



# **Diseño y Administración de Redes**

## **Práctica 1.1**

### **Diseño y gestión de escenarios IPv4. Configuración básica**

Dpto. Ingeniería Electrónica y Comunicaciones  
Área de Ingeniería Telemática



**Departamento de  
Ingeniería Electrónica  
y Comunicaciones**  
**Universidad Zaragoza**

Autores:

**Profesores del área de Ingeniería Telemática**

## 1. Introducción

### 1.1. Objetivos

Tras la realización de esta práctica, el alumno deberá ser capaz de:

Asignar un esquema de direcciones IPv4 de acuerdo a ciertas especificaciones de interconexión de redes.

Decidir y configurar el encaminamiento (estático o dinámico) que es necesario establecer para garantizar la conectividad completa en un escenario de interconexión de redes.

Configurar adecuadamente conexiones básicas mediante tecnología Ethernet.

Analizar y evaluar el comportamiento de un escenario de interconexión de redes:

- Emplear adecuadamente las herramientas de verificación necesarias.

- Detectar posibles errores de configuración, predecir resultados.

### 1.2. Contenidos

Los objetivos propuestos en esta práctica pretenden afianzar y complementar los **contenidos teóricos vistos en clase**. Por lo tanto, será necesario un estudio previo de los mismos, así como la utilización de los apuntes de clase (y cualquier material adicional que el alumno considere oportuno) como apoyo a la realización práctica.

A modo de orientación se enumeran aquellos aspectos de IPv4 que se consideran más relevantes:

- Direccionamiento

- Funcionalidad del Protocolo IPv4: PDU y Primitivas

- Fragmentación y reensamblado

- Encaminamiento

- Funciones de control: apoyo en otros protocolos

- Evaluación: medidas de ancho de banda y retardo

Pueden resultar de utilidad los documentos RFC relativos a los distintos aspectos a analizar: <http://tools.ietf.org/rfc/index>

Del mismo modo se recomienda, en caso necesario, consultar cualquier ayuda online, así como **man** de Linux, o **help** del interfaz de comandos de Windows.

Además, en los anexos se encuentra disponible información auxiliar que será necesaria durante la realización práctica:

Anexo I: Manuales de configuración y programas de análisis.

### 1.3. Equipos, tecnologías y herramientas en escenario real

La práctica propuesta consiste en la configuración, verificación y análisis de un escenario de interconexión de redes IP. Para ello, se trabajará con máquinas Lisa\_Centos y TinyCoreLinux para los diferentes equipos (vistas en la práctica de introducción de GNS3).

En cuanto a las **herramientas** necesarias para la verificación y el análisis de los escenarios, utilizaremos el software de captura **tcpdump** y el analizador de protocolos **Wireshark**.

### 1.4. Escenarios

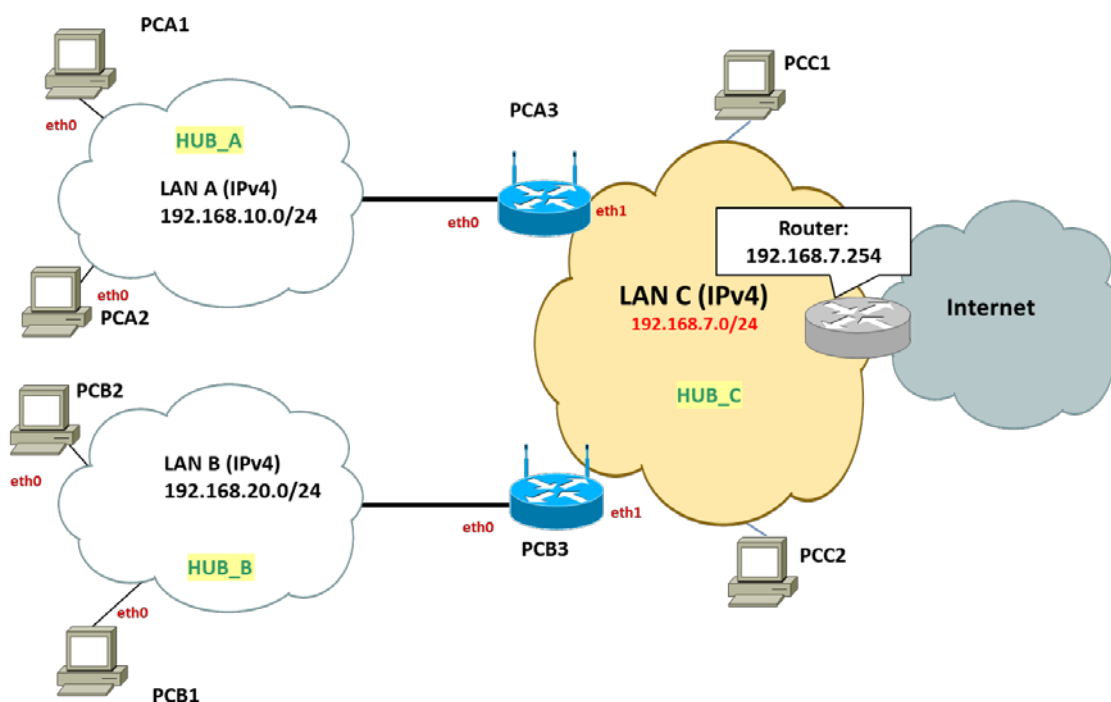


Figura 1: Escenario de interconexión de redes.

La figura 1 muestra el escenario con el que se trabajará en la práctica. En él configuraremos tres redes: LAN\_A, LAN\_B y LAN\_C. Los PC integrantes de las LAN A y B serán PCA1 (centos), PCA2 (tinycore), PCB1 (centos) y PCB2 (tinycore), que actúan como *host*, y PCA3 y PCB3 (ambas en centos) que tendrán funciones de *router* y estarán conectados también a LAN\_C. Se usará PCC1 (centos) y PCC2 (tinycore) para hacer las pruebas y comprobaciones de conexión que se consideren oportunas. Cada LAN utilizará direccionamiento IPv4 privado perteneciente a la red 192.168.X.0/24 donde la X varía en función de la LAN en la que nos encontremos.

En el esquema y en las explicaciones a lo largo de la práctica se hablará de los interfaces de las máquinas nombrándolos como eth0 y eth1. Estos nombres no tienen que ser éstos, sino que pueden cambiar. Es importante que lo tengamos en cuenta.

## 2. Realización de la práctica.

**\*\* La evaluación de la práctica requiere la entrega del escenario GNS3 con la configuración adecuada, las capturas correspondientes que avalen su correcto funcionamiento y un documento con la explicación oportuna. \*\***

**\*\* Para facilitar la elaboración del documento explicativo aparecen una serie de cuestiones a lo largo del enunciado de la práctica. \*\***

**\*\* Habrá cuestiones que hay que resolverlas antes de la práctica y también comprobar el grado de acierto en el transcurso de la misma. La previa se entregará individualmente y el resultado final por grupos \*\***

**\*\* Las cuestiones marcadas como tipo T, se resolverán después de la realización de la práctica y se entregarán individualmente \*\***

### 2.1. Configuración I del escenario: direccionamiento manual y encaminamiento estático

Para las máquinas tinycore seguiremos las explicaciones dadas en la práctica 0. Para la configuración de las máquinas Centos se deberán seguir las pautas indicadas a continuación.

