

# Práctica: Routing IP

## 1 Introducción

La práctica pretende adquirir cierta soltura en el direccionamiento IP y la gestión de rutas estáticas. Para ello se dispone de los equipos Mikrotik 2011 con sistema operativo OpenWRT que trabajarán como routers. Estos se interconectarán con otras redes según el esquema. Asimismo se dispone de los equipos de usuario para gestionar y configurar los routers, con sistemas operativos linux o Windows, y el programa ssh para la comunicación con los routers. Si la conexión se realizase desde equipos Windows, se podría utilizar el programa de emulación de terminal de libre distribución “putty”. La distribución OpenWRT escogida es la 19.07.10, versión avanzada de OpenWRT, donde en su zona web se pueden encontrar manuales (instalación - <http://www.openwrt.org>) y abundante documentación.

Esta práctica está diseñada para trabajo en grupos, y junto con cualquier otro entorno de enrutamiento (como por ejemplo Cisco) forma una unidad didáctica sobre enrutamiento IP estático en general.

### 1.1 Objetivos

El objetivo es conseguir desde los equipos conectividad total con el resto de las redes/subredes del esquema general del bloque de prácticas.

De forma mas práctica, los objetivos consisten en la configuración de los routers y los equipos de gestión con el direccionamiento deseado y presentado en el esquema general de la práctica.

Para ello se seguirá una lista de tareas, que de forma resumida son:

- Configuración de las interfaces de los equipos.
  - Configuración de los equipos gestores
  - Configuración de los routers
- Configuración de las rutas
  - Rutas estáticas en los routers
  - Rutas por defecto en los equipos gestores.
- Pruebas

### 1.2 Situación de partida

Dadas las características de compartición del laboratorio y por tanto de sus equipos, la configuración inicial de los equipos puede variar, por lo que habrá que uniformarla en la medida de lo posible.

- Situación de partida de los PC's

Los PCs deberán arrancar con un direccionamiento de red acorde a la red plana del laboratorio (192.168.29.0/24) y con salida a Internet a través de F0 (192.168.29.1). Por ejemplo: el PC13 arrancará con la dirección IP 192.168.29.113/24 y tendrá como puerta de enlace (Gateway) a F0.
- Situación de partida de los routers (RBM)

Los RouterBoardsMikrotik (RBM) arrancan con la dirección ip 192.168.29.A siendo A el número de RBM escrito en su carcasa, por ejemplo, el RBM21 arranca con la dirección IP 192.168.29.21. Esta dirección es accesible en la interfaz “br-lan” de la figura 1, es decir, el PC gestor de cada RBM deberá conectarse con el latiguillo con conectores RJ45 a cualquier puerto de la interfaz “lan”.

Para que funcionen como routers, será necesario acceder a ellos y ejecutar el comando:

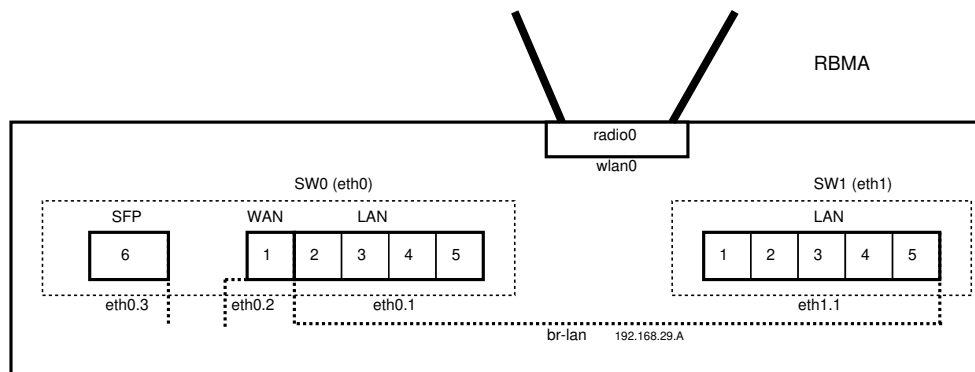


Figura 1: Situación de partida

Tabla 1: Situación de partida

Interfaz	Dirección IP
eth0.1	192.168.1.1/24
eth1.1	192.168.2.1/24
eth0.2	192.168.3.1/24
eth0.3	192.168.4.1/24
login	passwd
root	provisional

Comandos 1: Conversión en router.

```
root@RBMA:~# ./isa_router.sh
```

Desde ese momento, la configuración de las interfaces cambia, así como el direccionamiento IP, según se muestra en la figura 2 y en la tabla 1. Los RouterBoardMicrotik pasarán a convertirse en routers y tomarán el nombre de RA, siendo A el mismo número asociado al RBMA anterior.

