

En esta Unidad aprenderás a:

- 1 Conocer la estructura de la pila de protocolos TCP/IP.
- 2 Analizar el funcionamiento de las redes basadas en el protocolo TCP/IP.
- 2 Resolver problemas mediante los comandos y las utilidades de diagnóstico más conocidas.
- 4 Configurar los parámetros y servicios básicos en distintos *routers*.
- 5 Comprender el funcionamiento de los elementos implicados en las técnicas de enrutamiento.





5.1 Introducción al protocolo TCP/IP

Las siglas TCP/IP se refieren a un conjunto de protocolos para comunicaciones de datos. Este conjunto toma su nombre de dos de sus protocolos más importantes, el protocolo **TCP** (*Transmission Control Protocol*) y el protocolo **IP** (*Internet Protocol*).

La evolución del protocolo TCP/IP siempre ha estado muy ligada a la de Internet. En 1969 la agencia de proyectos de investigación avanzada, **ARPA** (*Advanced Research Projects Agency*) desarrolló un proyecto experimental de red conmutada de paquetes al que denominó **ARPAnet**.

ARPAnet comenzó a ser operativa en 1975, pasando entonces a ser administrada por el ejército de los EEUU. En estas circunstancias se desarrolla el primer conjunto básico de protocolos TCP/IP. Posteriormente, y ya entrados en la década de los ochenta, todos los equipos militares conectados a la red adoptan el protocolo TCP/IP y se comienza a implementar también en los sistemas Unix. Poco a poco ARPAnet deja de tener un uso exclusivamente militar, y se permite que centros de investigación, universidades y empresas se conecten a esta red. Se habla cada vez con más fuerza de **Internet** y en 1990 ARPAnet deja de existir oficialmente.

En los años sucesivos y hasta nuestros días las redes troncales y los nodos de interconexión han aumentado de forma imparable. La red Internet parece expandirse sin límite, aunque manteniendo siempre una constante: el protocolo TCP/IP. En efecto, el gran crecimiento de Internet ha logrado que el protocolo TCP/IP sea el

estándar en todo tipo de aplicaciones telemáticas, incluidas las redes locales y corporativas. Y es precisamente en este ámbito, conocido como **Intranet**, donde TCP/IP adquiere cada día un mayor protagonismo.

La popularidad del protocolo TCP/IP no se debe tanto a Internet como a una serie de características que responden a las necesidades actuales de transmisión de datos en todo el mundo, entre las cuales destacan las siguientes:

- Los estándares del protocolo TCP/IP son abiertos y ampliamente soportados por todo tipo de sistemas, es decir, se puede disponer libremente de ellos y son desarrollados independientemente del hardware de los ordenadores o de los sistemas operativos.
- TCP/IP funciona prácticamente sobre cualquier tipo de medio, no importa si es una red Ethernet, una conexión ADSL o una fibra óptica.
- TCP/IP emplea un esquema de direccionamiento que asigna a cada equipo conectado una dirección única en toda la red, aunque la red sea tan extensa como Internet.

La naturaleza abierta del conjunto de protocolos TCP/IP requiere de estándares de referencia disponibles en documentos de acceso público. Actualmente todos los estándares descritos para los protocolos TCP/IP son publicados como **RFC** (*Requests for Comments*) que detallan lo relacionado con la tecnología de la que se sirve Internet: protocolos, recomendaciones, comunicaciones, etcétera.



Pueden consultarse más de tres mil RFCs en la página web www.rfc-editor.org. El RFC Editor es un grupo fundado por la Sociedad Internet (www.isoc.org/esp). Por fortuna los estándares RFC pueden ser leídos en español gracias al trabajo desinteresado del Grupo de Traducción al castellano de RFC (www.rfc-es.org).

5.2 Arquitectura del protocolo TCP/IP

El protocolo TCP/IP fue creado antes que el modelo de capas OSI, así que los niveles del protocolo TCP/IP no coinciden exactamente con los siete que establece el OSI.

Existen descripciones del protocolo TCP/IP que definen de tres a cinco niveles. La Figura 5.1 representa un modelo de cuatro capas TCP/IP y su correspondencia con el modelo de referencia OSI.

Los datos que son enviados a la red recorren la pila del protocolo TCP/IP desde la capa más alta de aplicación hasta la más baja de acceso a red. Cuando son recibidos, recorren la pila de protocolo en el sentido contrario.

Durante estos recorridos, cada capa añade o sustrae cierta información de control a los datos para garantizar su correcta transmisión, como ya hemos visto en la Unidad 4.

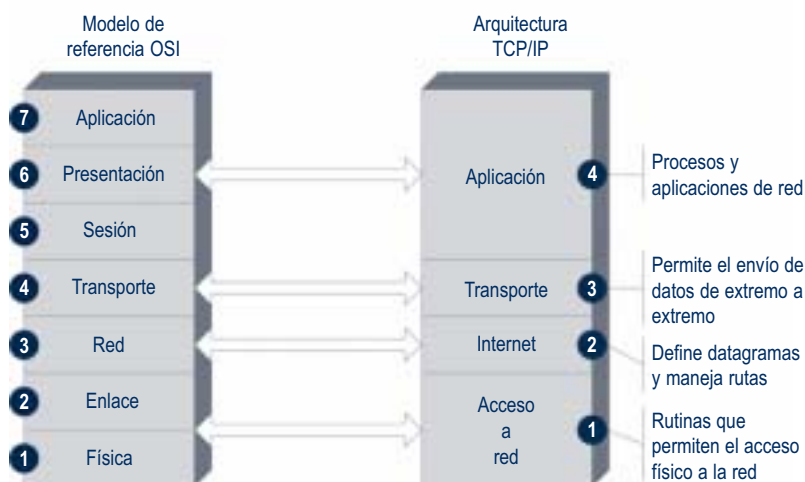


Fig. 5.1. Correspondencia del modelo OSI con TCP/IP.



5. Protocolo TCP/IP

5.2 Arquitectura del protocolo TCP/IP

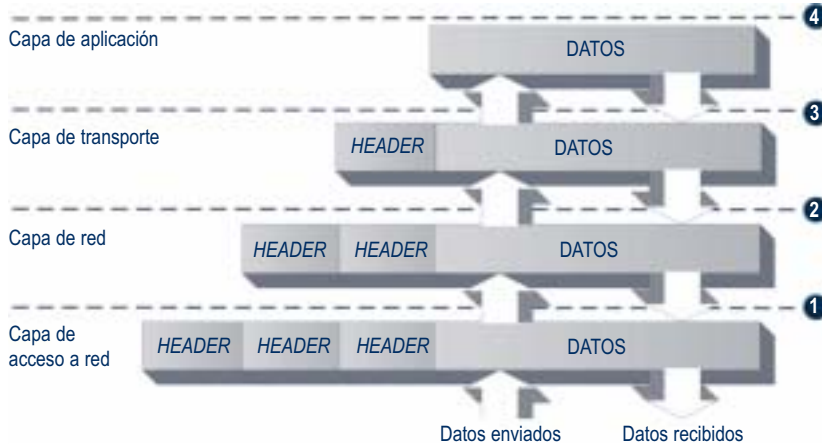


Fig. 5.2. Encapsulado de datos por los niveles TCP/IP.

Como esta información de control se sitúa antes de los datos que se transmiten, se llama cabecera (*header*). En la Figura 5.2 se puede ver cómo cada capa añade una cabecera a los datos que se envían a la red. Este proceso se conoce como **encapsulado**.

Si en vez de transmitir datos se trata de recibirlos, el proceso sucede al revés. Cada capa elimina su cabecera correspondiente hasta que quedan sólo los datos.

En teoría cada capa maneja una estructura de datos propia, independiente de las demás, aunque en la práctica estas estructuras de datos se diseñan para que sean compatibles con las de las capas adyacentes. Se mejora así la eficiencia global en la transmisión de datos.

A. Capa de acceso a red

Dentro de la jerarquía del protocolo TCP/IP la capa de acceso a red se encuentra en el nivel más bajo. Es en esta capa donde se define cómo encapsular un datagrama IP en una trama que pueda ser transmitida por la red, siendo en una inmensa mayoría de redes LAN una trama Ethernet.

Otra función importante de esta capa es la de asociar las direcciones lógicas IP a direcciones físicas de los dispositivos adaptadores de red (NIC). Por ejemplo: la dirección IP 192.168.1.5 de un ordenador se asocia a la dirección Ethernet 00-0C-6E-2B-49-65. La primera es elegida por el usuario (e, incluso, un mismo ordenador puede trabajar con diferentes direcciones IP). Sin embargo la segunda no puede cambiarse e identifica inequívocamente al adaptador NIC dentro de la red Ethernet.

Dentro de la capa de acceso a red opera el protocolo **ARP** (*Address Resolution Protocol*), que se encarga precisamente de asociar direcciones IP con direcciones físicas Ethernet. El estándar RFC 826 describe su funcionamiento.

Existe otra recomendación: la RFC 894 es el estándar para la transmisión de datagramas IP sobre redes Ethernet. Especifica cómo se encapsulan datagramas del protocolo IP para que puedan transmitirse en una red Ethernet.

B. Capa de red: Internet

La capa Internet se encuentra justo encima de la capa de acceso a red. En este nivel el protocolo IP es el gran protagonista. Existen varias versiones del protocolo IP: IPv4 es en la actualidad la más empleada, aunque el crecimiento exponencial en el tamaño de las redes compromete cada vez más su operatividad. El número de equipos que IPv4 puede direccionar comienza a quedarse corto. Para poner remedio a esta situación se ha desarrollado la versión IPv6, con una capacidad de direccionamiento muy superior a IPv4, pero totalmente incompatible.

El protocolo IP se ha diseñado para redes de paquetes conmutados no orientadas a conexión, lo cual quiere decir que cuando dos equipos quieren conectarse entre sí no intercambian información para establecer la sesión. IP tampoco se encarga de comprobar si se han producido errores de transmisión, confía esta función a las capas superiores. Todo ello se traduce en que los paquetes de datos contienen información suficiente como para propagarse a través de la red sin que haga falta establecer conexiones permanentes.

Para el protocolo IP un datagrama es el formato que debe tener un paquete de datos en la capa de red. La Figura 5.3 representa la estructura de un datagrama: muestra las seis primeras palabras de la cabecera y el punto desde el que se comienzan a transmitir los datos.

Las cinco (o seis) primeras palabras de 32 bits contienen la información necesaria para que el datagrama se propague por la red, y a continuación se adjuntan los datos. La lógica de funcionamiento del protocolo IP es simple: para cada datagrama consulta la dirección origen (palabra 4) y la compara con la dirección destino (palabra 5). Si resulta que origen y destino se corresponden con equipos (*hosts*) de la misma red, el datagrama se envía directamente de un equipo a otro. Si, por el contrario, los equipos pertenecen a redes distintas, se hace necesaria la intervención de una puerta de **enlace** o **gateway** que facilite el envío a redes diferentes.

El paso de datos de una red a otra a través de una puerta de enlace es conocido como «salto» (*hop*). Un datagrama puede realizar varios saltos a través de diversas redes hasta alcanzar su destino. El camino que siguen los datos enviados por un equipo a otro no tiene por qué ser siempre el mismo. La búsqueda del camino más adecuado a cada momento se denomina **enrutamiento**. De hecho, a las puertas de enlace se les denomina enrutadores (*routers*).



En la Figura 5.4 vemos un ejemplo elemental de dos redes unidas por un *router*. En una el equipo **A1** envía un datagrama **Z** al equipo **A2**. Como ambos pertenecen a la misma red 192.168.10.0, el datagrama **Z** es enviado directamente.

Cuando el equipo A1 pretende enviar otro datagrama Y al equipo B2 que se encuentra en otra red, resulta imprescindible la ayuda del *router*. Veamos esto con más detalle: como la dirección destino del datagrama Y no se encuentra en la red de origen, se envía directamente a la dirección de la puerta de enlace 192.168.10.3, que es uno de los interfaces del *router*. Es ahora este dispositivo quien decide qué camino debe seguir el datagrama Y. Para ello consulta sus tablas internas, y comprueba que la dirección destino coincide con la red en la que tiene conectado su interfaz con dirección 192.168.1.20. Una vez realizada esta consulta ya puede enviar el datagrama Y desde este puerto directamente al equipo B2 (192.168.1.22). El datagrama Y ha llegado a su destino realizando un salto entre redes.

Protocolo ICMP

En la misma capa de red en la que opera el protocolo IP tenemos también el importante protocolo ICMP (*Internet Control Message Protocol*), definido por la RFC 792.

ICMP envía mensajes en forma de datagramas que permiten al conjunto del protocolo TCP/IP realizar entre otras las siguientes funciones:

- **Control de flujo.** Si los datagramas llegan muy deprisa al host destino y éste se encuentra con dificultades para procesarlos, el host destino envía al host origen un mensaje ICMP solicitando que de forma temporal detenga su emisión.
- **Detección de destinos inalcanzables.** Cuando la dirección destino de un datagrama no logra ser asociada a ningún equipo, el host que la ha enviado recibe un mensaje ICMP indicando que el destino indicado es inalcanzable.
- **Redireccionamiento de rutas.** Una puerta de enlace puede enviar un mensaje ICMP a un *host* para hacerle saber que existe otra puerta de enlace dentro de la misma red, y que en ese momento resulta una mejor opción para encaminar sus datagramas hacia otras redes.
- **Pruebas de conectividad.** Esta funcionalidad es la más conocida ya que es la que emplea el comando `ping` (*Packet Internet Groper*). Desde un equipo se puede enviar a otro un mensaje ICMP «con eco». ¿Qué significa «con eco»? Pues quiere decir que cuando el *host* destino recibe el mensaje lo devuelve inmediatamente al *host* origen.



Fig. 5.3. Representación de la cabecera de una datagrama IP.

