



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

---

# Implementación de sensores geolocalizados para la obtencion de datos en un area metropolitana

---

Rodríguez Martinez, **David**



# Índice general

1	Introducción	1
2	¿Qué <i>Spanning-Tree-Protocol</i> ?	2
2.1	STP ( <i>Spanning Tree Protocol</i> )	2
2.2	RSTP ( <i>Rapid Spanning Tree Protocol</i> )	3
3	Objetivos	4
4	Material	5
5	Configuración	6
5.1	Configuración de la red	6
5.1.1	Arquitectura básica	7
5.1.2	Arquitectura básica + diagonal	8
5.2	Configuración inicial	9
5.2.1	Configuración <i>Wireless</i>	9
5.2.2	Configuración de los otros dispositivos	11
5.3	Configuración STP / RSTP	13
5.3.1	Configuración <i>out-of-the-box</i> de RSTP	13
5.3.2	Configuración para evitar otro dispositivo con STP no autorizado	14
5.3.3	Configuración para evitar bucles en el clientes	15
5.3.4	Configuración del <i>path cost</i>	16
5.3.5	Bucle para <i>Backup Port</i>	16
6	Pruebas y resultados	17
6.1	Introducción	17
6.2	Desactivar el STP en modo Seguro	18
6.3	Desconectar un enlace	18
6.4	Modificar el valor de <i>path cost</i>	19
6.5	Proteger la integridad de la estructura cuando se añada un dispositivo en un extremo de la red configurado con STP ( <i>edge</i> )	20
6.6	Proteger la estructura cuando un posible usuario cree un lazo cerrado entre dos puertos ( <i>horizon</i> )	21

6.7 Añadir enlace para tener redundancia ( <i>Backup Port</i> ) . . . . .	22
6.8 Evitar los dos problemas anteriores combinados. . . . .	23
7 Conclusiones . . . . .	24

# Capítulo 1

## ¿Qué *Spanning-Tree-Protocol*?

### 1.1 STP (*Spanning Tree Protocol*)

En comunicaciones, STP (del inglés *Spanning Tree Protocol*) es un protocolo de red de nivel 2 del modelo OSI (capa de enlace de datos).

Su función es la de gestionar la presencia de bucles en topologías de red debido a la existencia de enlaces redundantes, necesarios en muchos casos para garantizar la alta disponibilidad entre los dispositivos de una arquitectura de comunicación.

Los bucles impiden el funcionamiento normal de la red puesto que los dispositivos de interconexión de nivel de enlace de datos reenvían indefinidamente las tramas broadcast y multicast, creando así un bucle infinito que consume tanto el ancho de banda de la red como CPU de los dispositivos de enrutamiento.

Al no existir un campo TTL (tiempo de vida) en las tramas de capa 2, éstas se quedan atrapadas indefinidamente hasta que un administrador de sistemas rompa el bucle. Un router, por el contrario, sí podría evitar este tipo de reenvíos indefinidos. La solución consiste en permitir la existencia de enlaces físicos redundantes, pero creando una topología lógica libre de bucles. STP calcula una única ruta libre de bucles entre los dispositivos de la red pero manteniendo los enlaces redundantes desactivados como reserva, con el fin de activarlos en caso de fallo.

Si la configuración de STP cambia, o si un segmento en la red redundante llega a ser inalcanzable, el algoritmo reconfigura los enlaces y restablece la conectividad, activando uno de los enlaces de reserva.

El protocolo RSTP o STP, se basan en el algoritmo de caminos mínimos de Dijkstra, aplicado en el árbol lógico obtenido a partir de la configuración de red física, normalmente en estructura de malla para la alta disponibilidad, además, transforma una red física con forma de malla, en la que existen bucles, por una red lógica en forma de árbol (libre de bucles). Los puentes se comunican mediante mensajes de configuración llamados *Bridge Protocol Data Units*(BPDU).

El protocolo establece identificadores por puente y elige el que tiene la prioridad más alta (el número más bajo de prioridad numérica), como el puente raíz (*Root Bridge*), normalmente el equipo con mas carga dentro de la red. Este puente raíz establecerá el camino de menor coste para todas las redes y en caso de cambios dentro de la arquitectura de red, el protocolo

RSTP recalcula una nueva topología libre de redundancia dentro de un periodo de tiempo de convergencia.

El tráfico se redigirá desde cada puerto en función de dos parámetros principales: prioridad y dirección MAC.

La prioridad de cada *bridge*, se determina en función del que tiene la prioridad mas alta después del *Root Bridge*, que es el menor valor numérico después del *Root Bridge*. Además, se dispone de un parámetro configurable: el *Span path cost*, que se puede utilizar para redirigir el tráfico a través de los nodos de la configuración.

En caso de obtener el mismo coste en dos puertos, la prioridad se determinara en función del menor valor de MAC.

Para resolver el problema se utiliza el protocolo de routing, *Spanning Tree Protocol* (STP) así como sus variantes.

STP se ha convertido en el protocolo preferido para prevenir bucles de *layer 2* en topologías que incluyen redundancia aunque se pueden encontrar problemas que como: tiempos de convergencia demasiados altos y una configuración que puede complicarse si no se conoce bien el principio de funcionamiento.

- STP (**802.1d**): impide bucles usando un "*timer*".
- Rapid Spanning Tree (RSTP - **802.1w**).

## 1.2 RSTP (*Rapid Spanning Tree Protocol*)

Rapid Spanning Tree Protocol (RSTP) es un protocolo de red de la segunda capa OSI, (nivel de enlace de datos), que gestiona enlaces redundantes. Especificado en IEEE 802.1w, es una evolución del *Spanning tree Protocol* (STP), reemplazándolo en la edición 2004 del 802.1d. RSTP reduce significativamente el tiempo de convergencia de la topología de la red cuando ocurre un cambio en la topología.

Se ha convertido en el protocolo preferido para prevenir bucles de capa 2 en topologías que incluyen redundancia. Además de que el 802.1w contiene mejoras, retiene compatibilidad con su antecesor 802.1D dejando algunos parámetros sin cambiar. Por ejemplo, RSTP mantiene el mismo formato de BPDU que STP sólo que cambia el campo de versión, el cual se le asigna el valor de 2.

RSTP también define el concepto de *edge-port*, el cual también se menciona en STP como *Port-Fast*, en donde el puerto se configura como tal cuando se sabe que nunca será conectado hacia otro switch de manera que pasa inmediatamente al estado de direccionamiento sin esperar los pasos intermedios del algoritmo –etapas de escucha y aprendizaje– los cuales consumen tiempo. El tipo de enlace es detectado automáticamente, pero puede ser configurado explícitamente para hacer más rápida la convergencia.

## Capítulo 2

# ¿Qué *Spanning-Tree-Protocol*?

### 2.1 STP (*Spanning Tree Protocol*)

En comunicaciones, STP (del inglés *Spanning Tree Protocol*) es un protocolo de red de nivel 2 del modelo OSI (capa de enlace de datos).

Su función es la de gestionar la presencia de bucles en topologías de red debido a la existencia de enlaces redundantes, necesarios en muchos casos para garantizar la alta disponibilidad entre los dispositivos de una arquitectura de comunicacion.

Los bucles impiden el funcionamiento normal de la red puesto que los dispositivos de interconexión de nivel de enlace de datos reenvían indefinidamente las tramas broadcast y multicast, creando así un bucle infinito que consume tanto el ancho de banda de la red como CPU de los dispositivos de enrutamiento.

Al no existir un campo TTL (tiempo de vida) en las tramas de capa 2, éstas se quedan atrapadas indefinidamente hasta que un administrador de sistemas rompa el bucle. Un router, por el contrario, sí podría evitar este tipo de reenvíos indefinidos. La solución consiste en permitir la existencia de enlaces físicos redundantes, pero creando una topología lógica libre de bucles. STP calcula una única ruta libre de bucles entre los dispositivos de la red pero manteniendo los enlaces redundantes desactivados como reserva, con el fin de activarlos en caso de fallo.

