

Configuración e instalación red de area local

Xavier Gonzalez
Juan Luis Guillen
Miquel Testar
Ivan Morgado
Pere Toran
David Pérez

ROUTER

- Definición
- Router hardware
 - Protocolos de enrutamiento
- Router software

DEFINICIÓN

Un router es un dispositivo de hardware o de software que interconecta segmentos de red o redes enteras de ordenadores transportando paquetes de datos.

ROUTER HARDWARE

Son dispositivos capaces de tomar la decisión adecuada para mandar el paquete de datos por el camino más rápido según las circunstancias en cada momento.

ROUTER HARDWARE

Los protocolos de enrutamiento más utilizados son RIP, OSPF y BGP, que se distinguen por los algoritmos que utilizan para calcular las rutas. Todos ellos lo hacen de forma dinámica.

ROUTER SOFTWARE

Uso de un PC como router para redes pequeñas.

Muy simple con un SO GNU/Linux:
Añadir 2 interfaces de red y activar el soporte del kernel.

Opcionalmente: Quagga, Zebra o ZebOs. Actúan como daemons.

FIREWALL

- Definición
- Tipos:
 - De capa de red o filtrado de paquetes
 - De capa de aplicación
 - Personal

DEFINICIÓN

Es un elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas según la configuración del mismo.

TIPOS

DE CAPA DE RED

Actua como filtro de paquetes IP.
Permite o no la transmisión de datos
según la IP de origen o de destino.

TIPOS

DE CAPA DE APLICACIÓN

Se adapta al protocolo utilizado.
Por ejemplo, en tráfico HTTP (donde se conoce como Proxy) realiza filtrados según la URL a la que se está intentando acceder.

TIPOS

PERSONAL

Se instala en forma de software y controla el intercambio de información entre ese ordenador y el resto de la red.

CONFIGURACIÓN MÍNIMA DEL HARDWARE

- Un PC que actúe como servidor con acceso a Internet y interface ethernet.
- Resto de PC's con interface ethernet.
- Hub (red tipo estrella) o switch (más rápido, más directo, más seguro) para gestionar el envío de información.
- Cables con conectores RJ-45.

IPTABLES

- Es una aplicación en línea de comandos que gestiona el filtrado de paquetes en sistemas Linux
- Su función consiste básicamente en analizar todo el flujo de tráfico entrante y saliente hacia/desde él y tomar unas decisiones sobre cada paquete en base a unas reglas definidas.

Estructura iptables (I)

- La estructura del comando es la siguiente:

```
iptables <ubicacion> <especificacion>  
<accion>
```

- Para determinar la ubicación de la regla que se esta agregando se utilizan las siguientes opciones:

-t «tabla»

- indica en qué tabla se va a ubicar la reglas

-A «cadena»

- agrega una regla al final de la lista de reglas de la «cadena»

-I «cadena» [pos]

- inserta una regla dentro de la cadena «cadena» en la posición [pos]

Estructura iptables (II)

- Para determinar la especificación del paquete se utilizan estas opciones generalmente:
 - s «dirección»**
 - indica que el paquete proviene de la dirección «dirección», si se le agrega un ! se niega la opción
 - d «dirección»**
 - indica que el paquete va destinado a la dirección «dirección».
 - i «interfase»**
 - indica la interface de entrada
 - o «interfase»**
 - indica la interfase de salida
 - p «protocolo»**
 - indica el protocolo del paquete (los más comunes son tcp, udp, icmp)
 - -dport «puerto»**
 - indica el número de puerto destino
 - -sport «puerto»**
 - indica el número de puerto origen
 - m «módulo»**
 - indica que se va a utilizar un módulo en particular

Estructura iptables (III)

- Para determinar la acción de la regla se utiliza una única opción:

-j «acción o cadena»

- Con esta opción le indicamos que cuando un paquete coincida con las características expresadas en la regla, se deberá tomar la acción «acción» o saltar a la cadena «cadena».

Dependiendo la acción el kernel dejará de verificar las subsiguientes reglas o no.

