1. INTRODUCCIÓN SDN

Las redes van evolucionando con el tiempo. Hoy en día se buscan algunas modificaciones como las que vemos a continuación:

AUTOMATIZACIÓN DE REDES

Se busca que el aprovisionamiento, la gestión y la seguridad de la red se realicen de forma automática para así maximizar su eficiencia. Además, también se busca la automatización de configuración, de pruebas e implementación entre otros casos.

Como beneficios tenemos:

- Implementación rápida de cambios
- Simplificación de tareas rutinarias
- Reducción de errores humanos
- Aumento de disponibilidad y mejora del servicio.

ORQUESTACIÓN

La automatización conlleva un orden y coordinación, por ello la orquestación se encarga de implementar los procesos automáticos de forma que todos los sistemas trabajan de forma conjunta. La principal diferencia entre Automatización y orquestación es que en esta última debe existir un flujo de trabajo o workflow entre las tareas de forma que estén coordinadas unas con otras, mientras que en la automatización se realizan de forma independiente.

PROGRAMABILIDAD

Es el uso de aplicaciones externas con el fin de monitorizar, configurar o gestionar incidencias mediante software. La conexión entre dos aplicaciones se lleva acabo sobre una API (Application Programming Interface)

NETWORK ABSTRACTION

Es la capacidad de configurar redes sin la necesidad de conocer que elementos físicos están detrás de ella.

NETWORK ANALYTICS

El análisis de datos y estadísticas de red es tendencia en la actualidad. Mediante un análisis correcto puede llevarse a cabo la mejora de rendimiento de las redes.

Actualmente se están realizando análisis con herramientas de automatización e inteligencia artificial. Requiere un gran uso del Machine Learning, el cual hace uso de algoritmos para el análisis y aprendizaje de los datos y del Deep Learning, el cual usa redes neuronales para la comprensión de estos.

INTENT-BASED NETWORKING (IBN)

IBN es un capa, sobre el plano de control, que traduce lo que se quiere hacer, en cómo hacerlo, enviando dicha información al controlador, para que este lo implemente en la infraestructura o a veces esta capa actúa como controlador y se lo envía directamente a la infraestructura.

NETWORK FUNCTIONS VIRTUALIZATION (NFV)

NFV consiste en virtualizar las funciones de red, por ejemplo, creando máquinas virtuales.

Como veremos en el tema, NFV y SDN son tecnologías complementarias, ya que SDN construye el camino para unir los planos de datos y control, mientras que NFV despliega las funcionalidades de red en los puntos de los planos necesarias. Un ejemplo de esta implementación en perfecto funcionamiento es el IoT.

2. PROTOCOLOS DE PROGRAMACIÓN EN REDES: SBI Y NBI

Comenzaremos viendo las SOUTHBOUND INTERFACES (SBI).

Programación basada en modelos

Como hemos visto, la automatización ayuda a que un solo operario sea capaza de configurar miles de dispositivos en poco tiempo, pero para ayudar a realizar esto es importante la programabilidad basada en modelos, ya que con ellos podremos conseguir una mejor automatización de los procesos.

Con modelos de datos (como por ejemplo YANG), proporcionaremos una base estructurada que facilita la interacción con los dispositivos de la red a través de software. Con este tipo de modelos podemos **configurar** (velocidad de interfaz), **supervisar** el estado del dispositivo (leer cuantos paquetes se han perdido), **recibir notificaciones** (advertencias de fallos) o **invocar acciones de manera remota** (reestablecer controladores de paquetes perdidos).

Algunos modelos compatibles son:

Native: Son modelos creados por fabricantes concretos y que son incompatibles con dispositivos de otros fabricantes

OpenConfig: Son modelos creados para configurarlos de múltiples formas para adaptarse a distintos proveedores, ya que pueden adaptarse a las necesidades de cada uno.

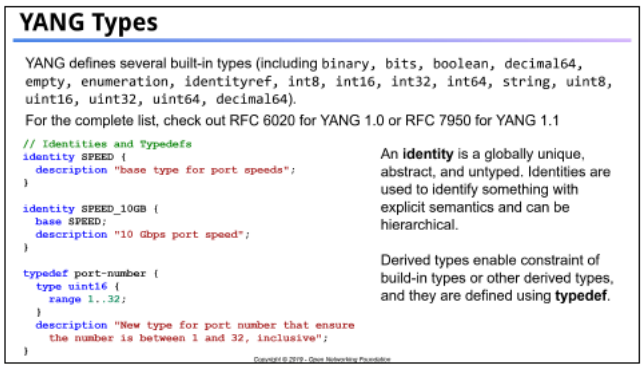
IETF: Son modelos que pueden usarse por cualquier fabricante, algunos de los estándares son YANG (modelo y lenguaje), NETCONF y RESTCONF (protocolos).

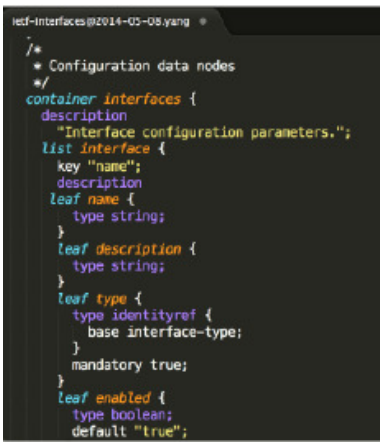
La separación en distintas partes del protocolo, la codificación y el modelo permite a los sistemas ser más flexibles.

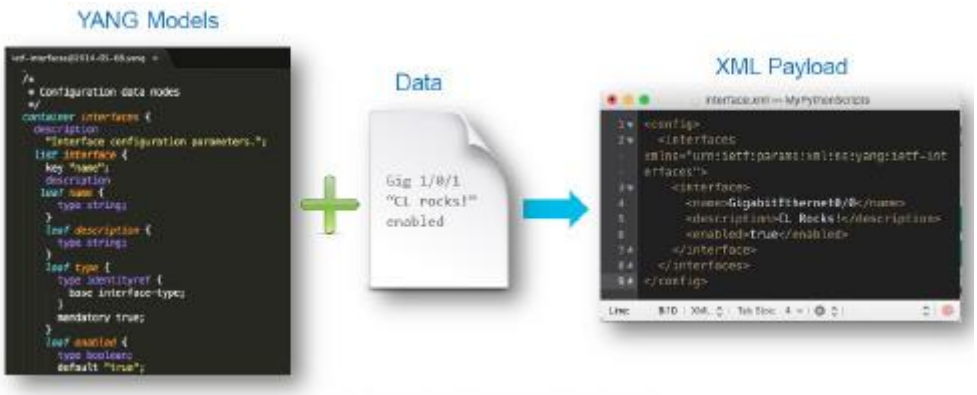
YANG

YANG, se trata de un modelo de datos optimizado para dispositivos de red, que va a ser empleado para la configuración de los dispositivos presentes en una red (una red la cual se quiere automatizar) , donde los datos son gestionados por los protocolos de NETCONG y RESTCONF y codificados en XML o JSON. (se usan contenedores agrupados en nodos, y listas para identificar los nodos).

