

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


1. Introducción

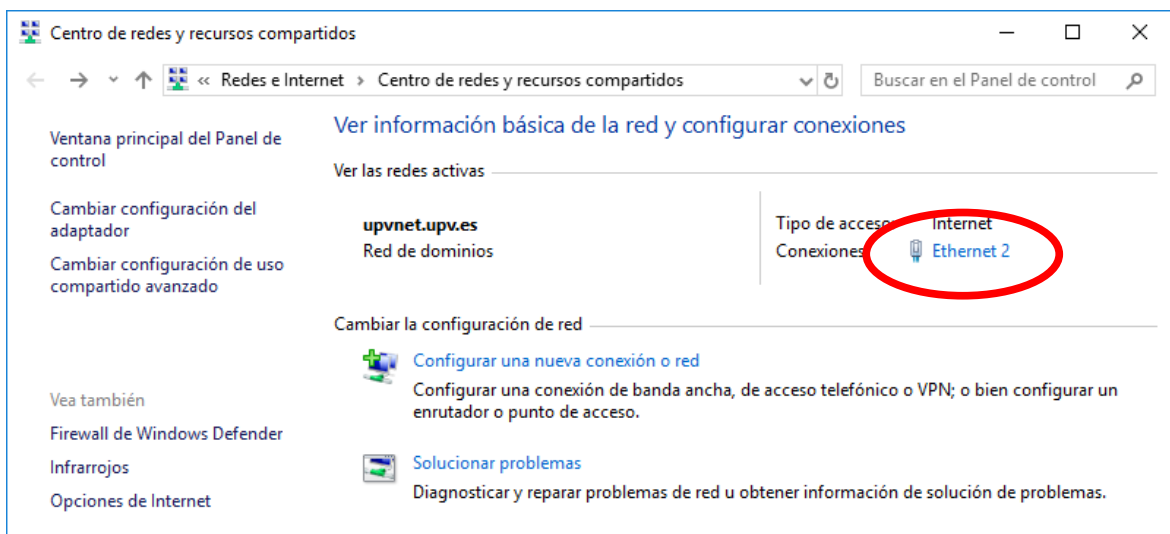
Esta práctica está dedicada a revisar el procedimiento básico de instalación y configuración de los protocolos TCP/IP en Windows 10 y Linux. Se muestra el uso de algunas herramientas útiles a la hora de resolver problemas con estos protocolos: configurar el software de red, verificar su funcionamiento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

2. Configuración de TCP/IP en Windows 10

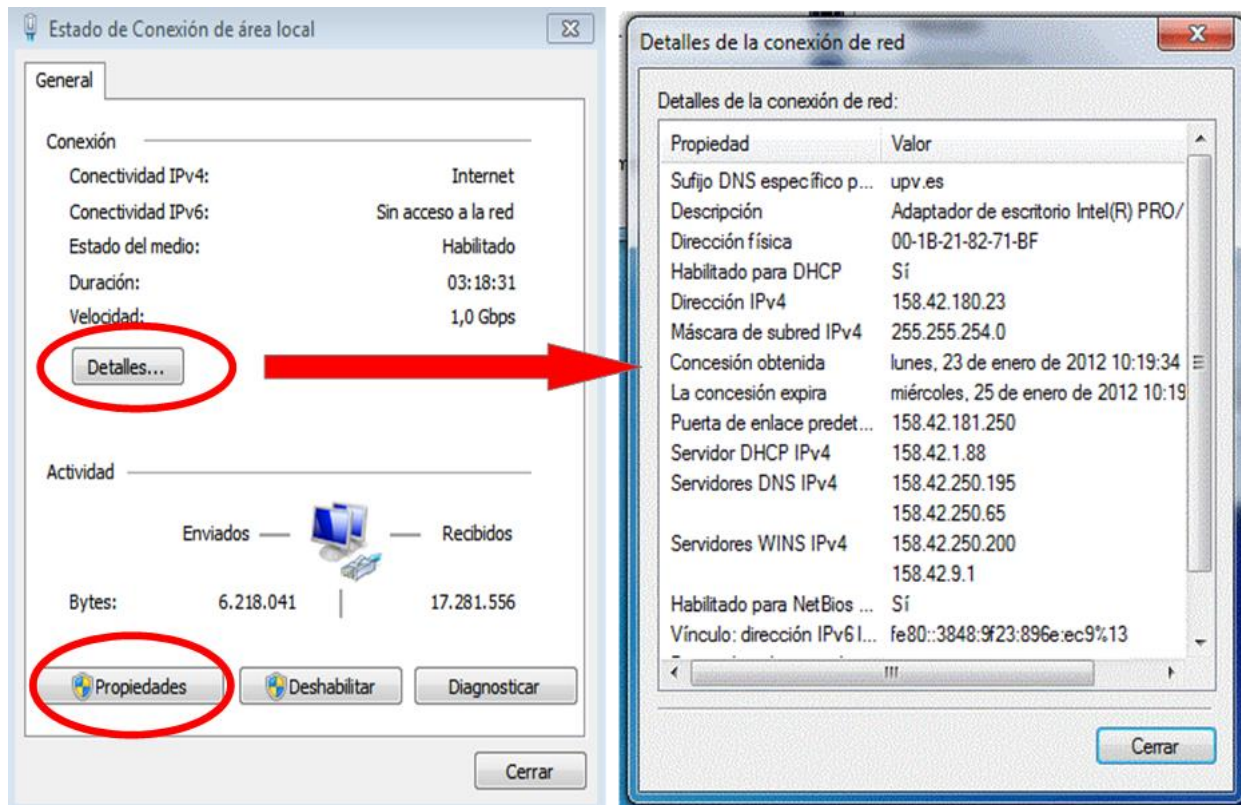
Para poder realizar este apartado debes arrancar tu ordenador en la partición de Windows 10, y utilizar para el acceso el nombre de usuario y la contraseña de tu cuenta de la UPV (dominio ALUMNO).

Para utilizar los protocolos TCP/IP desde una máquina Windows 10 conectada a una red de área local (Ethernet en nuestro caso) es necesario tener instalada una tarjeta adaptadora o NIC (*Network Interface Adapter*). En nuestros computadores esta tarjeta ya está instalada y configurada. 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 de prácticas, directamente conectados a la red local de la UPV, en **conexiones** aparece el adaptador Ethernet que se está empleando. Haz click sobre el enlace **Ethernet 2**



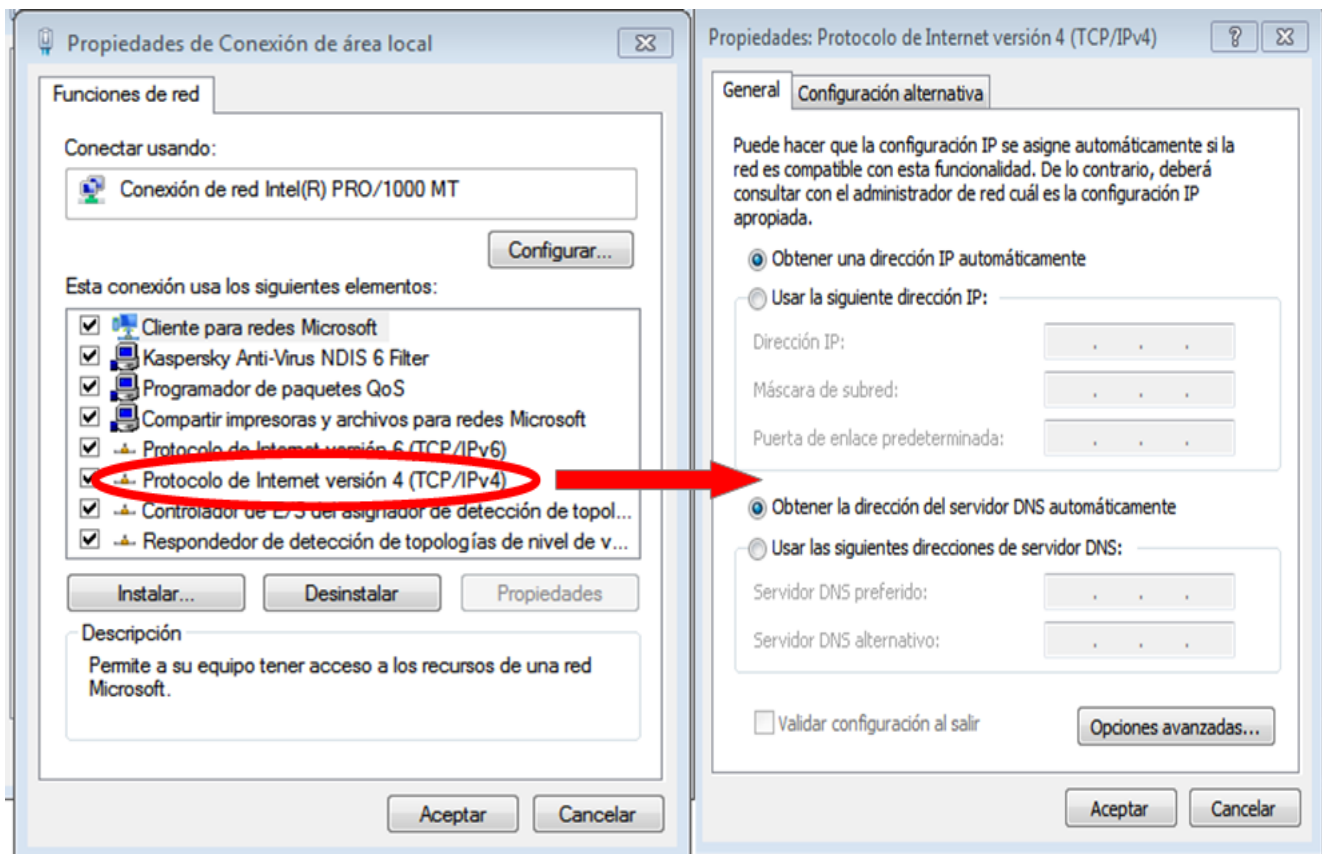
- Se abrirá la ventana de **Estado** del adaptador Ethernet. En ella, al pulsar el botón **Detalles**, podemos ver las principales propiedades de la conexión de red asociada al adaptador: dirección física del adaptador de red, dirección IP asociada a ese interfaz, máscara de red, etc.



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 (**tú no lo eres**) podría comprobarlo siguiendo los pasos siguientes. Nosotros lo observaremos 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.).



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).

El formato de la orden es el siguiente:

W:\>**ipconfig /?**

```
USO: ipconfig [/allcompartments] [/? | /all |
                                     /renew [adaptador] | /release [adaptador] |
                                     /renew6 [adaptador] | /release6 [adaptador] |
                                     /flushdns | /displaydns | /registerdns |
                                     /showclassid adaptador |
                                     /setclassid adaptador [id._clase] |
                                     /showclassid6 adaptador |
                                     /setclassid6 adaptador [id._clase] ]
```

donde:

adaptador Nombre de conexión (se permiten los caracteres comodín * y ?; consulte los ejemplos)

Opciones

/? Muestra este mensaje de ayuda.
/all Muestra toda la información de configuración.
/release Libera la dirección IPv4 para el adaptador especificado.
/release6 Libera la dirección IPv6 para el adaptador especificado.
/renew Renueva la dirección IPv4 para el adaptador especificado.

```
/renew6      Renueva la dirección IPv6 para el adaptador especificado.  
/flushdns    Purga la caché de resolución de DNS.  
/registerdns  Actualiza todas las concesiones DHCP y vuelve a  
              registrar los nombres DNS  
/displaydns  Muestra el contenido de la caché de resolución de DNS.  
/showclassid Muestra todos los Id. de clase DHCP permitidos para el  
              adaptador.
```

Tecleando simplemente **ipconfig** obtenemos 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

Identifica cuál de todos los adaptadores que aparecen es el que te conecta a Internet e indica el motivo.

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 comience 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 que empiece por "158.42"

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 el computador de prácticas?
¿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. El formato de la orden, que puedes ver tecleando `netstat -h`, es el 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:

a) Un paquete destinado a `zoltar.redes.upv.es`

b) Un paquete destinado a `www.upv.es`

c) 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 Ethernet. 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 en la partición “Prácticas DCLAN/RedLocal”.

Tu profesor te indicará la contraseña necesaria.

En Linux encontramos las mismas órdenes que acabamos de estudiar dentro del entorno Windows 10, en particular:

- Orden **tracert**, equivale a la orden **tracert** de Windows
- Orden **ping**, equivalente al **ping** de Windows. Por omisión, realiza *ping* de forma continua hasta que el usuario lo detiene con ctrl+C.
- Orden **arp**, equivale a la orden de Windows del mismo nombre.
- Orden **netstat**, equivalente a la que hemos estudiado para Windows.

Por ello, aquí sólo revisaremos algunas diferencias significativas entre los dos sistemas operativos. Puedes encontrar más información empleando `man <orden>`.

3.1 La orden ifconfig

La orden **ifconfig**, que puedes ejecutar **desde un terminal de red**, permite configurar y obtener información sobre la configuración de red. Utilizando esta orden con las opciones adecuadas podemos configurar todo el software de TCP/IP. En nuestro caso el software ya está instalado y, además, no tenemos permisos de administración para modificar los parámetros. Así que nos limitaremos a inspeccionar la información mediante **ifconfig**.

Si ejecutamos **ifconfig** seguido del nombre de una interfaz obtendremos información sobre la configuración de ésta. Si se ejecuta sin parámetros, presenta las características de todas las interfaces que se hayan configurado. A modo de ejemplo, la consulta de la configuración de la interfaz Ethernet **eth0**¹ sería:

```
redlocal@rdc01:~$ ifconfig eth0
eth0      Link encap:Ethernet  direcciónHW 10:c3:7b:94:e7:b4
          Direc. inet:158.42.180.1  Difus.:158.42.181.255  Másc:255.255.254.0
          Dirección inet6: fe80::12c3:7bff:fe94:e7b4/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:5114 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:776 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colatx:1000
          Bytes RX:5538624 (5.5 MB)  TX bytes:79082 (79.0 KB)
```

Los campos **MTU** y **Métrica** informan sobre los valores actuales de la MTU (Unidad Máxima de Transferencia) y de la métrica para una interfaz dada. Algunos sistemas operativos usan el valor **Métrica** para calcular el coste de una ruta. Linux no usa este valor por el momento, pero lo define por razones de compatibilidad.

Las líneas **Paquetes RX** y **Paquetes TX** dan información sobre el número de paquetes recibidos o transmitidos sin errores, el número de errores ocurridos, de cuántos paquetes han sido descartados, seguramente por memoria insuficiente, y cuántos se han perdido por desbordamiento, condición que ocurre cuando la recepción de paquetes es demasiado rápida y el sistema operativo es incapaz de dar servicio al paquete anterior antes de la llegada del nuevo paquete.