Si la configuración de STP cambia, o si un segmento en la red redundante llega a ser inalcanzable, el algoritmo reconfigura los enlaces y restablece la conectividad, activando uno de los enlaces de reserva.

El protocolo RSTP o STP, se basan en el algoritmo de caminos mínimos de Dijkstra, aplicado en el árbol lógico obtenido a partir de la configuración de red física, normalmente en estructura de malla para la alta disponibilidad, además, transforma una red física con forma de malla, en la que existen bucles, por una red lógica en forma de árbol (libre de bucles). Los puentes se comunican mediante mensajes de configuración llamados *Bridge Protocol Data Units*(BPDU).

El protocolo establece identificadores por puente y elige el que tiene la prioridad más alta (el número más bajo de prioridad numérica), como el puente raíz (*Root Bridge*), normalmente el equipo con mas carga dentro de la red. Este puente raíz establecerá el camino de menor coste para todas las redes y en caso de cambios dentro de la arquitectura de red, el protocolo

RSTP recalcula una nueva topología libre de redundancia dentro de un periodo de tiempo de convergencia.

El tráfico se redigirá desde cada puerto en función de dos parámetros principales: prioridad y dirección MAC.

La prioridad de cada *bridge*, se determina en función del que tiene la prioridad mas alta después del *Root Bridge*, que es el menor valor numérico después del *Root Bridge*. Además, se dispone de un parámetro configurable: el *Span path cost*, que se puede utilizar para redirigir el tráfico a través de los nodos de la configuración.

En caso de obtener el mismo coste en dos puertos, la prioridad se determinara en función del menor valor de MAC.

Para resolver el problema se utiliza el protocolo de routing, *Spanning Tree Protocol* (STP) así como sus variantes.

STP se ha convertido en el protocolo preferido para prevenir bucles de *layer 2* en topologías que incluyen redundancia aunque se pueden encontrar problemas que como: tiempos de convergencia demasiados altos y una configuración que puede complicarse si no se conoce bien el principio de funcionamiento.

- STP (**802.1d**): impide bucles usando un "*timer*".
- Rapid Spanning Tree (RSTP - **802.1w**).

## 2.2 RSTP (*Rapid Spanning Tree Protocol*)

Rapid Spanning Tree Protocol (RSTP) es un protocolo de red de la segunda capa OSI, (nivel de enlace de datos), que gestiona enlaces redundantes. Especificado en IEEE 802.1w, es una evolución del *Spanning tree Protocol* (STP), reemplazándolo en la edición 2004 del 802.1d. RSTP reduce significativamente el tiempo de convergencia de la topología de la red cuando ocurre un cambio en la topología.

Se ha convertido en el protocolo preferido para prevenir bucles de capa 2 en topologías que incluyen redundancia. Además de que el 802.1w contiene mejoras, retiene compatibilidad con su antecesor 802.1D dejando algunos parámetros sin cambiar. Por ejemplo, RSTP mantiene el mismo formato de BPDU que STP sólo que cambia el campo de versión, el cual se le asigna el valor de 2.

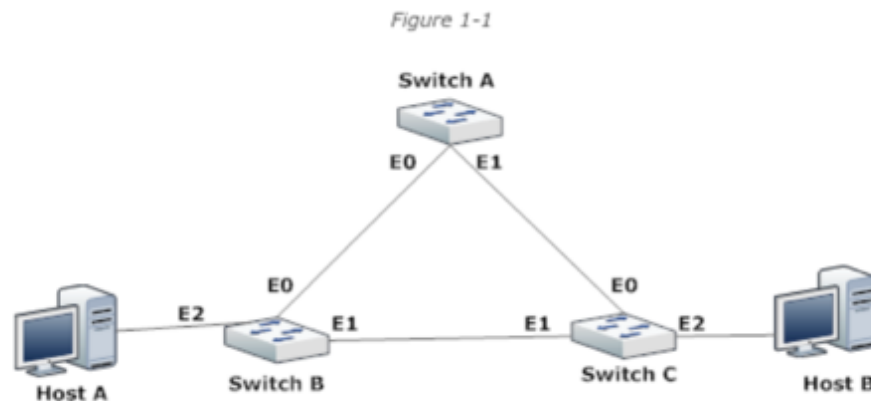
RSTP también define el concepto de *edge-port*, el cual también se menciona en STP como *Port-Fast*, en donde el puerto se configura como tal cuando se sabe que nunca será conectado hacia otro switch de manera que pasa inmediatamente al estado de direccionamiento sin esperar los pasos intermedios del algoritmo –etapas de escucha y aprendizaje– los cuales consumen tiempo. El tipo de enlace es detectado automáticamente, pero puede ser configurado explícitamente para hacer más rápida la convergencia.

## Capítulo 3

### Objetivos

El objetivo de la práctica es configurar y comprobar de primera mano todas las funcionalidades del *Spanning-Tree-Protocol*(STP), así como aprender a resolver los problemas que puedan suceder.

- Montaje de una red básica con 4 dispositivos, que son conectados a otra red.
- Implementar *bridge Rapid-Spanning-Tree-Protocol* (RSTP).
- Comprobar funcionamiento básico y realizar pruebas adicionales.
  - Desconectar una conexión entre dos dispositivos.
  - Añadir una nueva conexión al RSTP.
  - Intentar mejorar la fiabilidad y integridad de la jerarquía implementada inicialmente.
  - Evitar bucles entre puertos de clientes.
- Concluir con una configuración ideal para cada uno de los casos.



**Figura 3.1:** Esquema de una instalación STP (*Spanning-Tree-Protocol*).



## Capítulo 4

### Material

Para la realización el montaje del *Spaning-Tree-Protocol* (STP) se han utilizado dispositivos **RouterBOARD RB9512N**, cableado *ethernet* CAT-5E y equipos portátiles propios para realizar las configuraciones correspondientes.



**Figura 4.1:** RouterBOARD RB9512N

Como enlace a internet, el personal docente ha configurado un **RouterBOARD RB2011UAS**, en modo AP y NAT hacia el exterior.



**Figura 4.2:** RouterBOARD RB2011UAS

## Capítulo 5

# Configuración

En este capítulo se comentan las distintas configuraciones que tendremos en los equipos utilizados, necesarias para poder realizar las pruebas de funcionamiento posteriormente en el capítulo 5, donde veremos diferentes ventajas e inconvenientes relacionados con el protocolo.

### 5.1 Configuración de la red

Se ha decidido crear dos arquitecturas de red diferentes, una básica mas simple, y una ampliada.

Físicamente el montaje de equipos queda de la siguiente forma [figura 5.1](#).

Tendremos un router MikroTik dedicado a conectar con otra red, la cual permitirá el acceso a Internet de los equipos de nuestra LAN. Este dispositivo local, se configurará con el valor mínimo para garantizarle la mejor prioridad, así ejercerá el rol de '*Root Bridge*' sobre el resto de dispositivos. Además, se configurará un servidor DHCP para que atribuya direcciones IP de forma automática a todos los dispositivos interconectados en su red.

Solo tendremos habilitado la interfaz *wireless* en el dispositivo principal, ya que los dispositivos se han interconectado mediante ethernet.



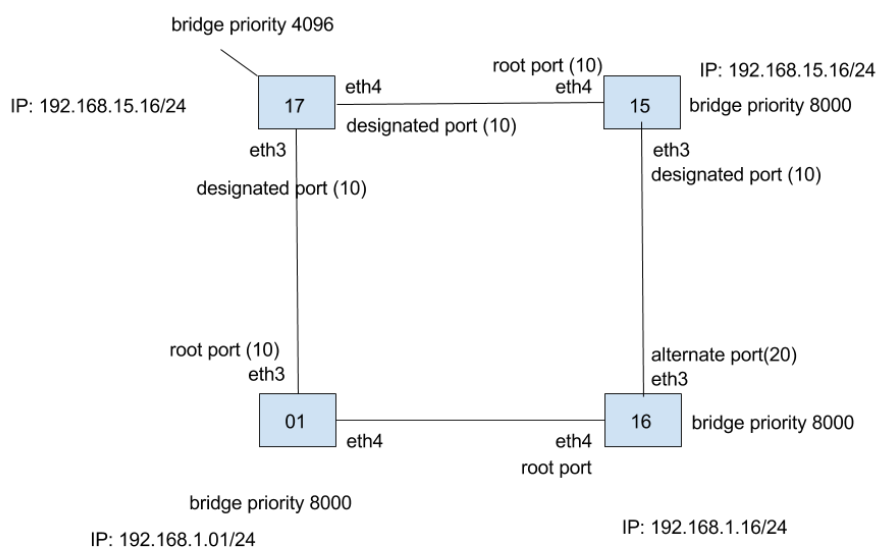
**Figura 5.1:** Fotografía de la instalación

### 5.1.1 Arquitectura básica

Podemos ver en la [figura 5.2](#) el esquema lógico de la conexión de los cuatro equipos MikroTik utilizados.

