UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA

Tesis para optar al Título de Magíster en Ingeniería en Computación.

Virtualización de Redes en la Empresa.

Autor: Ing. Pablo A. Gestido
Director de Tesis: Dr. Ing. Eduardo Grampín

Montevideo, Uruguay
2014

Historial de Revisiones

Fecha	Versión	Descripción	Autor	Aprobado por
29/01/2014	1.0	Entrega para el revisor	Pablo Gestido	
16/03/2014	2.0	Entrega para el tribunal	Pablo Gestido	

Tabla de Contenido

Virtualización de Redes en la Empresa

1. Introducción

1.1 Motivación

Antes de definir por qué se necesita de la virtualización de redes en la empresa, debemos ver que significa que una red se encuentra virtualizada.

1.1.1 ¿Qué es una red virtualizada?

Hoy en día las empresas brindan servicios a distintos grupos de usuarios, que tienen distintas necesidades desde el punto de vista de "networking".
Una red virtualizada es un entorno de red lógico aislado para cada uno de estos grupos dentro de la empresa. Cada red lógica es creada sobre una única infraestructura de red física, dando servicio a cada uno de estos grupos. Desde el punto de vista de la experiencia de usuario, cada uno de éstos tiene la visión de estar conectado a una única red dedicada (seguridad, conjunto de políticas individuales, decisiones de ruteo, etc). Debido a todo lo anterior virtualizar una red implica la segmentación lógica del transporte de red, dispositivos de red y servicios de red [1], como se puede apreciar en la Figura 1.

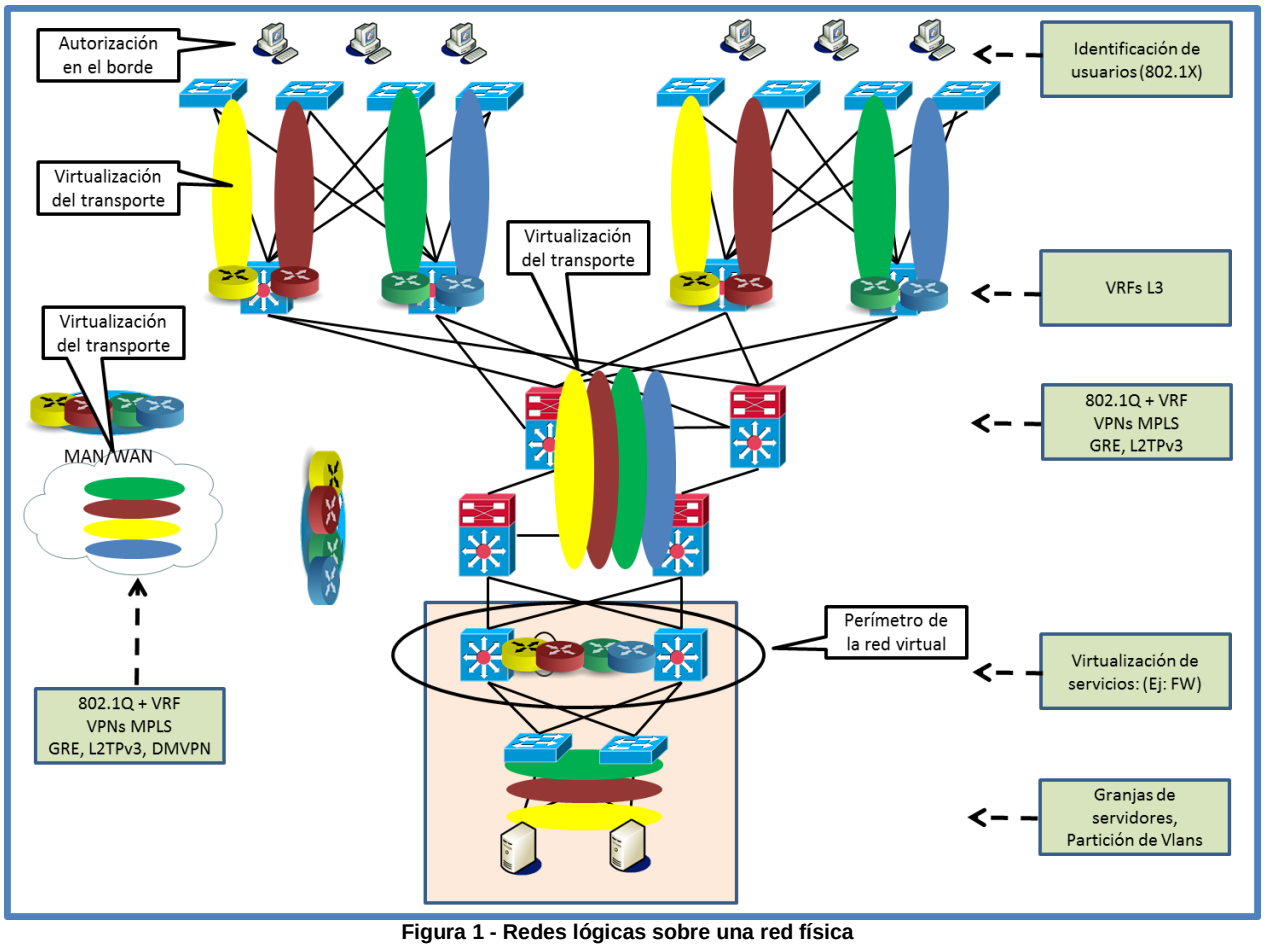

Figura 1 - Redes lógicas sobre una red física

1.1.2 ¿Qué impulsa la virtualización de redes?

Muchos factores pueden impulsar la virtualización de la red [1]; dentro de los más importantes se encuentran:

1. Reducción de costos.
   Esto permite que una única red física de soporte a múltiples usuarios y redes virtuales, con el consiguiente ahorro en dispositivos de red / servidores.

2. Mayor flexibilidad en acceso a los recursos y colaboración entre grupos de usuarios.
   Tal es el caso de una empresa donde distintos grupos de usuarios (empleados, visitantes, contratistas) acceden a los recursos y servicios de la red, pero lo pueden hacer desde distintas ubicaciones (no se encuentran limitados a una única ubicación física). En el otro extremo existen empresas donde en un campus común residen varios clientes. Como ejemplo de este último caso pueden ser aeropuertos o parques tecnológicos.

3. Simplifica la operación, administración y gerenciamiento (OAM)
   Mediante la virtualización se reduce la cantidad de dispositivos en la red, a ser gerenciados o monitoreados.

4. Alta disponibilidad.
   Al utilizarse técnicas de virtualización mediante la utilización de “clusters” (o “device pooling”), varios dispositivos se comportan como un único dispositivo con lo cual se logra aumentar considerablemente el “uptime” (con disminución de costos de administración y de operación al disminuir la cantidad de dispositivos).

5. Nuevos modelos de negocios.
   La naturaleza dinámica de los entornos empresariales actuales necesita frecuentemente la creación de equipos de trabajo virtuales. Estos equipos pueden estar compuestos por individuos pertenecientes a muchos grupos de dentro y fuera de la empresa. Los equipos de trabajo virtuales suelen crearse para realizar un proyecto específico, el cual puede ser a largo plazo o eventualmente muy corto. Un ejemplo de este tipo de equipos es la creación de un equipo de trabajo para despachar un vuelo en un aeropuerto para una compañía dada. Se deben crear los accesos a los recursos privados y compartidos de la línea aérea. Luego que el vuelo fue despachado, otra línea aérea podría usar el mismo equipamiento pero accediendo a su conjunto de recursos. El departamento de IT debe dar los permisos necesarios para crear bajo demanda el entorno de trabajo de ambas aerolíneas de tal manera que cada una tenga la “sensación de usuario” que los recursos son exclusivos.

6. Parques industriales con múltiples clientes.
   Los parques industriales proveen espacio de oficinas a varias empresas dentro de un mismo espacio físico equivalente a un campus. Las empresas alquilan el espacio físico y la infraestructura de red junto con servicios de telefonía, vigilancia, granjas de servidores, etc. Instalar un Data Center dedicado para cada empresa es extremadamente caro.

7. Datacenter virtualizados.
   En necesario su uso en escenarios de Datacenters donde existe uso intensivo de la virtualización de servidores / software (por ejemplo en ambientes de Cloud Computing) ya que es altamente común que los clientes mediante un portal de autoservicio puedan generar o expandir una infraestructura compuesta por servidores, redes locales, firewalls, rangos de direcciones IP, almacenamiento, sistemas operativos, software, etc. Para dar respuesta a tales requerimientos bajo demanda de manera ágil y eficaz, se debe utilizar tecnologías de virtualización de servidores como de redes. Dentro de la categorización de servicios de Cloud Computing, existe una llamada “Network as a

service" (o NaaS por sus siglas en inglés). Esta es una categoría de servicios de Cloud (nube) donde la capacidad principal que se brinda al usuario es el poder utilizar servicios de red y/o transporte así como servicios de conectividad entre distintas Clouds (nubes). Los servicios NaaS (Network as a Service – Red como Servicio) tradicionales incluyen el despliegue de VPNs y la asignación de ancho de banda bajo demanda.

1.1.3 Requerimientos de negocio y técnicos de una red empresarial virtualizada

Las empresas se enfrentan con desafíos técnicos (por lo general derivados de las necesidades del negocio) cuando intentan aplicar técnicas de virtualización de redes. Por ello se debe:

- Garantizar la privacidad entre grupos.
- Asignar los usuarios a grupos utilizando algún tipo de autenticación.
- Garantizar la seguridad para cada grupo de manera independiente.
- Asegurar mecanismos seguros de colaboración entre los grupos.
- Proveer servicios básicos de red para los distintos grupos.
- Proveer dominios de ruteo independiente y espacio de direccionamiento para cada grupo.
- Proveer extensión sobre la WAN para cada grupo.
- Implementar mecanismos de tarifación y contabilidad para cada grupo.
- Garantizar la aplicación de políticas de seguridad independientemente de la ubicación del usuario en la red.

Para implementar estos requerimientos, se puede utilizar diversas tecnologías. Alguna de ellas simples y otras con niveles de complejidad mayor. Las distintas necesidades de negocio van a demandar distinto grado de conectividad y/o diferente subconjunto de requerimientos. Como regla general cuando se superponen las distintas necesidades del negocio, resulta en una solución tecnológica más compleja. Cuando esto sucede, las tecnologías simples no escalan o no pueden proveer la conectividad adecuada.

1.2 Problema a resolver

El presente documento pretende describir el estado del arte de la virtualización de redes en la empresa para luego generar una clase de guía de diseño para orientar a los diseñadores de este tipos de redes qué tecnologías utilizar en los casos más comunes. Asimismo, mediante un caso de estudio se aplicarán estas guías de diseño para resolverlo mediante tecnologías cubiertas en el desarrollo del documento.

1.3 Alcance

El alcance del presente documento es presentar los distintos estándares, tecnologías y técnicas de virtualización de redes visto del lado de la empresa. Si una tecnología corresponde al dominio de la empresa o del proveedor de servicios es una línea gris, ya que depende de cuales aspectos la empresa quiere delegar (contratar) a un proveedor o implementar con sus recursos propios.
En el presente documento presentaremos los aspectos más relevantes de las tecnologías de virtualización de redes para luego presentar un caso de estudio donde se aplican tecnologías presentadas.

Para los ejemplos y/o casos de estudios que se presenten se tomará como referencia la línea de comando / configuración de equipos Cisco [2]. La motivación es que existen emuladores de estos equipos donde se pueden probar estas tecnologías en una escala casi 1:1. Por otro lado, Cisco siempre ha sido una empresa de vanguardia en "networking" y muchos estándares presentes hoy en día han surgido en una primera instancia como protocolos propietarios desarrollados por la empresa. Si bien el caso de estudio se puede implementar sobre una

plataforma de código abierto (p.e. Quagga [3]) se tomó la decisión de utilizar la primera opción ya que es un escenario concreto de equipamiento que una empresa podría utilizar en la implementación de una solución de virtualización.

1.4 Organización del documento

El presente documento se divide básicamente en cuatro partes.

En la primer parte del presente documento (capítulos 3 al 8) se estudian distintas alternativas de técnicas y tecnologías de virtualización de redes. En aquellos capítulos donde se detallan distintas tecnologías, al final del mismo se confecciona una tabla resumen con las principales características de los mismos, donde se puede discernir en cual caso de uso se ajusta mejor cada tecnología.

- Capítulo 3
  Se explican las diferencias en el diseño de una red tradicional vs una red virtualizada tanto en un entorno LAN / MAN como en un entorno WAN.

- Capítulo 4
  En este capítulo se cubren las distintas tecnologías utilizadas de base para la virtualización de redes.

- Capítulos 5
  En este capítulo se utilizan las tecnologías presentadas en el capítulo anterior y se proponen arquitecturas de virtualización como ser las VPNs tipo “Hop to Hop”, “overlay” y las opciones descritas en el RFC 2547 [4] (y trabajos derivados).

- Capítulo 6
  En este capítulo se describen las distintas opciones, desafíos y buenas prácticas para extender la red virtualizada de la empresa sobre la WAN.

- Capítulo 7
  En este capítulo se describe las “Software Defined Networks” (SDN), sus tecnologías asociadas, sus componentes y se describen los productos relacionados más importantes del ámbito académico y comercial.

- Capítulo 8
  En este capítulo se describe la virtualización de redes en el entorno del Datacenter, las estrategias de diseño y las maneras en que las distintas tecnologías existentes nos brindan apoyo para lograr la virtualización de la red y los servicios. Como un caso especial, se exploran las redes unificadas (datos y almacenamiento) que también nos brindan un grado de virtualización.

En la segunda parte (capítulo 9) se presenta un caso de estudio en el cual se aplican las técnicas vistas en los capítulos anteriores y se realiza una maqueta sobre una plataforma de emulación de dispositivos de red. Sobre esta maqueta, se toman distintas métricas las cuales se analizan en detalle en los anexos.

- Capítulo 9
  En este capítulo se presenta un caso de estudio en el cual se aplican las técnicas vistas en los capítulos anteriores y se realiza una maqueta sobre una plataforma de emulación de dispositivos de red (GNS3 [5]). Sobre esta maqueta, se toman distintas métricas las cuales

se analizan en detalle en los anexos (Capítulo 12).

En la tercera parte (capítulo 10) se exponen las conclusiones generales del trabajo que abarcan el estudio de las distintas tecnologías, así como del caso de estudio desarrollado.

- Capítulo 10
  En este capítulo se presentan las conclusiones generales del trabajo junto con un detalle de lo trabajado para la confección del presente documento así como los posibles trabajos a futuro.

En la cuarta parte (capítulos 11 y 12) se encuentra la bibliografía utilizada y anexos con la configuración de los dispositivos utilizada en el caso de estudio y distintos análisis realizados sobre el caso de estudio.

- Capítulo 11
  En este capítulo se presentan las referencias utilizadas.

- Capítulo 12
  En este capítulo se presentan los Anexos, compuestos con información detallada tanto de configuración como de distintos análisis del caso de estudio presentado en el Capítulo 10.

2. Resumen

Las empresas disponen de distintos grupos de usuarios con necesidades específicas. Muchas de las diferencias entre ellos se traducen en requerimientos específicos de “networking”. Dentro de la misma empresa, estos requerimientos suelen ser tan diferentes, que los diferentes grupos deben ser tratados como clientes distintos por el departamento de TI. A medida que el número de grupos aumenta, mantenerlos separados y seguros es un gran desafío.

El término virtualización es usado en varios contextos tales como virtualización de servidores, aplicaciones, dispositivos de almacenamiento e infraestructura de redes. La virtualización de redes delinea la virtualización de arquitecturas, tecnologías y técnicas correspondientes a la infraestructura de las mismas.

La virtualización de redes en la empresa apunta a resolver situaciones que aseguren una óptima utilización de los recursos existentes, tales como

- Reducción de costos.
- Simplificación de tareas operativas, de administración y gerenciamiento.
- Alta disponibilidad
- Creación de nuevos modelos de negocios.
- Instalación de parques industriales con múltiples clientes.
- Organización de Datacenters virtualizados.

En este sentido se debe proveer guías de diseño para las redes empresariales virtualizadas teniendo en cuenta las distintas tecnologías y los requerimientos de negocio a los cuales debe atender. Asimismo, un aspecto fundamental a considerar es la interacción con el Proveedor de Servicio de conectividad, a los efectos de garantizar tanto el Nivel de Servicio (especificado en mediante Niveles de Servicio – SLA, OLA) como el nivel de seguridad apropiado.

A lo largo del presente documento se estudia el estado del arte de la virtualización de redes, realizando un desglose de las diferentes tecnologías utilizadas para su implementación. Tanto en las tecnologías tradicionales de “networking” como en las nuevas tendencias, se pude observar una mezcla bastante compleja, donde existen muchas opciones disponibles para el diseñador / implementador de redes virtualizadas. Uno de los objetivos que se cubre en este trabajo es intentar plantear escenarios concretos junto con las tecnologías y técnicas que se entienden las más apropiadas para resolverlos. En ese sentido se proveen como forma de resumen, tablas que pueden utilizarse como guías para poder resolver un escenario particular. Estas son utilizadas en el caso de estudio donde se aplican técnicas de virtualización de redes para resolver un escenario dado con ciertas restricciones.

3. Arquitectura virtual de una red empresarial

3.1 Introducción

En el presente capítulo, se presentan conceptos básicos del diseño de redes en ambientes LAN y WAN para luego proceder a compararlo con el diseño de redes virtuales. Para la confección de este capítulo se tomó como base el capítulo 2 de [1], [6] y el aporte de otras fuentes.

3.2 Diseño de una red de Campus

3.2.1 El diseño de una red de campus tradicional

Se define como una red de campus una red de computadoras que conecta redes de área local a través de un área geográfica limitada, como un campus universitario, o una base militar. Además, todos los componentes, incluyendo conmutadores, enrutadores, cableado, y otros, le pertenecen a la misma organización. El método de diseño recomendado para el diseño de redes de campus es uno que es jerárquico y modular. Esta técnica de diseño genera un diseño de red altamente tolerante a fallos, escalable y fácil de mantener.

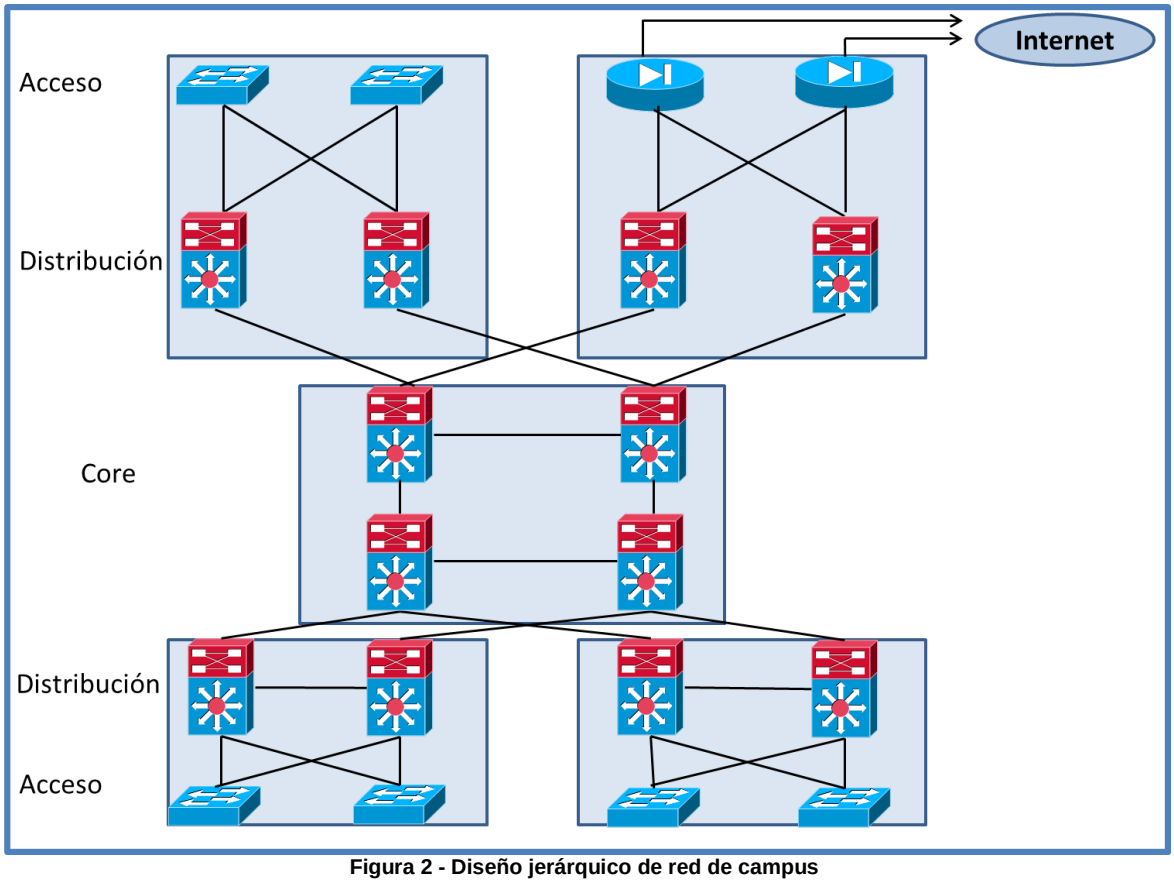

Figura 2 - Diseño jerárquico de red de campus

Según el diagrama de la Figura 2, en cada capa el tráfico es agregado de manera jerárquica desde la capa de acceso en la de distribución y luego en la de “core” mediante vínculos y equipamiento redundante, lo cual provee a la red con caminos redundantes y simétricos, los cuales optimizan la convergencia de los protocolos de red.

La modularidad se logra conectando cada dispositivo de acceso a 2 dispositivos de distribución y cada equipo de distribución a 2 equipos de “core”. Para la conexión hacia los routers de la WAN, Internet o hacia el Datacenter, puede utilizarse alguno módulo de distribución. La gran

ventaja de este esquema es el aislamiento de fallas ya que generalmente la capa de acceso son dispositivos de L2 y los dispositivos a partir de la capa de distribución son L3. Los dispositivos de "core", mediante la sumarización de rutas previene que un cambio una porción de la red cause la reconvergencia de todo el dominio ruteado. En este esquema el equipamiento en cada capa debe asumir distintos roles para el funcionamiento de la red. Por ejemplo:

- Switch de Acceso
  - In-line Power (POE)
  - Marcas de QoS trust boundary
  - Mecanismos de autenticación

- Switch de Distribución
  - Root bridge del Spanning Tree Protocol
  - First Hop Router
  - Punto de aplicación de "quality of service" (QoS)

- Switch de Core
  - Sumarizar las rutas
  - Manejar un IGP

En este diseño, una falla puede afectar a un módulo pero no se debe permitir su propagación en el core de la red. Prevenir la propagación de fallas no solo depende del diseño jerárquico (terminación de dominios L2), sino también depende en el despliegue de los protocolos de red utilizados ya que la configuración de un IGP que presenta el "core" como una sumarización de prefijos de red presentes en cada módulo de distribución es menos propenso a sufrir una reconvergencia global que uno sin las consideraciones de ruteo jerárquicos.

3.2.2 El diseño de una red de campus virtualizada

Cuando se virtualiza una red de Campus, se deben seguir las mismas pautas de diseño que para una red de campus no virtualizada. Para mantener todas las características ya vistas, se debe mantener un "core" ruteado. Como herramientas de virtualización se deben utilizar VLANS L2 para las capas de acceso y distribución y VPNs (L3) en el "core" ruteado de la red. Esta combinación de herramientas, aparte de mantener las características de la red de campus (escalabilidad, tolerancia a fallos, etc) crea una serie de redes "end-to-end" superpuestas en capas.

3.3 Diseño de una red WAN

3.3.1 El diseño de una red WAN tradicional

La "WAN" (Wide Area Network) provee conectividad entre los distintos sitios de una empresa. Generalmente en el contexto de una empresa el diseño de una WAN involucra la agregación de algunas sucursales con la casa matriz en un esquema llamado "hub-and-spoke", donde todas las sucursales tienen conexión directa con la casa matriz. Lo más común es que el Datacenter se encuentre cercano a la casa matriz.

Dependiendo los servicios de conectividad adquiridos por la empresa a un proveedor de servicios, existen distintas opciones en lo que refiere a la agregación de tráfico y seguridad. Los distintos tipos de servicios WAN pueden ser categorizados como servicios públicos o servicios privados.

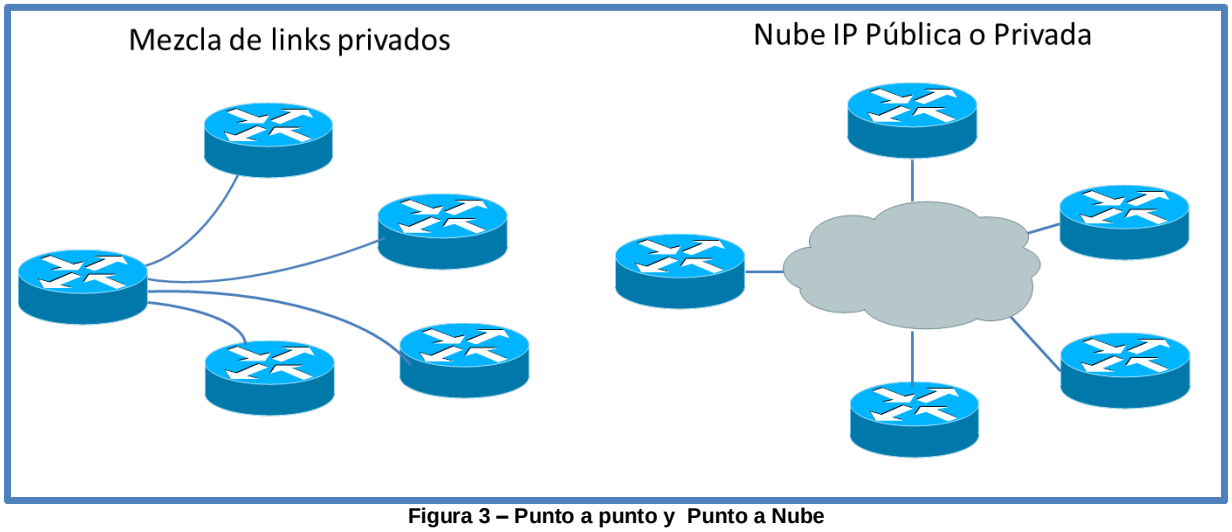

Figura 3 – Punto a punto y Punto a Nube

En general, los servicios privados del tipo punto a punto son percibidos como seguros y que no requieren el cifrado de datos. Como contraste, los servicios públicos como Internet requieren un conjunto de circuitos lógicos superpuestos para formar VPNs sobre la WAN compartida.

Aparte de Internet, los proveedores ofrecen la posibilidad de contratar servicios de WAN privada del tipo punto a nube. Este tipo de servicios se denomina IP VPN. En teoría este servicio no debería de requerir cifrado, sin embargo, algunos clientes piensan que las IP VPNs no son tan seguras como los circuitos privados virtuales y exigen el cifrado de datos, lo cual los hacen parecer a los servicios del tipo punto a nube implementados sobre Internet (ver Figura 3).

La arquitectura de una WAN debe ser jerárquica, modular y tolerante a fallos. De manera análoga al principio de jerarquía en la red de campus, el rol que cada dispositivo cumple depende de la posición en la jerarquía que este ocupa. Las capas que se definen en la WAN son similares a las que se definen en la red de campus: acceso, distribución y “core”.

Los routers de acceso en las sucursales proveen el acceso a la WAN y el tráfico proveniente de los dispositivos de la capa de acceso es agregado por los routers de agregación en la capa de distribución. Estos routers se conectan en el “core” de la red y proveen conectividad entre los routers de agregación, los dispositivos concentradores de VPN y el “core” de campus (ver Figura 4).

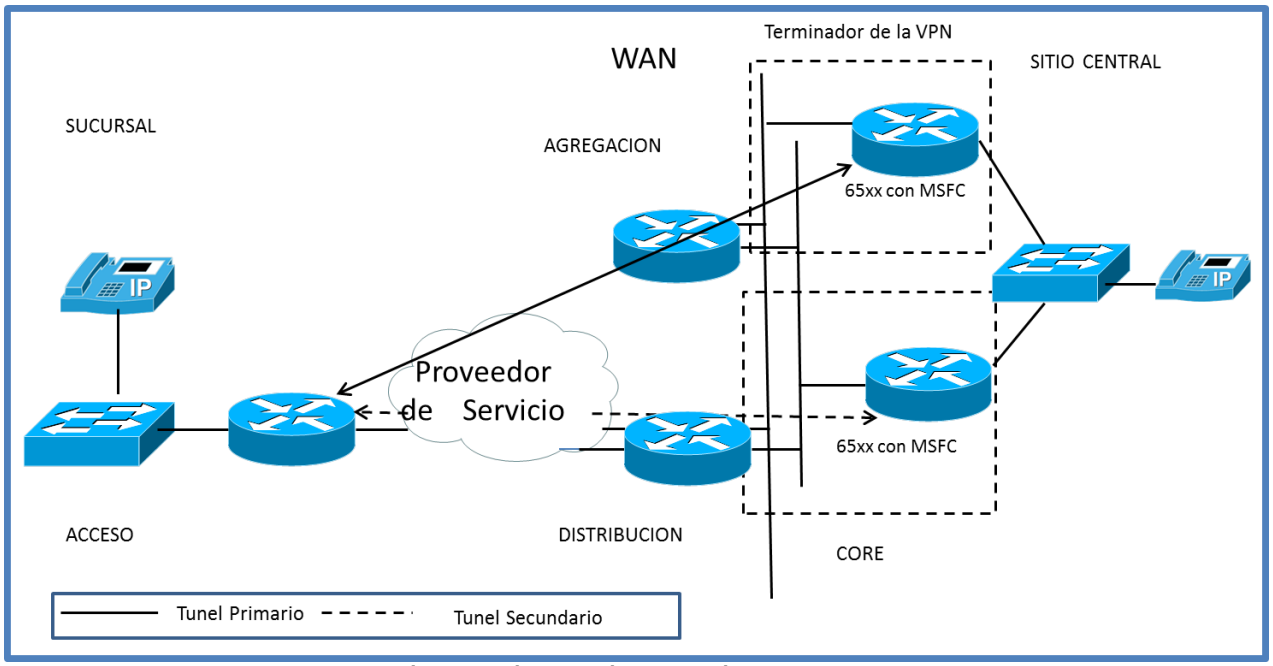

Figura 4 - Diagrama de una arquitectura WAN

Durante la fase de diseño de la WAN, debe prestarse especial atención a una serie de características. Entre ellas:

- Tolerancia a fallas
- Consideraciones de ruteo
- Seguridad.

3.3.2 El diseño de una red WAN virtualizada

El reto de virtualizar una WAN es el de poder mantener una jerarquía y resistencia a fallos mientras se despliegan un gran número de VPNs a lo largo de la WAN de una manera efectiva en costos. El hecho de poder realizar esto, puede llevar a la creación de VPNs o túneles dentro de las VPNs del proveedor. El aumento en número de las VPNs en la WAN se junta con el reto de poder mantener una topología lógica de mayor complejidad que pueda escalar para dar soporte a este escenario.

Se puede concluir que la arquitectura de una red virtualizada debe tener las mismas características que las de una red física como ser tolerancia a fallos, seguridad, escalabilidad en el campus, en el área metropolitana y en la WAN, por lo que se deben seguir los mismos principios de diseño.

4. Tecnología de base para la virtualización

4.1 Introducción

En el presente capítulo se detallan distintas tecnologías que se utilizan como "building blocks" de la virtualización de redes. Las tecnologías presentadas son las que fueron consideradas como más relevantes. Para el desarrollo, se tomó como guía el capítulo 4 de [1], complementando la información con [6] junto con distintos estándares como ser [7] , [8], [9], [10], [11], [12] y otras fuentes como por ejemplo [13] y [14].

4.2 Virtualización del equipamiento de red

Para poder comprender los retos que nos propone la virtualización de un dispositivo de red, debemos poder contestar las siguientes interrogantes:

¿De qué manera el tráfico es separado internamente en un dispositivo?
¿Cuáles son las primitivas utilizadas para el tráfico L2, L3 o L4?

Una de las características esenciales de una red virtual es el proveer la sensación de disponer comunicación privada sobre una infraestructura compartida. Este hecho, crea los siguientes requerimientos de la infraestructura:

- **El tráfico proveniente de un grupo no se debe mezclar con otro.** Al enviar y recibir tráfico sobre vínculos compartidos se debe garantizar la separación de datos. Los dispositivos de red deben asegurar la separación de grupos en su memoria interna (búsqueda en tablas de ruteo, procesamiento de listas de control de acceso, etc)

- **Cada red virtual tiene un espacio de direcciones propio.** Este requerimiento es heredado del hecho que una red virtual debe ofrecer las mismas características que una red física. El espacio de direccionamiento y el "forwarding" sobre la red son los dos aspectos más básicos de una red.

El primer desafío a resolver es como virtualizar el plano de "forwarding" de una manera que cumple los requerimientos de separación del espacio de direcciones y separación de tráfico. Dependiendo el tipo de dispositivo, esta separación puede tener los siguientes nombres:

- "Virtual LAN" (VLAN)
- "Virtual Routing and Forwarding" (VRF)
- "Virtual Forwarding Instance" (VFI)
- "Virtual Firewall Context" (Contextos virtuales)

4.2.1 Capa 2: VLANs

Las VLANs son un buen ejemplo de tecnología base de virtualización que se encuentra disponible durante un buen tiempo (el estándar 802.1q data del año 1988 [8]). Una VLAN es un grupo de puertos en un "switch" que forman un dominio de "broadcast". Un puerto en una VLAN solamente puede comunicarse con otros puertos en la misma VLAN. La manera que un "switch" hace esto es dependiente de la implementación, pero una solución común es marcar ("tag") cada "frame" con el número de VLAN cuando llega al puerto. Cuando se va a enviar un "frame" a otros puertos, se lo copia solamente si el puerto de destino se encuentra configurado con el mismo número de VLAN contenido en la marca del "frame".

En un "switch" Ethernet existe una tabla de MACs, que mapea puertos del dispositivo con direcciones MAC. Para dar soporte a las VLANs, (o virtualización de capa 2), la tabla de MACs

tiene un campo para el número de VLAN en la cual la estación fue descubierta (ver Figura 5).

```
Switch# show mac-address-table
...
Non-static Address Table:
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
0010.0de0.e289       Dynamic          1  FastEthernet0/1
0010.7b00.1540       Dynamic          2  FastEthernet0/5
0010.7b00.1545       Dynamic          2  FastEthernet0/5
0060.5cf4.0076       Dynamic          1  FastEthernet0/1
0060.5cf4.0077       Dynamic          1  FastEthernet0/1
0060.5cf4.1315       Dynamic          1  FastEthernet0/1
0060.70cb.f301       Dynamic          1  FastEthernet0/1
00e0.1e42.9978       Dynamic          1  FastEthernet0/1
00e0.1e9f.3900       Dynamic          1  FastEthernet0/1
```

Figura 5 - La tabla MAC

4.2.2 Capa 3: Instancias de VRF (Virtual Routing and Forwarding)

Las VRFs son a capa 3 (L3) lo que las VLANs son a capa 2 (L2) y delimitan el dominio de una red IP dentro de un router. Una posible definición formal es la siguiente.

> VRF- Una VRF consiste de una tabla de ruteo IP, una tabla de "forwarding" asociada y un conjunto de reglas y protocolos de ruteo que determinan el contenido de la tabla de "forwarding" [1].

En la Figura 6 se representa un router con dos VRFs: ROJO y VERDE. La tabla ROJO puede conmutar paquetes entre las interfaces E1/0, E1/2 y S2/0.102. La tabla VERDE por otro lado conmuta entre las interfaces E4/2, S2/0.103 y S2/1.103. Una interface no puede pertenecer a múltiples VRFs de manera simultánea. De esta manera se puede apreciar como las VRFs proveen caminos separados entre las interfaces ruteadas. Los paquetes correspondientes a la VRF ROJO nunca pueden terminar en una interface correspondiente a la VRF VERDE.

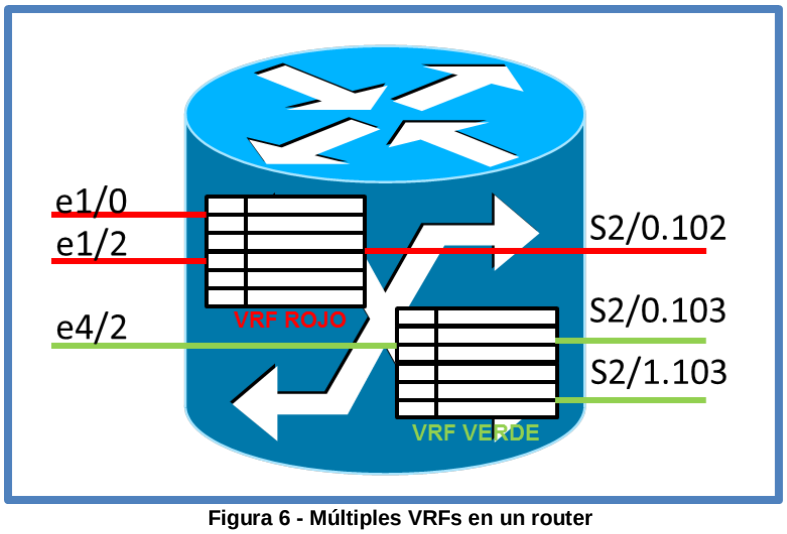

Figura 6 - Múltiples VRFs en un router

Adicionalmente a la información de ruteo para una VRF, se necesita ver un par de estructuras principales asociadas a la tabla de ruteo, las cuales son utilizadas para hallar la interface de egreso para un determinado paquete. Estas tablas se denominan "Forward Information Base" (FIB) y la "Routing Information Base" (RIB).

La base FIB es utilizada para realizar el forward de los paquetes. Cuando se recibe un paquete en una interface ruteada, el router busca el destino en la FIB para encontrar cual es el próximo hop que el paquete debe alcanzar. La estructura de la FIB es altamente eficiente para realizar búsquedas del tipo “longest-prefix”. En conjunto con esta información se mantiene una tabla de adyacencias. Se define como adyacencia a un nodo de la red que es alcanzable con un único hop de L2. Como esta estructura contiene información de L2 y L3, la FIB puede ser actualizada desde múltiples fuentes como ser los protocolos de ruteo o actualizaciones ARP (“Address Resolution Protocol”).

La RIB es la estructura en memoria donde se encuentra almacenada toda la información de ruteo IP. No es específica de ningún protocolo sino que es el repositorio donde todos los protocolos guardan las rutas. Las rutas son insertadas en la RIB cuando un protocolo de ruteo aprende una nueva ruta. Cuando un destino se convierte inalcanzable (“unreachable”), la ruta se marca como no utilizable (“unusable”) en una primera instancia y luego es removida de la RIB según la especificación del protocolo por el cual fue aprendida.

Cuando se habilitan las VRFs, existen múltiples instancias de información en la FIB y en la RIB. Se puede consultar la información de ruteo de una VRF en particular utilizando el comando **show ip route vrf** <nombre> donde nombre es el nombre de la VRF en cuestión (ver Figura 7).

```
R104#show ip route vrf RED
Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
     20.0.0.0/24 is subnetted, 1 subnets
O       20.0.0.0 [110/11121] via 40.0.0.1, 00:00:02, Tunnel0
     40.0.0.0/24 is subnetted, 1 subnets
C       40.0.0.0 is directly connected, Tunnel0
     30.0.0.0/24 is subnetted, 1 subnets
C       30.0.0.0 is directly connected, Ethernet0/0
```

Figura 7 - Información de Tabla de ruteo de la VRF RED

De manera análoga a lo explicado anteriormente, es posible consultar la FIB de una VRF e particular utilizando el comando **show ip cef vrf** <nombre> donde nombre es el identificador de la VRF en cuestión.

Una pregunta natural que puede surgir es el consumo de recursos al utilizar VRFs. La utilización de VRFs no impone una carga significativa al dispositivo, pero se debe tener cuidado que la suma de la cantidad de rutas en cada tabla de ruteo virtual no exceda la capacidad del dispositivo. Se debe manejar de manera similar a las mejores prácticas para administrar el tamaño de la tabla de ruteo de un dispositivo.

La manera en que se procesa el tráfico, ocurre de la misma manera que en un dispositivo sin VRFs

1. El tráfico ingresa en el router
2. Se ejecuta la política de ingreso es aplicada (“ingress policy”)
3. Los “lookups” de “forwarding” y ruteo son ejecutados
4. Se ejecuta la política de salida (“egress policy”)
5. Se realiza el forward del tráfico

Al disponer de múltiples instancias de ruteo y “forwarding” en un router, muchos de los subsistemas del router que utilizan estas tablas deben ser conscientes de la existencia de las VRF (“VRF aware”). Una funcionalidad o comando consciente de las VRFs puede ser configurado para referirse a la información de ruteo y “forwrading” de una VRF específica y que

solo ciertas subinterfaces pueden ser utilizadas con ciertas VRFs. Sin esta información, la funcionalidad utiliza la tabla global. Por ejemplo el comando **ping** para enviar mensajes ICMP fue modificado de la siguiente manera **ping vrf** <nombre> <destino> donde nombre es el nombre de la VRF y destino es la dirección / nombre del host de destino.

4.2.3 Virtualización del hardware

Una VRF no es lo mismo que un dispositivo totalmente virtualizado; solamente permite a los routers dar soporte a múltiples espacios de direcciones. Esto dista de un dispositivo totalmente virtualizado, donde los recursos pueden ser de alguna manera asignados a procesos.

Los dispositivos virtualizados existen y los podemos clasificar según la tecnología que se utilice para virtualizar.

4.2.3.1 Routers virtuales y lógicos

Según como se implemente la virtualización del componente, los dispositivos se pueden clasificar como:

Router lógico (Logical Router – LR). Esta clase de dispositivo usa el particionamiento de hardware para crear múltiples entidades de ruteo en un único dispositivo. Un LR puede ejecutar sobre distintos procesadores en diferentes tarjetas de expansión del router. Todo el hardware y software debajo se encuentra dedicado al LR. Esto incluye procesadores de red, interfaces y las distintas tablas de ruteo y "forwarding". Los LR proveen un aislamiento a fallas excelente, pero requieren cantidades grandes de hardware para ser implementados.

Router virtual (Virtual Router – VR). Un router virtual usa una emulación de software para crear entidades de ruteo. El hardware subyacente se comparte entre los distintos procesos de ruteo. En una implementación de un VR bien realizada, los usuarios pueden acceder a la configuración y estadísticas de su router únicamente.

La línea Nexus 7000 [15] de Cisco da soporte al sistema operativo NX-OS [16] [17], una nueva clase de sistema operativo diseñado para "Data Centers". Basado en la plataforma Cisco MDS 9000 SAN-OS, Cisco NX-OS introduce el soporte para "Virtual Device Contexts" (VDCs), que permite a los "switches" ser virtualizados a nivel de dispositivo. Cada VDC configurada se presenta como un único dispositivo a los usuarios conectados dentro del "framework" de ese "switch" físico. La VDC se ejecuta como una entidad lógica separada dentro del "switch", manteniendo su conjunto único de procesos de software, disponiendo de su propia configuración.

El sistema operativo de Cisco IOS XR [18] introduce soporte para la tecnología de virtualización "Hardware-Isolated Virtual Router" (HVR) denominada "Secure Domain Routers" (SDR). Esta tecnología dedica al plano de control y al plano de datos recursos por módulos de hardware a entidades virtuales individuales, por lo que no se comparten recursos del plano de control ni del plano de datos. Debido a la asignación dedicada de recursos al plano de control y al plano de datos, las aplicaciones de software y el hardware de "forwarding" no necesitan implementar la virtualización. Este tipo de tecnología se puede encontrar en la línea Cisco XR 1200 o en la línea Cisco ASBR 9000.

4.2.3.2 Device pooling

La técnica de Device pooling consiste en que varios dispositivos se conectan entre sí y pueden ser vistos como un único dispositivo. En dispositivos de L2 la gran ventaja de esta técnica es la

eliminación de “spanning tree”. La mayoría de las implementaciones existentes son propietarias de los fabricantes de los dispositivos, pero existen propuestas de estándares de la IETF para protocolos estándar [7].

- Cisco Virtual Switching System (VSS) [13] [14]

VSS es una tecnología presente en la línea de “switches” de “core” Cisco de la línea Catalyst 6500 que permite conectar varios “switches” de tal manera que se vean como uno solo. Solamente uno de los miembros del grupo tiene activo el plano de control, por lo que la administración se realiza en una única instancia. Debido a esto mismo, si se lo utiliza como default Gateway en una red, se necesita una única dirección IP en vez de tres (ya que no se requiere utilizar protocolos como HSRP, GLBP, VRRP). En combinación con “Multichassis Etherchannel” (ME) se puede realizar agregación de ancho de banda contra un “switch” de acceso (por ejemplo) conectando un link en cada chasis, sin la necesidad de ejecutar “spanning tree” (ver Figura 8). Los “switches” se pueden interconectar con interfaces estándar de 10 Gb Ethernet y la limitación de distancia está dada por el tipo de interface elegida (en casos de interfaces ópticas de 10 Gb puede ser de hasta 40km).

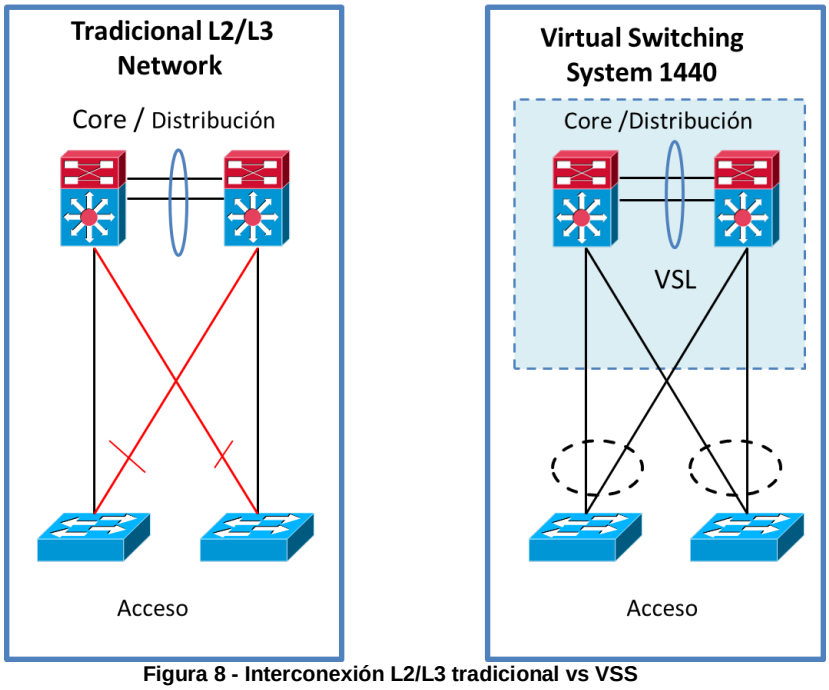

Figura 8 - Interconexión L2/L3 tradicional vs VSS

- Cisco Virtual Port Channel (vPC) [15]

“Virtual PortChannel” permite que los vínculos que se encuentran físicamente conectados a dos “switches” distintos sean vistos por un tercer dispositivo como provenientes de un único dispositivo y formando parte de un único “port channel”. Este tercer dispositivo puede ser un “switch”, un servidor o cualquier otro dispositivo de red que soporte “port channels” según la especificación de la norma 802.3ad de la IEEE [19]. El protocolo vPC se encuentra implementado en la familia de Cisco Nexus 5000 y 7000. Es una tecnología similar a VSS, pero su principal diferencia es que el plano de control se encuentra repartido entre los distintos dispositivos que forman el grupo (ver Figura 9).

La mayor ventaja de utilizar vPC en los data centers es que el tráfico entre los clientes y los servidores o entre los servidores puede utilizar todos los vínculos disponibles

simultáneamente. Dado que la familia Nexus tiene soporte para “Fibre Channel over Ethernet” (FCoE) y el hecho de contar con múltiples interfaces de 10 Gb Ethernet, se puede utilizar este tipo de equipamiento para la implementación de los Datacenters de nueva generación.

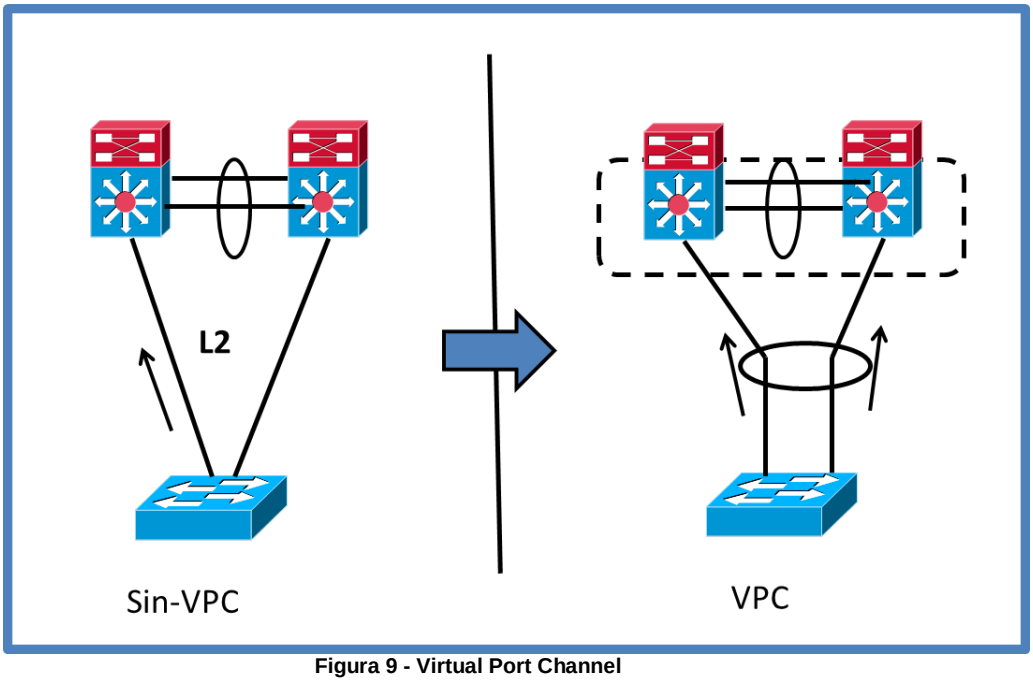

Figura 9 - Virtual Port Channel

- Inter Chassis Control protocol ICCP (draft-ietf-pwe3-iccp-11) [7]
  La propuesta de estándar ICCP pretende definir un protocolo no propietario para la conformación de un pool de dispositivos PE en el escenario de una VPN L2. La propuesta define un protocolo “inter-chassis” (ICCP) que permite la sincronización de estado y configuración entre dos o más PEs formando un grupo redundante (RG). Este protocolo soporta mecanismos de redundancia “multi-chassis” que pueden ser empleados sobre al “attachement circuit” o del lado del “pseudowire” (ver Figura 10).

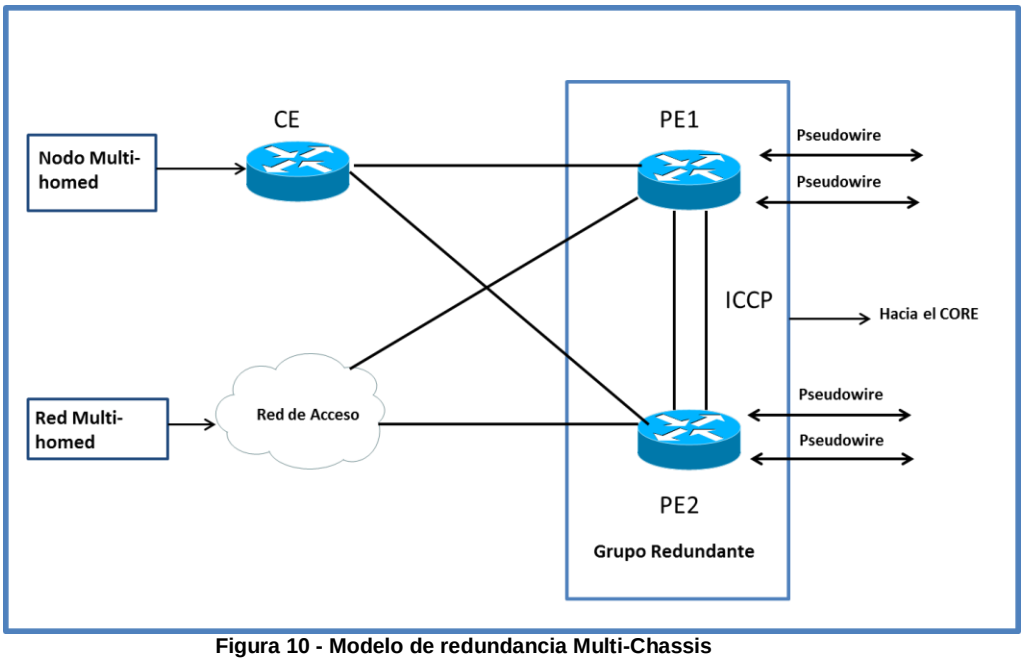

Figura 10 - Modelo de redundancia Multi-Chassis

4.2.4 Firewalls virtuales.

La virtualización de dispositivos no se limita a “switches” y “routers”. Es posible compartir un único “firewall” entre distintos clientes o segmentos de red. Cada “firewall” lógico necesita tener un conjunto completo de políticas, interfaces dedicadas para tráfico de entrada / salida y usuarios autorizados para administrar el “firewall”. En la línea de dispositivos Cisco (los cuales utilizamos para los ejemplos) el término contexto se refiere a un “firewall” virtual. Un contexto es una emulación de un dispositivo y es un ejemplo de un dispositivo del tipo Router Virtual (VR).

La definición de los contextos en el “firewall” virtual se realiza de manera distinta a una VRF. Cuando llega un paquete, el “firewall” inspecciona la dirección de destino o el “tag” de VLAN a la que pertenece y en base a esta información decide a cual contexto pertenece. Todo lo que necesita el “firewall” es que alguno de estos dos campos sea único, por lo que o bien cada contexto dispone de un espacio de direcciones único en sus interfaces o bien es compartido y cada contexto se encuentra sobre una VLAN distinta.

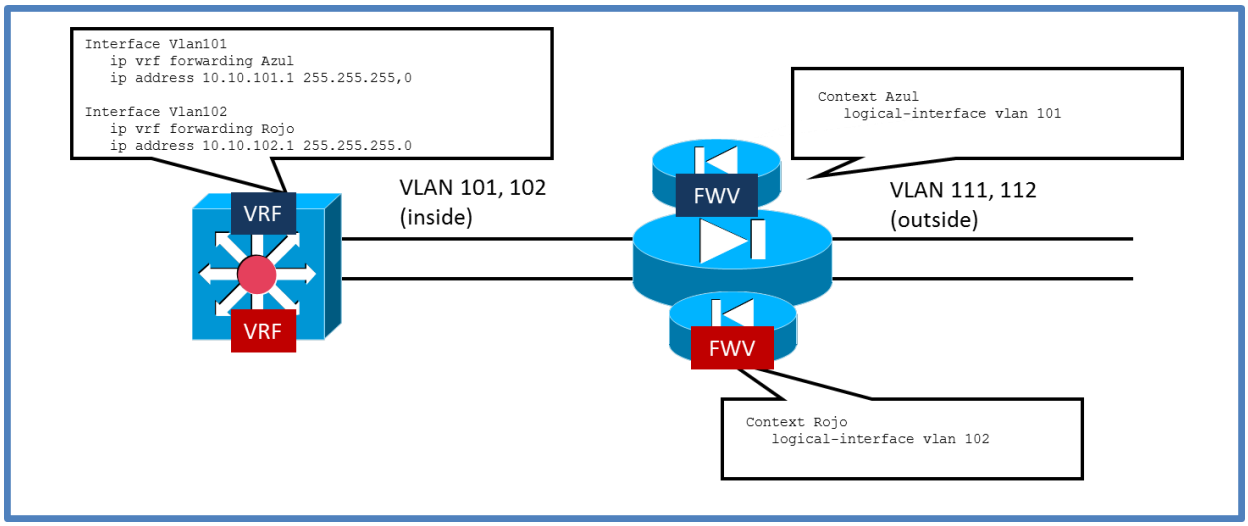

Figura 11 - Ejemplo de Firewall virtualizado

En el ejemplo de la Figura 11, se muestra un “firewall” virtualizado donde un “switch” se encuentra conectado a un “firewall” utilizando dos VLANs. El “switch” asigna una VLAN a una VRF Azul y la otra VLAN a una VRF Rojo. El “firewall” dispone de dos contextos distintos. El contexto Azul recibe los datos de la VLAN 101 y el contexto Rojo recibe los datos de la VLAN 102. De esta manera, los paquetes que se reciben en la interface “outside” del “firewall” sobre la VLAN 111 (que luego quedarán sobre la VLAN 101 del lado “inside”) pasan por un conjunto de reglas de “firewall” distintas de aquellas que pertenecen a la VLAN 112.

4.3 Virtualización del plano de datos

Cuando abordamos la virtualización del plano de datos, las principales cuestiones que se deben responder son:

> ¿De qué manera el tráfico es separado dentro de un camino dado en la red (network path)?
> ¿Qué herramientas se encuentran disponibles para lograr este propósito?

El desafío de conectar dispositivos mediante vínculos privados sobre una infraestructura compartida es un problema bien conocido. Los proveedores de servicio han resuelto este utilizando distintas soluciones de VPN. Estas mismas soluciones pueden ser utilizadas dentro de la empresa para crear conexiones virtualizadas en L2 o L3 utilizando una infraestructura común de redes. A continuación se realizará un inventario de las soluciones más comúnmente utilizadas.

4.3.1 802.1q

Generalmente no pensamos en 802.1q [8] como un protocolo de virtualización del plano de datos, pero este protocolo al insertar un “tag” de VLAN en vínculos Ethernet, garantiza la separación del espacio de direcciones en interfaces Ethernet.

Esta es una solución L2 y en cada hop se debe configurar correctamente para permitir la conectividad 802.1q en toda la red. Dado que VLAN es un sinónimo de dominio de “braodcast”, las VLANs del tipo “end-to-end” (donde las VLANs, son encapsuladas a lo largo de toda la organización, campus o edificio) son generalmente evitadas.

4.3.2 Generic Routing Encapsulation (GRE)

El protocolo GRE [9] provee un método de encapsular paquetes de un protocolo en otro protocolo (el RFC 2784 que lo define usa la expresión X sobre Y [9], lo cual da un indicio del problema que resuelve). A los datos del protocolo encapsulado (X) se les denomina carga (“payload”). La capa inferior se la denomina protocolo de entrega (“delivery protocol”). GRE permite transportar datos privados utilizando infraestructura pública mediante túneles punto a punto (“point-to-point tunnels”).
Si bien GRE es una solución genérica X sobre Y, lo más común es transportar IP sobre IP. Una versión modificada de GRE fue utilizada por Microsoft en su protocolo PPTP (Point to Point Tunneling Protocol). Hoy en día se pueden ver instalaciones donde se utiliza GRE para transportar MPLS sobre redes IP. Otra aplicación de GRE es el transporte sobre IP de protocolos legados como ser IPX (Internetwork Packet Exchange), AppleTalk, etc.

GRE es definido en el RFC 2784 [9] y tiene un encabezado (“header”) simple (ver Figura 12).

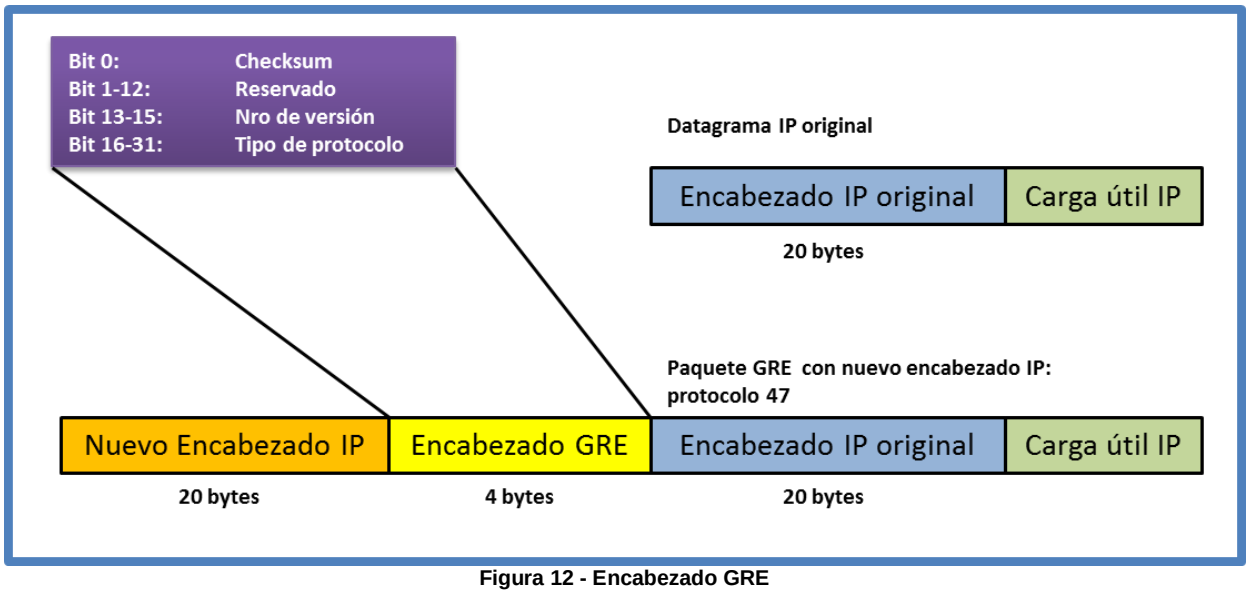

Figura 12 - Encabezado GRE

Los segundos 2 octetos del encabezado contienen la codificación de protocolo de carga, el cual es codificado según los números Ethernet provistos por el IANA (Internet Assigned Numbers Authority). El protocolo IP se codifica como 0x800.

La mínima expresión de un encabezado GRE es el campo “Protocol Type”. Todos los campos anteriores generalmente son 0 y los subsiguientes pueden ser omitidos.

GRE es un mecanismo de encapsulación / transporte. La manera en que los paquetes llegan a las puntas de los túneles (“endpoints”) es a consideración del usuario. No existe un protocolo de control ni estado de sesión que mantener ni auditar. Esta simplicidad permite que GRE sea fácilmente implementado en hardware en los sistemas de gran porte. La desventaja es que los “endpoints” de GRE no tienen manera de saber que sucede en la otra punta del túnel o siquiera si se encuentra accesible. La manera de poder detectar problemas de alcanzabilidad es ejecutar un protocolo de ruteo dinámico sobre el túnel. Los “keepalives” del protocolo de ruteo (Routing Protocol – RP) son descartados si el túnel se encuentra bajo y el RP por sí mismo declarará al vecino (“neighbor”) como no alcanzable (“unreachable”) por lo que intentará un camino alternativo para llegar a él.

La falta de un protocolo de control de GRE significa que no existe costo para mantener un túnel inactivo levantado. Los dispositivos en los extremos del túnel no intercambian información de estado y solamente debe encapsular los paquetes a medida que llegan. De la misma manera que otros mecanismos relacionados con la virtualización del plano de datos, el “core” de la red es inconsciente de su existencia y de la cantidad de túneles que lo atraviesan. Todo el trabajo es realizado en los extremos.

Si bien GRE es un protocolo genérico, no es la solución universal. Existe un costo asociado a encapsular / desencapsular, consultas de la tabla de ruteo, etc.

4.3.2.1 Ejemplo de aplicación de GRE

En el ejemplo de la Figura 13, se interconectan dos redes en el Sitio1 y el Sitio 2 de la empresa. Para realizar esto, se establece un túnel GRE entre los routers de borde en cada sitio.

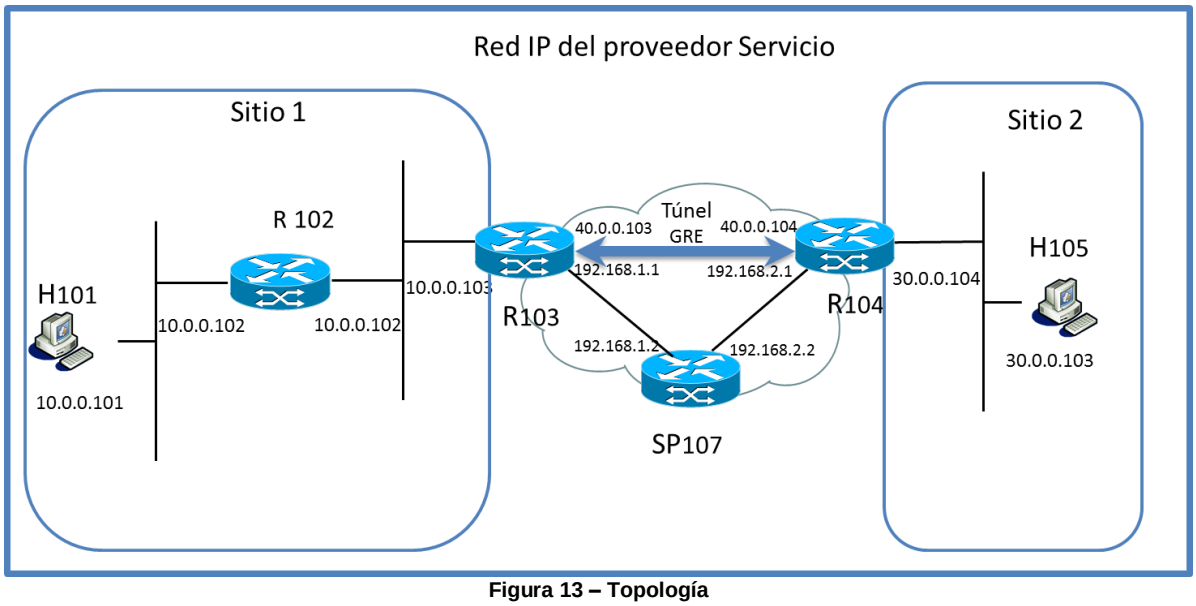

Figura 13 – Topología

Se aplica la configuración en R103 y R104 e independientemente del camino que los paquetes de datos tomen, se crea un túnel GRE entre R103 y R104.
En los dispositivos Cisco, los "endpoints" GRE se ven como interfaces regulares. Esto tiene mucho significado ya que todos los subsistemas que necesitan ver una interface (protocolos de ruteo, listas de acceso, etc) funcionarán automáticamente sobre un túnel GRE. En este caso el tipo de interface definido es "Tunnel". En la Figura 14, se incluyen los comandos necesarios para definir este tipo de interface.

```
Interface Tunnel0
  ip address 40.0.0.1   255.255.255.0
  tunnel source Serial1/0
  tunnel destination 192.168.2.1
```

Figura 14 - Configuración GRE de R103

Las direcciones "tunnel source" (origen) y "tunnel destination" (destino) son parte del espacio de direcciones correspondiente a la red de transporte (en este caso 192.168.1.1 y 192.168.2.1 respectivamente). Se necesita que coincidan en ambos extremos por lo que la dirección de origen en un router es la dirección de destino en el otro. En el router debe existir una entrada en la tabla de ruteo hacia la dirección de destino (el otro extremo del túnel).

En este ejemplo, R103 tiene una interface de tipo "Tunnel" con destino 192.168.2.1 en la red pública. La red 40.0.0.0/24 utilizada para la dirección IP del túnel pertenece al espacio de direcciones privado utilizado en ambos sitios de la empresa.

4.3.3 IPsec

IPsec [10] [11] [12] provee un conjunto completo de servicios de seguridad para redes IP. Fue originalmente concebido para proveer transporte seguro sobre redes IP. Los servicios que ofrece son los protocolos de autenticación ("Authentication Header" [AH]), cifrado ("Encryption Header" [EH]) y mecanismos seguros de intercambios de claves (Key Exchange).

IPsec es una parte obligatoria de IPv6, y su uso es opcional con IPv4. Aunque el estándar está diseñado para ser indiferente a las versiones de IP, el despliegue y experiencia hasta 2007 se refiere a las implementaciones de IPv4.
Los protocolos de IPsec se definieron originalmente en las RFCs 1825 y 1829 [20], publicadas

en 1995. En 1998 estos documentos fueron sustituidos por las RFCs 2401 [21] y 2412 [22], que no son compatibles con los RFCs 1825 y 1829, aunque son conceptualmente idénticas. En diciembre de 2005 se produjo la tercera generación de documentos, RFCs 4301 [10] y 4309 [11]. Son en gran parte un superconjunto de la 2401 y 2412, pero proporcionan un segundo estándar de Internet Key Exchange definido en el RFC 4306 [12] . Esta tercera generación de documentos estandarizó la abreviatura de IPsec como "IP" en mayúsculas y "sec" en minúsculas.

IPsec fue proyectado para proporcionar seguridad en modo transporte (host a host) del tráfico de paquetes, en el que los ordenadores de los extremos finales realizan el procesado de seguridad, o en modo túnel (“gateway to gateway”) en el que la seguridad del tráfico de paquetes es proporcionada a varias máquinas (incluso a toda la red de área local) por un único nodo.
IPsec puede utilizarse para crear VPNs en los dos modos, y este es su uso principal. Hay que tener en cuenta, sin embargo, que las implicaciones de seguridad son bastante diferentes entre los dos modos de operación.
Como el Protocolo de Internet no provee intrínsecamente de ninguna capacidad de seguridad, IPsec se introdujo para proporcionar servicios de seguridad tales como:

- Cifrar el tráfico (de forma que no pueda ser leído por nadie más que las partes a las que está dirigido).
- Validación de integridad (asegurar que el tráfico no ha sido modificado a lo largo de su trayecto).
- Autenticar a los extremos (asegurar que el tráfico proviene de un extremo de confianza).
- Anti-repetición (proteger contra la repetición de la sesión segura).

IPsec mantiene una base de datos de asociaciones de seguridad. Una asociación de seguridad (“Security Association” – SA) es un contrato entre pares en el cual se define lo siguiente:

- Los protocolos de cifrado / autenticación utilizados (por ejemplo Triple DES – Triple Data Encryption Standard)
- El servicio IPsec a utilizar (“Encapsulating Security Payload” [ESP] o “Authentication Header” [AH])
- Configuración necesaria para poder establecer comunicación con su par.

El SA se negocia cuando se inicia una sesión IPsec. Cada encabezado de IPsec contiene una única referencia a la SA para ese paquete un campo llamado “Security Parameter Index” (SPI), el cual es un entero de 32 bits que hace referencia a la SA necesaria para procesar el paquete. Los pares mantienen una lista de SAs para conexiones de entrada y salida. El valor de SPI se comparte entre pares y es parte de la información que se comparte durante la negociación de la sesión IPsec.

A continuación se describen los dos modos: modo transporte y modo túnel.

4.3.3.1 Modo transporte

En modo transporte, sólo la carga útil (los datos que se transfieren) del paquete IP es cifrada o autenticada. El enrutamiento permanece intacto, ya que no se modifica ni se cifra el encabezado IP; sin embargo, cuando se utiliza el encabezado de autenticación (AH), las direcciones IP no pueden ser traducidas, ya que eso invalidaría el hash. Las capas de transporte y aplicación están siempre aseguradas por un hash, de forma que no pueden ser modificadas de ninguna manera (por ejemplo traduciendo los números de puerto TCP y UDP). El modo transporte se utiliza para comunicaciones host a host.
Una forma de encapsular mensajes IPsec para atravesar NAT ha sido definida por los RFCs que describen el mecanismo de “NAT traversal” [23]. En la Figura 15 se muestra el paquete IP

original, mientras que en la Figura 16 se detallan los distintos componentes que se agregan para generar un paquete IPsec en modo transporte.

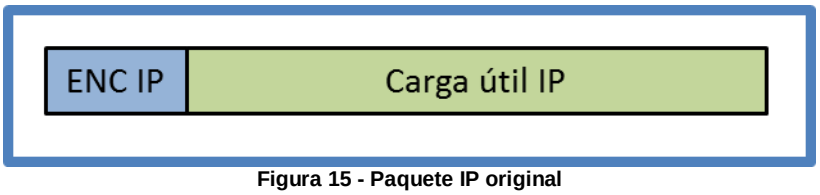

Figura 15 - Paquete IP original

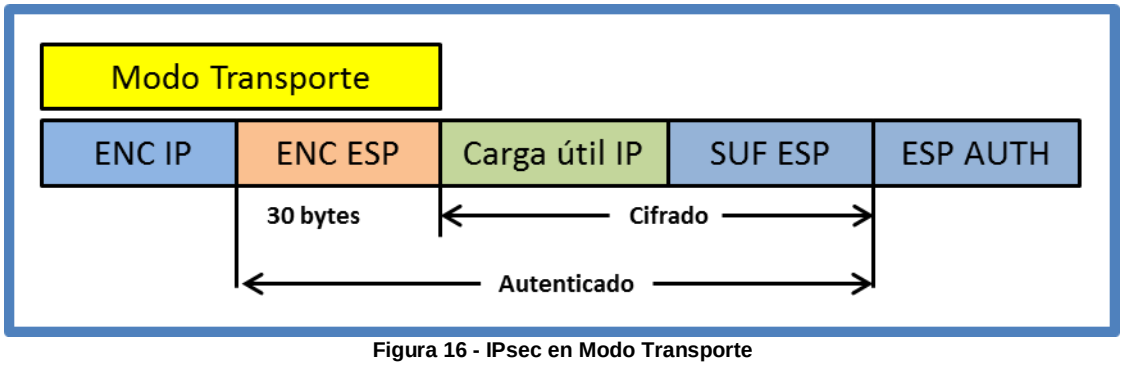

Figura 16 - IPsec en Modo Transporte

4.3.3.2 Modo túnel

En el modo túnel, todo el paquete IP (datos más encabezados del mensaje) es cifrado o autenticado. Debe ser entonces encapsulado en un nuevo paquete IP para que funcione el enrutamiento. El modo túnel se utiliza para comunicaciones red a red (túneles seguros entre routers, p.e. para VPNs) o comunicaciones host a red o host a host sobre Internet. El modo Túnel agrega una sobrecarga de 20 bytes. En la Figura 17 se detallan los distintos componentes que se agregan para generar un paquete IPsec en modo túnel.

