Redes de Datos 2

2º Parcial 2023

Solución

Esta es una posible solución a las preguntas planteadas. Por razones didácticas normalmente contiene bastante más información que la mínima necesaria para responder la pregunta.

Pregunta 1 (6 puntos)

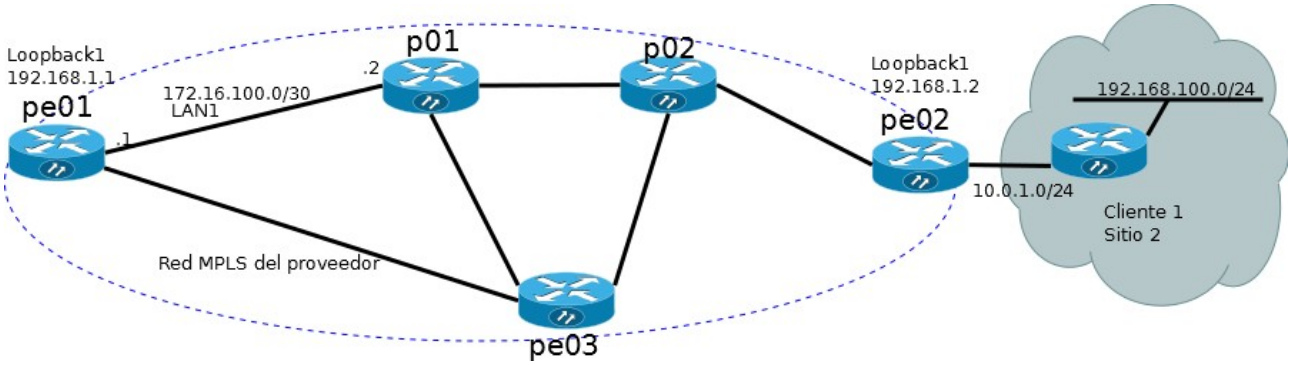
a) Explique cuáles son las FEC (clases de equivalencia de encaminamiento) en el protocolo LDP cuando este se utiliza para distribuir etiquetas asociado a un protocolo interno.
Cuando se asocia a un protocolo interno LDP asigna y distribuye etiquetas para cada una de las entradas de la tabla de rutas correspondientes a protocolos internos (incluyendo entradas correspondientes a rutas estáticas y directamente conectadas)
Entonces cada una de las FEC en este caso corresponde con todos los paquetes IP cuya dirección de destino coincida con una entrada de la tabla de rutas (y no con otra entrada más específica)

b) Explique qué mensajes LDP se esperan ver entre dos vecinos LDP, tanto para el descubrimiento de vecinos como para el intercambio de las relaciones entre las FEC y las etiquetas. No olvide explicar para qué sirven los mensajes, no importa el nombre del mensaje sino su función.
Para el descubrimiento de vecinos LDP envía periódicamente mensajes de tipo Hello, utilizando UDP y con destino la IP de multicast correspondiente a todos los enrutadores (224.0.0.2 en IPv4, ff02::2 en IPv6). De esta manera los enrutadores que reciben los mensajes Hello descubren a los enrutadores vecinos que están ejecutando LDP. Reciben además opciones y la dirección (transport address) con la cual establecer la sesión TCP para el intercambio de datos del protocolo
Para el intercambio de información se establece una conexión TCP entre cada par de vecinos, sobre la cual veremos distintos tipos de mensajes:
- *Mensajes de inicialización (Initialization). Se utilizan para intercambiar parámetros de la sesión*
- *Mensajes de keepalive. Para detección de fallos en la conexión*
- *Mensajes de tipo Address. Para enviar al vecino la lista de todas las direcciones de las interfaces donde se pueden recibir paquetes con etiquetas MPLS*
- *Mensajes del tipo Label Mapping para informar las relaciones FEC-Etiqueta que el enrutador asignó (y label withdraw para informar que una asociación dejó de ser válida)*
- *Mensajes de notificación para el manejo de errores*
- *En caso de trabajar en modo "on demand" hay otros mensajes para el manejo de relaciones FEC-Etiqueta*