MACS:

D4:CA:6D:94:68:9D MIKROTIK 01  
 D4:CA:6D:9A:8D:BF MIKROTIK 16  
 D4:CA:6D:9A:8D:C5 MIKROTIK 15  
 D4:CA:6D:9A:8D:C8 MIKROTIK 17



**Figura 5.2:** Montaje básico de STP y detalle

Los equipos portátiles personales utilizados para las pruebas se conectan al puerto ether5 de cada router, para poder acceder a configurar los mismos. Como disponemos de servidor DHCP no será necesario configurar ninguna IP ya que se realiza de forma automática.

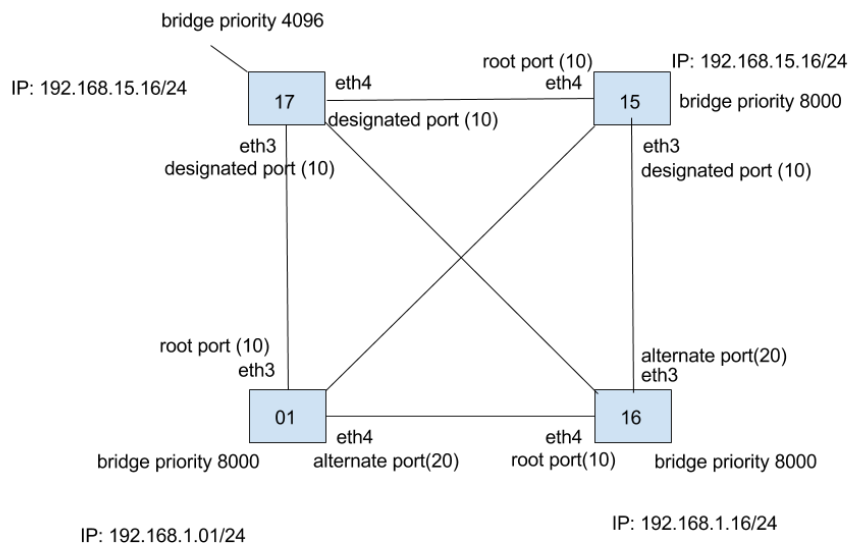
Los equipos portátiles personales se conectan al puerto ether5 de cada equipo, para poder entrar a configurar los mismos, como tenemos servidor DHCP no será necesario configurar ninguna IP.

### 5.1.2 Arquitectura básica + diagonal

Se han añadido dos conexiones adicionales al montaje básico, estableciendo dos nuevos enlaces y incrementando la redundancia en la configuración de la arquitectura para configurar STP. Se puede apreciar el detalle como vemos en la [figura 5.3](#).

MACS:

D4:CA:6D:94:68:9D MIKROTIK 01  
 D4:CA:6D:9A:8D:BF MIKROTIK 16  
 D4:CA:6D:9A:8D:C5 MIKROTIK 15  
 D4:CA:6D:9A:8D:C8 MIKROTIK 17



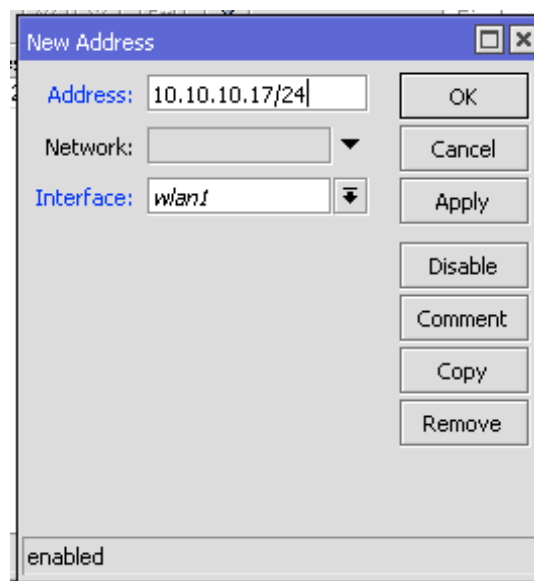
**Figura 5.3:** Montaje básico + diagonal de STP y detalle

## 5.2 Configuración inicial

### 5.2.1 Configuración *Wireless*

En el dispositivo MikroTik17, se ha configurado primero la red *wireless*, con las siguientes opciones:

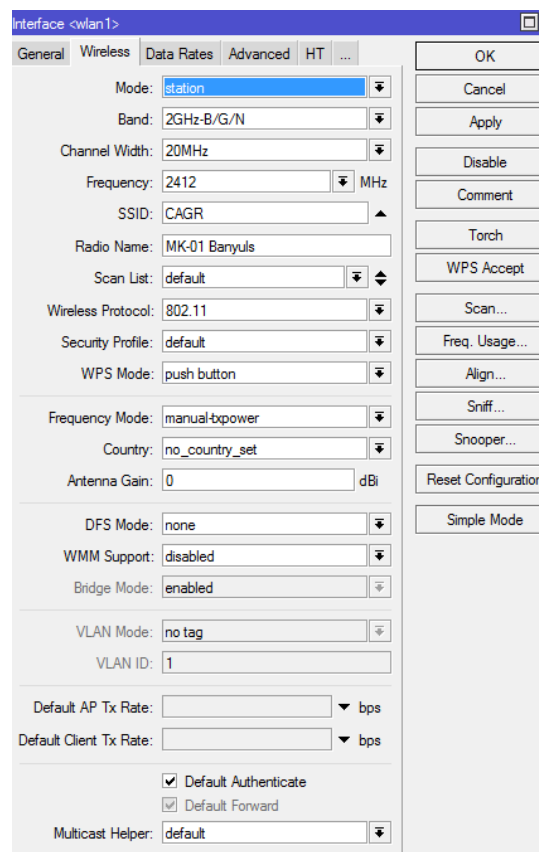
- Interfaz en **wlan1** figura 5.5.
- Perfil de seguridad en figura 5.6.
- IP: 10.10.10.17 en figura 5.4
- *Default gateway* al 10.10.10.100.



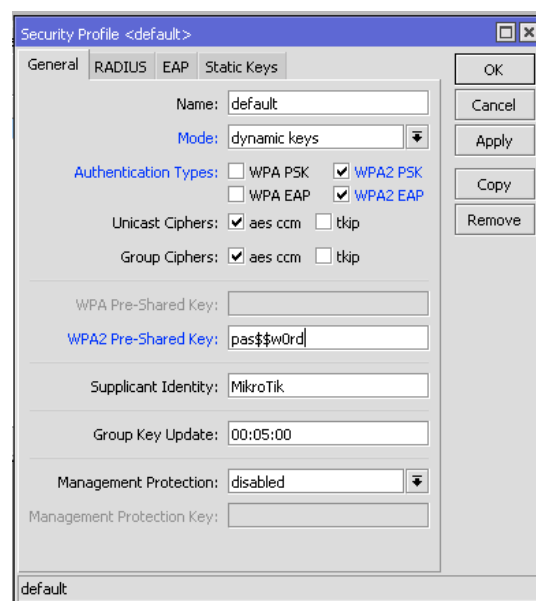
**Figura 5.4:** Configuración de la IP de la interfaz **wlan1**

Seguidamente, se ha configurado la interfaz *wireless* para poder conectarse al *Access-Point* y así disponer de Internet en los distintos equipos cliente.