#### 2.1.1. Direccionamiento manual para Centos:

##### Direccionamiento permanente (varía según el sistema Linux):

Esta configuración de direccionamiento y encaminamiento estático puede establecerse de modo permanente entre reinicios del sistema incluyendo la información en los archivos de sistema específicos. Concretamente, deben modificarse los archivos ifcfg-eth0 e ifcfg-eth1 correspondientes a las interfaces eth0 y eth1 respectivamente (sólo modificaremos los que estén en las redes A y B). Dichos archivos se ubican en el siguiente path: /etc/sysconfig/network-scripts/

Una vez modificados los archivos sería necesario realizar un reinicio del sistema (reboot) para que los cambios surtan efecto o ejecutar:

`ifdown eth(0 o 1)` (el que corresponda a las redes A y B)

`ifup eth(0 o 1)`

Lo usaremos para los interfaces de las redes A y B.

#### Direccionamiento no permanente:

El direccionamiento puede asignarse de manera no permanente, es decir, que no se mantiene entre reinicios del sistema o del servicio de red. Aunque este direccionamiento no es el más recomendado, puede ser adecuado, puesto que es más fácil de modificar. Para ello, se pueden utilizar los comando ***ifconfig ...*** o ***ip -4 addr ....*** Es conveniente usar el comando ***ip*** dado que ***ifconfig*** está ya en desuso en los nuevos sistemas. Algunos detalles acerca de su utilización se pueden consultar en el Anexo I.2.A (Sistema operativo Linux, configuración TCP/IP).

Lo usaremos para los interfaces de la red C. Deberemos asegurarnos que los ficheros `ifcfg-eth0` e `ifcfg-eth1` correspondientes sí que existan pero que no tengan ninguna configuración para no entrar en conflicto y que al ejecutarlos se borre la configuración. Además, será necesario crear un script de Shell donde guardemos los comandos utilizados para poderlo ejecutar en el caso de querer reiniciar la configuración (por ejemplo, en prácticas siguientes).

#### 2.1.2. Encaminamiento estático:

Para los PC internos (1) de las redes A y B, configuraremos el PC 3 como router por defecto.

**`ip route add 0.0.0.0/0 via <@IP_router>`**

Para los 4 PC de la red C definiremos la máquina denominada router como router por defecto utilizando el mismo comando descrito anteriormente.

Los PC 3 de las redes A y B deben tener habilitada la funcionalidad de encaminamiento por lo que es necesario que sean capaces de reenviar la información recibida de una interfaz a aquélla que sea necesario (de acuerdo a su tabla de rutas). La configuración de dicho reenvío (***forwarding***) exige la modificación de parámetros internos a los que podemos tener acceso mediante el comando ***sysctl*** (ver Anexo I.2.A, apartado Ficheros de configuración (interfaz `sysctl`)).

En los equipos PCC1 y PCC2 de la red C deberemos configurar el encaminamiento estático para conectarnos a las redes A y B. Para ello se utilizará el comando ***ip route***, siguiendo los ejemplos básicos mostrados a continuación. La información completa de dicho comando se detalla en el Anexo I.2.A (Sistema operativo Linux, Configuración TCP/IP):

**`ip route add <@red>/netmask via <@IP_router>`**

**`ip route add <@IPhost>/32 via <@IP_router>`**

#### 2.1.3. Pruebas de configuración

Una vez establecidas las configuraciones pertinentes, debemos ser capaces de comprobarlo.

Para comprobar el correcto funcionamiento del escenario vamos a ir realizando conexiones desde PC1 y PC2 a PC3 y los PC de la red C mediante **ping**:

Para una correcta comprobación, mientras realizamos cada uno de los **ping** vamos a capturar en los interfaces eth0 y eth1 de los *router* correspondientes para comprobar que el datagrama entra y sale por el interfaz correcto.

Existen herramientas que nos permiten comprobar las conexiones sin necesidad de tener acceso a los equipos por los que pasa la comunicación. La primera que vamos a utilizar es *traceroute* que en Linux se ejecuta mediante el comando **traceroute** y en Windows mediante el comando **tracert**. Vamos a utilizarla para corroborar el correcto funcionamiento de nuestro escenario IP. Tendremos que tener en cuenta que la comunicación atraviesa máquinas con dos direcciones IP (una en cada interfaz) y que nos devuelve la dirección IP por la que pasa, pero siempre mirando hacia el origen. Permite hasta 30 saltos. También, vamos a utilizar el comando **ping -R**. En este caso nos devuelve la dirección IP del interfaz de salida por los que va pasando la comunicación. Permite hasta 9 saltos contando ida y vuelta.

**Cuestión 1 (habrá que resolverla también previamente indicando qué debería obtenerse si ejecutamos los comandos).** Cuando el escenario esté correctamente configurado, verificarlo en PCA1 y PCA2 mediante las herramientas **traceroute** (en la máquina tinycore) y **ping -R** (en Centos) (Anexo I.2.B, las equivalentes para windows serían **tracert** y **ping -r 9**, Anexo I.2.B) y comprobándolo mediante las capturas de wireshark correspondientes a los tres saltos de la comunicación. Indica sobre el ejemplo concreto de comunicación desde PCA1 y PCA2 hacia PCB1, qué información proporciona cada una de estas herramientas y relaciónala con lo capturado.

## 2.2. Configuración II del escenario: direccionamiento manual y encaminamiento dinámico

Los equipos PCA1, PCA2, PCB1 y PCB2 no sufren modificaciones en la configuración, y mantiene las del apartado anterior. Por lo tanto, deberemos configurar los archivos necesarios para establecer el encaminamiento dinámico en los PC de la red C.

### 2.2.1. Configuración de encaminamiento dinámico en los router (PC3):

Debemos, de nuevo comprobar la configuración del reenvío (**forwarding**) mediante el comando **sysctl** (ver Anexo I.2.A, apartado Ficheros de configuración (interfaz sysctl)).

Una vez establecidas las direcciones y el encaminamiento en PCA1, PCA2, PCB1 y PCB2, así como el direccionamiento manual del interfaz eth0 de PCA3 y PCB3 y la función de reenvío en ambos, se configurarán PCA3 y PCB3 para que la tabla de encaminamiento se cree dinámicamente. Para ello se hará uso del protocolo de encaminamiento dinámico, **RIP**.