A continuación se muestra una lista de algunos parámetros reconocidos por **ifconfig** (puedes visualizar todas las opciones disponibles mediante la orden **man ifconfig**). Las opciones que simplemente activan alguna característica pueden usarse para desactivarla anteponiendo un guion (-). La utilización de estos parámetros permite a los usuarios administradores (**no es nuestro caso**) cambiar la configuración de la interfaz.

- **up**: marca la interfaz como "up" o activa, es decir, disponible para que sea usada por el nivel IP. (Esta opción corresponde a los indicadores UP RUNNING).
- **down**: marca la interfaz como "down" o inactiva, es decir, inaccesible al nivel IP.
- **netmask máscara**: asigna una máscara de subred a una interfaz.
- **metric número**: puede ser usada para asignar un valor de métrica a la tabla de encaminamiento creada para la interfaz.
- **mtu bytes**: fija la unidad máxima de transferencia, o lo que es lo mismo, el máximo número de octetos que la interfaz es capaz de manejar en una única transacción. Para Ethernet, la MTU toma el valor 1500 por defecto.
- **arp**: esta opción es específica de redes de difusión como Ethernet. Permite el uso de ARP, el Protocolo de Resolución de Direcciones, para detectar la dirección física de las máquinas conectadas a la red. Para redes de difusión, esta opción es habilitada por defecto. **ifconfig** avisa que ARP ha sido inhabilitado mediante el indicador NOARP. **-arp** inhabilita el uso de ARP para esta interfaz.

¹Puede ocurrir que en algunos puestos del laboratorio la información similar a la que se muestra corresponda a la interfaz 1 (eth1) en lugar de a la 0 (eth0).

- **promisc:** pone la interfaz en modo promiscuo. En una red de difusión, esto hace que la interfaz reciba todos los paquetes, independientemente de si están dirigidos a ella o no. Esto permite el análisis del tráfico de red mediante utilidades como filtros de paquetes. Se trata de una buena técnica para localizar problemas de red que de otra forma resultan difíciles. Por otro lado, esto también posibilita ataques, permitiendo al atacante analizar el tráfico de la red en busca de claves u otras cosas peligrosas. (Esta opción corresponde al indicador PROMISC). **-promisc** desactiva el modo promiscuo.

Ejercicio 8:

Ejecuta la orden **ifconfig eth0²** y, basándote en la descripción anterior, analiza la información obtenida.

3.2 La orden netstat

La orden **netstat** tiene un funcionamiento muy similar al descrito para Windows 10. Destacaremos únicamente la opción que permite consultar la tabla de encaminamiento para ver las diferencias de formato de ambas tablas.

Si se ejecuta **netstat** usando el indicador **-nr**, se puede ver la información de la tabla de encaminamiento del núcleo (produce una salida idéntica a la orden **route**). Por ejemplo:

```
redlocal@rdc01:~$ netstat -nr
Tabla de rutas IP del núcleo
Destino          Pasarela          Genmask           Indic   MSS  Ventana  irtt  Interfaz
0.0.0.0          158.42.181.250    0.0.0.0           UG      0 0           0 eth0
158.42.180.0     0.0.0.0           255.255.254.0     U       0 0           0 eth0
169.254.0.0      0.0.0.0           255.255.0.0       U       0 0           0 eth0
```

La segunda columna de la salida producida por **netstat** informa sobre los routers a los que apunta la información de encaminamiento. Si una ruta no usa router, el programa indica la dirección 0.0.0.0. La tercera columna muestra la máscara de red, es decir, cuántos bits de la dirección destino corresponden a la parte de red. A la hora de encaminar un paquete IP, el núcleo extrae la dirección del destino y recorre la tabla registro a registro aplicando un AND lógico de la dirección IP destino y la máscara (**Genmask**) antes de compararla con el destino que corresponde con dicho registro (**Destino**). La ruta por defecto tiene máscara 0.0.0.0. Al igual que ocurre en Windows, de todas las rutas en las que se produzca coincidencia se selecciona la que tenga más bits a uno en la máscara.

La cuarta columna muestra varios indicadores que describen la ruta:

- G La ruta utiliza un router.
- U La interfaz está activa.
- H Esta interfaz permite el acceso a una sola máquina.

Las columnas **MSS**, **Ventana** e **irrt** indican los valores iniciales de algunos parámetros que se utilizan en las conexiones TCP que se establecen a través de esta ruta. La columna **Interfaz** indica a través de qué interfaz se accede a estas rutas. En nuestro caso solo estamos interesados en las rutas que se establecen a través de la interfaz Ethernet (**eth0**).

²Recuerda que en tu puesto podría ser la eth1.

Ejercicio 9:

Utiliza la orden **netstat -nr** y rellena la tabla para las rutas relacionadas con la interfaz activa:

Destino	Pasarela (Router)	Máscara de red

Analiza qué ruta de la tabla se seleccionaría para:

- Un paquete destinado a **www.upv.es**.
- Un paquete destinado a **zoltar.redes.upv.es**.

Interpreta los resultados teniendo en cuenta el campo **Pasarela** (router de salida de la red o entrega directa de origen a destino).

3.3 La orden route

Esta orden no solo permite ver el contenido de las tablas de encaminamiento, sino también modificarlo, añadiendo o eliminando entradas de la misma. Para esto último se requiere permisos de administrador. En nuestro caso, para obtener estos permisos, emplearemos las órdenes anteponiendo el **sudo**.

La opción **-n** solicita que no se traduzcan las direcciones IP a nombres de dominio, y las opciones **add** y **del** permiten añadir o eliminar entradas, respectivamente. Puedes ver una ayuda detallada consultando el manual (`man route`).

```
redlocal@rdc01:~$ route -n
Tabla de rutas IP del núcleo
Destino          Pasarela          Genmask           Indic Métric Ref       Uso Interfaz
0.0.0.0          158.42.181.250    0.0.0.0           UG      0         0         0 eth0
158.42.180.0     0.0.0.0           255.255.254.0     U       0         0         0 eth0
169.254.0.0      0.0.0.0           255.255.0.0       U      1000      0         0 eth1
172.16.240.0     0.0.0.0           255.255.255.0     U       0         0         0 vmnet1
192.168.0.0      0.0.0.0           255.255.255.0     U       0         0         0 eth1
192.168.63.0     0.0.0.0           255.255.255.0     U       0         0         0 vmnet8
```

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 dos entradas de la tabla: la entrada por defecto, que nos permite alcanzar el resto de Internet y la entrega directa, que indica cómo enviar paquetes a destinos que están en la misma red que nuestro computador.

Ejercicio 10:

- Visualiza la tabla de encaminamiento de tu ordenador (orden **route -n**).
- 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 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 destino (que anotaste en el ejercicio 4b) en vez del nombre del servidor.
- 6) Intenta acceder a un destino que esté en la misma red IP que tu ordenador. Por ejemplo, mediante la orden **ping -c 2 158.42.180.62**, (es la IP de **zoltar.redes.upv.es**), o haciendo ping al ordenador de algún compañero (**158.42.180.<puesto>**).
- 7) Restaura la línea de la tabla de encaminamiento que habías eliminado (**sudo route add default gw dir_IP_de_tu_router**). Al ejecutar la orden puede que el sistema dé un mensaje de aviso porque no tiene acceso al DNS, pero no tiene mayor importancia.
- 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.

Ejercicio 11:

Lo ideal para comprobar cómo afecta al encaminamiento la entrada local sería eliminar de la tabla la entrada que corresponde a la red IP de tu equipo, pero no está permitido. Así es que vamos a emplear un truco para conseguir algo similar, la mantendremos en la tabla, pero impediremos su uso.

- 1) Ejecuta la orden **sudo route add -net 158.42.180.0 netmask 255.255.254.0 reject**.
- 2) Visualiza el estado de la tabla de encaminamiento (**route -n**)
- 3) Intenta acceder a un destino de tu red IP. Por ejemplo, mediante la orden **ping -c 2 zoltar.redes.upv.es**.
- 4) Intenta acceder a destinos fuera de tu red IP. Por ejemplo,
ping -c 2 www.upv.es
ping -c 2 www.google.es.
- 5) Comprueba lo que has observado y observa a qué se debe.
- 6) Restaura el estado original de la tabla de encaminamiento:
sudo route del -net 158.42.180.0 netmask 255.255.254.0 reject
- 7) Comprueba el estado de la tabla de encaminamiento. Debe ser el que había antes de eliminar la ruta.

Práctica 2: Fragmentación y reensamblado en IP

Para esta práctica debes arrancar en la partición de linux "Ubuntu".
Utiliza para el acceso el nombre de usuario y la contraseña de tu cuenta de la UPV

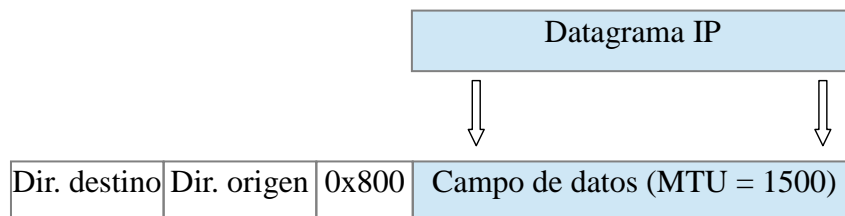
Lectura previa: Kurose 4.4.1 subapartado "Fragmentación del datagrama IP"

1. Introducción

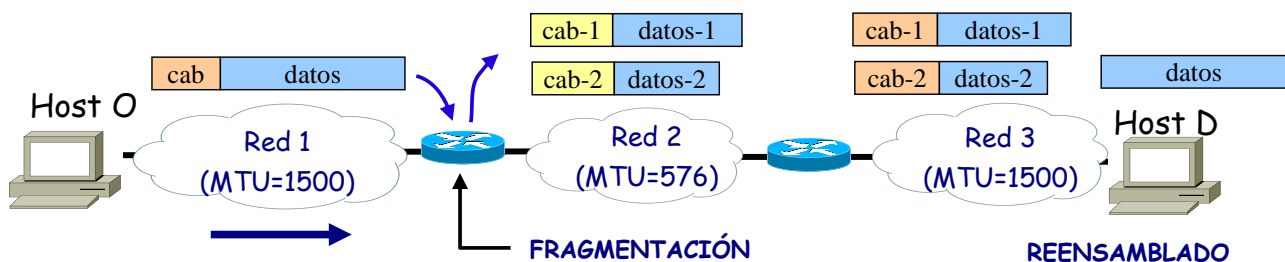
En esta práctica vamos a estudiar el problema de la fragmentación de datagramas IPv4.

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 y en la práctica suelen enviarse datagramas más pequeños.

Antes de transmitirse, el datagrama se encapsula 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 va a llevar. 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 inicial 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 el original. El router que separa las dos redes se encargará de esta tarea, antes de reenviar el datagrama a la red de salida. Posteriormente, el host destino tendrá que reensamblar el datagrama original una vez recibidos todos los fragmentos.

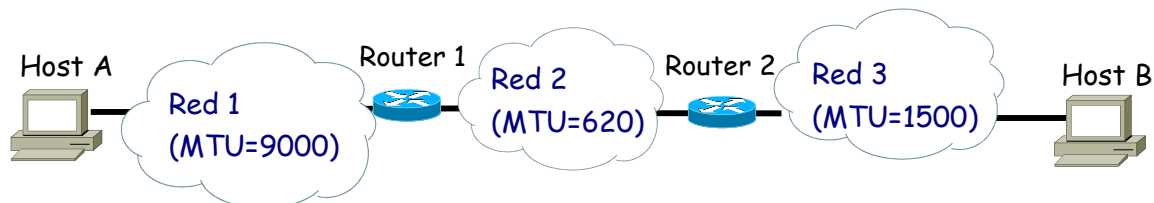


Puede ser necesario volver a fragmentar un datagrama ya fragmentado. En ese caso, todos los fragmentos tienen el mismo nivel y el desplazamiento se refiere al datagrama original.

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	MTU = 620 octetos	
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		

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

- Que la cabecera IP ocupa 20 bytes, si como es habitual no lleva opciones. 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.
- 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, que será múltiplo de 600 en los diferentes fragmentos, realmente aparecerá en la cabecera IP de los fragmentos

expresado en múltiplos de 75 dado que el campo desplazamiento tiene 13 bits de longitud y esto obliga a expresar los desplazamientos en múltiplos de 8 bytes.

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 en el laboratorio la fragmentación que se produce en los routers, pero podemos utilizar un pequeño truco para generar fragmentación en nuestro propio equipo de prácticas.

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. Muchos sistemas operativos como Linux, Microsoft Windows o MAC OS nos proporcionan una orden llamada “ping” que nos permite enviar a un destino paquetes ICMP 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 158.42.180.xx*

Para tu ordenador de prácticas debes sustituir xx por el número de host indicado en la etiqueta que tiene cada puesto de trabajo.

A continuación, abre un shell y teclea:

```
> ping -c 1 -s 3972 zoltar.redes.upv.es
```

Y una vez terminado el ping detén la captura de paquetes del wireshark.

La opción -c es para que se envíe un único paquete, ya que, por defecto, como se verá en la práctica donde se estudia con detalle el protocolo ICMP, la orden ping en Linux envía paquetes de forma ininterrumpida. La opción -s indica el tamaño del campo de datos ICMP (el paquete ICMP también tendrá una cabecera).

Como estamos conectados a una red Ethernet, 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 3: DHCP, funcionamiento y análisis de trazas.

1. Sesión L3

Lectura previa: Kurose 4.4.2, subapartado “Cómo obtener una dirección de host: Protocolo de configuración dinámica de host” (pags. 336-339).

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** porque, a diferencia de lo que ocurre en Linux, permite ejecutar el cliente DHCP sin necesidad de ser administrador del sistema.

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.