Se ha modificado además el *security profile* de la red.



**Figura 5.5:** Configurar conexión *wireless* al AP.



**Figura 5.6:** Configuración del *security profile* de la red.

### 5.2.2 Configuración de los otros dispositivos

En el resto de dispositivos, los puertos existente en cada uno de ellos se agruparan como un solo *bridge*, donde se aplicará la configuración del protocolo STP. Además, en el *router* principal, en este caso el MikroTik17, ejecutara un servidor DHCP para la configuración automática de direcciones IP.

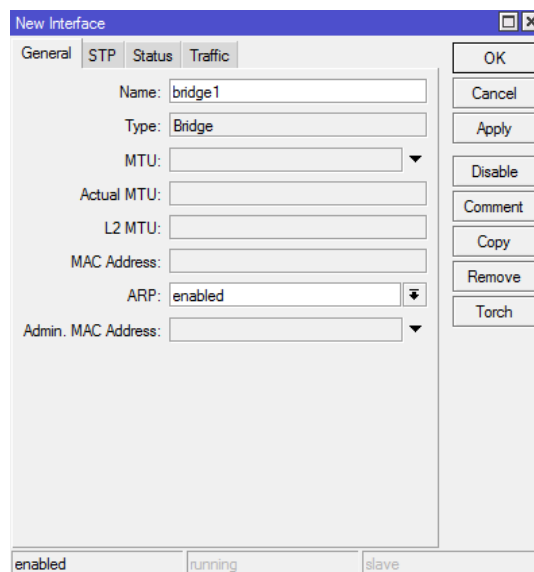
Estamos utilizando los siguientes dispositivos MikroTik proporcionados por el personal docente:

**Tabla 5.1:** Dispositivos utilizados y identificadores

Dispositivo <i>n</i>	<i>Identity</i>	IP	RouterOS Ver.
01	MK01 - Bañuls	192.168.1.1	6.28
15	MK15 - Torres	192.168.1.15	6.28
16	MK16 - Rodriguez	192.168.1.16	6.28
17	MK17 - Ibiza	192.168.1.17	6.32

Se ha utilizado el dispositivo MikroTik17 para conectar a través del AP *wireless* externo, ademas, se encargara del NAT y de dar servicio de DHCP.

Inicialmente, se ha configurado un *bridge* en todos los dispositivos para todas las interfaces *ethernet*.



**Figura 5.7:** Creación del nuevo *bridge*.

Una vez creado el '*bridge*' nos aparecerá como vemos en la [figura 5.8](#).

Accedemos a la pestaña 'Ports' y añadimos las interfaces del Mikrotik al *bridge*.

Despues se le ha asignado una dirección IP al *bridge*, de las comentadas en la [tabla 5.1](#), con esta configuración podremos acceder por IP a los dispositivos.

No se han configurado las rutas por defecto de los otros dispositivos, ya que no es necesaria.

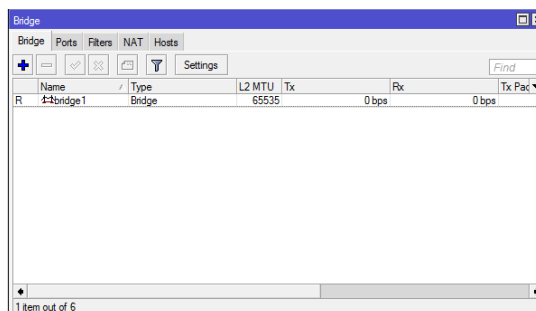


Figura 5.8: Visualización del nuevo *bridge*.

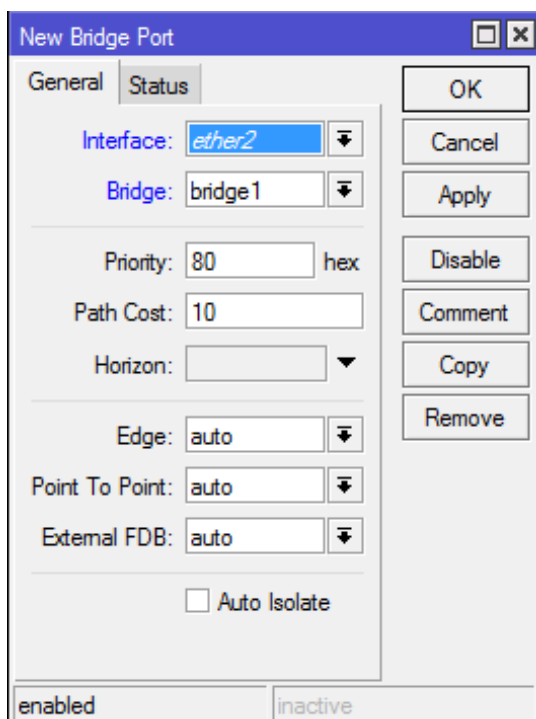
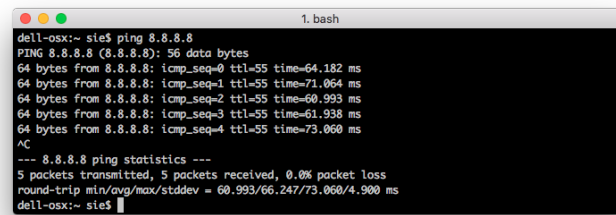


Figura 5.9: Configuración de puertos del *bridge*.

Se ha configurado y levantado también un servidor DHCP en la interfaz *bridge1*, sirviendo IP's desde 192.168.1.100-150, de esta forma, no es necesario configurar ninguna dirección en ningún equipo de la red.

Una vez realizada la conexión entre dispositivos de manera correcta, es posible observar un mal funcionamiento en nuestra red, ya que se pueden haber creado ya bucles que saturan la red y los dispositivos





```

dell-osx:~ sie$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=55 time=64.182 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=71.064 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=60.993 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=61.938 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=55 time=73.060 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 60.993/66.247/73.060/4.900 ms
dell-osx:~ sie$

```

Figura 5.10: Ping al DNS de Google.

## 5.3 Configuración STP / RSTP

El siguiente paso será la configuración propia del protocolo STP. En equipos MikroTik, RouterOS dispone de gran cantidad de herramientas avanzadas de configuración de redes, entre ellas, el protocolo de *routing* STP además de la versión mejorada RSTP.

### 5.3.1 Configuración *out-of-the-box* de RSTP

Partiendo un de un *bridge* creado en cada dispositivo, con IP para poder gestionar los dispositivos desde Winbox, vamos a proceder a configurar el protocolo de RSTP.

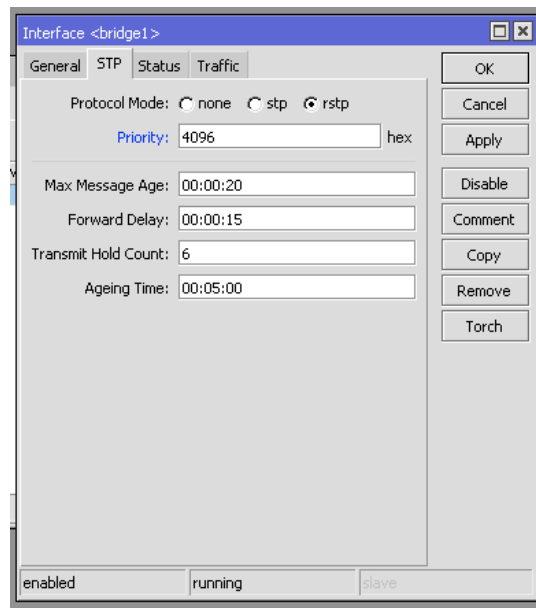
Cabe destacar, que para un funcionamiento adecuado de la red, debemos establecer una prioridad mas alta para un equipo en concreto, este equipo es el que se encarga de enrutar todo el tráfico hacia otra red y hacer NAT.

