REDES WAN DEFINIDAS POR SOFTWARE. SD-WAN

David Suárez Rubio
Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Integración de redes telemáticas

Antoni Morell Pérez
Pere Tuset Peiró

Junio de 2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	*REDES WAN DEFINIDAS POR SOFTWARE. SD-WAN*
Nombre del autor:	*David Suárez Rubio*
Nombre del consultor/a:	*Antoni Morell Pérez*
Nombre del PRA:	*Pere Tuset Peiró*
Fecha de entrega:	*06/2020*
Titulación:	*Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación*
Área del Trabajo Final:	*Integración de redes telemáticas*
Idioma del trabajo:	*Castellano*
Palabras clave	*SDWAN, SDN, REDES*

Resumen del Trabajo:

La tecnología SD-WAN, del inglés *software-defined wide area network*, es una tecnología para configurar e implementar una WAN empresarial, basada en redes definidas por software (SDN), para enrutar efectivamente el tráfico a ubicaciones remotas como sucursales y centros de datos, en las instalaciones o en la nube. La tecnología SD-WAN obtiene importantes beneficios de flexibilidad y agilidad al eliminar la carga de la gestión del tráfico de los dispositivos físicos y transferirla al software, la esencia de SDN.

Algunas de las ventajas de SD-WAN son la conmutación automática en caso de fallo, la redundancia, la administración simplificada y el ahorro de costes que reducen el costo de mantener la tecnología implementada en ubicaciones remotas.

La metodología seguida consiste en la descripción de diferentes plataformas de SD-WAN y la elección de una de ellas para desarrollarla en el proyecto.

Se ha elegido una plataforma *Open Source* para la que se han realizado pruebas de conectividad, rendimiento y redundancia. Los resultados obtenidos han sido buenos, se han instalado tres nodos SD-WAN, uno de ellos en la nube de Amazon AWS. Se ha conseguido instalar un nodo en un uCPE. La comunicación extremo a extremo entre nodos se ha establecido satisfactoriamente y se ha conseguido establecer una comunicación entre dos nodos redundada. Las pruebas de rendimiento han resultado igualmente satisfactorias.

Como conclusión, un entorno SD-WAN es similar en prestaciones a un entorno tradicional.

Abstract:

SD-WAN (software-defined wide area network), is a technology for configuring and implementing an enterprise WAN, based on software-defined networks (SDN), to effectively route traffic to remote locations such as branches and data centers , on-premises or in the cloud.

SD-WAN technology gains significant flexibility and agility benefits by removing the load of managing physical device traffic and transferring it to software, the essence of SDN.

Some of the advantages of SD-WAN are auto failover, redundancy, simplified administration, and cost savings, that reduce the cost of maintaining the technology deployed in remote locations.

The methodology for this proyect speak about different SD-WAN platforms and choosing one of them to develop it in the project.

An Open Source platform has been chosen for which connectivity, performance and redundancy tests have been carried out. The results obtained have been good, three SD-WAN nodes have been installed, one of them in the Amazon AWS cloud. A node has been successfully installed on a uCPE. End-to-end communication between nodes has been established successfully and a communication between two redundant nodes has been established. Performance tests have been equally successful.

In conclusion, an SD-WAN environment is similar in performance to a traditional environment.

Contenido

1. Introducción

Antes de entrar en este nuevo paradigma de las redes conviene repasar de forma breve la estructura de una red WAN tradicional.

Una red WAN es una red de comunicaciones que permite conectar dispositivos más allá de la red local (LAN) o de la red del campus (MAN). En las redes LAN el protocolo de acceso es en casi la totalidad de estas Ethernet. En cambio, en las redes WAN existe más variedad de protocolos, como por ejemplo ATM, RDSI, X.25, GPON, etc. Además, dentro de las redes WAN también existen varios tipos, como por ejemplo Internet, MPLS, etc.

Es posible interconectar redes LAN conectándolas a redes WAN que usan diferentes protocolos, los cuales corren en hardware específico. Esta interconexión debe ser privada y para ello se puede utilizar la red de Internet a través de un encapsulado encriptado o se pueden usar redes privadas como las MPLS, donde no se comparte el nivel lógico entre los usuarios.

En un conmutador de red existen tres planos funcionales:

- El plano de datos se encarga del proceso de almacenamiento y reenvío de paquetes.
- El plano de control se encarga de determinar el contenido de la tabla de reenvío a partir de la información que intercambian los conmutadores de la red de forma distribuida.
- El plano de gestión se ocupa de funciones relacionadas con la configuración, operación y mantenimiento del conmutador.

En las redes tradicionales se usan dispositivos que son independientes entre sí dentro de la red, manteniendo para ello el plano de gestión, el plano de control y el plano de datos en cada dispositivo.

En estas redes todos los dispositivos tienen un plano de control completo. Por ejemplo, routers IP que manejan el protocolo BGP intercambian rutas entre si a través de un algoritmo para decidir el encaminamiento de los paquetes. Este encaminamiento se ha construido de forma distribuida a través de la colaboración de los planos de control de cada dispositivo, pero todos ellos mantienen la topología de la red por separado.

1.1 ¿Qué son las redes definidas por software?

Existen principalmente dos tipos de redes definidas por software. Las redes SDN que son el equivalente a la LAN tradicional y las redes SD-WAN que son el equivalente a la WAN tradicional.

1.1.1 SDN

Como se ha comentado en el apartado anterior, existen tres planos de control, los cuales están integrados en cada dispositivo en una red tradicional. Por el contrario, en las redes SDN estos planos se desacoplan.

El plano de control se centraliza en un controlador que es el que se encarga de programar las tablas de reenvío en el plano de datos de los conmutadores SDN que integran la red.

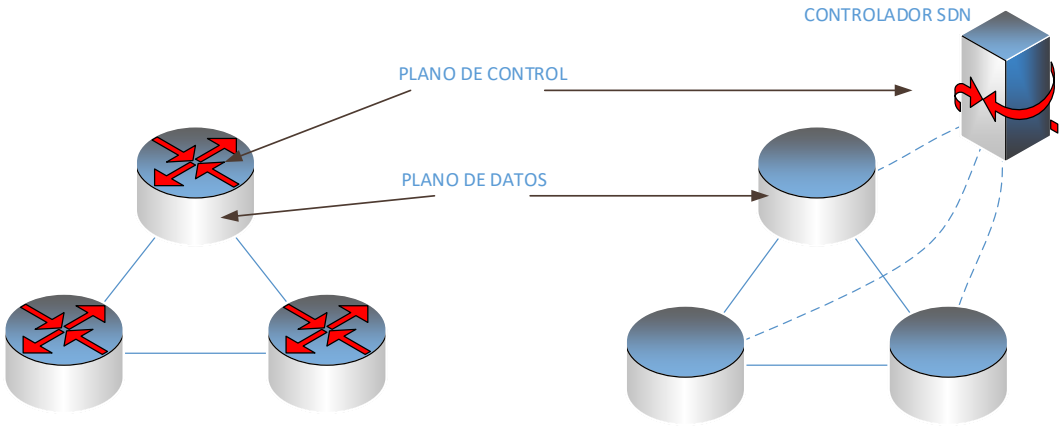


Figura 1 Planos funcionales en elementos de red.

En la figura 1 se aprecia como en las redes SDN, en los conmutadores solo está presente el plano de datos. El plano de control esta centralizado en un equipo o en varios equipos redundados.

Los conmutadores de una red SDN, al disponer solo del plano de datos, no pueden crear entradas en sus tablas de reenvío para un paquete nuevo que llega y del cual no tienen entrada en esta tabla. En este caso, reenvían ese paquete al controlador y este es el que lo examina y configura las tablas de reenvío de los conmutadores relevantes al mismo flujo.

1.1.2 SD-WAN

Software-Defined Wide Area Network (SD-WAN) es el resultado de aplicar las tecnologías SDN a las conexiones WAN. La aplicación de soluciones basadas en SDN en este entorno proporciona diferentes ventajas, como por ejemplo la sencillez respecto a otras tecnologías de la implantación de políticas de calidad de servicio, más facilidad para administrar grandes redes o una gestión más eficiente de los recursos de red.

Más formalmente, el Metro Ethernet Forum (MEF) [1] describe SD_WAN como un servicio que proporciona una red de superposición virtual consciente de las aplicaciones, orientada a políticas y orquestada entre las interfaces SD-WAN UNI, y proporciona la estructura lógica de una red enrutada privada virtual L3 para el suscriptor que transmite paquetes IP entre los puntos de presencia de los suscriptores.

Un servicio SD-WAN es consciente y encamina los paquetes basándose en los flujos de aplicación. El acuerdo de servicio incluye la especificación de los flujos de aplicaciones (paquetes IP que coinciden con un conjunto de criterios) y políticas que describen reglas y restricciones en el reenvío de los flujos de aplicaciones.

Los beneficios de SD-WAN se pueden manifestar en la capacidad de ajustar aspectos del servicio en tiempo casi real para satisfacer las necesidades del negocio. El suscriptor lo hace especificando los comportamientos deseados a nivel de conceptos de negocio conocidos, como aplicaciones y ubicaciones, y el proveedor de servicios supervisa el rendimiento del servicio y modifica cómo se envían los paquetes en cada flujo de aplicación en función de asignación de políticas y telemetría en tiempo real desde los componentes de red subyacentes.

Arquitectura SDN

El servicio SD-WAN puede operar sobre diferentes redes de transporte, definidas en el MEF como *Underlay Connectivity Services* o UCS. Algunos ejemplos de UCS son Internet o una VPN

basada en MPLS. Es importante entender que el proveedor de servicios SD-WAN puede ser diferente del proveedor de UCS, aunque también puede ser el mismo. En caso de ser diferentes el suscriptor será el encargado de coordinar la información que necesita el proveedor SD-WAN del proveedor de los UCS para la puesta en marcha del servicio.

El proveedor de SD-WAN se encarga de gestionar que tráfico se transporta sobre los UCS. Este, utilizando soluciones basadas en SDN, monitoriza el rendimiento del servicio y actúa en tiempo real para modificar la ruta que siguen las aplicaciones para conformar la política de QoS.

El servicio SD-WAN consta de varios componentes. Los nodos SD-WAN *edge*, son los encargados de implementar la función del plano de datos. Estos nodos tienen interfaces hacia los UCS y hacia el suscriptor del servicio SD-WAN. Los interfaces que conectan con el UCS se llaman UCS UNI (*User Network Interface*) y los que conectan con el suscriptor son los SD-WAN UNI.

Cuando se recibe un paquete IP en la interfaz SD-WAN UNI, se determina su interfaz UCS de salida utilizando información de diversa índole: políticas de calidad de servicio, información de los UCS y otros atributos del servicio SD-WAN. Toda esta clasificación termina definiendo unos flujos de aplicación que serán encaminados según la política definida para cada uno de ellos.

Un flujo de aplicación ingresa al Nodo SD-WAN Edge a través del interface SD-WAN UNI, según la política que hayamos definido se elegirá el UCS UNI correspondiente por el que se encaminará el tráfico.

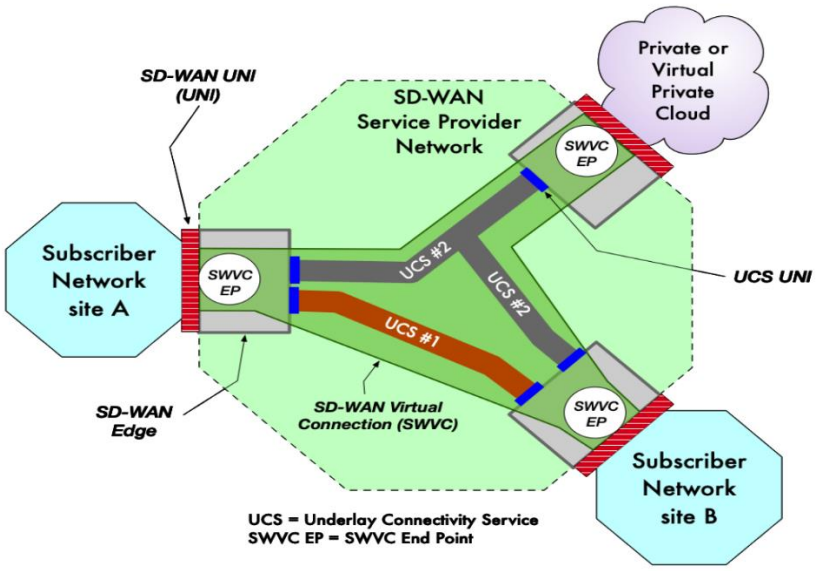


Figura 2 Componentes de una red SD-WAN

En la figura 2 se pueden apreciar los componentes descritos hasta ahora, concretamente, existen tres puntos de presencia a los que el suscriptor necesita dar conectividad. Las redes A y B son sedes físicas que se conectan entre si a través del UCS#1. Estas redes a su vez acceden a su nube privada a través de otro UCS, el UCS#2.

En la figura 3 se detalla uno de los nodos SD-WAN. En esta figura aparecen los túneles virtuales, (TVC, *Tunnel Virtual Connection*) que son conexiones punto a punto que el proveedor SD-WAN establece sobre los UCS y por los que encamina los paquetes según las políticas definidas para el flujo de aplicación que lo conforma.

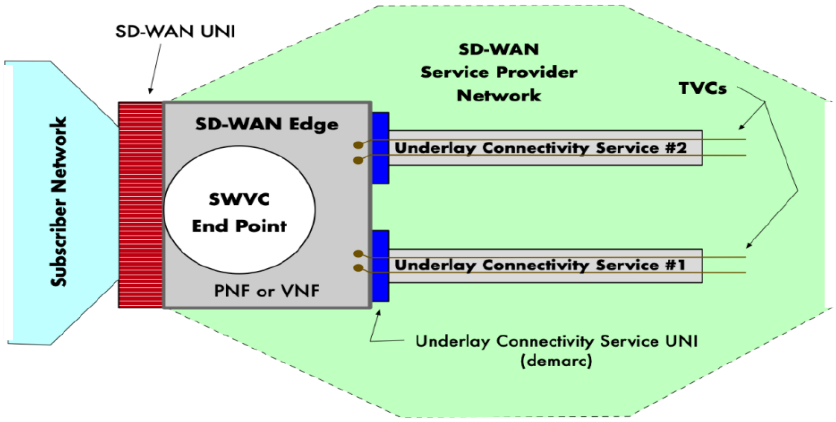


Figura 3 Detalle de un nodo SD-WAN

Uno de los UCS contratados por el suscriptor puede ser la conexión a internet. Al igual que para otros servicios, en este caso se define una política de aplicación que encaminara el flujo de aplicación que ingresa por el SD-WAN UNI a través del UCS UNI de internet en lugar de encaminarlo a otro SD-WAN UNI. A esta capacidad se le denomina *Internet Breakout*. El uso más común es para encaminar el tráfico de un sitio concreto del suscriptor a través de la conexión a internet local.

En la figura 4 se muestra este concepto, donde existen dos UCS con varios TVCs. El UCS#2 es una MPLS que el suscriptor ha contratado con un proveedor. Sobre esta MPLS hay definidos dos TVC por el que se encaminan dos flujos de aplicación diferentes hacia la nube del suscriptor. El UCS#1 es una conexión a internet normal, como por ejemplo un FTTH. Sobre esta conexión se establece un TVC por el que se encamina el flujo de aplicación entre sedes, y, además, está configurado como *Internet Breakout* para el sitio B del suscriptor.

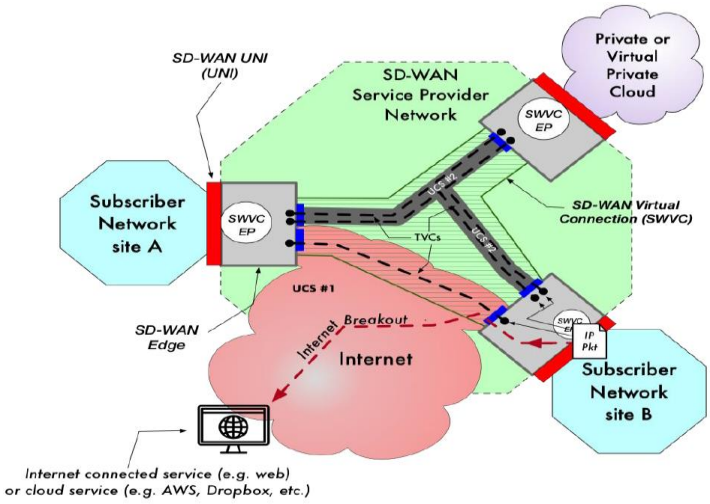


Figura 4 Internet Breakout

En este escenario es posible que un flujo de aplicación asignado a la política de *Internet Breakout*, y en el que el SD-WAN Edge por el que ingresa no disponga de un UCS con Internet, sea encaminado a través de un TVC a otro SD-WAN Edge en el que sí que exista este UCS y así darle conectividad a internet.

Para finalizar la revisión de la arquitectura se muestra en la figura 5 un ejemplo a modo de resumen.

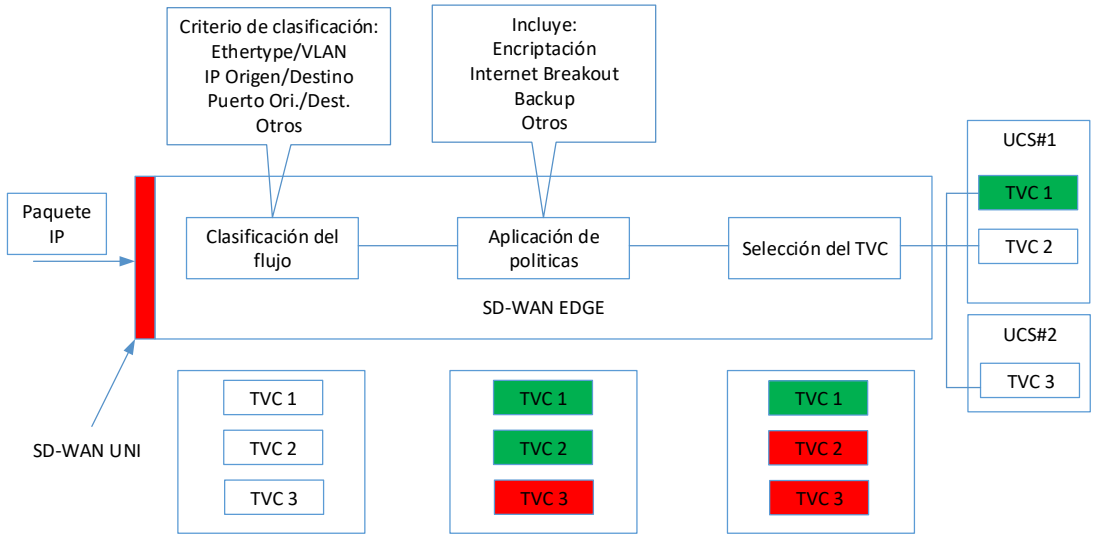


Figura 5 Flujo de aplicación

En la figura 5 se muestra un nodo SD-WAN Edge por el que ingresa un paquete IP que se debe clasificar y encaminar por un TVC que cumpla la política del flujo de aplicación para ese paquete, que va destinado a la nube privada del suscriptor. En el primer paso se clasifica el paquete según la cabecera, suponemos que la clasificación se basa en el origen y destino IP. La política para el flujo de aplicación de la comunicación con la nube privada establece que el paquete debe cifrarse, en este punto el nodo SD-WAN descarta el TVC 3 porque no dispone de esa capacidad. Finalmente, selecciona el TVC por el que debe enviar el paquete, como hay dos TVC que son compatibles con el flujo de aplicación, elegirá el que cumpla los requisitos marcados como mejor ruta, en este caso, el que tenga menor latencia por lo que elige el TVC 1.

Los nodos SD-WAN se pueden desplegar como equipos físicos o como función de red virtualizada (VNF) instanciada en un uCPE.

Orquestación de servicios SD-WAN.

Como se ha visto, las redes SD-WAN simplifican la gestión y la administración de las redes WAN. Establecer una VPN entre dos sedes de una red corporativa puede resultar complejo de configurar. Si, además, es necesario gestionar el ancho de banda de estas conexiones será necesaria la reconfiguración frecuente de estas conexiones.

Para gestionar estas necesidades se han creado APIs, que gracias a la separación de los planos de control y datos de las redes SD-WAN se pueden implementar de forma sencilla. Por ejemplo, para gestionar el ancho de banda de forma dinámica o para responder ágilmente ante eventos que requieren la reconfiguración de un gran número de elementos de la red.

Para diseñar una API se deben tener en cuenta algunos aspectos:

- Tendrá una visión global de la red, por motivos de seguridad se recomienda que esta sea una visión simplificada.
- Debe ser capaz de detectar conflictos entre políticas.
- Debe administrar el ancho de banda para garantizar un uso eficiente de los recursos de red.

1.1.2.1 Casos de uso SD-WAN

En la figura 6 se muestran varios casos de uso para conectar dos puntos de presencia de una red corporativa. De cara al suscriptor no hay diferencia, aunque como se ve en cada topología se usan diferentes UCS. [2]

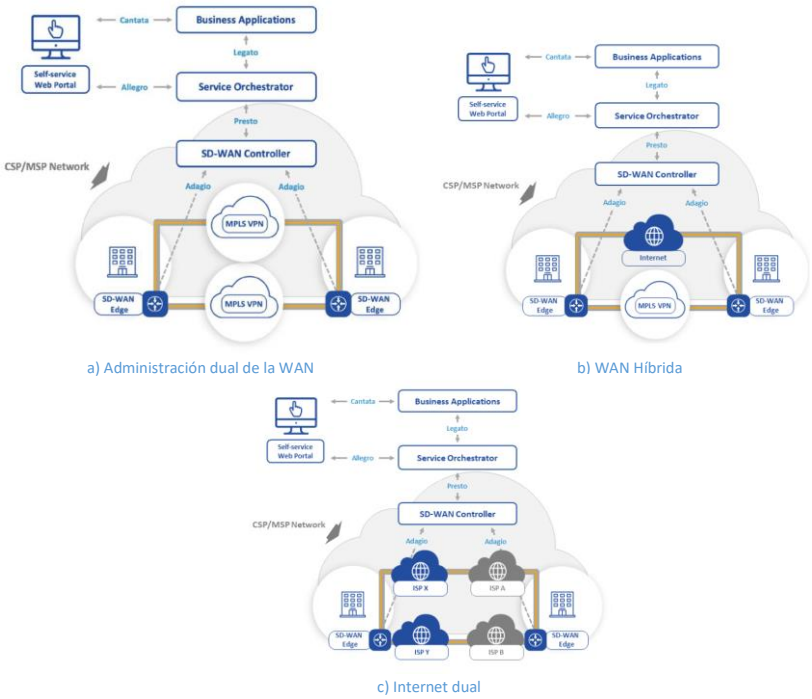


Figura 6 Casos de uso SD-WAN

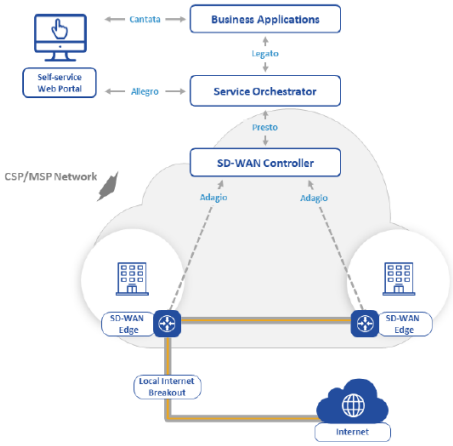
En el caso (a), la red corporativa consta de dos sedes conectadas a través de MPLS. En estos escenarios lo que se busca es la continuidad del negocio minimizando los problemas de conexión que puedan surgir por la caída de un enlace. Normalmente existe un enlace MPLS activo y uno en espera de que falle el principal, caso en el que conmutara al secundario.

Gracias a SD-WAN estos enlaces, que por lo general son costosos, se pueden aprovechar por igual, estableciendo para ello políticas para diferentes flujos que usen TVCs establecidos por cada uno de los enlaces.

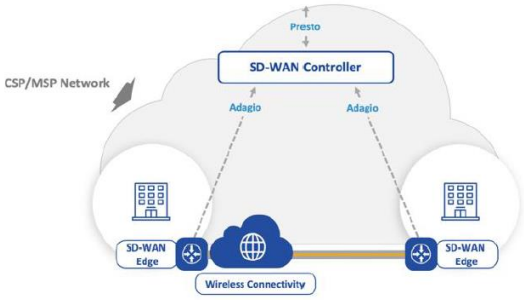
El caso (b) es similar al caso (a), aunque con la salvedad de que uno de los enlaces es una conexión estándar a Internet. Al igual que en el caso anterior, el enlace principal será el activo y el enlace secundario, internet, será el respaldo y no tendrá uso. Con los nodos SD-WAN ocurre lo mismo que en el caso (a), Internet será un UCS más por el que se pueden establecer TVCs entre sedes y por tanto encaminar flujos de aplicación.

Finalmente, el caso (c) contempla la posibilidad de que las sedes corporativas tengan conexión a través de ISPs diferentes, lo que normalmente impide el establecimiento de una MPLS debido a la falta de acuerdos entre operadores. Este caso, es transparente para el suscriptor SD-WAN, verá el mismo comportamiento que en los casos (a) o (b), con TVCs establecidos a través de los diferentes UCS.

A parte de estos casos de uso, el MEF propone otros dos más específicos.



d) Local Internet Breakout



e) Conectividad Wireless

Figura 7 Casos de uso SD-WAN específicos

El caso (d) ya se ha comentado anteriormente, el *Local Internet Breackout* es el escenario donde se usa la conexión a internet de una sede para que esta acceda a esta red, sin necesidad de navegar a través del CPD corporativo.

El caso (e), añade la posibilidad de que uno o varios de los UCS usen conectividad inalámbrica, por ejemplo, LTE, que a todos los efectos es un UCS más por el que se puede establecer un TVC.

1.2 Justificación del proyecto.

La tecnología SD-WAN cada vez tiene más relevancia en el sector de las telecomunicaciones y no para de crecer.

De acuerdo con el último informe de Omdia [3][4] sobre el equipamiento de data center, el mercado SD-WAN ha generado más de 600 millones de dólares de beneficio durante el último cuatrimestre de 2019, superando el doble de lo que generó en el mismo periodo de 2018. Además, el informe también indica que el beneficio para todo el periodo 2019 asciende a 2 billones de dólares, un 90% más que en 2018. Por último, este informe también pronostica que durante 2024 este mercado generará ingresos por más de 4,5 billones de dólares.

Las redes SD-WAN deben su gran previsión de crecimiento a las numerosas ventajas que aportan sobre una red WAN tradicional.

- Es mucho más sencilla la configuración para ajustar las políticas de calidad de servicio, pudiendo hacerlo de forma global, sin tener que acceder a cada nodo de la red.
- Las redes SD-WAN agilizan el despliegue de la red, acortan los tiempos de puesta en marcha de los equipos y automatizan la creación de VPNs.
- La gestión de los recursos de red es mucho más eficiente. Se pueden ajustar los flujos por diferentes caminos aprovechando al máximo la capacidad de los circuitos y sin necesidad de sobredimensionar los recursos.
- Los costes de operación son menores que en una red tradicional, tanto en los equipos hardware que pueden ser de propósito general como en el tiempo necesario para su puesta en marcha y reconfiguración. Los administradores de red suelen contratar caudales MPLS superiores al necesario para en caso de que aumente la necesidad no

tener que esperar a que el proveedor aumente el caudal ya que es un proceso lento. Esto conlleva un sobrecoste al sobredimensionar las conexiones MPLS, ya de por si caras. Además, gracias a la filosofía *overlay*, no es necesario disponer de enlaces dedicados para la conectividad entre sedes si no que cualquier circuito serviría.

Como desventajas de SDWAN, se puede indicar que ahora mismo adoptar esta tecnología puede resultar prematuro en según qué entornos.

También se puede considerar una desventaja el uso de circuitos de Internet, ya que la seguridad es menor y no hay garantía de calidad de servicio ya que es una red *best effort*. De todas formas, esta desventaja se suple estableciendo el servicio sobre un UCS MPLS.

1.3 Viabilidad económica.

Se va a realizar una valoración económica para una empresa que dispone, de unas oficinas centrales donde se encuentran todos sus empleados, un centro de datos *on-premise* donde aloja su propia infraestructura de virtualización y varios servidores en la nube de Amazon AWS.

Para el acceso entre las oficinas y el centro de datos *on-premise*, la empresa tiene contratada una red MPLS basada en ORLA. Asimismo, tanto en la sede de oficinas como en el datacenter *on-premise*, además de los circuitos MPLS tienen conexiones de FTTH para respaldo de este.

El datacenter en la nube tiene su propia conexión a la red, incluida en la cuota del servicio. [5]

Los costes de operación en los que incurre la empresa actualmente se detallan en la tabla 1,

Tabla 1

	Coste €/mes	Unidades	Total €/mes
Circuito MPLS 100 Mbps	601,38 €	2	1.202,76 €
FTTH 600/600	61,49 €	2	122,98 €
Amazon AWS	63,59 €	1	63,59 €
Mantenimiento MPLS	120,28 €	2	240,55 €
Mantenimiento FTTH	12,30 €	2	24,60 €
Mantenimiento Routers	129,08 €	2	258,15 €
Mano de obra	300,00 €	1	300,00 €
Datacenter	800,00 €	1	800,00 €

TOTAL por mes	3.012,63 €

Los costes mostrados son recurrentes y son los que está soportando la empresa actualmente. El coste de los circuitos MPLS se calcula sobre el precio de un circuito OLRA de 100 Mbps [6] con un margen del 50% para cubrir el resto de los componentes de la red y el margen del operador. El coste de la FTTH es el marcado por Movistar [7] para un circuito de esas características en el momento de escribir este texto. Los mantenimientos de los circuitos MPLS y FTTH se han calculado como un 20% del coste total de los mismos. El mantenimiento de los routers es el precio GPL de Cisco [8] para el router ISR-1109-4P. La mano de obra cubre los costes de una empresa externa que se encarga de mantener la red. El coste del datacenter está basado en una oferta de Interxion Madrid.