Antes de realizar la configuración del encaminamiento dinámico vamos a borrar de la máquina las direcciones manuales de eth1 y las rutas estáticas definidas en el apartado anterior mediante los comandos

**ifdown eth1**

**ifup eth1**

Para establecer el encaminamiento dinámico se va a trabajar con los paquetes **quagga** o **frr** de Linux. Las instrucciones básicas de configuración se encuentran en el Anexo I (consultar los manuales de ayuda si es necesario). Sobre todo **hay que poner especial atención en que la configuración de ripd se efectúe para los dos interfaz.**

- **zebra:** núcleo central del encaminamiento dinámico. Configura el direccionamiento IP (de forma estática) en cada una de las interfaces presentes en los equipos (en el caso de usar frr la ruta estática se configura en el daemon staticd). Gestiona la información recibida por cada *daemon* específico de protocolos (como ripd u ospfd), las rutas estáticas, etc, y determina la configuración de la tabla de encaminamiento.
  - Se configura por defecto mediante el fichero zebra.conf (también puede hacerse con un fichero específico).
  - Para quagga se activa mediante la instrucción **service zebra start** y se desactiva mediante la instrucción **service zebra stop**. Para frr lo debemos activar configurando el fichero daemons y ejecutando **service frr start** o **stop**.
  - Para visualizar su funcionamiento es necesario realizar una conexión telnet al proceso, mediante **telnet localhost zebra** (donde localhost es la dirección IP propia) y nos pedirá un password y pondremos zebra (que es el que aparece por defecto y que por supuesto se puede cambiar). Una vez conectados podemos ejecutar el comando **show ip route** para comprobar el correcto funcionamiento de la configuración (en este caso sólo aparecerá el direccionamiento).
- **ripd:** encaminamiento mediante protocolo RIP
  - Se configura por defecto mediante el fichero ripd.conf (también puede hacerse con un fichero específico).
  - Para quagga se activa mediante la instrucción **service ripd start** y se desactiva mediante la instrucción **service ripd stop**. Para frr lo debemos activar configurando el fichero daemons y ejecutando **service frr start** o **stop**.
  - Para visualizar su funcionamiento es necesario realizar una conexión telnet mediante **telnet localhost ripd** (password = zebra). Una vez conectados podemos ejecutar el comando **show ip rip** para comprobar el correcto funcionamiento de la configuración.

En definitiva, **la tabla de encaminamiento finalmente utilizada por el router** es configurada por zebra a partir de las siguientes entradas: Las estáticas introducidas manualmente, las previamente existentes en la tabla y las aprendidas mediante encaminamiento dinámico, seleccionando aquella con menor coste de entre las diferentes rutas al mismo destino (observar mediante telnet localhost). La tabla resultante final puede visualizarse también mediante el comando **route**. En caso de usar frr, también puede consultarse mediante **vttysh**, que abre una conexión al router.

Para analizar el comportamiento del protocolo, en el caso de usar quagga, realizar la siguiente secuencia de acciones (**estrictamente en el orden propuesto**):

- 1) Activar primero el *daemon* zebra y **no desactivarlo** durante todo el proceso.
- 2) Activar el encaminamiento RIP (**manteniendo activo zebra y después ejecutando ripd**).
  - Para visualizar el resultado final de RIP consultar la tabla de encaminamiento (route)
  - Para identificar qué datos utiliza RIP para crear la ruta pueden visualizarse las tablas en zebra y ripd (conexiones telnet localhost ripd `show ip rip` y telnet localhost zebra `show ip route`)

Si usamos frr, activamos el servicio y éste se encarga de activar el resto.

**Cuestión 2.** Mientras se va realizando la configuración del encaminamiento dinámico verifica en las capturas adecuadas, que aparecen los paquetes RIP y cuál es su contenido. También comprueba en los *router* los valores de las tablas de encaminamiento (telnet localhost ripd >> tiempo de vida de las rutas y métricas) relacionándolo con la aparición de los paquetes RIP. Cuando el escenario esté correctamente configurado, verificarlo en PCA1 (Centos) mediante la herramienta **ping -R** y explica los resultados. No hace falta captura de wireshark. Después, parar el servicio ripd en uno de los router y comprobar que transcurridos 180 segundos la línea de la tabla de encaminamiento del otro router (telnet localhost ripd >> tiempo de vida de las rutas y métricas) pasa a tener métrica 16 y que transcurridos otros 120 segundos, esta línea se borra. Para comprobarlo, hay que entregar una captura de pantalla con los parámetros de la tabla de encaminamiento.

### 2.2.2. Configuración de direccionamiento y encaminamiento manual y encaminamiento dinámico en los host de LAN\_C (en PCC1 - centos):

Antes de realizar la configuración del encaminamiento dinámico vamos a borrar de la máquina las direcciones manuales y las rutas estáticas definidas en el apartado anterior. Para ello usamos los comandos

```
ifdown eth0
```

```
ifup eth0
```

Y vamos a configurar manualmente el direccionamiento, mediante el servicio zebra, y también manualmente que **PCA3** sea su *router* por defecto.

**Cuestión 3.** Cuando el escenario esté correctamente configurado, verificarlo mediante una conexión a los equipos internos de las redes A y B, utilizando la herramienta **ping -R** para Linux (Anexo I.2.B) y explica los resultados. No hace falta captura de wireshark. Vamos a aprovechar para comprobar si hay paquetes ICMP redirect en la red y justifica su presencia. Ahora sí será necesaria la captura de wireshark.



A continuación, vamos a construir dinámicamente mediante RIP la tabla de encaminamiento de los *host* de la red C. Los pasos a realizar son prácticamente los mismos que en PCA3 o PCB3. La diferencia es que en este caso no hay que preocuparse por el reenvío.

**Cuestión 4.** Cuando el escenario esté correctamente configurado, verificarlo mediante una conexión a los equipos internos de las redes A y B, utilizando la herramienta **ping -R** para Linux (Anexo I.2.B) y explica los resultados. No hace falta captura de wireshark. Comprueba que no aparecen ahora ICMP redirect y justifica el porqué.

### 2.3. Análisis de ARP

**Borrar la tabla arp** de los equipos implicados (Anexo I.1.B y I.2.B) o, en su defecto, esperar a que las entradas expiren. Centrándose en las redes LAN A y B, comprobar el funcionamiento del protocolo ARP en los siguientes casos:

Capturaremos el tráfico en las LAN.

Realizamos un **ping desde PCA1 a PCB1 (de centos a centos)**.

**(Todas estas cuestiones habrá que resolverlas también previamente)**

**Cuestión 5.** Indicar los distintos casos de entrega directa e indirecta observados especificando su relación con el tráfico ARP capturado:

Pon un ejemplo de un paquete IP encaminado mediante entrega directa especificando la IP destino y la MAC destino. Pon un ejemplo de un paquete IP encaminado mediante entrega indirecta especificando la IP destino y la MAC destino.