Accediendo al menú *Bridges* en RouterOS, podemos configurar en cada uno que haga uso del protocolo del caso práctico, el cual se deberá de configurar en todos los *bridges* de cada equipo, donde además, en el *router* que actuará como raíz (*root bridge*), en nuestro caso el MikroTik17, se deberá de modificar el valor *priority* a un valor más bajo para especificar que esta será la "salida" mas prioritaria de nuestra red al exterior.

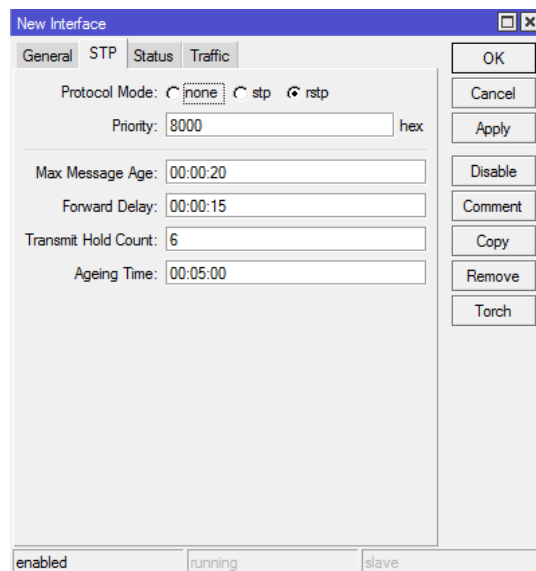
Por lo que en este dispositivo será en el que estableceremos una prioridad manual como vemos en la [figura 5.11](#).

Iremos a la gestión del *bridge* y establecemos el '*Protocol Mode*' como RSTP y la prioridad del *bridge*.

Cuando configuremos todos los *bridges* de los 4 dispositivos routers, deberíamos comprobar que después de un tiempo de convergencia la red funciona de forma estable y que todos los dispositivos pueden tener acceso a Internet, lo más práctico realizando PING sobre cualquier dirección.



**Figura 5.11:** Esableciendo la prioridad del dispositivo principal a 4096.



**Figura 5.12:** Configuración protocolo RSTP y prioridad del *bridge*.

### 5.3.2 Configuración para evitar otro dispositivo con STP no autorizado

Para evitar que cualquier usuario conecte un switch con STP activado y una prioridad más alta que nuestro *bridge*, tenemos que configurar en el puerto de usuario un parámetro para definir que a esa interfaz se va a conectar un extremo, por lo que no será necesario trabajar con STP a partir de ese punto.

En la interfaz en concreto, vamos a establecer el parámetro **Edge=Yes** como vemos en la [figura 5.13](#)

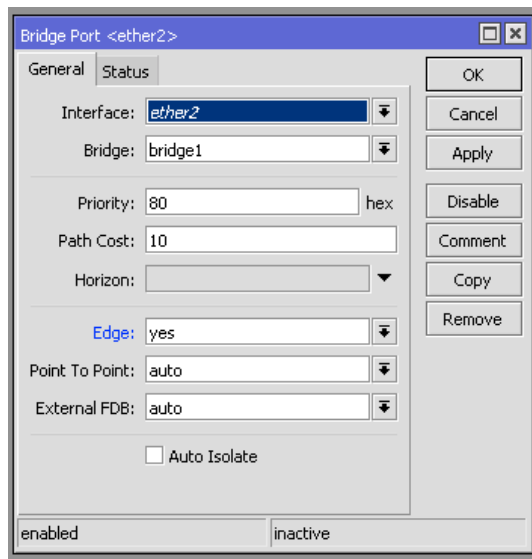


Figura 5.13: Establecer Edge=Yes.

Ahora ya evitamos que el dispositivo conectado a esa interfaz, trabaje por STP. Ya tendríamos el problema solucionado, pero si en dos puertos que tenemos **Edge=YES** creamos un bucle, bloquearemos el dispositivo.

En el punto siguiente, veremos como evitar bucles en la parte del cliente.

### 5.3.3 Configuración para evitar bucles en el clientes

Si creamos un bucle en interfaces Edge=Yes, bloquearemos el dispositivo, ya que se va a generar un bucle infinito de tráfico. Podemos evitar esto modificando unos parámetros.

Activaremos la opción de **horizon**, y estableceremos el mismo número. Evitaremos que se generen bucles, pero no vamos a poder tener comunicación entre estos dos dispositivos, en nuestro caso se ha establecido *Horizon=1* en dos interfaces del equipo, *ether1* y *ether2*, como se puede ver en la [figura 5.14](#).

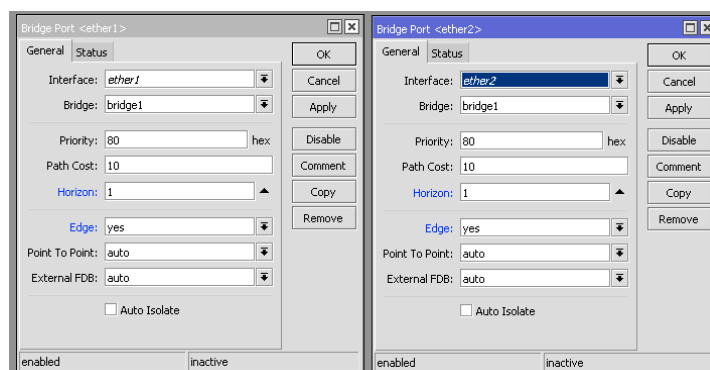


Figura 5.14: Establecer Edge=Yes.

### 5.3.4 Configuración del *path cost*

Una opción interesante en STP es poder establecer el coste de los caminos, con fin de optimizar la red y decidir por donde nos interesa mas llevar el tráfico.

Es posible que tengamos 2 caminos diferentes para llegar a un mismo destino, por lo que podemos configurar la interfaz y establecer un coste del camino (*Path cost*).

En el siguiente ejemplo(figura 5.15), se ha configurado un *path cost* de 99999, por lo que primero se utilizaran caminos con un *path cost* <99999.

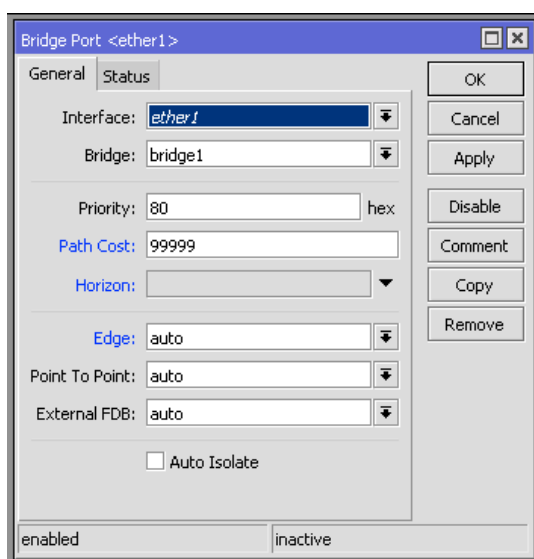


Figura 5.15: Configurando *path cost* muy elevado.

### 5.3.5 Bucle para *Backup Port*

Podemos crear bucles entre puertos configurados como STP, automáticamente el dispositivo asignara dicho puerto como *Backup Port*, en caso de fallar una conexión, seguirá funcionando como *Backup Port*, como vemos en la figura 5.16.

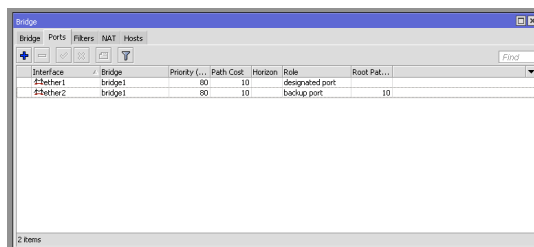


Figura 5.16: Configurando *path cost* muy elevado.