Se propone la migración a una red SD-WAN para minimizar costes. En este caso es necesario sustituir los equipos de red y, además, es muy recomendable sustituir los circuitos MPLS por circuitos FTTH.

Para la migración es necesaria una inversión inicial para adquirir el equipamiento necesario para desplegar el nodo SD-WAN. Como se verá más adelante, podemos encontrar un equipo de propósito general por entre 300 y 1000 €. Para este presupuesto se tomará un precio intermedio de 600€. Quedando los costes de inversión como se muestran en la tabla 2,

Tabla 2

	Coste	Unidades	Total
Nodos SD-WAN	600 €	2	1.200 €
Implantación	500 €	1	500 €
Alta FTTH	- €	2	0 €

TOTAL	1.700 €

Los costes de operación tras la migración a SD-WAN se detallan en la tabla 3,

Tabla 3

	Coste €/mes	Unidades	Total €/mes
Red MPLS 100 Mbps	601,38 €	0	- €
FTTH 600/600	61,49 €	4	245,96 €
Amazon AWS	63,59 €	1	63,59 €
Mantenimiento MPLS	120,28 €	0	- €
Mantenimiento FTTH	12,30 €	4	49,19 €
Mantenimiento Routers	129,08 €	0	- €
Mano de obra	150,00 €	1	150,00 €
Datacenter	800,00 €	1	800,00 €
Plataforma SD-WAN	- €	3	- €
Mantenimiento nodos	120,00 €	2	240,00 €

TOTAL	1.548,74 €

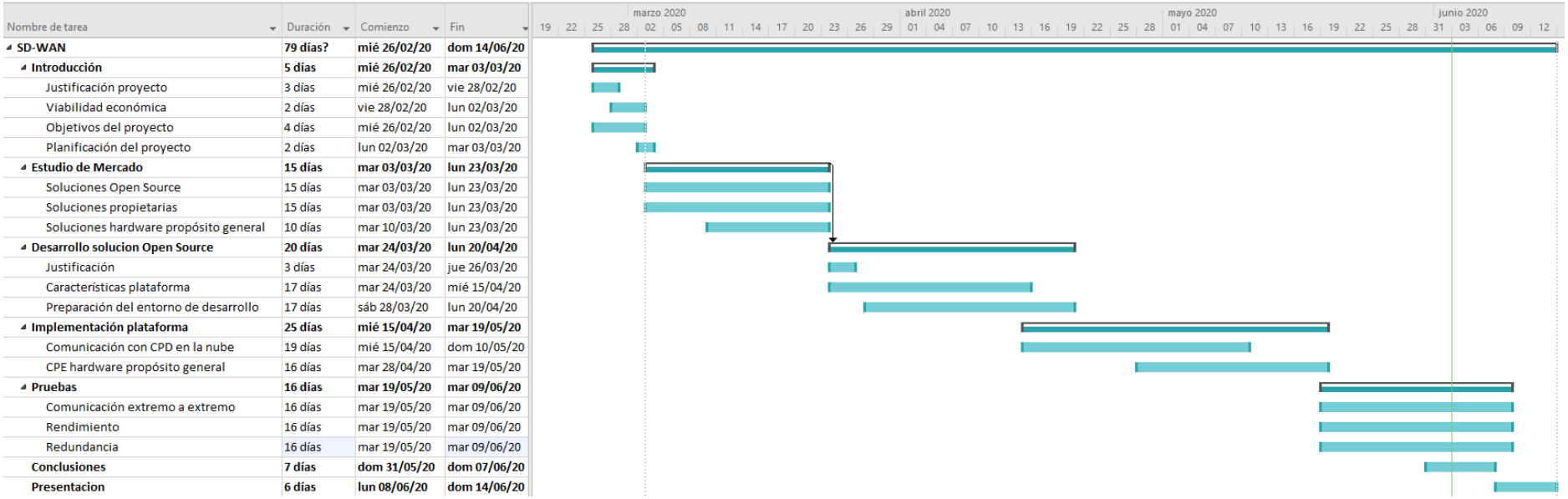
La plataforma SD-WAN es gratuita hasta los tres nodos.

Como se puede observar, en caso de aceptar la oferta, la empresa va a pasar a tener unos costes de operación que son la mitad de lo que actualmente paga con la solución MPLS.

1.4 Objetivos del proyecto.

- Definir que son las redes SD-WAN y de dónde vienen.
- Estudio de las soluciones que hay en el mercado y compararlas.
- Estudio de implantación de soluciones SDN/SDWAN en plataformas de propósito general.
- Elección de una solución SD-WAN para su desarrollo en el proyecto.
- Implementación de la plataforma SD-WAN elegida.
    - Dos nodos SD-WAN de los que colgarán varios servidores.
    - Configurar un uCPE para dotar de conectividad a la sede de la empresa con el CPD.
    - Uno de los nodos SD-WAN se debe instalar en la nube de Amazon AWS.
- Pruebas de comunicación entre nodos a través de SD-WAN.
    - Se requiere la comunicación extremo a extremo de una maquina en cada nodo. Pruebas de eco (ping) y camino seguido.
    - Pruebas de rendimiento, se busca ver si la infraestructura SD-WAN es limitante en latencia o ancho de banda.
    - Pruebas de redundancia. Los nodos con más de un enlace deben respaldar la conexión, para probarlo se simulará la caída del principal para ver que conmuta al secundario.

1.5 Planificación del proyecto.



Nombre de tarea	Duración	Comienzo	Fin
SD-WAN	**79 días?**	**mié 26/02/20**	**dom 14/06/20**
Introducción	**5 días**	**mié 26/02/20**	**mar 03/03/20**
Justificación proyecto	3 días	mié 26/02/20	vie 28/02/20
Viabilidad económica	2 días	vie 28/02/20	lun 02/03/20
Objetivos del proyecto	4 días	mié 26/02/20	lun 02/03/20
Planificación del proyecto	2 días	lun 02/03/20	mar 03/03/20
Estudio de Mercado	**15 días**	**mar 03/03/20**	**lun 23/03/20**
Soluciones Open Source	15 días	mar 03/03/20	lun 23/03/20
Soluciones propietarias	15 días	mar 03/03/20	lun 23/03/20
Soluciones hardware propósito general	10 días	mar 10/03/20	lun 23/03/20
Desarrollo solucion Open Source	**20 días**	**mar 24/03/20**	**lun 20/04/20**
Justificación	3 días	mar 24/03/20	jue 26/03/20
Características plataforma	17 días	mar 24/03/20	mié 15/04/20
Preparación del entorno de desarrollo	17 días	sáb 28/03/20	lun 20/04/20
Implementación plataforma	**25 días**	**mié 15/04/20**	**mar 19/05/20**
Comunicación con CPD en la nube	19 días	mié 15/04/20	dom 10/05/20
CPE hardware propósito general	16 días	mar 28/04/20	mar 19/05/20
Pruebas	**16 días**	**mar 19/05/20**	**mar 09/06/20**
Comunicación extremo a extremo	16 días	mar 19/05/20	mar 09/06/20
Rendimiento	16 días	mar 19/05/20	mar 09/06/20
Redundancia	16 días	mar 19/05/20	mar 09/06/20
Conclusiones	**7 días**	**dom 31/05/20**	**dom 07/06/20**
Presentacion	**6 días**	**lun 08/06/20**	**dom 14/06/20**

Figura 8 Planificación del proyecto

2. Estudio de mercado.

En el mercado se encuentran principalmente dos tipos de soluciones, las que son de código abierto u *Open Source* y las que son de código propietario. Las primeras suelen ser soluciones compatibles con todos los fabricantes que las quieran implementar y las segundas suelen ser soluciones completas de un fabricante, que no son interoperables con dispositivos o soluciones de terceros fabricantes.

2.1 Soluciones *Open Source*.

En primer lugar, es de interés mencionar que se entiende por código abierto. *Open Source* es un modelo de desarrollo de software de forma descentralizada que fomenta la colaboración altruista. El resultado de esta colaboración tiene que acabar en el acceso libre al código fuente, planos y demás documentación generada. [9]

2.1.1 OpenDaylight

OpenDaylight (ODL) es una plataforma modular abierta con la que se puede automatizar y gestionar redes de cualquier tamaño. El proyecto OpenDaylight surgió del movimiento SDN enfocado principalmente en la programabilidad de la red.

Es parte del proyecto Linux Fundation Networking, con un enfoque global, colaborativo y abierto a empresas. Actualmente está integrado en más de 35 soluciones empresariales para multitud de servicios.

Arquitectura ODL.

El core de la plataforma ODL es el MD-SAL (*Model-Driven Service Abstraction Layer*). En ODL los dispositivos y aplicaciones de red son representados como objetos o modelos, cuyas interacciones son procesadas en la capa de abstracción SAL.

Esta capa es el corazón de la arquitectura y permite la abstracción absoluta del comportamiento de los equipos de red y de otros servicios de la plataforma, al ser accedidos por las aplicaciones. La capa SAL es un mecanismo de intercambio de datos y adaptación entre modelos YANG, que representan los dispositivos de red, y las aplicaciones. Estos modelos contienen descripciones generales de dispositivos o de aplicaciones que permiten que se comuniquen entre ellos sin conocer los detalles del otro.

En la capa SAL existen diferentes roles, un productor implementa una API y proporciona los datos de esta; un consumidor usa la API y consume sus datos.

Por otro lado, en la capa SAL tenemos el interface *northbound*, usado para que las aplicaciones se conecten a esta, y el interface *southbound*, usado para la comunicación con los dispositivos de red.

En definitiva, la capa SAL empareja productores y consumidores de sus almacenes de datos para intercambiar información.

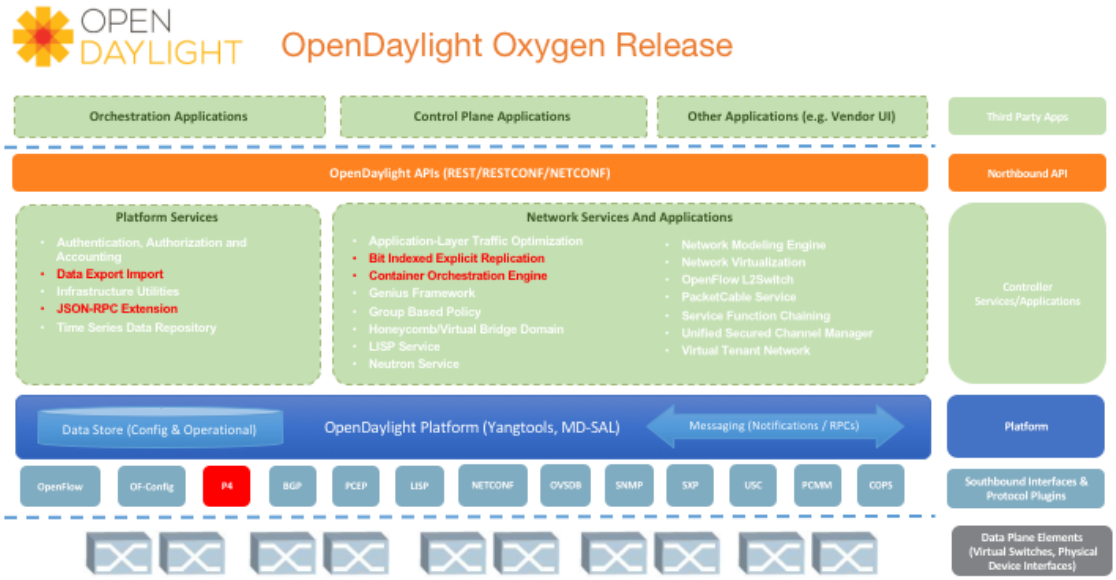


Figura 9 Arquitectura OpenDaylight

Modular y multiprotocolo.

La plataforma ODL está diseñada de una forma modular, lo que permite a los usuarios de está, desarrollar un controlador SDN que se ajuste a sus necesidades, permite aprovechar los servicios creados por otros y modificarlos en función de la necesidad o escribir los propios y compartirlos entre otras opciones. Soporta una gran cantidad de protocolos *southbound*, como por ejemplo OpenFlow, OVSDB, NETCONF, BGP y muchos más.

Debido a la naturaleza abierta de ODL se hace necesario un sistema para aislar cada característica de las otras y así evitar que puedan interferir entre ellas. Para ello, se hace uso de OSGi y Maven para construir características Karaf y sus interacciones.

S3P

La comunidad ODL dedica muchos esfuerzos para mejorar las áreas de seguridad, escalabilidad, estabilidad y rendimiento. Para ello hay establecidos grupos de prueba y de integración.

En la parte de seguridad, existe un gran equipo que responde a las vulnerabilidades inmediatamente, lo que ocurre gracias a que es una plataforma *Open Source* donde la comunidad de usuarios esta siempre alerta y cualquiera puede detectar y corregir una.

2.1.2 ONF CORD

CORD es un proyecto de la *Open Networking Fundation* (ONF) que integra varias de las soluciones que proporciona. El proyecto propone virtualizar la infraestructura de red que se despliega en las centrales locales de los operadores usando equipos de propósito general, a los que ONF llama POD. [10]

Las principales características de CORD son:

- Redes virtuales bajo demanda *Zero-Touch*.
- SLAs robustos y QoS para el tráfico empresarial.
- Pila de software que habilita la innovación de servicios.
- Equipos de cliente simples y de fácil mantenimiento.

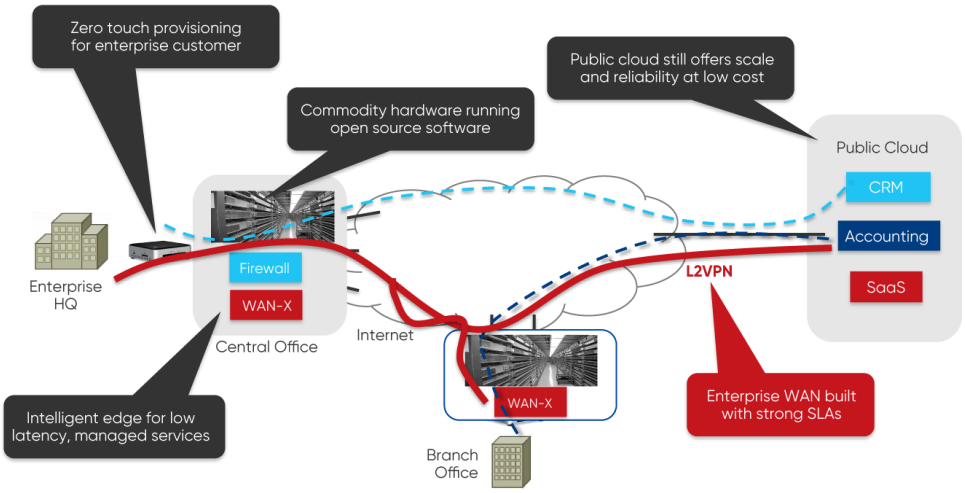


Figura 10 Red de datos usando CORD

CORD consta de tres elementos principales, OpenStack, ONOS y XOS. Los dos primeros se instalan en los POD.

Arquitectura

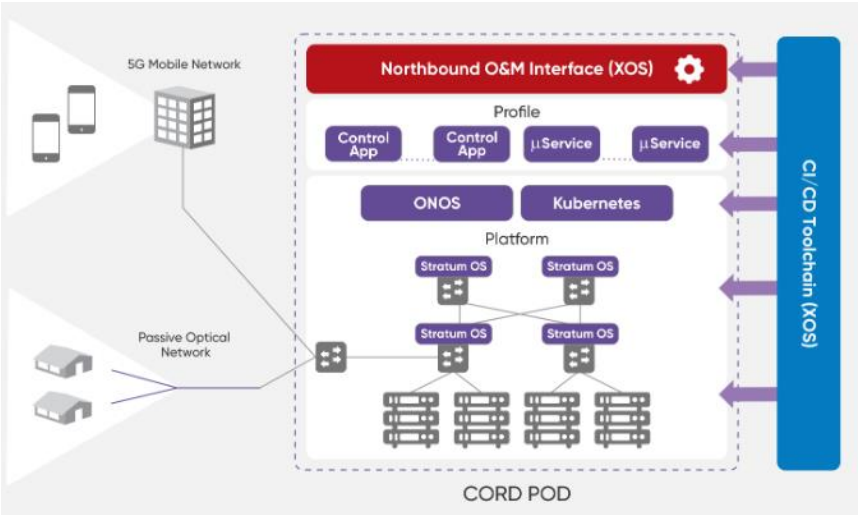


Figura 11 Arquitectura CORD

OpenStack es un sistema operativo *cloud* diseñado para controlar grandes centros de datos siguiendo el modelo, *Infrastructure as a Service* (IaaS). Este elemento es el encargado de proporcionar las máquinas y redes virtuales que implementan los servicios que proporciona CORD. Es decir, las VNF.

ONOS es un controlador SDN que gestiona los conmutadores de los POD para crear las redes virtuales. Es sobre este elemento donde se virtualizan las funciones de red, como el *routing*, el NAT, etc.

Por último, XOS es un entorno que permite componer servicios encadenando las funciones de red y aplicaciones que se despliegan sobre ONOS y OpenStack. En este escenario, XOS actúa como un orquestador NFV, gestionando el ciclo de vida de los servicios, funciones de red y aplicaciones.

2.1.3 FlexiWan

FlexiWAN nació con la idea de establecer una solución *Open Source* de SD-WAN 2.0, con un modelo de arquitectura y código abiertos. FlexiWan integra y coordina dentro su arquitectura diferentes soluciones *Open Source,* que incluyen vRouter, administración, orquestación y automatización para dotar la funcionalidad de SD-WAN. [11]

El desarrollo de esta solución de código abierto pretende democratizar el mercado SD-WAN, reduciendo drásticamente las barreras de entrada para que las empresas lo adopten y ofrezcan sus servicios.

Arquitectura.

La arquitectura de FlexiWAN se compone principalmente de dos dispositivos. flexiEdge, que es el nodo SD-WAN Edge desplegado en las diferentes ubicaciones. flexiManage, que es el sistema central de administración, el cual se conecta mediante API segura a flexiEdge para la administración y orquestación de este.

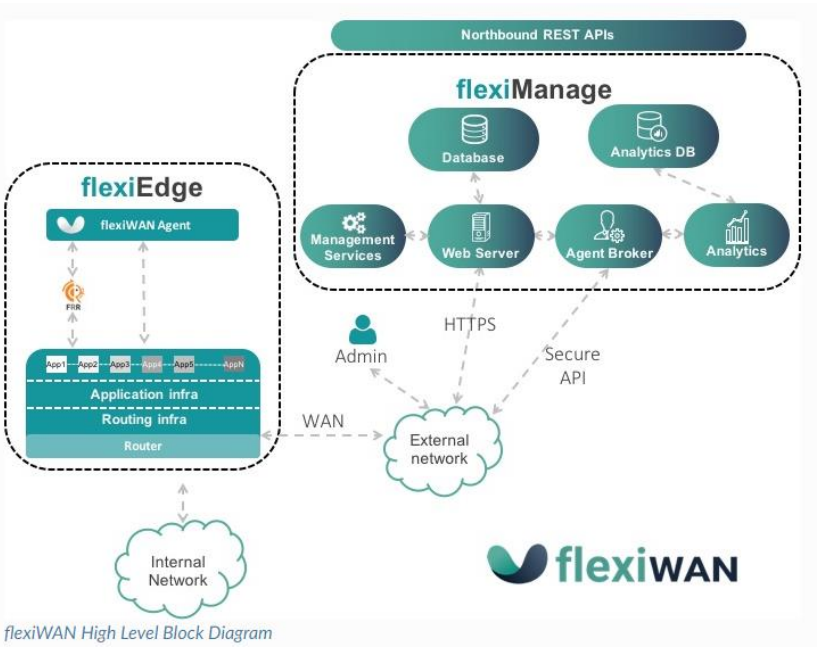


Figura 12 Arquitectura flexiWAN

flexiEdge comprende tres componentes clave:

- Infraestructura de router. Versión modificada de FD.io Vector Packet Processor (VRRP)
- Plano de control de routing. Free Range Routing (FRR)
- flexiWAN Agent. El componente software que se encarga de la comunicación a través de APIs seguras de flexiEdge y flexiManage.

flexiAgent se conecta con flexiManage usando un web socket bidireccional para la configuración y estadísticas que soporta las siguientes características:

- Comandos GET JSON simplificados
- APIs individuales provisionadas en Linux y en el Router.
- Configuración de almacenamiento del tipo clave-valor.
- Orquesta la secuencia de ejecución entre varios elementos.
- Mantiene el orden de configuración.
- Restaura el ultimo estado tras un reinicio.

- Procesamiento de las transacciones y posibilidad de restauración en caso de fallo.
- Monitorización de componentes y posibilidad de restaurarlos en caso de fallo.
- Proporcionar una estructura JSON a toda la configuración.
- Proporcionar comandos de consola para la resolución de errores.

flexiManage corre en un servidor web para la administración de la red. A través de su portal, un administrador de red puede gestionar todos los dispositivos que componen la red. Este agente es el encargado de la comunicación de todos los dispositivos flexiEdge que componen la red. También se encarga de la recopilación de las estadísticas de los dispositivos flexiEdge, los analiza y proporciona los informes al administrador.

En el momento de escribir este texto, FlexiWAN va por la versión 1.2.2. Es todavía una versión en desarrollo de lo que pretenden para el producto final pero ya dispone de características suficientes para su despliegue. Esta versión soporta *multi-WAN* y *multi-LAN* y se pretende reforzar en un futuro próximo con funcionen de *routing* basado en políticas y otras funciones.

Características soportadas, versión 1.2.2

- Instalación basada en Debian.
- Aprovisionamiento *Zero-Touch.*
- Cuentas y usuarios *Multi-Tenant*.
- Inventario basado en la organización.
- IPSec sobre túneles VxLAN.
- Métricas de calidad del túnel.
- *Internet Breakout*.
- Configuración de rutas estáticas.
- Cambios dinámicos de la configuración de flexiEdge.
- Monitorización mejorada de flexiEdge.
- Detección de errores y notificación de flexiEdge.
- *Dashboards*.
- Actualizaciones automáticas.
- Acceso a APIs *northbound*.
- Estabilidad y calidad de la plataforma mejoradas.

Características previstas en nuevas versiones.

- Selección de enlace basado en políticas.
- Identificación de aplicaciones.
- QoS avanzado.
- Más opciones de NAT.
- Otras muchas mejoras.

2.2 Soluciones propietarias.

2.2.1 Nuage Networks

Nuage networks es una empresa que pertenece a Nokia y es la encargada de desarrollar las tecnologías SDN. En concreto, para SD-WAN 2.0 lo hace a través de su producto VNS*, Virtualized Networks Services*. [12]

VNS se ofrece como una plataforma de gestión de redes extremo a extremo, muy potente y con el foco en la seguridad. Ofrece entorno de nubes múltiples y visibilidad y control de la red desde un solo panel de administración. Con su propuesta, Nuage quiere garantizar la conectividad, no

solo entre las sedes de un cliente, sino también con centros de datos públicos y privados y proveedores de SaaS o IaaS, todo ello desde su plataforma de gestión.

Nuage automatiza la provisión, configuración y gestión de las redes WAN para así poder ofrecer a sus clientes calidad de servicio adaptado a las políticas de negocio y seguridad de las aplicaciones, todo ello a bajo coste.

Arquitectura

La arquitectura de VNS corre bajo la plataforma *Virtualized Services Platform* (VSP) que también habilita el framework *Virtualized Cloud Services* (VCS).

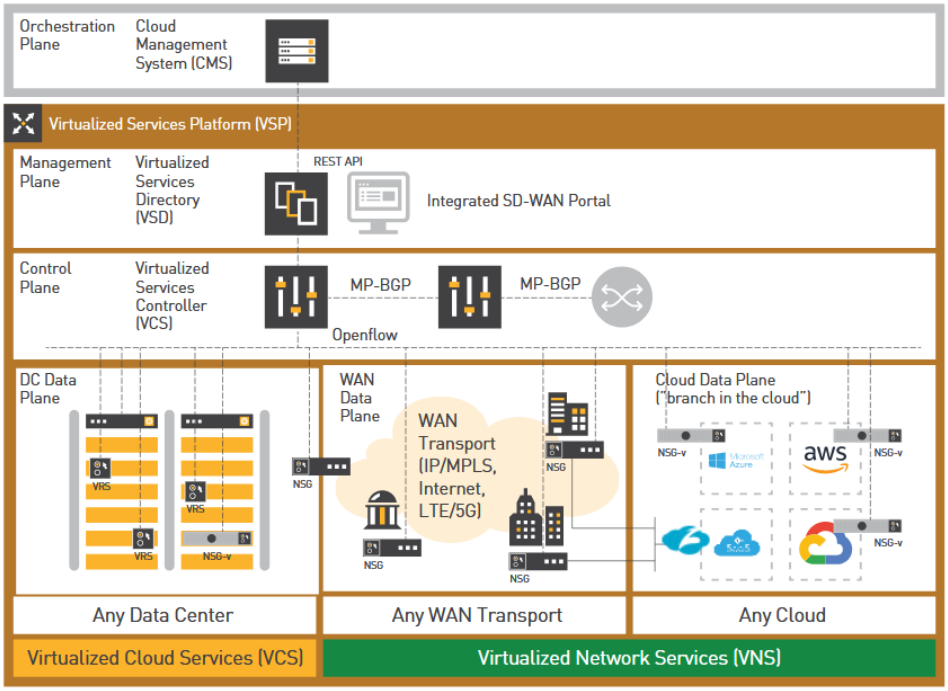


Figura 13 Arquitectura VSP/VNS

Para administrar esta plataforma se usa el framework *Virtualized Services Directory* (VSD) que se encarga de programar las políticas y del motor de análisis. Este *framework* proporciona flexibilidad en la política de red lo que habilita al administrador de red a definir, aplicar y asegurar las políticas de negocio de la red con un entorno de administración amigable. VSD permite la administración basada en roles y habilita la administración centralizada de la configuración de red con un interface gráfico intuitivo.

Gracias a VSD, los proveedores de red y las empresas tienen un entorno centralizado en el que pueden ver y cambiar las políticas de la red, desplegar nuevas en un solo sitio o en varios o según el tipo de red. VSD también se encarga de recopilar todos los reportes del tráfico de la red. Todo ello queda almacenado en una base de datos *Elasticsearch* que facilita el minado de datos y el rendimiento en los reportes.

Las funciones de red para el servicio se seleccionan utilizando la tienda de funciones de red de VSD. Esta tienda proporciona un conjunto integral de funciones de red comunes, como *stateful firewall* de capa 4, encadenamiento de servicios, IPSec, NAT/PAT, inteligencia empresarial, balanceo de carga, gestión de direcciones IP y servicios de nombres de dominio que se pueden insertar directamente en el servicio de red como opciones de servicio escalonadas. El VSD también es compatible con el despliegue de VNF, la gestión de repositorios y la gestión del ciclo

de vida de VNF, lo que complementa la capacidad de los NSG para alojar VNF de terceros. Esto reduce la necesidad de implementar elementos de red dedicados en ubicaciones remotas, lo que simplifica enormemente el modelo de operación para la empresa.

VSD también soporta una interfaz norte RESTful API que permite a todas las funciones una integración total con los sistemas de administración *cloud*.

También incluye un portal de cliente SD-WAN que permite a los proveedores de servicios establecer perfiles de cliente para que cada usuario pueda administrar sus funciones de red. Este portal incluye paneles separados por empresa y profundidad de análisis.

El portal de clientes SD-WAN ofrece flujos de trabajo sencillos para la gestión del ciclo de vida completo de las capas 2 y 3 de las VPN y sitios remotos. Para los proveedores de servicios, el portal proporciona gestión de perfiles de clientes. Los proveedores de servicios o los administradores de red de los clientes pueden crear un control de acceso detallado de acuerdo con la estrategia de control de acceso basado en roles de una organización.

Funciones del plano de control.

El Controlador de servicios virtualizados (VSC) es el controlador SDN más poderoso de la industria, según Nokia. Funciona como un plano de control de red centralizado y robusto para los servicios de red, manteniendo una vista completa de la red y las topologías de servicio. El VSC se basa en el sistema operativo SR y se beneficia de su rendimiento, robustez y posibilidades de escalado. A través del VSC, se establecen enrutamientos virtuales y funciones de conmutación para programar el plano de reenvío de red utilizando el protocolo OpenFlow. Se pueden federar varias instancias de VSC dentro y a través de la red aprovechando el protocolo MP-BGP, una tecnología de red probada y altamente escalable que permite que el servicio de red crezca con los requisitos del negocio.

Funciones del plano de datos.

