

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 2 equipos SOEKRIS 4503 con sistema operativo OpenWRT (R1 y R2), y la interconexión con otras redes según el esquema. Asimismo se dispone de los pares de equipos PC17 y PC25 para gestionar y configurar R1, y PC16 y PC24 para gestionar y configurar R2, 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 Kamikaze 10.3.01, 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 2 grupos, y junto con cualquier otro entorno de enrutamiento (como por ejemplo Cisco) forma una unidad didáctica sobre enrutamiento IP estático en general.

Grupo 1: conjunto R1, PC17 y PC25 para gestión remota. Grupo 2: conjunto R2, PC16 y PC24 para gestión remota.

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).
- Situación de partida de R1 y R2.

Esta siempre será la misma, pues los routers se reiniciarán al comienzo de cada práctica.

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.

- Fichero con información de la situación de partida de R1 (fijarse en las direcciones IP y en los parámetros de login y password)

```
R1
Dispositivo de Red SOEKRIS net4503
OpenWRT Backfire (10.03.1-RC6, r28680)
* Kernel 2.6.32
* teclado espanol
* Interfaces , Direcciones , Zonas :
  eth0      : 192.168.29.21/24
  eth1      : 192.168.2.21/24
  eth2      : 192.168.3.21/24
* Incluye servidor shell seguro (dropbear) configurado para acceder desde cualquier ←
  direccion.
* Incluye paquete zebra, ospfd.
* Se puede acceder a traves de puerto serie.
* login: root
* passwd: provisional
```

- Fichero con información de la situación de partida de R2 (fijarse en las direcciones IP y en los parámetros de login y password)

```
R2
Dispositivo de Red SOEKRIS net4503
OpenWRT Backfire (10.03.1-RC6, r28680)
* Kernel 2.6.32
* teclado espanol
* Interfaces , Direcciones , Zonas :
  eth0      : 192.168.29.22/24
  eth1      : 192.168.2.22/24
  eth2      : 192.168.3.22/24
* Incluye servidor shell seguro (dropbear) configurado para acceder desde cualquier ←
  direccion.
* Incluye paquete zebra, ospfd.
* Se puede acceder a traves de puerto serie.
* login: root
* passwd: provisional
```

- Situación de partida esquematizada, en la figura 1

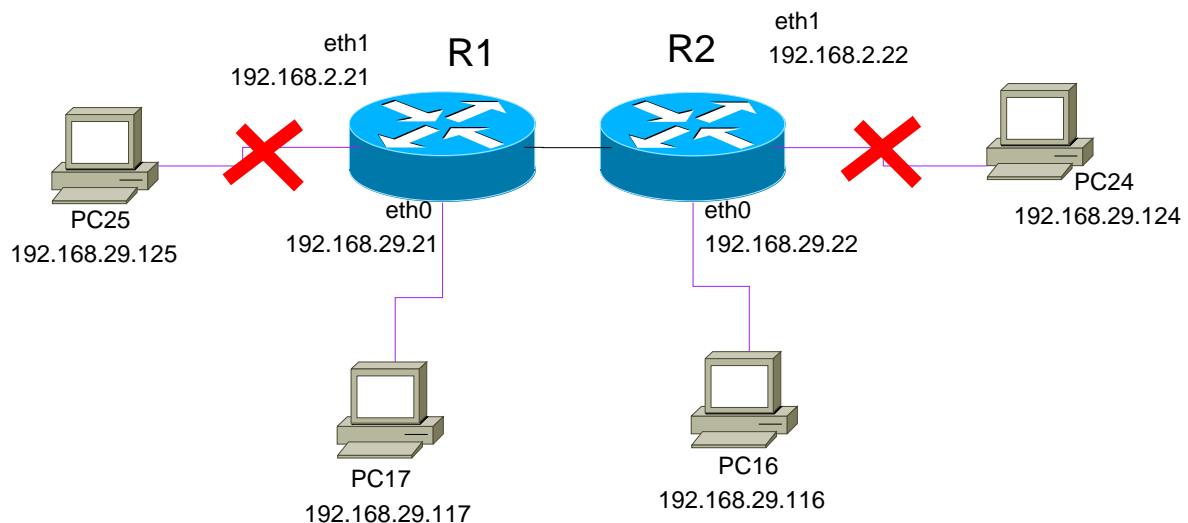


Figura 1: Situación de partida

donde puede verse que PC17 y PC16 pueden conectarse inicialmente a R1 y R2 respectivamente, ya que pertenecen a la misma subred, pero PC25 y PC24 necesitan cambiar su direccionamiento para conectarse a las interfaces eth1 de R1 y R2 respectivamente.