Ejemplos de reglas

Aceptar conexiones al puerto 80 en la tarjeta eth0

```
iptables -A INPUT -i eth0 -s 0.0.0.0/0 -p TCP --dport 80 -j ACCEPT
```

Redireccionar al puerto 3128 (proxy) todos los paquetes que entran por eth1 y con destino puerto 80 (HTTP), de esta manera conseguimos un proxy transparente.

```
iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

- D_{inamic}
- H_{ost}
- C_{onfiguration}
- P_{rotocol}

Introducción a DHCP

- DHCP es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente.
- Facilita incluir nuevos equipos en una red, y cambiar la configuración de todos ellos.

Reseña histórica(I)

- 1984 nace el RARP (bajo nivel)
- 1985 publicación BootP
- 1993 se publica el protocolo DHCP
- 1994 Servidores Win NT 3.5 lo incluyen
- 1997 ISC publica version 1.0 para UNIX.

Reseña histórica(II)

- 1999 Cisco IOS 12.0 con *server* DHCP
- 1999 version 2.0 para UNIX (ajustada a RFC)
- 2001 Sun añade DHCP al sistema Solaris
- 2003 se publica DHCPv6.

Opciones configurables

- Dirección del servidor DNS
- Nombre DNS
- Puerta de enlace de la dirección IP
- Dirección de Publicación Masiva (broadcast address)
- Máscara de subred
- Tiempo máximo de espera del ARP
- MTU (*Unidad de Transferencia Máxima*) para la interfaz
- Servidores NIS (*Servicio de Información de Red*)
- Dominios NIS
- Servidores NTP (*Protocolo de Tiempo de Red*)
- Servidor SMTP
- Servidor TFTP
- Nombre del servidor WINS

Métodos de funcionamiento

- Estático: Se asignan pares de direcciones MAC i IP. (equivalente a BOOTP)
- Automático: Se asigna una IP de forma permanente.
- Dinámico: El cliente obtiene una IP para un tiempo limitado.

Protocolo Obtención

- DHCP Discover (Cliente manda Broadcast)
- DHCP Offer (servidores disponibles responden)
- DHCP Request (El cliente escoge)
- DHCP Ack (Servidor asocia IP - MAC)

Protocolo mantenimiento

- DHCP Request (50% del tiempo asignado)
 - DHCP Ack (renovamos)
 - DHCP Nack (liberamos)
 - Silencio (esperamos 87.5%)
- DHCP Realease (liberación antes de hora)

Ventajas

- Ahorro de tiempo a la hora de configurar una subred.
- Permite aprovechar mejor IPv4. Ya que no hace falta mantener una IP para una MAC que no opera.
- Permite trabajar a máquinas sin disco en una subred

Inconvenientes

- Seguridad
- Incremento de tráfico en la red.

Configuración y puesta en marcha de un servidor DHCP

- Instalación del paquete DHCP
- Configuración del servidor
- Ejemplos de configuración
- Consideraciones de seguridad
- Puesta en marcha

Instalación del paquete DHCP

- `yum -y install dhcp` (Fedora)
- `up2date -i dhcp` (Si utilizas Red Hat Enterprise Linux o versiones posteriores)
- `apt-get install dhcp` (Debian, Ubuntu)

Configuración del servidor(I)

- Debemos conocer los datos de configuración de la red. Ejemplo:

Número de red 192.168.0.0

Máscara de sub-red: 255.255.255.0

Puerta de enlace: 192.168.0.1

Servidor de nombres: 192.168.0.1, 148.240.241.42 y 148.240.241.10

Servidor Wins: 192.168.0.1

Servidores de tiempo: 66.187.224.4 y 66.187.233.4

Rango de direcciones IP a asignar de modo dinámico: 192.168.0.11-192.168.0.199

Configuración del servidor(II)

- Creamos/Modificamos el fichero /etc/dhcpd.conf. Ejemplo:

```
ddns-update-style interim;
ignore client-updates;
shared-network miredlocal {
  subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.0.255;
    option domain-name "redlocal.net";
    option domain-name-servers 192.168.0.1, 148.240.241.42, 148.240.241.10;
    option netbios-name-servers 192.168.0.1;
    option ntp-servers 66.187.224.4, 66.187.233.4;
    range 192.168.0.11 192.168.0.199;
    default-lease-time 21600;
    max-lease-time 43200;
  }
  host m253 {
    option host-name "m253.redlocal.net";
    hardware ethernet 00:50:BF:27:1C:1C;
    fixed-address 192.168.0.253;
  }
  host m254 {
    option host-name "m254.redlocal.net";
    hardware ethernet 00:01:03:DC:67:23;
    fixed-address 192.168.0.254;
  }
}
```

Ejemplos de configuración(I)

- **Declaración de Subred**

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    option routers
    192.168.1.254;
    option subnet_mask
    255.255.255.0;

    option domain_name
    "example.com";
    option domain_name_servers
    192.168.1.1;

    option time_offset
    range 192.168.1.10 192.168.1.100;    _18000;
}
```

Ejemplos de configuración(II)

- **Declaración de shared-network, o red compartida**

```
shared_network mi_red local {
    option domain_name "test.fedora.com";
    option domain_name_servers ns1.fedora.com,
ns2.fedora.com;
    option routers 192.168.1.254;
    ---Declaración de subredes específicas---
    subnet 192.168.1.0 netmask 255.255.255.0 {
        parameters for subnet
        range 192.168.1.1 192.168.1.31;
    }
    subnet 192.168.1.32 netmask 255.255.255.0 {
        parameters for subnet
        range 192.168.1.33 192.168.1.63;
    }
}
```

Ejemplos de configuración(III)

- **Declaración de Grupo**

```
group {
  option routers                192.168.1.254;
  option subnet_mask            255.255.255.0;

  option domain_name            "example.com";
  option domain_name_servers    192.168.1.1;

  option time_offset            _18000;

  host apex {
    option host_name "apex.example.com";
    hardware ethernet 00:A0:78:8E:9E:AA;
    fixed_address 192.168.1.4;
  }

  host raleigh {
    option host_name "raleigh.example.com";
    hardware ethernet 00:A1:DD:74:C3:F2;
    fixed_address 192.168.1.6;
  }
}
```

Consideraciones de seguridad

- El servidor de DHCP utiliza el puerto 67 y 68 a través de UDP para recibir y enviar datos a los clientes. Es por esto que lo primero que deberemos retocar serán las reglas de nuestro cortafuegos, si es que este existe y si es que el servidor de DHCP comparte con él la misma máquina. Suponiendo que tenemos instalado iptables, la regla a introducir en el fichero de configuración será la siguiente:

```
$IPTABLES -I INPUT -i $LAN_IFACE -p udp --dport 67:68 --sport \
67:68 -j ACCEPT
```

Con esta regla se permite el tráfico hacia y desde los puertos UDP 67 y 68, lo cual nos deja abierto un pequeño agujero en el cortafuegos que permite el funcionamiento del servidor de DHCP.

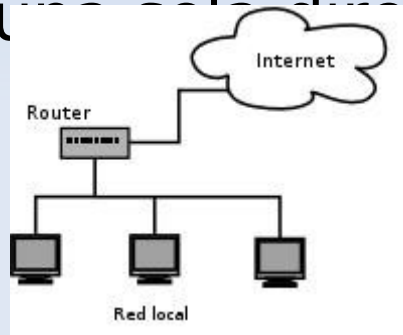
- En el caso de tener varios dispositivos de red configurados en la misma máquina, habrá que asegurarse de que nuestro servidor de DHCP sólo se ejecuta sobre el que tiene acceso a la red a la que va a proveer de servicio. En un sistema con *Fedora Core*, tenemos a nuestra disposición un fichero de configuración en la ruta `/etc/sysconfig/dhcpd`. Agregaremos lo siguiente a dicho fichero:

```
DHCPDARGS=eth0
```

eth0 se cambiará por el interfaz usado por DHCP.

NAT (I)

- La Traducción de Direcciones de Red, o NAT (*Network Address Translation*), es un sistema que se utiliza para asignar una red completa (o varias redes) a una sola dirección IP.



- NAT es necesario cuando la cantidad de direcciones IP que nos haya asignado nuestro proveedor de Internet sea inferior a la cantidad de ordenadores que queremos que accedan a Internet.

