

VLAN Trunking Protocol (VTP) e Inter VLAN Routing



Antes de Imprimir este documento
considere si es necesario
Ayudemos al Ambiente !!!!

Universidad San Carlos de Guatemala
—— **DANILO ESCOBAR** ——

(VTP) e Inter VLAN *Routing*

▶ VLAN Trunking Protocol

Es un protocolo propietario de Cisco que tiene por objetivo facilitar la administración y configuración de las VLANs dentro de una infraestructura, permitiendo que los cambios realizados en un dispositivo se propaguen automáticamente a todos los demás *switches* dentro de un mismo dominio.

Perteneciente a la capa 2 del modelo OSI, VTP previene inconsistencias al utilizar los enlaces troncales (requisito indispensable) para sincronizar información entre varios *switches*.

El seguimiento de los cambios se realiza gracias a un número de revisión, el cual se incrementa cada vez que se crea, elimina o se cambia de nombre una VLAN, entre más alto es este parámetro más reciente es considerada la información, por lo que todos los *switches* buscarán sincronizarse con la información que posea el número de revisión más alto.

Actualmente existen tres versiones de este protocolo, las primeras dos presentan ligeras diferencias en su funcionamiento, mientras que la versión 3 es una completa reestructuración del mismo.

Para configurar la versión a utilizar puede emplearse la siguiente instrucción.

```
Switch(config)# vtp version ?  
<1-3> Set the administrative domain VTP version number  
  
Switch(config)# vtp version 2
```

Además, VTP puede ser configurado para trabajar en una de las siguientes modalidades

- **Servidor (*Server*):** Este es el modo por defecto. Posibilita crear, eliminar y modificar VLANs y establecer parámetros (como la versión del protocolo) que serán utilizados en todos los *switches* dentro de un dominio, dentro del cual se recomienda la existencia de un solo dispositivo que cumpla con esta función.
- **Cliente (*Client*):** En este modo no es posible realizar ningún cambio en la configuración de las VLANs. Los dispositivos que tomen este rol heredarán aquella información proporcionada por el servidor. Sin embargo, si un *switch* en modo cliente es incorporado a la topología con un número de revisión mayor al de cualquier otro dispositivo, este enviará la información más reciente al servidor para luego ser propagada al resto del dominio.

(VTP) e Inter VLAN *Routing*

- Transparente (*Transparent*): Este modo no participa en el proceso de sincronización de VTP, los *switches* configurados para desempeñar este rol tendrán control sobre sus propias VLANs y se limitarán a reenviar las actualizaciones VTP recibidas a través de sus enlaces troncales para que estas puedan alcanzar otros dispositivos.
- Apagado (*Off*): Deshabilita VTP, lo *switches* tendrán control sobre sus propias VLANs y la información relacionada con VTP no será reenviada. Esta modalidad no está disponible en todas las plataformas.

Para configurar el modo VTP se puede emplear la siguiente instrucción.

```
Switch(config)# vtp mode ?  
client      Set the device to client mode.  
server      Set the device to server mode.  
transparent Set the device to transparent mode.
```

Otra característica de VTP es que en orden de funcionar correctamente este requiere de la capacidad de almacenar información relacionada con el estado de las VLANs y los cambios hechos en las mismas así como otros parámetros necesarios para su funcionamiento, de manera automática en un espacio de memoria no volátil (memoria que no necesita de energía para perdurar), funcionalidad presente en sistemas operativos antiguos pero no en el Cisco IOS.

Para superar este problema, se creó un archivo especial donde pudiera almacenarse dicha información de manera dinámica y se le dio el nombre de *vlan.dat*, usualmente referido como *VLAN Database*.