YDK es la API donde se encuentra definida toda la semántica del modelo de datos de YANG. Es decir que, si un programador sigue esta API, va a implementar un modelo de YANG, sin conocer los detalles de los protocolos y la codificación usada en este modelo.







3. SDN

Las redes definidas por software permiten separar el plano de datos y el plano de control que constituyen una red, lo que permite realizar todos los cambios de la red desde un controlador (forma centralizada), sin tener que ir configurando uno por uno los dispositivos que constituyen la red. SDN emplea el protocolo de OpenFlow, para realizar cambios en los dispositivos que constituyen la red, lo que aumenta la seguridad. SDN reduce los costes de las redes, además de proporcionarlas flexibilidad.

SDN centraliza el control de la red al separar el plano de control, del plano de datos.

Principios

PRINCIPIO DE SDN 1

Control programable. SDN usa API's en lugar de protocolos. Realizar este cambio permite ocultar detalles de la red, mejorando así la seguridad. Por otro lado, también permite el ocultar detalles relacionados con el servicio no importantes para las aplicaciones, facilitando así su lectura.

PRINCIPIO DE SDN 2

El reenvío de los paquetes se trata como un problema computacional que sigue los siguientes pasos:

- Recibir un paquete
- Observar sus campos
- Aplicar algoritmos de clasificación o decisión
- Editar paquete (Opcional)
- Reenviar paquete.

PRINCIPIO DE SDN 3

Los paquetes se manejan únicamente en función del flujo al que pertenecen. Los flujos pueden estar determinados por aspectos como:

- Prefijo IP
- Etiqueta MPLS
- Una VLAN

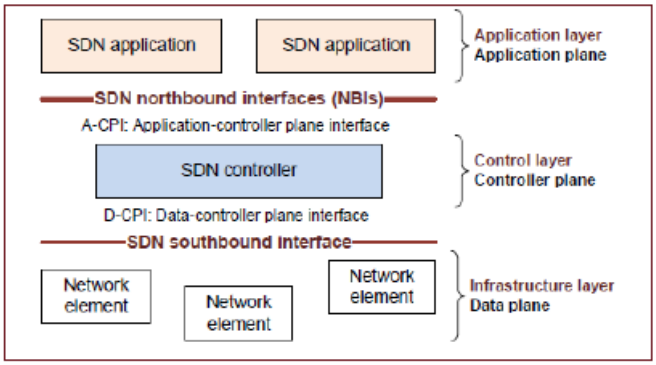
PRINCIPIO DE SDN 4

Eliminar protocolos distribuidos, es decir, que se reemplacen los protocolos de enrutamiento distribuidos con algoritmos en el controlador.

PRINCIPIO DE SDN 5

La configuración de los flujos se hace en los NE (conmutadores SDN) por el controlador SDN, por lo que los NE serán más simples.

Arquitectura



LA CAPA DE APLICACIÓN donde se encuentran las aplicaciones SDN, las cuales controlan un conjunto de recursos presentes en los controladores SDN.

LA CAPA DE CONTROL: el plano de control se encuentra presente en un conjunto de controladores, los cuales controlan un conjunto de recursos específicos del plano de datos. Un controlador SDN debe de ejecutar de manera obligatoria las solicitudes de las aplicaciones, pudiendo comunicarse con otros controladores.

LA CAPA DEL PLANO DE DATOS: compuesto por un conjunto de elementos de red, los cuales tienen un conjunto de procesamiento y reenvió de tráfico.

4. SOUTHBOUND INTERFACES (SBIS)

OPENFLOW

Este protocolo permite que el controlador les comunique a los dispositivos de red, presentes en la capa de datos, qué hacer con los paquetes que les llegan, además de programar las tablas de flujos de estos.

EL PLANO DE CONTROL:

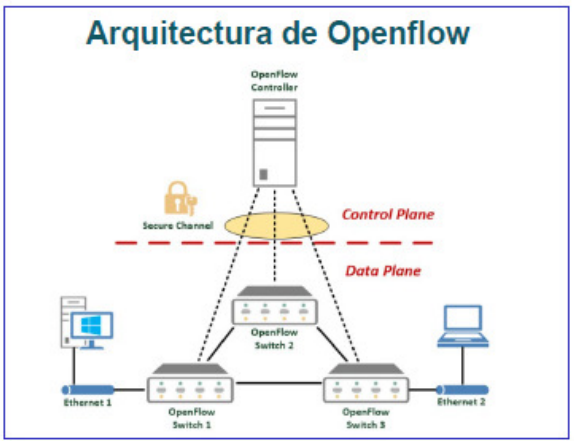
Con este protocolo, el controlador trabaja en modo centralizado, mientras que el plano de datos contiene en sus dispositivos los clientes de OpenFlow, quienes se comunican con el controlador a través de SSL (protocolo seguro).

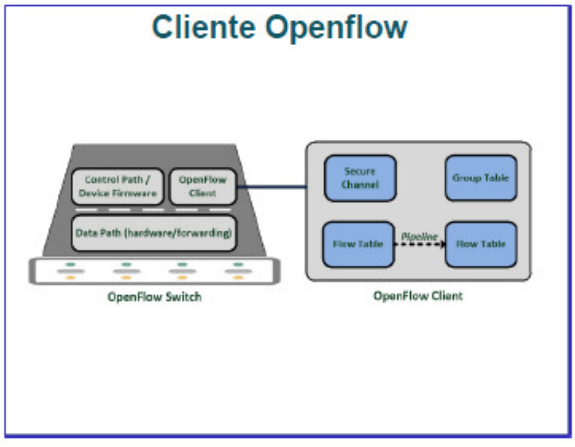
PLANO DE DATOS:

El plano de datos está compuesto por varias tablas de flujos, donde los paquetes de entrada son procesados.