El NSG constituye el plano de reenvío de red para VNS y SD-WAN 2.0. Encapsula el tráfico de datos, aplica las políticas de red de Capa 2 a Capa 4 y establece las VPN de superposición de Capa 2 o Capa 3 según lo definido por el VSD. El NSG también tiene posibilidad de Wi-Fi incorporado y soporte de LTE. En el NSG, los servicios avanzados se pueden activar, incluidas las funciones de red como el equilibrio de carga, el encadenamiento de servicios, la inteligencia empresarial y NAT / PAT. Las características de seguridad inherentes también se pueden habilitar, incluidos los túneles IPSec, los firewalls *stateful* de Capa 4 y Capa 7 y el filtrado de URL de Capa 7. El NSG también es compatible con máquinas virtuales y contenedores VNF, complementando el soporte de VSD para la incorporación de VNF, la gestión de repositorios y la gestión del ciclo de vida de VNF. Estos servicios se pueden aplicar a los NSG de manera centralizada, en todo el servicio o en función de un modelo de implementación específico de la ubicación. La implementación del NSG utiliza un proceso de arranque automático que incluye varias opciones de autenticación de múltiples factores. La naturaleza automatizada de esta función de arranque reduce la necesidad de recursos de red especializados en ubicaciones remotas. En la mayoría de los casos, el personal del sitio remoto sin cualificación técnica puede enchufar y olvidarse de lo demás. Esto reduce el costo de implementación del servicio.

2.2.2 Cisco SD-WAN

La solución de Cisco para el mundo SD-WAN viene de la mano de Viptela, empresa que adquirió en 2017 y que ha usado para desarrollar esta plataforma. [13]

Arquitectura

La arquitectura de cisco se divide en 4 planos, datos, control, administración y orquestación.

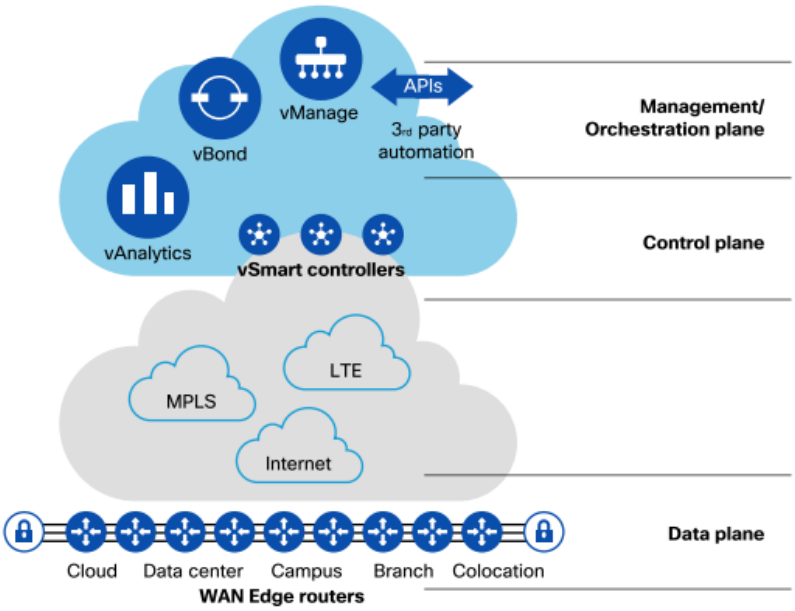


Figura 14 Arquitectura Cisco SD-WAN

Esta solución tiene cuatro componentes clave, vManage, vBond, vSmart y los routers SD-WAN Edge. [14]

Cisco vManage se encuentra en el plano de administración y es el encargado de proveer de un interfaz de usuario donde configurar los dispositivos, provisionarlos, monitorizar su actividad o en caso de fallos, ayudar en su resolución. También es el encargado de crear las políticas a aplicar en la red. Este componente se distribuye como panel único para una sola organización o como *multitenant* para que una organización pueda desplegar diferentes clientes desde un mismo panel de administración.

Cisco vBond pertenece al plano de orquestación y es el máximo responsable del aprovisionamiento *Zero-Touch,* así como la primera línea de autenticación, distribución de la información de control/administración y gestión del NAT *traversal*. Cuando un router arranca por primera vez, vBond se encarga de llevar el dispositivo a la SD-WAN *fabric*. Además, también es responsable de conocer la topología de la red y compartirla con todos los dispositivos.

Cisco vSmart es el cerebro de la solución y reside en el plano de control. Las políticas creadas con vManage son desplegadas y vigiladas desde vSmart. Cuando un sitio remoto conecta, su información de enrutamiento se intercambia con este componente en lugar de con el resto de las sedes. Es por ello, que con las políticas establecidas se puede ajustar como se conectan las sedes entre sí, pudiendo ajustar esta comunicación de forma individualizada. Esta configuración se intercambia a través del protocolo Overlay Management Protocol (OMP).

Cisco WAN Edge routers. Cisco propone variedad de equipos para su SD-WAN Edge según las necesidades de la sede. Existen dispositivos para diferentes niveles de funcionalidades, rendimiento y conectividad.

Todos estos componentes se combinan para formar el Cisco SD-WAN fabric.

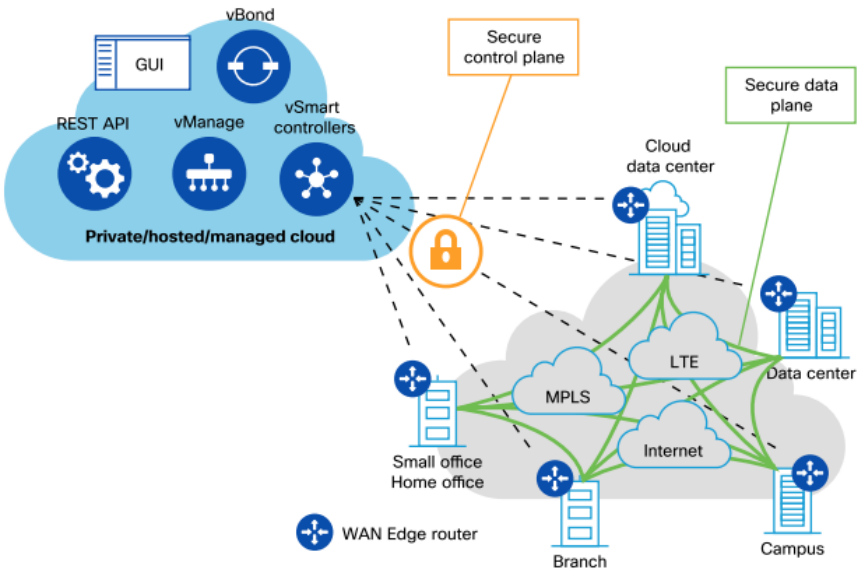


Figura 15 topología Cisco SD-WAN

En la figura 15 se muestra la relación entre todos estos componentes. Los router WAN Edge establecen túneles IPSEC entre ellos para formar la red *overlay* SD-WAN. Adicionalmente, también se establece un canal de control entre estos y cada uno de los elementos de control. A través de este canal de control cada componente recibe la configuración, aprovisionamiento e información de rutas.

Cisco usa su propio protocolo *southbound*, OMP, que se encarga de administrar la red *overlay* y corre entre los routers Edge y los controladores vSmart donde toda la información del plano de control se intercambia de forma segura. La configuración por defecto indica que OMP establezca una red *overlay* totalmente mallada, donde cada router Edge se conecta con todos los demás.

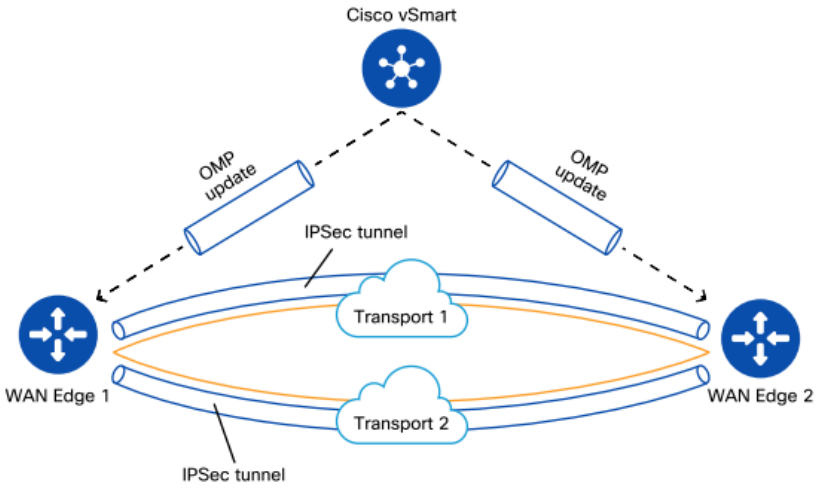


Figura 16 Intercambio de información a través de OMP

2.3 Soluciones compatibles con plataformas de propósito general.

En este contexto, una plataforma de propósito general, *general purpose* en inglés, es una pieza de hardware que no está diseñada específicamente para conmutar paquetes. Es un computador x86 normal que se utiliza para labores de red debido a su coste más asequible y su mayor facilidad de mantenimiento.

Normalmente todas las soluciones SDN *Open Source* se despliegan sobre plataformas de propósito general, denominadas uCPE, y las propietarias suelen utilizar hardware específico. El mejor ejemplo de esto último es Cisco, donde todos sus nodos SD-WAN Edge son routers con el software de Cisco, IOS.

Dos de los fabricantes de uCPE de coste más bajo son Edge-Core Networks y Advantech.

El modelo más económico de Edge-Core es el SAF4100I [15], un equipo con las siguientes características:

- CPU Intel D-1500 Series SoC (Broadwell-DE)
- Memory 2x DDR4 SODIMM slots (max 32GB). Default 4GB
- Ethernet Interfaces:
    - 1 x RJ-45 and 1 x SFP GbE Port for WAN, through 2 x Intel I210 controllers
    - 1 x SFP+ 10GbE port, through CPU built-in Ethernet controller
    - 8 x RJ-45 GbE port for LAN, through Marvell 88E6190X Ethernet Switch Hub
- Local Storage: 1 M.2 SATA SSD, default 32 GB
- Advanced Technology: Virtualization - VT-d, VT-x, SR-IOV, Security – H/W TPM 1.2, AES-NI DPDK

Figura 17 SAF4100I

Este modelo se puede conseguir en internet por algo más de 1400$ [16]

El modelo de Advantech más económico es el FWA-T011 [17]:

- CPU 2 Cores Intel Celeron J3355
- RAM 8GB DDR3L
- 4 x 10/100/1000BASE-T RJ45
- 1 puerto HDMI
- Almacenamiento SSD
- Virtualización VT-x, VT-d

Este modelo se puede encontrar en internet por 310$ [19]

Figura 18 FWA-T011

3. Desarrollo de una solución *Open Source.*

3.1 Justificación de la plataforma elegida.

Todas las plataformas vistas hasta ahora pueden ser válidas para desarrollar este proyecto, pero hay que tener en cuenta varios factores.

- Presupuesto.
- Acceso a documentación, software y equipamiento.
- Tiempo necesario para la implantación.
- Calidad de la solución aportada.

Lo más importante que se busca ofrecer es que el coste sea el menor posible sin merma en la calidad de la solución y ajustado a las necesidades del cliente.

Para poder elegir una solución concreta se realiza una ponderación de los siguientes indicadores,

- Inversión inicial (*hardware*, mano de obra, etc.)
    - En este indicador el valor se basa en que las soluciones Open Source se pueden implementar sobre plataformas de propósito general que son más baratas que las propietarias.
- Costes de operación (Cuotas de servicio, mantenimiento, complejidad de la plataforma, etc.)
    - Los costes de operación se ven afectados sobre todo por el precio de las licencias, por eso las soluciones propietarias tendrán menor puntuación en este apartado.
- Plataforma abierta (Código abierto, APIs propias, modificar el código, etc.)
    - Este indicador se basa en lo interoperable que es la plataforma con software de terceros y el acceso a su código, una vez más, cuanto más abierta es la plataforma más interoperable será.
- Documentación (Acceso libre a la documentación y soporte de usuarios).
    - Los fabricantes crean sus propias bases de conocimiento, este indicador refleja la calidad y disponibilidad de la documentación.
- Plazo de implantación (Obtener el hardware y el software y desplegar la solución).
    - Este indicador trata de determinar la rapidez con que la plataforma se puede instalar, esto dependerá de la cadena de suministro y la complejidad para la implantación.
- Fiabilidad (Estabilidad de la plataforma).
    - En este indicador se refleja la calidad de la solución de una forma subjetiva, basándose en la trayectoria del fabricante y la madurez de la tecnología.
- Escalabilidad (posibilidad de crecimiento de la red).
    - Se valora según la facilidad de implementar nodo nuevos en la red.
- Enfoque.
    - Como se ajusta la solución a las necesidades del cliente, una solución para conectar tres nodos, uno de ellos en la nube.

Tal como se ha comentado, se busca una solución barata, robusta y enfocada en el problema del cliente, por ello los puntos con mayor valoración son los de inversión, operación, fiabilidad y enfoque.

En la tabla se muestran todos estos indicadores con una valoración de 0 a 9 para cada plataforma. Además, cada indicador tiene su ponderador, siendo 0 el peor valor y 1 el mejor. Del resultado de multiplicar el ponderador por el valor de cada indicador y la suma de todos ellos resulta la puntuación total para cada solución.

Tabla 4

		ODL		ONF CORD		FlexiWan		Nuague		Cisco	
Indicador	Pond.	Valor	Pond	Valor	Pond	Valor	Pond	Valor	Pond	Valor	Pond
Inversión	1	8	8	8	8	8	8	3	3	3	3
Operación	1	7	7	6	6	9	9	4	4	4	4
Abierta	0,8	8	6,4	8	6,4	6	4,8	4	3,2	3	2,4
Document.	0,6	8	4,8	7	4,2	6	3,6	7	4,2	9	5,4
Plazo	0,7	7	4,9	7	4,9	7	4,9	5	3,5	5	3,5
Fiabilidad	1	8	8	8	8	7	7	8	8	9	9
Escalabilidad	0,6	9	5,4	9	5,4	9	5,4	9	5,4	9	5,4
Enfoque	1	6	6	4	4	9	9	7	7	8	8
TOTAL			50,5		46,9		51,7		38,3		40,7

Desde otra perspectiva, a través de un gráfico radial, se pueden ver de forma más clara las fortalezas y debilidades de cada solución.

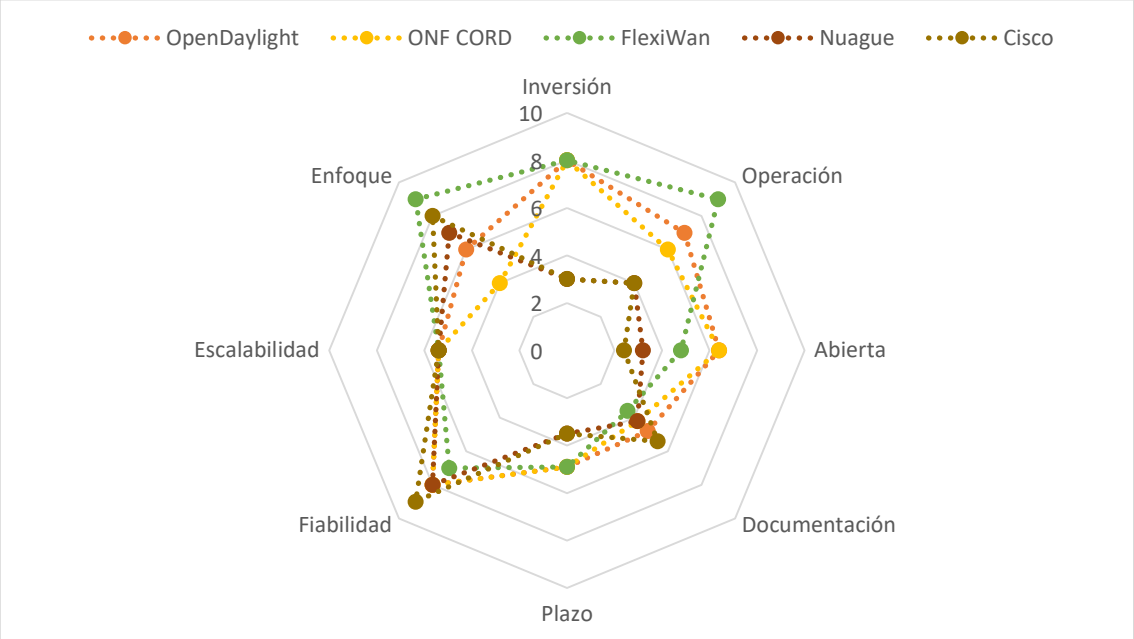


Figura 19 Comparativa plataformas

De estos datos se extraen varias conclusiones;

Por un lado, si se comparan las soluciones Open Source con las propietarias, las soluciones Open Source salen mejor paradas gracias a que para este proyecto se priman los costes y las soluciones propietarias tienen una alta carga recurrente debido a las licencias.

Por otro lado, las soluciones propietarias no tienen por qué ser mucho más fiables que las abiertas. Se puede creer que una solución propietaria al estar soportada por un fabricante es más estable, pero es cierto que las soluciones abiertas, al tener su código disponible para cualquiera, hace que sean altamente fiables, sobre todo según va madurando el proyecto.

Sobre las soluciones propuestas, la mejor valorada ha sido la de flexiWan. Como se ha comentado anteriormente, al ser una solución de código abierto su coste es inferior y no tiene por qué verse afectada en la calidad del producto. En la comparación entre plataformas Open Source, flexiWan ha terminado mejor parada gracias a que está enfocada en lo que realmente busca el cliente y a su facilidad de operación.

En conclusión, teniendo en cuenta las valoraciones de las plataformas se opta por ofrecer al cliente la solución Open Source de flexiWan.

3.2 Características de la plataforma.

En la figura 20 se encuentran, a modo de resumen de lo ya visto, los diferentes paquetes que forman la solución de FlexiWAN.

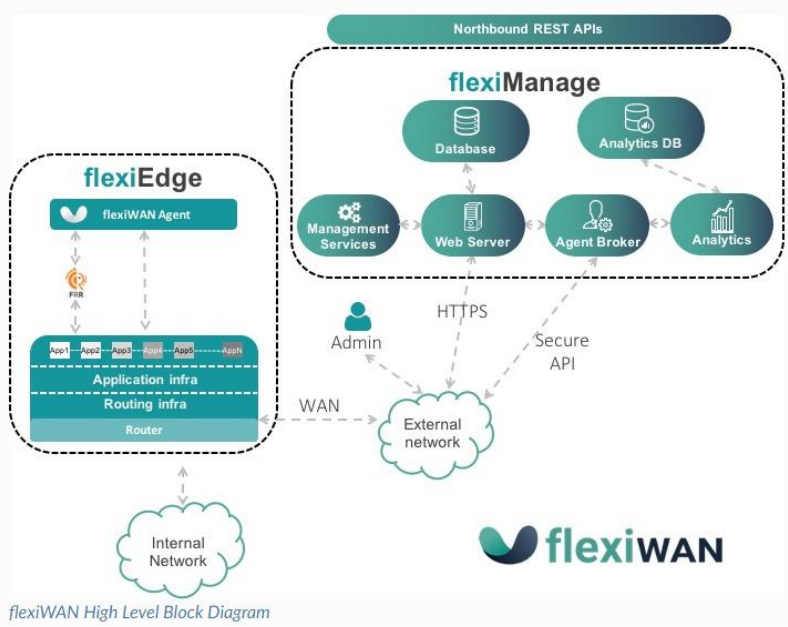


Figura 20 Arquitectura flexiWAN

FlexiEdge es un paquete de software para una distribución Linux Ubuntu 18.04 LTS y se puede instalar sobre varias plataformas:

- Bare metal
- Máquina virtual
- En la nube de Amazon AWS

La instalación está paquetizada por lo que es posible instalarlo en versiones de Ubuntu personalizadas. La instalación añade todos los componentes necesarios como servicios del sistema Ubuntu.

Durante la instalación, el software comprueba que tanto el hardware como el software cumplen con los requisitos.

FlexiEdge usa un token creado en flexiManage para registrarse. Durante la instalación se genera un token individual a partir de este que se usara para la autenticación con flexiManage, todo ello sin intervención del usuario en la provisión (*Zero-Touch provisioning*).

FlexiManage es el orquestador de la red SD-WAN. En él, es posible crear cuentas con diferentes niveles de administración dentro de la organización o cuentas para diferentes organizaciones, dándole capacidad *multi-tenant*. Cabe destacar que cada organización tiene su espacio aislado de las demás, no es posible ver los inventarios de otras o administrarlas.

Cada organización tiene un inventario que contiene los nodos SD-WAN (flexiEdge), los túneles y los tokens. Las operaciones de red se ejecutan a nivel de organización. Aunque las organizaciones estas aisladas entre sí, es posible asignar usuarios con permisos sobre varias.

Los dispositivos flexiEdge registrados son mostrados en flexiManage con su estado, donde el orquestador recoge información sobre ellos. Esta información incluye parámetros generales de configuración, interfaces, estadísticas de monitorización, rutas, logs y configuración interna.

flexiWAN soporta túneles IPSEC sobre VxLAN siguiendo la estructura de la figura 21,

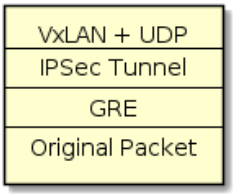


Figura 21 Túnel IPSEC sobre VxLAN

Los túneles se conectan a través del interfaz WAN. Las redes LAN son anunciadas a través de OSPF. Cada uno de estos túneles es creado entre dos dispositivos flexiEdge usando su interfaz *loopback* en el rango 10.100.X.Y/31

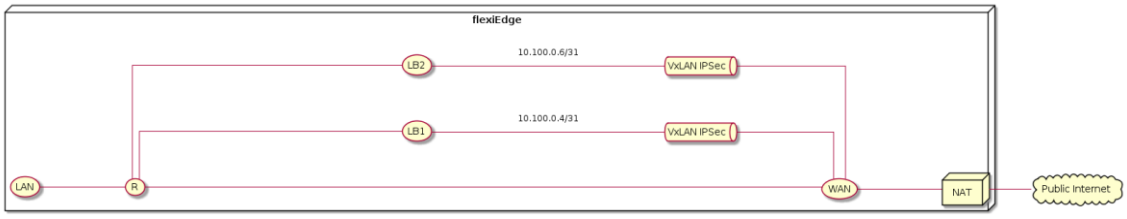


Figura 22 Topología túnel

Como se aprecia en la figura 22, en el tráfico enturado a través del túnel usa como *nexthop* la dirección IP *loopback* del otro extremo del túnel. Los paquetes que viajan a través del túnel usan VxLAN por el puerto 4789 UDP.

En algunos entornos, es necesario establecer rutas estáticas, para ello existe la posibilidad de establecerlas a través de los túneles desde flexiManage.

Además, el sistema permite configurar el túnel de forma flexible, como red totalmente mallada, *hub and spoke* o cualquier otra combinación.

flexiWAN permite encaminar el tráfico por los túneles en función de la etiqueta. Esta característica se denomina *Path Labels* y consiste en marcar los interfaces WAN por los que se quiere encaminar el tráfico entre dos flexiEdge. Está previsto en versiones futuras que con las etiquetas se pueda establecer el tipo de tráfico a encaminar.

Los parámetros de los túneles IPSEC son, protocolo IPSEC, modo túnel, algoritmos AES-CBC-128 para encriptación y SHA-256-128 para integridad.

El orquestador flexiManage se encarga de medir la conectividad, latencia y perdida de los túneles. Cada nodo flexiEdge lanza paquetes ICMP echo cada segundo para medir el tiempo de ida y vuelta (RTT) y las pérdidas que ocurren en estos.

Cualquiera de los dispositivos flexiEdge es capaz de encaminar tráfico directamente a internet usando la característica de *Internet Breakout*. El tráfico de internet pasa a través del NAT.

Otra de las características que permite el producto es la de NAT Traversal en modo 1:1 NAT, usando un reenvío de puertos o cuando el equipo que hace NAT preserva el puerto origen UDP.

flexiManage se encarga de recolectar estadísticas por dispositivo y muestra los bits por segundo o paquetes por segundo en el *dashboard*. Además de monitorizar también es el encargado de notificar, como, por ejemplo:

- Cuando es necesario actualizar el software de algún componente.
- Cuando un dispositivo se ha desconectado.
- Cuando un router deja de funcionar.
- Cuando el retardo de un túnel sea superior a 100ms
- Cuando las pérdidas de un túnel sean superiores al 50%

El dashboard de flexiManage presenta información relativa a la conectividad de red y el estado de cada túnel, al total del ancho de banda usado y los paquetes por segundo.

Por defecto, flexiWAN permite registrar y administrar 3 nodos flexiEdge de manera gratuita en su versión comercial del producto.

Las actualizaciones de funcionalidades y corrección de bugs se despliegan de forma periódica, se notifica al administrador vía email para que establezca una ventana de actualización.

flexiWAN soporta *northbound* APIs de tipo REST para administrar y provisionar las redes.

Algunos de los paquetes *Open Source* usados en flexiWAN son,

- VPP. *Vector Packet Procesing*, es un *framework* que provee funcionalidades de *router/switch*.
- Sqlitedict. Bases de datos para el nodo flexiEdge.
- DPDK. Kit de desarrollo del plano de datos para la acelerar el procesamiento de paquetes.
- *Libyang*. Parseador de lenguaje Yang.
- Python. Lenguaje de programación ampliamente utilizado en desarrollo de aplicaciones de red.
- Fping. Programa de ping avanzado que da más información que el ping del sistema.
- FRR. Es una suite de protocolos IP para Linux que incluye BGP, RIP y OSPF entre otros.
- MongoDB, base de datos para flexiManage.
- Sendmail. Encargado de enviar emails.

3.3 Preparación del entorno de desarrollo.

En primer lugar, es necesario conocer los requisitos mínimos del nodo flexiEdge para un entorno en producción.

- Ubuntu 18.04 LTS.
- CPU de dos cores y procesador de 64 bits.
- Mínimo 4GB de RAM, aunque es recomendable disponer de 8GB.
- Al menos 16GB de espacio en disco.
- Dos interfaces de red PCIe que soporten DPDK.
- Acceso a internet a través del interfaz WAN.

El nodo flexiEdge puede ser instalado en una máquina virtual, en una plataforma de propósito general o en la nube.

En la figura 23 se muestra un ejemplo de la topología de dos nodos, uno de ellos instalado *on-premise* y el otro en la nube de Amazon.

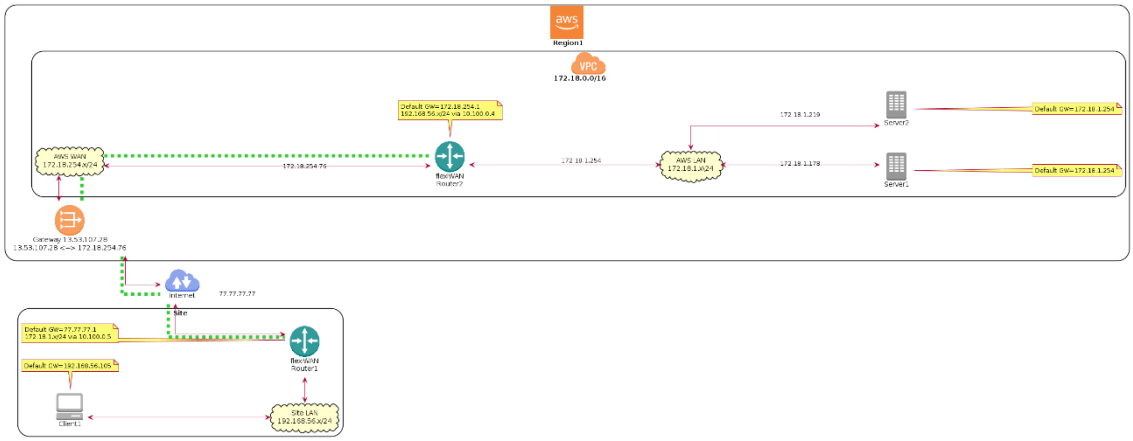


Figura 23 Topología de red entre nube y red local

3.3.1 Instalación de flexiEdge.

La instalación se puede realizar de dos formas, según el entorno elegido, una instalación atendida en una plataforma de propósito general o en un entorno de virtualización o prácticamente desatendida en la nube de Amazon AWS.

Para la instalación atendida es necesario disponer de un servidor con Ubuntu 18.04, en el que se ejecutara el script de instalación.

El script de instalación se encarga de actualizar los repositorios locales de Ubuntu para añadir los paquetes necesarios y de revisar si se cumplen los requisitos de instalación.

Tras la descarga y ejecución del script se procede a la instalación de flexiEdge en el nodo SD-WAN a través del comando de instalación de paquetes de Ubuntu.

Con estos dos simples pasos se ha conseguido instalar el software necesario para flexiEdge. Tras la instalación es importante revisar que el dispositivo cumple todos los puntos críticos.

Para que el nodo flexiEdge sea capaz de conectar con el orquestador flexiManage de la organización es necesario agregar el token al sistema. Para ello basta con copiar la cadena de texto que proporciona flexiManage a la ruta especificada de flexiEdge,

Tras la instalación se debe ejecutar la utilidad *fwsystem_checker* para reparar potenciales problemas de configuración y preparar el sistema para la operación de flexiEdge. Esta utilidad analizará el hardware, el sistema operativo del host y mostrará los resultados. Para que flexiEdge funcione de manera correcta es importante que los resultados referentes al hardware sean satisfactorios. Los fallos relativos al software suelen ser fácilmente solucionables directamente desde la utilidad.

Para la instalación desatendida en AWS es necesario un entorno Linux con Ansible que sea capaz de acceder a internet con las credenciales de flexiWAN. La instalación se ejecuta a través de un script de Ansible.