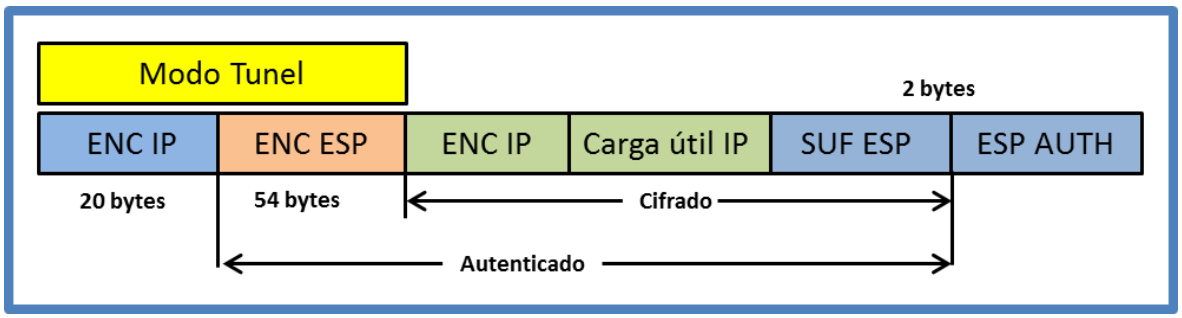

Figura 17 - IPsec en Modo Tunel

A nivel de protocolo existen dos encabezados IPsec:

- **Authentication Header (AH)** proporciona integridad, autenticación y no repudio si se eligen los algoritmos criptográficos apropiados.
- **Encapsulating Security Payload (ESP)** proporciona confidencialidad y la opción -altamente recomendable- de autenticación y protección de integridad.

Authentication Header (AH)

AH está dirigido a garantizar integridad, sin conexión y autenticación de los datos de origen de los datagramas IP. Para ello, calcula un “Hash Message Authentication Code” (HMAC) a través

de algún algoritmo hash operando sobre una clave secreta, el contenido del paquete IP y las partes inmutables del datagrama. Este proceso restringe la posibilidad de emplear NAT. Por otro lado, AH puede proteger opcionalmente contra ataques de repetición utilizando la técnica de ventana deslizante y descartando paquetes viejos. AH protege la carga útil IP y todos los campos de la cabecera de un datagrama IP excepto los campos mutantes, es decir, aquellos que pueden ser alterados en el tránsito. En IPv4, los campos de la cabecera IP mutantes (y por lo tanto no autenticados) incluyen TOS, Flags, Offset de fragmentos, TTL y suma de verificación del encabezado. AH opera directamente por encima de IP, utilizando el protocolo IP número 51. Un encabezado AH mide 32 bits. En la Figura 18 se detallan los campos que componen un encabezado AH.

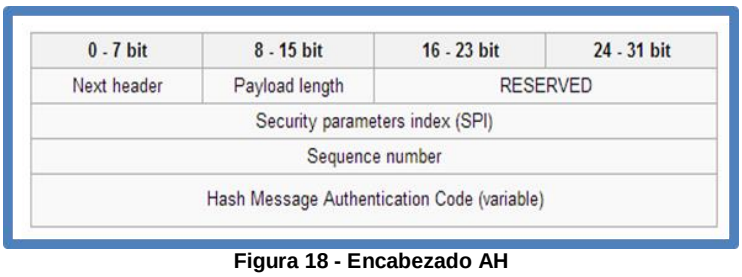

Figura 18 - Encabezado AH

Encapsulating Security Payload (ESP)

El protocolo ESP proporciona autenticidad de origen, integridad y protección de confidencialidad de un paquete. ESP también soporta configuraciones de sólo cifrado y sólo autenticación, pero utilizar cifrado sin autenticación está altamente desaconsejado porque es inseguro. Al contrario que con AH, la cabecera del paquete IP no está protegida por ESP (aunque en ESP en modo túnel, la protección es proporcionada a todo el paquete IP interno, incluyendo el encabezado interno; el encabezado externo permanece sin proteger). ESP opera directamente sobre IP, utilizando el protocolo IP número 50. En la Figura 19 se detallan los campos que componen un encabezado ESP.

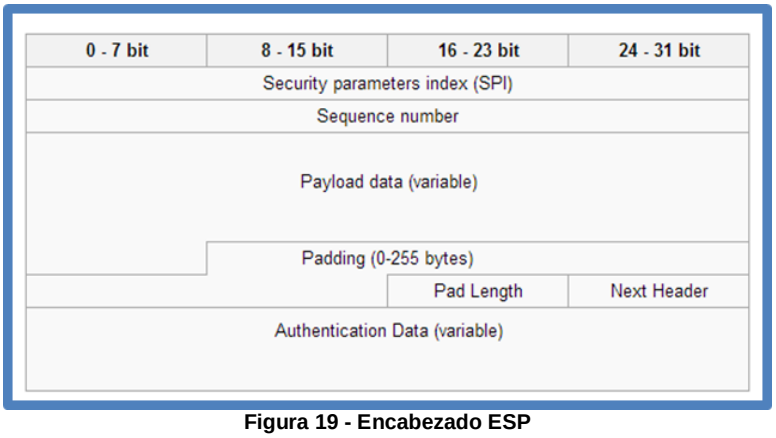

Figura 19 - Encabezado ESP

4.3.4 L2TPv3

El protocolo L2TPv3 [24] dispone de componentes para crear, mantener y cerrar sesiones y la capacidad de multiplexar distintos flujos L2 dentro de un túnel. Es una extensión del protocolo L2TP (RFC 2661 [25]) que permite transportar protocolo de capa de enlace (“link-layer”) distinto de PPP como lo indican la Figura 20 y la Figura 24. Las diferencias entre la versión 3 (definida en el RFC 3931 [24]) y la versión anterior, se refieren a hacer el protocolo menos específico a PPP. El encapsulamiento por defecto de L2TPv3 es IP.

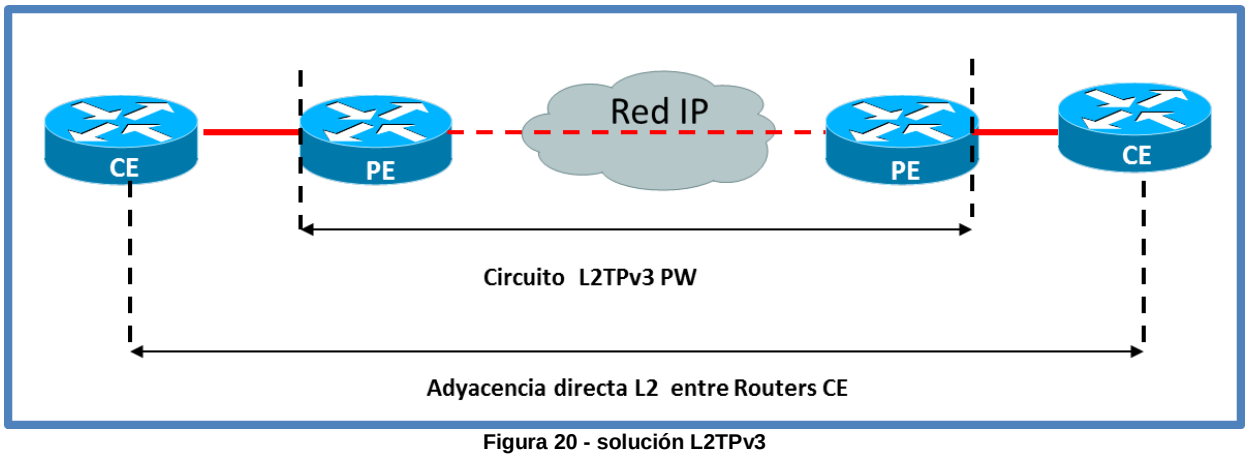

Figura 20 - solución L2TPv3

El protocolo L2TP dispone de un plano de control y uno de datos. El canal de control es confiable y dispone de 15 mensajes de control distintos. Los vecinos L2TPv3 pueden intercambiar información de capacidades durante la fase de establecimiento de la sesión. Las más importantes son el ID de sesión ("session ID") y la "cookie".

El ID de sesión es análogo a un identificador del canal de control y es un valor que el receptor asocia con el contexto que se negoció para una sesión en particular.

La cookie es un campo opcional de largo variable (ver el detalle en la Figura 21) de hasta 64bits. Es un número al azar que se utiliza para extender el ID de sesión de manera que hay pocas posibilidades de introducir tráfico en otra sesión debido a un ID de sesión corrupto. Como el dominio de la cookie (2^{64}) es grande, y el número es generado al azar, L2TPv3 es inmune a ataques de fuerza bruta tipo "spoofing", donde un atacante intenta inyectar paquetes dentro de una sesión activa.

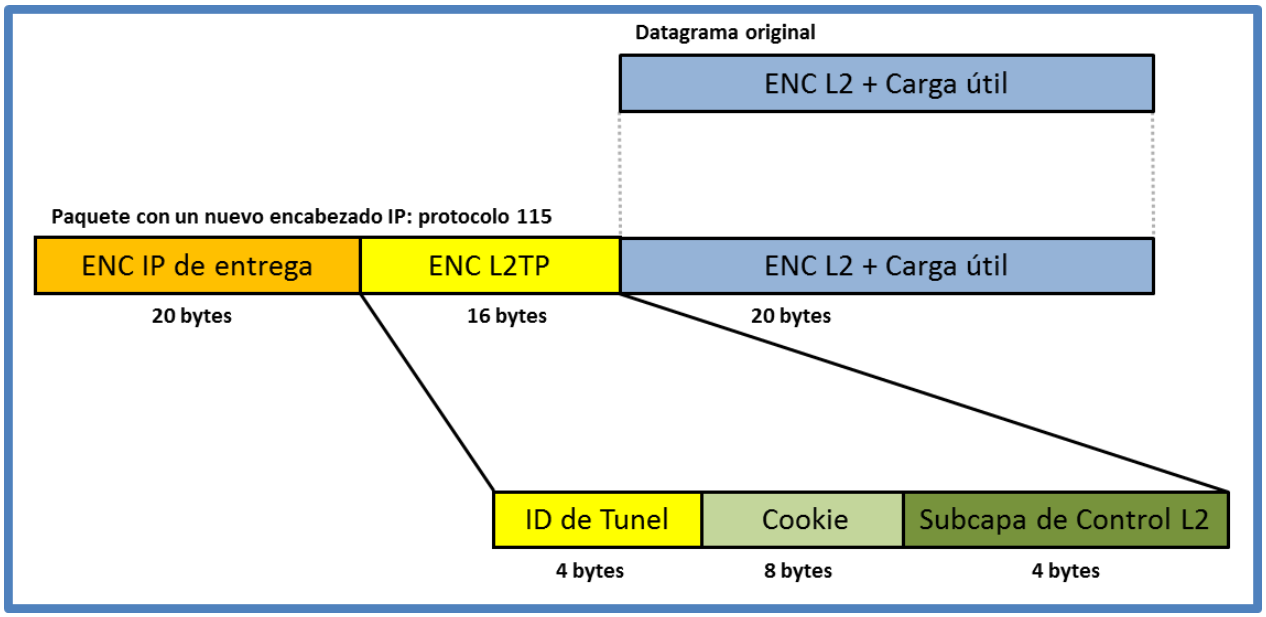

Figura 21 - Formato de paquete L2TPv3

Una vez que una sesión se establece, el "endpoint" L2TP se encuentra listo para enviar y recibir datos. Si bien el encabezado de datos ("data header") dispone de un campo "Sequence Number", el canal de datos no es confiable. El protocolo puede detectar paquetes perdidos,

duplicados o fuera de orden pero no los retransmite y delega esto a los protocolos de capas superiores.

4.3.4.1 Ejemplo de aplicación L2TPv3

Para la configuración del canal de control, se utiliza el comando l2tp-class como indica la Figura 22.

```
L2tp-class L2WAN
  password 7 00755657622
```

Figura 22 - l2tp-class

La segunda parte de la configuración se utiliza para el canal de datos. Se utiliza el comando "pseudowire-class", el cual especifica el encapsulado y refiere a la configuración del canal de control, así como el nombre de la interface que se usa como origen de paquetes L2TPv3 (ver Figura 23).

```
Pseudowire-class R103R104
  encapsulation l2tpv3
  protocol l2tpv3 L2WAN
  ip local interface Serial 1/0
```

Figura 23 - pseudowire-class

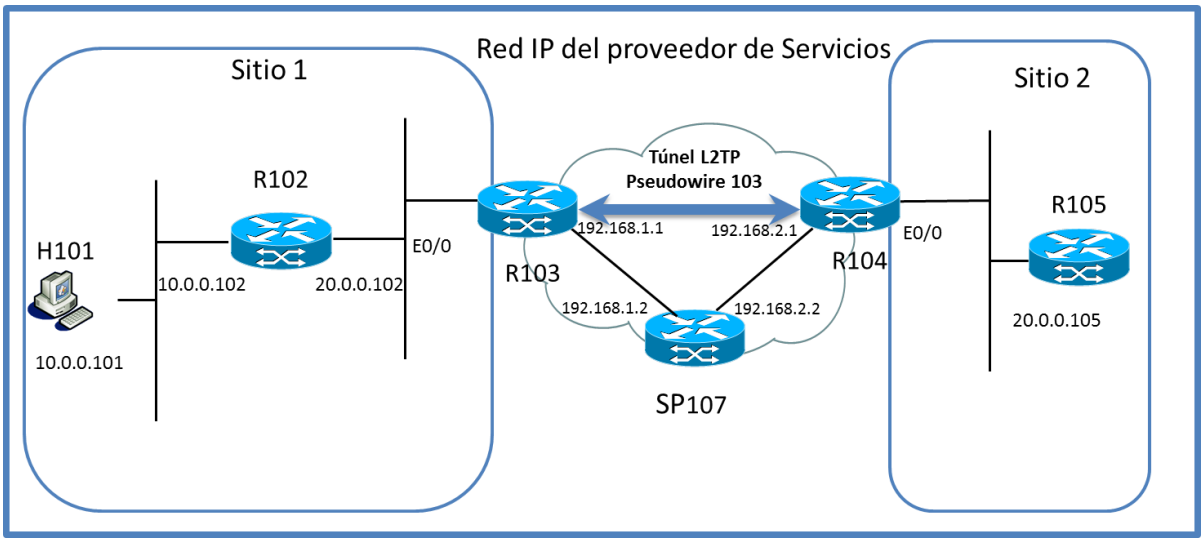

Figura 24 - Topología L2TPv3

La parte final de la configuración une el "attachement-circuit" que da hacia el cliente al "trunk", utilizando el comando "xconnect". Este comando define la dirección IP del nodo remoto y un identificador de circuito (VC ID) único que se utiliza en cada nodo para mapear la carga del paquete L2TPv3 al "attachement circuit" correspondiente. Los nodos L2TPv3 negocian valores únicos de ID de sesión y cookie para cada VC ID. Se debe configurar un VC ID distinto para cada VLAN, puerto o DLCI transportado sobre un túnel L2TPv3 (ver Figura 25).

```
Interface Ethernet 0/0
  description Puerto Hacia el Cliente
  no ip address
  no cdp enable
  xconnect 192.168.2.1 103 encapsulation 12tpv3 pw-class R103R104
```

Figura 25 – xconnect

4.3.5 Label Switched Paths

Los "Label Switched Paths" (LSPs) [26] son un híbrido, utilizando soluciones de L2 en el plano de datos y soluciones de L3 en el plano de control. Los LSP son encontrados en redes con tecnología "Multi Protocol Label Switching" (MPLS) [27]. MPLS es una tecnología híbrida que intenta combinar las mejores características de las técnicas conocidas para hacer llegar un paquete de un origen a un destino, tanto de L2 ("switching") como de L3 ("routing"), a través de una red de interconexión. MPLS surge como una evolución de ATM y/o Frame Relay, intentando mejorar algunos aspectos de estas tecnologías. MPLS está actualmente en uso en redes IP y sólo ha sido estandarizada por el IETF en RFC 3031 [26], aunque existen una serie de RFCs que la extienden o tratan temas anexos (p.e. RFC 3036 [28], RFC 5036 [29] - LDP). En la práctica, MPLS se utiliza principalmente para enviar datagramas IP y el tráfico de Ethernet. Las principales aplicaciones de la ingeniería de tráfico MPLS son las telecomunicaciones y MPLS VPN.

Los principales temas asociados a esta tecnología pueden resumirse como:

- Un LSP es un túnel que atraviesa una red MPLS conformado por varios segmentos del tipo hop-to-hop.
- Las redes MPLS utilizan el plano de control del protocolo IP.
- Los LSPs son configurados para todos los prefijos conocidos en la tabla de ruteo.
- Los LSPs son multiplexados dentro de vínculos físicos.
- Cada nodo en la red MPLS realiza toma la forward decisión de forwarding basándose en etiquetas de largo fijo en lugar de prefijos de largo variable.
- Las etiquetas son transportadas entre los encabezados de L2 y L3.
- Los nodos distribuyen las etiquetas a los nodos adyacentes utilizando un protocolo de distribución de etiquetas.
- La operación básica de MPLS es fácil de configurar.
- La configuración de MPLS debe ser realizada en todos los nodos.

En un escenario de ruteo tradicional, cuando un router necesita realizar el forward de un paquete, identifica la interface saliente buscando un prefijo IP que coincida en la tabla de ruteo. La interface que se utilizará para realizar el "forwarding" se corresponde al camino más corto al destino IP como se define por la política de ruteo. Otras políticas como QoS o seguridad pueden interferir con la elección de la interface. Este conjunto de criterios utilizado para tomar las decisiones de "forwarding" se conoce como "Forward Equivalency Class (FEC)". El mapeo de un paquete a una FEC se realiza en cada router a lo largo del camino IP y sucede de manera independiente de otros routers en la red.

MPLS desacopla la acción de "forwarding" de los paquetes de la información contenida en el encabezado IP. Un router MPLS realiza el forward de paquetes basado en etiquetas de largo fijo ("labels") en vez de coincidencias en prefijos IP de largo variable. Estas etiquetas son calculadas basadas en la información de topología de la red que se encuentra en la tabla de ruteo IP. El RFC 3031 (el cual define la arquitectura general de MPLS) establece lo siguiente:

> En MPLS, la asignación de un determinado paquete a una FEC determinada se realiza una única vez cuando el paquete ingresa a la red. La FEC a la cual el paquete es asignado se codifica en un campo de largo fijo llamado etiqueta ("label"). Cuando un paquete es enviado a su próximo salto, la etiqueta es enviada junto con el paquete, lo cual significa que el paquete es etiquetado antes de ser enviado.
> En MPLS, una vez que se le asigna una FEC a un paquete, no se realiza ningún análisis posterior de los encabezados por los routers que encuentre a lo largo del camino. Todas las acciones de "forwarding" son tomadas en base a las etiquetas.

En MPLS, los routers que componen la topología se clasifican según su ubicación y su función.

LSR (Label Switching Router). Es un componente de red que realiza el "forwarding" basado en etiquetas. Un LSR intercambia etiquetas. A diferencia de un router tradicional, un LSR no debe calcular a donde enviar un paquete basado en su encabezado IP, lo cual implica que no realiza clasificación por FEC cuando llega un paquete. Un LSR usa la etiqueta del paquete que llega para calcular la interface de salida (y la etiqueta asociada). Los LSR también suelen llamarse routers P ("Provider").

LSR de Borde (Edge LSR). Es un router que se encuentra en el borde de una red MPLS. El router de borde es quien se encarga de agregar / quitar las etiquetas a los paquetes. Este proceso de denomina "Pushing" y "Popping" porque se ve que las etiquetas se encuentran en un "stack". Los routers LSR de borde también se denominan como routers de borde del proveedor ("Provider Edge" – PE).

Routers de borde de cliente (Customer Edge – CE). Se denominan de esta manera a los routers del cliente que se conectan a los routers PE. Los routers CE realizan "forwarding" de paquetes IP. Los routers PE y los CE forman adyacencias mediante el uso de protocolos de ruteo dinámicos (o bien pueden utilizar ruteo estático).

Se utilizará el ejemplo que se encuentra en la Figura 26 para describir el funcionamiento de una red MPLS.

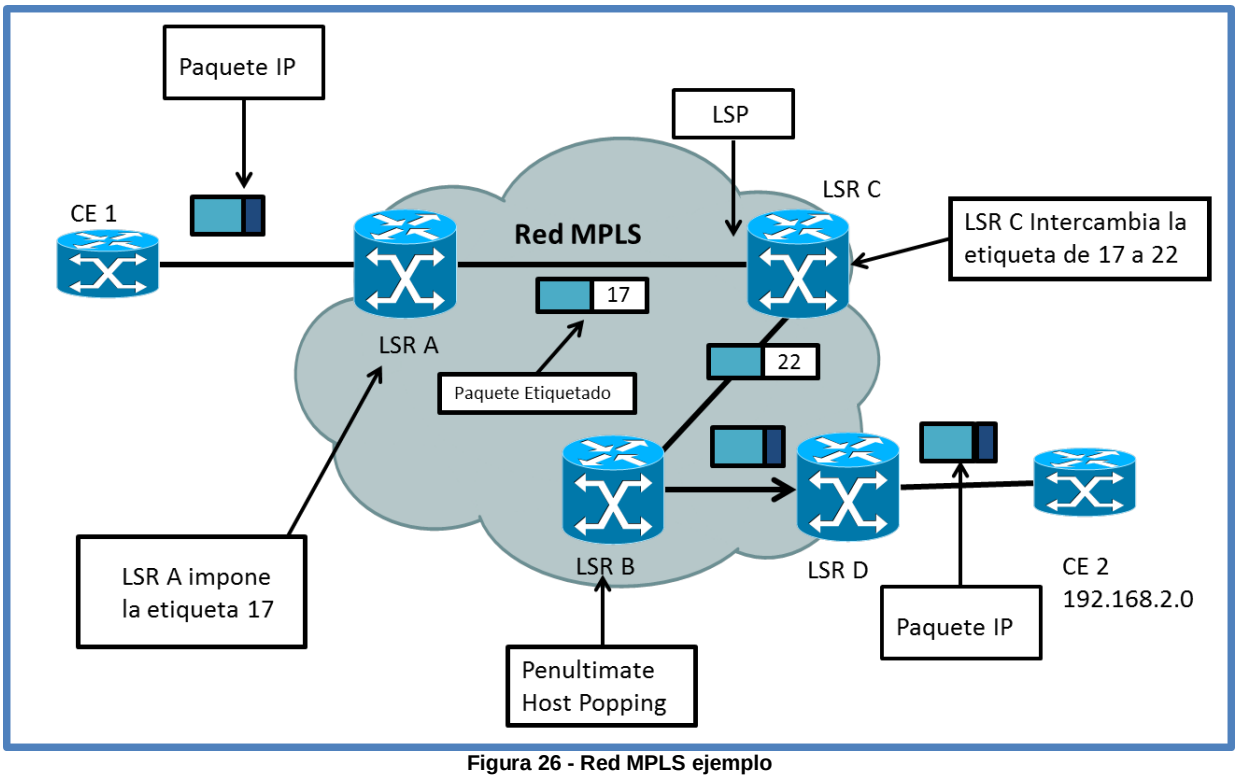

Figura 26 - Red MPLS ejemplo

A medida que un paquete recorre la red de ejemplo, es procesado por cada nodo de la siguiente manera:

1. En el borde de la red, el LSR A clasifica el paquete en una FEC y le asigna (“push”) la etiqueta 17. Una etiqueta tiene significado solamente local como puede ser el VPI/VCI de ATM o un DLCI de Frame Relay.

2. Los LSR en el “core”, como ser el LSR C y el LSR B intercambian (swap) los valores de las etiquetas. El LSR C remueve la etiqueta con valor 17 y asigna una nueva con valor 22. El valor de la etiqueta al ingreso y la interface son utilizados para encontrar los valores de la etiqueta e interface de salida.

3. El LSR B, al ser el penúltimo nodo del camino en la red MPLS, quita (pop) la etiqueta externa del “stack” de etiquetas. Este proceso se denomina “Penultimate Host Popping” (PHP). Esto se realiza para evitar que el LSR de borde tenga que realizar dos búsquedas de etiquetas MPLS (la correspondiente al LSP y a la VPN) y de esta manera reducir el consumo de CPU del mismo. Una vez que se ha quitado la etiqueta, los paquetes llegan al LSR D sin etiquetas, por lo cual se utiliza ruteo IP estándar para realizar la acción de “forwarding”.

4. Una vez que la etiqueta es quitada, el paquete es enviado utilizando ruteo IP.

La clasificación de un paquete en una FEC es realizada cuando un paquete entra en la red MPLS y no en cada nodo (como se realiza en el ruteo IP). Un LSR necesita solamente mirar la etiqueta de largo fijo para saber que interface de salida utilizar. Pueden existir distintas etiquetas en un LSR para un mismo destino IP, lo cual es lo mismo que decir que hay varios LSPs para el mismo destino.

El plano de control es idéntico en IP y MPLS ya que los LSR utilizan los protocolos de ruteo IP para construir sus tablas de ruteo. Un LSR tiene el trabajo adicional de asignar etiquetas para cada destino encontrado en la tabla de ruteo y publicar el mapeo entre etiqueta y FEC a los LSRs adyacentes.

Las redes MPLS utilizan una variedad de protocolos de señalización para distribuir etiquetas:

- **LDP** – Utilizado en todas las redes MPLS [29].
- **iBGP** – Utilizado para el servicio de L3 VPNs [30].
- **RSVP** – Utilizado para Ingeniería de Tráfico (TE) [31], [32].
- **LDP Dirigido** – Utilizado para el servicio de L2 VPNs [29].

LDP ("Label Distribution Protocol") utiliza el puerto TCP/646 es utilizado en todas las redes MPLS para distribuir etiquetas para todos los prefijos IP que se encuentran en la tabla de ruteo. En el ejemplo de la Figura 26, el LSR D y el LSR B establecen una sesión LDP. El LSR D está conectado a la red 192.168.2.0/24 del cliente y publica este prefijo a todos sus vecinos. Además, el LSR D envía una etiqueta al LSR B para la red 192.168.2.0/24. Cuando el protocolo de ruteo del LSR B converge y ve a la red como alcanzable, envía la etiqueta 22 al LSR C. Este proceso continúa hasta que el LSR A recibe una etiqueta desde el LSR C.
El conjunto de etiquetas desde el LSR A hasta el LSR D forma un LSP. Un LSP es unidireccional. Existe otro LSP, identificado por un conjunto de etiquetas distintas para el tráfico de retorno desde el LSR D hasta el LSR A.

Para que el LSP desde el LSR A hasta la red 192.168.2.0/24 sea funcional se deben dar las condiciones:

- El protocolo de ruteo del "backbone" debe converger para que el LSR A tenga una ruta hasta la red 192.168.2.0/24

- LDP debe converger para que las etiquetas se propaguen por toda la red.

Luego que el LSR A dispone de toda la información necesaria para enviar datos a través de la red MPLS, encapsula los paquetes salientes con un encabezado definido en el RFC 3032, el cual se inserta entre el encabezado L2 y el encabezado L3.

El encabezado (por mayor detalle ver la Figura 27) de MPLS es simple y se compone de:

- La etiqueta que define un espacio de direcciones de 20 bits.
- Los bits de TC ("Traffic Class") que se utilizan para QoS (fueron renombrados como "Traffic Class" en el RFC 5462).
- El bit S se prende en la etiqueta más interna cuando hay más de una etiqueta en el paquete ("stack").
- El campo TTL ("Time to live") es análogo al TTL de IP.

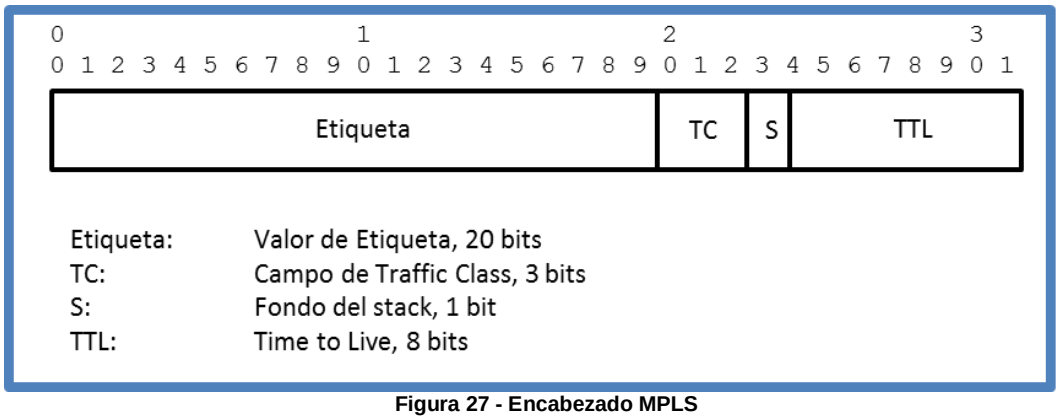

Figura 27 - Encabezado MPLS

Existen aplicaciones como las MPLS VPNs y MPLS FRR ("Fast ReRoute") que involucran varios niveles o "stack" de etiquetas. Un LSR realiza la acción de "forwarding" basado únicamente en la etiqueta más externa. Las etiquetas internas son utilizadas en aplicaciones como por ejemplo las MPLS VPNs de L2 o L3.

MPLS agrega una nueva vía de "forwarding" en el router. La FIB y la RIB contienen únicamente prefijos IP. LDP guarda etiquetas en una tabla llamada "Label Information Base" (LIB) y los valores de las etiquetas son agregados a la información de "forwarding" existentes en una tabla llamada "Label Forwarding Information Base" (LFIB). LDP debe mantener una entrada para cada ruta no-BGP de la tabla de ruteo y para todas las etiquetas publicadas por los vecinos LDP. La tabla LFIB se crea realizando una combinación de las tablas LIB y FIB. Dado un prefijo, si existe una etiqueta en la LIB se recibió desde un vecino LDP determinado por la FIB; esta etiqueta debe instalarse en la LFIB y utilizarse para "forwarding".

4.4 Virtualización del plano de control

La virtualización del plano de datos crea múltiples redes lógicas aisladas sobre una única infraestructura física compartida. Para mover los paquetes entre estas redes virtuales se necesita un protocolo de ruteo que sea consciente de este escenario.

El plano de control virtualizado más familiar es seguramente el "Per Vlan Spanning Tree" (PVST), que tiene una instancia de "spanning-tree" separada por cada VLAN configurada en un "switch". Si bien PVST existe desde antes que se popularizara el término virtualización, ejemplifica de manera muy clara el hecho que múltiples redes lógicas tienen topologías distintas y por lo tanto distintos caminos óptimos. Los "switches" deben realizar distintos cálculos de "spanning-tree" para cada red.

La virtualización del plano de control tiene dos aspectos, por un lado se definen las extensiones a los protocolos para que sea posible ejecutar varias instancias de ellos y por otro lado las utilidades que ejecutan sobre el plano de datos deben ser conscientes de la existencia de estas múltiples instancias de protocolos de ruteo.

4.4.1 Ruteo utilizando VRFs

Las implementaciones de los protocolos de ruteo deben ser conscientes de la existencia de las VRFs, lo que significa que comprenden que ciertas rutas deben ser instaladas únicamente en ciertas tablas de ruteo. Los protocolos de ruteo manejan esto haciendo vecindad con una topología restringida, donde una instancia de un protocolo de ruteo en una VRF hace vecindad con otras instancias en la misma red virtual. No se agrega información especial a las

publicaciones de las rutas para identificar los nombres de las VRFs, por lo que las distintas instancias deben comunicarse sobre vínculos privados.

Algunos protocolos de ruteo (por ejemplo BGP) permiten manejar en una única instancia de ruteo, múltiples VRFs. Otros protocolos (por ejemplo OSPF [33]) utilizan un único proceso por cada VRF. En ambos casos, cada VRF requiere que se realicen cálculos de optimización de rutas, por lo que un número elevado de VRFs tiene un impacto en la performance del dispositivo de red.

5. Arquitecturas de segmentación de redes virtuales

5.1 Introducción

Este capítulo se basa en las tecnologías presentadas en el capítulo anterior, organizadas en distintas arquitecturas y de esta manera proveer redes virtuales (“Virtual Private Network” – VPN). Se examina las distintas alternativas de utilización de los protocolos en las distintas arquitecturas. Las primeras arquitecturas que se presentan son las más simples (“Hop to Hop”); luego se cubren las distintas maneras en que se pueden realizar recubrimiento (“overlay”) de túneles para construir VPNs L2 y L3. Finalmente se presenta el modelo de peer VPN y dentro de esta clasificación el caso más notable que son las VPNs basadas en el RFC 2547 (y otros RFCs asociados – ver 5.4.1). En estos dos últimos casos se exploran distintos protocolos de transporte utilizados para la implementación de las VPNs mencionadas.

La realización de este capítulo se basó principalmente en los capítulos 5 y 6 de [1], en [17], en distintos estándares (p.e. [34], [35], [36], [37], [38], [4], [39], [40]) y otras fuentes.

5.2 VPN Hop to Hop (H2H)

Las VPNs h2h es una arquitectura simple donde cada nodo perteneciente a la VPN es miembro de todas las redes virtualizadas.

5.2.1 VPNs H2H L2

El ejemplo clásico de este tipo de VPN es una red switcheada donde cada uno de los “switches” en la red pertenece a todas las VLANs definidas. De esta manera se logra que una estación en particular pueda pertenecer a una VLAN particular de manera independiente a su ubicación física. En este tipo de escenario representado en la Figura 28, las VLANs abarcan toda la empresa y cada “switch” se configura como un dispositivo L2 ejecutando “Spanning Tree Protocol”.

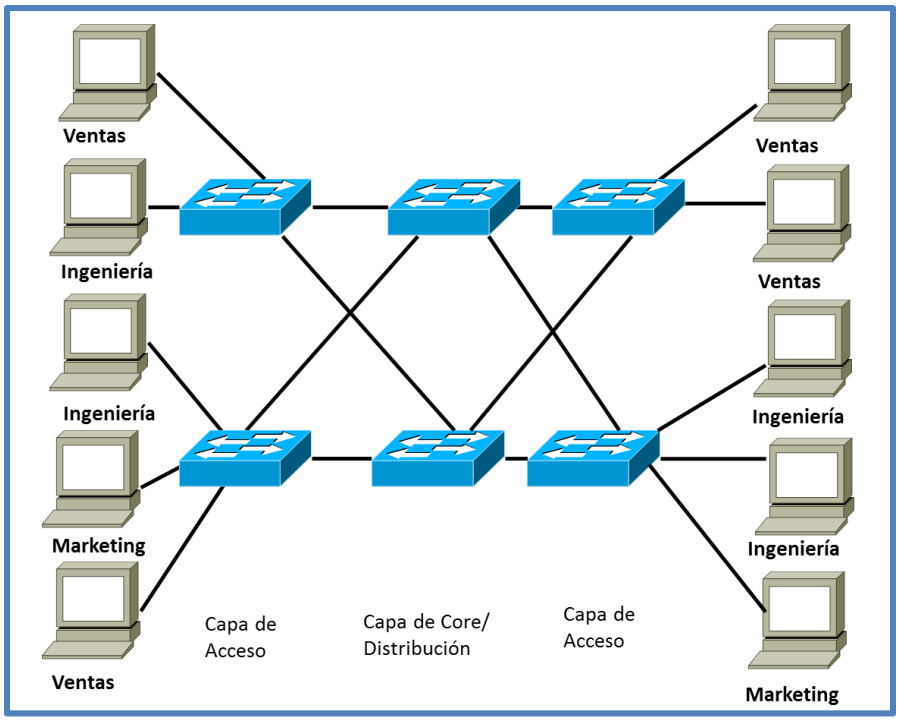

Figura 28- VPN H2H L2

La complejidad de este escenario comprende dos dimensiones; uno depende de la implementación y otro depende de la arquitectura. La limitación de implementación refiere al hecho de ejecutar STP ("Spanning Tree Protocol") a lo largo de la red. A menos que la red sea pequeña, STP es conocido por tener tiempos de convergencia altos, lo cual lo hace un protocolo lejos de lo óptimo para calcular caminos a lo largo de una red grande. Si bien existen optimizaciones (PVST+, RSTP, 802.1 w), esta arquitectura tiene problemas de diseño por sí misma. La limitación de diseño proviene del hecho que en el diseño de la Figura 28 en la red de core que se basa en Ethernet y sitios cliente con un único usuario. La ubicación de los sitios es impredecible ya que los lugares de los usuarios no son asignados por el departamento de redes.

5.2.2 VPNs H2H L3

Si se reemplaza el plano de control de la arquitectura y cada nodo ejecuta IP y algún tipo de protocolo de ruteo y se mantiene la virtualización del espacio de direcciones utilizando VRFs, se obtiene una solución H2H de capa 3 (esquematizado en la Figura 29). Es posible realizar una solución híbrida con acceso L2 realizando un mapeo de VLANs a L3 VRFs. En la solución se siguen utilizando VLANs, pero al utilizar dispositivos L3, se está removiendo la ejecución de "spanning tree" en toda la red, lo cual hace que el diseño sea más escalable. Como protocolo de ruteo, es posible que se use OSPF aunque la única restricción es contar con un protocolo de ruteo que soporte y entienda VRFs.

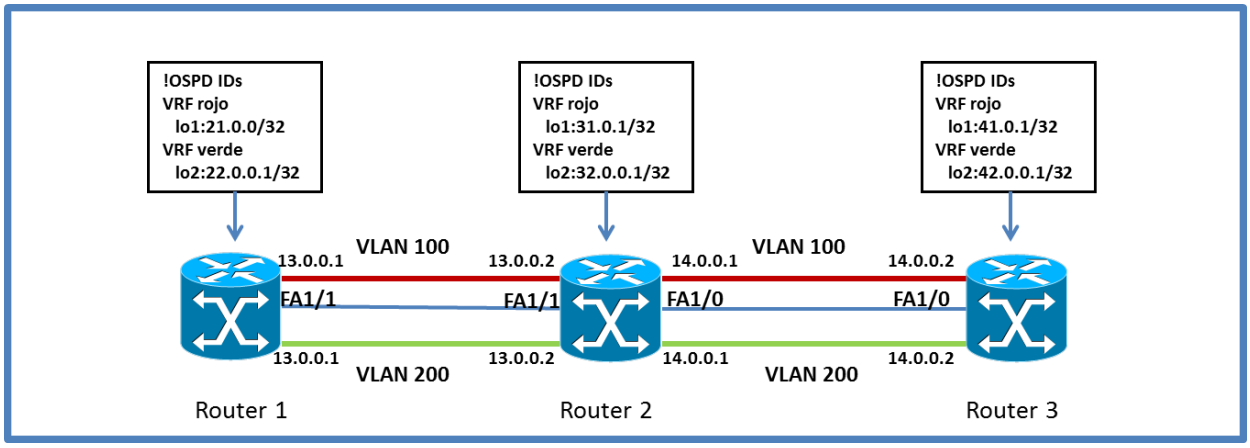

Figura 29 - VPN H2H L3

Para implementar este tipo de arquitecturas, se deben utilizar VRFs en cada nodo ruteado y la interconexión de las mismas mediante un circuito L2 como por ejemplo 802.1q (como se detallan la Figura 30). Como esta técnica requiere que las VRFs se encuentren configuradas y mapeadas a circuitos L2 en cada nodo, esta técnica generalmente se utiliza para segmentar la porción ruteada de redes de campus. Para las redes WAN IP (donde no se tiene control de cada nodo IP) existen otras alternativas técnicas.

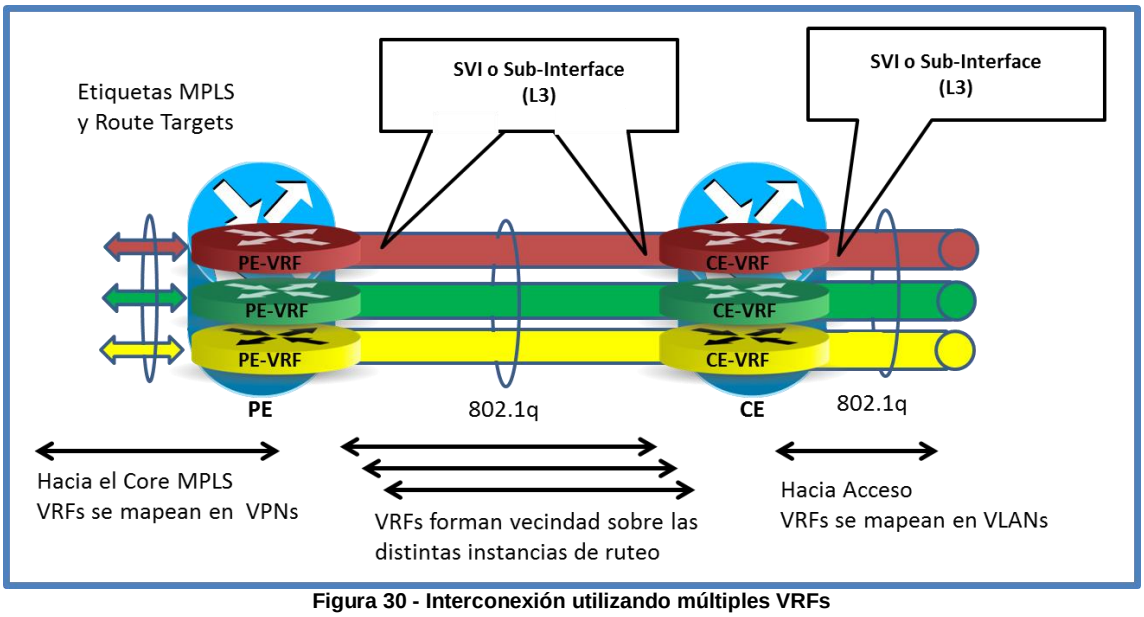

Figura 30 - Interconexión utilizando múltiples VRFs

5.3 Overlay VPNs

En este tipo de VPN, el proveedor brinda al cliente de un conjunto de líneas dedicadas emuladas. Estas líneas se llaman VC ("Virtual Circuit" – Circuito Virtual) y pueden estar disponibles constantemente o bajo demanda. El cliente establece comunicaciones del tipo router-router entre los equipos ubicados en sus distintos sitios sobre los VCs entregados por el proveedor. La información de ruteo se intercambia entre los equipos del cliente y el proveedor no tiene conocimiento de la estructura de la red del cliente. Este tipo de VPNs pueden ser implementadas en L1 utilizando líneas dedicadas/discadas, en L2 utilizando tecnologías como X.25/Frame Relay/ATM o en L3 utilizando túneles GRE o IPsec.
Cuando se implementan este tipo de VPNs en L2, el proveedor es responsable por brindar los VCs L2 entre los distintos sitios. El cliente es responsable por la capa IP y las superiores.

En el presente documento nos vamos a centrar en el uso de tecnologías IP para el armado de VPNs.

5.3.1 Overlay VPNs de L3

5.3.1.1 Overlay VPNs utilizando GRE e IPsec

Una topología básica para una "Overlay VPN" L3 basada en túneles es la arquitectura denominada "hub-and-spoke" donde existe un nodo central llamado "hub" a donde se encuentran conectados todos los otros sitios llamados "spokes".

La ventaja de este tipo de arquitecturas es que la operación de los routers en los "spokes" es sencilla, deben enviar todo el tráfico al router que se encuentra en el "hub" y utilizan para esto una ruta por defecto. El router ubicado en el "hub", debe enviar el tráfico al "spoke" correcto y anunciar el rango de direcciones correspondientes a cada "spoke" al resto de la red, generalmente agregando los prefijos utilizados en cada sitio (la posibilidad de agregación de prefijos va a depender del plan de direccionamiento de la red).

El aprovisionamiento de ésta solución es simple ya que la configuración es casi idéntica en cada router "spoke".

La solución con GRE e IPsec utiliza enlaces tipo P2P entre los equipos de la empresa y cifra el tráfico GRE con IPsec. Esta combinación permite utilizar GRE sobre redes públicas. La implementación con GRE permite utilizar "multicast", protocolos de ruteo dinámicos (OSPF utiliza "multicast"), e incluso transportar tráfico no IP entre los distintos sitios de la empresa. El hecho de agregar IPsec, si bien resuelve temas de seguridad, agrega cierta complejidad a la solución:

- Se instala un canal de control (Internet Key Exchange – IKE) entre el "hub" y cada "spoke".
- Se necesita realizar en la configuración de los dispositivos las asociaciones de seguridad (SA) que cifra el tráfico de paquetes GRE.

En la configuración de los dispositivos, se debe destacar que en el "hub" se debe contar con una interface tipo "Tunnel" para cada "spoke", por lo tanto el despliegue de un nuevo "spoke" requiere modificar la configuración en el "hub". Esto no solo agrega un paso en el despliegue sino que la configuración del "hub" puede crecer, lo cual puede dificultar las tareas de "troubleshooting".

La principal limitación de esta arquitectura depende de los patrones de tráfico de red. Si el tráfico fluye entre los "spoke" y el "hub" (en donde seguramente se provea el acceso a Internet y/o se encuentre el Datacenter de la empresa) todo fluye naturalmente y la topología es eficiente. Si hay tráfico entre los "spokes", la topología se vuelve ineficiente ya que todo el tráfico pasa por el "hub", creando un cuello de botella.

En el caso de redes con alto tráfico, se pueden generar cuellos de botella en casos que el tráfico fluye desde los "spokes" hacia el "hub" ya que la performance puede estar limitada por el router en el "hub". Si se tiene este tipo de limitación, una alternativa es utilizar múltiples routers en el "hub" y balancear la carga entre ellos como lo muestra la Figura 31.

En el ejemplo de la Figura 35, además de solucionar posibles problemas de performance, agrega cierto grado de alta disponibilidad a la topología. Si el router HUB1 tiene un inconveniente, el tráfico desde el SITIO1 puede ser redirigido hacia HUB2. Una vez solucionado el problema, el tráfico puede ser restaurado a su estado original.

La manera más común de realizar un "fail over" sobre túneles GRE es utilizar protocolos de ruteo para detectar cuando un túnel se encuentra caído; para este fin se configuran los túneles de manera estática en ambos "hubs" y hacer que el camino de respaldo tenga un costo más alto en el protocolo de ruteo que el camino principal. Los tiempos de convergencia van a estar de acuerdo con los del protocolo de ruteo utilizado.

DMVPN o "Dynamic Multipoint VPN" [41] es un tipo de túneles dinámicos que conforman una VPN soportada en routers Cisco. DMVPN provee la capacidad de crear túneles de manera dinámica (entre spokes) sin tener que pre configurar de manera estática todos los posibles túneles y end points. Soporta IPsec e ISAKMP ("Internet Security Association and Key Management Protocol" [34]). DMVPN se configura inicialmente para construir una topología del tipo "hub-and-spoke" configurando los "hubs" en los "spokes". No se necesita cambio de configuración en el "hub" para aceptar nuevos "spokes". Utilizando esta topología inicial, los túneles entre "spokes" se construyen bajo demanda sin configuración adicional. Esta funcionalidad permite el tráfico entre los "spokes" sin necesidad de ningún tipo de carga en el "hub".

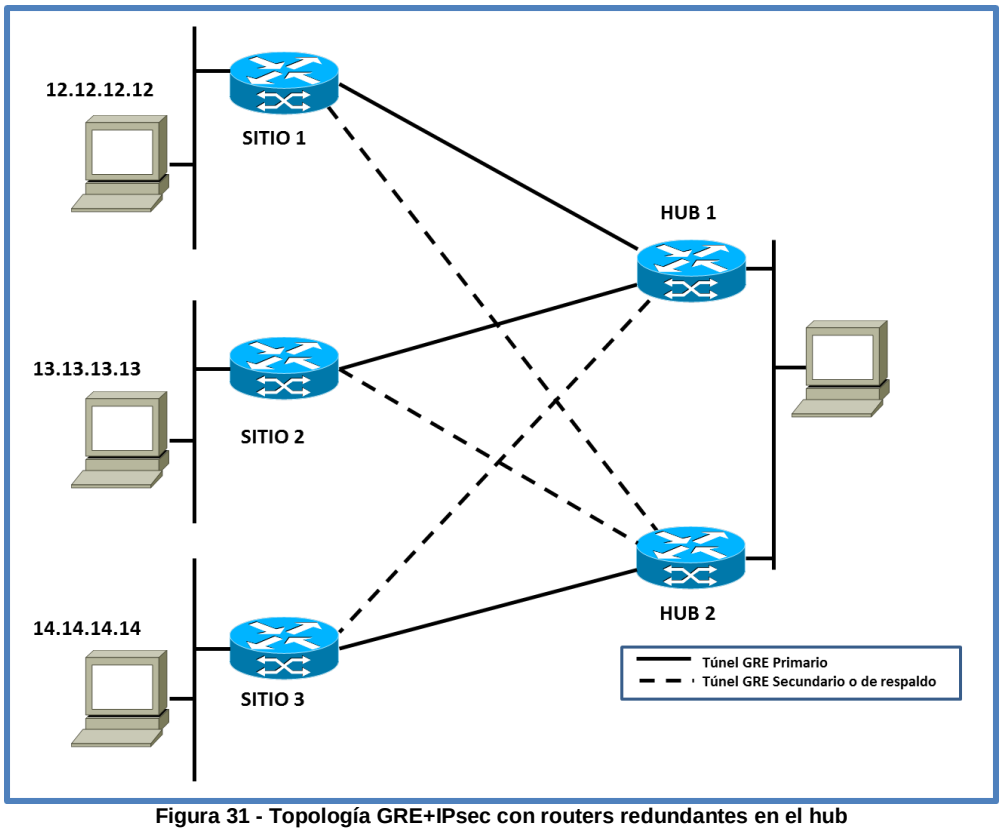

Figura 31 - Topología GRE+IPsec con routers redundantes en el hub

DMVPN es una combinación de las siguientes tecnologías:

- "Multipoint GRE" (mGRE) [42].
- "Next-Hop Resolution Protocol" (NHRP – RFC 2332 [35])
- Protocolos de ruteo dinámico (RIP [43], OSPF [33], BGP [30]).
- Cifrado utilizando IPsec [11].
- "Cisco Express Forwarding" (CEF) [44].

La primera ventaja de mGRE es que la configuración en el Hub se convierte en algo simple. Solamente se configura un único túnel estático, como se detalla en la Figura 32.

```
Interface Tunnel0
  description VPN hacia los sitios
  ip address 192.168.1.1  255.255.255.0
  no ip redirects
  ip nhrp map multicast dynamic
  in nhrp network-id 1007
  in nhrp holdtime 600
  ip nhrp server-only
  no ip split horizon eigrp 302
  tunnel source Serial0/0
  tunnel mode gre multipoint
tunnel key 1007
```

Figura 32 - Configuración del Tunnel mGRE en el Hub

La segunda ventaja es la capacidad de mGRE para desplegar dinámicamente los túneles. Si dos “spokes” necesitan comunicarse, el primer paquete se envía al Hub junto con un requerimiento de “Next Hop Routing Protocol” (NHRP – RFCs 2332, 2735 [36]) requiriendo la dirección IP del sitio. Cuando el router en el sitio recibe la respuesta de NHRP, construye el túnel directamente hacia el router de destino como se muestra en la Figura 33.

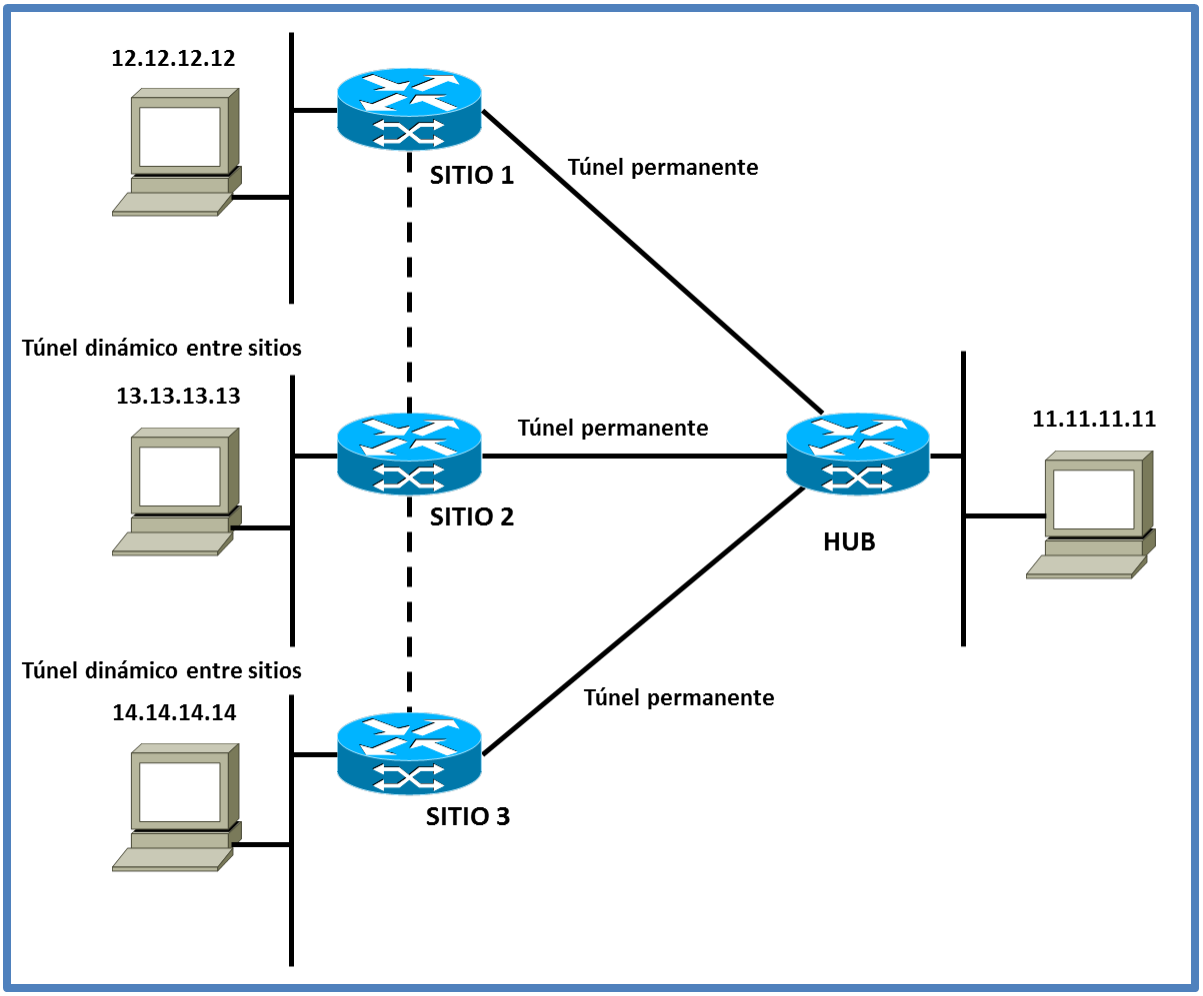

Figura 33 - Ejemplo de topología mGRE

NHRP (definido en el RFC 2332 [35] y extendido en el RFC 2735 [36]) es un protocolo de capa 2 para la resolución de direcciones similar a ARP. NHRP le indica a mGRE hacia donde debe tunelizar un paquete para poder alcanzar un cierto destino. NHRP es un protocolo cliente/servidor donde el servidor se encuentra en el “hub” y los clientes en los “spoke”. En el “hub” se mantiene una base de datos donde se registra para cada “spoke” el mapeo entre la dirección física usada como destino del túnel GRE y la dirección lógica a la interface tipo Tunel en el “spoke”. Cada “spoke” brinda esta información al “hub”, enviando un mensaje de registro NHRP en el momento de inicio del sistema.

Como la solución mGRE utiliza el mismo “stack” de protocolos, que el GRE “común”, el plano de datos no cambia. Existen túneles “inter-site” donde antes no existía nada y se puede utilizar NHRP para administrar la configuración de los túneles.

Si existen requerimientos de seguridad, se puede desplegar IPsec en conjunto con mGRE como lo hacíamos con GRE. El hecho de crear los túneles de manera dinámica conlleva una implicación de seguridad para la arquitectura ya que las credenciales criptográficas utilizadas

por el “hub” y los “spokes” deben coincidir. Con DMVPN no es posible conocer de antemano cual router va a establecer un túnel con cual otro router. Las alternativas son listar estáticamente cada dirección IP posible (lo cual es tedioso) o permitir a cualquier dirección IP establecer una sesión IPsec entrante. DMVPN utiliza la segunda alternativa, lo que establece un nivel de seguridad un poco más bajo. Sin embargo, una sesión IPsec se establece solamente si el dispositivo que se conecta dispone de las credenciales de seguridad correctas.

5.3.2 Overlay VPNs de L2

Una VPN de L3 puede no ser la solución adecuada para una red empresarial en algunos casos

- Las soluciones del tipo legadas que se basan en protocolos que no pueden ser transportados de manera nativa en L3. Para estos casos una VPN L2 es una manera de permitir que el resto de la red transicionar hacia IP mientras que esta parte de la red permanece en L2, permitiendo el uso de los sistemas legados.

- Algunas aplicaciones no son tolerantes a las características de latencias de la red L3 y necesitan de transporte sobre una red L2. Un ejemplo de esto es el intercambio de mensajes en un “cluster” para computar su estado y tecnologías de replicación de data center.

La característica más importante de las VPN L2 es que transportan paquetes L2 (“frames”) entre los sitios. Esto tiene sus ventajas y sus desventajas. La ventaja es que una L2 VPN es agnóstica acerca de los protocolos de niveles superiores que se utilizan en la red y necesitan menos recursos de los dispositivos que se encuentran en los extremos de la red. La desventaja es la ausencia del plano de control de L3 para manejar el “broadcast” en los segmentos de red y la alcanzabilidad en la VPN.

En la mayoría de escenarios empresariales, los dispositivos locales del cliente son “switches” Ethernet lo que implica que cuando se genera un “broadcast” en un sitio, debe ser transportado a todos los sitios. Como en cualquier tipo de VPN, un buen administrador de red deseará disponer de caminos redundantes entre los sitios. En una VPN L3, se dispone de sofisticados protocolos de ruteo para manejar caminos múltiples. En una VPN L2 se dispone de STP.

Se pueden construir VPNs L2 utilizando tecnologías P2P y multipunto. Por razones de escalabilidad que vimos anteriormente, es más simple construir topologías P2P que crear redes multiacceso sobre una infraestructura WAN.

5.3.2.1 L2TPv3 (P2P)

Una VPN L2 basad en L2TPv3 utiliza “pseudowires” P2P para transportar “frames” L2 entre los distintos sitios. Los “frames” L2 pueden ser del tipo 802.1q, Frame Relay, ATM o HDLC. En estos casos el dispositivo CE generalmente es un “switch” y el dispositivo PE generalmente es un router con soporte de L2TPv3. En el ejemplo que se presenta en la Figura 34 se utilizan túneles L2Tpv3 para construir una topología “hub-spoke”.

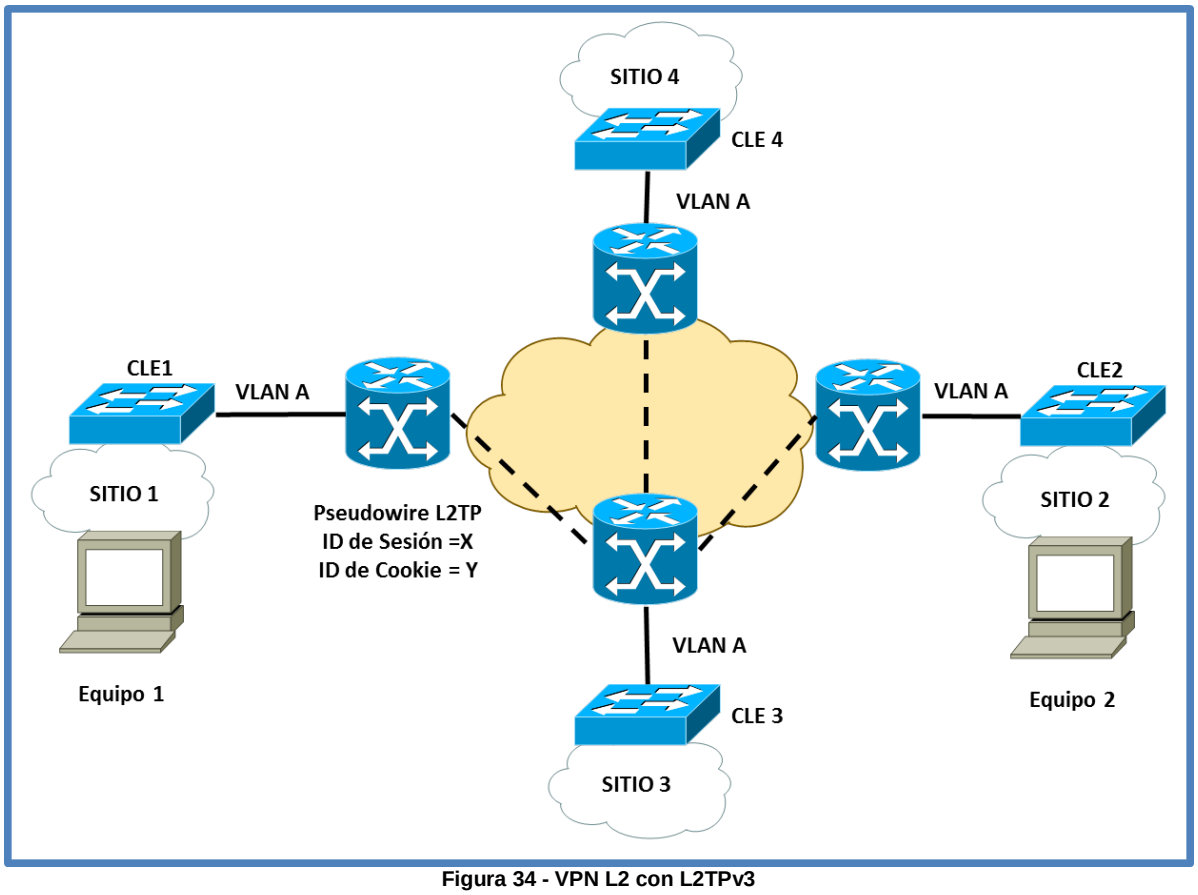

Figura 34 - VPN L2 con L2TPv3

El Sitio 3 es el Hub y los sitios 1, 2 y 4 son los “spoke”. Existe un canal de control L2TPv3 entre cada par “hub-spoke”, que los routers vecinos utilizan para negociar un ID de sesión y una cookie para cada VLAN o puerto asociado a un túnel L2TPv3. Una gran ventaja de esta solución es que la VPN y su topología son totalmente transparentes a los equipos CLEx, los que están configurados como cualquier equipo L2 en la red local.

Los equipos CLEx en esta red, van a ejecutar “spanning tree” por la VPN. Algunos proveedores de este tipo de servicio no permiten el tráfico de las BPDU (“Bridge Protocol Data Unit”), por lo que no es posible ejecutar “spanning tree” sobre los vínculos WAN.

La red IP de “core” sobre la cual se monta la solución L2TPv3 provee redundancia de caminos entre los sitios. Si el “core” puede converger rápidamente, el túnel hasta puede cambiar su camino sin siquiera se caigan las sesiones. En cualquier caso, el STP del cliente no necesita recalcular caminos alternativos sobre la VPN; el transporte L2TPv3 se encarga de esto.

Las topologías de VPN L3 (“Overlay VPN”) utilizan el plano de control de IP para crear la topología “hub-spoke”. En cada “spoke”, la ruta por defecto apunta hacia el “hub” que a su vez tiene rutas para los distintos prefijos IP de cada “spoke”. La mayoría de los protocolos de L2 no funciona de esta manera. Por ejemplo Ethernet realiza un “flood” de paquetes hacia todos sus vecinos.

Volviendo a la topología de ejemplo supongamos que el Host1 en el Site1 se desea comunicar con el Host2 en el Site2. Los sitios se encuentran interconectados mediante el Site3. Ambos Hosts se encuentran sobre la VLAN1, la cual se encuentra configurada también en el Site4.

- Cuando el Equipo 1 la solicitud ARP, ésta llega mediante "broadcast" a todos los sitios donde la VLAN se encuentra presente y todos los "switches" crean una entrada en la tabla de "mac-address", mapeando esta MAC al puerto correcto ("pseudowire").

- El Equipo 2 envía una respuesta de tipo "unicast", lo cual permite también al "switch" en el Site1 actualizar su tabla "mac-address".

- Cuando otro equipo en la misma VLAN envía datos, sucede el mismo proceso, con las acciones de "flooding" y aprendizaje sobre la VLAN.

En la topología de ejemplo no existen "loops", por lo que se podría deshabilitar STP sobre los vínculos de la WAN. En el caso que existan caminos redundantes, se requiere el uso de STP para bloquear uno de los dos caminos.

En lo que refiere al plano de datos, los RFCs asociados con L2TPv3 definen reglas de encapsulamiento según el protocolo de L2 que se transporte.

- RFC 4719 – "Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)" [45].
- RFC 4817 – "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3" [46].
- RFC 4349 – "High-Level Data Link Control (HDLC) Frames over Layer 2 Tunneling Protocol, Version 3" (L2TPv3) [47].
- RFC 4454 – "Asynchronous Transfer Mode (ATM) over Layer 2 Tunneling Protocol Version 3 (L2TPv3)" [48].
- RFC 4591 – "Frame Relay over Layer 2 Tunneling Protocol Version 3 (L2TPv3)" [49].

5.3.2.2 MPLS (P2P)

MPLS ofrece una manera alternativa para construir L2 VPNs.
MPLS puede ser visto como una serie de túneles P2P donde se utilizan etiquetas ("labels") en vez de "session IDs" para multiplexar el tráfico. De una manera análoga al escenario de L2 VPNs utilizando L2TPv3 una VPN L2 MPLS utiliza un protocolo del plano de control para intercambiar etiquetas para VLANs o puertos asociados a un "pseudowire" en un router PE.

Las dos cuestiones que se debe entender de su funcionamiento son la manera en que las etiquetas son intercambiadas entre los routers PE y la manera como los paquetes fluyen por el "core" de la red MPLS.

La implementación del intercambio de etiquetas es dependiente de la implementación. Las opciones son usar LDP ("Label Distribution Protocol") o BGP ("Border Gateway Protocol"). En MPLS, las etiquetas tienen significado local y a lo largo de un LSP, los distinto LSRs utilizan distintos valores. En una VPN L2 las sesiones del protocolo de intercambio de etiquetas se establecen entre los routers PE.

Cuando una interface (o VLAN) se asocia a un "pseudowire", el router PE crea una etiqueta para dicha asociación y la envía al PE remoto (en equipamiento Cisco se utiliza el comando "xconnect" para tal fin).

Los LSPs son unidireccionales, por lo que se utilizarán diferentes valores de etiquetas en cada extremo del “pseudowire” para una VLAN particular, en contraste con L2TPv3 que negocia un Id de sesión común y un Id de cookie.

Cuando un PE recibe un “frame”, es encapsulado en MPLS y la etiqueta que identifica esa VLAN o puerto es puesta en el “stack” de etiquetas (“label stack”). Esto no es suficiente ya que el próximo LSR espera una etiqueta que identifique la FEC para este paquete para encontrar la etiqueta de salida. Debido a esto, se agrega una segunda etiqueta correspondiente a la dirección IP del PE remoto. Esta etiqueta será intercambiada en cada hop y quitada del “stack” en el penúltimo hop antes que el paquete egrese de la red MPLS. La Figura 35 nos muestra cómo se utilizan etiquetas de MPLS para conformar un túnel L2TP y construir una VPN L2.

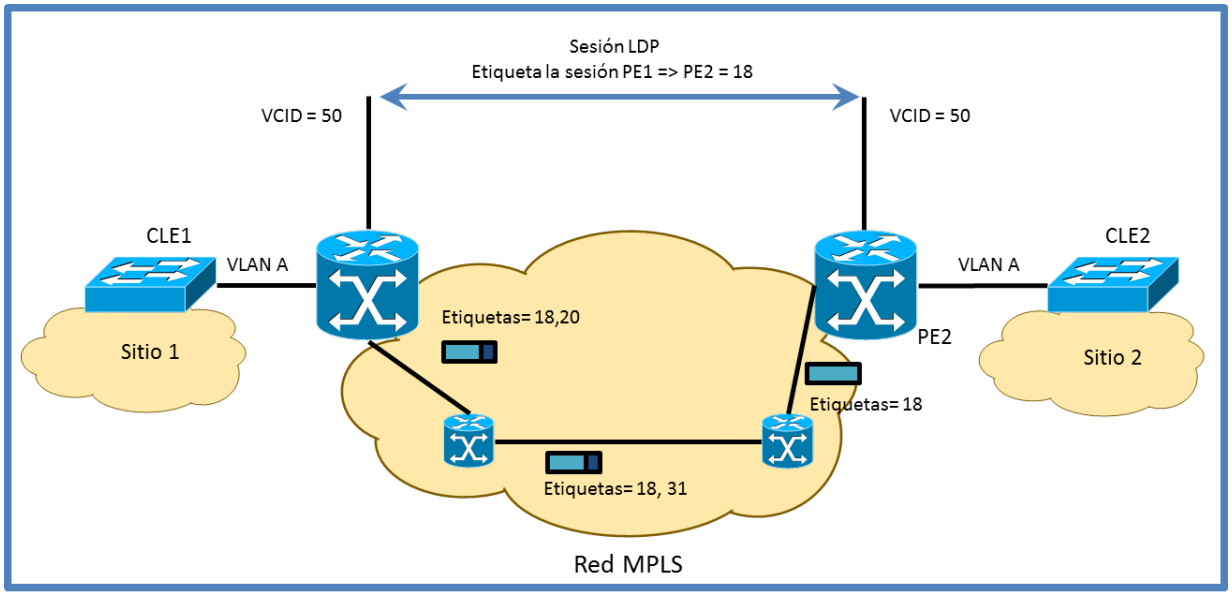

Figura 35 - L2VPN basada en MPLS

En el caso particular de Ethernet, si desea virtualizar la L2 VPN, se realiza por VLAN. Los LSP o túneles L2TP se comportan como interfaces asociados a VLANS 802.1q. El servicio de Ethernet VLAN L2 (el cual es el más común de encontrar como servicio comercial) es generalmente comercializado según las definiciones tomadas del Metro Ethernet Forum [50].

5.3.2.3 VPLS (MP2MP)

Desde un punto de vista de la arquitectura, VPLS (“Virtual Private Lan Service”) usa un “full mesh” de “pseudowires” MPLS para emular un switch Ethernet. La operación es la misma que un “bridge” Ethernet ya que reenvía “frames” utilizando direcciones MAC de destino, aprende direcciones de origen e inunda (“floods”) “frames” desconocidos o de “broadcast/multicast”.

Existen varios documentos (propuestas de estándares) con relación a este servicio. Los principales son:

- RFC 4761 [51]
  En este documento se describen las funciones requeridas para ofrecer un servicio de VPLS, un mecanismo de señalización para la VPLS y reglas para reenviar paquetes sobre una red conmutada de paquetes.

- RFC 4762 (Lasserre-Kompella) [52]
  En este documento las funciones del plano de control para la señalización de "pseudo-wires". Se describen también las funciones de "forwarding" del plano de datos haciendo detalle en el aprendizaje de las direcciones MAC.

- El encapsulado de los paquetes VPLS se describe en el RFC 4448 [53].

Implementar alguna de las características propias de Ethernet sobre una infraestructura P2P es un desafío. VPLS que es un servicio mp2mp implementado sobre "pseudowires" P2P intenta resolver estos temas de la siguiente manera:

- Auto-provisoning
  Los dispositivos pueden ser agregados a una red Ethernet sin una configuración previa. VPLS implementa aprovisionamiento automático de manera similar a un "switch" Ethernet. Cada dirección MAC nueva se envía ("floods") por todos los "pseudowires". La respuesta ARP se utiliza para crear una entrada en el PE correspondiente y mapear la MAC a un "pseudowire".

- Auto-discovery
  En el contexto de una VPN, existe el problema de proveer un mecanismo para que cada PE descubra sus vecinos de VLAN. Existen implementaciones que utilizan BGP (RFC 4761 – Kompella) para anunciar la pertenencia a una VLAN entre PEs. Otras ideas incluyen realizar una búsqueda en un repositorio central utilizando RADIUS [54] o DNS ("Domain Name System").

- Soporte Broadcast/Multicast
  El tráfico "broadcast" y "multicast" se trata de la misma manera. Los "frames" son enviados por todos los "pseudowires".

VPLS es una arquitectura compleja. Las VPNs L3 ofrecen una solución más escalable y robusta ya que mueve la tarea de ruteo desde el CE hacia la red del proveedor. Cuando se necesita conectividad L2, un conjunto de vínculos P2P suele ser suficiente.

5.4 Peer VPN Capa 3

El modelo de VPN Peer-to-Peer adopta un esquema de ruteo simple para el cliente. La red del proveedor y del cliente utilizan el mismo protocolo de red (generalmente IP) y todas las rutas del proveedor son trasladadas por el "core" de la red del proveedor. Los routers PE intercambian rutas con los routers CE y se crean vecindades de ruteo entre los routers PE y CE en cada sitio. EL agregado de nuevos sitios a la VPN es sencillo en comparación con el modelo de "Overlay". Debido a que el proveedor participa en el ruteo del cliente, las direcciones IP deben ser asignadas por el proveedor, por lo que el uso de direcciones privadas no es factible.

Para poder implementar este tipo de VPN, existen dos aproximaciones posibles:

1. Router PE compartido
   Existe un router PE compartido que transporta todas las rutas del cliente. Las rutas se separan con comunidades y filtros de rutas ("ACLs", "route maps", etc) en la interface del PE al CE. La complejidad de esta configuración resulta en altos costos de mantenimiento, utilización de CPU y requerimientos de memoria.

2. Router PE dedicado
   En este modelo, cada cliente dispone de un router PE dedicado que transporta únicamente sus rutas. La separación entre clientes mediante la falta de información de ruteo en el router PE. Los routers P contienen todas las rutas del cliente y filtran las actualizaciones de rutas entre los distinto PE utilizando comunidades de BGP. Debido a que cada cliente tiene un router PE dedicado, este enfoque es caro de desplegar y por lo tanto no es una solución viable en costo.

Las MPLS IP VPNs es un modelo peer-to-peer que unifica las características de seguridad y segregación del modelo de "overlay" con la simplificación de ruteo a nivel de cliente que se utiliza en el modelo peer-to-peer. La arquitectura de las MPLS VPNs es muy similar al modelo del router PE dedicado, teniendo en cuenta que el router dedicado por cliente se implementa como tablas de ruteo virtualizadas dentro del router PE. En otras palabras, la segregación de clientes se logra mediante el concepto de "Virtual Routing and Forwarding" (VRF) donde el PE router es dividido en varios routers virtuales que atienden a distintas VPNs (o sitios de cliente). Esto permite la superposición de direccionamiento IP en distintos clientes ya que a cada cliente se le asigna una tabla de ruteo independiente. Los routers PE contienen la información de ruteo solamente para las VPNs que están directamente conectadas; como resultado de esto el tamaño de las tablas de ruteo se reducen de manera significativa y la cantidad de información de ruteo almacenada es directamente proporcional a la cantidad de VPNs conectadas al router PE. Como consecuencia de todo lo anterior, la tabla de ruteo en el router PE crecerá cuando el número de VPNs directamente conectadas crezca. El router de cliente participará en el ruteo del cliente asegurando ruteo óptimo entre sitios y un aprovisionamiento sencillo, sin embargo en este tipo de VPN no se requiere "routing" dentro del "backbone" del proveedor ya que se utiliza MPLS para el "forwarding" de paquetes y no IP tradicional.