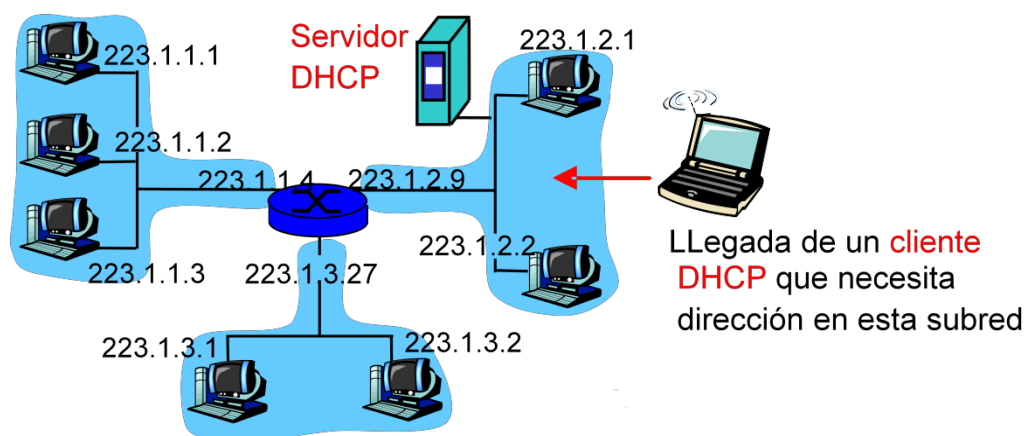
El protocolo DHCP permite realizar esta asignación automática de direcciones IP. Además de la asignación de direcciones IP a los nodos, DHCP también permite que un nodo obtenga información adicional necesaria para el funcionamiento en Internet:

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. **Etapas 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. **Etapas 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).

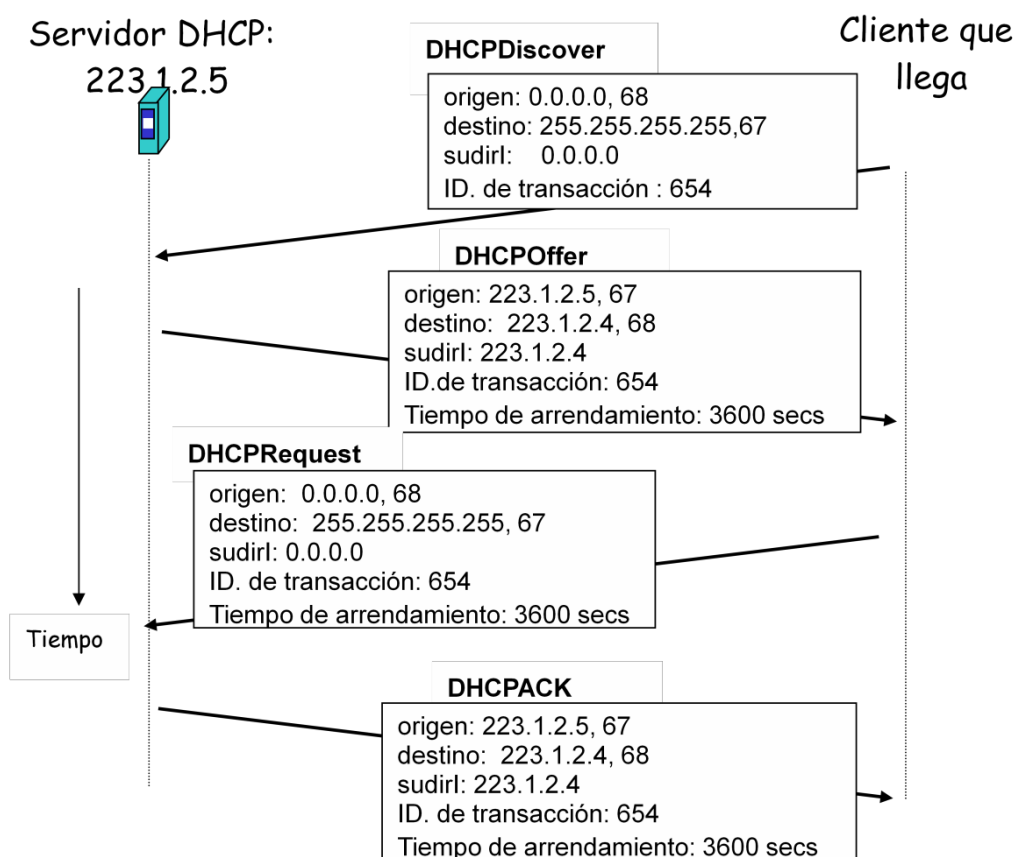
Por tanto, un cliente podrá recibir cero o más respuestas. Cada mensaje DHCPOFFER contendrá: el Identificador de Transacción que llevaba el mensaje DHCPDISCOVER, la dirección IP que el servidor ofrece al cliente, la máscara de red correspondiente y el tiempo de concesión de la dirección IP ofrecida: 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 averiguan los ordenadores del Laboratorio de Redes el tipo de configuración IP, es decir, cómo obtienen 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. Recordad que ese servidor DHCP estaba fuera de la red del Laboratorio de Redes. Por lo tanto, 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?

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
<i>ipconfig /renew</i>	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.

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.

Vamos a analizar los resultados obtenidos en la captura.

Por si acaso el proceso de obtención de mensajes DHCP nos da problemas, y 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. 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 DHCPOffer 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 DHCPOffer 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) Con la ayuda del comando `ipconfig /all` averigua 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 tu equipo?

e) Compara los valores del campo “DHCP Server Identifier” de los mensajes DHCPOffer 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?

Práctica 4: El protocolo ICMP

1. Sesión L4

Lectura previa: Kurose 4.4.3 subapartado “Protocolo de mensajes de control ICMP”

En la sesión de laboratorio debes arrancar el ordenador en la partición “**Ubuntu**” e iniciar sesión con tu usuario y contraseña de upvnet.

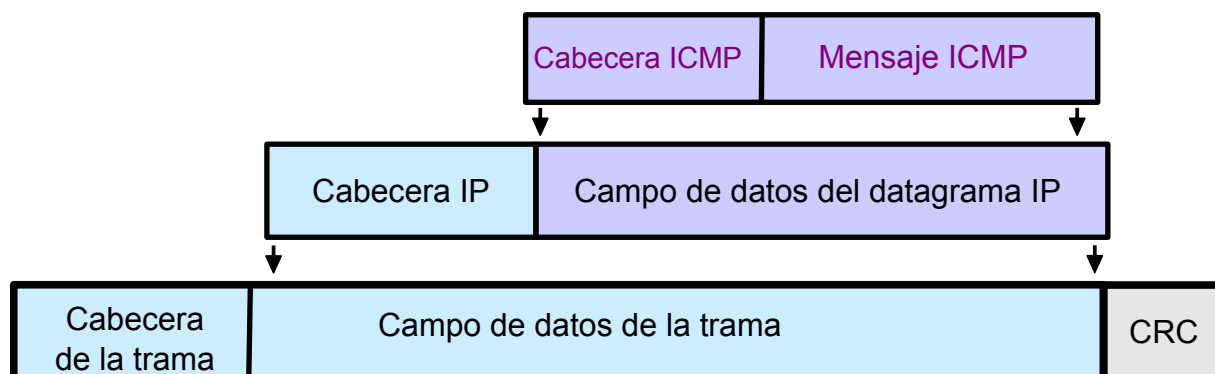
2. Introducción a ICMP

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.

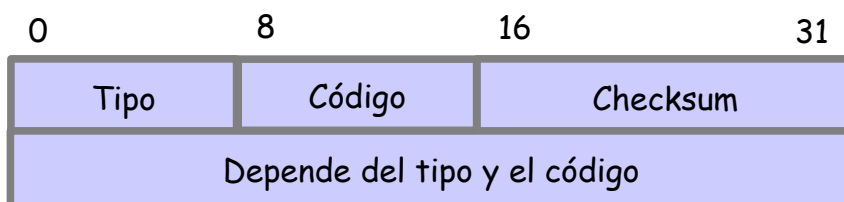
ICMP nos permite saber, por ejemplo, por qué no se ha entregado un datagrama (no hay ruta, el destino no responde, se ha agotado su tiempo de vida, etc.). Informa de errores sólo al origen del datagrama. Además, no se encarga de corregir el problema, sólo de avisar.

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* (desde el botón de la barra superior). Si te solicita contraseña teclea la de tu usuario de upvnet. 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). No olvides comprobar que la interfaz de captura es la adecuada (debe coincidir con la que tiene asociada la dirección IP pública). 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 varía 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. Para ver una información más completa puede consultarse el man de la orden.

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 -c 3` a las direcciones siguientes: `zoltar.redes.upv.es` (servidor dentro del Laboratorio de Redes), `www.upv.es` (servidor web de la UPV), `www.rediris.es` (servidor web de RedIris situado en Madrid), `www.uq.edu.au` (servidor web de la Universidad de Queensland en Australia), `www.berkeley.edu` (servidor web de la Universidad de California en Berkeley). La opción `-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
<code>zoltar.redes.upv.es</code>			
<code>www.upv.es</code>			
<code>www.rediris.es</code>			
<code>www.berkeley.edu</code>			

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 -c 3 zoltar.redes.upv.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 1.

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?
- ¿Para qué se utiliza el campo **identificador** de la cabecera ICMP?

5. La orden *traceroute*

La orden *traceroute* (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 *traceroute* 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 *traceroute* es que este mensaje ICMP contiene la dirección IP del router que lo ha enviado.

En el caso del *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 **traceroute** utiliza dos opciones distintas:

- 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.
- 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 **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 **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 **traceroute** para los siguientes destinos y anota el número de saltos.

	Saltos
www.uv.es	
www.ua.es	
www.usc.edu	

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 **traceroute** www.ua.es.

En el **traceroute** a www.usc.edu, 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 traceroute desde su máquina al destino que le hayas indicado. En nuestro caso le vamos a poner la dirección IP de nuestra máquina de trabajo en el Laboratorio de Redes. 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 traceroute al servidor **www.telstra.net**, para obtener la ruta inversa. Compara ambos resultados.

¿Se sigue el mismo recorrido desde la UPV a 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 `traceroute -n www.upv.es`. Para capturar también los paquetes enviados por tu ordenador tendrás que modificar el filtro para que incluya paquetes udp. Puedes emplear el filtro de captura “icmp or (udp and host 158.42.180.X)”, donde la X representa el valor en decimal del último octeto de la dirección IP de tu ordenador.

Para los paquetes que envía tu ordenador:

- En la cabecera IP: ¿cómo se modifica el TTL?
- En la cabecera UDP: ¿cómo se modifica el puerto destino?

Para los paquetes de respuesta completa la tabla siguiente (es suficiente un par de paquetes de cada uno de los tipos ICMP recibidos):

Cabecera ICMP		Datos ICMP (Información sobre el error)	
Tipo	Código	TTL (cabecera IP)	Protocolo (cabecera UDP)

Basándote en la información anterior, relaciona las respuestas recibidas con los paquetes que envió tu ordenador. Observa que las respuestas a los paquetes enviados pueden recibirse desordenadas, ya que vienen de distintos dispositivos.

Indica por qué se envía información sobre las cabeceras IP e ICMP en los paquetes ICMP de error.

2.7 El protocolo ICMP

- En Internet no disponemos de mecanismos hardware para comprobar la conectividad
- IP no proporciona herramientas para la detección de fallos y problemas
- Se introduce un nuevo módulo: **ICMP***
(*Internet Control Message Protocol*)
- Este protocolo permite a los hosts y routers enviar mensajes de control a otros hosts y routers

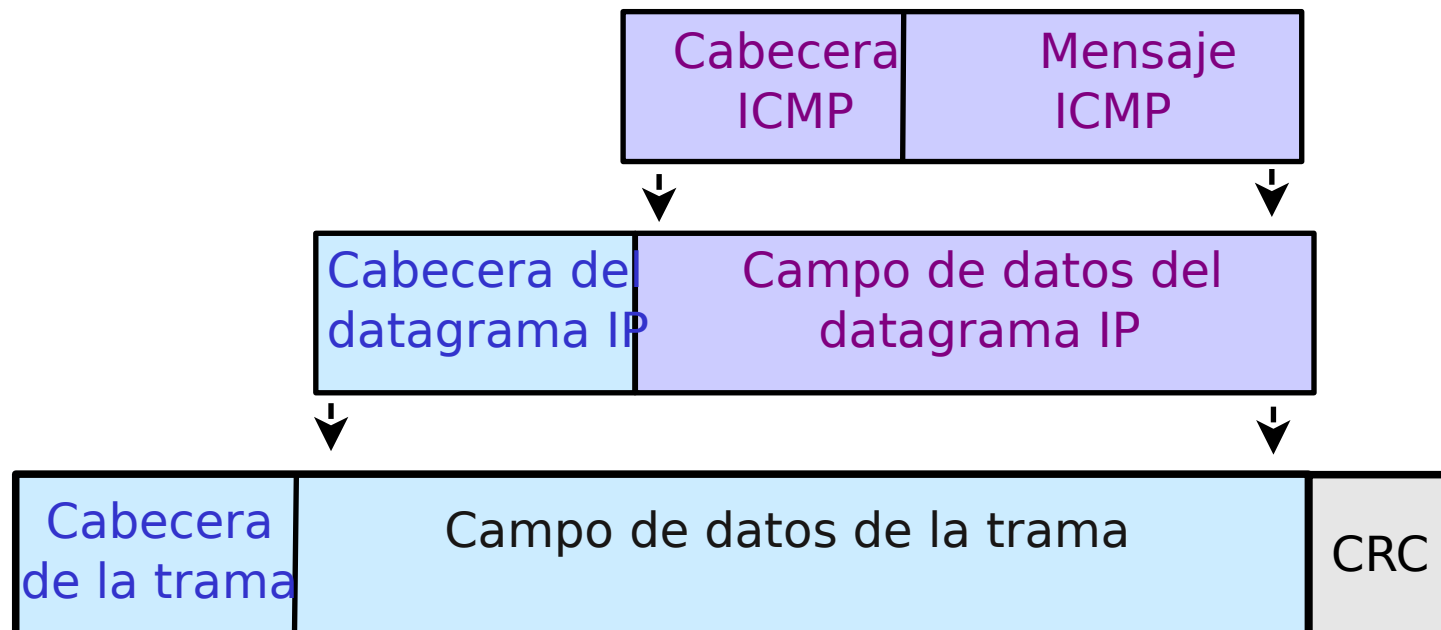
* RFC 792

Generalidades sobre ICMP

- ICMP nos permite saber, por ejemplo, por qué no se ha entregado un datagrama (no hay ruta, el destino no responde, agotado el tiempo de vida, etc.)
- Informa de **errores sólo al origen** del datagrama
- No corrige el problema (**sólo informa**)
- Los mensajes ICMP **viajan en el campo de datos de un datagrama IP**

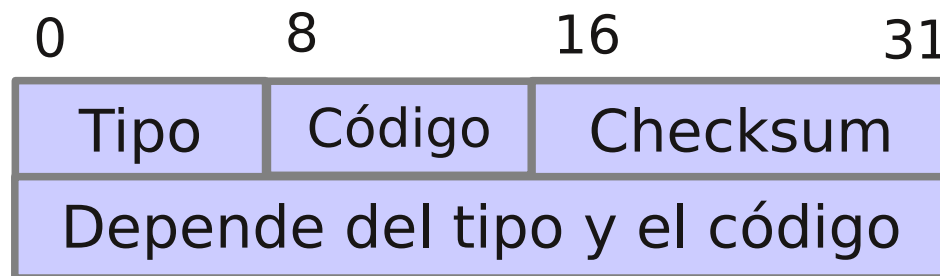
Encapsulamiento de un mensaje ICMP

- Los mensajes ICMP se encapsulan en datagramas IP
 - Pero ICMP no se considera un protocolo de nivel superior a IP



