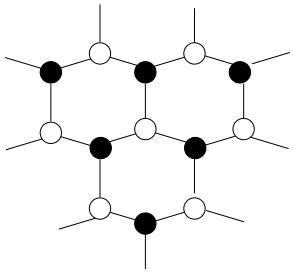


E52. Redes Informáticas

Práctica 6. El nivel de red en Internet: Protocolos IP, ARP e ICMP

Descripción del equipo y del software

1. Ordenadores de tipo PC con S.O. GNU/Linux.
2. Adaptador de red *Ethernet* con conector RJ-45 (estándares 10BaseT/100BaseT).
3. Conmutador 10BaseT.
4. Monitor de red y analizador de protocolos *Ethereal*.

1. Introducción

El conjunto de protocolos TCP/IP, desarrollados en los años 70 por la *Defense Advanced Research Projects Agency* de los EE.UU. y popularizados a raíz de su inclusión en el UNIX de Berkeley, consta de una serie de normas de distintos niveles que tienen como característica común el uso de IP (*Internet Protocol*) para el intercambio de bloques de información, llamados *datagramas*, entre estaciones conectadas a redes de cualquier naturaleza. IP define un servicio *sin conexión* y *no confirmado*, esto es, los datagramas se encaminan entre la estación de origen y la estación de destino sin necesidad de definir previamente una ruta y, además, el protocolo no establece mecanismos de comprobación para determinar con certeza si un bloque de información ha llegado o no a su destino. IP es un protocolo del nivel de red y, como tal, su misión principal es la del *encaminamiento*: determinar la mejor ruta para viajar entre dos estaciones. El camino seguido por un datagrama transmitido sobre una red de área extensa implicará normalmente el paso por una serie de nodos intermedios que son capaces, en función de una dirección de origen y una de destino, de encaminar el datagrama de forma adecuada. El formato del datagrama IP, estructurado en palabras de 32 bits, junto con una descripción completa de sus campos, puede encontrarse en el Anexo II.

Las unidades de datos de los protocolos del nivel de red necesitan encapsularse en unidades de los niveles inferiores y, en última instancia, convertirse en una secuencia de bits para su transmisión. En el caso concreto del envío de un datagrama IP a través de una red de área local como la del laboratorio, la trama de enlace que lo va a transportar ha de contener la dirección física (MAC) de la estación a la cual va dirigido. Para averiguarla, el módulo de adaptación de IP a MAC hace uso del protocolo ARP (*Address Resolution Protocol*). La estación interesada en averiguar la dirección MAC de una estación cuya dirección IP conoce, difunde un paquete ARP con la dirección IP mencionada; todas las estacionas de la red local lo analizan y contesta, mediante otro paquete ARP de respuesta, aquélla que reconoce la dirección IP como propia. Podéis encontrar el formato del paquete ARP y una descripción de sus campos en el Anexo I.

Para llevar a cabo con eficacia la función de encaminamiento, el protocolo IP se sirve a su vez de ICMP (*Internet Control Message Protocol*), que se encarga de enviar mensajes informativos a la estación que ha originado un datagrama cuando éste, por alguna razón, no puede llegar a su destino o se encuentra con cualquier problema en su camino hacia la estación destinataria. El formato del paquete ICMP esta detallado en el Anexo III.

2. Desarrollo de la práctica

2.1. Antes de comenzar...

Para realizar esta práctica hará falta que os agrupéis de la siguiente forma: formad dos grupos de 2-3 personas y tomad 2 ordenadores por cada grupo (a ser posible contiguos). Designad uno de los dos ordenadores como *nodo local* y el otro como *nodo pasarela*. Más adelante veremos la utilidad de esta designación. Los dos grupos trabajarán conjuntamente en el último apartado de la práctica, pero pueden hacerlo por separado en el resto.

En esta práctica vamos a hacer uso intensivo de los programas `ifconfig`, `route` y `ethereal`. Tenéis documentación de éstos en el Anexo IV y en la práctica anterior. También usaremos los programas `arp`, `ping` y `mtr` (consultad su manual respectivo cuando haga falta). A menudo no dispondréis del servicio de resolución de nombres (DNS), así que conviene que referenciéis las estaciones por su dirección IP. También es conveniente que uséis la opción `-n` (o una equivalente) de los programas que la acepten para evitar que hagan uso de ese servicio.

