Laboratorio de Redes en Ingeniería de Computadores

3º Ing. Informática

Cheuk Kelly Ng Pante alu0101364544@ull.edu.es

Tema 1. Manejo de dispositivos de red

1.1. Introducción

La configuración de los dispositivos se realizará mediante la consola. La consola de MikroTik es accesible a través de distintas interfaces como puertos seriales, telnet, SSH, . . .

1.2. Accesos al Router

- **Acceso a la consola:**

Para acceder a la consola, se utiliza un cable serial, que deberá conectar a la parte trasera del dispositivo.

La conexión se hará mediante el programa `screen` en GNU/Linux, que permite la conexión a terminales a través de una línea serie. Para establecer la conexión deberá conectarse con la orden:

```
screen /dev/ttyUSB0 115200
```

donde el primer argumento es el dispositivo correspondiente al puerto serie que estamos utilizando y el segundo argumento es la velocidad de transmisión en baudios, en este caso 115200.

- **Acceso por telnet:**

Lo habitual es acceder a los dispositivos de forma remota mediante telnet o ssh. Lo que haremos es utilizar el puerto 10 para acceder a la consola del dispositivo mediante telnet. Para ello, debemos asignar una dirección IP a nuestro dispositivo. Esto lo haremos de la siguiente manera:

```
ip address add address=192.168.88.x/24 interface=ether10
```

Debemos sustituir la 'x' por un número diferente en cada router.

A continuación, debemos asignar una dirección IP que se encuentre en la red 192.168.88.0/24 a nuestro PC, conectar un cable Ethernet entre el PC y el puerto 10 del router, y podremos conectar mediante telnet. En el PC tecleamos:

```
telnet 192.168.88.x
```

1.3. Principios básicos del funcionamiento de la consola

Jerarquía de comandos

Debido a que la cantidad de órdenes que permite este fabricante es bastante elevada, los distintos comandos están organizados jerárquicamente.

- Mostrar tabla de enrutamiento del router: `ip route print`
- Volver a la raíz: `/`
- Para subir un nivel se utiliza: `..`

Números de elementos y nombres de elementos

- Lista de interfaces de red: `interface print`
- La columna de más a la izquierda es el número de elementos de cada interfaz dentro de la lista de interfaces. Ahora para modificar la configuración de una interfaz se utiliza el comando `set` y el número de elemento.

```
[admin@MikroTik] > interface set 1 comment="Salida a internet"
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #     NAME                                TYPE       ACTUAL-MTU L2MTU
 0  R  ether1                              ether            1500
 1  R  ;;; Salida a internet
       ether2                              ether            1500
 2  R  ether3                              ether            1500
 3  R  ether4                              ether            1500
 4  R  ether5                              ether            1500
 5  R  ether6                              ether            1500
 6  R  ether7                              ether            1500
 7  R  ether8                              ether            1500
```

El comando anterior ha añadido un comentario al elemento 1 de la lista.
Otra forma de acceder a los elementos de una lista es utilizando sus nombres:

```
[admin@MikroTik] > interface set ether1 comment="Salida a internet"
```

1.4. Configuraciones básicas del router

Modificar el hostname del router

Suele ser conveniente modificar el hostname del router para distinguirlo de los demás cuando se hace un acceso remoto. Se utiliza:

```
[admin@MikroTik] > system identity set name=Router1
```

Establecimiento de la contraseña de acceso

Es un entorno de operación real, es necesario asignar una contraseña de acceso al dispositivo. Para fijar la contraseña del usuario *admin* al valor *class*, utilizamos:

```
[admin@MikroTik] > user set admin password=class
```

Asignación de dirección a una interfaz

La asignación de dirección a una interfaz se realiza mediante el comando `ip address`:

```
[admin@MikroTik] > ip address add adress=10.0.0.1/24 interface=ether1
```

Para eliminar una de las direcciones asignadas se utiliza el comando `remove`

```
[admin@MikroTik] > ip address remove numbers=1
```

donde el 1 es el número de elemento se va a eliminar según se muestra en el comando `print`.

Añadir una ruta estática a la tabla de enrutamiento

Para añadir una ruta estática a la tabla de enrutamiento se utiliza el comando `ip route`:

```
[admin@MikroTik] > ip route add dst-address=10.2.0.0/24 gateway=10.0.0.1
```

La red de destino se especifica con el parámetro `dst-address` y, en este caso, toma el valor `10.2.0.0/24`, mientras que la puerta de enlace se especifica el parámetro `gateway` y, en este ejemplo, toma el valor `10.0.0.1`

Guardar la configuración

En RouterOS, la configuración queda guardada según se va modificando. No es necesario guardarlo de manera explícita mediante un comando.

Volver a la configuración de fábrica

Para volver a la configuración de fábrica inicial se teclea la orden:

```
[admin@MikroTik] > system reset-configuration no defaults=yes keep-users=no
```

1.5. Configuración básica de switches

Introducción al switch TP-Link T2500G-10TS

Este switch dispone de 10 puertos en total. Ocho de ellos son puertos Gigabit Ethernet por par trenzado y dos son interfaces de fibra óptica.

La primera conexión al switch se hace mediante el cable de consola, utilizando el programa `screen`. La velocidad de la línea será de 38400 baudios.

```
screen /dev/ttyUSB0 38400
```

A partir de aquí tenemos acceso a la configuración del dispositivo, pero para realizar cambios en la configuración deberíamos cambiar al modo de configuración con el comando: `configure`

Resetear la configuración del dispositivo

Para volver a la configuración de fábrica del dispositivo se utiliza el comando `reset`

Revisar la configuración del dispositivo

Para mostrar la configuración completa del dispositivo se utiliza el comando `show running-config` estando en modo privilegiado

Guardar la configuración en ejecución

El switch no guarda la configuración en ejecución automáticamente. Esto deberá hacerse mediante el siguiente comando:

```
T2500G-10TS# copy running-config startup-config
```

Fijar el nombre del dispositivo

Es conveniente fijar un nombre de dispositivo para poder identificarlo cuando se accede de forma remota. Se hará en modo *config* a través del comando: `hostname S1`