Pregunta 2 (6 puntos)

a) Para las VPNs capa 3 vistas en el curso, ¿Qué función cumple el atributo "route-target" (comunidad extendida de tipo route-target)? ¿Cómo se utiliza para la exportación, propagación e importación de rutas entre VRFs?
En las VPN capa 3 (RFC 4364) se utilizan atributos comunidad extendida de BGP de tipo Route Target, para indicar las VRFs en las cuales se debe importar cada prefijo.
Recordemos que para la propagación de la información de rutas de las VRFs entre enrutadores PE se utiliza BGP multiprotocolo. En la RFC 4364 se decidió utilizar el atributo comunidades extendidas (extended Community) de BGP para propagar junto con el prefijo la información de las VRFs, en la forma de uno o mas valores de route-target para indicar en qué tablas (VRFs) debe importarse el prefijo
En cada VRF en cada PE se indican los valores de route-target para la exportación e importación de rutas.
Cada ruta que se exporta desde la VRF se "marca" con los valores de route-target indicados para exportación. Estos valores son propagados por BGP, y en el PE que recibe estos anuncios son utilizados para definir en qué VRFs debe ingresarse la información recibida: se agregarán las rutas en las VRFs que importen los valores de route-target recibidos.

b) En la red de la figura observamos parte de la red MPLS de un proveedor, y uno de los sitios del cliente 1, el cual está conectado al enrutador pe02. La sesión BGP entre los enrutadores pe01 y pe02 se establece entre las direcciones de loopback indicadas



En el enrutador pe01 se ejecutan los siguientes comandos, de los cuales se reproduce parcialmente la salida:

```
pe01# sh ip route
……..
C>* 172.16.100.0/30 is directly connected, eth0, 15:55:33
C>* 192.168.1.1/32 is directly connected, loopback0, 15:55:33
O>* 192.168.1.2/32 [110/40] via 172.16.100.2, eth0, label 21, 15:54:31
O>* 192.168.1.3/32 [110/30] via 172.16.100.2, eth0, label 19, 15:54:33
……..

pe01# sh ip route vrf cliente1
……...
VRF cliente1:
K>* 0.0.0.0/0 [0/8192] unreachable (ICMP unreachable), 00:06:39
C>* 10.0.0.0/24 is directly connected, eth1, 00:06:01
B>  10.0.1.0/24 [200/0] via 192.168.1.2(vrf default) (recursive), label 144, 00:01:49
B>  192.168.100.0/24 [200/0] via 192.168.1.2(vrf default) (recursive), label 144, 00:01:49
```

¿Qué valores de etiqueta MPLS espera ver en la red LAN1 para un paquete con destino 192.168.100.3 en la VPN cliente1? ¿qué función cumple cada una de las etiquetas?

Como vemos en el segundo comando (show ip route vrf cliente1), en pe01 los paquetes con dirección IP destino en la red 192.168.100.0/24 en la VRF cliente1 son encaminados a través del PE con IP 192.168.1.2 (pe02), encapsulado en MPLS con la etiqueta 144

A su vez, para llegar a 192.168.1.2 vemos que pe01 debe encaminar los paquetes con próximo salto la IP 172.16.100.2, y utilizando el valor de etiqueta MPLS 21

Entonces esperamos ver los paquetes indicados con un stack de dos etiquetas, la exterior con valor 21 y la interior con valor 144. La etiqueta exterior se utiliza para encaminar el paquete hacia el PE destino (pe02), mientras que la etiqueta interior (144) la utilizará el enrutador pe02 para identificar la VRF que corresponda (seguramente cliente1) y encaminar el paquete.

Pregunta 3 (3 puntos)

a) Para Enrutamiento por segmentos (Segment Routing) utilizando MPLS

- ¿Por qué se dice que se trata de una función de enrutamiento desde el origen?

En Segment Routing se dice que se está trabajando con enrutamiento desde el origen porque el enrutador de entrada a la red MPLS que está utilizando segment routing es el que define las instrucciones a ejecutar para el encaminamiento del paquete y las envía junto con el paquete

- ¿Cómo se representan las instrucciones para encaminar el paquete?