Tened también en cuenta que para modificar la configuración de la red de una máquina o para capturar tramas hará falta que actuéis con permiso de superusuario (*root*). Finalmente, recordaros que es muy aconsejable que vayáis apuntando las órdenes exactas que ejecutéis.

2.2. Recogida de información de los ordenadores

Usad los programas `ifconfig` y `route` para rellenar las tablas siguientes, incluyendo el nombre del dispositivo de red correspondiente (`eth0` o `eth1`). Usad también el programa `host` para encontrar la dirección IP correspondiente a Anubis (`anubis.uji.es`) y apuntadla.

	Nodo pasarela	Nodo local
Dirección MAC de la interfaz pública (`eth_`)		
Dirección IPv4 de la interfaz pública		
Dirección MAC de la interfaz privada (`eth_`)		
Dirección IPv4 de la interfaz privada		

Máscara de subred pública	
Máscara de subred privada	
Dirección IPv4 del encaminador por defecto	
Dirección IPv4 de `anubis.uji.es`	

2.3. Estudio de la red del laboratorio

Vamos a estudiar la estructura de la red de la UJI y del laboratorio de Sistemas Operativos y Redes (TD1108DL). Con la información anterior en mano, contestad a las preguntas siguientes:

- ¿Cuál es la dirección IP de la red de la UJI? ¿De qué clase (A, B, C...) es?
- ¿Cuál es su máscara por defecto? ¿Cuántas estaciones sería posible conectar con esta máscara?
- ¿Cuál es la máscara de subred que se usa en la red de la UJI? ¿En cuántas subredes queda dividida esta última? ¿Cuántas estaciones se pueden conectar a cada una de ellas?
- ¿Se encuentran los dos nodos de vuestro grupo en la misma subred, si tenéis en cuenta sus direcciones IP privadas (192.168.X.Y)? ¿Por qué?
- ¿Se encuentran los dos nodos de vuestro grupo en la misma subred, si tenéis en cuenta sus direcciones IP públicas (150.128.X.Y)? ¿Por qué?
- Escoged uno de los nodos. ¿Se encuentra en la misma subred que Anubis, si tenéis en cuenta sus direcciones IP públicas (150.128.X.Y)? ¿Por qué?

2.4. Redes locales y ARP

Antes de continuar usad `ifconfig` en los dos nodos para inhabilitar las interfaces de red públicas y privadas. Esto dejará incomunicadas vuestras máquinas. Comprovad que es realmente así usando `route` para ver las tablas de rutas actuales.

Ahora configuraréis una nueva subred privada con los dos nodos de vuestro grupo. La subred en cuestión será 192.168.X.0, donde X será el valor del último byte de la dirección IP privada que tenía vuestro nodo pasarela; la máscara de subred será 255.255.255.0. E.g. si el nodo pasarela era 192.168.49.182 la subred será 192.168.182.0/255.255.255.0. Asignad una nueva dirección privada dentro de la nueva subred a cada nodo y rellenad estas tablas:

	Nodo pasarela	Nodo local
Nueva dirección IP de la interfaz privada (`eth_`)		

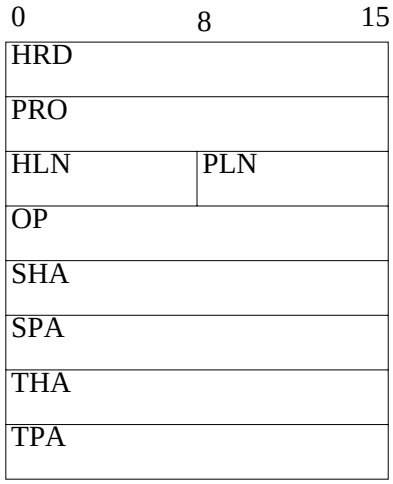
Nueva dirección de subred privada	
Nueva máscara de subred privada	

- Usad `ifconfig` en los dos nodos de vuestro grupo para asignar a la interfaz privada de cada uno su nueva dirección. Comprovad que la configuración funciona haciendo *ping* entre las dos máquinas.
- Si usáis `route` veréis que se ha añadido automáticamente una ruta hacia la nueva subred. ¿Cuál sería la orden exacta para crear esta ruta manualmente? Podéis comprobarla usando `route` para eliminar la ruta y volverla a activar después.