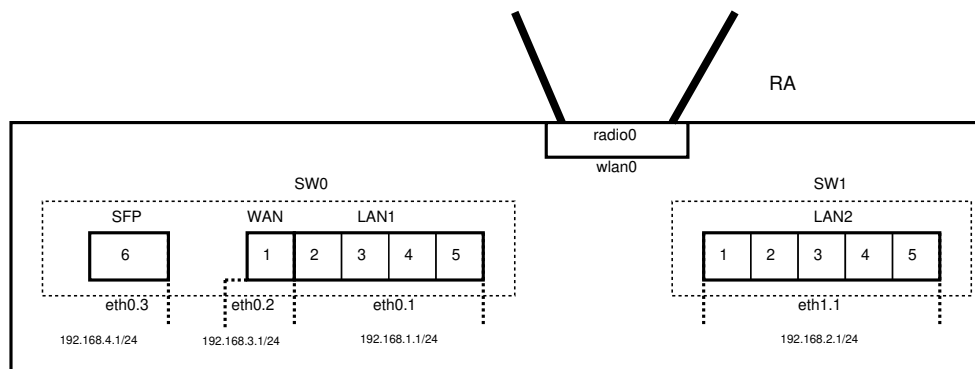


Figura 2: Situación de partida como router.

El sistema operativo OpenWRT incluye además de los demonios de enrutamiento y las utilidades básicas de los sistemas Linux, otras utilidades mas complejas, como lo son el firewall “shorewall” (deshabilitado en esta práctica), la weblet para visualizar el funcionamiento del router a través de un navegador y el demonio “sshd” para permitir conexiones remotas seguras y poder prescindir en los equipos con OpenWRT de tarjeta gráfica y monitor. El RBMA dispone de un puerto serie con interfaz RJ45 para conectividad directa.

- Tabla con información de la situación de partida de RBMA (atención a las interfaces y sus direcciones IP)

Incluye servidor shell seguro (dropbear) configurado para acceder desde cualquier direccion. Incluye paquete zebra, ospfd. Se puede acceder a traves de puerto serie.

Es muy importante resaltar que TODOS los routers tienen el mismo direccionamiento IP, por lo que la misma dirección IP estará repetida. Esto implica que habrá que modificar el direccionamiento de todas las interfaces de los routers y ponerlas según el objetivo de direccionamiento de la figura 4.

IMPORTANTE: Todos los RBMs adquieren la misma configuración y el mismo direccionamiento IP, por lo que será necesario extremar las precauciones para no interferir entre equipos con la misma dirección IP.

## 2 Desarrollo

Los comandos específicos imprescindibles para esta práctica son:

```
#ip addr [help]
#ip route [help]
```

Cuyo uso mas común tendrá como ejemplo:

```
# ip addr add <direccion/mask> dev <ethx>
ejemplo# ip addr add 192.168.12.112/24 dev eth1

# ip route add <red destino/mascara> via <gateway> dev <interfaz de salida>
ejemplo# ip route add 192.168.12.0 via 10.10.12.1 dev eth0.1
```

### 2.1 Acceso inicial a cada uno de los routers desde los equipos de gestión

Será necesario realizar el conexionado adecuado a la topología de la figura 3. En ella, la línea roja implica el uso de fibra óptica, las líneas negras son latiguillos de par de cobre con conectores RJ45, la “nube” es la red conmutada de nuestro laboratorio, RBM-B y RBM-C son RouterBoardMicrotik del laboratorio, y el PC-A es el PC de usuario.