En Segment Routing utilizando MPLS las instrucciones se representan mediante un stack de etiquetas MPLS, donde el valor de cada etiqueta representa una instrucción (segmento) que debe recorrer el paquete

- ¿por qué es necesario reservar un bloque de etiquetas global para Segment Routing?

En Segment Routing hay segmentos que son de tipo global, es decir que representan la misma instrucción para todos los enrutadores (por ejemplo segmentos de tipo IGP Prefix). Como queremos poder distribuir la información de segmentos utilizando extensiones a los protocolos de enrutamiento, precisamos poder representarlos respecto a valor que signifique lo mismo para todos los enrutadores (para esto se utiliza el offset dentro del segmento global de etiquetas)

Permite además evitar que haya colisiones entre los valores de etiqueta elegidos por Segment Routing y los valores elegidos por otros mecanismos (como LDP, RSVP, etc.)

Pregunta 4 (7 puntos)

a) ¿Para qué se utiliza OSPF en MPLS-TE?
¿Qué extensiones debieron realizarse sobre el OSPF para OSPF-TE?

*La premisa de Ingeniería de Tráfico es poder "controlar los flujos de datos en una red para cumplir con los **requerimientos** de los mismos y **optimizar la utilización de los recursos de la red**".*

Esto requiere conocer la topología de la red, el nivel de utilización de los enlaces y la capacidad disponible de los enlaces. Dado que OSPF de forma nativa ofrece la topología de la red, se optó por extender el protocolo de forma de cumplir el resto de los requerimientos.

De forma breve, OSPF-TE permite conocer la topología y el nivel de utilización de la red.
Esto es la base para poder realizar el mapeo de la demanda a la capacidad de la red.

El detalle de las extensiones agregadas a OSPF para OSPF-TE:

- *Información del Link (Link Type, Link ID, Local Interface IP)*
- *Traffic Engineering Metric (Weigth, puede ser diferente al peso de OSPF básico).*
- *Maximum Link Bandwith*
- *Maximum Reservable Bandwith*
- *Unreserved Bandwith (per priority level)*
- *Attribute Flags (atributos de links a considerar en las restricciones)*

b) ¿Cómo se utilizan los parámetros de priority y holding priority al momento de crear un túnel en MPLS-TE? Explique el concepto de preemption.

*Al momento de crear el túnel, el control de admisión compara el valor de prioridad **X** que solicita el túnel y el ancho de banda **B** con la capacidad disponible **D** en el link para dicha prioridad **X**.*
*La versión simple es que si **B** > D, no hay capacidad y por ende no se puede considerar dicho link para el túnel. La funcionalidad de preemption permite que los túneles de menor prioridad (> X) ya establecidos puedan ser expulsados para permitir crear el túnel de mayor prioridad.*

Ejemplo Link:

Priority	Allocated	Reservable
0	100000	400000
1	50000	350000
2	200000	150000
3	0	150000
4	0	150000
5	50000	100000
6	50000	50000
7	25000	25000

Si consideramos la sumatoria de reservas, hay reservados 475 Mbps, pero es posible crear un túnel de prioridad 5 y 50 Mbps expulsando a los túneles activos de prioridad menor. En este caso se pueden expulsar los túneles de prioridad 6 y 7, pero alcanza con expulsar (preemption) los túneles de prioridad 7.

El valor de holding-priority es el que se utiliza para, una vez creado un túnel, comparar con la prioridad de los nuevos túneles que se quieran establecer para definir si lo pueden expulsar. Continuando con el ejemplo, el túnel creado tiene prioridad 5, holding-priority 3, y ancho de banda 50 Mbps.

Priority	Allocated	Reservable
0	100000	400000
1	50000	350000
2	200000	150000
3	50000	100000
4	0	100000

5	50000	50000
6	50000	0
7	0	0

Luego de creado el túnel, ahora se observa que hay reservados 50 Mbps del túnel solicitado en prioridad 3, y ya no se observa la reserva de 25 Mbps en prioridad 7.

c) Asumiendo que estamos utilizando OSPF-TE. ¿Qué dificultades presentan los escenarios de múltiples áreas en OSPF-TE?