La arquitectura de este tipo de VPNs fue propuesta por primera vez en el RFC 2547 (y su serie de RFcs asociados). La mayor diferencia es que los routers CE y PE forman una relación de pares. Los routers PE utilizan MP-BGP ("Multiprotocol BGP") [55] para publicar información de ruteo del cliente.

Desde el punto de vista de este trabajo, este tipo de tecnologías se encuentran sobre todo en proveedores de servicio que manejan una gran infraestructura de red MPLS y no en la empresa. Sin embargo es necesario tener conocimiento de estas tecnologías ya que desde la empresa se consumen estos servicios y se deben conocer sus bondades / limitaciones para un mejor aprovechamiento de los mismos.

5.4.1 RFC 2547

El primer *RFC* que el *IETF* publicó en 1999 sobre las *MPLS L3VPNs* es el *RFC* 2547 *"BGP/MPLS VPNs"* [37] como una forma de estandarizar el protocolo de "*Tag Switching"* desarrollado por Cisco en esa época. Este *RFC* fue corregido mediante "*drafts"* (borradores) publicados en 2003 y 2004 (*draft-ietf-2547bis-0x*). En estas revisiones se le fueron agregando temas tales como *"Multi-AS backbone(Inter-AS)"* y *"Carriers' Carriers"*. El "*draft"* finalmente fue aprobado como *RFC 4364 "BGP/MPLS IP Virtual Private Networks (VPNs)"* [38] en Febrero de 2006, el cual hizo obsoleto al *RFC 2547*.

El *RFC 4364* fue actualizado mediante los *RFC 4577 "OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)"* [4], el *RFC 4684 "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)"* [39] y el *RFC 5462 "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field"* [40], los

cuales no modificaron la esencia del *RFC 4364* sino que actualizaron ciertas aspectos puntuales del mismo.

A los efectos de la elaboración del presente documento, cuando nos referimos al *RFC 2547*, nos estamos refiriendo a la familia del *RFCs* que comenzaron con el *RFC 2547*.
En un escenario L3, este tipo de VPN reduce la complejidad en los routers CE. Los routers PE son el "Next Hop" para cada router CE; en comparación con las soluciones de VPN tipo "overlay", en este tipo de soluciones, el router PE debe realizar un trabajo mayor.

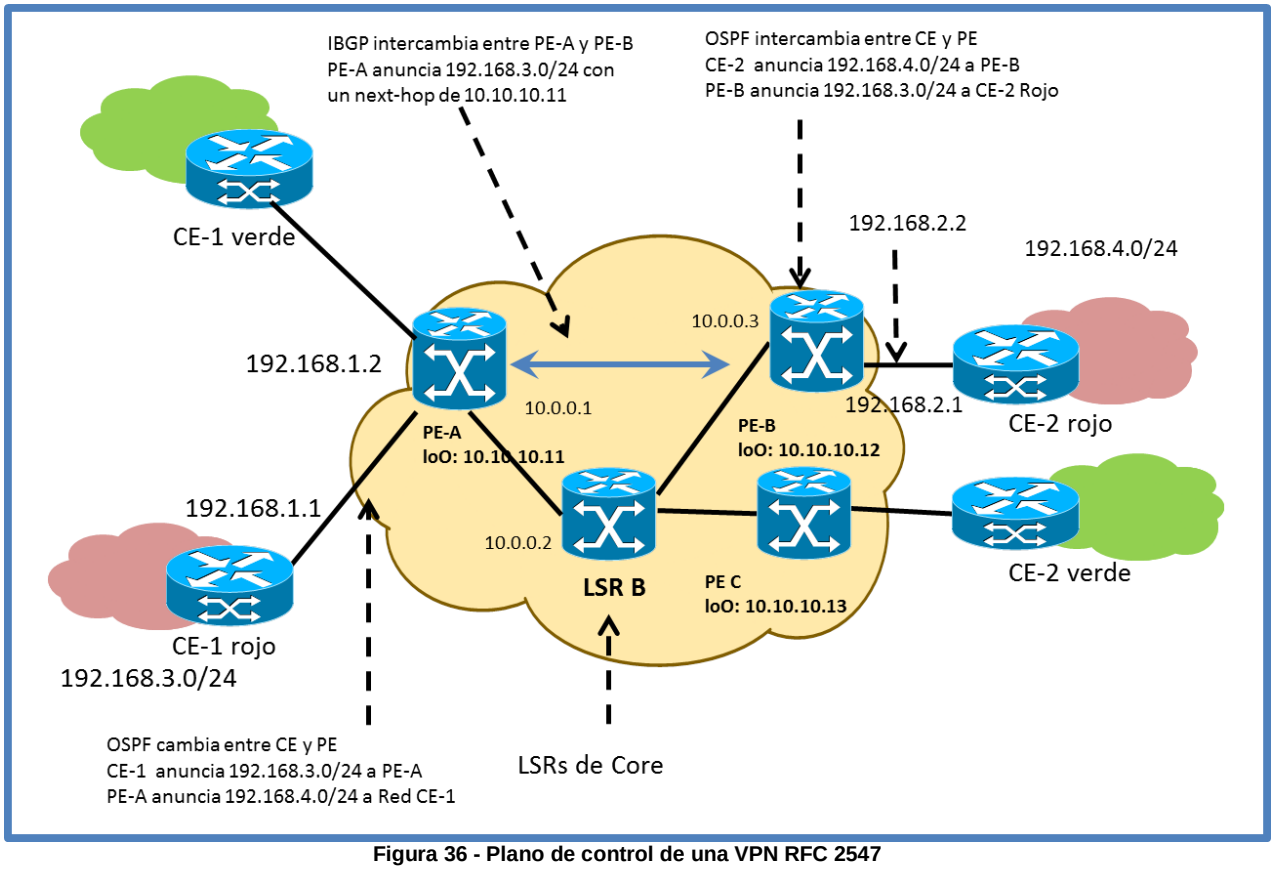

Figura 36 - Plano de control de una VPN RFC 2547

En el ejemplo de la Figura 36 se muestra una red con dos VPNs conectadas sobre la red de "core" de un proveedor. La red RED está compuesta por dos sitios que usan las redes 192.168.3.0/24 y la red 192.168.4.0/24; la VPN debe conectar estas redes entre sí.

El PE-A y el PE-B tienen vecindad con CE1 y CE2 respectivamente y utilizan un protocolo de ruteo dinámico (OSPF) para intercambiar rutas. CE1 publica la red 192.168.3.0/24 y CE2 publica la red 192.168.3.0/24 sobre el vínculo que conecta al router PE. Cuando los protocolos sobre la conexión VPN convergen, el "Gateway" para la red 192.168.3.0/24 en CE2 es 192.168.2.1 (PE B). Todos los vínculos CE-PE son parte del espacio de direcciones del cliente.

Cuando PE-A recibe una actualización de OSPF desde CE1, guarda información sobre la ruta en la tabla de ruteo correspondiente a la VRF roja. PE-A mantiene sesiones de MP-BGP abiertas con PE-B y PE-C. Para proveer alcanzabilidad de L3 sobre la red del proveedor, los PEs deben publicar las rutas de los clientes entre sí, por lo que para la VPN RED PE-A anuncia información de alcanzabilidad para el prefijo 192.168.1.3.0/24 a PE-B en un formato modificado llamado una dirección VPNv4. El PE-A también recibe actualizaciones de rutas desde sus vecinos y aprenderá que la red 192.168.4.0/24 es alcanzable a través de PE-B (10.10.10.12).

Para el funcionamiento de una VPN según el RFC 2547, BGP intercambia atributos adicionales según el detalle de la Figura 37.

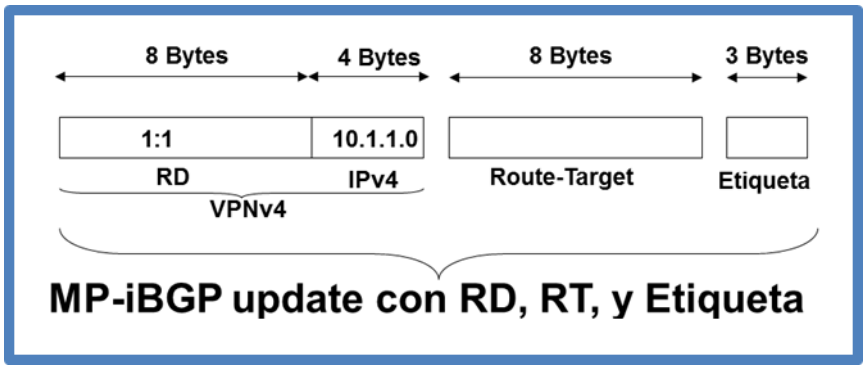

Figura 37 - Atributos de BGP en una VPN según el RFC 2547

Cada uno de estos atributos será explicado durante el desarrollo del tema.

El RFC 2547 establece la separación de direcciones entre distintas VPNs utilizando VRFs en las interfaces de los routers PE que implementan la conexión CE-PE y creando un espacio de direcciones denominado VPNv4 a lo largo de la red del proveedor. Estas direcciones surgen de concatenar un "Route Distinguisher" (RD) de 8 bytes con una dirección IP del espacio de direcciones del cliente (4 bytes).

Con el RD, se transforma una dirección IP de 32 bits en una dirección de 96 bits, única en la red. Los PEs no anuncian prefijos IP de 32 bits mediante MP-BGP sino prefijos de 96 bits. Los posibles formatos para el RD son que se encuentran en la Figura 38.

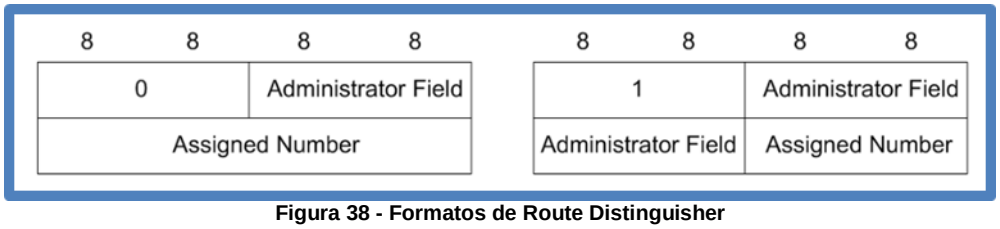

Figura 38 - Formatos de Route Distinguisher

El RFC 2547 define lo siguiente para los campos del RD:

- Tipo 0
  - "Administrator Filed" (2 bytes): Debe contener un número de sistema autónomo. Si es un número público de ASN, debe ser provisto por la autoridad competente.
  - "Assigned Number" (4 bytes): Es un número que proviene de un espacio de números administrado por la empresa a la que le fue asignada el ASN correspondiente.

- Tipo 1
  - "Administrator Filed" (4 bytes): Debe contener una dirección IP. Si es una IP pública, debe ser provisto por la autoridad competente.
  - "Assigned Number" (2 bytes): Es un número que proviene de un espacio de números administrado por la empresa a la que le fue asignada la IP correspondiente.

Cada VPN debe tener distintos RDs (pueden existir más de un RD para una misma VPN que se

utiliza en aplicaciones como ser el balanceo de carga entre distintos enlaces de un mismo CE hacia varios PE). Los PEs guardan rutas IP en tablas de ruteo correspondientes a VRFs e intercambian rutas del tipo VPNv4 con otros PEs. Cada PE se encuentra configurado para mapear un RD sobre una VRF. La sesión de iBGP utiliza este valor cuando intercambia los prefijos contenidos en la VRF con otros PEs. Los prefijos VPNv4 son utilizados únicamente en el plano de control ya que los paquetes mantienen su formato de dirección IP y los intercambios de rutas entre los CE y PE utilizan IPv4 normal.

Los PEs reescriben la dirección del próximo hop (un CE), reemplazándola con su propia dirección (dirección perteneciente al espacio de direcciones del proveedor). De esta manera, un PE puede enviar tráfico con destino a otra porción de una VPN del cliente sobre la red del proveedor utilizando búsquedas estándar sobre una tabla de ruteo ya que el PE en cuestión será el “next hop” para el prefijo publicado por el CE.

Cuando se desea enviar tráfico, hacia CE2, PE-A realizará una búsqueda para encontrar una ruta hacia 192.168.4.0 que en este caso resuelve a 10.10.10.12. PE-A debe realizar otra búsqueda en su FIB para determinar la ruta a tomar. En este caso es 10.0.0.2 que es un LSR.

Para ver este proceso en detalle, cuando PE-A recibe una actualización IPv4 desde CE1, lo traduce en una dirección VPNv4, realizando lo siguiente:

- Asigna un RT (RT=Red) según la configuración de la VRF.
- Reescribe el “NextHop” apuntando hacia sí mismo para el prefijo anunciado (192.168.3.0/24).
- Le asigna una etiqueta basado en la interface o la VRF (Et=100).

Una vez hecho esto, envía una actualización de MP-iBGP a los otros routers PE
PE-B recibe la actualización y verifica que RT=Red se encuentre configurado localmente dentro de alguna VRF. Si este es el caso, PE-B traduce la dirección VPN-v4 en una dirección IPv4 realizando:

- Instala el prefijo en la tabla de ruteo de la VRF que corresponda.
- Actualiza la tabla CEF de la VRF roja con etiqueta 100 para 192.168.3.0/24.
- Publica este prefijo IPv4 hacia CE2.

El modelo propuesto por el RFC 2547 crea un “full mesh” entre los distintos PEs de forma que cada CE puede alcanzar a otro CE en 2 saltos (“hops”). Es posible restringir la alcanzabilidad entre los CEs, incluso para crear una topología tipo “hub-spoke”, utilizando “Route Targets” (RTs). Los RTs son comunidades extendidas de BGP (64 bits) que son anunciadas entre los PE’s. Un PE puede estar configurado para exportar prefijos con un cierto RT e importar solamente prefijos que concuerdan con un RT dado. Los RTs permiten construir “mesh” arbitrarios entre sitios. La utilización de RTs hace factible la creación de distintas topología, combinando la exportación de rutas (“exports”) e importación de rutas (“imports”) en BGP.

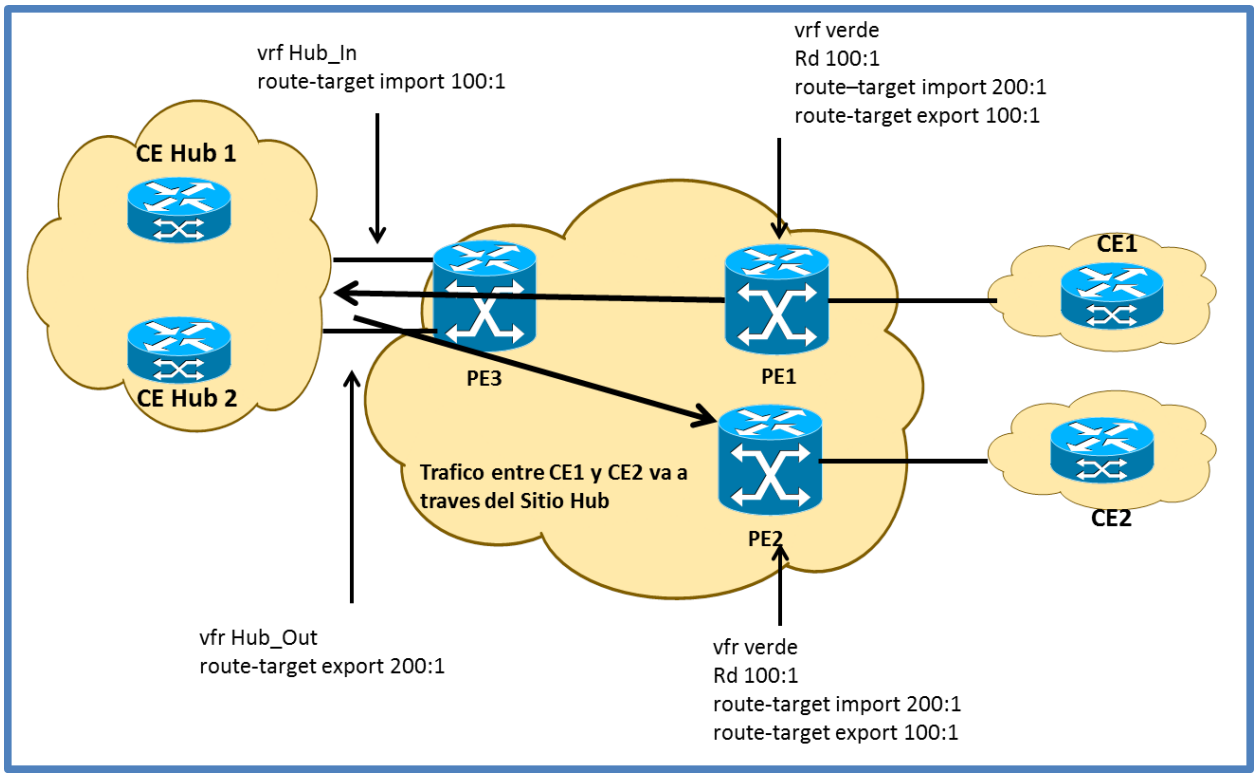

Figura 39 - Topología Hub & Spoke utilizando RTs

En el ejemplo de la Figura 39 los routers PE1 y PE2 exportan prefijos con un RT "Spoke" (100:1) y usan un RT "HUB" (200:1) para importar prefijos. PE3 importa las rutas 200:1 en la VRF_IN quien a su vez las distribuye hacia el router CE_HUB1. El router CE_HUB2 anuncia estas rutas hacia PE3, donde son instaladas en la VRF_OUT y luego exportadas con RT 200:1. Como resultado, el tráfico entre CE1 y CE2 es ruteado a través del HUB.

La secuencia de pasos de funcionamiento es la que sigue:

- PE1, PE2 y PE3 intercambian prefijos utilizando iBGP. PE1 importa rutas con RT 200:1, por lo que solamente las rutas provenientes de PE3 son cargadas en la VRF.
- PE3 importa rutas de PE1 y PE2 porque ellas tienen un RT de 100:1. Estas rutas con instaladas en la VRF_IN. PE3 exporta rutas recibidas desde CE_HUB2 a los otros vecinos BGP utilizando el RT 200:1
- Cuando CE1 envía un paquete a CE2, primero pasa por PE1.
- PE1 consulta el prefijo de CE2 en su VRF y descubre que el "next-hop" es PE3.
- El paquete se encapsula para atravesar la VPN y es enviado a PE3
- PE3 busca en su tabla de VRF y descubre que el "next-hop" para CE2 es HUB_CE1. El paquete es enviado sobre la red interna del HUB hacia HUB_CE2 quien tienen el "next-hop" para alcanzar CE2 hacia PE3, quien hace una consulta en VRF_OUT y descubre que el "next-hop" es PE2.
- PE2 realiza una consulta en la VRF correspondiente y envía el paquete IP hacia su destino sobre la interface que conecta con CE2.

El aprovisionamiento en una VPN RFC 2547 no es complicado, pero tampoco automático. Se debe configurar las rutas que anuncia un determinado PE y en todos los demás PEs se debe configurar para levantar una sesión iBGP con dicho PE. El RFC 2547 permite el uso de "Route Reflectors" (RRs), que remueve la conectividad del orden N^2 entre PEs, lo que ayuda a la escalabilidad de la solución y que el aprovisionamiento una operación única en cualquier PE ya que los PEs tienen vecindad BGP únicamente con los RRs.

La elección de BGP le da al RFC 2547 una arquitectura escalable y robusta. Debido a que el protocolo es utilizado en el "backbone" de Internet, es adecuado para redes de gran porte y a su vez dispone de políticas suficientemente flexibles. BGP fue extendido para trabajar con las arquitecturas del RFC 2547. El resultado de esta extensión se denomina "Multiprotocol BGP" (MP-BGP) y puede anunciar rutas VPNv4, VPNv6, IPv4 e IPv6. Los clientes son libres de elegir cualquier protocolo de ruteo que sea capaz de guardar rutas en una VRF (y por lo tanto consciente a la existencia de una VRF) en los vínculos CE-PE (en la práctica algunos proveedores fijan el protocolo CE-PE en sus servicios). La redistribución de rutas entre protocolos de ruteo dinámicos permite que las rutas sean anunciadas e importadas desde MP-BGP.

La MPLS-VPN es el caso más común de implementación del RFC 2547. El propio RFC describe el plano de control que utiliza MP-BGP y el plano de datos basado en MPLS. En la evolución del RFC 2547 (RFC 4364) menciona la posibilidad de tunelizar MPLS en GRE utilizando técnicas descritas en el RFC 4023 [56].
Adicionalmente a lo establecido en el RFC 4023, existe el RFC 4797 ("Use of Provider Edge to Provider Edge (PE-PE) Generic Routing Encapsulation (GRE) or IP in BGP/MPLS IP Virtual Private Networks") [57] que a modo informativo provee una estrategia de implantación de BGP/MPLS VPNs en redes cuyos dispositivo de borde son conscientes de la existencia de VPNs y MPLS pero los dispositivos interiores no. A su vez existe el draft-townsley-l3vpn-l2tpv3-01 ("BGP/MPLS IP VPNs over Layer 2 Tunneling Protocol" [58]) el cual plantea el uso de MPLS sobre L2TPv3 en el contexto de las VPNs del RFC 2547. En las próximas secciones, se analizará la utilización de MPLS en el plano de datos (la opción por defecto que plantea el RFC 2547 y sus RFCs derivados) y alternativas como L2TPv3 y mGRE.

5.4.1.1 Utilización de MPLS en el plano de datos de una VPN según el RFC 2547.

Las MPLS VPNs están implementadas utilizando 2 etiquetas. La más interna identifica la VRF del cliente y la externa identifica el "next hop" en el LSP hacia el PE de egreso. Para entender la operación de la VPN, se utilizará el ejemplo de la Figura 40.

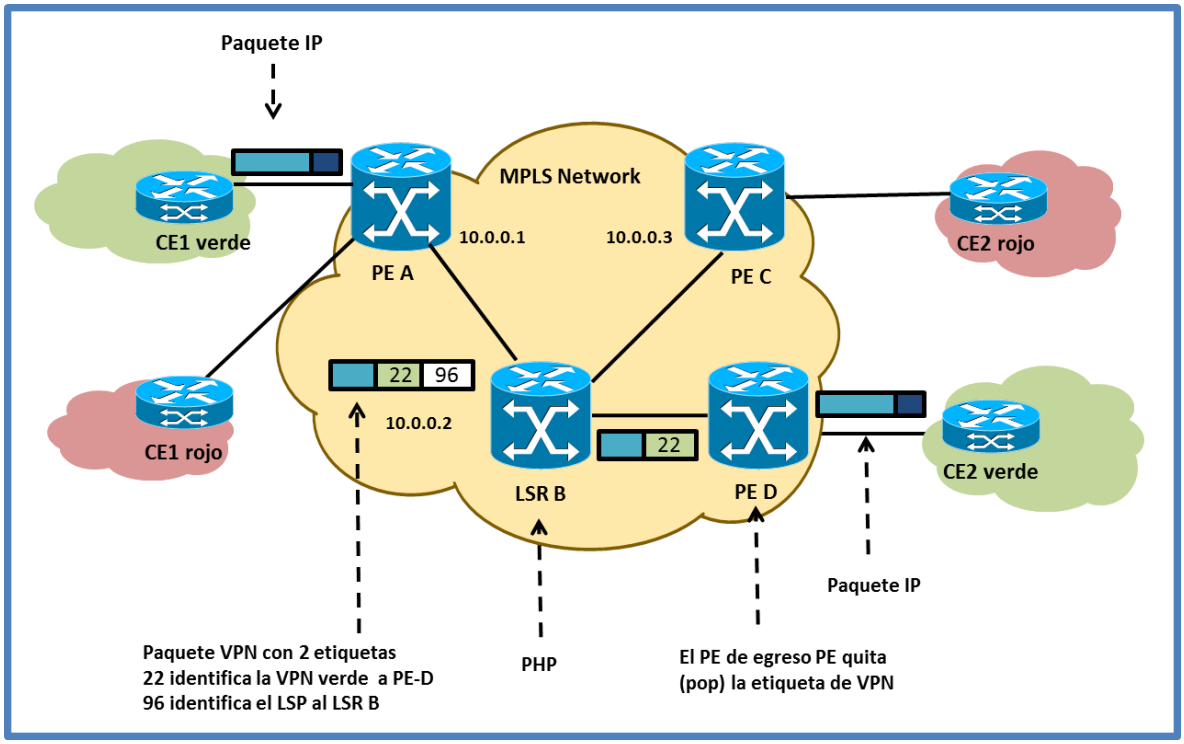

Figura 40 - Plano de datos VPN RFC 2547 (MPLS)

Antes que el tráfico sea enviado por un PE, se le debe agregar una etiqueta para que el paquete pueda llegar a la VRF correcta en el PE destino. En el ejemplo se mostrará un paquete que va desde el CE1 verde hacia CE2 verde.

- PE-A identifica el "next hop" (PE-D) para este paquete como un vecino BGP.
- PE-A inserta la etiqueta 22 que identifica la tabla de ruteo de la VPN hacia el PE-D. Esta etiqueta fue anunciada por el vecino (PE-D) durante el intercambio de prefijos BGP.
- El paquete debe viajar por la red MPLS, por lo que PE-A agrega otra etiqueta (96) que identifica el LSR "next-hop" en el camino hacia PE-D. Esta etiqueta fue anunciada por el LSR de "aguas abajo" (LSR-B) mediante LDP.
- Cada LSR en el camino intercambia etiquetas (swap) y reenvía el paquete normalmente (sin estar consciente que pertenece a una VPN) hacia PE-D. En el penúltimo "hop" se quita la etiqueta exterior (Penultimate Hop Popping – PHP). En el ejemplo hay un solo salto hacia el LSR de egreso, por lo que el LSR-B quita la etiqueta exterior.
- PE-D utiliza la etiqueta restante (22) para identificar la tabla de ruteo de la VPN verde que utilizará para este paquete y luego quita la etiqueta.
- PE-D realiza una búsqueda IP en la tabla de ruteo de la VPN para encontrar la interface de salida y luego envía el paquete IP a CE2 verde, quien lo rutea a su destino final.

Es importante comprender que los LSRs no tienen visibilidad del tráfico VPN, solamente realizan el "forwarding" de tráfico con etiquetas a lo largo de los LSP en concordancia con el protocolo de ruteo que corre en el "core" de la red. El IGP que se utiliza en la red MPLS, puede ser distinto al IGP utilizado en los vínculos CE-PE, ya que éstos no intercambian rutas directamente.

Los paquetes con etiquetas (MPLS) se encuentran únicamente en el "core" de la red. Los vínculos CE-PE utilizan IP. El último paso de la lista de actividades que ocurren en el tránsito de datos describen el "Penultimate Hop Popping" (PHP). Cuando el último LSR quita la etiqueta más externa, revela la etiqueta interior. PE-D usa una única búsqueda en la LFIB para encontrar la VRF a utilizar; realiza una segunda búsqueda para encontrar la interface par la cual se encuentra conectado el CE correspondiente.

Una VPN MPLS utiliza dos protocolos para señalización. La red MPLS utiliza LDP entre los LSRs para anunciar las etiquetas para los prefijos en sus tablas de ruteo. Los PEs utilizan MP-BGP para anunciar las rutas de las VPNs. No existe correlación entre ambos espacios de etiquetas.

5.4.1.2 Otras alternativas para el plano de datos

Las VPNs MPLS son un servicio exitoso, con cientos de redes operativas a lo largo del mundo. Sin embargo algunos "carriers" pueden sentirse atraídos por el modelo planteado por el RFC 2547, pero pueden tener resistencia por una implementación de MPLS o bien contar un con una infraestructura en su "backbone" IP tradicional. Como se mencionó anteriormente en el mismo RFC se plantean alternativas para estas situaciones y además existen trabajos en estas direcciones para resolver este escenario. Las dos alternativas más interesantes son las de transportar MPLS o bien sobre GRE (o mGRE) (Sección 5.4.1.2.1) o bien sobre L2TPv3 (Sección 5.4.1.2.2).

Estas alternativas están alineadas a la arquitectura propuesta en el RFC 2547 y utilizan MP-BGP para la redistribución de las rutas de los clientes y continúan usando etiquetas para la identificación de la tabla de ruteo de la VPN. Sin embargo, la red de "core" ya no utiliza MPLS y utiliza IP. La etiqueta externa es cambiada por algún otro método para marcar la ruta del PE origen al destino. Se mantienen el modelo de referencia PE-CE.

5.4.1.2.1 MPLS sobre mGRE

En esta arquitectura se utiliza túneles GRE dinámicos entre PEs para trasladar los datos del cliente. La distribución de rutas entre CE-PE y PE-PE funciona de la misma manera que en una solución MPLS-VPN. La diferencia importante es la manera que un PE envía el tráfico por la red de "core".

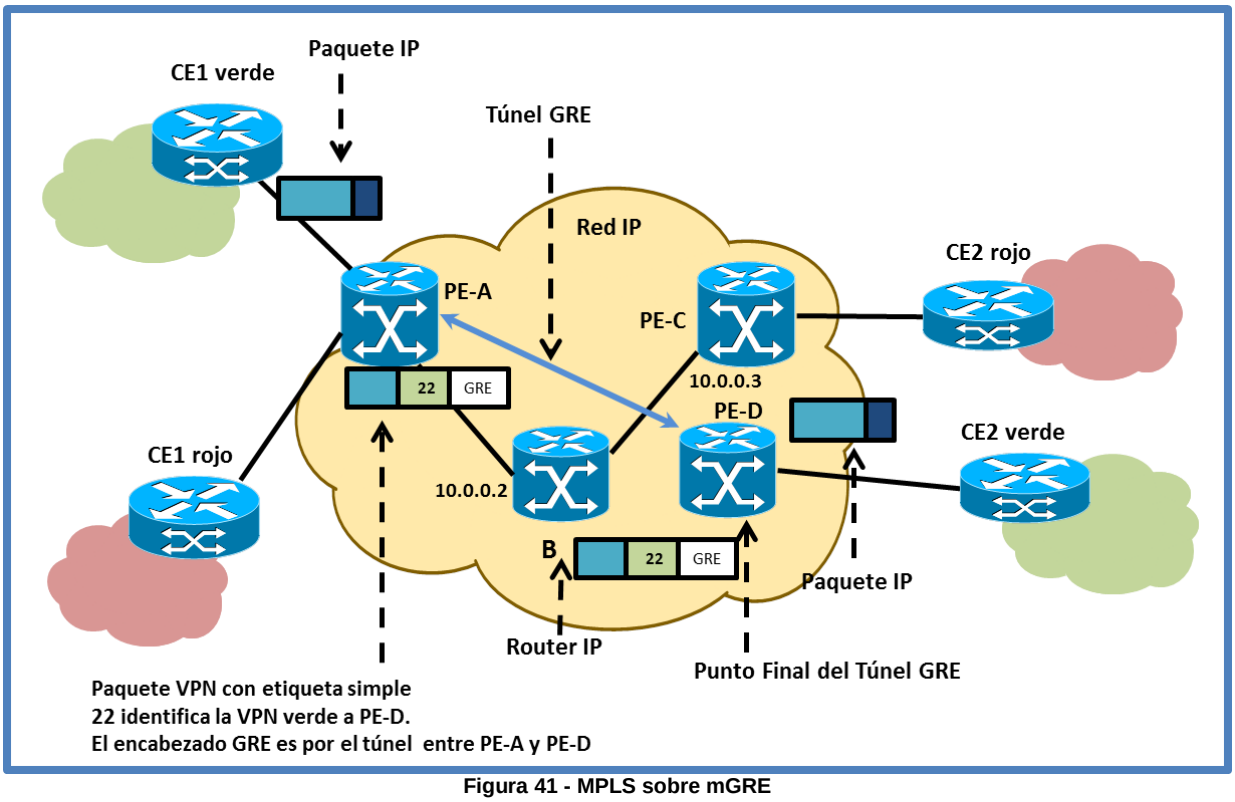

Figura 41 - MPLS sobre mGRE

En el ejemplo de la Figura 41, cuando PE-A necesita rutear un paquete desde CE1 a CE2, consulta la tabla de la VRF verde para encontrar el "next hop" BGP (PE-D) y la etiqueta asociada a la ruta correspondiente a la VPN anunciada por PE-D. Inserta la etiqueta delante del paquete y busca la dirección del "next-hop", la interface saliente y la información de encapsulamiento en la FIB. PE-A luego inserta un encabezado GRE delante del paquete etiquetado con dirección de origen y destino correspondiente a las direcciones públicas de PE-A y PE-D respectivamente. El atributo tipo del túnel GRE debe indicar un contenido de paquete MPLS (RFC 4023).

A nivel de paquete, se agrega la información que necesita GRE para transportar los datos sobre el túnel como lo muestra el esquema que se presenta en la Figura 42.

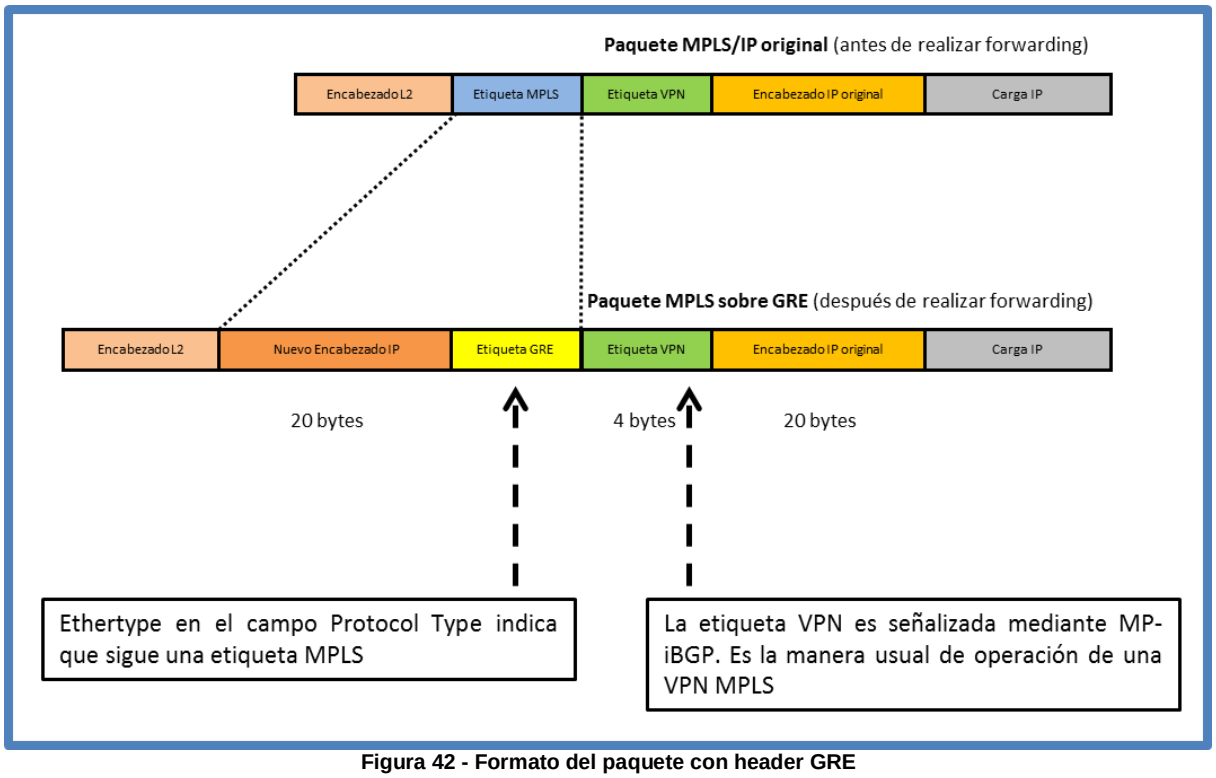

Figura 42 - Formato del paquete con header GRE

El uso del término dinámico se refiere que los túneles GRE no son utilizados como un camino posible por los protocolos de ruteo que ejecutan en el PE; solamente se utilizan como un método de encapsulamiento para atravesar un “core” basado en IP.

Un punto que se menciona en el RFC 4023 [56] es con respecto a la seguridad, Con respecto a este tema se discute sobre los distintos problemas posibles. Uno de ellos es el hecho que se pueda visualizar / alterar el contenido del paquete en su tránsito por el túnel GRE. Para mitigar este problema se puede utilizar IPsec. El otro problema potencial es el “spoofing” de paquetes dentro del túnel GRE ya que en este caso el PE recibe y envía paquetes IP, por lo tanto se debe utilizar algún mecanismo de filtrado de paquetes para mitigar este riesgo.

5.4.1.2.2 MPLS sobre L2TPv3

MPLS sobre L2TPv3 utiliza el mismo principio que la solución sobre GRE, excepto que utiliza el plano de datos de L2TPv3. BGP distribuye rutas de clientes y etiquetas correspondientes a rutas de las VPNs. L2TPv3 posee su propio plano de control y la información negociada durante el establecimiento de la sesión se puede encontrar en el encabezado del plano de datos; principalmente el “session ID“ y la “cookie ID”.

El “session ID” y la “cookie ID” son utilizados en esta arquitectura. El “session ID” se utiliza para multiplexar la sesión; diferentes sesiones tienen identificadores diferentes dentro del mismo túnel. En este entorno, se utiliza la etiqueta para este rol, por lo que el valor del “session ID” se utiliza para indicar al “endpoint” L2TPv3 que recibe que el paquete entrante corresponde a una VPN L3 y que se necesita procesamiento adicional por otro subsistema. EL “cookie ID” se sigue utilizando para implementar protección “antispoofing”. El valor del mismo puede ser generado tanto de manera estática como al azar y puede ser global (por PE) o local (por sesión). Tanto el “session ID” como el “cookie ID” son anunciados mediante MP-BGP, pero distintas implementaciones pueden usar otro protocolo.

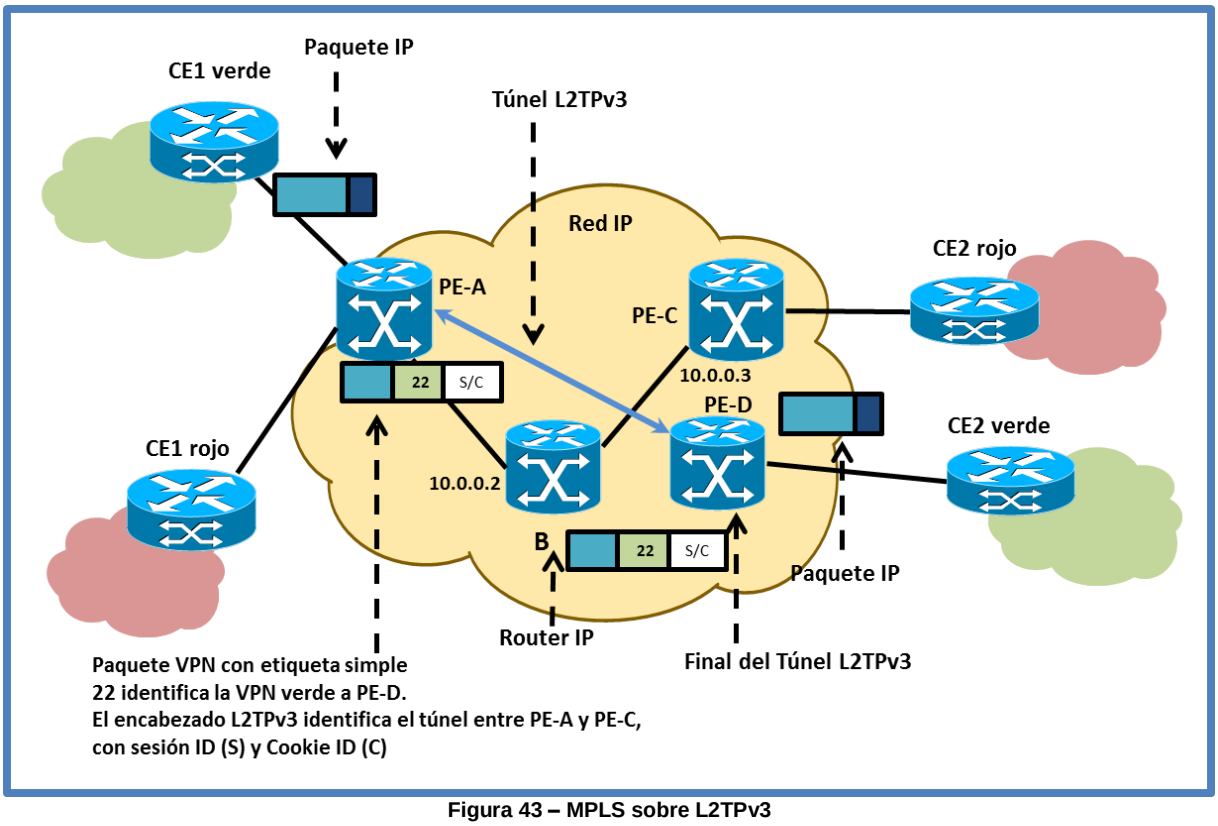

Figura 43 – MPLS sobre L2TPv3

En nuestra red de ejemplo de la Figura 43, cuando CE1 envía datos a CE2, PE-A realiza una búsqueda en la tabla de ruteo para encontrar el "next hop" de BGP así como la etiqueta que identifica a la VPN. A su vez encuentra el "session ID" y el "cookie ID" para el PE remoto. Luego, PE-A realiza una segunda búsqueda para encontrar la dirección IP de PE-D y encapsular el paquete dentro de un túnel L2TPv3. La red es una red estándar IPv4.

En el ingreso, PE-D quita el encabezado exterior de L2TPv3 y verifica la "session ID" y la "cookie ID". Si son válidos, la etiqueta del paquete se utiliza para identificar la VRF correcta y el procesamiento continúa como es usual. Si se compara con la solución de GRE, ésta es más robusta en el aspecto de seguridad ya que es posible descartar paquetes apócrifos (que no tienen una cookie ID correcta). En cuanto al cifrado de datos, es posible utilizar IPsec, tratando ambos extremos del túnel como "Security Associations" y debe utilizarse IPsec en modo transporte. Estas características de utilizar MPLS sobre L2TPv3 son abordadas en el "draft" del IETF draft-townsley-l3vpn-l2tpv3-01 (BGP/MPLS IP VPNs over Layer 2 Tunneling Protocol ver 3) y el RFC 4817 [46].

6. La red virtualizada de la empresa sobre la WAN

6.1 Introducción

En el presente capítulo se presentan las alternativas tecnológicas y mejores prácticas para mantener redes virtuales sobre las distintas opciones de conectividad que ofrecen los proveedores.

En la primer parte del capítulo se trata como se instancian las arquitecturas vistas en el capítulo anterior sobre las ofertas de los proveedores de servicio. Al final de la primera parte se muestra un cuadro resumen donde se presenta una guía para el diseñador de las distintas alternativas tratadas a modo de resumen.

En la segunda parte se presentan distintas alternativas para prestar y consumir servicios de red por parte de la empresa en el escenario de una red empresarial virtualizada, donde los recursos son compartidos entre varias redes virtuales.

La realización de este capítulo se basó principalmente en el capítulo 7 y 8 de [1] y otras fuentes.

6.2 Segmentación sobre la WAN

Debido a la variedad de tecnologías disponibles de los proveedores de servicios, la WAN es una porción interesante de la red empresarial. Dependiendo de los servicios ofrecidos por el proveedor, la empresa dispondrá de distintos niveles de control sobre la WAN. En muchos casos, la empresa requerirá el uso de un conjunto de redes lógicas para satisfacer sus necesidades de comunicación sobre la WAN. Esta conectividad se vuelve más sofisticada cuando se debe dar soporte a redes virtuales.

Los servicios WAN más comunes ofrecidos por los proveedores son:

- Servicios IP. VPNs L3 o enlaces a Internet.
- Circuitos capa2. Servicios tradicionales como ser ATM o Frame Relay o circuitos Ethernet (Metro Ethernet [50])

Es común que las empresas implementen sus propias redes sobre estos servicios. Alguno de los tipos de red que más comúnmente se implementan incluyen:

- GRE punto a punto (P2P GRE)
- GRE multipunto tipo “hub-and-spoke” (mGRE)
- DMVPN

Desde el punto de vista de la empresa, se puede proveer dos tipos de conectividad sobre la WAN:

- Interconexión de redes de área metropolitana (MAN) y redes de campus.
- Agregación de sucursales que requieran segmentación.

Las técnicas para extender las redes virtuales sobre la WAN incluyen las siguientes:

- MPLS sobre circuitos L2
- VPNs provistas por un proveedor de servicios
- MPLS sobre túneles GRE

- Múltiples VRFs interconectadas mediante mGRE/DMVPN
- VPNs del RFC 2547 sobre mGRE/DMVPN

Se debe tener en cuenta que en algunos escenarios, ´no se requiere segmentación en las sucursales. En este caso, la red de la sucursal se instala en una VRF propia o en la tabla de ruteo general, según los requerimientos de la empresa.

6.2.1 MPLS sobre circuitos L2

6.2.1.1 MPLS sobre circuitos L2 para interconectar MANs

El modelo más simple de utilizar para conectar MANs o redes de campus que han sido segmentadas utilizando VPNs de acuerdo al RFC 2547 es utilizar MPLS sobre circuitos L2 como se muestra en la Figura 44. El objetivo de este escenario es construir una gran red MPLS en la cual la WAN solamente brinda servicio de conexión L2 pero no participa en el ruteo IP de la empresa.
La solución incluye transformar los dispositivos de borde en equipos P (los llamaremos E-P ya que pertenecen a la red de la empresa) y por lo tanto convertir a la WAN en parte de la MAN MPLS de la empresa.

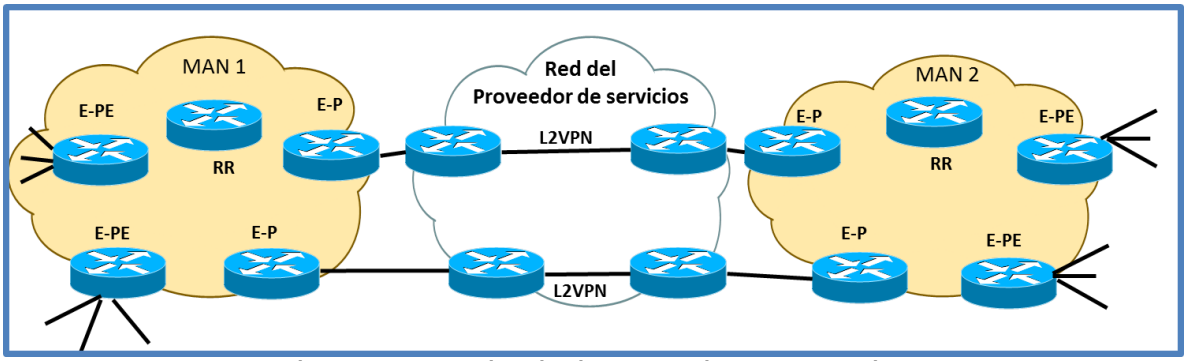

Figura 44 - MPLS sobre circuitos L2 para interconectar redes

Desde el punto de vista del plano de control, los siguientes protocolos van a ejecutar sobre los vínculos WAN.

- Un IGP (por ejemplo OSPF) para que los dispositivos E-P, E-PE, y los RR (“Route Reflectors”) puedan ser alcanzables entre sí.
- El protocolo LDP (“Label Distribution Protocol”).
- MP-iBGP para la distribución de etiquetas entre los dispositivos E-PE (preferentemente mediante un RR).

La solución es simple de configurar, lo que hay que hacer es expandir el “core” de la empresa para incluir la WAN. Desde la perspectiva de una VPN según el RFC 2547, se debe configurar lo siguiente en los routers E-P:

- El IGP se debe extender para que los routers E-P tengan vecindad sobre las interfaces WAN (configuración del protocolo de ruteo).
- Un protocolo de distribución de etiquetas (pe LDP) debe ser habilitado en cada E-P router sobre el borde de la WAN.
- Se debe habilitar MPLS en las interfaces hacia la WAN y hacia la MAN de los routers E-P de borde.

6.2.1.2 MPLS sobre circuitos L2 para interconectar Sucursales

El modelo detallado en la Figura 45 se basa en el hecho que la empresa dispone de vínculos L2 para conectar sucursales. Se debe habilitar MPLS sobre dichos vínculos para proveer segmentación sobre la VPN MPLS. Generalmente la conectividad L2 es del tipo "hub-and-spoke" o un "mesh" parcial, la cobertura MPLS hereda la arquitectura de conectividad subyacente. La comunicación "spoke-to-spoke" sucede a través del "HUB". El router de agregación (ubicado en el HUB) se convierte en un router E-P y los routers de las sucursales se convierten en routers E-PE con interfaces VRF hacia la sucursal y las interfaces con MPLS habilitado hacia la WAN.

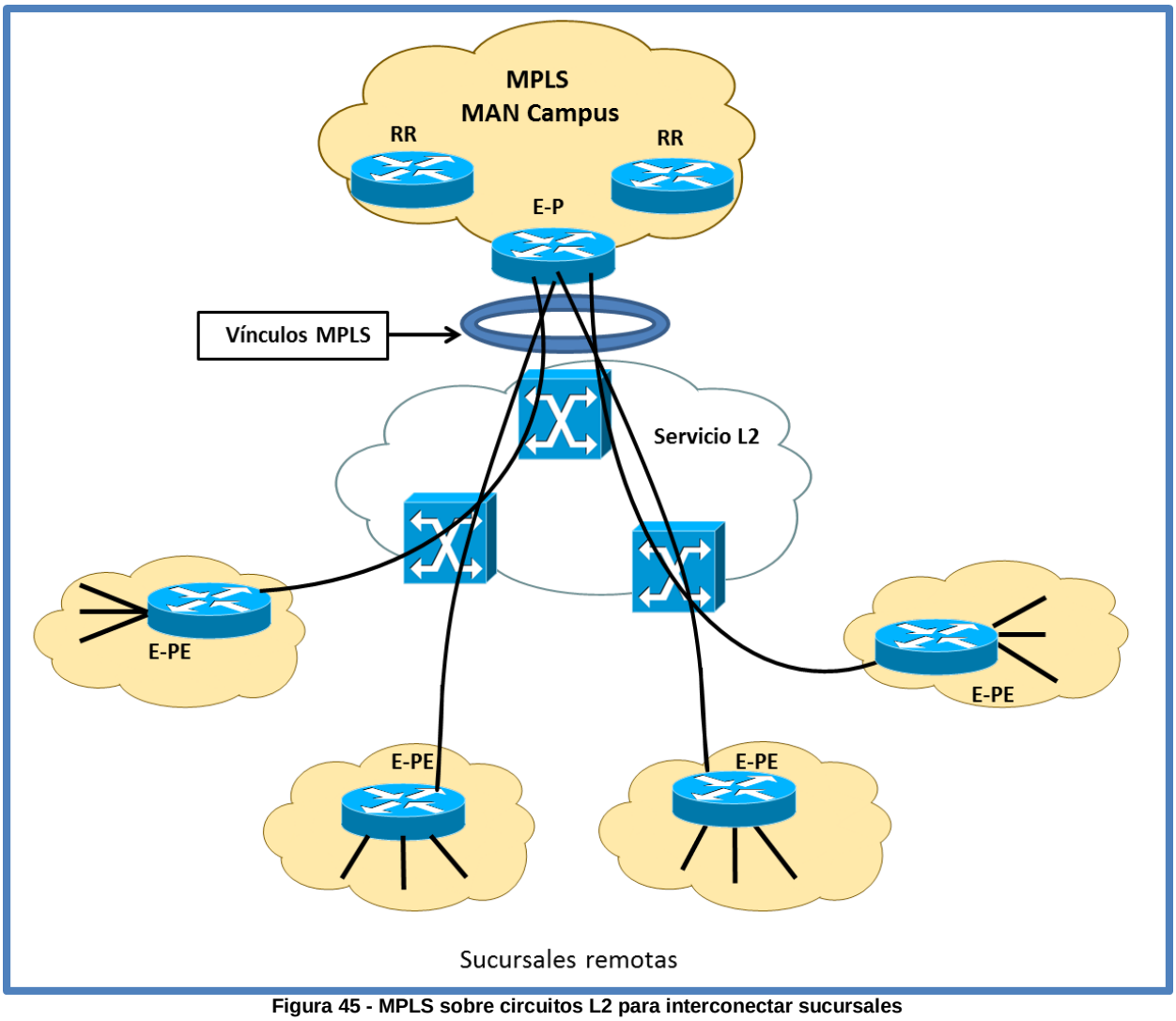

Figura 45 - MPLS sobre circuitos L2 para interconectar sucursales

Como los routers de las sucursales son del tipo E-PE, extienden el alcance de la red MAN. Como cualquier router PE, los routers de las sucursales mantienen una sesión LDP y una IGP con los routers P vecinos (en este caso el router de agregación en el HUB). Los routers PE también mantienen sesiones de MP-iBGP con los RRs (u otros routers PE), los cuales normalmente residen detrás del router de agregación.

La configuración del router en el HUB es idéntica a la vista en el caso anterior. Para los E-PE

routers de las sucursales, la configuración debe ser como sigue:

- El IGP debe ser configurado de tal manera que se establezca vecindad sobre la WAN entre los router E-P y los E-PE.
- Se debe configurar LDP en cada router E-PE.
- Se debe habilitar MPLS en la interface WAN de los E-PE.
- Se deben configurar las VRFs necesarias en los E-PE.
- Se debe configurar vecindad con los RRs.

6.2.1.3 Ventajas y desventajas de la solución

Las ventajas de desplegar MPLS sobre una infraestructura L2 incluye:

- La empresa tiene un control completo sobre la infraestructura IP.
- No hay dependencias en los servicios del proveedor para la extensión de la red virtual de la empresa o incluso la elección de la tecnología.
- Altamente escalable.

Las desventajas incluyen lo siguiente:

- El cifrado del tráfico es un desafío ya que se necesita un conjunto de túneles (generalmente GRE) para lograr dicho requerimiento.
- Los patrones de tráfico son tan buenos como la cobertura de circuitos L2. La conectividad “any-to-any” real requiere un “full-mesh” lo que generalmente suele ser caro.
- Los proveedores de servicio están dejando de brindar esta clase de servicio, volcándose al tipo de servicios de tipo VPNs IP o VPNs MPLS.
- Requiere un alto nivel de pericia operativa.

6.2.2 VPNs L3 provistas por un proveedor de servicios

Cuando una empresa necesita extender la segmentación a sucursales o bien extender su red MAN segmentada, una posible solución simple es contratar servicios de L3 VPN de un proveedor y mapear cada red virtual a una VPN L3. En el escenario que se muestra en la Figura 46, los routers en las sucursales se convierten en CEs multi-VRF y el router en la casa matriz puede convertirse en un multi-VRF CE o en un E-PE, dependiendo la estrategia utilizada en la casa matriz.

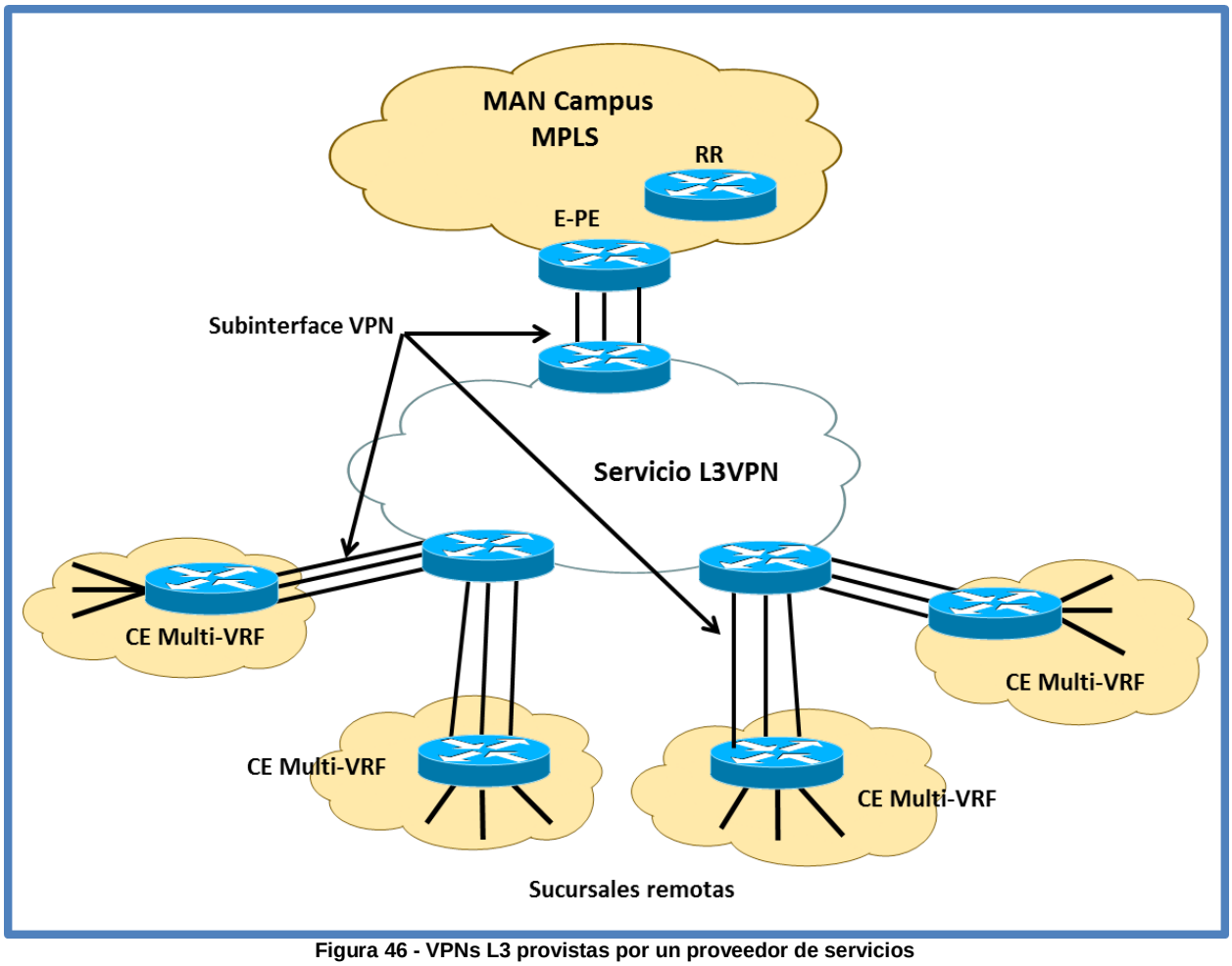

Figura 46 - VPNs L3 provistas por un proveedor de servicios

Para implantar esta solución, las redes virtuales de la empresa deben terminar en el borde de la WAN. La interconexión entre las redes virtuales de la empresa y las VPNs del proveedor se realiza conectando VRFs “back-to-back” en el E-PE. Estas conexiones “back-to-back” utilizan vínculos lógicos entre las VRFs de la empresa y las correspondientes VRFs del proveedor. La información de ruteo se intercambia sobre cada vínculo lógico asociado a las subinterfaces. Este intercambio puede ser realizado por una instancia de un IGP en cada subinterface.

Cada CE ejecuta un protocolo de ruteo IGP (p.e. OSPF) con el P-PE por cada VRF. Todas las recomendaciones de diseño y mejores prácticas que son aplicables para un servicio VPN, aplican para cada uno de las instancias de VPN.

6.2.2.1 Ventajas y desventajas de la solución

Debido al costo potencialmente alto de esta solución, es poco frecuente encontrarlo.

Las ventajas incluyen:

- Simple desde el punto de vista técnico.
- La administración se simplifica ya que el servicio de WAN se terceriza.
- No se requiere MPLS.

Las desventajas incluyen:

- Requiere un proceso de ruteo PE-CE por cada VPN en cada sitio.

- Tiene una alta dependencia con el proveedor de servicios.
- El costo puede ser muy alto basado en el número de VRFs y sitios.

Esta solución generalmente se limita a un número bajo de sucursales que requieren segmentación con un número bajo de VRFs. La limitación no se debe a un problema de escalabilidad, sino a un problema de costos ya que generalmente el proveedor cobra por el número de VPNs provistas y/o por el número de sitios conectados.

6.2.3 MPLS sobre GRE

Una posible alternativa para conectar la red con VPNs MPLS de la empresa sobre un servicio WAN IP es crear un conjunto de túneles GRE sobre la red WAN y habilitar MPLS en las interfaces GRE como se indica en la Figura 47. Esta configuración tiene los siguientes beneficios:

- Evita los nodos P y PE del proveedor sobre los cuales no tenemos control.
- Provee una topología lógica sobre la cual se puede habilitar MPLS y por lo tanto pertenecer a la red de la empresa.

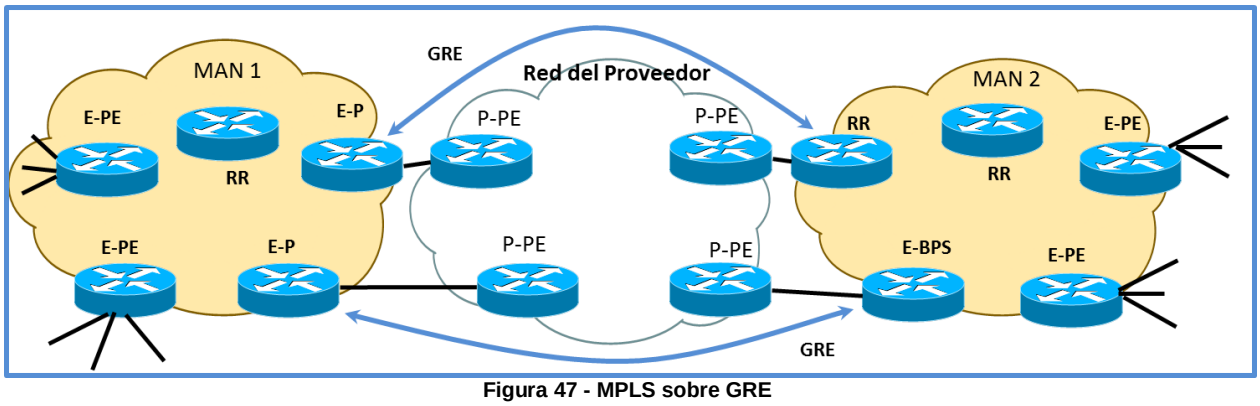

Figura 47 - MPLS sobre GRE

SI se desea un “full-mesh”, se debe crear un túnel GRE P2P entre cada par de routers de borde. Desde una perspectiva del plano de control, se espera que los siguientes protocolos ejecuten sobre los túneles GRE:

- Un IGP (p.e. OSPF) para asegurar que los dispositivos E-PE, E-P y RRs sean alcanzables entre sí.
- LDP para permitir la formación de LSPs sobre el cual el tráfico es enviado.
- MP-iBGP para la distribución de las rutas y etiquetas de la VPN entre los dispositivos E-PE.

Una vez que los túneles GRE se establecieron, la configuración de esta solución es idéntica a la usada para desplegar MPLS sobre circuitos L2. La única diferencia es que en vez de habilitar MPLS sobre las interfaces WAN, se lo habilita sobre las interfaces GRE.

Una vez que la distribución de rutas/etiquetas se completó en el plano de control, el dispositivo de borde de la empresa se comporta como un equipo P (LSR/P) donde se ven a las interfaces GRE como interfaces de acceso ordinarias.

6.2.3.1 Ventajas y desventajas

El despliegue de MPLS sobre un “mesh” de túneles GRE permite a la empresa extender su red MPLS sobre casi cualquier tipo de red IP.

Las ventajas de esta arquitectura incluyen lo siguiente:

- La solución es independiente del proveedor.
- Lo túneles GRE pueden ser cifrados con IPsec.
- Los routers de borde pueden tener el rol de P o PE, potencialmente permitiendo que una red MPLS sea desplegada a través de la frontera de una red MAN/red de campus/WAN.

Las desventajas incluyen lo siguiente:

- Mantener un “mesh” de túneles GRE puede ser trabajoso.
- La conectividad real “any-to-any” requiere un “full-mesh” de túneles.
- Se deben hacer consideraciones sobre el tamaño de la “Maximum Transmission Unit” (MTU) en el core de la WAN. Debido a que el “core” de la WAN no es controlado por la empresa, el tamaño del MTU debe ser impuesto en el borde de la red. [59]

La extensión de las MPLS VPNs sobre túneles GRE es útil en escenarios que requieren la agregación de un número limitado de sitios en una topología del tipo “hub-and-spoke”. Las topologías del tipo “any-to-any” son resueltas de mejor manera por mecanismos dinámicos.

6.2.4 Interconexión de VRFs mediante un overlay de túneles GRE o DMVPN

Este modelo mejora el “mesh” de túneles GRE para crear conexiones dedicadas “back-to-back” entre las VRFs. De esta manera, un grupo de VRFs en distintos sitios es interconectado por un “overlay” de túneles lógicos para crear una red virtual (VN). Cada red virtual dispone de un “overlay” dedicado y VRFs dedicadas en cada sitio, conformando redes lógicas separadas para cada grupo.

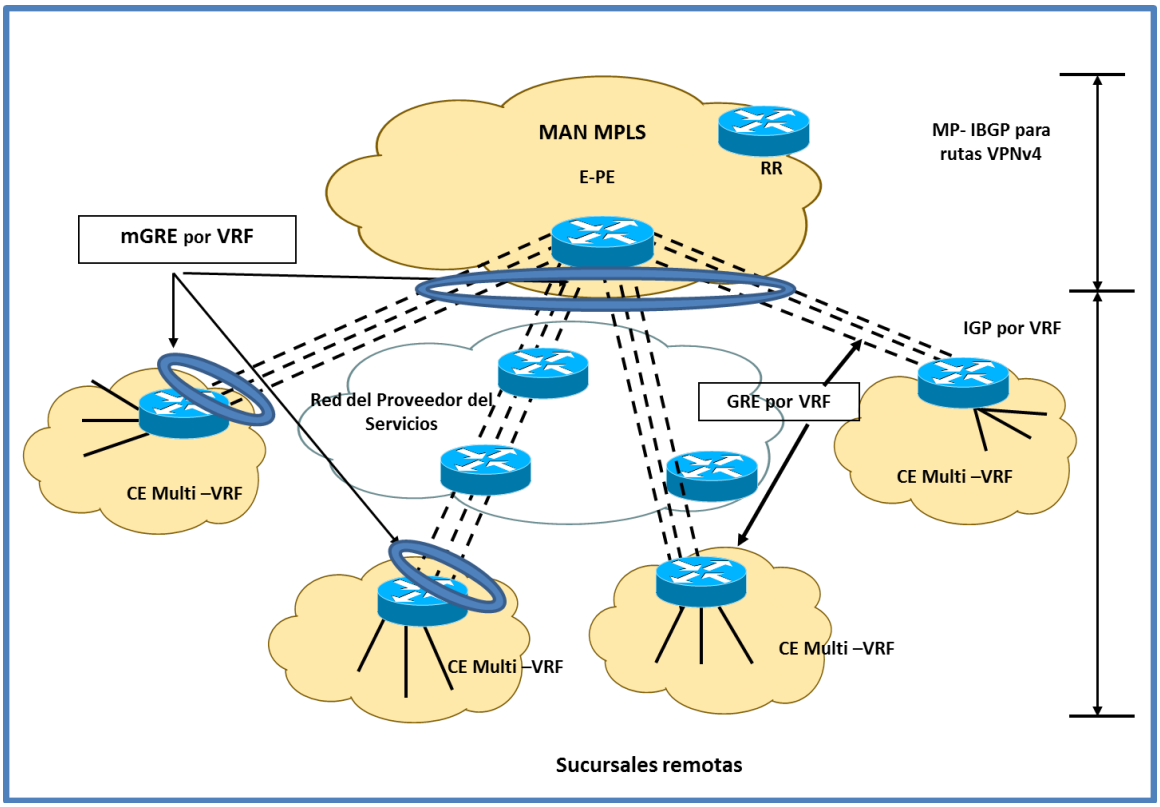

Figura 48 - VRFs múltiples interconectadas con mGRE/DMVPN

Este modelo puede ser utilizado sobre un servicio L2 o L3 de un proveedor. Si se hace esto, la empresa necesita adquirir solamente una única VPN o conjunto de circuitos L2. La empresa entonces, usa una combinación de multi-VRF y túneles GRE para realizar su propia red virtual sobre el servicio del proveedor.

En esta topología (descrita en la Figura 48), el router en la casa matriz dispone de un túnel mGRE por VRF. Las sucursales disponen de un túnel GRE o mGRE para cada VRF. Si no hay comunicación del tipo “spoke-to-spoke”, se utiliza GRE. En caso contrario, se utiliza mGRE.

El dispositivo en el HUB, aparte de realizar la agregación de las sucursales, también da soporte a las distintas VRFs. En el caso que este dispositivo implemente el RFC 2547, es visto como un PE. En este caso, se debe ejecutar una instancia de IGP por cada VPN entre el “HUB” y cada uno de los “spokes”. Las direcciones IPv4 que se aprenden de los “spokes”, son convertidas en direcciones VPNv4 antes de ser publicadas en la nube 2547 utilizando MP-iBGP. Como es usual en este tipo de implementaciones, el IGP cumple las tareas de:

- Intercambio de información de ruteo.
- Detección remota de fallo sobre túneles GRE.

Este tipo de despliegue es adecuado para la agregación de sucursales en una empresa con un número bajo de sucursales y un número bajo de grupos de usuarios o segmentos. La escalabilidad de esta solución, no solo viene dada por la capacidad de los dispositivos en el HUB, sino que también por la capacidad de administración de grupos y sitios.

Existen consideraciones similares para una solución basada en DMVPN ya que se requiere una DMVPN para interconectar cada grupo de VRFs y crear la VPN, lo cual involucra lo siguiente:

- Se deben crear túneles por cada VRF en cada sitio participante.
- Se requieren un servidor NHRP en el sitio HUB.
- Los routers en los “spokes” deben tener la posibilidad de ser cliente NHRP.
- Se deben considerar asociaciones de seguridad (IPsec sa) dinámicas para cifrado entre “spokes”

6.2.4.1 Ventajas y desventajas

Loa ventajas de usar esta arquitectura incluyen:

- El cifrado se encuentra nativo en la solución.
- Esta solución utiliza tecnologías que son familiares para el personal de operaciones de la empresa.
- Los requerimientos de hardware de esta solución, hace que se pueda elegir una amplia gama de plataformas para su implementación

La principal desventaja con esta solución es su limitada escalabilidad debido a la gran cantidad de túneles GRE o DMVPN a medida que la cantidad de sucursales crece. Este tipo de solución, generalmente aplica para empresas que ya están ejecutando DMVPN y necesitan agregar capacidad de llevar VRFs hacia los sitios.

6.2.5 VPN RFC 2547 sobre DMVPN

Es posible desplegar una única DMVPN y sobre esta multiplexar las diferentes redes virtuales, las cuales son multiplexadas utilizando etiquetas de VPN de la misma manera que se implementa una VPN según el RFRC 2547 con túneles GRE (ver Figura 49). Este modelo es recomendado principalmente para patrones de comunicaciones entre el HUB y los "spokes". Este modelo provee la flexibilidad de la creación de túneles dinámicos junto con la mejora en escalabilidad debido a que no existe un mapeo uno a uno entre los túneles y las VRFs, sino que existe un único túnel multipunto que se comparte para transportar varias redes virtuales.

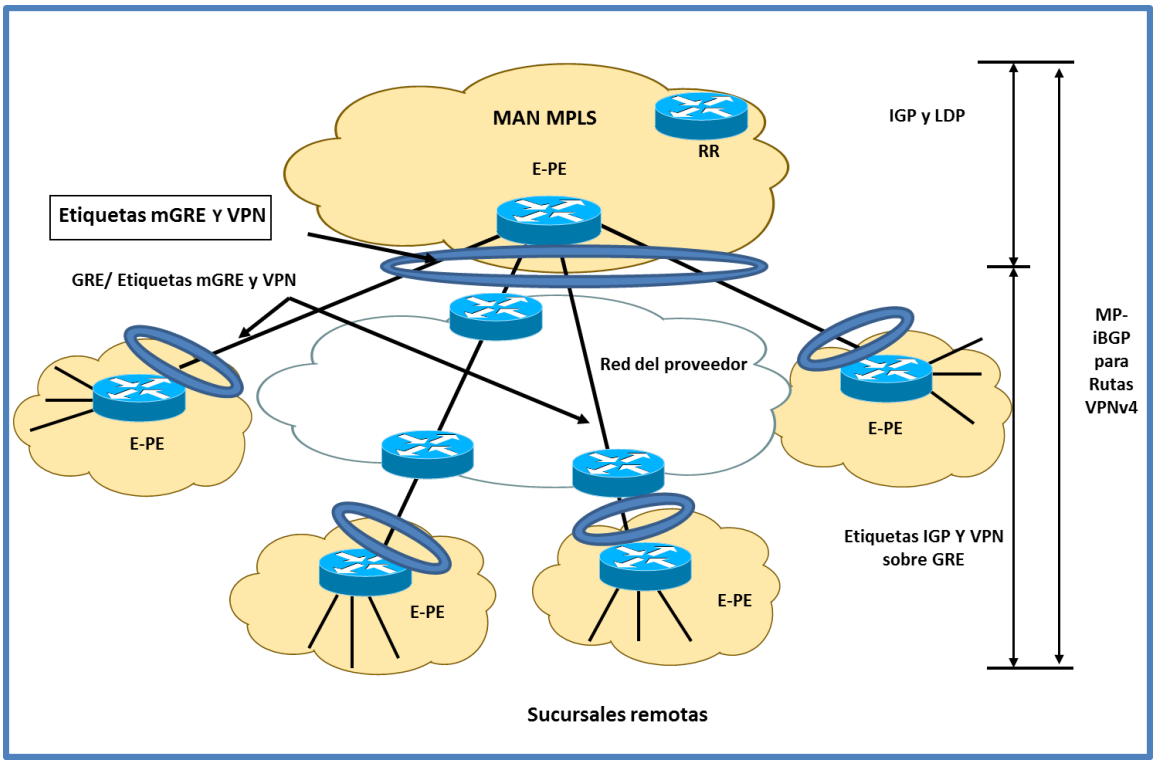

Figura 49 – RFC 2547 sobre mGRE/DMVPN

El plano de control para una VPN según el RFC 2547 sobre una DMVPN debe incluir lo siguiente:

- Las direcciones de los túneles para las sucursales como para la casa matriz deben ser publicadas por dentro de la red del proveedor ya sea de manera dinámica o estática.
- Un túnel GRE estático debe ser configurado entre el E-PE de la sucursal y cada E-PE de las sucursales.
- Un IGP ejecutando en el espacio global de la empresa y a través del túnel GRE es necesario para proveer conectividad entre los PEs y entre los PEs y los RRs.
- Las sesiones MP-iBGP son requeridas entre los RRs y los PEs, donde la dirección IP de origen publicada por BGP es la dirección de la interface del túnel. Esto fuerza que la búsqueda del BGP "next-hop" para la ruta de la VPN esté asociado con la interface tipo túnel.
- Se debe configurar NHRP entre el HUB y los "spokes".

