LABORATORIO

Tuning Spanning Tree

Topología

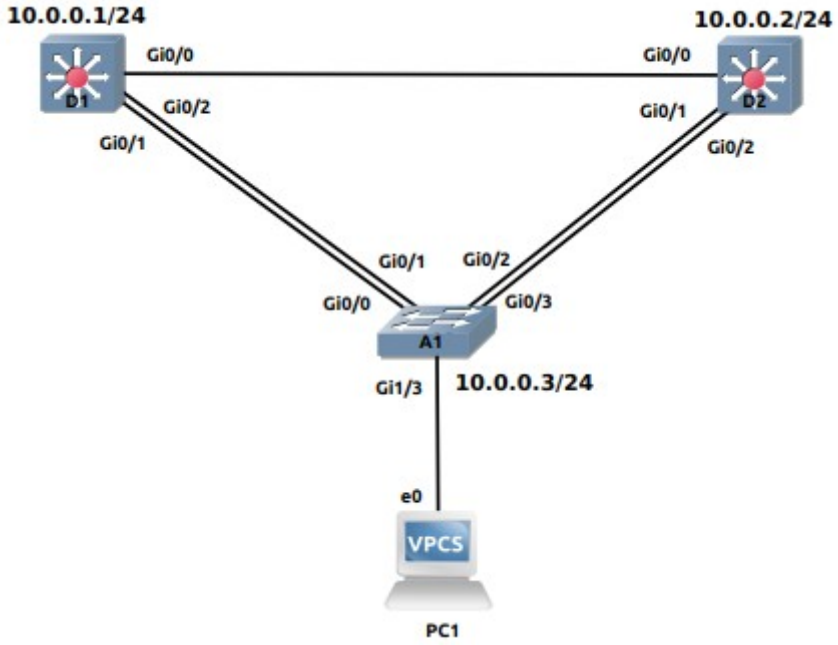


Figura 1.

Objetivos

- **Armado de la topología y cableado de la red.**
- **Configuración básica de los dispositivos.**
- **Configurar diferentes métodos de Tuning STP.**
- **Configurar y observar el comportamiento de los diferentes mecanismos de protección de la Topología STP.**

Introducción

A pesar de que Spanning Tree opera ni bien el switch es conectado a la red, los valores por defecto usados en las decisiones que toma para construir una topología lógica (libre de loops de Capa 2), puede no alinearse con lo que necesitamos de la red. Además, Spanning Tree "por defecto" es vulnerable a diversos escenarios donde el Root Bridge puede ser desplazado, o un loop podría ser introducido en la red.

En este laboratorio se configurarán y observarán diferentes formas de modificar la topología de Spanning Tree de forma tal que se adapte a nuestras necesidades, además se implementarán los diferentes mecanismos de protección de topología STP disponibles.

Actividades

Parte 1: Armado de la Red y Configuración Básica de los Dispositivos

Para el desarrollo del laboratorio, se utilizan tres switches Capa 3 montados en la plataforma GNS3. Al switch A1 se le asignó un símbolo diferente para distinguirlo de los switches de distribución.

Paso 1: Cableado de la red.

a) Realizar el montado y cableado de los dispositivos según la topología mostrada en la ***Figura 1***.

Paso 2: Configuraciones básicas.

a) Conectarse a la consola de cada switch, ingresar al modo de configuración global, y aplicar las siguientes configuraciones básicas:

Switch D1:

```
enable
configure terminal
hostname D1
spanning-tree mode rapid-pvst
banner motd # Switch D1, Lab Tuning STP #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range g0/0-2
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 no shutdown
 exit
interface range g0/3,g1/0-3
 shutdown
 exit
vlan 2
 name VLANDos
 exit
interface vlan 1
 ip address 10.0.0.1 255.255.255.0
 no shutdown
 end
```

Switch D2

```
enable
configure terminal
hostname D2
spanning-tree mode rapid-pvst
banner motd # Switch D2, Lab Tuning STP #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range g0/0-2
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 no shutdown
 exit
interface range g0/3,g1/0-3
 shutdown
 exit
vlan 2
 name VLANDos
 exit
interface vlan 1
 ip address 10.0.0.2 255.255.255.0
 no shutdown
 end
```

Switch A1

```
enable
configure terminal
hostname A1
spanning-tree mode rapid-pvst
banner motd # Switch A1, Lab Tuning STP #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range g0/0-3
 no negotiation auto
 speed 100                        ! Simulación de una interface FastEthernet*
 switchport trunk encapsulation dot1q
 switchport mode trunk
```

```
 switchport nonegotiate
 no shutdown
 exit
interface range g1/0-3
 shutdown
 exit
vlan 2
 name VLANDos
 exit
interface vlan 1
 ip address 10.0.0.3 255.255.255.0
 no shutdown
 end
```