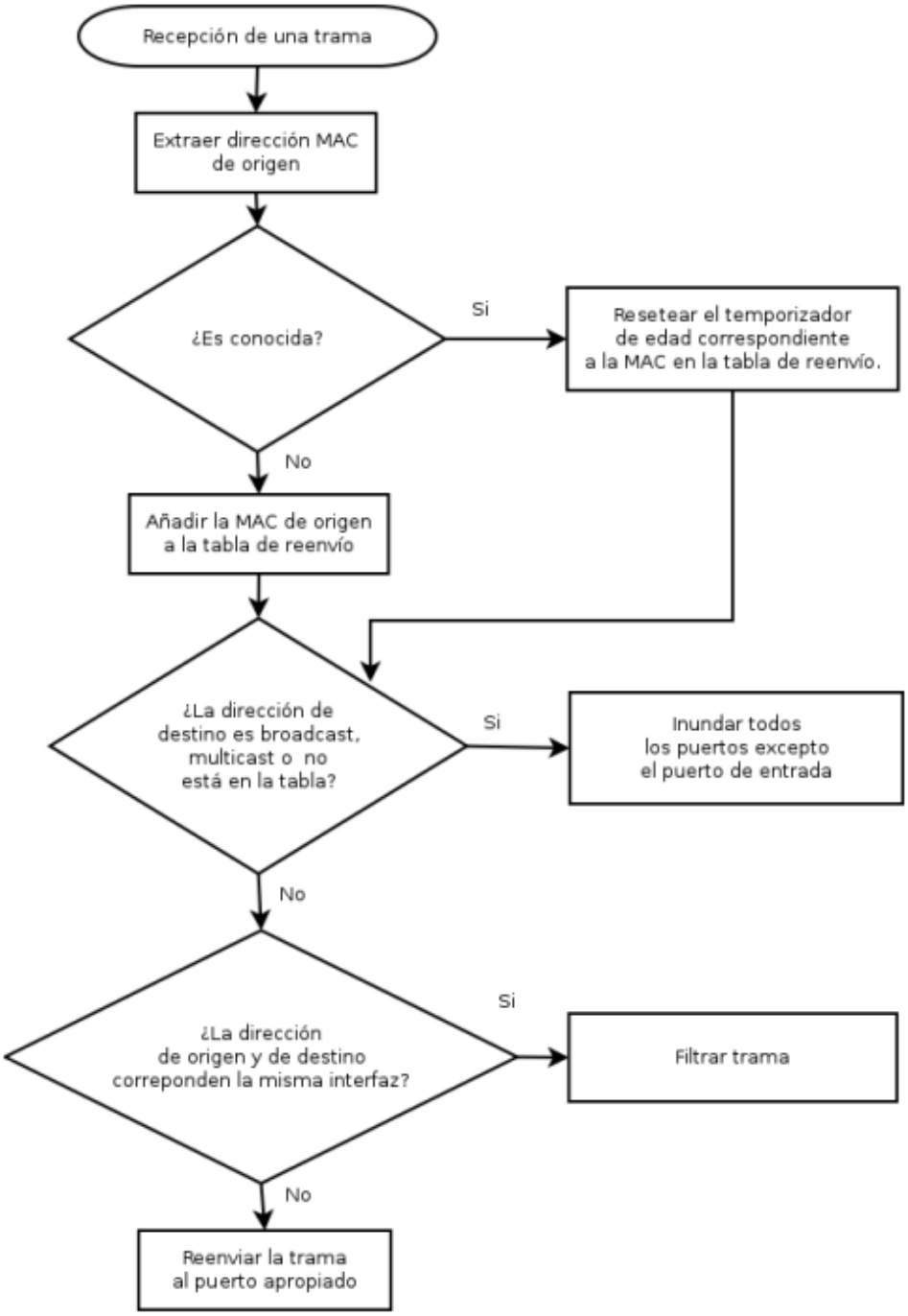
Tema 2. VLANs y enrutamiento entre VLAN

2.1. Introducción

Un switch es un dispositivo de la capa 2 del modelo OSI que opera como un *bridge* transparente. Su cometido es la conmutación de tramas entre los dispositivos conectados utilizando las direcciones MAC. Cada switch mantiene una tabla de conmutación como la siguiente:

MAC Address	Puerto de Salida	Edad (minutos)

La tabla de conmutación de un switch no es conocida, sino que se rellena mediante un proceso de aprendizaje.



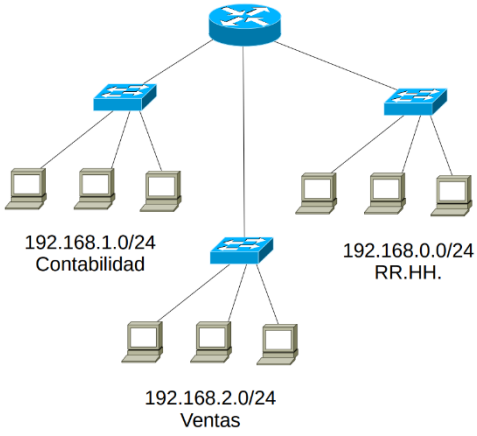
Ethernet es un medio compartido, se produce un problema de rendimiento cuando hay muchos dispositivos conectados a la LAN mediante un hub. Los dominios de colisión son áreas de una LAN en la que las estaciones compiten por el acceso al medio físico compartido. Al utilizar un switch cada host emisor y receptor forman un dominio de colisión separado solucionándose así el problema de rendimiento y escalabilidad.

Para solucionar los problemas de rendimiento se segmenta la LAN separando un dominio de colisión grande en varios de menor tamaño. Esto permite optimizar el ancho de banda ya que en cada domino de colisión habrá un menor número de hosts que compiten por el acceso al medio compartido. Esta separación es la que se consigue mediante un bridge. En una red conmutada con switches, cada puerto representa un dominio de colisión separado si se utiliza una comunicación half-duplex mientras que si se emplea full-duplex no se producen colisiones.

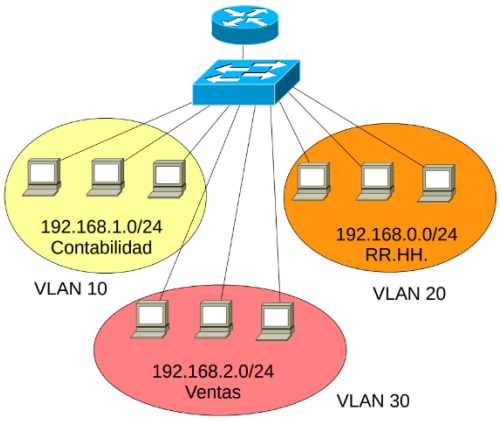
VLAN

Las VLAN o también conocidas como «**Virtual LAN**» nos permite crear redes lógicamente independientes dentro de la misma red física, haciendo uso de switches gestionables que soportan VLANs para segmentar adecuadamente la red. También es muy importante que los routers que utilicemos soportan VLAN, de lo contrario, no podremos gestionarlas todas ni permitir o denegar la comunicación entre ellas.

Configuración utilizando VLANs: La topología no coincide con la topología física:

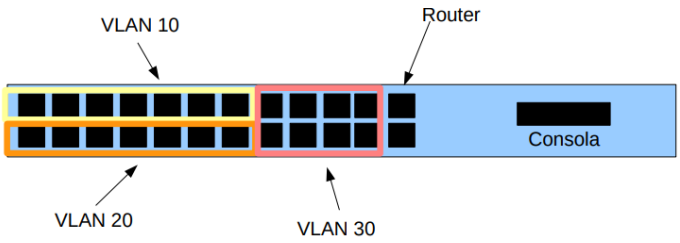


Topología Física

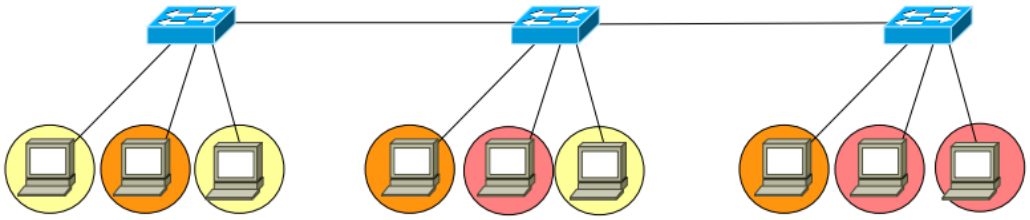


Topología Lógica

Los swiches permiten separar una LAN en múltiples dominios de broadcast utilizando VLANs (Virtual LANs). Las VLANs consisten en asignar los puertos del switch a distintos dominios de broadcast generando distintas redes que comparten hardware. Un dispositivo conectado a un switch con VLANs sólo podrá conectarse con otros dispositivos asignados a la misma VLAN, es decir, de su mismo dominio de broadcast. Como las VLANs se comportan como redes físicamente separadas, la interconexión de las mismas se hace mediante un router.