Con el entorno preparado, solo es necesario descargar e iniciar el script de instalación, donde en una solo línea de comando se añaden todos los parámetros necesarios para instalar flexiEdge.

Un comando de ejemplo sería el siguiente,

```
ansible-playbook ec2_create_customer.yml --extra-vars "region=us-east-2 vpc_name=VPC
vpc_cidr_block=172.18.0.0/16 cidr_lan=172.18.1.0/24 cidr_wan=172.18.254.0/24
lan_ip_address=172.18.1.254 flexiwan_token=ey****** stack=Ohio"
```

El script se encarga de crear toda la infraestructura necesaria en Amazon AWS y de instalar el software flexiEdge.

4. Implementación de la plataforma SD-WAN.

Para la implementación de la plataforma se han desplegado tres nodos SD-WAN flexiEdge, uno instalado en Amazon AWS (NODO SD1 AWS) y otros dos en un entorno de virtualización usando VMWare ESXi (NODO SD2 VMWARE) y (NODO SD3 VMWARE).

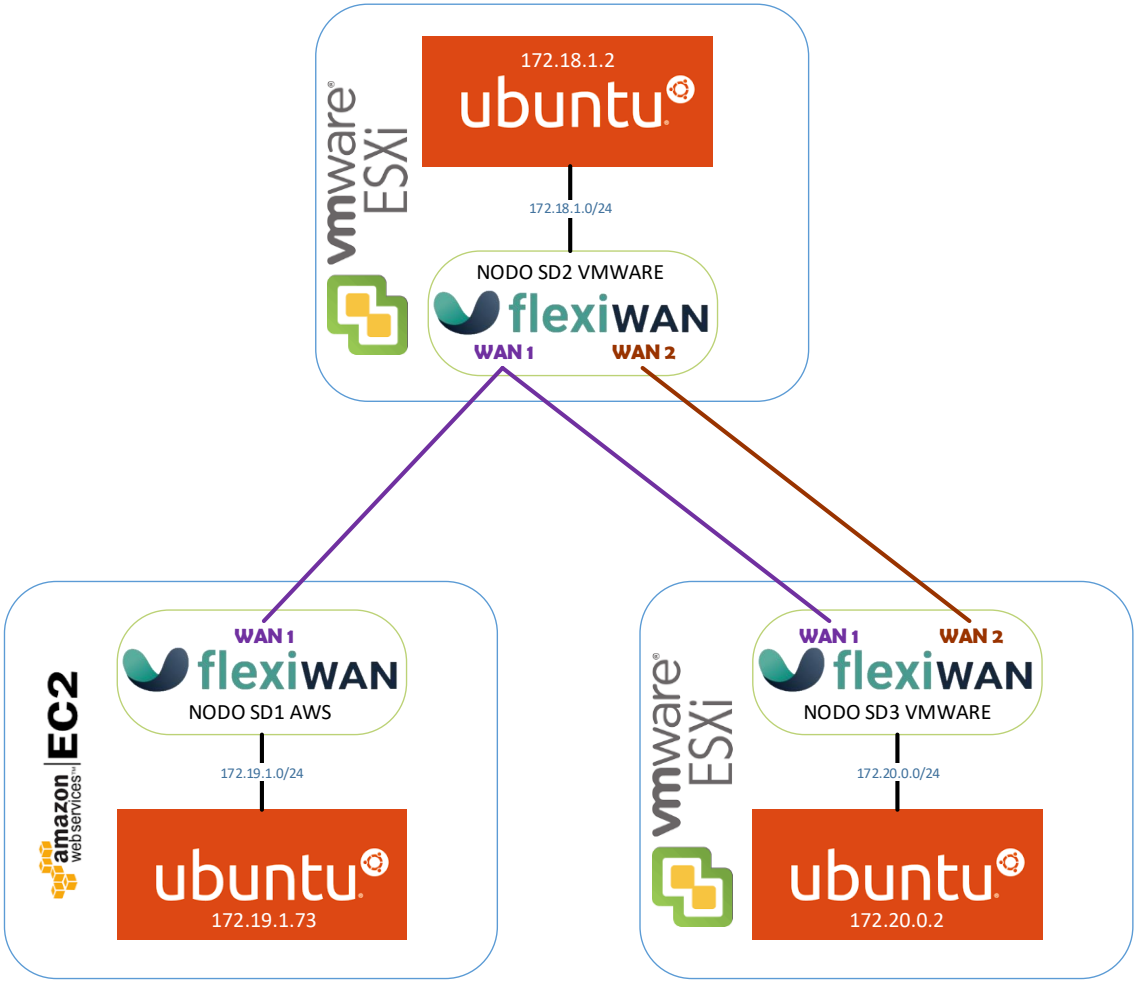


Figura 24 topología de red

Los nodos SD2 y SD3 disponen de dos interfaces WAN para establecer un doble túnel entre ellos y proporcionar redundancia. El nodo SD1 dispone de un interfaz WAN que establece un túnel con el NODO SD2 a través de su interfaz WAN1. Además, en cada nodo hay una maquina Ubuntu para verificar la comunicación entre todos ellos.

La sede corporativa se encuentra en el nodo SD2. La compañía dispone de dos datacenter, uno en la nube publica de Amazon AWS, el nodo SD1 y otro en un datacenter con infraestructura propia, el nodo SD3.

Además, la sede corporativa debe disponer de acceso a internet.

En el anexo I se muestra el detalle de los pasos a seguir para el despliegue del nodo en AWS.

A modo de resumen, es necesario disponer de un sistema Linux con Ansible instalado, acceso a Amazon AWS con credenciales IAM y el token de acceso al orquestador flexiManage. Del resto se encarga el script generado por flexiWan.

En caso de que el despliegue se realice sobre una plataforma de propósito general, los pasos de la instalación se detallan en el anexo II. Al igual que en el caso anterior y a modo de resumen, se deberá crear un máquina virtual con Ubuntu 18.04 en la que lanzar el script de instalación.

Tras la descarga e instalación del software, se debe ejecutar la herramienta de diagnóstico *fwsystem_checker* para verificar y solventar los posibles problemas que se hayan producido.

Con los sistemas instalados se procede a la configuración de los diferentes nodos. Estos pasos quedan detallados en el anexo III. A modo de resumen, una vez el equipo está instalado aparece en el panel de control del orquestador, donde es necesario ajustar su configuración de interfaces y rutas en el caso de multi-wan. Tras esto, se crean las etiquetas para los túneles y la del servicio de *Internet Breakout*, se asignan a los interfaces y se establecen los túneles.

A continuación, se muestra en detalle la topología desplegada mostrando los elementos de la arquitectura SD-WAN descritos en el apartado 1.1.2.

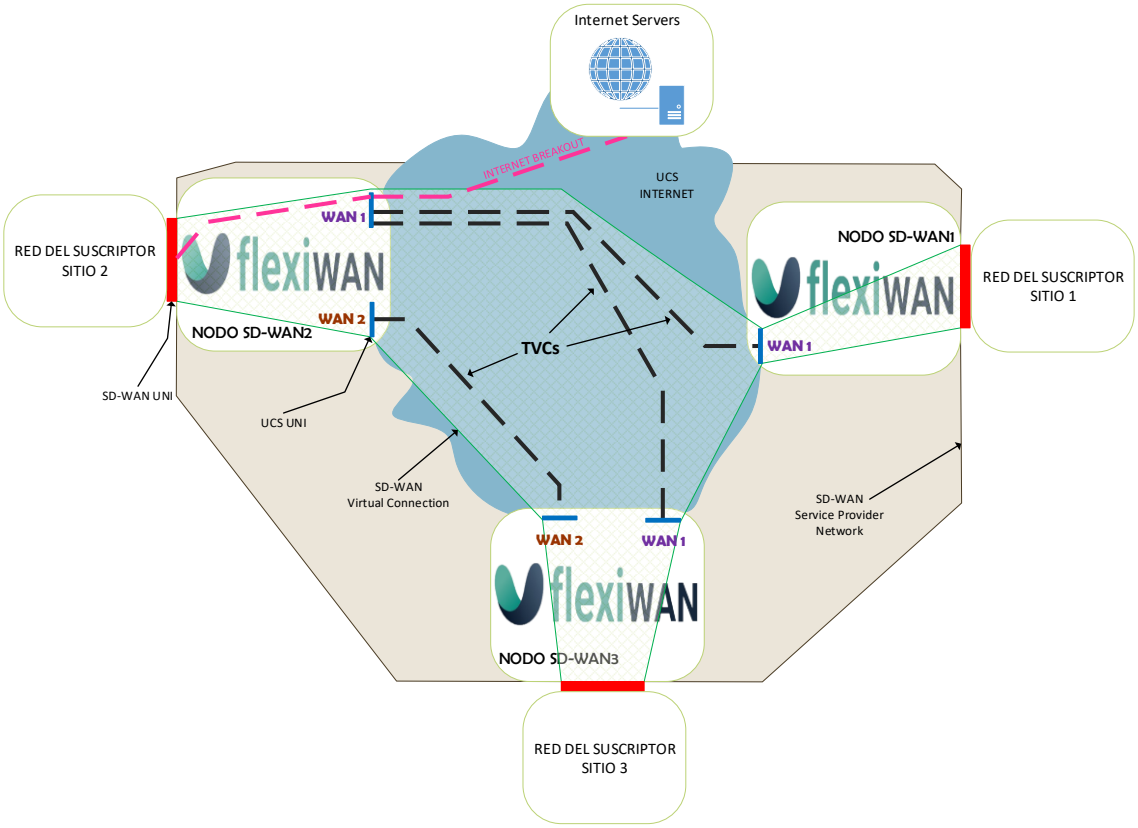


Figura 25 Topología de red SD-WAN MEF

Como se ve en la figura 25, solo hay un UCS, que es la red de internet, con dos conexiones por las cuales se establecen los TVCs en los nodos 2 y 3 y una conexión en el nodo 1. A modo de ejemplo, se puede encontrar en el anexo IV esta misma topología, pero usando los UCS que actualmente tiene en servicio el cliente, es decir, la red MPLS y las conexiones FTTH.

5. Pruebas.

En la siguiente figura se muestra la topología de red que se seguirá en las pruebas,

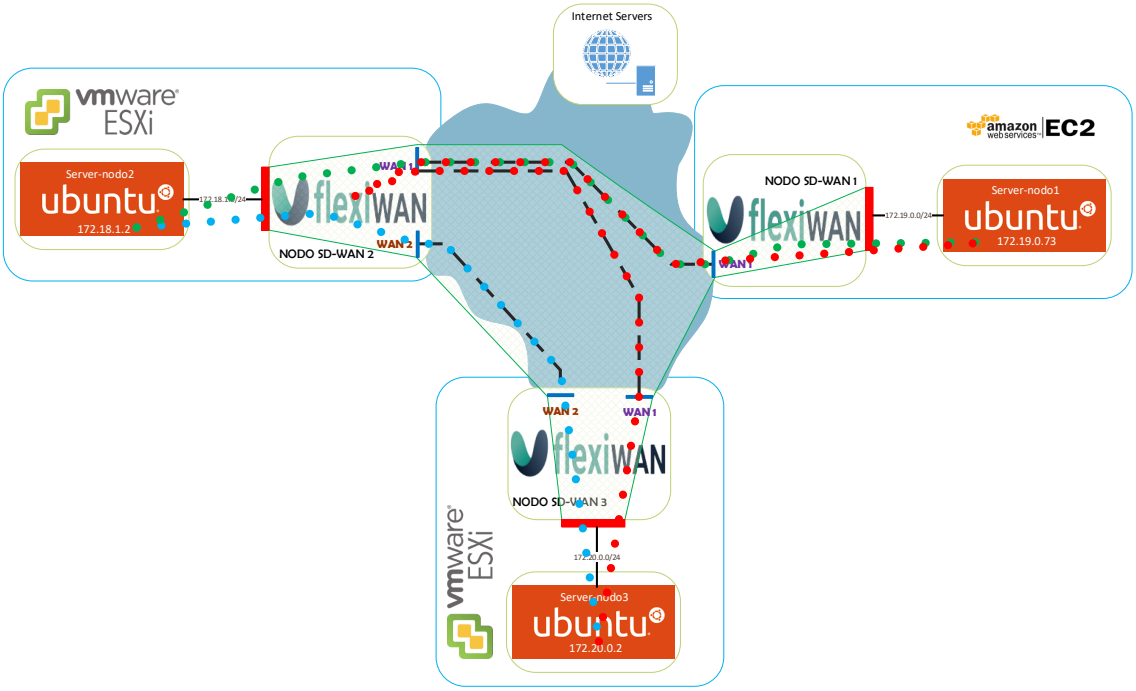


Figura 26 topología de red pruebas.

Se va a probar la comunicación extremo a extremo entre los diferentes servidores de la red, concretamente, entre *Server-nodo1* y *Server-nodo2* (línea punteada verde), entre *Server-nodo1* y *Server-nodo3* (línea punteada roja) y entre *Server-nodo2* y *Server-nodo3* (línea punteada azul).

Tras esto, la siguiente prueba consistirá en el rendimiento obtenido entre los servidores utilizando la herramienta iperf y las estadísticas que nos ofrece el orquestador.

Para finalizar, se probará a desconectar uno de los túneles entre el nodo sd2 y el nodo sd3 y verificar si sigue habiendo conectividad entre extremos.

5.1 Encapsulación VxLAN.

Antes de empezar las pruebas de comunicación, conviene revisar si los nodos están encapsulando correctamente el tráfico.

Tal como se ha visto en el apartado 3.2, se espera que la comunicación extremo a extremo se establezca a través de un túnel GRE encapsulado en VxLAN. Se va a comprobar que esto es así haciendo una captura de tráfico en uno de los nodos e interpretándola con wireshark.

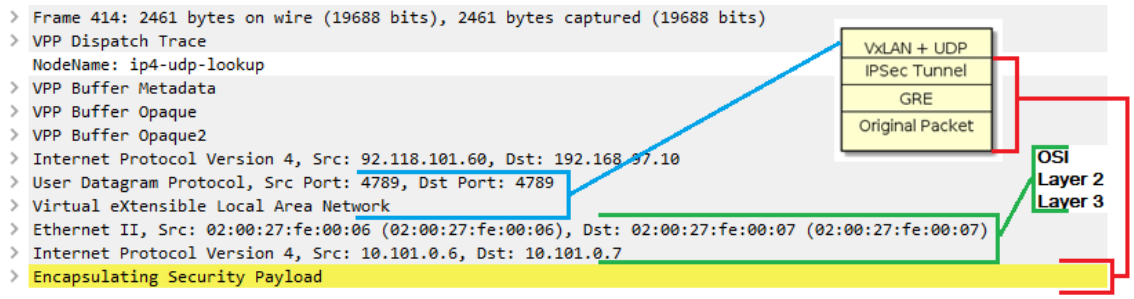


Figura 27 Captura de un paquete VxLAN

En la captura de la figura 27 se pueden relacionar las capas que se han visto en la documentación con la estructura real del paquete. Concretamente, se ve que tras la capa de red (3) del paquete original, hay un túnel VxLAN sobre UDP (marca azul). En el túnel VxLAN aparece encapsulado el tráfico ethernet entre los túneles, se ve la capa de enlace con las direcciones MAC de ambos extremos del túnel y la capa de red con las direcciones IP, también, de los extremos del túnel (marca verde). Por último, en wireshark se muestra un paquete encriptado en IPSEC, por el que va el túnel GRE y el paquete original (marca roja).

5.2 Comunicaciones extremo a extremo.

En primer lugar, se realizan pruebas de ping entre los diferentes nodos. El comando ping usa el protocolo ICMP que envía un *echo request* al host remoto y este responde con un *echo reply*. Debido a la naturaleza de esta prueba no es necesario ejecutar ping desde los dos extremos de la comunicación ya que en caso de que haya respuesta desde un extremo, indica que hay comunicación en ambas direcciones. A este respecto, se puede dar el caso de que en un extremo haya instalado un firewall que filtre los paquetes *icmp echo-request*, en este caso el resultado del ping seria que hay pérdida, aunque puede que haya comunicación.

Comunicación entre Server-nodo3 y Server-nodo2

Se ejecuta el comando *ping* desde *Server-nodo3* a *Server-nodo2.* Se van a lanzar 100 paquetes con un intervalo de 0,1 s a través del comando `ping 172.18.1.2 -c 100 -i 0.1`. Además, para una muestra más general, se lanza tres veces el mismo comando y se muestran los resultados en la figura 28.

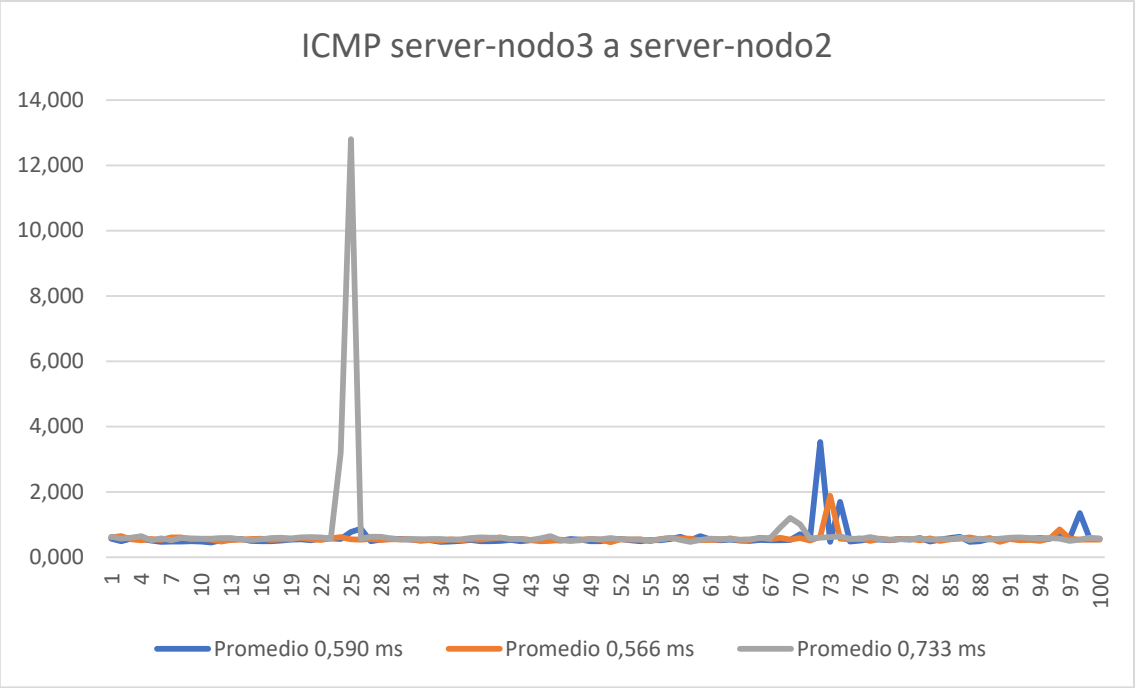


Figura 28 Ping de Server-nodo3 a Server-nodo2

El resultado de la ejecución de ping indica que hay comunicación extremo a extremo con una latencia media de 0,58, 0,566 y 0,733 ms para cada serie y 0,63 ms en total, como ese es el tiempo de ida y vuelta de los paquetes ICMP (RTT), la latencia del circuito será la mitad, 0,315 ms. Además, no se han producido perdidas.

A continuación, se comprueba el camino seguido por los paquetes usando el comando *tracepath*, en este caso desde los dos extremos, ya que, al existir dos posibilidades, el camino seguido desde un extremo no tiene por qué ser igual al camino seguido desde el otro.

```
david@server-nodo3:~$ tracepath -n 172.18.1.2
 1?: [LOCALHOST]                      pmtu 1350
 1:  172.20.0.1                                            0.134ms asymm  2
 1:  172.20.0.1                                            0.125ms asymm  2
 2:  10.100.0.7                                            0.450ms asymm  3
 3:  172.18.1.2                                            0.501ms reached
     Resume: pmtu 1350 hops 3 back 3
```

Figura 29 Resultado tracepath de Server-nodo3 a Server-nodo2

El host *Server-nodo3* tiene la IP 172.20.0.2 y como puerta de enlace la IP 172.20.0.1.

Los paquetes salen del host *Server-nodo3*, etiquetado como LOCALHOST, y atraviesan la puerta de enlace, 172.20.0.1. Entran al nodo SD2 a través del túnel, 10.100.0.7 y alcanzan el host remoto, *Server-nodo2*, 172.18.1.2.

El mismo tráfico desde el otro extremo,

```
david@server_nodo2:~$ tracepath -n 172.20.0.2
 1?: [LOCALHOST]                      pmtu 1350
 1:  172.18.1.1                                            0.182ms asymm  2
 1:  172.18.1.1                                            0.251ms asymm  2
 2:  10.100.0.6                                            0.597ms asymm  3
 3:  172.20.0.2                                            0.580ms reached
     Resume: pmtu 1350 hops 3 back 3
```

Figura 30 Resultado tracepath de Server-nodo2 a Server-nodo3

Con resultado similar al anterior, pero en sentido contrario.

El tráfico sigue la ruta marcada de azul en la topología de red.

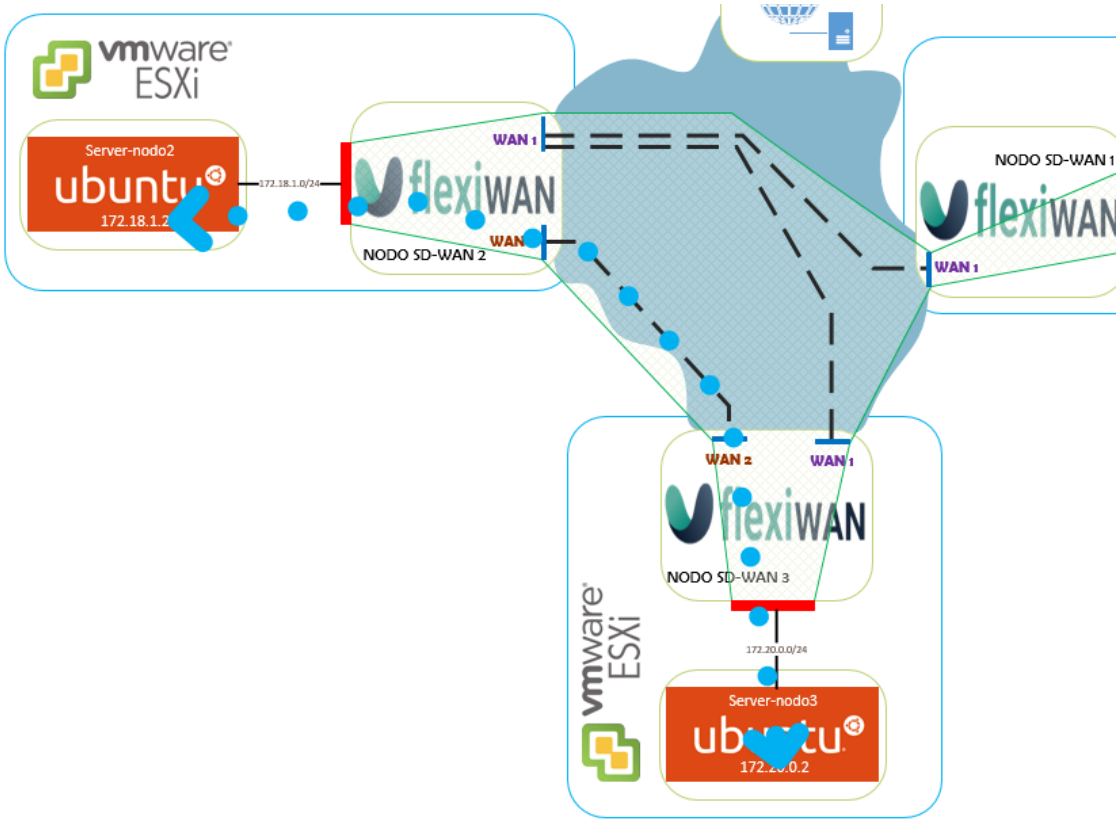


Figura 31 Resultado esquemático del camino seguido entre Server-nodo2 y Server-nodo3

Comunicación entre Server-nodo1 y Server-nodo2.

Al igual que en el caso anterior, primero se ejecuta el comando *ping* tres veces para extraer los datos para la gráfica.

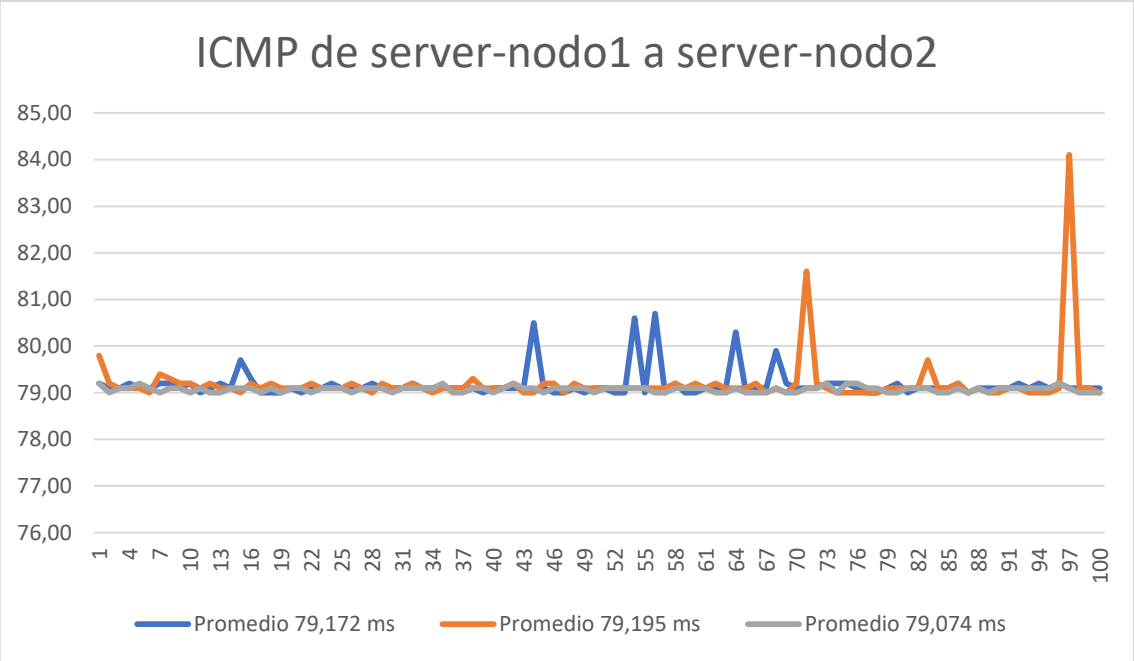


Figura 32 Ping de Server-nodo1 a Server-nodo2

Se comprueba que hay conectividad, sin perdidas, y que la latencia es más alta que en el caso anterior, con 79,147 ms como promedio de las tres series. Esto es normal ya que el servidor está ubicado en un centro de datos de Amazon AWS en Estados Unidos, por lo que la latencia aumenta al ser una conexión transatlántica. La latencia del circuito será la mitad, 39,57 ms.

A continuación, se muestra el camino que sigue el tráfico para la comunicación extremo a extremo desde ambos servidores.

```
ubuntu@server-nodo1:~$ tracepath -n 172.18.1.2
 1?: [LOCALHOST]                      pmtu 1500
 1:  172.19.1.100                                          0.490ms pmtu 1350
 1:  172.19.1.100                                          0.429ms asymm  2
 2:  10.100.0.9                                           80.783ms asymm  3
 3:  172.18.1.2                                           80.013ms reached
     Resume: pmtu 1350 hops 3 back 3
david@server_nodo2:~$ tracepath -n 172.19.1.73
 1?: [LOCALHOST]                      pmtu 1500
 1:  172.18.1.1                                            0.155ms pmtu 1350
 1:  172.18.1.1                                            0.159ms asymm  2
 2:  10.100.0.8                                           78.996ms asymm  3
 3:  172.19.1.73                                          79.464ms reached
     Resume: pmtu 1350 hops 3 back 3
```

Figura 33 Resultado tracepath entre server-nodo1 y server-nodo2

Donde el camino seguido es la ruta marcada en verde sobre la topología de red,

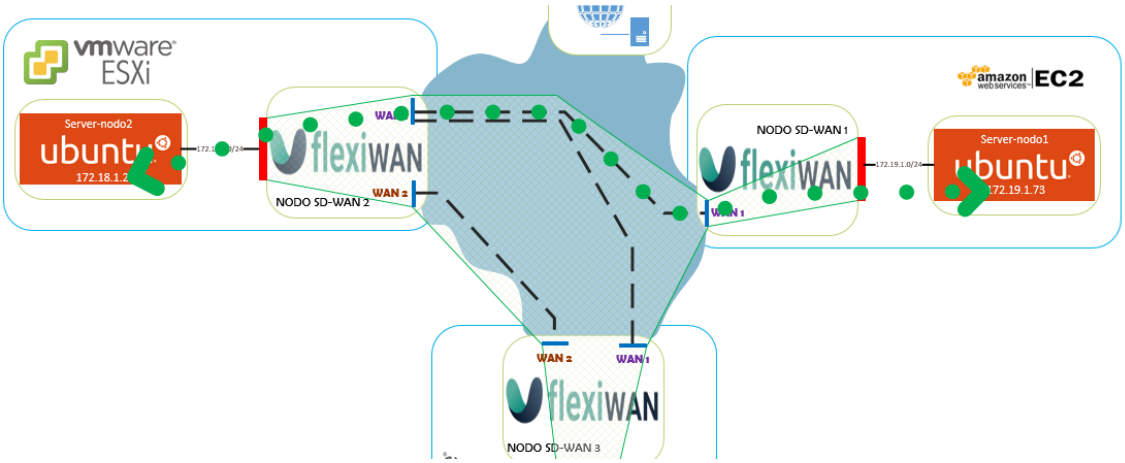


Figura 34 Resultado esquemático del camino seguido entre Server-nodo1 y Server-nodo2

Comunicación entre Server-nodo1 y Server-nodo3.