***Nota**: Se cambia la velocidad de las interfaces Gi0/0 a Gi0/3 para simular interfaces FastEthernet.

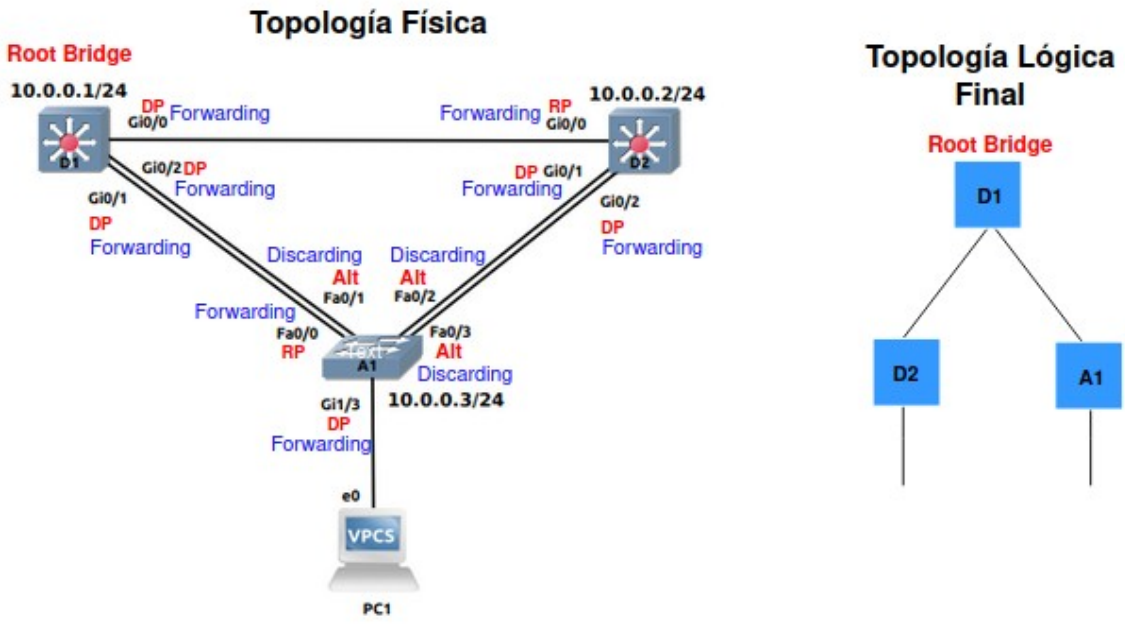
b) Guardar las configuraciones en startup-config

```
copy running-config startup-config      ! Repetir en cada switch
```

Paso 3: Configuraciones por defecto de Spanning Tree.

Con las configuraciones recientemente realizadas, **Rapid Spanning Tree** convergió en una topología lógica de red libre de loops. En este punto, es importante conocer cómo es esta topología lógica, así como saber quién es el **root bridge** y dónde están los **root ports**, **designated ports** y **alternate ports** en cada segmento y para cada VLAN. Puede ser de gran ayuda, volcar toda este información en un gráfico.

A continuación se provee una imagen con toda esta información, correspondiente al laboratorio empleado. La topología final por defecto, es la misma tanto para VLAN 1 como para VLAN 2.



Parte 2: Tuning Spanning Tree

Para esta parte del laboratorio, la PC1 se mantendrá apagada y la interfaz Gi1/3 del switch A1 no participará en STP.

Paso 1: Controlando el Root Bridge.

El **root bridge** actual, fue elegido en base al **Bridge ID más bajo**, conformado por:

- La **prioridad**.
- El **extended system ID** (igual al **VLAN ID**), y
- La **MAC address**.

Con los valores de **prioridad** y **extended system ID** siendo iguales en todos los switches, en una red sin configuraciones de STP realizadas, el **root bridge** es elegido en base a la MAC address numéricamente más baja. A pesar de esto, el resultado final, puede no ser el más optimo o el esperado.

Existen dos formas básicas de manipular y controlar la elección del root bridge:

- Con el comando **spanning-tree vlan** *<vlan_id>* **priority** *<value>* se puede configurar manualmente un valor de prioridad.
- Con el comando **spanning-tree vlan** *<vlan_id>* **root {primary|secondary}** se puede configurar un valor de prioridad de forma automática.

La diferencia entre estos dos comandos es que en el primero (**priority**), a la prioridad se le puede asignar un valor de forma manual (valores múltiplos de 4096). Con el segundo comando (**root primary**) la prioridad toma el valor de **24576** (o un valor de 4096 más bajo que la prioridad del actual **root bridge**), si no se realizaron modificaciones y las configuraciones son las por defecto. Notar que el valor 24576 es seis veces el valor 4096.

Con el comando **root primary** se configura un valor de prioridad lo suficientemente necesario como para ganar la elección del **root bridge**, dejando los valores entre 24576 y el valor por defecto 32768 para ser usado por **root bridges secundarios**.

El comando **spanning-tree vlan** *<vlan_id>* **root secondary** configura de forma estática el valor de la prioridad en **28672**. En una red **sin configuraciones previas**, la prioridad de todos los switches es el valor por defecto 32768, con el comando `root primary` se configura a la prioridad de un switch el valor de 24576 (dos decrementos de 4096 a la prioridad por defecto), mientras que con el comando **root secondary** se configura el valor de prioridad de 28672 al switch que funcionará como **root secundario** (un decremento a la prioridad por defecto).

a) Realizar las configuraciones necesarias para que D1 sea elegido como root primario para la VLAN 1 y D2 sea elegido como el root primario para la VLAN 2, D1 sea elegido como root secundario para la VLAN 2 y D2 sea elegido como root secundario para la VLAN 1.

```
D1#configure terminal
D1(config)#spanning-tree vlan 1 root primary
D1(config)#spanning-tree vlan 2 root secondary
D1(config)#end
```

```
D2#configure terminal
D2(config)#spanning-tree vlan 2 root primary
D2(config)#spanning-tree vlan 1 root secondary
D2(config)#end
```

b) Luego de configurar D1 y D2, ejecutar el comando `show spanning-tree root` en A1.

```
A1#show spanning-tree root

                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0001         24577 0c48.5749.0000        19    2   20  15  Gi0/0
VLAN0002         24578 0ca7.6d02.0000        19    2   20  15  Gi0/2
```

De la salida del comando, se puede observar que el **root port** para la VLAN 1 es Gi0/0 y el **root port** para la VLAN 2 es Gi0/2.