Formato de un mensaje ICMP

- Cada mensaje 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



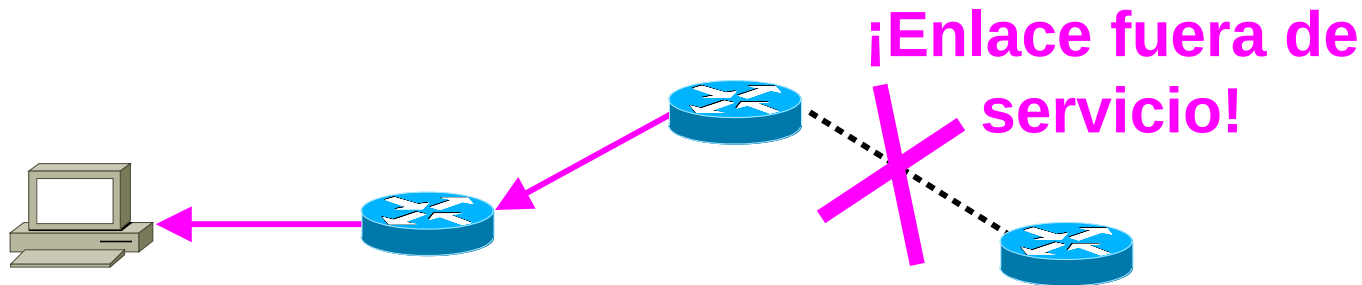
Tipos de mensajes ICMP

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

Tipo	Mensaje ICMP	Petición	Error
		✓	✓
0	Contestación de eco	✓	
3	Destino inalcanzable		✓
8	Petición de eco	✓	
11	Tiempo excedido en datagrama		✓

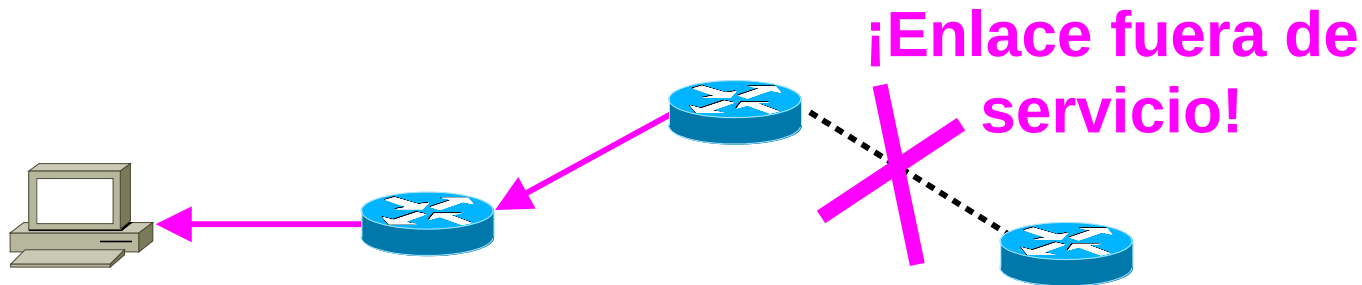
- Los mensajes de error contienen la cabecera IP + 8 primeros bytes de datos del datagrama original

Mensajes de error ICMP



- 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 máquina única (es decir, que la dirección origen no puede ser cero, la dirección de *loopback*, direcciones de difusión)
- Todo esto para prevenir *broadcast storms*

Mensajes de error ICMP



- 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 máquina única (es decir, que la dirección origen no puede ser cero, la dirección de *loopback*, direcciones de difusión)
- Todo esto para prevenir *broadcast storms*

Mensajes de eco (petición y respuesta)

- La respuesta devuelve los mismos datos que se recibieron en la petición
- 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

Mensaje de tiempo excedido

- Este tipo de mensajes pueden ser enviados por routers y hosts:
 - **Routers**: cuando descartan un datagrama al finalizar su tiempo de vida
 - **Hosts**: al ocurrir un timeout mientras se esperan todos los fragmentos de un datagrama
- El campo código explica cuál de los dos sucesos ha ocurrido

Mensaje de destino inalcanzable

- Son enviados por un *router* o host cuando no puede enviar o entregar un datagrama IP
- Se envían al emisor del datagrama original
- El campo código contiene un entero con información adicional. Algunos importantes son:

Cod.	Descripción
0	Red inalcanzable
1	<i>Host</i> inalcanzable
2	Protocolo inalcanzable
3	Puerto inalcanzable
4	Se requiere fragmentación pero bit DF activado
6	Red destino desconocida
7	<i>Host</i> destino desconocido

Práctica 5: NAT: funcionamiento y análisis de trazas

Lectura previa: Kurose 7ª edición, apartado 4.3.4 (páginas 286 – 288)

Video: <https://www.youtube.com/watch?v=hBloGXo2DIY>

Trabajo previo antes de la sesión de laboratorio:

- Lectura y comprensión de los apartados 1. Introducción y 2. Traducción de direcciones de red.
- Resolución del Ejercicio 1

1. Introducción

En esta práctica vamos a estudiar el funcionamiento del mecanismo NAT, o traducción de direcciones IP.

El mecanismo NAT nace como respuesta a la proliferación de pequeñas redes domésticas y de oficina con conexión a Internet. Cuando se contratan los servicios básicos de un ISP, éste nos proporciona una conexión a Internet con un ancho de banda determinado (de acuerdo al contrato elegido) y una única dirección IP con la que podemos identificarnos en Internet.

Esta configuración es suficiente si queremos conectar un único ordenador a Internet. Sin embargo, en el caso habitual de disponer de una pequeña red de área local y desear que los diferentes ordenadores de la misma puedan acceder a Internet simultáneamente, los servicios que nos proporciona el ISP no son suficientes. Más concretamente, el hecho de disponer de una única dirección IP (o hablando en términos más generales, de disponer de menos direcciones IP que ordenadores) nos crea el problema de que no todos los ordenadores de nuestra red van a poder conectarse a Internet de forma simultánea ya que no tienen una dirección IP con la que identificarse.

Una solución a este problema sería contratar un pequeño rango de direcciones IP para casa o la oficina. Pero esta opción, además de ser notablemente más costosa que una única dirección IP, conlleva además la necesidad de gestionar un router, lo cual queda fuera del alcance de los conocimientos de la mayoría de usuarios de Internet.

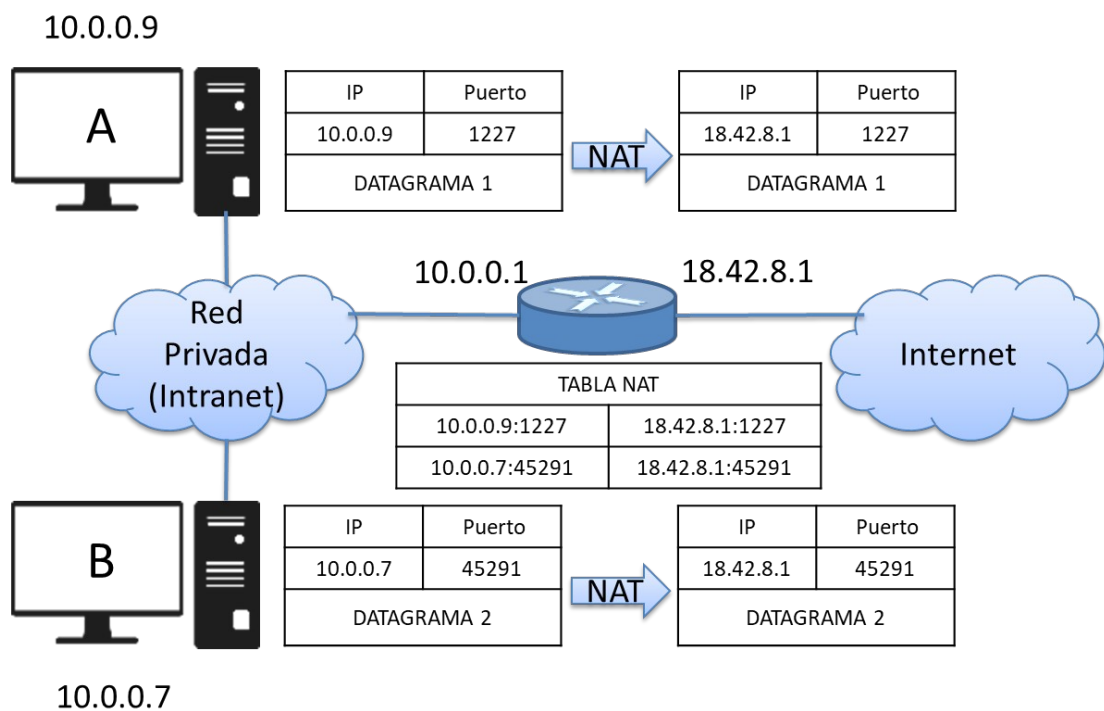
Una mejor opción es seguir contratando una única dirección IP e idear algún mecanismo para compartir esa dirección que nos ha proporcionado el ISP entre los ordenadores de la red de casa o de la oficina. Este mecanismo se conoce como traducción de direcciones, o NAT (*Network Address Translation*).

2. Traducción de direcciones de red

El mecanismo de traducción de direcciones de red (NAT) se complementa generalmente con el uso de direcciones privadas, de forma que es habitual encontrar dicho tipo de direcciones en la red local de casa o de la oficina (también conocida como intranet) que usan este mecanismo para acceder a Internet. No obstante, el funcionamiento del mecanismo es independiente del tipo de direcciones que se use en la intranet.

A modo de resumen (estudiar la sección 4.3.4 del Kurose para una información más detallada), el funcionamiento del mecanismo NAT es como sigue. Tal y como se muestra en la figura siguiente, cuando un ordenador de la red local accede a un servidor en Internet, envía el correspondiente datagrama al dispositivo NAT (también conocido como *router* NAT, a pesar de que sus funciones quedan lejos de las de un *router*). El *router* NAT actúa como puerta de enlace de la red local. Este dispositivo cambia la dirección origen del datagrama, que será en general una dirección privada, por la dirección pública que ha facilitado el ISP. Asimismo, en caso necesario (si el puerto ya está en uso en otro ordenador de la red interna), cambia el puerto origen del segmento TCP o del datagrama UDP por uno nuevo, con el fin de poder reenviar posteriormente la respuesta del servidor al host que originó la petición.

Durante el proceso de traducción el dispositivo NAT guarda en una tabla (tabla de traducciones) la equivalencia entre el puerto origen inicial y el nuevo puerto origen para cada uno de los datagramas que lo atraviesa. De esta forma, cuando llega una respuesta desde Internet, utiliza el puerto destino de la respuesta (puerto origen cambiado en la petición correspondiente anterior) para buscar en dicha tabla la entrada correspondiente y saber a qué host de la red local debe reenviar la respuesta del servidor. En el proceso de reenvío hacia el interior de la red local modifica la dirección destino y el puerto destino para que coincidan con los iniciales. En la figura siguiente se puede ver un ejemplo de la tabla de traducciones.



Ejercicio 1. Se pretende configurar manualmente la interfaz de la red pública de un dispositivo NAT con los siguientes valores:

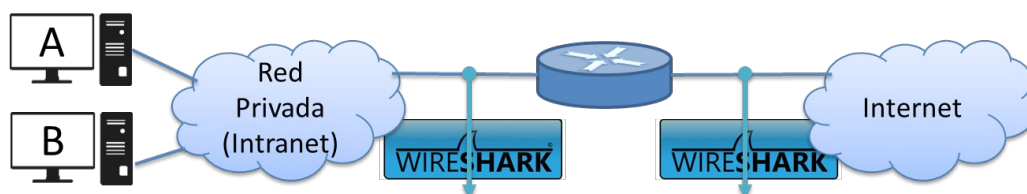
Dirección IP pública:	158.42.180.1
Máscara de subred:	255.255.255.0
Puerta de enlace:	158.42.181.250
Servidor de nombres 1:	158.42.249.8
Servidor de nombres 2:	158.42.1.8

Una vez introducidos esos valores, el dispositivo NAT nos dice que la dirección IP de la puerta de enlace es incorrecta. Sin embargo, tras comprobar los valores, tanto la dirección IP de la puerta de enlace como la dirección IP pública asignada al NAT son correctas.

1. ¿Por qué nos dice el NAT que la dirección IP de la puerta de enlace es incorrecta?
2. ¿Qué parámetro habría que cambiar para que la configuración fuera correcta? Propón un valor para dicho parámetro que haga correcta la configuración.
3. ¿Están los servidores DNS en la misma red que la interfaz configurada? En caso de no estarlo, ¿sería posible un correcto funcionamiento? Justifica la respuesta.

3. Análisis de tráfico

En esta sección vamos a analizar los paquetes que atraviesan un dispositivo NAT. En lugar de realizar nosotros las capturas de paquetes con el programa *Wireshark*®, vamos a usar unas capturas previamente realizadas y que están contenidas en los ficheros HTTP_LAN_1, HTTP_LAN_2, FTP_LAN_1, SSH_LAN_1 y también sus respectivas versiones en la parte del ISP. Estos ficheros se encuentran en PoliformaT. El motivo para utilizar unas capturas ya hechas es que, aunque podríamos fácilmente capturar el tráfico que genera nuestro ordenador en la red local, es complejo capturar el tráfico que sale del router NAT, dado que de normal no se tiene acceso a esa red. La siguiente figura muestra el escenario en el que vamos a trabajar.



Ejercicio	Captura Intranet	Captura Internet
Ejercicios 2, 3 y 4	HTTP_LAN_1	HTTP_ISP_1
Ejercicio 5	HTTP_LAN_2	HTTP_ISP_2
Ejercicio 6	FTP_LAN_1	FTP_ISP_1
Ejercicio 7	SSH_LAN_1	SSH_ISP_1

Ejercicio 2. El fichero HTTP_LAN_1 contiene una captura de tráfico generado dentro de la LAN (intranet). Abre el fichero HTTP LAN 1 con *Wireshark* y responde a las siguientes cuestiones:

1. ¿Cuántas conexiones TCP se realizan?
2. Localiza el datagrama que contiene la petición web “GET / HTTP/1.1”. ¿Cuáles son las direcciones IP y números de puerto TCP utilizados ¿Corresponden origen y destino a cliente y servidor web, respectivamente?
3. Localiza el datagrama que contiene la respuesta "HTTP 200 OK" enviada por el servidor web ¿Las direcciones IP y números de puerto corresponden a los de cliente y servidor encontrados en la petición web anterior?
4. ¿Qué puertos se usan en la segunda conexión TCP?