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 192.168.201.116/24 dev eth1

# ip route add <red destino/mascara> via <gateway> dev <interfaz de salida>
ejemplo# ip route add 192.168.201.0 via 10.0.0.1 dev eth2
```

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

- Acceso desde PC17 a R1 Este se hará por conexión ssh a la interfaz eth0 de R2. No necesita reconfiguración. El usuario será “root” y la clave la que aparece en las condiciones iniciales de R1.
- Acceso desde PC16 a R2 Este se hará por conexión ssh a la interfaz eth0 de R2. No necesita reconfiguración. El usuario será “root” y la clave la que aparece en las condiciones iniciales de R2.
- Acceso desde PC25 a R1 o PC24 a R2 Será necesario reconfigurar la interfaz de PC24 o PC25 para acceder a la red IP de la interfaz eth1 de R1 o R2. Para ello se hace de la forma habitual dependiendo del sistema operativo.

```
Linux: PCXX# ip addr add 192.168.2.1XX/24 dev eth0
Windows: C:\> netsh interface ip set address local1 static 192.168.2.1XX 255.255.255.0
```

Siendo XX el número del PC que corresponda (25 para PC25 y 24 para PC24) Tras ello se realizará la conexión por ssh a la interfaz eth1 de R1 o R2. El usuario será “root” y la clave la que aparece en las condiciones iniciales de R1 o R2 según corresponda.

El uso de la conexión ssh desde el PC de gestión a su router se realizará dependiendo del sistema operativo que se esté usando:

- Windows: Si se hiciese desde equipos Windows bastaría con ejecutar el programa “putty” y conectarse por el puerto 22 (SSH) a la dirección IP de la interfaz del router. Es recomendable hacer un “ping” previamente para asegurar que existe conectividad.
- Linux: Se hace directamente desde la línea de comandos si está el cliente “ssh” instalado. En los equipos del laboratorio, y en general en cualquier distribución GNU/Linux es así.

```
PCXX # ssh -l root <direccion IP de la interfaz del router>
```

2.1.1. Esquema tras la reconfiguración para el acceso a los routers

2.2. Configuración de las interfaces de los routers

Será necesario, tras tener el control del router, configurar sus interfaces según la arquitectura de la figura de la práctica, para lo cual se usará el comando “ip addr” en los routers, como se describe a continuación:

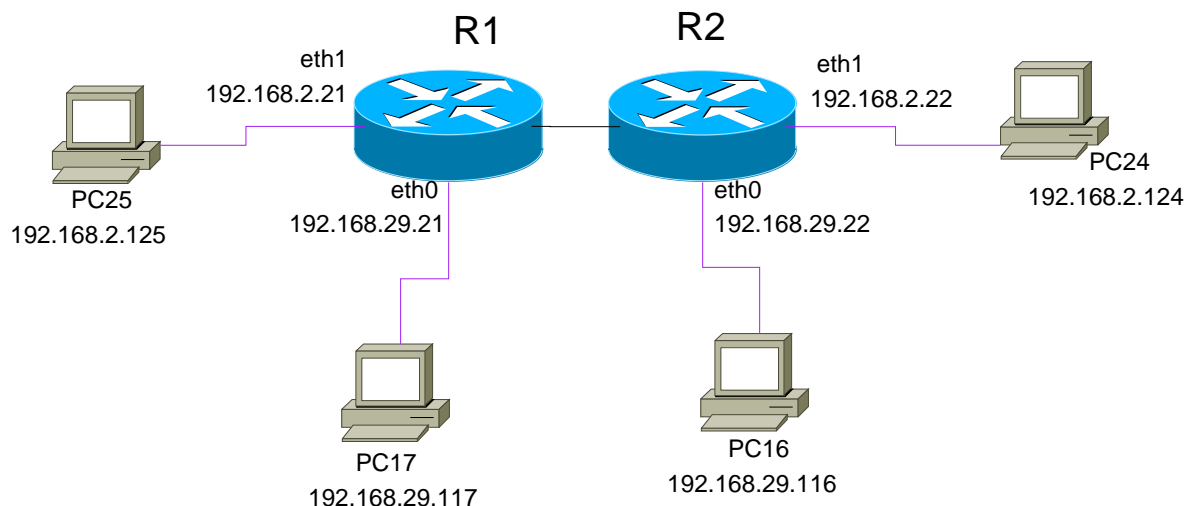


Figura 2: Situación intermedia

- Configurar R1 desde PC17