## Capítulo 6

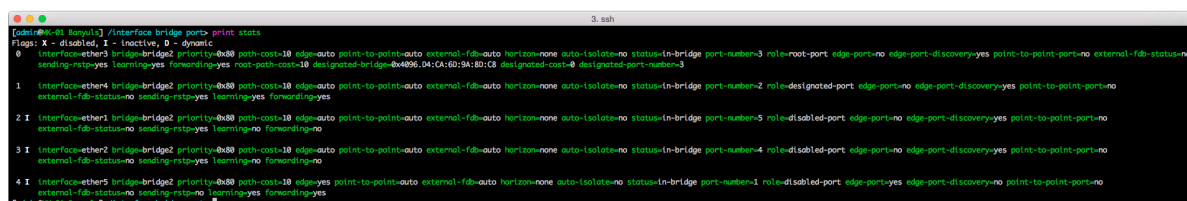
# Pruebas y resultados

### 6.1 Introducción

Se han realizado una serie de pruebas para evaluar el funcionamiento del protocolo *Spanning-Tree-Protocol* (STP), y generar situaciones en las que pueden generarse problemas y malfuncionamientos.

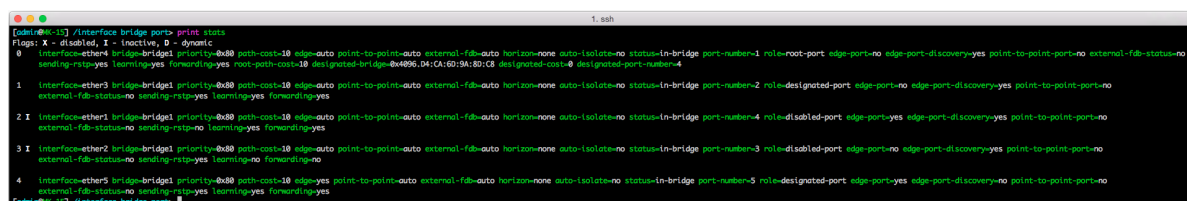
En el [Capítulo 5](#) se han comentado las estructuras de red creadas y las configuraciones utilizadas en cada caso.

La configuración inicial (podemos ver la configuración en [subsección 5.3.1](#)), muestra los siguientes parámetros, mostrados en las figuras [figura 6.3](#), [figura 6.4](#), [figura 6.1](#), [figura 6.2](#).



```
[admin@MK01 Rumpu] /interface bridge port> print state
Flags: X - disabled, I - inactive, D - dynamic
0 Interface-ether3 bridge-bridge2 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-3 role-root-port edge-port-no edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-yes forwarding-yes root-path-cost-10 designated-bridge-04:00:3A:80:C8 designated-cost-0 designated-port-number-3
1 Interface-ether4 bridge-bridge2 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-2 role-designated-port edge-port-no edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-yes forwarding-yes
2 I Interface-ether1 bridge-bridge2 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-5 role-disabled-port edge-port-no edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-no forwarding-no
3 I Interface-ether2 bridge-bridge2 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-4 role-disabled-port edge-port-no edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-no forwarding-no
4 I Interface-ether5 bridge-bridge2 priority-0x30 path-cost-10 edge-yes point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-1 role-disabled-port edge-port-yes edge-port-discovery-no point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-yes forwarding-yes
[admin@MK01 Rumpu] /interface bridge port>
```

Figura 6.1: Vista del estado y parámetros de RSTP en dispositivo MK01



```
[admin@MK15 Rumpu] /interface bridge port> print state
Flags: X - disabled, I - inactive, D - dynamic
0 Interface-ether4 bridge-bridge1 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-1 role-root-port edge-port-no edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-yes forwarding-yes root-path-cost-10 designated-bridge-04:00:3A:80:C8 designated-cost-0 designated-port-number-4
1 Interface-ether3 bridge-bridge1 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-2 role-designated-port edge-port-no edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-yes forwarding-yes
2 I Interface-ether1 bridge-bridge1 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-4 role-disabled-port edge-port-yes edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-yes forwarding-yes
3 I Interface-ether2 bridge-bridge1 priority-0x30 path-cost-10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-3 role-disabled-port edge-port-no edge-port-discovery-yes point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-no forwarding-no
4 Interface-ether5 bridge-bridge1 priority-0x30 path-cost-10 edge-yes point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number-5 role-designated-port edge-port-yes edge-port-discovery-no point-to-point-port-no external-fdb-status-no sending-rstp-yes learning-yes forwarding-yes
[admin@MK15 Rumpu] /interface bridge port>
```

Figura 6.2: Vista del estado y parámetros de RSTP en dispositivo MK15

```

[admin@Mik-16 RouterOS] /interface bridge port> print stats
Flags: I - disabled, I - inactive, D - dynamic
0 interface-ether3 bridge-bridge1 priority=0x80 path-cost=10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=2 role-alternate-port edge-port-no edge-port-discovery=yes point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=yes forwarding-no root-path-cost=20 designated-bridge=0x8000.04:CA:60:94:80:C5 designated-cost=10 designated-port-number=2
1 interface-ether4 bridge-bridge1 priority=0x80 path-cost=10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=1 role-root-port edge-port-no edge-port-discovery=yes point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=yes forwarding=yes root-path-cost=20 designated-bridge=0x8000.04:CA:60:94:80:A1 designated-cost=10 designated-port-number=2
2 I interface-ether2 bridge-bridge1 priority=0x80 path-cost=10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=3 role-disabled-port edge-port-no edge-port-discovery=yes point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=no forwarding-no
3 I interface-ether1 bridge-bridge1 priority=0x80 path-cost=10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=4 role-disabled-port edge-port-no edge-port-discovery=yes point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=no forwarding-no
4 interface-ether5 bridge-bridge1 priority=0x80 path-cost=10 edge-yes point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=5 role-designated-port edge-port-yes edge-port-discovery=no point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=yes forwarding=yes
[admin@Mik-16 RouterOS] /interface bridge port>

```

Figura 6.3: Vista del estado y parámetros de RSTP en dispositivo MK16

```

0 interface-ether3 bridge-bridge1 priority=0x80 path-cost=10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=3 role-designated-port edge-port-no edge-port-discovery=yes point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=yes forwarding=yes
1 interface-ether4 bridge-bridge1 priority=0x80 path-cost=10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=4 role-designated-port edge-port-no edge-port-discovery=yes point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=yes forwarding=yes
2 I interface-ether1 bridge-bridge1 priority=0x80 path-cost=10 edge-no point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=1 role-disabled-port edge-port-no edge-port-discovery=no point-to-point-port-no external-fdb-status-no sending-rstp=no learning=no forwarding-no
3 I interface-ether2 bridge-bridge1 priority=0x80 path-cost=10 edge-no point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=2 role-disabled-port edge-port-yes edge-port-discovery=no point-to-point-port-no external-fdb-status-no sending-rstp=no learning=yes forwarding=yes
4 interface-ether5 bridge-bridge1 priority=0x80 path-cost=10 edge-auto point-to-point-auto external-fdb-auto horizon-none auto-isolate-no status-in-bridge port-number=5 role-designated-port edge-port-yes edge-port-discovery=yes point-to-point-port-no external-fdb-status-no sending-rstp=yes learning=yes forwarding=yes
[admin@Mik-17 Ibiza] /interface bridge port> print count-only
5
[admin@Mik-17 Ibiza] /interface bridge port> print stats

```

Figura 6.4: Vista del estado y parámetros de RSTP en dispositivo MK17

## 6.2 Desactivar el STP en modo Seguro.

Una vez tengamos configurada de forma correcta el protocolo en todos los dispositivos y todos tengan acceso a internet, desactivaremos dentro del modo seguro de RouterOS el protocolo STP para ver cómo se altera el comportamiento normal de la red y se efectúa un bloqueo de los dispositivos al poco tiempo de su funcionamiento.

El modo seguro garantiza que si la configuración aplicada dentro de este modo cierra la sesión SAFE de forma inesperada, se cargará la ultima configuración buena conocida, la anterior con STP, de esta forma, si algún dispositivo entra en saturación por caer dentro de la redundancia de la red, se rectifica a la configuración anterior que era plenamente funcional para poder volver a acceder a cada uno de los routers.