Paso 2: Ajustar los valores de Port Cost para controlar la elección de los Root y Designated Ports.

De acuerdo a la implementación actual de la red, existen dos caminos (**paths**) entre A1 y el **root bridge** de cada VLAN. Se evalúa la suma **path cost + port cost** para determinar el **path más corto hacia el root bridge (root path cost)**. En el caso de que existan múltiples path con el mismo costo hacia el root bridge, se deben evaluar atributos adicionales. En nuestro caso, el número más bajo de interfaz (por ejemplo Gi0/0) es elegida como **root port**, y el número más alto de interfaz (por ejemplo Gi0/1) es puesta en estado **discarding**.

Con la ayuda de los comandos `show spanning-tree <vlan_id>` y `show spanning-tree blockedports` se puede ver los puertos puestos en estado **blocked**. Por ahora, se examinará la VLAN 1 en A1.

a) En A1, ejecutar los comandos `show spanning-tree 1` y `show spanning-tree blockedports`.

```
A1#show spanning-tree vlan 1
VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    24577
              Address     0c48.5749.0000
              Cost        19
              Port        1 (GigabitEthernet0/0)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0cc7.7269.0000
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec
```

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 19        128.1    P2p
Gi0/1               Altn BLK 19        128.2    P2p
Gi0/2               Altn BLK 19        128.3    P2p
Gi0/3               Altn BLK 19        128.4    P2p

A1#show spanning-tree blockedports
Name                 Blocked Interfaces List
-------------------- ------------------------------------
VLAN0001             Gi0/1, Gi0/2, Gi0/3
VLAN0002             Gi0/0, Gi0/1, Gi0/3
Number of blocked ports (segments) in the system : 6
```

Como se puede observar, el switch A1 tiene su **root port** en G0/0 para la VLAN 1. Gi0/1, Gi0/2 y Gi0/3 son **Alternate Blocking Ports**.

Para manipular qué puertos son elegidos como **root port** en **non-root switches**, se debe cambiar el valor de **port cost** o el de **priority port**.

Nota: Los cambios que se implementaran a continuación, son considerados como cambios en la topología y pueden tener un impacto significativo en el funcionamiento de la red. No se recomienda realizar estos tipos de cambios en una red en producción sin el debido planeamiento.

b) En el switch A1, deshabilitar (`shutdown`) las interfaces Gi0/0 y Gi0/1, asignar un nuevo valor de **port cost** a la interfaz Gi0/1, y habilitarlas nuevamente (`on shutdown`).

```
A1#configure terminal
A1(config)#interface range g0/0-1
A1(config-if-range)#shutdown
A1(config-if-range)#exit
A1(config)#interface g0/1
A1(config-if)#spanning-tree cost 12
A1(config-if)#exit
A1(config)#interface range g0/0-1
A1(config-if-range)#no shutdown
A1(config-if-range)#end
```

c) Verificar si hubo algún cambio en la elección del **root port** en A1:

```
A1#show spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID     Priority    24577
              Address     0c48.5749.0000
```

```
              Cost         12
              Port         2 (GigabitEthernet0/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
              Address     0cc7.7269.0000
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Altn BLK 19        128.1    P2p
Gi0/1               Root FWD 12        128.2    P2p
Gi0/2               Altn BLK 19        128.3    P2p
Gi0/3               Altn BLK 19        128.4    P2p

VLAN0002
  Spanning tree enabled protocol rstp
  Root ID    Priority    24578
              Address     0ca7.6d02.0000
              Cost         16
              Port         2 (GigabitEthernet0/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32770  (priority 32768 sys-id-ext 2)
              Address     0cc7.7269.0000
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Altn BLK 19        128.1    P2p
Gi0/1               Root FWD 12        128.2    P2p
Gi0/2               Altn BLK 19        128.3    P2p
Gi0/3               Altn BLK 19        128.4    P2p
```

De la salida del comando `show spanning-tree`, se puede observar que el **root port** elegido por A1 para la VLAN 1 ahora es la interface Gi0/1 y el **port cost** es de 12. También puede observarse que A1 selecciona a Gi0/1 como **root port** para la VLAN 2 y no a G0/2, esto es debido a que Gi0/1 tiene un **path cost** total de 16, que es un valor menor a cualquiera de las conexiones directas a D2 (19).

d) Ajustar el valor de **port cost** a 18 a la interfaz Gi0/1 de A1:

```
A1#configure terminal
A1(config)#interface range g0/0-1
A1(config-if-range)#shutdown
A1(config-if-range)#exit
A1(config)#interface g0/1
A1(config-if)#spanning-tree cost 18
A1(config-if)#exit
A1(config)#interface range g0/0-1
A1(config-if-range)#no shutdown
A1(config-if-range)#end

A1#show spanning-tree root
                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0001         24577 0c48.5749.0000        18    2   20  15  Gi0/1
VLAN0002         24578 0ca7.6d02.0000        19    2   20  15  Gi0/2
```

Paso 3: Ajustar los valores de Port Priority para manipular la elección del Root Port.

El siguiente método para manipular la elección del **root port** se configura en el **root bridge en sí.** En nuestra topología de red, A1 tiene dos conexiones hacia el **root bridge** de la VLAN 2 (switch D2). En este caso, el **root port** fue elegido en base al **port ID más bajo**. El port ID se compone de dos valores, etiquetados como:

- **Prio** (**prioridad**), y
- **Nbr** (**número**).

Nota: El **Nbr** de puerto es un **valor único** y no siempre se corresponde con el **ID de la interfaz.** El valor de **port priority** puede ser cualquier valor entre 0 y 240, con incrementos de 16.

a) Verificar los valores de **port ID** en A1 para la VLAN 2:

```
A1#show spanning-tree vlan 2
<output omitted>
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Altn BLK 19        128.1    P2p
Gi0/1               Altn BLK 18        128.2    P2p
Gi0/2               Root FWD 19        128.3    P2p
Gi0/3               Altn BLK 19        128.4    P2p
```

Como era de esperarse, con dos valores iguales de **path cost** hacia el **root bridge**, el valor más bajo de **port ID** es elegido como el **root port.**

b) Modificar el valor de **port priority** a la interfaz Gi0/2 de D2 para que sea elegida como el puerto preferido.