6.2.5.1 Ventajas y desventajas

Una de las características más atractivas de implantar una VPN RFC 2547 sobre DMVPN es la posibilidad de utilizar la topología DMVPN y habilitar el etiquetado de la VPN sobre ésta. Las redes virtuales que viajan sobre la infraestructura DMVPN heredan todos los beneficios de esta

topología como así sus desventajas.

Las ventajas de esta arquitectura incluyen:

- Las empresas pueden mejorar su infraestructura DMVPN para agregar sucursales
- Los túneles dinámicos del tipo “spoke-to-spoke” proveen comunicación directa entre las sucursales
- El cifrado nativo provee privacidad para el tráfico en tránsito sobre la WAN

Las desventajas incluyen lo siguiente:

- Este método es el indicado para conectividad tipo “hub-and-spoke”. Para patrones “any-to-any” se deben usar otras técnicas.
- Los límites de escalabilidad son más bajos en otras soluciones.
- Es una solución propietaria (Cisco), aunque existen proyectos de código abierto que interactúan con la misma [60].

6.2.6 Resumen

A continuación se presenta un cuadro resumen de las características de las distintas alternativas analizadas:

	Nro de VRFs objetivo	Eficiente spoke-to-spoke	Plano de control	Plano de datos	Esfuerzo requerido de implementación	Cifrado
MPLS sobre circuitos L2	Alto	Depende del circuito L2 (generalmente Hub-and-spoke)	IGP, LDP, MP-iBGP	MPLS	Alto	Requiere "mesh" de túneles
VPNs L3 provistas por un proveedor	Bajo (Debido al costo)	SI	IGP, BGP	IP	Bajo	Depende del proveedor. Generalmente requiere "mesh" de túneles.
MPLS sobre GRE	Alto	Requiere "full mesh"	IGP, LDP, MP-iBGP	MPLS sobre GRE	Alto si crece el número de túneles	IPsec sobre GRE
Interconexión de VRFs mediante overlay GRE o DMVPN	Bajo	Si	NHRP, IGP, MP-iBGP	IP sobre mGRE	Mediano	IPsec sobre GRE
VPN RFC 2547 sobre DMVPN	Alto	SI	MP-iBGP, IGP, LDP	VPN encapsulado en GRE	Bajo	IPsec sobre GRE

Como vimos anteriormente, se disponen de varias alternativas cuando se extiende la red virtualizada de la empresa. La elección de cual alternativa utilizar depende de los servicios

disponible por el proveedor de la red y la tecnología que a la que la empresa puede dar soporte (ya sea por equipamiento o calificación de su personal o costo). En general la extensión de la virtualización sobre la WAN requiere una de las siguientes tres condiciones:

- Control de todos los dispositivos IP en la WAN.
- Creación de un camino lógico capaz de saltearse los dispositivos IP sobre los cuales no se tiene control.
- Un servicio de un proveedor capaz de transportar la información de la red virtual para la empresa.

6.3 Servicios WAN compartidos

Una ventaja de la empresa virtualizada es la habilidad de crear patrones de comunicación flexibles entre las distintas redes virtuales. Esta capacidad permite el despliegue de servicios que están accesibles de manera central y son comunes a varias redes virtuales. La centralización al acceso de estos servicios provee un punto común donde se pueden aplicar políticas y control para las redes virtuales.

Los servicios compartidos de la empresa pueden agruparse como protegidos o no protegidos según la manera en que son accedidos.

6.3.1 Servicios no protegidos

Un servicio que puede ser accedido de manera abierta, sin control de tráfico o verificación de seguridad es considerado como un servicio no protegido. Es posible consumir un servicio no protegido desde una o más red virtual sin tener ningún control de seguridad entre el servidor y el cliente.

El uso de VRFs y MP-iBGP permite suficiente flexibilidad al tráfico para ser intercambiado entre VPNs. Esto se logra exportando e importando rutas entre las VRFs para proveer conectividad IP entre las distintas redes virtuales. Debido a la naturaleza “any-to-any” del tráfico en este tipo de redes, existe poca posibilidad de controlar el intercambio de tráfico luego que se intercambiaron las rutas. Debido q que no se realizarán chequeos de seguridad, se debe desplegar este tipo de conectividad con mucho cuidado ya que potencialmente se pueden crear “puertas traseras” entre las redes virtuales y de esta manera romper el concepto de zona de seguridad. En general la implementación de estos servicios se realiza creando lo que se llama una red virtual “Extranet”. Una red virtual “Extranet” se puede comunicar con muchas redes virtuales, pero no es una zona de tránsito entre las redes que se comunican con ella. En términos de ruteo, la Extranet tiene rutas hacia las redes virtuales cliente, las redes virtuales cliente tienen rutas hacia la Extranet, sin embargo las rutas de unos clientes hacia otros no son publicadas a través de la Extranet.

Es común que se requiera compartir servicios en una VPN con muchas otras. El uso de las cláusulas **“import”** y **“export”** de BGP a lo largo de distintas VPNs permite a varias VPNs comunicarse con una Extranet compartida. En este escenario, la VPN tiene rutas para la extranet y viceversa, pero la Extranet no actúa como una zona de tránsito entre VPNs. A pesar que las VPNs son mantenidas separadas unas de las otras, se pueden conectar a recursos compartidos.

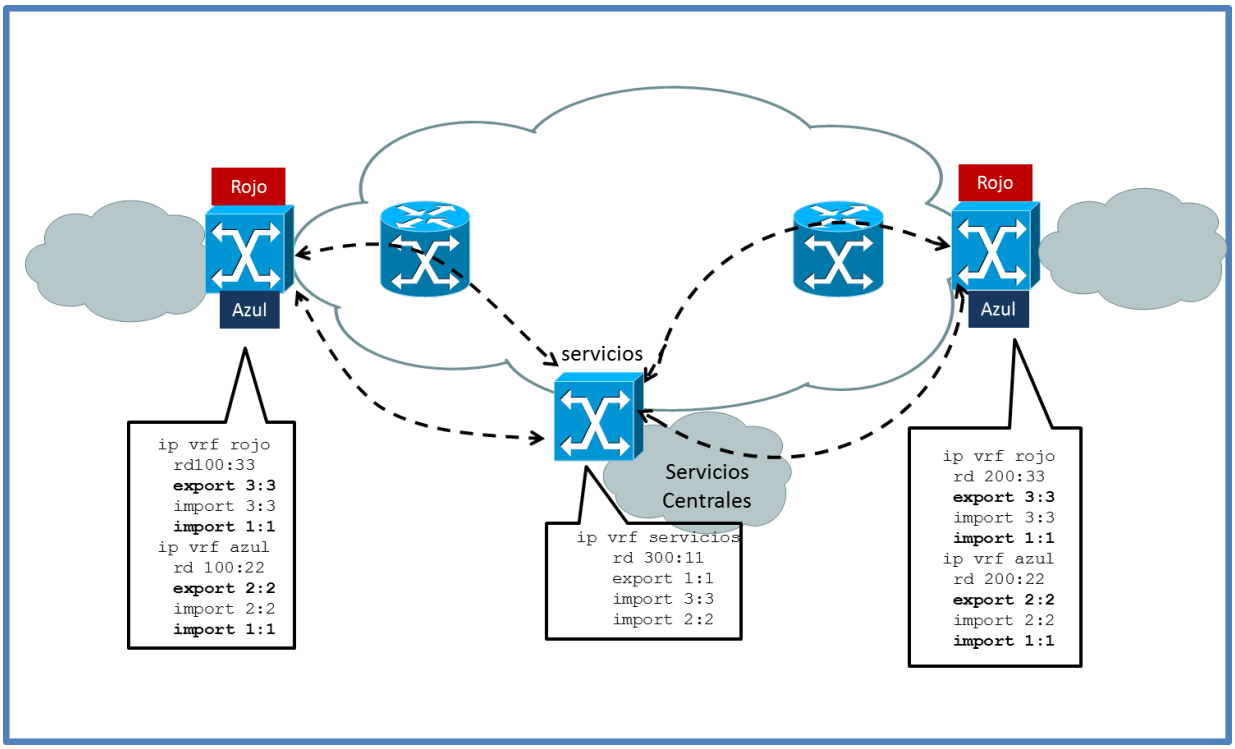

Figura 50 - Extranet VPN

En el ejemplo de la Figura 50, las rutas para la VRF de los servicios compartidos son exportadas con un RT 1:1. Estas rutas son importadas en las VRFs rojo y azul, permitiendo la comunicación de las VRFs hacia los servicios compartidos. La comunicación en el otro sentido se permite importando las rutas de las VRFs rojo y azul dentro de la VRF servicios. Esto se consigue importando los RTs 2:2 y 3:3, los cuales corresponden a las VRFs azul y rojo respectivamente. Es importante aclarar que las rutas de azul y rojo importadas en las VRF servicios mediante MP-iBGP no son vueltas a exportar. En otras palabras, las actualizaciones importadas con RTs 2:2 y 3:3 no son exportadas con RT 1:1 por la VRF servicios, por lo que esta configuración no provee conectividad entre las VRFs rojo y azul.

La posibilidad de crear una “Extranet” mediante el uso de “imports” y “exports” es versátil y tiene muchas aplicaciones dentro de las empresas, especialmente en Datacenters compartidos. El ejemplo visto anteriormente es un caso de aplicación de esta técnica y se expuso para mostrar los fundamentos de la técnica.

6.3.2 Servicios protegidos

Para permitir comunicaciones seguras a lo largo y entre redes virtuales, se debe crear puntos e ingreso y egreso desde y hacia cada red virtual. Una manera de realizar esto de manera fácil es configurar el ruteo dentro de cada red virtual para enviar el tráfico con destino fuera de la red virtual hacia un “Gateway” específico. Cuando el tráfico llega a este “Gateway”, se le pueden aplicar las políticas de seguridad utilizando dispositivos de seguridad. Esto es equivalente a tratar cada red virtual como si fuera una red física independiente.

Cuando se conectan varias redes a un recurso común, cada red debe ser terminada por un dispositivo de seguridad para controlar el acceso a la red. Para este fin generalmente se utiliza un “firewall”. Uno de los recursos compartidos más comunes es la conectividad con Internet, sin embargo existen recursos compartidos de la empresa que son consumidos desde varias redes virtuales. Estos servicios se encuentran desplegados en el perímetro de la red. Esto permite la

creación de políticas de seguridad específicas para cada red virtual. Para acceder a los servicios compartidos, los firewalls se encuentran conectados a un router de "fusión", el cual provee a las redes virtuales con conectividad a los servicios comunes, Internet o conectividad inter-redes virtuales si así se desea.

La presencia de este router de fusión, debe alertarnos de dos posibles preocupaciones:

- El intercambio potencial de tráfico entre redes virtuales.
- El riesgo que rutas de una red virtual sean anunciadas a otra red virtual.

La presencia de firewalls dedicados en el perímetro de cada red virtual previene el intercambio de tráfico entre las redes virtuales mediante el router de fusión ya que se solamente se permiten el retorno mediante el perímetro de la red virtual de conexiones previamente establecidas en dirección saliente.

El acceso a un recurso externo puede incluir el acceso a recursos de otras redes virtuales; por ejemplo el acceso a un recurso en la red virtual C se considera como un recurso externo para la red virtual A y por lo tanto debe ser accedido a través del perímetro de la red virtual.

A medida que la cantidad de redes virtuales crece, terminar cada red virtual en un dispositivo dedicado puede ser caro y difícil de administrar. Cisco provee "firewalls" que pueden ser virtualizados [61] y por lo tanto ofrecer un contexto separado y estanco para cada red virtual en el mismo dispositivo físico, como se puede observar en la Figura 51.

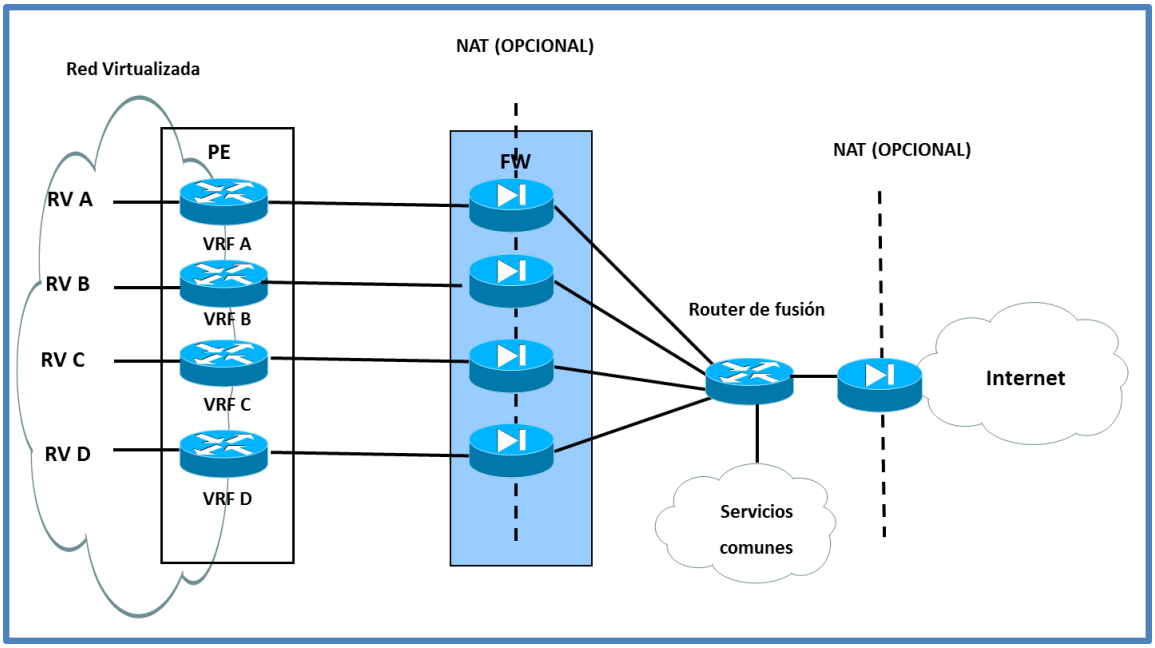

Figura 51 - Perímetro de redes virtuales con Firewall virtualizado

Tanto los firewalls físicos como los virtuales pueden operar en dos modos [61]:

- Modo transparente: El "firewall" se comporta como un bridge L2. Las interfaces "inside" y "outside" se encuentran sobre la misma subred (pero sobre distintas VLANs). El tráfico que pasa por el "firewall" transparente es objeto de inspección para la cual se puede utilizar información de capas L2, L3 y L4.
- Modo ruteado: El "firewall" se comporta como un router y soporta NAT.

Cada modo de funcionamiento de los firewalls requiere un estudio detallado del diseño y configuración cuando se realiza el ruteo entre las redes virtuales y el router de fusión.

6.3.2.1 Firewalls en modo ruteado

Los firewalls pueden ser utilizados en modo ruteado, pero sus capacidades de ruteo son limitadas cuando se lo compara con un router. Cuando un "firewall" es virtualizado, sus capacidades de ruteo se ven aún más disminuidas al punto que generalmente no permiten el uso de protocolos de ruteo dinámicos.

En el caso de servicios con un único sitio de intercambio, se dispone de un único punto de ingreso y egreso para servicios comunes protegidos. Para ejemplificar el caso se utilizará el servicio de Internet, pero las consideraciones que se verán aplican para cualquier otro tipo de servicio.

El ruteo entre el router de fusión y las diferentes redes virtuales debe ser configurado con cuidado ya que debido a su lugar, en la topología, el router de fusión tiene el potencial para mezclar las rutas de las distintas redes virtuales. Ya que el "firewall" se configura en modo ruteado con múltiples contextos, hace que solamente soporte ruteo estático y este hecho mitiga que sea posible la mezcla de rutas hacia las redes virtuales. La conectividad entre las redes virtuales se logra mediante configuración en el router de fusión. Dado que los firewalls se encuentran configurados para permitir el tráfico entrante únicamente de sesiones establecidas, cada red virtual puede alcanzar el router de fusión y éste puede retornar el tráfico a la red virtual correspondiente. Para permitir tráfico entre las redes virtuales a través del router de fusión, se deben configurar reglas para tal fin en el contexto correspondiente a las redes virtuales en cuestión.

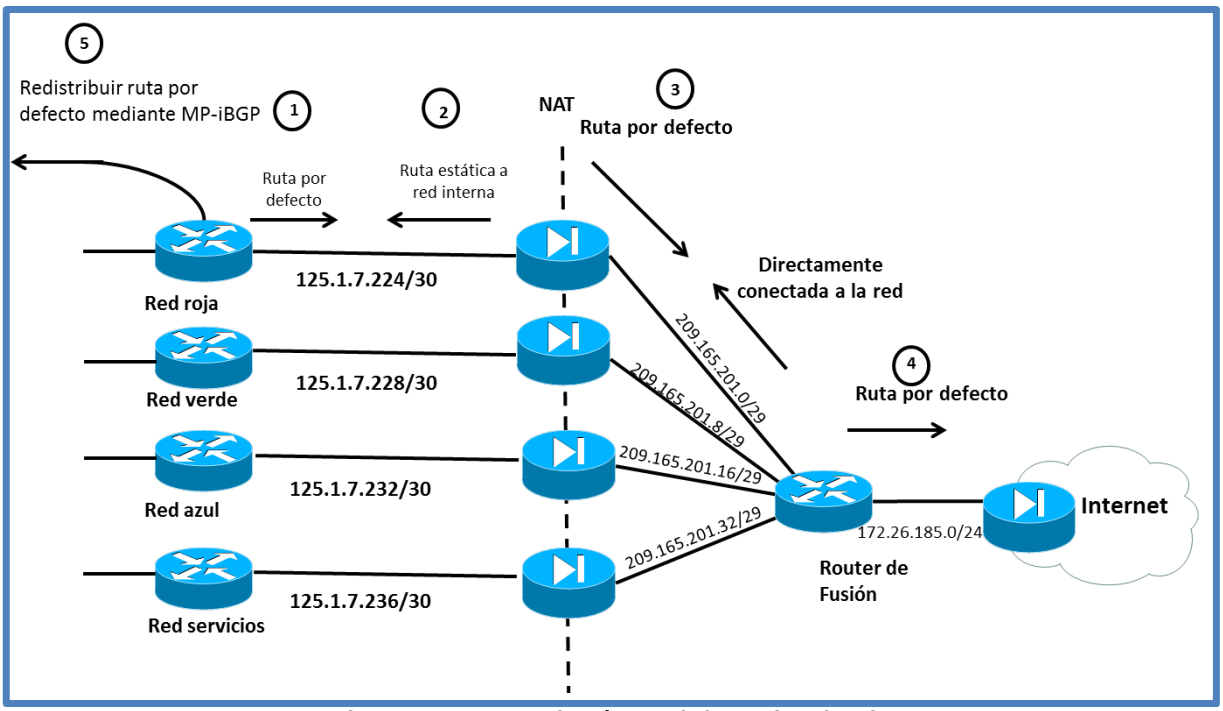

Figura 52 - Ruteo en el perímetro de las Redes Virtuales

Para configurar el ruteo para la red virtual Red roja en el contexto de la Figura 52, se deben seguir los siguientes pasos:

1. Crear la ruta por defecto para la VRF correspondiente.
2. Crear una ruta estática en la boca “inside” del “firewall” para alcanzar la red interna (Red-roja).
3. Crear una ruta estática por defecto para la boca “outside” del “firewall” para enviar tráfico al router de fusión.
4. Crear una ruta estática por defecto para el router de fusión para conectarse con el proveedor del servicio de Internet.
5. Inyectar la ruta por defecto creada en el paso (1) mediante MP-iBGP dentro de la red virtual.

En el caso de servicios con múltiples sitios de intercambio (Internet es un ejemplo), los objetivos que generalmente se persiguen son la capacidad de recuperación ante fallos y el balanceo de carga. La solución que se usa generalmente, utiliza dos sitios de servicios comunes e inyecta rutas por defecto en cada red virtual. La ruta preferida se elige en base a la proximidad con el sitio de servicios comunes. Esta proximidad se elige en base a la métrica del IGP que se ejecuta en la red.

En el caso particular de acceso a Internet, algunos sitios usarán un enlace y otros el otro enlace, basados en la proximidad al lugar de la red donde se encuentran instalados los servicios de Internet. En el caso de un fallo en un sitio de servicios donde está ubicado un enlace a Internet, la reconvergencia del propio IGP provoca que los usuarios que utilizaban el vínculo con problemas tengan conectividad por el otro enlace. En el caso de una falla en el ISP, se necesita algún mecanismo dinámico (p.e. Cisco IP SLA) para disparar la reconvergencia de la red y el reenvío del tráfico al enlace que funciona.

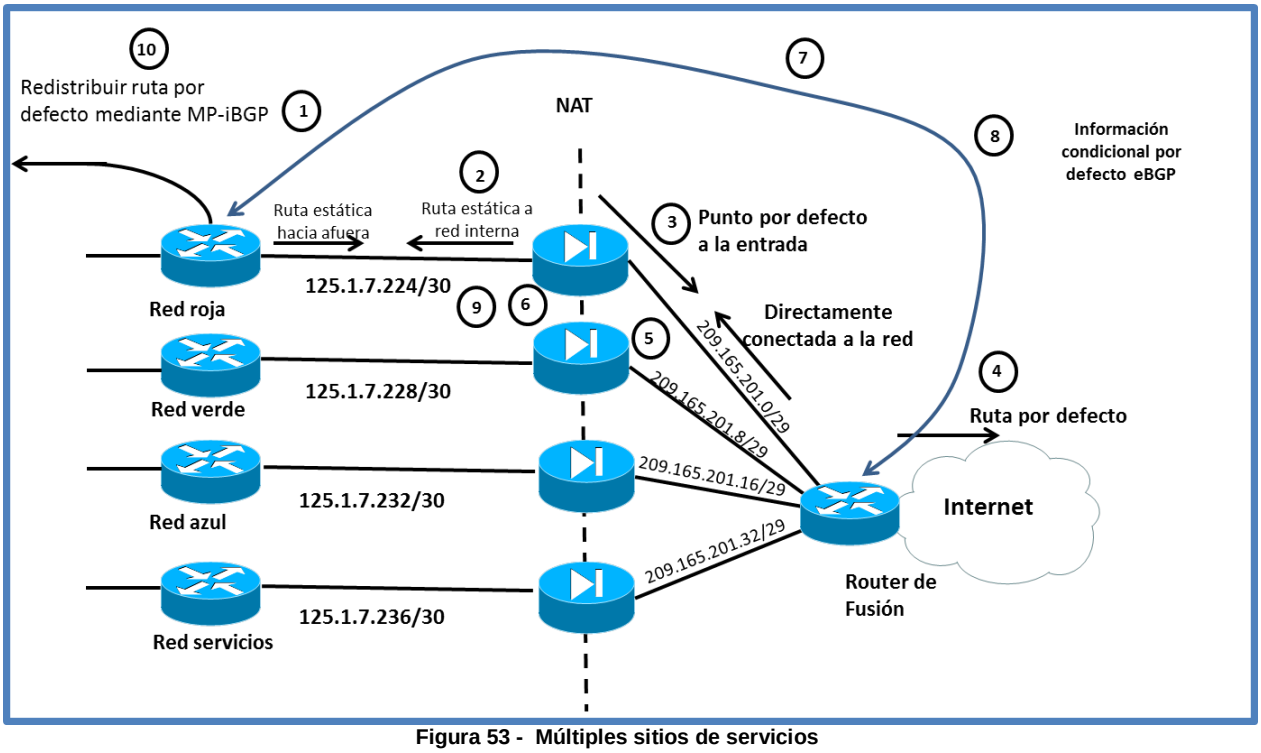

Figura 53 - Múltiples sitios de servicios

La configuración de este escenario (ilustrado en la Figura 53) involucra mayor cantidad de pasos ya que hay que establecer sesiones de BGP entre el router de fusión y el router PE de

cada red virtual. A su vez hay que permitir que el tráfico del protocolo de ruteo dinámico atraviese el "firewall".

1. En la VRF interna, crear una ruta estática al "firewall" exterior.
2. En la interface "inside" del "firewall", crear una ruta estática hacia la VPN interna
3. En la interface "outside" del "firewall", crear una ruta estática por defecto hacia el "Gateway" de Internet.
4. En el router de fusión, crear una ruta por defecto hacia el "Gateway" de Internet.
5. En el "firewall", configurar entradas de NAT para el vecino BGP de la VRF. Se debe crear una entrada de NAT estática para cada conexión que se inicia desde el exterior.
6. Crear las reglas necesarias en el "firewall" para permitir el tráfico BGP a través del "firewall".
7. Configurar la VRF interna y el router de fusión como vecinos BGP.
8. Configurar el router de fusión para que la ruta por defecto sea enviada por BGP hacia la VRF interna, en el caso que se encuentre disponible en la tabla de ruteo.
9. Filtrar las actualizaciones de eBGP desde la red interna hacia el router de fusión. Si se reciben actualizaciones de eBGP desde las redes virtuales, se van a replicar estas rutas hacia sus vecinos, inyectando rutas de una VPN dentro de otra.
10. Inyectar la ruta por defecto creada en el paso (8) en las redes virtuales. En este escenario esto implica inyectar la ruta por defecto mediante MP-iBGP en el PE.

Si utilizamos RRs, se debe tener en cuenta que al anunciar el mismo prefijo con dos rutas distintas, los RRs solamente van a publicar una. Para crear el efecto de un balanceo de carga, las distintas rutas se deben anunciar con RDs distintos, así BGP reflejará ambas rutas.

6.3.2.2 Firewalls en modo transparente

Como se vio anteriormente, los firewalls en modo transparente actúan como un bridge L2. En este escenario, la configuración es simple ya que todas las VRFs son vecinos con el router de fusión. El router de fusión inyecta una ruta por defecto en el IGP y esto va a ser redistribuido en MP-iBGP en el PE. Debido a que los firewalls están configurados en modo transparente, éstos no pueden ser usados para NAT y por lo tanto las VRFs deben usar direcciones IP válidas y únicas.

Para la configuración de este escenario (que se detalla en la Figura 54) se necesita realizar lo siguiente sobre cada red virtual:

1. En el router de fusión crear la ruta hacia Internet
2. Publicar las rutas en el IGP utilizado entre el router de fusión y las VRFs en el PE
3. Redistribuir el IGP en MP-iBGP en el PE y viceversa
4. Filtrar las publicaciones de rutas entre las VRFs en el router de fusión, para evitar la publicación de rutas de una VRF dentro de otra.

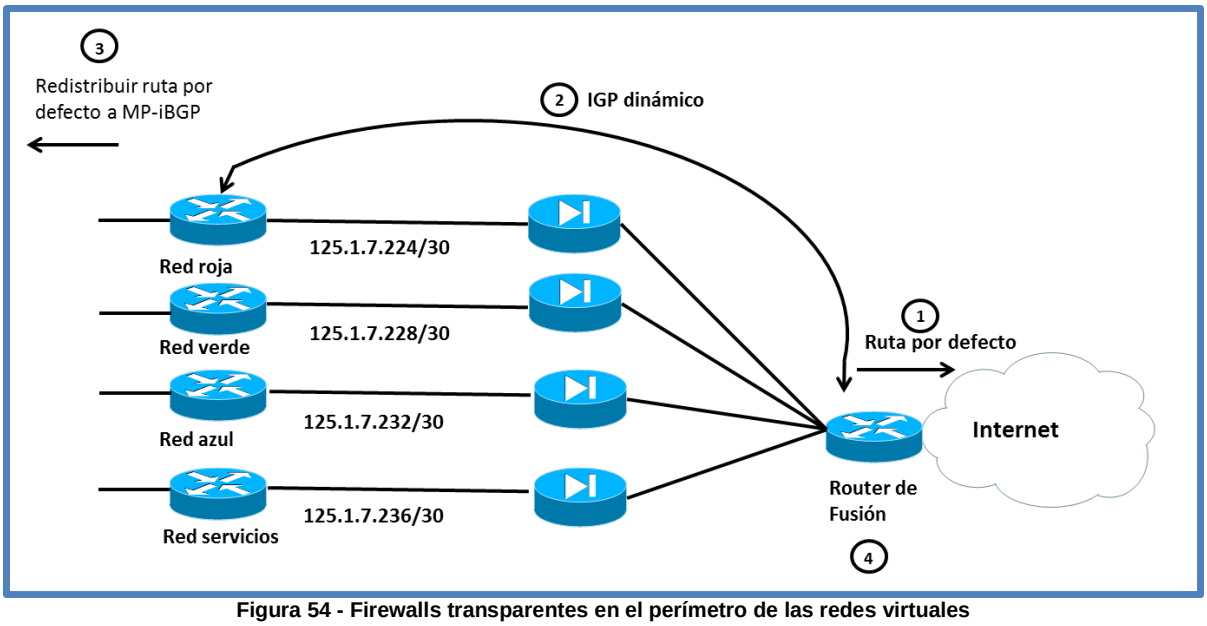

Figura 54 - Firewalls transparentes en el perímetro de las redes virtuales

6.3.3 Ejemplos de servicios compartidos

Los servicios de DNS [62] y DHCP [63] son muy comunes dentro de la empresa y pueden ser compartidos por varias redes virtuales. Como se encuentran muy relacionados con la estructura del direccionamiento IP, es necesario agregar cierta información de la red virtual en los requerimientos al servicio para poder virtualizar el servicio mismo y que sean conscientes de la existencia de redes virtuales.

6.3.3.1 DHCP

Para compartir un servidor de DHCP entre varias redes virtuales, es necesario virtualizar el servicio. El servidor contiene un conjunto de “scopes” para cada red virtual. Si tomamos como ejemplo la red de la Figura 55, el servidor DHCP puede contener una partición para la red virtual Azul con varios “scopes” y otra partición análoga para la red virtual Roja. Cada partición equivale a un servidor DHCP tradicional sin virtualizar y da servicio a una red virtual específica.

Se necesita un mecanismo para identificar de cual red virtual provienen los mensajes de solicitud de servicio. La infraestructura, además de realizar el “relay” de los mensajes, debe ser capaz de marcar los mismos. La opción 82 de DHCP puede ser utilizada para transportar la información de a qué red virtual pertenece la solicitud. Esta información es agregada por el agente de “relay”, por lo tanto cuando llega el mensaje al server, está asociado con la partición que corresponde a la red virtual mediante la opción 82.

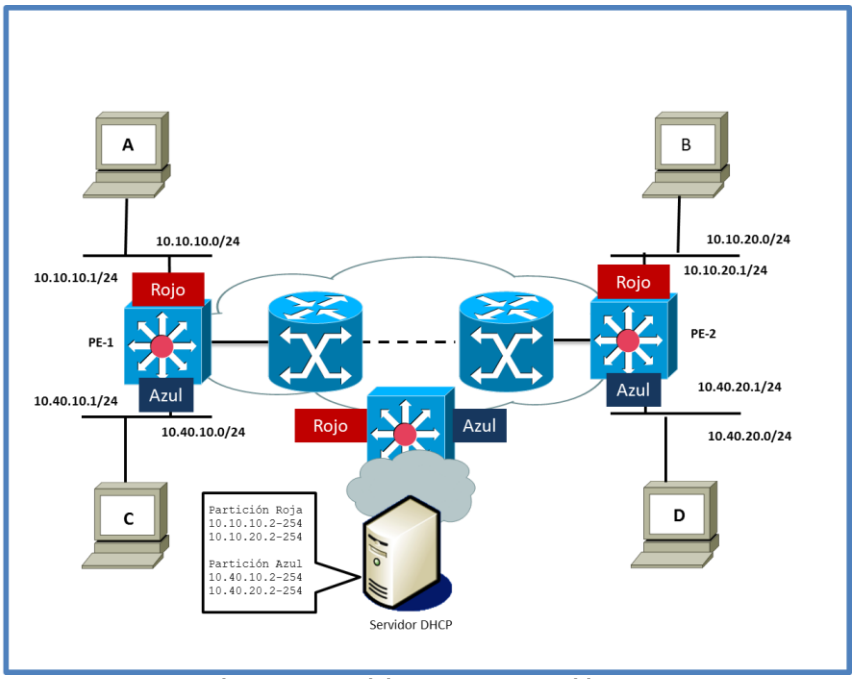

Figura 55 - Servicios DHCP compartidos

En el ejemplo, cuando el equipo C hace un pedido de DHCP, es visto por PE-1. PE-1 le asignará el identificador correspondiente a la red virtual Azul en el campo de la opción 82 de la solicitud y escribirá 10.40.10.1 en el campo "giaddr" (Gateway IP Address) del pedido, convirtiendo el paquete en "unicast" y enviándolo al servidor DHCP. Cuando el servidor recibe el pedido, seleccionará una dirección a asignar del "scope" que corresponda según el valor de la opción 82 y el "giaddr".

6.3.3.2 DNS

La implementación de servicios DNS para redes virtuales utilizando direcciones válidas en Internet es directa y no requiere ninguna consideración especial. Por el otro lado, la implementación de DNS para redes virtuales utilizando direcciones privadas según el RFC 1918 [64] impone algunos desafíos, dependiendo del nivel funcionalidad requerida.

En un nivel básico, los servicios DNS pueden ser compartidos por todas las redes virtuales si son utilizados solamente para resolver nombres que se encuentran fuera de las redes virtuales. En este escenario todas las redes virtuales deben poder alcanzar al servidor DNS que se debe encontrar en una zona de servicios comunes o en la DMZ.

Cuando se requiere resolver nombres de equipos internos y se utilizan direccionamiento IP privado es necesario instalar un servidor DNS dentro de cada red virtual. Este servidor interno permite la resolución de nombres internos y a su vez debe consultar servidores DNS externos para la resolución de nombres en Internet.

En general se recomienda que se instalen servidores internos y externos. Los servidores internos reenvían peticiones de los equipos internos hacia los servidores externos quienes realizan las consultas en nombre de los servidores internos. Para obtener el resultado esperado, es necesario que el servidor de DNS interno sea alcanzable desde el exterior de la red virtual. Para lograr estos se deben configurar en los firewalls una entrada de NAT estática y las reglas necesarias para permitir que los servidores internos sean accesibles desde el exterior.

7. Nuevas tendencias en virtualización de redes

7.1 Introducción

En el presente capítulo se tratan tecnologías alternativas al “networking” tradicional, como ser las “Software Defined Networks” (SDN). Debido al hecho que son una serie de tecnologías en desarrollo, se debieron tomar varias fuentes de libros como ser [65] y [66] (libros publicados en 2013) y principalmente artículos de las Universidades pioneras en el tema como ser el ON.LAB en la Universidad de Stanford [67] así como las definiciones del problema en los cuerpos de estándares [68]. Junto con las fuentes mencionadas, también se utilizaron artículos de difusión de los fabricantes de software/hardware (p.e. [69], [70]) y sitios de discusión especializados sobre el tema, entre otras fuentes.

El desarrollo del presente capítulo incluye una presentación del nuevo enfoque de arquitectura junto con una serie de tecnologías y componentes que dan soporte a la misma, para finalmente presentar una serie de casos de uso (donde la virtualización es uno de los más importantes) de SDN presentes en la empresa.

7.2 Software Defined Networking (SDN)

SDN en enfoque arquitectónico que optimiza y simplifica las operaciones de red vinculando de manera más cercana la interacción entre las aplicaciones, servicios e red y dispositivos sin importar sean reales o virtualizados. Esto se logra generalmente utilizando un punto de control centralizado en la red (denominado controlador SDN), el cual orquesta, media y facilita la comunicación entre las aplicaciones que desean interactuar con los elementos de red y los dispositivos de red que desean enviar información a las aplicaciones. El controlador expone y abstrae funciones de red y operaciones mediante interfaces programáticas bidireccionales [65] [68].

Desde el punto de vista arquitectónico y operacional, los dispositivos de red de hoy en día disponen básicamente de un plano de control y un plano de datos. El plano de datos es el encargado de realizar el “forwarding” de los paquetes; básicamente ejecuta acciones que son ordenadas por el plano de control. El plano de control, como lo sugiere su nombre es donde reside la inteligencia y complejidad del dispositivo; es donde se mantienen las tablas de ruteo/”forwarding”. El plano de aplicaciones o servicios es donde se implementan ciertos servicios que el dispositivo provee (p.e. “logging”, autorización, autenticación, estadísticas, protocolos de ruteo, etc). Esta arquitectura, representada en la Figura 56 ha estado presente por más de 35 años.

La propuesta de SDN es extraer el plano de control y aplicaciones del dispositivo y ejecutarlos de manera centralizada o distribuida en servidores. Como consecuencia de esto, los dispositivos son más sencillos (ver Figura 57).

SDN centraliza las aplicaciones y el plano de control. El controlador (o sistema operativo de red) es el corazón de las redes SDN ya que realiza las siguientes tareas:

- Descubre automáticamente la topología de la red.
- Determina las tablas de ruteo y de flujos para los dispositivos de red.
- Traduce comandos de alto nivel en comandos específicos para su ejecución.
- Mantiene la virtualización de la red.

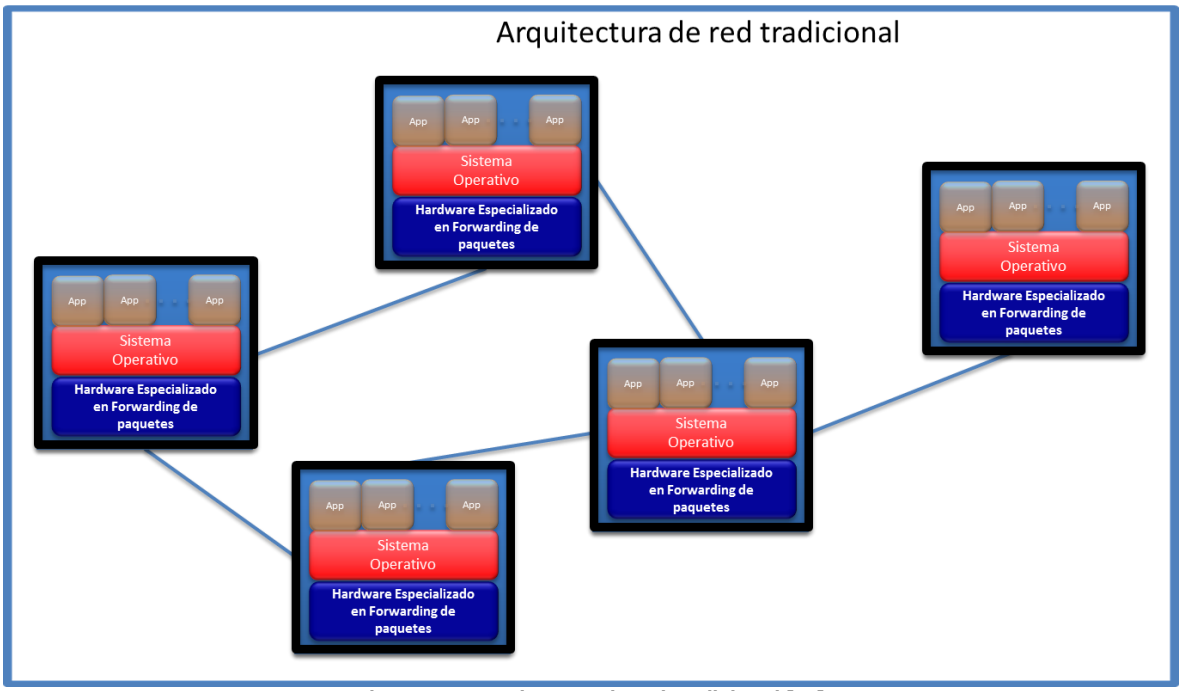

Figura 56 - Arquitectura de red tradicional [71]

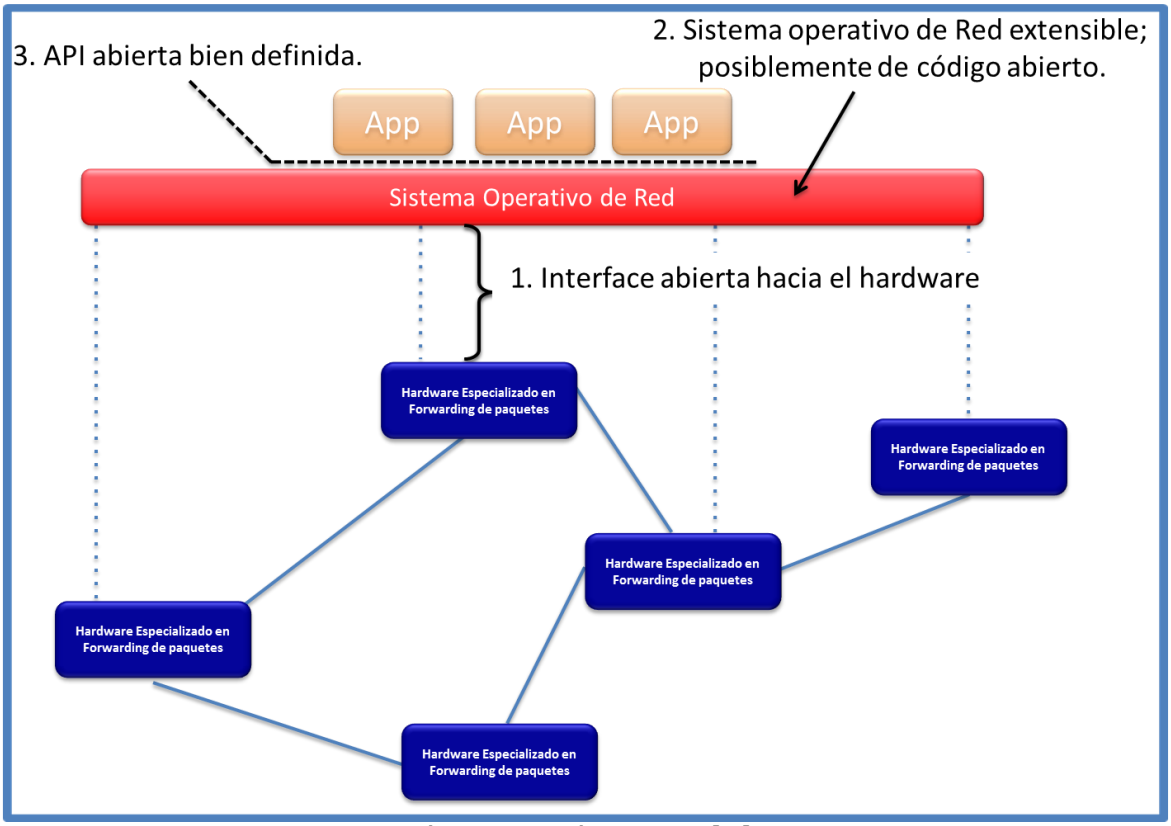

Figura 57 - Arquitectura SDN [71]

También es posible aplicar esta técnica sobre trozos (“slices”) de la red, donde cada porción es una red virtualizada con su respectivo controlador (y las aplicaciones que interactúan con él) que maneja la misma red física, utilizando los servicios de una capa de virtualización como se muestra en la Figura 68.

Los controladores ejecutan en servidores de aplicación, configurados de manera redundante para este fin, resultando en menores inversiones de capital (CAPEX). Estos servidores pueden ser potenciados a medida que sea necesario.

En una solución abierta de SDN, el protocolo de comunicación entre el controlador y los dispositivos es OpenFlow (se verán más detalles sobre este protocolo más adelante), el cual es especificado por la "Open Network Foundation" (ONF). Los dispositivos que adhieren al estándar son denominados "Open Flow Enabled", los cuales deben pasar por un programa de certificación de la ONF [72] La estandarización nos guía para cumplir uno de los objetivos de la SDN que es la creación de un switch genérico (a veces denominado "commodity switch") de bajo costo y alto rendimiento. SDN puede ser vista como una evolución de la red ya que puede interoperar con dispositivos comunes. Un atributo interesante de SDN es que toda la red puede presentarse como un dispositivo a una red adyacente o un "switch". Por ejemplo dos "switches" SDN pueden formar un MLAG contra un "switch" tradicional y el controlador puede administrar el "switch" tradicional mediante SNMP.

Las aplicaciones en SDN interactúan con el controlador mediante interfaces de programación definidas (APIs). El fabricante del controlador va a proveer distintas aplicaciones y el cliente puede programar aplicaciones para necesidades especiales sobre esta interface. Por ejemplo se puede definir redes virtuales para desarrollar un nuevo protocolo o para experimentación.
La presentación de la red virtualizada tiene la ventaja de mostrar únicamente lo que es necesario para la aplicación. Esto es útil para la visualización de la misma y la administración del tráfico. Se agrega el hecho que el controlador tiene una visión total de la topología de la red, incluyendo los servidores (posiblemente máquinas virtuales) conectados.

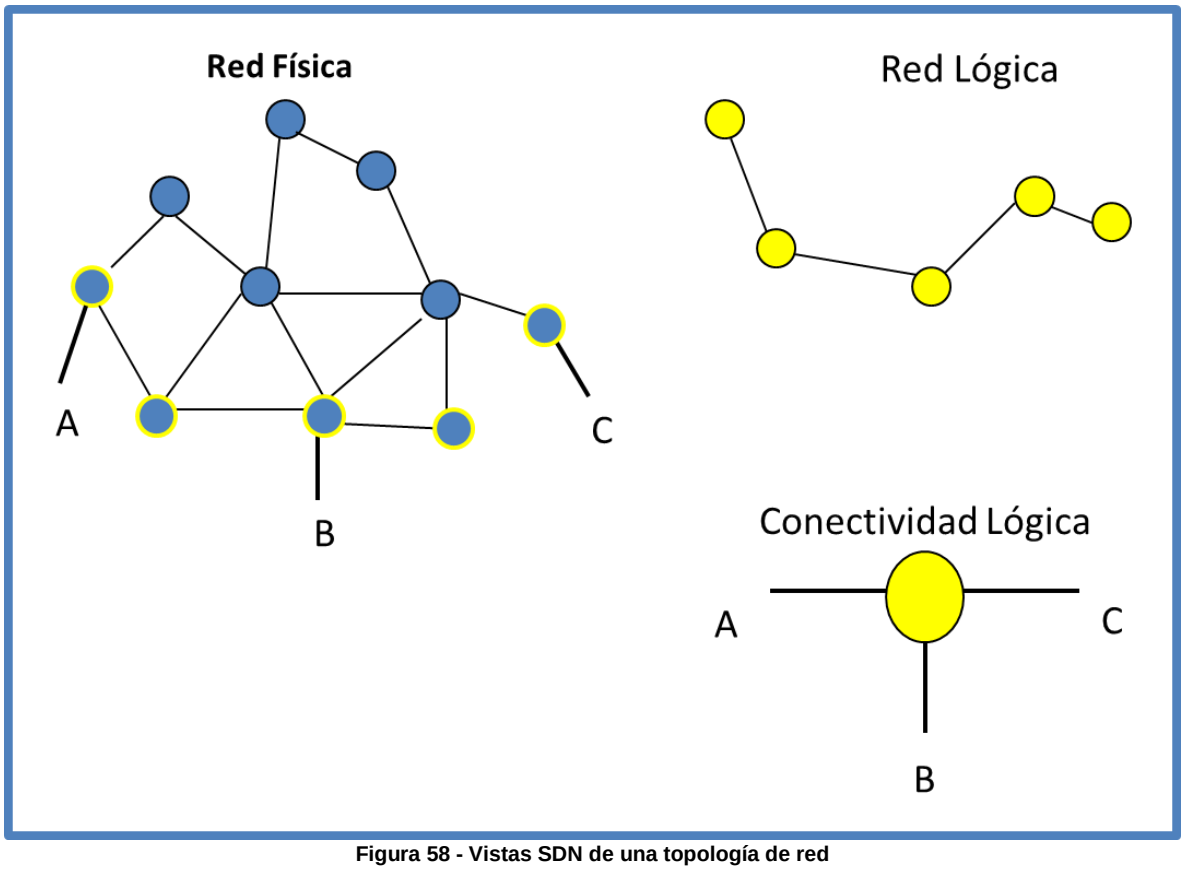

Figura 58 - Vistas SDN de una topología de red

Si por ejemplo tenemos la red física de la Figura 58, la visión lógica de la misma comprende a los elementos A, B, C más todos los elementos que los interconectan. La visión lógica de

conectividad que presenta SDN es similar a que A, B, C estuvieran todos conectados en un mismo “switch”. Lo interesante de SDN es que nos permite establecer condiciones y reglas de conectividad sobre A, B, C como si estuvieran conectados en el mismo “switch” y luego el controlador se encarga de configurar los dispositivos intermedios, haciendo uso de su conocimiento sobre la topología de la red completa, escondiendo la complejidad.

Definir la granularidad de los flujos puede hacerse fácilmente en una red en SDN. Se programan las necesidades de alto nivel en el controlador y éste ejecuta la implementación a lo largo de la red. El “forwarding” de tráfico en SDN puede estar basado en varios campos de un paquete de acuerdo a las reglas definidas. Por ejemplo se puede identificar un flujo en una dirección IP de origen y un puerto de destino como se indica en la Figura 59. En base a esto, se pueden definir acciones como por ejemplo manejar prioridades. Estas condiciones y acciones luego son implementadas mediante tablas de flujo en varios dispositivos a través de la red.

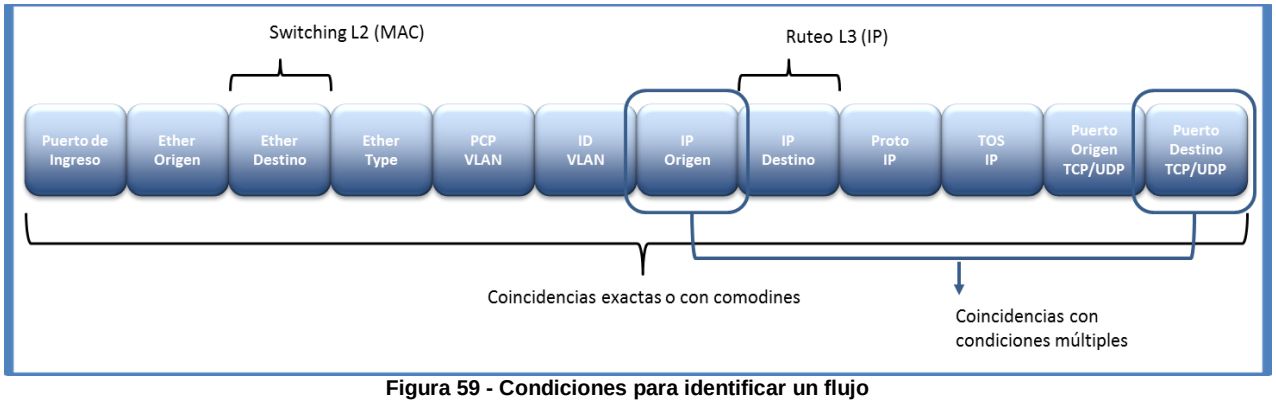

Figura 59 - Condiciones para identificar un flujo

En el ejemplo de la Figura 60 se puede analizar el envío condicional de tráfico entre dos puntos A y B de una red. Se puede definir un camino para un tipo de tráfico (en el ejemplo tráfico de video) y otro para el resto.

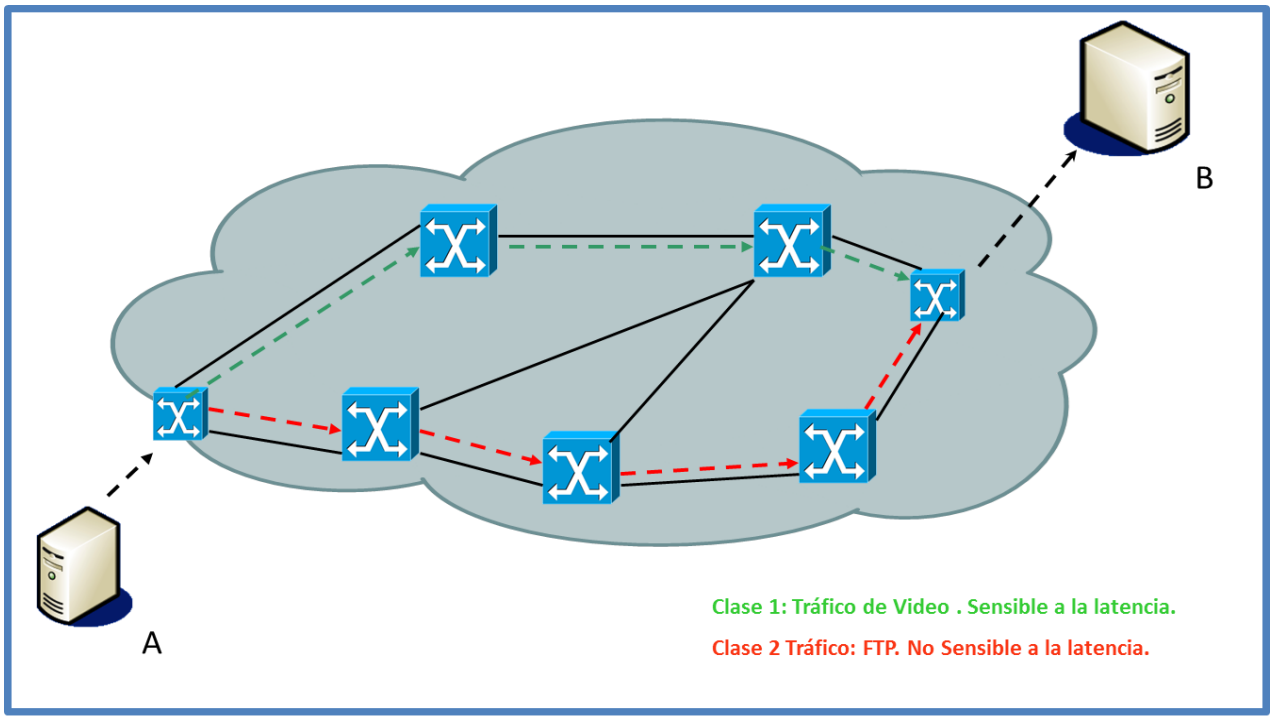

Figura 60 - Distintos comportamientos según el flujo de datos

El tráfico sensible a la latencia tomará un camino mientras que el resto tomará otro. Todo el tráfico tiene la misma dirección IP de origen y la misma dirección IP de destino, por lo que la decisión debe tomarse en base a otros campos como ser puerto TCP, protocolo, etc.
Otro ejemplo es la implementación de Datacenter con múltiples clientes. En estos Datacenters, cada una de las redes de clientes opera como una red segura y privada. Cada cliente define sus propias VLANs y su espacio de direcciones IP que en muchos casos se superponen ya que estas redes operan de manera aislada. Este tipo de configuraciones es común ya que muchas empresas se están cambiando a Datacenters públicos y necesitan mantener su esquema de direccionamiento, como se puede apreciar en la Figura 61.

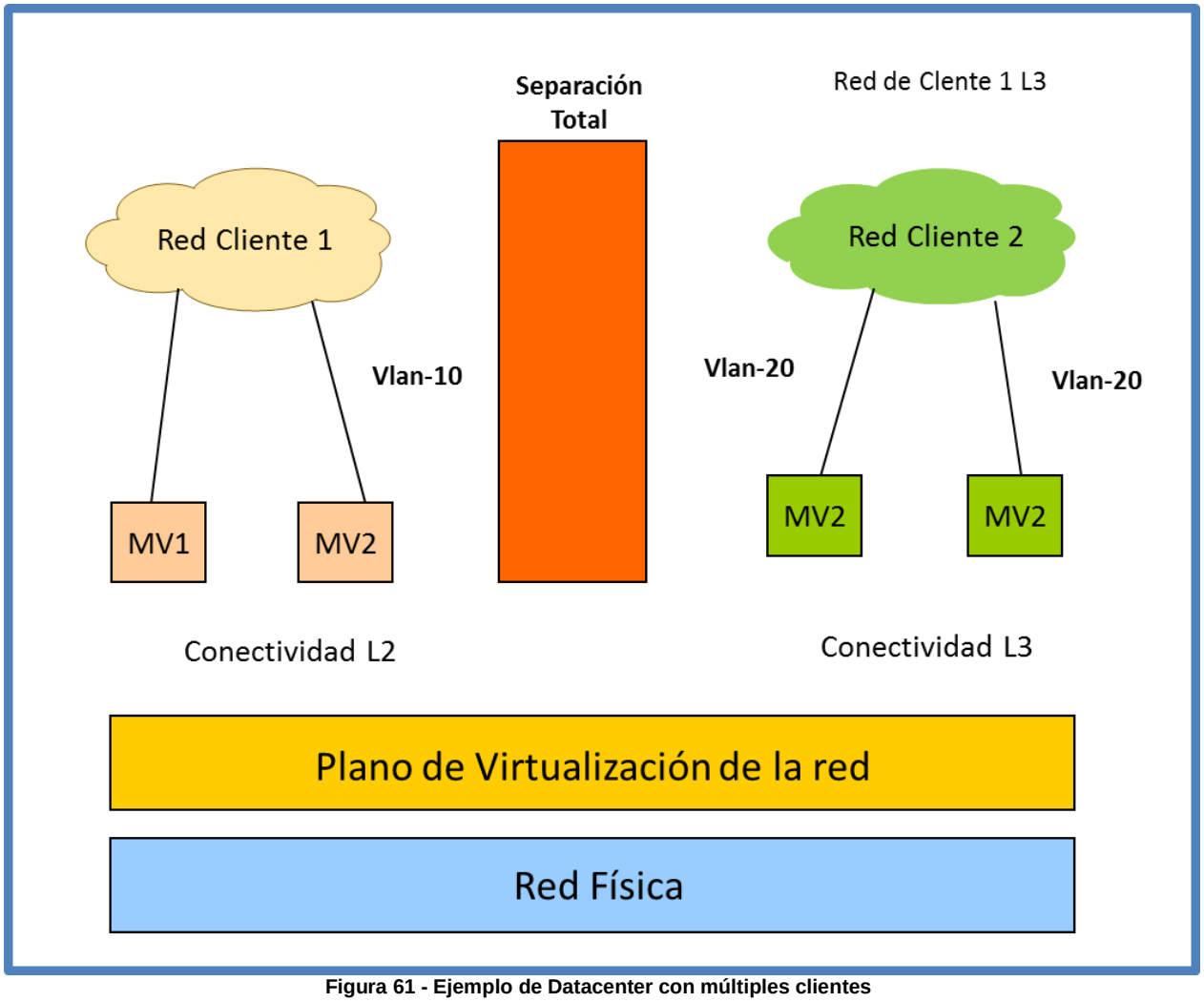

Figura 61 - Ejemplo de Datacenter con múltiples clientes

Otro tipo de funciones de red que se pueden implementar fácilmente con SDN son:

- Filtrado de paquetes – Los flujos son clasificados y sigue una acción de descartar o permitir el tráfico.
- Ruteo basado en políticas –Los flujos son clasificados y la acción es o bien configurar la interface de salida o se cambia la VLAN.
- Rutas estáticas – Los flujos son clasificados según la IP de destino y se configura el puerto de salida.
- NAT – Algunos “switches” OpenFlow pueden soportar la reescritura de la dirección IP de origen/destino o puerto.

En resumen, con el uso de SDN, disponemos de una red más simple, más automatizada, más potente, con menor costo.

Si bien desde sus inicios, el protocolo OpenFlow brinda una interface estandarizada hacia los dispositivos de red, recién en 2012 la ONF establece un grupo de trabajo para intentar definir un estándar de API hacia las aplicaciones / usuarios [73]. Este grupo, en su acta de constitución, establece que la motivación para la creación de este grupo de trabajo es cumplir la promesa a los clientes finales que sus aplicaciones puedan realizar un uso efectivo de los planos de control y datos desacoplados sin quedar atados a un fabricante en particular [74].

Existen otros esfuerzos para la estandarización de las APIs de red en el marco del proyecto OpenDaylight. El 8 de febrero de 2013, la “Linux Foundation” anunció la creación del proyecto OpenDaylight [75] como un proyecto dirigido por la comunidad y soportado por la industria de un "framework" de software libre para acelerar la adopción, incentivar la innovación y crear una aproximación más abierta y transparente a SDN [76]. Los miembros fundadores del proyecto son Arista Networks, Big Switch Networks, Brocade, Cisco, Citrix, Ericsson, HP, IBM, Juniper Networks, Microsoft, NEC, Nuage Networks, PLUMgrid, Red Hat y VMware, los cuales se encuentran comprometidos a donar software y recursos de ingeniería [77] para el proyecto OpenDaylight y de esta manera ayudar a definir el futuro de una plataforma SDN "open source". A diferencia de otros proyectos que buscan definir nuevos estándares, OpenDaylight se encuentra enfocado en crear una interface universal mediante la cual se pueden controlar "switches" (tanto físicos como virtuales) mediante "software", utilizando estándares como OpenFlow así como implementar un controlador de SDN de referencia.

7.2.1 OpenFlow

OpenFlow [78] es un protocolo de red diseñado para administrar y dirigir tráfico de “routers” y “Switches” de varios fabricantes. Programa el plano de control de los dispositivos, desacoplándolo del plano de datos. Es el resultado de una investigación conjunta entre la universidad de Stanford y la universidad de Berkeley. Entre 2008 y 2011, la especificación de OpenFlow fue manejada por el “OpenFlow Consortium”. A partir de 2011, este proyecto es dirigido por “Open Networking Foundation (ONF)”.

El concepto de OpenFlow surgió en la Universidad de Stanford y sus principales desarrolladores fueron los profesores Martín Casado y Nick McKeown. Su objetivo fue crear un ambiente de pruebas en el cual los investigadores pudieran realizar sus pruebas con nuevos protocolos de red sobre la red productiva del campus para que los experimentos se realizaran sobre entornos realistas sin impactar el tráfico productivo.

La primera consideración de OpenFlow fuera de la academia fue la investigación para ver la manera de escalar el ancho de banda en Datacenter masivos, en el cual el procesamiento en paralelo de conjuntos de data masivos tiene lugar en “clusters” de cientos o miles de servidores (como puede ser por ejemplo el buscador de Google). Haciendo un análisis preliminar del problema, se puede ver con facilidad que se requiere un gran ancho de banda horizontal y que una arquitectura en forma de árbol requerirá un “throughput” en los dispositivos de “core” que simplemente no se puede adquirir hoy en día. A partir de esta problemática, las soluciones con OpenFlow se convirtieron de gran interés para las empresas relacionadas con “cloud computing” o servicios virtualizados. En 2011, seis compañías (Deutsche Telekom, Facebook, Google, Microsoft, Verizon y Yahoo) formaron la ONF para dar soporte al desarrollo de tecnologías SDN y por extensión al protocolo OpenFlow.

7.2.1.1 Componentes de OpenFlow

La especificación de OpenFlow [78], define los siguientes componentes:

- Controlador (“Controller”).
- Canal seguro (“Secure Channel”).
- Tabla de flujos (“Flow Table”).
- Tabla de grupos (“Group Table”).
- Pipeline.

El controlador es el plano de control y provee la programación del plano de datos. El controlador maneja el plano de datos agregando, cambiando o borrando registros en las tablas de flujos de los “switches”. El controlador administra los dispositivos mediante el canal seguro generalmente autenticado y cifrado mediante TLS. Los mensajes que se envían a través del canal seguro son seguros y ordenados, aunque no se asegura la entrega ordenada. Los mensajes enviados por el canal seguro, se dividen en tres categorías:

- **Del controlador al dispositivo:**
  Estos mensajes son utilizados por el controlador para administrar el dispositivo. Mediante estos mensajes el controlador no solamente puede agregar, modificar o borrar entradas en las tablas de fuljo del dispositivo, sino que puede consultar su estado y estadísticas, configurar el dispositivo, modificar características de los puertos y enviar paquetes por un determinado puerto del dispositivo.
- **Mensajes asíncronos:**
  Estos mensajes son enviados desde el dispositivo al controlador y pueden ser originados por un paquete o un encabezado de un paquete que no verifica ninguna condición de en las entradas de las tablas de flujos (y por lo tanto debe ser enviado al controlador para su procesamiento), una notificación o cambio en el estado de un flujo o en el estado de un puerto, o un mensaje de error.
- **Mensajes simétricos:**
  Estos mensajes pueden ser enviados tanto por el controlador o el dispositivo y pueden ser mensajes de “hello” (keepalives) “echo requests” o “echo replies”, o mensajes experimentales que habilitan funcionalidades adicionales.

Las tablas de flujos están compuestas por una serie de entradas, donde cada una de ellas consisten de:

- **Campo de coincidencias (“match field”):**
  Especifica la expresión bajo la cual un paquete es identificado. Esta condición puede ser una combinación de: puerto entrante, campos de los encabezados IP o Ethernet, puertos de capas superiores o metadata.
- **Campo contador:**
  Aquí se graban las estadísticas de los paquetes coincidentes.
- **Campo de instrucciones:**
  Mediante este campo, se especifican las acciones a ser realizadas sobre los paquetes coincidentes.

Puede existir más de una entrada en la cual se cumpla la condición. En este caso, se utilizará la que tiene una cantidad mayor de coincidencias y se ejecutará su acción asociada. El paquete puede ser reenviado por un puerto físico o uno virtual como un túnel o una agregación (p.e. LAG). El paquete puede ser enviado al controlador, inundado en los puertos (flooded) o enviado al “switch” por defecto para un tratamiento normal.

Los paquetes pueden ser enviados a un grupo o ingresados en un pipeline hacia otra tabla de flujos.

Un grupo es un medio para aplicar un conjunto común de acciones para agregar flujos. Al designar a un paquete como miembro de un grupo las acciones pueden ser aplicadas eficientemente o cambiadas sobre múltiples flujos. Existe la tabla de grupos, la cual tiene registros de grupo con los siguientes campos:

- **Un identificador de grupo** de 32 bits
- **Una especificación del tipo de grupo**, el cual puede ser uno de los siguientes:
  - **All**: Ejecuta todos los grupos de acciones definidas para el grupo. Esto es utilizad o para “broadcast” o “multicast”, donde el paquete es replicado para cad grupo de acciones.
  - **Select**: Se utiliza un algoritmo de selección (Pej “round robin”) para enviar el paquete a un solo conjunto de acciones del grupo. Se puede lograr el balanceo de carga multi-camino utilizando este tipo de grupo.
  - **Indirect**: Ejecuta un único grupo de acciones para todos los paquetes enviados al grupo. Este tipo de grupo es utilizado de manera eficiente para aplicar un grupo de acciones a múltiples flujos.
  - **Fast Failover**: Este tipo de grupo tienen un conjunto de acciones asociado a cada puerto y ejecuta el primer conjunto vivo (determinado por un mecanismo de detección de “vida”). Como su nombre lo indica, este tipo de grupo permite que las acciones de “forwarding” sean movidas rápidamente de un puerto en falla a uno vivo sin que intervenga el controlador.
- **Un campo con contadores** para registrar estadísticas de los paquetes que tienen coincidencias.
- **Un campo de cubetas (“buckets”) de acciones,** donde cada uno contiene un conjunto de acciones a ejecutar; esto se comporta de manera similar a un conjunto de instrucciones en una entrada de la tabla de flujos. En concordancia con las descripciones de los cuatro tipos de grupo, existen cubetas de acciones para los distintos tipos de grupos existentes.

Otra acción que se puede especificar para un paquete que coincide con una regla es enviarlo hacia otra tabla. Este tipo de procesamiento se denomina “pipeline”, permitiendo crear una jerarquía de procesamiento con una lógica del estilo “if-then-goto”. El “pipeline” de OpenFlow es una serie de tablas de flujo numeradas de manera secuencial, comenzando en cero. Todos los paquetes entrantes al dispositivo se comienzan a procesar a partir de la tabla 0 y luego pueden ser derivados a otra tabla con numeración superior junto con su metadata que puede ser utilizada para generar una coincidencia en la próxima tabla.

7.2.1.2 Instrucciones OpenFlow

De las instrucciones definidas por OpenFlow existe un conjunto que son obligatorias (todos los dispositivos las deben implementar) y otro conjunto que son opcionales. Cuando un dispositivo se conecta a un controlador, lo primero que realiza es informar cuales son las instrucciones opcionales que soporta.

A medida que un paquete se procesa dentro del pipeline, acumula una o más acciones. Las acciones acumuladas se denominan conjunto de acciones (“action set”). Cuando finaliza el procesamiento del “pipeline”, se ejecutan todas las acciones en el conjunto de acciones.

Cuando un paquete entra en un “pipeline”, el conjunto de acciones es vacío. A lo largo del camino, las instrucciones en cada entrada de la tabla de flujos donde existe una coincidencia se modifica el conjunto de acciones agregando, borrando o modificando las instrucciones en el conjunto. Una instrucción en una entrada de la tabla de flujos puede modificar una acción sin modificar el conjunto de acciones.

Las instrucciones en una entrada de la tabla de flujos pueden incluir alguna de las siguientes, pero no más de una instancia de cada una:

- **“Apply-Actions”**: Una lista de una o más acciones se ejecuta inmediatamente sin modificar el conjunto de acciones.
- **“Clear-Actions”:** Borra todas las acciones en el conjunto de acciones, dando como resultado un conjunto vacío.
- **“Write-Actions”:** Agrega una o más acciones al conjunto de acciones. Si se especifica una acción que ya se encuentra en el conjunto, se sobrescribe la acción existente.
- **“Write-Metadata”:** Agrega valores de metadata al campo correspondiente. Se utiliza una máscara para indicar los registros de metadata a modificar.
- **“Goto-Table”:** Especifica la próxima tabla de flujos en el “pipeline” de procesamiento a la cual se va a enviar el paquete. El ID de la tabla debe ser numéricamente superior que la tabla actual.

7.2.1.3 Modos de funcionamiento de OpenFlow

7.2.1.3.1 Centralizado vs Distribuido

OpenFlow puede trabajar en una modalidad centralizada, donde un único controlador maneja todos los dispositivos o en una modalidad distribuida, donde varios controladores manejan la totalidad de dispositivos. La motivación de éste último modo es la escalabilidad del controlador ya que la cantidad de tráfico hacia el controlador crece con el número de dispositivos y a medida que la red crece en diámetro, no importa donde situemos el controlador, pueden haber tiempos altos de instalación de flujos en algunos dispositivos. Ya que la performance del sistema está acotada por el poder de procesamiento del controlador, los tiempos de instalación de entradas en las tablas de flujos pueden crecer significativamente con el aumento de la demanda al crecer la red. Existen alternativas mencionadas en [79] y en [80] resolver esta problemática.

7.2.1.3.2 Ruteo de Flujos vs Agregación de Flujos

En estas modalidades de funcionamiento se distinguen con la granularidad que OpenFlow trata el tráfico. En la modalidad de ruteo de flujos, cada flujo es instalado por el controlador y existe una entrada en la tabla de flujos por cada flujo, donde la condición es a medida para dicho flujo. Esta modalidad establece un control fino de flujos lo cual lo hace ideal para redes de campus. En la modalidad de agregación de flujos, una entrada en la tabla de flujos, corresponde un grupo (potencialmente grande) de flujos. Esto se logra utilizando comodines en las condiciones de la regla. Este modo de funcionamiento es ideal para un número grande de flujos, por ejemplo en una red de “backbone”.

7.2.1.3.3 Reactivo vs Proactivo

La diferencia entre estos modos es como va a reaccionar el dispositivo cuando le llega un paquete en función de la información instalada en sus tablas de flujos. En el modo reactivo, el dispositivo comienza con su tabla de flujos vacía y al llegar el primer paquete le consulta al controlador que hacer con él (en realidad el dispositivo envía al controlador la metadata del paquete recibido y la dirección de un buffer en el dispositivo donde está almacenado el

paquete). Una vez que el controlador decidió que hacer con el paquete envía al controlador las instrucciones para dar de alta las entradas en la tabla de flujos. La próxima vez que se reciba un paquete con las mismas características, el dispositivo ya sabe qué hacer con el mismo. Este modo de funcionamiento trae consigo una utilización eficiente de la tabla de flujos, pero como contrapartida cada flujo tiene un tiempo de inicialización y si el dispositivo se desconecta del controlador, tiene utilidad limitada.
El modo proactivo requiere que el controlador de alta en la tabla de flujos del dispositivo las reglas necesarias para su funcionamiento, lo cual implica un tiempo de inicialización cuando el dispositivo se enciende, pero no se necesita tiempo adicional de configuración como en la otra modalidad. En este modo si se pierde la conexión con el controlador, el tráfico sigue fluyendo normalmente. Este modo de funcionamiento lo normal es utilizar reglas del tipo agregación (con comodines).

7.2.1.4 Controladores OpenFlow

La arquitectura desacoplada del plano de control y de datos de SDN se puede comparar con un sistema operativo y el “hardware” de un servidor. El controlador OpenFlow provee una interface hacia los dispositivos OpenFlow. Utilizando esta interface, las aplicaciones de red (“NetApps”) pueden ser desarrolladas para realizar funciones como control y administración y ofrecer nuevas funcionalidades. Las aplicaciones se escriben como si la red se presenta como un único sistema.

Los controladores OpenFlow actúan como un sistema operativo de red y deben implementar al menos dos interfaces, una hacia el sur (“southbound”) que permite a los dispositivos OpenFlow comunicarse con el controlador y una interface hacia el norte (“northbound”) que presenta una interface de programación (API) a las aplicaciones de control y administración de la red (“NetApps”). La interface estándar hacia el sur existente es OpenFlow, la cual se encuentra bien definida y es un estándar de facto (aunque existen fabricantes que implementan otra interfaces hacia el sur). Sin embargo, no existe un estándar aceptado (aunque si existen procesos de estandarización como ser OpenDaylight [75] y un grupo de trabajo de la ONF [74]) en la interface hacia el norte y estas tienden a ser implementadas de manera particular en cada caso de uso para aplicaciones particulares. Existen muchos controladores OpenFlow; a continuación se brindará un resumen de las principales funcionalidades de algunos de ellos ya sea por razones históricas o de sus características resulten notables para el presente estudio.