La naturaleza de las VLANs permite que se puedan formar varias VLANs a partir de distintos segmentos LAN ubicados en lugares físicamente separados.



Enlaces troncales

Con un único switch que el tráfico de las distintas VLANs permanezca separado es sencillo.

Al conectar varios switches que soportan distintas VLANs es necesario etiquetar las tramas que se transmiten por los enlaces troncales con su identificador de VLAN, de modo que en el switch de destino sea posible conmutar a la VLAN correcta.

Al definir un puerto como troncal se pueden distinguir dos tipos de tramas:

- Tramas etiquetadas: Aquellas cuyas tramas se envían con el VID utilizando IEEE 802.1Q u otro protocolo de encapsulamiento para trunks.
- Tramas de la VLAN nativa: Las tramas de esta VLAN se envían de forma nativa, es decir, sin VID. Sólo puede existir una única VLAN nativa para el mismo enlace troncal.

Definición del tipo de puerto

- Puertos de acceso: Este tipo de puertos son los que se conectan host finales. Trabajan con las tramas clásicas de Ethernet, sin el agregado de las etiquetas de VLAN.
- Puertos troncales (trunk): Los puertos de troncales tienen una función especial que es la de conectar switches entre sí o un switch con un router. Cuando llega tráfico a un puerto de trunk proveniente desde el propio switch, éste es etiquetado con el identificador de VLAN y enviado por el puerto. El equipo que lo recibe, desencapsula la trama Ethernet y lo envía al puerto que corresponda.

Enlaces agregados

Los switches permiten agrupar enlaces para formar un enlace lógico de mayor ancho de banda.

- Agregación de enlaces (link aggregation)
- Bonding
- Trunking

2.2. Manejo de interfaces

- Para el estado de las distintas interfaces del switch: `show interface status`.
- Para configurar cualquier aspecto relacionado con una interfaz debemos pasar a modo de configuración y seleccionarla: `interface gigabitEthernet 1/0/1`
- Rango de interfaces para aplicarles la misma configuración mediante `interface range gigabitEthernet 1/0/1-5`

2.3. Manejo de VLANs

- Para mostrar las VLANs, su nombre y los puertos que tienen asociadas: `show vlan`
- Para crear una nueva VLAN se debe pasar a modo de configuración:
  ```
  S1(config)# vlan 99
  S1(config-vlan)# name management
  ```

  Con estos pasos se ha creado la VLAN con identificador 99 y se le ha asignado un nombre, en este caso *management*.
- Si deseamos eliminar una VLAN: `S1(config)# no vlan 10`

2.4. Añadir puertos a una VLAN

2.4.1. Puertos de acceso

Los puertos de acceso son aquellos destinados a conectar dispositivos y, por tanto, transportan tramas sin etiquetar. Por ejemplo, para hacer que el puerto Gi1/0/7 sea de acceso perteneciente a la VLAN 10:

```
S1(config)# interface Gi1/0/7
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 10
S1(config-if)# no shutdown
```

2.4.2. Puertos troncales

Los puertos troncales sirven para interconectar switches con switches o switches con routers. Las tramas transmitidas por este tipo de puertos son tramas etiquetadas. Por ejemplo, para hacer que el puerto Gi1/0/1 sea troncal y permita tramas de la VLAN 10,20 y 99:

```
S1(config)# interface Gi1/0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk allowed vlan 10,20,99
S1(config-if)# no shutdown
```

2.4.3. Puertos generales

Los puertos generales tienen las mismas reglas de ingreso que los puertos generales, es decir, que se permiten tramas no etiquetadas de una única VLAN que se especifica en el PVID del puerto. Sólo admite tramas etiquetadas y no etiquetadas que estén en la lista de VLAN admitidas por el puerto. Por otra parte, las tramas que egresan del puerto quedan controladas mediante las reglas de egreso en las que se indica si eran etiquetadas o no etiquetadas. Para pasar un puerto a modo general:

```
S1(config)# interface Gi1/0/1
S1(config-if)# switchport mode general
S1(config-if)# no shutdown
```

Comando válido solo para (TS-2500G-10TS-V1), la siguiente secuencia de comandos configura el PVID del puerto a una VLAN:

```
S1(config)# interface Gi1/0/1-3
S1(config-if)# switchport pvid 99
S1(config-if)# end
```

- El PVID es el identificador de la VLAN nativa del puerto, es decir, aquella que en la que se categorizan las tramas no etiquetadas que llegan al puerto.
- Para poder recibir y emitir tramas etiquetadas en el puerto debemos añadirlas a la lista de VLAN permitidas con la regla de egreso para tramas etiquetadas.

Por ejemplo, la siguiente secuencia de comandos añade las VLAN 10,20 con la regla de egreso de tramas etiquetadas a la lista de VLAN permitidas de los puertos indicados:

```
S1(config)# interface Gi1/0/1-3
S1(config-if)# switchport general allowed vlan 10,20 tagged
S1(config-if)# end
```

Si queremos que las tramas de una VLAN se emitan como no etiquetadas:

```
S1(config)# interface Gi1/0/1-3
S1(config-if)# switchport general allowed vlan 99 untagged
S1(config-if)# end
```

Si ejecutamos las tres secuencias de comandos anteriores tendríamos lo que tendremos es el equivalente a un puerto troncal que permite las VLAN 10,20 y 99, transmitiéndose las tramas de las VLAN 10 y 20 de forma etiquetada mientras que las de la VLAN 99 de forma no etiquetada (VLAN nativa).

