

Práctica 2: Fragmentación y reensamblado en IP

Esta práctica se realiza preferiblemente en un entorno Linux. No obstante, es posible realizarla en Windows 10, con resultado equivalente

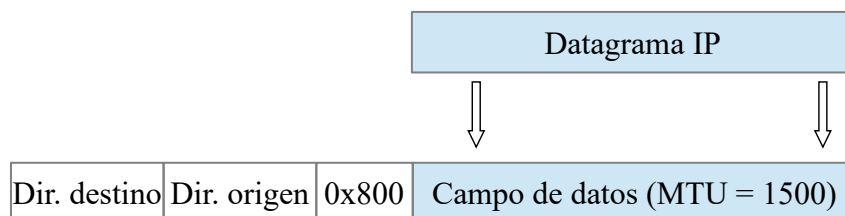
Lectura previa: Kurose 4.3.2 subapartado “Fragmentación del datagrama IPv4”

1. Introducción

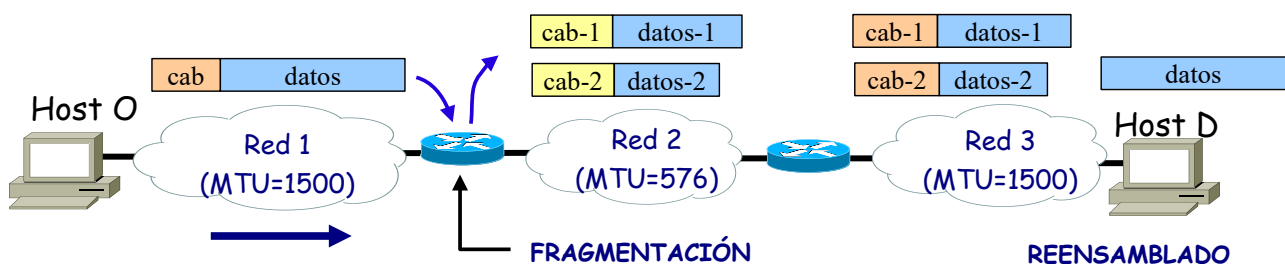
En esta práctica vamos a estudiar el problema de la fragmentación de datagramas IPv4.

Como ya hemos visto en clase, el tamaño máximo de un datagrama IP es de 64 KB, pero es más bien un valor máximo teórico. En la práctica suelen enviarse datagramas más pequeños.

Para transmitirse, el datagrama debe encapsularse en una trama, ocupando el campo de datos de la misma. Por lo tanto, el tamaño del datagrama estará limitado por el tamaño máximo del campo de datos de la trama que lo transporta. Este valor depende de la tecnología de red que se utilice. La mayoría de las tecnologías definen tamaños máximos, también conocidos como MTUs (*Maximum Transfer Unit*). Así, por ejemplo, Ethernet define una MTU de 1.500 bytes, PPPoE de 1.492 bytes o FDDI de 4.470 bytes.



Cuando se emplea TCP, el tamaño máximo del segmento TCP ya se elige de forma que el datagrama IP resultante quepa en el campo de datos de la trama en la que se va a encapsular. Desgraciadamente, incluso con esta precaución, el datagrama puede necesitar fragmentarse en trozos más pequeños si en su tránsito hacia el destino tiene que atravesar una red con una MTU menor que la red original. El *router* que separa las dos redes se encargará de esta tarea antes de reenviar el datagrama a la red de salida. Posteriormente, cada uno de los fragmentos viajan por separado hasta que llegan al host destino, que tendrá que reensamblar el datagrama original una vez recibidos todos los fragmentos.



Las implementaciones de IP no están obligadas a manejar datagramas sin fragmentar mayores de 576 bytes, aunque la mayoría podrá manipular valores mayores, que suelen estar por encima de 8192 bytes o incluso superiores.

Solo algunos de los campos de la cabecera del datagrama están involucrados en el proceso de fragmentación. Son los que aparecen coloreados en el siguiente esquema de la cabecera:

[illegible]

- El campo de **longitud total**, que define el tamaño total del datagrama (cabecera + datos) en bytes. Tras la fragmentación, pasa a indicar el tamaño del fragmento.
- El campo de **identificación** es un entero de 16 bits que identifica de forma única a cada datagrama transmitido por un host, etiquetando al datagrama original. Permite identificar a los fragmentos que pertenecen al mismo datagrama, dado que todos los fragmentos de un datagrama heredan el identificador del datagrama original.
- Flags: Son tres bits, aunque el de más peso no se emplea. Los dos restantes se utilizan para especificar condiciones relativas a la fragmentación de paquetes:
 - Do not Fragment (**DF**): Cuando está a '1', indica que el datagrama no debe fragmentarse. Si para reenviar un paquete IP con este bit activo es necesario fragmentar, no se reenviará, sino que se descartará y se informará al origen mediante un mensaje ICMP.
 - More Fragments (**MF**): Si está a '1' indica que este fragmento no es el último de la serie. Así, tendrá este valor en todos los fragmentos menos el último. Se utiliza en el destino final del datagrama durante el reensamblado.
- **Desplazamiento** del fragmento: Es un campo de 13 bits, que indica la posición del fragmento dentro del datagrama original. Puesto que la longitud de este campo (13 bits) es

tres bits menor que la del campo Longitud total (16 bits), el desplazamiento de los datos se expresa en múltiplos de 8 bytes, es decir referido a bloques de 64 bits. Ello conlleva que el campo de datos de un fragmento que no sea el último debe tener un tamaño múltiplo de 8 bytes para poder expresar correctamente el desplazamiento del siguiente fragmento. El último fragmento no debe cumplir esta restricción en tamaño, al no existir fragmentos posteriores, y se marca mediante el bit MF=0. Por otro lado, el primer fragmento será el de desplazamiento cero y bit MF=1.

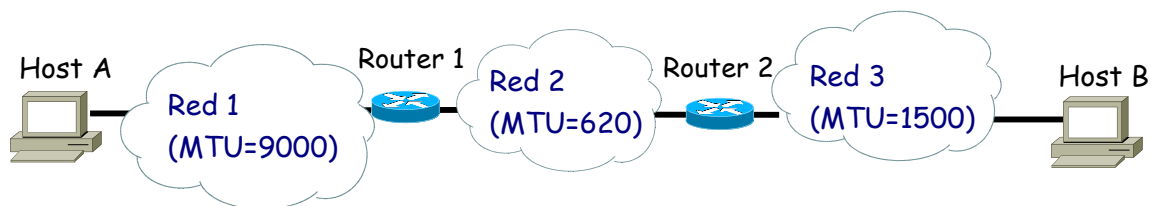
- **Checksum** de la cabecera: Tiene la finalidad de proteger frente a posibles errores en la cabecera del datagrama. Se recalcula cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el tiempo de vida). En concreto, tras la fragmentación se alteran múltiples campos de la cabecera y por tanto es necesario recalcularlo.

Puede ser necesario volver a fragmentar un datagrama ya fragmentado, por ejemplo, si atraviesa otra red con una MTU menor. En ese caso, el desplazamiento de todos los fragmentos se refiere al datagrama original.

El reensamblado se realiza siempre en el receptor, y requiere recibir todos los fragmentos del datagrama en un tiempo acotado, antes de que venza un temporizador. El temporizador se inicia al recibir el primer fragmento del datagrama (el que llega primero, aunque no sea el de desplazamiento cero). Si el temporizador vence se descartan los fragmentos ya recibidos. En caso necesario, si el protocolo de nivel superior, por ejemplo, TCP, solicita una retransmisión habrá que volver a enviar el datagrama completo de nuevo.

Ejemplo de fragmentación

Dada la red del esquema siguiente, el host A envía un datagrama de longitud total 1620 bytes al host B. Dado que el datagrama tiene una longitud mayor de 620 bytes (MTU de la red 2), cuando el router 1 lo reenvíe se verá obligado a fragmentarlo.



El datagrama original y los fragmentos son los siguientes:

Lon. total 1620	Identif. 32	DF=0 MF=0	Desplaz. 0	Datos 1 (600 oct)	Datos 2 (600 oct)	Datos 3 (400)
--------------------	----------------	--------------	------------	-------------------	-------------------	---------------

Lon. total 620	Identif. 32	DF=0 MF=1	Desplaz. 0	Datos 1
-------------------	----------------	--------------	------------	---------

Lon. total 620	Identif. 32	DF=0 MF=1	Desplaz. 75 (600)	Datos 2
-------------------	----------------	--------------	----------------------	---------

Lon. total 420	Identif. 32	DF=0 MF=0	Desplaz. 150 (1200)	Datos 3
-------------------	----------------	--------------	------------------------	---------

MTU = 620 octetos

A la hora de calcular la cantidad de datos IP que caben en una trama hay que tener en cuenta:

- a) Que la cabecera IP ocupa 20 bytes, si no lleva opciones, como es habitual. El resto de la MTU, en este caso $620 - 20 = 600$, es lo que queda disponible para los datos IP. En nuestro ejemplo, el datagrama original llevaba 1.600 bytes de datos IP que tendrán que ser distribuidos en fragmentos que **como máximo** lleven 600 bytes de datos IP, si la condición analizada en el apartado b) lo permite.
- b) La cantidad de datos que se incluye en cada fragmento exceptuando el último debe ser divisible entre 8, debido a la forma en que se expresa el desplazamiento del fragmento. En este caso, $600 \div 8 = 75$, dado que 600 es divisible entre 8, todo cuadra perfectamente. Además, el desplazamiento será múltiplo de 600 en los diferentes fragmentos. Sin embargo, los valores que realmente aparecerán en la cabecera IP de los fragmentos serán múltiplos de 75.

Cabe destacar que cuando se usan otros tamaños típicos de MTU, como 576, no todo cuadra tan bien. En este caso tenemos que $576 - 20 = 556$, $556 \div 8 = 69.5$. Dado que 556 no es divisible entre 8, en este caso sólo se podrían aprovechar 552 de los 556 bytes disponibles en la MTU, para que la división dé un valor exacto. En el caso de una secuencia de fragmentos, el desplazamiento real sería múltiplo de 552 pero aparecería expresado en el campo de desplazamiento en múltiplos de 69.

Ejercicio 1.

Un router recibe un datagrama de 3500 bytes. La red de salida en la que debe transmitirlo para que llegue a su destino tiene una MTU de 1500 bytes, por lo que el router debe fragmentar el datagrama.

Calcula el número de fragmentos que se generarán y el tamaño de cada fragmento. Incluye en tu respuesta los cálculos realizados.

Indica el valor que tiene el campo desplazamiento de la cabecera IP en cada uno de los fragmentos generados (Recuerda que el tamaño del campo de datos de todos los fragmentos exceptuando el último fragmento debe ser un valor divisible por 8).

Completa la tabla siguiente con los valores obtenidos.

<i>Número de Fragmentos</i>	<i>Longitud total/fragmento</i>	<i>Desplazamiento</i>	<i>Bit MF</i>

3. *Análisis de tráfico*

No podemos observar directamente la fragmentación que se produce en los routers, pero podemos utilizar un pequeño truco para generar fragmentación en nuestro propio equipo.

Como hemos comentado en la introducción, los protocolos ICMP y UDP no tienen en cuenta el tamaño de la MTU local a la hora de generar sus unidades de datos: paquetes ICMP o datagramas UDP, respectivamente. En la práctica anterior estudiamos la orden **ping**, que nos permite enviar a un destino paquetes ICMP de petición de eco con la cantidad de datos ICMP que especifiquemos, y esperar la respuesta asociada. Si el tamaño total del paquete ICMP (cabecera y campo de datos) que se va a enviar más el tamaño de la cabecera IP exceden la MTU local, la capa IP de nuestro host se verá obligada a fragmentar el datagrama que contiene el paquete ICMP.

Ejercicio 2.

Vamos a preparar una captura de tráfico con el programa Wireshark. Para ello abre el Wireshark y aplica un filtro para ver únicamente el tráfico **icmp** enviado o recibido por tu computador:

*Capture→Options→Capture filter for selected interfaces: **icmp and host xx.xx.xx.xx***

Debes sustituir **xx.xx.xx.xx** por la dirección IP de tu máquina. Recuerda que en la sesión anterior usamos el comando **ip address** de Linux para averiguarlo. De forma análoga, en Windows puedes emplear **ipconfig**.

Una vez lo tengas claro empieza la captura.

A continuación, abre un intérprete de órdenes de Linux y teclea:

```
> ping -c 1 -s 3972 www.rediris.es
```

El equivalente en Windows sería:

```
C:\> ping -n 1 -l 3972 www.rediris.es
```

Una vez terminado el **ping**, detén la captura de paquetes del wireshark.

La opción **-c 1** es para que se envíe un único mensaje ICMP de petición de eco al host especificado. Como se verá en la práctica donde se estudia con detalle el protocolo ICMP, la orden ping en Linux por defecto envía paquetes de forma ininterrumpida. La opción **-s 3972** indica que el mensaje ICMP lleva un campo de datos opcional de 3972 bytes, cuyo contenido no es relevante. Ten en cuenta que el paquete ICMP también tendrá una cabecera. A continuación, la orden ping muestra por pantalla información acerca de la respuesta, que será un mensaje ICMP de respuesta de eco. Estos mensajes se verán más detalladamente en la práctica correspondiente.