```
D2(config)#configure terminal
D2(config)#interface range g0/1-2
D2(config-if-range)#shutdown
D2(config-if-range)#exit
D2(config)#interface g0/2
D2(config-if)#spanning-tree port-priority 64
D2(config-if)#exit
D2(config)#interface range g0/1-2
D2(config-if-range)#no shutdown
D2(config-if-range)#end
```

c) Verificar si esta modificación en el switch D2 tuvo algún impacto en el switch A1:

```
A1#show spanning-tree vlan 2
<output omitted>
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Altn BLK 19        128.1    P2p
Gi0/1               Altn BLK 18        128.2    P2p
Gi0/2               Altn FWD 19        128.3    P2p
Gi0/3               Root BLK 19        128.4    P2p
```

Gi0/3 ahora es elegido como **root port**, esta elección se basa en el **valor de prioridad más bajo** de la interfaz Gi0/2 de D2. Notar que este valor más bajo de prioridad **no aparece en A1**, STP no propaga el port priority.

Paso 4: Implementar Spanning Tree Port Fast.

Tanto en STP como en RSTP, un puerto recién conectado debe garantizarse que no cree un bucle de Capa 2 antes de poder convertirse en un puerto en estado de reenvío (**forwarding**). Esto puede tomar hasta 30 segundos. Sin embargo, dicha verificación no es necesaria para puertos conectados a dispositivos finales como PCs, impresoras de red, servidores, etc. En RSTP, estos puertos se denominan **edge ports** (los puertos que se conectan a otros switches en la red se llaman **non-edge ports**). Los **edge ports** pueden entrar de manera segura en el estado de **forwarding** inmediatamente después de activarse, ya que no se conectan a ningún dispositivo capaz de reenviar frames.

Cisco desarrolló una característica llamada **portfast** que esencialmente permite definir un puerto como un **edge port**. Cualquier puerto habilitado con **portfast** entrará en estado de forwarding inmediatamente después de activarse, sin pasar por los estados intermedios, lo que ahorra 30 segundos cada vez que se establece una nueva conexión en el puerto. **Portfast** puede ser utilizado en todas las versiones de STP.

Además de permitir que un puerto pase directamente al estado de forwarding tan pronto como se conecta, el concepto de **edge port** es extremadamente importante en RSTP y MSTP. Como parte de las mejoras sobre STP, RSTP utiliza un mecanismo de **Proposal/Agreement** para habilitar rápidamente, pero de manera segura, un enlace entre switches si uno de los switches tiene su **root port** en ese enlace.

Al recibir un **Proposal** en su **root port**, un switch coloca todos sus **non-edge designated ports** en el estado de **discarding**, cortándose efectivamente de la red y previniendo un posible bucle de Capa 2 (esto se denomina operación de **Sincronización** o **Sync**). Una vez que se logra esto, el switch envía un **Agreement** de regreso por su **root port**, para que el **designated port upstream** que recibe este **Agreement** pueda colocarse inmediatamente en el estado de forwarding. Luego, el switch comenzará a enviar sus propios **Proposals** en todos sus **non-edge designated ports** que acababan de pasar a **discarding**. Esperará a que lleguen **Agreements** de los **switches downstream**, lo que permitirá que estos puertos se conviertan instantáneamente en forwarding nuevamente.

Si los dispositivos finales están conectados a puertos no configurados como **edge ports** (es decir, con **portfast**), estos puertos pasarán a **discarding** durante la operación de **Sync**. Dado que los hosts finales no soportan RSTP y no pueden enviar un **Agreement** de regreso, quedarán desconectados de la red durante 30 segundos hasta que los puertos alcancen el estado de forwarding utilizando temporizadores ordinarios. Como resultado, los usuarios experimentarán interrupciones significativas en la conectividad.

Los puertos configurados como **edge ports** no se ven afectados por la operación de **Sync** y permanecerán en el estado de forwarding incluso durante el manejo de **Proposal/Agreement**. Activar RSTP en una red sin configurar adecuadamente los puertos hacia hosts finales como **edge ports** puede hacer que la red funcione posiblemente peor que con STP. Con RSTP, la configuración adecuada de puertos hacia hosts finales como **edge ports** es crucial. Los switches Cisco configuran por defecto todos sus puertos como **non-edge ports**.

a) Encender la PC1.

b) En el switch A1, ejecutar el comando `debug spanning-tree events`, luego habilitar la interfaz Gi1/3, esperar unos segundos y habilitarla nuevamente, seguido a esto ejecutar `undebug all`.