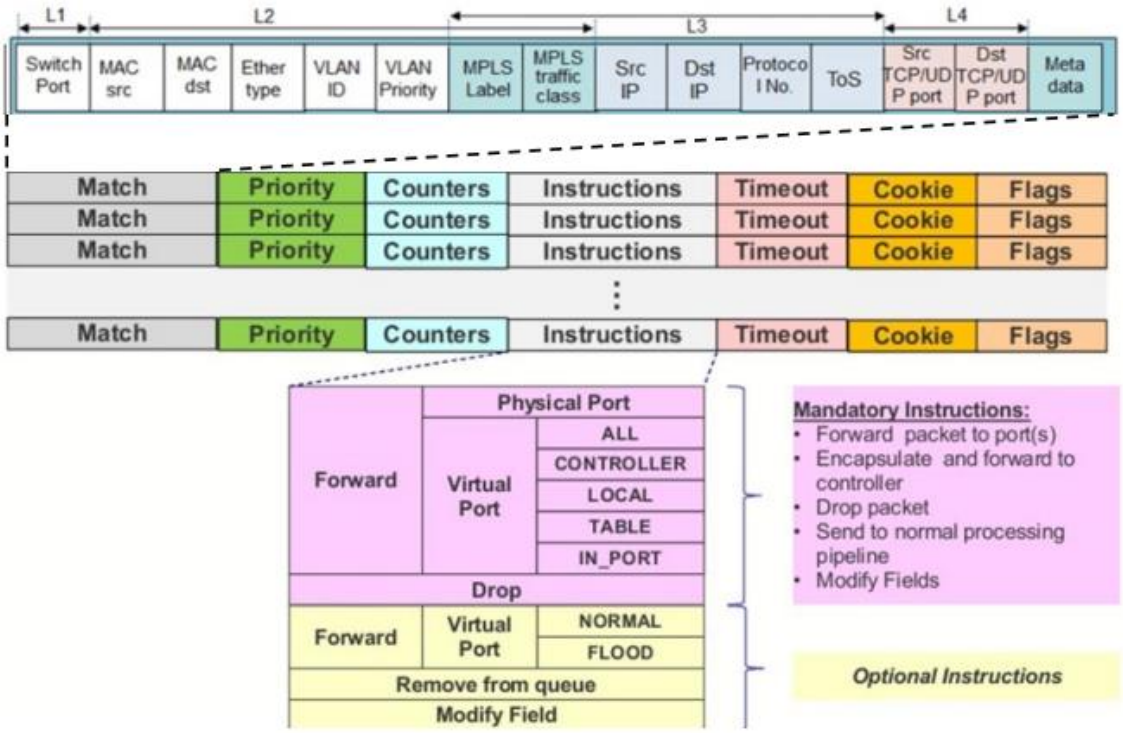
Una tabla de flujos está formada por varios flujos de entrada, donde cada uno de estos flujos contiene un campo matcher para realizar el macheo de los paquetes entrantes, es decir, la comprobación de ciertos campos para la elección de acción a realizar con los paquetes.

Además, presenta un campo de “timeout”, donde se establece un tiempo máximo antes de descartar un flujo, un campo de prioridad, para que un paquete pase antes por las entradas de mayor prioridad, un campo de cookie, el campo contador que va ir aumentando si los paquetes machean y otro de flags.



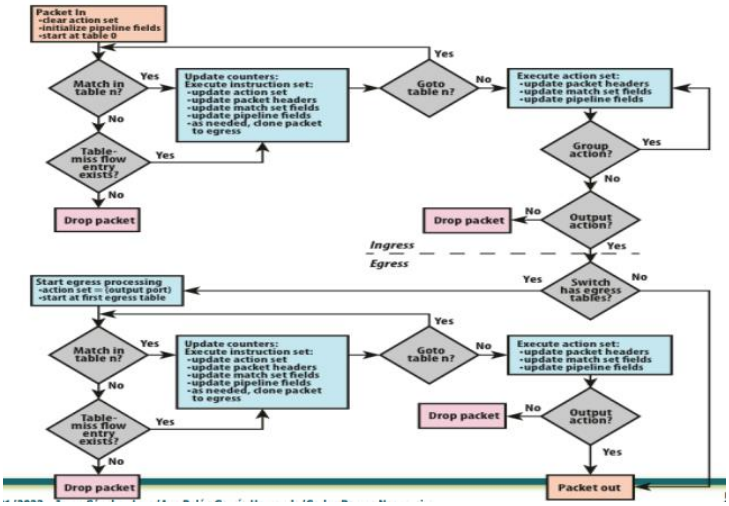


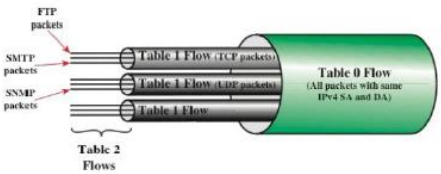
Las tablas de flujo siguen el siguiente aspecto:



Los paquetes entrarán siguiendo la estructura: paquete + puerto de entrada + valor de metadatos asociado + acciones asociadas. Para la tabla 0, el valor de los metadatos es 0 y las acciones son nulas. En el caso de que los paquetes no macheen con ninguna entrada, serán descartados pasando a la “table-miss”. Aquí reciben una última oportunidad de macheo, si no, serán descartados de forma directa. En caso de que en esta tabla se encuentre el primer paquete de un flujo, este se mandará al controlador para preguntar que hacer con él, ya que no ha macheado anteriormente.

En caso de que se maché con varias entradas, tendremos en cuenta la prioridad.