La comunicación entre los dos CPD no está habilitada de forma directa, pero sí que es posible atravesando el nodo SD2, a continuación, se muestran las pruebas de ping y traza como en los casos anteriores,

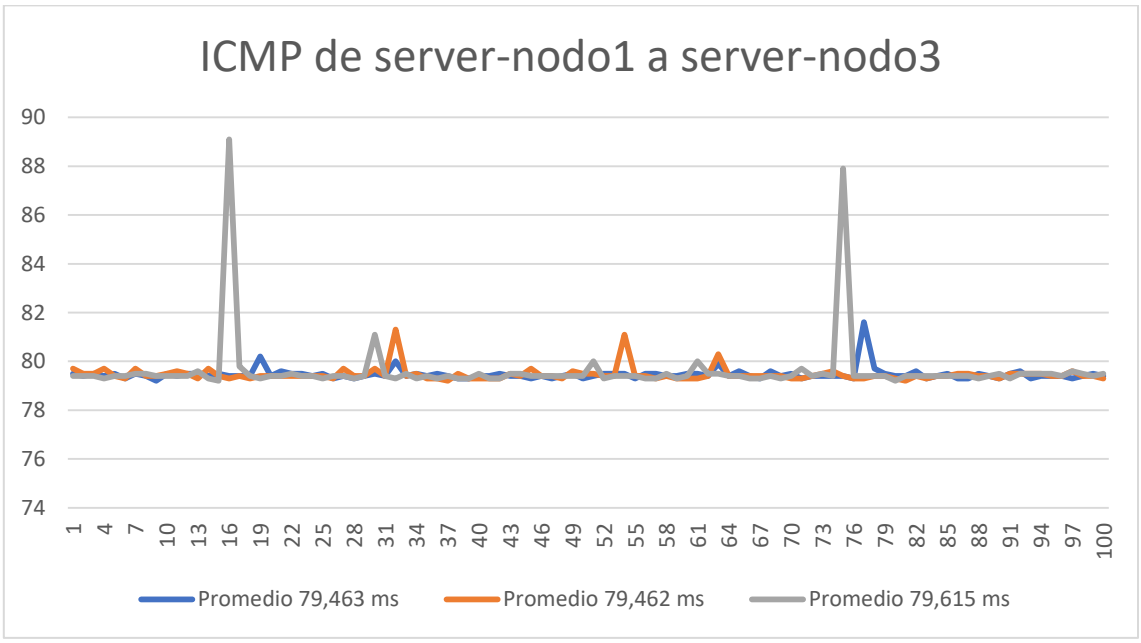


Figura 35 Ping de Server-nodo1 a Server-nodo3

La latencia media es de 79,51 ms, muy similar al caso anterior, aunque se aprecia que es unos 0,4 ms superior, debido al paso a través del nodo sd2. La latencia del circuito se establece en 39,75 ms.

Respecto al camino seguido por el tráfico, se muestran las trazas a continuación,

```
ubuntu@server-nodo1:~$ tracepath -n 172.20.0.2
 1?: [LOCALHOST]                      pmtu 1500
 1:  172.19.1.100                                          0.409ms pmtu 1350
 1:  172.19.1.100                                          0.332ms asymm  2
 2:  10.100.0.9                                           79.246ms asymm  3
 3:  10.100.0.4                                           79.600ms asymm  4
 4:  172.20.0.2                                           79.651ms reached
     Resume: pmtu 1350 hops 4 back 4
david@server-nodo3:~$ tracepath -n 172.19.1.73
 1?: [LOCALHOST]                      pmtu 1350
 1:  172.20.0.1                                            0.168ms asymm  2
 1:  172.20.0.1                                            0.187ms asymm  2
 2:  10.100.0.5                                            0.458ms asymm  3
 3:  10.100.0.8                                           79.083ms asymm  4
 4:  172.19.1.73                                          79.677ms reached
     Resume: pmtu 1350 hops 4 back 4
```

Figura 36 Resultado tracepath entre server-nodo1 y server-nodo3

Donde el camino seguido es la ruta marcada en rojo sobre la topología de red,

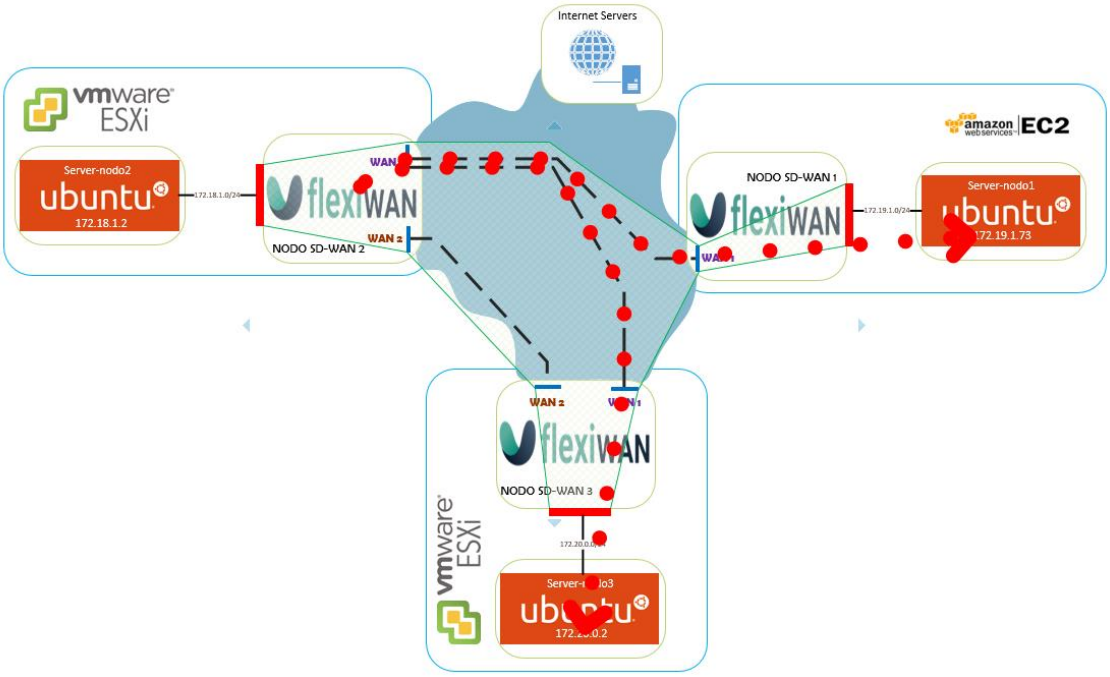


Figura 37 Resultado esquemático del camino seguido entre Server-nodo1 y Server-nodo3

A modo de resumen, se muestra una gráfica con las latencias entre los diferentes nodos,

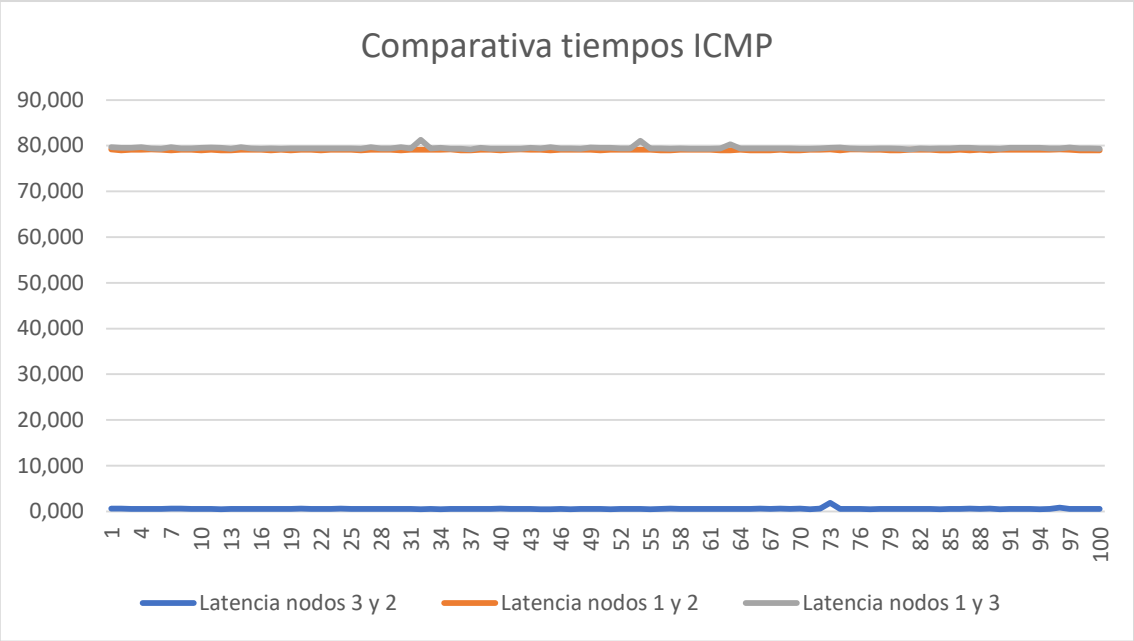


Figura 38 Latencias entre nodos

Como se ha comentado anteriormente, estas latencias corresponden a la suma del tiempo de ida y el tiempo de vuelta, que es el valor que proporciona ping. En la gráfica se aprecia de forma visual que hay una comunicación estable, que los nodos 2 y 3 están muy próximos entre si ya que tienen muy baja latencia y que el nodo 1 está más alejado. Efectivamente, los nodos dos y tres están en la misma ubicación mientras que el nodo uno está en Estados Unidos.

Otra forma de comprobar estos aspectos es a través del portal de flexiManage, donde se muestran estadísticas de latencia y perdidas en los túneles. En la figura 39 se muestran estos datos y se comprueba que son similares a los encontrados a través del comando ping.

Path Label	AVG Latency	Drop Rate
WAN1.SD2	0.69 ms	0.00 %
WAN2.SD2	0.84 ms	0.00 %
WAN1.SD2	75.88 ms	0.00 %

Figura 39 Datos de latencia y pérdidas mostrados en flexiManage

Pruebas de acceso web.

Además de las pruebas de ping, se ha desplegado un servidor web en *Server-nodo1* para realizar pruebas de conectividad.

Accediendo desde *Server-nodo2*,

NODO SD1

Servidor WEB de David Suarez

Figura 40 Acceso web a Server-nodo1

Y desde la consola del host *Server-nodo3*

```
david@serverVMWare:~$ curl http://172.19.1.73
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;">NODO SD1</h1>

<p>Servidor WEB de David Suarez</p>

</body>
</html>
david@serverVMWare:~$
```

Figura 41 Acceso web vía consola a Server-nodo1

5.3 Pruebas de rendimiento.

Para las pruebas de rendimiento se va a utilizar la herramienta iperf, usada para estos propósitos en entornos de producción.

Se busca ver si el ancho de banda se ve mermado por tener que atravesar los túneles creados por los nodos SD-WAN. Para ello, se instalan tres máquinas de *test* independientes a la topología SD-WAN, una en la infraestructura VMWare donde reside el nodo 3, otra junto al nodo 2, y otra en Amazon AWS donde reside el nodo 1. Se hará una prueba de rendimiento entre estas máquinas de *test* en paralelo a las pruebas de rendimiento echas a través de los túneles.

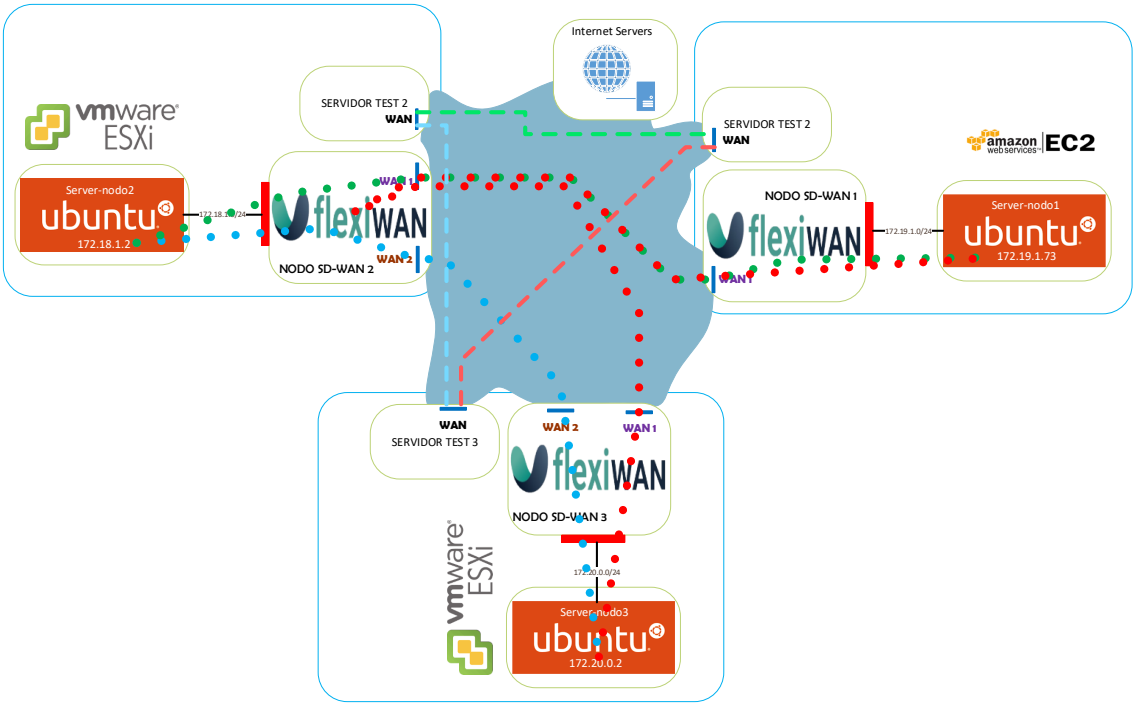


Figura 42 Esquema de las pruebas a realizar

En la figura 42 se muestra el resumen de estas pruebas, se probarán en paralelo los caminos del mismo tono de color (azul, rojo, verde). El camino en guiones representa la prueba a través de internet entre las tres máquinas de *test* y el de línea punteada entre los servidores corporativos a través de los túneles.

El comando iperf permite medir tanto en TCP como en UDP, en UDP aporta datos acerca del *jitter* y de las pérdidas de datagramas por lo que se usará este protocolo para medir estos parámetros. Para el ancho de banda real es más adecuado utilizar TCP, por lo que será el protocolo usado para este propósito.

En el lado servidor se usa el comando,

`iperf3 -s` tanto para UDP como para TCP, donde -s indica que es el servidor.

Para el lado cliente se usa el comando,

`iperf3 -c IP_REMOTA -u -b 100m` para UDP e `iperf3 -c IP_REMOTA` para TCP

donde -c indica que se conecta a IP_REMOTA, -u que se utiliza UDP y -b que se establece un ancho de banda de 100 Mbps (solo usado en UDP ya que no hay cálculo de ventana TCP).

Similar a como se ha hecho para el caso de las pruebas de ping, para el ancho de banda se realizan tres pruebas TCP con iperf y se comparan gráficamente junto a la media de las tres.

Pruebas entre el nodo SD1 y el nodo SD2

En primer lugar, se establece el lado servidor en el nodo 2 y el lado cliente en el nodo 1. Primero se ejecutan las pruebas en TCP y acto seguido en UDP.

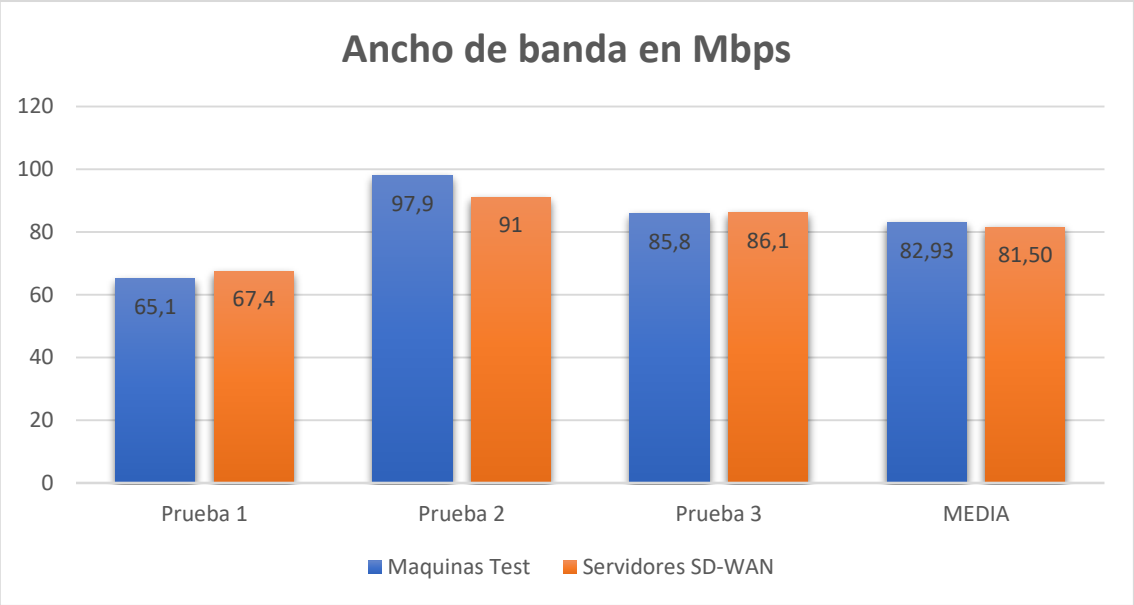


Figura 43 Pruebas TCP entre los nodos 1 y 2

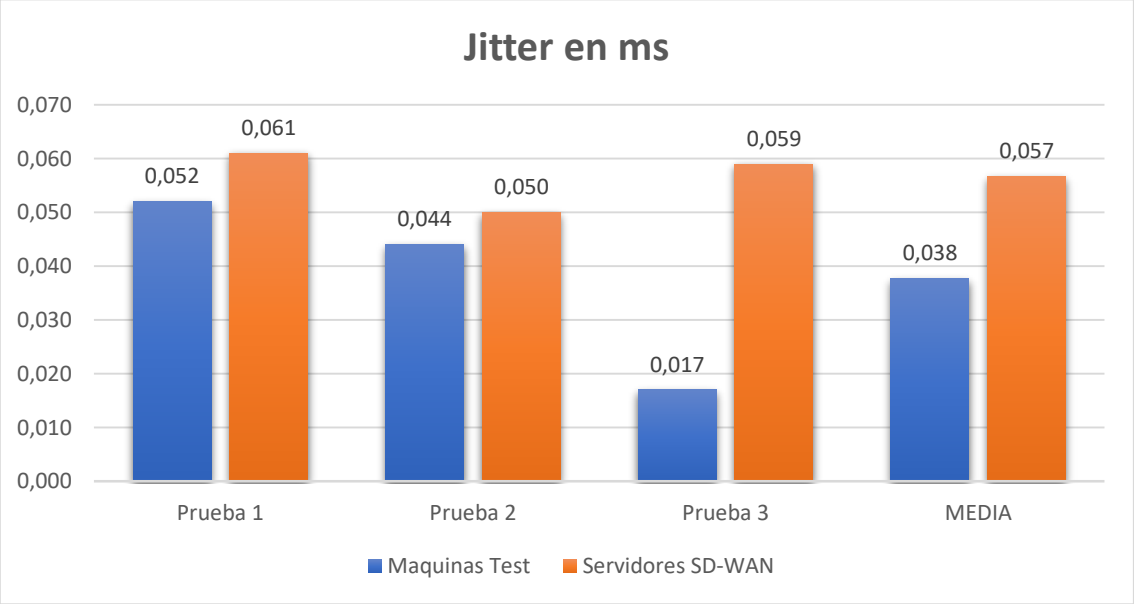


Figura 44 Pruebas UDP entre los nodos 1 y 2

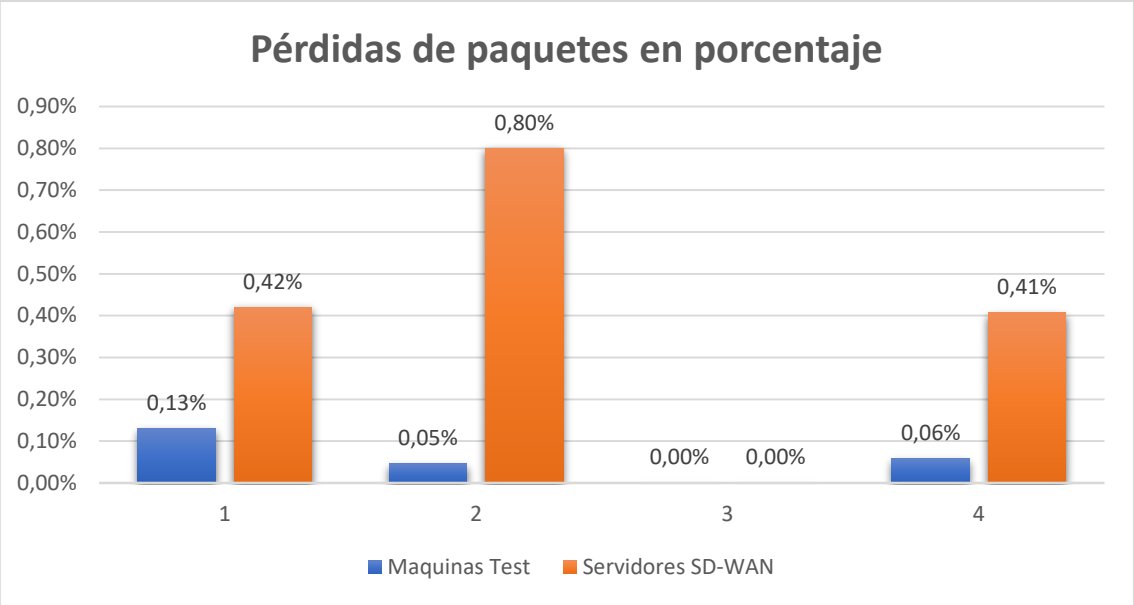


Figura 45 Pruebas UDP entre los nodos 1 y 2

En la figura 43 se aprecia que el ancho de banda es muy similar entre las pruebas realizadas a través de internet y entre las que usan la infraestructura SD-WAN, resultando en una media de 81,5 Mbps. Respecto a las pruebas de jitter y pérdidas, los valores medios son muy similares.

Pruebas entre el nodo SD3 y el nodo SD2

Para esta pruebas, se establece el lado servidor en el nodo 2 y el lado cliente en el nodo 3. Como en el caso anterior, en primer lugar, se muestran los datos de ancho de banda obtenidos a través de la pruebas TCP y a continuación los de jitter y pérdidas obtenidos a través de UDP.

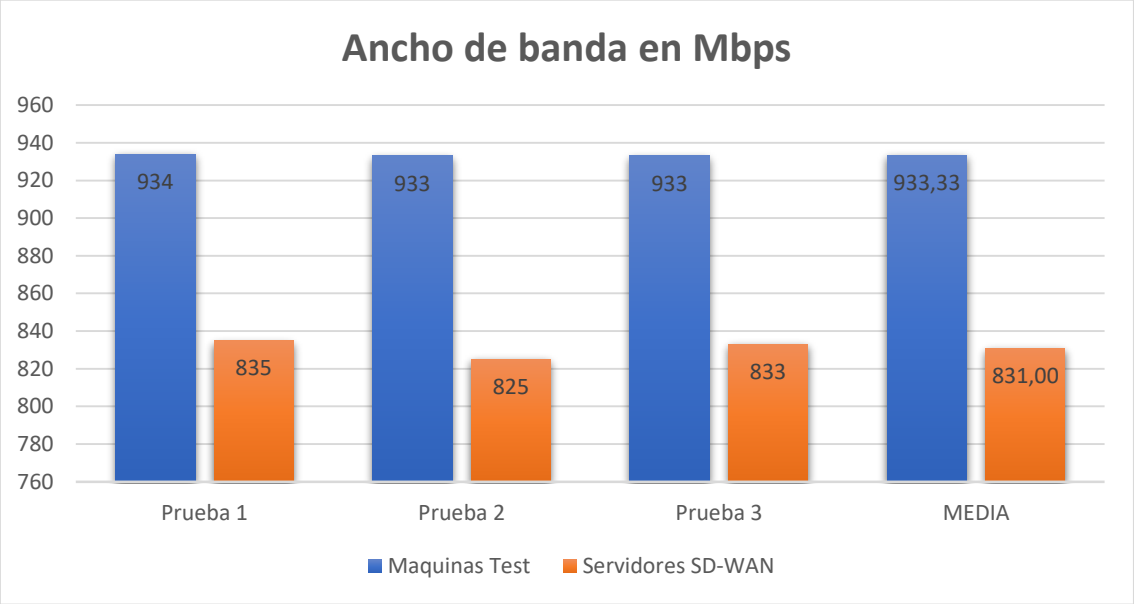


Figura 46 Pruebas TCP entre los nodos 2 y 3

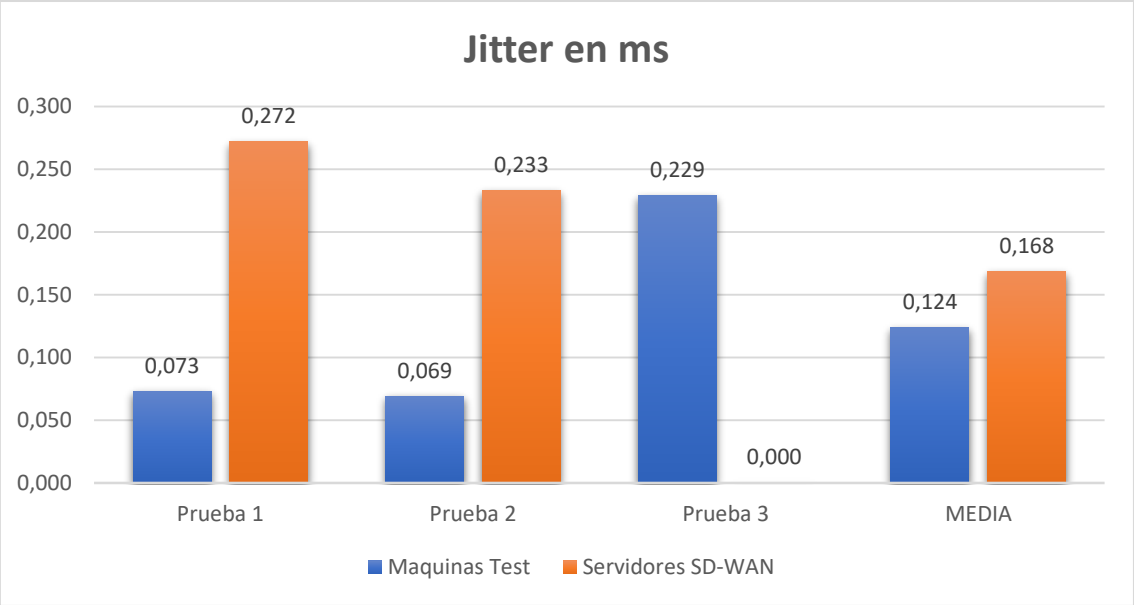


Figura 47 Pruebas UDP entre los nodos 2 y 3

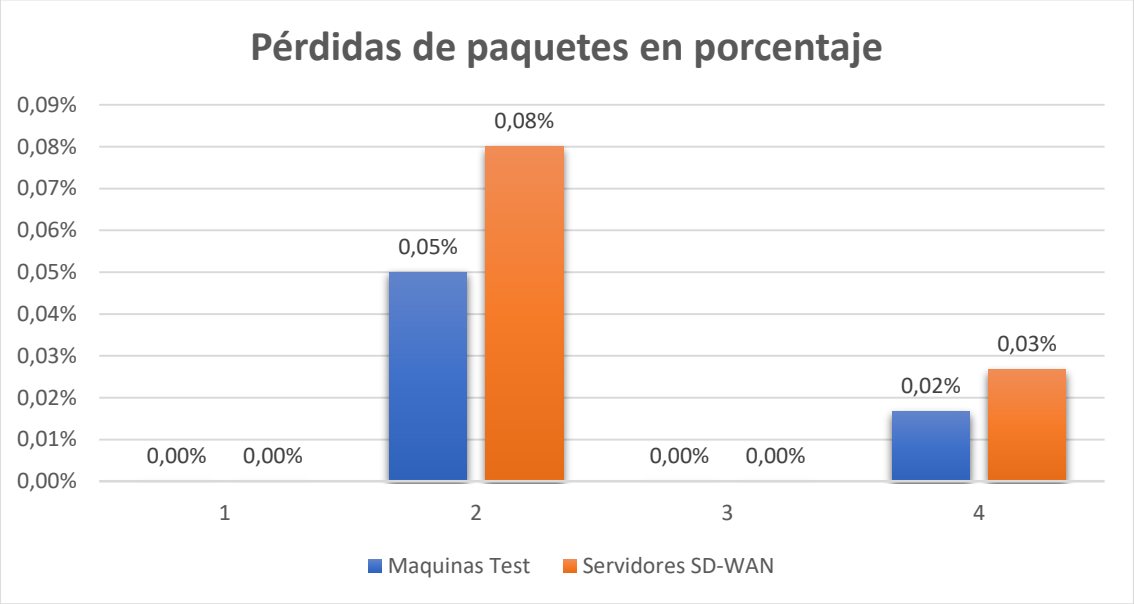


Figura 48 Pruebas UDP entre los nodos 2 y 3

Pruebas entre el nodo SD1 y el nodo SD3

Para finalizar, se ejecuta una prueba entre los dos nodos remotos, cuyo tráfico fluye a través del nodo SD2 para los datos tunelizados. La conexión entre las máquinas de test es directa al realizarse a través de Internet.