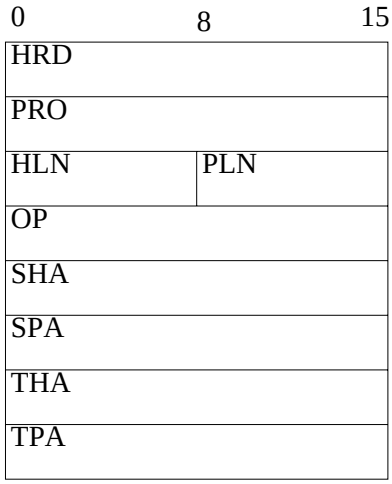
Vamos a comprobar el funcionamento del protocolo ARP. Si usáis `arp` veréis que la tabla ARP del núcleo no está vacía. Para partir con la tabla vacía, inhabilitad y reahabilitad la interfaz privada de los dos nodos del grupo con las órdenes que habéis usado antes, pero **no intercambiéis ningún paquete entre ellos**.

- Lanzad Ethereal como *root* en un nodo y en *Capture/Start* estableced un filtro para capturar únicamente los paquetes ARP originados en, o destinados a la dirección MAC privada de este nodo. Iniciad la captura.

- Lanzad un solo paquete de *ping* hacia la IP privada del otro nodo y parad la captura.
- ¿Cuál es la dirección de destino del primer paquete originado por este nodo? ¿Por qué? ¿Por qué tienen diferente tamaño las tramas ARP enviadas por este nodo y las recibidas desde el otro nodo?
- Elegid un paquete ARP de interrogación y otro de respuesta y rellenad, de forma inteligible, los campos correspondientes a las dos cabeceras ARP siguientes:



Interrogación

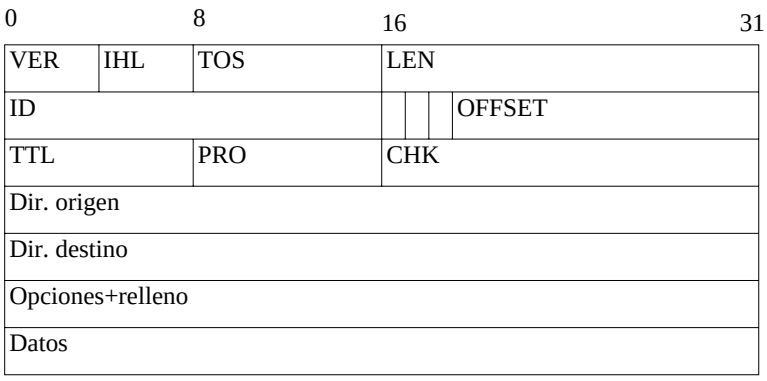


Respuesta

2.5. IP e ICMP

Vamos a analizar el funcionamento básico de IP en una red local. Para ello:

- Escoged un nodo y estableced en él con Ethereal un filtro para capturar únicamente los datagramas IP originados en, o destinados a la dirección IP privada de éste. Iniciad la captura.
- Lanzad un solo paquete de *ping* hacia la IP privada del otro nodo.
- Lanzad un solo paquete de *ping* con un tamaño de 2040 bytes de datos hacia la IP privada del otro nodo (es necesario que lo hagaís como *root*). Parad la captura.
- Analizad el datagrama de la primera petición de *ping*. Estudiad los campos de la cabecera IP y rellenad, de forma inteligible, el esquema siguiente:



- Estudiad las cabeceras IP de las dos peticiones de *ping*. ¿Se ha fragmentado el primer *ping*? ¿Y el segundo? ¿Cómo lo sabéis?
- Responded: ¿cómo están relacionados los datagramas pertenecientes al *ping* fragmentado? ¿Cómo se sabe que el datagrama del otro *ping* no tiene que ver con estos fragmentos?

Ahora estudiaremos por encima algunos detalles de los paquetes ICMP usados por `ping`: las peticiones de eco y las respuestas de eco.

