- Trabajo realizado por el equipo de la Biblioteca Digital de la Fundación Universitaria San Pablo-CEU
- Me comprometo a utilizar esta copia privada sin finalidad lucrativa, para fines de investigación y docencia, de acuerdo con el art. 37 del T.R.L.P.I. (Texto Refundido de la Ley de Propiedad Intelectual del 12 abril 1996)

Capítulo 7

PROTOCOLOS DE RED Y DE TRANSPORTE (II)

TCP/IP

El nombre *TCP/IP* proviene de dos de los protocolos más importantes de la familia de protocolos *Internet*, el *Transmission Control Protocol* (*TCP*) y el *Internet Protocol* (*IP*).

La principal virtud de *TCP/IP* estriba en que está diseñado para enlazar ordenadores de diferentes tipos, incluyendo *PCs*, minis y *mainframes*, que ejecuten sistemas operativos distintos, sobre redes de área local y redes de área extensa y, por tanto, permite la conexión de equipos distantes geográficamente.

Otro gran factor que ha permitido su expansión es la utilización de *TCP/IP* como estándar de *Internet*.

El mayor problema de *TCP/IP* estriba en la dificultad de su configuración, por lo que no es recomendable su uso para utilizarlo en una red pequeña.

TCP/IP fue desarrollado en 1972 por el Departamento de Defensa de los Estados Unidos, ejecutándose en *ARPANET* (una red de área extensa del Departamento de Defensa). Posteriormente, una red dedicada exclusivamente a aspectos militares denominada *MILNET* se separó de *ARPANET*. Fue el germen de lo que después constituiría *Internet*.

La arquitectura *TCP/IP* transfiere datos mediante el ensamblaje de datos en paquetes. Cada paquete comienza con una cabecera que contiene información de control seguida de los datos.

El *Internet Protocol* (*IP*), un protocolo del nivel de red de *OSI*, permite a las aplicaciones ejecutarse de forma transparente sobre las redes interconectadas. De esta forma, las aplicaciones no necesitan conocer qué *hardware* está siendo utilizado en la red y, por tanto, la misma aplicación puede ejecutarse en cualquier arquitectura de red.

El *Transmission Control Protocol* (*TCP*), un protocolo del nivel de transporte de *OSI*, asegura que los datos sean entregados, que lo que se recibe corresponde con lo que se envió y que los paquetes sean reensamblados en el orden en que fueron enviados.

UNIX se empezó a comercializar como el principal sistema operativo que utilizaba *TCP/IP* y llegaron a ser sinónimos.

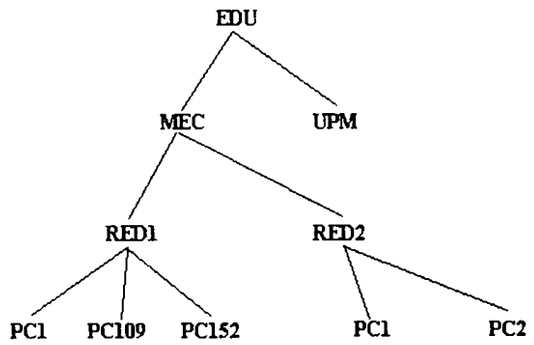
Cómo denominar a un ordenador en TCP/IP

Es importante que se establezca la identificación de la estación de trabajo de una forma que evite su duplicidad dentro de todos los ordenadores que puedan conectarse.

Para ello, en *TCP/IP*, se utiliza el nombre del usuario y el nombre del dominio de la red.

Para identificar al usuario es necesario nombrarlo evitando que pueda haber dos con el mismo nombre y produzca confusiones al servidor de la red.

Para identificar a la red se utiliza el concepto de dominio. La estructura del dominio se asemeja a un árbol invertido (es decir, el tronco se encuentra en la parte superior y las ramas en la parte inferior) y cada hoja corresponde a un dominio.



La identificación de un dominio está formada por varios apartados separados por un punto (por ejemplo, *RED1.MEC.EDU*). Cada uno de ellos recibe el nombre de subdominio. El subdominio situado más a la derecha es el de carácter más general y recibe el nombre de dominio de nivel alto.