Ejercicio 3. Vamos a centrarnos ahora en los mensajes HTTP que salen y entran del router NAT hacia o desde el servidor web (red ISP). Para ello, abre el fichero HTTP ISP 1, donde se muestra el tráfico que se ha generado a la parte externa del router NAT (internet) a partir del tráfico generado en la intranet (captura anterior). *Nota: los tiempos de ambas capturas no están sincronizados.*

Localiza el mensaje de salida “GET / HTTP/1.1”. Éste corresponde al reenvío por parte del router NAT del datagrama correspondiente generado en la intranet (ejercicio 2).

1. ¿En qué momento se transmite este mensaje? ¿Cuáles son las direcciones IP y números de puerto TCP utilizados ¿Corresponden origen y destino a cliente y servidor web, respectivamente?
2. ¿Cuál es la dirección IP pública del *router* NAT?
3. Respecto del datagrama correspondiente de la intranet ¿ha cambiado algún campo del mensaje HTTP? ¿y de la cabecera del datagrama IP? Para aquellos campos que se han modificado ¿cuál es el motivo?

Localiza el mensaje de entrada "HTTP 200 OK" enviado por el servidor web.

4. ¿En qué momento se recibe este mensaje? ¿Cuáles son las direcciones IP origen y destino del datagrama que lo contiene? ¿Se corresponden con las observadas en el ejercicio 2?

Ejercicio 4. Basándote en la información observada en los dos ejercicios anteriores, rellena la tabla de traducciones del router NAT, donde se recogen las correspondencias entre direcciones IP y números de puerto a ambos lados del router NAT en las dos conexiones TCP.

Conexión	Entrada desde LAN		Salida a red ISP	
	IP origen	Puerto origen	IP origen	Puerto origen
1				
2				

Ejercicio 5. En el ejercicio anterior se ha podido ver que el router NAT ha mantenido los mismos números de puerto origen en los segmentos de salida. Esta política es tan válida como cualquier otra, siempre que se lleve cuidado de que si el puerto a usar ya está en la tabla de traducciones, entonces habrá que usar un número de puerto nuevo. Precisamente eso es lo que vamos a comprobar en este ejercicio. Vamos a provocar que el router NAT tenga que modificar el número de puerto origen porque el que necesita ya está en uso. Para ello, en dos ordenadores de la red privada vamos a ejecutar el siguiente programa:

```
import java.net.*;
import java.io.*;

class ClienteTCP {
    public static void main(String args[]) throws Exception {
        String mi_IP = "192.168.1.2";
        InetAddress DirIP = InetAddress.getByName(mi_IP);
        Socket s = new Socket("www.redes.upv.es", 80, DirIP, 40000);
        PrintWriter esc= new PrintWriter(s.getOutputStream(), true);
        esc.println("GET / HTTP/1.1");
        esc.println("Host: www.redes.upv.es");
        esc.println();
        while(true);
    }
}
```

La variable “mi_IP” contendrá la dirección IP del ordenador en cuestión donde se esté ejecutando el programa. Básicamente, este programa abre una conexión TCP con un servidor web y la mantiene abierta. De esta forma, cuando ejecutemos el mismo programa en el segundo ordenador, el *router* NAT verá que el puerto que necesita está ocupado y no tendrá más remedio que usar uno nuevo que esté libre.

Abre el fichero HTTP LAN 2, que corresponde al tráfico generado por uno de los ordenadores de la red privada (LAN) donde se ha ejecutado el programa anterior.

1. ¿Cuáles son las direcciones IP de cliente y servidor y los números de puerto utilizados por cada uno de ellos?

Abre el fichero HTTP ISP 2. Este fichero contiene el tráfico generado en la red pública (ISP) por los dos ordenadores que ejecutaron el programa anterior en la red privada.

2. ¿Qué número de puerto origen asigna el *router* NAT a los segmentos de salida a la red pública en cada una de las dos conexiones?
3. ¿Cómo sabemos cuál de las dos conexiones TCP corresponde a la contenida en el fichero HTTP_LAN_2? Sugerencia: revisa identificador del paquete IP.

Basándote en la información proporcionada, rellena la tabla de traducciones NAT:

Conexión	Entrada desde LAN		Salida a red ISP	
	IP origen	Puerto origen	IP origen	Puerto origen
1				
2				

Ejercicio 6. Hasta ahora hemos visto que el *router* NAT modifica los campos de la cabecera IP y de la cabecera TCP (o UDP). Pero en ocasiones también se ve obligado a modificar el contenido del mensaje que viaja en el segmento TCP. Estudia los ficheros FTP LAN 1 y FTP ISP 1.

1. ¿Cuáles son las direcciones IP de cliente y servidor *ftp*?
2. Fíjate en uno cualquiera de los dos ficheros. Observarás que se establecen dos conexiones TCP ¿Para qué se usa cada una? Si no lo sabes revisa el funcionamiento básico del protocolo *ftp* ¿Cuáles son los números de puerto utilizados por cliente y servidor en cada una de las conexiones? Identifica los segmentos de fin de conexión de cada una de ellas.

Nota: en esta última pregunta, así como en el siguiente apartado, puedes ayudarte del filtro de pantalla para visualizar sólo una de las conexiones, por ejemplo, filtrando por número de puerto. Para ello escribe `tcp.port==Número_puerto`. Recuerda limpiar los filtros de visualización cuando termines de usarlos.

3. Fíjate en el datagrama que se genera en la red privada en el instante 4.307125 y compáralo con su correspondiente en la red externa. ¿Que cambia entre ambos? ¿Altera esto los números de secuencia de los subsiguientes segmentos TCP?

4. Servidores dentro de la intranet

Como hemos visto, NAT funciona de forma automática cuando un ordenador de la intranet se conecta a un servidor fuera de la intranet. Esto es así porque el router NAT modifica de forma automática los números de puerto de los segmentos que salen hacia el exterior, así como las direcciones IP de los datagramas que los contienen, guardando dichas correspondencias en su tabla de traducciones.

Sin embargo, si nos restringimos al procedimiento anterior, cuando se trata de acceder desde el exterior a un servidor en la intranet, no existirá ninguna correspondencia en la tabla y, por tanto, el *router* NAT no sabrá qué transformaciones debe realizar. Así pues, habrá que “enseñar” al *router* NAT lo que debe hacer con las peticiones entrantes hacia servidores internos. En particular, son necesarios dos pasos:

1. Hay que configurar el dispositivo NAT para que acepte peticiones destinadas al puerto del servidor y, además, cuando llegue una de estas peticiones, el dispositivo NAT debe saber a qué ordenador en la intranet debe reenviar la petición. Esto es lo que se conoce como *port forwarding*. Todo esto hay que configurarlo antes de poder dar servicio al exterior.
2. Dado que, en muchos casos, las direcciones IP de la intranet se asignan dinámicamente gracias a un servidor DHCP incorporado en el *router* NAT, debemos asegurarnos que el ordenador que haga de servidor siempre obtenga la misma dirección IP. Si no es así, cuando llegue una petición a un puerto del servidor, el *router* la reenviará a la dirección IP de la intranet que tenga configurada, pero el servidor puede no estar en esa dirección IP.

Ejercicio 7. En esta práctica no vamos a configurar un *router* NAT para que realice *port forwarding*, pero vamos a analizar los paquetes que llegan a un dispositivo NAT desde el exterior y que van destinados a un servidor *ssh* que está ejecutándose en un ordenador de la intranet. Para ello vamos a utilizar los ficheros de captura SSH LAN 1 y SSH ISP 1.

1. ¿Cuáles son las direcciones IP de cliente y servidor **ssh** y los números de puerto utilizados por cada uno de ellos?
2. ¿Qué diferencias observas en los datagramas y segmentos capturados dentro y fuera de la intranet? ¿Se está modificando el contenido de los mensajes, como en el caso de **ftp**?

Práctica 5 - NAT (*Network Address Translation*)

NAT (RFC 2663)

Problema:

Queremos que **m** hosts puedan acceder simultáneamente a la red externa disponiendo sólo de **n** IPs públicas (**m > n**)

Solución:

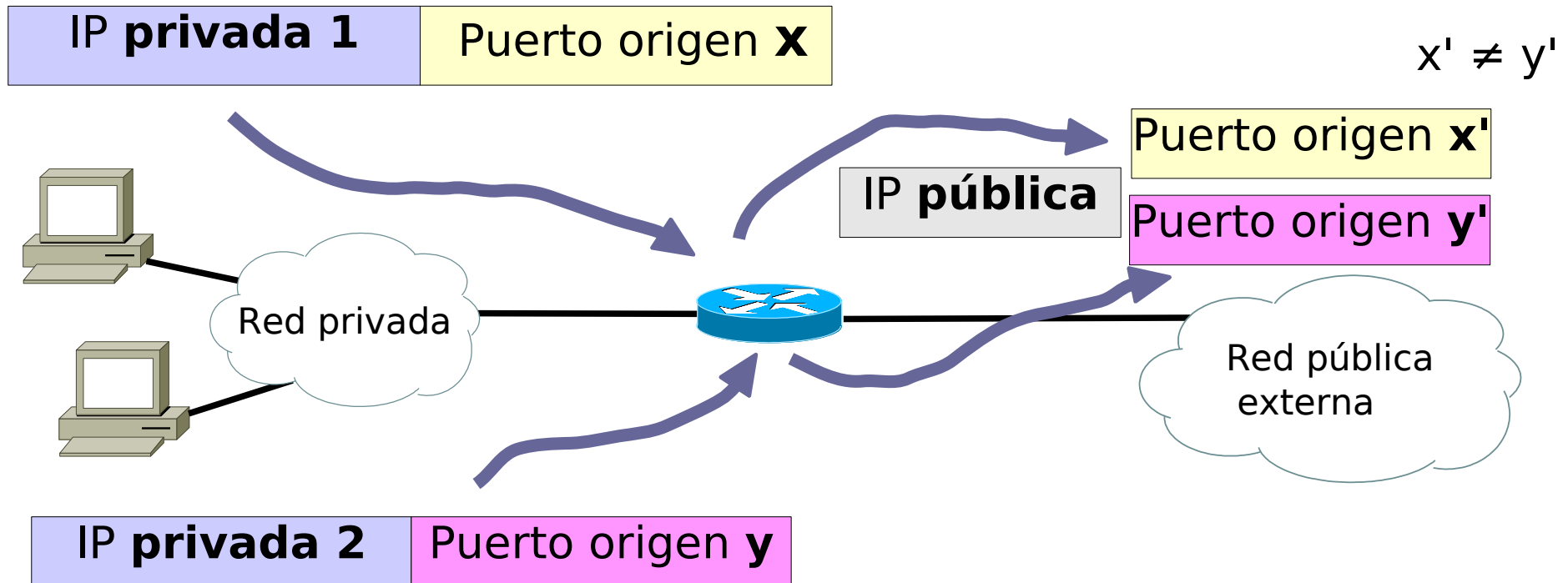
Asignar direcciones privadas en la red interna

Utilizar un procedimiento de traducción:

dirección privada ↔ dirección pública

para salir a la red externa

Funcionamiento de NAT



En el caso más simple una única IP pública es suficiente

Todos los datagramas que llegan al router tienen la misma IP destino: hay que demultiplexar por puerto

El router emplea una tabla de traducción

NAT: ejemplo

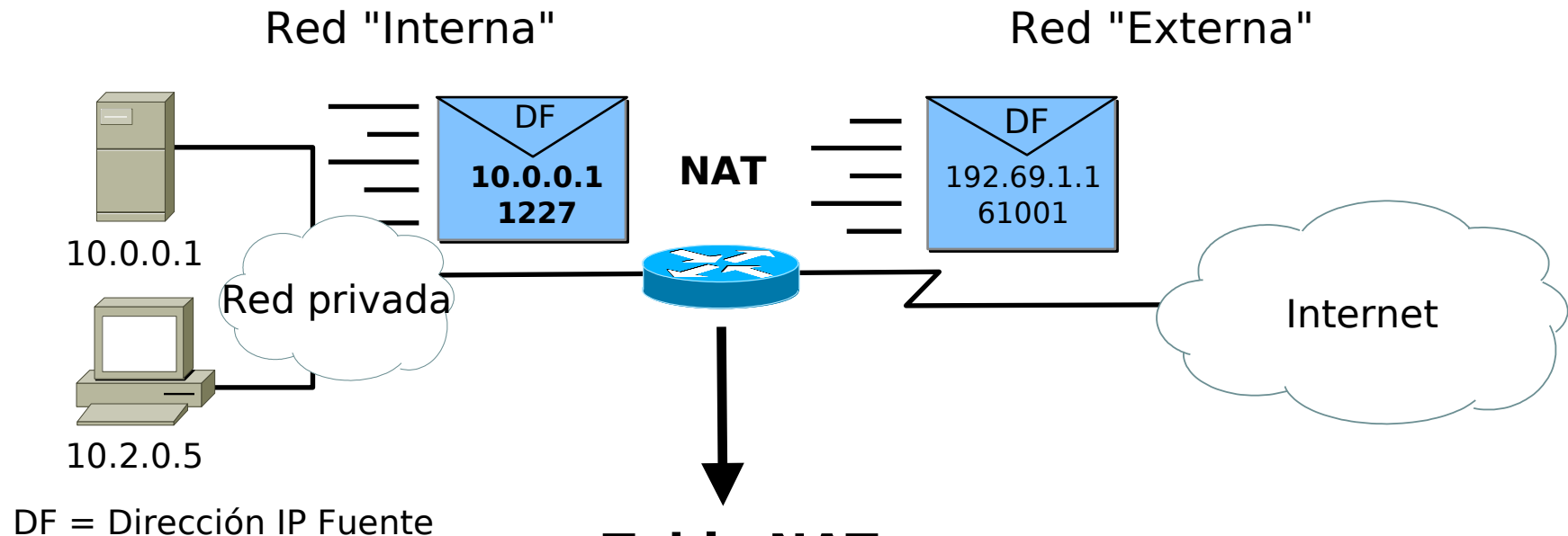


Tabla NAT

Dirección IP y puerto interno	Dirección IP y puerto externo
10.0.0.1: 1227	192.69.1.1: 61001
10.0.0.5: 1110	192.69.1.1: 61002

Desde el punto de vista "externo" todos los hosts utilizan una sola dirección IP

Se realiza multiplexación por puertos

NAT: Servidores en la red interna

¿Y si queremos tener un servidor en la red interna (dirección IP privada) que sea accesible desde el exterior?