NAT (II)

- Cuando un cliente en la red interna contacta con un máquina en Internet, envía paquetes IP destinados a esa máquina. Estos paquetes contienen toda la información de direccionamiento necesaria para que puedan ser llevados a su destino. NAT se encarga de estas piezas de información:

Dirección IP de origen (por ejemplo, 192.168.1.35)

Puerto TCP o UDP de origen (por ejemplo, 2132)

- Cuando los paquetes pasan a través de la pasarela de NAT, son modificados para que parezca que se han originado y provienen de la misma pasarela de NAT. La pasarela de NAT registra los cambios que realiza en su tabla de estado, para así poder:
 - a) invertir los cambios en los paquetes devueltos
 - b) asegurarse de que los paquetes devueltos pasen a través del cortafuegos y no sean bloqueados.

Configuración de NAT

El formato general para las reglas de NAT en /etc/pf.conf es parecido al siguiente:

```
nat on extif [af] from src_addr [port src_port] to \ dst_addr [port dst_port]  
-> ext_addr
```

- *extif*
 - El nombre de la interfaz de red.
- *af*
 - La familia de direcciones, que será inet para IPv4 ó inet6 para IPv6.
- *src_addr*
 - La dirección de origen (interna) de los paquetes que vayan a ser traducidos.
- *src_port*
 - El puerto de origen en la capa Layer 4 de la cabecera del paquete. en el campo.
- *dst_addr*
 - La dirección de destino de los paquetes que hay que traducir.
- *dst_port*
 - El puerto de destino en la capa Layer 4 de la cabecera del paquete.
- *ext_addr*
 - La dirección externa (la traducción) en la pasarela NAT a la que se traducirán los paquetes.

Ejemplo de regla NAT

- `nat on tl0 from 192.168.1.0/24:network to any -> 24.5.0.5`

Esta regla indica que hay que realizar NAT en la interfaz tl0 para cualquier paquete que venga de 192.168.1.0/24, y sustituir la dirección IP de origen con 24.5.0.5.

- `no nat on tl0 from 192.168.1.10 to any`
`nat on tl0 from 192.168.1.0/24 to any -> 24.2.74.79`

Con esta modificación los paquetes de 192.168.1.10 no serían traducidos, no cumplirían la norma.

Ejemplo de lan en linux

Qué necesitamos?

- **Un switch (en local)**
- **Un router (para salir a internet)**
- **Dos o más ordenadores**
- **Una tarjeta ethernet en cada ordenador**

Cada ordenador necesita una IP para identificarse.

Dos formas de configuración:

- **Usando DHCP -> más cómodo (autoconfiguración).**
- **Sin usar DHCP -> más laborioso, lo haremos así :)**

Sin DHCP (asignamos ip's, gateway, ... manualmente):

- **Mediante el comando ifconfig**
- **Mediante el archivo /etc/network/interfaces**

Ejemplo de lan en linux

Ifconfig: Comando unix usado para configurar las interfaces de red.

Qué opciones trae ifconfig? (las más básicas)

- ***Interface: El nombre de la interfaz***
- ***up/down: Activa/desactiva la interfaz de red***
- ***adress: La ip que asignaremos a esta interfaz***
- ***mtu N: Unidad máxima de transferencia para la interfaz***
- ***netmask addr: Asigna la máscara de red para ifconfig***

***ifconfig eth0 192.168.0.x netmask 255.255.255.0
up***

Activa la interfaz eth0 con una ip 192.168.0.x y una máscara de red 255.255.255.0

Repetimos la misma instrucción en cada ordenador, modificando el caracter x (1..254).

Ejemplo de lan en linux

/etc/network/interfaces: Archivo de configuración de las interfaces de red.

Se estructura con un conjunto de “stanzas” (etiquetas) y opciones dentro de cada “stanza”.

Qué etiquetas trae?

- ***auto: Para identificar interfaces físicas.***
- ***mapping: Para crear interfaces lógicas a partir de interfaces físicas.***
- ***iface: Indica el inicio de configuración de cada interfaz lógica (si no hay irá antes de cada interfaz física).***
- ***allow-: Permite activar la interfaz dependiendo de algun subsistema.***

Ejemplo de lan en linux

Parámetros para iface (opciones de configuración):

- ***NAME: Nombre de la interfaz lógica (si no hay, física).***
- ***ADDRFAM: Protocolo de internet (inet para TCP/IP, ipx, inet6)***
- ***METHOD: Modo de configuración (static, dhcp,...).***

Si el método es estático debes indicarle la ip y la máscara.