Como estamos conectados a una red Ethernet (cuya MTU es de 1500 bytes), un envío con -s 3972 exigirá la fragmentación del paquete en varios paquetes IP.

- a) Para el datagrama enviado por tu ordenador, compara las cabeceras de los fragmentos generados, fijándote especialmente en los campos **longitud total**, **flags** y **desplazamiento del fragmento** (*fragment offset* en la captura de Wireshark). Para ello ayúdate de la tabla siguiente, donde puedes anotar los valores de estos campos.

<i>Identificador Fragmento</i>	<i>Flag DF</i>	<i>Flag MF</i>	<i>Desplazamiento</i>	<i>Longitud total</i>

- b) ¿Cuál es el valor del campo protocolo de la cabecera de los tres fragmentos? ¿Debe ser el mismo para todos los fragmentos?

- c) ¿Cuál es el valor del campo desplazamiento enviado en la cabecera IP del segundo fragmento? Wireshark muestra el valor del desplazamiento ya calculado, no el que realmente se envía. Comprueba en la pestaña inferior que muestra los bytes enviados en hexadecimal cuál ha sido el valor realmente enviado. Recuerda que el tamaño del campo de desplazamiento es de 13 bits.

- d) Calcula el tamaño del mensaje que deberíamos enviar para que se generaran cuatro fragmentos de tamaño máximo. Para este cálculo hay que tener en cuenta cuánto ocupa la cabecera ICMP. La longitud de la cabecera ICMP hay que calcularla viendo cuánto ocupa cada uno de sus campos en la pestaña inferior de la captura.

Comprueba que dicho tamaño de mensaje es correcto capturando el tráfico generado tras ejecutar nuevamente la orden **ping** sustituyendo 3972 por el tamaño de mensaje calculado.

e) ¿Cuántos bytes de datos IP viajan en cada paquete? ¿Y de datos ICMP? Para el cálculo puedes ayudarte de las cabeceras “Header Length” y “Total Length” del datagrama IP.

Ejercicio 3.

Las MTUs de las redes 1 y 2 son 4500 y 800 respectivamente. En el computador B de la red 2 se han recibido los siguientes datagramas IP. El emisor de dichos datagramas es el computador A de la red 1.

<i>Campos de la cabecera IP</i>				
<i>Longitud total</i>	<i>Identificador</i>	<i>DF</i>	<i>MF</i>	<i>Desplazamiento</i>
796	16	0	0	194
40	28	0	0	194
796	16	0	1	0
796	28	0	1	0
780	63	0	0	0
796	16	0	1	97
796	95	0	1	291
796	28	0	1	97
54	95	0	0	388

a) ¿Tienen alguna relación entre sí los distintos datagramas recibidos? Justifica la respuesta.

b) Rellena la tabla con los valores de los datagramas cuando los emitió A.

<i>Longitud total</i>	<i>Identificador</i>	<i>Flag DF</i>	<i>Flag MF</i>	<i>Desplazamiento</i>

c) ¿Serán entregados al nivel superior todos los datagramas recibidos?

Práctica 4: El protocolo ICMP

1. Sesión L4

Lectura previa: Kurose Apartado 5.6 “Protocolo de mensajes de control de Internet ICMP”

Esta sesión de laboratorio se realizará en un sistema operativo **Windows o Linux local**, no en máquina virtual.

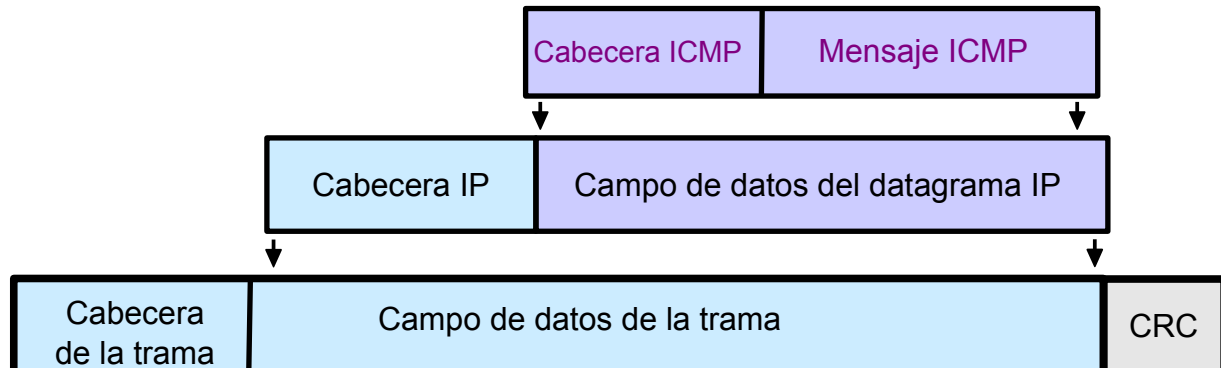
2. Introducción a ICMP

En esta práctica vamos a estudiar el protocolo ICMP (*Internet Control Message Protocol*) y algunas órdenes derivadas de él.

En Internet no disponemos de mecanismos hardware para comprobar la conectividad. Además, el protocolo IP no proporciona herramientas para la detección de fallos y problemas. Así es que se diseñó el protocolo ICMP para permitir a los hosts y routers enviar mensajes de control a otros hosts y routers. Está definido en el RFC 792.

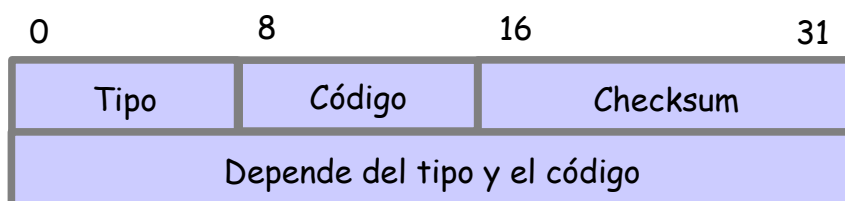
ICMP nos permite saber, por ejemplo, por qué no se ha entregado un datagrama (no hay ruta, el destino no responde, se ha agotado su tiempo de vida, etc.). Informa de errores sólo al origen del datagrama. Además, no se encarga de corregir el problema, sólo de avisar.

Como los mensajes ICMP pueden generarse en el exterior de la red IP donde se generó el datagrama original necesitan viajar en el campo de datos de un datagrama IP, pero ICMP no se considera un protocolo de nivel superior a IP, sino de nivel de red.



Cada mensaje ICMP tiene su propio formato, pero todos comienzan con los mismos campos:

- Tipo (8 bits): Identifica el tipo de mensaje
- Código (8 bits): Más información sobre el tipo de mensaje
- *Checksum* (16 bits): Utiliza el mismo algoritmo que IP.



El tipo de mensaje determina su significado y su formato. Hay 15 tipos distintos. Entre los principales tenemos:

- Tipo = 0. Respuesta de eco.
- Tipo = 3. Destino inalcanzable.
- Tipo = 8. Petición de eco.
- Tipo = 11. Tiempo de vida excedido en datagrama (TTL=0).

Los mensajes de error contienen la cabecera IP y los ocho primeros bytes de datos del datagrama original. Hay que señalar que esa información contine las direcciones IP fuente y destino, así como los puertos fuente y destino del datagrama que ha causado el error.

Para evitar problemas en la red, en particular *broadcast storms*, nunca se generan mensajes de error en respuesta a:

- Un mensaje de error ICMP.
- Un datagrama destinado a una dirección IP de difusión.
- Un fragmento que no sea el primero.
- Un datagrama cuya dirección origen no defina una conexión de red única (es decir, que la dirección origen no puede ser cero, la dirección de *loopback*, direcciones de difusión).

Mensajes ICMP de eco

La respuesta a una petición de eco devuelve los mismos datos que se recibieron en la petición. Estos mensajes se utilizan para construir la herramienta *ping*, empleada por administradores y usuarios para detectar problemas en la red.

Permite :

- Comprobar si un destino está activo y si existe una ruta hasta él.
- Medir el tiempo de “ida y vuelta”.
- Estimar la fiabilidad de la ruta.
- Puede ser utilizado tanto por hosts como por routers.

Mensajes ICMP de tiempo excedido

Este tipo de mensajes pueden ser enviados por routers y por hosts:

- Routers: cuando descartan un datagrama al llegar a cero su tiempo de vida.
- Hosts: al vencer un temporizador mientras esperan todos los fragmentos de un datagrama.

El campo código explica cuál de los dos sucesos ha ocurrido.

En estos mensajes se apoya la orden *traceroute*, que se estudiará después.

Mensajes de destino inalcanzable

Son enviados por un router o un host cuando no puede enviar o entregar un datagrama IP.

Se envían al emisor inicial del datagrama.

El campo código contiene un entero con información adicional. Los más importantes son:

- Código = 0. Red inalcanzable.
- Código = 1. Host inalcanzable.

- Código = 2. Protocolo inalcanzable.
- Código = 3. Puerto inalcanzable. Se genera usualmente cuando se recibe un datagrama UDP destinado a un puerto UDP que está cerrado en el destino.
- Código = 4. Se requiere fragmentación pero bit DF activado.
- Código = 6. Red destino desconocida.
- Código = 7. Host destino desconocido.

Ejercicio 1:

El computador B ha recibido los datagramas IP mostrados en la tabla, que han sido enviados por el computador A. Durante la recepción de los datagramas los únicos puertos abiertos en B eran los puertos **TCP 22** y **30.000**.

Nº	Identificador	MF	OFFSET	Long. Total	Protocolo	Tipo (si ICMP)/ Puerto si UDP o TCP
1	1340	1	185	1500	ICMP	8
2	1341	0	0	877	UDP	8.000
3	1342	1	0	1500	TCP	22
4	1340	0	370	78	ICMP	8
5	1342	0	185	1340	TCP	22

- ¿Qué datos recibirá el nivel de transporte? Justifica la respuesta.
- ¿Se generarán mensajes ICMP? Justifica la respuesta. En caso afirmativo indica qué datagrama(s) lo(s) generará(n).

3. Análisis de la cabecera IP

Ejercicio 2:

Inicia el analizador de protocolos *Wireshark*. Captura los paquetes que se generan al cargar en el navegador la página www.uv.es. Utiliza un filtro de captura para eliminar el resto del tráfico. Recuerda que los protocolos de aplicación se filtran indicando el puerto del servidor (port 80 para http o port 443 en el caso de https). Detén la captura, analiza los primeros 4 paquetes generados, y responde a las siguientes cuestiones referidas a la cabecera IP de dichos paquetes:

	<i>Identificador</i>	<i>TTL</i>	<i>Dirección IP fte.</i>	<i>Dir. IP destino</i>
Paquete 1				

Paquete 2				
Paquete 3				
Paquete 4				

Con respecto al campo TTL (*Time To Live*) de la cabecera IP de los paquetes capturados:

- ¿Tiene siempre el mismo valor?
- En general, todos los paquetes que envía un ordenador, ¿tienen siempre el mismo TTL inicial?
- ¿Cuál sería el valor inicial del TTL en el paquete 2 (el primero que ha enviado el servidor)?

Observa cómo varia el campo identificador en el cliente y en servidor. Describe lo que observas. Anota el valor del campo protocolo. En este caso, ¿a qué protocolo se refiere?

4. La orden *ping*

Mediante la orden *ping* (que se ejecuta desde un terminal) se obtiene una estimación del tiempo de ida y vuelta de un paquete (RTT), desde la estación origen a una estación destino que se especifica. Para ello se almacena el instante de tiempo en el que se envía el paquete y cuando llega la respuesta al valor almacenado se le resta del tiempo actual. El funcionamiento de la orden *ping* se basa en el uso de mensajes ICMP de tipo 0 (*Echo reply*) y 8 (*Echo request*).

Otras utilidades de la orden *ping* son:

- Averiguar si un destino está operativo, conectado a la red y sus protocolos TCP/IP en funcionamiento.
- Conocer la fiabilidad de la ruta entre origen y destino (calculando el porcentaje de paquetes que obtienen respuesta).

Ejemplo:

```
user@rdc14:~$ ping www.uji.es
```

```
PING www.uji.es (84.124.83.62) 56(84) bytes of data.
```

```
64 bytes from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp_seq=1 ttl=112
time=22.1 ms
```

```
64 bytes from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp_seq=2 ttl=112
time=21.6 ms
```

```
64 bytes from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp_seq=3 ttl=112
time=21.6 ms
```

```
64 bytes from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp_seq=4 ttl=112
time=22.0 ms
```

```
--- www.uji.es ping statistics ---
```

```
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
```

```
rtt min/avg/max/mdev = 21.646/21.872/22.139/0.217 ms
```

La orden **ping** admite un serie de opciones, algunas de las más útiles se muestran a continuación, según trabajemos con un sistema operativo Windows o Linux. Para ver una información más completa puede consultarse el manual de la orden.

WINDOWS :

```
ping [-n count] [-l packetsize] [-i ttl] destino
```

Opciones:

```
-n cantidad  Cantidad de solicitudes de eco a enviar.
-l tamaño    Número de bytes de datos.
-i ttl       Tiempo de vida.
```

LINUX:

```
ping [-b] [-c count] [-s packetsize] [-t ttl] destino
```

Opciones:

```
-b           Permite hacer broadcast a una dirección de difusión
-c cantidad Cantidad de solicitudes de eco a enviar.
-s tamaño   Número de bytes de datos.
-t ttl      Tiempo de vida.
```

Para interrumpir la ejecución del programa ping: hay que presionar Ctrl-C.