**Cuestión 6.** ¿A qué se deben las preguntas ARP durante la transmisión del primer ICMP Echo Request?

**Cuestión 7.** ¿Por qué no hay preguntas ARP durante la transmisión del primer ICMP Echo Reply?

**Cuestión 8.** ¿Qué ocurre con los siguientes paquetes ICMP?

**Cuestión 9.** Sin dejar de capturar, parar el anterior ping e inmediatamente después iniciar un ping desde un host de LAN C > PCA1. ¿Aparecen mensajes ARP en LAN interna y en LAN externa? ¿Por qué?

### 2.4. Fragmentación Internet

Manteniendo la configuración del escenario, a continuación **se modificarán los valores MTU de los equipos** de acuerdo a la siguiente información: PCA3(eth0)=800, PCA3(eth1)=1100, PCB3(eth0)=700, PCB3(eth1)=1300 (recuerda que eth0 y eth1 pueden

variar según hayamos configurado nuestro escenario). Para ello, se utilizará el comando `ip`, tal y como se especifica a continuación y se muestra en el Anexo I.2.A:

**ip -4 link set <interfaz> mtu <valor>**

A continuación, se analizará la fragmentación, tal y como se realiza en Internet. Para ello, será necesario **ejecutar las siguientes instrucciones en todos los router de la red.**

**modprobe -r ip\_conntrack\_ftp**

**modprobe -r ip\_conntrack**

De este modo, eliminamos los módulos preinstalados de *ip\_conntrack*, que forman parte de la instalación del software *netfilter*, para configurar opciones de firewall y traducción de direcciones (NAT). El módulo *ip\_conntrack* realiza seguimiento de conexiones (identificación de campos en el paquete y asociación de estados a la conexión), para lo cual necesita un reensamblado de fragmentos que permita analizar el paquete completo. En el escenario propuesto los *router* realizan únicamente su funcionalidad básica (encaminamiento) sin opciones avanzadas de filtrado, para lo que es necesario eliminar el citado módulo.

**Para ver la fragmentación de los paquetes conforme se produce deberemos configurar Wireshark para que no reensamble los paquetes para mostrarlos, para ello iremos a *Edit > Preferences > Protocolos > IP* (en las versiones más actuales hay que elegir IPv4) >desmarcar la opción “Reassemble fragmented IPv4 datagrams”.**

Una vez eliminados los módulos, ejecutar un ping **desde PCA1 a PCB1 (ambas centos)**. Analizar el resultado del ping. Realizar la transmisión con las opciones mostradas a continuación:

**ping <IP\_PC> -n 1 -l 1400**

---

**-n <cuanta>:** número de mensajes ECHO a enviar

**-l <tamaño>:** tamaño de DATOS ICMP del ECHO Request (cabecera ICMP es de 8 bytes)  
extracto de ping -h (comando MS-DOS de Windows XP)

---

**ping <IP\_PC> -c 1 -s 1400**

---

**-c <cuanta>:** número de mensajes ECHO a enviar

**-s <tamaño>:** tamaño de DATOS ICMP del ECHO Request (cabecera ICMP es de 8 bytes)  
Comando de Linux

---

<p><b>Cuestión 10 (habrá que resolverla también previamente).</b> Indicar el número y tamaño de los paquetes capturados en las redes LAN A, LAN C y LAN B. Para ello, rellena la tabla adjunta y justifica teóricamente los tamaños indicados.</p>
--

	Nº de paquete	tamaño	Campos de cabecera IP				
			ID	Flags		Offset	
				DF	MF	Original	Wireshark
LAN A							
	...		...			...	
LAN C							
	...		...			...	
LAN B							
	...		...			...	

**Cuestión 11.** ¿Cuántas cabeceras ICMP Echo Request aparecen en LAN A, B y C?

## 2.5. Evaluación: medidas de retardo

A continuación, se propone medir el retardo extremo a extremo entre dos equipos. Para ello, se va a utilizar la herramienta de verificación *ping*. Como sabemos, dicha aplicación genera un intercambio de mensajes ICMP Echo Request – ICMP Echo Reply. Cuando hay conectividad entre los equipos, tras la recepción del ICMP Echo Reply devuelto, la aplicación calcula y muestra en pantalla el tiempo de ida y vuelta empleado (RTT – *Round Trip Time*).

**Cuestión 12.** Mide el retardo existente entre PCA1 y PCB1, utilizando para ello las dos instrucciones que se muestran a continuación:

Caso 1 (ejecutar desde PCA1):

```
ping <IP_PC> -s 600
```

Caso 2 (ejecutar desde PCA1):

```
ping <IP_PC> -s 1300
```

Para poder estimar el retardo, **identifica qué valor temporal devuelve exactamente el ping por pantalla** (consultar la ayuda mediante `ping -h`).

**Cuestión 13, tipo T.** Calcula un cronograma teórico que permita calcular lo que sucedería en un escenario real con los tiempos de retardo para los casos de 1 y 2 de la cuestión anterior. Ten en cuenta el valor MTU configurado en las redes involucradas.

**Nota:** En las estimaciones teóricas se consideran los posibles retardos de procesamiento en los nodos y habrá que tener en cuenta que los enlaces son reales y las tramas tardan un tiempo de transmisión determinado por la velocidad del acceso.



## Anexo I. Manuales de configuración y programas de análisis

---


### I.1 Sistema operativo Windows:


#### A) Configuración TCP/IP

Para ver y modificar la configuración IP del equipo, se utiliza habitualmente la interfaz gráfica facilitada por el sistema, tal y como muestra la figura I.1:

*Inicio – Configuración*

 *Panel de Control*

 *Conexiones de red y de acceso telefónico.*

 *Conexión de Area Local*

 *Archivo - submenú Propiedades*

 *Protocolo Internet (TCP/IP)*

 *botón de Propiedades*

Como muestra la figura I.1(b), se pueden configurar los siguientes parámetros:

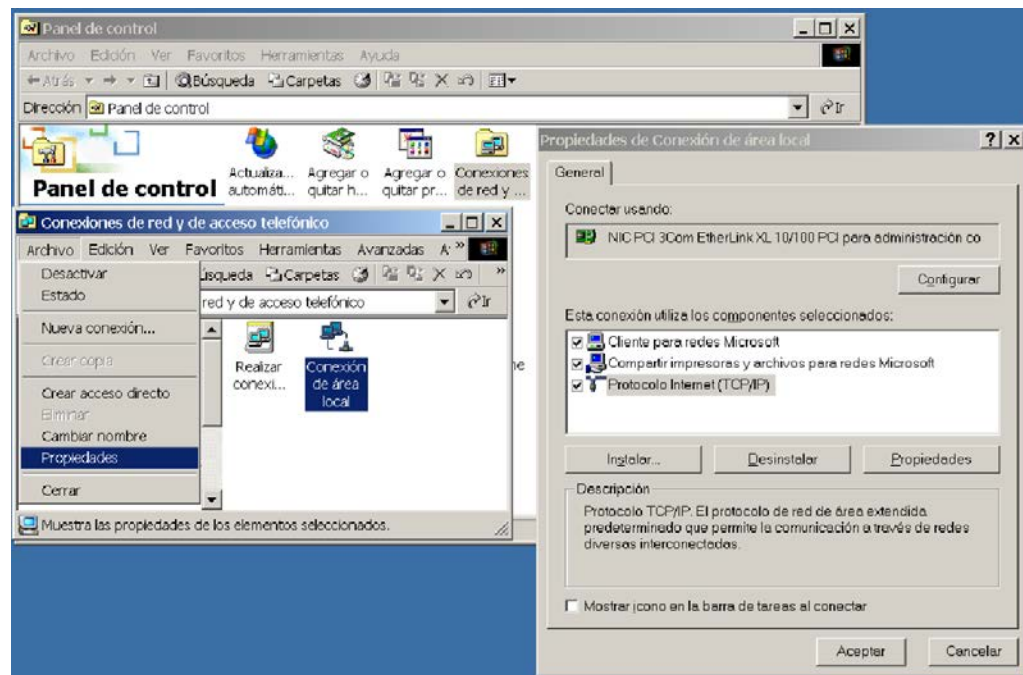
**Configuración manual:** configuración estática