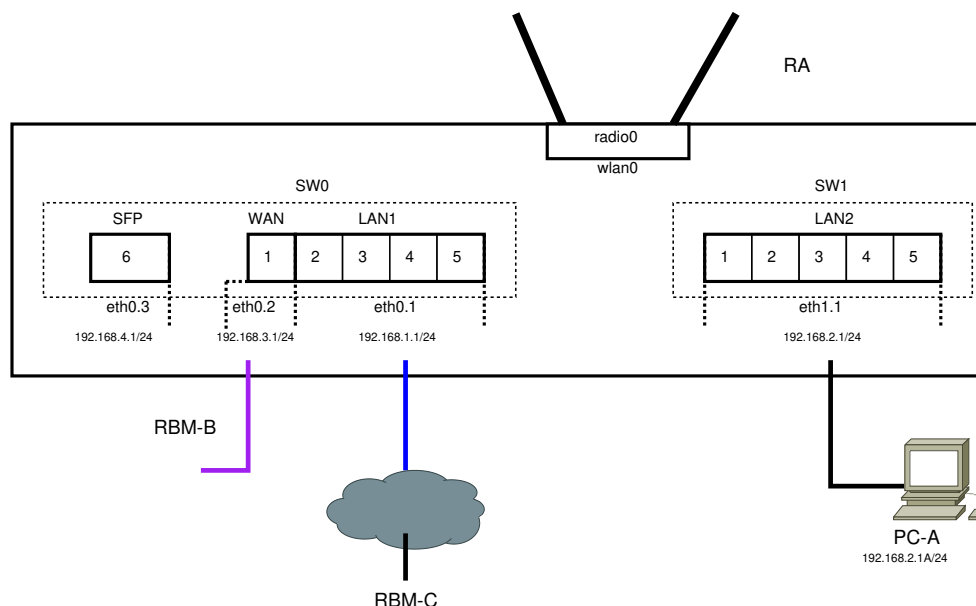


Figura 3: Conexionado

Para que cada equipo de gestión (PC-A) acceda a su Router tendrá que realizar además de un cambio en su conectividad hacia el router también en su direccionamiento, de forma que para conectarse al router RA

a través de la interfaz eth1.1 con la dirección IP de partida 192.168.2.1/24, el PC-A tendrá que configurarse con la IP 192.168.2.1A/24.

```
PC-A# ip addr add 192.168.2.1A/24 dev eth0
```

Nota: la interfaz en PC-A puede tener un nombre distinto a “eth0”. Será necesario sustituirlo en el comando anterior.

Tras ello se realizará la conexión por ssh a la interfaz eth1 de RA. El usuario será “root” y la clave la que aparece en las condiciones iniciales de RA o RB según corresponda.

```
PC-A # ssh -l root <direccion IP de la interfaz del router RA>
```

## 2.2 Configuración de las interfaces de los routers

Será necesario, tras tener el control del router, configurar sus interfaces según la figura 4, para lo cual se usará el comando “ip addr” en los routers.