Ejercicio 3:

Haz un `ping -n 3` (en Linux: `ping -c 3`) a las direcciones siguientes: `www.uv.es` (servidor web de la Universidad de Valencia), `www.uvigo.gal` (servidor web de la Universidad de Vigo), `www.uq.edu.au` (servidor web de la Universidad de Queensland en Australia) y `www.berkeley.edu` (servidor web de la Universidad de California en Berkeley). La opción `-n 3/ -c 3` configura la orden **ping** para que realice únicamente tres intentos. Anota los resultados en la tabla siguiente:

	Tiempo de ida y vuelta (ms)		
	Mínimo	Medio	Máximo
www.uv.es			
www.uvigo.gal			
www.berkeley.edu			
www.uq.edu.au			

Analiza a qué se debe la disparidad de los resultados entre los distintos destinos.

Los resultados que se obtienen mediante la orden **ping** son, a veces, difíciles de interpretar. El usuario obtiene poca información de por qué el tiempo de ida y vuelta es mayor en unos destinos que en otros. Incluso cuando no hay respuesta al **ping**, no es posible conocer cuál es el problema: el destino solicitado está fuera de servicio, no existe una ruta desde el origen al destino o la

saturación de la red es tan alta que no se obtiene respuesta del destino en un tiempo razonable. También, en ocasiones por motivos de seguridad y para evitar dar información sobre los ordenadores conectados a la red, los administradores de las redes filtran los mensajes de *ping* en los cortafuegos o desactivan el servicio en los propios ordenadores. A pesar de lo dicho, es una de las herramientas que más utilizan los administradores y usuarios de equipos conectados en red.

Ejercicio 4:

Antes de iniciar la captura lee el ejercicio hasta llegar a las cuestiones.

Aplica un filtro de captura (no de visualización) que capture únicamente los paquetes ICMP generados tras la ejecución de la orden `ping -n 3 www.uv.es` (en Linux: `ping -c 3 www.uv.es`). Ejecuta la orden dos veces.

Detén la captura cuando terminen los seis intentos y observa cuántos mensajes ICMP se producen, prestando especial atención a los **campos de la cabecera ICMP: tipo, código, y bytes de datos**. Observa la diferencia entre los mensajes ICMP de petición y de respuesta de eco. Asimismo, analiza las **cabeceras IP** de cada uno de ellos, y en concreto los campos **longitud de la cabecera, longitud total y bytes de datos**. Compara el valor del campo protocolo con el que observaste en el ejercicio 2.

Respecto a los mensajes ICMP:

- ¿Por qué los mensajes ICMP no llevan números de puerto fuente y destino?
- ¿Para qué se utiliza el **número de secuencia** de la cabecera ICMP?

5. La orden *tracert*/*tracert*

La orden *tracert* en Windows o *tracert* en Linux (que se ejecuta desde un terminal) permite conocer el camino (secuencia de routers) que debe atravesar un paquete para llegar desde la estación origen a la estación destino. El funcionamiento se basa en gestionar adecuadamente un parámetro de la cabecera de los datagramas IP (el campo TTL: tiempo de vida) y en la información que aportan los mensajes ICMP que generan los routers cuando les llega un datagrama cuyo tiempo de vida se ha agotado.

Por cada nuevo router atravesado por el datagrama se dice que hay un *salto* en la ruta. Podemos decir, que el programa *tracert*/*tracert* calcula y describe el número de saltos de una ruta.

Generalmente, el campo TTL tiene 8 bits que el emisor inicializa a algún valor. El valor recomendado actualmente en el RFC de números asignados (RFC 1700) es de 64. Cada router que atraviesa el datagrama debe reducir el TTL en una unidad. Cuando un router recibe un datagrama IP con TTL igual a uno y decrementa este valor obtiene un cero. Consecuentemente, el router descarta el datagrama y envía un mensaje ICMP de tipo 11 (*tiempo excedido*) al origen que generó el datagrama. La clave para el funcionamiento del programa *tracert*/*tracert* es que este mensaje ICMP contiene la dirección IP del router que lo ha enviado.

En el caso del *tracert/traceroute*, el primer datagrama IP se envía al ordenador destino con TTL igual a 1. Si el destino no está en la misma red que el host origen, el primer router con el que se encuentre este datagrama decrementará el TTL y al obtener un cero lo descartará, enviando un mensaje ICMP de “tiempo excedido” (*Time exceed*) al origen. Así se identifica el primer router en el camino. A continuación se envía un datagrama con TTL igual a 2 para encontrar la dirección del segundo router, y así sucesivamente.

Cuando el datagrama alcance un valor de TTL suficiente para llegar a su destino, necesitaremos que el destino envíe un mensaje que nos permita detener el proceso. Para ello *tracert/traceroute* utiliza dos opciones distintas:

- **Tracert:** Enviar mensajes ICMP de eco (es la que se usa en Microsoft Windows). La respuesta al alcanzar el destino será un mensaje de respuesta de eco.
- **Traceroute:** Enviar mensajes UDP a un puerto arbitrariamente grande (en principio es el 33434) y muy probablemente cerrado (es la opción que se usa en Linux/Unix). El sistema responderá con un mensaje ICMP de puerto inalcanzable si el puerto está cerrado en el destino, pero si estuviera abierto no se recibiría respuesta y no se detectaría que ya se ha alcanzado el destino.

Por defecto, para averiguar cada nuevo salto se envían tres datagramas y para cada uno de ellos se calcula el valor del tiempo de ida y vuelta. Si en un tiempo máximo (configurable) no hay respuesta se indica en la salida mediante un asterisco.

Algunas puntualizaciones:

- No hay ninguna garantía de que la ruta que se ha utilizado una vez vaya a ser utilizada la siguiente.
- No hay ninguna garantía de que el camino seguido por el paquete de vuelta sea el mismo que ha seguido el paquete de ida. Esto implica que a partir del tiempo de ida y vuelta que ofrece *tracert/traceroute* puede no ser directo estimar el tiempo de ida o de vuelta por separado (si el tiempo que tarda el paquete en ir desde el origen hasta el *router* es de 1 segundo y el tiempo que tarda el paquete de vuelta es de 3 segundos, el valor que nos proporcionará *tracert/traceroute* será de 4 segundos).
- La dirección IP que se devuelve en el mensaje ICMP es la dirección de la interfaz entrante del router (aquella por la que se recibió el paquete).

Ejercicio 5:

Ejecuta la orden *tracert* (*traceroute* en Linux) para los siguientes destinos y anota el número de saltos.

	Saltos
www.uv.es	
www.ua.es	
www.uq.edu.au	

Observa que si se alcanza el destino, la última línea mostrada corresponde a dicho destino (en nuestro caso un servidor web) y no a un router.

Analiza cuáles pueden ser las causas de la respuesta obtenida al ejecutar la orden *tracert* www.ua.es.

En el *tracert* a www.uq.edu.au, aparecen diferencias importantes en el retardo de los enlaces que se observa, ¿cuál crees que es el motivo?

Ejercicio 6:

Desde el navegador accede a la página <http://www.telstra.net/cgi-bin/trace>. En esta página puedes indicar una dirección ip destino. El servidor de telstra que está en Melbourne hará un *tracert* desde su máquina al destino que le hayas indicado. En nuestro caso le vamos a poner la dirección IP de nuestra máquina de trabajo.

Tenemos que tener en cuenta que para que el servidor de telstra pueda hacer un *tracert* hasta nuestra máquina es necesario proporcionarle la dirección IP pública y no la privada que es la que nos proporciona nuestro router doméstico y la que vemos mediante el comando *ipconfig*. Para obtener nuestra IP pública hay muchas páginas web especializadas en ello. Una de ellas es www.cualesmiip.es.

Así obtendremos todos los routers por los que pasa un datagrama desde el servidor de Telstra hasta llegar a nuestra red.

A continuación, realiza un *tracert*/*tracert* al servidor www.telstra.net, para obtener la ruta inversa. Compara ambos resultados.

¿Se sigue el mismo recorrido desde tu máquina hasta www.telstra.net y viceversa?

Observarás que algunos routers tienen nombres similares en los dos casos pero con direcciones IP distintas, ¿a qué crees que es debido?

Ejercicio 7:

Captura los paquetes IP derivados de la ejecución de la orden *tracert -d* www.uv.es. (en Linux: *tracert -n* www.uv.es). Para capturar también los paquetes del protocolo involucrados en este comando tendrás que utilizar el filtro de captura “*icmp*” (en Linux: “*icmp or (udp and host X.X.X.X)*”, donde X.X.X.X representa la dirección IP de tu ordenador) .

Para los paquetes que envía tu ordenador:

- ¿Qué tipo de mensajes envía tu máquina? ¿Cuál es la dirección destino de estos mensajes? ¿Es siempre la misma? ¿A quién pertenece esta dirección?
- En la cabecera IP: ¿cómo varía el TTL?
- ¿Cuántos paquetes envía tu máquina con el mismo TTL?
- ¿Cuál es el TTL del último mensaje que envía tu ordenador? ¿Cuál es el número de saltos que obtuviste en la ejecución del *tracert*?

Para los paquetes de respuesta:

- ¿Qué tipos diferentes de mensajes ICMP recibe tu máquina?
- Observa la dirección IP origen de cada uno de los mensajes ICMP que recibe tu máquina. Relaciona estas direcciones con el resultado obtenido en la ejecución del *tracert*.
- Indica por qué se envía información sobre las cabeceras IP e ICMP en los paquetes ICMP de error.

Práctica 1: Configuración de TCP/IP en Windows 10 y Linux

1. Introducción

Para hacer esta práctica necesitas dos entornos distintos: Windows 10 y Linux.

- **WINDOWS 10:**

Si no dispones de este sistema operativo, puedes:

1. instalar la máquina virtual Windows 10, a partir de un su fichero .ova. Para ello, obtén el fichero WinDev2012Eval.ova desde alguno de estos enlaces:

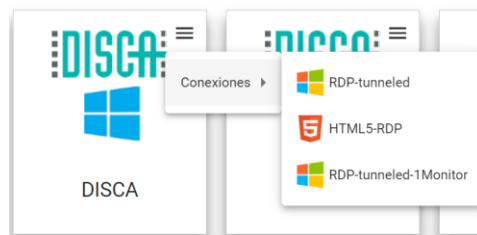
<https://developer.microsoft.com/es-es/windows/downloads/virtual-machines/>

<https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/>

<https://developer.microsoft.com/es-es/microsoft-edge/tools/vms/>

En estos dos últimos enlaces, la máquina virtual está configurada de forma predeterminada en idioma inglés. Para cambiar el teclado a español, accede "Settings", y busca la opción "Keyboard language", donde podrás seleccionar el idioma español como predeterminado.

2. acceder a Polilabs y utilizar la máquina virtual WIDOWS de DISCA:



- **LINUX:**

Si no dispones de este sistema operativo, puedes usar la misma máquina virtual que has usado durante el primer cuatrimestre en esta asignatura.

Esta práctica está dedicada a revisar el procedimiento básico de instalación y configuración de los protocolos TCP/IP en Windows 10 y Linux. Se muestra el uso de algunas herramientas útiles a la hora de resolver problemas con estos protocolos: configurar el software de red, verificar su funcionamiento y ajustar los parámetros relacionados con TCP/IP.


2. Configuración de TCP/IP en Windows 10

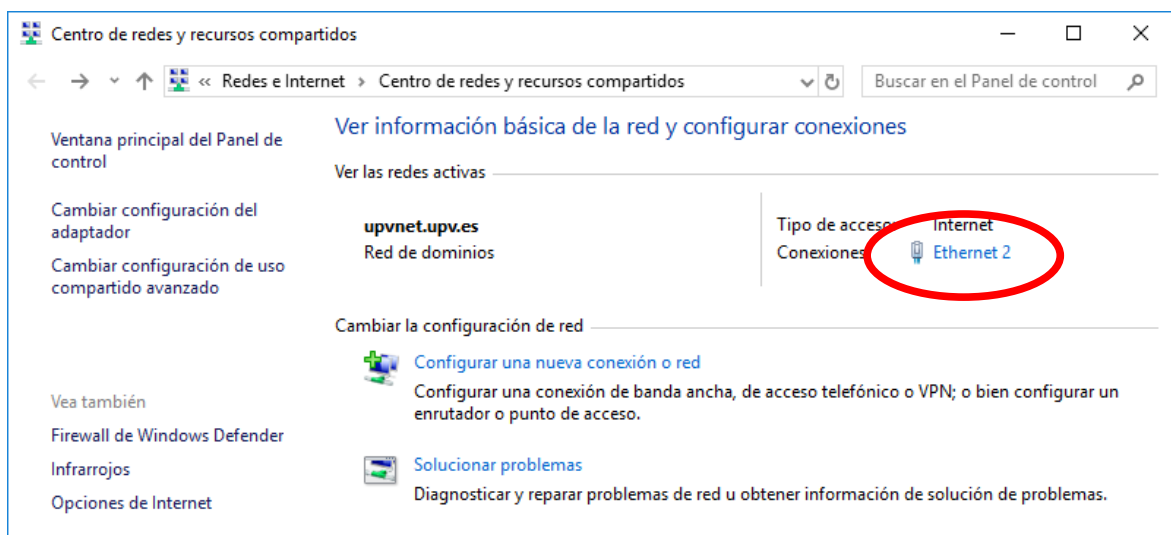
Este apartado requiere, evidentemente, iniciar en tu ordenador un entorno Windows 10.