Ejemplo de lan en linux

Configuramos interfaces pero como las activo/desactivo?

Mediante los comandos *ifup*, *ifdown*:

- ***ifup*: Activa la interface.**
- ***ifdown*: Desactiva la interface.**

***ifup -a*: Activa todas las interfaces definidas por "auto".**

***ifup eth0*: Activa la interfaz eth0.**

***ifup eth0=home*: Activa la interfaz física eth0 con el nombre de interfaz lógica home.**

***ifdown -a*: Desactiva todas las interfaces.**

Ejemplo de lan en linux

Ejemplo del fichero /etc/network/interfaces:

```
auto lo eth0  
allow-hotplug eth1  
iface lo inet loopback
```

```
mapping eth0  
script /usr/local/sbin/map-scheme  
map HOME eth0-home  
map WORK eth0-work
```

```
iface eth0-home inet static  
address 192.168.1.1  
netmask 255.255.255.0  
up flush-mail
```

```
iface eth0-work inet dhcp  
iface eth1 inet dhcp
```

Ejemplo de lan en linux

Hasta ahora no podíamos salir a internet, qué nos falta?

- **Servidor DNS**
- **Route**

Cómo configurar el servidor DNS?

- **Con el fichero /etc/resolv.conf**

Opciones de configuración de /etc/resolv.conf:

- **nameserver: Dirección ip del servidor dns.**
- **domain: dominio de la red donde estás conectado.**
- **search: Lista de dominios a buscar si no pones "domain".**
- **sortlist: Lista de direcciones que el servidor resolverá.**

Ejemplo de fichero /etc/resolv.conf:

```
search dummy.net  
nameserver 193.152.63.197
```

Ejemplo de lan en linux

Qué es y para qué sirve “route”?

- **Cuando mandas paquetes con ip's que no se encuentran en tu red a quién se los mandas?**
- **Sirve para enrutar paquetes según su dirección de destino.**
- **Route se muestra en forma de tabla que contiene los campos: destination, gateway, genmask, flags,...**

Cómo configurar route?

Para añadir, según la sintaxis:

**route [-v] [-A family] add [-net|-host] target
[netmask Nm] [gw Gw] [metric N] [mss M] [window
W] [irtt I] [reject] [mod] [dyn] [reinststate] [[dev] If]**

Para borrar, según la sintaxis:

**route [-v] [-A family] del [-net|-host] target [gw Gw]
[netmask Nm] [metric N] [[dev] If]**

Ejemplo de lan en linux

Opciones de configuración:

- **add/del:** Añade/borra una fila de la tabla route.
- **target:** La red o host de destino.
- **-net/-host:** Indica que el target es red o host.
- **netmask:** Mascara de red.
- **gw GW:** Gateway, puerta de enlace.
- **metric, mss, window,...**

```
route add default gw 192.168.1.1
```