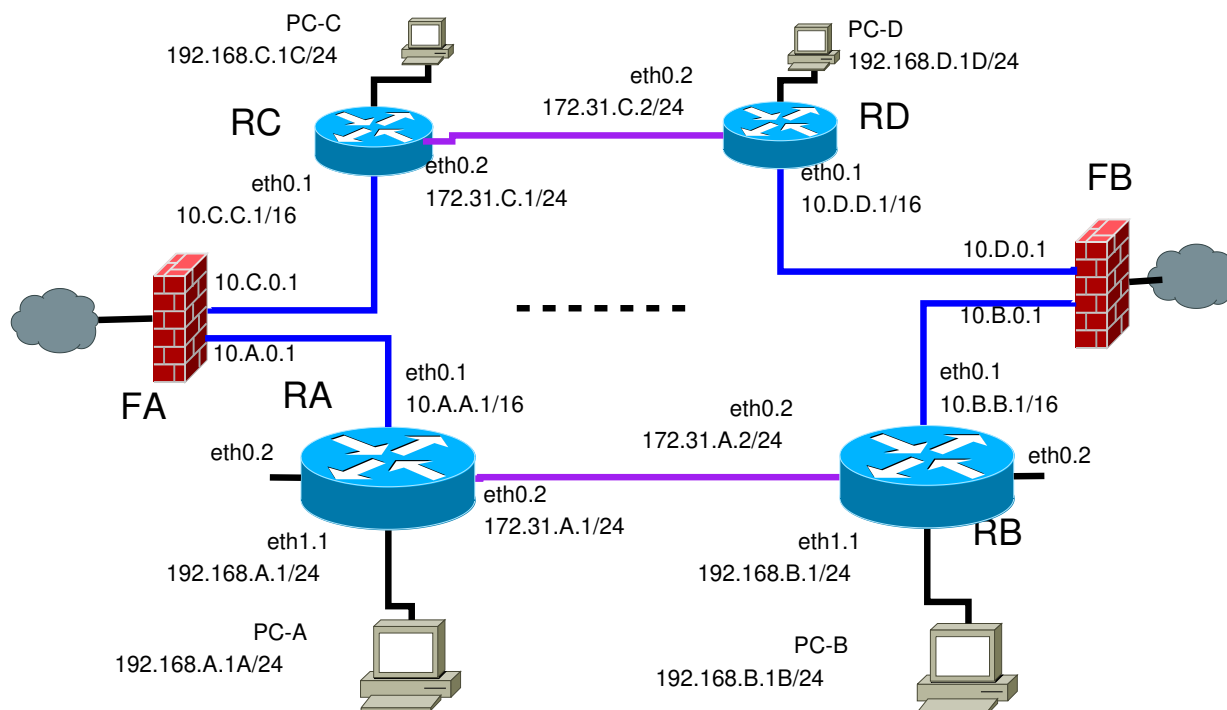


Figura 4: Objetivo de direccionamiento

En la figura 4, los números de los routers RA, RB, RC y RD cumplen  $A < B < C < D$ , de forma que en la práctica se usarán 4 routers de los cuales A será el menor y D el mayor. Es importante tener en cuenta que esa codificación repercute directamente en el direccionamiento propuesto. No obstante se recomienda atender a cada situación de interconexión particular, prestando especial atención a la conectividad tanto a nivel de enlace como a nivel IP.

Un ejemplo indicativo de cómo quedaría el direccionamiento con los R20, R21, R22 y R23 puede verse en la figura 5

Se deberá probar la conectividad de las interfaces con equipos próximos y routers.

## 3 Configuración de las rutas

Una vez que existe acceso ssh a los routers, se procederá a configurar las rutas de los mismos y las rutas por defecto en los equipos gestores.

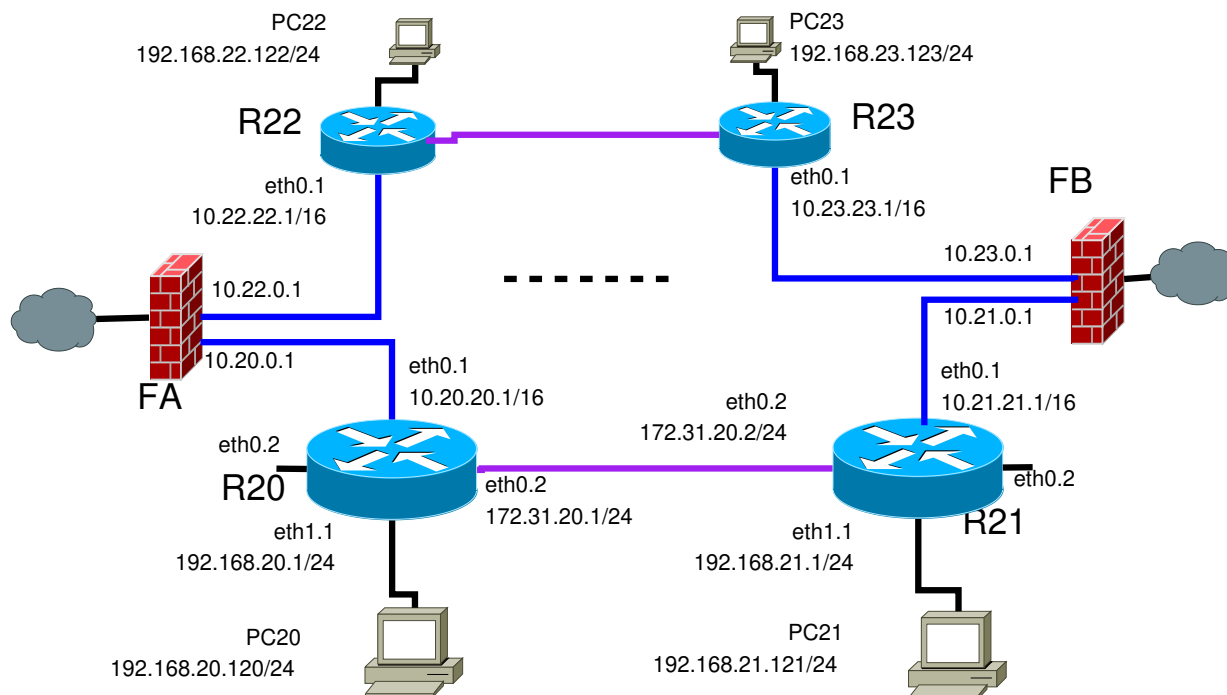


Figura 5: Ejemplo de direccionamiento

Las rutas que se configuran son estáticas, por lo que se realizará la inserción de las mismas de forma manual con el comando “ip route” de la forma:

```

RX# ip route add <red destino/mascara> via <gateway> dev <interfaz de salida>

```

- En RA Rutas directas ya están configuradas. Pueden visualizarse con el comando

```

RA # ip route

```

Será necesario configurar las rutas hacia las redes de PC-B, PC-C y PC-D.