- Escoged un nodo y estableced en él con Ethereal un filtro para capturar únicamente los paquetes ICMP originados en, o destinados a la dirección IP privada de éste. Iniciad la captura.
- Lanzad unos pocos paquetes de *ping* hacia la IP privada del otro nodo. Una vez hecho esto, volved a lanzar la misma orden. Finalmente, parad la captura.
- Estudiad los campos de las cabeceras ICMP de los paquetes capturados. ¿En qué coinciden las peticiones de eco de una misma ejecución de `ping`? ¿En qué se diferencian? ¿Y en qué se diferencian las peticiones de eco de dos ejecuciones diferentes?

2.6. Encaminamiento IP

Para realizar este apartado será necesario que trabajéis *los dos grupos de forma coordinada*. Vamos a dar a los nodos pasarela acceso al exterior y a la subred del otro grupo, de forma que se puedan intercambiar paquetes entre ellas. Podéis comprobar que esto no es posible ahora haciendo un *ping* desde un nodo de una subred hacia un nodo de la otra.

- Usad `ifconfig` en los dos nodos pasarela para asignar a la interfaz pública de cada uno su dirección y máscara públicas originales. Haced *ping* entre las dos máquinas (usando la IP pública) para comprobar su funcionamento. Con esto tenemos acceso a la subred pública del laboratorio.
- Para poder acceder al exterior se necesitará configurar un *encaminador por defecto* en los dos nodos pasarela. Usad `route` para establecer en ellos el encaminador por defecto que tenían en un principio. Comprobad que esto funciona haciendo un *ping* a Anubis.
- Para intercomunicar ambas subredes usaremos la red pública del laboratorio. Añadid con `route` en cada nodo pasarela una ruta hacia la subred del otro grupo a través de la dirección IP pública de su nodo pasarela. Haced *ping* a la IP privada del otro nodo pasarela para ver si lo habéis hecho bien.
- Usad `route` en los nodos locales para establecer en ellos como encaminador por defecto el nodo pasarela del mismo grupo (mediante su dirección IP privada). Si podéis hacer *ping* entre los nodos locales, lo habéis hecho bien.

Una vez lo tengáis todo listo, cada grupo deberá realizar las experiencias siguientes y contestar las cuestiones respectivas:

- Ejecutad `mtr` desde el nodo local hacia las siguientes direcciones IP y apuntad por qué nodos pasa cada trazado:

- La privada de vuestro nodo pasarela.
- La pública del nodo pasarela del otro grupo.
- La privada del nodo pasarela del otro grupo.
- La privada del nodo local del otro grupo.
- La pública de Anubis.

¿Llegan todos los trazados a su destino? ¿Por qué suponéis que es así?

- Repetid la experiencia anterior desde el nodo pasarela. En el primer caso substituid el nodo pasarela por el nodo local. ¿Llegan todos los trazados? ¿Se corresponde con lo que suponíais antes? ¿Cuál es el primer nodo por donde pasa el trazado hacia Anubis? ¿Conocíais ya este nodo?
- Estableced con Ethereal en el nodo pasarela una captura de los paquetes ICMP originados en, o destinados a la dirección IP privada del nodo. A continuación haced un *ping* hacia vuestro nodo local. ¿A quién suponéis que pertenece la dirección MAC de destino de los paquetes emitidos?
- Estableced con Ethereal en el nodo pasarela una captura de los paquetes ICMP originados en, o destinados a la dirección IP pública del nodo. Haced un *ping* hacia Anubis. ¿A quién suponéis que pertenece ahora la dirección MAC de destino de los paquetes emitidos?
- Comprobad si vuestras suposiciones eran acertadas usando el programa `arp` para ver las direcciones IP correspondientes a las direcciones MAC anteriores.

3. Conclusiones

- ¿Qué estrategia se sigue en la UJI para tener varias subredes partiendo de la asignación de la red IP única 150.128.0.0?
- A la vista de los paquetes de datos que habéis analizado, ¿diríais que los protocolos ARP e ICMP son del mismo nivel? ¿Diríais que ARP e IP son del mismo nivel, basándoos en su situación en la pila de protocolos? ¿Y basándoos en su funcionalidad?
- ¿Seríais capaces de dibujar un esquema que comprenda la topología final de vuestras subredes (incluyendo a cada nodo y sus interfaces y direcciones IP) y la red de la UJI (incluyendo a Anubis)?