```
A1#debug spanning-tree events
Spanning Tree event debugging is on
A1#configure terminal
A1(config)#interface g1/3
A1(config-if)#no shutdown
*Nov  4 14:34:55.887: RSTP(1): initializing port Gi1/3
*Nov  4 14:34:55.888: RSTP(1): Gi1/3 is now designated
*Nov  4 14:34:55.891: RSTP(1): transmitting a proposal on Gi1/3
*Nov  4 14:34:56.325: RSTP(1): transmitting a proposal on Gi1/3
A1(config-if)#shutdown
*Nov  4 14:35:10.332: RSTP(1): transmitting a proposal on Gi1/3
*Nov  4 14:35:10.888: RSTP(1): Gi1/3 fdwhile Expired
A1(config-if)#
*Nov  4 14:35:13.665: %LINK-5-CHANGED: Interface GigabitEthernet1/3, changed state to
administratively down
*Nov  4 14:35:14.665: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/3, changed state to down
A1#undebug all
All possible debugging has been turned off
```

Lo que se observa aquí es que el switch intenta realizar el proceso de **Proposal/Agreement** en la interfaz Gi1/3. Sin embargo, esto no tiene sentido porque el dispositivo conectado a Gi1/3 es un host final y no entiende Spanning Tree. Esto añade el riesgo de un retraso de 30 segundos antes de que el host pueda enviar datos, como por ejemplo una solicitud DHCP a la red.

c) En el switch A1, ejecutar nuevamente el comando `debug spanning-tree events`, luego configurar la interfaz Gi1/3 con el comando `spanning-tree portfast` seguido del comando `no shutdown`. Esto designa la interfaz Gi1/3 como una interfaz que nunca se conectará a otro switch y, por lo tanto, nunca causará un bucle en la topología, permitiendo posteriormente que dicha interfaz pase inmediatamente a forwarding. Observar la salida del comando.

```
A1#debug spanning-tree events
Spanning Tree event debugging is on
A1#configure terminal
A1(config)#interface g1/3
A1(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
 host. Connecting hubs, concentrators, switches, bridges, etc... to this
 interface  when portfast is enabled, can cause temporary bridging loops.
 Use with CAUTION
%Portfast has been configured on GigabitEthernet1/3 but will only
 have effect when the interface is in a non-trunking mode.
A1(config-if)#no shutdown
*Nov  4 14:48:24.342: RSTP(1): initializing port Gi1/3
*Nov  4 14:48:24.343: RSTP(1): Gi1/3 is now designated
*Nov  4 14:48:26.196: %LINK-3-UPDOWN: Interface GigabitEthernet1/3, changed state to
up
*Nov  4 14:48:26.553: %SYS-5-CONFIG_I: Configured from console by console
*Nov  4 14:48:27.196: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/3, changed state to up
A1(config-if)#end
```

Como se puede observar, RSTP ve a la interfaz Gi1/3 como **designated port**, y nunca enviará un **Proposal** por esa interfaz, ya que está configurada como **portfast**.

Existen dos formas adicionales para configurar una interfaz como **portfast**, haciendo uso del comando de configuración de interfaz `switchport host` y con el comando de configuración global `spanning-tree portfast default`.

- `switchport host` no solamente habilita **portfast**, además configura la interfaz estáticamente en **modo access**, deshabilitando los protocolos de link aggregation.
- `spanning-tree portfast default` configura por defecto con **portfast** a todas las interfaces configuradas como **access ports**. Sólo se debe configurar la interfaz con `switchport mode access` y **portfast** se habilitará automaticamente.

Para verificar que un puerto está en modo **portfast** se puede revisar la configuración en ejecución (running-config) de ese puerto o examinando los detalles de Spanning Tree para ese puerto.

Por ejemplo, utilizando el comando `show spanning-tree interface <interface-id>` para confirmar que la interfaz está en modo **edge**, como se muestra a continuación:

```
A1#show spanning-tree interface g1/3
Vlan                Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
VLAN0001            Desg FWD 4         128.8    P2p Edge
```

También se puede hacer uso del siguiente comando:

```
A1#show spanning-tree detail | section GigabitEthernet1/3
 Port 8 (GigabitEthernet1/3) of VLAN0001 is designated forwarding
   Port path cost 4, Port priority 128, Port Identifier 128.8.
   Designated root has priority 24577, address 0c48.5749.0000
   Designated bridge has priority 32769, address 0cc7.7269.0000
   Designated port id is 128.8, designated path cost 18
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   The port is in the portfast edge mode
   Link type is point-to-point by default
   BPDU: sent 756, received 0
```

Parte 3: Implementar los Diferentes Mecanismos de Protección de la Topología

En esta parte del laboratorio, se implementará y observará la operación de los diferentes mecanismos de protección de la topología que existen, tales como **root guard**, **bpdu guard**, **bpdu filter** y **loop guard**.

Paso 1: Implementar Root Guard.

Root Guard ayuda a prevenir la toma de control del **root bridge** o del **root port.** Se configura en el puerto que se desea proteger. Si un puerto protegido por **root guard** recibe un **BPDU** superior que normalmente causaría que el puerto se convierta en **root port**, el **BPDU** será descartado y el puerto pasará al estado **Root-Inconsistent**. Un estado inconsistente de STP difiere del estado de **error disabled** en que el puerto no se deshabilita por completo, en su lugar, solo se coloca en el estado de **Blocking** (**discarding**) y volverá a su rol y estado apropiados una vez que desaparezca la causa de su inconsistencia. Con **root guard**, un puerto se reintegrará automáticamente a su rol y estado adecuados cuando deje de recibir **BPDUs superiores**.

Nota: **root guard** es un mecanismo de protección utilizado en situaciones donde se necesita combinar dos redes (por ejemplo nuestra red y la red de un cliente) para formar un único dominio STP, pero se desea que el **root bridge** esté en nuestra porción de la red y no queremos que un switch del cliente se convierta en el **root bridge** de la topología, ni que el tráfico deba pasar a través de la red del cliente. En estos casos, se colocaría **root guard** en los puertos hacia el cliente. Sin embargo, usar **root guard** dentro de nuestra propia red, sería perjudicial. Nuestra red puede considerarse confiable y no hay un **root bridge rogue** contra el cual protegerse. Usar **root guard** en nuestra propia red causaría que STP no pueda converger a una nueva topología funcional si fallara cualquiera de los enlaces primarios, y

también impediría que la red converja a un **root bridge secundario** si el **root bridge primario** fallara por completo.

a) Para observar el comportamiento de **root guard**, se configurará en un **designated port** en D2. D2 es el **root bridge** en la VLAN 2, por lo tanto, todos sus puertos son **designated ports**.