Ruta hacia PC-B (192.168.B.0/24) a través de RB

```

RA# ip route add 192.168.B.0/24 via 172.31.A.2 dev eth0.2

```

Ruta hacia PC-C (192.168.C.0/24) a través de RC

```

RA# ip route add 192.168.C.0/24 via 10.A.C.1 dev eth0.1

```

Ruta hacia PC-D (192.168.D.0/24) a través de RB

```

RA# ip route add 192.168.D.0/24 via 172.31.A.2 dev eth0.2

```

Ruta por defecto (0.0.0.0) a través de FA

```

RA# ip route add default via 10.A.0.1

```

Nota: existen rutas alternativas a cada uno de estos destinos. Se invita al alumnado a realizar configuraciones alternativas coordinándose con sus compañeros/as.

- En RB Rutas directas ya están configuradas. Pueden visualizarse con el comando

```
RB # ip route
```

Será necesario configurar las rutas hacia las redes de PC-B, PC-C y PC-D.

Ruta hacia PC-A (192.168.A.0/24) a través de RA

```
RB# ip route add 192.168.A.0/24 via 172.31.A.1 dev eth0.2
```

Ruta hacia PC-C (192.168.C.0/24) a través de RA

```
RB# ip route add 192.168.C.0/24 via 172.31.A.1 dev eth0.2
```

Ruta hacia PC-D (192.168.D.0/24) a través de RD

```
RB# ip route add 192.168.D.0/24 via 10.B.D.1 dev eth0.1
```

Ruta por defecto (0.0.0.0) a través de FA

```
RA# ip route add default via 10.A.0.1
```

Se invita al alumnado crear las rutas en RC y en RD. En ambos casos la puerta de enlace es FC.

Nota: existen rutas alternativas a cada uno de estos destinos. Se invita al alumnado a realizar configuraciones alternativas coordinándose con sus compañeros/as.

Rutas por defecto en los equipos

- Ruta por defecto en PC-A

Esta será la interfaz eth1.1 de RA (192.168.A.1). Los comandos a ejecutar son:

```
Linux: PC-A # ip route add default via 192.168.A.1
```

- Ruta por defecto en PC-B

Esta será la interfaz eth1.1 de RB (192.168.B.1). Los comandos a ejecutar son:

```
Linux: PC-B # ip route add default via 192.168.B.1
```

- Ruta por defecto en PC-C y PC-D

Se invita al alumnado a configurar correctamente esas rutas.

## 4 Varias salidas a Internet

La idea genérica de que un dispositivo (o red) pueda conectarse a varias redes (por ejemplo disponiendo de varias salidas a Internet) se llama en la literatura científica “multihoming”.

En Linux, la gestión de varias salidas a Internet puede hacerse mediante el Policy Based Routing (PBR) o enrutamiento basado en políticas. Este tipo de enrutamiento existe en Linux desde el kernel 2.2. y se gestiona mediante herramientas basadas en *iproute2*.

Parte de la base de que existen varias tablas de enrutamiento y la posibilidad de seleccionar una u otra en base a unas reglas. Si un paquete cumple la condición de una regla, el paquete se enrutará tomando la tabla de enrutamiento asociada a esa regla.

**Tablas** Las tablas están dadas de alta en el fichero `/etc/iproute2/rt_tables`. Estas tablas son gestionadas por reglas, que pueden visualizarse con el comando `ip rule`. El contenido habitual de `/etc/iproute2/rt_tables` es:

```
RX# cat /etc/iproute2/rt_tables
#
# reserved values
#
255     local
254     main
253     default
0       unspec
#
# local
#
#1      inr.ruhep
```