MENSAJES OPENFLOW

En el protocolo podemos encontrar 3 tipos:

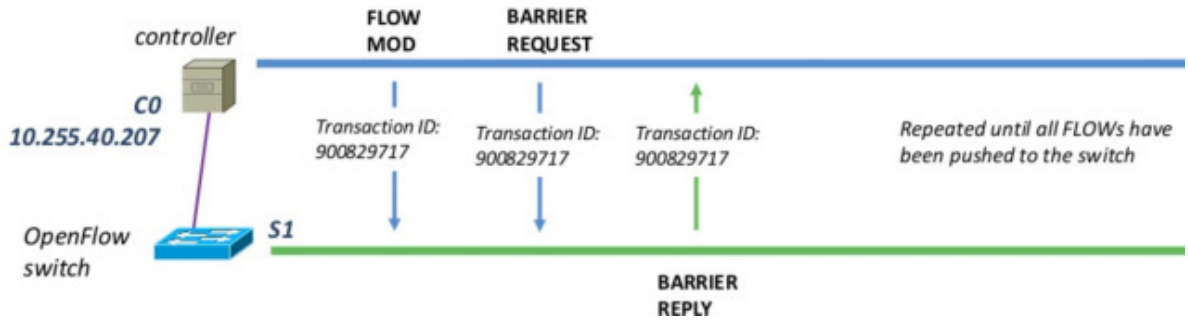
Del controlador al switch, sirven para preguntar las capacidades implementadas para crear tablas, modificar flujos… Ejemplos de estos mensajes son el features, configuration, packet out, barrier…

Mensajes asíncronos, son mensajes enviados del switch al controlador que informan de que el flujo ha sido terminado debido al TTL o para informar de errores. Ejemplos de estos mensajes son packet in, error, port status…

Mensajes simétricos, los cuales pueden ser enviados en ambas direcciones, son mensajes como el hello, echo o experimentales/vendedor, los cuales sirven para proveer de un futuro camino para extensiones de la tecnología OpenFlow.

Los Mensajes Barrier podemos definirlos como:

- Request, utilizados por el controlador para comprobar que el flujo se ha modificado correctamente
- Replay, para procesar la modificación de flujo y contestar al controlador.
- El valor de *transaction ID* se usa para identificar las operaciones de “Flow Mod”, “Barrier Request” y “Barrier Replay”

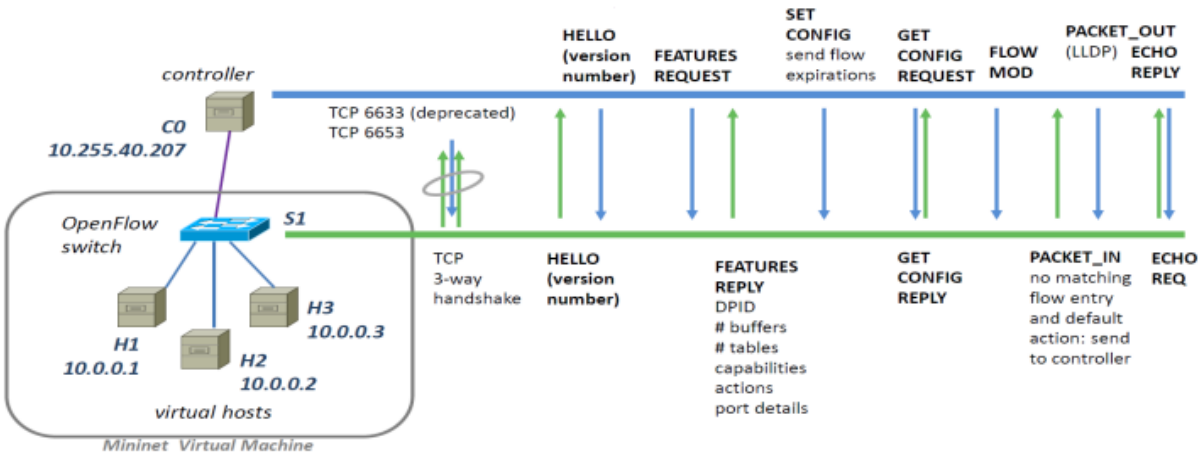


SECURE CHANNEL (SC)

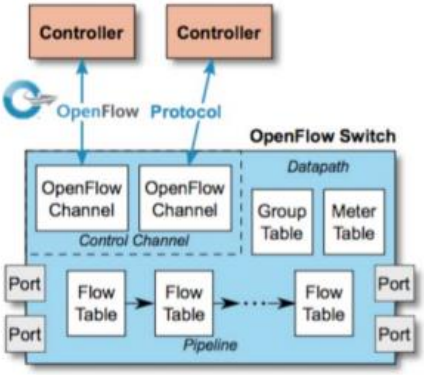
El SC se trata de la interfaz que conecta el switch que alberga el cliente de OpenFlow, con el controlador. Además, el SC establece y finaliza las conexiones de OpenFlow, entre el switch y el controlador, siendo estas unas conexiones TLS (Switch y Controlador se autentican mutuamente mediante el intercambio de verificados de clave privada).

Un switch puede estar conectado a través de un SC a un único controlador, o a través de varios SC a varios controladores distintos para así compartir la gestión de conmutación de dicho switch entre varios controladores.

La inicialización de los Switches se hace de la siguiente forma:



MULTICONTROLADOR



El Switch OpenFlow puede establecer comunicación con varios controladores de forma simultánea, lo cual mejora la confiabilidad del sistema, ya que si uno dejase de funcionar el otro lo respaldaría. El cambio de contexto de un controlador a otro y el traspaso entre ellos está completamente gestionado por los propios controladores.

Open vSwitch (OVS) y vSwitchDatabase (OVSDB)

Un OVS, es un software que se comporta como un Switch virtual, encargado de reenviar el tráfico de máquinas virtuales albergadas en la misma maquina física, incluso reenviando el tráfico de las maquinas virtuales a la red física.

Además, permite la implementación en varias máquinas físicas separadas a través de los hipervisores.

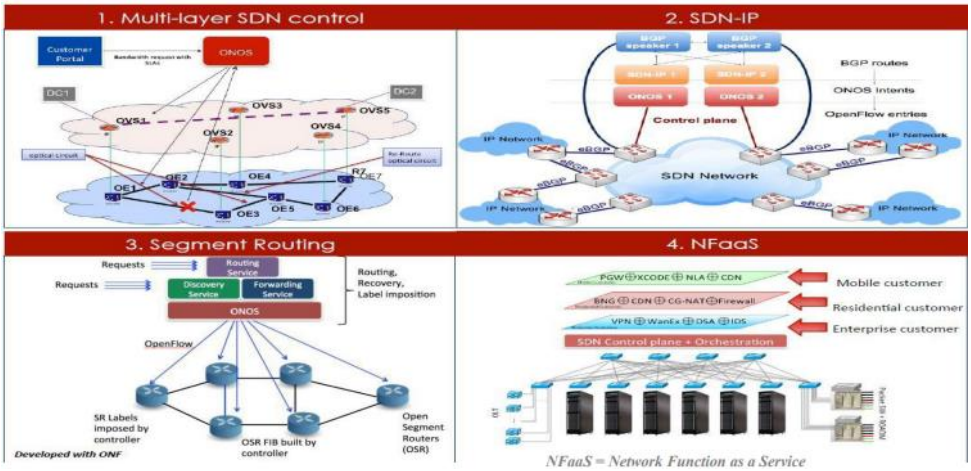
Existe un protocolo, OVSDB, encargado de la configuración de estos Switches virtuales, definiendo el numero de “virtual bridges”, creando, configurando y borrando puertos y túneles.