Podemos asociar un puerto de entrada en el dispositivo NAT con la IP privada del servidor y el número de puerto donde atiende las peticiones (*port forwarding*)

Limitaciones de NAT

NAT funciona bien para direcciones IP en la cabecera IP

Algunas aplicaciones envían dir. IP en la parte de datos de aplicación

Por ejemplo: orden PORT de FTP

En general NAT no sabe manejar estas aplicaciones

Aunque es habitual que resuelvan algunos casos sencillos como el de FTP

Práctica 6: El protocolo ARP.

Lectura previa: Kurose2017, apartado 6.4.1

Videos:

<https://www.youtube.com/watch?v=2XdAXD3uS8c>

<https://media.upv.es/#/portal/video/8e5cbcf2-0f19-8740-988f-536217d4442a>

1. Introducción.

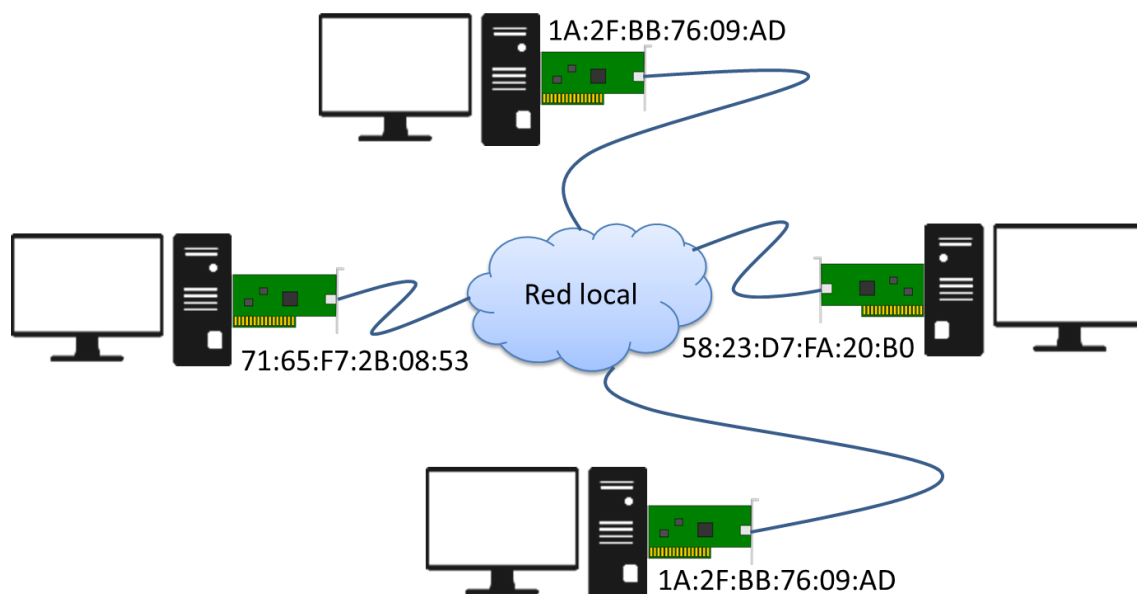
Cada uno de los hosts y routers que están conectados a Internet se identifica mediante una dirección del nivel de red: dirección IP. Pero, además, cada adaptador de red instalado en cada nodo dispondrá de una dirección del nivel de enlace: dirección física.

En esta práctica nos ocuparemos de estudiar el direccionamiento a nivel de enlace de red y también el Protocolo de Resolución de Direcciones (ARP, Address Resolution Protocol) que se encarga de traducir las direcciones IP en direcciones físicas.

2. Direcciones físicas

Las direcciones de la capa de enlace se asignan a los adaptadores de red y se les denomina de diversas formas: dirección LAN, dirección física o dirección MAC.

La dirección física tiene 6 bytes de longitud, lo que nos da 2^{48} posibles direcciones físicas. Estas direcciones se suelen expresar en notación hexadecimal, indicándose cada byte como una pareja de números hexadecimales. Lo podemos ver en la siguiente figura:



La dirección física de un adaptador tiene una estructura plana (no jerárquica como IP) y no variará, aunque el nodo al que pertenece cambie de red.

IEEE se encarga de gestionar el espacio de direcciones físicas, garantizando que dichas direcciones son únicas, independientemente de los fabricantes y de las redes. Cuando una empresa quiere fabricar adaptadores, debe comprar una parte del espacio de direcciones compuesto por 2^{24} direcciones. IEEE asigna el fragmento de 2^{24} direcciones fijando los primeros 24 bits de sus direcciones físicas y dejando que la empresa diseñe combinaciones únicas de los últimos 24 bits para cada adaptador.

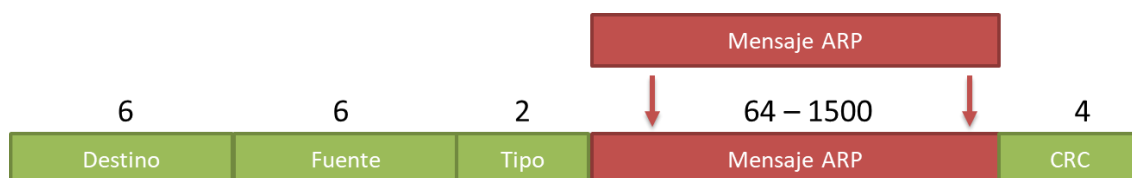
Cuando dos nodos pertenecientes a la misma red quieren comunicarse van a necesitar conocer no solo la dirección IP del otro sino también su dirección física. Un computador solo necesita averiguar la dirección física de otro si ambos comparten la misma red IP.

3. Protocolo ARP

Para que un datagrama IP viaje por la red de área local, este debe encapsularse dentro de una trama (Ethernet en nuestro caso). Esa trama Ethernet contiene la dirección física del siguiente destino, que puede tratarse del computador final al que van dirigidos los paquetes (origen y destino en la misma red local) o del *router* que encaminará el paquete hacia el exterior (origen y destino en distintas redes).

En TCP/IP se utiliza un protocolo para la obtención de direcciones físicas a partir de direcciones IP dentro de una red de área local. Este protocolo se conoce con el nombre ARP (*Address Resolution Protocol*).

Los mensajes del protocolo ARP pertenecen al nivel de enlace y son encapsulados en el campo de datos de una trama. El campo **Tipo** en la cabecera de la trama permitirá identificar el tipo de mensaje: 0x806 identifica al protocolo ARP en el caso de Ethernet.



Para averiguar una dirección física, la capa de enlace enviará un paquete ARP de consulta que contendrá la dirección IP origen, dirección física origen y dirección IP destino. Este mensaje irá dirigido a todos los nodos de la red, empleando para ello la dirección de difusión: FF:FF:FF:FF:FF:FF como dirección destino.



Esta consulta ARP llega a todos los nodos de la red. Cada uno de ellos comprueba si la dirección IP por la que se está consultando le pertenece. Aquel cuya dirección coincida con la pregunta responderá a esta consulta y lo hará mediante un mensaje ARP de respuesta en el que añadirá su dirección física. Este mensaje ya no se enviará por difusión, sino que será dirigido al nodo que ha realizado la consulta.



Cada nodo tiene en su memoria una tabla ARP donde almacena temporalmente las correspondencias entre las direcciones IP y direcciones físicas que ha utilizado, y presumiblemente podría volver a utilizar. Cada entrada tiene asociado un valor de tiempo

de vida (TTL), que indica cuándo se eliminará de la tabla. ARP se considera un protocolo *plug-and-play* porque la tabla ARP se construye automáticamente, no necesita ser configurada por el administrador del sistema.

Cuando un nodo necesite una correspondencia entre dirección IP y física, consultará en primer lugar su tabla ARP, y en caso de no disponer de la información, lanzará la consulta ARP a la red.

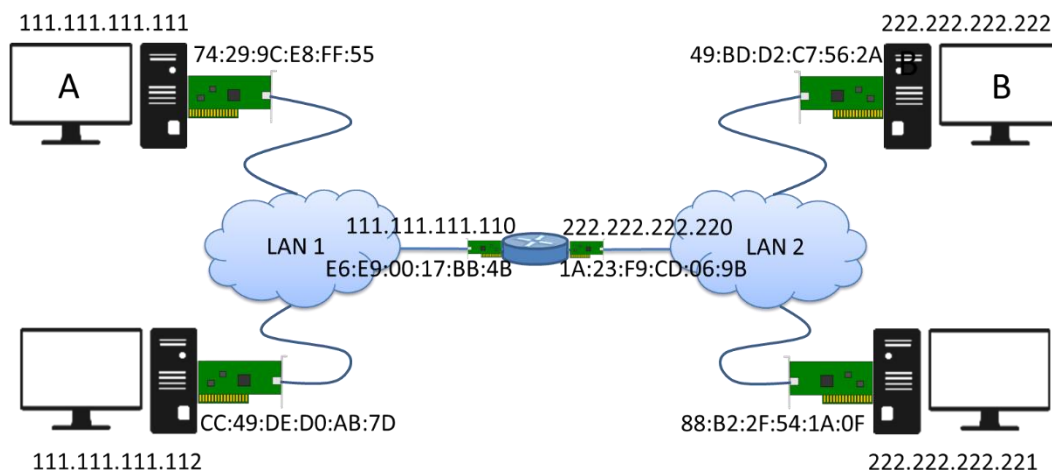
Los nodos no añaden a la tabla ARP únicamente las correspondencias entre direcciones IP y físicas por las que preguntan, sino que también añade aquellas correspondencias que son capaces de aprender por todos los paquetes ARP que capturan. Hay que tener en cuenta que todos los paquetes ARP llevan, al menos, la correspondencia de la dirección IP origen y la dirección física origen.

Existen dos situaciones por las que un nodo puede añadir una entrada a la tabla ARP (suponiendo que no la tenía ya):

- El nodo que realiza la petición ARP, cuando recibe la respuesta, apuntará en la tabla el contenido de dicha respuesta.
- El nodo al que va destinada la petición ARP apuntará en su tabla ARP la dirección IP del nodo origen junto con su dirección física.

Además, cuando un nodo recibe una petición ARP mediante difusión donde no es ni el origen ni el destino, actualizará su tabla ARP únicamente si la dirección IP del emisor ya se encuentra en la tabla.

Este mecanismo es suficiente cuando dos computadores pertenecientes a la misma red desean intercambiar un datagrama IP. El origen averigua la dirección física del destino y encapsula en datagrama IP en una trama destinada al mismo. Sin embargo, cuando el nodo destino pertenece a otra red IP no es posible esta entrega directa. Estudiemos esta situación basándonos en la siguiente figura.



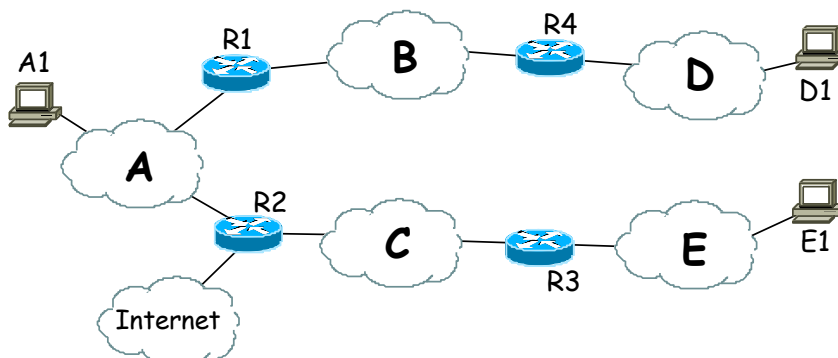
En la figura aparecen dos redes: 111.111.111.0/24 y 222.222.222.0/24, ambas conectadas mediante un *router* R. Como vemos, dicho *router* dispone de dos adaptadores, con sus correspondientes direcciones físicas e IP. Supongamos que el nodo A quiere enviar un datagrama IP al nodo B. La secuencia de eventos será la siguiente:

1. El nodo A generará un datagrama con dirección IP origen 111.111.111.111 y dirección IP destino 222.222.222.222. Al aplicar su algoritmo de reenvío, A decidirá que B no está en su red IP, por lo que consultará sus tablas de reenvío, que le indican que la puerta de enlace adecuada es el *router* R.
2. Indicando que el datagrama ha de entregarse a la IP 111.111.111.110 (*router* R), A lo pasará a su nivel de enlace.
3. El nivel de enlace necesita generar una trama con dirección MAC origen la del nodo A (74-29-9C-E8-FF-55) y como dirección destino la del adaptador del *router* R perteneciente a su misma subred (E6-E9-00-17-BB-4B). Para averiguar esta dirección física destino el nodo A consultará su caché ARP y, si es necesario, realizará una consulta ARP en su red. El campo de datos de la trama contendrá el datagrama generado en el nivel de red.
4. Cuando la trama llegue al *router* R, el datagrama IP pasará al nivel de red, desapareciendo la trama original. El *router* R consultará su tabla de reenvío para determinar qué interfaz ha de emplear para alcanzar el destino, y si éste puede hacer una entrega directa. Se decidirá que el datagrama ha de utilizar la interfaz 222.222.222.220, que pasa el datagrama a su el nivel de enlace.
5. El datagrama IP deberá ser encapsulado en una nueva trama que tiene como dirección física origen la del *router* R (1A-23-F9-CD-06-9B) y como dirección física destino la del nodo B (49-BD-D2-C7-56-2A). Nuevamente, será necesaria la consulta a la caché ARP, y quizá una nueva consulta ARP, para que el *router* R averigüe la dirección física del nodo B.
6. Una vez la trama llega al nodo B, el nivel de enlace extraerá el datagrama y lo pasará al nivel de red, que lo procesará convenientemente.

Por último, comentar que el protocolo ARP para Ethernet está definido en el documento RFC 826.

Ejercicio 1.- En la figura se muestra un conjunto de redes locales Ethernet (A, B, C, D y E) de una empresa conectadas entre sí por cuatro routers (R1, R2, R3 y R4). La red se conecta a Internet a través del router R2. Emplearemos la notación IP(D1) e IP(R4D) para denotar las direcciones IP del host D1 y el Router 4, adaptador conectado a la red D. Del mismo nodo, MAC(D1) y MAC(R4D) se refieren a las direcciones físicas correspondientes. Supondremos todas las máquinas correctamente configuradas y las resoluciones DNS en caché.

- a) Si suponemos que inicialmente las cachés ARP asociadas a los adaptadores están vacías, indica cómo quedarán las cachés ARP de todos los adaptadores después de que A1 envíe un mensaje a D1 y después de que D1 le conteste a A1.
- b) Si a continuación, E1 envía un mensaje a D1, ¿cómo quedan las cachés ARP?