El nombre de un dominio completamente calificado (*FQDN, Full Qualified Domain Name*) ha de empezar por el nombre de la estación de trabajo (*HOST*), un punto, y el nombre de la red (*DOMINIO*). Por ejemplo, si se denomina al *PC* como *PC109* y a la red principal como *RED1*, la identificación completa de la estación de trabajo sería *PC109.RED1*.

Si, a su vez, esta red formara parte de otra red superior, se volvería a poner otro punto y el nombre de dicha red (por ejemplo, *PC109.RED1.MEC*). En este caso, después del *HOST* vendría el *SUBDOMINIO* (es posible tener varios niveles de subdominios) y, para finalizar, el *DOMINIO*.

También es interesante identificar a la institución de la que forma parte la red, así como la organización o el país a la que pertenece. Para ello, se le habrán de añadir estos dos nuevos conceptos separados, también, por puntos.

DOMINIO DE ALTO NIVEL DE ORGANIZACIÓN	
DOMINIO	**SIGNIFICADO**
com	Organización comercial
edu	Institución educativa
gov	Institución gubernamental
int	Organización internacional
mil	Organización militar
net	Organización de red
org	Organización sin ánimo de lucro
es	Organización española

Si se toma como ejemplo la identificación *RODRIGUEZJL@PC109.RED1.MEC.EDU* se ve que el usuario (*RODRIGUEZJL*) se separa con una arroba del dominio, que está formado por el nombre de la estación (*PC109*), de la red (*RED1*), de la institución (*MEC*) y de la organización (*EDU*).

Existe una institución que se encarga del registro de todas las direcciones *IP* y sus correspondientes dominios que se denomina *INTERNIC* y que ha delegado para España sus funciones en *REDIRIS*.

Es necesario hacer constar que la definición de dominio dada en este apartado no tiene nada que ver con los dominios de redes locales definidos en *Windows NT*.

Direcciones IPv4

Las direcciones *IP* consiguen que el envío de datos entre ordenadores se realice de forma eficaz, de forma parecida a como se utilizan los números de teléfono en las llamadas telefónicas.

Actualmente, las direcciones *IP* de la versión actual (*Ipv4*) tienen 32 *bits*, formados por cuatro campos de 8 *bits* (octeto), cada uno, separados por puntos.

Por tanto, las direcciones *IP* están en representación binaria (por ejemplo, 01111111.00000000.00000000.00000001). Cada uno de los campos de 8 *bits* puede tener un valor que esté comprendido entre 00000000 (cero en decimal) y 11111111 (255 en decimal).

Normalmente y debido a la dificultad del sistema binario, la dirección *IP* se representa en decimal. Por ejemplo, la dirección *IP* indicada anteriormente 01111111.00000000.00000000.00000001 (en representación binaria) tiene su correspondencia con 127.0.0.1 (en representación decimal).

La forma de pasar de un sistema binario a un sistema decimal se hace por potencias de dos en función de la posición de cada uno dentro del octeto, correspondiendo cero a la primera posición a la derecha y siete a la primera posición de la izquierda (por ejemplo, 00000001 corresponde a 1 ya que 2^0=1, 00000010 corresponde a 2 ya que 2^1=2 y 00001000 corresponde a 8 ya que 2^3=8).

Si hay varios unos en el octeto, se deberán sumar los resultados de las potencias de dos correspondientes a su posición (por ejemplo, 00001001 corresponde a 9 ya que 2^3+2^0=8+1=9 y 01001001 corresponde a 73 ya que $2^6+2^3+2^0$=64+8+1=73).

Los cuatro octetos de la dirección *IP* componen una dirección de red y una dirección de equipo que están en función de la clase de red correspondiente.

Existen cinco clases de redes: *A*, *B*, *C*, *D* o *E* (esta diferenciación viene dada en función del número de ordenadores que va a tener la red).