Anexo I. Formato del paquete ARP

La estructura del paquete ARP se muestra en la figura 1. Los campos sombreados tienen longitud variable, que depende de los valores de los campos HLN y PLN. Los campos del paquete tienen el significado siguiente:

HRD Tipo de dirección física o de hardware (p.e. Ethernet).

PRO Tipo de dirección del nivel de red o de protocolo (p.e. IP).

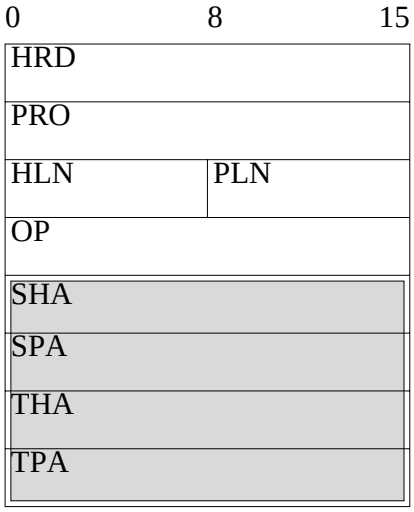


Figura 1: Estructura de un paquete ARP.

HLN Longitud de la dirección física, en bytes.

PLN Longitud de la dirección del nivel de red, en bytes.

OP Código de operación (`REQUEST` o `REPLY`).

SHA Dirección física del emisor.

SPA Dirección del nivel de red del emisor.

THA Dirección física del destinatario (a 0 si `OP=REQUEST`).

TPA Dirección del nivel de red del destinatario.

Anexo II. Formato del datagrama IP

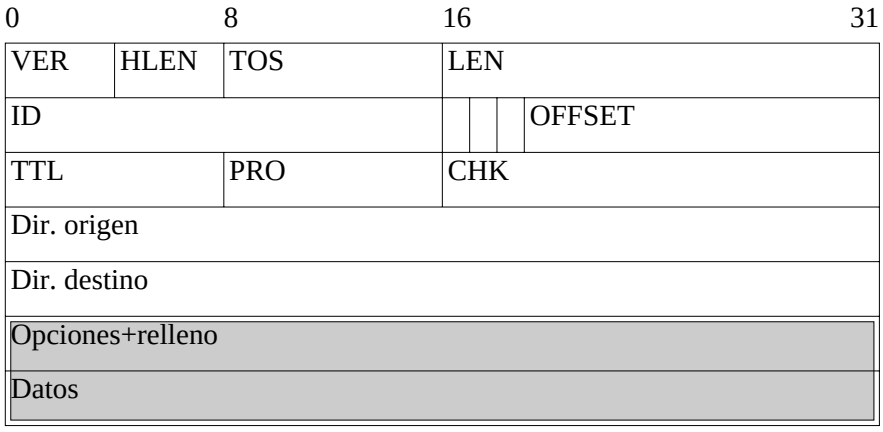


Figura 2: Estructura de un datagrama IP.

La estructura del datagrama IP se muestra en la figura 2. Los campos sombreados en la figura tienen longitud variable. Los campos del datagrama tienen el significado siguiente:

VER Versión del protocolo IP.

IHL (*IP Header Length*) Indica la longitud, en palabras de 32 bits, de la cabecera del datagrama. Su valor mínimo es 5 (20 bytes).

TOS (*Type Of Service*) Indica el tipo de servicio que la estación espera de la subred. Suele ser un compromiso entre fiabilidad y velocidad; la forma de solicitar un servicio determinado por un datagrama es especificándolo según el esquema siguiente:

P1	P2	P3	D	T	R		

Los tres primeros bits codifican la prioridad del datagrama (000 es la mínima, 111 la máxima); el bit D puesto a 1 indica que se debe minimizar el retardo en su transmisión; el bit T, que hay que tratar de elegir las rutas de mayor ancho de banda; y el R, que hay que buscar la máxima fiabilidad. Los dos últimos bits están reservados. Estas opciones están contempladas por el protocolo IP, aunque rara vez hacen uso de ellas los protocolos usuarios.