5. ONOS

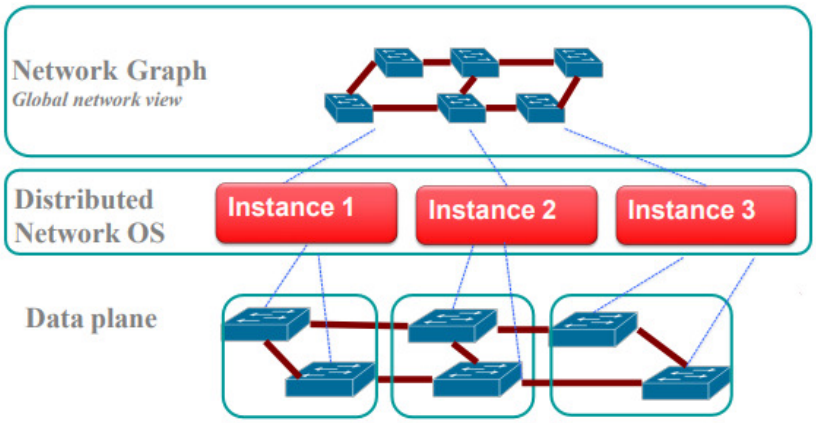
ONOS implementa la tecnología SDN, por lo que las aplicaciones van a mandar ordenes a la capa de control para modificar la configuración de la red, mientras que la capa de control mandará ordenes a los dispositivos de red que forman la capa de datos, configurándolos a través de OpenFlow.

Servicios primarios de ONOS

ONOS almacena el número de dispositivos de red, tiene un inventario de enlaces de la infraestructura, tienen otro inventario de los hosts conectados a los switches, además, sabe dónde se distribuyen geográficamente sus elementos y encuentra las rutas y los dispositivos que desea a través de OpenFlow. Por último, almacena información de los flujos establecidos y permite a la aplicación procesar los paquetes que llegan y definir por donde salen, así como la creación de los flujos.



Escalado de ONOS

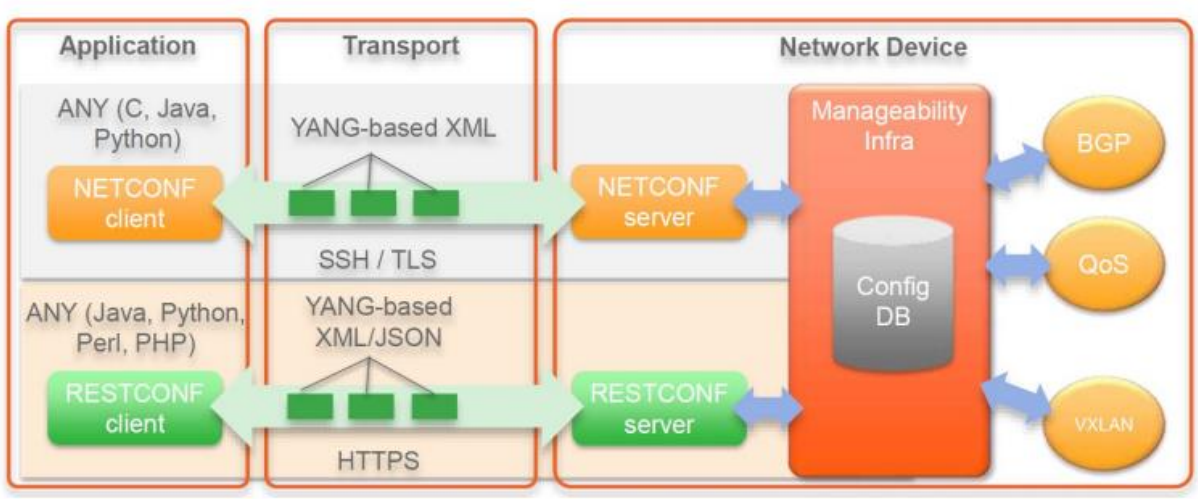


En las redes como al de la figura, cada instancia es responsable de mantener su parte de la red. En ONOS, los Switches suelen tener conexiones a más de un controlador, funcionando uno de ellos como primario y el resto como secundario.

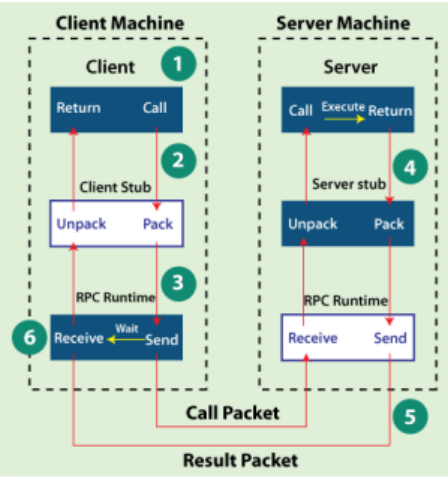
6. PROTOCOLOS GESTION REDES (NETCONF Y RESTCONF)

Se trata de protocolos que se van a encargar de la configuración de los dispositivos que conforman una red.
Antiguamente se empleaba el protocolo de NETCONF, pero actualmente se encuentra más implementado el protocolo de RESTCONF.

La única diferencia entre estos protocolos es que en la capa de transporte el protocolo de NESTCONF emplea SSH/TLS, mientras que el protocolo de RESTCONF emplea HTTPS, para la conexión (segura).



NETCONF



Proporciona mecanismos para modificar, eliminar e instalar configuraciones en los dispositivos de una red. Además, es capaz de transportar los datos de configuración y realizar operaciones de solicitud/respuesta empleando RPC.