De esta manera en aquellos modos donde VTP es completamente funcional (Servidor y cliente) toda información respecto a su configuración y a las VLANs es almacenada en dicha base de datos y no será mostrada en el archivo de configuración siendo el caso contrario en las otras modalidades.

```
Switch(config)# vtp mode server  
Device mode already VTP SERVER.  
Switch# show running-config | include vlan  
Switch#
```

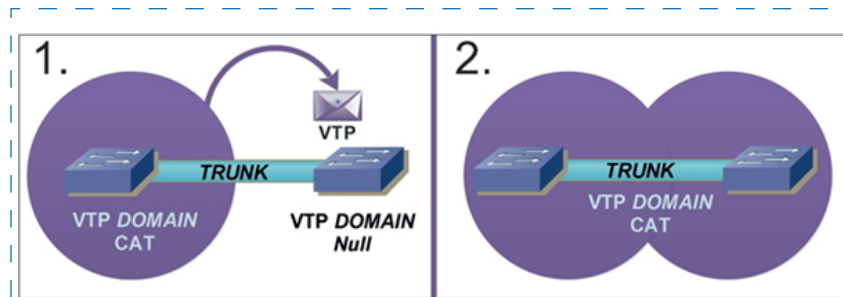
```
Switch(config)# vtp mode transparent  
Setting device to VTP TRANSPARENT mode.  
Switch# show running-config | include vlan  
vlan 10  
vlan 20  
vlan 30  
Switch(config)#
```

A pesar de haber establecido la versión y el modo a utilizar, VTP no comenzará a enviar publicaciones hasta que se haya especificado un dominio administrativo dentro del *switch* ya que de manera predeterminada estos no pertenecen a ninguno, estando este campo indefinido por lo que se dice que tiene un valor indeterminado (*Null*).

(VTP) e Inter VLAN Routing

Un dominio indeterminado (*Domain Null*) requiere especial atención, ya que además del comportamiento descrito anteriormente ocasionará que el dispositivo, al recibir una publicación VTP en uno de sus enlaces troncales, pase a formar parte del dominio incluido en la misma para luego heredar toda su información.

Este proceder, exclusivo de las versiones 1 y 2 de VTP, se incluyó en el diseño de este protocolo con el objetivo de facilitar la incorporación de nuevos *switches* a una infraestructura, aunque hoy en día el mismo no es conveniente desde el punto de vista de la seguridad en la red.



Al recibir una publicación VTP un switch con un dominio indeterminado pasará a formar parte del dominio indicado en la misma.

Para establecer un dominio, se utiliza el siguiente comando, nótese que un *switch* puede pertenecer solamente a un dominio administrativo.

```
Switch(config)# vtp domain CAT  
Changing VTP domain name from NULL to CAT
```

Otra medida que puede tomarse es la configuración de una contraseña para su uso dentro del dominio. Dicha contraseña no impedirá que un *switch* pase a formar parte de un dominio de manera automática, pero sí evitará la sincronización de la información entre dispositivos.

```
Switch(config)# vtp password FOX
```

Un protocolo que depende de la correcta configuración de VTP es el *Dynamic Trunking Protocol* (DTP).

DTP negociará un enlace troncal entre dos *switches* si ambos se encuentran en el mismo dominio o en el caso de que uno o los dos dispositivos tengan un dominio indeterminado. De otra forma, no se negociará dicho enlace debido a un error llamado *domain mismatch*.

(VTP) e Inter VLAN *Routing*

```
Sw1(config)# vtp domain CAT
Changing VTP domain name from NULL to CAT
```

```
Sw2(config)# vtp domain HAT
Changing VTP domain name from CAT to HAT
00:10:22 %DTP-5-DOMAINMISMATCH: Unable to perform trunk
negotiation on port Fa0/1 because of VTP domain mismatch.
```

La única manera de visualizar la configuración de VTP (almacenada en el vlan.dat) es utilizando el comando *show vtp status*, el cual se muestra a continuación, los demás comandos son para proporcionar contexto y completar el ejemplo.



```
Switch(config)#vtp mode server
Device mode already VTP SERVER.
Switch(config)# vtp password FOX
Setting device VLAN database password to FOX
Switch(config)# vtp domain CAT
Changing VTP domain name from NULL to CAT
Switch(config)# vtp version 2
```

```
Switch#
%SYS-5-CONFIG_I: Configured from console by console
```

```
Switch# show vtp status
VTP Version                : 2
Configuration Revision     : 0
Maximum VLANs supported locally : 255
Number of existing VLANs   : 5
VTP Operating Mode         : Server
VTP Domain Name            : CAT
VTP Pruning Mode           : Disabled
VTP V2 Mode                : Disabled
VTP Traps Generation       : Disabled
MD5 digest                 : 0xC5 0x1F 0xC8 0x2C 0x6F 0xF9 0x91 0x53
```

```
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
```