Puede verse la existencia de las tablas “local”, “main” y “default”, además de otras como “unspec”. Se puede visualizar el contenido de cada una de ellas mediante el comando `ip route show table <tabla>`, siendo “<tabla>” el nombre de la tabla. Por ejemplo:

```
RX# ip route show table local
broadcast 127.0.0.0 dev lo  proto kernel  scope link    src 127.0.0.1
local 127.0.0.0/8 dev lo  proto kernel  scope host    src 127.0.0.1
local 127.0.0.1 dev lo  proto kernel  scope host    src 127.0.0.1
broadcast 127.255.255.255 dev lo  proto kernel  scope link    src 127.0.0.1
broadcast 193.136.92.0 dev eth0  proto kernel  scope link    src 193.136.93.114
local 193.136.93.114 dev eth0  proto kernel  scope host    src 193.136.93.114
broadcast 193.136.93.255 dev eth0  proto kernel  scope link    src 193.136.93.114
```

Aquí pueden verse rutas directas de las interfaces locales.

**Reglas** La gestión de las reglas que permiten el uso de una tabla u otra se hace con el comando `ip rule`.

La salida habitual del comando `ip rule` es:

```
RX# ip rule list
0:  from all lookup local
32766:  from all lookup main
32767:  from all lookup default
```

Donde puede verse la prioridad en el primer campo (menor valor implica mayor prioridad) y el formato básico de las reglas.

Cuando un paquete con una dirección IP origen y destino llega al router, este consulta su lista de reglas por orden de prioridad y aplica la tabla de rutas pertinente. En este caso por defecto, se consultará la primera regla *0: from all lookup local* que significa que un paquete con cualquier origen (“from all”) usará la tabla local (“lookup local”). Si no hay destino en esa tabla, consultará la siguiente regla *32766: from all lookup main*, es decir, consultará la tabla principal (“lookup main”) y así sucesivamente.

Aplicar esto a la posibilidad de disponer de dos salidas a Internet implica la creación de reglas y tablas específicas. Existen varias soluciones:

## 4.1 Rutas por defecto duplicadas

A esta solución se la llama “deflection routing” o “hot potato routing”.

Se plantea la posibilidad de que haya dos salidas a Internet, una por FA y otra por FB. De esa forma RA puede seleccionar que la red de PC-A salga por FB y la red de PC-B salga por FA (y todas las demás redes salgan por FA). Al igual puede hacerse con RB, que la red de PC-B salga por FA y la red de PC-A salga por FB.

Primeramente será necesario conocer las tablas de enrutamiento operativas, que en general son con las que arranca el sistema. Para ello basta con conocer el orden de aplicación de las reglas del PBR y ver a qué tablas afecta:

```
RX# ip rule list
0:    from all lookup local
32766: from all lookup main
32767: from all lookup default
```

lo que muestra la prioridad de las reglas y la lista de tablas a las que afecta (local, main y default). La lista de tablas puede visualizarse además como:

```
RX# more /etc/iproute2/rt_tables
```

La tabla “local” es una tabla de enrutamiento especial que contiene rutas de alta prioridad y permiten a las interfaces conocer cómo acceder a diversas redes o direcciones básicas, como a sí mismas o a direcciones de broadcast. Esta tabla no suele modificarse. Se puede visualizar su contenido de la forma:

```
RA:~#ip route show table local
broadcast 192.168.3.0 dev eth2  proto kernel  scope link    src 192.168.3.21
broadcast 192.168.2.255 dev eth1  proto kernel  scope link    src 192.168.2.21
broadcast 127.255.255.255 dev lo    proto kernel  scope link    src 127.0.0.1
local 192.168.29.21 dev eth0  proto kernel  scope host    src 192.168.29.21
local 192.168.3.21 dev eth2  proto kernel  scope host    src 192.168.3.21
broadcast 192.168.29.0 dev eth0  proto kernel  scope link    src 192.168.29.21
broadcast 192.168.3.255 dev eth2  proto kernel  scope link    src 192.168.3.21
broadcast 192.168.2.0 dev eth1  proto kernel  scope link    src 192.168.2.21
local 192.168.2.21 dev eth1  proto kernel  scope host    src 192.168.2.21
broadcast 192.168.29.255 dev eth0  proto kernel  scope link    src 192.168.29.21
broadcast 127.0.0.0 dev lo    proto kernel  scope link    src 127.0.0.1
local 127.0.0.1 dev lo    proto kernel  scope host    src 127.0.0.1
local 127.0.0.0/8 dev lo    proto kernel  scope host    src 127.0.0.1
```

La tabla “main” es la tabla habitual que contiene las rutas basadas en destino. Se puede visualizar su contenido de la forma:

```
RA:~#ip route show table main
192.168.3.0/24 dev eth2  proto kernel  scope link    src 192.168.3.21
192.168.2.0/24 dev eth1  proto kernel  scope link    src 192.168.2.21
192.168.29.0/24 dev eth0  proto kernel  scope link    src 192.168.29.21
default via 192.168.29.1 dev eth0
```

La tabla “default” está vacía de comienzo. Se utiliza para post-procesamiento si un paquete no sabe a qué tabla está asignado su destino, será pues como una “tabla por defecto”.