Si hubiésemos puesto esta otra secuencia de comandos, sin permitir las otras VLAN etiquetadas, tendríamos un comportamiento similar al puerto de acceso para la VLAN 99, descrito anteriormente:

```
S1(config)# interface Gi1/0/1-3
S1(config-if)# switchport pvid 99
S1(config-if)# switchport general allowed vlan 99 untagged
S1(config-if)# end
```

2.5. VLAN de gestión

Es conveniente que los dispositivos puedan accederse de manera remota. Para eso es necesario que tengan activa una interfaz de gestión que pertenece a una VLAN de gestión. Para configurar la VLAN de gestión se usa el comando `ip management-vlan` y se indica el identificador de la VLAN de gestión. Además, debemos asignar la dirección IP a la interfaz de gestión dentro de esta VLAN:

```
S2(config)#ip management-vlan 99
S1(config)#interface vlan 99
S1(config-if)#ip address 10.10.10.3 255.255.255.0
```

2.6. Enrutamiento entre VLANs con routers MikroTik

Para que exista conectividad entre la VLAN los paquetes deberían pasar por un router, ya que están cambiando de red. Para que el router pueda recibir paquetes de ambas VLANs deberá tener asignada una dirección en cada una de ellas. Utilizaremos la interfaz ether1 para conectar con el switch y crearemos dos interfaces virtuales dependientes de ether1, una en cada VLAN:

```
[admin@MikroTik] > interface vlan add interface=ether1 name=ether1.10
vlan-id=10
[admin@MikroTik] > interface vlan add interface=ether1 name=ether1.20
vlan-id=20
[admin@MikroTik] > interface vlan add interface=ether1 name=ether1.99
vlan-id=99
```

El argumento `vlan-id` indica el identificador de VLAN a esperar y el argumento `name` es un nombre que utilizaremos para identificar la interfaz virtual en procesos de configuración posteriores

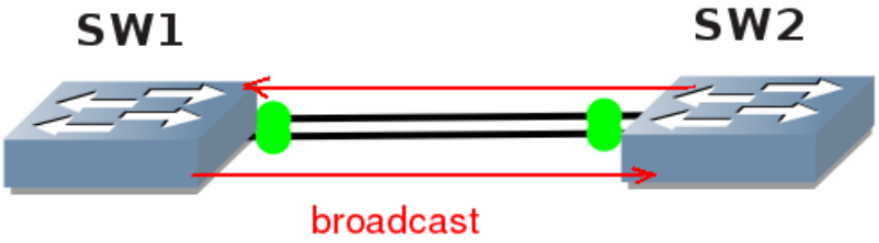
Asignar la dirección al router dentro de cada una de las interfaces virtuales dentro de su VLAN:

```
[admin@MikroTik] > ip address add address=192.168.0.1/24 interface=ether1.10
[admin@MikroTik] > ip address add address=192.168.1.1/24 interface=ether1.20
[admin@MikroTik] > ip address add address=10.10.10.1/24 interface=ether1.99
```

Tema 3. Protocolos Spanning Tree

3.1. Introducción

Los enlaces redundantes pueden ser una buena idea, pero también son una fuente de problemas. Esto es porque se puede producir lo que se denominan tormentas de broadcast. Si no se implementa un mecanismo de prevención de bucles, cuando se produce una inundación, debido a un broadcast, la trama pasa de un switch a otro a través del enlace redundante y el switch receptor, al tratarse de un broadcast, reproduce la trama inundando nuevamente, con lo que la trama vuelve al primer switch, donde se produce una nueva inundación, y así sucesivamente.



El protocolo *Spanning Tree (STP)* es un protocolo que asegura que no se produzcan bucles en una LAN con switches. Su función básica es prevenir las tormentas de difusión resultantes de los bucles. Tiene las siguientes características:

- Bloquea enlaces con el fin de prevenir bucles en una red Ethernet. Los enlaces bloqueados se pueden activar en el caso de que fallen los enlaces activos.
- El árbol de expansión se recalcula automáticamente cuando se produce un cambio en la topología, pero a costa de una parada momentánea de los reenvíos cuya duración depende de la variante del protocolo utilizada.

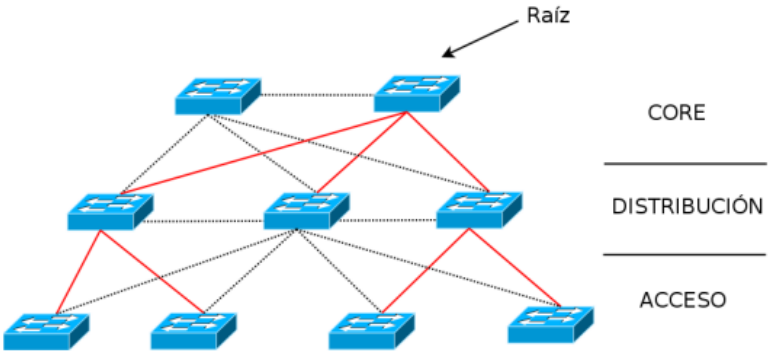
Existen algunas variantes del protocolo STP original, que tratan de reducir los tiempos necesarios para recalcular el árbol de expansión o generador.

- *RSTP (Rapid Spanning Tree Protocol)*: es una evolución para reducir el tiempo de parada de la red cuando hay que recalcular el árbol de expansión.
- *MSTP (Multiple Spanning Tree Protocol)*: es una evolución del protocolo *RSTP*, que permite mantener un árbol de expansión distinto para distintas asociaciones de VLANs, lo que permite, utilizar los enlaces redundantes para cada asociación de VLANs, en vez de desactivarlos completamente aprovechando mejor el ancho de banda.

Cada uno de los switches del spanning tree quedan identificados mediante un identificador de puente (*BID – Bridge Identifier*). Este identificador está conformado por dos partes: una prioridad asignada al switch y la dirección MAC del switch:

El BID, aparte de identificar a cada uno de los switches que forman parte del spanning tree, se utiliza para establecer el puente raíz del spanning tree. Una vez se inician los switches que forman parte de la red, intercambian una serie de BPDUs (Bridge Protocol Data Units) que, entre otros datos, llevan el BID de cada switch. De esta forma,

cada switch compara el BID de la BPDU entrante con su propio BID y establece como switch raíz aquel con BID menor. Después del proceso de intercambio de BPDUs el switch con menor BID se convierte en raíz. Para controlar qué switch ejerce de puente raíz el administrador de red deberá modificar las prioridades que por defecto están establecidas a 32768.



3.2. Configurar el protocolo Spanning Tree (STP)

```
S1# spanning-tree
S1# spanning-tree mode stp
```

Además, deberá seleccionar los puertos troncales y activar el protocolo de los mismos:

```
S1(config)# interface range gi 1/0/1-4
S1(config-if-range)# spanning-tree
```

3.3. Comandos sobre STP

- `S1(config)# show spanning-tree active`
- `S1(config)# show spanning-tree interface`
- Cambiar la prioridad para que otro switch se convierta en nodo raíz:
  `SX(config)# spanning-tree priority 0`

3.4. Cambiar los tipos de STP en los switches

- Protocolo *RSTP*:
  ```
  Switch(config)#spanning-tree mode rstp
  Switch(config)#spanning-tree
  ```
- Protocolo *MSTP*:
  ```
  Switch(config)# spanning-tree mode mstp
  Switch(config)#spanning-tree mst configuration
  Switch(config-mst)#name 1
  Switch(config-mst)#revision 100
  Switch(config-mst)#instance 1 vlan 99
  Switch(config-mst)#instance 2 vlan 10,20
  ```

Los comandos anteriores cambian el modo de spanning tree para que utilice MSTP. Luego se asigna el nombre de la región y la revisión. A continuación, se crean dos instancias indicando las VLAN asociadas

Forzar raíz en MSTP: `SX(config)#spanning-tree mst instance 2 priority 0`
Ver estado de MSTP: `SX(config)#show spanning-tree mst instance 2`

Tema 4. Enrutamiento dinámico con OSPF

4.1. Introducción

OSPF es un protocolo de enrutamiento de estado de enlace. Este tipo de protocolos cada nodo:

- Mide el estado de los enlaces con sus vecinos.
- Asigna un coste a los enlaces con sus vecinos (por administrador)
    - Número de saltos.
    - Pesos inversamente proporcionales al caudal.
- Cada nodo inunda la red con esta información.
- Todos los nodos conocen el grafo completo de la topología de red.
- Cada nodo calcula la ruta de coste mínimo entre él mismo y todos los posibles destinos de la red, utilizando el algoritmo de Dijkstra.
- Actualiza estado cada 30 min o en cambios.