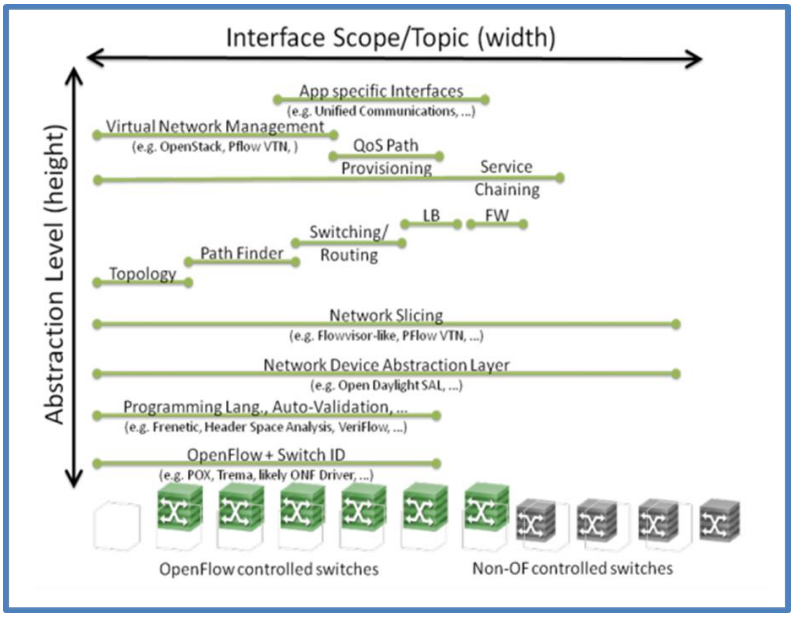

Figura 62 - Completitud de la Interface Openflow [73]

7.2.1.4.1 NOX/POX

NOX [81] fue desarrollado por Nicira que luego donó el código a la comunidad de investigadores en 2008 [81]. Fue el primer controlador OpenFlow de código abierto y fue extendido y soportado por la actividad en el ON.LAB [67] en la Universidad de Stanford con contribuciones de UC Berkeley. NOX provee una interface en C++ hacia OpenFlow (v 1.0) y un modelo de programación asíncrono orientado a eventos.

NOX es un controlador y a su vez un "framework" para el desarrollo de aplicaciones SDN. NOX provee métodos y APIs para interactuar con "switches" OpenFlow que incluyen un manejador de conexiones y un motor de eventos. Existen componentes adicionales que extienden la API como ser ruteo, topología (LLDP) y una interface implementada como un "wrapper".
NOX se utiliza de manera frecuente en investigaciones académicas para desarrollar aplicaciones de SDN en la investigación de protocolos de red. Un efecto colateral de este uso por parte de la comunidad académica es que existe código de ejemplo que emula un "switch" con capacidad de aprendizaje y un "switch" global de red, el cual puede ser utilizado como punto de partida para varios proyectos de programación y experimentación.

Las aplicaciones más populares de NOX son SANE y Ethane [65]. SANE es un enfoque de representar la red como un sistema de archivos. Ethane es una aplicación que parte como un proyecto de investigación de la Universidad de Stanford para seguridad centralizada al nivel de listas de control de acceso. Ambas aplicaciones demostraron la efectividad de SDN en reducir el número de líneas de código requerido para implementar funcionalidades similares [82]
POX es la nueva versión de NOX basada en Python (o NOX en Python). La idea detrás de este desarrollo es generar una nueva plataforma en Python con una API de alto nivel, incluyendo un grafo de topología consultable de manera programática y soporte para virtualización.
POX tiene las siguientes ventajas sobre NOX:

- Dispone de una interface Python de OpenFlow.
- POX dispone de componentes reusables de ejemplo como ser selección de caminos, descubrimiento de topología, etc.
- POX ejecuta sobre cualquier plataforma y puede ser empaquetado con un "runtime" de Python para un despliegue fácil.
- POX es multiplataforma.
- POX utiliza el mismo entorno de visualización (GUI) que NOX.
- POX tienen un rendimiento comparable a las aplicaciones escritas en Python para NOX.

NOX y POX se comunican con "switches" OpenFlow v1.0 e incluyen soporte especial para Open vSwitch [83]

7.2.1.4.2 Beacon

Beacon [84] [85] es un controlador, portable entre distintas plataformas ("cross-platform") y modular de alto rendimiento escrito en Java que soporta la operación tanto basada en eventos como basada en "threads" [85]. Otros de los objetivos del desarrollo de Beacon es el ser amigable con el usuario para el desarrollo de aplicaciones. El desarrollo de Beacon comenzó en 2010 y ha sido utilizado en varios proyectos de investigación, cursos de redes y despliegues de prueba. Los módulos en Beacon pueden ser iniciados/parados/reiniciados/instalados durante la ejecución ("runtime") sin interrumpir otros módulos no dependientes.

Beacon exploró nuevas áreas del diseño de controladores OpenFlow, poniendo foco en ser amigable para el desarrollador de aplicaciones, de alta performance [84] y tener la habilidad de iniciar/parar aplicaciones existentes en tiempo de ejecución. Beacon mostró alta performance y

la capacidad de escalar de manera lineal con la cantidad de “cores”.

7.2.1.4.3 Floodlight / BigSwitch

Floodligth [86] [69] es un controlador SDN, contribuido a la comunidad por Big Switch Networks. Se basa en Beacon de la Universidad de Stanford. La arquitectura de controlador, así como su API son comunes con la versión comercial (BNC) producida por Big Switch Networks (en la versión comercial del controlador se ofrece virtualización y la aplicación BigTap [69]).
La arquitectura principal de Floodlight [86] es modular con componentes que incluyen administración de topología, manejo de dispositivos, cálculo de caminos, acceso a estadísticas (OpenFlow) y una abstracción de almacenamiento para guardar el estado.
Estos componentes son definidos como un conjunto de servicios instalables que exportan un estado. El controlador se presenta como como un conjunto extensible de APIs REST así como un sistema de notificación de eventos. El API permite a las aplicaciones consultar y guardar el estado del controlador, así como suscribirse a eventos emitidos desde el controlador utilizando “Java Event Listeners”.

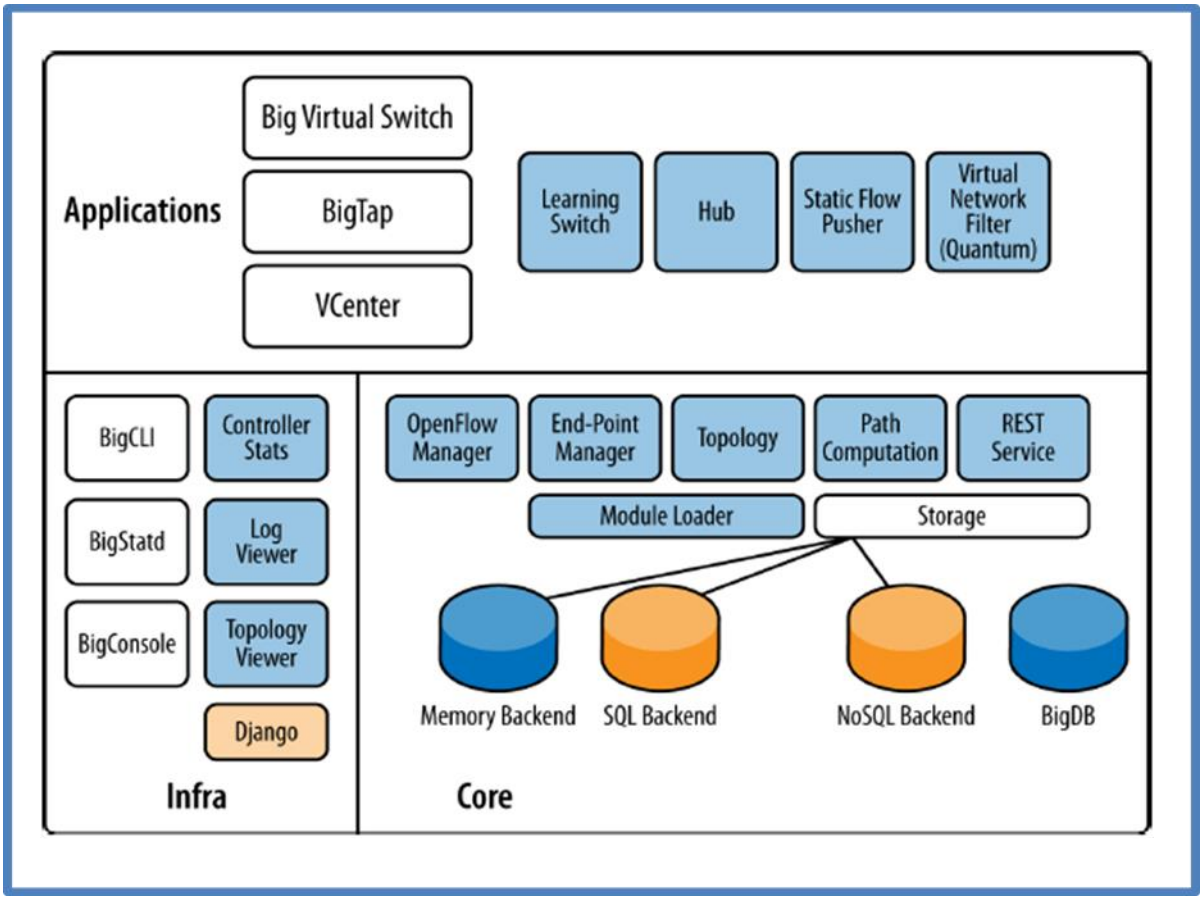

Figura 63 - Arquitectura de Floodlight / BNC [65]

En la Figura 63, los componentes no coloreados corresponden exclusivamente a la versión comercial (BNC); a su vez, la versión comercial tiene mejoras en varios de los módulos.
El módulo principal maneja la E/S desde los “switches” y traduje los mensajes OpenFlow a eventos de Floodlight, creando un “framework” manejado por eventos asíncronos. Floodlight incorpora un modelo de “threads” que permite a los módulos compartir “threads” con otros módulos. El manejo de eventos dentro de esta estructura ocurre en el contexto del “thread” del módulo que publica el evento. Para proteger las estructuras de datos compartidas se utilizan “locks” sincronizados. Las dependencias entre componentes se resuelven al momento de carga

mediante configuración.

El manejador de topologías utiliza LLDP para descubrir dispositivos y verificar si cumplen con OpenFlow o no.

El controlador Floodlight puede interoperar con cualquier dispositivo que soporte OpenFlow
Las aplicaciones incluidas por este controlador incluyen:

- **Plug-in Neutron**: Como interface para el sistema de administración de "cloud" OpenStack
- **Static Flow Pusher**: Permite a los usuarios insertar flujos de manera manual.
- **Circuit Pusher**: Permite crear entradas permanentes en los "switches" a lo largo de un camino.
- **Firewall**: Aplica listas de control de acceso a los paquetes
- **Virtual Switch**: Es una aplicación de virtualización de redes que ejecuta sobre el controlador y permite la creación de múltiples redes L2 lógicas sobre un único dominio L2.
- **Big Tap (BNC)**: Es una aplicación de monitoreo avanzado que permite analizar el tráfico en cualquier punto de la red y entregar estas estadísticas a las aplicaciones de monitoreo o seguridad según la política definida.
- **Big Virtual Switch (BNC)**: Es una aplicación de virtualización de redes orientado a Datacenters, permitiendo automatizar el aprovisionamiento de la infraestructura de red en ambientes de "cloud" con múltiples clientes.

7.2.1.4.4 OpenDailyght

El controlador del proyecto Open Daylight [75] (ODL) es una infraestructura altamente disponible, modular, extensible, escalable con soporte para múltiples protocolos construido para despliegues de SDN en redes modernas heterogéneas de múltiples vendedores. El modelo basado en una capa de abstracción de servicios ("Service Abstraction Layer" – SAL) (ver Figura 64) provee las abstracciones necesarias para soportar múltiples protocolos hacia el sur ("SouthBound") mediante "plugins"; por ejemplo OpenFlow. La arquitectura extensible expuesta hacia el norte ("NorthBound") provee interfaces REST para las aplicaciones débilmente acopladas y servicios mediante la especificación OSGi [87] para aplicaciones con alto nivel de acoplamiento. El "framework" OSGi es responsable por la naturaleza modular y extensible de la solución, proveyendo versionado y administración del ciclo de vida para los módulos y servicios OSGi.

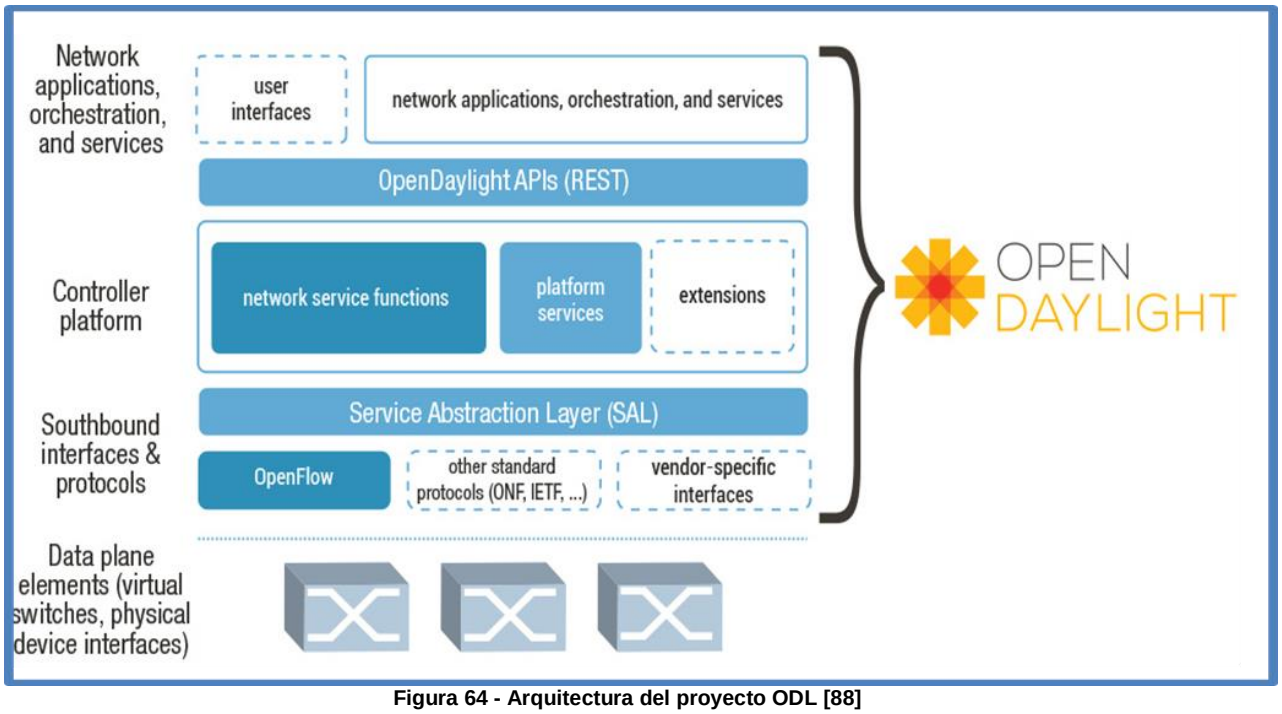

Figura 64 - Arquitectura del proyecto ODL [88]

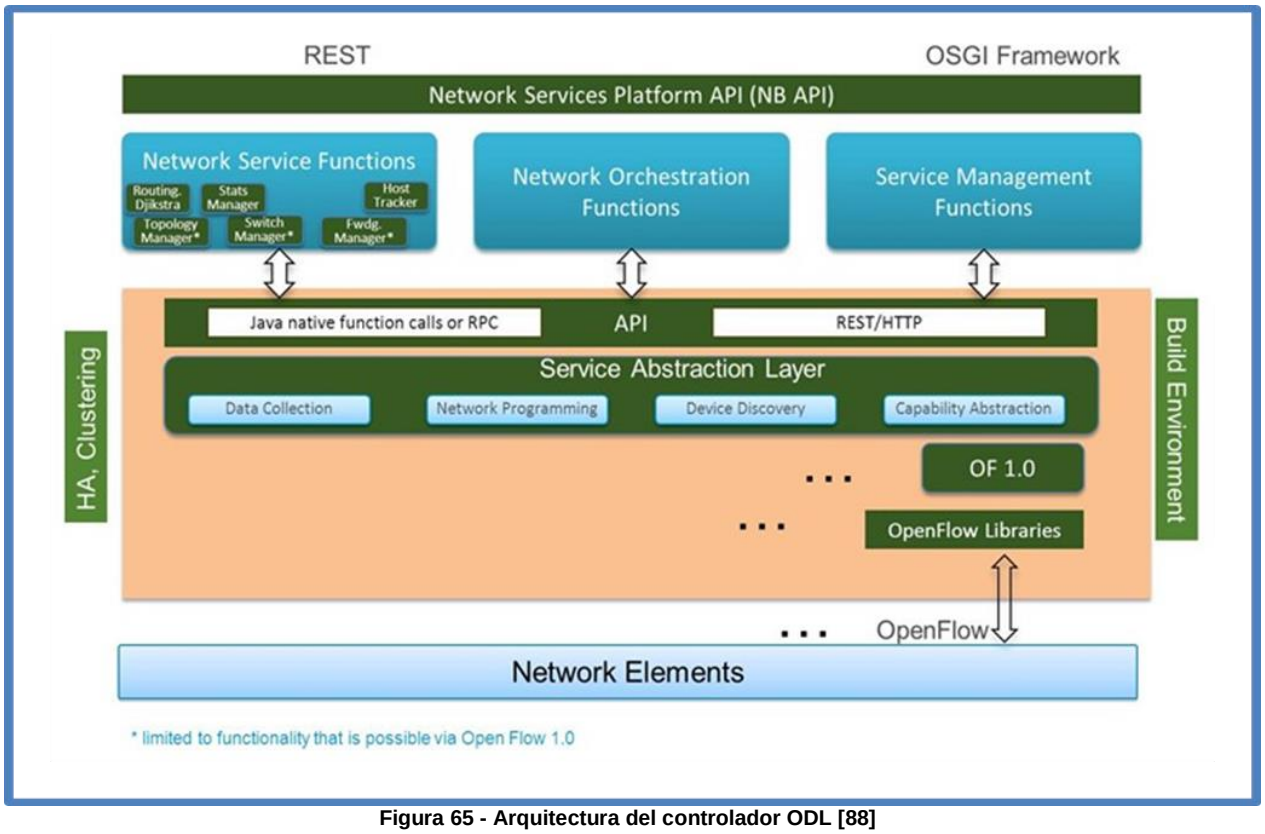

Figura 65 - Arquitectura del controlador ODL [88]

La interface hacia el sur (como se aprecia en la Figura 65) puede soportar varios protocolos anexados como “plugins”. Por el momento soporta OpenFlow, pero se espera que varios de los contribuyentes al proyecto [77] agreguen estos módulos como parte de sus contribuciones/proyectos. Estos módulos son enlazados de manera dinámica con la SAL. La SAL expone los servicios que son consumidos en las capas superiores e intenta descubrir cómo satisfacer el servicio requerido sin tener en consideración el protocolo que se utiliza entre

el controlador y el dispositivo (generalmente OpenFlow). Este mecanismo provee una protección de la inversión hacia las aplicaciones en la medida que el protocolo OpenFlow y otros protocolos evolucionan. La información acerca de las capacidades y la alcanzabilidad de los dispositivos de red se almacena por el "Topology Manager". Los otros componentes (por ejemplo "ARP handler", "Host Tracker", "Device Manager" y "Switch Manager") ayudan generando la base de datos de topología para el "Topology Manager". La API del "Switch Manager" mantiene los detalles del elemento de red. A medida que los elementos de red se descubren, el "Switch Manager" guarda sus atributos en una base de datos. El controlador expone una API abierta hacia el Norte utilizando el "framework" OSGi y REST bidireccional, las cuales son utilizadas por las aplicaciones. La lógica de negocio y algoritmos reside en las aplicaciones ("NetApps"), las cuales utilizan el controlador para obtener datos de la red, ejecutan su algoritmo para realizar los análisis y luego utilizan el controlador para orquestar nuevas reglas a lo largo de la red. El controlador ODL soporta alta disponibilidad basada en un clúster, donde existen varias instancias del controlador ODL que actúan como una única instancia lógica. Esto no sólo brinda redundancia, sino que permite escalar el modelo de manera lineal.

7.2.1.4.5 OnePk

El controlador OnePK [70] (Cisco) es un controlador comercial que sigue el concepto de "framework" al integrar muchos "plugins" en su interface hacia el sur (OpenFlow entre ellos) e incluyendo un "plugin" Que implementa un protocolo propietario que es la API de Cisco OnePK.

La arquitectura del controlador es un "framework" OSGI basado en Java que utiliza un modelo de almacenamiento en memoria y provee una interface REST bidireccional (ver Figura 66). Es posible su utilización en clúster utilizando los servicios de JBoss.

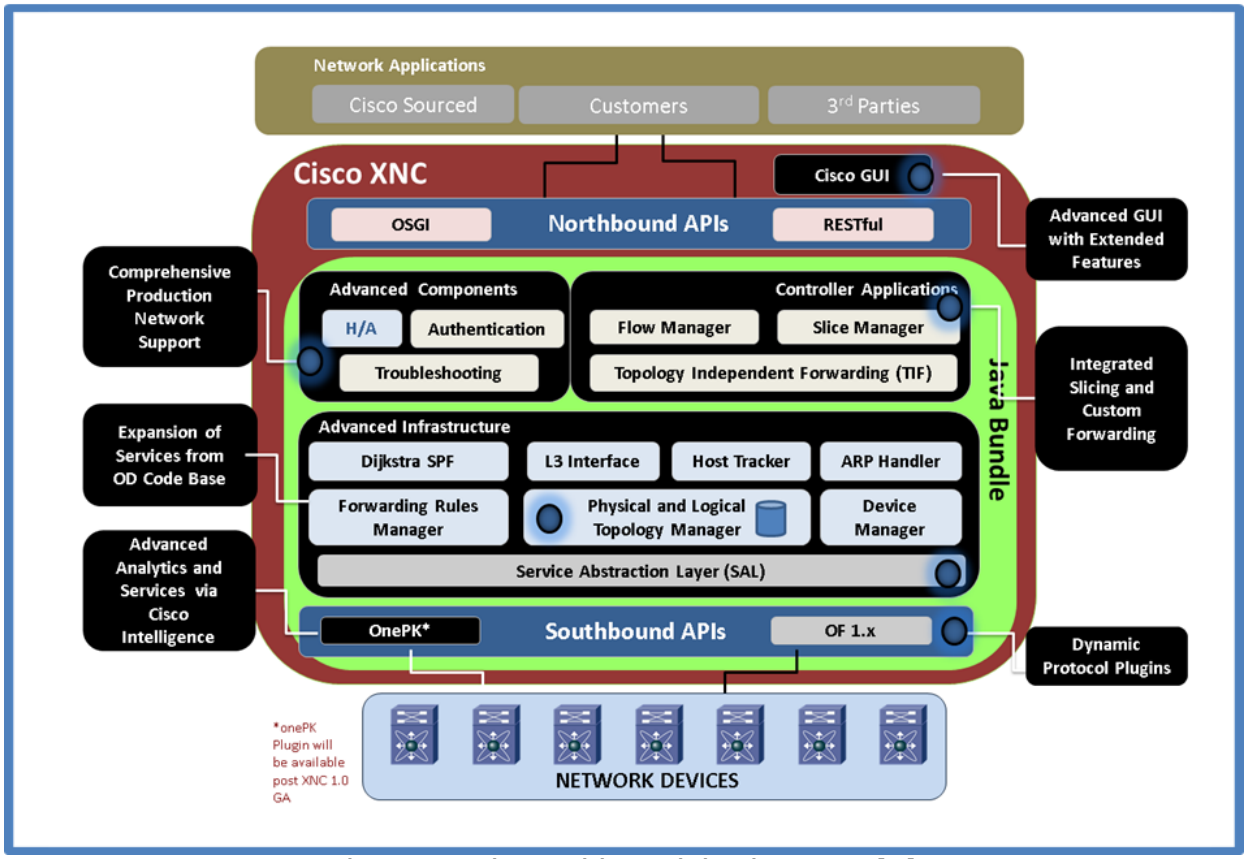

Figura 66- Arquitectura del controlador Cisco OnePK [89]

Cisco afirma que la lógica del controlador [90] [70] es capaz de reconciliar distintas decisiones

de “forwarding” se superponen desde múltiples aplicaciones y a su vez brindar abstracción para hacer posible el “troubleshooting” así como también el descubrimiento de capacidades y mapeo.

Si bien no es inusual para los fabricantes proveer a los clientes con un SDK (elementos propietarios de programación de los dispositivos previos a SDN), el controlador de Cisco implementa esto como un “plugin” en un “framework” general. Esto permite el uso de la SDK propietaria en un entorno de soluciones SDN (p.e. haciendo interactuar la API de OnePK con OpenFlow) en lugares donde el SDK pueden agregar valor.

En controlador OnePK se acerca al controlador idealizado en un ambiente SDN. Provee una API RESTful extensible, un entorno de programación integrado, múltiples motores de cómputo, soporte para múltiples protocolos hacia el sur que pueden ser utilizados para conectarse con una amplia variedad de dispositivos (reales y virtuales). Este controlador también tiene capacidad para el manejo de su estado y almacenamiento de la configuración “online”, “offline” y distribuida. También tienen la capacidad de comunicación horizontal y coordinación de la forma controlador-controlador. A su vez, implementa una capa de abstracción que facilita la comunicación del estilo muchos-contra-muchos. Es esta característica lo que lo diferencia de otros controladores ya que esto le permite la extensión de manera fácil.

Este controlador se utiliza como base para el controlador de referencia definido por el proyecto Open Daylight ya que Cisco aportó parte de su código para el proyecto de la Linux Foundation [77].

7.2.1.4.6 RouteFlow

RouteFlow [91] es un proyecto de código abierto que provee ruteo IP virtualizado sobre dispositivos compatibles con OpenFlow. Está compuesto por una aplicación OpenFlow (controlador), un servidor independiente y un entorno de red virtual que reproduce la conectividad de la infraestructura de ruteo IP, como se puede observar en la Figura 67. Los motores de ruteo IP generan la FIB (“Forwarding Information Base”) en las tablas de ruteo de Linux, dependiendo de los protocolos de ruteo configurados (OSPF, BGP, etc). RouteFlow combina la flexibilidad de los motores de ruteo de código abierto (Pej Quagga) con la performance de los dispositivos OpenFlow. Los principales componentes de RouteFlow son:

- Cliente RouteFlow (RF-Client)
- Servidor RouteFlow
- Proxy RouteFlow (RF-Proxy)

El objetivo principal de RouteFlow es desarrollar un “framework” de código abierto para soluciones IP de ruteo virtual sobre dispositivos que implementan la API de OpenFlow. RouteFlow apunta hacia una arquitectura de ruteo que la performance de “hardware” comercial con la flexibilidad de soluciones de ruteo de código abierto ejecutando en servidores de propósito general.

Los principales resultados en el diseño de aplicaciones de ruteo utilizando RouteFlow son:

- La migración de soluciones de ruteo tradicionales hacia soluciones SDN/OpenFlow.
- Entornos de código abierto para dar soporte a distintos caso de uso de virtualización de redes (routers lógicos, agregación de routers).
- Modelo de red IP “Routing-as-a-Service”.
- Interoperabilidad con dispositivos tradicionales.

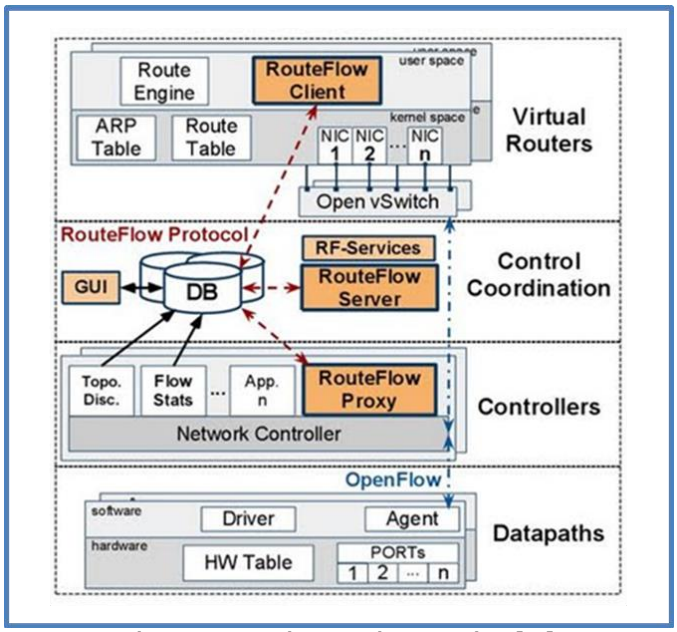

Figura 67 - Arquitectura de RouteFlow [91]

7.2.1.4.7 FlowVisor

Una red SDN puede tener algún nivel de descentralización mediante el uso de múltiples controladores lógicos. Un tipo interesante de controlador tipo proxy llamado FlowVisor [92] [93] se utiliza para agregar a la red OpenFlow un nivel de virtualización. De esta manera, se permite a un grupo de controladores que controlen conjuntos de dispositivos físicos cuya intersección no es vacía. Esta facilidad se desarrolló en un principio para conducir experimentos en la misma red productiva sin afectar el tráfico, pero demostró la facilidad para desplegar nuevos servicios en entornos SDN. FlowVisor se puede considerar como un controlador especial ya que actúa como un "proxy" transparente entre los dispositivos OpenFlow por un lado y los controladores OpenFlow por el otro como se muestra en la Figura 68 y en la Figura 69.

FlowVisor crea "trozos" de recursos de red y delega el control de cada trozo a un controlador distinto, promoviendo el aislamiento entre trozos. FlowVisor fue desarrollado originalmente por la Universidad de Stanford y se utiliza en redes académicas para dar soporte a virtualización donde los experimentos obtienen su trozo de red aislado y lo controlan utilizando su propio controlador y una serie de aplicaciones para control y administración. FlowVisor permite ejecutar experimentos en ambientes productivos reales con tráfico real. Dado que es de código abierto, es posible modificar el código para adaptarlo a las necesidades. Ya que cuenta con una interface en "JavaScript Open Notation" (JSON) para los usuarios, el lenguaje de programación Java para los desarrolladores, se dispone de un entorno donde se puede ajustar FlowVisor fácilmente.

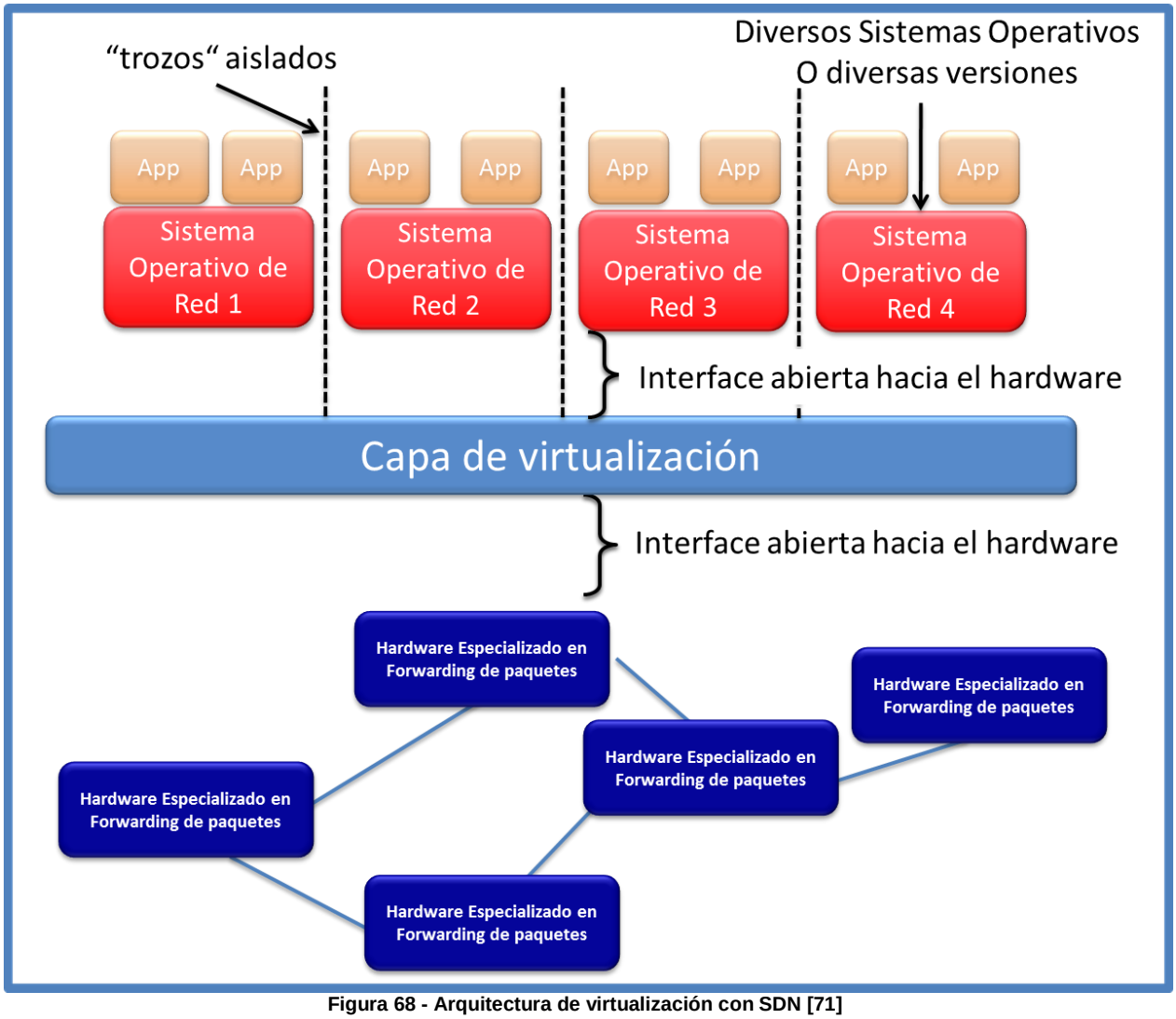

Figura 68 - Arquitectura de virtualización con SDN [71]

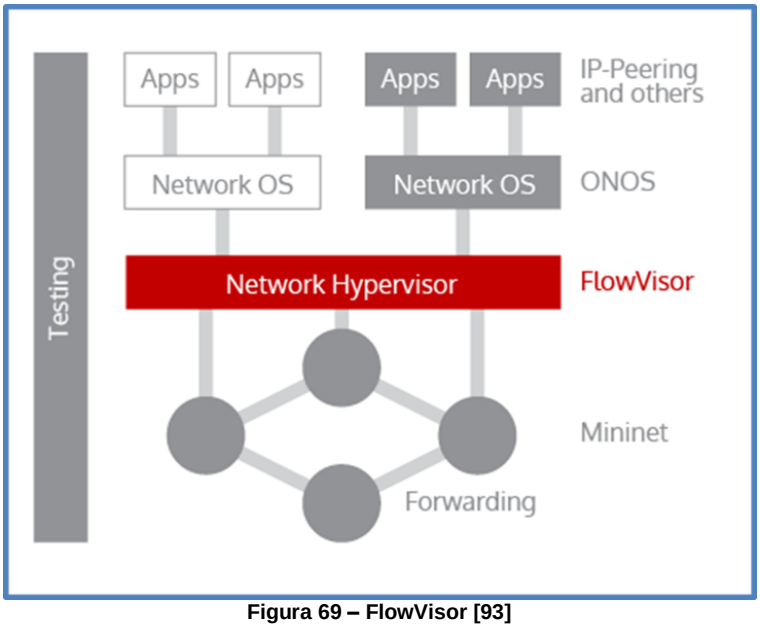

Figura 69 – FlowVisor [93]

7.2.1.5 Switches compatibles con OpenFlow

7.2.1.5.1 Implementaciones en Hardware

A continuación se presenta una breve lista de las opciones disponibles hoy en el mercado de "switches" compatibles con OpenFlow. Esta lista pretende ofrecer opciones comerciales disponibles en el mercado [94].

Fabricante	Modelo
Brocade	NetIron CES 2000 Series, CER 2000
Hewlett Packard	3500/3500yl, 5400zl,6200zl,6600, 8200zl
IBM	RackSwitch G8264, G8264T
Juniper	EX9200 Programmable switch
NEC	PF5240, PF5820
Pica8	P-3290, P-3295, P-3780, P3920
Pronto	3290 and 3780
Broadcom	BCM56846
Extreme Networks	BlackDiamond 8K, Summit X440, X460, X480
Netgear	GSM7352Sv2
Arista	7150, 7500, 7050 series
Cisco	Cisco Nexus 1000V Switch para VMware vSphere
Cisco	Catalyst 3650/3850 [95] , 4500-E [96]

7.2.1.5.2 Implementaciones en software

Existen varis implementaciones de "switches" que soportan OpenFlow que se pueden utilizar para implementar una red basada en OpenFlow o probar aplicaciones de red.

- **Open vSwitch**
  Es un "switch" virtual multicapa utilizable para redes en producción. Tiene la capacidad de operar integrado a una plataforma de virtualización o directamente sobre dispositivos de comunicaciones [83].

- **Indigo**
  Indigo es una implementación OpenFlow de código abierto que ejecuta en "switches" físicos. Se basa en la implementación de referencia de OpenFlow de la Universidad de Stanford y es parte del proyecto Floodlight [97].

- **Pantou**
  Pantou convierte un "Access Point" comercial inalámbrico en un "switch" que soporta OpenFlow, implementado como una aplicación sobre OpenWrt. El módulo se basa en la implementación de referencia de la Universidad de Stanford [98].

7.2.2 Casos de uso de SDN en la empresa

SDN comprende un conjunto de tecnologías que permite que la red sea programable, identificando los flujos de red y actuando sobre ellos de manera granular. Mucho se habla de lo que es SDN, pero lo interesante es descubrir que se puede realizar con ella. A continuación se presentan una serie de casos de uso de SDN, que son aplicables en las distintas áreas de la

empresa. La lista no pretende ser exhaustiva sino que ver las principales aplicaciones y el lugar donde aplican a la empresa. En el contexto del presente trabajo, se deben resaltar aquellos casos relacionados con la virtualización [99].

Caso de Uso	Lugar de aplicación	Aporte de SDN	Beneficio
Virtualización de redes. Redes Multi Usuario	Datacenter	Creación dinámica de redes segregadas equivalentes desde el punto de vista topológico a lo largo del Datacenter. Escala más allá del límite de 4K VLANs	Mejor utilización de los recursos del Datacenter. Tiempos más rápidos de despliegues (de días a minutos) mediante las APIs de automatización.
Virtualización de redes. Redes Elásticas	Datacenter	Creación de redes agnósticas a la ubicación a lo largo de racks o Datacenters, con movilidad de máquinas virtuales y reubicación dinámica de recursos.	Se pueden generar aplicaciones más simples con una mayor tolerancia a fallas sin una codificación compleja y un mejor uso de recursos ya que las máquinas virtuales se mueven de manera transparente para consolidar las cargas de trabajo. Se mejora los tiempos de recuperación ante desastres.
Inserción de servicios	Datacenter	Creación de cadenas dinámicas de servicios L4-7 en una base por usuario para implementar la selección de estos servicios en una modalidad de autoservicio o basado en políticas como respuesta a ciertos eventos.	Los tiempos de despliegue se reducen de días a minutos. La mayor agilidad y modalidad autoservicio generan nuevos ingresos y oportunidades de servicio con costos más bajos.

Monitoreo	Datacenter	Enviar el tráfico a analizar (potencialmente identificado por flujo) hacia los dispositivos de red / seguridad desde cualquier parte de la red.	Reduce la necesidad de cableado extra hacia los dispositivos de análisis.
Ancho de Banda bajo demanda.	Borde de la Empresa / Proveedor de Servicios	Permite de manera programática controlar los vínculos del proveedor para requerir ancho de banda extra cuando se necesite (en escenarios de Recuperación de Desastres / Respaldos)	Reduce los costos operativos permitiendo el autoservicio por parte de la empresa e incrementa la agilidad, ahorrando tiempo en la provisión del servicio.
Aplicaciones conscientes del ruteo	Datacenter / Campus	Algunas aplicaciones pueden tener necesidades especiales de ruteo / reserva de ancho de banda. Las aplicaciones pueden comunicarse con el controlador para indicarle sus requerimientos y el controlador SDN programar el flujo según estas necesidades.	Mejor utilización de los recursos de red / obtener caminos óptimos globales al conocer la topología completa.

Movilidad	Campus	Permite a la empresa desarrollar sus propias extensiones a las aplicaciones en ambientes de movilidad (pej entornos BYOD) sin tener que esperar desarrollos de los fabricantes.	El tráfico se envía directamente al AP más cercano al dispositivo móvil (por un camino óptimo) ya que el controlador conoce la ubicación del mismo y la topología de la red y no a través de un controlador para luego transmitirlo hacia el AP que corresponda.
Aplicaciones colaborativas / transmisión de video	Campus	Al conocer las fuentes / destinos y la topología de la red, el controlador SDN puede programar una topología multicast bajo demanda con una complejidad menor que las configuraciones multicast tradicionales.	Se utiliza una configuración con menor complejidad que la tradicional y altamente eficiente.
Seguridad y aplicación de políticas	Campus	El controlador SDN puede entender el contexto de un flujo y a partir de allí programar la red para aplicar políticas.	Permite aplicar las políticas de seguridad con una mayor granularidad.

Para poder implementar estos casos de uso se necesita los distintos componentes de una solución SDN. Una lista muy completa de productos de código abierto que soportan OpenFlow puede encontrarse en [100] y [99].

7.3 Conclusiones

El controlador SDN es un punto de control estratégico en la red, ya que es esencialmente su cerebro. El estado actual del mercado de los controladores SDN es tal que mientras los usuarios esperan comportamientos basados en estándares, los fabricantes utilizan técnicas y/o protocolos propietarios para crear mercados para sus productos. Esto se explica ya que las aplicaciones se encuentran altamente acopladas al controlador mediante el uso de APIs propietarias.

Los controladores SDN, vistos como un grupo tienen los siguientes atributos:

- Proveen distintos niveles de soporte para el desarrollo (lenguajes, herramientas, etc).
- Los productos comerciales tienen generalmente interfaces propietarias, pero ofrecen soluciones más robustas de almacenamiento y escalabilidad.
- Pocos controladores tienen soporte para más de un protocolo para la interacción con los dispositivos.
- Todas las soluciones de controladores disponen de APIs propietarias para interfaces de aplicación (interface hacia el norte – “northbound”), si bien algunos están trabajando en la estandarización de este protocolo como el proyecto Open Daylight. Desafortunadamente la ONF se ha resistido en trabajar para este fin hasta hace poco tiempo [73] y otras organizaciones como el IETF ([101], [102], [103], [68]) y ETSI ([104]) ya comenzaron a trabajar en esta área. El proyecto Open Daylight producirá código abierto que se espera sea de referencia y haga converger los distintos estándares.
- En este momento, los controladores que permiten escalabilidad mediante entornos con múltiples controladores, sincronización de bases de datos o estrategias de clúster. Estas estrategias dificultan la interoperabilidad entre los distintos fabricantes.

El soporte para OpenFlow 1.3 no es universal aún y muchos controladores y fabricantes solamente soportan OpenFlow 1.0, lo cual es crítico debido a la cantidad de mejoras existentes entre ambas versiones del protocolo. Algunos fabricantes de los dispositivos implementan extensiones propietarias al protocolo, lo cual es un obstáculo entre la interoperabilidad de las soluciones.

Sin una portabilidad real de las aplicaciones e interoperabilidad SDN no progresará y decaerá como una moda pasajera. Queda por ver si el enfoque de “framework” de los controladores y el proyecto Open Daylight creará un estándar “de facto” o si alguno de los caminos de estandarización como la IETF, ETSI, ITU, ONF van a aportar una solución o solamente agregar una mayor confusión a la existente hoy en día. Es de esperar que el proyecto Open Daylight, al tener un fuerte apoyo de los fabricantes, se convierta en una solución estándar de la industria y sea seguida por los organismos de estandarización. Lo que tienen las tecnología SDN a su favor es que normalmente surgen de la academia y luego son tomadas por los consorcios / fabricantes. Algo que queda pendiente hoy en día y donde SDN tiene un alto impacto es el área operativa. Se debe recordar que algunos años atrás, se trabajó duro en la optimización de las áreas operativas de nuevas tecnologías (en ese entonces) como ser MPLS.

Por último debemos preguntarnos cuál es la posición de la industria al respecto o cómo ve esta tecnología a futuro. Según IDC [105] realizó las siguientes predicciones sobre el mercado SDN:

- El tamaño del mercado será de U$ 3.7 Billones (1 billón = 1.000 millones) para 2016.
- SDN ocupará el 35% del mercado de "switching" en los Data Centers para 2016 (en 2012 casi ni participaba de este mercado).
- Los principales fabricantes de "networking" (Cisco, Juniper, Brocade, Citrix, HP, Dell, IBM) continuarán adquiriendo tecnología SDN (de otras empresas más chicas).

Dados estos números, parece lógico esperar que en los próximos 3 años, SDN tenga un crecimiento explosivo.

8. Virtualización de la red en el Data Center

8.1 Introducción

Los requerimientos para la disponibilidad de aplicaciones cambió el modo en que las aplicaciones son alojadas en los Datacenters hoy en día. Ha ocurrido una evolución en los distintos componentes como ser la virtualización de servidores, virtualización del almacenamiento y también la virtualización de redes.

Las mayores motivaciones para la virtualización de servidores estuvieron asociadas en una primera instancia con reducción de costos y redundancia, luego evolucionaron hacia escalabilidad y agilidad. Las tecnologías LAN de los Datacenters han tomado un camino análogo, primero con un objetivo de redundancia para luego crear una red densa más escalable (“fabric”) dentro y entre Datacenters.

Las nuevas necesidades de negocios han cambiado el foco de las redes de la próxima generación desde la redundancia (duplicación de sistemas) hacia la resiliencia que es la capacidad de adaptarse a las consecuencias de una falla. Los Datacenters de hoy en día deben estar alineados con ciertos objetivos de negocio y algunos obstáculos de diseño para alcanzar la resiliencia.

Objetivos de negocio:

- Mejorar la performance de las aplicaciones
- Cumplir con normas regulatorias
- Agilidad del negocio, desde el punto de vista de TI

Obstáculos de diseño:

- La densidad crece rápidamente.
    - Nuevas aplicaciones son desplegadas en nuevos servidores de manera continua.
    - La mejora en performance de los servidores resulta en un número mayor de máquinas virtuales por servidor.
    - El incremento en la cantidad de máquinas virtuales por servidor incrementa la cantidad de tráfico por servidor.
- Despliegue dinámico de aplicaciones y asignación de recursos.

En el presente capítulo se presentan las distintas alternativas de diseño de la red del Datacenter junto con las tecnologías que le dan soporte. Como caso particular se estudian las dos principales tecnologías para la generación de caminos redundantes (que también dan soporte a la virtualización de la red) en la red del Datacenter para luego realizar una comparación de ambas.

Como caso particular de virtualización se estudia la convergencia de la red de almacenamiento junto con sus tecnologías asociadas y las mejores prácticas para su aplicación.

Finalmente se provee como resumen, una tabla con las distintas tecnologías presentadas con su escenario de aplicación a modo de guía, presentando las tecnologías más relevantes.

La realización de este capítulo se basó principalmente en [6] y [106] junto con otras fuentes (estándares, artículos y manuales de los fabricantes de hardware y software, artículos de sitios especializados, etc).

8.2 Objetivos de diseño en una red de Data Center

Los objetivos de diseño de los Datacenters modernos se encuentran alineados con sus objetivos de negocios y de los requerimientos de las aplicaciones alojadas en los mismos. De todos los mencionados anteriormente, los objetivos de diseño incluyen:

- Rendimiento
- Escalabilidad y agilidad
- Flexibilidad para dar soporte a varios servicios
- Seguridad
- Redundancia / Alta disponibilidad
- Soporte para la administración
- Disminuir costos operativos (OPEX y CAPEX)
- Viabilidad a largo plazo

No existe una única solución que aplique a todos los casos, pero se intentará exponer una serie de guías y recomendaciones generales que satisfagan los requerimientos de diseño expuestos por la empresa. Las redes LAN de los Datacenters están en plena evolución, los requerimientos de negocio presionan a las organizaciones de TI para adoptar nuevos modelos de aplicaciones. La evolución del Datacenter desde un modelo de servidores centralizados a un modelo de nube privada se encuentra en camino y es potenciada por servicios de nube híbridos y públicos.
El tráfico de red en el Datacenter se está convirtiendo en menos cantidad de cliente-servidor y evolucionando a la forma de servidor-servidor por lo que nuevas topologías están emergiendo. Las redes de ayer que eran altamente segmentadas se están convirtiendo en redes con menor segmentación a nivel físico y mayor segmentación a nivel lógico. La segmentación virtual permite la reducción de equipamiento físico y por lo tanto un ahorro de costos importantes.

8.3 Facilitadores que impulsan los cambios

Existen varios factores que impulsan los cambios a nivel del Datacenter para evolucionar hacia los Datacenters de nueva generación.

- La necesidad de un nivel más alto de confiabilidad, minimizando el tiempo que los componentes no se encuentren disponibles debido a actualizaciones o cambios en la configuración. Una vez que se ha consolidado la arquitectura, es crítico que se mantenga disponible con la mínima disrupción posible.

- La necesidad de optimizar la infraestructura de red del Datacenter, moviéndose a una topología donde ningún vínculo se mantiene ocioso. Las topologías basadas en “Spanning Tree” son conocidas por ser ineficientes ya que bloquea uno de los caminos o por ser activo/pasivo cuando se realiza “teaming” de tarjetas de red en servidores. Esta necesidad se resuelve utilizando técnicas de “multipath” en L2, las cuales suelen ser propietarias de los fabricantes de equipos de comunicaciones (anteriormente vimos VPC, una solución de Cisco para cubrir esta necesidad).

- La necesidad de optimizar los recursos de cómputo, reduciendo la tasa de crecimiento de servidores físicos. Esta necesidad se resuelve mediante la virtualización de servidores.

- La necesidad de reducir el tiempo que llevar provisionar un nuevo servidor. Esto se resuelve mediante la habilidad de configurar perfiles de servidores y luego impactarlos en la infraestructura física o virtual.

- La necesidad de reducir el consumo de energía en el Datacenter. Esta necesidad se resuelve con varias tecnologías, como ser la virtualización y “hardware” energéticamente más eficiente.

- La posibilidad de mantener la segmentación de la empresa dentro del Datacenter (suponiendo sea un sitio remoto)

Como consecuencia de las necesidades listadas anteriormente, se desarrollan las siguientes capacidades:

- Arquitecturas capaces de soportar la LAN y la SAN sobre la misma red (para reducir energía y consolidar servidores)

- Arquitecturas capaces de distribuir tráfico L2 en todos los vínculos disponibles.

- Cableado simplificado: Para flujos de aire más eficientes, menor consumo de energía y menos costo de implementación de redes de alta velocidad.

8.4 Conectividad y topología

8.4.1 Diseño en 2 capas

Los diseños en dos capas son muy populares para las redes de los Datacenters. Los “switches” de acceso para para dar conectividad a los servidores se conectan en los “switches” de agregación de alta velocidad, los cuales proveen funciones de “switching” y “routing” para varias “VLANs” de servidores. Este diseño que se ejemplifica en la Figura 70 tiene varias ventajas:

- Diseño simple (menos “switches” significa menos elementos que administrar).
- Se reduce la latencia de la red ya que hay menos saltos en los “switches”.
- Bajo consumo de energía

Sin embargo la desventaja de un diseño en dos capas es su escalabilidad limitada. Cuando los puertos en un “switch” de agregación se encuentran completamente ocupados, agregar un nuevo par de “switches” de agregación es complejo. La conexión entre los pares de “switches” de agregación debe ser de la forma de un “full mesh” con alto ancho de banda de manera de no introducir cuellos de botella en el diseño de la red. Ya que en los “switches” de agregación se ejecutan protocolos de ruteo, más “switches” significa más vecinos en los protocolos de ruteo, más interfaces de ruteo y por lo tanto mayor complejidad que es introducida por el diseño tipo “full mesh”.

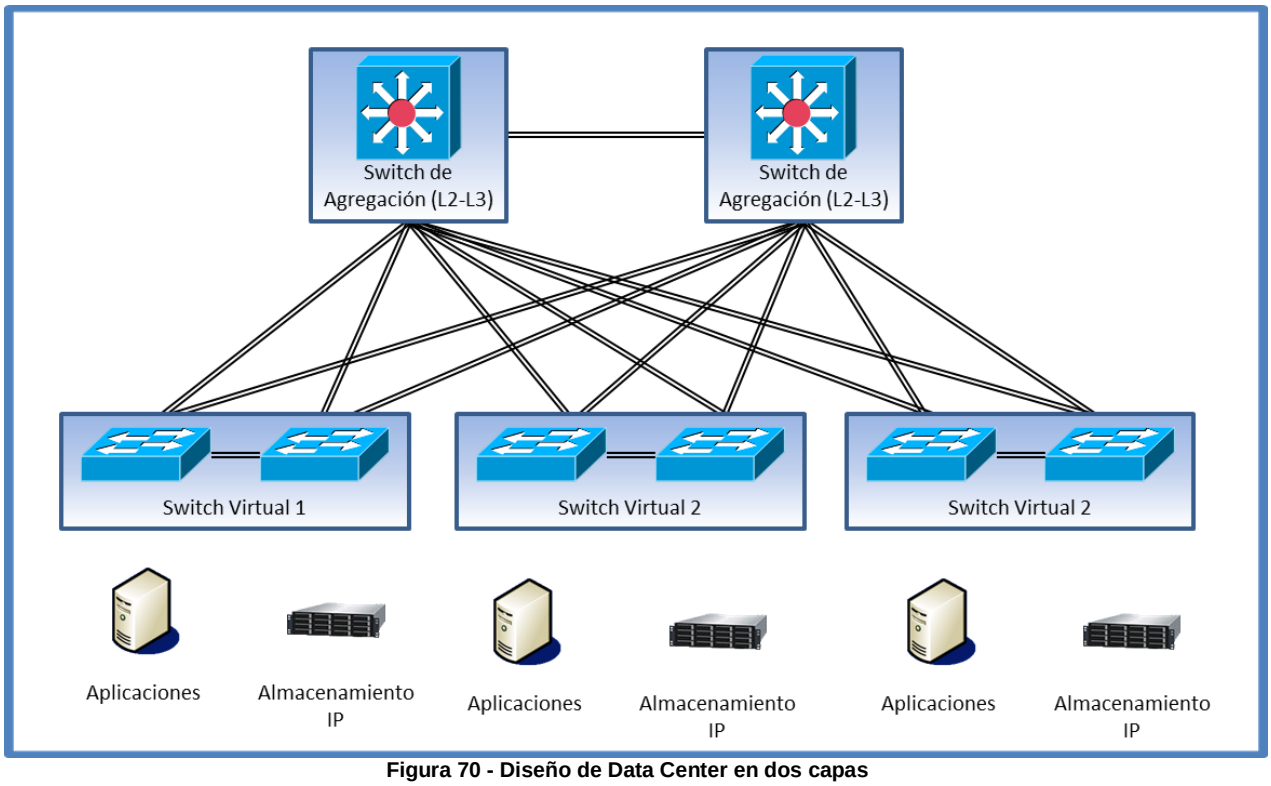

Figura 70 - Diseño de Data Center en dos capas

8.4.2 Diseño en 3 capas

El diseño de las redes de Datacenters en tres capas se compone de “switches” de acceso conectado a los servidores, “switches” de agregación para los “switches” de acceso y “switches” de “core” de Datacenter que proveen ruteo desde y hacia la red de la empresa. El diseño basado en tres capas, se basa en un diseño jerárquico, por lo que su principal ventaja es la escalabilidad. Se pueden agregar pares de “switches” de agregación si necesidad de modificar los pares de agregación existentes. Debido a la utilización de ruteo en los “switches” de “core”, no se necesita utilizar un esquema “full mesh”. La desventaja de un diseño en tres capas es la de una latencia elevada debido a una capa adicional, sobresuscripción adicional en el diseño (a menos que el ancho de banda se agregue en cantidades importantes), más dispositivos (y con una complejidad mayor) para configurar y operar, mayor consumo de energía y utilización de espacio de rack. En la Figura 71, podemos encontrar un ejemplo de este tipo de diseño.

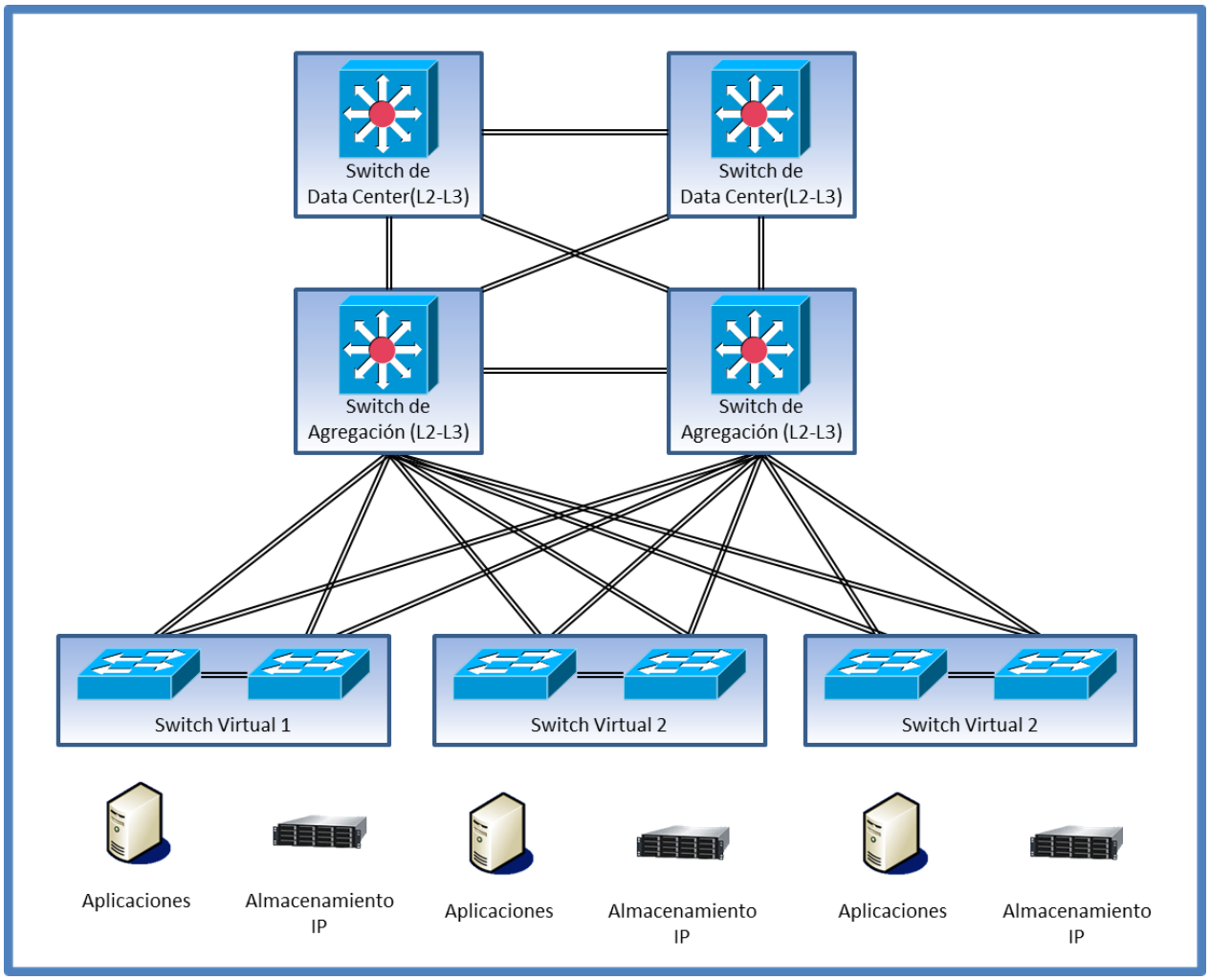

Figura 71 - Diseño de Data Center en tres capas

8.4.3 Top of Rack (TOR)

Es muy común encontrar diseños TOR en los Datacenters de hoy. Su diseño modular hace que la implementación de los mismos sea más sencilla y que el cableado se perciba como más sencillo, especialmente cuando existe una alta densidad de servidores con adaptadores Gigabit Ethernet. En este tipo de diseños, se instala uno o más “switches” en un “rack” para atender a los servidores alojados allí y luego se interconectan con la capa de agregación. Un ejemplo de este diseño puede apreciarse en la Figura 77.

A su vez, TOR tiene algunas desventajas como ser:

- El número de servidores en al rack cambia a lo largo del tiempo, por lo tanto varía la cantidad de puertos en el “switch” a ser provistos.
- El número de puertos no usados va a ser más alto que en el modelo “End of Row” (EOR). Esto resulta en un mayor consumo de energía, y mayor necesidad de refrigeración.
- Mejoras en la tecnología (pasaje de velocidades de 1G a 10G o “uplinks” de 40G) tiene como resultado el reemplazo del “switch” TOR.

- TOR agrega problemas de escalabilidad, específicamente en la congestión de los enlaces de “uplink”. En el modelo EOR, esto se logra agregando distintas placas a un “chassis” modular.

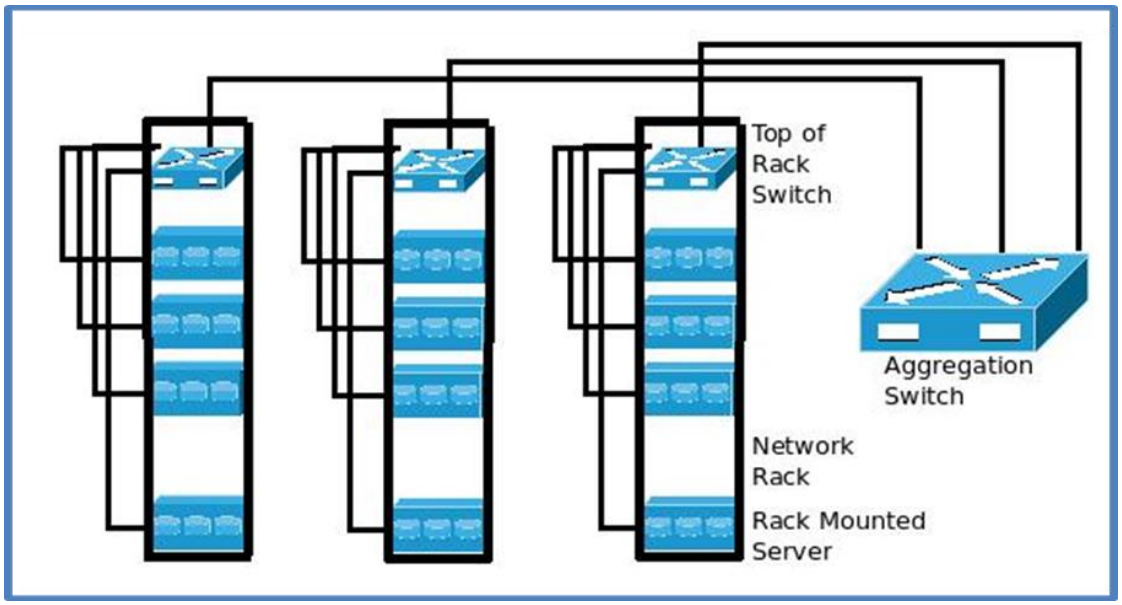

Figura 72 – Diseño TOR [106]

8.4.4 End of Rack (EOR)

Otra posible topología de Datacenter es la opción End of Rack (EOR), la cual se basa en un “switch” basado en “chassis” para la conectividad de los servidores. Este diseño coloca este “switch” al final o en medio de la fila de “racks” y conectar todos los servidores en una fila de “racks” contra dicho “switch”, como se puede apreciar en la Figura 73.

Comparado con el diseño TOR, los servidores pueden ser ubicados en cualquier lugar de los racks, por lo que se pueden evitar áreas de gran concentración de servidores y por ende de calor. En el uso de los equipos en modalidad EOR, se optimiza en espacio de “rack”, consumo de energía y enfriamiento. El número de “switches” que se utilizan es menor y trae consigo las ventajas de un diseño escalable y de alta disponibilidad. Los típicos “switches” de chasis proveen más funcionalidades y escalan mejor en el modelo EOR comparado con equipamiento de menor porte, típicos de los diseños TOR. Por otro lado, el cableado es más complejo a medida que la densidad aumenta en el “rack” EOR.

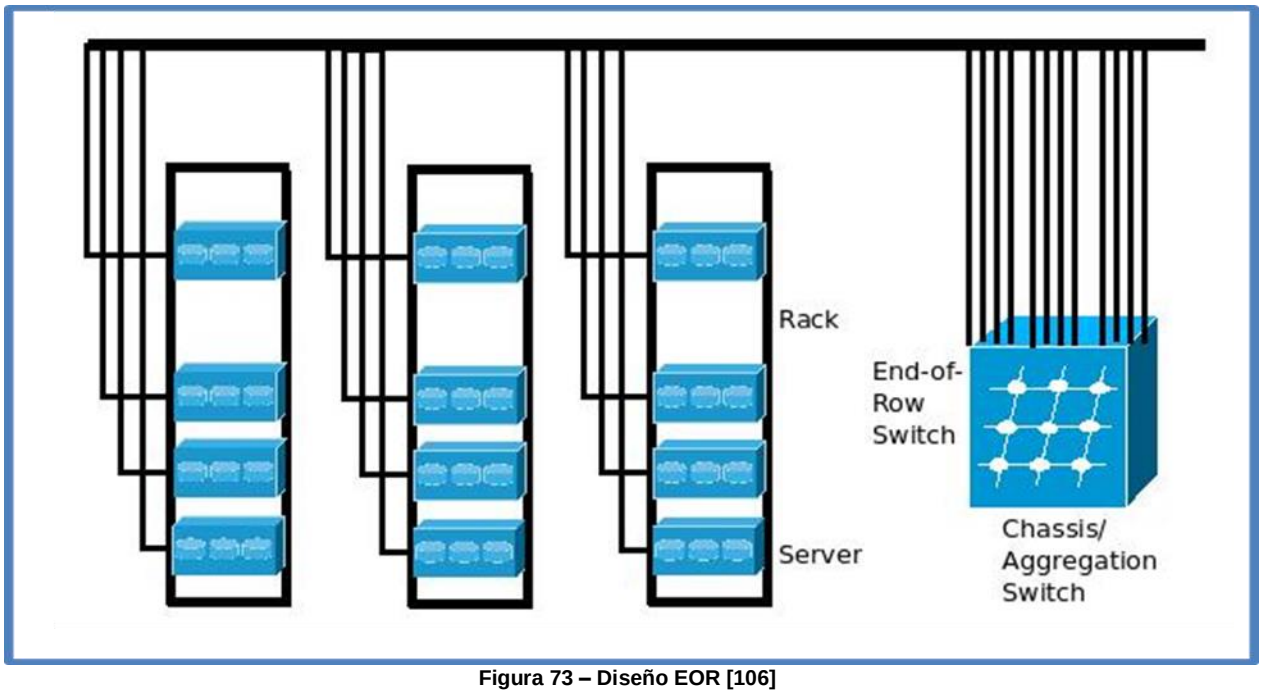

Figura 73 – Diseño EOR [106]

8.4.5 Resiliencia de la red LAN

La virtualización de servidores cambió los requerimientos de cómo se conectan los servidores a la red. Sin importar la topología física de conexión y el "hypervisor" utilizado, existe un conjunto de requerimientos básicos que los sistemas requieren de la red. A la vez que la consolidación crece, también lo hace la necesidad de resiliencia.

La conectividad de servidores, tiene varios requerimientos:

- Se debe contar con conexiones redundantes.
- Se debe armar un esquema de balanceo de carga.
- El despliegue de la conectividad debe ser altamente automatizada.

La agregación de vínculos se representa por el estándar IEEE 802.3ad, el cual define balanceo de carga y redundancia en la interconexión de dos nodos utilizando un número arbitrario de vínculos. Existen soluciones desarrolladas por los fabricantes de interfaces de red (NIC) para prevenir puntos únicos de fallo basándose en el desarrollo de manejadores ("device drivers") especiales que permiten a dos interfaces de red estar conectadas a dos "switches" distintos o a dos puertos del mismo "switch". Si una interface falla, la interface secundaria asume la dirección IP del servidor y toma la operación sin disrupción en la conectividad. Los distintos tipos de soluciones incluyen activo/pasivo y activo/activo. Para poder implementar estas soluciones se requiere que las interfaces dispongan de adyacencia L2 entre ellas.

8.4.6 Switches virtuales (capa de acceso)

Como "switches" virtuales se entienden tecnologías que nos presentan dos "switches" físicos como uno solo. Estas tecnologías suelen ser propietarias de los fabricantes y de las que se han mencionado anteriormente se destacan VSS y VPC de Cisco. Estas tecnologías proveen redundancia en la capa de acceso de servidores y también se encuentran en la capa de agregación. El "switch" virtual es una entidad lógica que dispone un único punto de administración (se pueden encontrar soluciones con varios puntos de administración y herramientas de sincronización de configuraciones) que da soporte a conexiones utilizando los protocolos de agregación definidos en IEEE 802.3ad. En entornos de virtualización, la asignación de servidores virtuales a vínculos agregados, provee mejor performance y reduce la

necesidad de configuraciones de red especializadas en el hypervisor.

Los “switches” virtuales deben ofrecer las siguientes características:

- Agregación de vínculos automatizada entre “switches” físicos.
- Envío de tráfico de aplicación de manera continua.
- Automatización de perfiles de seguridad o red por servidor virtualizado, por puerto, etc.
- Soporte de miles de equipos por sistema

Se debe estar consciente que la virtualización de “switches” puede reducir la disponibilidad, cuando ocurren errores de configuración por parte del administrador de red. Debido a que el “switch” virtual funciona como un único dispositivo, un error de este tipo tiene la potencialidad de disminuir la disponibilidad de la solución. Una posible mitigación de este tipo de problemas es la utilización de protocolos de “Spanning Tree” como un mecanismo de contingencia.

Existen protocolos como IEEE Shortest Path Bridging (SPB) o el IETF “Transparent Interconnection of Lots of Links” (TRILL) (aún en fase de estandarización), los cuales permiten la utilización de “switches independientes en el “core” del Datacenter

8.4.7 Diseño de core L2

La resiliencia de la red se requiere a lo largo de la red, y no solo en una capa en particular si es que se deben alcanzar las necesidades de las aplicaciones de negocio. Los diseños de L2 de hoy en día derivan la resiliencia generalmente basándose en los protocolos estándares de la industria (también existen soluciones propietarias como vimos anteriormente), los cuales incluyen los protocolos IEEE 802.1s [107] “Multiple Spanning Tree Protocol” (MSTP) y IEEE 802.1w [108] “Rapid Spanning Tree Protocol” (RSTP). Estos protocolos no solo proveen interoperabilidad con el equipamiento, sino que debido a su madurez proveen al administrador un gran número de herramientas para implementar y mantener la topología de la red.

Existen estándares emergentes que aumentan la resiliencia de las redes de los Datacenters de la nueva generación, los cuales incluyen:

- “Shortest Path Bridging” (SPB), el cual se encuentra en desarrollo por el grupo de trabajo IEEE 802.1aq. La definición de este protocolo se basa en el estándar de la IEEE 802.1aq junto con el RFC 6329 [109] que propone extensiones al protocolo de ruteo IS-IS para dar soporte a SPB.
- “Transparent Interconnect of Lots of Links”, el cual se encuentra basado en el RFC 5556 [110] y el RFC 6326 [111] que propone el uso de IS-IS para dar soporte a TRILL. A su vez existen una serie de RFCs (RFC 6325 [112], RFC 6847 [113], etc) los cuales definen temas adicionales como el transporte de ciertos protocolos sobre TRILL.

Estos nuevos estándares van a mejorar la resiliencia de las futuras redes ya que:

- Tienen la posibilidad de utilizar toda la conectividad física disponible.
- Permiten una rápida recuperación de la conectividad ente fallos.
- Restringen los fallos de tal manera que solo el tráfico directamente afectado se impacta durante la recuperación; el resto del tráfico continúa fluyendo sin ser afectado.

8.4.8 Diseño de core L3

Las redes de “core” L3 hacen hincapié en dos principios fundamentales como ser la disponibilidad de las rutas y la disponibilidad del “gateway”. Los protocolos estándar OSPF y VRRP (también existen protocolos propietarios) proveen a las redes L3 de esta capacidad.

8.4.8.1.1 Balanceo de carga y disponibilidad (OSPF, VRRP)

OSPF permite a las redes L3 negociar caminos L3 para poder balancear la conectividad mediante distintos caminos en la red. Esto permite a los diseñadores de redes asegurarse que los caminos son utilizados y no solo utilizados como respaldo. A su vez OSPF posee posibilidades de ingeniería de tráfico [114] para asegurar que las aplicaciones críticas disponen del ancho de banda y caminos necesarios para su correcto funcionamiento. Combinando OSPF con el mecanismo de redundancia de "Gateway" provisto por VRRP se provee una resiliencia alta con una madurez alta en las herramientas disponibles para el administrador.

8.4.8.1.2 Separación de redes (MPLS, VRF)

VRF permite que existan múltiples instancias de ruteo en un único router, separa completamente clientes o departamentos basado en dominios de ruteo, permitiendo la asignación de recursos dedicados de ruteo para aplicaciones críticas. Provee una solución simple para la LAN de campus y las aplicaciones en el Datacenter. Es la extensión y el precursor de los servicios de VPNs MPLS dados por un proveedor, sin exponerlo a la complejidad de MPLS. Con las soluciones vistas anteriormente es posible transportar la segmentación dentro de la empresa hacia el Datacenter, incluso cuando éste es compartido por varias organizaciones (ver Figura 74).

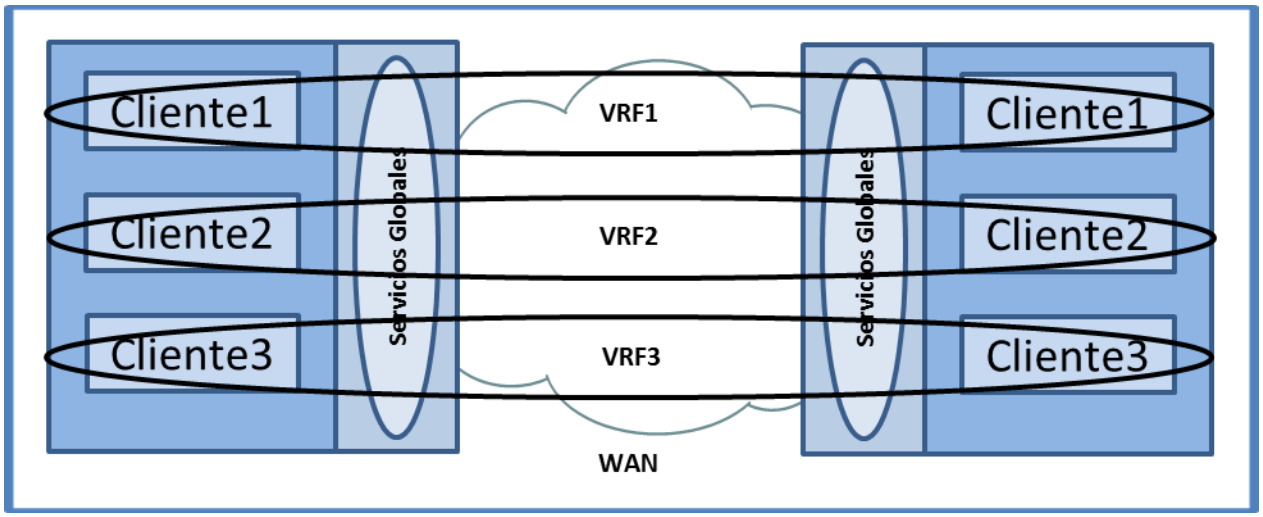

Figura 74 - Diseño de interconexión de DataCenters con VRFs

8.4.9 Nuevas tendencias

8.4.9.1 Transparent Interconnection of Lots of Links (TRILL)

TRILL es una tecnología L2 multicamino ("multipath") definida en los RFCs 5556, 6325, 6327, 6326, 6439 del IETF [115], [116], [117], [113], [118], [119], [120] y otros documentos borradores (drafts) [121] en espera de aprobación. TRILL es el sucesor de "Spanning Tree" ya que se diseñó para superar sus deficiencias. El proceso de estandarización de TRILL ha llevado más de lo esperado para que la industria de "networking" pueda avanzar con su implementación en dispositivos de red.