(VTP) e Inter VLAN *Routing*

Entre la información más importante puede apreciarse el número de revisión (*Configuration Revision*), el nombre del dominio y el modo que se está utilizando. También puede observarse la dirección IP del último dispositivo en actualizar la base de datos de las VLAN (*Configuration last modified by*) y la dirección IP del dispositivo mismo (*Local Updater ID*), suponiendo en ambos que los dispositivos cuentan con al menos una dirección asignada a través de alguna interfaz virtual. Para configurar manualmente la dirección a ser utilizada por VTP, se debe seleccionar la interfaz que posea la dirección deseada de la siguiente manera.

```
Switch(config)# vtp interface ?  
WORD The name of the interface providing the VTP updater ID for this device.
```

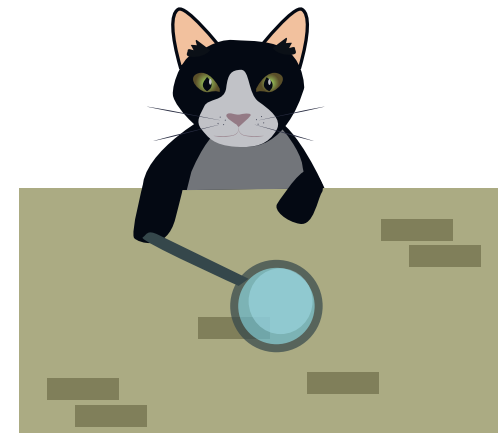
Un parámetro importante que no se muestra utilizando los métodos descritos anteriormente es la contraseña utilizada por VTP, misma que solo se puede visualizar utilizando el siguiente comando.

```
Switch# show vtp password  
VTP Password: FOX
```

A pesar de todos los beneficios explicados anteriormente VTP introduce también el riesgo de interrumpir todas las operaciones de la red, de manera intencional o accidental ya sea por atacantes o usuarios inexperimentados por lo que en la mayoría de las organizaciones no es utilizado.

Dichos riesgos han sido mitigados o completamente eliminados en la versión 3 de este protocolo, no obstante, dicha versión no está disponible para todas las plataformas.

Finalmente es necesario agregar que incluso si no se planea implementar este protocolo es necesario conocer su funcionamiento y comportamiento a manera de poder desactivarlo apropiadamente.



(VTP) e Inter VLAN Routing

► Inter VLAN Routing

La creación de las VLANs permite la segmentación de la red al posibilitar la existencia de varios dominios de *broadcast* dentro de un mismo *switch*.

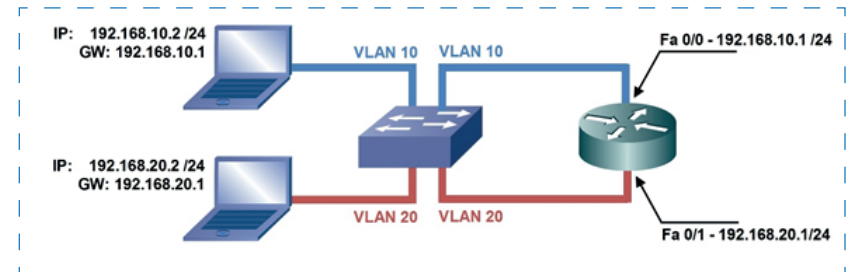
Estos dominios funcionan de manera independiente, sin comunicación alguna entre ellos a pesar de residir físicamente en el mismo dispositivo, por lo que cada uno de los mismos puede ser utilizado para albergar una red diferente.

Por esta razón, para volver a establecer comunicación entre varias VLANs (*Inter Vlan Routing*), se necesita de un dispositivo capaz de enrutar entre varias redes, pudiendo elegirse entre las siguientes opciones.