Para utilizar los protocolos TCP/IP desde una máquina Windows 10 conectada a una red de área local es necesario tener instalada una tarjeta adaptadora o NIC (*Network Interface Adapter*). En nuestros computadores esta tarjeta ya suele estar instalada y configurada.

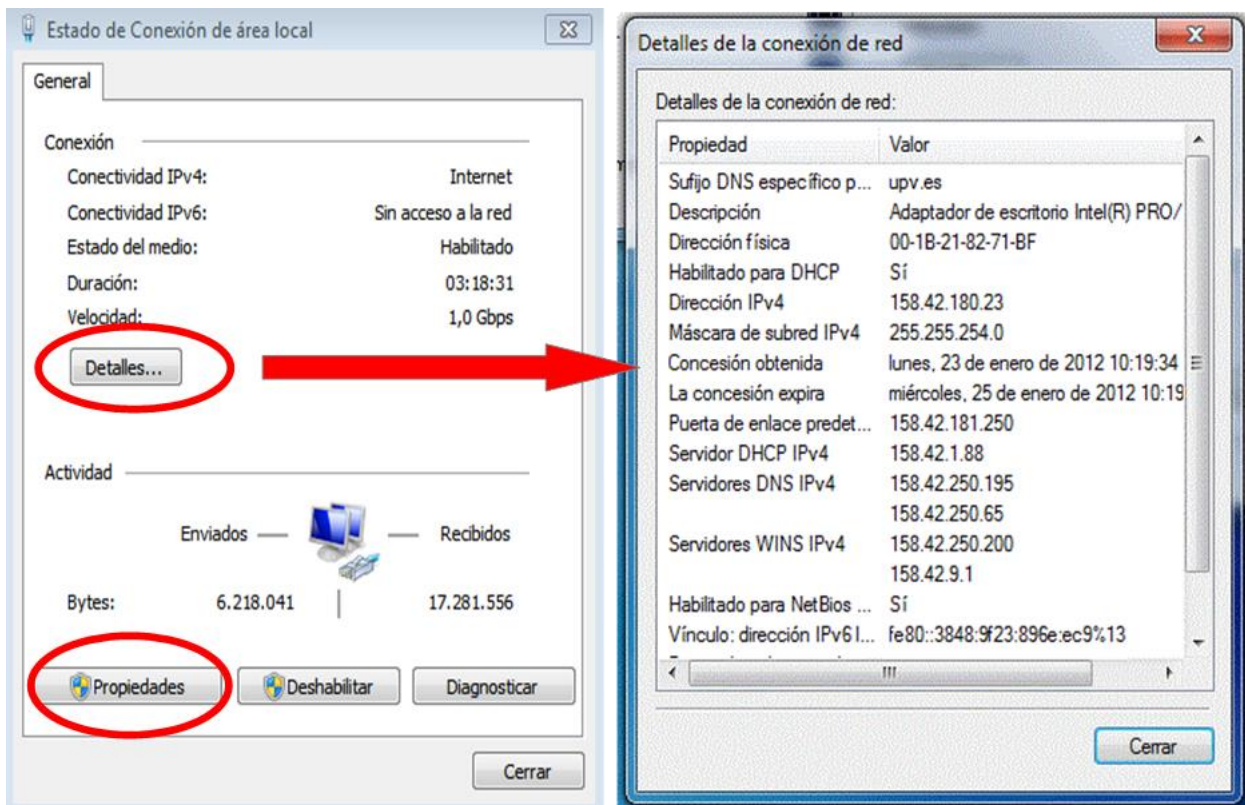
Dependiendo de si tienes una tarjeta de red Ethernet o Wifi la forma de ver la configuración puede ser ligeramente distinta. Vamos a ver los dos casos:

Para ver la configuración de la tarjeta adaptadora Ethernet:

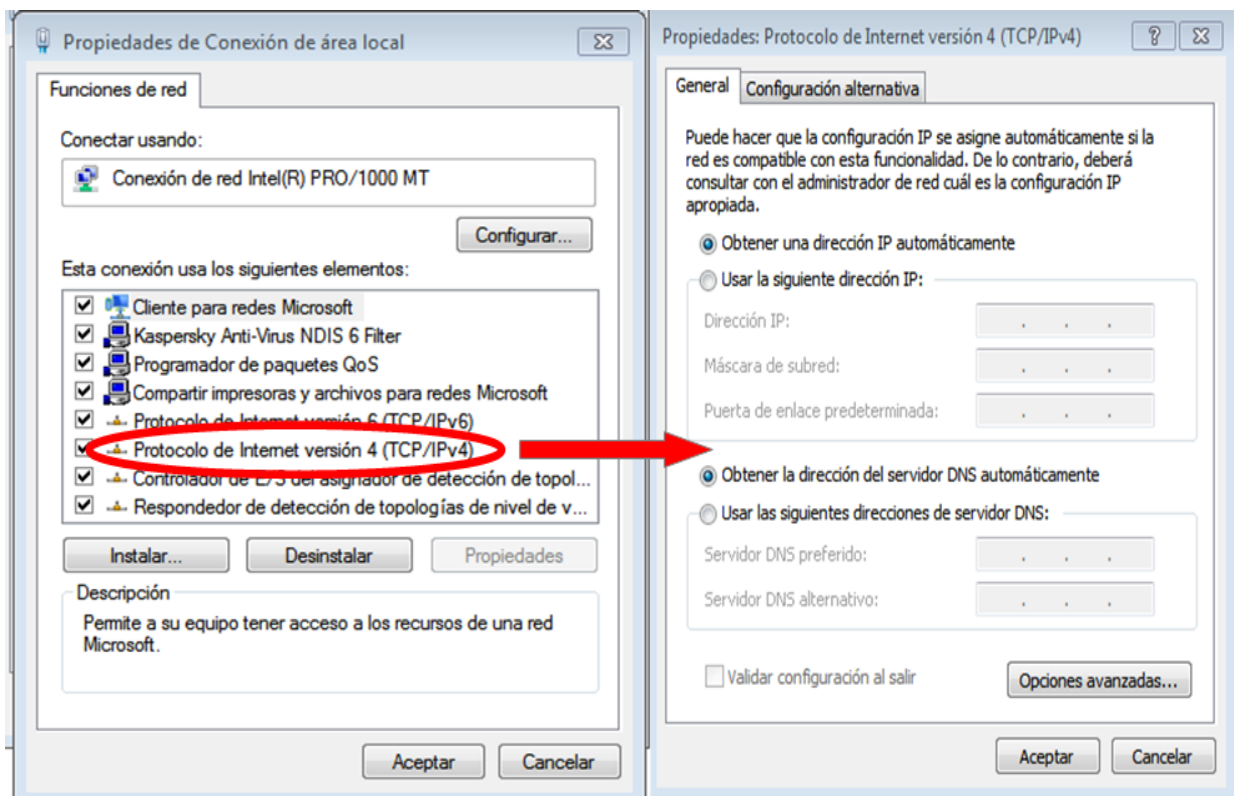
1. Pulsa el botón **Inicio**, y después **Configuración** (o  icono) → **Red e Internet**.
2. En el apartado **Centro de redes y recursos compartidos** aparece, entre otras informaciones, la red activa en la que se encuentra tu PC y el tipo de acceso a esa red. En los computadores, directamente conectados a una red local, en **conexiones** aparece el adaptador Ethernet que se está empleando. Haz clic sobre el enlace **Ethernet** (en nuestro ejemplo **Ethernet2**)



3. Se abrirá la ventana de **Estado** del adaptador Ethernet. En ella, al pulsar el botón **Detalles**, podemos ver las principales propiedades de la conexión de red asociada al adaptador: dirección física del adaptador de red, dirección IP asociada a ese interfaz, máscara de red, etc.




Además del adaptador de red, para poder utilizar aplicaciones Internet es necesario que estén instalados los protocolos TCP/IP. De nuevo, estos protocolos ya están instalados en nuestros ordenadores. Un administrador podría comprobarlo siguiendo los pasos siguientes. Si tú no lo eres, puedes observarlo en la figura siguiente.

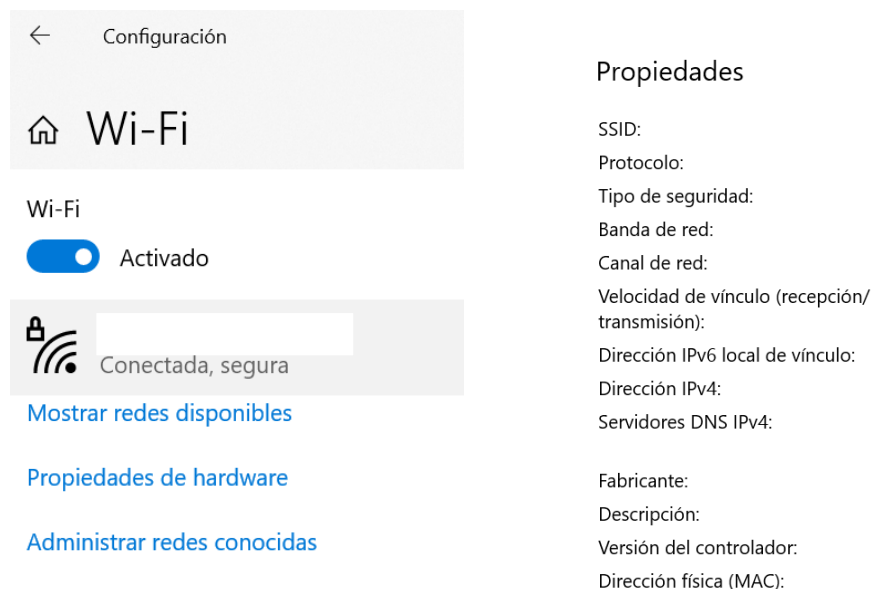


- Desde la ventana **Estado** del adaptador Ethernet (que se muestra en el gráfico anterior a la izquierda) si se pulsa sobre el botón **Propiedades**, aparece la ventana de **Propiedades**, donde se muestran todos los elementos que emplean este adaptador de red. Entre ellos destaca el protocolo de Internet versión 4, que representa la pila TCP/IP. Tras seleccionarlo, el botón **Propiedades** permite acceder a una nueva ventana que muestra los parámetros de funcionamiento.

Como podemos observar, la configuración más habitual consiste en que la mayoría de los parámetros necesarios para el funcionamiento de TCP/IP (¡incluyendo la propia dirección IP!) no se configuran manualmente, sino que se obtienen automáticamente durante el proceso de arranque de la máquina. Esto es posible gracias al protocolo DHCP (*Dynamic Host Configuration Protocol*) que permite a un cliente solicitar al servidor una dirección IP. Este protocolo, de empleo frecuente, se estudiará en una práctica posterior durante este cuatrimestre. Además de la dirección IP, el servidor DHCP proporciona información adicional necesaria para el funcionamiento de los protocolos TCP/IP (dirección IP del servidor DNS, dirección IP del router, etc.).

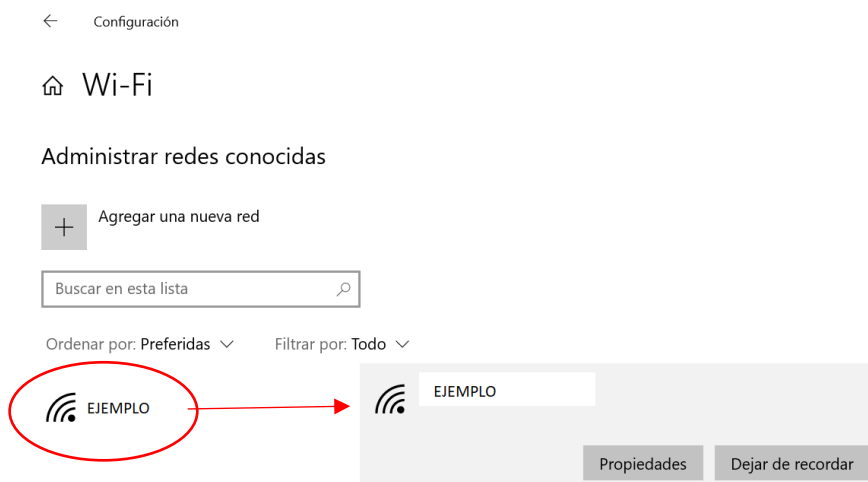
Para ver la configuración de la tarjeta adaptadora WiFi:

- Pulsa el botón **Inicio**, y después **Configuración** (o el icono ) → **Red e Internet**.
- En el apartado **Wi-Fi** aparece, entre otras informaciones, la red activa en la que se encuentra tu PC (parte izquierda de la figura)
- En **propiedades de hardware** (parte derecha de la figura) podemos ver las principales propiedades de la conexión de red asociada: dirección física del adaptador de red, dirección IP asociada a ese interfaz, etc.