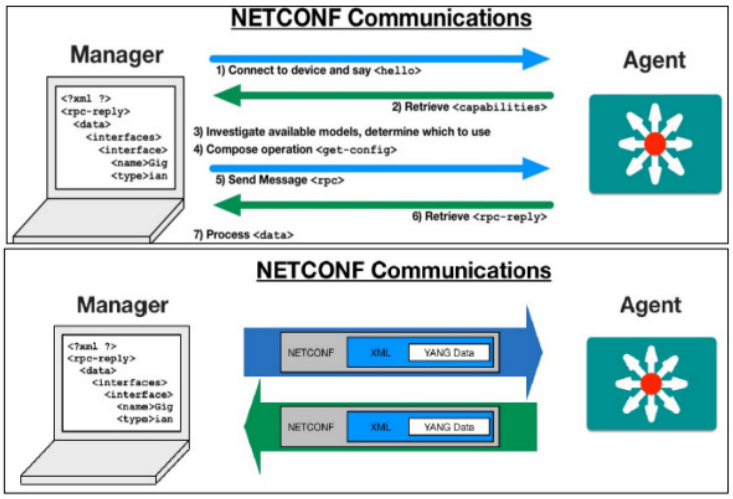
Los datos se codifican en XML, los cuales son transportados a través de la conexión SSH.

RPC es un mecanismo cliente-servidor que permite que un cliente que reside en un equipo realice una llamada a un procedimiento en otra máquina para que la otra lo ejecute y le envíe los resultados de la ejecución al cliente que ha hecho la llamada del procedimiento.

El proceso es: El cliente llama a un procedimiento local (“stub routine”) que empaqueta los argumentos del procedimiento en un mensaje (“marshalling”), el cual es enviado a través de la red al servidor. A continuación, el stub del cliente se bloquea. Mientras tanto el servidor descomprime al mensaje, llama al procedimiento y lo ejecuta, empaquetando los resultados de la ejecución, devolviendo este mensaje al stub del cliente, quien se desbloquea cuando le llega el mensaje, descomprime el mensaje que el servidor le ha enviado y se lo envía a la aplicación, la cual esta alojada en el cliente que realizo la llamada de procedimiento.

INTERACCIÓN DE RPC

Como bien sabemos el protocolo de NETCONF sirve para realizar la configuración de los dispositivos de red. Para poder ejecutar estas configuraciones en ellos se necesita de RPC.



El proceso de comunicación con NETCONF sería el siguiente: El administrador de red desde una aplicación (presente en la capa de aplicación) va a establecer una conexión desde el controlador (presente en la capa de control) con un dispositivo de red (presente en la capa de datos) enviando un mensaje de <hello>. Cuando el dispositivo de red recibe este mensaje responde mediante un mensaje de <capabilities> para informar al administrador de red de las capacidades que presente dicho dispositivo. Una vez que el administrado posee esta información, busca posibles configuraciones, las cuales puedan ser soportadas por el dispositivo y programará la configuración adecuada <get-config>. A continuación, el administrado va a ordenarle al controlador (cliente) que a través de RPC le envíe al dispositivo (servidor) la configuración que tiene que ejecutarse en él. Cuando este mensaje le llega al dispositivo, ejecuta la configuración la cual se le ha ordenado y los resultados de dicha ejecución, se los envía al controlador, para que este sepa si se ha ejecutado la configuración definida para el dispositivo de manera correcta.

CONFIGURATION DATASTORES

En el protocolo de NETCONF, en los configurations datastores, es donde se almacena la información de la configuración requerida, para que un dispositivo pase de su estado inicial, a un estado operativo. Contiene los datos definidos por YANG. Encontramos los siguientes tipos:

- <startup> configuration datastore: representa la configuración guardada para el próximo inicio de dispositivo, es decir si se hace un “copy-config” de la configuración activa del dispositivo se hará una copia para que en el siguiente inicio el dispositivo presente la misma configuración.
- -<running> configuration datastore: representa la configuración activa de un dispositivo.
- -<candidate> configuration datastore: representa la configuración que se puede llegar a activar <running> a través de un “commit”.

RESTCONF

Se trata de un protocolo similar al protocolo de NETCONF, pero se permite a las aplicaciones el acceso a los datos de configuración de los dispositivos.

RESTCONF emplea conexiones HTTPS, para así poder acceder a los datos de configuración de los dispositivos (codificados en XML o JSON) definidos en el modelo de YANG a través de la operación CRUD (operaciones propias de HTTP: GET, POST, DELETE, POST, PUT, PATCH…), que se encuentran almacenados en los "configuration datastores" del protocolo NETCONF.

Las operaciones principales de CRUD son:

- DELETE, POST, PUT y PATCH: son empleados para modificar los datos de configuración almacenados.
- GET: para recuperar los datos de configuración y de estado almacenados.
- POST: invocar las operaciones RPC.

RESTCONF	NETCONF
POST	create
PUT	replace
PATCH	merge
PATCH	any edit operation
DELETE	delete
POST	any <rpc> operation
GET	<get>, <get-config>
GET	<create-subscription>

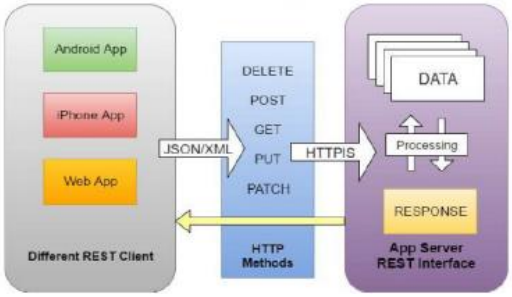
RESTCONF vs NETCONF

RESTCONF	NETCONF
Especifica las operaciones usando métodos HTTP: **GET, POST, PUT, PATCH, DELETE**	Especifica las operaciones encerrando el contenido del mensaje en un "XML tag": **<get-config>, <edit-config>**
Mensajes RPC codificados en XML o JSON	Mensajes RPC **siempre** codificados en XML
Utiliza la ruta del recurso en la URL de la solicitud para especificar la ubicación de los datos.	Utiliza varios niveles de profundidad en un mensaje XML para especificar la ubicación de los datos.
Tendencia de las nuevos desarrollos de equipos de red	Se debe soportar para integrar equipamiento existente

7. SDN NORTHBOUND INTERFACES: REST API

Esta API, permite a las aplicaciones interactuar con la red, para así poder obtener datos de ella o ejecutar funciones.