TRILL se puede describir como ruteo de L2. El ruteo se piensa comúnmente como en términos de L3, donde el tráfico que tiene como destino una red remota se envía por una interface

particular, indicada por el cómputo del mejor camino. TRILL toma la idea del ruteo y la aplica a direcciones MAC de Ethernet, incluso agregando un campo con información de tiempo de vida (Time To Live – TTL).

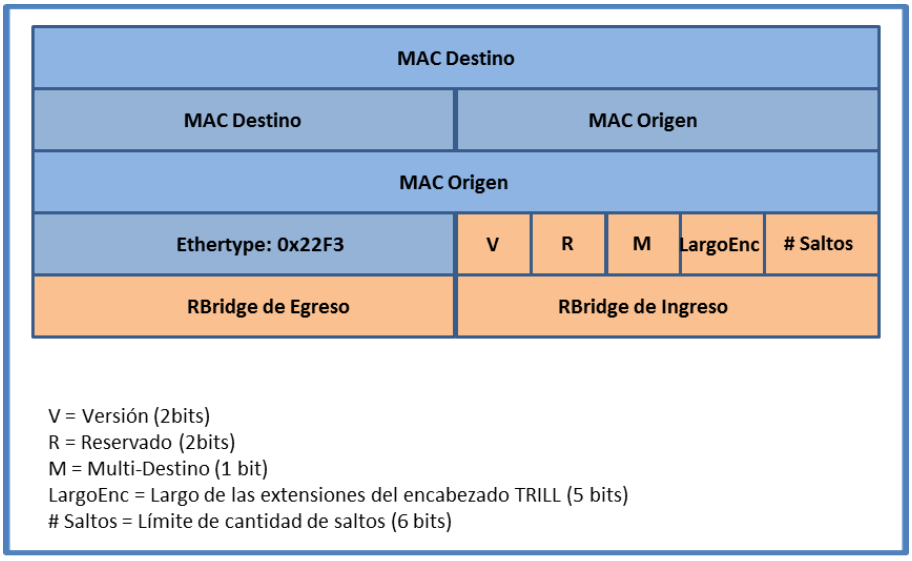

Figura 75 - Formato del paquete de TRILL [122]

A los dispositivos que soportan TRILL se los denomina RBridges (“Routing Bridges”). Estos dispositivos deben calcular el mejor camino hacia direcciones MAC remotas utilizando el protocolo de ruteo IS-IS. Cuando un RBridge recibe un paquete Ethernet, determina que dispositivo TRILL debe recibir ese paquete, lo encapsula (con el formato descrito en la Figura 75) con el destino del RBridge que corresponde y lo envía. Si existen otros RBridges en la topología entre los RBridges de ingreso y egreso, estos envían el paquete hacia el RBridge de egreso. Cada salto TRILL decrementa el campo TTL que es una característica que Ethernet no posee y es la razón principal por la cual una red Ethernet con un bucle (“loop”) colapsa rápidamente. Cuando el RBridge de egreso recibe el paquete, desencapsula el paquete Ethernet y lo envía hacia su destino.

Por el momento, las implementaciones de los distintos fabricantes difieren de alguna manera del estándar, lo que les da a los distintos fabricantes su distintivo.

- La implementación de TRILL por parte de Brocade es un componente crucial de su solución de clúster virtual.

- Cisco describe a FabricPath (disponible en la línea Nexus 5000 y 7000) como un derivado de TRILL. Cisco se encuentra involucrado de manera activa en la estandarización de TRILL y es posible que algunas de las características de FabricPath se conviertan en características de TRILL.

Al ser una tecnología nueva, todavía no existe interoperabilidad completa entre los distintos fabricantes, por lo que si bien es posible realizar implementaciones de TRILL en el Datacenter, seguramente se traducirá en una relación de larga duración con el fabricante que se elija.

8.4.9.2 Shortest Path Bridging 802.1aq (SPB)

SPB es una tecnología similar a TRILL basada en el estándar 802.1aq del IEEE [123], la cual fue ratificada en marzo de 2012.

SPB al igual que TRILL es una tecnología L2 multicamino cuyo objetivo es reemplazar a "Spanning Tree".

Si bien conceptualmente SPB es igual a TRILL (que también utiliza IS-IS para el cálculo del camino óptimo), la manera en que SPB envía los paquetes difiere ya que SPB aplica una etiqueta (VID) al paquete cuando éste ingresa al dominio (o la traduce a otra etiqueta si ya existe una) y la quita (o la vuelve a traducir) cuando el paquete sale del dominio. La otra variante de SPB es "Shortest Path Bridging MAC" (SPBM), donde se usa el encapsulado MAC-en-MAC. Cuando un paquete ingresa al dominio, el dispositivo SPB de ingreso determina el dispositivo de egreso. El paquete original se encapsula en otro paquete Ethernet con la MAC de destino del dispositivo SPB de egreso, donde el paquete original es desencapsulado y se envía a su destino final [122].

SPB ofrece características que se ajustan bien a redes de varios clientes independientes o a redes de proveedores de servicios y los fabricantes alineados con SPB comparten esta orientación.

Alcatel, Avaya y Huawei son fabricantes que ofrecen soluciones SPB. En estas tecnologías se nota que los fabricantes eligieron una o la otra, si bien HP anunció su deseo de dar soporte a ambas [124].

8.4.9.3 Comparativa entre SPB y TRILL

En la siguiente tabla, se resumen las características más importantes de cada una de estas tecnologías, realizando una comparación entre ellas. Por lo general, es de esperar encontrar a TRILL en los Datacenters de las empresas y a SPB en los proveedores de servicios.

	General	**Cantidad de VLANs soportadas**	**Tamaño de paquete**	**Complejidad del cálculo de rutas** **N = Nro de switches** **k= Nro de multicaminos**	**Vecindad**	**Spanning Tree**
TRILL	Protocolo del IETF con base en el RFC 6325 Aprobado 15/03/2010	4K VLANs	+20 bytes para Ethernet +8 bytes para PPP	Para paq Unicast O(NxLog(N)) Para paq multi-destino O(kxNxlog(N))	Forma vecindad a traves de cualquier switch	Bloquea SPT en cada dispositivo
SPB	Protocolo de la IEEE 802.1aq Proyecto del WG 802.1 Aprobado en 03/2012	4K VLANs	+22 bytes para Ethernet +24 bytes para Eth sobre PPP No soporta PPP nativo	Unificado: O(kxN^2xlog(N))	Forma vecindad solo con switches SPB vecinos	Mantiene SPT. Los paquetes son enviados por SPT o mediante el camino más corto, dependiendo de la VLAN

8.4.10 Convergencia de almacenamiento.

La conectividad de almacenamiento ("storage") hoy en día es una mezcla de FibreChannel (FC), iSCSI y NFS (iSCSI y NFS son basados en IP). Si se realiza un despliegue de FC, se necesita dos conjuntos distintos de "hardware", cables, herramientas y conocimientos. La conectividad del almacenamiento en el futuro estará basada en una única red convergente (con una etapa intermedia de una única interface convergente entre el servidor y el "switch" de acceso) con nuevos protocolos y hardware. Este cambio redundará en menos adaptadores, cables y nodos, resultando en una operación más eficiente de la red.

8.4.10.1 Data Center Bridging (DCB)

Las redes Ethernet 10Gb se están convirtiendo en la opción más común de acceso para los servidores. Solamente como una innovación “incremental”, permite la agregación de interfaces Gigabit Ethernet. En 2011, la IEEE ratificó y publicó varios mejoras al protocolo que colectivamente se conocen como “Data Center Bridging” ([125] [126] [127]), que entre otras innovaciones dotaron a 10 Gb Ethernet la capacidad de emular el comportamiento de “Fibre Channel”, dando como resultado las bases para “Fibre Channel over Ethernet (FCoE). Definida en el estándar T11 FC-BB-5 del “International Committee for Information Technology Standards” (INCITS) en 2009, esta tecnología permite la convergencia de las redes Fibre Channel (fabrics) y las redes Ethernet, como se muestra en el ejemplo de la Figura 76.

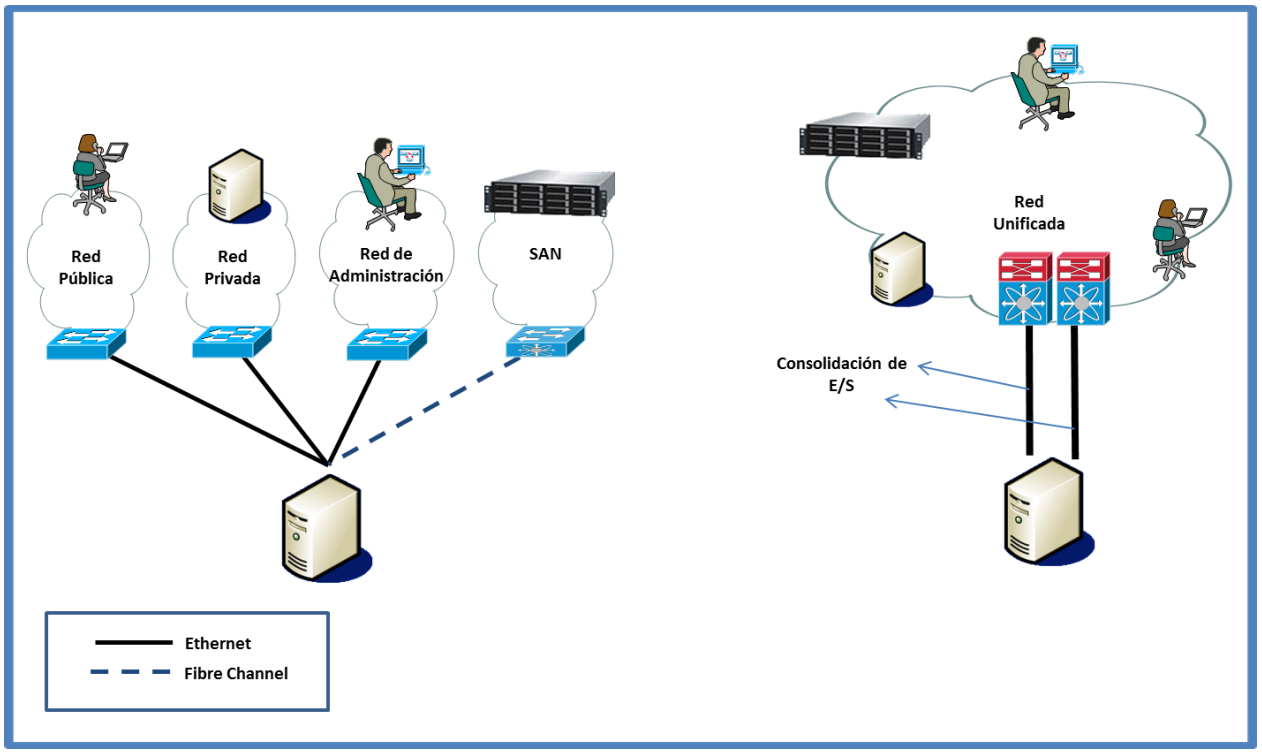

Figura 76 – Consolidación de E/S

Para convertirse en el transporte común a varias tecnologías en los Datacenters, Ethernet debe presentar las mismas características de “no descarte” ante cargas de tráfico seleccionadas. Como resultado de una serie de innovaciones, el IEEE publicó un grupo de estándares que definen como una red Ethernet puede soportar distintos patrones de tráfico, incluyendo algunos altamente sensitivos a las pérdidas como ser “Fibre Channel”. Conocidos como “Data Center Bridging” (DCB), estos estándares son mejoras que permiten transporte sin pérdida, asignación de recursos, configuración automática de dispositivos y notificación de congestión sobre redes 10 Gigabit Ethernet “full-duplex”. Como lo dice su nombre, DCB fue diseñado para entornos de Datacenters que se caracterizan por “switches” de alta velocidad con baja latencia y pocos saltos entre el origen y el destino.

8.4.10.2 Priority-Based Flow Control (PFC)

El protocolo IEEE 802.1Qbb [126] (Priority-based Flow Control) (PFC) permite el control de flujo por clase de tráfico en vínculos Ethernet “full-duplex”, con cada clase identificada por su prioridad basada en la VLAN. En resumen, intenta eliminar la pérdida de “frames” causada por contención de la misma manera que lo realiza la norma IEEE 802.3x PAUSE, pero sobre

prioridades individuales. En una interface con PFC, un “frame” de una prioridad sin pérdida no estará disponible para su transmisión si esta prioridad se encuentra en pausa en dicho puerto, como se ejemplifica en la Figura 77.

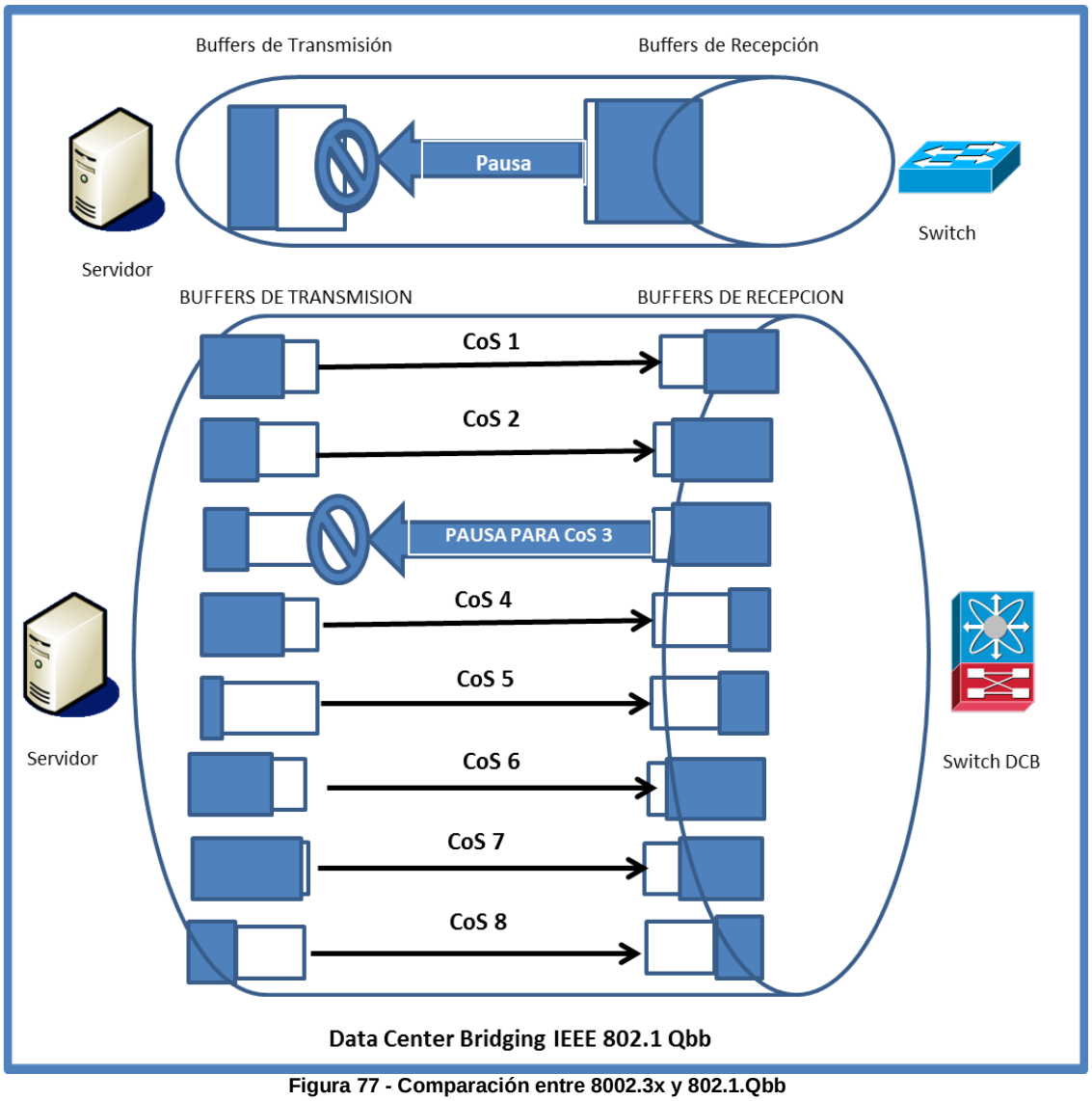

Figura 77 - Comparación entre 8002.3x y 802.1.Qbb

Contando con PFC, es posible transportar un protocolo sensible como Fibre Channel, previniendo que se descarte un “frame” encapsulado debido a falta de recursos.

8.4.10.3 Enhaced Transmission Selection (ETS)

Considerando que el estándar 802.1p permite la creación de ocho clases de servicio (o cables virtuales) en cada vínculo Ethernet, se utilizó otra extensión de Ethernet para definir la asignación de recursos para cada uno de estos “cables”.

El estándar IEEE 802.1Qaz (“Enhaced Transmission Selection” – ETS) [125] controla la asignación de ancho de banda entre las distintas clases de servicio. En una conexión con ETS habilitado, cuando una clase de tráfico no está utilizando el tráfico asignado, ETS permite que otras clases de tráfico utilizar el ancho de banda disponible. De acuerdo al estándar 802.Qaz, un “bridge” ETS debe soportar al menos tres clases de tráfico (una con PFC activado, una sin PFC y una con prioridad estricta), un porcentaje de granularidad de 1% y una asignación con

una precisión mínima de 10%.
A pesar que ETS determina que el ancho de banda se debe dividir entre clases de servicio, el método de encolamiento o el algoritmo de asignación de ancho de banda no están definidos en el estándar.

8.4.10.4 Datacenter Bridging eXchange Protocol (DCBX)

Desde el momento que PFC y ETS se encuentran implementados en ambos extremos de la conexión, los equipos conectados allí, deben estar configurados de acuerdo a las políticas impuestas por DCB. Si bien esto trae beneficios para la red del Datacenter, también se convierte en una gran tarea de configurar potencialmente miles de dispositivos. Afortunadamente todas estas operaciones se pueden evitar utilizando “Data Center Bridging eXchange Protocol” (DCBX) [125]. Este protocolo se utiliza en los dispositivos que soportan DCB para:

- Descubrir las capacidades del puerto vecino.
- Intercambiar configuración con los vecinos directamente conectados.
- Detectar parámetros de configuración erróneos.

DCBX se define en el estándar 802.1Qaz y se utiliza para aplicar ETS, PFC y las políticas de prioridad en los dispositivos que soportan DCB. DCBX utiliza “Link Layer Discovery Protocol” (LLDP) [128], el cual es un estándar para intercambiar atributos entre dos vecinos interconectados por un vínculo.

Los atributos de DCBX se transmiten mediante los campos de LLDP (“Type-Length-Values” – TLV) y pueden ser del tipo:

- **Informativos**: si no hay negociación o participación de la máquina de estados de DBCX.
- **Asimétricos:** si la configuración que se envía al vecino remoto, no debe coincidir con la configuración local.
- **Simétrico:** si la configuración debe coincidir en los puertos de ambos vecinos.

8.4.10.5 Quantized Congestion Notification (QCN)

Con el estándar IEEE 802.1Qau [127], se agregó a la iniciativa DCB un método para la notificación de congestión “end-to-end” en las redes de Datacenter, denominado “Quantized Congestion Notification” (QCN).

Un bridge L2 puede señalizar una situación de congestión mediante el mensaje “congestion notification message” (CNM) destinado al origen de los “frames” Este mensaje contiene información acerca de la congestión en el “switch” que implementa QCN (denominado como “Congestion Point” o CP). Utilizando los parámetros de cogestión transportado en el mensaje recibido, un limitador de tráfico (“rate limiter” o RL) asociado con una dirección de origen de “frames” disminuye la tasa de envío. Luego, el RL incrementa la tasa de envío de manera unilateral para recuperar el ancho de banda perdido y explorar si existe ancho de banda extra disponible. En la Figura 78, se da un ejemplo de esta situación.

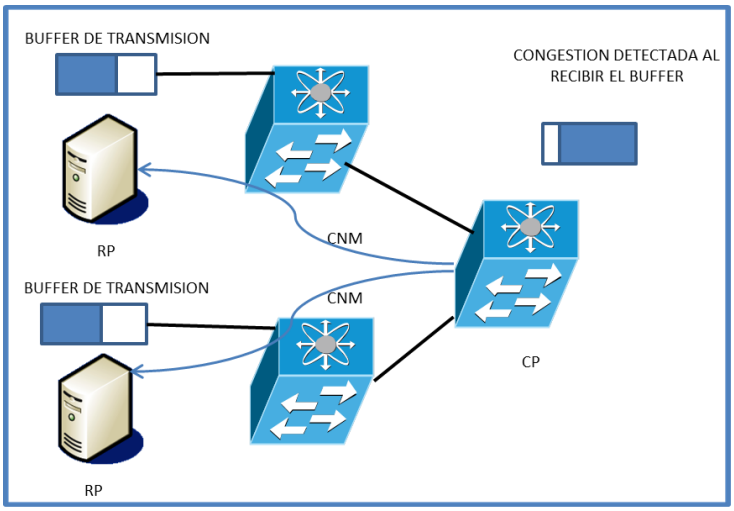

Figura 78 - Quantified Congestion Notification

8.4.10.6 Fibre Channel over Ethernet (FCoE)

En paralelo con el desarrollo de las mejoras DCB a Ethernet, se fue desarrollando la primera aplicación para un cable virtual PFC sin pérdidas. “Fibre Channel over Ethernet” (FCoE o FC-BB_E) es un protocolo que se definió en 2009 en el estándar INCITS T11 FC-BB-5. FCoE permite el transporte de “Fibre Channel” (FC) sobre una red Ethernet sin pérdidas, utilizando las técnicas abordadas en DCB. FCoE encapsula los “frames” Fibre Channel (Clase 2,3 o F [106]) sobre “frames” de Ethernet apropiados con el valor del campo “Ethertype” particular (0x8906), como se puede observar en la Figura 79.

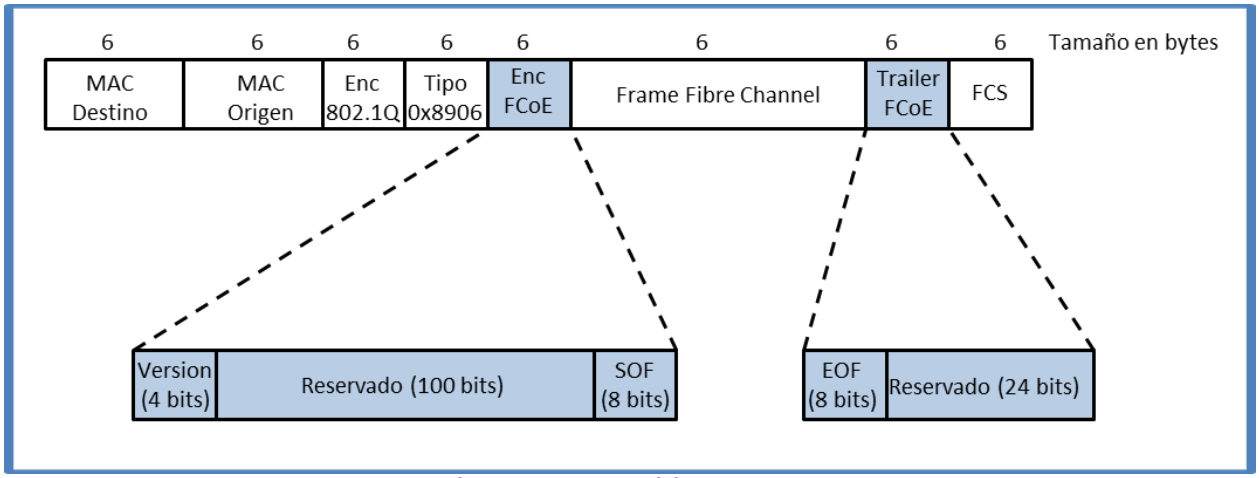

Figura 79 - Formato del Frame FCoE

De acuerdo con el formato, los “frames” FCoE, deben soportar un MTU de 2140 bytes para evitar la fragmentación

Desde el punto de vista de Fibre Channel, FCoE funciona como otro tipo de medio que puede transportar “frames” de Fibre Channel.

Los servidores pueden conectarse a FCoE mediante adaptadores convergentes (Converged Network Adapters – CNA) que contiene la funcionalidad de un HBA de “Fibre Channel” y una NIC Ethernet en la misma tarjeta. El encapsulado FCoE puede realizarse mediante software en tarjetas Ethernet convencionales, pero las CNAs FCoE quitan trabajo al procesador (CPU) realizando el procesamiento de bajo nivel del protocolo de paquetes y funciones del protocolo

SCSI que es realizado de manera tradicional por los HBAs.

Generalizando, una CNA combina las características de una tarjeta de red Ehernet (NIC) y un adaptador “Fibre Channel” (HBA) en un único adaptador físico.

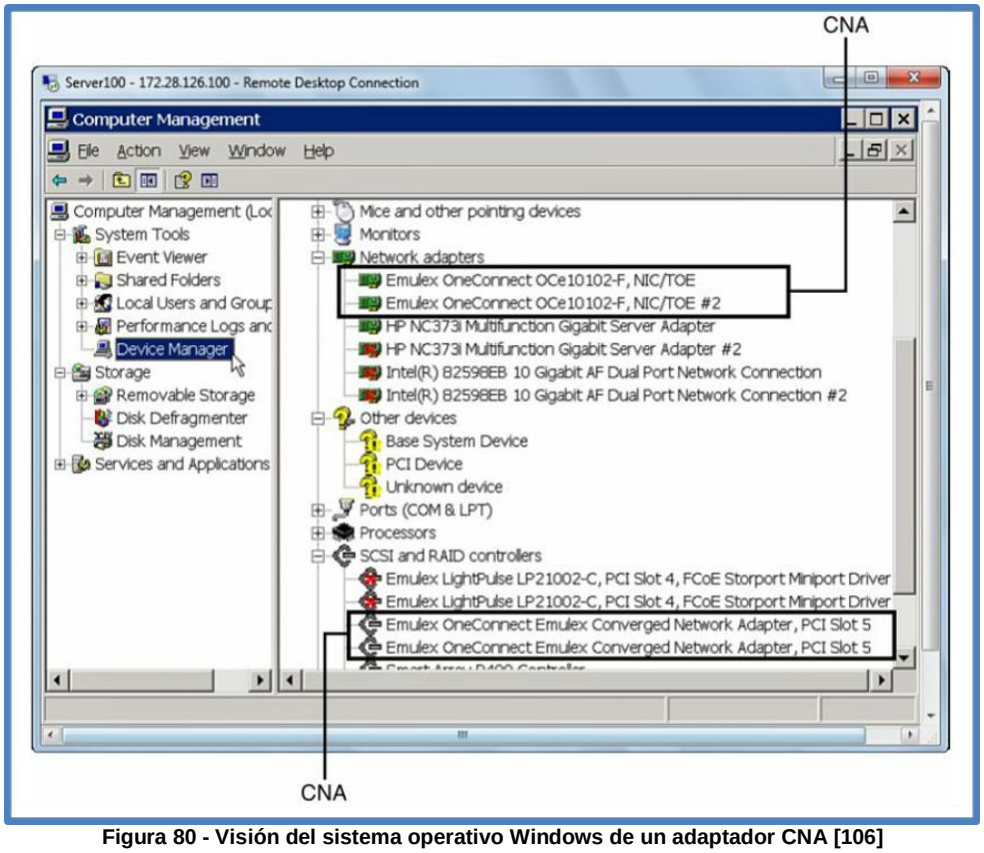

Figura 80 - Visión del sistema operativo Windows de un adaptador CNA [106]

En el ejemplo de la Figura 80 se observa que el sistema operativo (en este caso Windows) detecta dos adaptadores de red estándar y dos adaptadores “Fibre Channel”, correspondientes a un único adaptador CNA dual (con dos puertos).

Esta visión tiene la ventaja para los administradores/operadores del servicio que siguen manejando ambas redes (“Fibre Channel” y Ethernet) con las mismas herramientas y técnicas que utilizaban en las redes no convergentes, lo cual redunda en aprovechar todos los procesos y conocimientos existentes en la empresa con su consiguiente ahorro comparado con el entrenarse y crear nuevos procesos en tecnologías desconocidas.

8.4.11 Interconexión entre Datacenters

Los patrones de tráfico de clústeres, servidores y de las soluciones de virtualización de almacenamiento, requieren nuevos esquemas de redundancia. Estos esquemas proveen la tecnología de transporte utilizada para la conectividad entre Datacenters y se convierten en críticas a medida que la el diseño de la red evoluciona para proveer altos niveles de estabilidad, resiliencia y rendimiento.

La tecnología de transporte que se elige entre Datacenters depende de ciertos requerimientos. Los más importantes son:

- La tolerancia de las aplicaciones virtualizadas y el almacenamiento ante “jitter” y

latencia.
- La tolerancia ante “jitter” y latencia de las soluciones de clúster.
- El ancho de banda disponible por clase de tráfico.
- Si la interconexión se realiza en L2 o L3.
- Si se requiere cifrado del tráfico o no.

8.4.11.1 Desafíos de la interconexión de Datacenters

La manera más confiable para la interconexión de Datacenters para proveer alta disponibilidad es mediante la utilización de tráfico IP ruteado entre los mismos. Sin embargo se ha vuelto muy común extender la red Ethernet sobre la WAN, debido al uso de aplicaciones como ser los clústeres y la migración de máquinas virtuales.

La extensión de la red L2 entre Datacenters conlleva varios desafíos:

Escalabilidad: El diámetro STP se define como en número de saltos entre dos “switches” que pertenecen a un “spanning tree”. Cuando se interconectan Datacenters, este valor puede exceder fácilmente el valor de 7 recomendado por la IEEE [129], pudiendo dar resultados inesperados [130].

Aislamiento: Cuando se extiende una instancia de STP a múltiples sitios, solo una va a contener el “root switch”. Si este dispositivo falla ocurre un cambio en la topología, todas las VLANs dentro de esta instancia se afectan debido a la reconvergencia de STP. Debido a que STP es un protocolo conservador, prefiere la pérdida de conectividad a “loops” temporales, por lo que el tráfico puede ser interrumpido en todos los sitios durante el proceso.

Multihoming: Debido a que STP selecciona un único camino entre el dispositivo “root” y cualquier otro dispositivo en la misma instancia de STP, los vínculos múltiples entre los sitios no van a ser utilizados en su totalidad.

Otro efecto colateral se llama “tromboning” y puede suceder entre Datacenters cuando existe ruteo interno no óptimo entre las VLANs extendidas como se ejemplifica en la Figura 81. Esto puede suceder cuando un elemento de una clase activo situado en un Datacenter se comunica con un elemento activo situado en otro Datacenter, cuando era posible la comunicación entre elementos en el mismo Datacenter.

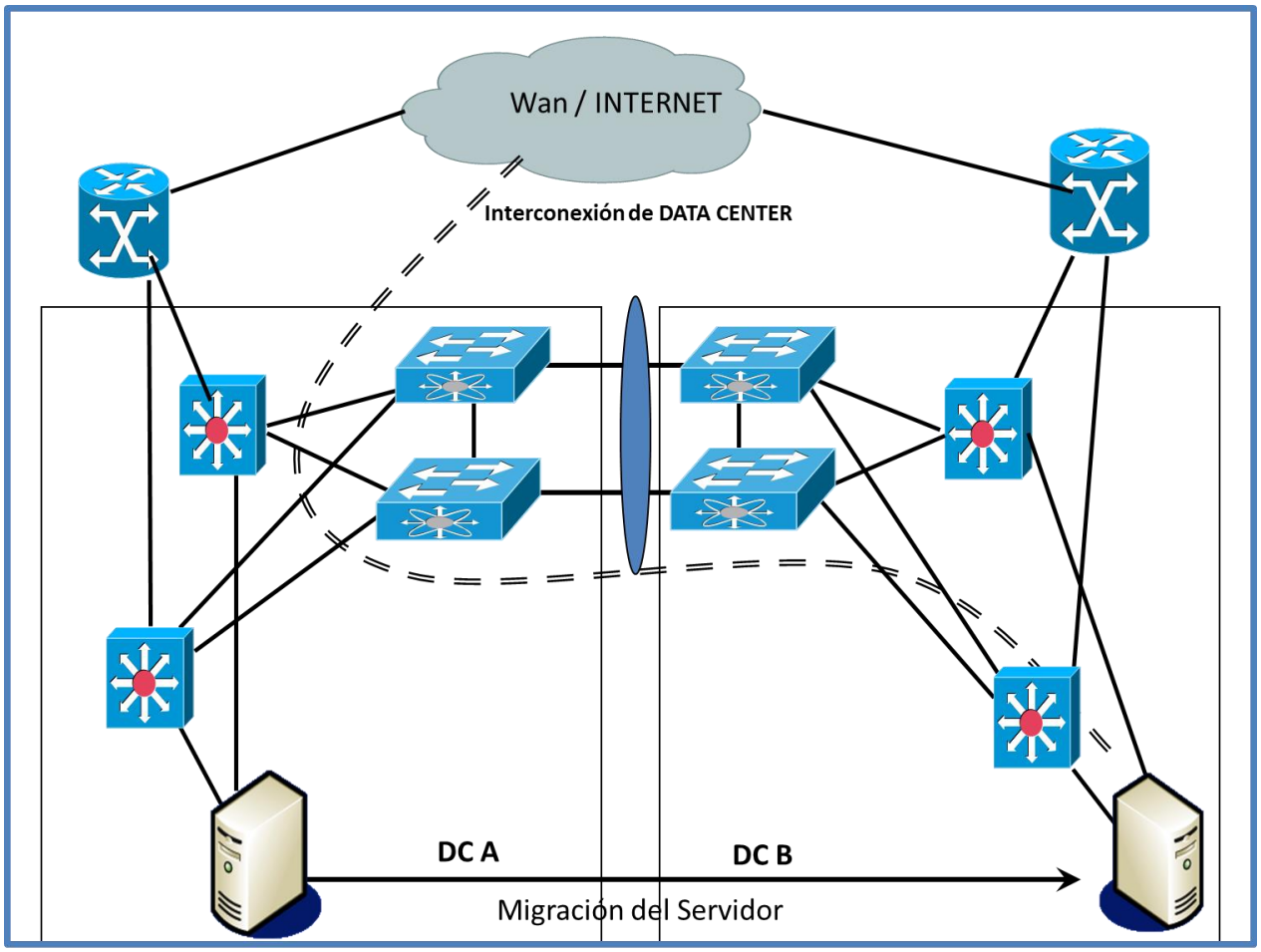

Figura 81 - Problemas en el ingreso - Tromboning

La confidencialidad de datos es otro tema que surge en las empresas cuando utilizan vínculos de comunicaciones que no están bajo su control. Algunas industrias se rigen por normas regulatorias que exigen el cifrado de datos en la interconexión entre los Datacenters.

8.4.11.2 Soluciones a la interconexión de Datacenters

Existen varias soluciones a la interconexión de Datacenters, algunas basadas en estándares y otras basadas en soluciones provistas por fabricantes. Estas últimas soluciones suelen aproximarse a estándares, los cuales han sido mejorados por los fabricantes y se comercializan como soluciones propietarias.

- Fibra Oscura:
  Fibra oscura es un término utilizado para describir cables de fibra óptica dedicados o servicios que emulan cables dedicados. Los cables físicos tienen una cobertura de entre 50 y 75 km (la distancia cubierta por un transmisor láser “single-mode”). Lo más usual es que un proveedor local brinde el servicio DWDM (“Dense Wavelenght Division Multiplexer”) que presenta en cada sitio una fibra que parece ser un servicio dedicado. El láser se encuentra físicamente multiplexado y DWDM tiene conocimiento limitado de los protocolos que la empresa transmite por el vínculo. EL servicio DWDM provee redundancia adicional a nivel de circuito mediante el uso de topologías tipo anillo y se pueden proveer varios servicios sobre el mismo cable.

- Pseudowires MPLS:
  En lo que respecta a MPLS, la mayoría de las organizaciones deciden contratar servicios L2 de un proveedor de servicios. Los proveedores de servicios utilizan el protocolo MPLS internamente en sus redes para proveer una amplia gama de servicios WAN e Internet. Por lo general, el proveedor entrega un puerto Ethernet en los dominios de la empresa, el cual se conecta a la red MPLS del proveedor. Junto con MPLS se han desarrollado un gran conjunto de protocolos que pueden proveer emulación de Ethernet sobre la red MPLS. Las tecnologías como VPLS, EoMPLS, GREoMPLS y L2TPv3 proveen maneras de emular redes Ethernet. Las empresas grandes, pueden construir sus propias redes MPLS para tener mayor control sobre los servicios y la seguridad de la WAN, pero para la mayoría de las empresas esto no es viable ya que MPLS es un grupo relativamente complejo de protocolos que requieren mucho tiempo para aprender y comprender. Al nivel de la empresa, el construir sistemas de misión crítica es realmente difícil y por lo tanto generalmente debe evitarse.

- Multichassis Link Agregation (MLAG):
  MLAG describe la unión lógica de dos “switches” físicos en un único dispositivo lógico. El plano de control lógico es una única entidad de software. Esto previene condiciones de “loop” y reduce el riesgo operacional asociado. Es simple de configurar, usar y mantener comparado con otras alternativas. En este escenario el proveedor debe ofrecer servicios L2 para interconectar los dispositivos mencionados. Se debe aclarar que MLAG no es un estándar. Cada vendedor tiene un nombre distinto para esta tecnología como ser vPC (Cisco), IRF (HP), MC-LAG (Juniper). Para utilizar MLAG en la interconexión de Datacenters, se debe conectar el servicio L2 en los puertos MLAG del “switch” para evitar “loops”. Es posible conectar varios servicios L2 en estos dispositivos, pero se debe tener en cuenta que una falla en el proveedor de servicios puede reducir el ancho de banda y requiere un diseño cuidados si se está utilizando calidad de servicio para proteger aplicaciones críticas de negocio.

- Equal Cost Multipath (ECMP)/TRILL:
  ECMP es una de las nuevas opciones para la interconexión de Datacenters. Este estándar del IETF [131] provee un protocolo de múltiples caminos (“multipath”) que rutea paquetes Ethernet a través de hasta 16 caminos que tienen el mismo ancho de banda o costo. Si bien este estándar surgió como una tecnología de Datacenter, puede ser utilizado para la interconexión de Datacenters. Provee alta disponibilidad ya que se utilizan “switches” redundantes en todos los sitios y también provee aislamiento nativo de STP. Una característica única de TRILL como tecnología de interconexión es el hecho que soporta topologías del tipo “partial mesh” para múltiples Datacenters ya que el tráfico L2 es ruteado sobre el “core” TRILL. Si bien las funcionalidades principales se encuentran completas, el protocolo TRILL todavía se encuentra en fase de desarrollo. Muchos de los principales fabricantes aún no han liberado implementaciones que cumplen con el estándar completo, por lo que si bien es posible realizar implementaciones con equipos de un único proveedor, se pueden encontrar problemas de interoperabilidad en ambientes heterogéneos. Algunos fabricantes (que han participado en el desarrollo del estándar) han extendido el estándar para agregarle funcionalidades propietarias; en este caso encontramos VCS de Brocade y FabricPath de Cisco.

- Soluciones propietarias:
  De todo lo anterior, se puede concluir que existen desafíos complejos en el área técnica para extender las redes Ethernet entre los Datacenters. Sin embargo, los fabricantes desarrollan protocolos propietarios para resolver estas complejidades; este es el caso del protocolo “Overlay Transport Virtualization” (OTV) [132] (para este protocolo, Cisco

lo presentó ente el IETF para su estandarización [133]) y del protocolo “Ethernet Virtual Interconect” (EVI) [134]. Estos protocolos encapsulan Ethernet en IP para transportarlo sobre la WAN. Para estos protocolos los dispositivos que se encuentran en el borde de la red proveen funcionalidades como ser aislamiento de STP, caminos redundantes (“multipath”), balanceo de carga. El diferencial que agregan estas soluciones (o por lo menos sus fabricantes prometen) es la facilidad de configuración y el mantenimiento de la misma. Como contrapartida, el uso de estas funcionalidades se encuentra en equipamiento de alta gama en donde se deben adquirir las licencias de uso para estas funcionalidades. La simplicidad de estos enfoques hace muy atractivo su uso en la empresa.

Para el cifrado de datos en la interconexión de Datacenters, va a depender su aplicación según la tecnología que se utilice. Si la interconexión se realiza a nivel L2, existe la posibilidad (según si el equipamiento lo soporta) de realizarlo utilizando el estándar de la IEEE 802.1AE (MACSEC) [135]. El intercambio y gestión de claves de cifrado forma parte de una extensión del protocolo 802.1X recogida en la revisión de 2010 (802.1X-2010) [136]. A todos los efectos, el funcionamiento es el mismo que el de un cliente Wi-Fi que sigue se comunica de forma segura con un punto de acceso siguiendo el estándar 802.11i, sólo que el medio físico de transmisión es el par trenzado. MACSEC refuerza la seguridad en el último tramo físico entre el puerto del “switch” y el punto de red de dispositivo final, proporcionando integridad, acreditación y privacidad en las comunicaciones de datos evitando de este modo ataques de tipo "man-in-the-middle", y escucha y/o repetición alterada de la comunicación. El cliente conectado al puerto del “switch” tiene que ser compatible con los protocolos 802.1AE y 802.1X. Se debe señalar que el cifrado con MACSEC aplica a enlaces punto a punto como ser la interconexión entre dos Datacenters situados en dos edificios cercanos. En otros escenarios (sobre todo donde se configura una malla de túneles GRE o no existe soporte de MACSEC) el cifrado de datos se realiza mediante el tunelizado de los vínculos mediante IPsec, como hemos visto anteriormente en las secciones referidas a las VPNs.

8.4.11.3 Resumen

Antes de comenzar un proyecto de interconexión de Datacenters, nos debemos plantear si es la solución para nuestras necesidades (generalmente como implementación de un plan de recuperación ante desastres). En el caso que necesitemos un plan para evitar desastres, donde existen instancias de servidores virtuales que pueden ser movidos entre Datacenters como respuesta a un evento mayor, la interconexión es el camino a recorrer. Se deben considerar cuidadosamente los requerimientos del negocio ya que la migración de máquinas virtuales entre Datacenters crea varias dificultades técnicas debido a la complejidad de los flujos de tráfico. Existen algunos problemas que no son tan obvios:

- La verificación de aplicaciones se dificulta cuando los servidores pueden estar en dos ubicaciones.
- La interconexión genera latencia a las aplicaciones, la cual puede producir efectos impredecibles de performance.
- Las fallas de supresión de “loops”, pueden producir indisponibilidades en ambos Datacenters.
- El consumo excesivo de ancho de banda puede resultar en la pérdida del servicio y no puede ser fácilmente controlada.

A modo de resumen, se presenta un cuadro con las principales tecnologías de interconexión L2 junto con sus principales características:

	Tecnología				
Característica	Interconexión propietaria	TRILL (FabricPAth)	EoMPLS	VPLS	OTV [133] [132]
Transporte	Óptica o EoMPLS	Óptica	MPLS o IP (GRE)	MPLS o IP (GRE)	IP
Mecanismos para evitar Loops	Agregación de vínculos multichassis (MLAG)	IS-IS	Agregación de vínculos multichassis (MLAG)	Split Horizon	IS-IS y AED
Multihoming	Depende del fabricante	ECMP (nativo)	Depende del fabricante	Depende del fabricante	Distribución de VLANs
Aislamiento de STP	Filtro de BPDU o regiones de MST	Dominio STP	Filtro de BPDU o regiones de MST	Nativo	Nativo
Cifrado	IEEE 802.1ae	No soportado	IEEE 802.1ae o IPsec (GRE)	IPsec	IPsec (con dispositivo externo)
Bytes agregados al paquete Ethernet original	No	22 byte	30 o 54 bytes (GRE)	30 o 54 bytes (GRE)	42 bytes

8.4.12 Conclusiones

En los Datacenters de nueva generación seguramente encontremos una alta densidad de poder de cómputo concentrados en espacios cada vez más reducidos ya que los servidores han aumentado su poder de cómputo y han disminuido su tamaño. Dentro de cada uno de estos servidores se alojan cada vez un número mayor de máquinas virtuales que a su vez se ejecutan las aplicaciones de negocio. Debido a este motivo, también se debe aumentar drásticamente la capacidad de transmitir información (ya sea a nivel de red como de almacenamiento). En este sentido, los adaptadores convergentes permiten el tráfico unificado de red de datos y almacenamiento sobre la misma red, lo que obliga por un lado al aumento de la velocidad en estos adaptadores así como en los "switches" de acceso y a su vez en los "switches" de distribución. En este sentido, la tecnología Ethernet ha mejorado, incorporando velocidades de 10Gb/s, 40 Gb/s y 100Gb/s [137] (ya se ha formado un grupo de estudio de la IEEE para empezar a trabajar en el nuevo estándar de 400Gb/s [138]). Otra necesidad es la de los múltiples caminos, ya sea para disponer de alta disponibilidad o para poder aumentar la tasa de transferencia. Debido a que protocolos como STP, bloquean estos caminos, se deben hallar otro tipo de soluciones para este problema. Para solucionar esta dificultad existe el protocolo TRILL, que según vimos antes, habilita Ethernet con múltiples caminos (multipath) y mejora las capacidades de Ethernet en el Datacenter virtualizado. El inconveniente de esta opción es que el protocolo se encuentra en fase de implementación por parte de los fabricantes y al momento no existe una implementación completa del mismo. Por otro lado existen soluciones propietarias como "FabricPath" (en realidad es una extensión propietaria de TRILL) que solucionan este problema. Mirando el futuro, en la medida que el rendimiento de OpenFlow

mejore en conjunto con el desarrollo de integrados con funcionalidades nativas, es altamente probable que las técnicas de “forwarding” utilizando OpenFlow puedan desplazar a las soluciones equivalentes que utilizan arquitecturas y equipamiento tradicional (plano de control y plano de datos en el mismo dispositivo), por lo que las redes SDN puedan posicionarse en el Datacenter. Con esta tecnología, se puede desarrollar nuevos protocolos para cubrir estas necesidades [139].

Al poder disponer de la red convergente, anchos de banda adecuados, y posibilidad de caminos múltiples es posible la utilización sobre el mismo medio de FCoE. La gran ventaja de la utilización de este protocolo es que Ethernet solamente se convierte en un transporte para FC, las demás características como ser la posibilidad de virtualización (“zoning”) [106], las herramientas de administración son las mismas que con FC tradicional, por lo que no se debe invertir en la capacitación del personal de administración / operación. Combinado con los adaptadores convergentes (CNA), los cuales se presentan hacia el sistema operativo como adaptadores de red (NIC) y adaptadores FC (HBA), la familiaridad con la plataforma es total.

Si bien las ventajas de las redes convergentes en el Datacenter son altas, se debe analizar la estructuras de costos, en algunos casos se da la situación que dos redes separadas bien diseñadas es una mejor opción [140]. Es claro que las empresas que se encuentran desarrollando nuevos Datacenters o realizando una expansión grande de los mismos deben considerar una infraestructura unificada [141].

9. Caso de estudio

9.1 Introducción

En este capítulo se presenta un caso práctico de estudio el cual tiene el objetivo de aplicar el conjunto de tecnologías vistas en los capítulos anterior de una manera práctica y ver las interacciones entre ellas por un lado y por otro lado implementar en una maqueta, utilizando un simulador de equipamiento de redes concreto. En este caso se utilizó la plataforma de simulación GNS3 [5].

El capítulo se divide en cuatro partes:

- Hipótesis de trabajo.
  En esta sección se presenta el caso a resolver junto con la realidad a solucionar.
- Análisis de la solución.
  En esta sección, utilizando las guías desarrolladas en el trabajo, se realiza un análisis de las tecnologías / protocolos a utilizar para resolver el escenario planteado.
- Implementación de la solución.
  En esta sección se detalla (en función de los resultados de la sección anterior) los pasos utilizados para construir la maqueta en detalles, las herramientas utilizadas, las limitaciones de la solución así como la estrategia de implementación de la misma, para finalmente realizar pruebas de validación sobre la misma.
- Conclusiones.
  En base a las secciones anteriores, en esta sección se analiza cómo impacta el hecho de utilizar virtualización de redes en el caso de estudio y las diferencias que hubieran ocurrido de no utilizarla.

9.2 Hipótesis de trabajo

Para el presente trabajo, se propone un caso de estudio donde se presenta una empresa con dos sitios interconectados entre sí, la cual solicita a un proveedor un servicio para gestionar sus Datacenters ubicados en dos sitios externos, con un esquema de conectividad según Figura 82. El presente caso de estudio se corresponde con una red actualmente en producción en la que participó en sus etapas de diseño, implementación y operación. Debido al compromiso de confidencialidad asumido con la empresa, no es posible revelar ciertos detalles de implementación de la red así como la fuente de alguna información (presupuestos, cotizaciones, etc) que se utilizan en el presente trabajo.

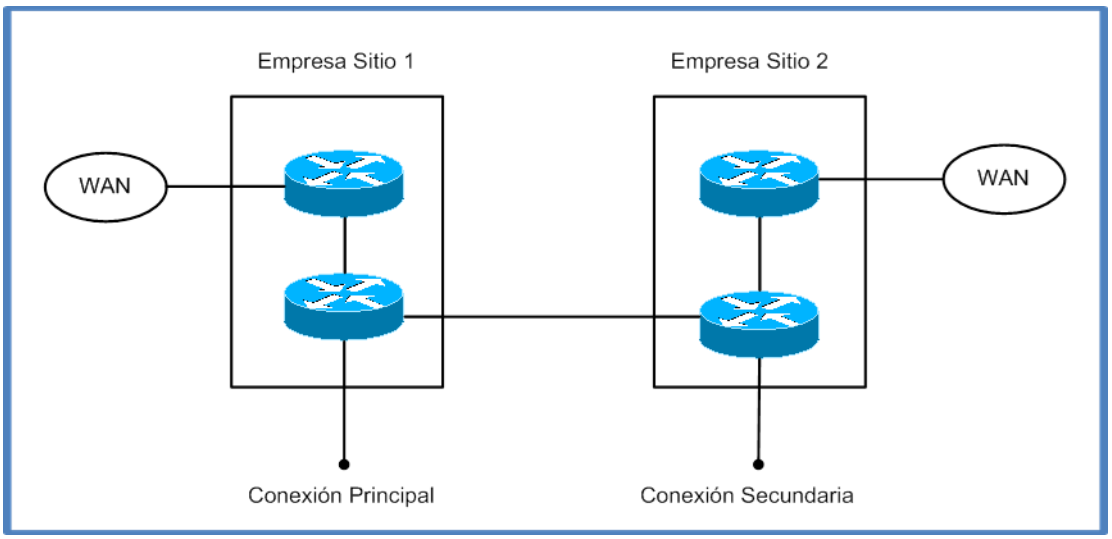

Figura 82 - Esquema de conectividad

La solución debe cumplir con los siguientes requerimientos:

- Las comunicaciones que se realicen entre los entrantes/salientes de la empresa y/o los Datacenters deberá ser cifrada.
- Debido al tipo de aplicaciones que dispone la empresa (p.e. Clusters de Máquinas virtuales), la misma VLAN generada en un Datacenter debe estar definida e interconectada en el otro Datacenter como un único dominio L2.
- Se debe segmentar las redes y realizar controles de flujo de datos entre ellas.
- Dentro de los Datacenters, cualquier conexión entre dos redes/subredes distintas deberá atravesar un "firewall" de manera obligatoria.
- Se debe diseñar la solución para soportar un fallo a la vez de forma automática. En caso de múltiples fallas se debe tomar acción de manera manual.
- Para la interconexión en los sitios de la empresa se utilizará equipamiento dedicado y no se deberá intercambiar información de ruteo mediante protocolos dinámicos entre el equipamiento del proveedor y el equipamiento de la empresa.
- Se preferirá la utilización de protocolos estándar para la implementación de la solución.
- Para la interconexión de los sitios, se dispone de un servicio VPLS provisto por un proveedor de telecomunicaciones.

9.3 Análisis de la solución

9.3.1 Justificación de las decisiones de diseño

Para el diseño de la solución, se debieron tomar en cuenta los requerimientos y en base a ellos aplicar las herramientas necesarias para poder construir la solución deseada.
Uno de los requerimientos más fuertes que se tiene es el cifrado de las comunicaciones ya que nos implica utilizar algún protocolo para encapsular y luego poder cifrar el tráfico entre los distintos puntos de la sección WAN de la solución (los routers que dan acceso a los distintos sitios). Para este cometido utilizaremos GRE que es un mecanismo para encapsular paquetes de un protocolo dentro de otro protocolo. Dentro de los túneles GRE se cifra el tráfico utilizando IPsec en modo transporte. El modo transporte fue seleccionado. Esta decisión fue tomada basándonos en el hecho que el modo transporte de IPsec introduce menor "overhead" (nos estamos ahorrando un nuevo encapsulamiento IP, con lo que se maximiza la carga útil) y es el mecanismo apropiado para cifrar conexiones nodo-a-nodo (entre las puntas de los túneles GRE) sobre la red de un proveedor de servicios (que a la vez nos asegura que las redes de los distintos clientes se encuentran totalmente aisladas). Configurando y cifrando los túneles de esta manera, nos aseguramos la posibilidad de cursar tráfico de "multicast" que es necesario para dar soporte los protocolos de ruteo dinámico (p.e. OSPF). Con las consideraciones vistas anteriormente, se dispones de una estructura de túneles GRE + IPsec entre los routers de acceso a los distintos sitios, pudiendo armar hasta un esquema de comunicaciones tipo "full mesh".

Hasta este momento disponemos de una malla de túneles GRE cifrados con IPsec entre los distintos sitios; para solucionar ruteo parece lo más apropiado elegir un IGP ya que el ruteo lo vamos a resolver dentro de un mismo sistema autónomo como se plantea en los requerimientos a implementar. La justificación de este requerimiento proviene del hecho que la empresa y el proveedor tiene cada uno su propio dominio de administración, por lo cual no es razonable el intercambio de rutas mediante un IGP. Si aplicáramos las buenas prácticas, se debería implementar el intercambio de rutas mediante un EGP entre ambos sistemas autónomos. En el caso particular, debido a que solamente existen dos puntos contactos y se

entiende que implementar un EGP tiene un alto costo tanto en la implementación como en la operación de la solución, se decide implementar esto mediante rutas estáticas y la redistribución de las mismas dentro del IGP de la empresa. Este aspecto se detallará más adelante en la sección correspondiente al diseño de la capa de ruteo de la solución. Como IGP se decidió la utilización de OSPF, el cual es un protocolo de ruteo estándar que cuenta con implementaciones estables y nos permite bastante flexibilidad en su configuración.

En lo que refiere al requerimiento de la inspección de tráfico, según los requerimientos, cada vez que exista tráfico entrante a alguna de las redes del Datacenter, éste deberá ser inspeccionado por un "firewall" (incluso si provino de otra red del Datacenter). Para poder cumplir con este requerimiento, el "firewall" se debe ubicar entre el router de acceso al sitio y el equipamiento donde se definen las distintas redes ("Switch" L3 del Datacenter). Para poder aplicar seguridad a estos tráficos, se debe definir redes de integración entre el router y el "switch" en la forma de subinterfaces encapsuladas mediante dot1q en el router y un "trunk" de VLANs (encapsulado en do1q también). En el medio se interconecta el "firewall" que se lo configura en modo transparente (funciona como un bridge más inspección de paquetes) y al cual se lo virtualiza en distintos contextos. Cada contexto funciona como un bridge el cual conmuta los paquetes (o frames) entre la subinterface "inside" y la subinterface "outside". En cada contexto se pueden definir reglas independientes de los demás contextos. El detalle de la configuración lo veremos más adelante en el presente documento donde se detallan las zonas de seguridad de la solución. En el "switch" del Datacenter, para terminar de implementar la red de integración se define una VRF (Virtual Routing and Forwarding) la cual se asocia por un lado con la interface VLAN hacia el "firewall" y por el otro lado con la interface VLAN asociada con la VLAN donde se conectan los servidores de una determinada subred; esta última será el "default Gateway" de los servidores sobre la subred en cuestión. Las VRFs definidas aquí son las que generan las distintas redes del Datacenter, por lo cual deben ser publicadas en el protocolo de ruteo. Esto implica tener un proceso de OSPF por cada VRF y debe formar vecindad con el router de acceso al sitio para poder propagar las rutas por toda la solución. Para que esto sea posible, debe prestarse atención de permitir el tráfico OSPF (recordar que OSPF opera directamente sobre IP con número de protocolo 89) en cada contexto virtual del router.

Debido al tipo de aplicaciones que se están utilizando ("clusters" de máquinas virtuales) el dominio de colisión de cada VLAN L2 se debe extender entre los Datacenters. Para poder implementar esto, lo que se debe lograr es transportar un protocolo de L2 (en nuestro caso Ethernet) sobre un protocolo de transporte de L3 (en este caso, debe poder transportarse por dentro de un túnel GRE cifrado con IPsec). Para lograr esto se utilizó el protocolo L2TPv3 el cual trabaja sobre IP con número de protocolo 115. Para el transporte agrega un nuevo encabezado IP y un encabezado de L2TP de 16 bytes. A nivel de implementación para cada VLAN L2 a transportar se crea un "pseudowire" contra el otro "endpoint" asociado a una subinterface encapsulada en 802.1q. Otra alternativa para implementar el transporte de las VLANs es utilizar AToM ("Any Transport over MPLS"). Se entiende que en este escenario como MPLS no es requerido para otros servicios, L2TPv3 es una solución simple y eficiente para resolver la situación ("Good Fit").
Si bien se verá en detalle más adelante, es de orden destacar que a nivel de ruteo en los routers de acceso a los sitios (porción WAN de la solución) se ha designado a las interfaces físicas de los mismos como "passive", obligando que el tráfico fluya por los túneles que están cifrados. De esta manera ni siquiera la información del plano de control es transmitida en texto claro.

Como estrategia de implementación, en una primera etapa se resolvieron los problemas de conectividad sin la instalación de los firewalls. Durante esa etapa, se habían dejado por defecto en los túneles el valor que usa OSPF (en OSPF el costo es inversamente proporcional al ancho

de banda de la interface) y resultaba que era posible que el tráfico en una dirección tomara un camino y el tráfico de retorno tomara otro camino distinto ya que todos los túneles tenían el mismo costo asociado. Cuando se agregaron los firewalls, surgió el inconveniente que el "firewall" filtraba el tráfico saliente que no estaba asociado con tráfico entrante. Es decir que si un tráfico se encaminaba por un determinado camino hacia los servidores y retornaba por otro distinto; era bloqueado por el "firewall". Para solucionar este inconveniente se "endureció" el ruteo, es decir que las dos alternativas posibles, siempre se escoge una como preferida. Esto resuelve el problema y no le quita disponibilidad a la solución ya que cuando la opción preferida no se encuentre disponible, el protocolo de ruteo va a elegir la otra opción. Para poder implementar esto, se configura en cada interface un ancho de banda nominal que se usa para el cálculo del costo (en OSPF costo = 10^8 / ancho de banda en b/s). Más adelante en el presente documento se mostrará este aspecto de la solución con mayor detalle.

9.3.2 Esquema de conectividad L2

Para implementar la solución se utilizará un servicio "LAN Service" provisto por el proveedor de servicios de Telecomunicaciones. La implementación del servicio "LAN Service" se realiza a nivel de capa 2. Dicho servicio está basado en EPLAN definido por el MEF en su norma 6.1. El servicio "LAN Service" permite que múltiples sitios se conecten a un único dominio "bridgeado", sobre una la red MPLS. Los sitios simulan estar ubicados en una misma LAN, a pesar de su locación dispersa. Con esta unificación se obtiene una arquitectura de red sencilla y eficiente.
Este servicio se implementa en la red MPLS del proveedor, basándose en el servicio VPLS ("Virtual Private LAN Service"), RFC 4762 [52], donde los PEs ("Provider Equipment") son capaces de aprender, "bridgear" y replicar tráfico "Ethernet" sobre una VPLS. Con MPLS se logra que las redes de los distintos clientes del proveedor estén totalmente aisladas unas de otras, de manera que ninguna de ellas podrá inyectar tráfico de forma no autorizada en otra red. La información se enviará de forma segura sobre una red privada sin la posibilidad de intrusiones. Es posible transportar tráfico de tipo Ethernet con VLAN marcado como se define en IEEE 802.1Q [8], segmentando en varias redes según las necesidades que puedan surgir.
La conectividad L2 entre sitios se establece a través de una red MPLS VPLS brindada por el proveedor de servicios de Telecomunicaciones.

Los puntos de conectividad de la red son:

- EmpS01: Sitio 1 de la empresa.
- EmpS02: Sitio 2 de la empresa.
- DC01: Datacenter 1.
- DC02: Datacenter 2.

La red VPLS brinda un conexionado símil a una LAN entre los cuatro sitios. Para su conexión se utilizan 4 routers, los cuales deben dar soporte al trabajo de cifrado, redundancia y performance esperados para esta red. El rol de este equipamiento así como del otro que se utilizará, será detallado en las secciones siguientes del presente documento.
En la Figura 83, se realiza un esquema de alto nivel de la conectividad propuesta.

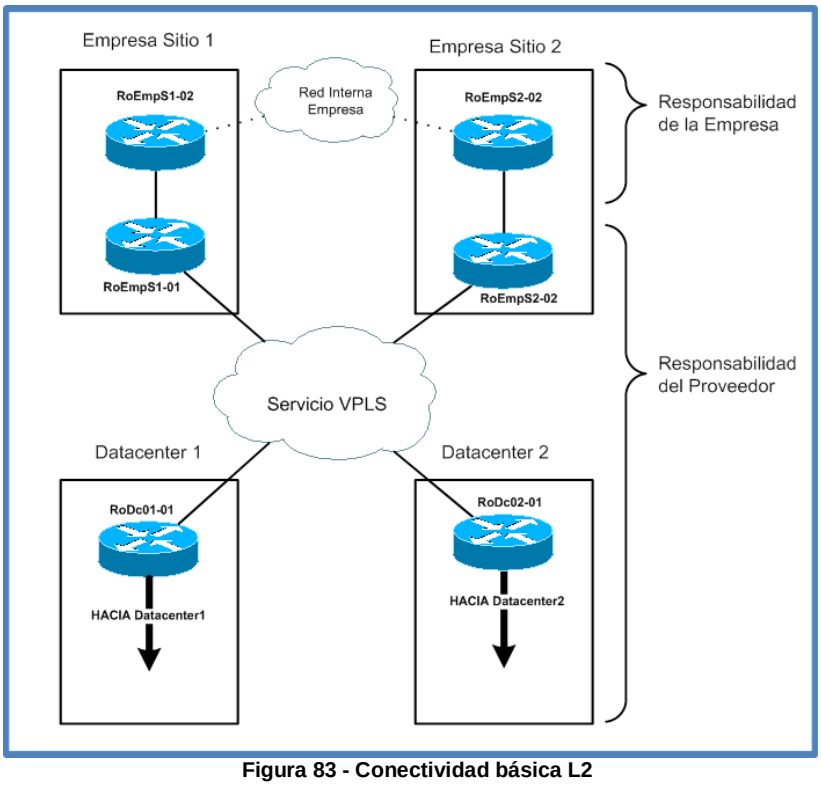

Figura 83 - Conectividad básica L2

9.3.3 Esquema de conectividad Capa 3

La conectividad L3 se divide en tres áreas que se detallan en las siguientes secciones.

9.3.3.1 Direccionamiento IP

Para el desarrollo de la solución, se utilizan direcciones IP en el espacio de direccionamiento de la empresa. Esta nos ha asignado los rangos 10.10.229.0/24 y 10.10.230.0/24, sobre las cuales se crearán subredes IP.

Para la interconexión entre el equipamiento de la solución y el equipamiento de la empresa, se utilizarán subredes de integración 10.10.220.24/29 y 10.10.220.200/29.

Los detalles del plan de direccionamiento se explicarán a medida que se describan los componentes de la solución donde intervienen. En el Anexo I – Plan de direccionamiento IP del caso de estudio, se detalla el plan completo.

9.3.3.2 Area WAN

A fin de mantener cifradas las conexiones salientes de la empresa hacia los Datacenters DC1 y DC2 se crearon túneles GRE + IPsec entre sitios, los cuales simulan conexiones punto a punto sobre la red MPLS/VPLS, como se puede apreciar en la Figura 84. Sobre dicha malla de túneles punto-a-punto, y las redes generadas en ambos sitios DC, se ejecuta el protocolo de ruteo OSPF a fin de dar redundancia automática entre sitios. Para dar soporte protocolos de rutero dinámico (como OSPF), se configuró IPsec en modo transporte ya que es la única modalidad que permite tráfico del tipo “multicast” necesario para la ejecución del protocolo de ruteo. Esta configuración se realiza mediante la generación de túneles GRE utilizando subredes /30 para generar enlaces virtuales punto a punto entre los routers de la solución. Como se aprecia en la Figura 84; en los Túneles 2, 3, 5, 6 se ha utilizado esta técnica. El Túnel 1 es un caso especial ya que se utiliza para transportar tráfico entre los Datacenters como para

la extensión de LAN (L2TPv3).
Cuando se utiliza IPsec en modo Transporte debe tenerse especial cuidado con temas relacionados al MTU y la fragmentación ya que hay dos niveles de encapsulamiento (en el Túnel 1 hay tres) [59].

A nivel de ruteo, se eligió OSPF como protocolo de ruteo dinámico. En esta solución, estos cuatro routers están ubicados dentro del área 0 de OSPF y se obliga a intercambiar la información de ruteo a través de las interfaces tipo Túnel, lo cual se logra haciendo la interface donde se conecta el router al servicio pasiva para el protocolo. De esta manera nos aseguramos que no se trafica información con datos potencialmente sensibles por fuera de los Túneles cifrados (ni siquiera información del plano de control). En esta solución se han "endurecido" la solución de ruteo, dándole a los enlaces contra el Sitio1 preferencia sobre los enlaces contra el Sitio2 al fijar el ancho de banda nominal de los enlaces, que luego utilizará OSPF para elegir la mejor ruta. Esto no disminuye la disponibilidad de la solución ya que al faltar un camino, el protocolo de ruteo converge automáticamente para generar el tráfico por la otra ruta disponible. Si bien aquí se describe la solución de ruteo para el área WAN, ésta se integra con la solución de ruteo para las subredes generadas en el área de los Datacenters así como el área de integración con los Sitios de la empresa. El detalle se irá describiendo a medida que se detallen el resto de las áreas que componen la solución.

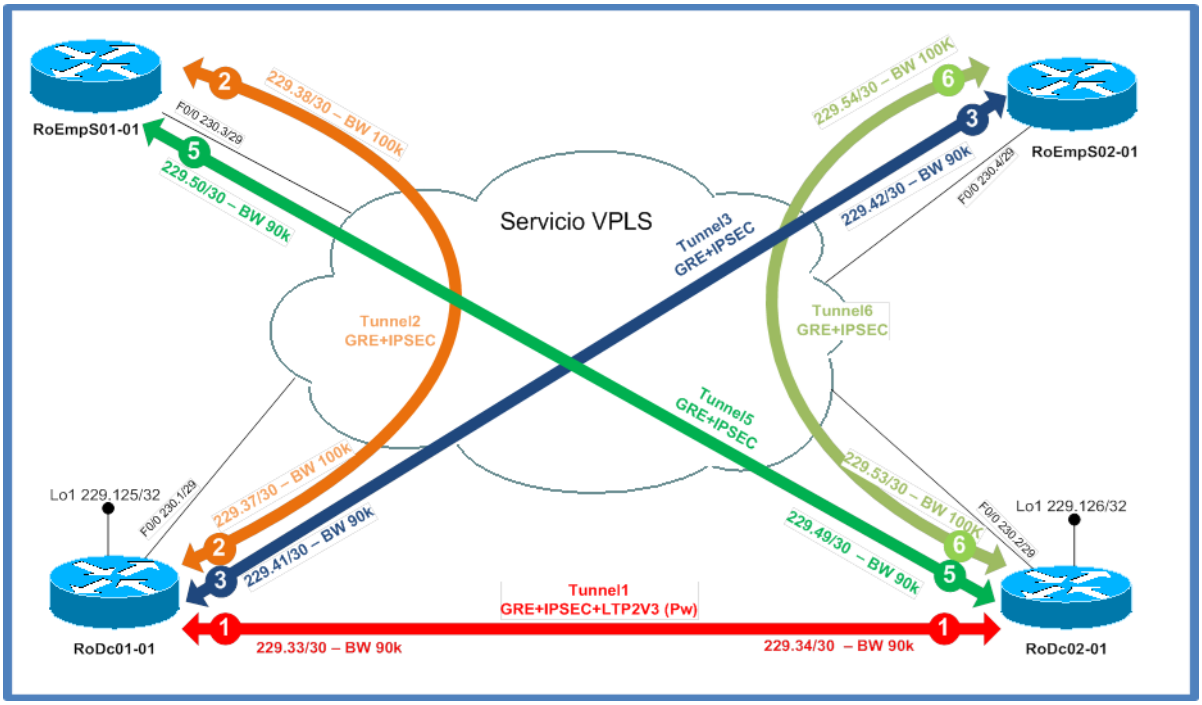

Figura 84 - Red de Túneles GRE+IPsec

9.3.3.3 Área Datacenter

Dentro de cada Datacenter, se dispone de un Router, en el cual según vimos en la sección anterior, terminan las conexiones WAN y los túneles GRE+IPsec (o GRE+IPsec+L2TPv3 en caso del Tunel1), un "firewall" virtualizable en modalidad transparente y un "switch" capa3 donde se virtualiza mediante la creación de VRFs las, distintas redes o zonas de seguridad.
En cada DC se encuentran dos Zonas de seguridad (podrían ser más; se muestran dos para ejemplificar) distintas, cada una de ellas es generada en forma separada lógicamente por una VRF distinta dentro del "Switch" L3.

Cada distinta VRF cuenta con al menos dos VLANs, una interior, la que crea la red de servicio a los servidores y otra exterior de interconexión. En la Figura 85 se observa un detalle de los componentes mencionados anteriormente.

En cada VRF se ejecuta OSPF para lograr redundancia entre sitios a Nivel WAN/MAN. Por ejemplo el router RoDC01-01 establece una relación de vecindad a nivel OSPF con el router virtual implementado por la Vrf101. Este hecho hace que en el router RoEmpS01-01 la ruta hacia la vrf101 sea aprendida por OSPF mediante el Tunel2.

Para la conexión entre el router y el “firewall”, se utiliza una única conexión, sobre la cual se encapsula utilizando 802.1q del lado de router mediante sub-interfaces y del lado del “firewall”, cada VLAN se conecta con un contexto del “firewall”, en el cual se definen las reglas de acceso para esa zona de seguridad.

Para interconexión entre el “firewall” y el “switch” L3, se usa una técnica similar ya que se encapsula la VLAN de interconexión y ésta es asociada a la VRF correspondiente. Luego se asocia la interface VLANxx (la interna) con cada VRF y los servidores de esa red son conectados a puertos de acceso definidos sobre la VLANxx (L2). En la Figura 85 se puede observar un detalle de la conexión dentro de un Datacenter.

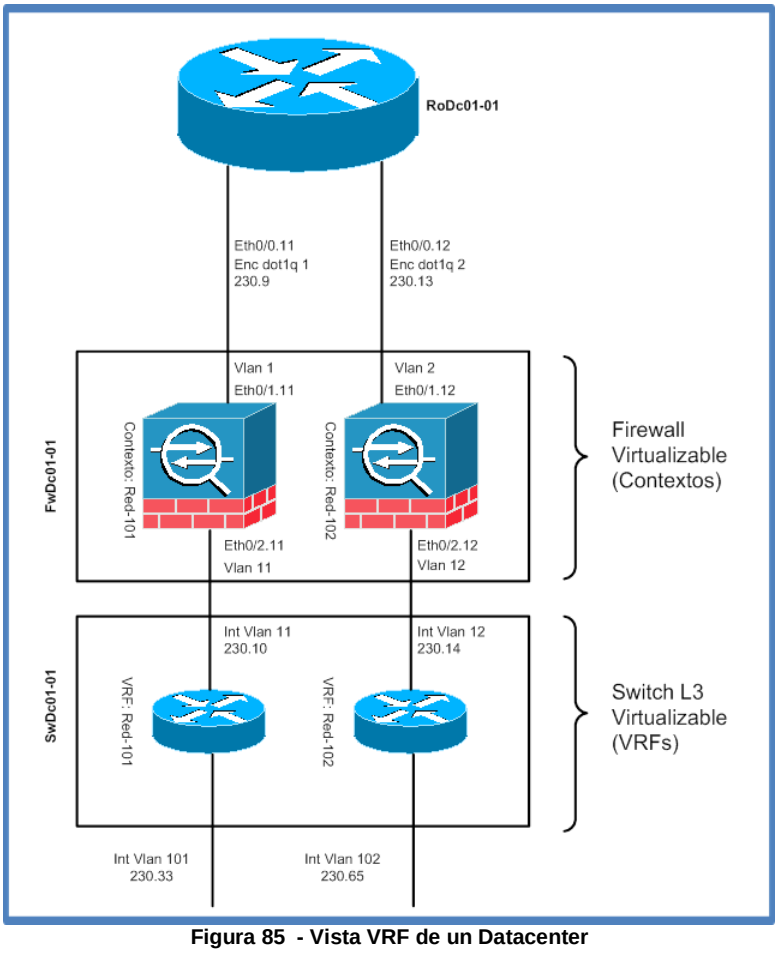

Figura 85 - Vista VRF de un Datacenter

9.3.3.4 Interconexión de Datacenters

A fin de brindar interconexión L2 entre los distintos DC y así implementar el requerimiento planteado en el caso de estudio, se crearon túneles L2TPv3 (“Pseudowire”) entre sitios a

través de los equipos Router (RoDC01-01, RoDC02-02) sobre una nueva interface configurada L2 solo para tal fin. Desde el punto de vista de los "switches" L3, ésta conexión es un simple "trunk" de VLANs (802.1q). Esta conexión no pasa por el "firewall" ya que no es posible que cambie de VLAN al pasar por el "pseudowire" porque los paquetes ("frames") llevan "tag" de VLAN, los cuales se asocian a subinterfaces en el router y a su vez se asocia a un túnel L2TPv3 específico para cada red L2.