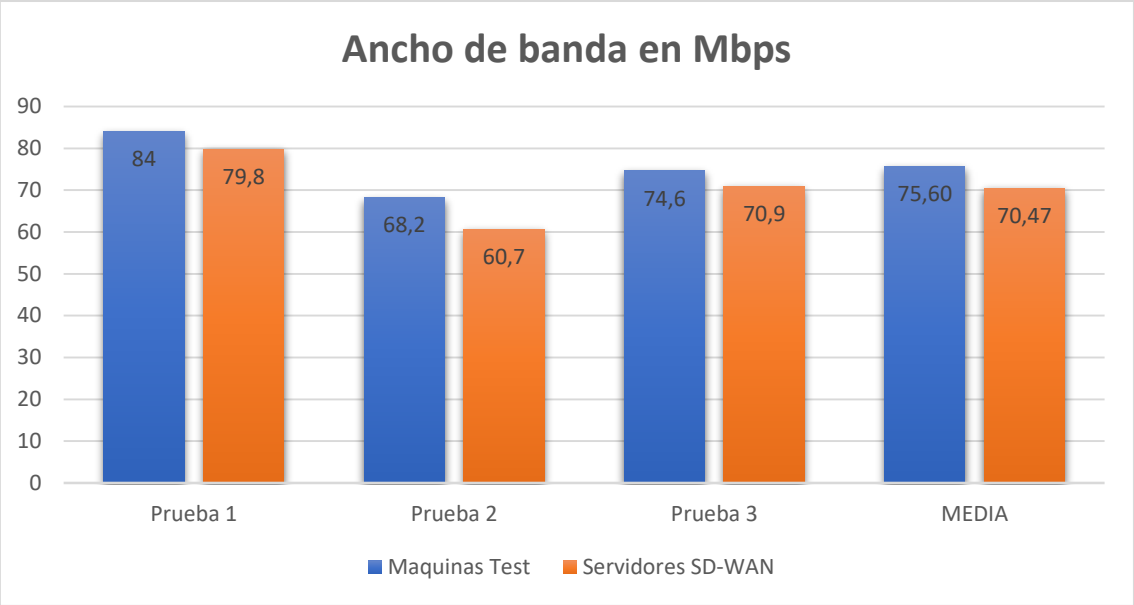


Figura 49 Pruebas TCP entre los nodos 1 y 3

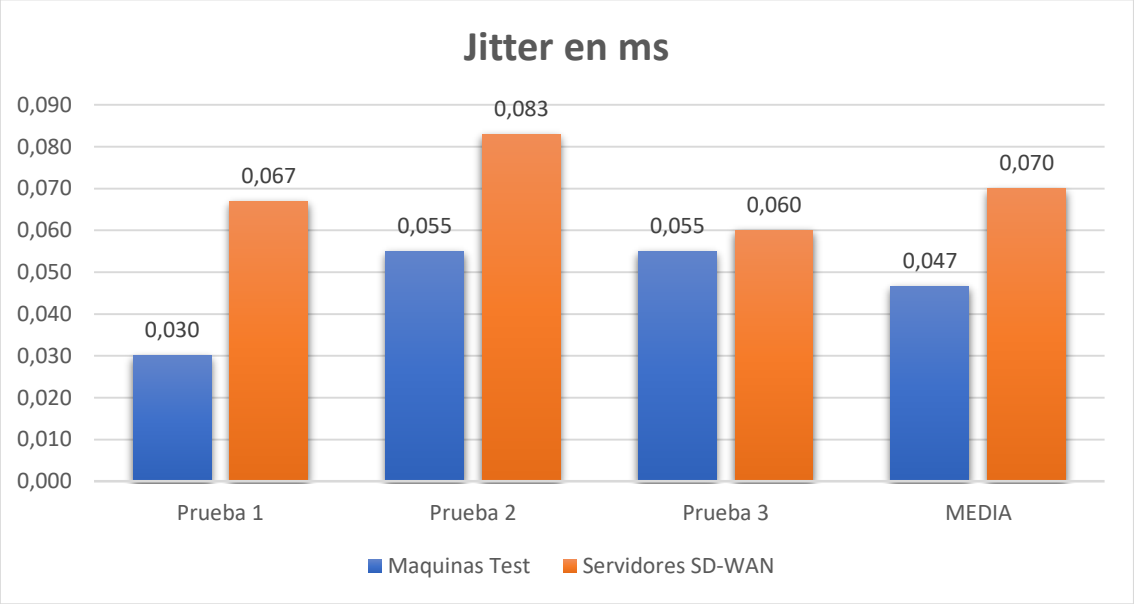


Figura 50 Pruebas UDP entre los nodos 1 y 3

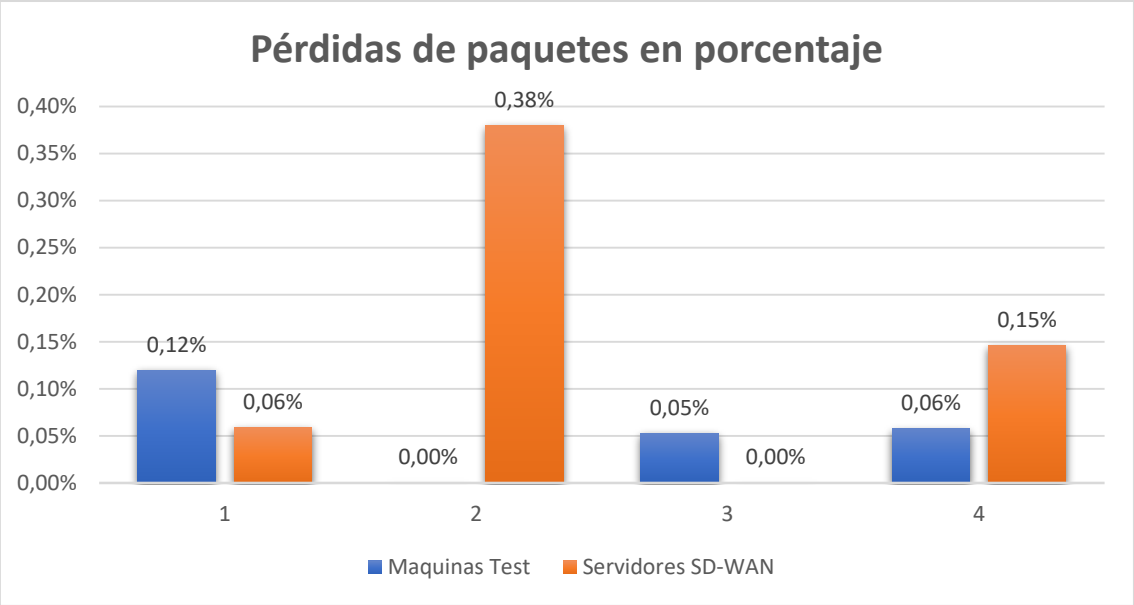


Figura 51 Pruebas UDP entre los nodos 1 y 3

Resumen de las pruebas de rendimiento

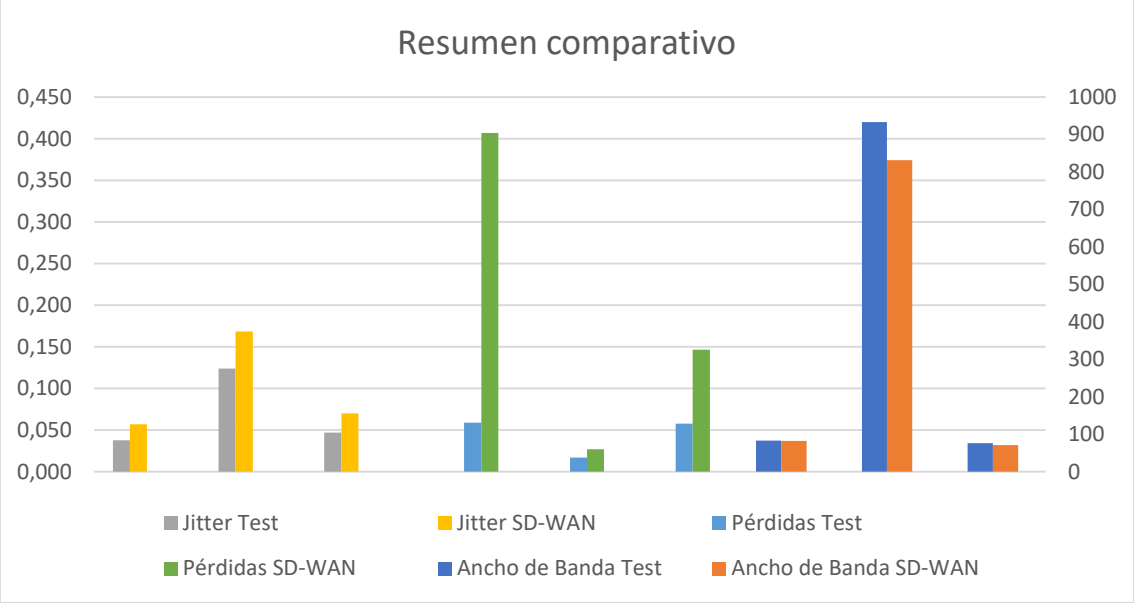


Figura 52 Resumen gráfico de las pruebas de rendimiento.

La conexión transatlántica entre el nodo uno y los nodos dos y tres, ronda los 100 Mbps y no se aprecia que al atravesar el túnel se reduzca el ancho de banda, excepto al hacerlo entre los nodos uno y tres, que el tráfico fluye a través del nodo dos. En la conexión entre los nodos dos y tres, se observa que el ancho de banda ronda el gigabit por segundo, pero en la prueba a través del túnel este ancho de banda se reduce.

Estos datos indican que los túneles afectan al ancho de banda, aunque esto es solo apreciable para tasas de transferencia altas. Esto se debe a que el tráfico a través del túnel tiene una MTU de 1350 bytes, mientras que el tráfico entre las máquinas de test, que van a través de internet, tiene una MTU de 1500 bytes, diferencia existente ya que el tráfico tunelizado requiere cabeceras TCP/IP extra (VxLAN, GRE, IPSEC). Esta diferencia influye en la cantidad de paquetes a transmitir, ya que el total de los datos debe ser dividido en paquetes de un máximo de tamaño

igual a la MTU menos las cabeceras, haciendo que a menor MTU sea necesario enviar más paquetes para el mismo volumen de datos y, por tanto, necesitar más tiempo para completar el envío. [24]

Respecto al jitter y a las perdidas, se observa que los valores a través del túnel son ligeramente superiores, aunque despreciables, ya que son valores muy bajos para ambos casos. El aumento de estos valores cuando se mide a través del túnel se debe a que los nodos SD-WAN requieren de más operaciones de proceso por paquete, al tener que encriptarlos y encapsularlos. Como se ha comentado, este aumento no es significativo para una operativa normal por lo que se considera que la conexión es de buena calidad.

5.4 Pruebas de redundancia.

Uno de los elementos centrales de flexiEdge es el router virtual FD.io, que tiene un interface de configuración similar al IOS de Cisco. A través de este interface se van a comprobar las rutas y estado de OSPF para esta prueba de redundancia.

Como se ha visto en la topología utilizada para las pruebas, solo hay redundancia entre los nodos SD2 y SD3, por lo que las pruebas irán centradas en ellos.

Para acceder al interface de FD.io se usa el comando,

```
sudo vtysh
```

En el nodo-sd3 aparecen las sesiones OSPF establecidas a través de los dos túneles.

```
nodo-sd3# sh ip ospf neighbor

Neighbor ID     Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
172.18.1.1        1 Full/DROther      38.774s 10.100.0.8      vpp3:10.100.0.9          0     0     0
172.18.1.1        1 Full/DROther      38.774s 10.100.0.6      vpp5:10.100.0.7          0     0     0
```

Figura 53 Sesiones OSPF en el nodo-sd3

Hay dos sesiones, una a través del túnel de la WAN1 (10.100.0.7) y otra a través del túnel de la WAN2 (10.100.0.9).

En el nodo-sd2, además de las dos sesiones con el nodo-sd3, aparece la sesión del nodo-sd1,

```
nodo-sd2# sh ip ospf neighbor

Neighbor ID     Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
172.19.1.100      1 Full/DROther      34.219s 10.100.0.5      vpp3:10.100.0.4          0     0     0
172.20.0.1        1 Full/DROther      34.743s 10.100.0.7      vpp5:10.100.0.6          0     0     0
172.20.0.1        1 Full/DROther      34.743s 10.100.0.9      vpp7:10.100.0.8          0     0     0
```

Figura 54 Sesiones OSPF en el nodo-sd2

La sesión con el nodo-sd1 se establece a través de la IP 10.100.0.4, la sesión a través de la WAN1 con la IP 10.100.0.6 y a través de la WAN2 con la IP 10.100.0.8.

El túnel que va por la WAN2 se puede identificar a través del orquestador,

ID	Device A	Interface A	Device B	Interface B	Path Label	AVG Latency	Drop Rate	Status	Actions
1	NODO SD2 VBOX	enp0s3 (loopback: 10.100.0.4)	NODO SD1 AWS	ens5 (loopback: 10.100.0.5)	WAN1.SD2	94.59 ms	0.00 %	Connected	
2	NODO SD2 VBOX	enp0s3 (loopback: 10.100.0.6)	NODO SD3 VMWARE	ens160 (loopback: 10.100.0.7)	WAN1.SD2	7.17 ms	0.00 %	Connected	
3	NODO SD2 VBOX	enp0s8 (loopback: 10.100.0.8)	NODO SD3 VMWARE	ens192 (loopback: 10.100.0.9)	WAN2.SD2	6.89 ms	0.00 %	Connected	

Figura 55 Túneles creados en flexiManage

El túnel que va por la WAN2 es el que tiene la etiqueta WAN2.SD2 (en rojo en la imagen) que como se muestra, está establecido entre las IPs 10.100.0.8 y 10.100.0.9, por los interfaces vpp7 en nodo-sd2 y vpp3 en nodo-sd3.

Con los túneles ya identificados, se muestran las rutas en el nodo-sd2. Estas indican que el tráfico a la red del nodo SD3 se encamina a través del túnel WAN2.SD2 (10.100.0.9),

```
nodo-sd2# sh ip ro
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

K>* 0.0.0.0/0 [0/0] via 192.168.149.2, vpp0, 00:35:55
C>* 10.0.2.0/24 is directly connected, vpp1, 00:35:55
O   10.100.0.4/31 [110/10000] is directly connected, vpp3, 00:35:55
C>* 10.100.0.4/31 is directly connected, vpp3, 00:35:55
O   10.100.0.6/31 [110/10000] is directly connected, vpp5, 00:35:55
C>* 10.100.0.6/31 is directly connected, vpp5, 00:35:55
O   10.100.0.8/31 [110/1000] is directly connected, vpp7, 00:35:55
C>* 10.100.0.8/31 is directly connected, vpp7, 00:35:55
O   172.18.1.0/24 [110/10000] is directly connected, vpp2, 00:35:55
C>* 172.18.1.0/24 is directly connected, vpp2, 00:35:55
O>* 172.19.1.0/24 [110/20000] via 10.100.0.5, vpp3, 00:35:45
O>* 172.20.0.0/24 [110/11000] via 10.100.0.9, vpp7, 00:25:15
C>* 192.168.149.0/24 is directly connected, vpp0, 00:35:55
nodo-sd2#
```

Figura 56 Tabla de rutas en el nodo-sd2

Concretamente la línea,

```
O>* 172.20.0.0/24 [110/11000] via 10.100.0.9, vpp7, 00:25:15
```

Indica que se ha insertado la ruta a la red del nodo-sd3 desde OSPF (O) con *nexthop* la IP 10.100.0.9 e interface vpp7.

Las rutas en el nodo-sd3 indican que el tráfico se encamina a través del túnel WAN2.SD2,

```
nodo-sd3# sh ip ro
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

K>* 0.0.0.0/0 [0/0] via 92.118.100.1, vpp2, 00:34:29
O>* 10.100.0.4/31 [110/11000] via 10.100.0.8, vpp3, 00:13:25
O   10.100.0.6/31 [110/10000] is directly connected, vpp5, 00:34:29
C>* 10.100.0.6/31 is directly connected, vpp5, 00:34:29
O   10.100.0.8/31 [110/1000] is directly connected, vpp3, 00:13:25
C>* 10.100.0.8/31 is directly connected, vpp3, 00:34:29
C>* 92.   .100.0/24 is directly connected, vpp2, 00:34:29
K>* 92.   .102.  /32 [0/1] via 192.168.97.8, vpp0, 00:34:29
O>* 172.18.1.0/24 [110/11000] via 10.100.0.8, vpp3, 00:13:25
O>* 172.19.1.0/24 [110/21000] via 10.100.0.8, vpp3, 00:13:25
O   172.20.0.0/24 [110/10000] is directly connected, vpp1, 00:34:29
C>* 172.20.0.0/24 is directly connected, vpp1, 00:34:29
C>* 192.168.97.0/24 is directly connected, vpp0, 00:34:29
nodo-sd3#
```

Figura 57 Tabla de rutas en el nodo-sd3

Concretamente la línea,

```
O>* 172.18.1.0/24 [110/11000] via 10.100.0.8, vpp3, 00:13:25
```

Por último, se realiza una traza para ver el camino que sigue el tráfico y confirmar que las rutas que se han mostrado son las activas,

```
david@serverVMWare:~$ tracepath -n 172.18.1.2
 1?: [LOCALHOST]                      pmtu 1500
 1:  172.20.0.1                                            0.085ms pmtu 1350
 1:  172.20.0.1                                            0.037ms asymm  2
 2:  10.100.0.8                                            6.942ms asymm  3
 3:  172.18.1.2                                            6.908ms reached
     Resume: pmtu 1350 hops 3 back 3
david@serverVMWare:~$
```

Figura 58 Traza desde el nodo-sd3 al nodo-sd2

En la imagen se ve que la traza atraviesa el interfaz con IP 10.100.0.8, que corresponde al túnel a través de la WAN2.

Ahora se quiere simular la caída del enlace WAN2, para ello, se establece una ruta de *blackhole* para la IP de la WAN2 del nodo-sd3 en el router intermedio de sd2, que está simulando el acceso a internet de este nodo.

Transcurrido el tiempo de convergencia de OSPF, la sesión a través de la WAN2 cae,

En el nodo-sd2,

```
nodo-sd2# sh ip ospf neighbor

Neighbor ID     Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
172.19.1.100      1 Full/DROther      36.159s 10.100.0.5      vpp3:10.100.0.4          0     0     0
172.20.0.1        1 Full/DROther      36.678s 10.100.0.7      vpp5:10.100.0.6          0     0     0
172.20.0.1        1 Init/DROther      36.678s 10.100.0.9      vpp7:10.100.0.8          0     0     0
```

Figura 59 Sesiones OSPF en el nodo-sd2

La sesión a través del interface con IP 10.100.0.8 está en estado *Init*, lo que indica que no está establecida.

En el nodo-sd3,

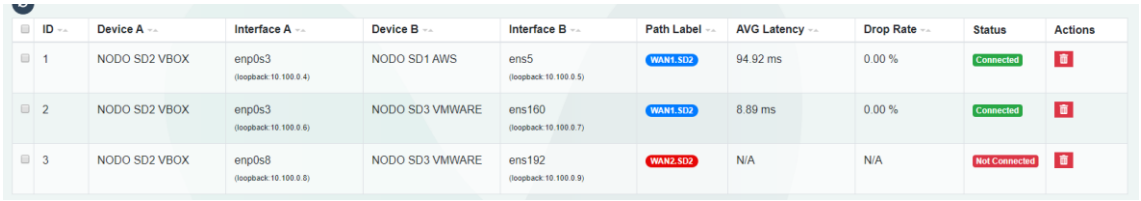
```
nodo-sd3# sh ip ospf neighbor

Neighbor ID     Pri State           Dead Time Address         Interface            RXmtL RqstL DBsmL
172.18.1.1        1 Full/DROther      39.098s 10.100.0.6      vpp5:10.100.0.7          0     0     0
```

Figura 60 Sesiones OSPF en el nodo-sd3

Solo aparece la sesión a través de la interfaz WAN1, por lo tanto, no hay sesión establecida a través del túnel de la WAN2.

En el portal de flexiManage se puede ver que el túnel que va a través de la WAN2 no está conectado,



ID	Device A	Interface A	Device B	Interface B	Path Label	AVG Latency	Drop Rate	Status	Actions
1	NODO SD2 VBOX	enp0s3 (loopback:10.100.0.4)	NODO SD1 AWS	ens5 (loopback:10.100.0.5)	WAN1.SD2	94.92 ms	0.00 %	Connected	
2	NODO SD2 VBOX	enp0s3 (loopback:10.100.0.6)	NODO SD3 VMWARE	ens160 (loopback:10.100.0.7)	WAN1.SD2	8.89 ms	0.00 %	Connected	
3	NODO SD2 VBOX	enp0s8 (loopback:10.100.0.8)	NODO SD3 VMWARE	ens192 (loopback:10.100.0.9)	WAN2.SD2	N/A	N/A	Not Connected	

Figura 61 Túnel WAN2 caido

Con en el túnel no establecido, se comprueba que la ruta ha conmutado al túnel de la WAN1,

En el nodo-sd2,

```
nodo-sd2# sh ip ro
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

K>* 0.0.0.0/0 [0/0] via 192.168.149.2, vpp0, 00:42:38
C>* 10.0.2.0/24 is directly connected, vpp1, 00:42:38
O   10.100.0.4/31 [110/10000] is directly connected, vpp3, 00:42:38
C>* 10.100.0.4/31 is directly connected, vpp3, 00:42:38
O   10.100.0.6/31 [110/10000] is directly connected, vpp5, 00:42:38
C>* 10.100.0.6/31 is directly connected, vpp5, 00:42:38
O   10.100.0.8/31 [110/1000] is directly connected, vpp7, 00:42:38
C>* 10.100.0.8/31 is directly connected, vpp7, 00:42:38
O   172.18.1.0/24 [110/10000] is directly connected, vpp2, 00:42:38
C>* 172.18.1.0/24 is directly connected, vpp2, 00:42:38
O>* 172.19.1.0/24 [110/20000] via 10.100.0.5, vpp3, 00:42:28
O>* 172.20.0.0/24 [110/20000] via 10.100.0.7, vpp5, 00:01:58
C>* 192.168.149.0/24 is directly connected, vpp0, 00:42:38
nodo-sd2#
```

Figura 62 Tabla de rutas del nodo-sd2

La línea,

```
O>* 172.20.0.0/24 [110/20000] via 10.100.0.7, vpp5, 00:01:58
```

Muestra que ahora la ruta va a través de la IP 10.100.0.7 (WAN1)

Y en el nodo-sd3,

```
nodo-sd3# sh ip ro
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r - rejected route

K>* 0.0.0.0/0 [0/0] via 92.118.100.1, vpp2, 00:42:18
O>* 10.100.0.4/31 [110/20000] via 10.100.0.6, vpp5, 00:01:48
O   10.100.0.6/31 [110/10000] is directly connected, vpp5, 00:42:18
C>* 10.100.0.6/31 is directly connected, vpp5, 00:42:18
O   10.100.0.8/31 [110/1000] is directly connected, vpp3, 00:21:14
C>* 10.100.0.8/31 is directly connected, vpp3, 00:42:18
C>* 92.118.100.0/24 is directly connected, vpp2, 00:42:18
K>* 92.118.102.61/32 [0/1] via 192.168.97.8, vpp0, 00:42:18
O>* 172.18.1.0/24 [110/20000] via 10.100.0.6, vpp5, 00:01:48
O>* 172.19.1.0/24 [110/30000] via 10.100.0.6, vpp5, 00:01:48
O   172.20.0.0/24 [110/10000] is directly connected, vpp1, 00:42:18
C>* 172.20.0.0/24 is directly connected, vpp1, 00:42:18
C>* 192.168.97.0/24 is directly connected, vpp0, 00:42:18
```

Figura 63 Tabla de rutas del nodo-sd3

Donde la línea,

```
O>* 172.18.1.0/24 [110/20000] via 10.100.0.6, vpp5, 00:01:48
```

Muestra que la ruta al nodo-sd2 es a través de la IP 10.100.0.6, que corresponde a la WAN1.

Tras la conmutación se prueba la traza desde el host server-nodo3,

```
david@serverVMWare:~$ tracepath -n 172.18.1.2
 1?: [LOCALHOST]                      pmtu 1350
 1:  172.20.0.1                                            0.098ms asymm  2
 1:  172.20.0.1                                            0.089ms asymm  2
 2:  10.100.0.6                                            6.752ms asymm  3
 3:  172.18.1.2                                            6.932ms reached
     Resume: pmtu 1350 hops 3 back 3
david@serverVMWare:~$
```

Figura 64 Traza desde server-nodo3

Confirmando que el túnel de respaldo ha entrado en funcionamiento.

A modo de resumen, en la figura 65 se muestran los dos caminos que hay para que el tráfico fluya entre nodos. El tráfico se enruta a través del túnel de la WAN2, en azul en la figura. Cuando el túnel cae, conmuta de forma automática al túnel de la WAN1, en rojo en la figura.

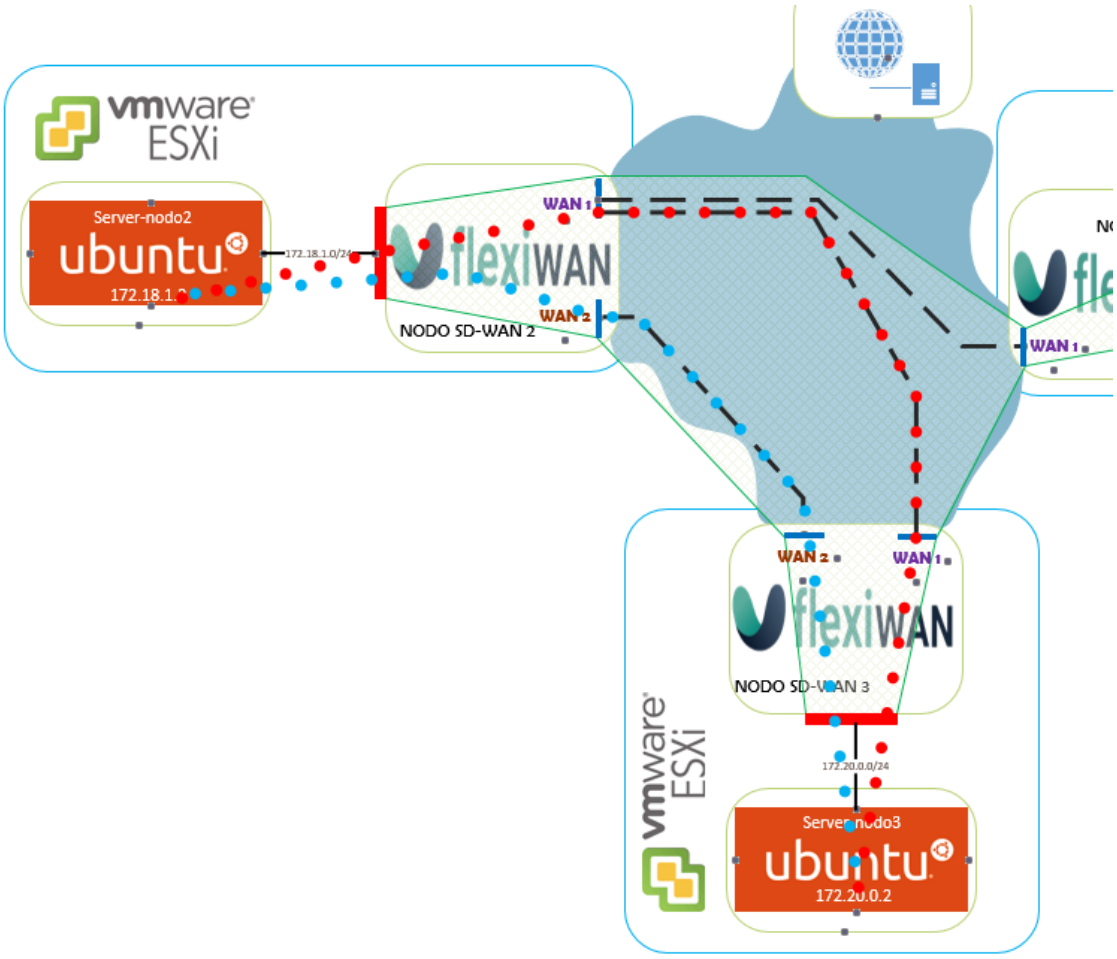


Figura 65 Resumen de encaminamiento del tráfico

5.5 Internet Breakout

Finalmente, se comprueba que el nodo 2 tiene habilitado el acceso a internet.

```
david@server_nodo2:~$ curl -I https://www.uoc.edu/
HTTP/1.1 200 OK
Date: Sun, 07 Jun 2020 11:02:25 GMT
Server: Apache
Last-Modified: Mon, 03 Feb 2020 09:03:02 GMT
Accept-Ranges: bytes
X-Powered-By: ModLayout/5.1
Content-Type: text/html
Set-Cookie: BIGipServerportal_webserver=818587840.20480.0000; path=/; Httponly
```

Figura 66 Obtención de cabeceras HTTP

Como se ve en la figura 66, desde el servidor se accede a la web, para ello, se ha mostrado la obtención de las cabeceras de esta mediante curl.