- La **clase A** contiene 7 *bits* para direcciones de red (el primer *bit* del octeto siempre es un cero) y los 24 *bits* restantes representan a direcciones de equipo. De esta manera, permite tener un máximo de 128 redes (aunque en realidad tienen 126, ya que están reservadas las redes cuya dirección de red empieza por cero y por 127), cada una de las cuales puede tener 16.777.216 ordenadores (aunque en realidad tienen 16.777.214 ordenadores cada una, ya que se reservan aquellas direcciones de equipo, en binario, cuyos valores sean todos ceros o todos unos). Las direcciones, en representación decimal, estarán comprendidas entre 0.0.0.0. y 127.255.255.255 y la máscara de subred será de 255.0.0.0.

- La **clase B** contiene 14 *bits* para direcciones de red (ya que el valor de los dos primeros *bits* del primer octeto ha de ser siempre 10) y 16 *bits* para direcciones de equipo, lo que permite tener un máximo de 16.384 redes, cada una de las cuales puede tener 65.536 ordenadores (aunque en realidad tienen 65.534 ordenadores cada una, ya que se reservan aquellas direcciones de equipo, en binario, cuyos valores sean todos ceros o todos unos). Las direcciones, en representación decimal, estarán comprendidas entre 128.0.0.0 y 191.255.255.255 y su máscara de subred será de 255.255.0.0.

- La **clase C** contiene 21 *bits* para direcciones de red (ya que el valor de los tres primeros *bits* del primer octeto ha de ser siempre 110) y 8 *bits* para direcciones de equipo, lo que permite tener un máximo de 2.097.152 redes, cada una de las cuales puede tener 256 ordenadores (aunque en realidad tienen 254 ordenadores cada una, ya que se reservan aquellas direcciones de equipo, en binario, cuyos valores sea todos ceros o todos unos). Las direcciones, en representación decimal, estarán comprendidas entre 192.0.0.0 y 223.255.255.255 y su máscara de subred será de 255.255.255.0.

- La **clase D** se reserva todas las direcciones para multidestino (*multicasting*), es decir, un ordenador transmite un mensaje a un grupo específico de ordenadores de esta clase. El valor de los cuatro primeros *bits* del primer octeto ha de ser siempre 1110 y los últimos 28 *bits* representan los grupos multidestino. Las direcciones, en representación decimal, estarán comprendidas entre 224.0.0.0 y 239.255.255.255.

- La **clase E** se utiliza con fines experimentales únicamente y no está disponible para el público. El valor de los cuatro primeros *bits* del primer octeto ha de ser siempre 1111 y las direcciones, en representación decimal, estarán comprendidas entre 240.0.0.0 y 255.255.255.255.

La dirección de equipo indica el número que corresponde al ordenador dentro de la red (por ejemplo, al primer ordenador de una dirección de red de clase *C* 192.11.91 se le otorgará la dirección *IP* 192.11.91.1, al segundo 192.11.91.2, al cuarto 192.11.91.4 y así sucesivamente).

Segmentación de la red

Actualmente debido al uso masivo de aplicaciones cliente-servidor y multimedias que requieren la transmisión de grandes volúmenes de información, la tecnología de redes de área local, que en algunos casos data desde hace unos 20 años, se ha visto en la necesidad de transmitir un gran volumen de datos y con mayor rapidez. Esta necesidad ha obligado a buscar tecnologías que permitan aumentar el ancho de banda y mejorar, e incluso intentar asegurar, los tiempos de respuestas.

Como se ha visto anteriormente, la red *Ethernet* funciona a una velocidad de 10 *Mbps*, implementada en una topología física de configuración en bus o en una topología física de configuración en *estrella* dentro de una topología lógica de *bus*

DETERMINAR EL NÚMERO DE SUBREDES NECESARIAS

El primer paso a seguir cuando se desea segmentar una red, es decidir el número de subredes que se necesitan y, así, establecer las direcciones *IP* de cada subred y su mascara correspondiente.

Si se toma, como ejemplo, que la red que se va a segmentar es una clase *B* (con máscara de red 255.255.0.0) y con dirección 164.56.0.0 (en representación decimal), resulta que su dirección (en representación binaria) es:

10100100	00111000	00000000	000000000

Una vez realizadas las evaluaciones pertinentes, se considera que con ocho subredes es suficiente para cubrir las necesidades actuales. El primer paso a seguir es convertir el número decimal 8 a su representación binaria (1000).