De esta manera se logra tunelizar los distintos dominios de "Broadcast" L2 sobre una red L3. En la Figura 86, se muestra el detalle de interconexión de los equipos.

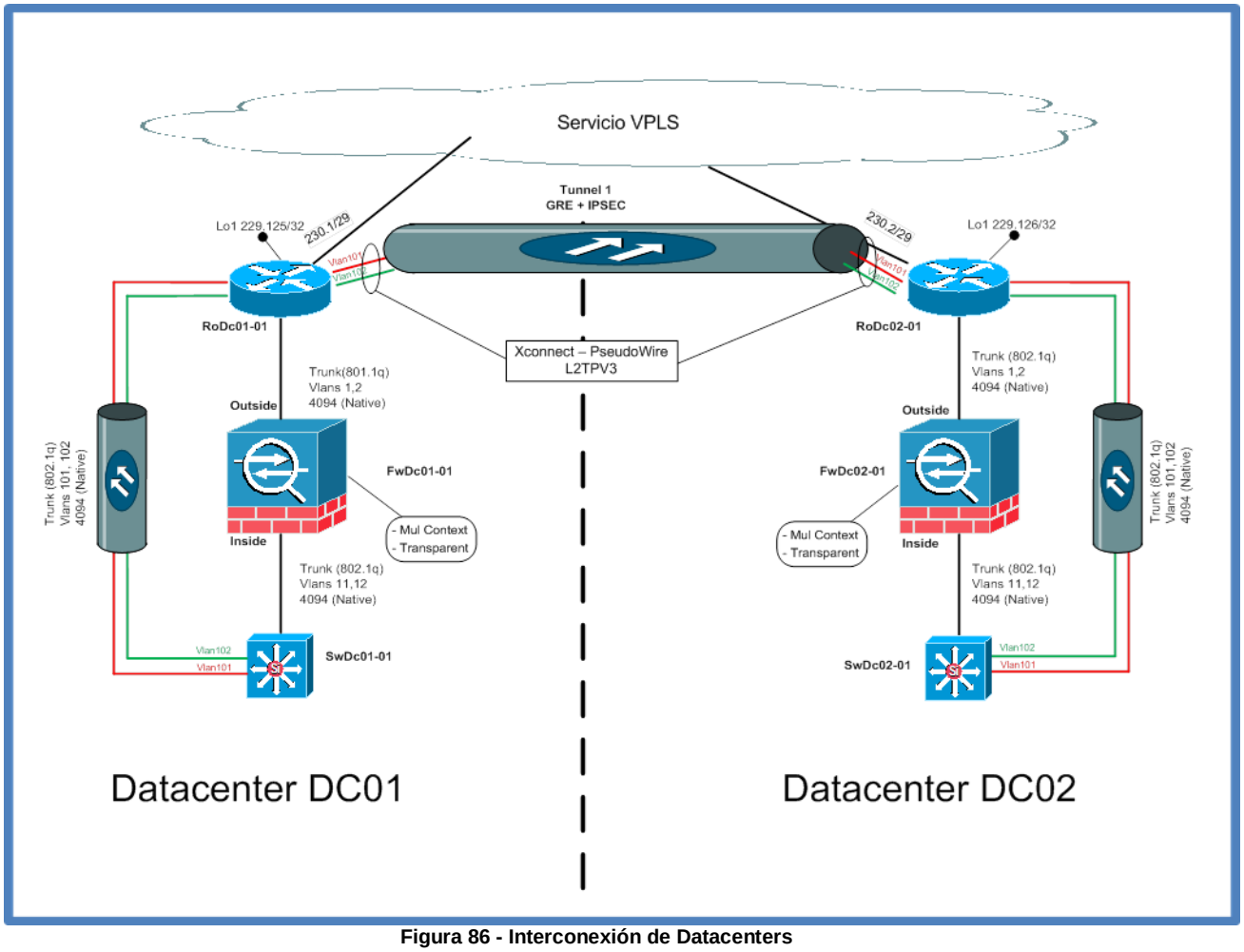

Figura 86 - Interconexión de Datacenters

9.3.3.5 Área Interconexión Empresa

La interconexión con la empresa se realiza en dos sitios e la misma. Para este propósito la empresa destina dos routers para dicha implementación (RoEmpS01-02 y RoEmpS02-02). Para cumplir con la hipótesis de trabajo de no intercambiar protocolos de ruteo dinámicos, se implementa de la siguiente manera:

- Se dan de alta en los routers RoEmpS0x-02 rutas estáticas hacia las redes 10.10.230.0/24 y 10.10.229.0/24, las cuales son redistribuidas mediante un protocolo de ruteo dinámico hacia su red interna. Se "pasivisa" la interface hacia el proveedor de tal manera que esta no participe en el protocolo de ruteo. Las rutas estáticas antes mencionadas son instaladas sobre la interface, para que de esta manera cuando la

interface cae, esta ruta se quita de la tabla de ruteo y por ende no se publica más en el protocolo de ruteo.

- En los routers RoEmpS0x-01 se “pasivisan” la interface contra la WAN y la interface contra el RoEmpS0x-02 correspondiente para:
  1. No intercambiar información de ruteo dinámico con la empresa.
  2. Intercambiar información de control y datos únicamente mediante los vínculos cifrados.

- En los routers RoEmpS0x-01 se debe hacer un “tracking” de los túneles que terminan allí y cuando uno de estos cae, se debe “señalizar” al router de la empresa que por este camino no hay rutas disponibles. Como no hay un protocolo de ruteo ejecutando, la manera que tenemos de hacer esto es bajando la interface hacia la empresa. Cuando el túnel se levanta, debemos levantar la mencionada interface. En la Figura 87 se muestra un detalle de la conectividad entre la empresa y el servicio VPLS.

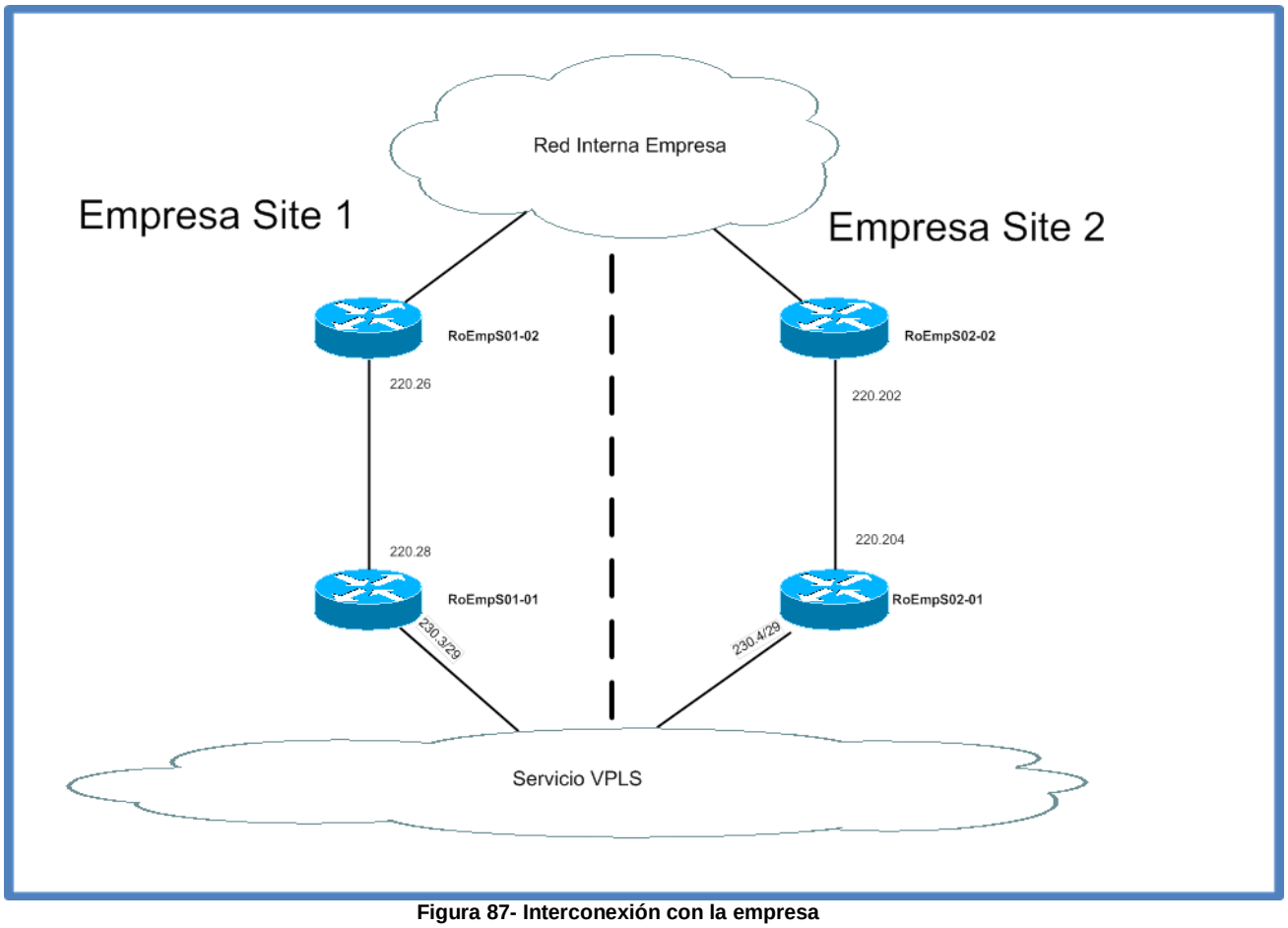

Figura 87- Interconexión con la empresa

9.3.4 Alta disponibilidad

1. Caída de enlace en DC
   OSPF da de baja las conexiones hacia el DC caído. Los clusters de aplicaciones

resuelven la alta disponibilidad de las aplicaciones (con menor poder de procesamiento).

2. Caída del Switch en el DC
   En este dispositivo se genera la "Interface" VLAN, por lo que se define un grupo de HSRP para cada Interface VLAN, la cual es la ruta por defecto de los servidores sobre esa VLAN.

3. Caída del FW
   Ante la caída del "firewall", existirá comunicación entre los servidores sobre la misma VLAN, pero no existirá comunicación inter-VLAN. Una manera de llegar a un escenario estable es que al caer la interface del router que da hacia el "firewall", bajar la interface del router que da hacia la WAN, quedando en el escenario (1).

4. Caída de un enlace en los sitios de la Empresa
   La técnica explicada en 9.3.3.5 (Área Interconexión Empresa) permite (mediante la convergencia del protocolo de ruteo) disponer de un mecanismo para señalizar al protocolo IGP de la empresa la caída de uno de los puntos de acceso y poder redirigir el tráfico internamente hacia el otro punto de conexión.

En el Anexo III (Casos de test del caso de estudio) se describen las pruebas realizadas para demostrar estas características de la solución.

9.4 Implementación de la solución

9.4.1 Maqueta de la solución

En base al diseño que se presentó, se realizó una implementación de la solución como una maqueta utilizando el simulador GNS3 [5], el cual permite emular dispositivos del fabricante Cisco Systems ejecutando de manera directa el sistema operativo de los dispositivos ("router", "switches", "firewalls", IPS, etc) y pudiendo realizar la interconexión entre ellos. A su vez este simulador permite la conexión del PC anfitrión con los dispositivos de red y la conexión de máquinas virtuales a la solución. Con estas máquinas se puede inyectar tráfico a la solución para estudiar su comportamiento.

9.4.2 Herramientas utilizadas

Para implementar la maqueta de la solución se utilizaron las siguientes herramientas

- Simulador de redes GNS3.
- VPCS [142] (emulador de PCs con funcionalidades básicas, ping, traceroute, etc)
- Software de virtualización VirtualBox [143].
- Analizador de protocolo y software de captura de redes Wireshark [144] (integrado con GNS3).
- Herramientas de "troubleshooting" integradas en la plataforma:
  - Ping, traceroute, logs de la plataforma.
  - Captura de tráfico en la propia plataforma.
  - Herramienta de administración del "firewall" (ASDM) [145].

9.4.3 Definiciones

Para la implementación de la maqueta utilizando GNS3, se tomaron las siguientes definiciones:

- Para los routers, se emuló con GNS3 un router Cisco de la línea 7200 (más específicamente el modelo 7206 [146]) con una imagen de sistema operativo C7200-ADVIPSERVICESK9-M), Version 15.0(1)M [147] la cual posee todas las características necesarias para implementar la maqueta.

- Para los “switches” en los Datacenters, se utilizó la emulación en GNS3 de routers Cisco modelo 3725 [148] con imagen de sistema operativo C3725-ADVIPSERVICESK9-M Version 12.4(25b) [149] con una placa de “switching” modelo NM-16ESW [150] que si bien no es exactamente lo mismo que un “switch”, posee todas las características necesarias para implementar la maqueta.

- Para los firewalls, se utilizó la emulación que provee GNS3 mediante QEMU [151] (plataforma de virtualización) de un “firewall” Cisco ASA 5520 [152] con sistema operativo versión 8.4.2 [153]

- Para los Pc’s/servidores se utilizó una emulación básica que se encuentra incluida con GNS3 (permite solamente realizar ping, traceroute) y también el ejecutar una máquina Windows sobre la plataforma de virtualización VirtualBox.

- El servicio VPLS fue emulado con un router modelo 3725 con imagen de sistema operativo C3725-ADVIPSERVICESK9-M Version 12.4(25b) y una placa de “switching” modelo NM-16ESW. Para le emulación de este servicio se mantuvo la configuración por defecto del equipo y se conectaron los enlaces de los sitios en puertos de la placa de “switching”. El efecto final es que todos estos puertos se encuentran sobre la misma VLAN.

9.4.4 Limitaciones de la solución

- El servicio VPLS no es real, es emulado mediante una placa de “switching” en un router (a su vez emulado por GNS3)
- La solución se emula en un único PC y cuando se le inyecta tráfico el consumo de CPU y memoria crece haciendo que la emulación funcione con un rendimiento bajo, por lo cual en esta maqueta no podrán medirse parámetros de performance
- La emulación del “firewall” (ASA) es intensiva en CPU, por lo cual a lo sumo se puede ejecutar una instancia en la misma plataforma donde se ejecuta el resto de la emulación. Por este motivo, solamente se ejecutará una instancia de “firewall” en el Datacenter DC01. De todas maneras agregar un segundo en “firewall” en DC02 es simétrico.

9.4.5 Estrategia de implementación de la maqueta

Para la implementación de la maqueta, se tomó la decisión de realizarla en al menos dos etapas. En la primera etapa, se realizaría una primera maqueta dejando de lado los firewalls de la solución. En una segunda etapa, se incorporaron los firewalls (en realidad el “firewall” del Datacenter DC01) debido a las limitaciones de la plataforma.

9.4.5.1 Etapa1 – Maqueta sin Firewalls

En esta etapa de implementación de la maqueta se implementó sin los firewalls, para poder validar el diseño sin los elementos de seguridad que pueden bloquear cierto tráfico y hacernos pensar que el diseño es inválido o aumentar considerablemente el tiempo necesario para hacer “troubleshooting”. En la Figura 88 se muestra el detalle de la topología que se confeccionó en utilizando GNS3 para esta etapa del proyecto.

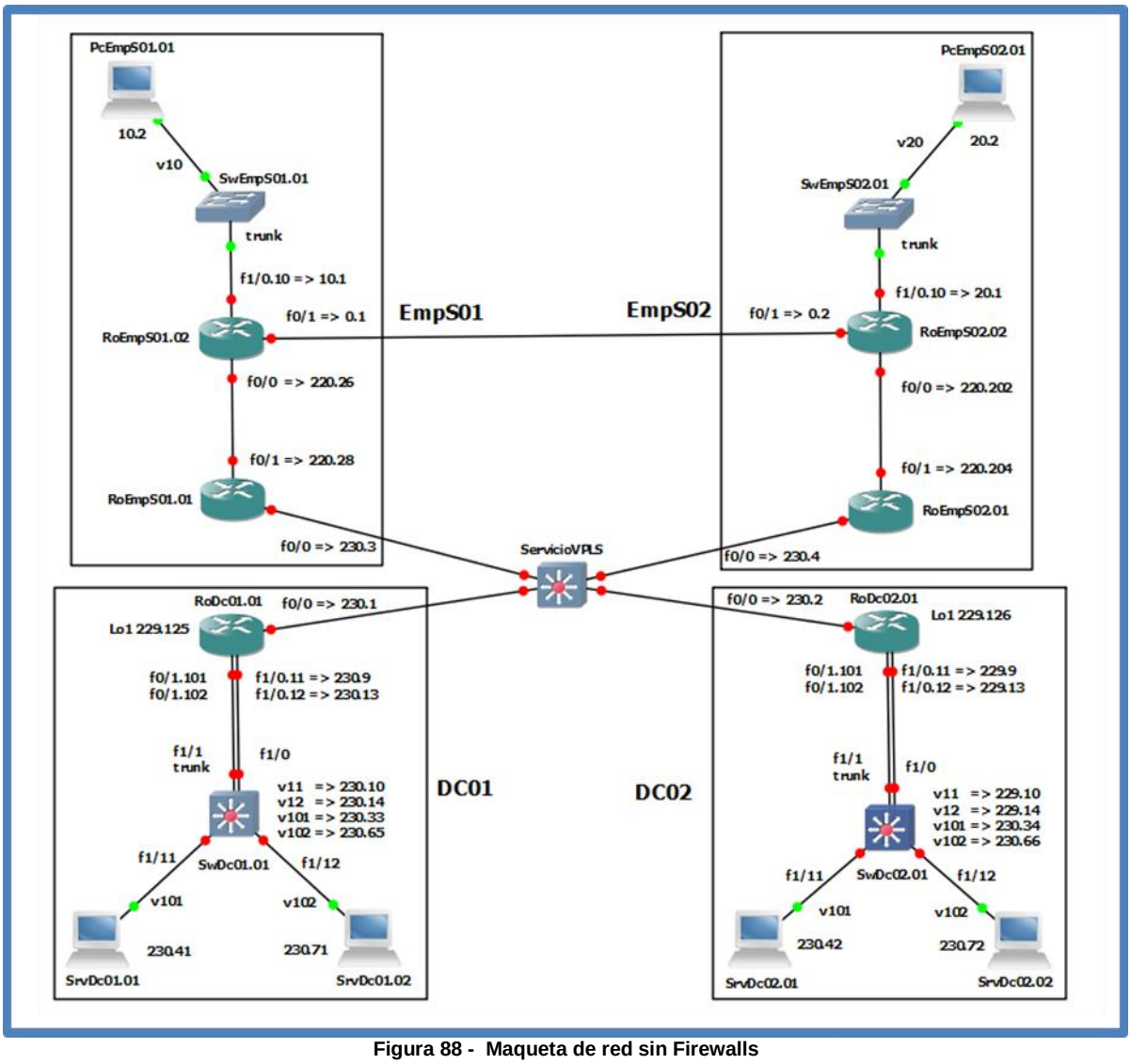

Figura 88 - Maqueta de red sin Firewalls

9.4.5.2 Etapa2 – Maqueta con Firewalls

Como se mencionó anteriormente, en esta etapa se incluyó únicamente el "firewall" del Datacenter DC02 debido a las limitaciones de la plataforma.

Para adicionar el "firewall", se lo realizó en la modalidad transparente (funciona de manera similar a un bridge) y de múltiples contextos (virtualizado) entre el router de acceso al Datacenter el switch multicapa (L3) en el cual se encuentran virtualizadas las distintas redes de Datacenter. Al funcionar como un dispositivo de capa 2, el primer impacto es en el vínculo entre estos equipos que está implementado como un "trunk" de VLANs (en este caso se transportan las VLANs 11 y 12). Al instalar este equipo entre los dos equipos antes mencionados, hay que hacer un cambio en una de las puntas. Se eligió cambiar las VLANs 11 y 12 en el router por las VLANs 1 y 2 (cambiando la VLAN nativa a la 4094). En resumen, la sección "outside" usará las VLANs 1 y 2 mientras que la sección "inside" utilizará las VLAN 11 y 12.

Otro cambio que impactó en el diseño inicial de la solución es el hecho que en toda la malla de túneles, se había dejado su configuración por defecto en los pesos asignados para el protocolo de ruteo. Al ser todas las interfaces idénticas, esto resultaba en un ruteo totalmente asímetrico; es decir que un paquete podía hacer un camino (path) desde un sitio de la empresa hasta un

servidor en el Datacenter y a la vuelta hacer un camino distinto. Esto genera un problema en el "firewall" ya que al realizar inspección del tipo "stateful", deniega el tráfico de retorno del cual no tiene información ya que no pasó por el "firewall". Para solucionar este problema, como se mencionó anteriormente, se "endureció" el ruteo básicamente favoreciendo ciertos caminos ("path") frente a otros de tal manera que los paquetes utilicen el mismo camino ("path") en ambos sentidos. En la Figura 89 se muestra el detalle de la topología que se confeccionó en utilizando GNS3 para esta etapa del proyecto.
La configuración completa de los dispositivos se encuentra en el Anexo II (Configuración de la solución del caso de estudio).

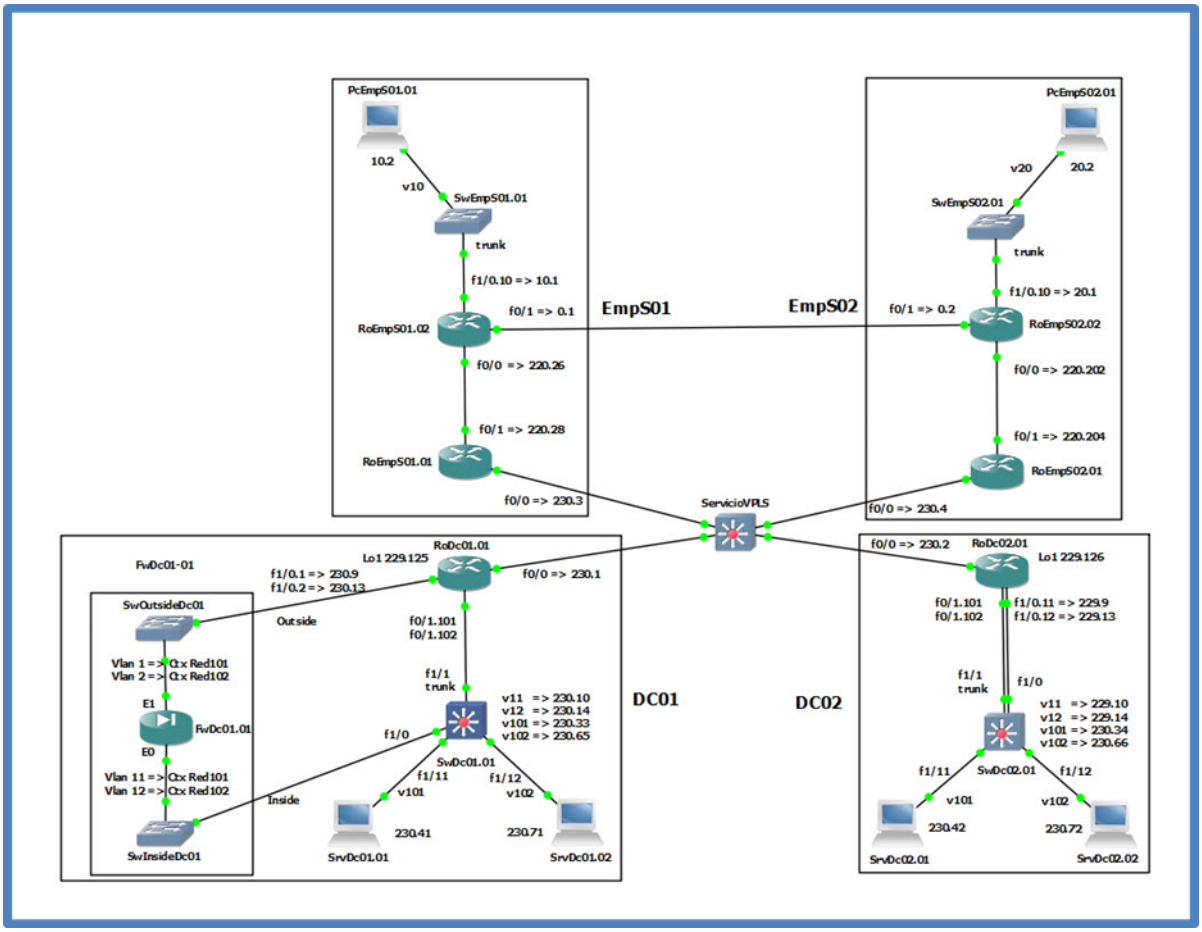

Figura 89 Maqueta de red incluyendo Firewall

9.4.6 Puntos a destacar de la maqueta

El punto más destacable de la solución es el transporte de las redes sobre la malla de túneles GRE. En este sentido, se deben diferenciar dos casos

- Tunel1: Entre RoDc01.01 y RoDc02.01 el cual s encuentra cifrado y además de transportar tráfico IP y datos del protocolo de ruteo dinámico, transporta las redes L2 mediante L2TPv3 correspondientes a los servidores del Datacenter

- Túneles 2,3,5,6: Definidos según el esquema de la Figura 84 - Red de Túneles GRE+IPsec. Sobre estos túneles GRE se transporta el plano de datos e información del protocolo de ruteo dinámico.

La cuestión con esta solución surge del ajuste de MTU de la solución para evitar la fragmentación excesiva al utilizar GRE junto con IPsec (en este caso en modo transporte). En

esta solución fueron combinados de esta manera ya que IPsec por sí solo no soporta "multicast", con lo cual no se permite la utilización de protocolos de ruteo dinámicos sobre una VPN IPsec. Los túneles GRE permiten la utilización de "multicast", encapsulando primero los paquetes "multicast" de los protocolos de ruteo en un paquete GRE y luego cifrándolo con IPsec. En este tipo de configuraciones, IPsec se configura en modo transporte ya que los vecinos IPsec y los extremos de los túneles GRE son los mismos. Un beneficio adicional de esta solución es que se ahorran 20 bytes de encabezado IPsec.

Un caso interesante resulta cuando un paquete grande se debe fragmentar en dos trozos y luego se encapsula mediante GRE. En este caso IPsec ve dos paquetes independientes (GRE + IPsec) y es común que el paquete resultante luego de cifrar el más grande, deba ser fragmentado nuevamente para su transmisión. El vecino IPsec deberá re ensamblar este paquete antes de poder descifrarlo. Esta doble fragmentación (una vez debida a GRE y luego por IPsec) incrementa la latencia y disminuye el rendimiento del dispositivo que envía los paquetes. A su vez, en el dispositivo que recibe se produce un "hit" a la CPU cuando esto sucede ya que el re ensamblado de paquetes es un proceso en la cual se debe interrumpir a la CPU para que lo realice.

Esta situación se puede evitar configurando el MTU en el túnel GRE (en este caso con el comando "ip mtu") lo suficientemente bajo para tomar en consideración la sobrecarga de GRE e IPsec (también de L2TPv3 en el Túnel 1). Por defecto el MTU de los túneles GRE se configuran como el MTU de la interface menos el tamaño de los encabezados GRE (24 bytes).

En el caso de esta solución, las sobrecargas impuestas sobre los vínculos tipo túnel son las siguientes:

Combinación de Túneles	Sobrecarga	Observaciones
GRE + IPSEC	58 bytes	GRE (24) + IPSEC Transp (34)
L2TP + GRE + IPSEC	74 bytes	L2TPv3 (16) + GRE (24) + IPSEC Transp (34)

Una solución genérica es configurar el MTU de todos los túneles en 1400 bytes (valor los suficientemente bajo para "acomodar" todos los posibles encabezado y aún con buen rendimiento).

Otras medidas adicionales que se pueden utilizar [59] son:

- Realizar un ajuste del parámetro MSS ("maximun Segment Size") en las interfaces tipo Túnel (utilizando el comando **ip tcp adjust-mss**) para que de esta manera el router disminuya el valor en el paquete SYN de las conexiones TCP. Esto ayuda a los equipos que envían y reciben tráfico TCP a usar paquetes lo suficientemente chicos para que la doble fragmentación no ocurra.

- Crear una política en la interface de ingreso del router para que se ponga en cero el bit DF ("Don't Fragment") en el encabezado IP antes de llegar a la interface GRE. De esta manera, los paquetes IP son fragmentados antes de ser encapsulados en el túnel GRE.
  Esta política se crea con el comando:

```
route-map clear-df permit 10
 match ip address 101
 set ip df 0
```

Este comando debe ser aplicado a las interfaces de ingreso a los routers de borde de la solución.

```
interface FastEthernet1/0.1
 description to_FwDc01-01 ctx v1-v11-Red101
 bandwidth 100000
 encapsulation dot1Q 1
 ip address 10.10.230.9 255.255.255.252
 ip policy route-map clear-df
```

9.4.7 Pruebas realizadas sobre la maqueta.

Para validar la configuración de los dispositivos y el haber logrado los objetivos de diseño, se realizar una serie de pruebas sobre la maqueta, las cuales se detallan en el Anexo III. Dichas pruebas se pueden dividir en tres grandes grupos:

1. Pruebas básicas de conectividad (ver detalle en 12.3.1) las cuales tienen como objetivo asegurarnos que todos los dispositivos que correspondan son alcanzables desde los sitios que corresponda.

2. Pruebas de alta disponibilidad (ver detalle en 12.3.2) las cuales tienen como objetivo observar el comportamiento de la red ante casos de contingencia como ser la caída de enlaces de datos.

3. Pruebas de verificación del MTU sobre los túneles (ver detalle en 12.3.3) las cuales tienen como objetivo medir este parámetro de la red, e cual impacta directamente en el rendimiento de la misma.

9.5 Conclusiones

En el caso de estudio se tomó una situación donde una empresa prestadora de servicios decidió implementar una solución con virtualización de redes para dar servicios a otra empresa. Se tomó la decisión de implementar la solución utilizando virtualización a nivel de redes.

De las opciones de conectividad disponibles en el mercado se optó por la opción de “Lan Service” (VPLS) ya que para realizar una conectividad similar con enlaces del tipo “LAN to LAN”, el precio es un 25% superior (ver Anexo IV – Estudio de costos involucrados en el caso de estudio).

Sobre la conectividad planteada, se debe resolver dos aspectos:
1) Conexión L2 entre los Datacenters cifrada.
2) Conectividad IP hacia/desde la empresa cifrada.

En cuanto al punto (1) y según las distintas opciones planteadas en el presente trabajo, se decidió utilizar el protocolo L2TPv3 sobre un túnel GRE cifrado con IPsec ya que la otra opción posible es utilizar MPLS sobre GRE (EoMPLS), pero esta solución es bastante más compleja de configurar y mantener (impactando en una mano de obra más calificada para su implementación y mantenimiento). Por otro lado no se necesitan todo el conjunto de servicios

MPLS en esta instalación, por lo que la opción utilizada parece acertada con la dimensión y complejidad de la red.

Para el punto (2) se utilizó la malla de túneles GRE ya que esta solución permite el cifrado de la información en plataformas de ruteo de porte medio (incluso con hardware de cifrado).

La utilización de virtualización en los “firewalls” y “switches” de los Datacenters en la presente solución supone un ahorro de costos ya que si utilizamos equipamiento sin capacidad de virtualización, se debe incluir un dispositivo por cada red que desea implementar (por ejemplo con los firewalls). Además resulta más conveniente adquirir un equipamiento de más alta gama virtualizado que varios de gama más baja sin virtualizar ya que los primeros generalmente disponen de características como fuentes redundantes, menor consumo eléctrico, menor ocupación de espacio de “rack” (si estamos alquilando este espacio, se traduce en ahorro), menor necesidad de refrigeración en el Datacenter, etc (ver Anexo IV – Estudio de costos involucrados en el caso de estudio).

En cuanto a los costos de los recursos humanos necesarios para la implementación y operación de la solución, se tiene un comportamiento similar de la economía de escala al tener que contratar una plantilla adicional para un nuevo proyecto en oposición a contratar posiciones extra, acompañando la escala de crecimiento de la solución al agregarse nuevos clientes o crecer la cantidad de redes o sitios a administrar (ver Anexo IV – Estudio de costos involucrados en el caso de estudio).

En cuanto a la escalabilidad de la situación, podemos diferenciar dos casos,

1) Que llegue un nuevo cliente
2) Que se agreguen sitios al actual cliente

En el caso (1), se deben definir VRFs en los “routers” de borde (RoDC01.01 y RoDC02.01), donde se conectan los vínculos de los sitios del nuevo cliente. Dentro de los firewalls se deben definir nuevos contextos, los cuales se deben “conectar” con nuevas VRFs definidas en los “switches” L3 de los Datacenters. A nivel de ruteo estas nuevas VRFs en el “switch” L3 deben establecer vecindad con la VRF en el “router” de borde. De esta manera, es que las redes internas de cada cliente se interconectan, sin tener acceso a la red del otro cliente. Dentro de los “switches” se crean VLANs, las cuales se mapean a distintas redes.
Para la malla de túneles, se deben utilizar túneles GRE conscientes de VRFs [154] para implantar la nueva malla de túneles.

En el caso (2), solamente se debe configurar la dirección IP de la WAN del nuevo “router”, configurar los túneles hacia los Datacenters y publicar en el protocolo de ruteo dinámico las redes que conecta.

En conclusión, es altamente ventajosa la utilización de equipamiento de red con capacidad de virtualización, ya sea desde el punto de vista técnico y económico como se ha visto en el presente trabajo.

10. Conclusiones

Para resolver el problema tal cual como se definió en 1.2, se realizó el estudio del estado del arte en virtualización de redes utilizando técnicas de "networking" tradicionales como introduciendo nuevas herramientas que han sido desarrolladas en el ámbito académico y hoy en día se encuentra en pleno desarrollo en el ámbito comercial. Si analizamos los motivos que impulsan la virtualización de redes (1.1.2) (sin perder de vista los requerimientos del negocio como se establece en 1.1.3), podemos ver que detrás de estos facilitadores, se encuentran motivos que siempre se encuentran en la empresa y que tienen una importancia alta como ser disminuir los costos operativos, simplificar y hacer más eficientes y flexibles los procesos de negocio de la empresa.

Como se ha visto a lo largo de este trabajo, existen muchas tecnologías distintas para la virtualización de los dispositivos de red. Cada día existen nuevas tecnologías cuyo objetivo es abstraer el dispositivo de red. El problema con estas tecnologías (tanto estándar como propietarias) es que existe una infinidad de maneras de utilizarlas para implementar una solución. A lo largo del trabajo se han explorado dentro de cada escenario (LAN, WAN, Datacenter) distintas soluciones para la virtualización las cuales fueron oportunamente discutidas en el capítulo correspondiente y desglosadas como una tabla guía que nos indica el escenario de aplicación de cada una de ellas.

Básicamente existen dos corrientes que se han identificado, una más orientada al dispositivo o tradicional (impulsada por los fabricantes de dispositivos), la cual tiene la gran desventaja que se debe configurar salto por salto ("hop by hop"). Por ejemplo la utilización de VRFs es un modo de lograr la virtualización de redes. Esta tipo de tecnología si bien se encuentra en un estado maduro, es complicada de configurar, desplegar y mantener. A su vez, las aplicaciones que las utilizan deben ser conscientes de la existencia de VRFs para poder funcionar. La mínima unidad que se utiliza para virtualizar tráfico es una subinterface o una VLAN, por lo que no es posible aislar tráfico en el contexto de un flujo.

Por otro lado, SDN con su planteo trae el software al centro de discusión para dar la "flexibilidad" a la red donde los fabricantes de dispositivos pasan a tener un rol más de desarrolladores de software que de fabricantes de hardware. Con SDN se le da a la empresa un rol más activo en el desarrollo de nuevos protocolos ajustados a sus necesidades.
Si bien SDN parece ser lo que se necesita para soporte a la virtualización de redes, es una tecnología (o conjunto de tecnologías) que todavía se encuentra en su infancia ya que no hay una plataforma que cubra todos los casos (como se ve en la Figura 62). Es de esperarse que en los próximos tres años, SDN tenga un crecimiento explosivo ya que según IDC [155] el mercado de SDN para 2016 será de U$ 3.7 billones y los principales fabricantes de equipos tradicionales seguirán desarrollando tecnologías basadas en SDN. Lo que si será necesario es un esfuerzo de estandarización por parte de la industria de las tecnologías SDN. En ese terreno, el proyecto OpenDaylight puede ser la solución ya que muchos fabricantes se han alineado detrás, los cuales tienen la capacidad para definir las bases de nuevos estándares. Sería un despropósito que dentro de tres o cuatro años más, tengamos que hablar de soluciones propietarias SDN (cerradas a un solo fabricante) y no disponer de interfaces abiertas estándar para dialogar con la red.

Como a trabajo a futuro resulta interesante generar guías análogas a las realizadas en este trabajo, pero enfocadas a tecnologías SDN y la implementación de un caso de estudio utilizando una solución de código abierto de SDN como ser el Proyecto OpenDaylight. Como se mencionó en 7.3, se espera que esta solución se convierta en estándar de "facto" ya que varios de los principales fabricantes de equipamiento y "software" de redes contribuyen en desarrollo

de esta solución aportando distintos componentes de la misma [77]. De las soluciones analizadas en este trabajo OpenDaylight es la más completa ya que no solo implementa las interfaces con los dispositivos (mediante OpenFlow u otros protocolos que se pueden adicionar) sino que implementa una interface completa hacia las aplicaciones, como se indica en la Figura 64 y en la Figura 65 la cual nos permite desde programar un dispositivo de red, ejecutar algoritmos sobre la red o tomar datos de rendimiento de la misma hasta orquestar un conjunto de acciones sobre un conjunto de dispositivos. La completitud de esta herramienta, el respaldo de los fabricantes, junto con el hecho de ser de código abierto la hacen ideal para experimentar en futuros proyectos, conocer su interna o colaborar en el desarrollo de la misma.

11. Bibliografía

[1] V. Moreno and Kumar Reddy, Network Virtualization, 1 ed., CiscoPress, 2006, p. 408.

[2] Cisco Systems, «Cisco IOS Configuration Fundamentals Command Reference,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/td/docs/ios/fundamentals/command/reference/cf_book.html. [Último acceso: 01 03 2014].

[3] «Quagga Routing Suite,» [En línea]. Available: http://www.nongnu.org/quagga/index.html. [Último acceso: 30 11 2013].

[4] E. Rosen, P. Psenak y P. Pillay-Esnault, «OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs) - RFC 4577,» 06 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4577. [Último acceso: 30 11 2013].

[5] GNS3.net, «GNS3 Graphical Network Simulator,» [En línea]. Available: http://www.gns3.net/. [Último acceso: 30 11 2013].

[6] M. Arregoces, Data Center Fundamentals, Indianapolis: Cisco Press, 2010.

[7] M. Luca , S. Salam, A. Sajassi, S. Matsushima, M. Bocci y T. Nadeau, «Inter-Chassis Communication Protocol for L2VPN PE Redundancy,» 13 10 2013. [En línea]. Available: http://tools.ietf.org/html/draft-ietf-pwe3-iccp-12. [Último acceso: 23 11 2013].

[8] IEEE, «IEEE 802.1Q - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks— Corrigendum 2: Technical and editorial corrections,» 2 11 2012. [En línea]. Available: http://www.ieee802.org/1/pages/802.1Q.html. [Último acceso: 30 11 2013].

[9] D. Farinaci, T. Li, S. Hanks, D. Meyer y P. Traina, «Generic Routing Encapsulation (GRE) - RFC 2784,» 03 2000. [En línea]. Available: http://tools.ietf.org/html/rfc2784. [Último acceso: 30 11 2013].

[10] S. Kent y K. Seo, «Security Architecture for the Internet Protocol - RFC 4301,» 12 2005. [En línea]. Available: http://tools.ietf.org/html/rfc4301. [Último acceso: 30 11 2013].

[11] R. Housley, «Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP) - RFC 4309,» 12 2005. [En línea]. Available: http://tools.ietf.org/html/rfc4309. [Último acceso: 30 11 2013].

[12] C. Kaufman, «Internet Key Exchange (IKEv2) Protocol - RFC 4306,» 12 2005. [En línea]. Available: http://tools.ietf.org/html/rfc4306. [Último acceso: 30 11 2013].

[13] Cisco Systems, «Virtual Switching System (VSS) Q&A,» [En línea]. Available: http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps9336/prod_qas0900aecd806ed74b.html. [Último acceso: 10 10 2013].

[14] Cisco Systems, «Virtual Switching Systems (VSS),» [En línea]. Available: http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/vss.html. [Último acceso: 01 11 2013].

[15] Cisco Systems, «Data Center Design with Cisco Nexus Switches and Virtual PortChannel: Overview,» [En línea]. Available: http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9670/C07-572831-00_Dsgn_Nexus_vPC_DG.pdf. [Último acceso: 10 11 2013].

[16] Cisco Systems, «Cisco Nexus 7000 Series Switches Command References,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/products-command-reference-list.html. [Último acceso: 30 11 2013].

[17] G. A. Donahue, Network Warrior, Sebastopol: O'Reilly, 2011.

[18] Cisco Systems, «Secure Domain Router Commands on Cisco IOS XR Software,» [En

línea]. Available: http://www.cisco.com/c/en/us/td/docs/ios_xr_sw/iosxr_r3-4/system_management/command/reference/sys_r34/yr34sdr.pdf. [Último acceso: 30 11 2013].

[19] IEEE, «IEEE 802.3ad - Amendment to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications — Aggregation of Multiple Link Segments,» 2000 03 20. [En línea]. [Último acceso: 30 11 2013].

[20] P. Karn, P. Metzger y W. Simpson, «The ESP DES-CBC Transform - RFC 1829,» 08 1995. [En línea]. Available: http://tools.ietf.org/html/rfc1829. [Último acceso: 30 11 2013].

[21] S. Kent y R. Atkinson, «Security Architecture for the Internet Protocol - RFC 2401,» 11 2009. [En línea]. Available: http://tools.ietf.org/html/rfc2401. [Último acceso: 30 11 2013].

[22] H. Orman, «The OAKLEY Key Determination Protocol - RFC 2414,» 11 1998. [En línea]. Available: http://tools.ietf.org/html/rfc2412. [Último acceso: 30 11 2013].

[23] T. Kivinen, B. Swander, A. Huttunen y V. Volpe, «Negotiation of NAT-Traversal in the IKE - RFC 3947,» 01 2005. [En línea]. Available: http://tools.ietf.org/html/rfc3947. [Último acceso: 30 11 2013].

[24] J. Lau, M. Townsley y I. Goyret, «Layer Two Tunneling Protocol Version 3 (L2TPv3) - RFC 3931,» 03 2005. [En línea]. Available: http://tools.ietf.org/html/rfc3931. [Último acceso: 30 11 2013].

[25] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn y B. Palter, «Layer Two Tunneling Protocol "L2TP" - RFC 2661,» 08 1999. [En línea]. Available: http://tools.ietf.org/html/rfc2661. [Último acceso: 30 11 2013].

[26] E. Rosen, A. Viswanathan y R. Callon, «Multiprotocol Label Switching Architecture - RFC 3031,» 01 2001. [En línea]. Available: http://tools.ietf.org/html/rfc3031. [Último acceso: 30 11 2013].

[27] T. D. Nadeau, MPLS Network Management. MIBs, Tools and Tecniques, San Francisco: Morgan Kaufmann, 2003.

[28] L. Andersson, P. Doolan, N. Feldman, A. Fredette y B. Thomas, «LDP Specification - RFC 3036,» 01 2001. [En línea]. Available: http://tools.ietf.org/html/rfc3036. [Último acceso: 30 11 2013].

[29] L. Andersson, T. Minei y B. Thomas, «LDP Specification - RFC 5036,» 10 2007. [En línea]. Available: http://tools.ietf.org/html/rfc5036. [Último acceso: 30 11 2013].

[30] Y. Rekhter, T. Li y S. Hares, «A Border Gateway Protocol 4 (BGP-4) - RFC 4271,» 01 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4271. [Último acceso: 30 11 2013].

[31] R. Braden, L. Zhang, S. Berson, S. Herzog y S. Jamin, «Resource ReSerVation Protocol (RSVP,» 09 1997. [En línea]. Available: http://tools.ietf.org/html/rfc2205. [Último acceso: 30 11 2013].

[32] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan y G. Swallow, «RSVP-TE: Extensions to RSVP for LSP Tunnels - RFC 3209,» 12 2001. [En línea]. Available: https://tools.ietf.org/html/rfc3209. [Último acceso: 30 11 2013].

[33] J. Moy, « OSPF Version 2 - RFC 2328,» 04 1998. [En línea]. Available: http://tools.ietf.org/html/rfc2328. [Último acceso: 30 11 2013].

[34] . D. Maughan, M. Schertler, M. Schneider y J. Turner, «Internet Security Association and Key Management Protocol (ISAKMP) - RFC 4306,» 11 1998. [En línea]. Available: http://tools.ietf.org/html/rfc2408. [Último acceso: 30 11 2013].

[35] J. Luciani, D. Katz, D. Piscitello, B. Cole y N. Doraswamy, «NBMA Next Hop Resolution Protocol (NHRP) - RFC 2332,» 04 1998. [En línea]. Available: http://tools.ietf.org/html/rfc2332. [Último acceso: 30 11 2013].

[36] B. Fox y B. Petri, «NHRP Support for Virtual Private Networks - RFC 2735,» 12 1999. [En línea]. Available: http://tools.ietf.org/html/rfc2735. [Último acceso: 30 11 2013].

[37] E. Rosen y Y. Rekhter, «BGP/MPLS VPNs - RFC 2547,» 03 1999. [En línea]. Available: http://tools.ietf.org/html/rfc2547. [Último acceso: 30 11 2013].

[38] E. Rosen y Y. Rekhter, «BGP/MPLS IP Virtual Private Networks (VPNs) - RFC 4364,» 02 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4364. [Último acceso: 30 11 2013].

[39] P. Marques, R. Bonica, L. Fang, L. Martini , R. Raszuk, K. Patel y J. Guichard, «Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs) - RFC 4684,» 11 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4684. [Último acceso: 30 11 2013].

[40] L. Andersson y R. Asati, «Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field - RFC 5462,» 02 2009. [En línea]. Available: http://tools.ietf.org/html/rfc5462. [Último acceso: 30 11 2013].

[41] Cisco Systems, «Cisco IOS DMVPN,» 2008. [En línea]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/security/dynamic-multipoint-vpn-dmvpn/DMVPN_Overview.pdf. [Último acceso: 30 11 2013].

[42] Cisco Systems, «Dynamic Multipoint IPsec VPNs (Using Multipoint GRE/NHRP to Scale IPsec VPNs),» [En línea]. Available: http://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/41940-dmvpn.html#related. [Último acceso: 30 11 2013].

[43] G. Malkin, «RIP Version 2 - RFC 2453,» 11 1998. [En línea]. Available: https://tools.ietf.org/html/rfc2453. [Último acceso: 30 11 2013].

[44] Cisco Systems, «Cisco Express Forwarding Overview,» [En línea]. Available: http://www.cisco.com/c/en/us/td/docs/ios/12_2/switch/configuration/guide/fswtch_c/xcfcef.html. [Último acceso: 30 11 2013].

[45] R. Aggarwal, M. Townsley y M. Dos Santos, «Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3) - RFC 4719,» 11 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4719. [Último acceso: 30 11 2013].

[46] M. Townsley, C. Pignataro , S. Wainner, T. Seely y J. Young, «Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3 - RFC 4817,» 03 2007. [En línea]. Available: http://tools.ietf.org/html/rfc4817. [Último acceso: 30 11 2013].

[47] C. Pignataro y M. Townsley, «High-Level Data Link Control (HDLC) Frames over Layer 2 Tunneling Protocol, Version 3 (L2TPv3) - RFC 4349,» 02 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4349. [Último acceso: 30 11 2013].

[48] S. Singh, M. Townsley y C. Pignataro, «Asynchronous Transfer Mode (ATM) over Layer 2 Tunneling Protocol Version 3 (L2TPv3) - RFC 4454,» 05 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4454. [Último acceso: 30 11 2013].

[49] M. Townsley, G. Wilkie, S. Booth, S. Bryant y J. Lau, «Frame Relay over Layer 2 Tunneling Protocol Version 3 (L2TPv3) - RFC 4591,» 07 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4591. [Último acceso: 30 11 2013].

[50] Metro Ethernet Forum, «Metro Ethernet Forum,» Metro Ethernet Forum, [En línea]. Available: http://www.metroethernetforum.org/. [Último acceso: 05 10 2013].

[51] K. Kompella y Y. Rekhter, «Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling - RFC 4761,» 01 2007. [En línea]. Available: http://tools.ietf.org/html/rfc4761. [Último acceso: 11 30 2013].

[52] M. Lasserre y V. Kopella, «Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling - RFC 4762,» 01 2007. [En línea]. Available: http://tools.ietf.org/html/rfc4762. [Último acceso: 30 11 2013].

[53] L. Martini, E. Rosen, N. El-Aawar y G. Heron, «Encapsulation Methods for Transport of

Ethernet over MPLS Networks - RFC 4448,» 04 2006. [En línea]. Available: http://tools.ietf.org/html/rfc4448. [Último acceso: 30 11 2013].

[54] C. Rigney, S. Willens, A. Rubens y W. Simpson, «Remote Authentication Dial In User Service (RADIUS) - RFC 2865,» 06 2000. [En línea]. Available: https://tools.ietf.org/html/rfc2865. [Último acceso: 30 11 2013].

[55] T. Bates, R. Chandra, D. Katz y Y. Rekhter, «Multiprotocol Extensions for BGP-4 - RFC 4760,» 01 2007. [En línea]. Available: http://tools.ietf.org/html/rfc4760. [Último acceso: 30 11 2013].

[56] T. Worster, Y. Rekther y E. Rosen, «Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE) - RFC 4023,» 03 2005. [En línea]. Available: http://tools.ietf.org/html/rfc4023. [Último acceso: 30 11 2013].

[57] Y. Rekhter, R. Bonica y E. Roden, «Use of Provider Edge to Provider Edge (PE-PE) Generic Routing Encapsulation (GRE) or IP in BGP/MPLS IP Virtual Private Networks - RFC 4797,» 01 2007. [En línea]. Available: http://tools.ietf.org/html/rfc4797. [Último acceso: 30 11 213].

[58] M. Townsley y T. Seely, «BGP/MPLS IP VPNs over Layer 2 Tunneling Protocol ver 3 - draft-townsley-l3vpn-l2tpv3-01,» 01 2004. [En línea]. Available: http://tools.ietf.org/html/draft-townsley-l3vpn-l2tpv3-01. [Último acceso: 30 11 2013].

[59] Cisco Systems, «Resolve IP Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPSEC,» 14 Enero 2008. [En línea]. Available: http://www.cisco.com/en/US/tech/tk827/tk369/technologies_white_paper09186a00800d6979.shtml. [Último acceso: 23 11 2013].

[60] T. Teras, «OpenNHRP,» 16 10 2007. [En línea]. Available: http://sourceforge.net/projects/opennhrp/. [Último acceso: 30 11 2013].

[61] O. Santos y J. Frahim, Cisco ASA: All-in-One Firewall, IPS, and VPN Adaptive Security Appliance - Cisco ASA Security Contexts, CiscoPress, 2005.

[62] P. Mockapetris, «DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION - RFC 1035,» 11 1987. [En línea]. Available: https://tools.ietf.org/html/rfc1035. [Último acceso: 30 11 2013].

[63] R. Droms, «Dynamic Host Configuration Protocol - RFC 2131,» 03 1997. [En línea]. Available: https://tools.ietf.org/html/rfc2131. [Último acceso: 30 11 2013].

[64] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot y E. Lear, «Address Allocation for Private Internets - RFC 1918,» 08 1996. [En línea]. Available: http://tools.ietf.org/html/rfc1918. [Último acceso: 30 11 2013].

[65] T. Nadeau y K. Gray, SDN Software Defined Networks, Sebastopol, CA: O´Reilly, 2013.

[66] S. Azodolmolky, Software Defined Networking with OpenFlow, Birmingham: Packt Publishing Ltd, 2013.

[67] U. d. Stanford, «ON.LAB,» [En línea]. Available: http://onlab.us/tools.html. [Último acceso: 1 12 2013].

[68] T. Nadeau y . P. Pan, «Software Driven Networks Problem Statement,» IETF, [En línea]. Available: http://tools.ietf.org/search/draft-nadeau-sdn-problem-statement-01.

[69] Big Switch Networks, Inc, «Big Network Controller Is The Network Application Platform For Open SDN | Big Switch Networks, Inc,» [En línea]. Available: http://www.bigswitch.com/products/SDN-Controller. [Último acceso: 10 12 2013].

[70] Cisco Systems, «onePK Use Cases,» Cisco Systems, [En línea]. Available: https://developer.cisco.com/site/tech/networking/sdn/onepk-developer/overview/use-cases/index.gsp. [Último acceso: 30 11 2013].

[71] R. Sherwood, «An Experimenter's Guide to OpenFlow,» 2010. [En línea]. Available: http://www.rob-sherwood.net/GENI-Experimenters-Workshop.ppt. [Último acceso: 10 11

2013].

[72] O. N. Foundation, «OpenFlow™ Conformance Testing Program,» Open Networking Foundation, [En línea]. Available: OpenFlow™ Conformance Testing Program. [Último acceso: 20 12 2013].

[73] Open Networking Foundation, «Northbound Interface Working Group,» Open Networking Foundation, [En línea]. Available: https://www.opennetworking.org/working-groups/northbound-interface. [Último acceso: 12 10 2013].

[74] S. (. Raza y D. (. Lenrow, «Open Networking Foundation North Bound Interface Working Group (NBI-WG) Charter,» 6 10 2013. [En línea]. Available: https://www.opennetworking.org/images/stories/downloads/working-groups/charter-nbi.pdf. [Último acceso: 25 11 2013].

[75] Linux Foundation, «Open Daylight,» [En línea]. Available: http://www.opendaylight.org/. [Último acceso: 30 12 2013].

[76] E. (. Knorr, «OpenDaylight: A big step toward the software-defined data center,» 8 4 2013. [En línea]. Available: http://www.infoworld.com/t/sdn/opendaylight-big-step-toward-the-software-defined-data-center-215960. [Último acceso: 15 11 2013].

[77] Linux Foundation, «Industry Leaders Collaborate on OpenDaylight Project, Donate Key Technologies to Accelerate Software-Defined Networking,» 8 4 2013. [En línea]. Available: http://www.linuxfoundation.org/news-media/announcements/2013/04/industry-leaders-collaborate-opendaylight-project-donate-key. [Último acceso: 24 11 2013].

[78] Open Networking Foundation, «OpenFlow Switch Specification - Version 1.4.0,» 14 10 2013. [En línea]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf. [Último acceso: 10 12 2013].

[79] A. Tootoonchian y Y. Ganjali, «HyperFlow: A Distributed Control Plane for OpenFlow,» [En línea]. Available: https://www.usenix.org/legacy/events/inmwren10/tech/full_papers/Tootoonchian.pdf. [Último acceso: 20 12 2013].

[80] Volkan Yazıcı, Oğuz Sunay y Ali Ö. Ercan, «Controlling a Software-Defined Network via Distributed Controllers,» [En línea]. Available: http://faculty.ozyegin.edu.tr/aliercan/files/2012/10/YaziciNEM12.pdf. [Último acceso: 25 11 2013].

[81] M. McCauley, «NOX - POX,» [En línea]. Available: http://www.noxrepo.org/. [Último acceso: 5 12 2013].

[82] M. Casado, N. McKeown, N. Gude, T. Koponen, J. Pettit, B. Pfaff y S. Shenker, «NOX: Towards an Operating System for Networks,» [En línea]. Available: http://yuba.stanford.edu/~casado/nox-ccr-final.pdf. [Último acceso: 01 12 2013].

[83] openvswitch.org, «Open vSwitch,» [En línea]. Available: http://openvswitch.org/. [Último acceso: 12 11 2013].

[84] D. Erickson, «The Beacon OpenFlow Controller,» [En línea]. Available: http://yuba.stanford.edu/~derickso/docs/hotsdn15-erickson.pdf. [Último acceso: 10 12 2013].

[85] D. Erickson, «Home - Beacon - Confluence,» 4 2 2013. [En línea]. Available: https://openflow.stanford.edu/display/Beacon/Home. [Último acceso: 20 11 2013].

[86] «Documentation « Floodlight OpenFlow ControllerProject Floodlight,» [En línea]. Available: http://www.projectfloodlight.org/documentation/. [Último acceso: 10 12 2013].

[87] OSGi Alliance, «OSGi Alliance | Technology / The OSGi Architecture,» [En línea]. Available: http://www.osgi.org/Technology/WhatIsOSGi. [Último acceso: 10 12 2013].

[88] OpenDaylight, «OpenDaylight Controller:Architectural Framework - Daylight Project,»

[En línea]. Available: https://wiki.opendaylight.org/view/OpenDaylight_Controller:Architectural_Framework. [Último acceso: 10 12 2013].

[89] Cisco Systems, «Cisco XNC FAQ,» Cisco Systems, [En línea]. Available: https://developer.cisco.com/site/tech/networking/sdn/xnc/faq/index.gsp. [Último acceso: 10 12 2013].

[90] Cisco Systems, «Cisco XNC: An Overview,» Cisco Systems, [En línea]. Available: https://developer.cisco.com/site/tech/networking/sdn/xnc/overview/overview/.

[91] «RouteFlow,» [En línea]. Available: https://sites.google.com/site/routeflow/home. [Último acceso: 01 12 2013].

[92] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown y G. Parulkar, «FlowVisor: A Network Virtualization Layer,» 2009. [En línea]. Available: http://archive.openflow.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf. [Último acceso: 15 11 2013].

[93] ON.Lab, «FlowVisor,» ON.Lab, [En línea]. Available: http://onlab.us/flowvisor.html. [Último acceso: 03 01 2014].

[94] Open Networking Foundation, «SDN Product Directory,» Open Networking Foundation, 2013. [En línea]. Available: https://sdndirectory.opennetworking.org/products/. [Último acceso: 20 11 2013].

[95] Cisco Systems, «Benefits of Migrating to Cisco® Catalyst® 3850 and 3650 Switches,» 03 2014. [En línea]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/switches/catalyst-3650-series-switches/feature-comparison-c83-731054.pdf. [Último acceso: 03 2014].

[96] Cisco Systems, «Benefits of Migrating to Cisco® Catalyst® 4500-E Switches with Supervisors 7-LE and 8-E,» Cisco Systems, 3 2014. [En línea]. Available: http://www.cisco.com/c/dam/en/us/products/collateral/switches/catalyst-4500-series-switches/feature-comparison-c83-731050.pdf. [Último acceso: 3 2014].

[97] Project Floodlight, «Indigo,» [En línea]. Available: http://www.projectfloodlight.org/indigo/. [Último acceso: 10 12 2013].

[98] OpenFlow.org, «Pantou : OpenFlow 1.0 for OpenWRT,» OpenFlow.org, [En línea]. Available: http://archive.openflow.org/wk/index.php/Pantou_:_OpenFlow_1.0_for_OpenWRT. [Último acceso: 01 12 2013].

[99] R. Chua, «Controller Wars 2.0 – ON.LAB & Juniper Re-Ignite the Open-Source Battleground. Part 1 of 2,» SDNCentral.com, 21 12 2013. [En línea]. Available: http://www.sdncentral.com/market/controller-onlab-juniper-open-source-sdn-battleground-part1/2013/12/. [Último acceso: 10 01 2014].

[100] SDNCentral.com, «Software Defined Networking Open Source Projects,» [En línea]. Available: http://www.sdncentral.com/comprehensive-list-of-open-source-sdn-projects/. [Último acceso: 15 12 2013].

[101] A. Atlas y E. Crabbe, «Interface to the Routing System (i2rs) - Charter for Working Group,» IETF, [En línea]. Available: https://datatracker.ietf.org/wg/i2rs/charter/.

[102] M. Bocci y B. Schliesser , «Network Virtualization Overlays (nvo3) - Charter for Working Group,» IETF, [En línea]. Available: https://datatracker.ietf.org/wg/nvo3/charter/.

[103] E. Marocco y V. Gurbani , «Application-Layer Traffic Optimization (alto) - Charter for Working Group,» IETF, [En línea]. Available: http://datatracker.ietf.org/wg/alto/charter/.

[104] European Telecommunication Standards Institute - ETSI, «Network Functions Virtualisation,» ETSI, [En línea]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv.

[105] F. (. Gens, «Top 10 Predictions - IDC Predictions 2013 : Competing on the 3rd Platform,» 11 2012. [En línea]. Available: http://www.idc.com/research/Predictions13/downloadable/238044.pdf. [Último acceso: 10 11 2013].

[106] G. Santana, Data Center Virtualization Fundamentals, Indianapolis: Cisco Press, 2013.

[107] IEEE, «IEEE 802.1s - Multiple Spanning Trees,» [En línea]. Available: http://www.ieee802.org/1/pages/802.1s.html. [Último acceso: 30 11 2013].

[108] IEEE, «IEEE 802.1w - Rapid Reconfiguration of Spanning Tree,» 9 6 2004. [En línea]. Available: http://www.ieee802.org/1/pages/802.1w.html. [Último acceso: 30 11 2013].

[109] D. Fedyk, P. Ashwoord-Smith, D. Allan, N. Bragg y P. Unbehagen, «IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging - RFC 6329,» 04 2012. [En línea]. Available: http://tools.ietf.org/html/rfc6329. [Último acceso: 30 11 2013].

[110] J. Touch y R. Perlman, «Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement - RFC 5556,» 05 2009. [En línea]. Available: http://tools.ietf.org/html/rfc5556. [Último acceso: 30 11 2013].

[111] D. Eastlake, A. Banerjee, D. Dutt, R. Perlman y A. Ghanwani, «Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS - RFC 6326,» 07 2011. [En línea]. Available: http://tools.ietf.org/html/rfc6326. [Último acceso: 30 11 2013].

[112] R. Perlman, D. Eastlake 3rd, D. Dutt, S. Gai y A. Ghanwani, «Routing Bridges (RBridges): Base Protocol Specification - RFC 6325,» 07 2011. [En línea]. Available: http://tools.ietf.org/html/rfc6325. [Último acceso: 30 11 2013].

[113] D. Melman, T. Mizrahi y D. Eastlake 3rd, «Fibre Channel over Ethernet (FCoE) over Transparent Interconnection of Lots of Links (TRILL) - RFC 6847,» [En línea].

[114] D. Katz, K. Kompella y D. Yeung, «Traffic Engineering (TE) Extensions to OSPF Version 2 - RFC 3630,» 09 2003. [En línea]. Available: http://tools.ietf.org/html/rfc3630. [Último acceso: 01 12 2013].

[115] P. Ashwood-Smith, D. A. Ericsson, . N. Bragg y P. Unbehagen, «IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging - RFC 6329,» 04 2012. [En línea]. Available: http://tools.ietf.org/html/rfc6329. [Último acceso: 30 11 2013].

[116] R. Perlman, D. Eastlake 3rd, D. Dutt, S. Gai y A. Ghanwani, «Routing Bridges (RBridges): Base Protocol Specification - RFC 6325,» 07 2011. [En línea]. Available: http://tools.ietf.org/html/rfc6325. [Último acceso: 30 11 2013].

[117] R. Perlman, D. Eastlake 3rd, Y. Li, A. Banerjee y F. Hu, «Routing Bridges (RBridges): Appointed Forwarders - RFC 6439,» 11 2011. [En línea]. Available: http://tools.ietf.org/html/rfc6439. [Último acceso: 30 11 2013].

[118] D. Eastlake 3rd, R. Perlman, A. Ghanwani, D. Dutt y V. Manral, «Routing Bridges (RBridges): Adjacency - RFC 6327,» 07 2011. [En línea]. Available: http://tools.ietf.org/html/rfc6327. [Último acceso: 30 11 2013].

[119] A. Banerjee, D. Eastlake, D. Dutt, R. Perlman y A. Ghanwani, «Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS - RFC 6326,» 07 2011. [En línea]. Available: http://tools.ietf.org/html/rfc6326. [Último acceso: 30 11 2013].

[120] J. Touch y R. Perlman, «Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement - RFC 5556,» 05 2009. [En línea]. Available: http://tools.ietf.org/html/rfc5556. [Último acceso: 30 11 2013].

[121] IETF, «Search Internet-Drafts and RFCs (TRILL),» [En línea]. Available: https://datatracker.ietf.org/doc/search/?name=&activedrafts=on&sort=&by=group&group=trill. [Último acceso: 01 11 2013].

[122] R. Van der Pol, «TRILL and IEEE 802.1aq Overview,» 04 2012. [En línea]. Available: http://www.rvdp.org/publications/TRILL-SPB.pdf. [Último acceso: 30 11 2013].

[123] IEEE, «IEEE 802.1aq - Shortest Path Bridging,» 29 06 2012. [En línea]. Available: http://www.ieee802.org/1/pages/802.1aq.html. [Último acceso: 30 11 2013].

[124] D. E. Eastlake, «The IETF TRILL Protocol,» 02 2013. [En línea]. Available: http://conference.apnic.net/__data/assets/pdf_file/0004/58882/trillapricot8_1361288177.pdf. [Último acceso: 30 11 2013].

[125] IEEE, «IEEE 802.1Qaz - Enhanced Transmission Selection,» 2011.

[126] IEEE, «IEEE 802.1Qbb - Priority Based Flow Control,» IEE, 2011.

[127] IEEE, «IEEE 802.Qau - Congestion Notification,» 2010.

[128] IEEE, «IEEE 802.1ab - Station and Media Access Control Connectivity Discover,» 2009.

[129] Cisco Systems, «Understanding and Tuning Spanning Tree Protocol Timers,» 30 1 2006. [En línea]. Available: http://www.cisco.com/en/US/tech/tk389/tk621/technologies_tech_note09186a0080094954.shtml. [Último acceso: 23 11 2013].

[130] IEEE, «IEEE 802.1D - Media Access Control (MAC) Bridges,» 09 06 2004. [En línea]. Available: http://www.ieee802.org/1/pages/802.1D.html. [Último acceso: 10 10 2013].

[131] IETF, «Transparent Interconnection of Lots of Links (trill),» [En línea]. Available: http://datatracker.ietf.org/wg/trill/charter/. [Último acceso: 04 11 2013].

[132] Cisco Systems, «Cisco Overlay Transport Virtualization Technology Introduction and Deployment Considerations,» 17 01 2012. [En línea]. Available: http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DCI/whitepaper/DCI3_OTV_Intro_WP.pdf. [Último acceso: 25 10 2013].

[133] H. Grover, D. Rao, D. Farinacci y V. Moreno, «Overlay Transport Virtualization - draft-hasmit-otv-04,» 23 02 2013. [En línea]. Available: http://tools.ietf.org/id/draft-hasmit-otv-04.txt. [Último acceso: 22 10 2013].

[134] Hewlett Packard, «HP Ethernet Virtual Interconnect and Multitenant Device Context,» 13 08 2012. [En línea]. Available: http://h17007.www1.hp.com/docs/814/factsheet.pdf. [Último acceso: 27 11 2013].

[135] IEEE, «IEEE 802.1AE - Media Access Control (MAC) Security,» 16 08 2006. [En línea]. Available: http://www.ieee802.org/1/pages/802.1ae.html. [Último acceso: 12 05 2013].

[136] IEEE, «IEEE 802.1X - Port-Based Network Access Control,» 05 02 2010. [En línea]. Available: http://www.ieee802.org/1/pages/802.1x-2010.html. [Último acceso: 05 10 2013].

[137] Brocade, «The Evolution of Ethernet Nomenclature,» 01 02 2011. [En línea]. Available: http://www.brocade.com/downloads/documents/technical_briefs/Ethernet_Nomenclature_GA-TB-357.pdf. [Último acceso: 5 12 2013].

[138] IEEE 802.3 Ethernet Working Group, «400 Gigabit Ethernet Call-For-Interest Consensus,» 03 2013. [En línea]. Available: http://www.ieee802.org/3/cfi/0313_1/CFI_01_0313.pdf. [Último acceso: 10 12 2013].

[139] S. Fang, Y. Yu, C. H. Foh y K. M. M. Aung, «A Loss-Free Multipathing Solution for Data Center Network Using Network Using,» *IEEE TRANSACTIONS ON MAGNETICS,,* vol. 49, nº 6, pp. 2723-2730, 2013.

[140] Gartner, «Gartner Says Don't Assume That a Single Converged Data Center Network Is More Efficient Than Two Well-Designed Separate Networks,» 16 03 2010. [En línea]. Available: http://www.gartner.com/newsroom/id/1323313. [Último acceso: 01 12 2013].

[141] COMMSCOPE, «A Comparison of Total Costs of Ownership of 10 Gigabit Network Deployments in the Data Center,» 11 2010. [En línea]. Available: https://www.anixter.com/content/dam/Suppliers/CommScope/Documents/10GbE_TCO_whitepaper.pdf. [Último acceso: 15 12 2013].

[142] «Virtual PC Simulator VPCS,» 04 01 2014. [En línea]. Available: http://sourceforge.net/projects/vpcs/. [Último acceso: 20 01 2014].

[143] Oracle Corp., «VirtualBox,» [En línea]. Available: https://www.virtualbox.org/. [Último acceso: 30 11 2013].

[144] «Wireshark,» [En línea]. Available: http://www.wireshark.org/. [Último acceso: 30 11 2013].

[145] Cisco Systems, «Cisco Adaptive Security Device Manager,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/products/security/adaptive-security-device-manager/index.html. [Último acceso: 30 11 2013].

[146] Cisco Systems, «Cisco 7206 Router,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/products/routers/7206-router/index.html. [Último acceso: 30 11 2013].

[147] Cisco Systems, «Cisco IOS 15.0M,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-15-0m/index.html. [Último acceso: 30 11 2013].

[148] Cisco Systems, «Cisco 3700 Series Multiservice Access Routers,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/products/collateral/routers/3700-series-multiservice-access-routers/product_data_sheet09186a008009203f.html. [Último acceso: 30 11 2013].

[149] Cisco Systems, «Cisco IOS Software Releases 12.4 Mainline,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-software-releases-12-4-mainline/index.html. [Último acceso: 30 11 2013].

[150] Cisco Systems, «Cisco EtherSwitch Modules for the Integrated Services Routers,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/products/collateral/routers/2600-series-multiservice-platforms/product_data_sheet09186a00801aca3e.html. [Último acceso: 30 11 2013].

[151] «QEMU Open Source processor emulator,» [En línea]. Available: http://wiki.qemu.org/Main_Page. [Último acceso: 30 11 2013].

[152] Cisco Systems, «Cisco ASA 5520 Adaptive Security Appliance,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/support/security/asa-5520-adaptive-security-appliance/model.html. [Último acceso: 30 11 2013].

[153] Cisco Systems, «Cisco ASA 5500 Series Configuration Guide using the CLI, 8.4 and 8.6,» Cisco Systems, [En línea]. Available: http://www.cisco.com/c/en/us/td/docs/security/asa/asa84/configuration/guide/asa_84_cli_config.html. [Último acceso: 30 11 2013].

[154] Cisco Systems, «GRE Tunnel with VRF Configuration Example,» Cisco Systems, 19 01 2006. [En línea]. Available: http://www.cisco.com/en/US/tech/tk436/tk428/technologies_configuration_example09186a00801e1294.shtml. [Último acceso: 30 11 2013].

[155] IDC, «IDC Analyze the future,» IDC, [En línea]. Available: http://www.idc.com. [Último acceso: 30 11 2013].

[156] Microsoft, «Path Maximum Transmission Unit (PMTU) Discovery,» Microsoft Technet, 08 04 2010. [En línea]. Available: http://msdn.microsoft.com/en-us/library/aa916746.aspx. [Último acceso: 30 11 2013].

[157] IpFiled.net, «Determine MTU values for each hop with mturoute.exe,» IpFiled.net, [En línea]. Available: http://ipfield.net/2010/07/determine-mtu-values-for-each-hop-with-mturoute-exe/. [Último acceso: 30 11 2013].

[158] Iea Software, «MTU Path - Maximum network path size scan utility,» Iea Software, [En línea]. Available: http://www.iea-software.com/products/mtupath.cfm. [Último acceso: 11

30 2013].

[159] Cisco Systems, «Cisco® ASA 5500-X Series Next-Generation Firewalls – ASA 5512-X, 5515-X, 5525-X, 5545-X and 5555-X,» Cisco Systems, 01 10 2013. [En línea]. Available: http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6032/ps6094/ps6120/at_a_glance_c45-701635.pdf. [Último acceso: 30 11 2013].

[160] A. S. Tanenbaum, Computer Networks, New Jersey: Prentice Hall, 1996.

[161] D. E. Comer y D. L. Stevens, Internetworking with TCP/IP Voulme I / II, New Jersey: Prentice Hall .

[162] S. Kent y R. Atkinson, «IP Authentication Header - RFC 2402,» 11 1998. [En línea]. Available: http://tools.ietf.org/html/rfc2402. [Último acceso: 30 11 2013].