4. Análisis de tráfico

En esta práctica utilizaremos el analizador de protocolos *Wireshark*, que ya conocemos, para analizar el tráfico ARP generado en la red. Para ello necesitamos conocer algo más de la máquina con la que estamos trabajando.

Ejercicio 2.- Mediante el comando *ifconfig* en Linux o *ipconfig /all* en Windows, averigua cuántos adaptadores de red tiene la máquina con la que estás trabajando. Anota la dirección física de cada uno de ellos y observa si son del mismo fabricante. ¿Todos los adaptadores tienen direcciones IP asignadas? ¿De qué tipo es cada una de las direcciones IP que te aparece?

Adaptador	Dirección Física	Dirección IP

En las prácticas anteriores hemos centrado el análisis de la información proporcionada por las capturas de *Wireshark* en los niveles superiores de la pila de protocolos: aplicación, transporte y red. En esta práctica, en cambio, estudiaremos la información relacionada con el nivel de enlace de datos.

Ejercicio 3.- Realiza una captura con el *Wireshark* mientras cargas en el navegador la página web: <http://www.upv.es> (no omitas el protocolo en la URL). Utiliza el filtro de captura “**tcp port 80 and host www.upv.es**” para poder quedarte con el tráfico HTTP. Selecciona el primer mensaje HTTP de petición que transporta el GET y analiza en la ventana intermedia las diferentes cabeceras de la pila de protocolos TCP/IP.

- ¿Qué tipo de direccionamiento se utiliza en el nivel de transporte? ¿Y en el nivel de red?
- Expande la información relacionada con el nivel de enlace (Ethernet) ¿Cuál es la dirección física origen? ¿Y la dirección física destino?
- Observa que *Wireshark* expresa las direcciones físicas en dos formatos. ¿Qué dígitos identifican al fabricante del adaptador?
- En la cabecera de Ethernet aparece un campo más que indica el Tipo. ¿Qué se identifica con este campo? ¿Qué valor tiene en nuestro caso y a quién identifica?

El analizador *Wireshark* muestra, por defecto, información del nivel más alto posible. Sin embargo, es posible centrarnos en los niveles inferiores (enlace y físico). Para ello, a través de la opción *Analyze->EnabledProtocols*, deshabilitaremos el protocolo IPv4. Observa cómo ha cambiado el aspecto de las ventanas del *Wireshark*, especialmente la superior. ¿Qué valores aparecen ahora en las columnas Origen, Destino y Protocolo? Vuelve a habilitar el protocolo IP para tener una visión completa de TCP/IP.

Recordemos que el protocolo ARP mantiene una caché con las últimas correspondencias *dirección IP-dirección MAC* averiguadas. El comando *arp* nos permite tanto ver como manipular los contenidos de esta caché:

- **arp -a**: nos permite visualizar el contenido de la caché local de ARP.
- **arp -d <dir_IP>**: nos permite eliminar entradas manualmente de la caché. Para ello el usuario ha de tener permisos de root.
- **arp -s <dir_IP> <dir_Eth>**: para añadir entradas manualmente a la caché. También necesitamos permisos de root.

A lo largo de la práctica podrás encontrar momentos en los cuales ya está disponible en caché ARP una dirección física, y por tanto no se producen los mensajes ARP que quieres obtener. En tal caso, puedes emplear la orden **sudo arp -d <dirIP>** (necesitarás ser administrador, por lo que, en Windows, debes ejecutar el “Símbolo del sistema” como administrador. En Linux, debes usar “sudo arp -d <dirIP>”) para invalidar la entrada en la tabla, lo cual obliga a volver a consultar la dirección física empleando ARP cuando esta información es necesaria.

Ejercicio 4.- Desde una ventana de comandos ejecuta la orden **arp -a** para comprobar el contenido de la caché. Anota los resultados. ¿A qué máquina o máquinas pertenece la información que obtienes? ¿Qué relación tiene una de las direcciones obtenidas con el ejercicio anterior?

A continuación, realiza una captura con el *Wireshark* empleando el filtro “**(arp or icmp)**”.

Vamos a realizar ping hacia otro dispositivo que esté conectado a tu misma red Wifi (tendrás que averiguar la dirección IP de dicho dispositivo). Si utilizas dispositivos móviles o portátiles asegúrate que están conectados a la misma red Wifi y no a los datos.

Nota: Para dispositivos Android puede buscar en “Ajustes” → “Acerca del teléfono” → “Estado”, o bien, “Ajustes” → “Sobre el teléfono” → “Todas las especificaciones” → “Estado”, o alguna opción similar dentro de “Ajustes”, dependiendo de tu modelo dispositivo.

Ejecuta la orden **ping** a dicho dispositivo y examina nuevamente el contenido de la caché ARP.

Ahora mira la captura obtenida y localiza la consulta y respuesta ARP que se ha generado relacionada con la orden ping. Puedes utilizar el filtro *ARP* en la ventana de visualización. Tras seleccionar la consulta ARP, responde a las siguientes cuestiones:

- ¿Qué niveles de la pila de protocolos TCP/IP aparecen en el mensaje ARP? ¿Por qué? ¿Dónde se encapsulan los mensajes ARP?
- Evalúa la cabecera Ethernet de la petición ARP. Observa los valores de las direcciones origen y destino, así como el campo de Tipo. ¿Qué identifica este último campo?
- Observa los campos del mensaje ARP. ¿Qué información proporciona la consulta ARP al resto de nodos de la red? ¿En qué campo indica la dirección IP consultada?

Selecciona ahora el mensaje de respuesta ARP asociado a la consulta realizada:

- Observa las direcciones origen y destino de la cabecera Ethernet, e identifica a quién pertenece la dirección física origen. Comprueba que el campo Tipo identifica nuevamente al protocolo ARP.
- ¿Qué direcciones aparecen en el interior del mensaje ARP de respuesta? ¿A quién pertenecen? ¿Dónde aparece la información que nuestro equipo había solicitado?
- ¿Qué campo permite diferenciar una consulta ARP de una respuesta?

Práctica 8: Análisis de tráfico WIFI

1. Video

<https://media.upv.es/#/portal/video/c702b81a-35f9-469f-89e1-d1852b93ac83>

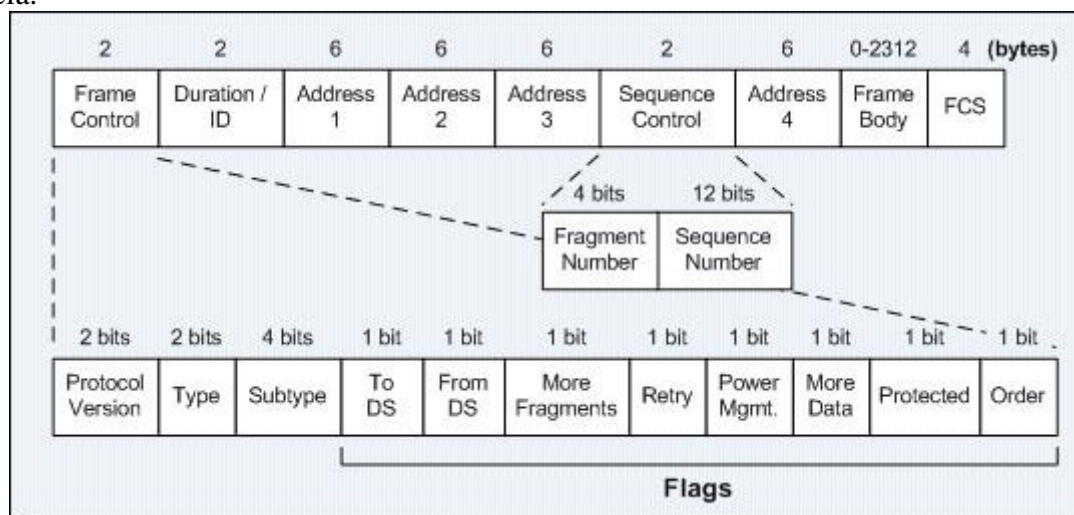
2. Introducción

Las redes de datos inalámbricas prácticamente se han convertido en algo imprescindible en nuestra vida cotidiana. Tras su introducción masiva en los ordenadores portátiles, ahora hacemos uso de ellas en todo tipo de dispositivos móviles (teléfonos, tabletas, etc.). El motivo de su gran aceptación es claro: permiten una flexibilidad que las redes cableadas no tienen. Además, al no necesitar conectores físicos también se permite la fabricación de dispositivos más compactos y manejables. Sin embargo, el hecho de no proteger la señal que transporta los datos en el medio de transmisión, como ocurre en las redes cableadas, provoca que las prestaciones de las redes inalámbricas sean, en general, inferiores a las de las redes cableadas. No obstante, la mayor flexibilidad que aportan suele compensar esta pérdida de prestaciones en la mayoría de los casos.

En esta práctica vamos a realizar un primer estudio de las redes de datos inalámbricas que siguen el estándar IEEE 802.11. Nótese que estas redes son mucho más complejas que las redes cableadas como Ethernet. Por ello, en esta práctica vamos a realizar un estudio sencillo sin entrar en muchos de los detalles de funcionamiento de las redes IEEE 802.11. También vamos a incluir en este estudio cuestiones relacionadas con la red cableada a la que se conecta el punto de acceso de una red inalámbrica dada.

3. Formato de las tramas 802.11

El formato general de las tramas 802.11 se presenta en la siguiente figura, donde también se muestran los formatos de dos de los campos de la cabecera de las tramas: el campo de control y el de secuencia.



A modo de breve introducción, el significado de cada uno de los campos de la trama 802.11 es el siguiente:

- **Frame control:** este campo incluye diversos subcampos y flags necesarios para el correcto funcionamiento del protocolo. El significado de estos subcampos y flags se describirá un poco más abajo.
- **Duration/ID:** Este campo tiene diversos significados dependiendo del valor de los bits 15 y 14 (bits MSB). De forma resumida, este campo puede indicar, o bien el número de microsegundos que el medio de transmisión va a estar ocupado con la transmisión de la trama en curso, o bien cuestiones relacionadas con el ahorro de energía del dispositivo inalámbrico (en concreto, que se transmitan las tramas almacenadas en el punto de acceso mientras el dispositivo inalámbrico estaba en modo de bajo consumo).
- **Direcciones 1, 2, 3 y 4:** Estos campos de 48 bits contienen las direcciones MAC involucradas con la trama en curso. A modo de regla general, y para las tramas de datos únicamente, el campo “dirección 1” contiene la dirección MAC del dispositivo que debe recibir la trama que se está transmitiendo. El campo “dirección 2” contiene la dirección MAC del dispositivo que ha transmitido la señal inalámbrica.

Nótese que puede ocurrir que la MAC contenida en el campo “dirección 1” no sea el destino final de la trama (por ejemplo, el destino final de la trama no es un dispositivo inalámbrico sino una tarjeta de red en la parte cableada, más allá del punto de acceso). Asimismo, puede darse el caso de que la MAC contenida en el campo “dirección 2” no sea el origen inicial de los datos contenidos en la trama 802.11 (es decir, el origen de los datos sería una tarjeta de red en la parte cableada, por ejemplo). Por este motivo, hace falta un tercer campo (“dirección 3”) que indique dicha información. El campo “dirección 3” contiene el origen real de los datos (ya sea en la parte cableada o en la parte inalámbrica) o el destino final de los datos (en la parte cableada o en la parte inalámbrica). Para poder interpretar correctamente la información contenida en el campo “dirección 3” habrá que hacer uso de los flags “To DS” y “From DS” contenidos en el campo “Frame control”.

Finalmente, el campo “dirección 4” se usa para la comunicación entre puntos de acceso y queda por tanto fuera de este estudio inicial de las redes IEEE802.11.

- **Sequence Control:** Este campo se usa para tareas de reensamblado de mensajes del nivel superior que se enviaron fragmentados y también para detectar la recepción de tramas duplicadas. Para ello tiene dos subcampos:
 - **Fragment number:** Cuando un mensaje del nivel superior se fragmenta, este subcampo indica el número de fragmento (el primer fragmento tiene el número 0 y a partir de ahí se usan números consecutivos).
 - **Sequence number:** Este subcampo se usa a modo de identificador, de forma que dos tramas diferentes tienen valores diferentes en este subcampo. Si una trama debe retransmitirse (porque no se ha recibido el ACK correspondiente, por ejemplo), entonces la trama retransmitida mantiene el mismo número de secuencia. De esta forma, el receptor puede saber si los datos recibidos son nuevos o si ya se recibieron. Nótese que cuando una trama se fragmenta, todos los fragmentos tienen el mismo número de secuencia.
- **Frame body:** este campo contiene los datos enviados en la trama. La longitud máxima de este campo son 2304 bytes (2312 incluyendo los datos WEP).
- **FCS (Frame Check Sequence):** Es el código CRC que permite saber si la trama se ha recibido sin errores de transmisión.

Respecto al campo “frame control”, los diferentes subcampos y flags que contiene son los siguientes:

- **Protocol version:** indica qué versión exacta del protocolo 802.11 está contenida en el resto de la trama. Actualmente solo hay una versión de 802.11 y se le asigna el valor 0. En el futuro quizá haya nuevas versiones.
- **Type y subtype:** Indica qué tipo de trama es la trama en curso, así como el subtipo.