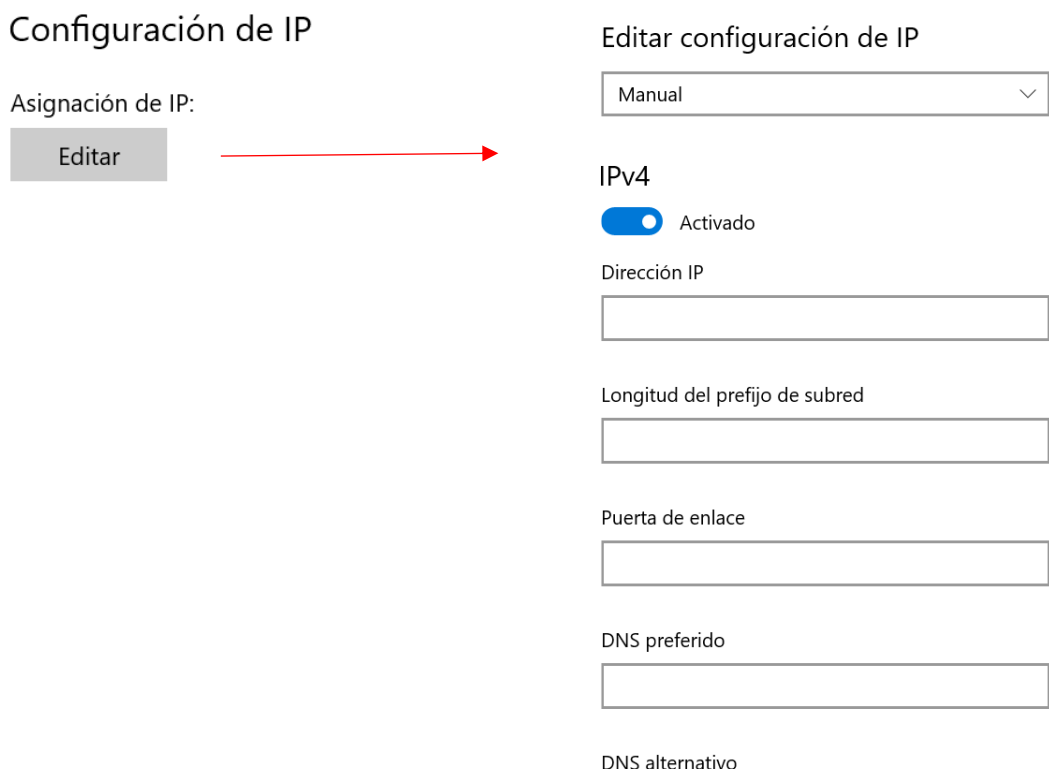


Además del adaptador de red, para poder utilizar aplicaciones Internet es necesario que estén instalados los protocolos TCP/IP. De nuevo, estos protocolos ya están instalados en nuestros ordenadores.

4. Desde la opción **Administrar redes conocidas** si se pulsa sobre nuestra red Wi-Fi (EJEMPLO), aparece la ventana de **Propiedades**, donde se muestran los elementos que relacionados con este adaptador de red.



5. Entre ellos destaca el protocolo de Internet versión 4, que representa la pila TCP/IP. Tras seleccionar en **Configuración de IP**, el botón **Editar**, si no tenemos una configuración automática puesta por defecto, podemos acceder a una nueva ventana que muestra los parámetros de funcionamiento.



La configuración más habitual consiste en que la mayoría de los parámetros necesarios para el funcionamiento de TCP/IP (¡incluyendo la propia dirección IP!) no se configuran manualmente, sino que se obtienen automáticamente durante el proceso de arranque de la máquina. Esto es posible gracias al protocolo DHCP (Dynamic Host Configuration Protocol) que permite a un cliente solicitar al servidor una dirección IP. Este protocolo, de empleo frecuente, se estudiará en una

práctica posterior durante este cuatrimestre. Además de la dirección IP, el servidor DHCP proporciona información adicional necesaria para el funcionamiento de los protocolos TCP/IP (dirección IP del servidor DNS, dirección IP del router, etc.).

2.1 La orden **ipconfig**



Una vez que los protocolos TCP/IP están instalados, la orden **ipconfig** (se ejecuta desde una ventana de DOS – Símbolo del sistema – a la que puede accederse desde el menú “Inicio” tecleando `cmd` en la ventana de texto inferior) proporciona información sobre la configuración de la red en nuestra máquina (para cada uno de los adaptadores de red instalados).

Si ejecutamos **ipconfig /?** obtendremos información sobre la orden

ipconfig nos permite, entre otras cosas, obtener información para cada uno de los adaptadores de red instalados en nuestro ordenador sobre:

- **Dirección IPv4 e IPv6:** direcciones IP asignadas a nuestra máquina, en nuestro caso de forma dinámica mediante el protocolo DHCP.
- **Máscara de subred:** indica qué parte de la dirección IPv4 identifica la red, y qué parte identifica al computador (a un adaptador de red). La red de la UPV globalmente tiene asignado el bloque de direcciones IPv4 158.42.0.0/16, que se ha desglosado en una serie de subredes. La máscara de subred (255.255.254.0) indica que, en el caso de la subred del laboratorio, los 23 bits más significativos de cada dirección IPv4 (bits a 1 en la máscara) deben considerarse identificador de red, y los 9 últimos (bits a 0 en la máscara) identificador de *host*.
- **Puerta de enlace predeterminada:** dirección IP del router que conecta nuestra subred con el resto de la red de la UPV y con el exterior (Internet).

Ejercicio 1

Desde el intérprete de ordenes de Windows ( > Sistema de Windows >  **Símbolo del sistema**) ejecuta `ipconfig /?` para visualizar sus opciones. A continuación, identifica cuál es la forma de averiguar los parámetros explicados en párrafos anteriores (**Dirección IPv4 e IPv6**, **Máscara de subred**, **Puerta de enlace predeterminada**) usando esta orden.

En la respuesta obtenida, identifica cuál de todos los adaptadores que aparecen es el que te conecta a Internet y por qué.

Si ejecutamos la orden **ipconfig /all** obtenemos información adicional, entre la cual destaca:

- **Dirección física:** es la dirección física que corresponde a la tarjeta adaptadora de red (Ethernet en nuestro caso) que está instalada en nuestro computador y nos permite el acceso a la red.
- **Servidores DNS:** la dirección IP de la(s) máquina(s) que realiza(n) las traducciones de nombres a direcciones IP (servidor de nombres).
- **Servidor DHCP:** dirección IP de la máquina que nos ha asignado la dirección IP y la mayoría de parámetros que aparecen en esta ventana.
- **Concesión obtenida (la concesión expira):** fecha en la que fue obtenida (caducará) la dirección IP actual. Aplicable únicamente en el caso de información obtenida por DHCP.

Las órdenes **ipconfig /release** e **ipconfig /renew** permiten liberar y renovar la dirección IPv4 obtenida mediante DHCP.

Ejemplos:

```
> ipconfig /renew           Renueva todos los adaptadores.
> ipconfig /renew EL*      Renueva cualquier conexión cuyo nombre con EL.
> ipconfig /release *Con*  Libera todas las conexiones coincidentes, por ejemplo:
                           "Conexión de área local 2".
```

Ejercicio 2

Ejecuta la orden **ipconfig /all** y completa la información siguiente relativa al adaptador que tenga asociada una dirección IP.

Dirección física del adaptador Ethernet conexión de área local	
Dirección IPv4	
Máscara de subred	
Dirección IP del router (puerta de enlace)	
Servidores DNS	
Servidor DHCP	

Según la información obtenida:

- ¿Cuál es la dirección IP de la red a la que está conectado tu equipo?
- Los servidores DNS y DHCP, ¿están en la misma subred que tu ordenador? ¿Cómo lo has averiguado?

Ejercicio 3

Comprueba el contenido de la caché DNS. Anota en la tabla los valores de uno de los registros que aparecen que sea de tipo 1:

Tipo registro	
Nombre registro	
Valor registro (un registro <host>)	

2.2 La orden ping

Mediante la orden **ping** (que se ejecuta desde una ventana DOS) se obtiene una estimación del tiempo de ida y vuelta de un paquete (RTT, *Round Trip Time*), desde la estación donde se ejecuta la orden a la estación destino que se especifica. El funcionamiento de la orden **ping** se basa en el uso de mensajes ICMP de eco, que se estudiarán en una práctica posterior.

Ejemplo: ejecuta la orden `ping www.upc.es` y observa lo que hace.

La orden **ping** admite una serie de opciones, la única que nos interesa de momento se muestra a continuación:

```
ping [-n cantidad] destino
```

`-n cantidad` número de solicitudes de eco a enviar.

Esta orden se analizará con detalle en la práctica destinada al estudio del protocolo ICMP. Hasta ese momento la emplearemos únicamente con el propósito de saber si un destino determinado puede alcanzarse o no a través de la red y cuál es su dirección IP.

2.3 La orden tracert

La orden **tracert** (se ejecuta desde una ventana DOS) permite conocer el camino (secuencia de routers) que debe atravesar un paquete para llegar desde la estación origen a la estación destino. Se basa en el empleo de mensajes ICMP y, por lo tanto, también se estudiará en la práctica destinada a este protocolo.

Ejemplo: ejecuta la orden `tracert www.upc.es` y observa lo que hace. Cuando finalice, ejecuta `ping www.upc.es` e intenta justificar el valor del campo TTL de las respuestas.

2.4 La orden netstat

La orden **netstat** (desde **Símbolo del sistema**) ofrece diversa información sobre el estado y estadísticas de los protocolos de red. Se pueden obtener datos sobre los principales sucesos Ethernet, IP, ICMP, UDP y TCP. Ejecuta `netstat -h`, y confirma que obtienes, entre otras, la información que se muestra a continuación:

```
netstat [-a] [-e] [-n] [-s] [-p proto] [-r]
```

`-a` Muestra todas las conexiones y puertos escucha.

`-e` Muestra estadísticas Ethernet. Se puede combinar con `-s`.

`-n` Muestra números de puertos y direcciones en formato numérico.

`-p proto` Muestra conexiones del protocolo especificado por **proto**; que puede ser TCP, UDP, TCPv6 o UDPv6. Si se usa con la opción `-s` para mostrar estadísticas por protocolo, **proto** puede ser TCP, UDP, TCPv6 o UDPv6.

`-r` Muestra el contenido de la tabla de rutas.

`-s` Muestra estadísticas por protocolo. De forma predeterminada, se muestran para los protocolos IP, IPv6, TCP, UDP e IP; se puede utilizar la opción `-p` para especificar un subconjunto de los valores predeterminados.

Mediante la orden `netstat -r` obtenemos información sobre la tabla de encaminamiento (produce la misma salida que la orden `route print`).

Cuando hay que encaminar un datagrama, para averiguar la ruta se sigue el proceso de reenvío tal como se estudió en las clases de teoría. En concreto, el mecanismo es el siguiente:

1. Para cada línea de la tabla de encaminamiento, se realiza un AND lógico entre la **dirección IP destino** del datagrama y la **máscara de red**. IP compara el resultado con la **Red destino** y marca todas las rutas en las que se produce coincidencia.

- De la lista de rutas coincidentes IP selecciona la ruta que tiene más bits en la máscara. Esta es la ruta más específica y se conoce como la **ruta de máxima coincidencia** (*longest matching*).
- Si hay varias rutas de máxima coincidencia, se usa la ruta con menor **métrica**. Si hay varias con la misma métrica se usa una cualquiera de ellas.

Ejercicio 4:

Visualiza la tabla de encaminamiento (apartado IPv4) del ordenador en el que estás trabajando. Averigua y anota las IPs de los destinos siguientes (recuerda los ejercicios anteriores) y analiza qué ruta de la tabla se seleccionaría para cada uno de ellos:

- Un paquete destinado a `zoltar.redes.upv.es`
- Un paquete destinado a `www.upv.es`
- Un paquete destinado a `www.usc.edu`

Ejercicio 5:

La orden **netstat -e** proporciona estadísticas sobre el número de bytes y tramas enviadas y recibidas por el adaptador de red. Se detalla el número de tramas unicast (un solo destino), no unicast (múltiples destinos y difusiones), paquetes erróneos y descartados.

Ejecuta esta orden y anota los resultados en la tabla siguiente:

	Recibidos	Enviados
Paquetes de unidifusión (unicast)		
Paquetes no de unidifusión (no unicast)		
Descartados		
Errores		

Comprueba también la ejecución de **netstat -es**. Indica las diferencias que observas entre el formato de salida de las dos ejecuciones.

Ejercicio 6:

La orden **netstat -sp IP** produce estadísticas sobre el tráfico IP. Ejecuta esta orden y anota los resultados en la tabla siguiente:

	Cantidad
Paquetes recibidos	
Errores de encabezado recibidos	
Errores de dirección recibidos	
Datagramas reenviados	
Protocolos desconocidos recibidos	
Datagramas correctamente fragmentados	

Ejercicio 7:

Análogamente la orden **netstat –sp TCP** produce estadísticas sobre el tráfico TCP (también se pueden solicitar estadísticas sobre los protocolos ICMP y UDP). Ejecuta esta orden y anota los resultados en la tabla siguiente:

	Cantidad
Activos abiertos	
Pasivos abiertos	
Intentos de conexión erróneos	
Conexiones actuales	

¿A qué hacen referencia las dos primeras filas de la tabla (“Activos abiertos” y “Pasivos abiertos”)?