12. Anexos

12.1 Anexo I – Plan de direccionamiento IP del caso de estudio

Plan de direccionamiento IP

El nro de vlan depende de que lado del fw estén ya que funciona como un bridge (L2)

Nombre	Rango IP	Host Disponibles	Not
WAN-MPLS	10.10.230.0/29(248)	230.1 - 230.6	
InterconexionRO-SW-DC1-Red101	10.10.230.8/30(252)	230.9 - 10	Vlan 11 - 1
InterconexionRO-SW-DC1-Red102	10.10.230.12/30(252)	230.13- 14	Vlan 12 - 2
InterconexionRO-SW-DC2-Red101	10.10.229.8/30(252)	229.9 - 10	Vlan 11 - 1
InterconexionRO-SW-DC2-Red102	10.10.229.12/30(252)	229.13- 14	Vlan 12 - 2
DC-Red101	10.10.230.32/27 (224)	230.33 - 62	Vlan 101
DC-Red102	10.10.230.64/26 (192)	230.65 - 126	Vlan 102
Red Tunnel GRE DC1-DC2	10.10.229.32/30 (252)	229.33 -34	Tunel1
Red Tunnel GRE DC1-EmpS1	10.10.229.36/30 (252)	229.37 - 38	Tunel2
Red Tunnel GRE DC1-EmpS2	10.10.229.40/30 (252)	229.41 -42	Tunel3
Red Tunnel GRE DC2-EmpS1	10.10.229.48/30 (252)	229.49 - 50	Tunel5
Red Tunnel GRE DC2-EmpS2	10.10.229.52/30 (252)	229.53 -54	Tunel6
Red Loopbacks Ro-DC1-01, Ro-DC2-01	10.10.229.124/30 (252)	229.125 - 229.126	Loop1, Loop2
Redes de Interconexion EmpS1	10.10.220.24/29 (248)	220.25 - 30	
Redes de Interconexion EmpS2	10.10.220.200/29 (248)	220.201 - 206	

Interconexión Empresa

Red de Interconexion EmpS1	
Red	10.10.220.24/29 (248)
Rango	220.25 - 30
Direcciones	
IP	10.10.220.192/29
10.10.220.24	dirección de red
10.10.220.25	
10.10.220.26	Router Ro-EmpS1-02
10.10.220.27	
10.10.220.28	Router Ro-EmpS1-01
10.10.220.29	
10.10.220.30	
10.10.220.31	broadcast

Red de Interconexion EmpS2	
Red	10.10.220.200/29 (248)
Rango	220.201 - 206
Direcciones	
IP	10.10.220.200/29
10.10.220.200	dirección de red
10.10.220.201	
10.10.220.202	Router Ro-EmpS2-02
10.10.220.203	
10.10.220.204	Router Ro-EmpS2-01
10.10.220.205	
10.10.220.206	
10.10.220.207	broadcast

Red DC01

DC-Red101		
Red	10.10.230.32/27 (224)	
Mascara	255.255.255.224	
Broadcast	10.10.230.63	
Rango	230.33 - 62	
Vlan	101	
Direcciones		
IP	10.10.230.32/27	
10.10.230.33	Switch DC1 Fisica	
10.10.230.34	Switch DC2 Fisica	
10.10.230.35	HSRP activo en DC1	Def gw en equipos DC-Red101 En Dc1
10.10.230.36	HSRP activo en DC2	Def gw en equipos DC-Red101 En Dc2
10.10.230.37		
10.10.230.38		
10.10.230.39		
10.10.230.40		
10.10.230.41	Servidor v101-230.41	Ubicado en DC1
10.10.230.42	Servidor v101-230.42	Ubicado en DC2
10.10.230.43		
10.10.230.44		
10.10.230.45		
10.10.230.46		
10.10.230.47		
10.10.230.48		
10.10.230.49		
10.10.230.50		
10.10.230.51		
10.10.230.52		
10.10.230.53		
10.10.230.54		
10.10.230.55		
10.10.230.56		
10.10.230.57		
10.10.230.58		
10.10.230.59		
10.10.230.60		
10.10.230.61		
10.10.230.62		
10.10.230.63	Broadcast	

Red DC02

DC-Red102		
Red	10.10.230.64/26 (192)	
Mascara	255.255.255.192	
Broadcast	10.10.230.127	
Rango	230.65 - 126	
Vlan	102	
Direcciones		
IP	10.10.230.64/26	
10.10.230.65	Switch DC1 Fisica	
10.10.230.66	Switch DC2 Fisica	
10.10.230.67	HSRP activo en DC1	Def gw en equipos DC-Red102 En Dc1
10.10.230.68	HSRP activo en DC2	Def gw en equipos DC-Red102 En Dc2
10.10.230.69		
10.10.230.70		
10.10.230.71	Servidor v102-230.71	Ubicado en DC1
10.10.230.72	Servidor v102-230.72	Ubicado en DC2
10.10.230.73		
10.10.230.74		
10.10.230.75		
10.10.230.76		
10.10.230.77		
10.10.230.78		
10.10.230.79		
10.10.230.80		
10.10.230.81		
10.10.230.82		
10.10.230.83		
10.10.230.84		
10.10.230.85		
10.10.230.86		
10.10.230.87		
10.10.230.88		
10.10.230.89		
10.10.230.90		
10.10.230.91		
10.10.230.92		
10.10.230.93		
10.10.230.94		
10.10.230.95		
10.10.230.96		
10.10.230.97		
10.10.230.98		
10.10.230.99		
10.10.230.100		
10.10.230.101		
10.10.230.102		
10.10.230.103		
10.10.230.104		
10.10.230.105		
10.10.230.106		
10.10.230.107		
10.10.230.108		
10.10.230.109		
10.10.230.110		
10.10.230.111		
10.10.230.112		
10.10.230.113		
10.10.230.114		
10.10.230.115		
10.10.230.116		
10.10.230.117		
10.10.230.118		
10.10.230.119		
10.10.230.120		
10.10.230.121		
10.10.230.122		
10.10.230.123		
10.10.230.124		
10.10.230.125		
10.10.230.126		
10.10.230.127	Broadcast	

Red Intercoexiónn RO-SW (DC1)

InterconexionRO-SW-DC1-Red101	
Red	10.10.230.8/30(252)
Rango	230.9 - 230.10
Vlan	11
Direcciones	
IP	10.10.230.8/30(252)
10.10.230.9	Ro-DC01-01
10.10.230.10	Sw-DC01-01
10.10.230.11	Broadcast

InterconexionRO-SW-DC1-Red102	
Red	10.10.230.12/30(252)
Rango	230.13 - 230.14
Vlan	12
Direcciones	
IP	10.10.230.12/30(252)
10.10.230.13	Ro-DC01-01
10.10.230.14	Sw-DC01-01
10.10.230.15	Broadcast

Red Intercoexiónn RO-SW (DC2)

InterconexionRO-SW-DC2-Red101	
Red	10.10.229.8/30(252)
Rango	230.9 - 230.10
Vlan	11
Direcciones	
IP	10.10.229.8/30(252)
10.10.229.9	Ro-DC02-01
10.10.229.10	Sw-DC02-01
10.10.229.11	Broadcast

InterconexionRO-SW-DC2-Red102	
Red	10.10.229.12/30(252)
Rango	230.13 - 230.14
Vlan	12
Direcciones	
IP	10.10.229.12/30(252)
10.10.229.13	Ro-DC02-01
10.10.229.14	Sw-DC02-01
10.10.229.15	Broadcast

Red Intercoexiónn Tuneles GRE

Red Tunnel GRE DC1-DC2	
Red	10.10.229.32/30 (252)
Rango	229.33 - 229.34
Vlan	Tunel1
Direcciones	
IP	10.10.229.32/30 (252)
10.10.229.33	Ro-DC01-01
10.10.229.34	Ro-DC02-01
10.10.229.35	Broadcast

Red Tunnel GRE DC1-EmpS1	
Red	10.10.229.36/30 (252)
Rango	230.37 - 230.38
Vlan	Tunel2
Direcciones	
IP	10.10.229.36/30 (252)
10.10.229.37	Ro-DC01-01
10.10.229.38	Router Ro-EmpS1-01
10.10.229.39	Broadcast

Red Tunnel GRE DC2-EmpS1	
Red	10.10.229.48/30 (252)
Rango	229.49 - 230.50
Vlan	Tunel5
Direcciones	
IP	10.10.229.48/30 (252)
10.10.229.49	Ro-DC02-01
10.10.229.50	Router Ro-EmpS1-01
10.10.229.51	Broadcast

Red Tunnel GRE DC1-EmpS2	
Red	10.10.229.40/30 (252)
Rango	229.41 - 229.42
Vlan	Tunel3
Direcciones	
IP	10.10.229.40/30 (252)
10.10.229.41	Ro-DC01-01
10.10.229.42	Router Ro-EmpS2-01
10.10.229.43	Broadcast

Red Tunnel GRE DC2-EmpS2	
Red	10.10.229.52/30 (252)
Rango	229.53 -229.54
Vlan	Tunel6
Direcciones	
IP	10.10.229.52/30 (252)
10.10.229.53	Ro-DC02-01
10.10.229.54	Router Ro-EmpS2-01
10.10.229.55	Broadcast

Red WAN

WAN-MPLS-VPLS	
Red	10.10.230.0/29 (248)
Mascara	255.255.255.248
Broadcast	10.10.230.7
Rango	230.1 - 230.6
Vlan	N/A
Direcciones	
IP	10.10.230.0/29 (248)
10.10.230.1	WAN DC1
10.10.230.2	WAN DC2
10.10.230.3	WAN EmpS1
10.10.230.4	WAN EmpS2
10.10.230.5	Libre
10.10.230.6	Libre
10.10.230.7	BroadCast

12.2 Anexo II – Configuración de la solución del caso de estudio

12.2.1 RoDc01.01

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RoDc01-01
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone UY -3
ip source-route
ip cef
!
no ip domain lookup
ip domain name pg.edu.uy
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
pseudowire-class vlan-xconnect
 encapsulation l2tpv3
 ip local interface Loopback1
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 02-01 address 10.10.230.2
crypto isakmp key 01-03 address 10.10.230.3
crypto isakmp key 04-01 address 10.10.230.4
!
crypto ipsec transform-set TSET esp-des esp-sha-hmac
crypto ipsec df-bit clear
!
crypto ipsec profile VTI
 set security-association lifetime seconds 28800
 set transform-set TSET
!
interface Loopback1
 description Loopback-RoDc01.01
 ip address 10.10.229.125 255.255.255.255
!
interface Tunnel1
 description Tunnel GRE - RoDc01-01 - RoDc01-02
 bandwidth 90000
 ip address 10.10.229.33 255.255.255.252
```

```
 ip tcp adjust-mss 1400
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.1
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.2
 tunnel protection ipsec profile VTI
!
interface Tunnel2
 description Tunnel GRE - RoDc01-01 - RoEmpS01-01
 bandwidth 100000
 ip address 10.10.229.37 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.1
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.3
 tunnel protection ipsec profile VTI
!
interface Tunnel3
 description Tunnel GRE - RoDc01-01 - RoEmpS02-01
 bandwidth 90000
 ip address 10.10.229.41 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.1
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.4
 tunnel protection ipsec profile VTI
!
interface FastEthernet0/0
 description to-WAN
 ip address 10.10.230.1 255.255.255.248
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description to-RoDC01-Xconnect
 no ip address
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1.101
 description Red-101
 encapsulation dot1Q 101
 xconnect 10.10.229.126 101 encapsulation l2tpv3 pw-class vlan-xconnect
!
interface FastEthernet0/1.102
 description Red-102
 encapsulation dot1Q 102
 xconnect 10.10.229.126 102 encapsulation l2tpv3 pw-class vlan-xconnect
!
interface FastEthernet1/0
 description to-RoDC01
```

```
 no ip address
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet1/0.1
 description to_FwDc01-01 ctx v1-v11-Red101
 bandwidth 100000
 encapsulation dot1Q 1
 ip address 10.10.230.9 255.255.255.252
 ip policy route-map clear-df
!
interface FastEthernet1/0.2
 description to_FwDc01-01 ctx v2-v12-Red102
 bandwidth 100000
 encapsulation dot1Q 2
 ip address 10.10.230.13 255.255.255.252
 ip policy route-map clear-df
!
interface FastEthernet1/0.4094
 encapsulation dot1Q 4094 native
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 1
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 passive-interface FastEthernet0/0
 network 10.10.229.125 0.0.0.0 area 0
 network 10.10.229.0 0.0.0.255 area 0
 network 10.10.230.0 0.0.0.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
route-map clear-df permit 10
 match ip address 101
 set ip df 0
!
control-plane
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
```

```
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

12.2.2 RoDc02.01

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RoDc02-01
!
boot-start-marker
boot-end-marker
!
logging queue-limit 100
!
no aaa new-model
!
clock timezone UY -3
ip source-route
ip cef
!
no ip domain lookup
ip domain name pg.edu.uy
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
pseudowire-class vlan-xconnect
 encapsulation l2tpv3
 ip local interface Loopback1
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 02-01 address 10.10.230.1
crypto isakmp key 02-03 address 10.10.230.3
crypto isakmp key 04-02 address 10.10.230.4
!
crypto ipsec transform-set TSET esp-des esp-sha-hmac
crypto ipsec df-bit clear
!
crypto ipsec profile VTI
 set security-association lifetime seconds 28800
 set transform-set TSET
!
interface Loopback1
```

```
 description Loopback-RoDc02.01
 ip address 10.10.229.126 255.255.255.255
!
interface Tunnel1
 description Tunnel GRE - RoDc01-02 - RoDc01-01
 bandwidth 90000
 ip address 10.10.229.34 255.255.255.252
 ip tcp adjust-mss 1400
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.2
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.1
 tunnel protection ipsec profile VTI
!
interface Tunnel5
 description Tunnel GRE - RoDc01-02 - RoEmpS01-01
 bandwidth 90000
 ip address 10.10.229.49 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.2
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.3
 tunnel protection ipsec profile VTI
!
interface Tunnel6
 description Tunnel GRE - RoDc02-01 - RoEmpS02-01
 bandwidth 100000
 ip address 10.10.229.53 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.2
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.4
 tunnel protection ipsec profile VTI
!
interface FastEthernet0/0
 description to-WAN
 ip address 10.10.230.2 255.255.255.248
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description to-RoDC02-Xconnect
 no ip address
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1.11
!
interface FastEthernet0/1.12
!
interface FastEthernet0/1.101
```

```
 description Red-101
 encapsulation dot1Q 101
 xconnect 10.10.229.125 101 encapsulation l2tpv3 pw-class vlan-xconnect
!
interface FastEthernet0/1.102
 description Red-102
 encapsulation dot1Q 102
 xconnect 10.10.229.125 102 encapsulation l2tpv3 pw-class vlan-xconnect
!
interface FastEthernet1/0
 description to-RoDC01
 no ip address
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet1/0.11
 description Int-v101
 bandwidth 100000
 encapsulation dot1Q 11
 ip address 10.10.229.9 255.255.255.252
 ip policy route-map clear-df
!
interface FastEthernet1/0.12
 description Int-v102
 bandwidth 100000
 encapsulation dot1Q 12
 ip address 10.10.229.13 255.255.255.252
 ip policy route-map clear-df
!
interface FastEthernet1/0.4094
 encapsulation dot1Q 4094 native
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 1
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 passive-interface FastEthernet0/0
 network 10.10.229.126 0.0.0.0 area 0
 network 10.10.229.0 0.0.0.255 area 0
 network 10.10.230.0 0.0.0.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
route-map clear-df permit 10
 match ip address 101
 set ip df 0
!
control-plane
```

```
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
dial-peer cor custom
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

12.2.3 RoEmpS01.01

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RoEmpS01-01
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone UY -3
ip source-route
ip cef
!
no ip domain lookup
ip domain name pg.edu.uy
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
track 1 ip sla 10 reachability
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 01-03 address 10.10.230.1
crypto isakmp key 02-03 address 10.10.230.2
```

```
!
crypto ipsec transform-set TSET esp-des esp-sha-hmac
crypto ipsec df-bit clear
!
crypto ipsec profile VTI
 set security-association lifetime seconds 28800
 set transform-set TSET
!
interface Tunnel2
 description Tunnel GRE - RoEmpS01-01 - RoDc01-01
 bandwidth 100000
 ip address 10.10.229.38 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.3
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.1
 tunnel protection ipsec profile VTI
!
interface Tunnel5
 description Tunnel GRE - RoEmpS01-01 - RoDc01-02
 bandwidth 90000
 ip address 10.10.229.50 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.3
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.2
 tunnel protection ipsec profile VTI
!
interface FastEthernet0/0
 description to-WAN
 ip address 10.10.230.3 255.255.255.248
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description to-integ-Ent
 ip address 10.10.220.28 255.255.255.248
 ip policy route-map clear-df
 load-interval 30
 duplex auto
 speed auto
!
router ospf 1
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute static
 passive-interface FastEthernet0/0
 passive-interface FastEthernet0/1
 network 10.10.220.24 0.0.0.7 area 0
 network 10.10.229.0 0.0.0.255 area 0
 network 10.10.230.0 0.0.0.255 area 0
 default-information originate metric-type 1
!
```

```
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 FastEthernet0/1 10.10.220.26 track 1
!
ip sla 10
 icmp-echo 10.10.220.26 source-interface FastEthernet0/1
 frequency 10
ip sla schedule 10 life forever start-time now
!
route-map clear-df permit 10
 match ip address 101
 set ip df 0
!
control-plane
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
event manager applet shut-f0/1-t2
 event track 12 state down
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "shutdown"
event manager applet no-shut-f0/1-t2
 event track 12 state up
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "no shutdown"
event manager applet shut-f0/1-t5
 event track 15 state down
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "shutdown"
event manager applet no-shut-f0/1-t5
 event track 15 state up
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "no shutdown"
```

```
!
end
```

12.2.4 RoEmpS01.02

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RoEmpS01-02
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone UY -3
ip source-route
ip cef
!
no ip domain lookup
ip domain name pg.edu.uy
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
track 1 ip sla 10 reachability
!
interface FastEthernet0/0
 description to_R3
 ip address 10.10.220.26 255.255.255.248
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description to_RoEntS2
 ip address 10.10.0.1 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet1/0
 description to_SwEntS1
 no ip address
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet1/0.10
 description to_SwEntS1_v10
 encapsulation dot1Q 10
```

```
 ip address 10.10.10.1 255.255.255.0
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 1
 log-adjacency-changes
 passive-interface FastEthernet0/0
 network 10.10.0.0 0.0.0.3 area 0
 network 10.10.10.0 0.0.0.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip route 10.10.229.0 255.255.255.0 FastEthernet0/0 10.10.220.28 track 1
ip route 10.10.230.0 255.255.255.0 FastEthernet0/0 10.10.220.28 track 1
ip route 10.10.229.0 255.255.255.0 10.10.0.2 10
ip route 10.10.230.0 255.255.255.0 10.10.0.2 10
!
ip sla 10
 icmp-echo 10.10.220.28 source-interface FastEthernet0/0
 frequency 10
ip sla schedule 10 life forever start-time now
!
control-plane
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

12.2.5 RoEmpS02.01

```
!
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
```

```
hostname RoEmpS02-01
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone UY -3
ip source-route
ip cef
!
no ip domain lookup
ip domain name pg.edu.uy
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
track 2 ip sla 20 reachability
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 group 2
crypto isakmp key 04-01 address 10.10.230.1
crypto isakmp key 04-02 address 10.10.230.2
!
crypto ipsec transform-set TSET esp-des esp-sha-hmac
crypto ipsec df-bit clear
!
crypto ipsec profile VTI
 set security-association lifetime seconds 28800
 set transform-set TSET
!
interface Tunnel3
 description Tunnel GRE - RoEmpS02-01 - RoDc01-01
 bandwidth 90000
 ip address 10.10.229.42 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.4
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.1
 tunnel protection ipsec profile VTI
!
interface Tunnel6
 description Tunnel GRE - RoEmpS02-01 - RoDc02-01
 bandwidth 100000
 ip address 10.10.229.54 255.255.255.252
 load-interval 30
 keepalive 5 3
 tunnel source 10.10.230.4
 tunnel mode ipsec ipv4
 tunnel destination 10.10.230.2
 tunnel protection ipsec profile VTI
```

```
!
interface FastEthernet0/0
 description to-WAN
 ip address 10.10.230.4 255.255.255.248
 load-interval 30
 duplex auto
 speed aut
!
interface FastEthernet0/1
 description to-integ-Ent
 ip address 10.10.220.204 255.255.255.248
 ip policy route-map clear-df
 load-interval 30
 duplex auto
 speed auto
!
router ospf 1
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 passive-interface FastEthernet0/0
 passive-interface FastEthernet0/1
 network 10.10.220.200 0.0.0.7 area 0
 network 10.10.229.0 0.0.0.255 area 0
 network 10.10.230.0 0.0.0.255 area 0
 default-information originate metric-type 1
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 FastEthernet0/1 10.10.220.202 track 2
!
ip sla 20
 icmp-echo 10.10.220.202 source-interface FastEthernet0/1
 frequency 10
ip sla schedule 20 life forever start-time now
!
route-map clear-df permit 10
 match ip address 101
 set ip df 0
!
control-plane
!
mgcp fax t38 ecm
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
```

```
 login
!
event manager applet shut-f0/1-t3
 event track 13 state down
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "shutdown"
event manager applet no-shut-f0/1-t3
 event track 13 state up
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "no shutdown"
event manager applet shut-f0/1-t6
 event track 16 state down
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "shutdown"
event manager applet no-shut-f0/1-t6
 event track 16 state up
 action 05 cli command "enable"
 action 10 cli command "conf t"
 action 20 cli command "interface f0/1"
 action 30 cli command "no shutdown"
!
end
```

12.2.6 RoEmpS02.02

```
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname RoEmpS02-02
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
clock timezone UY -3
ip source-route
ip cef
!
no ip domain lookup
ip domain name pg.edu.uy
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
```

```
track 2 ip sla 20 reachability
!
interface FastEthernet0/0
 description to_R4
 ip address 10.10.220.202 255.255.255.248
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description to_RoEntS1
 ip address 10.10.0.2 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet1/0
 description to_SwEntS2
 no ip address
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet1/0.10
 description to_SwEntS1_v10
 encapsulation dot1Q 20
 ip address 10.10.20.1 255.255.255.0
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
router ospf 1
 log-adjacency-changes
 passive-interface FastEthernet0/0
 network 10.10.0.0 0.0.0.3 area 0
 network 10.10.20.0 0.0.0.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip route 10.10.229.0 255.255.255.0 FastEthernet0/0 10.10.220.204 track 2
ip route 10.10.230.0 255.255.255.0 FastEthernet0/0 10.10.220.204 track 2
ip route 10.10.229.0 255.255.255.0 10.10.0.1 10
ip route 10.10.230.0 255.255.255.0 10.10.0.1 10
!
ip sla 20
 icmp-echo 10.10.220.204 source-interface FastEthernet0/0
 frequency 10
ip sla schedule 20 life forever start-time now
!
control-plane
!
mgcp fax t38 ecm
```

```
mgcp behavior g729-variants static-pt
!
gatekeeper
 shutdown
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login
!
end
```

12.2.7 SwDc01.01

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SwDc01-01
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
clock timezone UY -3
ip cef
!
ip vrf Red-101
 rd 101:0
!
ip vrf Red-102
 rd 102:0
!
no ip domain lookup
ip domain name pg.fing.edu.uy
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
ip ssh version 2
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
```

```
 speed auto
!
interface FastEthernet1/0
 description to-Ro01
 switchport trunk allowed vlan 1,2,11,12,1002-1005
 switchport mode trunk
 load-interval 30
!
interface FastEthernet1/1
 description to-Ro01-XC
 switchport trunk allowed vlan 1,2,101,102,1002-1005
 switchport mode trunk
 load-interval 30
!
interface FastEthernet1/2
!
interface FastEthernet1/3
!
interface FastEthernet1/4
!
interface FastEthernet1/5
!
interface FastEthernet1/6
!
interface FastEthernet1/7
!
interface FastEthernet1/8
!
interface FastEthernet1/9
!
interface FastEthernet1/10
!
interface FastEthernet1/11
 description Vlan101
 switchport access vlan 101
!
interface FastEthernet1/12
 description Vlan102
 switchport access vlan 102
!
interface FastEthernet1/13
 description Vlan101 - PC
 switchport access vlan 101
!
interface FastEthernet1/14
!
interface FastEthernet1/15
!
interface Vlan1
 no ip address
!
interface Vlan11
 description Int-ROSW-Red101
 bandwidth 100000
 ip vrf forwarding Red-101
 ip address 10.10.230.10 255.255.255.252
```

```
!
interface Vlan12
 description Int-ROSW-Red102
 bandwidth 100000
 ip vrf forwarding Red-102
 ip address 10.10.230.14 255.255.255.252
!
interface Vlan101
 description Red-101
 bandwidth 90000
 ip vrf forwarding Red-101
 ip address 10.10.230.33 255.255.255.224
 standby 11 ip 10.10.230.36
 standby 101 ip 10.10.230.35
 standby 101 priority 101
 standby 101 preempt
!
interface Vlan102
 description Red-102
 bandwidth 90000
 ip vrf forwarding Red-102
 ip address 10.10.230.65 255.255.255.192
 standby 12 ip 10.10.230.68
 standby 102 ip 10.10.230.67
 standby 102 priority 101
 standby 102 preempt
!
router ospf 11 vrf Red-101
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network 0.0.0.0 255.255.255.255 area 0
!
router ospf 12 vrf Red-102
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
control-plane
!
mgcp behavior g729-variants static-pt
!
line con 0
 exec-timeout 0 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
!
end
```

12.2.8 SwDc02.01

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SwDc02-01
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
clock timezone UY -3
ip cef
!
ip vrf Red-101
 rd 101:0
!
ip vrf Red-102
 rd 102:0
!
no ip domain lookup
ip domain name pg.fing.edu.uy
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
ip ssh version 2
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/0
 description to-Ro01
 switchport mode trunk
 load-interval 30
!
interface FastEthernet1/1
 description to-Ro01-XC
 switchport mode trunk
 load-interval 30
!
interface FastEthernet1/2
!
interface FastEthernet1/3
!
interface FastEthernet1/4
```

```
!
interface FastEthernet1/5
!
interface FastEthernet1/6
!
interface FastEthernet1/7
!
interface FastEthernet1/8
!
interface FastEthernet1/9
!
interface FastEthernet1/10
 description PC
 switchport access vlan 101
!
interface FastEthernet1/11
 description Vlan101
 switchport access vlan 101
!
interface FastEthernet1/12
 description Vlan102
 switchport access vlan 102
!
interface FastEthernet1/13
!
interface FastEthernet1/14
!
interface FastEthernet1/15
!
interface Vlan1
 no ip address
!
interface Vlan11
 description Int-ROSW-Red101
 bandwidth 100000
 ip vrf forwarding Red-101
 ip address 10.10.229.10 255.255.255.252
!
interface Vlan12
 description Int-ROSW-Red102
 bandwidth 100000
 ip vrf forwarding Red-102
 ip address 10.10.229.14 255.255.255.252
!
interface Vlan101
 description Red-101
 bandwidth 90000
 ip vrf forwarding Red-101
 ip address 10.10.230.34 255.255.255.224
 standby 11 ip 10.10.230.36
 standby 11 priority 101
 standby 11 preempt
 standby 101 ip 10.10.230.35
!
interface Vlan102
 description Red-102
```

```
 bandwidth 90000
 ip vrf forwarding Red-102
 ip address 10.10.230.66 255.255.255.192
 standby 12 ip 10.10.230.68
 standby 12 priority 101
 standby 12 preempt
 standby 102 ip 10.10.230.67
!
router ospf 11 vrf Red-101
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network 0.0.0.0 255.255.255.255 area 0
!
router ospf 12 vrf Red-102
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network 0.0.0.0 255.255.255.255 area 0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
mac-address-table static 0000.0c07.ac0c interface FastEthernet1/1 vlan 102
!
control-plane
!
mgcp behavior g729-variants static-pt
!
line con 0
 exec-timeout 0 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
end
```

12.2.9 FwDC01-01

12.2.9.1 Contexto System

```
firewall transparent
hostname FwDC01-01
enable password 8Ry2YjIyt7RRXU24 encrypted
no mac-address auto
!
interface GigabitEthernet0
 description Inside
!
interface GigabitEthernet0.11
 description Inside11
 vlan 11
!
interface GigabitEthernet0.12
 description Inside12
 vlan 12
```

```
!
interface GigabitEthernet1
 description Outside
!
interface GigabitEthernet1.1
 description Outside1
 vlan 1
!
interface GigabitEthernet1.2
 description Outside2
 vlan 2
!
interface GigabitEthernet2
 description Management
!
interface GigabitEthernet3
 shutdown
!
interface GigabitEthernet4
 shutdown
!
interface GigabitEthernet5
 shutdown
!
class default
  limit-resource All 0
  limit-resource Mac-addresses 65535
  limit-resource ASDM 5
  limit-resource SSH 5
  limit-resource Telnet 5
!
ftp mode passive
pager lines 24
no failover
no asdm history enable
arp timeout 14400
console timeout 0

admin-context admin
context admin
  config-url disk0:/admin.cfg
!
context Red101
  allocate-interface GigabitEthernet0.11 INSIDE
  allocate-interface GigabitEthernet1.1 OUTSIDE
  config-url disk0:/Red101.cfg
!
context Red102
  allocate-interface GigabitEthernet0.12 INSIDE
  allocate-interface GigabitEthernet1.2 OUTSIDE
  config-url disk0:/Red102.cfg
!
prompt hostname context
no call-home reporting anonymous
crashinfo save disable
```

12.2.9.2 Contexto Admin

```
firewall transparent
hostname ciscoasa
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
names
!
pager lines 24
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
user-identity default-domain LOCAL
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart warmstart
telnet timeout 5
ssh timeout 5
no threat-detection statistics tcp-intercept
```

12.2.9.3 Contexto Red101

```
firewall transparent
hostname Red101
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
names
!
interface BVI1
 ip address 1.1.1.1 255.255.255.0
!
interface INSIDE
 description Inside
 nameif inside
 bridge-group 1
 security-level 100
!
interface OUTSIDE
 description Outside
 nameif outside
 bridge-group 1
 security-level 0
!
access-list OUTSIDE-IN extended permit ospf any any
access-list OUTSIDE-IN extended permit icmp any any log
access-list OUTSIDE-IN extended permit icmp any any echo-reply log
access-list OUTSIDE-IN extended permit ip any any log
pager lines 24
mtu inside 1500
mtu outside 1500
```

```
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
access-group OUTSIDE-IN in interface outside
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
user-identity default-domain LOCAL
http server enable
http 0.0.0.0 0.0.0.0 inside
no snmp-server location
no snmp-server contact
telnet timeout 5
ssh 0.0.0.0 0.0.0.0 inside
ssh timeout 5
no threat-detection statistics tcp-intercept
!
class-map inspection_default
 match default-inspection-traffic
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect ip-options
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
!
service-policy global_policy global
```

12.2.9.4 Contexto Red102

```
firewall transparent
hostname Red102
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
names
!
```

```
interface BVI1
 ip address 2.2.2.2 255.255.255.0
!
interface INSIDE
 description Inside
 nameif inside
 bridge-group 1
 security-level 100
!
interface OUTSIDE
 description Outside
 nameif outside
 bridge-group 1
 security-level 0
!
access-list OUTSIDE-IN extended permit ospf any any
access-list OUTSIDE-IN extended permit icmp any any log
access-list OUTSIDE-IN extended permit udp any any
access-list OUTSIDE-IN extended permit ip any any log
access-list OUTSIDE-IN extended permit icmp any any echo-reply log
pager lines 24
mtu inside 1500
mtu outside 1500
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
access-group OUTSIDE-IN in interface outside
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
user-identity default-domain LOCAL
http server enable
http 0.0.0.0 0.0.0.0 inside
no snmp-server location
no snmp-server contact
telnet timeout 5
ssh 0.0.0.0 0.0.0.0 inside
ssh timeout 5
no threat-detection statistics tcp-intercept
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
```

```
  inspect h323 h225
  inspect h323 ras
  inspect ip-options
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
!
service-policy global_policy global
```

12.3 Anexo III – Casos de Test del caso de estudio

12.3.1 Pruebas básicas de conectividad

12.3.1.1 Prueba 1

Descripción:
Desde los equipos instalados en DC01 verificar que sean alcanzables los equipos en DC02, EmpS01 y EmpS02

Desde 10.10.230.41

Destino	Red	Sitio
10.10.230.71	Red-102	DC01
10.10.230.42	Red-101	DC02
10.10.230.72	Red-102	DC02
10.10.10.2	Red-E01	Emp01
10.10.20.2	Red-E02	Emp02

Desde 10.10.230.71

Destino	Red	Sitio
10.10.230.41	Red-101	DC01
10.10.230.42	Red-101	DC02
10.10.230.72	Red-102	DC02
10.10.10.2	Red-E01	Emp01
10.10.20.2	Red-E02	Emp02

Resultado esperado
Conectividad exitosa

Resultado Obtenido
Conectividad exitosa

12.3.1.2 Prueba 2

Descripción:
Desde los equipos instalados en DC02 verificar que sean alcanzables los equipos en DC01, EmpS01 y EmpS02

Desde 10.10.230.42

Destino	Red	Sitio
10.10.230.41	Red-101	DC01
10.10.230.71	Red-102	DC01
10.10.230.72	Red-102	DC02
10.10.10.2	Red-E01	Emp01
10.10.20.2	Red-E02	Emp02

Desde 10.10.230.72

Destino	Red	Sitio
10.10.230.41	Red-101	DC01
10.10.230.71	Red-102	DC01
10.10.230.42	Red-101	DC02
10.10.10.2	Red-E01	Emp01
10.10.20.2	Red-E02	Emp02

Resultado esperado
Conectividad exitosa

Resultado Obtenido
Conectividad exitosa

12.3.1.3 Prueba 3

Descripción:
Desde un equipo instalado en DC01 verificar el camino que se utiliza para alcanzar los equipos en DC01, DC02, EmpS01 y EmpS02. Utilizando el utilitario "traceroute" desde el dispositivo con dirección IP 10.10.230.41

Traceroute 10.10.230.71

```
230.41[1]> trace 10.10.230.71
trace to 10.10.230.71, 8 hops max, press Ctrl+C to stop
 1   10.10.230.33   26.000 ms  9.000 ms  19.000 ms
 2   10.10.230.9   83.000 ms  62.000 ms  31.000 ms
 3   10.10.230.14   41.000 ms  50.000 ms  32.000 ms
 4   *10.10.230.71   193.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.230.42

```
230.41[1]> trace 10.10.230.42
trace to 10.10.230.42, 8 hops max, press Ctrl+C to stop
 1   *10.10.230.42   30.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.230.72

```
       230.41[1]> trace 10.10.230.72
trace to 10.10.230.72, 8 hops max, press Ctrl+C to stop
 1   10.10.230.33   18.000 ms  10.000 ms  20.000 ms
 2   10.10.230.9   59.000 ms  61.000 ms  127.000 ms
 3   10.10.230.14   54.000 ms  42.000 ms  54.000 ms
 4    *10.10.230.72    118.000 ms (ICMP type:3, code:3, Destination port unreachable)
22.000 ms
```

Traceroute 10.10.10.2

```
230.41[1]> trace 10.10.10.2
trace to 10.10.10.2, 8 hops max, press Ctrl+C to stop
 1   10.10.230.33   19.000 ms  33.000 ms  22.000 ms
 2   10.10.230.9   88.000 ms  57.000 ms  22.000 ms
 3   10.10.229.38   65.000 ms  118.000 ms  94.000 ms
 4   10.10.220.26   139.000 ms  128.000 ms  105.000 ms
 5   *10.10.10.2   88.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.20.2

```
230.41[1]> trace 10.10.20.2
trace to 10.10.20.2, 8 hops max, press Ctrl+C to stop
 1   10.10.230.33   10.000 ms  85.000 ms  10.000 ms
 2   10.10.230.9   69.000 ms  29.000 ms  164.000 ms
 3   10.10.229.38   75.000 ms  293.000 ms  124.000 ms
 4   10.10.220.26   223.000 ms  132.000 ms  142.000 ms
 5   10.10.0.2   183.000 ms  90.000 ms  140.000 ms
 6   *10.10.20.2   99.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

12.3.1.4 Prueba 4

Descripción:
Desde un equipo instalado en DC02 verificar el camino que se utiliza para alcanzar los equipos en DC01, DC02, EmpS01 y EmpS02. Utilizando el utilitario “traceroute” desde el dispositivo con dirección IP 10.10.230.72.

Traceroute 10.10.230.41

```
230.72[4]> trace 10.10.230.41
trace to 10.10.230.41, 8 hops max, press Ctrl+C to stop
 1   10.10.230.65   63.000 ms  114.000 ms  55.000 ms
 2   10.10.230.13   120.000 ms  143.000 ms  101.000 ms
 3   10.10.230.10   89.000 ms  69.000 ms  89.000 ms
 4   *10.10.230.41   79.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.230.71

```
230.72[4]> trace 10.10.230.71
trace to 10.10.230.71, 8 hops max, press Ctrl+C to stop
 1   *10.10.230.71   37.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.230.42

```
230.72[4]> trace 10.10.230.42
trace to 10.10.230.42, 8 hops max, press Ctrl+C to stop
 1   10.10.230.65   69.000 ms  50.000 ms  34.000 ms
 2   10.10.230.13   113.000 ms  89.000 ms  155.000 ms
 3   10.10.230.10   85.000 ms  81.000 ms  103.000 ms
 4   *10.10.230.42   136.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.10.2

```
230.72[4]> trace 10.10.10.2
trace to 10.10.10.2, 8 hops max, press Ctrl+C to stop
 1   10.10.230.65   83.000 ms  31.000 ms  31.000 ms
 2   10.10.230.13   89.000 ms  99.000 ms  130.000 ms
 3   10.10.229.38   118.000 ms  127.000 ms  91.000 ms
 4   10.10.220.26   118.000 ms  109.000 ms  108.000 ms
 5   *10.10.10.2   125.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.20.2

```
230.72[4]> trace 10.10.20.2
trace to 10.10.20.2, 8 hops max, press Ctrl+C to stop
 1   10.10.230.65   85.000 ms  60.000 ms  40.000 ms
 2   10.10.230.13   136.000 ms  134.000 ms  141.000 ms
 3   10.10.229.38   128.000 ms  108.000 ms  74.000 ms
 4   10.10.220.26   200.000 ms  112.000 ms  122.000 ms
 5   10.10.0.2   137.000 ms  125.000 ms  133.000 ms
 6   *10.10.20.2   106.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

12.3.1.5 Prueba 5

Descripción:
Desde un equipo instalado en un sitio de la empresa (EmpS01) verificar el camino que se utiliza para alcanzar los equipos en DC01, DC02 y EmpS02. Utilizando el utilitario “traceroute” desde el dispositivo con dirección IP 10.10.10.2.

Traceroute 10.10.230.41

```
trace to 10.10.230.41, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   32.000 ms  44.000 ms  12.000 ms
 2   10.10.220.28   76.000 ms  40.000 ms  25.000 ms
 3   10.10.229.37   160.000 ms  102.000 ms  127.000 ms
 4   10.10.229.34   152.000 ms  191.000 ms  218.000 ms
 5   10.10.229.10   125.000 ms  92.000 ms  68.000 ms
 6   *10.10.230.41   216.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.230.71

```
10.2[5]> trace 10.10.230.71
trace to 10.10.230.71, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   41.000 ms  19.000 ms  10.000 ms
 2   10.10.220.28   42.000 ms  42.000 ms  21.000 ms
 3   10.10.229.37   90.000 ms  109.000 ms  80.000 ms
 4   10.10.230.14   83.000 ms  67.000 ms  79.000 ms
 5   *10.10.230.71   140.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.230.42

```
10.2[5]> trace 10.10.230.42
trace to 10.10.230.42, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   55.000 ms  12.000 ms  11.000 ms
 2   10.10.220.28   58.000 ms  47.000 ms  23.000 ms
 3   10.10.229.37   90.000 ms  82.000 ms  98.000 ms
 4   10.10.230.10   111.000 ms  89.000 ms  79.000 ms
 5   *10.10.230.42   141.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.230.72

```
10.2[5]> trace 10.10.230.72
trace to 10.10.230.72, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   75.000 ms  4.000 ms  10.000 ms
 2   10.10.220.28   54.000 ms  22.000 ms  21.000 ms
 3   10.10.229.37   137.000 ms  95.000 ms  54.000 ms
 4   10.10.230.14   97.000 ms  74.000 ms  85.000 ms
 5   *10.10.230.72   122.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Traceroute 10.10.20.2

```
10.2[5]> trace 10.10.20.2
trace to 10.10.20.2, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   159.000 ms  19.000 ms  10.000 ms
 2   10.10.0.2   89.000 ms  43.000 ms  50.000 ms
 3   *10.10.20.2   86.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

12.3.2 Pruebas de alta disponibilidad

12.3.2.1 Prueba 1

Descripción
Se apaga en “switch” correspondiente ubicado en DC01

Resultado esperado:

Se espera la caída en la ip hsrp de las vlans activas en DC01
Solamente contestan las ip's configuradas en el equipamiento situado en DC02 y las ip's hsrp activas en DC02

Resultado obtenido:
Según lo esperado

```
SwDc02-01>
*Mar  1 01:47:39.171: %HSRP-5-STATECHANGE: Vlan101 Grp 101 state Standby -> Active
*Mar  1 01:47:39.187: %HSRP-5-STATECHANGE: Vlan102 Grp 102 state Standby -> Active
SwDc02-01>
*Mar  1 01:48:07.907: %OSPF-5-ADJCHG: Process 11, Nbr 10.10.230.33 on Vlan101 from FULL to
DOWN, Neighbor Down: Dead timer expired
*Mar  1 01:48:07.919: %OSPF-5-ADJCHG: Process 12, Nbr 10.10.230.65 on Vlan102 from FULL to
DOWN, Neighbor Down: Dead timer expired
SwDc02-01>

RoDc01-01#
*Jan 22 00:15:47.511: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.230.65 on FastEthernet1/0.2 from
FULL to DOWN, Neighbor Down: Dead timer expired
*Jan 22 00:15:47.591: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.230.33 on FastEthernet1/0.1 from
FULL to DOWN, Neighbor Down: Dead timer expired

10.10.10.2[5]> ping 10.10.230.72
10.10.230.72 icmp_seq=1 ttl=59 time=119.000 ms
10.10.230.72 icmp_seq=2 ttl=59 time=71.000 ms
10.10.230.72 icmp_seq=3 ttl=59 time=64.000 ms
10.10.230.72 icmp_seq=4 ttl=59 time=111.000 ms
10.10.230.72 icmp_seq=5 ttl=59 time=98.000 ms

10.10.10.2[5]> ping 10.10.230.42
10.10.230.42 icmp_seq=1 ttl=59 time=83.000 ms
10.10.230.42 icmp_seq=2 ttl=59 time=81.000 ms
10.10.230.42 icmp_seq=3 ttl=59 time=59.000 ms
10.10.230.42 icmp_seq=4 ttl=59 time=90.000 ms
10.10.230.42 icmp_seq=5 ttl=59 time=80.000 ms

10.10.10.2[5]> ping 10.10.230.41
10.10.230.41 icmp_seq=1 timeout
10.10.230.41 icmp_seq=2 timeout
10.10.230.41 icmp_seq=3 timeout
10.10.230.41 icmp_seq=4 timeout
10.10.230.41 icmp_seq=5 timeout

10.10.10.2[5]> ping 10.10.230.71
10.10.230.71 icmp_seq=1 timeout
10.10.230.71 icmp_seq=2 timeout
10.10.230.71 icmp_seq=3 timeout
10.10.230.71 icmp_seq=4 timeout
10.10.230.71 icmp_seq=5 timeout

SwDc02-01#sh ip vrf
  Name                             Default RD          Interfaces
  Red-101                          101:0               Vl11
                                                       Vl101
  Red-102                          102:0               Vl12
```

```
                                                    Vl102
SwDc02-01#sh ip route vrf Red-101

Routing Table: Red-101
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.10.229.9 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 16 subnets, 5 masks
O       10.10.220.200/29 [110/30] via 10.10.229.9, 00:04:29, Vlan11
O       10.10.229.125/32 [110/22] via 10.10.229.9, 00:04:29, Vlan11
O       10.10.229.126/32 [110/11] via 10.10.229.9, 00:04:29, Vlan11
O       10.10.230.64/26 [110/31] via 10.10.229.9, 00:04:29, Vlan11
O       10.10.229.36/30 [110/31] via 10.10.229.9, 00:04:29, Vlan11
O       10.10.220.24/29 [110/31] via 10.10.229.9, 00:04:29, Vlan11
O       10.10.229.32/30 [110/21] via 10.10.229.9, 00:04:31, Vlan11
C       10.10.230.32/27 is directly connected, Vlan101
O       10.10.229.40/30 [110/31] via 10.10.229.9, 00:04:31, Vlan11
O       10.10.229.52/30 [110/20] via 10.10.229.9, 00:04:31, Vlan11
O       10.10.229.48/30 [110/21] via 10.10.229.9, 00:04:32, Vlan11
O       10.10.230.0/29 [110/20] via 10.10.229.9, 00:04:32, Vlan11
O       10.10.229.12/30 [110/20] via 10.10.229.9, 00:04:32, Vlan11
O       10.10.230.12/30 [110/31] via 10.10.229.9, 00:04:32, Vlan11
C       10.10.229.8/30 is directly connected, Vlan11
O       10.10.230.8/30 [110/31] via 10.10.229.9, 00:04:32, Vlan11
O*E1 0.0.0.0/0 [110/21] via 10.10.229.9, 00:04:32, Vlan11
SwDc02-01#sh ip route vrf Red-102

Routing Table: Red-102
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.10.229.13 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 16 subnets, 5 masks
O       10.10.220.200/29 [110/30] via 10.10.229.13, 00:04:51, Vlan12
O       10.10.229.125/32 [110/22] via 10.10.229.13, 00:04:51, Vlan12
O       10.10.229.126/32 [110/11] via 10.10.229.13, 00:04:51, Vlan12
C       10.10.230.64/26 is directly connected, Vlan102
O       10.10.229.36/30 [110/31] via 10.10.229.13, 00:04:51, Vlan12
O       10.10.220.24/29 [110/31] via 10.10.229.13, 00:04:51, Vlan12
O       10.10.229.32/30 [110/21] via 10.10.229.13, 00:04:52, Vlan12
O       10.10.230.32/27 [110/31] via 10.10.229.13, 00:04:52, Vlan12
O       10.10.229.40/30 [110/31] via 10.10.229.13, 00:04:52, Vlan12
O       10.10.229.52/30 [110/20] via 10.10.229.13, 00:04:52, Vlan12
```

```
O       10.10.229.48/30 [110/21] via 10.10.229.13, 00:04:54, Vlan12
O       10.10.230.0/29 [110/20] via 10.10.229.13, 00:04:55, Vlan12
C       10.10.229.12/30 is directly connected, Vlan12
O       10.10.230.12/30 [110/31] via 10.10.229.13, 00:04:55, Vlan12
O       10.10.229.8/30 [110/20] via 10.10.229.13, 00:04:55, Vlan12
O       10.10.230.8/30 [110/31] via 10.10.229.13, 00:04:55, Vlan12
O*E1 0.0.0.0/0 [110/21] via 10.10.229.13, 00:04:55, Vlan12
SwDc02-01#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

12.3.2.2 Prueba 2

Descripción:
Se apaga el router correspondiente a DC01

Resultado esperado:
Solamente contestan las ip's configuradas del el equipamiento situado en DC02.

Resultado obtenido:
Según lo esperado

Previo a la caída.

```
SwDc01-01#sh standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl101       11  100    Standby  10.10.230.34    local           10.10.230.36
Vl101       101 101  P Active   local           10.10.230.34    10.10.230.35
Vl102       12  100    Standby  10.10.230.66    local           10.10.230.68
Vl102       102 101  P Active   local           10.10.230.66    10.10.230.67

SwDc02-01#sh standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl101       11  101  P Active   local           10.10.230.33    10.10.230.36
Vl101       101 100    Standby  10.10.230.33    local           10.10.230.35
Vl102       12  101  P Active   local           10.10.230.65    10.10.230.68
Vl102       102 100    Standby  10.10.230.65    local           10.10.230.67
```

Luego de la caída

```
RoDc02-01
*Jan 22 23:54:28.519: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state
to down
*Jan 22 23:54:28.555: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.229.125 on Tunnel1 from FULL to
DOWN, Neighbor Down: Interface down or detached
```

```
RoEmpS01-01
*Jan 23 00:08:54.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel2, changed state
to down
*Jan 23 00:08:54.867: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.229.125 on Tunnel2 from FULL to
DOWN, Neighbor Down: Interface down or detached

RoEmpS02-01#
*Jan 22 23:52:15.227: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel3, changed state
to down
*Jan 22 23:52:15.247: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.229.125 on Tunnel3 from FULL to
DOWN, Neighbor Down: Interface down or detached

SwDc02-01
*Mar  1 01:17:23.807: %HSRP-5-STATECHANGE: Vlan102 Grp 102 state Standby -> Active
*Mar  1 01:17:23.927: %HSRP-5-STATECHANGE: Vlan101 Grp 101 state Standby -> Active
*Mar   1 01:17:46.663: %OSPF-5-ADJCHG: Process 11, Nbr 10.10.230.33 on Vlan101 from FULL to
DOWN, Neighbor Down: Dead timer expired
*Mar   1 01:17:46.703: %OSPF-5-ADJCHG: Process 12, Nbr 10.10.230.65 on Vlan102 from FULL to
DOWN, Neighbor Down: Dead timer expired

SwDc02-01#sh standby brief
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl101       11  101  P Active   local           unknown         10.10.230.36
Vl101       101 100    Active   local           unknown         10.10.230.35
Vl102       12  101  P Active   local           unknown         10.10.230.68
Vl102       102 100    Active   local           unknown         10.10.230.67

10.2[5]> trace  10.10.230.72
trace to 10.10.230.72, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   39.000 ms  28.000 ms  9.000 ms
 2   10.10.220.28   70.000 ms  28.000 ms  31.000 ms
 3     *10.10.229.49   336.000 ms  *
 4   10.10.229.49   48.000 ms  351.000 ms  *
 5   10.10.229.14   45.000 ms  41.000 ms  351.000 ms
 6     **10.10.230.72   7.000 ms (ICMP type:3, code:3, Destination port unreachable)

10.2[5]> trace  10.10.230.42
trace to 10.10.230.42, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   38.000 ms  30.000 ms  10.000 ms
 2   10.10.220.28   59.000 ms  30.000 ms  38.000 ms
 3     *10.10.229.49   888.000 ms  *
 4   10.10.229.49   64.000 ms  808.000 ms  *
 5   10.10.229.10   72.000 ms  47.000 ms  865.000 ms
 6     **10.10.230.42   98.000 ms (ICMP type:3, code:3, Destination port unreachable)

 20.2[6]> trace 10.10.230.42
trace to 10.10.230.42, 8 hops max, press Ctrl+C to stop
 1   10.10.20.1   42.000 ms  19.000 ms  11.000 ms
 2   10.10.220.204   82.000 ms  29.000 ms  21.000 ms
 3   10.10.229.53   81.000 ms  95.000 ms  94.000 ms
 4   10.10.229.10   88.000 ms  59.000 ms  59.000 ms
 5   *10.10.230.42   78.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

```
20.2[6]> trace 10.10.230.72
trace to 10.10.230.72, 8 hops max, press Ctrl+C to stop
 1   10.10.20.1   48.000 ms  19.000 ms  49.000 ms
 2   10.10.220.204   69.000 ms  40.000 ms  39.000 ms
 3   10.10.229.53   113.000 ms  80.000 ms  88.000 ms
 4   10.10.229.14   84.000 ms  70.000 ms  71.000 ms
 5   *10.10.230.72   67.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

12.3.2.3 Prueba 3

Descripción

Se levanta el switch 01 de DC01 y el router de DC01.

Resultado esperado

Una vez levantado el switch y el router, contestan las ip's definidas en los hsrp en DC01 y DC02

Resultado obtenido:

Según lo esperado

Log del momento en que levantan nuevamente los equipos

```
RoDc02-01#
*Jan 23 20:49:25.299: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state
to up
*Jan 23 20:49:25.531: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.229.125 on Tunnel1 from LOADING to
FULL, Loading Done

RoEmpS02-01#
*Jan 23 20:54:47.403: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel2, changed state
to up
*Jan 23 20:54:52.223: %OSPF-4-ERRRCV: Received invalid packet: Bad Checksum from 10.10.229.49,
Tunnel5
*Jan 23 20:54:52.519: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.229.125 on Tunnel2 from LOADING to
FULL, Loading Done

RoEmpS02-01#
*Jan 23 20:51:35.651: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.229.126 on Tunnel6 from LOADING to
FULL, Loading Done
*Jan 23 20:52:10.631: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel3, changed state
to up
*Jan 23 20:52:11.019: %OSPF-5-ADJCHG: Process 1, Nbr 10.10.229.125 on Tunnel3 from LOADING to
FULL, Loading Done

SwDc01-01#
*Mar  1 00:26:30.507: %OSPF-5-ADJCHG: Process 12, Nbr 10.10.230.66 on Vlan102 from LOADING to
FULL, Loading Done
*Mar  1 00:26:30.571: %OSPF-5-ADJCHG: Process 11, Nbr 10.10.230.34 on Vlan101 from LOADING to
FULL, Loading Done
*Mar  1 00:26:34.047: %HSRP-5-STATECHANGE: Vlan101 Grp 11 state Speak -> Standby
*Mar  1 00:26:34.127: %HSRP-5-STATECHANGE: Vlan102 Grp 12 state Speak -> Standby

SwDc02-01#
*Mar  1 00:26:30.391: %HSRP-5-STATECHANGE: Vlan102 Grp 102 state Active -> Speak
*Mar  1 00:26:32.127: %HSRP-5-STATECHANGE: Vlan101 Grp 101 state Active -> Speak
*Mar  1 00:26:37.539: %OSPF-5-ADJCHG: Process 12, Nbr 10.10.230.65 on Vlan102 from LOADING to
```

```
FULL, Loading Done
*Mar  1 00:26:37.587: %OSPF-5-ADJCHG: Process 11, Nbr 10.10.230.33 on Vlan101 from LOADING to
FULL, Loading Done
*Mar  1 00:26:40.391: %HSRP-5-STATECHANGE: Vlan102 Grp 102 state Speak -> Standby
*Mar  1 00:26:42.127: %HSRP-5-STATECHANGE: Vlan101 Grp 101 state Speak -> Standby
```

12.3.2.4 Prueba 4

Descripción:
Cae router RoEmpS01 en la empresa.

Resultado Esperado:
Al caer RoEmpS01.01, el tráfico desde el Sitio 1 de la empresa hace tránsito por dentro de la empresa RoEmpS01.02 => RoEmpS02.02 => RoEmpS02.01.

Resultado obtenido:
Según lo esperado

Antes de la caída de RoEmpS01.01

```
RoEmpS01-02#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 9 subnets, 4 masks
C        10.10.0.0/30 is directly connected, FastEthernet0/1
L        10.10.0.1/32 is directly connected, FastEthernet0/1
C        10.10.10.0/24 is directly connected, FastEthernet1/0.10
L        10.10.10.1/32 is directly connected, FastEthernet1/0.10
O        10.10.20.0/24 [110/2] via 10.10.0.2, 00:24:40, FastEthernet0/1
C        10.10.220.24/29 is directly connected, FastEthernet0/0
L        10.10.220.26/32 is directly connected, FastEthernet0/0
S        10.10.229.0/24 [1/0] via 10.10.220.28, FastEthernet0/0
S        10.10.230.0/24 [1/0] via 10.10.220.28, FastEthernet0/0

10.2[5]> trace 10.10.230.41
trace to 10.10.230.41, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   47.000 ms  27.000 ms  9.000 ms
 2   10.10.220.28   53.000 ms  20.000 ms  24.000 ms
 3   10.10.229.37   84.000 ms  55.000 ms  73.000 ms
 4   10.10.230.10   92.000 ms  245.000 ms  258.000 ms
 5   *10.10.230.41   101.000 ms (ICMP type:3, code:3, Destination port unreachable)  11.000 ms
```

Cae RoEmpS01.01

```
RoEmpS01-02#
*Jan 23 21:04:50.239: %TRACKING-5-STATE: 1 ip sla 10 reachability Up->Down

RoEmpS01-02#
*Jan 23 21:04:50.239: %TRACKING-5-STATE: 1 ip sla 10 reachability Up->Down
```

```
RoEmpS01-02#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 9 subnets, 4 masks
C        10.10.0.0/30 is directly connected, FastEthernet0/1
L        10.10.0.1/32 is directly connected, FastEthernet0/1
C        10.10.10.0/24 is directly connected, FastEthernet1/0.10
L        10.10.10.1/32 is directly connected, FastEthernet1/0.10
O        10.10.20.0/24 [110/2] via 10.10.0.2, 00:27:52, FastEthernet0/1
C        10.10.220.24/29 is directly connected, FastEthernet0/0
L        10.10.220.26/32 is directly connected, FastEthernet0/0
S        10.10.229.0/24 [10/0] via 10.10.0.2
S        10.10.230.0/24 [10/0] via 10.10.0.2

10.2[5]> trace 10.10.230.41
trace to 10.10.230.41, 8 hops max, press Ctrl+C to stop
 1   10.10.10.1   39.000 ms  33.000 ms  9.000 ms
 2   10.10.0.2   51.000 ms  31.000 ms  28.000 ms
 3   10.10.220.204   69.000 ms  50.000 ms  50.000 ms
 4   10.10.229.53   109.000 ms  110.000 ms  109.000 ms
 5   10.10.229.10   110.000 ms  113.000 ms  81.000 ms
 6   *10.10.230.41   118.000 ms (ICMP type:3, code:3, Destination port unreachable)
```

Levanta RoEmpS01.01

```
RoEmpS01-02#
*Jan 23 21:05:55.315: %TRACKING-5-STATE: 1 ip sla 10 reachability Down->Up
```

12.3.3 Verificación del MTU sobre los túneles

Para la determinación del MTU se utilizaron varias técnicas

- Utilitario Ping [156]
- Utilitario mturoute.exe [157]
- Utilitario mtupath.exe [158]

Para este fin se instalaron dos maquinas virtuales con Windows 7 y se integraron con la plataforma GNS3. Uno de estoas máquinas virtuales se configuró con la dirección IP 10.10.20.3 (Sitio 2 de la empresa) y la otra con la dirección IP 10.10.230.50 (Datacenter 2). Al intercambiar tráfico entre ellas, se utiliza como vínculo el Tunel 6, por lo que es medido.
El procedimiento es análogo para los demás túneles

12.3.3.1 Tunel sin cifrar

En este caso se quitó el cifrado del túnel y se ejecutaron los casos de test.

```
Tunnel6 is up, line protocol is up
  Internet address is 10.10.229.54/30
```

```
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1476 bytes


10.10.230.50 => 10.10.20.3

C:\tmp>tracert 10.10.20.3

Tracing route to WIN7 [10.10.20.3]
over a maximum of 30 hops:

  1    20 ms    10 ms     8 ms  10.10.230.34       // SwDc02.01 v101
  2    57 ms    74 ms   110 ms  10.10.229.9        // RoDc02.01 f1/0.11
  3   104 ms    81 ms    90 ms  10.10.229.54       // RoEmpS02.01 Tunnel 6
  4   186 ms   162 ms    73 ms  10.10.220.202      // RoEmpS02.02 f0/0
  5   113 ms    71 ms   145 ms  WIN7 [10.10.20.3]

Trace complete.

C:\tmp>ping -l 1442 -f 10.10.20.3

Pinging 10.10.20.3 with 1442 bytes of data:
Reply from 10.10.20.3: bytes=1442 time=107ms TTL=124
Reply from 10.10.20.3: bytes=1442 time=79ms TTL=124
Reply from 10.10.20.3: bytes=1442 time=77ms TTL=124
Reply from 10.10.20.3: bytes=1442 time=81ms TTL=124

Ping statistics for 10.10.20.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 77ms, Maximum = 107ms, Average = 86ms

C:\tmp>ping -l 1443 -f 10.10.20.3

Pinging 10.10.20.3 with 1443 bytes of data:
Reply from 10.10.20.3: bytes=1443 time=93ms TTL=124
Reply from 10.10.20.3: bytes=1443 time=87ms TTL=124
Reply from 10.10.20.3: bytes=1443 time=69ms TTL=124
Reply from 10.10.20.3: bytes=1443 time=60ms TTL=124

Ping statistics for 10.10.20.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 60ms, Maximum = 93ms, Average = 77ms

C:\tmp>mturoute  -m 2000 10.10.20.3
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
- ICMP payload of 1472 bytes is too big.
+ ICMP payload of 92 bytes succeeded.
+ ICMP payload of 782 bytes succeeded.
+ ICMP payload of 1127 bytes succeeded.
+ ICMP payload of 1299 bytes succeeded.
+ ICMP payload of 1385 bytes succeeded.
```

```
+ ICMP payload of 1428 bytes succeeded.
- ICMP payload of 1450 bytes is too big.
+ ICMP payload of 1439 bytes succeeded.
+ ICMP payload of 1444 bytes succeeded.
+ ICMP payload of 1447 bytes succeeded.
+ ICMP payload of 1448 bytes succeeded.
- ICMP payload of 1449 bytes is too big.
Path MTU: 1476 bytes.

C:\tmp>mturoute -t  -m 2000 10.10.20.3
mturoute to 10.10.20.3, 30 hops max, variable sized packets
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
 1  +-  host: 10.10.230.34  max: 1500 bytes
 2  +-  host: 10.10.229.9  max: 1500 bytes
 3  -++++++-.++++-  host: 10.10.229.54  max: 1476 bytes
 4  +-  host: 10.10.220.202  max: 1476 bytes
 5  +.-  host: 10.10.20.3  max: 1476 bytes

 C:\tmp>mtupath -4  10.10.20.3

MTU path scan to 10.10.20.3, ttl=64, limit=48
# 16 processing - best MSS 1448 (estimated MTU 1476) [pPPPPpPppPpPPppp]
# 08 nearest minimum MTU on 10.10.229.54 (3 hops away)

        #1 MSS IN RANGE     1 <==  1447 ==>  1448
        #2 MSS EXCEEDED  1449 <== 14935 ==> 16384

C:\Users\pablo>tracert 10.10.230.50

Tracing route to WIN7 [10.10.230.50]
over a maximum of 30 hops:

  1    15 ms     8 ms    10 ms  10.10.20.1           // RoEmpS02.02 f1/0.10
  2    69 ms    35 ms    21 ms  10.10.220.204        // RoEmpS02.01 f1/1
  3   100 ms   102 ms    84 ms  10.10.229.53         // RoDc02.01 Tunel 6
  4    83 ms    73 ms    69 ms  10.10.229.10         // SwDc02.01 v11
  5    74 ms    75 ms    76 ms  WIN7 [10.10.230.50]

Trace complete.

C:\Users\pablo>mturoute -m 2000 10.10.230.50
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
- ICMP payload of 1472 bytes is too big.
+ ICMP payload of 92 bytes succeeded.
+ ICMP payload of 782 bytes succeeded.
+ ICMP payload of 1127 bytes succeeded.
+ ICMP payload of 1299 bytes succeeded.
+ ICMP payload of 1385 bytes succeeded.
+ ICMP payload of 1428 bytes succeeded.
- ICMP payload of 1450 bytes is too big.
+ ICMP payload of 1439 bytes succeeded.
+ ICMP payload of 1444 bytes succeeded.
```

```
+ ICMP payload of 1447 bytes succeeded.
+ ICMP payload of 1448 bytes succeeded.
- ICMP payload of 1449 bytes is too big.
Path MTU: 1476 bytes.

C:\Users\pablo>mturoute -t -m 2000 10.10.230.50
mturoute to 10.10.230.50, 30 hops max, variable sized packets
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
 1  +-  host: 10.10.20.1  max: 1500 bytes
 2  +-  host: 10.10.220.204  max: 1500 bytes
 3  -++++++-++++-  host: 10.10.229.53  max: 1476 bytes
 4  +-  host: 10.10.229.10  max: 1476 bytes
 5  +.-  host: 10.10.230.50  max: 1476 bytes

C:\Users\pablo>mtupath -4 10.10.230.50

MTU path scan to 10.10.230.50, ttl=64, limit=48
# 16 processing - best MSS 1448 (estimated MTU 1476) [pPPPPpPppPpPPppp]
# 08 nearest minimum MTU on 10.10.229.53 (3 hops away)

        #1 MSS IN RANGE     1 <==  1447 ==>  1448
        #2 MSS EXCEEDED  1449 <== 14935 ==> 16384
```

De este estudio surge que la máxima cantidad de datos que se puede enviar en un paquete si que sea fragmentado es 1476 bytes (1500 menos el encabezado GRE de 24 bytes).

12.3.3.2 Tunel cifrado

En este caso se volvió a cifrar el túnel 6 y se ejecutaron los tests nuevamente.

```
Tunnel6 is up, line protocol is up
  Internet address is 10.10.229.53/30
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1446 bytes

10.10.230.50 => 10.10.20.3

C:\tmp>tracert 10.10.20.3

Tracing route to WIN7 [10.10.20.3]
over a maximum of 30 hops:

  1    20 ms    10 ms     8 ms  10.10.230.34       // SwDc02.01 v101
  2    57 ms    74 ms   110 ms  10.10.229.9        // RoDc02.01 f1/0.11
  3   104 ms    81 ms    90 ms  10.10.229.54       // RoEmpS02.01 Tunnel 6
  4   186 ms   162 ms    73 ms  10.10.220.202      // RoEmpS02.02 f0/0
  5   113 ms    71 ms   145 ms  WIN7 [10.10.20.3]

Trace complete.

C:\tmp>ping -l 1419 -f 10.10.20.3
```

```
Pinging 10.10.20.3 with 1419 bytes of data:
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.
Packet needs to be fragmented but DF set.

Ping statistics for 10.10.20.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\tmp>ping -l 1418 -f 10.10.20.3

Pinging 10.10.20.3 with 1418 bytes of data:
Reply from 10.10.20.3: bytes=1418 time=112ms TTL=124
Reply from 10.10.20.3: bytes=1418 time=84ms TTL=124
Reply from 10.10.20.3: bytes=1418 time=97ms TTL=124
Reply from 10.10.20.3: bytes=1418 time=113ms TTL=124

Ping statistics for 10.10.20.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 84ms, Maximum = 113ms, Average = 101ms

C:\tmp>mturoute  -m 2000 10.10.20.3
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
- ICMP payload of 1472 bytes is too big.
+ ICMP payload of 92 bytes succeeded.
+ ICMP payload of 782 bytes succeeded.
+ ICMP payload of 1127 bytes succeeded.
+ ICMP payload of 1299 bytes succeeded.
+ ICMP payload of 1385 bytes succeeded.
- ICMP payload of 1428 bytes is too big.
+ ICMP payload of 1406 bytes succeeded.
+ ICMP payload of 1417 bytes succeeded.
- ICMP payload of 1422 bytes is too big.
- ICMP payload of 1419 bytes is too big.
+ ICMP payload of 1418 bytes succeeded.
Path MTU: 1446 bytes.

C:\tmp>mturoute -t  -m 2000 10.10.20.3
mturoute to 10.10.20.3, 30 hops max, variable sized packets
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
 1  +-  host: 10.10.230.34  max: 1500 bytes
 2  +-  host: 10.10.229.9  max: 1500 bytes
 3  -++++.+-++--+  host: 10.10.229.54  max: 1446 bytes
 4  +-  host: 10.10.220.202  max: 1446 bytes
 5  +.-  host: 10.10.20.3  max: 1446 bytes

C:\tmp>mtupath -4  10.10.20.3

MTU path scan to 10.10.20.3, ttl=64, limit=48
# 16 processing - best MSS 1418 (estimated MTU 1446) [pPPPPpPppPPPpPPp]
# 08 nearest minimum MTU on 10.10.229.54 (3 hops away)
```

```
        #1 MSS IN RANGE     1 <==  1417 ==>  1418
        #2 MSS EXCEEDED  1419 <== 14965 ==> 16384



10.10.20.3 => 10.10.20.3

C:\Users\pablo>tracert 10.10.230.50

Tracing route to WIN7 [10.10.230.50]
over a maximum of 30 hops:

  1    15 ms     8 ms    10 ms  10.10.20.1           // RoEmpS02.02 f1/0.10
  2    69 ms    35 ms    21 ms  10.10.220.204        // RoEmpS02.01 f1/1
  3   100 ms   102 ms    84 ms  10.10.229.53         // RoDc02.01 Tunel 6
  4    83 ms    73 ms    69 ms  10.10.229.10         // SwDc02.01 v11
  5    74 ms    75 ms    76 ms  WIN7 [10.10.230.50]

Trace complete.

C:\Users\pablo>mturoute  -m 2000 10.10.230.50
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
- ICMP payload of 1472 bytes is too big.
+ ICMP payload of 92 bytes succeeded.
+ ICMP payload of 782 bytes succeeded.
+ ICMP payload of 1127 bytes succeeded.
+ ICMP payload of 1299 bytes succeeded.
+ ICMP payload of 1385 bytes succeeded.
- ICMP payload of 1428 bytes is too big.
+ ICMP payload of 1406 bytes succeeded.
+ ICMP payload of 1417 bytes succeeded.
- ICMP payload of 1422 bytes is too big.
- ICMP payload of 1419 bytes is too big.
+ ICMP payload of 1418 bytes succeeded.
Path MTU: 1446 bytes.

C:\Users\pablo>mturoute -t  -m 2000 10.10.230.50
mturoute to 10.10.230.50, 30 hops max, variable sized packets
* ICMP Fragmentation is not permitted. *
* Speed optimization is enabled. *
* Maximum payload is 2000 bytes. *
 1  +-  host: 10.10.20.1  max: 1500 bytes
 2  .+-  host: 10.10.220.204  max: 1500 bytes
 3  -++++-++--+  host: 10.10.229.53  max: 1446 bytes
 4  +-  host: 10.10.229.10  max: 1446 bytes
 5  +.-  host: 10.10.230.50  max: 1446 bytes

C:\Users\pablo>mtupath -4  10.10.230.50

MTU path scan to 10.10.230.50, ttl=64, limit=48
# 16 processing - best MSS 1418 (estimated MTU 1446) [pPPPPpPppPPPpPPp]
# 08 nearest minimum MTU on 10.10.229.53 (3 hops away)
```

```
#1 MSS IN RANGE     1 <==  1417 ==>  1418
#2 MSS EXCEEDED  1419 <== 14965 ==> 16384
```

En este caso se concluye que el tamaño máximo de paquete a enviar sin ser fragmentado es de 1446 bytes (1500 menos el encabezado GRE menos el encabezado IPsec).

12.4 Anexo IV – Estudio de costos involucrados en el caso de estudio

12.4.1 Enlaces de datos

Para el estudio de la topología a utilizar, se evaluaron propuestas de conectividad de un proveedor en Uruguay. Los precios que se reflejarán provienen de un presupuesto para una instalación concreta.

Tipo de enlace	Cantidad	Costo mensual (U$S)	Observaciones
LAN to LAN	3	10.540	3 enlaces LAN to LAN 100 Mbps DC1 <=> S01 DC2 <=> S02 DC1 <=> DC2
LAN Service	1	8.432	1 servicio VPLS 100 Mbps DC1 DC2 S01 S02

Estos precios tienen una base de un contrato a tres años y se debe aclarar que la cotización de donde se tomaron estos datos incluye otros servicios (incluidos por igual en ambos costos) que no se encuentran desglosados.

Tomando en cuenta que el costo de instalación es igual a un arrendamiento mensual, surge que la opción “LAN to LAN” tiene un costo de un 25% que una configuración similar utilizando “LAN Service” (VPLS).

12.4.2 Equipamiento

Item	Precio Unitario (U$S)	Observaciones
ASA 5512	5.000	Sin fuente redundante
ASA 5545	18.000	Con fuente redundante
		Escala = 3 x 5512
ASA 5 contextos	4.000	Licencia
ASA 10 contextos	8.000	Licencia
ASA 20 contextos	13.000	Licencia
ASA 50 contextos	25.000	Licencia
Router 2921 /sec	4.800	Fuente redundante
Switch 3560 48 port	5.000	Sin fuente redundante
Switch 4948 48 port	13.000	Con fuente redundante

En el caso de los “firewalls” virtualizable, se ve claramente que por U$ 4000 se pueden tener 5 contextos adicionales y el precio unitario va decreciendo a medida que se adquieren paquetes con un número mayor de contextos. Esta diferencia es mayor a medida que se crece en la

gama de equipo, pero se empiezan a disponer de características de mayor porte como ser fuentes redundantes. Para una comparativa completa se puede ver en [159]. Para los costos también se debe considerar el mantenimiento. En este caso particular, se ha cotizado únicamente los costos de mano de obra y escalamiento de incidentes al fabricante.

Equipo	Mantenimiento Anual Unitario (Solo horas hombre) (U$$)
ASA 5545	2.200
CISCO2921-	864
WS-C3560G-48TS	723

12.4.3 Personal

En la siguiente tabla se presenta una estimación de salarios de los roles necesarios para conformar la dotación de un NOC para mantener la red de esta solución.

Rol	Nominal mensual (U$$)	Costo empresa anualizado (U$$) Nominal*12*1,4	Costo empresa mensual promedio (U$$)
Gte Operaciones Networking	7.000	117.600	9.800
Adm Networking Senior	5.000	84.000	7.000
Adm Networking Junior	3.000	50.400	4.200
Operador de red Senior	2.000	33.600	2.800
Operador de red Jr	1.500	25.200	2.100

Un equipo básico para mantener esta red está compuesto por
- 1 Gerente de operaciones de Networking
- 1 Administrador Senior
- 1 Operador Junior

En la medida que se agregan clientes con redes similares y/o sitios, se agrega personal, pero no se debe armar un equipo nuevo. La virtualización de los equipos de red ayuda a que la incorporación de personal sea gradual y en lo posible, promocionando las posiciones Junior en Senior e incorporar posiciones Junior.

12.4.4 Proyecto a 5 años

En la tabla que se detalla a continuación se analizan los costos de un proyecto de networking a 5 años de la envergadura de detallado en el caso de uso, con los siguientes supuestos:

- La instalación de la infraestructura dura dos meses.
- El costo de instalación de los enlaces corresponde a un mes de servicio.
- Para cubrir el costo de mantenimiento de hardware, se adquiere al inicio una unidad extra de cada tipo de dispositivo.
- La plantilla se mantiene constante.

		Instalación		Año 1		Año 2		Año 3		Año 4		Año 5	
		Cantidad	Costo (U$$)	Cantidad	Costo (U$$)	Cantidad	Costo (U$$)	Cantidad	Costo (U$$)	Cantidad	Costo (U$$)	Cantidad	Costo (U$$)
Item	**Costo unitario (U$$)**												
Equipamiento													
ASA 5545	18.000	3	54.000	0	0	0	0	0	0	0	0	0	0
Router 2921 /sec	4.800	5	24.000	0	0	0	0	0	0	0	0	0	0
Switch 3560 48 port	5.000	5	25.000	0	0	0	0	0	0	0	0	0	0
Enlaces													
LAN Service	8.432	3	25.296	12	101.184	12	101.184	12	101.184	12	101.184	12	101.184
RRHH													
Gte Operaciones	9.800	2	19.600	12	117.600	12	117.600	12	117.600	12	117.600	12	117.600
Adm Networking Senior	7.000	2	14.000	12	84.000	12	84.000	12	84.000	12	84.000	12	84.000
Operador de red Jr	2.100	2	4.200	12	25.200	12	25.200	12	25.200	12	25.200	12	25.200
Mantenimiento													
ASA 5545	2.200	0	0	1	2.200	1	2.200	1	2.200	1	2.200	1	2.200
Router 2921 /sec	864	0	0	1	864	1	864	1	864	1	864	1	864
Switch 3560 48 port	723	0	0	1	723	1	723	1	723	1	723	1	723
Subtotal			166.096		331.771		331.771		331.771		331.771		331.771
Total													1.824.951

ISSN 1510-7264 (Tesis de Maestría en Ingeniería en Computación)
Pablo Gestido
Tesis de Maestría en Ingeniería en Computación
Facultad de Ingeniería
Universidad de la República

Montevideo, Uruguay, 2014