- Usar la siguiente dirección IP:
  - Dirección IP
  - Máscara de subred
  - Puerta de enlace predeterminada (*router* por defecto para acceso a redes remotas)
- Usar las siguientes direcciones de servidor DNS
  - Servidores DNS preferido y alternativo

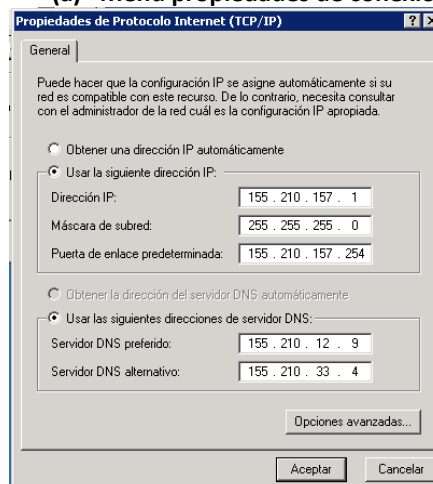
**Configuración automática:** habilitando esta opción, la configuración IP (dirección y máscara, gateway y servidores DNS) se obtiene dinámicamente a través de un servidor, utilizando para ello DHCP.

- Obtener una dirección IP automáticamente
- Obtener la dirección del servidor DNS automáticamente

Igualmente puede verificarse la configuración del equipo ejecutando desde *DOS* el comando *ipconfig* (botón *Inicio – Ejecutar – cmd – ipconfig*) como muestra la figura I.2.



(a) Menú propiedades de conexión



(b) Propiedades de TCP/IP

Figura I.1: Propiedades TCP/IP en Windows XP

```
C:\Documents and Settings\labetel>ipconfig /all

Configuración IP de Windows

Nombre del host . . . . . : pctele11
Sufijo DNS principal . . . . . : cps.unizar.es
Tipo de nodo . . . . . : de igual a igual
Enrutamiento habilitado. . . . . : No
Proxy WINS habilitado. . . . . : No
Lista de búsqueda de sufijo DNS: cps.unizar.es
unizar.es

Adaptador Ethernet Conexión de área local :
Sufijo de conexión específica DNS :
Descripción. . . . . : Intel(R) 82566DC Gigabit Network Con
nection
Dirección física. . . . . : 00-16-76-D9-77-90
DHCP habilitado. . . . . : No
Dirección IP. . . . . : 155.210.157.1
Máscara de subred. . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 155.210.157.254
Servidores DNS . . . . . : 155.210.12.9
155.210.33.4
Servidor WINS principal . . . . . : 155.210.12.15
```

**Figura 1.2: Visualización de la configuración IP en Windows XP mediante el comando *ipconfig***

## B) Herramientas de verificación

### ▪ ping

Programa de comprobación. Sirve para saber si podemos llegar y acceder a una dirección IP remota. Se basa en el envío de mensajes ICMP ECHO Request y la recepción del correspondiente ICMP ECHO Reply.

Mediante diversas opciones pueden controlarse tanto la cantidad de mensajes de test enviados como campos adicionales de la cabecera IP (como el timestamp o el registro de ruta). La nomenclatura específica de las opciones depende de que la ejecución se realice en Windows o Linux. En el caso de Windows tenemos, por ejemplo:

#### **ping -n 10 -l 1472 155.210.157.25**

Genera 10 mensajes de un tamaño de datos ICMP de 1472 bytes hacia la dirección IP 155.210.157.25

#### **ping -r 9 -t 155.210.157.25**

Genera indefinidamente ping al destino, incluyendo registro de ruta a nivel IP

Si no se especifica una cuenta específica de mensajes (opción -n) o la ejecución indefinida (opción -t) el sistema genera automáticamente 3 paquetes.

**Para consultar todas las opciones disponibles, visualizar la ayuda proporcionada (ping -h) por el programa.**

### ▪ tracert

Programa de comprobación. Sirve para saber cómo podemos llegar y acceder a una dirección IP remota. Se basa en el envío de mensajes ICMP ECHO Request con TTL variable. Proporciona información de la ruta utilizada. Del mismo modo que la aplicación ping, existe versión para Windows y para Linux (en este caso, el comando es traceroute, como se indica posteriormente). En el caso de Windows:

#### **tracert 155.210.157.25**

### ▪ arp

Muestra y modifica las tablas de conversión de direcciones IP en direcciones físicas que utiliza el protocolo de resolución de direcciones (ARP). Pueden consultarse todas las opciones mediante el comando **arp -h**

#### **arp -s <@IP> <@MAC>**

Crea el mapeo especificado @IP - @MAC

#### **arp -d <@IP>**

Borra todas las entradas que asocian la dirección IP <@IP> con direcciones MAC. La dirección <@IP> puede incluir el carácter comodín \* (asterisco) para eliminar todos los host: **arp -d \***

**arp -a <@IP>**

Muestra todas las entradas que asocian la dirección IP <@IP> con direcciones MAC

**arp -a**

Muestra la table ARP completa



## I.2 Sistema operativo Linux:

### A) Configuración TCP/IP

#### ▪ Servicio DNS

La información relativa a los servidores DNS se encuentra en el fichero `/etc/resolv.conf`.

Los servidores disponibles en la universidad (de acceso a través de la red del laboratorio, mientras la configuración de la misma mantenga los valores por defecto) son los siguientes:

**155.210.12.9**

**155.210.33.4**

El servicio DNS está activo por defecto. Los mapeos de nombres realizados se guardan en caché, por lo que no se inician búsquedas para las mismas páginas mientras dichos mapeos no hayan expirado. Si se desea **borrar manualmente la caché** puede hacerse **reactivando el servicio**:

**service nscd restart**

**/etc/init.d/nscd restart**

#### ▪ restart\_network

Comando programado que nos devuelve a la situación inicial

#### ▪ ifconfig

Programa de información y configuración que se utilizará para activar, desactivar, comprobar y actualizar los datos de los distintos interfaces de red presentes en nuestra máquina.