Para poder crear un túnel de ingeniería de tráfico es necesario conocer la topología y el nivel de reserva de los links. Cuando OSPF-TE trabaja con varias áreas, un router solo puede conocer el detalle de la topología del área a la que pertenece. Si el router origen y destino del túnel se encuentran en diferentes áreas, no es posible el calculo de camino.

Por este motivo se creo el concepto de nodos loose en la especificación de restricciones de camino, un nodo loose es un nodo por el cual debe de pasar el camino explícito del túnel. La forma de resolver el inconveniente de crear un túnel de Ingeniería de tráfico cuando el origen y el destino están en diferentes áreas es utilizando a los routers de borde de área origen (y área cero) y los routers de borde de área destino (y área cero) como nodos loose en la especificación de restricciones del camino.

d) Existen varias formas de especificar un camino (restricciones de camino) para realizar el túnel de ingeniería de tráfico.

Una de ellas es de forma explícita especificando la totalidad del camino (“strict”), otra es indicando nodos “loose” (nodo por el cual deseo pasar, pero sin especificar el camino de forma explícita).
¿Quién y cómo resuelve la definición del camino entre el origen y destino cuando se específica el camino mediante nodos “loose”?

Cuando se especifica un camino utilizando nodos loose, se conoce el head-end, y luego la lista de nodos loose ordenados en orden de aparición (por ejemplo 1. router de borde de área origen y 2. router de borde de área destino).

El head-end busca un camino explícito, que cumpla las restricciones de camino (ancho de banda y afinidad), hacia el primer nodo loose. Una vez hallado, construye un mensaje RSVP path con el objeto Explicit_Route “camino explícito hasta el primer nodo loose, lista restante de nodo loose”. Cuando el mensaje RSVP path llega al primer nodo loose, este busca un camino explícito desde el hasta el siguiente nodo loose, el mensaje RSVP path contiene un objeto Explicit_Route “camino explicito entre el primer nodo loose hasta el segundo nodo loose, lista restante de nodos loose”. El último nodo loose, busca un camino explícito hasta el tail-end, una vez hallado el mensaje RSVP path se propaga con un objeto de Explicit_Route “camino explicito desde el último nodo loose hasta el tail-end”.

En resumen, la construcción del camino explícito se realiza de forma parcial comenzando en el head-end hasta el primer nodo loose, el siguiente tramo del camino “se delega” al nodo loose, y así sucesivamente

e) Explique las diferencia entre las “restricciones de camino” de un túnel y el camino explícito del túnel.

Las restricciones de camino son una declaración de los requerimientos del camino, que pueden ser de forma genérica, puede no existir restricciones de camino, garantizar que pasen por un nodo o que no pase por un nodo, o que pase por un conjunto de nodos o incluso el camino explícito entre el origen y destino del túnel.

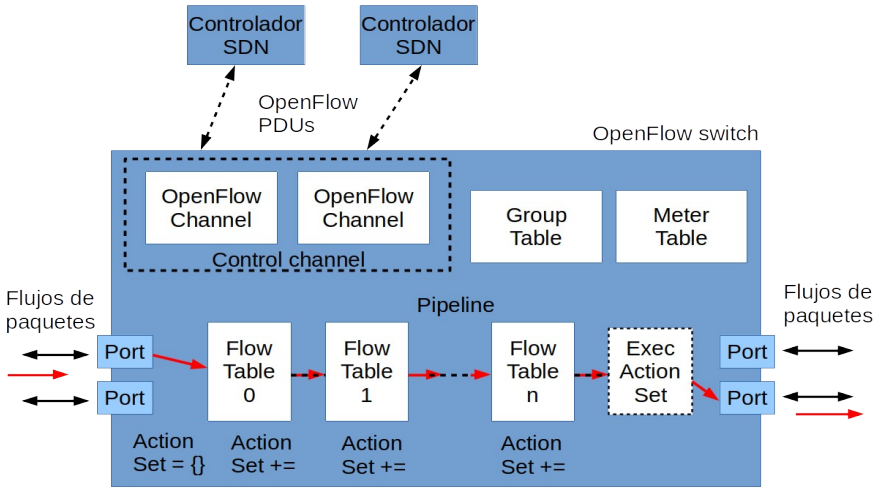
El camino explícito es la declaración de por que links se reservaran etiquetas para el LSP que implementa el túnel, por donde se propaga el mensaje RSVP para gestionar dichas etiquetas.
Para poder gestionar las etiquetas por RSVP debo tener el detalle del camino, por lo cual dada las restricciones de camino, el algoritmo CSPF (Constraint Shortest Path First) busca un camino explícito entre el origen y destino que cumpla la restricciones. Para esto completa la información faltante, por ejemplo, si no debo pasar por un nodo, “elimina” el nodo de la topología y busca los posibles caminos