Los tipos de trama son:

- Management (type = 00)
- Control (type = 01)
- Data (type = 10)

Cada tipo de trama contiene diversos subtipos. Por ejemplo, dentro del tipo “management” encontramos, entre otros, los subtipos

- *Association request (0000)*
- *Association response (0000)*
- *Probe request (0100)*
- *Probe response (0101)*
- *Beacon (1000)*

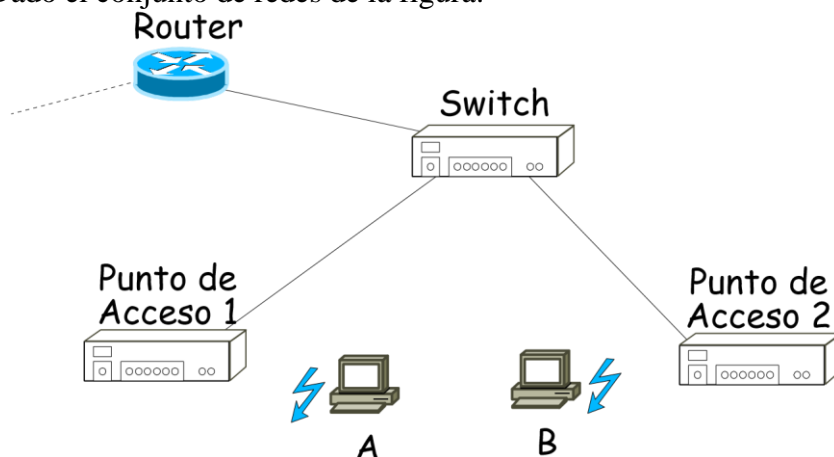
Dentro del tipo “control” están, entre otros, los subtipos:

- *RTS (1011)*
- *CTS (1100)*
- *Acknowledgment (1101)*

- **ToDS y FromDS:** Estos flags se usan principalmente para indicar si la trama en curso viene del punto de acceso (“DS” significa “Distribution System”, que hace referencia a la red cableada), o si va al punto de acceso.
 - ToDS = 1 y FromDS = 0: la trama se origina en la parte inalámbrica y va destinada hacia el punto de acceso
 - ToDS = 0 y FromDS = 1: la trama se transmite desde el punto de acceso y va destinada a una estación inalámbrica
 - Las otras dos combinaciones de estos flags también se usan y tienen su propio significado, pero no los vamos a revisar en esta práctica.
- **More fragments:** Este flag se usa de forma similar al bit “more fragments” de la cabecera del datagrama IP. En este sentido, cuando un mensaje proveniente de la capa superior ha sido fragmentado en varias tramas, todos los fragmentos menos el último llevan este flag a uno.
- **Retry:** Este bit se pone a uno cuando se está retransmitiendo una trama previamente transmitida. Así se ayuda al receptor a eliminar tramas duplicadas.
- **Power management:** Cuando una estación transmite una trama, si pone este bit a uno indica que tras finalizar la transmisión en curso se pondrá en modo de ahorro de energía (y por tanto no recibirá tramas, las cuales tendrán que ser temporalmente almacenadas por el punto de acceso).

- **More data:** Como el punto de acceso tiene que almacenar tramas destinadas a dispositivos que se han puesto en modo de bajo consumo, este bit es usado por el punto de acceso para indicar si tiene tramas almacenadas pendientes de envío.
- **Protected (WEP):** Cuando una trama está cifrada, este bit se pone a uno
- **Order:** Si se requiere que las tramas se transmitan en el orden en que se han generado (o recibido en el punto de acceso), este bit se pone a uno. Nótese que el uso de transmisión en orden requiere usar recursos adicionales tanto en el transmisor como en el receptor.

Ejercicio 1. Dado el conjunto de redes de la figura:



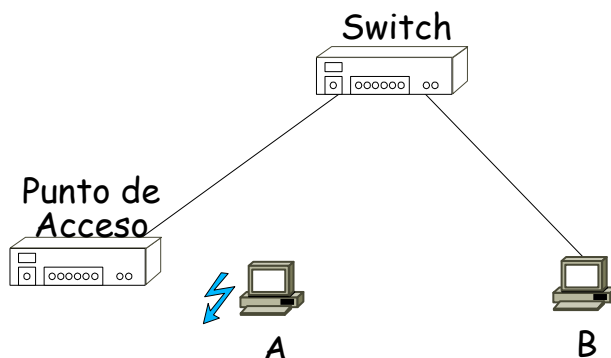
Todas las redes cumplen el estándar IEEE 802.3 o IEEE 802.11. El conmutador (*switch*) conoce la ubicación de todas las máquinas tras un periodo de funcionamiento. El *router* está correctamente configurado. La estación A está asociada al punto de acceso 1, mientras que la estación B está asociada al punto de acceso 2. Las cachés ARP de todos los sistemas están vacías, a excepción de las del router, que contiene todas las direcciones físicas necesarias.

Indica, completando una tabla como la siguiente (usa tantas filas como necesites), la secuencia de tramas que se generarán en el envío de un mensaje ICMP de tipo 8 desde el ordenador A al ordenador B (solamente el envío; no hay que describir las tramas que se generarían en la respuesta). Para las direcciones físicas utiliza los valores simbólicos A, B, PA1, PA2, etc. Para las direcciones IP utiliza los valores simbólicos IP A, IP B, IP R, etc.

Tipo de trama IEEE	Dir. destino/ Dir. 1	Dir. origen/ Dir. 2	Dir. 3	Dir IP origen relacionada (si la hay)	Dir IP destino relacionada (si la hay)	Papel que desempeña la trama	Máquinas (nodos) cuya tarjeta de red recibe copia de la trama

4. Análisis de tráfico. Escenario 1

Para comenzar nuestro estudio del tráfico generado por estaciones inalámbricas, vamos a usar el escenario mostrado en la figura:



En este escenario tenemos dos ordenadores. El ordenador A está asociado al punto de acceso, mientras que el ordenador B tiene una conexión de red cableada. Por otra parte, en lugar de realizar nosotros las capturas de paquetes con el programa wireshark, vamos a usar unas capturas previamente realizadas y que están contenidas en los ficheros **wifi2cable_1** y **wifi2cable_2**. Estos ficheros se encuentran en poliformaT. El motivo para utilizar unas capturas realizadas previamente es que no resulta sencillo capturar tráfico inalámbrico. Para poder llevar a cabo dichas capturas (y ver los campos asociados a IEEE802.11) es necesario que la tarjeta de red inalámbrica pueda operar en modo monitor. Por otra parte, en el laboratorio de redes hay mucho tráfico inalámbrico, lo cual podría complicar sobremanera el análisis de las capturas. Por ello, las capturas que vamos a estudiar se han realizado en un entorno con muy poco tráfico.

Por otra parte, el tráfico a analizar se ha generado ejecutando la orden “**ping -c 1 dir_IP_B**” en el ordenador A, y se ha capturado ejecutando el programa wireshark tanto en el ordenador A como en el ordenador B (de ahí que haya dos ficheros de captura).

Ejercicio 2. Abre desde el wireshark el fichero **wifi2cable_1** y responde a las siguientes cuestiones.

1. De acuerdo con el formato de las tramas que muestra wireshark, ¿se trata de las tramas capturadas en el segmento de red cableado o en el segmento inalámbrico?
2. ¿Cuál es la dirección IP del ordenador A? ¿Cuál es la dirección IP del ordenador B?
3. ¿Por qué se generan las dos primeras tramas de la captura?
4. ¿Cuáles son las direcciones físicas que aparecen en las tramas de la captura? ¿A quiénes corresponden dichas direcciones físicas?
5. ¿Cuál es la longitud de los datagramas enviados? ¿Ha hecho falta fragmentar dichos datagramas debido a un MTU demasiado pequeño? ¿Cuál es la longitud de los datos contenidos en el mensaje ICMP?

Ejercicio 3. Tras analizar en el ejercicio anterior parte del tráfico generado por la orden “**ping -c 1 dir_IP_B**”, en este ejercicio vamos a analizar el resto del tráfico generado. Para ello, abre la captura **wifi2cable_2** (en esta captura se han eliminado todas las tramas que no son consecuencia directa de la orden ping, con el fin de facilitar el análisis del tráfico. La captura completa se puede encontrar en el fichero **wifi2cable_2_completa**).

1. De acuerdo con la información contenida en dicha captura, ¿se trata de las tramas capturadas en el segmento de red cableado o en el segmento inalámbrico?
2. ¿Puedes ver el contenido de los mensajes ARP e ICMP que has visto en la captura anterior? ¿Cuál crees que es la razón por la que no puedes ver el contenido de dichos mensajes?
3. Intenta localizar en esta captura las tramas generadas al ejecutar la orden ping y que complementan el tráfico analizado en el ejercicio anterior.

El motivo por el cual en la captura del ejercicio 3 no se muestran los datos contenidos en las tramas capturadas es porque dichos datos van cifrados de acuerdo con el protocolo WPA-PSK. Para poder ver los mensajes ARP e ICMP que viajan en las tramas de la captura sería necesario disponer de la clave asociada al punto de acceso. Esta clave es del estilo de las que habitualmente se introducen por pantalla cuando nos asociamos desde Windows o Linux a un punto de acceso. En nuestro caso, y dado que el ordenador desde el que se ha hecho la captura ya disponía de dicha clave, basta con ir al menú “Edit”-->”Preferences” y buscar en la ventana que aparece el protocolo “IEEE802.11”. En las opciones que nos permite modificar, buscaremos la que dice “**Ignore the protection bit**” y seleccionaremos “**Yes – with IV**”. A partir de ese momento podemos acceder al contenido de las tramas de la captura.

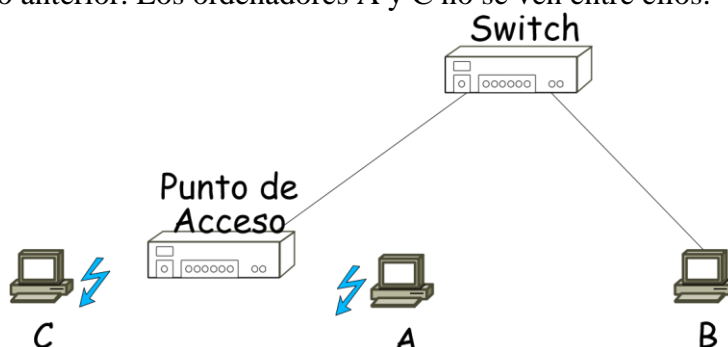
Ejercicio 4. Tras modificar las opciones de cifrado de acuerdo con la descripción anterior, responde a las siguientes cuestiones:

1. Localiza las tramas asociadas al protocolo ARP. ¿Cuántas tramas hay? ¿Hay el mismo número de tramas ARP que en la captura anterior? ¿Por qué?
2. ¿Cuál es la dirección física del punto de acceso? ¿Se trata de la dirección física en la parte cableada o en la parte inalámbrica? ¿Aparecía dicha dirección física en la captura anterior? ¿Por qué?
3. Céntrate en las dos primeras tramas ARP que se envían (tramas 1 y 2 en la captura). Aparentemente son iguales. ¿Qué diferencias observas entre ellas?
4. Analiza los bits “From DS” y “To DS” de ambas tramas. ¿Qué conclusiones extraes? ¿Estas conclusiones son coherentes con las diferencias que has observado en el punto anterior?
5. Busca en la captura la respuesta ARP. ¿Coinciden las direcciones MAC con las de la captura anterior? ¿Son diferentes? ¿Por qué?
6. Centrémonos ahora en la petición y respuesta del ping. ¿Cuáles son las direcciones MAC que intervienen en ambas tramas? Esas direcciones, ¿son coherentes con las direcciones MAC de la captura anterior?
7. ¿Cuáles son las direcciones IP que aparecen en esta captura? ¿Son las mismas que las que aparecían en la captura anterior? ¿Cuál es la dirección IP del punto del acceso?

8. Fíjate en los números de secuencia de las diferentes tramas que se generan como consecuencia de la orden ping (compara la respuesta del ARP y la respuesta del ping). ¿Son consecutivos?
9. De acuerdo con la captura que estás analizando, ¿se han fragmentado los mensajes ICMP? ¿Y los mensajes ARP?

5. Análisis de tráfico. Escenario 2

Vamos a continuar nuestro estudio de las redes inalámbricas introduciendo un ordenador adicional en el escenario de captura. Ahora vamos a tener el escenario de la figura siguiente, donde se ha introducido la estación inalámbrica C, que está asociada al mismo punto de acceso que el ordenador A del escenario anterior. Los ordenadores A y C no se ven entre ellos.



De forma similar al análisis anterior, en este caso vamos a analizar el tráfico generado por la orden “**ping -c 1 dir_IP_C**”, que se ejecuta, de nuevo, en el ordenador A. Para capturar dicho tráfico se ha ejecutado el programa wireshark tanto en el ordenador B como en el ordenador A (de nuevo tenemos dos ficheros de captura: **wifi2wifi_1** y **wifi2wifi_2** (en esta ocasión no se han limpiado los ficheros de captura)).

Ejercicio 5. Abre con el programa wireshark el fichero de captura **wifi2wifi_1** y contesta a las siguientes preguntas:

1. ¿Qué contiene este fichero de captura? ¿Es correcto que contenga la(s) trama(s) que aparecen en la pantalla? ¿Falta alguna trama?
2. Comprueba las direcciones MAC y las direcciones IP que aparecen en el fichero. ¿A quién corresponden estas direcciones? ¿Son las mismas que en el escenario anterior?
3. A partir de la información contenida en esta captura, ¿puedes conocer las direcciones MAC y/o IP de los ordenadores incluidos en el escenario de la figura?

Ejercicio 6. Abre con el programa wireshark el fichero de captura **wifi2wifi_2** y contesta a las siguientes preguntas:

1. ¿En cuántas tramas aparece la dirección MAC del ordenador A? ¿Qué papel desempeñan dichas tramas?
2. Respecto a la primera trama que contiene la dirección MAC del ordenador A como origen de la trama, ¿qué papel desempeña dicha trama? ¿qué información se busca con

dicha trama? ¿Cuáles son las direcciones MAC involucradas en esa primera trama? ¿A quiénes pertenecen esas direcciones MAC?

3. Volviendo sobre la trama anterior, ¿qué direcciones IP aparecen? ¿a quiénes pertenecen dichas direcciones IP?
4. La trama anterior parece que esté repetida en la captura. Analiza el por qué de esta duplicidad.
5. Localiza la trama que se genera como contestación a la trama anterior. ¿Quién transmite dicha trama? ¿Quién es el creador original de dicha trama? ¿Aparece en la captura la trama original transmitida por el creador inicial de dicha trama? ¿Por qué?
6. De acuerdo a la trama anterior, ¿cuál es el valor de la información buscada en la primera trama que envía el ordenador A?
7. Vuelve sobre las tramas analizadas y comprueba los bits “ToDS” y “FromDS” de las mismas. ¿El valor de dichos bits para cada una de las tramas es el esperado?
8. Busca las tramas que contienen el echo request y el echo reply que se transmiten como consecuencia de la orden “ping -c 1 dir_IP_C”. ¿Cual es la dirección MAC que transmite cada una de esas tramas? ¿Y la dirección MAC del creador del mensaje inicial contenido en las tramas? ¿Qué direcciones destino se utilizan? Observa que tanto la dirección MAC del punto de acceso como las de la fuente y destino de las tramas aparecen en la cabecera de la trama en distinto orden dependiendo de que actúen como dirección 1, 2 o 3 (puedes mirar el apartado 3 de la práctica para revisar el formato de las tramas). ¿Quién debe quedarse copia de cada una de las tramas? ¿Serán retransmitidas dichas tramas?
9. Analiza el campo número de secuencia de las tramas siguientes: beacon, ARP y ping. ¿Encuentras alguna relación entre los números de secuencia que usa el punto de acceso?
10. Teniendo en cuenta que las estaciones A y C no se ven entre ellas, pero que ambas ven al punto de acceso, ¿falta alguna trama relacionada con la orden ping en la captura?