La orden **netstat** sin argumentos ofrece información sobre las conexiones activas en nuestra máquina. Si se utiliza con la opción **–a**, además de la información anterior se indica también la relación de puertos TCP y UDP en los que hay alguna aplicación escuchando (dispuesta a aceptar conexiones TCP o datagramas UDP).

2.5 La orden arp

Esta orden también resulta de mucha utilidad para la configuración y diagnóstico de problemas en redes. Para analizarla de forma detallada, se dedicará una práctica al protocolo ARP más adelante durante este cuatrimestre.

Ejemplo: ejecuta la orden **arp –a** para observar las direcciones IP de las máquinas con las cuales ha interactuado tu PC, y su dirección física asociada.

3. Configuración de TCP/IP en Linux

Para este apartado debes arrancar tu computador en LINUX, o emplear la máquina virtual LAB-REDES.

En Linux podemos encontrar utilidades equivalentes a las que se han estudiado en el apartado anterior. Para verificar la conectividad con una determinada máquina disponemos de las órdenes **tracert**, y **ping**, que cumplen la misma misión, respectivamente, que **tracert** y **ping** de Windows 10.

En cuanto al acceso a la configuración local, tradicionalmente, en Linux encontrábamos órdenes equivalentes a las que acabamos de estudiar dentro del entorno Windows 10, en particular:

- Orden **ifconfig**, muy parecida a **ipconfig** de Windows.
- Orden **arp**, equivale a la orden de Windows del mismo nombre.
- Orden **netstat**, equivalente a la que hemos estudiado para Windows.

No obstante, en las últimas versiones de Linux se han sustituido todas ellas por la orden **ip**, mucho más potente. Por ello, aunque en sistemas más antiguos debamos emplear las funciones anteriores, esta práctica se centra en la orden **ip**.

3.1 La orden *ip*

La orden **ip**, que puede ejecutarse desde un terminal de red, permite configurar y obtener información sobre diversos aspectos de la configuración de red. Al ejecutar **ip** sin parámetros se obtiene una descripción de sus posibilidades:

```
usulocal@rdcvm:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
                  tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm |
                  netns | l2tp | fou | macsec | tcp_metrics | token | netconf | ila |
                  vrf | sr }
      OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                  -h[uman-readable] | -iec |
                  -f[amily] { inet | inet6 | ipx | dnet | mpls | bridge | link } |
                  -4 | -6 | -I | -D | -B | -O |
                  -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                  -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                  -rc[vbuf] [size] | -n[etns] name | -a[ll] | -c[olor]}
```

usulocal@rdcvm:~\$

Como puede verse, **ip** tiene muchas más posibilidades que **ipconfig** de Windows 10. Permite trabajar sobre varios elementos de la pila de protocolos: mediante **ip link** se accede a los adaptadores de red; **ip address** se refiere a las direcciones IPv4 e IPv6 del adaptador; **ip route** accede a la tabla de reenvío mientras que **ip neighbour** permite acceder a las tablas de caché de vecinos (protocolos ARP en IPv4 y ND en IPv6), mientras que **ip ntables** permite gestionar estas tablas. Existen más posibilidades que se comentarán en prácticas posteriores.

3.1.1 La orden *ip address*

Para obtener la información relativa a los adaptadores de red del sistema y sus direcciones IP asociadas, la orden adecuada es **ip address list** o **ip address show**. También valdría emplear **ip address**, puesto que **show** es la opción por defecto. Además, pueden abreviarse los parámetros siempre que no exista ambigüedad; **ip address** puede escribirse como **ip addr** e incluso **ip a**.

```
usulocal@rdcvm:~$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:84:e5:1b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 85360sec preferred_lft 85360sec
    inet6 fe80::a00:27ff:fe84:e51b/64 scope link
        valid_lft forever preferred_lft forever
```

Como puede verse, la máquina virtual de pruebas tiene dos interfaces de red. El interfaz **lo** corresponde a una tarjeta de red virtual que implementa los accesos a *local loopback*, es decir, recibe los datagramas IP enviados a direcciones del tipo 127.x.x.x y los devuelve como si hubieran sido recibidos desde la red.

Por el contrario, el interfaz **enp0s3** corresponde a la tarjeta de red de la máquina virtual. En un sistema real, no virtualizado, correspondería al adaptador de red que une físicamente el computador con la red local.

Para cada interfaz, **ip** muestra cuáles son sus direcciones IP asociadas, tanto IPv4 como IPv6, su prefijo de red y su rango: En el caso de las direcciones IPv4, todas las direcciones son globales, mientras que la dirección IPv6 asignada pertenece a un ámbito más restringido (direcciones locales).

También aparece la información relativa al adaptador a nivel de enlace de datos: Su dirección física (que veremos en prácticas sucesivas), la MTU de la red (fundamental para la siguiente práctica de fragmentación), y si el interfaz está activo (UP) o no.

Ejercicio 8:

Utiliza la orden **ip** para averiguar cuántos adaptadores de red existen en tu computador, cual es la dirección IP y máscara de red de cada uno de ellos. Interpreta si permiten difusiones (broadcast), y multidifusión (multicast) y cuál es la MTU de cada red accesible a través de cada uno. Verifica, además, si son direcciones permanentes o deben ser renovadas periódicamente, así como si son globales o locales.

La orden **ip address** ofrece a un administrador del sistema la posibilidad de añadir o eliminar direcciones de red sobre un adaptador. Pueden asignarse varias direcciones IP al mismo interfaz. Así, ejecutando **ip address add 192.168.1.153/255.255.255.0 dev enp0s3** se asigna una segunda dirección al interfaz:

```

sulocal@rdevm:~$ sudo ip address add 192.168.1.153/255.255.255.0 dev enp0s3
[sudo] contraseña para usulocal:
usulocal@rdevm:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
    link/ether 08:00:27:84:e5:1b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 70735sec preferred_lft 70735sec
    inet 192.168.1.153/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe84:e51b/64 scope link
        valid_lft forever preferred_lft forever

```

Como es habitual, la primera vez que se emplea **sudo** es necesario repetir la contraseña. También podría indicarse una dirección de difusión para el mismo mediante **ip address add broadcast 192.168.1.255 dev enp0s3**, o bien realizar ambas simultáneamente mediante **ip address add 192.168.1.153/24 broadcast + dev enp0s3**. El “+” indica que para obtener la dirección de difusión debe sustituir cada uno de los bits de *host* de la IP proporcionada por “1”. De forma análoga, sustituir “add” por “del” permite eliminar esta asignación.

Esta funcionalidad permite, por ejemplo, comprobar localmente un programa cliente sobre un servidor de IP fija, asignando dicha IP al adaptador **lo**. Por ejemplo:

```

usulocal@rdevm:~$ sudo ip address add 158.41.1.1/255.255.0.0 dev lo
usulocal@rdevm:~$ ping 158.41.1.1

```

```

PING 158.41.1.1 (158.41.1.1) 56(84) bytes of data.
64 bytes from 158.41.1.1: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 158.41.1.1: icmp_seq=2 ttl=64 time=0.032 ms
64 bytes from 158.41.1.1: icmp_seq=3 ttl=64 time=0.030 ms
^C
--- 158.41.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.020/0.027/0.032/0.006 ms
usulocal@rdevm:~$ traceroute 158.41.1.1
traceroute to 158.41.1.1 (158.41.1.1), 30 hops max, 60 byte packets
 1  158.41.1.1 (158.41.1.1)  0.024 ms  0.009 ms  0.008 ms

```

Ejercicio 9:

Accede a la página web principal de la UPV (www.upv.es). A continuación, utiliza la orden **ip** para asignar la dirección IP correspondiente (158.42.4.23/16) a tu adaptador **lo**. Repite el acceso y comprueba las diferencias.

Revierte el cambio, eliminando esta dirección IP de tu adaptador, y comprueba nuevamente el acceso.

3.1.2 La orden *ip link*

Es posible obtener una relación de los interfaces de red existentes en el computador ejecutando **ip link**. Sin embargo, esta orden no proporciona información acerca de la configuración relativa a protocolos por encima de nivel de enlace.

```

usulocal@rdevm:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
DEFAULT group default qlen 1000
    link/ether 08:00:27:84:e5:1b brd ff:ff:ff:ff:ff:ff

```

También se proporcionan estadísticas de uso de los interfaces mediante el parámetro **-s**. Se indican tanto los bytes/paquetes transmitidos y recibidos correctamente como el número de paquetes que han sufrido algún error.

```

sulocal@rdevm:~$ ip -s link show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
DEFAULT group default qlen 1000
    link/ether 08:00:27:84:e5:1b brd ff:ff:ff:ff:ff:ff
    RX: bytes    packets  errors  dropped  overrun  mcast
    978092      3033    0       0       0       0
    TX: bytes    packets  errors  dropped  carrier  collsns
    408153      2765    0       0       0       0

```

Ejercicio 10:

Utiliza la orden **ip** para averiguar cuántos adaptadores de red existen y el número de bytes transmitidos y recibidos a través de cada uno de ellos.

Comprueba que las cifras se incrementan tras ejecutar **ping localhost** y **ping www.rediris.es**

Esta orden permite a los administradores (en nuestro caso, anteponiendo **sudo**) añadir y eliminar interfaces de red de forma manual, si bien esta funcionalidad supera las expectativas de esta práctica. Sí puede resultar útil la capacidad de modificar algunos parámetros del mismo. Por ejemplo, es posible desactivar el interfaz mediante la orden **ip link set <interfaz> down**, impidiendo pues su utilización. Para restaurar su funcionalidad se emplea **ip link set <interfaz> up**.

Ejercicio 11:

Desactiva uno de los interfaces de tu computador, y comprueba el funcionamiento de los *pings* anteriores. A continuación, repite la prueba tras activar el primero y desactivar otro. Observa los resultados.

Otra de las opciones interesantes de **ip link set** consiste en la posibilidad de manipular casi cualquier parámetro del enlace. Por ejemplo:

- **ip link set dev** <adaptador> **arp (on|off)** permite habilitar y deshabilitar las respuestas ARP –protocolo que comentaremos en prácticas siguientes – desde el interfaz.
- **ip link set dev** <adaptador> **multicast (on|off)** permite el tráfico *multicast* a través del mismo.
- **ip link set dev** <adaptador> **promisc (on|off)** cambia el adaptador en modo promiscuo, es decir, el adaptador recibe todo el tráfico aunque no vaya dirigido a él. Es especialmente útil para la monitorización de redes, por ejemplo, empleando **Wireshark**.
- **ip link set dev** <adaptador> **address** <dirección> permite cambiar manualmente la dirección física del adaptador.
- **ip link set dev** <adaptador> **mtu** <tamaño> modifica el tamaño máximo del campo de datos de la unidad de enlace de datos (trama).

3.1.3 La orden *ip route*

La orden **ip route** permite acceder a las tablas de reenvío del sistema. Linux emplea varias tablas para gestionar las rutas. En general, las rutas se almacenan en la tabla 254, también denominada **main**, y el sistema únicamente emplea esta tabla al calcular rutas. La tabla 255, denominada local, almacena las rutas hacia direcciones locales y de difusión. Esta distribución en tablas resulta especialmente interesante al aplicar distintas políticas de reenvío.

La forma más sencilla de utilizarla es **ip route show**:

```
usulocal@rdcvm:~$ ip route show
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
```

Al no indicar la tabla, se muestra la tabla principal (**main**). La interpretación de esta tabla de reenvío es sencilla. La ruta por defecto – default – consiste en transmitir a 10.0.2.2 (el router por defecto) empleando el adaptador de red **enp0s3**. Esta configuración ha sido obtenida mediante DHCP, el protocolo de configuración automático de parámetros de IP, que se estudiará en prácticas sucesivas. Finalmente, la IP origen de los datagramas IP enviados por defecto será 10.0.2.15 – nuestra dirección IP – y la métrica asociada a este enlace es 100.

Esta métrica sirve como parámetro adicional para la selección de rutas. Como se ha visto en las clases de teoría, si varias entradas de la tabla de reenvío son aplicables se elige la ruta con el prefijo de red más largo. En igualdad de longitud, se empleará esta métrica, que representa el coste de uso del enlace, para seleccionar la ruta óptima (de menor coste). La segunda línea se aplica para las entregas directas, es decir, el reenvío de los datagramas IP cuyo destino se encuentra en la misma red IP. En este caso, la información ha sido incluida en la tabla a instancias del propio sistema operativo (*kernel*).

Es posible acceder a la tabla de reenvío para el protocolo IPv6 empleando el parámetro **-6** :

```
usulocal@rdcvm:~$ ip -6 route show
fe80::/64 dev enp0s3 proto kernel metric 256 pref medium
```

Ejercicio 12:

Utiliza la orden **ip route show table all** para visualizar todas las reglas de reenvío de tu sistema, e interpreta el resultado. Prueba también con las tablas **main** y **local** sustituyendo **all** en la orden anterior.