OSPF agrupa las redes de un sistema autónomo en áreas. Cada una de las áreas mantiene una *base de datos de estado de enlace*, que es un grafo cuyos nodos son los routers y las redes agrupadas en el área. Si un router pertenece a más de un área se dice que es un router fronterizo de área (Area Border Router, ABR). Durante el enrutamiento se pueden producir dos situaciones. Si el origen y el destino pertenecen al mismo área, se utiliza la base de datos de estado de enlace del área. Si las áreas del origen y el destino son distintas, la ruta puede dividirse en tres tramos:

- El tramo desde el origen hasta el router fronterizo de área
- El tramo de red troncal entre el ABR del área de origen y el ABR del área de destino.
- El tramo desde el ABR del área destino hasta la red de destino.

Todas las redes OSPF deben contener al menos un área troncal, el área 0.0.0.0, que se encarga de enrutar el tráfico entre áreas no troncales. Todos los routers fronterizos de áreas deben pertenecer al área troncal.

En OSPF se distinguen cuatro tipos de routers:

- **Router interno**: Todas las redes directamente conectadas pertenecen al mismo área.
- **Router fronterizo de área (ABR)**: Es un router conectado a varias áreas.
- **Routers troncales**: Tienen una interfaz conectada a una red del área troncal. Todos los routers fronterizos de área son también routers troncales, pero no al revés.
- **Router fronterizo del sistema autónomo (ASBR)**: Es un router que intercambia información con otro sistema autónomo.

Asimismo, las interfaces OSPF reconocen tres tipos de redes:

- **Broadcast**: No se puede saber de antemano cuantos routers hay conectados. Las redes con tecnología Ethernet son un ejemplo de red broadcast.
- **Punto-a-punto**: Son redes en las que no sólo participan dos nodos.
- **Redes de acceso multiple sin capacidad broadcast (NBMA)**: Son redes con tecnología X.25, Frame Relay o ATM.

Para calcular los costes OSPF utiliza un coste de referencia que por defecto es de 10000Kbps. El coste de cada enlace se calcula:

$$Coste = \left[\frac{Coste\ de\ referencia}{Ancho\ de\ banda\ del\ enlace\ en\ Kbps}\right]$$

El coste de referencia se puede modificar a nivel de router y el ancho de banda se debe configurar para cada interfaz de red.

4.2. Tipos de Anuncios de Estado de Enlace (LSA)

Tipo LS	Nombre LS	Generado por	Distancia	Descripción LSA
1	Router-LSA	Cada router interno dentro de un área	Área-local	Originado por todos los enrutadores. El ID de estado de enlace del LSA de tipo 1 es el ID del enrutador de origen.
2	Network-LSA	Router designado	Área-local	Este LSA contiene la lista de enrutadores conectados a la red. El ID de estado de enlace del LSA de tipo 2 es la dirección de interfaz IP del DR.
3	Summary-LSA	Area Border Router (ABR)	Dominio de enrutamiento	Un enrutador de borde de área (ABR) toma la información que ha aprendido en una de sus áreas adjuntas y la resume antes de enviarla a otras áreas a las que está conectado. Cada resumen-LSA describe una ruta a un destino fuera del área, pero aún dentro del AS (es decir, una ruta entre áreas). Este resumen ayuda a proporcionar escalabilidad al eliminar la información de topología detallada para otras áreas, ya que su información de enrutamiento se resume en solo un prefijo de dirección y una métrica.
4	ASBR-Router	Area Border Router (ABR)	Dominio de enrutamiento	Esto es necesario porque los LSA externos de tipo 5 se inundan en todas las áreas con el origen como ID de enrutador de límite de sistema autónomo (ASBR), pero las ID de enrutador no se anuncian entre áreas. Esto se soluciona con un Area Border Router que inunda la información del ASBR donde se originó el tipo 5. El ID de estado de enlace es el ID de enrutador del ASBR descrito para los LSA de tipo 4.

5	ASBR-Summary	Enrutador del límite del Sistema Autónomo	Dominio de enrutamiento	Estos LSA contienen información importada a OSPF desde otros procesos de enrutamiento. Se inundan todas las áreas sin cambios (excepto las áreas stub y NSSA).
7	NSSA-Summary	El ASBR dentro de un NSSA	Intra-area	Los LSA tipo 7 son idénticos a los LSA tipo 5
				Los LSA de tipo 7 solo se inundan dentro de la NSSA.
				Esto permite que los enrutadores en NSSA envíen información de enrutamiento externo para su redistribución. Utilizan LSA de tipo 7 para informar a los ABR sobre estas rutas externas, que el enrutador de borde de área luego traduce a LSA externas de tipo 5 e inunda el resto de la red OSPF de manera normal
				En el enrutador de borde de área, los LSA de tipo 7 seleccionados se traducen en LSA de tipo 5 y se inundan en la red troncal.

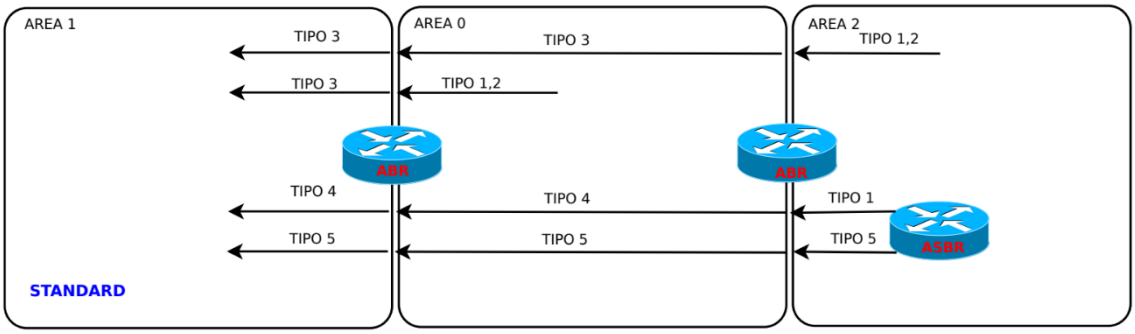
4.3. Tipos de Áreas OSPF

4.3.1. Área Troncal o Backbone

El área *troncal* o *backbone* es el área principal de una topología OSPF. **Se etiqueta como área 0 y debe existir en todas las redes**.Tiene como misión interconectar otras áreas y por tanto permite mensajes de tipo 1 y 2 internos a la misma. Además, permite la entrada de mensajes de tipo 3, 4 y 5 de las demás áreas (no así el mensaje de tipo 7). El área troncal es la que está identificada mediante el identificador de área 0 y debe existir en toda tipología OSPF.