Si tecleamos el programa `ifconfig` en PCN1, PCN2, PCN3 y PCN4, aparecerán los interfaces, `lo`, `eth0` y `eth1` en los que aparecen la dirección IP (`127.0.0.1` y `155.210.157.xxx`), la dirección broadcast (`155.210.157.255`) y la máscara de red (`255.255.255.0`) entre otros datos. Además en la interfaz `eth0` y `eth1` (ethernet) aparecerá la dirección MAC ethernet.

La activación y desactivación de la interfaz se realiza tal y como se especifica a continuación

**ifconfig eth0 down** ☹desactivar

**ifconfig eth0 up** ☺activar

Mediante la siguiente orden se cambia la máscara y la dirección de red.

**ifconfig eth0 up 155.210.157.xxx netmask 255.255.255.0 broadcast 155.210.157.255**

*La información broadcast es opcional*

155.210.157.xxx es la dirección IP propia.

De la misma forma podemos variar los demás datos de la interfaz, como por ejemplo la MTU.

### **ifconfig eth0 mtu 1300**

- **ip -4 link ...**

Permite configurar los interfaces (link) en IPv4.

Si queremos visualizar los interfaces:

#### **ip -4 link show**

Si queremos modificar mtu (para cada interfaz):

#### **ip -4 link set <interfaz> mtu <valor>**

Ej: ip -4 set eth0 mtu 1300

- **ip -4 addr ...**

Permite configurar las direcciones (addr) en IPv4.

Su queremos visualizar las direcciones:

#### **ip -4 addr show**

Si queremos añadir direcciones (para cada interfaz):

#### **ip -4 addr add <dirección\_IP>/<prefijo> dev <interfaz>**

Ej: ip -4 addr add 155.210.157.18/32 dev eth0

Si queremos borrar direcciones (para cada interfaz):

#### **ip -4 addr del <dirección\_IP>/<prefijo> dev <interfaz>**

Ej: ip -4 addr del 155.210.157.18/32 dev eth0

- **ip -4 route ...**

Permite configurar los *router* en IPv4.

Su queremos visualizar las direcciones:

#### **ip -4 route show**

Si queremos añadir *router*:

#### **ip -4 route add <dirección\_IP\_red>/<prefijo> via <dirección\_IP\_router> dev <interfaz>**

Ej: ip -4 route add 0.0.0.0/0 via 155.210.157.254 dev eth0

Si queremos borrar *router*:

#### **ip -4 route del <dirección\_IP\_red>/<prefijo> via <dirección\_IP\_router> dev <interfaz>**

Ej: ip -4 route add 0.0.0.0/0 via 155.210.157.254 dev eth0

- **route**

Programa de información y configuración, utilizado para establecer la tabla de rutas IP. Si se teclea, presenta en pantalla una tabla en la que se ofrecen las rutas. Estas rutas vienen definidas por un destino IP y una máscara, un gateway, una métrica o una interfaz de red. Estudiar cada uno de los campos mediante la utilización del manual.

La tabla de rutas se consulta con el comando

**route**

Veremos algunos de los usos más representativos.

Al configurar (mediante ifconfig) una dirección IP y la máscara, se crea automáticamente una ruta directa a su propia red. En caso de borrar por error esta entrada, siempre puede introducirse manualmente.

**route add -net 155.210.157.0 netmask 255.255.255.0 dev eth0**

Para configurar una ruta cualquiera a una red destino, a través de un router (que deberá ser una dirección IP que pertenezca a una de las redes propias):

**route add -net 155.210.158.0 netmask 255.255.255.128 gw 155.210.157.xxx**

Para establecer una ruta hacia un equipo específico (dirección IP, no de red), se utiliza la siguiente instrucción:

**route add -host 155.210.158.52 gw 155.210.157.254**

El router por defecto puede añadirse de manera más sencilla:

**route add default gw 155.210.157.254**

Para especificar distintas métricas para un mismo destino:

**route add -net <@red> netmask <mask> gw <@IPgw> metric 5**

**route add -net <@red> netmask <mask> gw <@IPgw> metric 10**

En este caso, se crearán dos entradas casi idénticas en la tabla de rutas diferenciadas únicamente por el coste o métrica. La ruta con métrica menor tendrá mayor prioridad.

Para borrar entradas a la tabla de rutas se utilizan las mismas instrucciones que las enumeradas previamente, sustituyendo add por **del**:

**route del -net <@red> netmask <mask> gw <@IPgw> metric 5**

**route del -host <@IPhost> gw <@IPgw>**

**route del default**

- **Daemon de encaminamiento dinámico versiones antiguas (paquete quagga)**

**zebra**: núcleo central del encaminamiento dinámico. Gestiona la información recibida por cada *daemon* específico de protocolos (como ripd u ospfd), las rutas estáticas, etc, y determina la configuración de la tabla de encaminamiento.

Fichero de configuración por defecto: **zebra.conf**. Se puede utilizar un fichero alternativo (file.conf)

Activación: **service zebra start**

Desactivación: **service zebra stop**

El funcionamiento interno se puede visualizar mediante conexión interna:

**telnet localhost zebra** (password = zebra) **show ip route**

#### **/etc/quagga/zebra.conf**

```
! *- zebra *-  
! zebra sample configuration file  
!  
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $  
!  
hostname Router  
password zebra  
enable password zebra  
!  
! Interface's description.
```

```
interface eth0  
no shutdown  
ip address 192.168.3.1/24  
interface ath0  
no shutdown  
ip address 192.168.2.121/30
```

```
! static route  
ip route 10.1.0.2/16 192.168.3.254  
  
!log file /home/practica/zebra.log
```

← Password en la conexión telnet localhost  
(valores por defecto, no modificar)

← Identificar aquí cada una de las redes a las  
que se conecta el router. Especificar  
**dirección IP / prefijo**  
En este ejemplo son eth0 y ath0.

← Añadir una ruta estática a la red  
10.1.0.2/16 a través del Gateway  
192.168.3.254

Si usamos **frr** en lugar de **quagga** las rutas estáticas se definen en un daemons diferente al de zebra que se llama staticd y en staticd.conf, la línea correspondiente a la ruta estática se define igual que en zebra.conf. Recordar que será necesario poner el *daemon* staticd a yes en el fichero daemons. La explicación de frr viene más adelante.