► Un router con una interfaz para cada VLAN

En este caso la solución es directa, por cada VLAN existe una interfaz separada en el *router* quien luego se encarga de establecer comunicación entre ellas. No obstante, esta solución no es económica ni escalable ya que se tiene que incorporar una nueva interfaz por cada nueva VLAN.



Router con una interfaz por cada VLAN

(VTP) e Inter VLAN Routing

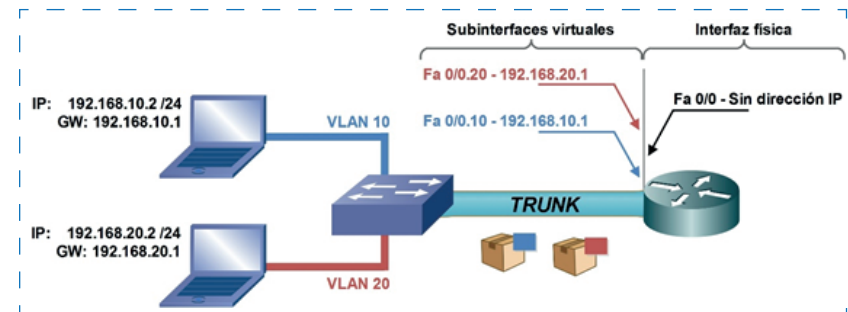
Router en un palo (Router on a stick)

En esta solución se utiliza un solo enlace troncal para llevar el tráfico de todas las VLANs a una interfaz física del *router* (de ahí su nombre), en donde se crearán varias subinterfaces virtuales, siendo cada una de ellas destinada a manejar los datos y servir como puerta de enlace predeterminada de una VLAN específica.

Para que dichas subinterfaces puedan separar y procesar correctamente el tráfico proveniente de cada VLAN es necesario configurar, dentro de cada una, el protocolo utilizado en el enlace troncal así como la etiqueta específica asociada con el tráfico que se pretende manejar.

Este arreglo es más económico y escalable que la solución presentada anteriormente, sin embargo introduce un único punto de falla, el enlace troncal, así como un cuello de botella dentro de la red ya que el tráfico de todas las VLANs debe pasar necesariamente por dicho enlace.

Para mostrar la implementación de esta solución se usará la siguiente topología, donde se ha configurado previamente el *switch* con las VLANs indicadas y se han configurado los puertos en los modos necesarios.



Router en un palo (Router on a Stick)

Antes de comenzar, es importante comprender que dentro de cada red existen dos tipos diferentes de topologías: la física y la lógica. La topología física es la que nos muestra la disposición de los dispositivos, las interconexiones entre ellos y los cables utilizados para las mismas. Mientras que la topología lógica está compuesta por aquellas construcciones invisibles, formadas por dispositivos e interfaces virtuales, las cuales afectan las rutas que atraviesa la información de un punto a otro de la red y que deben configurarse de la misma manera que sus contrapartes reales.

De esta manera al analizar la topología anterior, se tiene que físicamente el tráfico de todas las VLANs llega a la interfaz *FastEthernet* 0/0 del *router* mostrado a través de un enlace troncal. No obstante, desde un punto de vista lógico el enlace troncal no existe y cada VLAN está conectada a este dispositivo a través de su propia interfaz.

(VTP) e Inter VLAN *Routing*

Así pues, físicamente la interfaz *FastEthernet* 0/0 debe estar encendida para recibir la información proveniente del enlace troncal, sin embargo, a nivel lógico dicha interfaz no recibe ningún tipo de tráfico, por lo que no necesita de ninguna otra configuración, ni siquiera de una dirección IP.

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#no shutdown
Router(config-if)#exit
```

```
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
```

Esta dualidad es un poco difícil de comprender en un inicio. Si bien es cierto que el flujo de la información está determinado por la composición lógica de la red no puede olvidarse que esta depende del correcto funcionamiento de la parte física en todo momento.

Siguiendo esta línea de razonamiento se crearán, sobre la interfaz física *FastEthernet* 0/0, subinterfases virtuales para manejar el tráfico de las VLANs presentadas, como se muestra a continuación.