4.3.2. Área standar

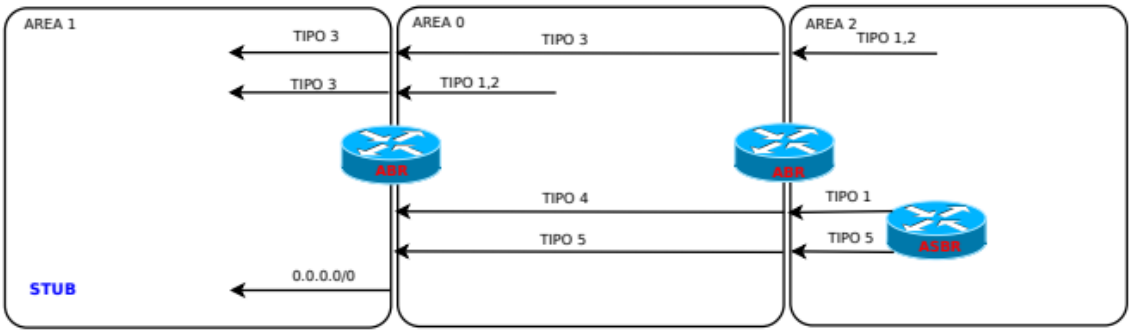
Las áreas *stantard* son el tipo de área por defecto. Admiten la entrada de mensajes de tipo 3, 4 y 5 a través de sus routers frontera. Esto implica que los routers internos a este tipo de áreas tiene la misma información detallada de fuera del sistema autónomo (fuente de enrutamiento distinta de OSPF), de otras áreas y por supuesto de los demás routers internos.



Este tipo de área se conecta a la de backbone o Área 0. Todos los routers del área conocen los demás routers del área y tiene la misma base de datos topológica. Sin embargo, cada router tiene su propia tabla de enrutamiento.

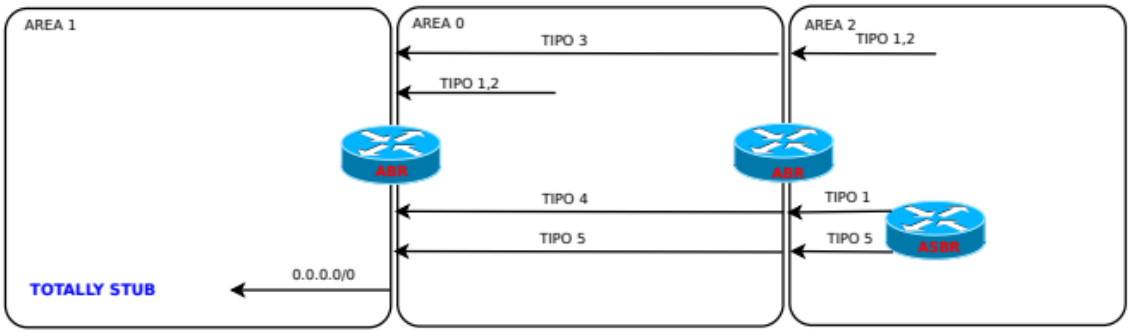
4.3.3. Área Stub

Las áreas *Stub* admiten la entrada de mensajes de tipo 3 a través de sus routers frontera, pero bloquean los mensajes de tipo 4 y 5 y en su lugar inyectan una ruta por defecto hacia el ABR. Esto implica que los routers internos a este tipo de áreas tienen la información detallada de otras áreas y por supuesto de los demás routers internos, pero no tienen todos los prefijos hacia fuentes de enrutamiento externas, porque se sustituyen por una ruta por defecto (0.0.0.0/0).



4.3.4. Área Totally Stub

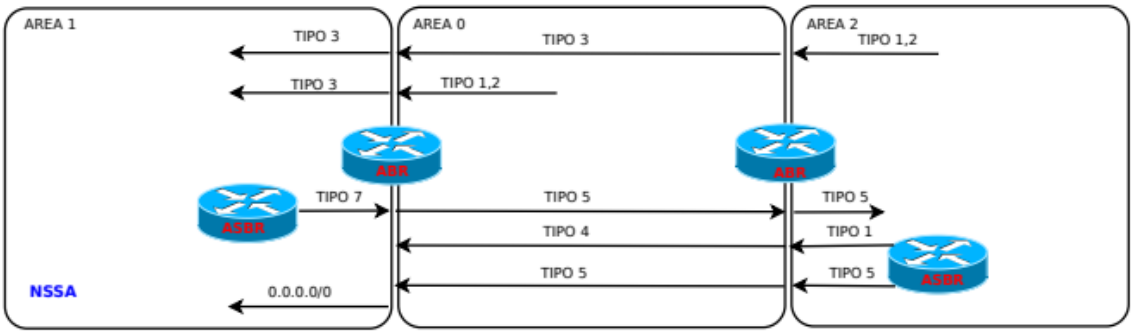
Las áreas *Totally Stub* ni siquiera admiten la entrada de mensajes de tipo 3 a través de sus routers frontera. El ABR bloquea los mensajes de tipo 3, 4 y 5 y en su lugar inyectan una ruta por defecto hacia el ABR. Esto implica que los routers internos al área sólo tienen información de otros routers internos. Toda la información de enrutamiento externa al área se sustituye por una ruta por defecto (0.0.0.0/0) hacia el ABR.



La única forma de salir del área es mediante una ruta por defecto. Este tipo de área es muy útil para sitios remotos con pocas redes y conectividad limitada con el resto de la empresa.

4.3.5. Área Not So Stubby Area (NSSA)

Las NSSA son área tipo *stub* por lo que se comportan igual que estas en cuanto a los mensajes que permiten entrar desde los routers frontera. Sin embargo, los routers internos de las ´áreas tipo stub no permiten la inundación mediante mensajes de tipo 4 y 5. Esto impide que se pueda colocar un ASBR en el interior de un área de tipo stub. Para evitar esta restricción nacen las NSSA. En este tipo de áreas un ASBR interno propagará mensajes de tipo 7, que cumplen la misma función que los de tipo 5, pero sí pueden propagarse hacia el ABR, en el que se sustituyen por mensajes de tipo 5 que se inundan hacia el resto de la red.