```
R1# ip addr add 192.168.201.1/24 dev eth0
R1# ip addr del 192.168.29.21/24 dev eth0
```

Tras la ejecución de este último comando se perderá la conexión entre PC17 y R1, debiéndose reconfigurar la interface ethernet de PC17 con la dirección IP marcada en la arquitectura de la figura (192.168.201.117), para recuperar esa conexión. Esto, dependiendo del sistema operativo se hace con el comando:

```
Linux: PC17 # ip addr add 192.168.201.117/24 dev eth0
Windows: C:\> netsh interface ip set address local1 static 192.168.201.117 255.255.255.0
```

Se deberá recuperar la conectividad ssh con R1 antes de proceder a realizar el cambio desde PC25.

- Configurar R1 desde PC25

```
R1# ip addr add 192.168.203.1/24 dev eth1
R1# ip addr del 192.168.2.21/24 dev eth1
```

Tras la ejecución de este último comando se perderá la conexión entre PC25 y R1, debiéndose reconfigurar la interface ethernet de PC25 con la dirección IP marcada en la arquitectura de la figura (192.168.203.125), para recuperar esa conexión. Esto, dependiendo del sistema operativo se hace con el comando:

```
Linux: PC25 # ifconfig eth0 192.168.203.125/24
Windows: C:\> netsh interface ip set address local1 static 192.168.203.125 255.255.255.0
```

- Configurar R2 desde PC16

```
R2# ip addr add 192.168.202.1/24 dev eth0
R2# ip addr del 192.168.29.22/24 dev eth0
```

Tras la ejecución de este último comando se perderá la conexión entre PC16 y R2, debiéndose reconfigurar la interface ethernet de PC16 con la dirección IP marcada en la arquitectura de la figura

(192.168.202.116), para recuperar esa conexión. Esto, dependiendo del sistema operativo se hace con el comando:

```
Linux:      PC16 # ifconfig eth0 192.168.202.116/24
Windows:    C:\> netsh interface ip set address local1 static 192.168.202.116 255.255.255.0
```

Se deberá recuperar la conectividad ssh con R1 antes de proceder a realizar el cambio desde PC25.

■ Configurar R2 desde PC24

```
R2# ip addr add 192.168.204.1/24 dev eth1
R2# ip addr del 192.168.2.22/24 dev eth1
```

Tras la ejecución de este último comando se perderá la conexión entre PC24 y R2, debiéndose reconfigurar la interface ethernet de PC24 con la dirección IP marcada en la arquitectura de la figura (192.168.204.124), para recuperar esa conexión. Esto, dependiendo del sistema operativo se hace con el comando:

```
Linux:      PC24 # ifconfig eth0 192.168.204.124/24
Windows:    C:\> netsh interface ip set address local1 static 192.168.204.124 255.255.255.0
```

La interfaz eth2 en R1 y R2 puede configurarse de cualquiera de los equipos gestores asignados al router respectivo, y se realiza de nuevo con el comando:

```
R1# ip addr add 10.0.0.1/8 dev eth2
R2# ip addr add 10.0.0.2/8 dev eth2
```

2.2.1. Esquema de direccionamiento final

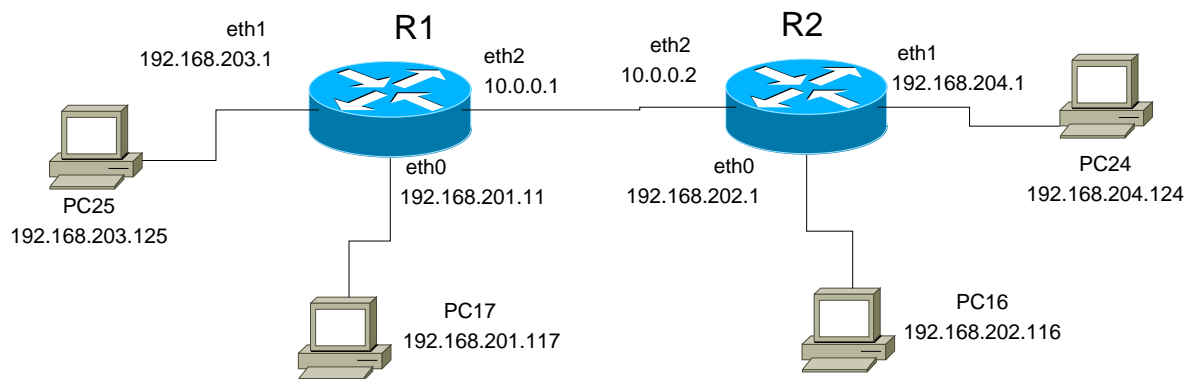


Figura 3: Situación final

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.

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:

```
R1# ip route add <red destino/mascara> via <gateway> dev <interfaz de salida>
```

Para las rutas por defecto de los equipos gestores, se utilizará el comando “netsh” o “route” si se está trabajando sobre un sistema operativo Microsoft (Ver para el comando “netsh routing ip”: <http://technet2.microsoft.com/WindowsServer/en/library/cb852ab7-d19f-4230-b6e7-5422691f33ad1033.mspx?mfr=true>) o el comando “route” o “ip route” si se trabaja sobre un sistema operativo tipo Linux. La inserción de estas rutas puede realizarse desde cualquiera de los equipos gestores que le corresponda a cada router, y estas rutas serán:

- En R1 Rutas directas ya configuradas (redes A1 y A3). Pueden visualizarse con el comando

```
R1 # ip route
```

Ruta hacia A2 (192.168.202.0/24) a través de R2

```
R1# ip route add 192.168.202.0/24 via 10.0.0.2 dev eth2
```

Ruta hacia A4 (192.168.204.0/24) a través de R2

```
R1# ip route add 192.168.204.0/24 via 10.0.0.2 dev eth2
```

Ruta hacia Internet (ruta por defecto 0.0.0.0/0) a través de F1

```
R1# ip route add default via 192.168.201.2 dev eth0
```

- En R2

Rutas directas ya configuradas (redes A2 y A4). Pueden visualizarse con el comando

```
R2 # ip route
```

Ruta hacia A1 (192.168.201.0/24) a través de R1

```
R2 # ip route add 192.168.201.0/24 via 10.0.0.1 dev eth2
```

Ruta hacia A3 (192.168.203.0/24) a través de R1

```
R2 # ip route add 192.168.203.0/24 via 10.0.0.1 dev eth2
```

Ruta hacia Internet (ruta por defecto 0.0.0.0/0) a través de F2

```
R2# ip route add default via 192.168.202.2 dev eth0
```

Rutas por defecto en los equipos

- Ruta por defecto en PC16

Esta será la interfaz eth0 de R2 (192.168.202.1). Los comandos a ejecutar dependerán del sistema operativo que se esté utilizando, así:

```
Linux:      PC16 # ip route add default via 192.168.202.1
Windows:    C:\> netsh routing ip add dest=0.0.0.0 mask=0.0.0.0 name="local1" nhop←
             =192.168.202.1
Windows:    C:\ route add default 192.168.202.1
```

- Ruta por defecto en PC17

Esta será la interfaz eth0 de R1 (192.168.201.1). Los comandos a ejecutar dependerán del sistema operativo que se esté utilizando, así:

```
Linux:      PC17 # ip route add default via 192.168.201.1

Windows:    C:\> netsh routing ip add dest=0.0.0.0 mask=0.0.0.0 name="local1" nhop↔
             =192.168.201.1