LEN Es la medida, en bytes, del tamaño total (cabecera más datos) del datagrama. El tamaño máximo del datagrama es, por tanto, de 65536 bytes.

ID Entero que utiliza el emisor para identificar cada uno de sus datagramas.

FLAGS Este campo y el siguiente sirven para controlar la fragmentación que se puede producir en en tráfico entre redes. El campo de marcadores (*flags*) está formado por tres bits, de los cuales el primero no se usa y se transmite a 0; el siguiente es el bit DF (*Don't Fragment*), que se pone a 1 para indicar que el datagrama no puede ser fragmentado; el tercero es el bit MF (*More Fragments*), que indica, cuando vale 0, que el fragmento es el último de un datagrama original.

OFFSET Posición relativa de un fragmento dentro del datagrama original. Se mide en unidades de 8 bytes. Si un datagrama no puede ser fragmentado, DF=1, MF=0 y OFFSET=0.

TTL (*Time-To-Live*) Mantiene un contador que es decrementado por cada entidad que procesa la cabecera, de forma que cuando una entidad IP detecta un datagrama con TTL=0, lo descarta.

PRO Identifica el protocolo del nivel superior que debe recibir el datagrama en la estación de destino. Los valores más comunes son 6 para TCP, 17 para UDP y 1 para ICMP.

CHK Se usa para detectar errores en la transmisión de la cabecera. No se realiza ninguna comprobación sobre el campo de datos. El valor que se almacena es el complemento a uno de la suma (usando aritmética de complemento a uno y tomando el campo CHK como 0) de todas las palabras de 16 bits de la cabecera.

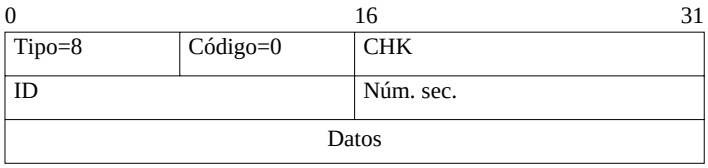
Adreces Los cuatro bytes de las direcciones IP de origen y destino del datagrama.

Opcions Es un campo de longitud variable y opcional en los datagramas, aunque cualquier entidad IP debe implementar la funcionalidad requerida. Un datagrama puede incluir más de una opción. Cada una de éstas vendrá identificada por un campo de código de opción (de un byte) que, en algunos casos, irá seguido de un campo de longitud (de un byte) y éste, a su vez, determinará el tamaño de otro campo que contendrá los datos específicos de la opción.

Anexo III. Formato del paquete ICMP

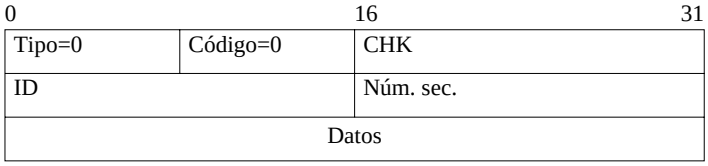
El formato del paquete ICMP depende de cada uno de los mensajes que se envíe para comunicar situaciones especiales o anómalas que se hayan podido producir a lo largo de la ruta que sigue un datagrama. Este protocolo hace uso de IP para la transmisión de sus bloques de información. El primer byte del paquete indica el tipo de mensaje, el segundo un código de información adicional, los bytes tercero y cuarto contienen la suma de comprobación (*checksum*) de ICMP, que se calcula de forma idéntica a la de la cabecera del datagrama IP. Los tipos de mensaje más comunes y sus formatos correspondientes se describen a continuación.

Echo request (Solicitud de eco)



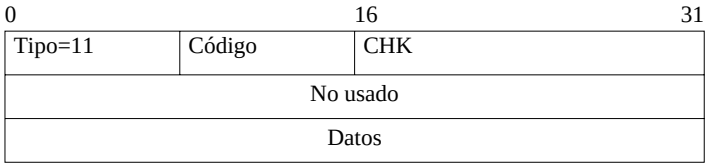
Es una petición de eco de una entidad ICMP a otra entidad remota del mismo protocolo. Una petición de eco puede constar de varios paquetes; en este caso el campo *Identificador* (ID) indica la petición, mientras que el campo *Número de secuencia* identifica los paquetes individuales. La respuesta será un paquete del tipo 0 (respuesta de eco). El campo de datos se rellena con caracteres arbitrarios.