4.4. Comandos router QuaggaRouter para OSPF

- Configurar OSPF para las redes del área *X*:
  ```
  routerX(config)# router ospf
  routerX(config-router)# network <dirección IP> area X
  ```
- Revisar las tablas de enrutamiento: `routerX# show ip route`
- Ver los vecinos de un router: `show ip ospf neighbor`
- Asignar un identificador fijo a cada router:
  ```
  routerX(config)# router ospf
  routerX(config-ospf)# router-id X.X.X.X
  ```
- Configurar un área como área *stub* o *totally stub*:
  ```
  routerX(config)# router ospf
  routerX(config-router)# area X stub

  routerX(config)# router ospf
  routerX(config-router)# area X stub no-summary
  ```
- Activar mecanismo de sumarización de rutas:
  ```
  routerX(config-router)# area X range <dirección IP>
  ```
- Coste de referencia se puede modificar a nivel de router y el ancho de banda se debe configurar para cada interfaz de red:
  ```
  router(config)# int eth0
  router(config-if)# bandwidth 100000
  ```

4.5. Comandos router MikroTik para OSPF

- Crear interfaces de loopback: `/interface bridge add name=loX`
- Asignar direcciones a las interfaces de red:
  ```
  [admin@RX] > ip address add address=IP interface=Y
  ```
- Activar OSPF (la instancia default y el area0 ya existen):
  ```
  routing ospf instance add name=default
  routing ospf area add name=areaX area-id=X.X.X.X
  routing ospf network add network=<dir IP> area=<area name>
  ```
- Ver estado OSPF:
  `routing ospf route print` (ver rutas)
  `routing ospf neighbor print` (ver vecinos ospf)
- Sumarización de rutas:
  ```
  routing ospf area range add area=areaX range=<dir IP>
  ```
- Conversión a *Totally Stub*:
  ```
  routing ospf area set areaX type=stub inject-summary-lsa=yes
  ```

Tema 5. Redistribución de rutas y VRRP: Redundancia del primer salto

5.1. Redistribución de rutas

La redistribución de rutas es un concepto por el que se puede llegar a comunicar varios protocolos de enrutamiento, consiguiendo así una coexistencia entre ellos.

Casos de uso:

- Fusión de empresas: Supongamos que dos empresas se fusionan, cada una de ellas hasta el momento de fusión ha estado usando un protocolo de pasarela interior y ahora que se ha funcionado necesitan comunicarse entre ambas instalaciones con lo cual la redistribución de rutas puede ser una solución viable.
- Cambio del protocolo de enrutamiento: Puede llegar un momento en que nos veamos en la situación de tener que cambiar el protocolo actual de una red en esta.
- Razones de diseño: A pesar de intentar que nuestra red emplee un único protocolo de enrutamiento puede darse la situación en la que una parte de la red necesariamente utilice un protocolo de enrutamiento diferente al resto de la red. Y que ambas partes interoperan entre sí

Problemas en redes con redistribución de rutas:

Para poder realizar la redistribución de rutas se tiene que establecer métricas para una conversación de datos ya que cada protocolo usa sus métodos para calcular las rutas de enrutamiento.

Al usar este tipo de métricas puede darse el caso que se calculen rutas subóptimas. Esto suele darse en situaciones en las que ocurren bucles.

- Distintos Costes (RIP: saltos, OSPF: estado de enlace)
- Se redefinen costes al pasar de un protocolo a otro.
- Se pueden calcular **rutas subóptimas** si obtienen igual métrica al pasar de un protocolo a otro → **Distancia administrativa** (↑0 a 255) → Hay que **evitar bucles**.

5.1.1. Comandos router Quagga y MikroTik

- Asignar métrica (en RIP a OSP) en Quagga:
  ```
  router rip
  redistribute ospf metric 1
  ```
- Activar RIP en Quagga:
  `routerX(config)# router rip` (activar RIPv2)
  `routerX(config-router)# version 2`
  `routerX(config-router)# network <dirección IP> (añadir redes)`
- Activar RIP en MikroTik:
  `routing rip network add network=<dirección IP>`
- Redistribución de rutas two-way desde redes OSPF a RIP (MikroTik):
  `routing ospf instance set redistribute-rip=as-type-2`
- Redistribución de rutas two-way desde redes OSPF a RIP (MikroTik):
  `routing rip set redistribute-ospf=yes`
- Conversión a totally stub:
  `routing ospf area set areaX inject-summary-lsas=no type=stub`

5.2. VRRP: Redundancia del primer salto

Cuando se establece una ruta estática en los dispositivos hacia su puerta de enlace utilizando una dirección IP fija, surge un problema. En el caso de fallar el router que tenga asignada esta dirección, los equipos pierden la conexión con Internet, aun teniendo un enlace redundante hacia Internet, ya que la puerta de enlace configurada en los PCs indica que este es precisamente el router caído. La solución a este problema no viene de la mano de los protocolos de enrutamiento, sino del protocolo VRRP (Virtual Router Redundancy Protocol).

El protocolo VRRP permite:

- Tener varios routers conectados a la misma LAN, de forma que estos se coordinen entre sí para permitir un primer salto redundante.
- Varios routers físicos se configuran como un router virtual, al que se le asigna la dirección IP de la puerta de enlace.
- En el grupo de routers, es uno el que ejerce de maestro mientras que los demás funcionan en modo backup.
- El maestro envía anuncios de presencia y cuando se dejan de recibir, uno de respaldo toma el control.

5.2.1. Comandos router MikroTik

- Activar VRRP en MikroTik: `interface vrrp add interface=etherX vrid=XX`
  Activa el proceso VRRP sobre la interfaz `etherX` que es la que está conectada a la LAN. La variable `vrid=XX` es el identificador del router virtual.
- Asignar la dirección a un router virtual:
  `ip address add address=<dirección IP> interface=vrrp1`
- Información sobre las instancias VRRP: `interface vrrp print detail`
- Manejo de prioridad: `interface vrrp set vrrp1 priority=150`

Tema 6. Border Gateway Protocol (BGP) y redundancia del primer salto

6.1. Introducción

6.1.1. Sistemas autónomos

Un sistema autónomo se define como "Un grupo conectado de uno o más prefijos IP promovidos por uno o más operadores de red con una política de enrutamiento **única** y **claramente definida**.

6.1.2. Identificadores de sistemas autónomos

Cada sistema autónomo se identifica mediante un número entero asignado por la IANA (Internet Assigned Numbers Authority). Estos identificadores eran de 16 bits, lo que permitía hasta un máximo de 65535 sistemas autónomos. Ahora los identificadores son de 32 bits. Se consideran dos notaciones "asplain", que consiste en expresar el identificador como un número entero en notación decimal, y "asdot" en el que se separa en dos enteros de 16 bits mediante un punto `x.y`.

Los identificadores de sistemas autónomos se asignan por los registros regionales de internet (RIR – Regional Internet Registers), cada uno de los cuales se encarga de una zona geográfica distinta.

6.1.3. Tipos de sistemas autónomos

- ***Multihomed***: Son los que tiene conexiones con más de un sistema autónomo. Esto permite seguir conectado con Internet en el caso de fallar alguna conexión. Al contrario que los AS de tránsito no permiten el tráfico a través del sistema autónomo.
- ***Stub***: Es un sistema autónomo que está conectado únicamente con otro sistema autónomo.
- ***Tránsito***: Permite la interconexión entre otros sistemas autónomos a través de si mismo.
- ***Internet Exchange Point (IXP)***: Es una infraestructura física a través de la cual los proveedores de servicio o de contenido intercambian tráfico entre sus sistemas autónomos.

6.2. Enrutamiento entre sistemas autónomos: BGP (Border Gateway Protocol)

El protocolo BGP es un protocolo mediante el cual se intercambia información de encaminamiento entre sistemas autónomos. BGP utiliza TCP como protocolo de transporte, en el puerto 179. Dos routers BGP forman una conexión TCP entre ellos. Las parejas de routers con una conexión entre ellos se denominan *peers*. BGP contempla distintos tipos de mensajes que son intercambiados entre dos *peers*:

- ***OPEN***: Se utiliza para el establecimiento de una sesión BGP una vez haya sido establecida la conexión TCP. Se suelen negociar ciertos parámetros que caracterizan a esa sesión. Por ejemplo, los *peers* se ponen de acuerdo en la versión de BGP que se va a utilizar.