Windows:    C:\ route add default 192.168.201.1
```

- Ruta por defecto en PC25

Esta será la interfaz eth1 de R1 (192.168.203.1). Los comandos a ejecutar dependerán del sistema operativo que se esté utilizando, así:

```
Linux:      PC25 # ip route add default via 192.168.203.1

Windows:    C:\> netsh routing ip add dest=0.0.0.0 mask=0.0.0.0 name="local1" nhop↔
             =192.168.203.1

Windows:    C:\ route add default 192.168.203.1
```

- Ruta por defecto en PC24

Esta será la interfaz eth1 de R2 (192.168.204.1). Los comandos a ejecutar dependerán del sistema operativo que se esté utilizando, así:

```
Linux:      PC24 # ip route add default via 192.168.204.1

Windows:    C:\> netsh routing ip add dest=0.0.0.0 mask=0.0.0.0 name="local1" nhop↔
             =192.168.204.1

Windows:    C:\ route add default 192.168.204.1
```

4. Varias salidas a Internet

La idea genérica de que un dispositivo (o red) puedan 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*.

Básicamente parte de la base de que existen varias tablas de enrutamiento, 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.

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 F1 y otra por F2. De esa forma R1 puede seleccionar que la red A1 salga por F1 y la red A3 salga por F2. Al igual puede hacerse con R2, que la red A2 salga por F2 y la red A4 salga por F1.

Primeramente será necesario conocer las tablas de enrutamiento operativas, que en general son con las que arranca el sistema:

```

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. 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:

```

R1:~#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:

```
R1:~#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:

En R1:

```
R1# echo 200 TablaF2 >> /etc/iproute2/rt_tables
```

En R2:

```
R2# echo 200 TablaF1 >> /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:

En R1:

```
R1# ip route add 192.168.201.0/24 via 192.168.201.1 dev eth0 table TablaF2
R1# ip route add 192.168.203.0/24 via 192.168.203.1 dev eth1 table TablaF2
R1# ip route add default via 10.0.0.2 dev eth2 table TablaF2
R1# ip route show table TablaF2
default via 10.0.0.2 dev eth2
192.168.201.0/24 via 192.168.201.1 dev eth0
192.168.203.0/24 via 192.168.203.1 dev eth1
```

En R2:

```
R2# ip route add 192.168.202.0/24 via 192.168.202.1 dev eth0 table TablaF1
R2# ip route add 192.168.204.0/24 via 192.168.204.1 dev eth1 table TablaF1
R2# ip route add default via 10.0.0.1 dev eth2 table TablaF1
R2# ip route show table TablaF1
default via 10.0.0.1 dev eth2
192.168.204.0/24 via 192.168.204.1 dev eth1
192.168.202.0/24 via 192.168.202.1 dev eth0
```

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 (TablaF1 o TablaF2 según corresponda), se crearán nuevas reglas con el comando `ip rule`:

En R1:

```
R1# ip rule add from 192.168.203.0/24 table TablaF2 prio 1000
R1# ip rule
0:      from all lookup local
1000:   from 192.168.203.0/24 lookup TablaF2
32766:  from all lookup main
32767:  from all lookup default
```

En R2:

```
R2# ip rule add from 192.168.204.0/24 table TablaF1 prio 1000
R2# ip rule
0:      from all lookup local
1000:   from 192.168.204.0/24 lookup TablaF1
32766:  from all lookup main
32767:  from all lookup default
```

que viene a decir que cualquier paquete que llegue con una dirección de la red 192.168.203.0/24 utilizará la tabla TablaF2 (o similar en R2).

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 TablaF1 o TablaF2).

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 porqué ruta enviarlo, dependiendo del parámetro “weight”.

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

En R1:

```
R1# ip route add default scope global nexthop via 192.168.201.2 dev eth0 weight 1 nexthop via ↵
    10.0.0.2 dev eth2 weight 1
```

En R2:

```
R2# ip route add default scope global nexthop via 192.168.202.2 dev eth0 weight 1 nexthop via ↵
    10.0.0.1 dev eth2 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*.