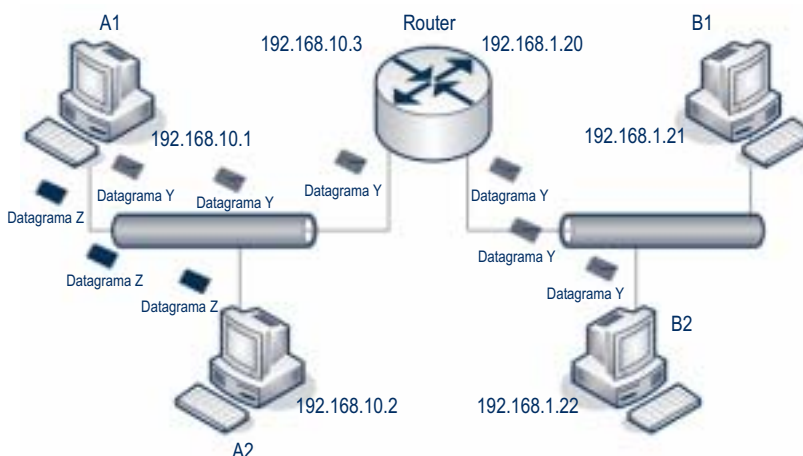


Fig. 5.4. Dos formas diferentes de enviar datagramas.

Aunque el comando `ping` fue creado para sistemas Unix, casi cualquier sistema proporciona el software adecuado para su ejecución. `ping` se sirve normalmente de dos mensajes específicos: `ECHO_REQUEST` y `ECHO_REPLY`. La conectividad IP entre equipos queda contrastada cuando se completa el camino de ida y vuelta de los mensajes ICMP. Además el comando `ping` ofrece información acerca del tiempo que tardan el ir y volver los paquetes de datos.

Pruebas de conectividad con `ping`

El comando `ping` puede presentar ligeras diferencias y distintas opciones según el sistema o equipo desde el que se ejecute. La versión de `ping` que viene con los sistemas operativos Windows envía por defecto cuatro paquetes ICMP de 32 bytes, y si el *host* de la dirección destino se encuentra activo y recibe los mensajes, responde a cada uno. La forma correcta de ejecutar el programa `ping` será desde la una ventana que permita teclear comandos. En



5. Protocolo TCP/IP

5.2 Arquitectura del protocolo TCP/IP

```
Símbolo de sistema
C:\> ping 192.168.0.50

Haciendo ping a 192.168.0.50 con 32 bytes de datos:

Respuesta desde 192.168.0.50: bytes=32 tiempo=1ms TTL=30
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=30
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=30
Respuesta desde 192.168.0.50: bytes=32 tiempo<1m TTL=30

Estadísticas de ping para 192.168.0.50:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Fig. 5.5. Respuesta correcta a los mensajes ICMP de ping.

```
Símbolo de sistema
C:\> ping 192.168.0.50

Haciendo ping a 192.168.0.50 con 32 bytes de datos:

Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.0.50:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 4
    (100% perdidos).
```

Fig. 5.6. El host destino no está activo o conectado.

Cada vez que se envía un paquete ECHO_REQUEST se guarda en el mismo paquete el instante exacto de su salida, y el *host* destino copia esta información en el paquete ECHO_REPLY que devuelve al *host* origen. Al recibir la respuesta se calcula el tiempo empleado, comparando la hora guardada en el paquete con la hora a la que ha sido recibido.

¿Qué tiempo de ida y vuelta es el más adecuado? La respuesta dependerá del tipo de redes por las que viajen los paquetes. Para una LAN los valores que superan tiempos de 100 ms se consideran un error de funcionamiento, aunque ya desde los 10 ms se interpreta que la red LAN es lenta o está congestionada. Si las pruebas ping se hacen a través de Internet, los tiempos se incrementan notablemente de tal manera que 200 ms es considerado un buen valor, siendo aceptables tiempos de hasta 500 ms.

Ping también se puede utilizar para medir la tasa de transferencia de datos entre dos equipos. Supongamos que deseamos conocerla para una conexión a Internet. En primer lugar enviaremos un paquete con un tamaño de 100 bytes al equipo remoto y seguidamente otro de 1 100 bytes. Utilizaremos los parámetros **-n** para enviar un solo paquete y **-l** para modificar su longitud:

```
C:\> ping -n 1 -l 100 www.mcgraw-hill.com
```

```
C:\> ping -n 1 -l 1100 www.mcgraw-hill.com
```

El primer ping ofrece un tiempo de ida y vuelta de 180 ms y el segundo de 443 ms: por tanto, para enviar 1 000 bytes adicionales (o, lo que es lo mismo, 8 000 bits) se emplean 263 ms en ida y vuelta, o bien 131,5 ms para uno de los trayectos. La tasa de transferencia medida para esta conexión es de aproximadamente 60,8 Kbps (8 000 bits/131,5 ms).

A veces resulta interesante testear una conexión de forma continua. Al añadir **-t** a ping, se envían paquetes de datos hasta que se ordene lo contrario pulsando **Control-C**.

```
C:\> ping -t 192.168.0.50
```

Pero la utilización de ping no se encuentra exenta de problemas, ya que ping depende del protocolo ARP que convierte la dirección IP en una dirección MAC y de los servidores DNS que convierten un nombre de dominio en una dirección IP. Si estos fallan, ping también falla. Además, por razones de seguridad, muchos administradores bloquean en los servidores la respuesta a mensajes ICMP en general o a ECHO_REQUEST en particular. Se debe a que a veces el comando ping es utilizado por usuarios malintencionados para perpetrar sobre los servidores ataques **DoS** (*Denial of Service*) consistentes en envíos masivos de mensajes ICMP que degradan el rendimiento de la red.



Una variante de ping es **fping** (www.fping.com), un programa que permite especificar en la línea de comandos más de un *host* destino o un bien fichero con una lista de todos los *hosts* destino.

Windows XP se abre la ventana *Símbolo del sistema* siguiendo esta secuencia de selección con el cursor:

Inicio > *Programas* > *Accesorios* > *Símbolo de sistema*

En el ejemplo de la Figura 5.5 el *host* destino responde a los cuatro mensajes ICMP recibidos desde el *host* origen.

A continuación de ping se suele teclear la dirección IP del *host* destino, aunque también puede escribirse su nombre. Por ejemplo:

```
C:\> ping www.mcgraw-hill.com
```

Cuando el *host* destino no responde a los mensajes ICMP la pantalla muestra un mensaje como el de la Figura 5.6.



Caso práctico 1

En este Caso práctico veremos distintas formas de emplear el comando `ping` y también cómo interpretar los resultados obtenidos al ejecutarlo. Las pruebas se realizarán sobre una pequeña red de área local conectada a Internet, cuyo esquema se representa en la Figura 5.7.

En nuestra red de ejemplo aparecen tecnologías tanto cableadas (Ethernet 100BaseT) como inalámbricas (802.11). Todos los equipos de la red LAN tienen instalado Windows XP.

La red cableada está compuesta por los ordenadores E1 y E2, el punto de acceso AP (*Access Point*) y el *router*. Bajo la cobertura del AP dos ordenadores portátiles P1 y P2 se integran en la red. Las direcciones IP de todos los equipos mencionados pertenecen a la red 192.168.135.0, si bien en el caso del *router* también se indica la dirección IP asignada al interfaz de que se conecta a la red Internet.

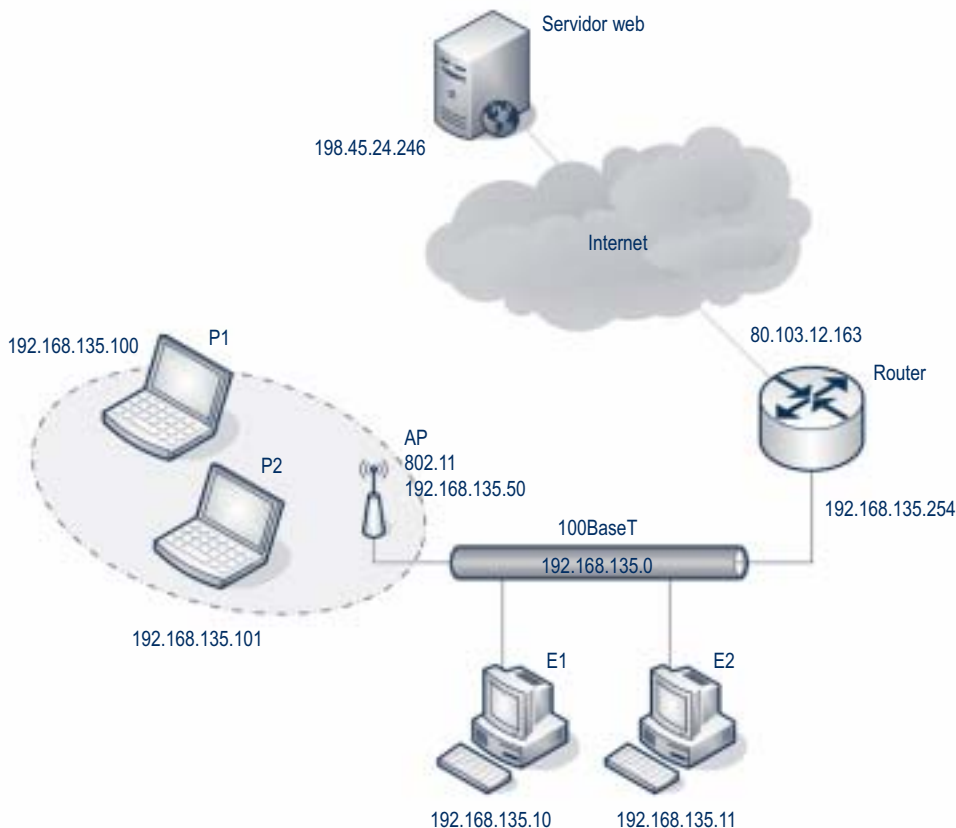


Fig. 5.7. Esquema de la red propuesta.

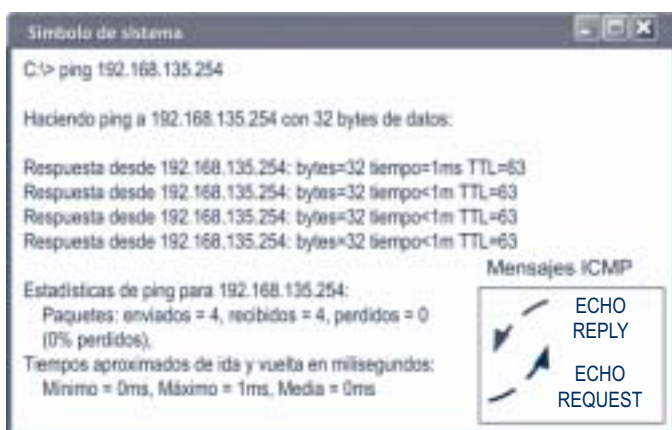


Fig. 5.8. Prueba de conectividad entre E1 y el router.

Para probar si funciona bien la conexión a Internet, se ha buscado un servidor remoto que permita procesar solicitudes de eco entrante ICMP (ECHO_REQUEST). Es importante pues cada vez más equipos conectados a Internet establecen mecanismos de protección (cortafuegos) para evitar ataques externos.

Las comprobaciones se harán dentro de la capa de red. Su objetivo fundamental será el de establecer la conectividad entre equipos dentro de esta capa, así como la calidad de la misma.

Comenzaremos con la red cableada. Para ello lo más adecuado será hacer un ping desde E1 y E2 al *router*. La Figura 5.8 muestra el resultado obtenido al ejecutar el comando desde el ordenador E1.

Como se puede apreciar el *router* responde correctamente a todas las peticiones hechas desde E1, y además lo hace en un tiempo razonable, siendo la media de menos de 1 ms. Una vez que sabemos que la conexión está dentro de los valores adecuados ejecutamos de nuevo el comando ping desde E1; en esta ocasión se trata de comprobar que el *router* puede enviar los mensajes ICMP a un equipo perteneciente a otra red. La Figura 5.9 muestra los resultados obtenidos.



5. Protocolo TCP/IP

5.2 Arquitectura del protocolo TCP/IP

Caso práctico 1 (cont.)



```

Símbolo de sistema
C:\> ping 198.45.24.246

Haciendo ping a 198.45.24.246 con 32 bytes de datos:

Respuesta desde 198.45.24.246: bytes=32 tiempo=213ms TTL=47
Respuesta desde 198.45.24.246: bytes=32 tiempo=218ms TTL=47
Respuesta desde 198.45.24.246: bytes=32 tiempo=212ms TTL=47
Respuesta desde 198.45.24.246: bytes=32 tiempo=213ms TTL=47

Estadísticas de ping para 198.45.24.246:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 212ms, Máximo = 218ms, Media = 214ms

Mensajes ICMP
    ECHO REPLY
    ECHO REQUEST
  
```



Fig. 5.9. Resultados desde E1 a un servidor web externo.

```

Símbolo de sistema
C:\> ping 192.168.135.10

Haciendo ping a 192.168.135.10 con 32 bytes de datos:

Respuesta desde 192.168.135.10: bytes=32 tiempo=6ms TTL=128
Respuesta desde 192.168.135.10: bytes=32 tiempo=7ms TTL=128
Respuesta desde 192.168.135.10: bytes=32 tiempo=7ms TTL=128
Respuesta desde 192.168.135.10: bytes=32 tiempo=7ms TTL=128

Estadísticas de ping para 192.168.135.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 6ms, Máximo = 7ms, Media = 6ms

Mensajes ICMP
    ECHO REPLY
    ECHO REQUEST
  
```



Fig. 5.10. Prueba desde la red inalámbrica a la cableada.

De nuevo se obtiene una respuesta positiva para los cuatro paquetes de datos enviados, si bien en esta ocasión los tiempos de ida y vuelta son muy superiores a los del caso anterior. Esto no indica ninguna anomalía. Un tiempo medio de respuesta de 214 ms para un equipo

remoto entra dentro de lo normal. Téngase en cuenta que para ello los paquetes de datos han realizado varios saltos de una red a otra hasta alcanzar su objetivo y luego deben volver al origen.