Echo reply (Respuesta de eco)



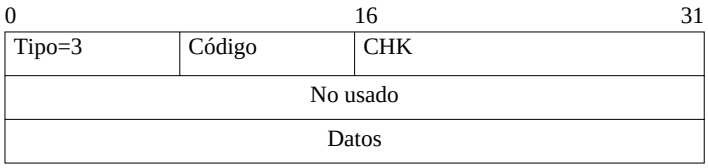
Es una respuesta a una petición de eco. Una entidad ICMP envía a otra una solicitud, y ésta última le responde con un paquete de estas características. Los campos *Identificador* y *Número de secuencia* son los descritos en el paquete de solicitud. En el campo de datos se retorna la misma información que llevara el paquete de solicitud.

Time Exceeded (Tiempo de vida excedido)



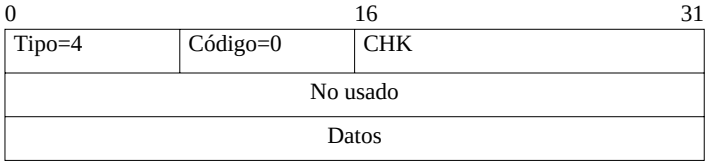
Cuando un encaminador IP detecta un datagrama con el campo TTL igual a 0, debe descartarlo. Este mensaje sirve para informar a la entidad IP emisora de este hecho. En el caso de que un datagrama haya sido fragmentado, este mensaje puede ser generado por el hecho de que el desinatario no haya recibido todos los fragmentos y no pueda, por tanto, reconstruir el datagrama original en un tiempo prefijado. El primer problema se indica con un valor 0 en el campo de *Código*, mientras que el segundo se indica con un valor 1. El campo de *Datos* contiene la cabecera IP básica y los primeros 64 bits de datos del datagrama que ha originado el error.

Destination Unreachable (Destino inalcanzable)



Se envía al emisor de un datagrama cuando un encaminador no puede hacerlo llegar a su destino por cualquier razón. El campo de *Código* indica el tipo de problema; por ejemplo, 0 si la red no es alcanzable, 1 si la estación no es alcanzable, 2 si el protocolo no es alcanzable, etc. El campo de *Datos* contiene la cabecera IP básica y los primeros 64 bits de datos del datagrama que ha originado el error.

Source Quench (Detener el origen)



Se envía al emisor de un datagrama cuando la entidad IP destinataria no tiene tiempo de procesar la información que le llega. Es una petición para que el origen disminuya la frecuencia de sus envíos. El campo de *Datos* contiene la cabecera IP básica y los primeros 64 bits de datos del datagrama que ha originado el error.

Anexo IV. Los programas `ifconfig` y `route`

Configuración de las interfaces: `ifconfig`

El programa `ifconfig(8)` se utiliza para manipular la configuración de las interfaces de red del núcleo Linux, activándolas, desactivándolas y asignándoles direcciones de varios protocolos de red. Veamos la sintaxis básica que usaremos en esta práctica (orientada a IP; podéis obtener más información en la página correspondiente del manual):

Consulta de la configuración de las interfaces

Sintaxis:
```
ifconfig
ifconfig <interfaz>
```

La primera forma muestra la configuración de todas las interfaces activas del núcleo, la segunda la de una interfaz concreta. Esta información incluye (entre otros): el tipo de dispositivo,

la dirección física, las direcciones asignadas de los diferentes protocolos de red (en el caso de IP incluye la dirección de difusión o *broadcast* y la máscara de subred asociadas), los marcadores de funcionamento, la unidad máxima de transferencia (MTU) y la métrica (o *distancia* hasta la red). Por ejemplo, `ifconfig eth0` muestra la configuración del primer dispositivo Ethernet.