```
Router(config)#interface fastethernet 0/0.?
<0-4294967295> FastEthernet interface number

Router(config)#interface fastethernet 0/0.10
```

Al agregar un punto (".") después del nombre de la interfaz puede accederse a la configuración de las subinterfases virtuales. El rango de posibles valores (0-4294967295) tiene como propósito proporcionar flexibilidad a la hora de elegir un valor y no indica la cantidad de subinterfases que este *router* puede manejar.

Sea cual sea el valor elegido para designar una subinterfaz este no tendrá ninguna injerencia sobre su funcionamiento. Sin embargo, se recomienda elegir un nombre relacionado con el propósito de la misma, lo que más adelante facilitará su manejo y la resolución de problemas dentro de la red.

Para utilizar la subinterfaz recién creada para procesar el tráfico proveniente de una VLAN, debe especificarse el protocolo utilizado por el enlace troncal para encapsular la información (*Encapsulation*), así como el valor de la etiqueta respectiva.

En este caso, dicho protocolo será el estándar abierto 802.1Q o "punto 1Q" (dot1q) y el tráfico que se quiere manejar es el de la VLAN 10.

```
Router(config-subif)# encapsulation dot1q 10
```

(VTP) e Inter VLAN *Routing*

Una vez configurada tanto la encapsulación como la etiqueta del tráfico a procesar se vuelve posible especificar una dirección IP para la subinterfaz de la misma manera como se haría con una interfaz real

```
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)# exit
```

Ahora esta subinterfaz puede ser utilizada como puerta de enlace predeterminada de todos los dispositivos dentro de la VLAN 10.

Es posible seguir el mismo proceso para crear la subinterfaz encargada de manejar el tráfico de la VLAN 20.

```
Router(config)# interface fastethernet 0/0.20
Router(config-subif)# encapsulation dot1q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
```

Para mostrar el estado de todas las interfaces, físicas y virtuales, se utiliza nuevamente el comando *show ip interface brief*.

```
Router# show ip interface brief
```

Interface	P-Address	K?	Method	Status	Protocol
FastEthernet0/0	unassigned	ES	unset	ip	up
FastEthernet0/0.10	192.168.10.1	ES	manual	ip	up
FastEthernet0/0.20	192.168.20.1	ES	manual	ip	up
FastEthernet0/1	1 unassigned	ES	unset	administratively	down
Vlan1	unassigned	ES	unset	down	down
				administratively	down

Finalmente, al revisar la tabla de enrutamiento se puede apreciar que el *router* tiene conocimiento de las redes utilizadas para cada una de las VLANs ya que estas se encuentran conectadas directamente a través de las subinterfaces creadas anteriormente.

Todo tráfico entre VLANs debe pasar necesariamente a través del *router*, el cual redirigirá el tráfico y modificará las etiquetas para establecer conectividad.



(VTP) e Inter VLAN *Routing*

```
Router# show ip route
```

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

```
C    192.168.10.0/24 is directly connected, FastEthernet0/0.10
C    192.168.20.0/24 is directly connected, FastEthernet0/0.20
```

► Switch multicapa (*Multilayer switch*)

Es un dispositivo que además de proveer las funciones típicas de un *switch* presta otros servicios que operan en otras capas del modelo OSI (de ahí su nombre).

Una de las capacidades del *switch* multicapa es la de enrutar paquetes a gran velocidad utilizando hardware dedicado, misma que se encuentra deshabilitada por defecto pero que puede activarse utilizando el siguiente comando

```
MLSwitch(config)# ip routing
```

De esta manera los puertos de este dispositivo pueden funcionar tanto a nivel de la capa 2, como los de un *switch* tradicional o a nivel de la capa 3, en cuyo caso podrá asignársele una dirección IP y utilizarla junto con algún protocolo de enrutamiento.



(VTP) e Inter VLAN Routing

La funcionalidad por defecto depende de la plataforma, pero es posible alterarlo utilizando el comando *switchport* (Capa 2) y la negación del mismo no *switchport* (Capa 3), como se muestra a continuación.