Para disponer de dos rutas por defecto, será necesario que ambas rutas estén en diferentes tablas, por lo que se creará una tabla específica para la segunda ruta por defecto:

## Comandos 2: Creación de tablas de rutas en RA (TablaFB) y en RB (TablaFA)

```
RA# echo 200 TablaFB >> /etc/iproute2/rt_tables
RB# echo 200 TablaFA >> /etc/iproute2/rt_tables
```

Por supuesto, habrá que añadir las rutas por defecto alternativas a las nuevas tablas de rutas creadas, además de las rutas hacia las propias redes que saldrían por esas rutas por defecto, para que el router conozca el camino de retorno también en esa tabla. Es decir, cada una de estas tablas deberá ser prácticamente idéntica a la tabla “main” salvo por la selección de su “puerta de enlace”:

## Comandos 3: Añadido de rutas en RA (TablaFB)

```
RA# ip route add 192.168.A.0/24 via 192.168.A.1 dev eth1.1 table TablaFB
RA# ip route add 10.A.0.0/16 via 10.A.A.1 dev eth0.1 table TablaFB
RA# ip route add default via 172.31.A.2 dev eth0.2 table TablaFB
RA# ip route show table TablaFB
default via via 172.31.A.2 dev eth0.2
192.168.A.0/24 via 192.168.A.1 dev eth1.1
10.A.0.0/16 via 10.A.A.1 dev eth0.1
```



#### Comandos 4: Añadido de rutas en RB (TablaFA)

```
RB# ip route add 192.168.B.0/24 via 192.168.B.1 dev eth1.1 table TablaFA
RB# ip route add 10.B.0.0/16 via 10.B.B.1 dev eth0.1 table TablaFA
RB# ip route add default via 172.31.A.1 dev eth0.2 table TablaFA
RB# ip route show table TablaFA
default via via 172.31.A.1 dev eth0.2
192.168.B.0/24 via 192.168.B.1 dev eth1.1
10.B.0.0/16 via 10.B.B.1 dev eth0.1
```

A pesar de estar creada la tabla, ninguna regla la usa. Las reglas son registros de la base de datos de rutas con campos de “prioridad”, “selector” y “acción” y “tabla”. El router arranca con tres reglas básicas:

1. Con prioridad 0 (máxima prioridad), con selección “cualquier origen” (from all), con acción “enruta por tabla local” (lookup table local). Esta regla es fundamental para cualquier conectividad IP, ya que la tabla “local” contiene las direcciones básicas. Esta regla no puede ser borrada ni sobreescrita.
2. Con prioridad 32766, con selección “cualquier origen” (from all), con acción “enruta por tabla main” (lookup table main). Básicamente, si no había ruta en la tabla “local”, prueba a ver si hay en la tabla “main”, que es la tabla de rutas estáticas habituales, así como de rutas de interface.
3. Con prioridad 32767, con selección “cualquier origen” (from all), con acción “enruta por tabla default” (lookup table default). Esta es la última regla que se aplica hacia una tabla que contenga rutas alternativas y no encontradas previamente.

Para seleccionar los paquetes que usan la tabla recién creada (TablaFA o TablaFB según corresponda), se crearán nuevas reglas con el comando `ip rule`:

#### Comandos 5: Añadido de reglas en RA

```
RA# ip rule add from 192.168.A.0/24 table TablaFB prio 1000
RA# ip rule
0:      from all lookup local
1000:   from 192.168.A.0/24 lookup TablaFB
32766:  from all lookup main
32767:  from all lookup default
```

#### Comandos 6: Añadido de reglas en RB

```
RB# ip rule add from 192.168.B.0/24 table TablaFA prio 1000
RB# ip rule
0:      from all lookup local
1000:   from 192.168.B.0/24 lookup TablaFA
32766:  from all lookup main
32767:  from all lookup default
```

que viene a decir que cualquier paquete que llegue con una dirección de origen perteneciente a la red 192.168.A.0/24 utilizará la tabla TablaFB en RA, o cuando el paquete llega con una dirección de origen perteneciente a la red 192.168.B.0/24 utilizará la tabla TablaFA en RB.