El número binario 1000 necesita cuatro *bits* para representarse y, por tanto, se han de tomar cuatro *bits* de la dirección de equipo para indicar la dirección de subred.

La dirección en representación binaria del ejemplo que se está indicando será:

10100100	00111000	1000	0000	00000000

Y su máscara de red es 255.255.240.0 que corresponde a:

11111111	11111111	1111	0000	00000000

que indica que hay 20 *bits* para marcar la dirección de red y 12 *bits* para la dirección de equipo (el tercer octeto será 11110000 que corresponde a 240 en decimal).

Otra manera de representar la máscara de subred es indicarla en notación alternativa indicando la dirección *IP* en decimal de la red y el número de *bits* que se toman para indicar la dirección de red (en el ejemplo, sería 164.56.0.0/20).

Otro aspecto a considerar es el número de subredes posibles que se pueden tener con la máscara 255.255.240.0.

00000000	0
00010000	16
00100000	32
00110000	48
01000000	64

01010000	80
01100000	96
01110000	112
10000000	128
10010000	144
10100000	160
10110000	176
11000000	192
11010000	208
11100000	224
11110000	240

Como se puede observar, hay 16 posibles combinaciones que se pueden obtener utilizando los primeros cuatro *bits* del octeto. Pero no todas las combinaciones son susceptibles de utilización. Así, la combinación con todos ceros (dirección 164.56.0.0/20) no se puede utilizar porque es equivalente a la de la dirección 164.56.0.0/16 y puede ocasionar problemas a los protocolos de encaminamiento y, de la misma manera, la combinación con todos unos daría una dirección de difusión (164.56.255.255) equivalente a la dirección de difusión de la dirección 164.56.0.0/16. Por lo que quedarían 14 subredes posibles.

Se puede usar la ecuación $2^n - 2$ para determinar el número de subredes que se pueden obtener (*n* indica el número de *bits* que se va a utilizar).

De esta manera, se obtienen las subredes que se indican en la tabla siguiente:

1	-
2	2
3	6
4	14
5	30
6	62
7	126
8	254

Ahora se deberá considerar si las subredes que se necesitan actualmente (8) y las posibles combinaciones que se pueden obtener con los cuatro *bits* (14) son suficientes para las necesidades futuras o se debe ampliar el número de *bits* que se pasan pasar a dirección de subred.

DETERMINAR EL NÚMERO DE EQUIPOS DISPONIBLES

El siguiente paso en determinar el número de equipos disponibles en cada segmento de la red que está en función del número de *bits* que se han dejado para determinar la dirección de equipo (en el ejemplo anterior es 12).

Para ello, se utiliza la misma formula anterior ya que tampoco se puede utilizar las direcciones de equipos con todos ceros o todos unos. De esta manera, se obtiene que con 12 *bits* se puede disponer de 4.094 equipos en cada segmento.

ESTABLECIENDO EL DEPÓSITO DE DIRECCIONES IP

El último paso es identificar las direcciones *IP* que se pueden usar en los segmentos de red y que estará determinado por la primera dirección *IP* de la red (en el ejemplo, 164.56.0.0) y su máscara de subred (255.255.240.0 ó 164.56.0.0/20).

De esta manera, se obtiene la tabla siguiente con las 16 posibles combinaciones de direcciones de red (para esta máscara):

~~10100100 00111000 00000000 00000000~~	~~164.56.0.0~~
10100100 00111000 00010000 00000000	164.56.16.0
10100100 00111000 00100000 00000000	164.56.32.0
10100100 00111000 00110000 00000000	164.56.48.0
10100100 00111000 01000000 00000000	164.56.64.0
10100100 00111000 01010000 00000000	164.56.80.0
10100100 00111000 01100000 00000000	164.56.96.0
10100100 00111000 01110000 00000000	164.56.112.0
10100100 00111000 10000000 00000000	164.56.128.0
10100100 00111000 10010000 00000000	164.56.144.0
10100100 00111000 10100000 00000000	164.56.160.0
10100100 00111000 10110000 00000000	164.56.176.0
10100100 00111000 11000000 00000000	164.56.192.0
10100100 00111000 11010000 00000000	164.56.208.0
10100100 00111000 11100000 00000000	164.56.224.0
~~10100100 00111000 11110000 00000000~~	~~164.56.240.0~~

De ella se eliminan la primera y la última dirección (según se indicó anteriormente) por lo que quedan 14 combinaciones.