```
MLSwitch(config)# interface fastEthernet 0/1
MLSwitch(config-if)# switchport
MLSwitch(config-if)# switchport mode access
MLSwitch(config-if)# switchport access vlan 10
```

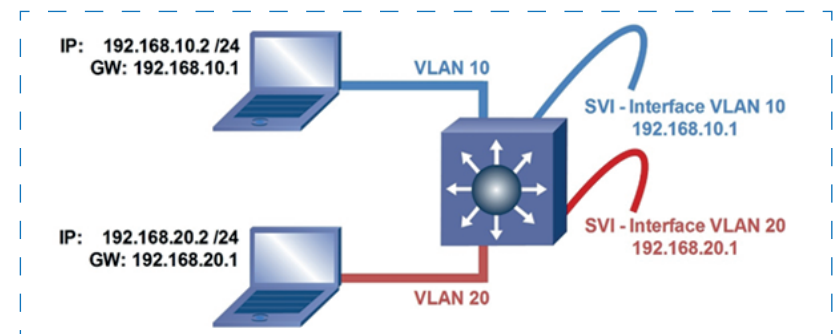
```
MLSwitch(config-if)# interface fastEthernet 0/2
MLSwitch(config-if)# no switchport
MLSwitch(config-if)# ip address 172.16.1.1 255.255.255.0
```

Otra posibilidad que ofrece este switch, es la creación de múltiples *Switched Virtual Interfaces* (SVIs), interfaces virtuales íntimamente ligadas a las VLANs y que pueden ser usadas por estas como puertas de enlace predeterminadas. Para crear una SVI, puede utilizarse el siguiente comando.

```
MLSwitch(config)# interface vlan 100
MLSwitch(config-if)# no shutdown
```

Para que una SVI sea operacional, la VLAN relacionada debe existir y estar activa dentro del switch, es decir, que debe haber por lo menos un puerto activo perteneciente a dicha VLAN y/o un enlace troncal en donde dicha VLAN sea permitida y que no haya sido bloqueada ya sea manualmente o por algún protocolo (VTP, Spanning Tree, etc.,).

Para mostrar la implementación de las SVIs se muestra la siguiente topología, donde las VLANs y los puertos han sido previamente configurados. Se muestra nuevamente cómo se activan las capacidades de enrutamiento para que el ejemplo este completo.



Creación y configuración de SVIs.

(VTP) e Inter VLAN *Routing*

```
MLSwitch(config)# ip routing

MLSwitch(config)# interface vlan 10
MLSwitch(config-if)# no shutdown
MLSwitch(config-if)# ip address 192.168.10.1 255.255.255.0

MLSwitch(config)# interface vlan 20
MLSwitch(config-if)# no shutdown
MLSwitch(config-if)# ip address 192.168.20.1 255.255.255.0
```

Al examinar la tabla de enrutamiento..