```
D2#show spanning-tree detail | include VLAN0002
 VLAN0002 is executing the rstp compatible Spanning Tree protocol
 Port 1 (GigabitEthernet0/0) of VLAN0002 is designated forwarding
 Port 2 (GigabitEthernet0/1) of VLAN0002 is designated forwarding
 Port 3 (GigabitEthernet0/2) of VLAN0002 is designated forwarding
```

b) En A1, identificar cuál es el **root port** para la VLAN0002.

```
A1#show spanning-tree root
                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0001         24577 0c48.5749.0000        18    2   20  15  Gi0/1
VLAN0002         24578 0ca7.6d02.0000        19    2   20  15  Gi0/3
```

c) En el switch D2, configurar **root guard** en los puertos conectados a A1.

```
D2#configure terminal
D2(config)#interface range g0/1-2
D2(config-if-range)#spanning-tree guard root
*Nov  4 19:35:08.095: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard enabled on port GigabitEthernet0/1.
*Nov  4 19:35:08.100: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard enabled on port GigabitEthernet0/2.
D2(config-if-range)#end
```

d) Para verificar la operatividad de **root guard**, configurar A1 para que se convierta en el **root bridge** de la VLAN0002:

```
A1#configure terminal
A1(config)#spanning-tree vlan 2 priority 16384
```

e) En el switch D2, observar la salida del comando `show spanning-tree vlan 2`:

```
D2#show spanning-tree vlan 2
VLAN0002
  Spanning tree enabled protocol rstp
  Root ID     Priority    16386
              Address     0cc7.7269.0000
              Cost        8
              Port        1 (GigabitEthernet0/0)
```

```
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24578  (priority 24576 sys-id-ext 2)
             Address     0ca7.6d02.0000
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Root FWD 4         128.1    P2p
Gi0/1               Desg BKN*4         128.2    P2p *ROOT_Inc
Gi0/2               Desg BKN*4          64.3    P2p *ROOT_Inc
```

Esta salida presenta dos indicadores del problema. Primero, **BKN*** que es la abreviatura de **BROKEN**, y ***ROOT_Inc** representa el mensaje de Inconsistencia del Root (**Root Inconsistent**). También se puede solicitar una lista de todos los puertos inconsistentes de STP, incluyendo el motivo de su inconsistencia, haciendo uso del comando `show spanning-tree inconsistentports`.

```
D2#show spanning-tree inconsistentports
Name                 Interface              Inconsistency
-------------------- ---------------------- ------------------
VLAN0002             GigabitEthernet0/1     Root Inconsistent
VLAN0002             GigabitEthernet0/2     Root Inconsistent
Number of inconsistent ports (segments) in the system : 2
```

f) Revertir las configuraciones realizadas con los comandos `no spanning-tree vlan 2 priority 16384` en A1, y luego `no spanning-tree guard root` en las interfaces Gi0/1 y Gi0/2 de D2 para deshabilitar **root guard.**

```
A1(config)#no spanning-tree vlan 2 priority 16384
A1(config)#end
D2#configure terminal
D2(config)#interface range g0/1-2
D2(config-if-range)#no spanning-tree guard root
*Nov  4 19:50:59.860: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard disabled on port GigabitEthernet0/1.
*Nov  4 19:50:59.863: %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard disabled on port GigabitEthernet0/2.
D2(config-if-range)#end
```

Paso 2: Implementar BPDU Guard.

PortFast hace que una interfaz pase inmediatamente al estado de forwarding. Existe el riesgo de que, si dos puertos habilitados con **portfast** se conectan inadvertidamente o de manera maliciosa, ambos se activen como puertos en estado forwarding, creando inmediatamente un bucle de Capa 2.

El comportamiento esperado por defecto de un puerto **portfast** que recibe una **BPDU** es que dicho puerto revierta a un puerto spanning-tree normal **no-edge**. Existe la posibilidad de que la carga en un switch dado sea demasiado alta para manejar adecuadamente los **BPDUs** recibidos, prolongando la condición de bucle.

BPDU Guard añade otra capa de protección. Siempre que un puerto protegido por **BPDU Guard** recibe inesperadamente una **BPDU**, se coloca inmediatamente en estado **err-disabled**. Cualquier interfaz puede protegerse con **BPDU Guard**, pero su uso típico es en puertos habilitados con **portfast**.

BPDU Guard puede configurarse de manera global o por interfaz individual. Si **BPDU Guard** se configura a nivel global mediante el comando `spanning-tree portfast bpduguard default`, **BPDU Guard** se habilitará automáticamente en todos los puertos habilitados con **portfast** del switch. Si **BPDU Guard** se configura en una interfaz mediante el comando `spanning-tree bpduguard enable`, se aplicará a este puerto de forma incondicional, independientemente de si es un puerto habilitado con **portfast**.

Para este ejemplo, se configurará **BPDU Guard** en una interfaz de **trunking** y que sea un puerto **no-root** en A1. Configurar **BPDU Guard** en una interfaz destinada a ser un **trunk** no es una práctica recomendada, lo haremos sólo para demostrar la funcionalidad de la herramienta.

a) Verificar los puertos **trunks** y los **root ports** en el switch A1 utilizando los comandos `show spanning-tree root` y `show interface trunk`.