- ***UPDATE***: Es un mensaje de actualización que contiene los anuncios de nuevos prefijos. Se generarán mensajes de actualización cada vez que se determine una nueva ruta óptima para cierto destino o haya una modificación en alguna existente.
- ***KEEPALIVE***: Cuando la sesión BGP está activa se envía periódicamente un mensaje para mantener viva la conexión.
- ***NOTIFICATION***: Se envía al cerrar una sesión BGP. Esto sucede cuando ocurre algún error que requiera el cierre de la misma.

6.2.1. Atributos de paquete BGP

- ***ORIGIN***: Identifica el mecanismo por el cual se anunció el prefijo IP por primera vez. Se puede especificar como IGP (0), EGP (1) o INCOMPLETE (2).
- ***AS-PATH***: El atributo *AS-PATH* almacena una secuencia de números de AS que identifican la ruta de los AS por los que ha pasado el anuncio.
- ***NEXT-HOP***: Identifica la dirección IP del router correspondiente al siguiente salto hacia el destino. Se modifica cuando se anuncia una ruta fuera del sistema autónomo o cuando se desea redirigir tráfico a otro interlocutor. La información contenida en este campo sirve para incluir los prefijos IP contenidos en el anuncio en la tabla de enrutamiento.
- ***LOCAL-PREF***: Representa el grado de preferencia que el operador de red tiene por una determinada ruta dentro del sistema autónomo. El valor más alto indica una preferencia mayor. Por defecto, tiene el valor 100. El valor de este atributo es local al AS.
- ***Otros***: Existen algunos atributos más que no se van a tratar aquí.

6.2.3. El proceso de enrutamiento: iBGP y eBGP

- **iBGP**: Se produce cuando se comunican routers del mismo sistema autónomo intercambiando información BGP.
- **eBGP**: Se produce cuando se comunican routers fronterizos de sistemas autónomos diferentes.

6.2. Comandos en Quagga

- Activar sesiones BGP entre *peers*:
```
routerX(conf)# router bgp <as-number>
routerX(conf)# network <network-ip>
```
- Poner NEXT-HOP a IP del router que propaga por iBGP:
```
neighbor <ip-address> next-hop-self
```
- Anunciar un prefijo de red:
```
routerX(conf)# router bgp <as-number>
routerX(conf)# network <network-ip>
```
- Comprobar los prefijos de red conocidos:
```
show bgp ipv4 unicast
```
- Activar default route en RIP (desde default):
```
router rip
default-information originate
```

- Activar mapas de ruta (reglas o políticas de routing):

```
(config)# route-map ASXX-entrada permit 10
(config-route-map)# set local-preference 200
(config)# router bgp 100
(config-router)# neighbor 20.0.0.1 route-map ASXX-entrada in

(config)# ip as-path access-list 1 permit ^$
(config)# route-map ASXX-salida permit 10
(config-route-map)# match as-path 1
(config)# router bgp 100
(config-router)# neighbor 20.0.0.1 route-map ASXX-salida out
```

- Forzar sync de políticas de filtrado (soft sin reinicio):

```
clear ip bgp *soft
```

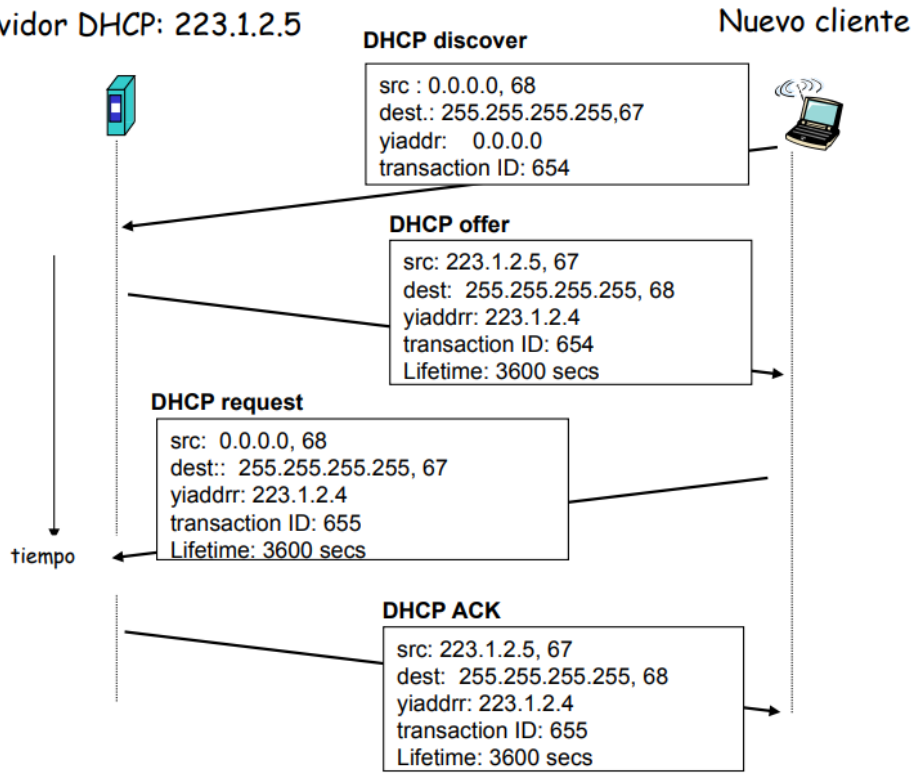
Tema 7. Servicios: DHCP

7.1. Introducción

DHCP es un protocolo que permite a los hosts de una red obtener su dirección IP de forma automática. Un servidor asigna de forma dinámica las direcciones durante un espacio de tiempo limitado.

Un servidor DHCP debe mantener lo que se conoce como pool de direcciones, que es un conjunto de direcciones IP que pueden ser asignadas a un host mediante este protocolo. Cada subred deberá tener su propio pool de direcciones. Cuando un host pide una dirección mediante DHCP, la dirección se retira de forma temporal del pool hasta que venza el tiempo de préstamo sin que se produzca una renovación (*lease time*) o hasta que el host libere la dirección

El funcionamiento del protocolo DHCP es el siguiente:



1. El host cliente envía un mensaje del tipo *`DHCP discover`* destinado a broadcast, puesto que desconoce la dirección IP del servidor DHCP.
2. El servidor DHCP responde con un mensaje *`DHCP offer`* ofreciendo una dirección IP al host solicitante.
3. El host cliente debe pedir la dirección ofrecida mediante un *`DHCP request`*.
4. Finalmente, el servidor confirma la reserva mediante el mensaje *`DHCP ack`*.

Con el fin de centralizar todos los pools DHCP en un único servidor, se utiliza lo que se denomina *DHCP relay*, que consiste en que en cada red broadcast que utilice DHCP exista un agente de reenvío que recoja los mensajes y los envíe al servidor, que puede estar situado en otra red. Una vez el servidor genera las respuestas las envía al agente de reenvío y ´este las inyecta de nuevo en la red del cliente.