```
MLSwitch# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

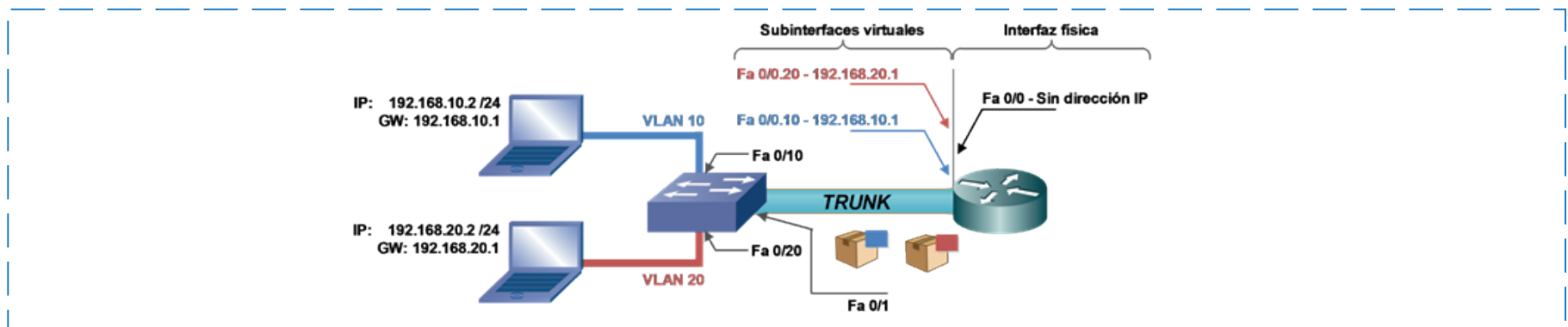
C      192.168.10.0/24 is directly connected, Vlan10
C      192.168.20.0/24 is directly connected, Vlan20
```

Al utilizar un *switch* multicapa para comunicar varias VLANs, se eliminan cuellos de botella y retrasos introducidos por otros dispositivos, por lo que desde el punto de vista del mejor diseño de la red, es la solución predilecta. Estando su utilización limitada en el pasado debido a su costo, hoy en día su uso se hace cada vez más frecuente, siendo implementados en segmentos de la red antes reservados para los *switches* tradicionales tendencia que irá en aumento conforme su precio se vuelva más asequible.

Finalmente, es necesario añadir que a pesar de ejecutar muchas funciones que en el pasado eran exclusivas a los *routers*, no es posible reemplazar estos últimos por *switches* multicapa en todos los casos. Antes de tomar una decisión respecto a cuál de los dos dispositivos debe utilizarse, es necesario tener completo entendimiento de los requerimientos que el diseño debe satisfacer, poniendo especial atención a características especiales como conexiones WAN, calidad de servicio, seguridad, entre otras.

(VTP) e Inter VLAN Routing

Resumen de la configuración *Router on a Stick*



Switch

```
SW(config)# vlan 10
SW(config-vlan)# name Tecnicos

SW(config)# vlan 20
SW(config-vlan)# name Secretarias

SW(config)# interface fastEthernet 0/1
SW(config-if)# switchport mode trunk
SW(config-if)# switchport nonegotiate
SW(config-if)# switchport trunk allowed vlan 10,20

SW(config)# interface fastEthernet 0/10
SW(config-if)# switchport mode access
SW(config-if)# switchport access vlan 10

SW(config)# interface fastEthernet 0/20
SW(config-if)# switchport mode access
SW(config-if)# switchport access vlan 20
```

Router

```
Router(config)# interface fastEthernet 0/0
Router(config-if)# no shutdown
Router(config-if)# exit

Router(config)# interface fastEthernet 0/0.10
Router(config-subif)# encapsulation dot1q 10
Router(config-subif)# ip address 192.168.10.1 255.255.255.0

Router(config)# interface fastEthernet 0/0.20
Router(config-subif)# encapsulation dot1q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
```

(VTP) e Inter VLAN *Routing*

Resumen de la configuración *Switch* multicapa

Switch Multicapa