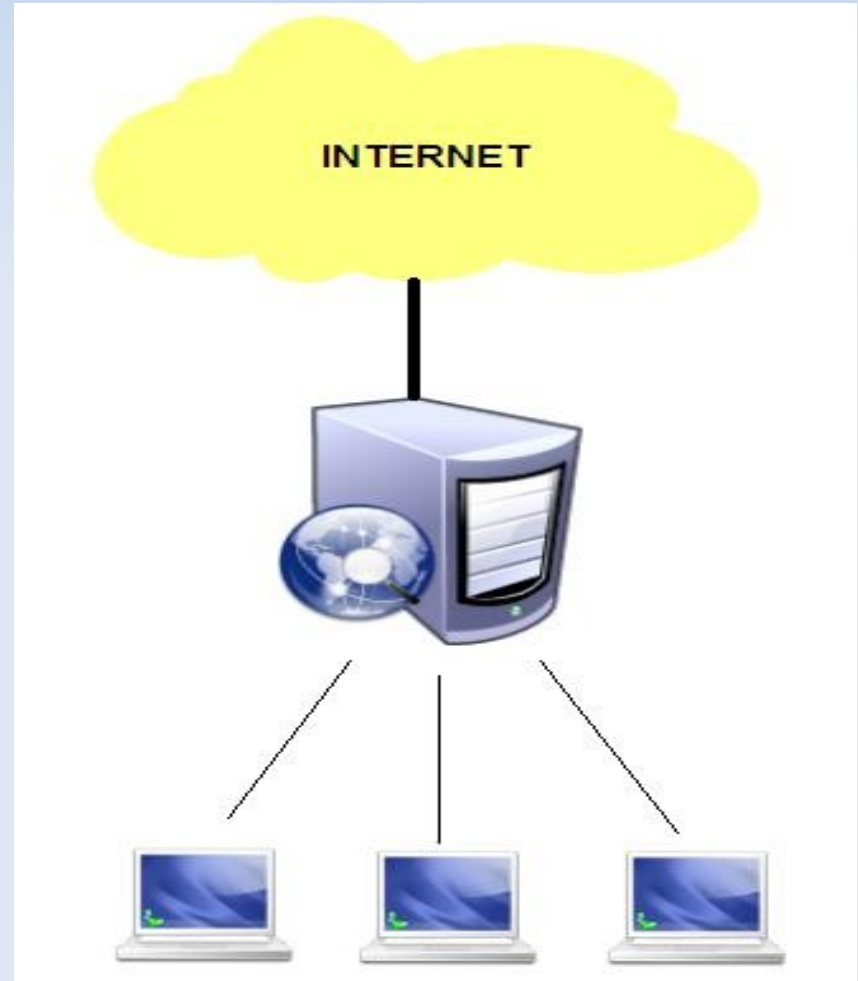
Ejemplo de tabla route:

Destination	Gateway	Genmask	Flags		
Metric	Iface				
192.168.1.0	*	255.255.255.0	U	0	eth1
default	192.168.1.1	0.0.0.0	UG	0	
eth1					

Los paquetes que van a 192.168.1.x ya sabe donde mandarlos. El resto de paquetes se envian a la 192.168.1.1

Configuración red área local

- **Normalmente solo disponemos de un punto de acceso a Internet.**
 - Reaprovechamos direcciones publicas.
 - Tan solo pagamos una dirección.
- **Tendremos 2 zonas diferenciadas, la red de área local y el servidor con**



Pasos para configurar nuestra red

- Configuración del servidor.
 - Configuración del servidor DHCP.
 - Configuración de iptables.
 - Configuración del protocolo de traducción de direcciones.
- Configuración de los clientes.
 - Configuración de las direcciones.

Pasos para configurar nuestra red

- Configuración del servidor.
 - Configuración del servidor DHCP.
 - Configuración de iptables.
 - Configuración del protocolo de traducción de direcciones.
- Configuración de los clientes.
 - Configuración de las direcciones.

- Si nuestro servidor ofrece la asignación de direcciones privadas mediante DHCP tendremos que configurar el servidor DHCP tal y como hemos explicado anteriormente.
- Con esto conseguiremos un mayor reaprovechamiento de direcciones privadas.

Pasos para configurar nuestra red

- Configuración del servidor.
 - Configuración del servidor DHCP.
 - Configuración de iptables.
 - Configuración del protocolo de traducción de direcciones.
- Configuración de los clientes.
 - Configuración de las direcciones.

- A través de la configuración de las reglas mediante iptables conseguiremos definir claramente quien, como y a donde se puede conectar cualquier dirección.

Pasos para configurar nuestra red

- Configuración del servidor.
 - Configuración del servidor DHCP.
 - Configuración de iptables.
 - Configuración del protocolo de traducción de direcciones.
- Configuración de los clientes.
 - Configuración de las direcciones.

- Como ya hemos visto normalmente tan solo se dispone de una dirección pública. De manera que para dotar a todas las direcciones privadas de acceso a Internet necesitaremos configurar algún sistema de traducción de direcciones.
- Tal y como hemos explicado usaremos NAT.
 - Alternativamente se podría usar otros protocolos como PAT.

Pasos para configurar nuestra red

- Configuración del servidor.
 - Configuración del servidor DHCP.
 - Configuración de iptables.
 - Configuración del protocolo de traducción de direcciones.
- Configuración de los clientes.
 - Configuración de las direcciones.

- En el caso de los clientes tan solo hay que configurar la dirección IP, si no se usa DHCP.
- Y añadir a la tabla de routing la entrada por defecto de la dirección privada de nuestro servidor.