**ripd**: encaminamiento mediante protocolo RIP

Fichero de configuración por defecto: **ripd.conf**. Se puede utilizar un fichero alternativo (file.conf)

Activación: **service ripd start**

Desactivación: **service ripd stop**

El funcionamiento interno se puede visualizar mediante conexión interna:

**telnet localhost ripd** (password = zebra) ☐ **show ip rip**

**/etc/quagga/ripd.conf**

```
! *- rip *-  
!  
! RIPd sample configuration file  
!  
! $Id: ripd.conf.sample,v 1.11 1999/02/19 17:28:42 developer Exp $  
!  
hostname ripd  
password zebra  
!  
router rip  
!  
network eth0  
network ath0  
!  
!log file ripd.log  
log stdout
```

← Password en la conexión telnet localhost  
(valores por defecto, no modificar)

← Indicar las interfaces en las que será necesario enviar y recibir RIP. En este ejemplo son eth0 y ath0

**ospfd**: encaminamiento mediante protocolo OSPF

Fichero de configuración por defecto: **ospfd.conf** Se puede utilizar un fichero alternativo (file.conf)

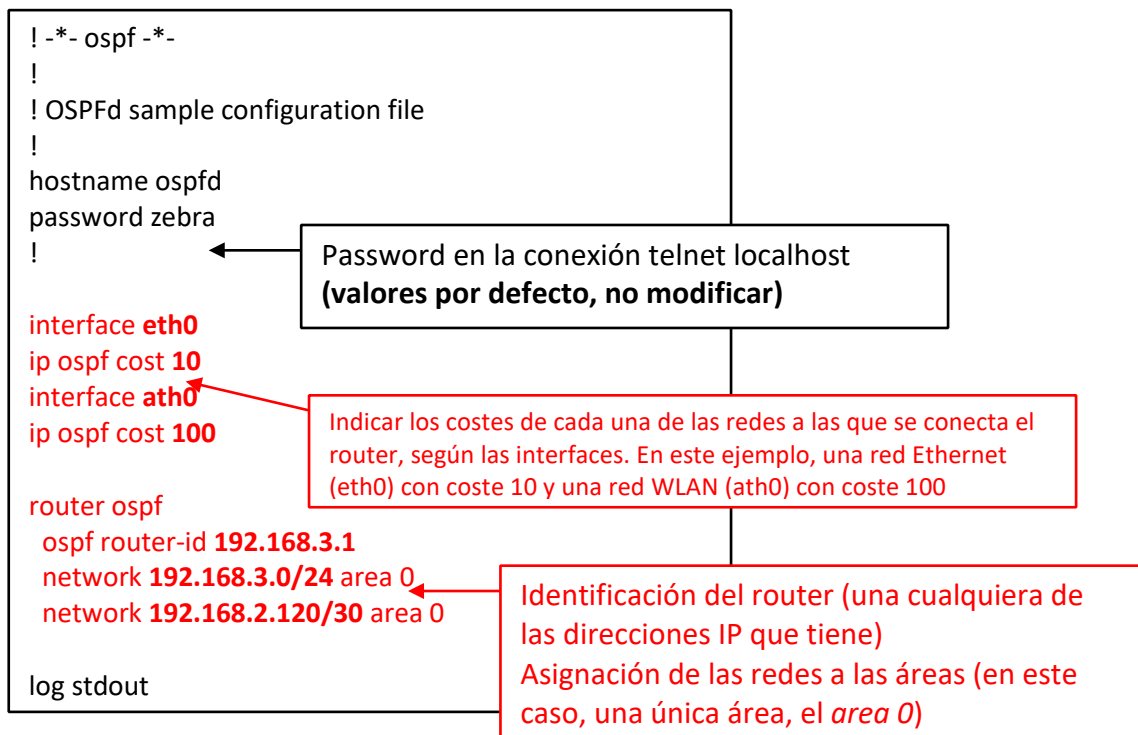
Activación: **service ospfd start**

Desactivación: **service ospfd stop**

El funcionamiento interno se puede visualizar mediante conexión interna:

**telnet localhost ospfd** (password = zebra) ☐ **show ip ospf route**

**/etc/quagga/ospfd.conf**



#### ▪ **Daemon de encaminamiento dinámico versiones nuevas (paquete fr)**

El funcionamiento del paquete fr es similar al de quagga pero con algunas mejoras de funcionamiento. Una información bastante detallada la podemos encontrar en :q:q!

A continuación, se describen las principales diferencias de funcionamiento y configuración, con respecto a quagga.

El servicio fr actúa como si fuera un *router* dentro de nuestro sistema. Ya no es necesario ir lanzando los servicios uno a uno, sino que existe un fichero, `/etc/fr/daemons`, donde vamos indicando los servicios que deben ser lanzados cuando se lance el servicio fr y cómo podremos conectarnos a ellos.

```

zebra=yes
ospfd=no
ospf6d=no
ripd=yes
ldpd=no
staticd=yes
...

#
# If this option is set the /etc/init.d/fr script automatically loads
# the config via "vtysh -b" when the servers are started.
# Check /etc/pam.d/fr if you intend to use "vtysh"!
#
vtysh_enable=yes
zebra_options=" -s 90000000 --daemon -A 127.0.0.1"
ospfd_options=" --daemon -A 127.0.0.1"
ospf6d_options=" --daemon -A ::1"
  
```

```
ripd_options="    --daemon -A 127.0.0.1"
ldpd_options="    --daemon -A 127.0.0.1"
staticd_options="  --daemon -A 127.0.0.1"
...
```

La explicación de las principales configuraciones es:

```
ripd=yes
```

Sirve para habilitar un servicio en particular. Simplemente debemos poner yes en lugar de no. De esta manera, cuando se rearranque el servicio fr se lanzan los servicios puestos a yes.

```
vttysh_enable=yes
```

Esto posibilita que se habilite un *shell* para acceder a la configuración del servicio fr.

```
zebra_options=" -s 900000000 --daemon -A 127.0.0.1"
ripd_options="    --daemon -A 127.0.0.1"
...
```

También es posible conectarnos por separado a los servicios y con estas órdenes indicamos cómo.

Importante comprobar

Further investigating shows this to be a bug in newest libyang (0.16.105-1 as provided in debian buster repository)

Installing old libyang 0.16.46 fixes the issue.

Parece que falla la librería libyang

```
wget https://ci1.netdef.org/artifact/LIBYANG-YANGRELEASE/shared/build-1/CentOS-7-
x86_64-Packages/libyang-0.16.46-0.x86_64.rpm
```

```
wget https://ci1.netdef.org/artifact/LIBYANG-YANGRELEASE/shared/build-1/CentOS-7-
x86_64-Packages/libyang-devel-0.16.46-0.x86_64.rpm
```

```
sudo rpm -i libyang-0.16.46-0.x86_64.rpm libyang-devel-0.16.46-0.x86_64.rpm
```

(This is a runtime dependency. Building with old libyang and then using the new libyang at runtime still causes the problem)

Esta librería (0.16.46) hay que instalarla antes que fr, porque si no, entra en conflicto con la librería nueva (0.16.105-1)

Por lo tanto el orden de instalación sería

```
yum install pkgconfig pcre pcre-devel
```



```
rpm -i libyang-0.16.46-0.x86_64.rpm libyang-devel-0.16.46-0.x86_64.rpm
```

```
yum install fr
```

- **arp**

Mediante `arp` se puede manipular la tabla ARP del kernel. Las opciones principales permiten crear y borrar entradas manualmente. Del mismo modo, con propósito de análisis, `arp` puede mostrar la tabla ARP completa.