La segunda dirección a determinar será la dirección de difusión para cada una de las posibles redes (esta dirección corresponde a poner todos unos en los *bits* de equipo).

De esta manera, se obtiene la tabla siguiente con las 14 posibles combinaciones de direcciones de difusión (para esta máscara):

10100100 00111000 00011111 11111111	164.56.31.255
10100100 00111000 00101111 11111111	164.56.47.255
10100100 00111000 00111111 11111111	164.56.63.255
10100100 00111000 01001111 11111111	164.56.79.255

10100100 00111000 01011111 11111111	164.56.95.255
10100100 00111000 01101111 11111111	164.56.111.255
10100100 00111000 01111111 11111111	164.56.127.255
10100100 00111000 10001111 11111111	164.56.143.255
10100100 00111000 10011111 11111111	164.56.159.255
10100100 00111000 10101111 11111111	164.56.175.255
10100100 00111000 10111111 11111111	164.56.191.255
10100100 00111000 11001111 11111111	164.56.207.255
10100100 00111000 11011111 11111111	164.56.223.255
10100100 00111000 11101111 11111111	164.56.239.255

Utilizando las dos tablas se obtiene el depósito de direcciones *IP* que se puede utilizar en cada una de las 14 combinaciones posibles (para esta máscara):

164.56.16.0	164.56.16.1	164.56.31.254	164.56.31.255
164.56.32.0	164.56.32.1	164.56.47.254	164.56.47.255
164.56.48.0	164.56.48.1	164.56.63.254	164.56.63.255
164.56.64.0	164.56.64.1	164.56.79.254	164.56.79.255
164.56.80.0	164.56.80.1	164.56.95.254	164.56.95.255
164.56.96.0	164.56.96.1	164.56.111.254	164.56.111.255
164.56.112.0	164.56.112.1	164.56.127.254	164.56.127.255
164.56.128.0	164.56.128.1	164.56.143.254	164.56.143.255
164.56.144.0	164.56.144.1	164.56.159.254	164.56.159.255
164.56.160.0	164.56.160.1	164.56.175.254	164.56.175.255
164.56.176.0	164.56.176.1	164.56.191.254	164.56.191.255
164.56.192.0	164.56.192.1	164.56.207.254	164.56.207.255
164.56.208.0	164.56.208.1	164.56.223.254	164.56.223.255
164.56.224.0	164.56.224.1	164.56.239.254	164.56.239.255

CONSTRUIR UNA TABLA DE SUBREDES

Todo el proceso que se ha estado siguiendo para realizar los cálculos anteriores para una determinada máscara ha sido complejo. Se puede construir una tabla rápida que simplifique los cálculos a seguir.

Esta tabla de doble entrada tendrá en la fila superior el número de *bits* de la dirección de equipo tomados para la dirección de subred y las tres columnas de información siguientes:

Incremento								
Máscara de subred								
N° de redes								

En la fila *Incremento* se indicará el valor decimal correspondiente al *bit* dentro del octeto (recuerde que este número se calcula con 2^n donde *n* toma los valores de cero a siete de derecha a izquierda). De esta manera, los valores serán los siguientes:

Incremento	128	64	32	16	8	4	2	1
Máscara de subred								
Nº de redes								

Este valor va a determinar el valor de incremento de las direcciones de inicio del depósito de direcciones de cada subred.

La segunda fila determina el valor de la máscara de subred y está basado en el número de *bits* de la dirección de equipo tomados para la dirección de subred. Este valor se calcula sumando los valores de incremento de los *bits* que se toman y que se muestran en la tabla siguiente:

Incremento	128	64	32	16	8	4	2	1
Máscara de subred	128	192	224	240	248	252	254	255
Nº de redes								

La última fila es el número de redes que se pueden obtener en cada caso (se calcula con la formula $2^n - 2$ donde *n* toma los valores de los *bits* tomados):

Incremento	128	64	32	16	8	4	2	1
Máscara de subred	128	192	224	240	248	252	254	255
Nº de redes	0	2	6	14	30	62	126	254

Usar la tabla de subredes para una clase A

Ahora se va a mostrar un ejemplo de utilización de la tabla anterior para una segmentación de una dirección de clase *A*.