entre el origen y destino, selecciona uno de ellos y este es el que considera como candidato y utiliza para completar el objeto de Explicit_Route del mensaje RSVP path.

Pregunta 5 (5 puntos)

a) Explique cómo es la arquitectura lógica de un switch OpenFlow, identificando sus principales componentes e indicando qué función cumplen.

Un switch SDN tiene una arquitectura lógica como se muestra en la siguiente figura:



Se pueden identificar los siguientes bloques:

- *Puertos de entrada y salida: para ingreso y salida de los paquetes*
- *Cadena o pipeline de tablas de flujo: Los paquetes pueden recorrer una o más tablas de flujo una vez que ingresan al switch y en cada una de ellas, dependiendo de los "matches" en las entradas de cada tabla y las instrucciones que indiquen las entradas, se establecerán un conjunto de acciones a aplicar al paquete. Si el paquete no es descartado o enviado al controlador durante el procesamiento, será finalmente enviado por alguno de los puertos de salida del switch.*
  En la cadena de tablas de flujo se pueden identificar el procesamiento de ingreso (ingress processing) con una serie de tablas que se pueden aplicar antes de definir la interfaz de salida y el procesamiento de egreso (egress processing) que puede también incluir varias tablas y se ejecuta en el contexto del puerto de salida.
- *Tablas de grupo: Los paquetes pueden ser enviados a una tabla de grupo donde se pueden aplicar acciones a un conjunto de flujos. Las tablas de grupo permiten implementar multicast, broadcast, failover, colas con pesos y también aplicar acciones masivas sobre muchos flujos.*
- *Tablas de medida: permiten establecer acciones relacionadas con medidas de performance para medir calidad de servicio.*
- *Canales de comunicación con el controlador. Este bloque implementa el protocolo openflow para la comunicación con el controlador. Es importante que haya redundancia en la comunicación con el controlador.*

b) ¿Por qué es necesario definir una arquitectura genérica de los switches?

Una de las ventajas de la arquitectura SDN es poder gestionar equipos de diferentes fabricantes y para eso es necesario poder identificar componentes dentro del switch de forma genérica. El controlador debe poder escribir y borrar entradas en la tabla X del switch y para eso es necesario que todos los switches tengan la misma arquitectura independientemente de cómo sea luego la implementación de cada uno internamente.

Pregunta 6 (6 puntos)

Bajo el modelo de QoS DiffServ:

a) ¿A qué se llaman y en qué se diferencian los nodos exteriores de los nodos interiores?

En el modelo Diffserv, la priorización del tráfico se hace por clase, a cada paquete IPv4 se le define (se le marca) una clase utilizando los DSCP bits del encabezado IPv4. De esta forma las estrategias de QoS se realizan solo considerando la marcas, sin diferenciar a que flujo pertenece.

***Nodos exteriores** se llama a los nodos que están mas cercanos el origen del tráfico, en la periferia de "la red que entiende de marcas DSCP o puede aplicar priorización de tráficos". Son los nodos donde el tráfico pueden llegar sin marcar los DSCP bits, y se encarga de crear las marcas.*
Además de poder marcar el tráfico, pueden confirmar las marcas que reciben (el origen podría generar ya tráfico con marcas), remarcar el tráfico o realizar "shapping o policing" del tráfico de acuerdo al contrato de servicio (conformar el tráfico).

En las interfaz de salida cada marca de DSCP se asigna a diferentes colas de servicio, luego un scheduler (o despachador) elige una estrategia para poder garantizar QoS a las diferentes clases de tráfico.

