sumamente sencillo, es literalmente un par de líneas de configuración, las cuales serían la IP del CE y el routing-instance respectivo, ya que los parámetros en común se lograron agrupar de manera que se puedan heredar a cada monitoreo nuevo como se puede observar en la sección C. Se debe mencionar que la cantidad de tests que puede soportar un mismo equipo se desconoce, tiene capacidad para cientos de ellas, pero en realidad cuantas puede llegar a soportar es algo que no se ha logrado determinar, afortunadamente aún se tiene capacidad para un número significativo de servicios.

Es necesario prestar atención a lo logrado con este proyecto, lo cual se ha mencionado varias veces en este trabajo, y es el demostrar al cliente que se posee control de la red, que se presta un servicio y también una atención proactiva, que se puede despreocupar hasta cierto punto. En los casos de incidentes en estos enlaces ya no debe ser el cliente quien reporte al carrier lo que es de su responsabilidad. Utilizando este sistema también se aumenta la disponibilidad de la red, ya que se inicia el diagnóstico y reparación casi inmediatamente al caer el servicio.

VIII. CONCLUSIONES

Al finalizar este trabajo se obtuvo un sistema funcional, rápido y sencillo. Se puede observar la simplicidad y efectividad de su funcionamiento en la sección de Resultados. Se puede concluir que los objetivos establecidos inicialmente fueron cumplidos. Estos servicios seleccionados cuentan con un monitoreo proactivo, el cual ayuda a reducir la cantidad de tiempo que los mismos permanecen caídos.

Cabe mencionar que con la implementación este sistema se le está agregando valor al servicio, de esta manera se aumenta la satisfacción del cliente, lo cual es sumamente importante y respalda los objetivos de la realización de este proyecto.

Asimismo los enlaces cuentan con gráficas históricas respaldando dicho monitoreo proactivo y, a la vez, ayudando a mantener un control más extenso del comportamiento de los circuitos. Se recalca el hecho de tomar en cuenta la operatividad presente del enlace así como la histórica como una obligación para lograr un sistema completo.

Se debe concluir la importancia de la cantidad de recursos utilizados para implementar este proyecto, es decir, la inexistencia de inversión económica, y a la vez, la permanencia del desempeño óptimo de los equipos y enlaces de la red. En otras palabras, no se utilizaron recursos económicos de la compañía, ni de la red.

Es justo mencionar la dificultad que se tuvo durante la elaboración de este trabajo para demostrar la importancia de la aplicación de un monitoreo de este tipo de una manera más directa, lo que complica concluir en su necesidad de la misma en una red de este tipo al tener que mantener confidencial toda información sobre los enlaces tratados.

IX. RECOMENDACIONES

Se recomienda evaluar detenidamente si el valor que se agregará a los servicios será notado por el cliente, quien debe ser el mayor beneficiado. De esta manera se logra decidir si el proyecto deberá ser realizado, o proceder a cambiar el planteamiento inicial.

Se recomienda el análisis extenso de la topología de la red del carrier y los servicios que se desean monitorear antes de iniciar el diseño. Al realizar esto es posible identificar formas más simples y directas de monitorear los servicios sin interacción entre más plataformas.

Se recomienda tener siempre presente la importancia de la escalabilidad de la aplicación y la facilidad de la adición de futuros servicios a la misma. Al tomar esto en cuenta se asegura un largo tiempo de vida del sistema y también el uso.

Se recomienda fomentar la importancia actitud proactiva al y la necesidad de la misma con el cliente con el equipo encargado del monitoreo. De esta manera se logra aprovechar en su totalidad de la plataforma desarrollada.

Se recomienda mantener la organización y la nomenclatura de las configuraciones homogenizadas en toda la red, de igual manera la definición los puntos que realizarán pruebas. Esto con el propósito de lograr un fácil desarrollo y adaptación de futuras automatizaciones, así como la migración de los datos a una nueva plataforma.

Se recomienda la utilización de protocolos más seguros como SNMPv3 y SSL para el transporte de la información en la infraestructura. De esta manera se logra la mitigación de ataques activos o pasivos que puedan existir en la red del carrier.

X. BIBLIOGRAFÍA

(s.f.).

Blank, A. G. (2004). *TCP/IP Foundations* (1ra ed.). Hoboken: John Wiley & Sons.

Clemm, A. (2006). syslog protocol. En *Network Management Fundamentals.* Cisco.