```

Símbolo de sistema
C:\> ping 192.168.135.101

Haciendo ping a 192.168.135.101 con 32 bytes de datos:

Respuesta desde 192.168.135.101: bytes=32 tiempo=15ms TTL=128
Respuesta desde 192.168.135.101: bytes=32 tiempo=8ms TTL=128
Respuesta desde 192.168.135.101: bytes=32 tiempo=7ms TTL=128
Respuesta desde 192.168.135.101: bytes=32 tiempo=7ms TTL=128

Estadísticas de ping para 192.168.135.101:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 7ms, Máximo = 15ms, Media = 9ms

Mensajes ICMP
    ECHO REPLY
    ECHO REQUEST
  
```



Fig. 5.11. Prueba entre equipos de la red inalámbrica.

Las dos comprobaciones anteriores se deben repetir desde el ordenador E2. Con ello tendremos la certeza de que los equipos conectados a la red cableada funcionan en la capa de red. Aún resta por comprobar si el punto de acceso inalámbrico responde correctamente a las pruebas ping, pero esto queda indirectamente resuelto si enviamos un ping desde uno de los portátiles al ordenador E1 (véase la Fig. 5.10).

Para que los paquetes de datos enviados por P1 puedan llegar al E1 deben pasar a través del AP (192.168.135.50). Por tanto, si E1 recibe datos y los devuelve a P1, hemos constatado al mismo tiempo que el AP funciona en la capa de red. No obstante el tiempo medio de respuesta que se observa no es el óptimo, pues 6 ms es un valor un poco alto. Para despejar dudas hacemos otra prueba, esta vez entre los dos equipos de la red inalámbrica (véase la Fig. 5.11).

El valor medio de respuesta entre P1 y P2 sigue siendo un poco elevado (9 ms). Con todo lo visto se puede afirmar sin duda que la conectividad en red entre todos los equipos es la adecuada, pero que los tiempos de respuesta cuando está implicado P1 son un poco lentos. Esto podría obedecer a una mala configuración de la red inalámbrica, aunque esta hipótesis queda descartada porque al hacer un ping desde P2 al router (o a cualquiera de los ordenadores E1 o E2) los tiempos medios de respuesta que se obtienen son de 2 ms.



Caso práctico 1 (cont.)

Parece claro que el problema se localiza en el portátil P1, ya que tanto la electrónica de red cableada como la inalámbrica funcionan correctamente.

Pero si P1 devuelve y envía los mensajes ICMP, ¿por qué los tiempos de ida y vuelta son tan lentos? Antes de responder volvamos e ejecutar el comando ping, esta vez desde P2 a P1 (véase la Fig. 5.12).

Ahora ya no hay dudas: si los mensajes enviados por P2 tardan una media de 234 ms en regresar a P1, hay algo en P1 que no funciona bien.

Debemos comprobar la configuración del adaptador de red inalámbrico. No estamos hablando del canal de radio empleado o de la clave WEP elegida; como ya sabemos, si estos valores no son los mismos en P1 y en el AP, no habría siquiera conectividad. Pero sí es cierto que muchos portátiles trabajan con rendimientos mínimos para ahorrar energía, y que hay puntos de acceso que no se llevan bien con esta táctica. En nuestro caso P1 tiene un procesador Intel Centrino®, que por defecto funciona con valores mínimos de rendimiento para ahorrar batería. Abrimos el cuadro de *Estado de conexiones de red inalámbricas* de Windows XP y hacemos la siguiente selección:

Propiedades > Configurar > Avanzadas > Gestión de alimentación

Desmarcamos el valor predeterminado (*Mínimo*) y movemos el cursor hasta la posición *Máximo*. El problema desaparecerá.

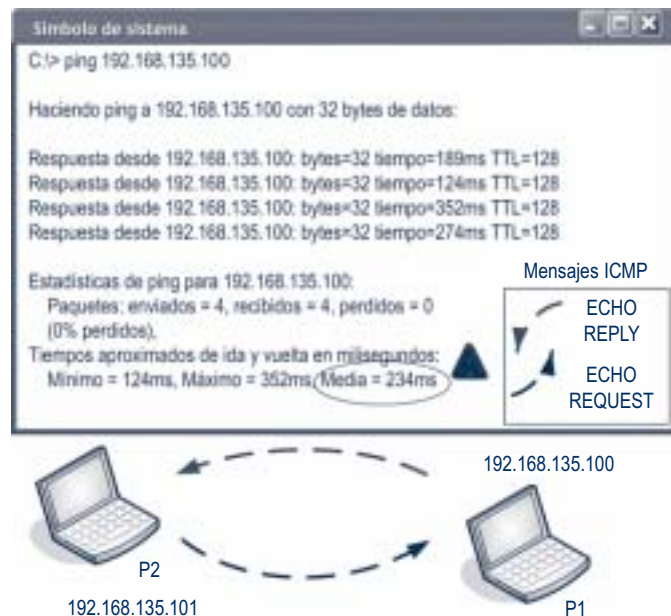


Fig. 5.12. Tiempos de ida y vuelta excesivamente lentos.

C. Capa de transporte

En esta capa se encuentran definidos el protocolo TCP y el protocolo UDP (*User Datagram Protocol*). TCP permite enviar los datos de un extremo a otro de la conexión con la posibilidad de detectar errores y corregirlos. UDP, por el contrario, reduce al máximo la cantidad de información incluida en la cabecera de cada datagrama, ganando con ello rapidez a costa de sacrificar la fiabilidad en la transmisión de datos. La Figura 5.13 muestra el formato de un mensaje UDP.

Ciertas aplicaciones prefieren utilizar en la capa de transporte el protocolo UDP aunque éste no haga corrección ni detección de errores. Como estas aplicaciones necesitan transmitir pequeñas cantidades de datos, resulta más eficaz reenviar los datagramas defectuosos que no sobrecargar cada uno con información de control en la cabecera.

Si se requiere más fiabilidad en los datos transmitidos las aplicaciones recurren en la capa de transporte al protocolo TCP. Al revés que UDP, es un protocolo orientado a conexión, y el formato de los datos que maneja es muy distinto al de los datagramas. Para TCP los datos son una secuencia o trama continua de bytes, cada comunicación es seleccionada en la trama mediante segmentos. En la Figura 5.14 (véase la pág. 100) vemos el formato de un segmento TCP.



Fig. 5.13. Representación del formato de mensaje UDP.

El protocolo TCP necesita que se establezca una conexión entre los equipos situados en ambos extremos de la misma. Antes de iniciar la transferencia de datos TCP efectúa una negociación entre los dos equipos basada en el intercambio de tres segmentos de datos. Precisamente por esta razón se la conoce como negociación de tres vías.

La Figura 5.15 (véase la pág. 100) representa las fases de una negociación de tres vías entre dos equipos a los que llamaremos *host A* y *host B*. La primera iniciativa la tiene el *host A*, quien envía al *host B* un segmento de sincronización (SYN) que contiene un identificador numérico. Al recibir esta información, el *host B* tiene constancia de la intención de iniciar una comunicación por parte del



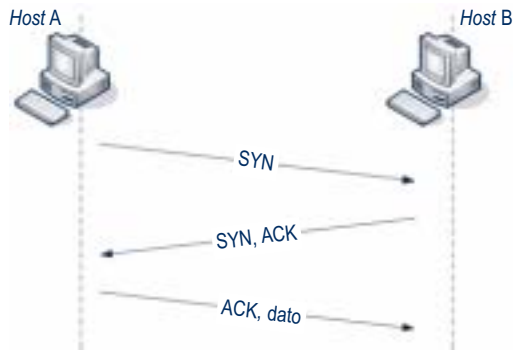
5. Protocolo TCP/IP

5.2 Arquitectura del protocolo TCP/IP



Fig. 5.14. Campos de cabecera para un segmento TCP.

Fig. 5.15. Negociación de tres vías en el protocolo TCP.



host A y, además, gracias al identificador numérico recibido, conoce el punto exacto en el que el host A señala el inicio de su transmisión de datos.

En segundo lugar el host B responde al host A con un segmento de confirmación (ACK) y lo acompaña del identificador que él empleará en la conexión.

La negociación finaliza con éxito cuando el host A recibe esta confirmación y responde con otra, que ya va acompañada de los primeros datos.

Los identificadores numéricos de secuencia son fundamentales para que TCP pueda determinar dentro del flujo continuo de bytes el segmento correspondiente a la comunicación entre los equipos A y B.

Al margen de estas negociaciones, la capa de transporte es responsable de hacer llegar los datos a las aplicaciones que los requieren en las capas superiores. Para ello se asocia cada aplicación a un número de 16 bits al que se denomina **número de puerto**. Tanto TCP como UDP hacen que la primera palabra de sus cabeceras contenga el puerto origen y destino de los datos que se transmi-

ten. Esta operación se conoce como **multiplexación**. En apartados posteriores se estudiará con más detalle la multiplexación de datos.

D. Capa de aplicación

Ésta es la capa más alta dentro de la estructura jerárquica del protocolo TCP/IP (véase la Fig. 5.1), e incluye las aplicaciones y procesos con los que intercambia datos la capa de transporte. TCP/IP tiene en esta capa protocolos que soportan servicios de conexión remota, correo electrónico y transferencia de archivos. De todos los protocolos de aplicación los más conocidos son:

- **Telnet** (*Network Terminal Protocol*). Es un protocolo que permite establecer conexiones con terminales remotos, de tal manera que se puedan ejecutar en ellos comandos de configuración y control.
- **FTP** (*File Transfer Protocol*). Protocolo orientado a conexión dedicado a la transferencia de archivos. FTP ofrece una gran fiabilidad con este servicio, en gran parte debido a que se basa en el protocolo TCP dentro de la capa de transporte. **TFTP** (*Trivial File Transfer Protocol*) es una versión de FTP que funciona más rápido, pero es menos fiable porque se sirve de mensajes UDP en la capa de transporte.
- **SMTP** (*Simple Mail Transfer Protocol*). Posibilita el funcionamiento del correo electrónico en las redes de ordenadores. SMTP recurre al protocolo de oficina postal **POP** (*Post Office Protocol*) para almacenar mensajes en los servidores de correo electrónico. Existen dos versiones: **POP2**, que necesita la intervención de SMTP para enviar mensajes; y **POP3**, que funciona de forma independiente.
- **HTTP** (*Hipertext Transfer Protocol*). Es un estándar de Internet que permite la transmisión de gran variedad de archivos de texto, gráficos, sonidos e imágenes. HTTP regula el proceso mediante el cual navegadores como Netscape, Mozilla o Internet Explorer solicitan información a los servidores web.