Para un usuario administrador (anteponiendo **sudo**), **ip route add default via 192.168.1.150/24** permitirá modificar la ruta por defecto del sistema. También pueden añadirse rutas específicas en esta tabla (**ip route add 172.16.32.0/24 via 192.168.1.150/24 dev enp0s3**) e incluso rutas particulares para direcciones individuales (**ip route add 172.16.32.32 via 192.168.1.150/24 dev enp0s3**). Al igual que en el caso anterior, “del” permite anular el efecto de “add”.

En este apartado emplearemos la posibilidad de modificar las tablas para ver el efecto sobre el encaminamiento de los paquetes de las entradas más importantes. En particular, analizaremos la entrada por defecto, que nos permite alcanzar el resto de Internet.

Ejercicio 13:

- 1) Visualiza la tabla de encaminamiento de tu ordenador (orden **ip route show**).
- 2) Encuentra y anota la dirección del router de salida de la red. Nos referiremos a ella como **dir_IP_de_tu_router**.
- 3) Elimina la entrada de la dirección de red por defecto (**sudo ip route del default**) y visualiza la tabla de encaminamiento.
- 4) Intenta acceder a un destino fuera de tu red IP. Por ejemplo, mediante la orden

ping -c 2 www.upv.es

Explica que pasa.

- 5) Vuelve a probar el ping empleando ahora la dirección IP de www.upv.es (158.42.4.23) en vez del nombre del servidor.
- 6) Intenta acceder a un destino que esté en la misma red IP que tu ordenador. Por ejemplo, al router por defecto mediante la orden **ping -c 2 dir_IP_de_tu_router**.
- 7) Restaura la línea de la tabla de encaminamiento que habías eliminado (**sudo ip route add default via dir_IP_de_tu_router**).
- 8) Comprueba el estado de la tabla de encaminamiento. Debe ser el mismo que era antes de eliminar la ruta. Verifica también que ahora sí funcionan los pings a computadores fuera de tu red.

Práctica 3: DHCP, funcionamiento y análisis de trazas.

1. Sesión L3

Lectura previa: Kurose 4.3.3, subapartado “Cómo obtener una dirección de host: Protocolo de configuración dinámica de host” (pags. 284-286).

Trabajo previo a realizar antes de la sesión de laboratorio:

- Lectura del apartado: **Introducción.**
- Estudio del apartado: **Funcionamiento del protocolo DHCP.**

La práctica se realizará en **WINDOWS**, aunque en los Ejercicios 1 y 2 se plantean las alternativas para realizar dichos ejercicios en Linux. Y a partir del Ejercicio 3 se podría realizar igualmente en Linux.

2. Objetivos

Al acabar esta práctica deberías conocer lo suficiente sobre el protocolo DHCP para ser capaz de:

- Explicar la utilidad del protocolo DHCP.
- Describir sus mensajes básicos y cómo se llaman.
- Interpretar los principales campos de un mensaje DHCP capturado mediante el programa Wireshark.
- Explicar el papel que desempeña un agente retransmisor DHCP.
- Interpretar en una captura Wireshark si se utiliza o no un agente retransmisor y, en el caso de que se utilice, cuál es su dirección IP.

3. Introducción.

En esta práctica vamos a estudiar la forma más habitual que tiene un nodo para obtener una dirección IP: mediante la utilización del **Protocolo de Configuración Dinámica de Host** (Dynamic Host Configuration Protocol).

Hemos estudiado en clase, cómo una organización puede obtener un bloque de direcciones IP para ser distribuido entre todos los nodos que pertenecen a su subred. También hemos visto que un nodo necesita tener una dirección IP para quedar perfectamente identificado en Internet y esta dirección se la tiene que asignar su organización (habitualmente su ISP).

Esta asignación se puede hacer de forma manual, es decir, es el administrador el que configura el equipo asignando una dirección IP fija. Esta forma de configuración es habitual en los routers.

Pero otras veces es interesante que la asignación se realice de forma automática, en el proceso de arranque del sistema. Esto resulta especialmente útil cuando las computadoras tienen la opción de conectarse a redes diferentes, como es el caso hoy en día de los equipos portátiles.

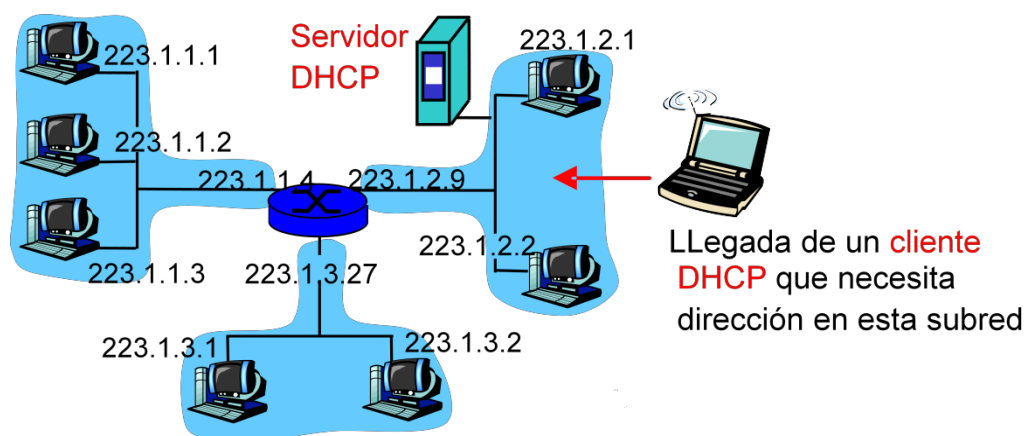
El protocolo DHCP permite realizar esta asignación automática de direcciones IP. Además de la asignación de direcciones IP a los nodos, DHCP también permite que un

nodo obtenga información adicional necesaria para el funcionamiento en Internet: máscara de subred, dirección del router asignado (gateway) y dirección de su servidor DNS local.

A pesar de que abordamos el estudio del protocolo DHCP junto con el nivel de red de la pila de protocolos TCP/IP, es necesario aclarar que el protocolo DHCP es un protocolo del **nivel de aplicación** que se apoya en el protocolo UDP. DHCP permite que dos partes se comuniquen:

- **Cliente DHCP:** nodo que se conecta a una subred y solicita dirección IP.
- **Servidor DHCP:** nodo que se encarga de gestionar el bloque de direcciones IP de una organización. El servidor DHCP se identifica mediante el puerto 67.

En el caso más simple cada subred tendrá un servidor DHCP. Si en la subred no hay ningún servidor, será necesario un agente de retransmisión DHCP (normalmente un router) que conozca la dirección de un servidor DHCP para dicha red. La figura que mostramos a continuación nos presenta los dos casos



4. Funcionamiento del protocolo DHCP

Cuando un nodo arranca y no tiene todavía configuración IP, tendrá que pasar por cuatro etapas para conseguirla:

1. **Etapla de Descubrimiento:** cuando un nodo con configuración IP dinámica arranca ni siquiera tiene la información de la dirección de la red a la que se está conectando, ni mucho menos la dirección de un servidor DHCP de esa red. La primera tarea, por tanto, será encontrar un servidor DHCP con el que interactuar.

Para ello, el nodo enviará un mensaje DHCP de descubrimiento (DHCPDiscover) al puerto 67 y dirigido a toda la red. Como no conoce su propia IP ni la del servidor, el datagrama IP que contenga el mensaje de descubrimiento utilizará las siguientes direcciones:

- Dirección IP origen: 0.0.0.0. Ya que el nodo todavía no tiene dirección IP asignada.
- Dirección IP destino: 255.255.255.255. Dirección IP de difusión para que llegue a toda la red.

Por último, comentar que el mensaje DHCPDISCOVER contendrá un **Identificador de Transacción** que permite asociar las respuestas con la petición.

2. **Etapla de ofrecimientos:** El mensaje DHCPDISCOVER lo van a recibir todos los elementos de la red local, incluidos todos los servidores DHCP de la red. Pero sólo los

servidores que hayan sido programados para responder a un cliente en particular enviarán un mensaje DHCP de ofrecimiento (DHCPOFFER).

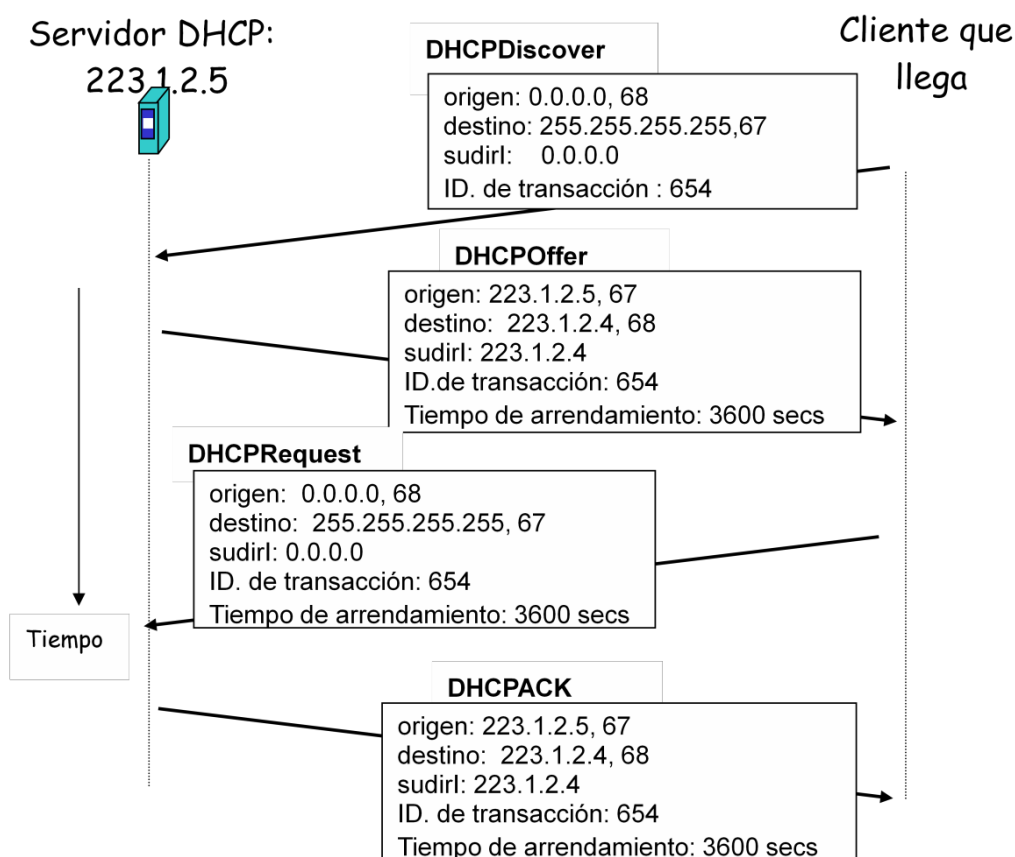
Por tanto, un cliente podrá recibir cero o más respuestas. Cada mensaje DHCPOFFER contendrá: el Identificador de Transacción que llevaba el mensaje DHCPDISCOVER, la dirección IP que el servidor ofrece al cliente, la máscara de red correspondiente y el tiempo de concesión de la dirección IP ofrecida: tiempo de validez de dicha dirección.

Los mensajes DHCPOFFER son unicast, salvo que el cliente solicite que le devuelvan la respuesta por difusión:

- Dirección origen: dirección IP del servidor DHCP.
- Dirección destino: dirección IP que el servidor ofrece al cliente o 255.255.255.255 si el cliente lo ha solicitado.

3. **Etapa de Petición DHCP:** El nodo cliente deberá seleccionar una de las respuestas según algún criterio: primero en llegar, mayor tiempo asignado,... Después de elegir, el cliente mandará un mensaje de petición DHCP al servidor elegido: DHCPREQUEST. Este mensaje repetirá los parámetros de configuración que le han propuesto al cliente.
4. **Etapa de Confirmación:** Enviado el mensaje de petición, el cliente espera la confirmación del servidor, que le llegará mediante un mensaje DHCP de reconocimiento (DHCPACK).
5. A partir de este momento el nodo cliente pasará a un estado estable en el que ya podrá utilizar la dirección IP asignada durante el tiempo de arrendamiento.

La siguiente figura muestra estos cuatro pasos:



Una vez el cliente ha conseguido su dirección IP, puede dejar de necesitarla en cualquier momento. Para finalizar el arrendamiento de la dirección IP antes del tiempo asignado, el cliente deberá mandar un mensaje **DHCPRELEASE** al servidor. A partir de ese momento el cliente ya no podrá usar esa dirección IP y el servidor DHCP podrá asignarla a cualquier otro nodo que lo solicite.

Por el contrario, si un nodo agota el tiempo de arrendamiento que se le ha concedido en la asignación de la dirección IP y desea seguir utilizándola puede renovar su tiempo de arrendamiento mediante un mensaje DHCPREQUEST.

El servidor podrá contestar afirmativamente mediante un DHCPACK, o bien, denegar la prórroga de tiempo mediante un mensaje DHCPNACK (mensaje DHCP de reconocimiento negativo). En este último caso, el cliente abandonará la dirección IP inmediatamente.