Conviene consultar las posibilidades de selección de reglas que ofrece `iproute2`.

El demonio de enrutamiento consulta las reglas por orden de prioridad. Cuando un paquete encuentra su ruta de salida, el demonio deja de consultar reglas y tablas para ese paquete. De esa forma, aunque exista una ruta estática hacia la red “default” en la tabla “main”, utilizará la ruta que encuentre en la tabla de menor prioridad (en este caso TablaFA o TablaFB).

Finalmente conviene borrar la caché de rutas para evitar que las rutas anteriores estén cacheadas.

```
RX# ip route flush cache
```

## 4.2 Balanceo de carga

Algunos equipos pueden plantearse balancear la carga.

Existen dos formas de balanceo de la salida a Internet, una por flujos y la otra por paquetes.

- Balanceo por flujos

Está basado en el mantenimiento de una ruta para un flujo de datos (mismo ID de conexión). Uno de los aspectos mas destacados es el “cacheado” de la ruta, de forma que para un mismo destino siempre se utilizará la misma ruta. Cuando el destino no está en caché, es el demonio de enrutamiento el que decide a través de qué ruta enviarlo, dependiendo del parámetro “weight”.

En este caso se trabaja con la opción de rutas multipath.

#### Comandos 7: Creación de rutas balanceadas en RA y en RB

```
RA# ip route add default scope global nexthop via 10.A.0.1 dev eth0.1 weight 1 nexthop via ↔
172.31.A.2 dev eth0.2 weight 1
RB# ip route add default scope global nexthop via 10.B.0.1 dev eth0.1 weight 1 nexthop via ↔
172.31.A.1 dev eth0.2 weight 1
```

El valor del parámetro “weight” puede cambiarse para dar prioridad a una sobre otra.

- Balanceo por paquetes

En este caso, la decisión de una ruta u otra se hace en cada paquete. No es la opción mas recomendada para flujos orientados a conexión, pues aumenta la probabilidad de que los paquetes lleguen desordenados. No se produce cacheado.

En ese caso se usa la opción “equalize” de iproute2.

#### Comandos 8: Balanceo por paquetes en RA

```
RA# ip route add default equalize nexthop via 10.A.0.1 dev eth0.1 weight 1 nexthop via ↔
172.31.A.2 dev eth0.2 weight 1
RA# ip route show
default equalize
nexthop via 10.A.0.1 dev eth0.1 weight 1
nexthop via 172.31.A.2 dev eth0.2 weight 1
```

#### Comandos 9: Balanceo por paquetes en RB

```
RB# ip route add default equalize nexthop via 10.B.0.1 dev eth0.1 weight 1 nexthop via ↔
172.31.A.1 dev eth0.2 weight 1
RB# ip route show
default equalize
nexthop via 10.B.0.1 dev eth0.1 weight 1
nexthop via 172.31.A.1 dev eth0.2 weight 1
```

Se invita al usuario a probar el funcionamiento del balanceo, así como a comprobar si la caída de alguna de las salidas a Internet mantiene correctamente el acceso a Internet a través de la otra.

## 5 Enrutamiento en el núcleo

Actualmente, la mayoría de las redes tienen una arquitectura conmutada, de forma que comparten el dominio de broadcast o lo segmentan utilizando tecnología VLAN. Cuando se dispone de varias subredes IP (capa 3) en una misma red conmutada (capa 2), la forma de comunicarse entre ellas es a través de un enrutador con una sólo interfaz de capa 2 y tantas direcciones IP como subredes IP haya. La figura XXXX muestra la topología recomendada.

FIGURA XXXX. RBM-B,C,D.... haciendo de switches y RBA haciendo de router

## 6 Pruebas

Finalmente, se comprobará la conectividad (con el comando “ping”) con el resto de la red, así como hacia internet, dependiendo esto de la correcta implementación de las demás prácticas implicadas.

- RA ↔ RB

- $RA(RB) \leftrightarrow PC-A, PC-B$
- $RA(RB) \leftrightarrow RC, RD$
- $RA(RB) \leftrightarrow PC-A, PC-B, PC-C, PC-D$
- $RA(RB) \leftrightarrow INTERNET$
- $RA(RB) \leftrightarrow \dots\dots\dots$