- **DNS** (*Domain Name Service*). Esta aplicación convierte nombres de dispositivos y de nodos de red en direcciones IP. Por ejemplo, el nombre `www.mcgraw-hill.es`, se convierte en la dirección `198.45.24.91`.

Los servidores de red proporcionan servicios esenciales para las comunicaciones entre ordenadores. A diferencia de lo que ocurre con muchos programas de aplicación, estos servicios no facilitan el acceso al usuario final. Para más información se aconseja consultar el estándar RFC 1122.



5.3 Direccionamiento IP

Hasta el momento nos hemos referido con frecuencia a las direcciones IP, que son números de 32 bits que constituyen la dirección unívoca de todo dispositivo conectado a una red que funcione con el protocolo TCP/IP. Las direcciones IP se escriben mediante la denominada notación punto decimal, o de cuatro octetos.

Con el fin de facilitar el manejo de las direcciones IP, los 32 bits se dividen en cuatro grupos de 8 bits cada uno, y cada uno de estos bytes se traduce a su equivalente en decimal. De cada conversión resulta un número comprendido entre 0 y 255. Estos cuatro números se escriben separados entre sí por un punto (véase la Fig. 5.16).

Las direcciones IP proporcionan dos datos: el número de red y el número de *host*. Para que un sistema pueda transmitir datos debe determinar con claridad la dirección destino de red y *host*, además de poder informar al resto de sistemas de cuál es su propia dirección de red y *host*.

Los sistemas de red se pueden direccionar de tres formas:

- **Unicast.** Los paquetes de datos tienen como destino la dirección de un único *host*.
- **Multicast.** Los datos se pueden enviar de forma simultánea a un determinado conjunto de *hosts*.
- **Broadcast.** Dirección de difusión que permite enviar datos a todos los sistemas que forman parte de una red. Este tipo de direccionamiento está siempre supeditado a las capacidades físicas de los dispositivos conectados en la red.

El número de bits empleado para definir la red y el número de bits que identifican al *host* pueden variar entre unos casos y otros. Cada dirección IP tiene un prefijo cuya longitud indica qué bits corresponden al identificador de red y cuáles al *host*. La longitud de este prefijo la establecen los bits de la máscara.

Éste es el funcionamiento de los bits de máscara: si un bit de la máscara es 1, su bit equivalente en la dirección IP corresponde a la dirección de red. Si un bit de la máscara es 0, el bit equivalente en la dirección IP pertenece a la dirección de *host*.

Tal como se puede ver en la Figura 5.17 la dirección IP 198.16.23.102 tiene asociada la máscara 255.255.255.0, o lo que es lo mismo, tiene un prefijo de 24 bits, por lo que la dirección de *host* utiliza los 8 bits restantes. La forma abreviada para la dirección y prefijo sería: 198.16.23.102/24. Aplicando la máscara de red (es decir realizando la operación lógica AND entre ambas), resulta

que la dirección de red es 198.16.23 y la dirección de *host* 102. Como la cifra 0 en el campo de *host* está reservada para identificar la red, la dirección de red también se escribe: 198.16.23.0

Como el prefijo determina la parte de la dirección IP destinada a identificar la red, los bits disponibles para direcciones de *hosts* se obtienen restando a 32 la longitud del prefijo. Un ejemplo: la dirección 192.168.10.0/20 destina 12 bits para direccionar *hosts* ($32 - 20 = 12$). Por tanto, la red 192.168.0 dispondrá de un bloque de 12 bits de direcciones ($2^{12} = 4096$), de la dirección 0.1 a la 15.254, estando reservadas las direcciones 0.0 y 15.255 a red y difusión respectivamente.

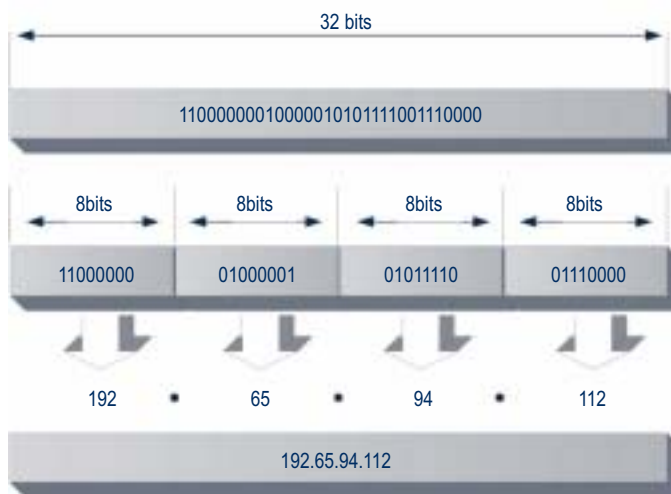


Fig. 5.16. Distintas formas de representar una dirección IP.

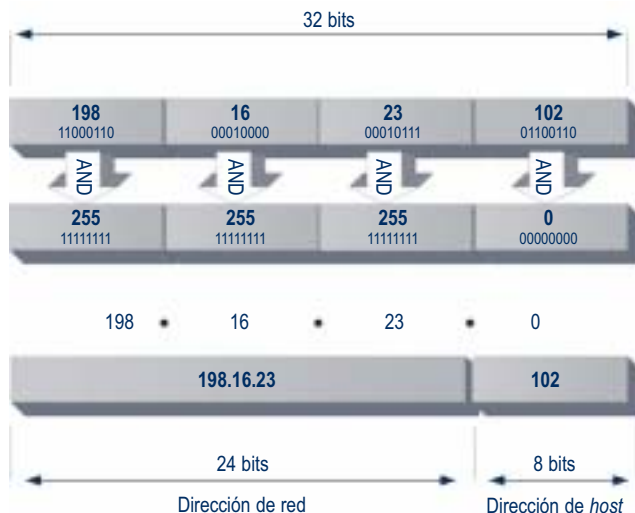


Fig. 5.17. Solución de direcciones de red y *host*.



5. Protocolo TCP/IP

5.3 Direccionamiento IP

Clases de direcciones IP

Aunque existen cinco, son tres las principales **clases de direcciones IP: A, B y C**. El protocolo IP distingue la clase a la que pertenece una dirección analizando el valor de sus bits de mayor peso. El concepto de clase está asociado al de máscara por defecto. Para determinar la clase de una dirección IP se siguen las siguientes reglas:

- **Clase A.** Si el bit de mayor peso es «0» la máscara por defecto tendrá un prefijo de 8 bits. Se tienen por tanto 8 bits para direcciones de red y 24 bits *hosts*.
- **Clase B.** Si los dos primeros bits son «1» «0», la máscara por defecto tendrá una longitud de 16 bits (prefijo 16). Con ello los primeros 16 bits son para identificar la red; los 16 últimos, para identificar los *hosts*.
- **Clase C.** Si los tres primeros bits son «1» «1» «0» la máscara por defecto tiene un prefijo de 24 bits. Para esta clase se contempla la existencia de una gran cantidad de redes, en concreto 224. En cada una de ellas el número de equipos es como máximo 253, una vez restadas las direcciones de red y difusión.
- **Clase D.** Si los cuatro primeros bits de la dirección son «1» «1» «1» «0» nos encontramos frente a una dirección *multicast*. Entonces, no se habla de una dirección de red, sino de un grupo de equipos a los

que se desea enviar datos simultáneamente. Todos los bits de una dirección *multicast* son significativos, así que la máscara por defecto es de 32 bits (prefijo 32).

- **Clase E.** Si los cuatro primeros bits de la dirección son unos lógicos, la dirección IP pertenece a un rango que se ha reservado para experimentación. Dentro de esta clase aparece la dirección IP de difusión 255.255.255.255.

Las reglas descritas se refieren al análisis de direcciones IP representadas en binario. Pero como siempre se trabaja con direcciones IP escritas en notación punto decimal, se emplea con mayor frecuencia otra técnica consistente en analizar la primera cifra según estas reglas:

- Si el primer octeto es menor de 128, la máscara por defecto es de 8 bits (clase A, 255.0.0.0). La excepción a esta regla la tenemos en la red 127.0.0.0, que se ha reservado al completo para la dirección especial 127.0.0.1. Ésta es la dirección de reenvío o bucle cerrado (*loopback*), y sirve para que un host compruebe si su NIC funciona correctamente en el protocolo IP.
- Si la primera cifra decimal está comprendida entre 128 y 191, la máscara por defecto tendrá un prefijo de 16 bits (clase B, 255.255.0.0).



Caso práctico 2

Estudiaremos un caso real de **segmentación IP** de una red mediante la creación de subredes.

En la Figura 5.18 se muestra una red de clase C con dirección 192.168.15.0/24 que está compuesta por varios equipos con conexiones cableadas e inalámbricas.

El tráfico de red detectado es el siguiente: los equipos desde E1 a E4 utilizan con frecuencia el equipo multifunción MF1 y leen y escriben archivos en el servidor S1, quien además almacena copias de seguridad de ellos.

Los portátiles que acceden a la red a través del punto de acceso AP1 utilizan exclusivamente un servicio FTP configurado en el servidor S2.

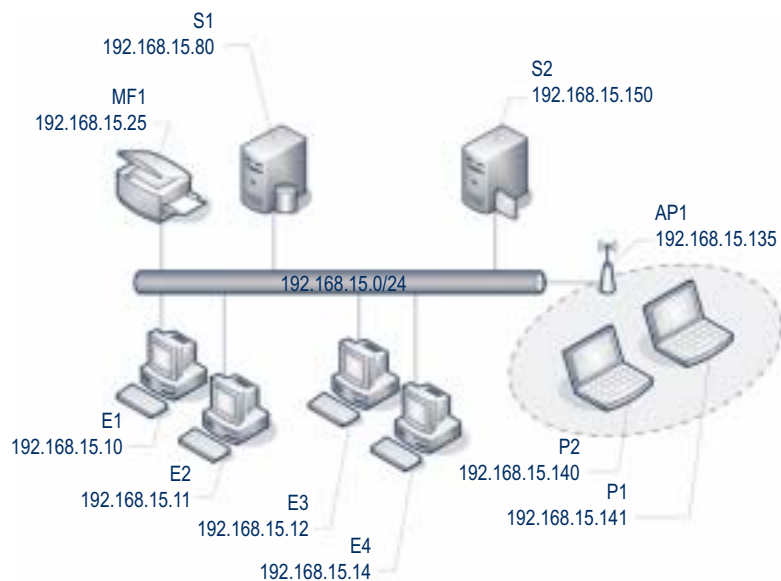


Fig. 5.18. Esquema de la red propuesta para el Caso práctico.



Caso práctico 2 (cont.)

De este comportamiento del tráfico de red se deduce que a pesar de que todos los equipos están concentrados con la misma electrónica de red, resulta viable dividir la red en dos subredes independientes respecto del protocolo IP.

Para obtener dos subredes basta con ampliar el prefijo en un bit, razón por la que en todos los equipos se cambiará la máscara de subred a 255.255.255.128 (véase Fig. 5.19).

Las subredes creadas admiten cada una de ellas 126 direcciones de *host*, que son:

- **Subred 1** (192.168.15.0). Comprende todos los hosts con direcciones entre 1 y 126. Esto incluye el servidor S1, el equipo multifunción MF1 y los ordenadores E1 a E4. La dirección de difusión de la subred 1 será 192.168.15.127.
- **Subred 2** (192.168.15.128). A ella pertenecen todos los hosts con direcciones entre 129 y 254. Lo cual afecta al punto de acceso AP1, a los equipos portátiles P1 y P2, y al servidor S2. La dirección de difusión es 192.168.15.255.

Verificamos la segmentación de redes realizando pruebas de conectividad. La respuesta entre equipos de la misma subred será la correcta (véase la Fig. 5.5).

La Figura 5.20 muestra el mensaje que debe visualizarse en cualquier equipo de la subred 1 cuando lanza un mensaje ICMP a otro equipo de la subred 2, en este caso a P2.

Lo mismo sucederá si un equipo de la subred 2 ejecuta el comando `ping` hacia otro de la subred 1 (véase la Fig. 5.21).

Hay que tener en cuenta que `ping` da como respuesta «Host de destino inaccesible», porque el equipo desde el que se ejecuta no tiene configurada una puerta de acceso.

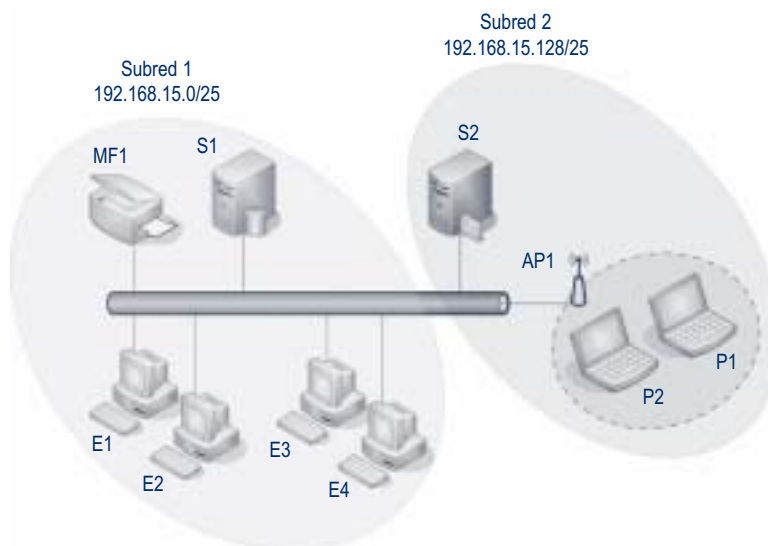


Fig. 5.19. Direcciones y prefijos para crear dos subredes.