Downing, M. (2013). *The importance of network monitoring*. Recuperado el 13 de Agosto de 2013, de http://www.animate.com/the-importance-of-network-monitoring/

Expedient Holding. (2004). *Ping and traceroute*. Recuperado el 22 de Septiembre de 2013, de http://help.expedient.com/general/ping_traceroute.shtml

Internet Assigned Numbers Authority. (20 de Septiembre de 2013). *Service Name and Transport Protocol Port Number Registry.* Recuperado el 25 de Septiembre de 2013, de http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

Jiménez-Zarco, A. I., & Torrent-Sellens, J. (2009). Orientación proactiva hacia el cliente, cooperación y uso de las TIC: un análisis empírico sobre sus interrelaciones y efectos como potenciadores de la innovación en producto. REV. INNOVAR.

Juniper Networks. (2009). JUNOS routing essentials, revision 9.b. California: Juniper Networks, Inc.

Juniper Networks. (2010). *Networking Fundamentals.* Recuperado el Enero de 2013, de Juniper Learning Portal: https://learningportal.juniper.net/juniper/user_activity_info.aspx?id=769

Juniper Networks. (2010a). *Real-time performance monitoring on juniper network devices.* California: Juniper Networks, Inc.

Juniper Networks. (2010b). Junos MPLS and VPNs. California: Juniper Networks, Inc.

Juniper Networks. (11 de Noviembre de 2011). *Juniper Technical Documentation*. Recuperado el Junio de 2013, de Configuring RPM Probes: http://www.juniper.net/techpubs/en_US/junos10.4/topics/task/configuration/rpm-probes-configuring.html

Juniper Networks. (2013). VPN review. En *JNCIS-SP study guide part 3* (págs. 157-170). California: Juniper Networks, Inc.

Juniper Networks. (2013a). Border gateway protocol. En *JNCIS-SP study guide part 1* (págs. 83-110). California: Juniper Networks, Inc.

Juniper Networks. (2013b). Open shortest path first. En *JNCIS-SP study guide part 1* (págs. 51-82). California: Juniper Networks, Inc.

Juniper Networks. (2013c). Layer 3 VPNs. En *JNCIS-SP study guide part 3* (págs. 171-192). California: Juniper Networks, Inc.

Juniper Networks. (2013d). MPLS fundamentals. En *JNCIS-SP study guide part 3* (págs. 9-38). California: Juniper Networks, Inc.

Juniper Networks. (2013e). *Configuring RPM Probes*. Obtenido de Juniper technical documentation: http://www.juniper.net/techpubs/en_US/junos10.4/topics/task/configuration/rpm-probes-configuring.html

Python Software Foundation. (2013). *socket — low-level networking interface*. Recuperado el 2 de Agosto de 2013, de http://docs.python.org/2/library/socket.html?highlight=socket%20server

Soricelli, J. M. (2003). Tutorial: introduction to MPLS. Salt Lake City: Juniper Networks, Inc.

Soricelli, J. M. (2006). *JNCIA study guide* (1ra ed.). Sybex, Inc.
Stallings, W. (1996). *SNMP, SNMPv2, and RMON: practical network management* (2da ed.). Michigan: Addison-Wesley Pub. Co.