***Nodos interiores** se llama a los nodos donde el tráfico llega con marcas DSCP, por lo cual el nodo lo único que tiene que hacer es detectar la marca, y luego en la interfaz de salida, asigna a la cola que corresponde a dicha marca, luego el scheduler (o despachador) elije una estrategia para poder garantizar QoS a las diferentes clases de tráfico.*

La mayor diferencia entre ambos tipos de nodos radica en que los nodos exteriores tienen que revisar al detalle el tráfico entrante, de forma de marcar/verificar/remarcar los bits DSCP, así como controlar que el tráfico sea conforme de acuerdo al contrato de tráfico. Los nodos interiores, solo revisan las marcas.

b) ¿Quién marca el tráfico y de acuerdo a qué criterio lo hace?

Como se comentó en la parte a) las marcas las definen los nodos externos, aunque siendo estrictos pueden definirse en el origen del tráfico (un servidor) y confiar en las marcas, o si el tráfico no cumple el contrato de tráfico, se modificarán las marcas.

Para marcar el tráfico se puede utilizar criterios simples o criterios complejos; uno simple es todo lo que entra por una interfaz le asigno una marca, por dirección IP de origen o destino, por protocolo de transporte, por puerto de origen o destino, etc. Los criterios complejos permiten combinar estos criterios simples.

c)
i) Para el tráfico IPv4 se utilizan las marcas en los bits DSCP. Detalle cómo se adapta el concepto de marcas a otros protocolos vistos en el curso.

El éxito de la estrategia de Diffserv de trabajar con un conjunto de clases y definir QoS por clase hace natural que se realicen equivalencias en otros protocolos. Los protocolos vistos en el curso son IPv6 y MPLS.

En MPLS tenemos los experimental bits, que permiten generar hasta 8 clases distintas. Si bien esto es restrictivo en comparación con los 6 bits DSCP de IPv4, en la práctica no se suelen utilizar los 64 posibles valores. Los equipos deben de definir la estrategia de dado el DSCP de IPv4 a que valor de exp bit de MPLS mapear, de forma que los nodos internos sigan simplemente revisando la marca y definiendo a que cola de prioridad dirige el tráfico. La estrategia debe ser consistente, en el sentido que el tráfico EF (Expedited Forwarding) mantenga las características de bajo retardo cuando se hace el mapeo a EXP bit = 5.

En IPv6 utilizaremos el campo Traffic Class de 8 bits del encabezado para llevar la marca de priorización o tratamiento del tráfico, luego los nodos internos deben revisar estas marcas, y asignar a la cola de prioridad correspondiente. Al igual que antes los tratamientos de colas deben ser consistentes, no hay que perder de vista que por el mismo link puedo tener tráfico IPv4, tráfico IPv6 y tráfico MPLS.

ii) ¿Qué precauciones se deben tener cuando el origen es tráfico IPv4 y la red de transporte es MPLS?

Cuando la red de transporte es MPLS y el tráfico nativo IPv4, debemos de traducir las marcas de un universo a otro, pero manteniendo la consistencia, es decir el trafico por defecto en IPv4 debe de traducirse al tráfico por defecto de MPLS (tráfico elástico sin priorizar), el tráfico de EF en IPv4 (asociado al voz sobre ip VoIP) que se traduzca a una marca de MPLS que luego lo dirija a una cola de atención de baja latencia.

d) A los fines de brindar o garantizar QoS, ¿alcanza solamente con marcar el tráfico? En caso negativo explicar que función o funciones adicionales se deben realizar.

No alcanza con marcar el tráfico si luego los nodos no realizan tratamiento diferenciado de acuerdo a las marcas. Para ello de acuerdo a la marca DSCP los paquetes se asignan a diferentes colas de servicio de paquetes, y luego un despachador es el que garantiza la prioridad entre las diferentes colas de servicio.

El punto crucial es que son dos funciones separadas, las marcas se generaron en los nodos externos, y los nodos internos deben de implementar las estrategias de priorización, que eventualmente podrían diferir nodo a nodo, ya que no existe un protocolo para sincronizar la misma estrategia entre los nodos.

Las funciones adicionales son básicas, revisar las marcas, asignar cola de paquetes de acuerdo a la marca y luego la estrategia de despacho de paquetes de las diferentes colas.