```
Simbolo de sistema
C:\> ping 192.168.15.140

Haciendo ping a 192.168.15.140 con 32 bytes de datos:

Host de destino inaccesible.
Host de destino inaccesible.
Host de destino inaccesible.
Host de destino inaccesible.

Estadísticas de ping para 192.168.15.140:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),
```

Fig. 5.20. Ping desde la subred 1 a la subred 2.

```
Simbolo de sistema
C:\> ping 192.168.15.80

Haciendo ping a 192.168.15.80 con 32 bytes de datos:

Host de destino inaccesible.
Host de destino inaccesible.
Host de destino inaccesible.
Host de destino inaccesible.

Estadísticas de ping para 192.168.15.80:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4
    (100% perdidos),
```

Fig. 5.21. Ping desde la subred 2 a la subred 1.



5. Protocolo TCP/IP

5.3 Direccionamiento IP

- Desde 192 a 223 la máscara por defecto tiene un prefijo de 24 bits (clase C, 255.255.255.0).
- Desde 224 a 239 la dirección IP es *multicast*, y la máscara por defecto tendrá una longitud de 32 bits (clase D, 255.255.255.255).
- De 240 a 255 las direcciones IP son de uso experimental. No se asignan a ningún sistema o red concreto.

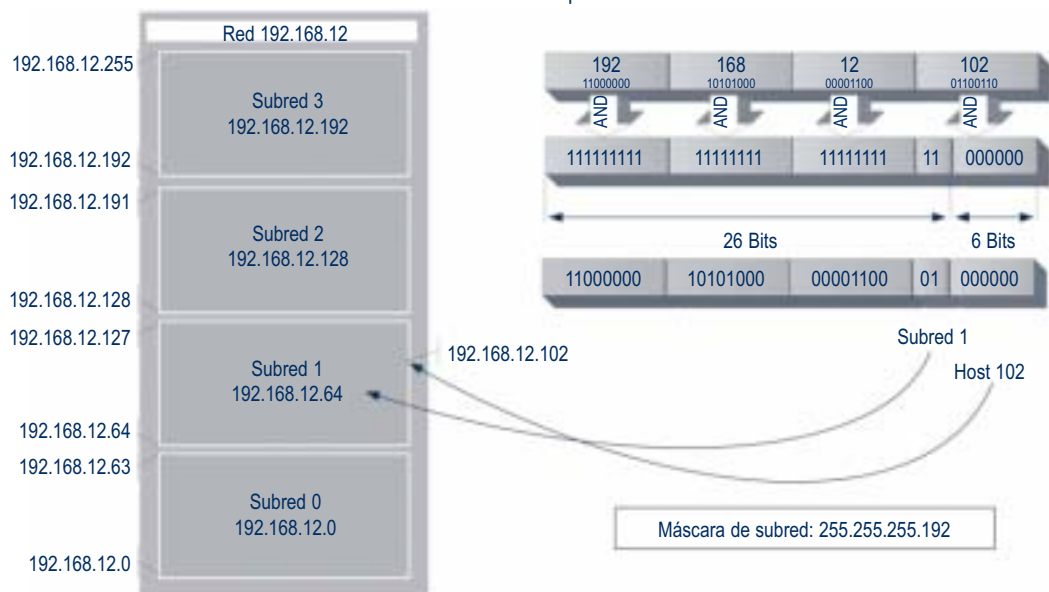
Subredes

La estructura de una dirección IP puede ser localmente modificada al destinar parte de los bits de dirección de *host* para bits de red adicionales o de subred.

La creación de subredes reduce el número posible de hosts que pueden existir en una de ellas. Los bits de subred definen un nuevo bloque de direcciones dentro del bloque de direcciones de red.

En el ejemplo de la Figura 5.22 podemos apreciar cómo dentro de la red de 192.168.12.0/C se crean cuatro subredes. Para conseguirlo se aumenta en dos bits la longitud del prefijo, haciendo que pase de 24 a 26 bits. Si lo vemos desde el punto de vista de la máscara de red tendremos que pasa de ser 255.255.255.0 a 255.255.255.192.

Con estas cuatro subredes se puede mejorar la eficiencia de la red, pues queda restringido el tráfico de datos al ámbito local, y cada una de ellas tendrá además una dirección única.



Dependiendo de la clase de red y del número de veces que se pretenda segmentar, se obtienen una serie de direcciones válidas. La directiva RFC 1878 contiene una lista con todas las posibles combinaciones y direcciones válidas.

En la Figura 5.18 se han utilizado dos bits del bloque de direcciones de red para definir subredes. De ello se desprende que los seis bits restantes sirven para definir un bloque de 64 direcciones ($2^{(8-2)} = 64$) de *hosts*, del que, como ya se sabe, la primera y la última están reservadas para identificar la subred y la dirección de difusión.

De la operación AND entre la dirección 192.168.12.102 de ejemplo, y la máscara de subred 255.255.255.192, se obtiene que pertenece a la subred 1 (01) con direcciones válidas que oscilan entre 192.168.12.65 y 192.168.12.126. El equipo con esta dirección sólo es capaz de resolver las direcciones que se encuentran dentro de este rango.

Se podrían plantear otros números de *host* como ejemplo:

- **192.168.12.10** AND 255.255.255.192 da como resultado que pertenece a la subred 0 («00») con direcciones posibles entre 192.168.12.1 y 192.168.12.62.
- **192.168.12.204** AND 255.255.255.192 resulta que pertenece a la subred 3 («11») con direcciones comprendidas entre 192.168.12.193 y 192.168.12.254, ambas inclusive.

La eficiencia referida consiste en que cuando un equipo de la subred 1 envía un paquete de datos, éste es atendido sólo por los demás equipos de la subred 1. Los de las otras subredes ignoran este tráfico de datos y se concentran en el suyo propio. Esto se traduce en un mejor aprovechamiento del ancho de banda de la red.

Fig. 5.22. Cuatro subredes dentro de una red de clase C.



5.4 Tablas de enrutamiento

Como ya se ha dicho en otras ocasiones la puerta de enlace de una LAN permite enviar datos a otras redes distintas a la de origen. Encaminar o enrutar (ambos términos son válidos) los datos es el principal cometido de estos dispositivos. La decisión de a quién se envían los datos no es patrimonio exclusivo de los *routers*.

En la práctica, un *host* está continuamente tomando decisiones sobre **adónde** envía los datos. La lógica de decisión se resume en dos puntos:

- Si se constata que el *host* destino se encuentra en la misma red que el *host* origen, los datos son enviados directamente, de *host* a *host*, sin intermediarios.
- La otra posibilidad es que la dirección del *host* destino no pertenezca al mismo bloque de direcciones IP que el *host* origen. En este caso los datos se envían a la puerta de enlace de la red local, y será este *router* quien los encamina a partir de que los recibe en uno de sus puertos.

Tal como se mostraba en el **Caso práctico 2**, cuando no se introduce en la configuración de un equipo la dirección de la puerta de enlace, la dirección destino de otras redes resulta inaccesible para el sistema y los datos no se envían.

Los *hosts* consultan una tabla interna de enrutamiento cuyo estado puede mostrarse con el comando `netstat`. La Figura 5.23 ofrece un ejemplo típico de tabla de rutas. En este caso `netstat` se ha ejecutado desde un equipo con el sistema operativo Windows XP. En otros sistemas la tabla puede tener otro aspecto, pero en esencia ofrecerá la misma información. `Netstat -nr` garantiza que las direcciones se vean con la notación punto decimal.

La tabla de rutas de la Figura 5.23 muestra en primer lugar una lista con los adaptadores de red detectados en el equipo. En este caso sólo hay uno con MAC 00-0c-6e-2b-49-66 y otro virtual para pruebas de reenvío a la dirección de bucle cerrado 127.0.0.0; a continuación aparecen las rutas activas clasificadas en cinco columnas cuyos valores son consultados cada vez que se quiere enviar un dato:

- **Destino de red.** El destino puede ser un *host* o una red completa. Para buscar la ruta adecuada se trabaja con el dato de la siguiente columna de máscara de red. La búsqueda se realiza priorizando las rutas únicas sobre las genéricas. Por ejemplo: 192.168.135.0 es una dirección de subred, 224.0.0.0 es *multicast* y 192.168.135.10 una dirección cuyo único destino es un *host*.

```
Símbolo de sistema
C:\> netstat -nr

Tabla de rutas
-----
Lista de interfaces
0x1 ..... MS TCP Loopback interface
0x10003 ...00 0c 6e 2b 49 66 ..... NVIDIA nForce MCP Networking Adapter
-----
Rutas activas:
Destino de red      Máscara de red      Puerta de acceso      Interfaz      Métrica
0.0.0.0             0.0.0.0             192.168.135.254      192.168.135.10  20
127.0.0.0           255.0.0.0           127.0.0.1            127.0.0.1       1
192.168.135.0       255.255.255.0       192.168.135.10      192.168.135.10  20
192.168.135.10     255.255.255.255     127.0.0.1            127.0.0.1       20
192.168.135.255    255.255.255.255     192.168.135.10      192.168.135.10  20
224.0.0.0           240.0.0.0           192.168.135.10      192.168.135.10  20
255.255.255.255    255.255.255.255     192.168.135.10      192.168.135.10  1
Puerta de enlace predeterminada: 192.168.135.254
-----
Rutas persistentes:
ninguno
```

Fig. 5.23. Tabla de rutas mostrada por el comando `netstat`.

- **Máscara de red.** Permite diferenciar la parte de la dirección destino que se refiere a red de la de *host*. En el caso particular 255.255.255.255 las direcciones destino de red y *host* son coincidentes. Por ejemplo: si el destino de un paquete de datos es la dirección del propio *host* que los envía (192.168.135.10) o la dirección de difusión de la red a la que pertenece (192.168.135.255).
- **Puerta de acceso.** Aquí se especifica la dirección IP del dispositivo al que se debe enviar el paquete de datos después de detectar una coincidencia con los valores anteriores. Por ejemplo: si el destino está en la propia red (192.168.135.0) del *host* (192.168.135.10) su propia interfaz es la puerta de acceso al medio; si el destino es la dirección del *host* (192.168.135.10) la interfaz a la que se recurre es la de bucle cerrado (127.0.0.0); y si no hay coincidencias de destino (0.0.0.0) la puerta de acceso designada es la de la puerta de enlace predeterminada (192.168.135.254).
- **Interfaz.** Dirección IP del adaptador de red encargado de enviar el paquete de datos en la ruta designada.
- **Métrica.** Número máximo de saltos hasta alcanzar el destino. Por ejemplo: cuando la ruta es la de bucle cerrado la métrica es 1; para el resto aparece asignado el valor 20 por defecto.



5. Las comunicaciones urgentes

5.4 Tablas de enrutamiento



Caso práctico 3

Vamos a estudiar las tablas de rutas de los sistemas ya vistos en el Caso práctico 2, si bien en esta ocasión entra en juego un *router* que realiza la interconexión a Internet de la red. Se da por supuesto que en todos los ordenadores el sistema operativo instalado es Windows XP.

Al seguir manteniendo la segmentación en dos subredes, cabe preguntarse qué dirección IP se debe asignar al *router*. La respuesta depende de qué equipos se pretenda que tengan acceso a Internet. Si como parece más lógico se facilita el acceso a todos los equipos, la dirección IP del *router* no puede circunscribirse a una de las subredes. La Figura 5.24 muestra una posible solución:

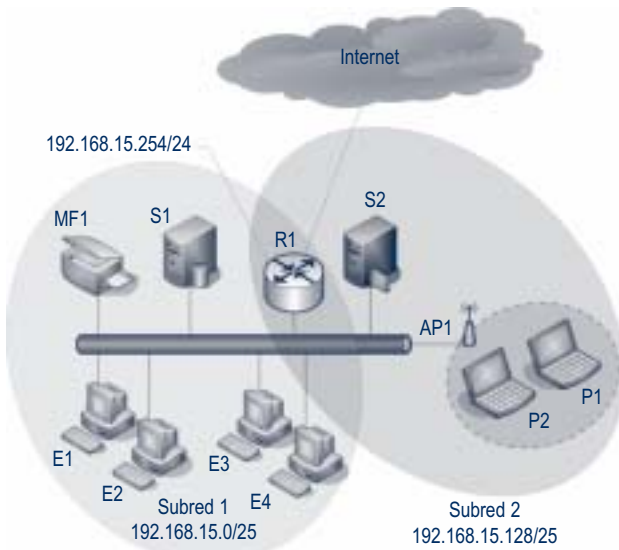


Fig. 5.24. Conexión a Internet de dos subredes.

La dirección IP 192.168.135.254/24 otorga al *router* la capacidad de resolver la dirección de cualquier *host* comprendido entre 1 y 254, lo cual le sitúa tanto en el dominio de enrutamiento de la subred 1 como en el de la subred 2.

En cuanto a los equipos dentro de las subredes, tenemos que en todos ellos se ha tenido que especificar la dirección IP del *router* como puerta de enlace predeterminada. Esto se traduce en que las tablas de rutas en cada uno de ellos se ven modificadas de forma automática con la aparición de nuevas entradas. En un equipo de la subred 1, por ejemplo E1 (192.168.15.10), tecleamos en la ventana de símbolo de sistema el comando: `route print`. La tabla de rutas que aparecerá se muestra en la Figura 5.25.

La primera entrada nos informa de que los paquetes de datos para los que no exista coincidencia son enviados al *router* a través del único adaptador de red disponible con métrica 30. La tercera entrada de la tabla determina que cuando la red destino sea la subred 1, los datos son enviados directamente, sin que salgan de la subred (no emplea al *router* como puerta de enlace). Es interesante fijarse en que no aparece ninguna ruta que haga referencia a la subred 2: si un equipo de la subred 1 intenta enviar datos a la subred 2, no sucede que el destino sea inalcanzable. Ahora estos datos se envían al *router* y él se encarga de resolver la dirección destino. Así la eficiencia en el tráfico de red se mantiene, pues los mensajes para equipos de una misma subred no son encaminados o replicados a la otra.

La Figura 5.26 corresponde con la tabla de rutas que presenta el equipo P2 (192.168.15.140) en la subred 2. Se aprecia que P2 tiene dos adaptadores de red, y que el Ethernet no ha sido configurado en el protocolo IP. El resto de explicaciones es similar al de la Figura 5.25.

```
Windows de sistema
C:\> route print

Tabla de rutas
-----
Lista de interfaces
0x1 ..... MS TCP Loopback interface
0x10003 ...00 c0 3f 1a b6 fe ..... Adaptador Fast Ethernet compatible VIA
-----
Rutas activas:
Destino de red   Máscara de red   Puerta de acceso   Interfaz   Métrica
0.0.0.0         0.0.0.0         192.168.15.254    192.168.15.10  30
127.0.0.0       255.0.0.0       127.0.0.1        127.0.0.1    1
192.168.15.0    255.255.255.128 192.168.15.10    192.168.15.10  20
192.168.15.10   255.255.255.255 127.0.0.1        127.0.0.1    20
192.168.15.255 255.255.255.255 192.168.15.10    192.168.15.10  20
224.0.0.0       240.0.0.0       192.168.15.10    192.168.15.10  20
255.255.255.255 255.255.255.255 192.168.15.10    192.168.15.10  1
Puerta de enlace predeterminada: 192.168.15.254
-----
Rutas persistentes:
ninguno
```

Fig. 5.25. Tablas de ruta de un equipo en la subred 1.

```
Windows de sistema
C:\> route print

Tabla de rutas
-----
Lista de interfaces
0x1 ..... MS TCP Loopback interface
0x10003 ...00 0c f1 03 c9 4d ..... Intel(R) PRO/Wireless LAN 2100 3B Mini PCI Adapter
0x10004 ...00 02 3f 11 33 30 ..... Realtek RTL8139/10x Family Fast Ethernet NIC
-----
Rutas activas:
Destino de red   Máscara de red   Puerta de acceso   Interfaz   Métrica
0.0.0.0         0.0.0.0         192.168.15.254    192.168.15.140  30
127.0.0.0       255.0.0.0       127.0.0.1        127.0.0.1    1
192.168.15.128 255.255.255.128 192.168.15.140    192.168.15.140  30
192.168.15.140 255.255.255.255 127.0.0.1        127.0.0.1    30
192.168.15.255 255.255.255.255 192.168.15.140    192.168.15.140  30
224.0.0.0       240.0.0.0       192.168.15.140    192.168.15.140  30
255.255.255.255 255.255.255.255 192.168.15.140    192.168.15.140  1
Puerta de enlace predeterminada: 192.168.15.254
-----
Rutas persistentes:
ninguno
```

Fig. 5.26. Tablas de ruta de un equipo en la subred 2.

5. Las comunicaciones urgentes

5.4 Tablas de enrutamiento



También se puede obtener la tabla de rutas con el comando **route**: basta con teclear a continuación del símbolo de sistema: `route print`. La información visualizada será la misma que con `netstat -r`. No obstante, la verdadera utilidad de `route` se encuentra en su capacidad de añadir o eliminar entradas en la tabla de rutas de un sistema. Estas entradas no persisten al reiniciar el sistema.

Resolución de direcciones

El resultado de combinar las direcciones IP y la información contenida en las tablas de rutas consiste en que los datos pueden recorrer distintas redes hasta llegar al destino específico. Sin embargo, las capas inferiores de la red no procesan ni entienden esta información.

En la capa física no se distingue entre redes y equipos. Para este nivel la única información válida es la de la dirección física de cada adaptador de red, conocida como dirección MAC, y que se analizó en la Unidad 4. Cuando los datos llegan a la capa más baja, hay que encargarse de asociar las direcciones IP con las direcciones MAC. Ésta es la principal labor del protocolo ARP.

Cada *host* mantiene en la memoria caché de su adaptador de red una tabla de equivalencias entre direcciones IP y direcciones IP. El contenido de esta tabla se puede ver al introducir en línea de comandos la orden: `arp -a`.

La Figura 5.27 muestra un ejemplo básico de tabla ARP. Del cual se debe deducir que cuando el *host* 192.168.135.10 quiere enviar sus paquetes de datos a través de su adaptador de red al *host* 192.168.135.254, traduce la dirección destino de su formato IP al formato MAC 00-a0-f9-02-a7-94. Y es éste el que se envía en las tramas Ethernet.

Cuando no se encuentra la equivalencia del *host* destino en la tabla ARP, el *host* origen envía a todos los equipos de su red una petición ARP. A esta petición responde el propio equipo aludido o un equipo capaz de resolver direcciones IP en direcciones MAC, por ejemplo la puerta de enlace de la red.

La respuesta a la consulta es almacenada de forma automática en la tabla ARP del *host* origen. Cuando una entrada es creada de esta forma es clasificada como dinámica y todo este proceso sucede sin que las capas superiores del protocolo TCP/IP tengan que estar pendientes de él.

También es posible crear manualmente una entrada en estas tablas mediante el comando `arp`. Por ejemplo:

```
C:\> arp -s 192.168.135.100 00-0c-f1-03-c9-4d
```

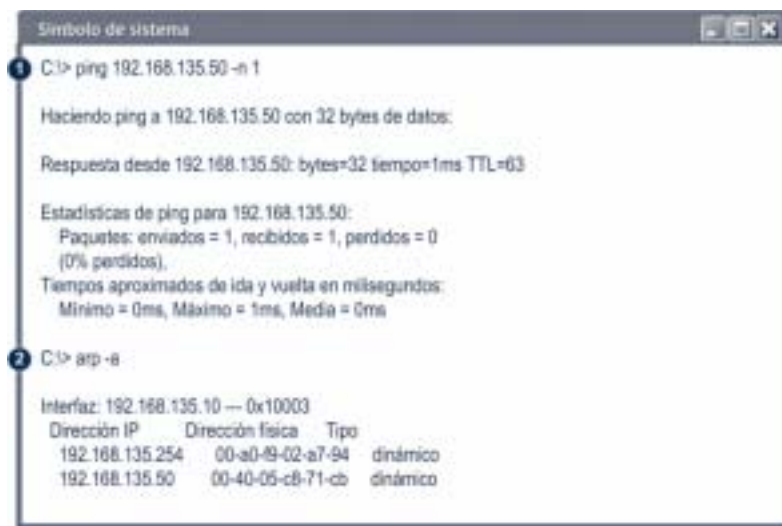
Para borrar entradas la orden sería `arp -d` seguida de la dirección o grupo de direcciones que se pretendan elimi-



```
Símbolo de sistema
C:\> arp -a

Interfaz: 192.168.135.10 --- 0x10003
Dirección IP      Dirección física  Tipo
192.168.135.254  00-a0-f9-02-a7-94  dinámico
```

Fig. 5.27. Tabla ARP con una sola entrada.



```
Símbolo de sistema
1 C:\> ping 192.168.135.50 -n 1

Haciendo ping a 192.168.135.50 con 32 bytes de datos:

Respuesta desde 192.168.135.50: bytes=32 tiempo=1ms TTL=63

Estadísticas de ping para 192.168.135.50:
    Paquetes: enviados = 1, recibidos = 1, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms

2 C:\> arp -a

Interfaz: 192.168.135.10 --- 0x10003
Dirección IP      Dirección física  Tipo
192.168.135.254  00-a0-f9-02-a7-94  dinámico
192.168.135.50   00-40-05-c8-71-cb  dinámico
```

Fig. 5.28. Proceso de registro de una entrada dinámica.

nar. Se puede obtener más información sobre el comando `arp` tecleando en línea de comandos `arp /?`.

La Figura 5.28 representa en dos pasos cómo se añade una entrada dinámica a una tabla ARP. En primer lugar se envía un paquete de 32 bytes con el comando `ping`, para conseguir que el *host* destino se identifique y notifique su dirección MAC. Al visualizar en el segundo paso la tabla ARP se aprecia que esta respuesta ha sido registrada como una entrada dinámica.

Los comandos `arp` y `ping` pueden utilizarse conjuntamente para solucionar problemas en la red. La Figura 5.29 muestra que al enviar un mensaje ICMP al *host* 192.168.135.60, no se obtiene respuesta a la solicitud de eco.

Lo más lógico sería pensar que el equipo no se encuentra conectado a la red. Sin embargo, para asegurarnos de que esto es así visualizamos la tabla ARP. Es entonces cuando vemos que la dirección del *host* destino ha sido asociada a su dirección MAC generando una entrada dinámica. Por tanto, podemos concluir que el equipo 192.168.135.60 sí está conectado a la red, y que probablemente su administrador ha instalado un filtro para no responder a mensajes ICMP.



5. Las comunicaciones urgentes

5.4 Tablas de enrutamiento

◆ Determinar la traza de una ruta

La conexión de redes con enrutadores tal como se hace en Internet, trae consigo que las conexiones entre equipos locales y remotos involucren a gran cantidad de sistemas.

Hasta ahora hemos estudiado el funcionamiento de comandos que realizan pruebas de conectividad, imprimen tablas de enrutamiento o resuelven direcciones IP. El comando **tracert** engloba todas estas funcionalida-

des y además nos informa del camino exacto que siguen los paquetes de datos desde el *host* local al *host* remoto.

Un ejemplo típico lo encontramos cuando un ordenador se conecta a una página web (p. ej.: www.mcgraw-hill.com). ¿Qué camino recorren realmente los paquetes de datos? En la representación gráfica de la Figura 5.30 se observa con detalle la respuesta a esta pregunta.

Tracert envía paquetes de datos UDP desde el *host* local al *host* remoto, muestra el nombre de cada puerta de enlace (siempre que se haya configurado) y su dirección IP. Mientras se determina la traza, cada vez que es detectada una puerta de enlace activa, se numera y se registran sus datos en una entrada denominada salto. Por defecto se imprimen trazas con un máximo de 30 saltos.

En cada salto se indica el tiempo que ha empleado cada paquete en ir y volver desde el *host* local a la puerta de enlace. La ejecución de **tracert** puede finalizar bien por completarse la traza hasta el *host* destino o porque se haya superado el número máximo de saltos.

Una traza incompleta proporciona información muy valiosa a la hora de resolver problemas de enrutamiento entre redes. Cuando se agotan los tiempos de espera para un paquete de datos, la localización del salto en el que se produce este retardo nos permite plantear una hipótesis más acertada del problema. Por ejemplo:

Fig. 5.29. Solución de problemas de red con ping y arp.

```
Símbolo de sistema
C:\> ping 192.168.135.60 -n 1

Haciendo ping a 192.168.135.60 con 32 bytes de datos:

Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 192.168.135.60:
    Paquetes: enviados = 1, recibidos = 0, perdidos = 1
            (100% perdidos)

C:\> arp -a

Interfaz: 192.168.135.10 --- 0x10003
Dirección IP      Dirección física  Tipo
192.168.135.254  00-a0-f9-02-a7-94  dinámico
192.168.135.50   00-40-05-c8-71-cb  dinámico
192.168.135.60   00-0c-41-ba-6c-57  dinámico
```

```
Símbolo de sistema
C:\> tracert www.mcgraw-hill.com

Traza a la dirección www.elb.mcgraw-hill.com [198.45.19.151]
sobre un máximo de 30 saltos:

 1  2 ms  2 ms  2 ms  192.168.1.254
 2  72 ms 195 ms 74 ms naba1-1-100.net.uni2.es [62.36.196.21]
 3  96 ms 77 ms 89 ms rmba1-fe0-1-0.net.uni2.es [62.36.196.65]
 4  76 ms 71 ms 77 ms rba1-GE4/1.1.net.uni2.es [62.36.196.145]
 5  77 ms 77 ms 72 ms 62.36.196.234
 6  71 ms 71 ms 71 ms 62.36.204.67
 7  70 ms 71 ms 77 ms P1-0.BARBB1.Barcelona.opentransit.net [193.251.251.33]
 8  82 ms 89 ms 89 ms P2-0.PASCR1.Pastourelle.opentransit.net [193.251.241.66]
 9  80 ms 95 ms 83 ms P13-0.PASCR3.Pastourelle.opentransit.net [193.251.129.62]
10 165 ms 167 ms 167 ms P2-0.OAKCR2.Oakhill.opentransit.net [193.251.242.96]
11 173 ms 167 ms 173 ms P1-0.ASHCR1.Ashburn.opentransit.net [193.251.243.89]
12 169 ms 167 ms 173 ms si-st21-ash-15-3.sprintlink.net [144.223.246.21]
13 165 ms 174 ms 167 ms si-st20-ash-12-0.sprintlink.net [144.232.19.240]
14 167 ms 167 ms 167 ms 0.so-5-0-0.BR1.DCAS.ALTER.NET [204.255.168.13]
15 175 ms 167 ms 173 ms 0.so-0-3-0.XL2.DCAS.ALTER.NET [152.63.43.178]
16 177 ms 180 ms 179 ms 0.so-0-0-0.TL2.DCAS.ALTER.NET [152.63.38.73]
17 181 ms 173 ms 179 ms 0.so-6-0-0.TL2.NYCB.ALTER.NET [152.63.13.10]
18 173 ms 179 ms 179 ms 0.so-5-0-0.XL2.NYCB.ALTER.NET [152.63.23.130]
19 171 ms 179 ms 179 ms POS7-0.GW6.NYCB.ALTER.NET [152.63.24.69]
20 181 ms 179 ms 179 ms rgh-13-gw.customer.ALTER.NET [157.130.18.70]
21 186 ms 185 ms 180 ms gw2.mcgraw-hill.com [198.45.19.20]
22 182 ms 179 ms 180 ms 198.45.19.151

Traza completa.
```

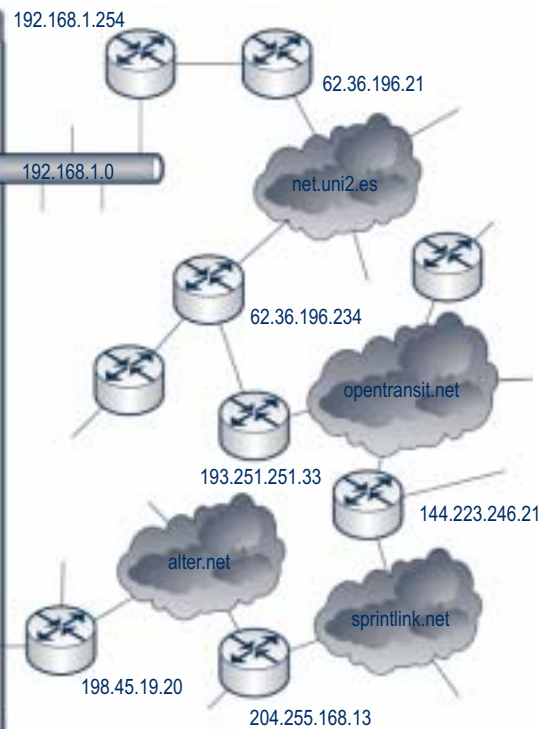


Fig. 5.30. Traza completa obtenida con el comando tracert.



- Si ya no hay respuesta en el salto 1, el problema lo tenemos en la puerta de enlace de nuestra propia red.
- Si la respuesta se pierde entre los saltos 2 y 4, quien está interrumpiendo la conexión es nuestro proveedor de acceso a Internet (en nuestro ejemplo, Uni2).
- En los saltos intermedios intervienen otras redes de tránsito de diferentes operadores. A veces se deja de tener respuesta porque están congestionadas de

tráfico. En estos casos es recomendable repetir la orden `tracert` pasado un tiempo.

- Si, por último, el tiempo de respuesta se excede en el salto 21, es el servidor de la página web a la que pretendemos acceder quien se encuentra con problemas.

`Tracert` simboliza los tiempos de espera agotados imprimiendo el carácter * (asterisco) en lugar del tiempo en milisegundos.

5.5 Multiplexación de datos

Cuando los datos pasan de la capa de transporte a la capa de aplicación, resulta imprescindible indicar con exactitud a qué servicio de red están destinados dichos datos.

Esto es posible, entre otras cosas, por la coordinación que realiza la autoridad de asignación de números en Internet **IANA** (*Internet Assigned Numbers Authority*), que mantiene un registro actualizado con los números de puerto asignados a cada servicio de red. Para una información más exhaustiva al respecto, se recomienda consultar el informe RFC 1700, en el cual se describe con todo detalle el proceso de asignación de puertos para el protocolo TCP/IP.

El número de puerto tiene un tamaño de 16 bits y siempre se trabaja con dos números: el primero es el número de puerto origen, que identifica al proceso que envía los datos; y el segundo es el número de puerto destino que identifica al puerto que los recibe. Ambos números se sitúan en la primera palabra de la cabecera tanto de los datagramas UDP como de los segmentos TCP.

En la Figura 5.31 el *host*, al que se ha denominado *pegasus*, (192.168.1.100) está utilizando dos servicios de red.

Por un lado mantiene una sesión telnet para configurar el *router* que actúa como puerta de enlace de la red, y al mismo tiempo con su navegador ha accedido a una página web. Para conocer el estado de ambas conexiones se puede emplear el comando `netstat` de varias formas. La Figura 5.32 muestra dos de las más empleadas.

Ambos servicios trabajan con el protocolo TCP. El comando `netstat` ha sido ejecutado desde el *host* *pegasus*, por ello su dirección IP aparece siempre en la columna de dirección local.

En la Figura 5.32 vemos que los puertos 23 (telnet) y 80 (http) son asociados a las direcciones de los *host* remotos, es decir, son puertos destino. Es necesario que exista también un puerto origen. En el *host* local se utiliza el puerto 1048 para la sesión telnet, y los puertos 1051 y 1052 para el servicio http.

Según la numeración de los puertos se hace la siguiente clasificación para los protocolos TCP y UDP:

- Los puertos con números inferiores a 1.024 están reservados para servicios muy definidos, como telnet, SMTP, POP3... Estas asignaciones son fijas y no pueden ser utilizadas por otros servicios. A menudo estos puertos son llamados «puertos bien conocidos».
- Los puertos numerados entre 1024 y 49151 son puertos registrados. Significa que la IANA intenta ordenar el uso de este rango, pero sin las restricciones que existen para los puertos bien conocidos.
- Por último, los puertos numerados entre 49152 y 65535 son puertos privados de los que se puede disponer para cualquier uso.



El registro IANA puede ser consultado en la página web de esta organización:

www.iana.org/assignments/port-numbers/

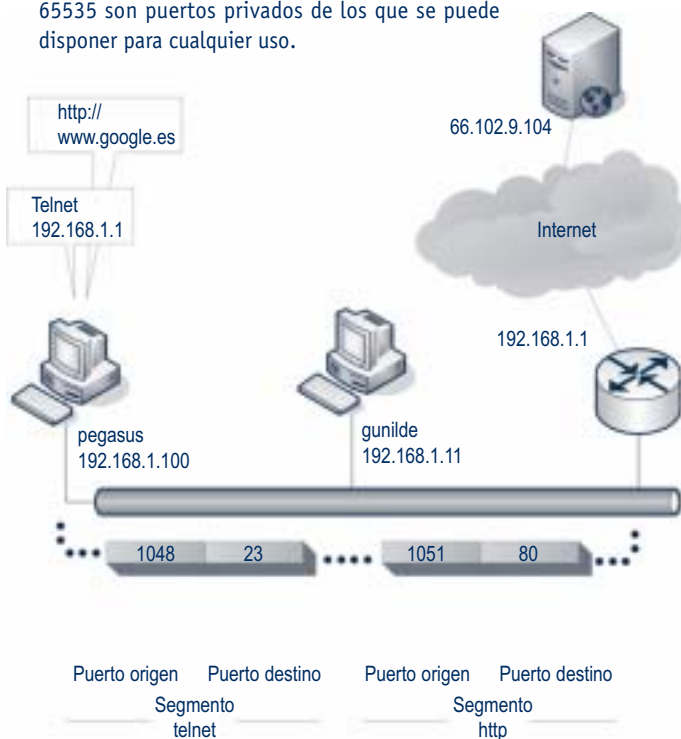


Fig. 5.31. Puertos origen y destino en segmentos TCP.



5. Protocolo TCP/IP

5.5 Multiplexación de datos