Utiliza el protocolo HTTP (con los métodos, GET, POST, PUT, PATCH, DELETE) para ejecutar operaciones en el servidor), es un protocolo sin estado, cliente-servidor, cacheable (indica si la información puede ser almacenada) y presenta un sistema de capas (mejora la abstracción, ya que el cliente no necesita saber si está comunicándose con un servidor concreto o elementos intermedios). Además, sigue una arquitectura REST para diseñar aplicaciones en red.



La ventaja de REST es que es completamente independiente de la plataforma y del lenguaje de programación donde se implemente, algo que ayuda claramente a la abstracción. Funciona en presencia de Firewalls además de tener seguridad basada en acceso user/password y encriptación de HTTPS.

8. VIRTUALIZACIÓN

La virtualización de recursos es una técnica que permite ocultar las características físicas de la plataforma informática en la que estamos trabajando. La virtualización utiliza el software, para crea una capa de abstracción sobre el hardware, de tal manera que permite dividir los elementos del hardware (memoria, procesador…), en varias máquinas virtuales, consiguiendo así una rentabilidad en el coste del hardware.

Cada máquina virtual presenta su propio sistema operativo y actúa como un ordenador independiente.

Una máquina virtual se puede cambiar fácilmente de una maquina física a otra.

Entorno de virtualización

- Hipervisores: es la capa de software que coordina las máquinas virtuales. Se trata de una interfaz entre la máquina virtual y el hardware físico. Existen dos tipos de hipervisores:

  -virtualización completa (1): sustituyen por completo el sistema operativo tradicional, por lo que interactúa con los recursos físicos.

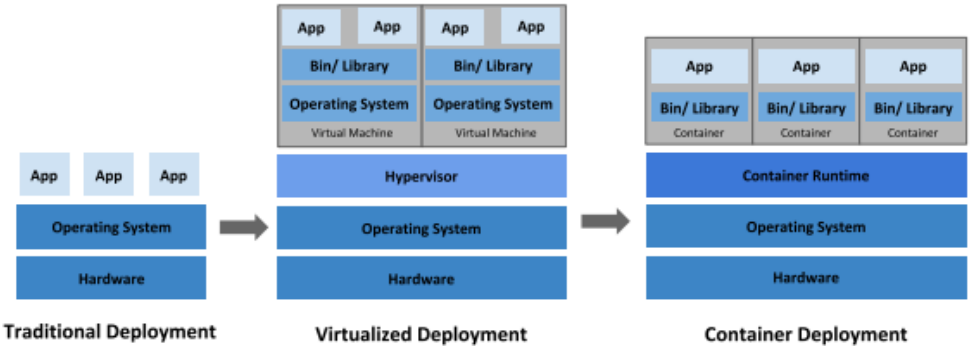
  -virtualización del sistema operativo (2): se ejecuta como una aplicación en un sistema operativo.

- Software de gestión: ayuda a configurar y mantener la infraestructura.
- Software de implementación: ayudar a implementar e integrar la aplicación en la nube.
- Red: permite conectar servicios en la nube a través de internet.
- Servidor: ayuda a calcular el uso compartido de recursos, presentes en la nube.
- Almacenamiento: la nube emplea un sistema de archivos distribuidos redundante para el almacenamiento de los datos.

Containers

Los contenedores son una forma de aislar un grupo de procesos o aplicaciones del resto del SO, de esta forma solucionamos el problema de cómo hacer que el software se ejecute de forma correcta al moverlo desde un sistema informático a otro.

NOTA: Recordar ejemplo de guille, en la asignatura de Seguridad tienes 9 MV, pero con 20 alumnos es un número increíble de máquinas para un servidor, por tanto, realmente tienes solo 1 MV, pero con 9 Containers o Jaulas, permitiendo así ahorrar recursos.

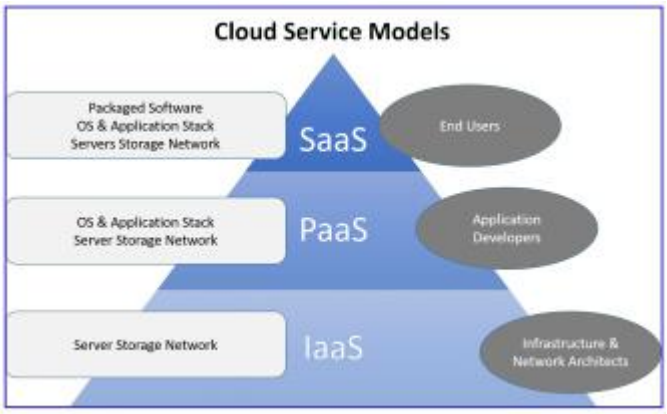


Cloud Computing

Una Cloud Computing proporciona computación, almacenamiento y red.

Existen tres modelos de servicios en Cloud:

- Cloud SaaS: es una nube empleada para distribuir las aplicaciones ya creadas a los usuarios a través de internet (Gmail)
- Cloud PaaS: proporciona un entorno de desarrollo listo para usar, para que los desarrolladores puedan crear aplicaciones personalizadas (Flynn)
- Cloud IaaS: proporciona el hardware virtualizado (VMware)



NFV

Consiste en la virtualización de los servicios de la red (firewalls, servidores…). Estos servicios virtualizados se empaquetan como máquinas virtuales, que se ejecutan en el hardware de la maquina física.

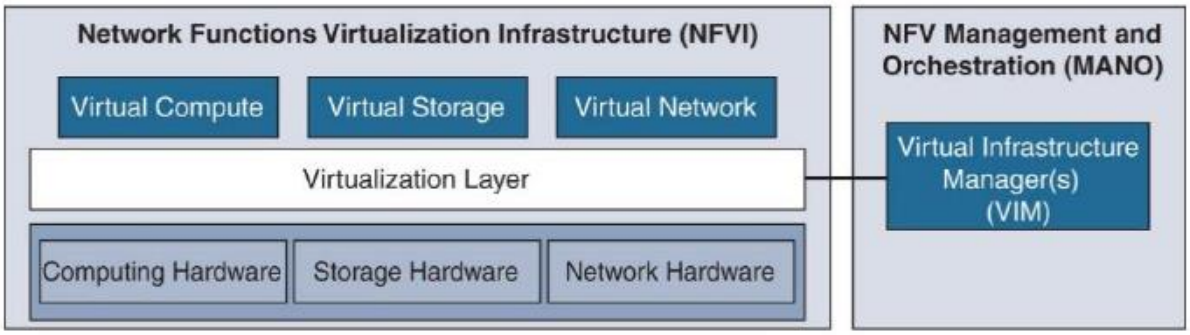
Con NFV se consigue que un mismo hardware físico, abarque varias funciones de la red.