## 6.3 Desconectar un enlace.

Para ver como se reestablece el cauce de datos por la red frente a la caída de un equipo o desconexión de un enlace de red, se realizará la siguiente prueba.

Con los dispositivos plenamente funcionales bajo el protocolo STP, se desconectara el cable que conecta el equipo Mikrotik1 con el Mikrotik16, donde esta conexión es la predeterminada o 'root port' para el dispositivo Mikrotik16, siendo el puerto principal para salida de este dispositivo.

Para comprobar cómo se mantiene la disponibilidad de la conexión a pesar de la desconexión o caída de un equipo colindante, se realizará un PING continuo desde el equipo conectado al MikroTik16, y acto seguido se desconectara el enlace entre los dos routers.

Si todo está correctamente configurado, una vez retiremos el cable, en el terminal donde se está realizando el PING podremos observar una perdida de paquetes o dirección inalcanzable

por unos momentos, pero posteriormente, después de un tiempo de convergencia la conexión se debería de reanudar y continuar con el tráfico anterior de forma normal.

```

Símbolo del sistema
Respuesta desde 8.8.8.8: bytes=32 tiempo=12ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=25ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=21ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=11ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=20ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=28ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=36ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=40ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=23ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=22ms TTL=52
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Respuesta desde 8.8.8.8: bytes=32 tiempo=16ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=20ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=43ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=21ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=17ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=22ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=18ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=21ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=20ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=25ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=22ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=17ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=16ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=17ms TTL=52

```

Figura 6.5: Imagen del ping, devolviendo *timeouts* y recuperando la conectividad

## 6.4 Modificar el valor de *path cost*

El valor Path Cost es un parámetro configurable para establecer el coste que tiene un enlace para ser utilizado. Este parámetro se utiliza en el calculo del recorrido y coste total que seguira el trafico desde el puerto de un dispositivo. De esta manera, la modificación de este parámetro permitirá redirigir el trafico en función de los costes establecidos para cada enlace y asi poder forzar que siga un camino en concreto dentro de la red.

Para probar el funcionamiento de este parametro, vamos a realizar un cambio en el valor de '*Path Cost*' para modificar de esta forma el camino que tomará el trafico para enlazar con el *root bridge*.

El '*Path Cost*' inicial de todos los interfaces es de 10 como podemos ver en la [figura 6.6](#).

Vemos como el puerto '**ether1**' está configurado como '*root port*' i tiene un *Root Path Cost* de 10.

Modificamos el valor del '*Path Cost*' del puerto '**ether1**' y establecemos un '*Path cost*' de 1000, como vemos en la [figura 6.7](#).

Automáticamente se modifican los roles de los puerto donde ahora actúa como '*root port*' el interfaz '**ether4**' que estaba antes como '*alternated port*' debido a que ahora tiene menos coste de enlace.

```
[admin@MK-16 Rodriguez] /interface bridge port> print stats
Flags: X - disabled, I - inactive, D - dynamic
 0 interface=ether3 bridge=bridge1 priority=0x80 path-cost=10 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=no forwarding=no root-path-cost=20 designated-bridge=bridge1
 1 interface=ether4 bridge=bridge1 priority=0x80 path-cost=10 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=yes forwarding=yes root-path-cost=20 designated-bridge=bridge1
 2 I interface=ether2 bridge=bridge1 priority=0x80 path-cost=10 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=no forwarding=no root-path-cost=20 designated-bridge=bridge1
 3 I interface=ether1 bridge=bridge1 priority=0x80 path-cost=10 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=no forwarding=no root-path-cost=20 designated-bridge=bridge1
 4 interface=ether5 bridge=bridge1 priority=0x80 path-cost=10 edge=yes point-to-point=auto external-fdb-status=no sending-rstp=yes learning=yes forwarding=yes root-path-cost=20 designated-bridge=bridge1
[admin@MK-16 Rodriguez] /interface bridge port>
```

Figura 6.6: *Path cost* = 10, por defecto

```
[admin@MK-16 Rodriguez] /interface bridge port> print stats
Flags: X - disabled, I - inactive, D - dynamic
 0 interface=ether3 bridge=bridge1 priority=0x80 path-cost=10 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=no forwarding=no root-path-cost=20 designated-bridge=bridge1
 1 interface=ether4 bridge=bridge1 priority=0x80 path-cost=10 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=yes forwarding=yes root-path-cost=20 designated-bridge=bridge1
 2 I interface=ether2 bridge=bridge1 priority=0x80 path-cost=10 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=no forwarding=no root-path-cost=20 designated-bridge=bridge1
 3 interface=ether1 bridge=bridge1 priority=0x80 path-cost=1000 edge=auto point-to-point=auto external-fdb-status=no sending-rstp=yes learning=no forwarding=no root-path-cost=1000 designated-bridge=bridge1
 4 interface=ether5 bridge=bridge1 priority=0x80 path-cost=10 edge=yes point-to-point=auto external-fdb-status=no sending-rstp=yes learning=yes forwarding=yes root-path-cost=20 designated-bridge=bridge1
[admin@MK-16 Rodriguez] /interface bridge port>
```

Figura 6.7: Cambios en *path cost* a 1000.

## 6.5 Proteger la integridad de la estructura cuando se añada un dispositivo en un extremo de la red configurado con STP (*edge*)

El protocolo STP orientado a routing, basa su funcionamiento en la transmisión de BPDUS donde se informa a los diferentes dispositivos conectados dentro de la red de quien posee mas prioridad para dirigir el trafico. En este caso, la raíz de una configuración STP se define en el 'Root Bridge' donde un dispositivo en concreto es el que dispone de una mayor prioridad sobre los otros, definida esta por un valor en el campo *Priority*.

El problema surge cuando dentro de una arquitectura de red, en un dispositivo del limite de la misma, se conecta otro dispositivo con el protocolo STP configurado y ademas con un valor de prioridad mas bajo que el *Root* de nuestra configuración. Esto genera que se recalcule el árbol



lógico de prioridades y que el nuevo dispositivo conectado redirija el tráfico de la red a través de sí, lo cual supone un grave problema en seguridad e integridad de la red.

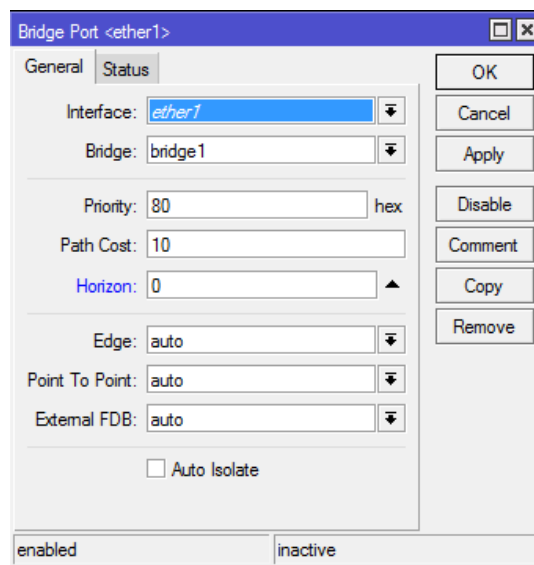
Para evitar este problema, los dispositivos que conforman el extremo de la red considerados como límite, se configuran sobre un parámetro disponible en RouterOS llamado EDGE.

Este parámetro determina que el *bridge* del dispositivo se considere extremo, y a partir de él ya no se reenviarán *BPDUS*, con lo cual, elimina el problema de reestructuración de la red a través de otro dispositivo externo conectado.

Para verificar este problema, se conectará un dispositivo configurado con STP y un valor de prioridad más bajo que el valor de prioridad del root de nuestra red. Realizando la conexión del dispositivo no autorizado a un puerto de un dispositivo extremo, comprobamos que tras un tiempo de convergencia se han reestructurado todos los roles de los bridges de todos los dispositivos, encaminando el tráfico ahora hacia el dispositivo nuevo, considerado ahora 'Root Bridge' por tener el menor valor de prioridad.