Asignación de direcciones a una interfaz

Sintaxis: `ifconfig <interfaz><dirección>[netmask <máscara de subred>]`
Esta orden asocia la `dirección` IP especificada a la `interfaz` indicada, de forma que (i) los paquetes IP que se envíen por la interfaz lleven como dirección de origen esta dirección IP (ii) se establece el mecanismo ARP sobre la interfaz para responder con su dirección física cuando se detecten en su enlace peticiones ARP de la dirección IP configurada.

El argumento opcional `netmask` se utiliza para fijar una `máscara de subred` diferente a la que corresponde por defecto a la clase de la `dirección`. Con la dirección y la máscara de subred, el sistema es capaz de añadir automáticamente una ruta a la subred correspondiente[1]. Así, para asignar la dirección 10.0.1.2 a la interfaz `eth1`, con máscara 255.255.255.0 (diferente a la máscara por defecto, 255.0.0.0) ejecutaríamos `ifconfig eth1 10.0.1.2 netmask 255.255.255.0`, que añadiría de paso una ruta a la red 10.0.1.0/255.255.255.0 a través de la interfaz `eth1`.

Desactivación de una interfaz

Sintaxis: `ifconfig <interfaz>down`
Esta es la forma de inhabilitar una interfaz determinada. Al hacer esto todas las direcciones asociadas a ella dejan de estar activas. Además, las rutas que usan esta interfaz son eliminadas, así como todas las parejas de direcciones de red y enlace resueltas vía ARP sobre la interfaz.

Manipulación de las rutas: `route`

El programa `route(8)` se usa para configurar las rutas estáticas del núcleo hacia las redes IP, ya sea directamente a través de interfaces (redes locales) o indirectamente a través de encaminadores (redes no locales). Ésta es la sintaxis de las órdenes que usaremos en esta práctica:

Consulta de las rutas

Sintaxis: `route [-n]`
Si ejecutamos `route` sin argumentos, se nos mostrará una tabla con las rutas IP actuales del núcleo. Por cada línea (ruta) veremos, entre otros, a qué red lleva, la máscara de subred, y a través de qué encaminador o interfaz se llega a ella. La red especial 0.0.0.0 o `default` indica una ruta por defecto, y el encaminador 0.0.0.0 o `*` que el acceso a la red en cuestión es directo (red local). La opción `-n` evita la resolución de los nombres de las redes y encaminadores.

[1] La mayoría de las veces esta ruta será la deseada, pero habrá casos donde no, así que no está de más comprobar si ésta es correcta con `route`.

Adición de una ruta directa a red

Sintaxis: `route add -net <red>netmask <máscara de subred>`
`dev <interfaz>`

Esta orden añade una ruta directa hacia la `red` especificada con la `máscara de subred` correspondiente. La red es local y se encuentra accesible directamente en la `interfaz` nombrada. Se pueden añadir varias rutas a una misma red, siempre que no usen la misma salida (interfaz o encaminador). Por ejemplo, la orden `route add -net 10.0.1.0 netmask 255.255.255.0 dev eth1` añadiría la ruta automática nombrada en el ejemplo de asignación de direcciones con `ifconfig`.

Adición de una ruta indirecta a red

Sintaxis: `route add -net <red>netmask <máscara de subred>`
`gw <encaminador>`
`route add default gw <encaminador>`

Esta orden añade una ruta indirecta hacia la red especificada a través del `encaminador` indicado, que ha de ser alcanzable en alguna red local. En el primer caso establecemos una ruta hacia una `red` determinada con la correspondiente `máscara de subred`. En el segundo caso se establece una *ruta por defecto*, es decir, una ruta por donde enviar los paquetes para los que no exista ninguna otra ruta mejor. En ambos casos se encuentra la interfaz adecuada hacia el encaminador automáticamente usando el resto de las rutas.

Por ejemplo, `route add -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.1.1` establecería 10.0.1.1 como encaminador para llegar a la red 10.0.2.0/255.255.255.0.

Eliminación de una ruta a red

Para eliminar una ruta sólo hace falta ejecutar la misma orden que habríamos invocado para añadirla, pero sustituyendo en ella el argumento `add` por `del`. De esta forma, para eliminar la ruta del ejemplo anterior escribiríamos `route del -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.1.1`.