XI. ANEXOS

A. CÓDIGO PARA POLLING DE OID EN CACTI

```
#!/usr/bin/perl

#Codigo mencionado el inciso D en seccion de Diseño Experimental

use strict;

my $node = $ARGV[0];
my $comm = $ARGV[1];
my $action1 = $ARGV[2];
my $action2 = $ARGV[3];
my $theindex = $ARGV[4];

my $snmppath = "/usr/bin/snmpwalk";
my $snmpget = "/usr/bin/snmpget";
my $version = "2c";
my $jnxRpmResCalcAverage = '.1.3.6.1.4.1.2636.3.50.1.3.1.5';
my $jnxRpmResCalcPkToPk = '.1.3.6.1.4.1.2636.3.50.1.3.1.6';
my @index;
my $owner;
my $test;
my @Tests;

sub indexes {
        my $queryOrIndex = shift;

        my @Index = `$snmppath -On -v $version $node -c $comm $jnxRpmResCalcAverage`;

        foreach (@Index) {

                if ($_ =~m/^\.1\.3\.6\.1\.4\.1\.2636\.3\.50\.1\.3\.1\.5\.(\d+).*\.1\.1 \=
Gauge32.*$/i){ #Encuentra la longitud de la cadena owner
                if ($_
=~m/^\.1\.3\.6\.1\.4\.1\.2636\.3\.50\.1\.3\.1\.5\.(\d+)((\.\d+){$1})\.(\d+).*\.1\.1 \=
Gauge32.*$/i){ #Encuentra la cadena owner y la longitud de la cadena test
                if ($_
=~m/^\.1\.3\.6\.1\.4\.1\.2636\.3\.50\.1\.3\.1\.5\.(\d+)((\.\d+){$1})\.(\d+)((\.\d+){$4})\.1\.1 \=
Gauge32.*$/i){ # Encuentra la cadena test
                        my $owner = oid_to_ascii($2);
                        my $test = oid_to_ascii($5);
                        if ($queryOrIndex eq "query"){
                                push @Tests , "$owner.$test:$owner.$test\n";
                        }
                        elsif ($queryOrIndex eq "index"){
```

```
                                        push @Tests, "$owner.$test\n" ;
                                }
                        }
                }
}
}
return @Tests
}
sub get_average {
        my $index_value = shift;
        my @values = split /\./,$index_value;
        my $lengthOwnerOid =  length ($values[0]);
        my $lengthTestOid =  length ($values[1]);
        my $ownerOid =  ascii_to_oid($values[0]);
        my $testOid = ascii_to_oid($values[1]);
        my $oidIndex =  $lengthOwnerOid . $ownerOid . '.' . $lengthTestOid . $testOid;
        my @rtt =  split (/ /,`$snmpget -Ov  -v $version $node -c $comm
$jnxRpmResCalcAverage.$oidIndex.2.1`);
        return $rtt[1];
}
sub get_jitter {
        my $index_value = shift;
        my @values = split /\./,$index_value;
        my $lengthOwnerOid =  length ($values[0]);
        my $lengthTestOid =  length ($values[1]);
        my $ownerOid =  ascii_to_oid($values[0]);
        my $testOid = ascii_to_oid($values[1]);
        my $oidIndex =  $lengthOwnerOid . $ownerOid . '.' . $lengthTestOid . $testOid;
        my @jitter =  split (/ /,`$snmpget -Ov  -v $version $node -c $comm
$jnxRpmResCalcPkToPk.$oidIndex.2.1`);
        return $jitter[1];
}

ARGUMENTS: {
        if ($action1 eq "index") { print indexes("index") ; last ARGUMENTS; }
        if ($action1 eq "query") { print indexes("query") ; last ARGUMENTS; }
        if ($action2 eq "average") { print get_average("$theindex") ; last ARGUMENTS; }
        if ($action2 eq "jitter") { print get_jitter("$theindex") ; last ARGUMENTS; }

        print "usage:\n\n./juniper-rpm.pl IP COMMUNITY index\n./juniper-rpm.pl IP
COMMUNITY query \n./juniper-rpm.pl IP COMMUNITY get {average|jitter} index_oid\n";
}

sub oid_to_ascii ($)
{
        # Convierte 8 bytes en hexadecimal a un ASCII
        (my $str = shift)  =~ s/([0-9]{1,3})/chr($1)/eg;
        my @noDotArry = split(/\./,$str);
```

```
        $str = join ('',@noDotArry);
        return $str;
}
sub ascii_to_oid($)
{
        my @chars = split(//,shift);
        my $oid = '';
        foreach (@chars){
                $oid .= '.' . ord ($_);
        }
        return $oid;
}
exit;
```

XII. GLOSARIO

ABR Area Border Router
AS Autonomous System
ASBR Autonomous System Border Router
BGP Border Gateway Protocol
CE Customer Edge
CPE Customer Premises Equipment
CPU Central Processing Unit
DNS Domain Name System
HTTP Hypertext Transfer Protocol
ICMP Internet Control Message Protocol
IP Internet Protocol
ISP Internet Service Provider
LSP Label Switched Path
LSR Label Switching Router
MIB Management Information Base
MPLS Multi-Protocol Label Switching
MTTR Mean Time To Repair
NOC Network Operations Center
OID Object Identifier
OS Operating System
OSPF Open Shortest Path First
PE Provider Edge
RPM Real-Time Performance Monitoring
SLA Service Level Agreement
SMTP Simple Mail Transfer Protocol
SNMP Simple Network Management Protocol
VPN Virtual Private Network
VRF VPN Routing and Forwarding