6. Conclusiones

6.1 Migración de servicios a SD-WAN

Migrar los servicios de una red tradicional a una red basada en software distribuido tiene una gran ventaja económica. Los costes de operación de una red MPLS son altos, además, si la red es grande todavía es más evidente el coste de estos servicios. Gracias a las tecnologías SD-WAN se pueden reducir costes de infraestructura y en recursos humanos, es bien sabido que administrar una red MPLS es complejo y cada operación requiere de manipular varios elementos de la red. En caso de una red grande esta labor lleva mucho tiempo y es peligrosa, ya que cuanto más se intervenga, más probable es que se genere un error humano que provoque una falla.

En las redes SD-WAN la administración es sencilla, no se requieren grandes conocimientos técnicos para crear conexiones entre sedes y encaminar tráfico entre ellas, gracias a esto se reducen mucho los costes en mano de obra.

6.2 Implementación y pruebas

La solución elegida no ha resultado ser tan estable como se esperaba en un principio. El desarrollo del proyecto se ha encontrado con varias dificultades que ha habido que ir solventando, a veces rodeándolas y otras consultando con el fabricante. [25][26]

Al crear los túneles entre sedes, en el nodo 3 no se establecían los túneles. Tras consultar al fabricante este confirma que hay un bug con el servicio VPP que hace que cada vez que se manipulan datos referentes a la red sea necesario parar el servicio y volverlo a arrancar. Esto ha provocado que pueda haber inconsistencias en los interfaces vpp y sus IPs descritos durante la implantación y pruebas, ya que al arrancar se reasigna el nombre de interfaz y su IP.

Otro de los problemas encontrados ha sido el servicio de *MultiWAN*, para que funcione es necesario crear rutas estáticas a través de la WAN2 para alcanzar el otro extremo del túnel. Este punto también ha sido reconocido por el fabricante, que ha comentado que está previsto resolverlo en próximas actualizaciones.

Un tercer problema ha venido derivado de la implementación del protocolo OSPF. En el caso de una red *MultiWan*, los routers no dan más peso a un túnel que ha otro, y por tanto el tráfico es balanceado entre los túneles disponibles por igual. Para evitar este comportamiento se ha modificado la configuración de OSPF para dar más prioridad al túnel principal y así solo usar el de respaldo cuando el primero falle.

6.3 Conclusión final

Aunque sobre el papel la solución SD-WAN de flexiWAN está preparada para un entorno en producción, tras las pruebas se ha comprobado que todavía tiene margen de mejora y problemas pendientes de resolver.

Respecto a los objetivos marcados, pese a las dificultades se han conseguido alcanzar todos ellos, se han instalado tres nodos SD-WAN, uno de ellos en la nube de Amazon AWS. Se ha conseguido instalar un nodo en un uCPE. La comunicación extremo a extremo entre nodos se ha establecido satisfactoriamente y se ha conseguido establecer una comunicación entre dos nodos redundada.

Además, las pruebas de rendimiento han dado un buen resultado, confirmando que un entorno SD-WAN es perfectamente válido para su operación.

7. Acrónimos

UCS. *Underlay Connectivity Services*. Son las redes de transporte por donde se establecen las conexiones SD-WAN. Internet, MPLS, etc.

UNI. *SD-WAN User Network Interface*. Son los interfaces que conectan con los UCS o con el suscriptor.

TVC. *Tunnel Virtual Connection*. Túneles virtuales. Conexiones punto a punto sobre los UCS. Por ejemplo, VPNs construidas sobre MPLS.

VNF. *Virtual Network Function*. Función de red virtualizada.

uCPE. *universal Custome Premise Equipment*. Es una plataforma de computación que se despliega en las instalaciones de la corporación y que tiene capacidad para ejecutar VNFs.

Application Flow, flujos de aplicación. Se describe como un conjunto de paquetes IP que comparten el valor de algunos campos de sus cabeceras (desde nivel 2 al nivel 7). Por ejemplo, se puede definir un flujo de aplicación como todos los paquetes que se transporten sobre RTP (*Real Time Protocol*), o como todos los paquetes que pertenezcan a una sesión de video conferencia

SaaS. *Software as a Service*. Paquete software que está instalado en la infraestructura de terceros y se puede arrendar.

IaaS. *Infraestructure as a Service*. Infraestructura que está instalada en una ubicación de un tercero y se puede arrendar.

Red Overlay. Se dice de una red virtual que funciona sobre otra, normalmente física.

On-premise. Se dice de una instalación que se efectúa en una ubicación propia en lugar de en la nube.

LAN. *Local Area Network.* Red de área local.

WAN. *Wide Area Network*. Red de área extensa.

ORLA. Oferta de Referencia de líneas Alquiladas. Es la oferta regulada por la que telefónica está obligada a arrendar su infraestructura de líneas terminales.

Latencia. El tiempo, en milisegundos, necesario para recorrer de un extremo a otro un circuito.

RTT. Tiempo que tarda un paquete de datos enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor de destino.

Jitter. Variabilidad temporal durante el envío de señales digitales.

IOS. Sistema operativo que corre en los equipos del fabricante Cisco Systems.

OSPF. *Open Shortest Path First*. Protocolo de enrutamiento dinámico.

8. Bibliografía

[1] Metro Ethernet Forum. *MEF 70 - SD-WAN Service Attributes and Services Definition.* [Online] mef.net. Disponible en https://wiki.mef.net/display/CESG/MEF+70+-+SD-WAN+Service+Attributes+and+Services+Definition (Accedido en Marzo de 2020)

[2] Metro Ethernet Forum. *MEF 3.0 SD-WAN Services.* [Online] mef.net. Disponible en https://www.mef.net/mef-white-paper-request/ (Accedido en Marzo de 2020)

[3] Omdia. *Data Center Network Equipment Market Tracker*. [Online] informa.com. Disponible en https://technology.informa.com/550561/data-center-network-equipment-market-tracker (Accedido en Abril de 2020)

[4] Sdxcentral. *Omdia SD-WAN Report: Fuel to VMware, Cisco Fire?.* [Online] sdxcentral.com. Disponible en https://www.sdxcentral.com/articles/news/omdia-sd-wan-report-fuel-to-vmware-cisco-fire/2020/04/ (Accedido en Abril de 2020)

[5] Amazon. *Precios de Amazon EC2*. [Online] amazon.com. Disponible en https://aws.amazon.com/es/ec2/pricing/on-demand/ (Accedido en Abril de 2020)

[6] CNMC. *OFERTA DE LÍNEAS ALQUILADAS DE TELEFÓNICA DE ESPAÑA*. [Online] cnmc.es. Disponible en https://www.cnmc.es/sites/default/files/1517407_25.pdf (Accedido en Abril de 2020)

[7] Movistar. *Telefónica empresas*. [Online] movistar.es. Disponible en http://www.movistar.es/empresas/para-tu-oficina/conectividad-internet/adsl-empresas/ (Accedido en Abril de 2020)

[8] Cisco. *Cisco CCW*. [Online] cisco.com. Disponible en https://apps.cisco.com/cfgcor/public/app/servicesubscriptionui/views/servicesubscriptionui.jsp?rl=y&appid=BNP&sId=1589739543125&plp=Y&pId=5ec1804ccc4c3c486bf26fb7#/home (Accedido en Abril de 2020)

[9] Levine, S. S., & Prietula, M. J. (2014). *Open collaboration for innovation: Principles and performance.* [Online] ssrn.com. Disponible en https://poseidon01.ssrn.com/delivery.php?ID=210112099027097067024079092064002006006053019011070090014119022123120079006064067098037057101062049011109094030112127094113068106036006035001086096072090004002067009075052118012112080087115110107066070075115091028085092074108005124006064065002106100&EXT=pdf (Accedido en Marzo de 2020)

[10] ONF. *E-CORD*. [Online] opennetworking.org. Disponible en https://www.opennetworking.org/e-cord/ (Accedido en Abril de 2020)

[11] Flexiwan. *Welcome to flexiWAN's documentation!* [Online] flexiWAN.com. Disponible en https://docs.flexiwan.com/ (Accedido en Abril de 2020)

[12] Cisco. *Viptela.* [Online] cisco.com. Disponible en https://www.cisco.com/c/en/us/about/corporate-strategy-office/acquisitions/viptela.html (Accedido en Abril de 2020)

[13] Nokia. *Nuage Networks VNS Solution Sheet*. [Online] nokia.com. Disponible en https://onestore.nokia.com/asset/183178 (Accedido en Abril de 2020)

[14] Cisco. *Cisco SD-WAN Cloud scale architecture.* [Online] cisco.com. Disponible en https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/sd-wan/nb-06-cisco-sd-wan-ebook-cte-en.pdf (Accedido en Abril de 2020)

[15] Edge-Core. *Virtualization of CPEs*. [Online] Edge-dore.com. Disponible en https://www.edge-core.com/solution-inquiry.php?cls=5&id=55 (Accedido en Abril de 2020)

[16] CompSource. *Precio SAF4100I*. [Online] compsource.com. Disponible en https://www.compsource.com/buy/SAF4100I/vid=Smc-Networks-407 (Accedido en Abril de 2020)

[17] Advantech. *FWA-T011.* [Online] advantech.com. Disponible en https://www.advantech.com/products/6e5d3cbf-5420-4e95-9a4a-9a51d929623b/fwa-t011/mod_c9ae1b84-eedc-48f9-8fd2-d6c0cf79db84 (Accedido en Abril de 2020)

[18] Enea. *Enea NFV Access - Software Platform for Advantech White box uCPEs*. [Online] enea.com. Disponible en https://www.enea.com/globalassets/downloads/nfvi-platforms/enea-nfv-access/solution-brief---scalable-ucpe-from-enea-and-advantech (Accedido en Abril de 2020)

[19] Luchengtech. *Precio FWA-T011-2CA1S Tiny Network Appliance.* [Online] alibaba.com. Disponible en https://xinghuatech.en.alibaba.com/product/62093585397-804805630/Advantech_FWA_T011_2CA1S_Tiny_Network_Appliance.html (Accedido en Abril de 2020)

[20] Ubuntu. *Ubuntu documentation*. [Online] Ubuntu.com Disponible en https://help.ubuntu.com/ (Accedido en Mayo de 2020)

[21] Oracle. *VirtualBox*. [Online] virtualbox.org. Disponible en https://www.virtualbox.org/ (Accedido en Mayo de 2020)

[22] VMWare. *ESXi.* [Online] vmware.com. Disponible en https://www.vmware.com/es/products/esxi-and-esx.html (Accedido en Mayo de 2020)

[23] Amazon. *AWS Cloud Computing.* [Online] amazon.com. Disponible en https://aws.amazon.com/es/ (Accedido en Mayo de 2020)

[24] Vandenberghe, Ruben. Measuring throughput: effect of used TCP settings. [Online] excentis.eu. Disponible en https://www.excentis.eu/blog/measuring-throughput-effect-used-tcp-settings (Accedido en Mayo de 2020)

[25] Flexiwan Forum. *Tunnel Not Connected*. [Online] Google.com. Disponible en https://groups.google.com/a/flexiwan.com/forum/#!topic/flexiwan-users/4CeU-AOtYTA (Accedido en Mayo de 2020)

[26] Flexiwan Forum. *How does the multilink feature work?* [Online] Google.com. Disponible en https://groups.google.com/a/flexiwan.com/forum/#!topic/flexiwan-users/5mQ02jMav5I (Accedido en Mayo de 2020)

Anexos

I. Despliegue de flexiEdge en Amazon AWS.

Como se ha visto en el apartado 3, se procede a la instalación desde una maquina Linux, instalando en primer lugar los paquetes necesarios de Ansible.

```
sudo apt update
sudo apt install software-properties-common
sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible python-boto3 python3-boto3 python-boto python3-boto
```

En el panel de administración de AWS es necesario crear un usuario para obtener el ID de clave y la clave secreta que el script usará para crear la instancia en AWS.

Para ello hay que abrir el panel de administración IAM y crear un usuario con permiso de acceso mediante programación.

Tras crear el usuario se muestran las claves necesarias para la instalación a través del script.

Antes de ejecutar el script, es necesario fijar las variables para la instalación. Además, flexiWan por defecto crea una instancia de tipo m5.large. En caso de que sea necesario una instancia de otro tipo, hay que modificar el script de Ansible ya que no hay opción a través de variable tal cual viene el script.

Las variables configurables por el script son,

```
región=eu-central-1
vpc_name=VPC
vpc_cidr_block=172.18.0.0/16
cidr_lan=172.18.1.0/24
cidr_wan=172.18.254.0/24
lan_ip_address=172.18.1.254
flexiwan_token=ey******
stack= RouterAWS
```

Resultando en el siguiente comando para este caso,

```
ansible-playbook ec2_create_customer.yml --extra-vars "region=eu-west-1 vpc_name=VPC
vpc_cidr_block=172.18.0.0/16 cidr_lan=172.18.1.0/24 cidr_wan=172.18.254.0/24
lan_ip_address=172.18.1.254 flexiwan_token=*** stack=RouterAWS"
```

Tras la instalación aparece un resumen del resultado de las tareas realizadas.

```
PLAY RECAP *********************************************************************************************************
127.0.0.1                  : ok=27   changed=11   unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

18.196.41.7                : ok=17   changed=13   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figura anexa 1 Resumen instalación

Tras lo cual, y transcurridos unos minutos, aparece el nodo en el inventario de flexiManage.

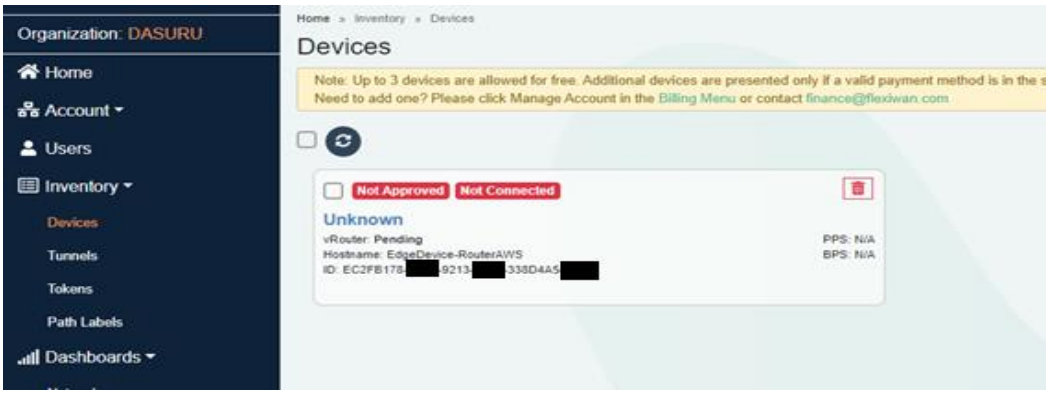


Figura anexa 2 Inventario flexiManage

Los pasos a seguir son comunes a todas las plataformas y se explican más adelante.

En la plataforma de AWS es necesario crear una nube privada virtual (VPC) y subredes para dar el servicio de red a las maquinas. El script ejecutado anteriormente crea todos estos espacios, además de las rutas necesarias para la comunicación.

En la siguiente figura se muestra la VPC creada por el script junto a la que Amazon añade por defecto,

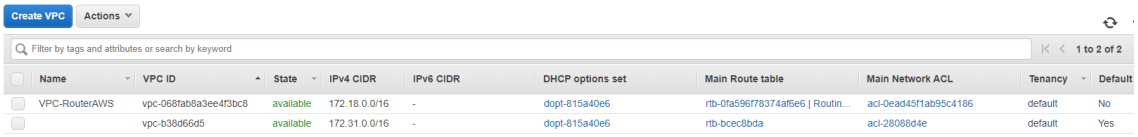


Figura anexa 3 VPC en AWS

Además de la VPC, se crean subredes, rutas, puntos de salida a internet e IPs públicas (*Elastic* IP)

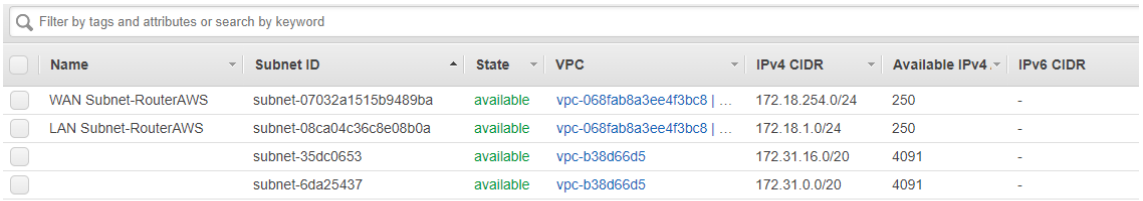


Figura anexa 4 Tabla de redes en AWS

Tras esto, ya se puede disponer de un nodo SD-WAN en la nube corporativa, para poder conectar los servidores de esta nube con los sitios remotos.

Para este proyecto, se va a crear un servidor web que será accesible a través de la red SD-WAN.

Para ello, hay que navegar en AWS hasta el panel de Instancias EC2 y pulsar sobre *Launch Instance*. Aparece una guía para el despliegue donde se configuran los parámetros de CPU, memoria, almacenamiento, red y la seguridad que proporciona AWS.

Existe un tipo de instancia que se puede usar de forma gratuita, lo que Amazon llama *free-tier*, que es la que se va a desplegar, con un sistema Linux Ubuntu.

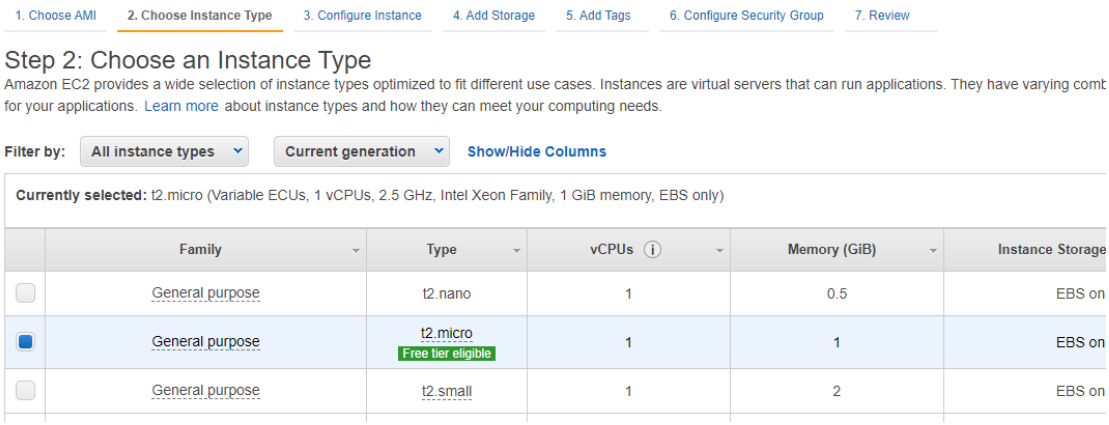


Figura anexa 5 Selección de instancia en AWS

En la parte de red es importante seleccionar que el servidor estará en la parte LAN de la VPC,

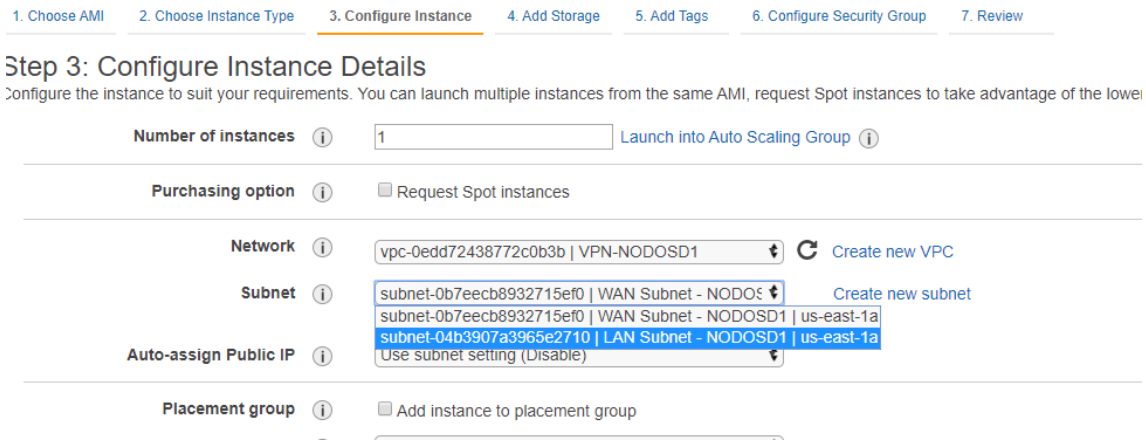


Figura anexa 6 Selección de VPC para la instancia en AWS

Al finalizar el asistente es necesario establecer las claves SSH para conectarse a la maquina o usar unas claves creadas anteriormente.

AWS por defecto no permite el tráfico a una instancia que no va dirigido a ella, por lo que es necesario habilitar en el nodo SD-WAN la comprobación de origen/destino.

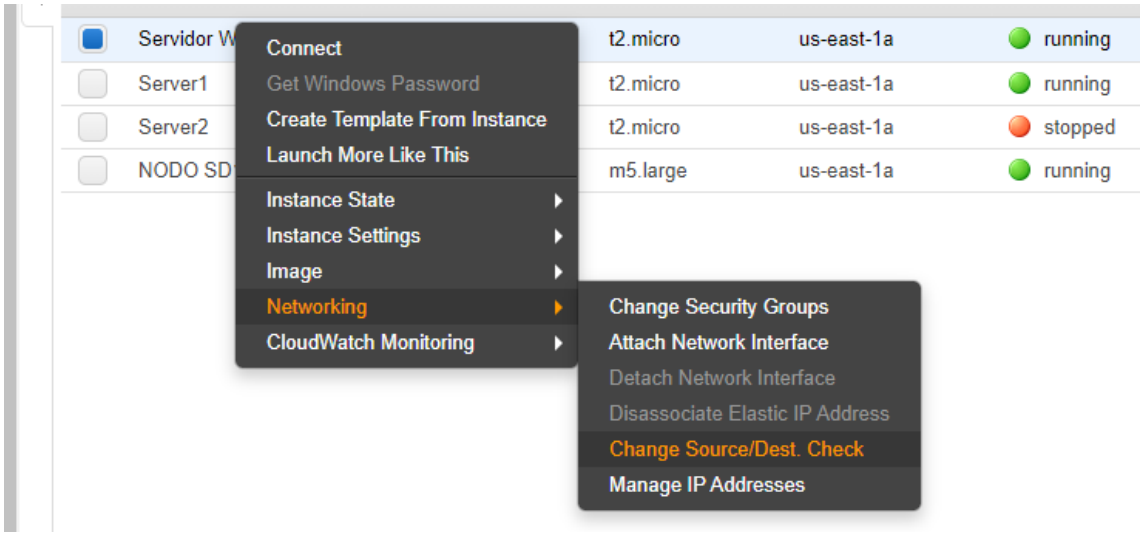


Figura anexa 7 Habilitar en el nodo el forwarding

Tras iniciar la maquina se puede comprobar que hay continuidad entre el nodo SD-WAN y el servidor WEB.