Con la utilización de hardware COTS, se reduce el CAPEX, pero pasando a contabilizarse como OPEX. Se reduce el consumo de la energía, las funcionalidades se implementan en software y posibilita la localización de las funciones de red "network funtions", en cualquier lugar de la red.

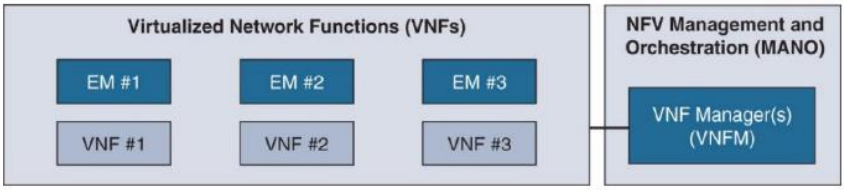
ARQUITECTURA DE REFERENCIA ETSI

Este punto teóricamente es complejo y suele entrar, por lo que miraremos directamente como funciona desde las respuestas de exámenes anteriores:

- VNF: son las aplicaciones de software que ofrecen las funciones de red (configuración de IP…) que se ejecutan en el hardware físico.
- NFVi: consiste en los elementos de la infraestructura, formado por los recursos físicos del hardware la capa de virtualización y los recursos virtuales.
- MANO: es el marco para gestionar la infraestructura de NFVi. Dentro se encuentran la VIM, que se encarga de controlar y administrar los recursos de NFVi.



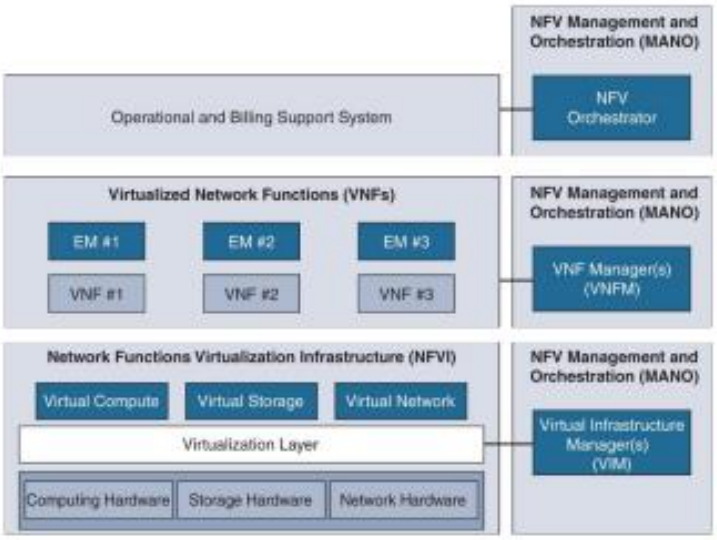
- CAPA VNF: capa donde se implementa la virtualización de las funciones de red. Formada por bloques EM y VNF. Se puede ejecutar en cualquier hardware con suficientes interfaces de red.
- VNFM: gestiona el ciclo de vida de las instancias VNF. Coordina y adapta los eventos producidos entre NVFi y EM.



- NFVO: gestiona el ciclo de vida de los servicios de la red, las políticas para instanciar los "network services" y la validación y la autorización de las solicitudes de recursos NFVI.

Preguntas de examen:

b) En la siguiente figura se representan gráficamente las capas de la arquitectura de referencia NFV, incluyendo sus principales componentes. Describa la funcionalidad de cada una de las capas y sus componentes mostrados. (0,5 puntos)



- NFV Infrastructure: Proporciona una virtualización del HW a las funciones virtuales de red. Comprende los recursos físicos de hardware, la capa de virtualización y los recursos virtuales.
- Virtualized Network Functions (VNF): Es la capa donde se implementa la virtualización de las funciones de red, haciendo uso de la virtualización del HW proporcionada por la NFVI. El bloque VNF se define como una combinación de bloques VNF y Element Management (EM).
- OSS/BSS: Sistema de operación y facturación del operador.
- Management and Orchestration (MANO): contiene el orquestador que se encarga de la gestión de los servicios de red (que pueden involucrar varias VNF cada uno), así como la gestión del ciclo de vida de las VNF y la administración de los recursos de computación, almacenamiento y red de NFVI.

a) ¿En qué consiste el "Service Chaining" en NFV? Explique cómo se relaciona este concepto con los de "Network Service" y "Virtual Network Function". (0,5 puntos)

Service Chaining es una funcionalidad que permite crear una cadena ordenada de VNF (Virtual Network Function) a aplicar al tráfico, gracias al establecimiento automático de conexiones entre dichas VNF con SDN, para componer servicios de red (Network Services) sofisticados entre extremos de la red (end points).

La principal ventaja de "Network Service Chaining" es automatizar la forma en que se pueden configurar las conexiones de red virtuales, para definir los flujos de tráfico de los servicios conectados. SDN y NFV permiten que el tráfico se encamine de manera inteligente a través de la red para que los paquetes pasen a través de las VNF requeridas, y solo esos dispositivos, por cada cliente.

SD_WAN

b) Defina y compare los siguientes conceptos de SD-WAN: "Underlay" vs "Overlay". ¿Cómo se relacionan con los túneles cifrados que se establecen entre los nodos "edge"? (0,5 puntos)

Mediante SD-WAN se consigue virtualizar la red física (denominada "Underlay") que interconecta los extremos de la red (los nodos "edge") mediante el establecimiento de túneles cifrados que conforman una red virtual denominada "Overlay". De esta manera se permite la implementación de soluciones versátiles sobre la red así virtualizada, abstrayéndose de las múltiples tecnologías que pueden estar presentes en el "Underlay" (MPLS, redes celulares, Internet, ...).

SD-WAN se trata de una aplicación especifica de tecnología de red, definida por software, aplicada a conexiones WAN, como internet. (4G…). Optimiza la dinámica de rutas múltiples. Conecta redes empresariales, incluidas sucursales y centros de datos, en grandes distancias geográficas.