En R1:

```

R1# ip route add default equalize nexthop via 192.168.201.2 dev eth0 weight 1 nexthop via ↵
    10.0.0.2 dev eth2 weight 1
R1# ip route show
default equalize
    nexthop via 192.168.201.2 dev eth0 weight 1
    nexthop via 10.0.0.2 dev eth2 weight 1

```

En R2:

```

R2# ip route add default equalize nexthop via 192.168.202.2 dev eth0 weight 1 nexthop via ↵
    10.0.0.1 dev eth2 weight 1
R2# ip route show
default equalize
    nexthop via 192.168.202.2 dev eth0 weight 1
    nexthop via 10.0.0.1 dev eth2 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. 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.

R1 <-> R2 R1(R2) <-> PC17, PC16, PC24, PC25 R1(R2) <-> Resto de equipos R1(R2) <-> INTERNET
R1(R2) <->

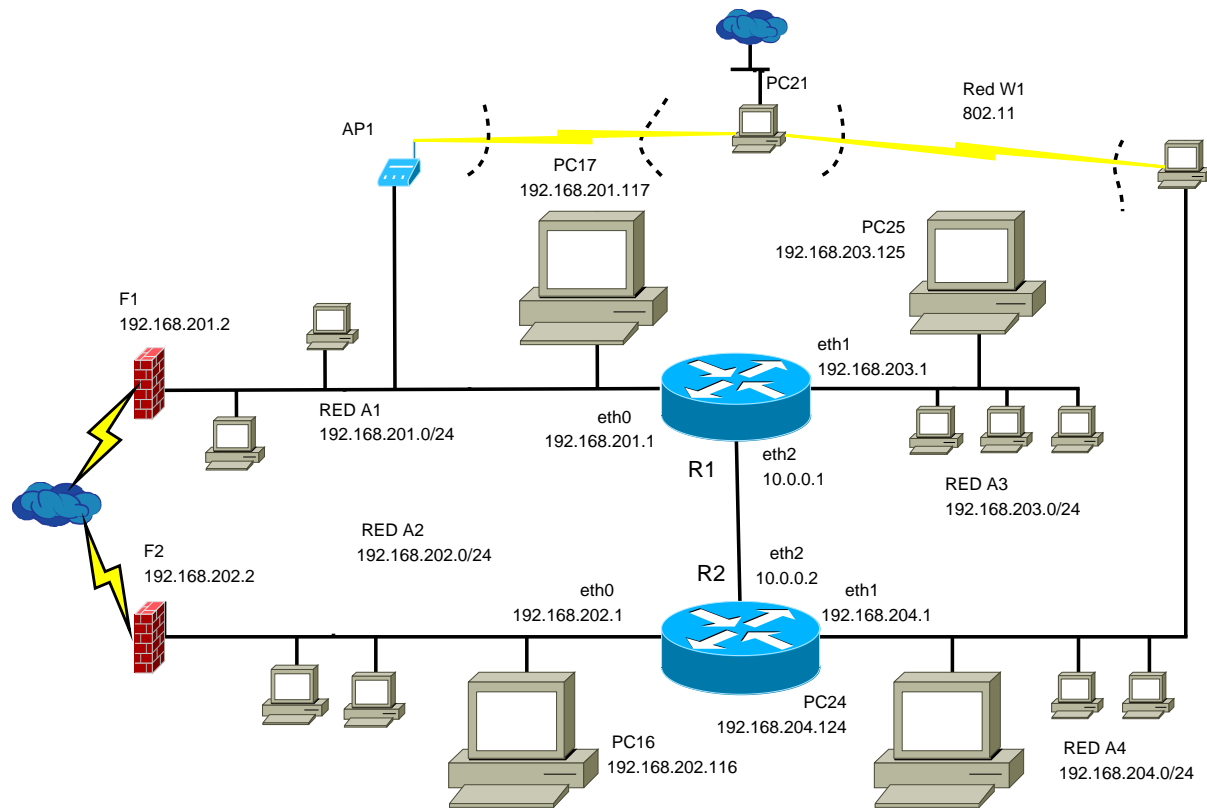


Figura 4: Visión general de la práctica