```
MLSwitch(config)# vlan 10  
MLSwitch(config-vlan)# name Tecnicos
```

```
MLSwitch(config)# vlan 20  
MLSwitch(config-vlan)# name Secretarias
```

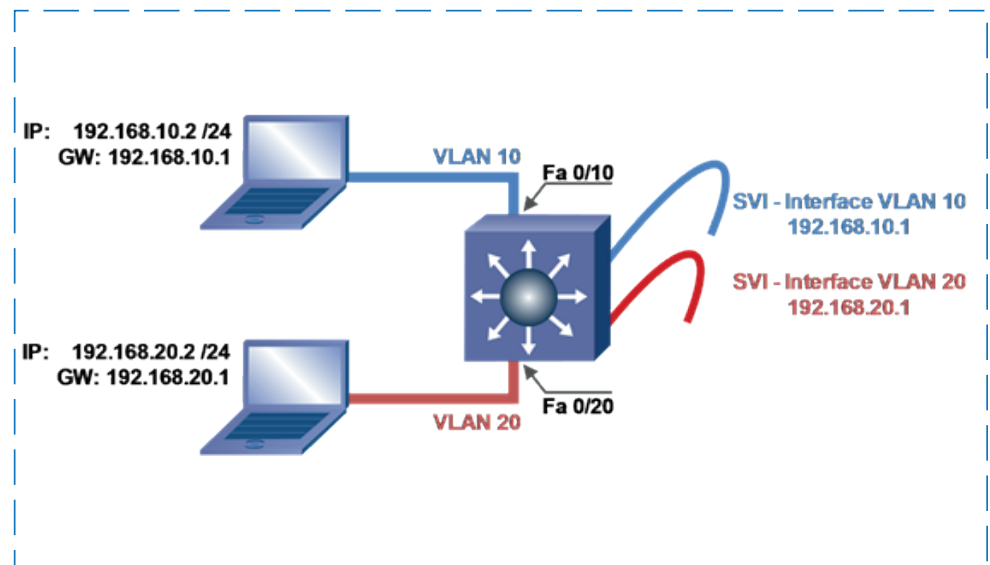
```
MLSwitch(config)# interface fastEthernet 0/10  
MLSwitch(config-if)# switchport mode access  
SMLSwitch(config-if)# switchport access vlan 10
```

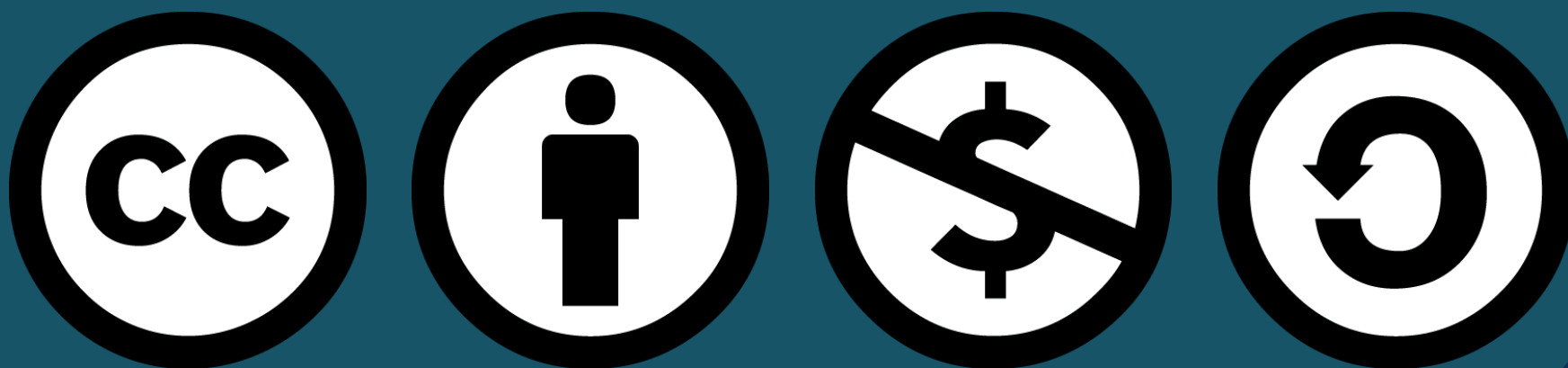
```
MLSwitch(config)# interface fastEthernet 0/20  
MLSwitch(config-if)# switchport mode access  
MLSwitch(config-if)# switchport access vlan 20
```

```
MLSwitch(config)# ip routing
```

```
MLSwitch(config)# interface vlan 10  
MLSwitch(config-if)# no shutdown  
MLSwitch(config-if)# ip address 192.168.10.1 255.255.255.0
```

```
MLSwitch(config)# interface vlan 20  
MLSwitch(config-if)# no shutdown  
MLSwitch(config-if)# ip address 192.168.20.1 255.255.255.0
```





Diseño y edición:

María Esther Pineda
Carolina Villatoro

Descargo de Responsabilidad

El autor y los colaboradores de este trabajo han hecho su mejor esfuerzo en la preparación del mismo para asegurar que su contenido sea lo más exacto posible, sin embargo, no se hacen responsables por el uso de la información en este documento así como de errores u omisiones que pudieran resultar en pérdida de cualquier tipo.

La información está proporcionada “como está” para ser utilizada bajo “su propia cuenta y riesgo”.