Agente retransmisor DHCP

Como hemos visto, varios de los mensajes DHCP se envían por difusión, mediante la dirección destino 255.255.255.255. Las difusiones realizadas de esta forma son filtradas por los routers, y por tanto, no pueden alcanzar a destinos externos a la red local donde se haya originado la difusión. Esto supone que, en principio, se requeriría un servidor DHCP en cada red que deseara utilizar el servicio. Para evitar este requerimiento pueden emplearse agentes retransmisores DHCP (*DHCP relay agent*). Son dispositivos que reciben las solicitudes de los clientes enviadas como difusiones y las reenvían en modo unicast a la dirección del servidor DHCP. Se puede configurar como agente retransmisor el router de salida de la red, o bien un host que esté en la misma red que el cliente DHCP.

Formato del mensaje DHCP

Tanto los mensajes DHCP de petición como los de respuesta tienen el mismo formato que se muestra a continuación:

0	8	16	24	31
OP	HTYPE	HLEN	HOPS	
ID DE TRANSACCION				
SEGUNDOS		BANDERAS		
DIRECCION IP DEL CLIENTE				
TU (CLIENTE) DIRECCION IP				
DIRECCION IP DEL SERVIDOR				
DIRECCION IP DEL ROUTER				
DIRECCION DE HARDWARE DE CLIENTE (16 OCTETOS)				
.				
NOMBRE DE SERVIDOR (64 OCTETOS)				
.				
NOMBRE DEL ARCHIVO DE ARRANQUE (128 OCTETOS)				
.				
OPCIONES (VARIABLE)				
.				

Veamos el significado de cada uno de los campos:

- **OP:** (1) Solicitud (2) Réplica.
- **HTYPE:** Tipo de Hardware de Red. Ej: (1) Ethernet.
- **HLEN:** Longitud de la dirección Hardware. Ej: Ethernet (6).
- **HOPS:** Número de saltos. El cliente lo pone a cero. Si el servidor recibe la solicitud y decide pasarla a otra máquina lo incrementa.
- **ID. DE TRANSACCIÓN:** entero que la máquina utiliza para emparejar las respuestas con las solicitudes.
- **SEGUNDOS:** el cliente apunta el número de segundos desde que comenzó el arranque.
- **BANDERAS:** sólo el bit de orden superior tiene asignado significado. El resto se pone a cero. Un 1 en el bit de orden superior indica que el servidor debe responder con la difusión IP, lo que implicará también una difusión hardware. Con un 0 se solicita una respuesta unicast.
- **DIR. IP DEL CLIENTE:** Si el cliente ya tiene asignada una dirección IP utilizará este campo para ponerla. En caso contrario, pondrá a cero este campo.
- **TU (CLIENTE) DIRECCIÓN IP:** cuando el cliente todavía no tiene dirección asignada, se utiliza este campo para indicar la dirección IP que se le ofrece.
- **DIR. IP DEL SIGUIENTE SERVIDOR:** lo utiliza el servidor en sus respuestas al cliente (DHCP OFFER y DHCP ACK) para indicarle la dirección del siguiente servidor DHCP que debe usar en el proceso de arranque.
- **DIR. IP DEL AGENTE RETRANSMISOR:** si se quiere especificar.
- **NOMBRE DEL SERVIDOR:** funciona igual que la dirección IP del servidor.
- **NOMBRE DEL FICHERO DE ARRANQUE:** un administrador puede querer tener varios tipos de arranque (Ej. UNIX, WINDOWS, etc.). En ese caso, puede especificarlo en este campo.
- **OPCIONES:** con este campo DHCP puede codificar una gran variedad de cosas diferentes: duración del arrendamiento, tipo de mensaje, máscara de red, ... Cada opción está a su vez formada por tres campos:
 1. **Código:** indica el tipo de opción. Ocupa un byte.
 2. **Longitud:** indica el número de bytes del campo datos. Ocupa un byte
 3. **Datos:** la información concreta que atañe a esa opción. Tiene longitud variable.

Ejemplo de opción: Una opción que aparece en todos los mensajes DHCP es la número 53, que indica el tipo de mensaje DHCP. Se trata de una opción de 3 octetos que tienen el siguiente significado y valor:

CODIGO (53)	LONGITUD (1)	TIPO (1-7)
-------------	--------------	------------

Donde el tipo de mensaje puede ser:

TIPO	MENSAJE
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNACK
7	DHCPRELEASE

El mensaje DHCPDECLINE lo utiliza el cliente para comunicar al servidor que la dirección IP ofrecida ya está en uso.

4. Análisis de tráfico

En esta práctica vamos a utilizar el analizador de protocolos que ya conocemos, *Wireshark*, para analizar el tráfico que generan un cliente y un servidor DHCP, tanto cuando el cliente solicita dirección IP al servidor como cuando la libera.

Pero antes de ponernos a trabajar con el analizador de protocolos, es interesante recordar de la práctica 1 cómo averigua mi ordenador el tipo de configuración IP, es decir, cómo obtiene la dirección IP. Veámos en la práctica 1 que nuestros equipos de trabajo obtienen la configuración IP de forma automática, es decir, mediante el diálogo con un servidor DHCP. Ese servidor DHCP puede estar dentro de mi red (suele ser así en nuestros ordenadores domésticos), o bien, estar fuera de la red como es el caso de los ordenadores del Laboratorio de Redes. En este último caso, en ese diálogo entre nuestra máquina (cliente DHCP) y el servidor DHCP que le proporciona la información IP intervendrán agentes retransmisores.

Ejercicio 1. Abre una ventana de DOS y ejecuta la orden ***ipconfig /all***. De toda la información que se te muestra, ¿qué parámetros están relacionados con el diálogo DHCP inicial que se ha producido en el arranque del sistema? ¿Cuándo se ha obtenido la concesión de la dirección IP? ¿Cuándo caduca?

(Recuerda que el comando equivalente en Linux sería: ***ip address***)

La particularidad que tiene el protocolo DHCP es que el tráfico DHCP se genera fundamentalmente en el arranque del equipo. Una vez el nodo está listo para poder tomar capturas con el Wireshark, el diálogo DHCP ya ha finalizado. Puede aparecer un nuevo diálogo DHCP cuando se cumpla el tiempo de concesión de la dirección IP y se quiera renovar, pero no es cuestión de esperar hasta entonces.

Podríamos utilizar una orden que ya estudiasteis en la práctica 1. Esta nos permite liberar la dirección IP que le han asignado a la máquina en el arranque, y volver a pedir después una nueva asignación mediante un diálogo DHCP.

Esta orden es el ***ipconfig***, de la que destacamos 3 opciones solamente:

<i>ipconfig /all</i>	Muestra toda la información de configuración
<i>ipconfig /release</i>	Libera la dirección IPv4

ipconfig /renew Renueva la dirección IPv4

Ejercicio 2. Ponemos en marcha el analizador de protocolos Wireshark e iniciamos una captura filtrando el tráfico UDP que utilice el puerto 67. A continuación, utilizamos el comando ***ipconfig /release*** para liberar la información IP que tiene nuestro ordenador y después el comando ***ipconfig /renew*** para volver a obtener dirección IP, mientras está en marcha la captura con el Wireshark. Una vez finalizado este proceso paramos la captura.

Los comandos equivalentes en **Linux** serían:

- ***sudo dhclient -r*** para liberar la dirección IP asociada a nuestro ordenador.
- ***sudo dhclient*** para solicitar nueva dirección IP

Con este proceso hemos obtenido todo el diálogo DHCP que realizan un cliente y servidor DHCP para conseguir una dirección IP. Este diálogo, en nuestro caso, estará precedido por el diálogo mediante el cual el cliente ha renunciado a la configuración IP que había obtenido en el arranque del sistema. Observa si los mensajes obtenidos en la captura se corresponden con los que se han explicado en apartados anteriores.

Con el objetivo de que trabajemos todos con las mismas capturas hemos dejado en Poliformat dos capturas: **Captura1Practica3.pcap** que recoge el proceso de obtención de configuración IP y **Captura2Practica3.pcap** que recoge el proceso de liberación de la configuración IP. Estas capturas han sido realizadas en un ordenador del Laboratorio de Redes y, por tanto, nos van a permitir conocer un poco sobre la configuración IP de dicha red.

Ambas están listas para su análisis y podemos trabajar con ellas a partir ahora.

Ejercicio 3. Descárgate del Poliformat la captura **Captura1Practica3.pcap** y ábrela con el programa Wireshark.

- a) Nos centramos en primer lugar en el primer mensaje DHCP que interviene en el proceso de obtención de dirección IP: DHCPDISCOVER. Basándote en la información obtenida, ¿qué servicio utiliza DHCP, TCP o UDP? Mirando las direcciones IP origen y destino del datagrama de este primer mensaje DHCP que te aparece, ¿podrías justificar la elección de DHCP por un servicio sin conexión?
- b) Selecciona el **mensaje DHCPDiscover** y **b u s c a** la información para rellenar los siguientes campos (se trata de información que corresponde a distintos niveles de la arquitectura y está en diferentes cabeceras):

<i>Tipo de Mensaje (en el campo de opciones DHCP)</i>	
<i>Dir. IP origen del datagrama (en cab. IP)</i>	
<i>Dir. IP destino del datagrama (en cab. IP)</i>	
<i>Puertos origen y destino (en cab. UDP)</i>	
<i>Id. Transacción (en cabecera DHCP)</i>	
<i>Campo dir. IP Cliente (en cabecera DHCP)</i>	
<i>Campo Tu dirección IP (en cabecera DHCP)</i>	

<i>Campo Dir. IP del Agente Retransmisor (cab. DHCP)</i>	
---	--

c) En este mensaje el nodo no solicita una dirección IP cualquiera, sino que pide una dirección concreta que está asociada a su dirección física. ¿En qué campo del mensaje se hace esta solicitud?

d) Entre las distintas opciones aparece la lista de parámetros que el cliente solicita al servidor. Cita las cuatro primeras.

Ejercicio 4. A continuación podemos ver los mensajes de ofrecimiento de los servidores: DHCPOFFER.

a) ¿Cuántos mensajes de este tipo hay? ¿Qué conclusiones podemos sacar acerca del número de servidores DHCP disponibles en la red de la UPV?

b) Busca en el primer mensaje DHCP Offer la información para rellenar los siguientes campos:

<i>Tipo de Mensaje (en el campo de opciones DHCP)</i>	
<i>Dir. IP origen del datagrama (en cab. IP)</i>	
<i>Dir. IP destino del datagrama (en cab. IP)</i>	
<i>Puertos origen y destino (en cab. UDP)</i>	
<i>Id. Transacción (en cabecera DHCP)</i>	
<i>Campo dir. IP Cliente (en cabecera DHCP)</i>	
<i>Campo Tu dirección IP (en cabecera DHCP)</i>	
<i>Campo Dir. IP del Agente Retransmisor (cab. DHCP)</i>	

c) ¿En qué campo de los mensajes DHCP Offer aparece la mayor parte de la información de configuración IP que el Servidor ofrece al Cliente? Comprueba todo lo que ofrecen los servidores DHCP. En particular, fíjate en el valor del campo “**DHCP Server Identifier**”. ¿Coincide con alguno de los que has anotado en la tabla?

d) Busca y anota en cualquiera de los mensajes DHCP Offer la siguiente información IP que los servidores DHCP están ofreciendo al equipo que ha solicitado la configuración IP: dirección IP ofrecida, máscara de subred, router asignado y nombre de dominio .

e) A quién pertenece la dirección IP origen de estos datagramas (DHCPOFFER). ¿Qué conclusiones puedes sacar acerca de la localización de los servidores DHCP de la UPV con respecto a la subred en la que está el cliente?

f) Compara los valores del campo “DHCP Server Identifier” de los mensajes DHCP Offer restantes que aparecen en la captura. ¿Cuántos servidores DHCP distintos están contestando?

Ejercicio 5. Analizamos ahora el mensaje DHCPREQUEST con el que contesta el cliente a uno de los servidores que le ha realizado una oferta.

a) Completa la siguiente tabla con la información sobre este mensaje:

<i>Tipo de Mensaje (en el campo de opciones DHCP)</i>	
<i>Dir. IP origen del datagrama (en cab. IP)</i>	
<i>Dir. IP destino del datagrama (en cab. IP)</i>	
<i>Puertos origen y destino (en cab. UDP)</i>	
<i>Id. Transacción (en cabecera DHCP)</i>	
<i>Campo dir. IP Cliente (en cabecera DHCP)</i>	
<i>Campo Tu dirección IP (en cabecera DHCP)</i>	
<i>Campo Dir. IP del Agente Retransmisor (cab. DHCP)</i>	

b) Busca entre las distintas opciones del mensaje la dirección IP del servidor DHCP al que el cliente está contestando.

Ejercicio 6. Por último, tenemos los mensajes DHCPACK de los servidores de la UPV que confirman la obtención de la dirección IP por parte del cliente. Busca en estos mensajes las direcciones IP de los diferentes servidores DHCP de la UPV.

Ejercicio 7. Y para terminar, vamos a analizar una nueva captura con el Wireshark con el fin de estudiar el tráfico DHCP que se genera cuando un nodo libera su dirección IP. Para ello abrimos la captura **Captura2Practica3.pcap** (la hemos obtenido ejecutando el comando: *ipconfig /release*).

¿Qué tipo de mensaje DHCP interviene en este proceso? ¿Quién es el origen y el destino de este mensaje? ¿Hay contestación a este mensaje?