## 6.6 Proteger la estructura cuando un posible usuario cree un lazo cerrado entre dos puertos (*horizon*)

Podemos evitar bucles cuando se establezca una conexión mediante un enlace físico utilizando un cable de red y conectando los dos extremos del mismo a diferentes puertos de un mismo dispositivo Mikrotik.



**Figura 6.8:** Configuración valor horizon interfaz *ether1*

Configuraremos ([subsección 5.3.3](#)) todas las interfaces del mismo dispositivo Mikrotik que conecten directamente con los clientes con el mismo valor de *horizon*, como vemos en la imagen [figura 6.8](#).

El resultado es no poder comunicarnos con otro dispositivo con el mismo valor de *horizon* ([figura 6.9](#)).

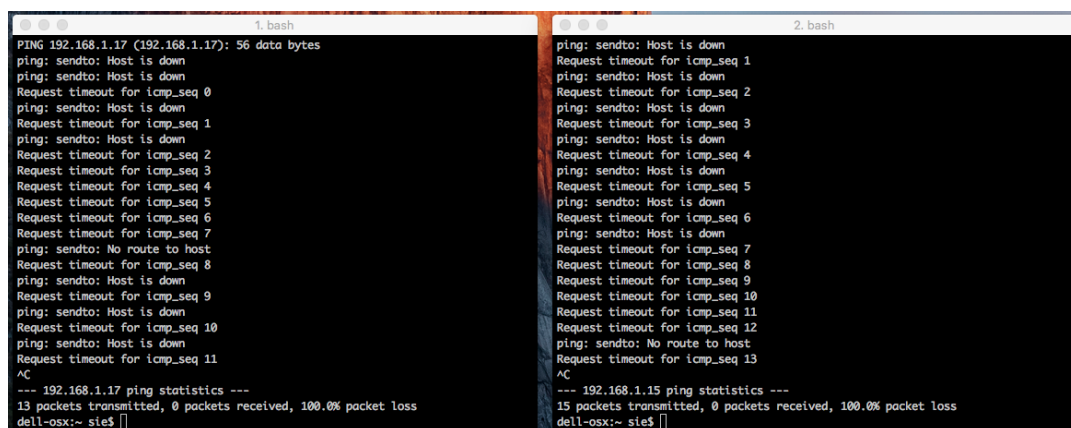
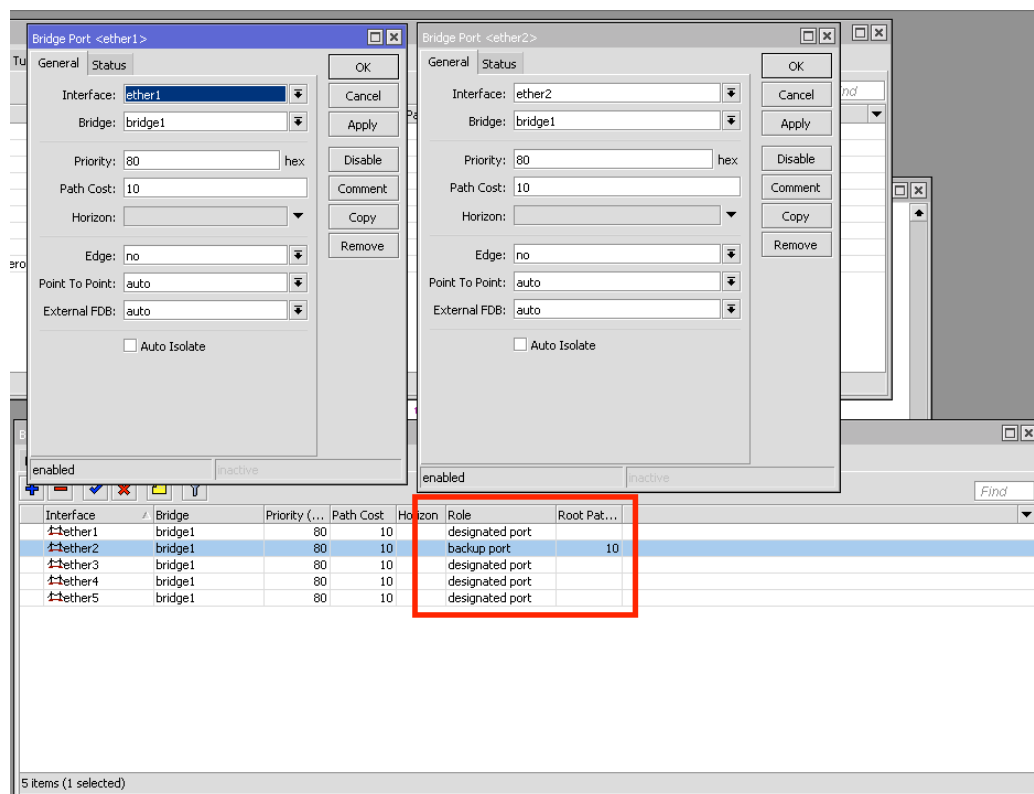


Figura 6.9: Dispositivos incommunicables con mismo horizon.

## 6.7 Añadir enlace para tener redundancia (*Backup Port*)

Si replicamos una conexión entre dos dispositivos previamente ya conectados, que forman parte del conjunto de dispositivos configurados como RSTP y *Edge=No*, automáticamente el dispositivo establecerá el nuevo enlace marcándolo como *backup port*.

Podemos ver el resultado de esta prueba en la [figura 6.10](#).

Figura 6.10: Configuración de *backup port*

## 6.8 Evitar los dos problemas anteriores combinados.

Podemos combinar las soluciones propuestas anteriormente ([sección 6.5](#), [sección 6.6](#)), utilicemos la que utilicemos, vamos a obtener una série de ventajas e inconvenientes.

Si utilizamos **Edge=Yes**, obtendremos más control sobre la integridad de la estructura STP, evitando que si se conecta un dispositivo a la red, y éste implementa STP y además tiene una prioridad menor que el dispositivo root de nuestra red, puede alterar la jerarquía previamente configurada, con esta configuración no vamos a evitar bucles en dichos puertos (**Edge=Yes**).

Para evitar los bucles en puertos (**Edge=Yes**), deberemos establecer un valor en el campo *horizon*, que tiene que ser el mismo para todos los puertos que que queramos aislar, de esta forma, garantizaremos que no se podrán generar bucles, pero, no va a poder existir comunicación entre los dispositivos conectados a estos puertos.

## Capítulo 7

# Conclusiones

Después de realizar la presente práctica hemos entendido los problemas que se encuentran en una red cableada donde coexisten diferentes formas de llegar a un mismo destino, entendiendo como poder solucionar esto mediante la utilización del protocolo (R)STP.

Se ha visto diferentes formas de eliminar problemas que puedan surgir en nuestra red , como la posibilidad de crear bucles mediante la conexión de los extremos de un cable de red en un mismo dispositivo.

Además, poder ofrecer una fiabilidad estructural de la jerarquía de STP, eliminando la posibilidad de que se conecte otro dispositivo con STP implementado y altere el funcionamiento deseado.

Se ha entendido el funcionamiento a nivel práctico del protocolo STP utilizando dispositivos MikroTik hemos podido ver de forma práctica la configuración de los mismos de una forma pragmática.

Es sabido que, si no se implementa el protocolo adecuadamente, este no funcionará correctamente ya que se han realizado diversas pruebas de diferente naturaleza, para comprobar el alcance de cada parámetro sobre la configuración en conjunto.

En otro caso con mas complejidad, con un numero más elevado de nodos, se puede apreciar que aumenta dificultad y se debería elegir, por ejemplo, que prioridad y que caminos se deben elegir para optimizar el funcionamiento de la red.

Así que podemos resumir que en esta práctica hemos aprendido el funcionamiento del protocolo (R)STP sobre dispositivos MikroTik.

# Bibliografía

- [1] Llinares R. (2015) *Documentación de Configuración, administración y gestión de redes*, Departamento de Comunicaciones, Universitat Politècnica de València.
  
- [2] <http://wiki.mikrotik.com/wiki/> *MikroTik Documentation*, MikroTik maintained documentation pages, Mikrotikls Ltd.