```
A1#show interfaces trunk
Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1
Gi0/1       on               802.1q         trunking      1
Gi0/2       on               802.1q         trunking      1
Gi0/3       on               802.1q         trunking      1
Port        Vlans allowed on trunk
Gi0/0       1-4094
Gi0/1       1-4094
Gi0/2       1-4094
Gi0/3       1-4094
Port        Vlans allowed and active in management domain
Gi0/0       1-2
Gi0/1       1-2
Gi0/2       1-2
Gi0/3       1-2
Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       none
Gi0/1       1
Gi0/2       none
Port        Vlans in spanning tree forwarding state and not pruned
Gi0/3       2
```

A partir de la salida que se muestra, se observa que la interfaz Gi0/0 cumple con los requisitos para esta demostración (**trunk** y **no root**).

```
A1#show spanning-tree root
                                        Root    Hello Max Fwd
Vlan                Root ID             Cost    Time  Age Dly  Root Port
---------------- ------------------- --------- ----- --- ---  ------------
VLAN0001         24577 0c48.5749.0000        18    2   20  15  Gi0/1
VLAN0002         24578 0ca7.6d02.0000        19    2   20  15  Gi0/3
```

b) En la interfaz Gi0/0 del switch A1, ejecutar el comando `spanning-tree bpduguard enable`.

```
A1#configure terminal
A1(config)#interface g0/0
A1(config-if)#spanning-tree bpduguard enable
*Nov  5 14:04:19.630: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU on port Gi0/0 with BPDU Guard enabled. Disabling port.
*Nov  5 14:04:19.630: %PM-4-ERR_DISABLE: bpduguard error detected on Gi0/0, putting Gi0/0 in err-disable state
*Nov  5 14:04:20.632: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down
*Nov  5 14:04:21.634: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to down
```

Como se puede observar, la interfaz pasa casi inmediatamente al estado **err-disabled**.

Ejecutar el comando `shutdown`, eliminar **BPDU Guard** con el comando `no spanning-tree bpduguard enable` y, finalmente, ejecutar el comando `no shutdown` en la interfaz Gi0/0 para reactivarla.

```
A1(config-if)#shutdown
A1(config-if)#spanning-tree bpduguard enable
*Nov  5 14:04:40.395: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to administratively down
A1(config-if)#no spanning-tree bpduguard enable
A1(config-if)#no shutdown
*Nov  5 14:05:02.857: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Nov  5 14:05:03.857: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
A1(config-if)#end
```

Verificar que el **trunk** esté operativo mediante el comando `show interface trunk`.

```
A1#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       on               802.1q         trunking      1
Gi0/1       on               802.1q         trunking      1
```

```
Gi0/2       on              802.1q          trunking      1
Gi0/3       on              802.1q          trunking      1

Port        Vlans allowed on trunk
Gi0/0       1-4094
Gi0/1       1-4094
Gi0/2       1-4094
Gi0/3       1-4094

Port        Vlans allowed and active in management domain
Gi0/0       1-2
Gi0/1       1-2
Gi0/2       1-2
Gi0/3       1-2

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       none
Gi0/1       1
Gi0/2       none

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/3       2
```

Paso 3: Implementar BPDU Filter.

Ni **portfast** ni **BPDU Guard** impiden que el switch envíe **BPDUs** por una interfaz, si se quiere que una interfaz deje de enviar **BPDUs**, se debe utilizar **BPDU Filter**. **BPDU Filter** puede configurarse tanto a nivel global como en una interfaz específica.

Si **BPDU Filter** se configura a nivel global mediante el comando de configuración global `spanning-tree portfast bpdufilter default`, **BPDU Filter** sólo se aplica a los puertos con **portfast** habilitado. Cuando estos puertos se activan, enviarán **hasta 11 BPDUs** y luego dejarán de enviar **BPDUs**. Si una interfaz configurada con **BPDU Filter** recibe un **BPDU** en cualquier momento, **BPDU Filter** y **portfast** se desactivarán en ese puerto, y este se convertirá en una interfaz de spanning tree normal. Como resultado, **BPDU Filter** configurado globalmente no impide que los puertos reciban y procesen **BPDUs**, sólo intenta detener el envío de **BPDUs** en puertos donde, probablemente, no hay un dispositivo conectado que los procese.

Si se configura una interfaz con el comando `spanning-tree bpdufilter enable`, el puerto dejará de enviar y procesar los **BPDUs** recibidos por completo. Esto se puede usar, por ejemplo, para dividir una red en dos o más dominios STP independientes, cada uno con su propio **root bridge** y topología resultante. Sin embargo, dado que estos dominios ya no están protegidos contra bucles por STP, es responsabilidad del administrador de red asegurarse de que estos dos dominios nunca se conecten por más de un solo enlace.

Para esta demostración, se configurará **BPDU Filter** a nivel de interfaz.

a) Con ayuda del comando `show spanning-tree interface g1/3 | include BPDU`, observar cuántos **BPDUs** fueron enviados por la interfaz Gi1/3 en el switch A1. Repetir el comando para verificar que el conteo se incrementa:

```
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 7, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 9, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 11, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 13, received 0
```

b) Configurar **BPDU Filter** a nivel de interfaz en la interfaz Gi1/3 del switch A1:

```
A1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#interface g1/3
A1(config-if)#spanning-tree bpdufilter enable
A1(config-if)#end
```

c) Repetir los pasos del punto a:

```
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 103, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 103, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 103, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 103, received 0
```

Como se puede observar, se dejaron de enviar **BPDUs** por la interfaz Gi1/3 de A1.

d) Deshabilitar **BPDU Filter** con el comando `no spanning-tree bpdufilter enable`:

```
A1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
A1(config)#interface g1/3
A1(config-if)#no spanning-tree bpdufilter enable
A1(config-if)#end
```

e) Verificar que los **BPDUs** volvieron a ser enviados por la interfaz:

```
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 191, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 193, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 195, received 0
A1#show spanning-tree interface g1/3 detail | include BPDU
   BPDU: sent 197, received 0
```

Paso 4: Implementar Loop Guard.

Loop Guard evita que los puertos **root** y **alternate** se conviertan en puertos **designated** si repentinamente dejan de recibirse **BPDU** en ellos.

En una red STP normal, todos los puertos reciben y procesan **BPDU**, incluso los puertos en estado **blocking** (**discarding**). Así es como saben que el dispositivo en el otro extremo del enlace está vivo y aún es superior a ellos (**BID superior**). Si un puerto **bloqueado** deja de recibir estos **BPDU**, solo puede asumir que el dispositivo del otro lado ya no está presente y que ahora él debería pasar a estado **forwarding** para el segmento dado. Un ejemplo de cuándo podría ocurrir esto es cuando la **fibra Rx** en un cable óptico se desconecta, se corta o se conecta a un puerto o dispositivo diferente al de la **fibra Tx** correspondiente, creando un enlace unidireccional. Esto podría causar bucles de Capa 2 permanentes en la red, por lo que **Loop Guard** ayuda a prevenirlos.

Loop Guard se puede habilitar globalmente usando el comando de configuración global `spanning-tree loopguard default`, o por interfaz usando el comando `spanning-tree guard loop`. **Loop Guard** nunca debe habilitarse en puertos con **portfast** activado.

Para este ejemplo, se configurará **Loop Guard** en un puerto **alternate** del switch A1 y luego se dejará de enviar **BPDU** desde el puerto **designated** correspondiente en el otro extremo del enlace.

a) Verificar que puertos son **alternate** para la VLAN 2 en el switch A1:

```
A1#show spanning-tree vlan 2 | begin Interface
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Altn BLK 19        128.1    P2p
Gi0/1               Altn BLK 18        128.2    P2p
Gi0/2               Altn BLK 19        128.3    P2p
Gi0/3               Root FWD 19        128.4    P2p
```

b) En el switch A1, configurar la interfaz Gi0/0 con **Loop Guard**:

```
A1#configure terminal
A1(config)#interface g0/0
A1(config-if)#spanning-tree guard loop
A1(config-if)#end
```

c) En el switch D1, configurar la interfaz Gi0/1 (interfaz que se conecta a Gi0/0 del switch A1) con **BPDU Filter**:

```
D1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface g0/1
D1(config-if)#spanning-tree bpdufilter enable
D1(config-if)#end
```

d) En el switch A1, se verá un mensaje de **SYSLOG** informando que **Loop Guard** bloqueó el puerto Gi0/0. Ejecutar el comando `show spanning-tree vlan 2` para ver el estado de la interfaz Gi0/0:

```
A1#
*Nov  7 14:35:15.983: %SPANTREE-2-LOOPGUARD_BLOCK: Loop guard blocking port
GigabitEthernet0/0 on VLAN0002.

A1#show spanning-tree vlan 2
VLAN0002
<output ommited>
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Gi0/0               Desg BKN*19        128.1    P2p *LOOP_Inc
Gi0/1               Altn BLK 18        128.2    P2p
Gi0/2               Altn BLK 19        128.3    P2p
Gi0/3               Root FWD 19        128.4    P2p
```

Se puede ver que la interfaz Gi0/0 se muestra en estado **broken.** Ejecutar el comando `show spanning-tree inconsistentports` para obtener más información:

```
A1#show spanning-tree inconsistentports
Name                 Interface               Inconsistency
-------------------- ----------------------- ------------------
VLAN0001             GigabitEthernet0/0      Loop Inconsistent
VLAN0002             GigabitEthernet0/0      Loop Inconsistent
Number of inconsistent ports (segments) in the system : 2
```

La interfaz Gi0/0 se muestra como **Loop Inconsistent.**

e) Deshabilitar **BPDU Filter** en la interfaz Gi0/1 del switch D1:

```
D1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
D1(config)#interface g0/1
D1(config-if)#no spanning-tree bpdufilter enable
D1(config-if)#end
```

f) En el switch A1 se debería observar un mensaje de **SYSLOG** informando que se deshabilitó **Loop Guard** en la interfaz Gi0/0:

```
A1#

*Nov  7 14:49:06.976: %SPANTREE-2-LOOPGUARD_UNBLOCK: Loop guard unblocking port
GigabitEthernet0/0 on VLAN0002.

A1#show spanning-tree inconsistentports

Name                 Interface               Inconsistency

-------------------- ----------------------- ------------------

Number of inconsistent ports (segments) in the system : 0
```

Remover la configuración de loop guard en dicha interfaz:

```
A1#configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

A1(config)#interface g0/0

A1(config-if)#no spanning-tree guard loop

A1(config-if)#end
```

Conclusión

Este laboratorio permitió interiorizar no solo la teoría de STP y RSTP, sino también su comportamiento real en una topología con redundancia.

El análisis de los roles, costos, prioridades y mecanismos preventivos proporciona una comprensión completa de cómo Cisco IOS construye y mantiene una red estable y libre de loops.

La experiencia adquirida aquí establece una base sólida para escenarios más avanzados, como MST, EtherChannel con STP, y optimización de redes empresariales.