```
ubuntu@ip-172-19-1-73:~$
ubuntu@ip-172-19-1-73:~$ ping 172.19.1.1
PING 172.19.1.1 (172.19.1.1) 56(84) bytes of data.
64 bytes from 172.19.1.1: icmp_seq=1 ttl=255 time=0.320 ms
64 bytes from 172.19.1.1: icmp_seq=2 ttl=255 time=0.277 ms
64 bytes from 172.19.1.1: icmp_seq=3 ttl=255 time=0.296 ms
64 bytes from 172.19.1.1: icmp_seq=4 ttl=255 time=0.320 ms
64 bytes from 172.19.1.1: icmp_seq=5 ttl=255 time=0.406 ms
^C
--- 172.19.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4092ms
```

Figura anexa 8 Ping desde el servidor al nodo

II. Despliegue de flexiEdge en una plataforma de propósito general o máquina virtual.

Como se ha visto en el apartado 3.3.1, es necesario una versión de Ubuntu 18.04 que se puede descargar de su web.

Se puede desplegar en un servidor dedicado o en una plataforma de virtualización. A continuación, se muestra la instalación en una máquina virtual usando Oracle Virtual Vox.

Al pulsar en crear nueva máquina se inicia el asistente para la configuración.

Siguiendo los requisitos del sistema flexiEdge, necesitamos que la maquina disponga de 4 gigabytes de memoria RAM y 2 CPUs. también hacen falta al menos dos adaptadores de red, uno para la WAN y otro para la LAN. En este caso se han seleccionado tres adaptadores, dos para WAN y uno para LAN.

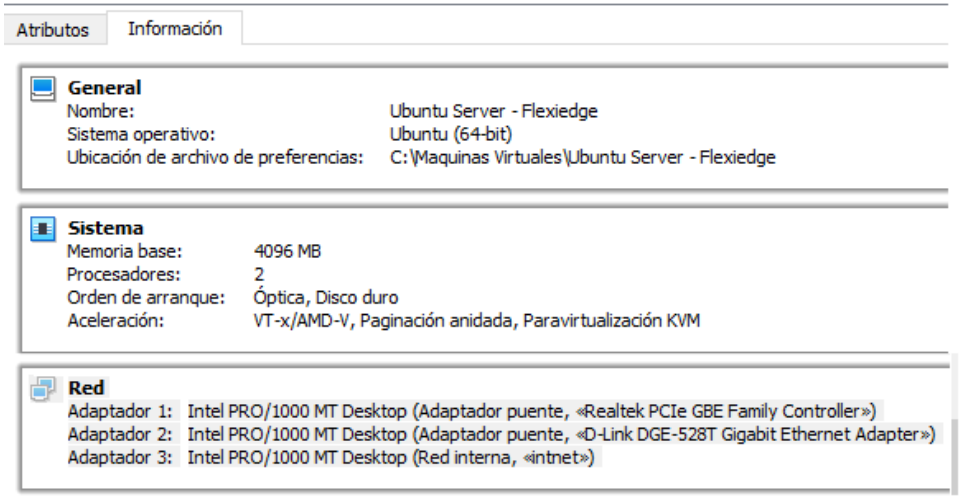


Figura anexa 9 Resumen de la configuración de Oracle Virtualbox

La ventaja del despliegue en una plataforma de nube publica es que ya dispone de imagen de los principales sistemas operativos por lo que no es necesario como en este caso la instalación del sistema.

Tras instalar el sistema Ubuntu, hay que proceder a la instalación de flexiEdge siguiendo los pasos comentados en el apartado 3.3.1.

Tras la instalación de flexiEdge se muestra un resumen del cumplimiento de los requisitos del sistema,

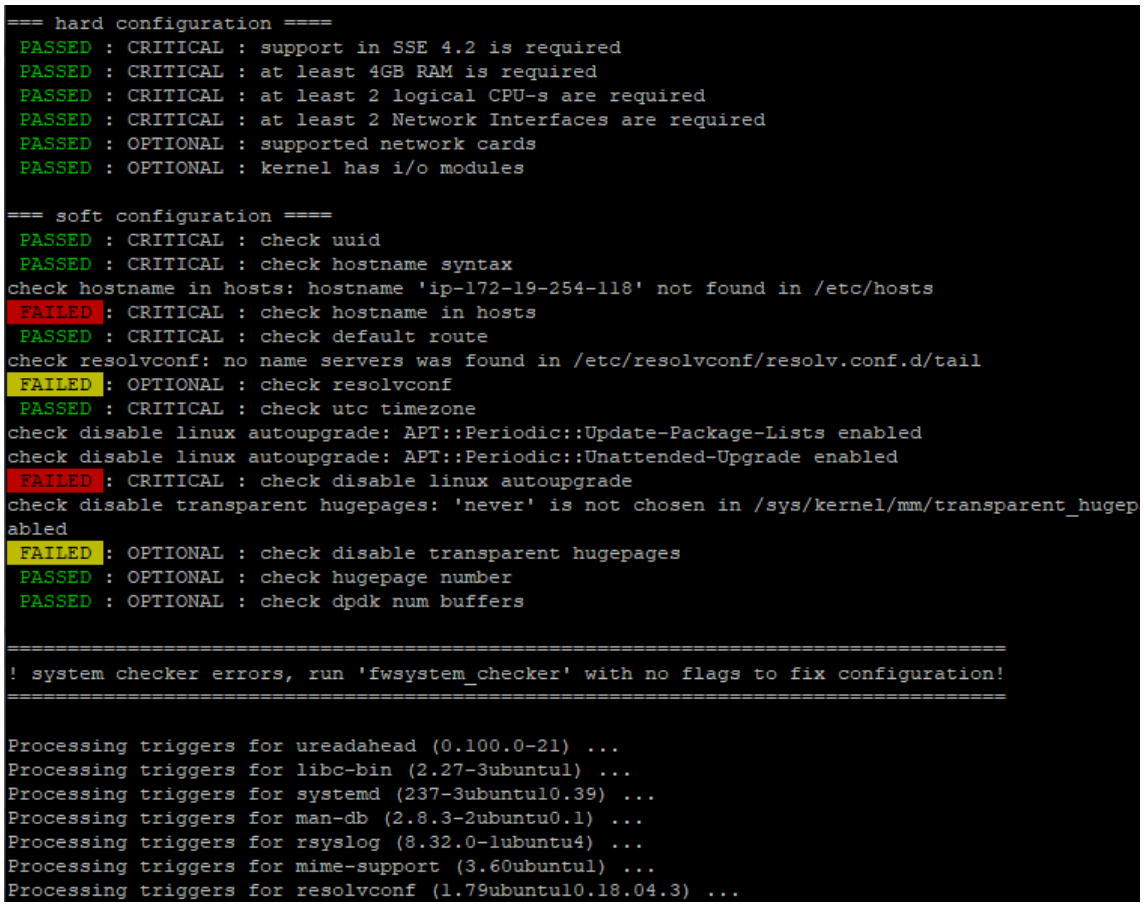


Figura anexa 10 Resumen de cumplimiento de los requisitos del sistema

Los requisitos de software son fácilmente subsanables, basta con ejecutar la utilidad *fwsystem_checker* y elegir la opción 3.

Tras ejecutar la opción 3, se puede comprobar el estado con la opción 2 y se apreciará que se han subsanado los errores críticos que aparecían,

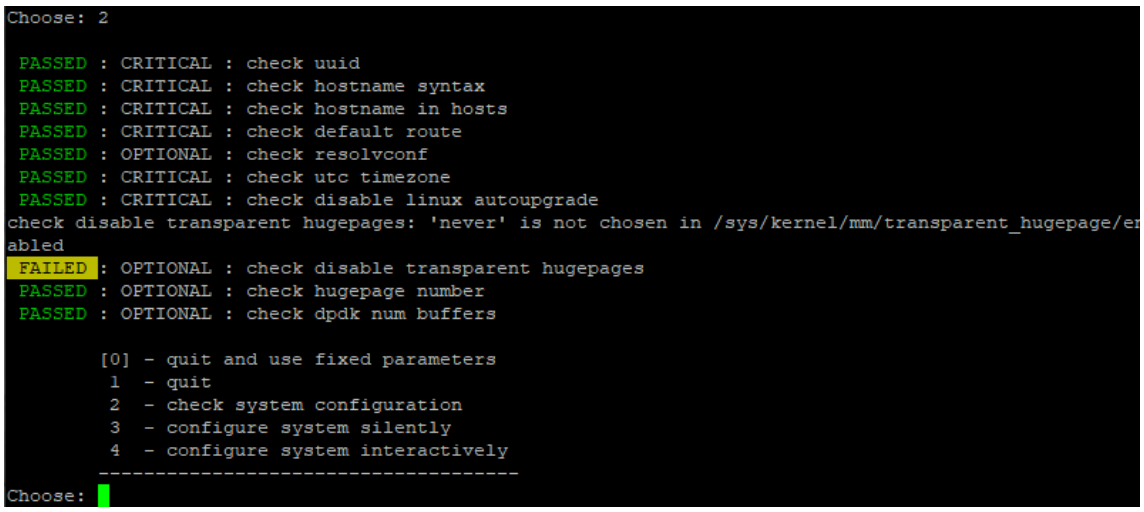


Figura anexa 11 Resultado de fwsystem_checker tras ejecutar la opción 2

Con el agente ya instalado solo resta añadir el token de conexión a nuestra organización al fichero /etc/flexiwan/agent/token.txt.

III. Pasos comunes para el despliegue del nodo flexiEdge.

Tras la instalación del agente, el nodo aparecerá en la consola de flexiManage. Lo primero que hay que hacer es aprobarlo para que se conecte con el orquestador. Para ello, se pulsa sobre el nombre del dispositivo y aparece la siguiente pantalla,

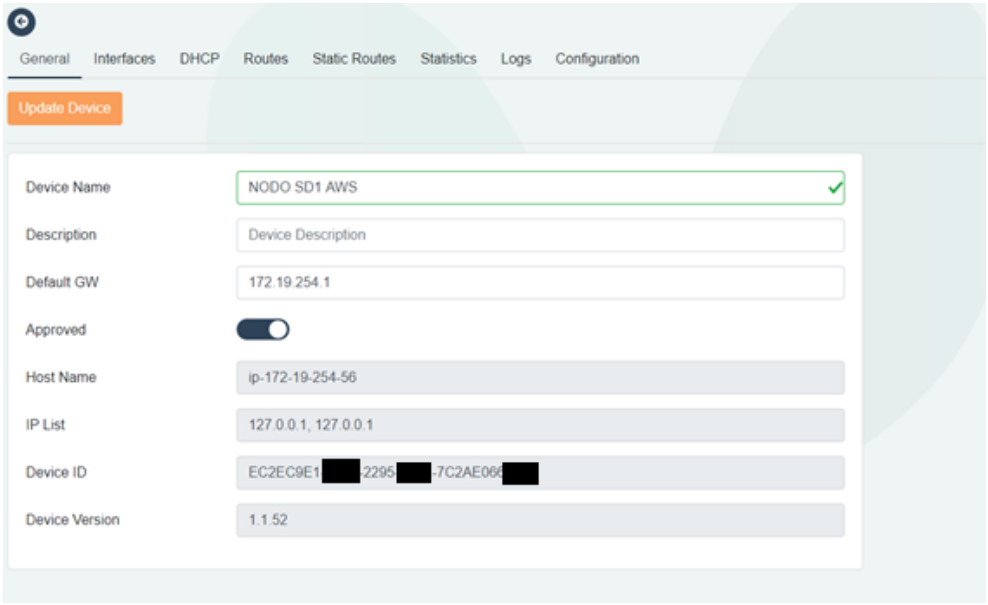


Figura anexa 12 Propiedades del nodo flexiEdge desde flexiManage

En la que se le da un nombre descriptivo y se aprueba activando el pulsador *Approved*. Tras ello, es necesario pulsar en *Update Device* para aplicar los cambios.

Con el nodo aprobado y conectado, es necesario revisar la configuración de los interfaces. En el caso del nodo desplegado en AWS aparecen los dos interfaces, uno configurado como WAN y otro como LAN, con las IPs asignadas a cada uno. El interfaz WAN tiene asignada la IP publica y el LAN usa como protocolo de enrutamiento OSPF. En este caso no es necesario ajustar nada en la configuración. Este nodo ya estaría listo para entrar en servicio.

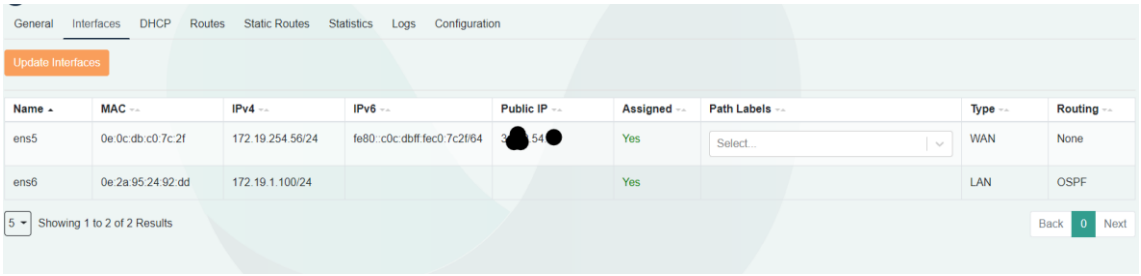


Figura anexa 13 Configuración de los interfaces de un nodo flexiEdge

El nodo SD2 se ha desplegado sobre Virtualbox y dispone de dos interfaces WAN,

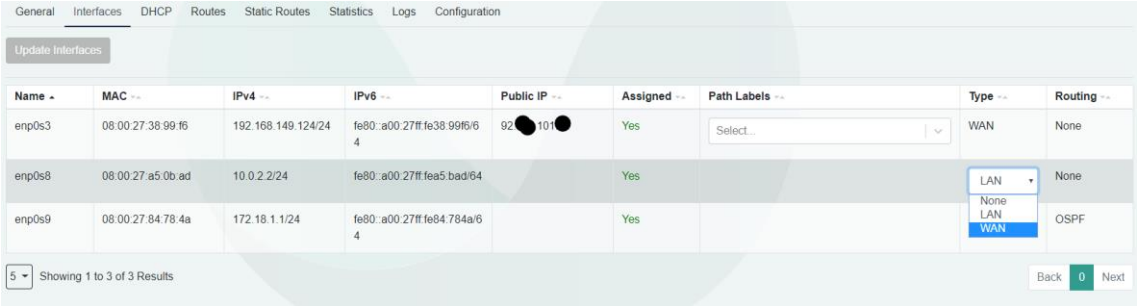


Figura anexa 14 Configuración de los interfaces del nodo SD2 flexiEdge

En este caso es necesario ajustar el segundo interfaz WAN ya que el sistema lo reconoce como un interfaz LAN. Para ello, se selecciona en el desplegable *Type* la opción WAN y se configura la IP publica en el campo *Public IP*.

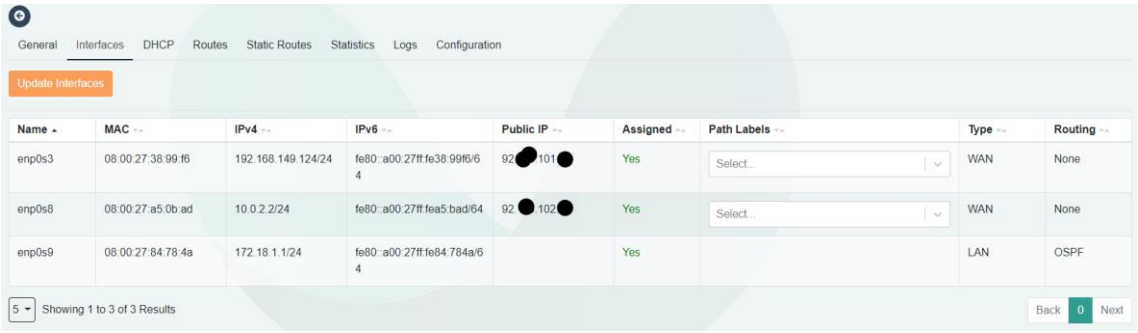


Figura anexa 15 Configuración de los interfaces del nodo SD2 flexiEdge

Además de los valores de interfaces se pueden configurar otros valores, como el servidor DHCP en caso de necesitar que se asignen IPs dinámicamente dentro de la red o enrutamiento estático para alcanzar subredes que quedan fuera del rango de los interfaces

El nodo SD3 también esta desplegado con dos interfaces WAN.

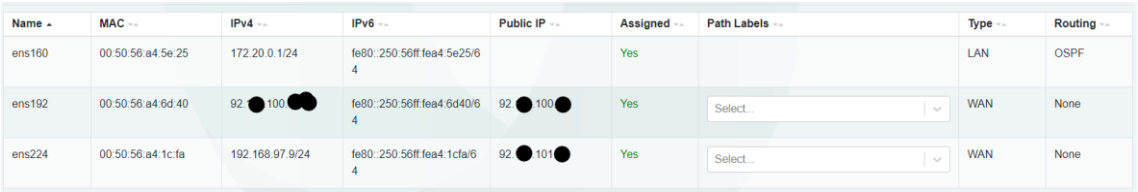


Figura anexa 16 Configuración de los interfaces del nodo SD3 flexiEdge

En este caso, como el nodo tiene una de sus WAN con la IP publica configurada directamente en la interfaz, aparecerá tanto en el campo IPv4 como en el de *Public IP*.

Una vez que esta todo configurado es necesario arrancar el enrutamiento en el agente flexiEdge. Para ello hay que darle al botón con forma de *play* que aparece en la ficha del nodo,

Una vez arrancado el router, aparecerá en ejecución en las fichas de los nodos,

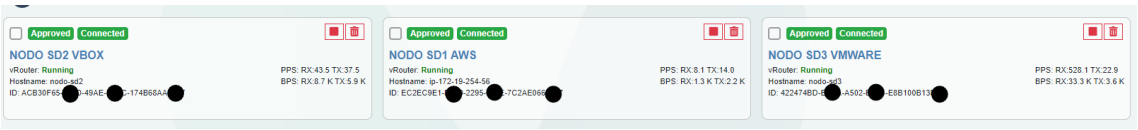


Figura anexa 17 Nodos configurados en el panel de flexiManage

Cuando los nodos SD-WAN estén ejecutándose será cuando se puedan conectar entre sí. Para ello, se seguirá la topología descrita anteriormente.

Para la conexión de los túneles, lo primero que hay que hacer es crear las etiquetas que decidirán el encaminamiento de los paquetes. Se van a establecer dos etiquetas, una para conectar las WAN1 de los nodos SD2 y SD3 y el nodo SD1 con el SD2, y otra para conectar las WAN2 de los nodos SD2 y SD3.

Figura anexa 18 Configuración de etiquetas para clasificar el trafico

Las etiquetas en sí mismas no configuran nada, es necesario aplicarlas a los dispositivos. Las dos etiquetas creadas se han llamado WAN1.SD2 y WAN2.SD2, para que sea fácil la identificación del interfaz usado.

Tras la creación de las etiquetas hay que aplicarlas a los interfaces. Para ello, en el panel de dispositivos hay que acceder a cada uno de los dispositivos y asignar las etiquetas a los interfaces.

En el caso del nodo SD2 y SD3 que tienen dos WAN, se asignará una etiqueta a cada interfaz.

Figura anexa 19 Configuración de las etiquetas en los interfaces

En el nodo SD1 se configurará solo una etiqueta para conectar con la WAN1 de SD2.

Figura anexa 20 Configuración de la etiqueta en el nodo SD2

El siguiente paso es crear los túneles, para ello, desde el panel de dispositivos se seleccionan los equipos entre los que se quieren crear. En el botón *Actions* se selecciona la opción de crear túneles y en el desplegable se eligen las etiquetas que seguirá el túnel.

Siguiendo la topología descrita se crearán tres túneles. Por ejemplo, para el túnel entre la WAN1 del nodo SD2 y la WAN del nodo SD1 se elegirá la etiqueta WAN1.SD2.

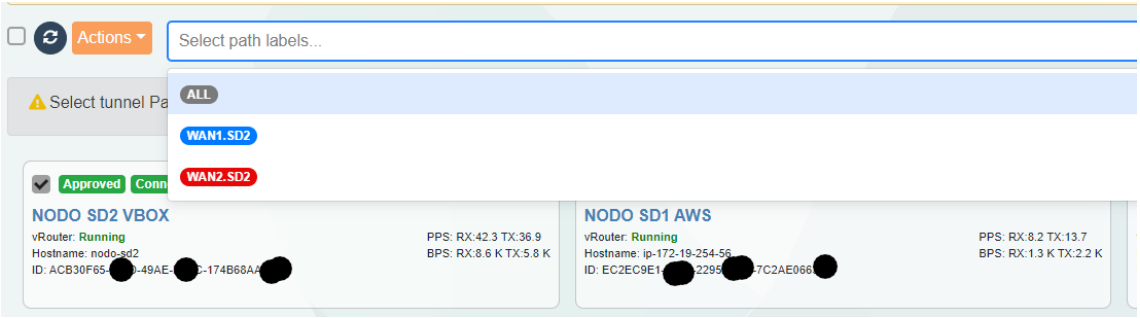


Figura anexa 21 Creación de túneles según la etiqueta

Después de crear todos los túneles se puede comprobar su estado en el panel de túneles,

ID	Device A	Interface A	Device B	Interface B	Path Label	AVG Latency	Drop Rate	Status	Actions
1	NODO SD2 VBOX	enp0s3 (loopback:10.100.0.4)	NODO SD3 VMWARE	ens192 (loopback:10.100.0.5)	WAN1.SD2	6.81 ms	0.00 %	Connected	
2	NODO SD2 VBOX	enp0s3 (loopback:10.100.0.6)	NODO SD1 AWS	ens5 (loopback:10.100.0.7)	WAN1.SD2	95.22 ms	0.00 %	Connected	
3	NODO SD2 VBOX	enp0s8 (loopback:10.100.0.8)	NODO SD3 VMWARE	ens224 (loopback:10.100.0.9)	WAN2.SD2	6.55 ms	0.00 %	Connected	

Figura anexa 22 Estado de los túneles

Otra característica importante es la de dotar de salida a internet a las ubicaciones que lo requieran, para ello es necesario crear una etiqueta que habilite la función de Internet Breakout y aplicarla al nodo en el interfaz por el que se provea esta conectividad.

Para ello, en el panel de etiquetas hay que activar la opción de *Direct Internet Access*.

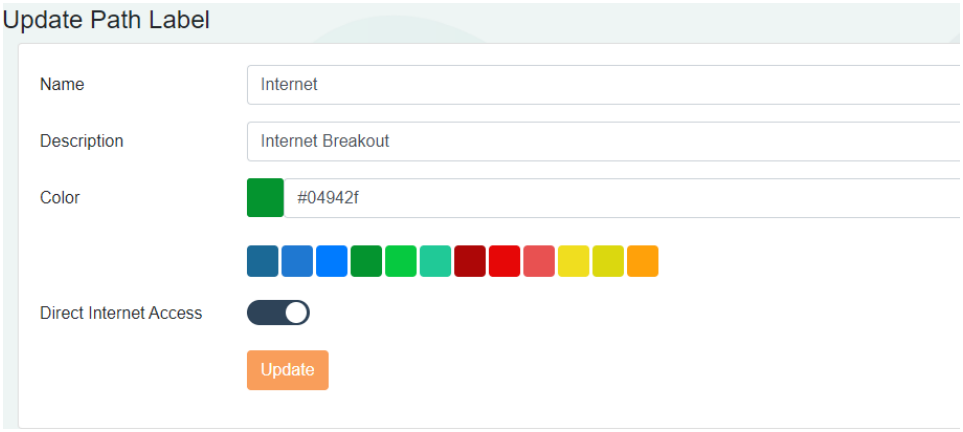


Figura anexa 23 Configuración de Internet Breakout

En el resumen del panel de etiquetas ya aparen las creadas para los túneles, que son del tipo *tunnel*, y la creada para internet, que es del tipo DIA.

Name	Type	Description
Internet	DIA	Internet Breakout
WAN1.SD2	Tunnel	Tunel usando WAN1 de SD2 a SD3 y SD1
WAN2.SD2	Tunnel	Tunel usando WAN2 de SD2 a SD3 y SD1

Figura anexa 24 Resumen de las etiquetas creadas

El siguiente paso por tanto será aplicar la etiqueta de internet en el nodo que lo requiera, por ejemplo, en la interfaz WAN1 del nodo SD2

Name	MAC	IPv4	IPv6	Public IP	Assigned	Path Labels	Type
enp0s3	08:00:27:38:99:f6	192.168.149.124/24	fe80::a00:27ff:fe38:99f6/64	92.[redacted]101	Yes	WAN1_SD2 × Internet ×	WAN
enp0s8	08:00:27:a5:0b:ad	10.0.2.2/24	fe80::a00:27ff:fea5:bad/64	92.[redacted]102	Yes	WAN2_SD2 ×	WAN
enp0s9	08:00:27:84:78:4a	172.18.1.1/24	fe80::a00:27ff:fe84:784a/64		Yes		LAN

Figura anexa 25 Configuración de Internet Breakout a un interfaz

En el panel de red de flexiManage se puede ver de forma esquemática la topología que se ha configurado. Al poner el cursor sobre un túnel, se muestran las estadísticas de latencia y perdidas de este.

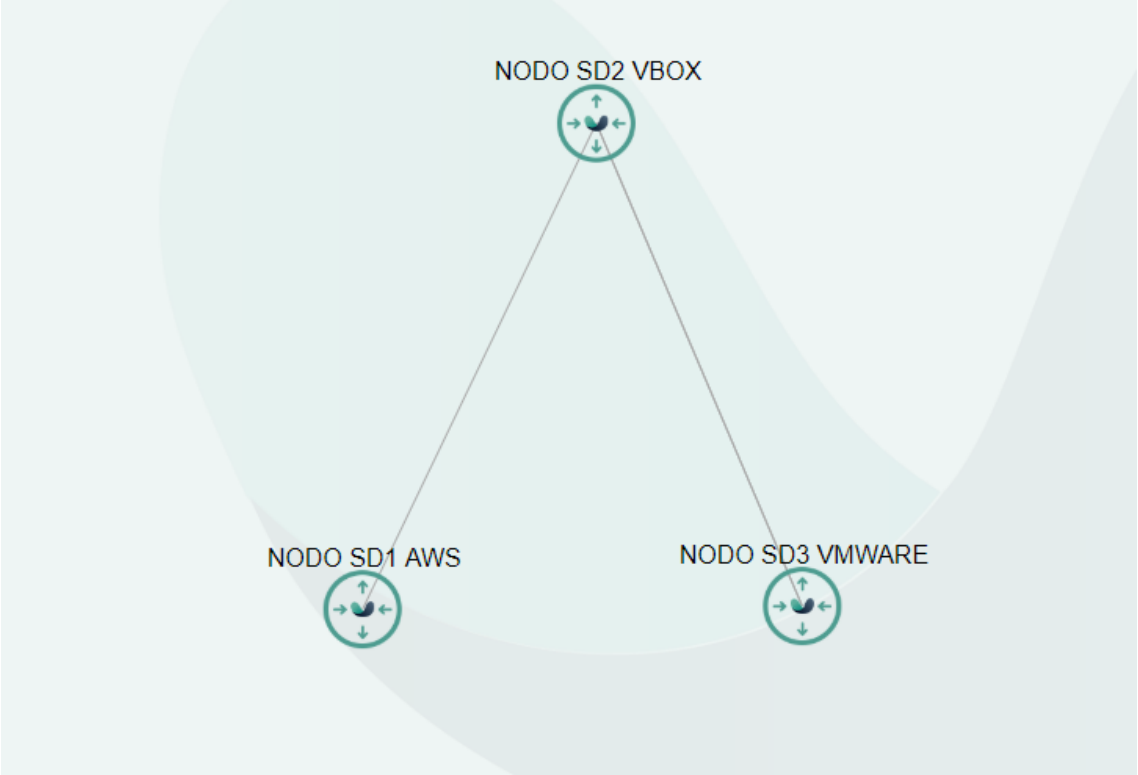


Figura anexa 26 Topología mostrada en flexiManage

IV. Topología SD-WAN del cliente sin cambiar sus UCS actuales.

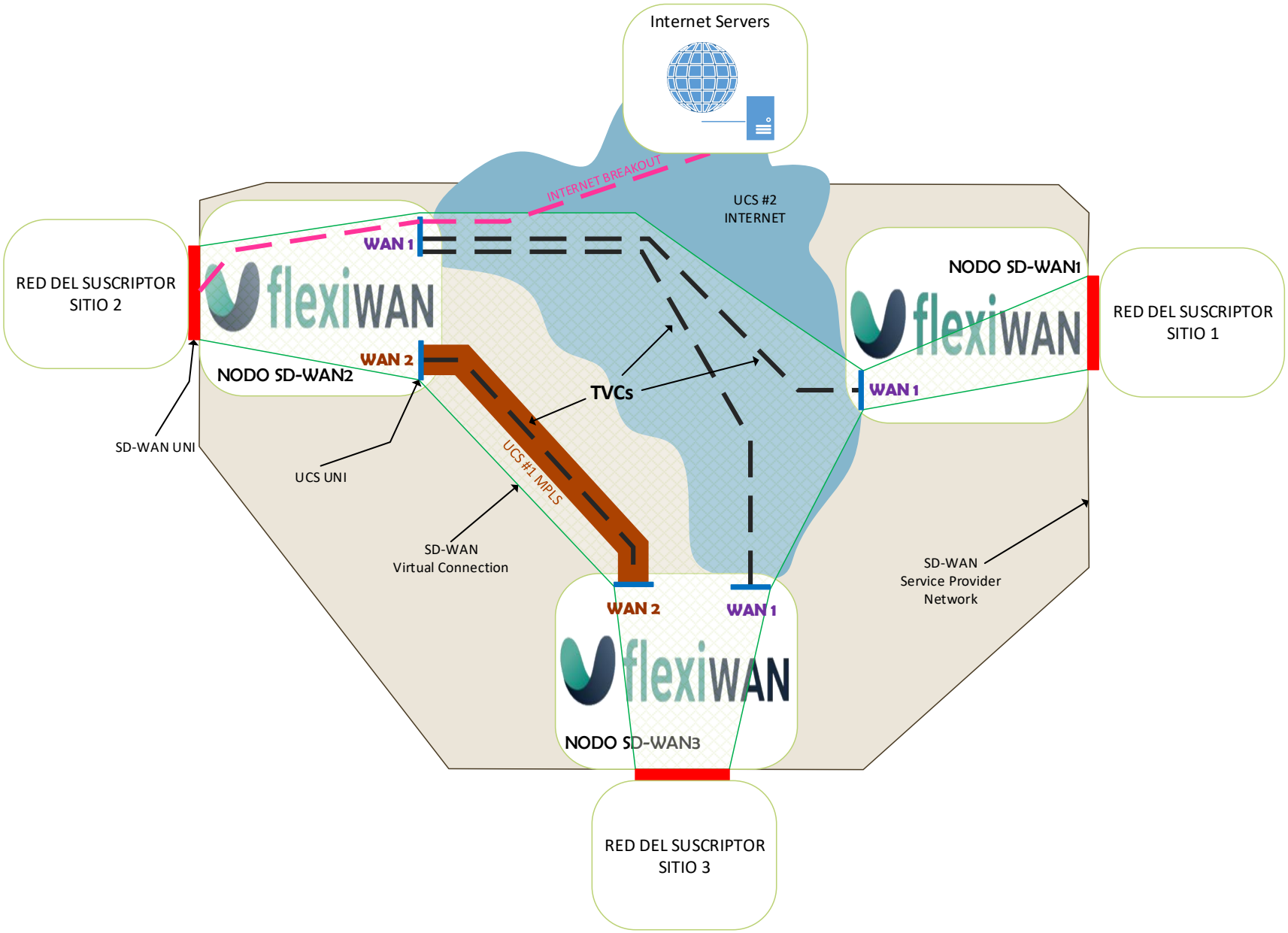


Figura anexa 27 Topología SD-WAN con un UCS tipo MPLS

V. Topología de red usada en las pruebas

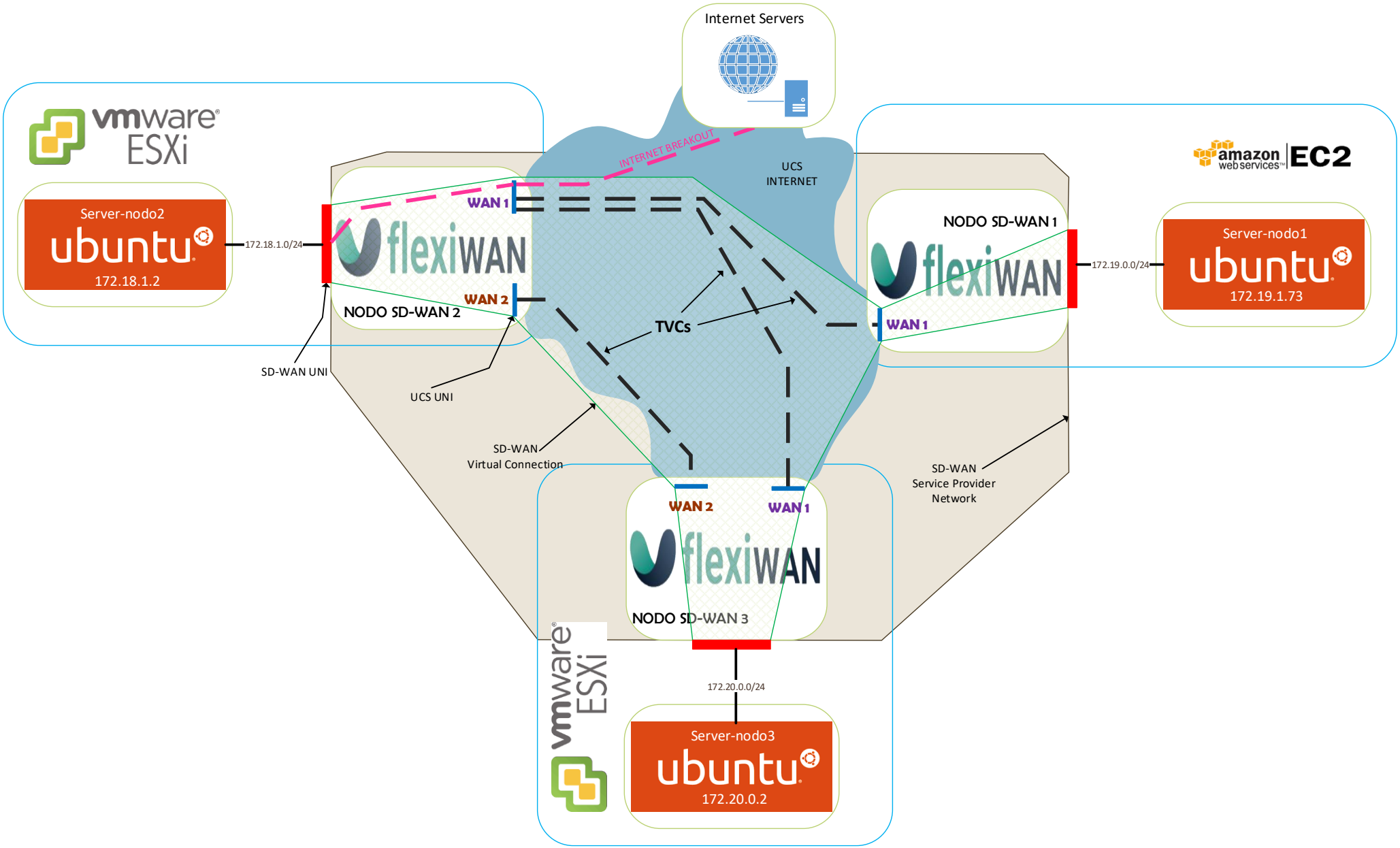


Figura anexa 28 Topología seguida durante las pruebas