```

Simbolo de sistema
C:\>netstat -n

Conexiones activas

Proto Dirección local      Dirección remota      Estado
TCP    192.168.1.100:1048      192.168.1.1:23       ESTABLISHED
TCP    192.168.1.100:1051      66.102.9.104:80      ESTABLISHED
TCP    192.168.1.100:1052      66.102.9.104:80      ESTABLISHED

C:\>netstat

Conexiones activas

Proto Dirección local      Dirección remota      Estado
TCP    Pegasus:1048          192.168.1.1:telnet   ESTABLISHED
TCP    Pegasus:1051          66.102.9.104:http    ESTABLISHED
TCP    Pegasus:1052          66.102.9.104:http    ESTABLISHED
  
```

Fig. 5.32. Resultados impresos para netstat -n y netstat.

Aplicación	Puerto TCP	Puerto UDP
FTP	21	21
Telnet	23	23
SMTP	25	25
DNS	53	53
TFTP	69	69
HTTP	80	80
POP3	110	110
NetBIOS	139	139
SNMP	161	161

Tabla 5.1. Selección de puertos bien conocidos.

En la capa de transporte un segmento TCP puede emplear el mismo puerto que un datagrama UDP, sin que exista riesgo de confusión, ya que los procesos son identificados por la combinación protocolo-puerto. Algunos de los puertos TCP y UDP bien conocidos se encuentran en la Tabla 5.1, con su aplicación asociada. Muchas aplicaciones pueden funcionar con puertos TCP o UDP. En negrita se ha resaltado el protocolo que utiliza con mayor frecuencia cada una.

La Tabla 5.2 resume una selección de los puertos registrados para aplicaciones de descarga compartida, videoconferencia, control remoto, juegos, etcétera.

Aplicación	Puerto TCP	Puerto UDP
kaaza	1214	1214
MSN Messenger	1863	
VNC	5800+ y 5900+	
CU-SeeMe	7648 y 7649	7648 a 7652, 24
WinGate 2.1	8010	032
HTTP alternativo	8080	
Quake	26000 a 28000	8080

Tabla 5.2. Ejemplos de puertos registrados.

Algunas aplicaciones reservan distintos números de puerto, como por ejemplo CU-SeeMe, que recurre a puertos TCP y UDP. Otras utilizan puertos consecutivos a partir de un número concreto, como hace el programa VNC (desde los puertos 5800 y 5900). Y las hay que requieren rangos con miles de puertos (véase el ejemplo del juego Quake, desde el puerto 26000 al 28000).

Asignación dinámica de puertos

Los puertos bien conocidos simplifican las conexiones, ya que los equipos involucrados en ambos extremos saben de sobra a qué aplicación van destinados los datos.

Además de los puertos bien conocidos, cada conexión necesita asignar dinámicamente un segundo número de puerto. La elección de este número es aleatoria, y lógicamente no será una cifra inferior a 1024.

La principal ventaja de la asignación dinámica es que permite que un servicio soporte simultáneamente más de un usuario. En la Figura 5.32 se ve cómo el servicio http mantiene al mismo tiempo dos usuarios a los que se han asignado dinámicamente los puertos TCP 1051 y 1052.

Durante la negociación de una conexión en el protocolo TCP los *host* origen y destino se intercambian los números de puertos. La Figura 5.33 muestra un ejemplo de ello.

La combinación de la dirección IP y el número de puerto es denominada **socket**. Un *socket* identifica un proceso de red de manera única en Internet.

En el ejemplo de la Figura 5.33 para el equipo pegasus, un *socket* es 192.168.1.100.1051, es decir, su dirección IP seguida del puerto asignado dinámicamente; mientras que para el servidor web será 66.102.9.104.80.

Este par de *sockets* es conocido por ambos extremos una vez que finaliza con éxito la negociación y se establece la conexión, que es identificada por ellos. No existirá en Internet otra conexión que posea un par de *sockets* coincidentes con ésta.

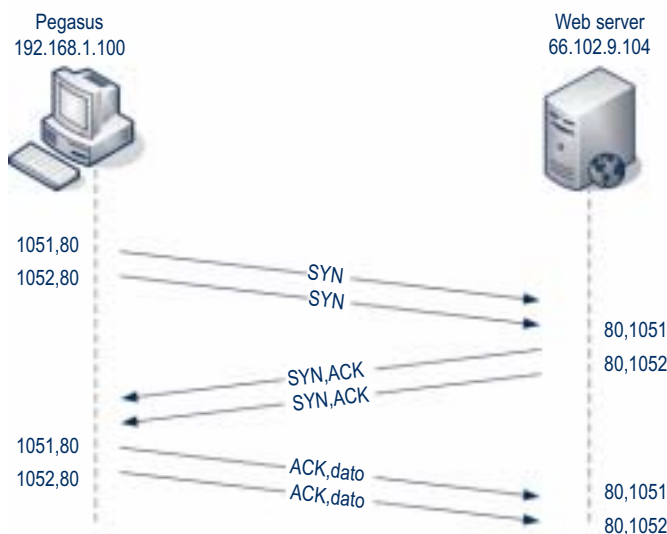


Fig. 5.33. Intercambio de puertos en la negociación TCP.



Caso práctico 4

Muchos usuarios de ordenadores conectados en red creen que sus equipos establecen por su cuenta conexiones con otros equipos remotos, y que transmiten datos sin su consentimiento; efectivamente, están en lo cierto.

Sistemas operativos como Windows XP mantienen abiertos muchos puertos como parte intrínseca de funcionamiento. Esto no supone un riesgo para el equipo, pero quien tenga dudas puede intentar disiparlas.

El comando `netstat` visualiza los números de puertos TCP locales y remotos, las direcciones IP de los *hosts* y el estado de las conexiones. Podemos ampliar esta información y ver también los puertos UDP (opción `-a`), o el identificador del proceso **PID** que utiliza cada puerto UDP y TCP (opción `-o`). La Figura 5.34 muestra una ejecución del comando `netstat` desde un equipo que tiene instalado el sistema Windows XP.

La Figura 5.34 muestra que en los puertos UDP no aparece el *socket* de dirección remota ni los estados de la conexión. Esto es lo normal, ya que el protocolo UDP no está orientado a conexión, y la única información de la que dispone el sistema en cuanto a los datagramas UDP es local. Con el protocolo TCP no sucede lo mismo: la negociación de la conexión a tres vías proporciona información acerca del *host* remoto y del estado de la conexión. Netstat puede presentar en la columna de estado estos valores:

- **SYN_SEND** indica que el puerto se abre y es activo.
- **SYN_RECEIVED:** el *host* remoto indica que ha recibido la secuencia de sincronismo (SYN) del *host* local.
- **ESTABLISHED:** el *host* local ha recibido la secuencia de sincronismo del *host* remoto: conexión establecida.
- **LISTENING:** el *host* remoto está preparado para aceptar conexiones en los puertos indicados.
- **TIME_WAIT** es un estado introducido por el *host* local justo antes de cerrar la conexión. A veces una aplicación libera un *socket* y netstat continua mostrándolo en el estado TIME_WAIT. No debe considerarse anómala esta situación siempre y cuando no se prolongue durante más de cuatro minutos.
- **CLOSE_WAIT** indica un estado de cierre de conexión; en concreto, aparece cuando el equipo remoto recibe la petición de cierre (FIN) del equipo local.

En nuestro caso vemos cómo tres procesos se encuentran dentro del estado ESTABLISHED. Por tanto, existen otras tantas conexiones activas con sistemas remotos. Analizando con detalle los *sockets* remotos se aprecia que se

Proto	Dirección local	Dirección remota	Estado	PID
TCP	0.0.0.0:135	0.0.0.0	LISTENING	732
TCP	0.0.0.0:445	0.0.0.0	LISTENING	4
TCP	0.0.0.0:1025	0.0.0.0	LISTENING	776
TCP	0.0.0.0:1046	0.0.0.0	LISTENING	1692
TCP	0.0.0.0:1047	0.0.0.0	LISTENING	1708
TCP	0.0.0.0:1049	0.0.0.0	LISTENING	804
TCP	0.0.0.0:5000	0.0.0.0	LISTENING	960
TCP	192.168.135.100:139	0.0.0.0	LISTENING	4
TCP	192.168.135.100:1046	86.102.9.99.80	ESTABLISHED	1692
TCP	192.168.135.100:1047	86.102.9.99.80	ESTABLISHED	1708
TCP	192.168.135.100:1048	216.49.88.118.80	TIME_WAIT	0
TCP	192.168.135.100:1049	216.49.88.121.80	ESTABLISHED	804
TCP	192.168.135.100:1050	216.49.88.118.80	TIME_WAIT	0
UDP	0.0.0.0:445	**	**	4
UDP	0.0.0.0:500	**	**	576
UDP	0.0.0.0:1026	**	**	802
UDP	0.0.0.0:1027	**	**	802
UDP	127.0.0.1:123	**	**	776
UDP	127.0.0.1:1044	**	**	1692
UDP	127.0.0.1:1045	**	**	1708
UDP	127.0.0.1:1900	**	**	960
UDP	192.168.135.100:123	**	**	776
UDP	192.168.135.100:137	**	**	4
UDP	192.168.135.100:138	**	**	4
UDP	192.168.135.100:1900	**	**	**

Puertos
UDP

Fig. 5.34. Puertos UCP, TCP e identificador de proceso PID.

corresponden con el puerto 80 (http) y que son utilizados por los procesos 1692, 1708 y 804 respectivamente. ¿Cómo saber de qué procesos se trata? Windows XP permite conocerlo gracias al *Administrador de tareas*, que aparece al pulsar **CTRL+ALT+SUPR.**

Por defecto el administrador de tareas de Windows no incluye dentro de la ficha denominada *Procesos* la columna con los identificadores de proceso PID. Tendremos que pulsar la opción *Ver* en el menú principal y escoger en el submenú desplegable *Seleccionar columnas*. Vemos una serie de opciones y señalamos la del identificador de proceso (PID).

La Figura 5.35 muestra la información de los procesos actuales en el equipo donde se ha ejecutado `netstat -aon`.

Al revisar los valores se ve que los PID 1.692 y 1.708 pertenecen al programa `iexplorer.exe` (la aplicación Internet Explorer). El otro PID que usa el puerto 80 es 804: la tabla lo asocia al programa ejecutable `mcupdate.exe` (lo emplea el antivirus McAfee para actualizaciones automáticas).



5. Protocolo TCP/IP

5.5 Multiplexación de datos



Caso práctico 4 (cont.)

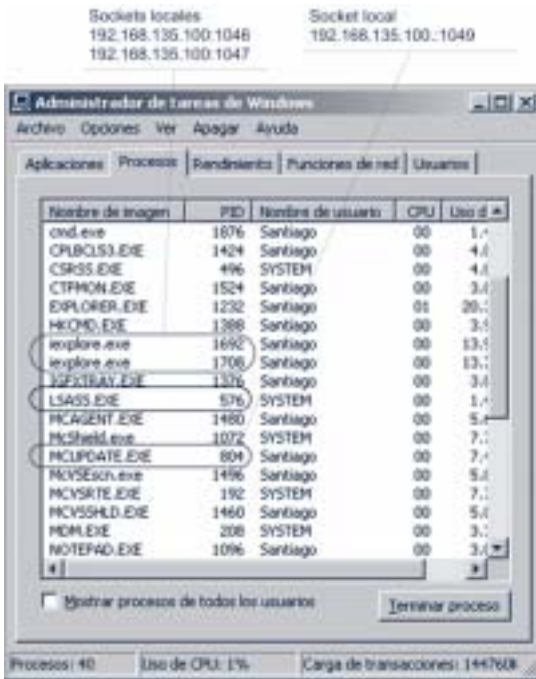


Fig. 5.35. Administrador de tareas de Windows XP.

Parece, por tanto, que los tres procesos son seguros.

El análisis de las tablas de procesos se debe hacer con paciencia y cierta intuición. En este sentido ayuda mucho la experiencia, porque no siempre todo es lo que parece ser.

Nos fijamos ahora en un proceso que casi pasa inadvertido: LSASS.EXE con el PID 576 asociado. ¿Por qué este proceso? Quien lo conoce lo sabe bien, y quien no, seguro que ha oído hablar del gusano W32-Sasser.

El programa original LSASS.EXE, es un proceso que Windows emplea para verificar la identidad de los usuarios que acceden al sistema. Lamentablemente W32-Sasser, al igual que otros gusanos, suplanta la identidad de este archivo, abre una conexión simulando ser un servidor FTP a través del puerto TCP 5554 y crea un acceso remoto en el puerto TCP 9996. Este virus también escanea las direcciones IP de puertos TCP que se encuentren en el estado LISTENING a partir del puerto 1068.

Afortunadamente para el usuario de este sistema, LSASS.EXE está utilizando el puerto UDP 500, así que no se trata de un equipo infectado. Otra cosa muy distinta sería que viésemos al PID 576 asociado con un *socket* cuya dirección IP y puerto TCP se correspondiese con lo descrito para el famoso gusano.

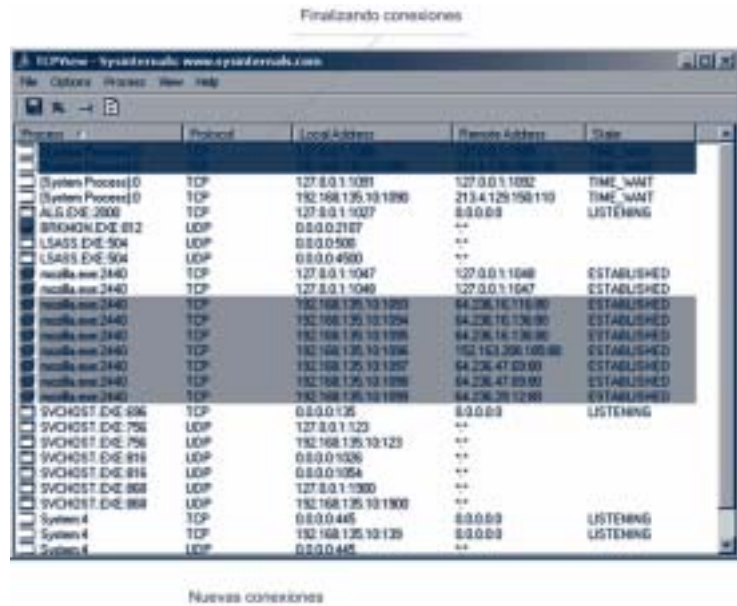


Fig. 5.36. Pantalla principal del programa TCPView.

El análisis continuo de las conexiones abiertas y los puertos utilizados es ya una tarea rutinaria para los programas antivirus. En este sentido nets-tat se queda corto, pues su ejecución cuando existen muchas conexiones abiertas es lenta y la actualización de su información no es automática. Afortunadamente existen multitud de programas, la mayoría de ellos de distribución gratuita, que permiten visualizar el estado de los puertos en tiempo real y con detalle.