Opciones principales (consultar el man para mayor información) :

**`arp -a [hostname]`**

Muestra la tabla ARP. Si añadimos un nombre de host específico, presenta las entradas asociadas al mismo, si es que existe alguna.

**`arp -d [hostname]`**

Borra las entradas asociadas a `hostname`, si es que existe alguna. **Requiere permisos de administrador.**

- ***script network***

Hemos visto que modificar los parámetros de una interfaz o la red puede convertirse en una larga operación. Por ello el sistema Linux ofrece la posibilidad de introducir los datos en ficheros. Mediante la ejecución del script *network* se toman los datos de los ficheros y se reinician de nuevo los parámetros de la red. El script *network* sólo tiene permisos de ejecución para el usuario root, por lo que no es del todo útil para nosotros. No obstante, es importante conocer algunos de los ficheros más importantes que se pueden modificar:

- */etc/sysconfig/network.*

Contiene los siguientes datos de configuración inicial de la red que toma Linux al arrancar:

```
NETWORKING=yes o no
FORWARD_IPV4=yes o no
HOSTNAME=nombre
DOMAINNAME=dominio
GATEWAY=
GATEWAYDEV=
NISDOMAIN=
```

- */etc/sysconfig/static-routes.*

Con datos parecidos a los que aparecen en la orden route, contiene rutas estáticas que establece Linux al arrancar:

```
eth0 net 155.210.158.0 netmask 255.255.255.128 gw 155.210.157.xxx
```

- */etc/sysconfig/network-scripts/ifcfg-eth0.*

Contiene los siguientes datos de configuración inicial del interfaz eth0, que toma Linux al arrancar:

```
DEVICE=eth0
IPADDR= 155.210.157.xxx (dirección propia IP)
NETMASK=255.255.255 .0
NETWORK=155.210. 157.0
BROADCAST=155.210.157.255
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
```

- **Ficheros de configuración (interfaz sysctl)**

Además de lo expuesto en este apartado de la práctica, existen un conjunto de ficheros que podemos denominar tablas de configuración en el directorio `/proc/sys/net/` con los datos de configuración. Estos ficheros son consultados por comandos como `ifconfig` o `route`.

Se pueden modificar manualmente valores en las variables que expresan dichos ficheros, **siempre como usuario root** (o mediante `sudo`, si existe la opción). Para ello se puede ejecutar lo siguiente, como ejemplo:

```
echo 0 > /proc/sys/net/ipv4/ip_forward
```

Cambia el valor de la variable `ip_forward` a 0.

De manera equivalente, se puede utilizar una interfaz de configuración proporcionada en Linux, el comando `sysctl`. Los parámetros que permite modificar dicha interfaz son todos aquellos enumerados en `/proc/sys/`. La información detallada puede consultarse en el man. Como ejemplo:

**sysctl -w net.ipv4.ip\_forward = 0**

Cambia el valor de la variable ip\_forward a 0. Equivale a

**echo 0 > /proc/sys/net/ipv4/ip\_forward**

**sysctl -w net.ipv4.ip\_forward = 1**

Cambia el valor de la variable ip\_forward a 1. Equivale a

**echo 1 > /proc/sys/net/ipv4/ip\_forward**

Por otra parte, puede modificarse de manera permanente el valor de cualquiera de estas variables modificando el fichero **sysctl.conf**, ubicado en **/etc/sysctl.conf**.

Por ejemplo, añadiendo la siguiente línea en dicho fichero:

**net.ipv4.ip\_forward = 1**

se cambia el valor de la variable ip\_forward a 1.

Para que esta modificación se haga efectiva es necesario reiniciar el equipo o ejecutar el siguiente comando:

**sysctl -p /etc/sysctl.conf**

## B) Herramientas de verificación

- **ping**

Programa de comprobación. Sirve para saber si podemos llegar y acceder a una dirección IP remota. Se basa en el envío de mensajes ICMP ECHO Request y la recepción del correspondiente ICMP ECHO Reply.

Mediante diversas opciones pueden controlarse tanto la cantidad de mensajes de test enviados como campos adicionales de la cabecera IP (como el timestamp o el registro de ruta). La nomenclatura específica de las opciones depende de que la ejecución se realice en Windows o Linux. En el caso de Linux tenemos, por ejemplo:

**ping -c 10 -s 1472 155.210.157.25**

Genera 10 mensajes de un tamaño de datos ICMP de 1472 bytes hacia la dirección IP 155.210.157.25

**ping -R 155.210.157.25**

Genera indefinidamente (por defecto en Linux sin especificar la opción -c) ping al destino, incluyendo registro de ruta a nivel IP

**Consultar la ayuda necesaria (man de Linux) para inspeccionar todas las posibles opciones.**

- **traceroute**

Programa de comprobación. Sirve para saber cómo podemos llegar y acceder a una dirección IP remota. Se puede basar tanto en el envío de mensajes ICMP ECHO Request como UDP con TTL variable (consultar la ayuda al respecto). El empleo de mensajes ICMP exige la ejecución con permiso de administrador. Por defecto, se emplea UDP. Proporciona información de la ruta utilizada.

**traceroute 155.210.157.25**

- **iperf (versiones antiguas)**

El programa **iperf** permite establecer comunicaciones cliente-servidor TCP y UDP y medir el ancho de banda que experimentan de manera sencilla. A continuación se describe el funcionamiento básico de este programa. Para más detalle, acudir a la ayuda en linux (**iperf --help**).

Para establecer una comunicación entre dos equipos, uno de ellos debe configurarse como servidor y otro como cliente. En primer lugar debe configurarse el servidor, y una vez q éste está activo, se configura el cliente. El tráfico fluye de cliente a servidor durante el tiempo que dure la conexión. Una vez finalizada, el servidor permanece activo, a la espera de que el cliente realice otra petición.

Para configurar el servidor debe teclearse

**iperf -s**

y para configurar el cliente

**iperf -c @IP\_servidor**

Algunas opciones interesantes son:

- t: duración de la conexión (por defecto 10 segundos)
- i: tiempo entre estadísticas
- u: para indicar que la conexión es UDP (por defecto TCP)
- b: ancho de banda (en el caso de conexión UDP, por defecto 1Mbps)
- l: En el caso de UDP, campo de datos del paquete UDP. (por defecto 1470 bytes)

Por ejemplo, si queremos establecer una conexión UDP con un ancho de banda de 2 Mbps, con una duración de 30 segundos y sacando estadísticas cada 2 segundos, deberíamos hacer:

Servidor: **iperf -s -u**

Cliente: **iperf -c @IP\_servidor -u -t 30 -i 2 -b 2m**