Un ejemplo simple son dos colas de servicio, una de prioridad estricta sobre la otra, los paquetes de EF (VoIP) los asignamos a la cola que llamamos prioritaria, luego la estrategia de despacho de paquetes es mientras haya paquetes en la cola de prioridad atenderlos primero, y solo cuando no haya paquetes en dicha cola, despachar los paquetes de la cola de menor prioridad.

Pregunta 7 (5 puntos)

a) Explique brevemente el nuevo paradigma de NFV

En pocas palabras NFV propone desacoplar las funciones de red tradicionales (Hardware y Software propietario), para que de esta manera las funciones puedan ejecutarse en servidores de propósito general y Entornos Virtualizados (ejemplo Virtual Machine o contenedores).

b) ¿Qué es una Virtual Network Function (VNF)?

Una VNF (Virtual Network Function) es una implementación de una Función de Red tradicional desplegada dentro de un entorno virtualizado NFVI (Network Function Virtualization Infraestructure) de acuerdo al paradigma NFV.

c) ¿Para qué se utilizan hipervisores y contenedores en NFV?

Los hipervisores permiten realizar una abstracción del HW, particionar los recursos y controlar el acceso a dichos recursos HW, para luego desplegar VM (máquinas virtuales) las cuales implementan los componentes software necesarios. Desde el punto de vista de una VM requiere un sistema operativo tradicional y es incapaz de distinguir si se esta ejecutando directamente sobre un componente hardware real o sobre componentes virtualizados por el hipervisor. La separación se hace en la separación de las llamadas del kernel al HW.

Los contenedores se basan en el concepto de particionar y limitar los recursos del sistema operativo, permiten "empaquetar" la aplicación y las componentes software del sistema operativo (librerías) requeridos. Los contenedores pueden tener un stack de TCP/IP diferente del sistema operativo host. La separación entre contenedores se realiza mediante funciones dentro del kernel.

Ambas estrategias son la base del concepto de virtualización que busca el paradigma NFV.

d) ¿Porqué se dice que los contenedores son nativos al ambiente de nube y no así las VM (máquinas virtuales)?

El surgimiento de las VM viene de la mano del concepto de consolidación, varias empresas de software imponían un modelo de una Aplicación para uso empresarial (servidor de correo, servidor Web, etc) un servidor. El resultado era que buena parte del tiempo los servidores tenían valores de utilización de CPU bajos. Las VM son el mismo sistema operativo y Aplicación que corría antes en un servidor físico, ahora sobre recursos virtuales. No hay ninguna adaptación en el diseño del software al entorno.

En caso de requerir más capacidad, debo iniciar una nueva VM, que es equivalente al proceso de booteo de todo el sistema operativo de dicha VM.

Los contenedores permiten desplegar varias instancias de una misma aplicación "contenerizada", ejecutarse con un stack TCP/IP diferente al del sistema operativo base, y distribuir la carga asociada a una aplicación entre estas instancias. Estas instancias pueden incrementarse de acuerdo a la necesidad (el tiempo que demora en levantar una aplicación en un servidor), y distribuir dichas instancias en diferentes servidores. Si bien no lo vimos en detalle en el curso, existen mecanismos nativos en el entorno de contenedores, que permite crear un overlay, permitiendo que se comuniquen las aplicaciones entre si.

Las aplicaciones para ser contenerizadas requieren ajustes o un diseño específico para el entorno de contenedores.

En resumen las aplicaciones en contenedores son diseñadas para ser distribuidas, con varias instancias de un mismo contenedor que pueden instanciarse en diferentes servidores físicos, sin importar en que equipo especifico se encuentra en contenedor, adaptarse a la demanda (crear o eliminar contenedores) o re-instanciar un contenedor (en caso de falla del servidor). Es por este motivo que se considera que son nativos en cloud (premisa de que no es importante en que servidor esta el contenedor y adecuarse a la demanda).

Como observación final, formalmente se requiere un orquestador de contenedores, por ejemplo, kubernetes, junto con otras herramientas que permiten distribuir la carga y realizar el overlay de networking, para poder decir que son cloud native.