Para este caso práctico se ha elegido TCPView v2.34 de Sysinternals (www.sysinternals.com/ntw2k/utilities.shtml). Al iniciar el programa una pantalla muestra todos los puertos UDP y TCP que se encuentran activos y qué procesos los están utilizando. Además resuelve nombres de dominio y *host* en direcciones IP, e informa del estado de las conexiones IP (véase la Fig. 5.36).

Por defecto TCPView actualiza los datos cada segundo, variando su presentación: cuando una conexión cambia de estado entre actualizaciones es resaltada en amarillo; cuando finaliza o va ser borrada, en rojo; y las nuevas conexiones activas, en verde.

Con este programa se evita recurrir al administrador de tareas para conocer el proceso asociado al identificador PID. TCPView muestra en la primera columna el nombre del programa o proceso, seguido del código PID.

También podemos decidir finalizar uno de estos procesos o concluir una conexión. Basta con resaltar la entrada en cuestión, hacer clic sobre el botón derecho del ratón y elegir la opción deseada.



Actividad práctica 1



Objetivos

- Determinar que tipo de máscara de subred se debe emplear para asignar una cantidad adecuada de *hosts* y de subredes.
- Comprobar la segmentación efectiva de una subred de clase C empleando las herramientas de diagnóstico más adecuadas.

Materiales

- Una red LAN, compuesta al menos por ocho equipos a los que se pueda asignar una dirección IP. Resulta indiferente si los medios de transmisión son cableados o inalámbricos.
- El sistema operativo no tiene por que ser el mismo en todos los equipos, sólo es necesario que funcionen con el protocolo TCP/IP y que permitan ejecutar sus comandos.

Tiempo estimado: 45 min.

Enunciado

En una LAN con dirección 192.168.1.0 se pretende hacer una segmentación IP de cuatro subredes para mejorar la eficiencia de la misma.

Procedimiento

- a) Determinar la máscara de subred que permita la creación de las cuatro subredes. Anotar su valor primero como número binario y después en notación binario decimal.
- b) Calcular para la subred 1, la dirección de red, dirección de difusión, primera y última dirección válida de *host*.
- c) Repetir el punto anterior para el resto de las subredes, a las que denominaremos: subred 2, subred 3 y subred 4.
- d) Introducir en la configuración TCP/IP de cada uno de los equipos la máscara de subred calculada en el primer apartado. Modificar sus direcciones IP de tal manera que cada subred esté compuesta al menos por dos de ellos.
- e) Una vez introducidos los cambios debemos reiniciar los equipos en los que el sistema operativo lo demande para que tengan efecto.

- f) Comprobar la conectividad IP mediante el comando `ping` entre equipos de la misma subred. Visualizar las tablas dinámicas ARP antes y después de cada prueba. ¿Aparece una nueva entrada ARP después de cada `ping`? En caso de que la respuesta sea negativa, se deben revisar todas las conexiones físicas y comprobar que los diodos led de enlace en los adaptadores de red están encendidos.
- g) Ejecutar los comandos `ipconfig` y/o `wiipcfg` desde cada equipo de la red. Comprobar que la máscara de subred es la misma para todos ellos, que la dirección de *host* se encuentra dentro del rango adecuado de su subred, y que no aparece dirección alguna para la puerta de enlace.
- h) Una vez verificada la conectividad entre equipos de la misma subred comprobar que realmente existe segmentación IP entre equipos que pertenezcan a distintas subredes. Ejecutar el comando `ping` desde un equipo de la subred 1 a direcciones IP de las subredes 2, 3 y 4.

Observaciones

- Puede suceder que se disponga de equipos en los que el sistema operativo no sea el mismo; por ejemplo: Windows en unos y Linux en otros, o que las versiones sean distintas. Todo ello no modifica el proceso de esta actividad siempre y cuando el protocolo instalado en todos ellos sea TCP/IP.
- Si en el apartado 6 del procedimiento se detectan errores de conectividad, primero se revisará que todos los equipos están correctamente conectados a nivel físico, después se pasará el apartado g y se revisará la configuración TCP/IP de cada equipo antes de volver de nuevo al apartado f.
- Al consultar las tablas ARP que se guardan en la memoria caché de cada adaptador de red, se debe tener en cuenta que la mayoría de ellos sólo almacena cuatro entradas.
- Al modificar las direcciones IP y máscaras de subred de cada equipo comprobaremos que en ninguno de ellos hay una dirección IP especificada para la puerta de enlace.
- Para identificar adecuadamente los mensajes esperados después de ejecutar el comando `ping`, se recomienda consultar los casos prácticos uno y dos de esta unidad.
- Para responder a los apartados 2 y 3 se recomienda utilizar el informe RFC 1878. Dentro de él hay una tabla (Tabla 1-2) con todas las combinaciones posibles. Considerar que el identificador genérico N.N.N de la tabla equivale a 192.168.1 en esta actividad.



5. Protocolo TCP/IP

Actividades prácticas


Actividad práctica 2



Objetivos

- Emplear utilidades *software* para realizar pruebas de conectividad IP y medir la eficiencia de la red.

Materiales

- Una LAN con un esquema similar al visto en el Caso práctico 1 (se puede prescindir de la parte Wi-Fi). Los datos de configuración LAN pueden ser los mismos.
-  Utilidad de diagnóstico QCheck. Esta utilidad se puede descargar desde <http://www.netiq.com/qcheck/>

Tiempo estimado: 30 min.

Enunciado

Sobre una configuración de red adaptada del Caso práctico 1 (véase la Figura 5.7) realizaremos pruebas de conectividad, eficiencia en la tasa de transferencia de datos para los protocolos TCP y UDP.

Procedimiento

- Instalar en cada uno de los equipos de la red el programa de diagnóstico QCheck.
- Ejecutar desde uno de los equipos de la red cableada Ethernet la utilidad QCheck. Aparecerá la consola mostrada en la Figura 5.37.
- Para medir la eficiencia de la red con otro equipo de la red inalámbrica hacemos clic sobre los botones TCP y *Throughput* de la consola. El campo *From Endpoint 1* tendrá el valor *localhost* y en el campo *To Endpoint 2* tecleamos la dirección IP de uno de los equipos de la red inalámbrica, por ejemplo 192.168.135.101. Para comenzar el diagnóstico hacemos clic sobre el botón *Run*. Anotamos el valor obtenido. (El tamaño de los datos empleados será de 100 Kbps).
- Realizar el apartado c cambiando al protocolo UDP.
- Comprobar los apartados 3 y 4 haciendo las pruebas en dirección contraria. Es decir, ejecutando la utilidad *Check* desde el otro equipo.
- Repite los apartados 3, 4 y 5 de la siguiente manera: primero entre dos equipos de la red inalámbrica, y después entre dos equipos de la red cableada. Comparar los resultados obtenidos.

- Realiza las pruebas de conectividad desarrolladas en el Caso práctico 1, utilizando en esta ocasión QCheck.
- Para ello hacemos clic sobre los botones TCP y *Response Time* de la consola, en el campo *From Endpoint 1* introducimos la dirección IP del *host* origen, y en el campo *To Endpoint 2* la dirección IP del *host* destino. La aplicación QCheck se ejecuta desde el *host* origen.

Observaciones

- QCheck realiza las pruebas de diagnóstico sólo entre dos equipos cada vez. Es imprescindible que la utilidad esté instalada en ambos, por lo que se recomienda instalarlo en todos los equipos de la red.
- Cada vez que se haga una prueba, la aplicación se ejecuta en uno solo de los extremos, en el otro QCheck se ha ejecutado en memoria residente al iniciar el PC.
- Los resultados obtenidos en las medidas de eficiencia de la red se corresponden con el momento concreto en el que se hacen. Para evitar resultados no representativos se realizarán en diferentes ocasiones y admitiremos como válido el que se repita con más frecuencia.



Fig. 5.37. Consola de la aplicación QCheck.



Preguntas de evaluación



- 1 ¿Qué diferencias hay entre una dirección física y una dirección lógica?
- 2 ¿Por qué el protocolo TCP se considera más fiable que el protocolo UDP?
- 3 ¿Cómo se denomina a los paquetes de datos en cada capa del protocolo TCP/IP?
- 4 ¿Cómo se consigue separar en una dirección IP la dirección de red de la dirección de *host*?
- 5 En una dirección IP escrita en binario, ¿qué reglas se siguen para determinar a qué clase pertenece?
- 6 En una dirección IP escrita en notación punto decimal, ¿qué reglas se siguen para determinar a qué clase pertenece?
- 7 ¿Qué ventajas se obtienen al segmentar redes?
- 8 ¿En qué capa opera el protocolo ARP? ¿Cuál es su principal cometido?
- 9 Describe los pasos que se siguen durante una negociación a tres vías del protocolo TCP.
- 10 ¿Qué mensajes ICMP emplea el comando `ping`?
- 11 Enumera varios de los servicios de red disponibles en la capa de aplicación del protocolo TCP/IP.
- 12 Cuando se habla de números de puertos TCP o UDP, ¿qué significan los denominados «bien conocidos»?
- 13 En una conexión TCP, ¿en cuál de los extremos se realiza una asignación dinámica de puerto?
- 14 ¿Qué significa el término *socket* en una conexión TCP?
- 15 ¿Para qué sirven las tablas de rutas?
- 16 Explica como se puede determinar la traza de una ruta.
- 17 Las conexiones del protocolo TCP que muestra el comando `netstat` están acompañadas de un estado. ¿Cuántos estados posibles hay? ¿Cuál es su significado?
- 18 El comando `netstat` puede mostrar un código de identificación de proceso para cada puerto activo. Indica al menos dos formas de averiguar a qué aplicación se corresponde ese código PID.

Ejercicios



- 1 Calcula el número de redes que puede haber en cada una de las clases de direcciones IP (clases A, B y C).
- 2 Calcula el número de *hosts* que puede haber en cada una de las clases de direcciones IP (clases A, B y C).
- 3 Indica a qué clase pertenece cada una de las siguientes direcciones IP:
 - a) 64.81.234.120
 - b) 124.23.45.28
 - c) 192.154.23.14
- 4 ¿Cuántas subredes crea cada una de estas máscaras al ser asociada a una dirección IP de clase C?
 - a) 255.255.255.192
 - b) 255.255.255.224
 - c) 255.255.255.248
- 5 Indica para cada una de las siguientes direcciones IP, cuál es el identificador de red y cuál es el identificador de *host*:
 - a) 127.0.0.1
 - b) 66.102.9.104
 - c) 245.34.67.14
 - d) 122.23.9.167
 - e) 171.34.59.201
 - f) 114.50.34.29