Se dispone de la dirección *IP* 56.0.0.0 para una red de clase *A* y se desea segmentar la red en 3 subredes de 6.000 equipos cada una que podrían aumentar a 12.000 equipos en los próximos años.

Para ello, fíjese en la fila *Número de redes* y busque el valor correspondiente a 3 (o su superior inmediato). Ese valor corresponde a tomar 3 *bits* para la dirección de subred (que permite tener hasta 6 subredes con lo cual se podría aumentar la red en un futuro).

Ahora ya se sabe que la máscara de subred que se va a disponer es 255.224.0.0 (o su notación alternativa que es 56.0.0.0/11).

Al tener que tomar 11 *bits* para la dirección de red y subred, se obtienen 21 *bits* para la dirección de equipo con lo que se pueden tener 2^{21}-2=2.097.150 ordenadores en cada uno de los segmentos de red (cubre perfectamente las necesidades presentes y futuras).

Finalmente y utilizando el valor de incremento (32) correspondiente a la columna 3, se obtiene la siguiente tabla con todas las direcciones de las 6 subredes posibles:

56.32.0.0	56.32.0.1	56.63.255.254	56.63.255.255
56.64.0.0	56.64.0.1	56.95.255.254	56.95.255.255
56.96.0.0	56.96.0.1	56.127.255.254	56.127.255.255
56.128.0.0	56.128.0.1	56.159.255.254	56.159.255.255
56.160.0.0	56.160.0.1	56.191.255.254	56.191.255.255
56.192.0.0	56.192.0.1	56.223.255.254	56.223.255.255

Usar la tabla de subredes para una clase B

Ahora se va a mostrar un ejemplo de utilización de la tabla anterior para una segmentación de una dirección de clase *B*.

Se dispone de la dirección *IP* 180.10.0.0 para una red de clase *B* y se desea segmentar la red en 24 subredes de 1.000 equipos cada una que podrían aumentar a 2.000 equipos en los próximos años.

Para ello, fíjese en la fila *Número de redes* y busque el valor correspondiente a 24 (o su superior inmediato). Ese valor corresponde a tomar 5 *bits* para la dirección de subred (que permite tener hasta 30 subredes con lo cual se podría aumentar la red en un futuro).

Ahora ya se sabe que la máscara de subred que se va a disponer es 255.255.248.0 (o su notación alternativa que es 180.10.0.0/21).

Al tener que tomar 21 *bits* para la dirección de red y subred, se obtienen 11 *bits* para la dirección de equipo con lo que se pueden tener 2^{11}-2=2.046 ordenadores en cada uno de los segmentos de red (cubre perfectamente las necesidades presentes y futuras).

Finalmente y utilizando el valor de incremento (8) correspondiente a la columna 5, se obtiene la siguiente tabla con todas las direcciones de las 30 subredes posibles:

180.10.8.0	180.10.8.1	180.10.15.254	180.10.15.255
180.10.16.0	180.10.16.1	180.10.23.254	180.10.23.255
180.10.24.0	180.10.24.1	180.10.31.254	180.10.31.255
180.10.32.0	180.10.32.1	180.10.39.254	180.10.39.255
180.10.40.0	180.10.40.1	180.10.47.254	180.10.47.255
180.10.48.0	180.10.48.1	180.10.55.254	180.10.55.255
180.10.56.0	180.10.56.1	180.10.63.254	180.10.63.255
180.10.64.0	180.10.64.1	180.10.71.254	180.10.71.255
180.10.72.0	180.10.72.1	180.10.79.254	180.10.79.255
180.10.80.0	180.10.80.1	180.10.87.254	180.10.87.255
180.10.88.0	180.10.88.1	180.10.95.254	180.10.95.255
180.10.96.0	180.10.96.1	180.10.103.254	180.10.103.255
180.10.104.0	180.10.104.1	180.10.111.254	180.10.111.255
180.10.112.0	180.10.112.1	180.10.119.254	180.10.119.255
180.10.120.0	180.10.120.1	180.10.127.254	180.10.127.255
180.10.128.0	180.10.128.1	180.10.135.254	180.10.135.255
180.10.136.0	180.10.136.1	180.10.143.254	180.10.143.255
180.10.144.0	180.10.144.1	180.10.151.254	180.10.151.255
180.10.152.0	180.10.152.1	180.10.159.254	180.10.159.255
180.10.160.0	180.10.160.1	180.10.167.254	180.10.167.255
180.10.168.0	180.10.168.1	180.10.175.254	180.10.175.255
180.10.176.0	180.10.176.1	180.10.183.254	180.10.183.255
180.10.184.0	180.10.184.1	180.10.191.254	180.10.191.255
180.10.192.0	180.10.192.1	180.10.199.254	180.10.199.255
180.10.200.0	180.10.200.1	180.10.207.254	180.10.207.255
180.10.208.0	180.10.208.1	180.10.215.254	180.10.215.255
180.10.216.0	180.10.216.1	180.10.223.254	180.10.223.255
180.10.224.0	180.10.224.1	180.10.231.254	180.10.231.255
180.10.232.0	180.10.232.1	180.10.239.254	180.10.239.255
180.10.240.0	180.10.240.1	180.10.247.254	180.10.247.255

Usar la tabla de subredes para una clase C

Ahora se va a mostrar un ejemplo de utilización de la tabla anterior para una segmentación de una dirección de clase *C*.

Se dispone de la dirección *IP* 196.32.10.0 para una red de clase *C* y se desea segmentar la red en 8 subredes de 8 equipos cada una que podrían aumentar a 14 equipos en los próximos años.

Para ello, fíjese en la fila *Número de redes* y busque el valor correspondiente a 8 (o su superior inmediato). Ese valor corresponde a tomar 4 *bits* para la dirección de subred (que permite tener hasta 14 subredes con lo cual se podría aumentar la red en un futuro).

Ahora ya se sabe que la máscara de subred que se va a disponer es 255.255.255.240 (o su notación alternativa que es 196.32.10.0/28).

Al tener que tomar 28 *bits* para la dirección de red y subred, se obtienen 4 *bits* para la dirección de equipo con lo que se pueden tener 2^4-2=14 ordenadores en cada uno de los segmentos de red (cubre perfectamente las necesidades presentes y futuras).

Finalmente y utilizando el valor de incremento (16) correspondiente a la columna 4, se obtiene la siguiente tabla con todas las direcciones de las 14 subredes posibles:

196.32.10.16	196.32.10.17	196.32.10.30	196.32.10.31
196.32.10.32	196.32.10.33	196.32.10.46	196.32.10.47
196.32.10.48	196.32.10.49	196.32.10.62	196.32.10.63
196.32.10.64	196.32.10.65	196.32.10.78	196.32.10.79
196.32.10.80	196.32.10.81	196.32.10.94	196.32.10.95
196.32.10.96	196.32.10.97	196.32.10.110	196.32.10.111
196.32.10.112	196.32.10.113	196.32.10.126	196.32.10.127
196.32.10.128	196.32.10.129	196.32.10.142	196.32.10.143
196.32.10.144	196.32.10.145	196.32.10.158	196.32.10.159
196.32.10.160	196.32.10.161	196.32.10.174	196.32.10.175
196.32.10.176	196.32.10.177	196.32.10.190	196.32.10.191
196.32.10.192	196.32.10.193	196.32.10.206	196.32.10.207
196.32.10.208	196.32.10.209	196.32.10.222	196.32.10.223
196.32.10.224	196.32.10.225	196.32.10.238	196.32.10.239

LAS MÁSCARAS DE SUBRED DE LONGITUD VARIABLE

Hasta ahora se han estado manejando máscaras de subred de longitud fija con lo cual cada uno de los segmentos en que se segmentaba la red tenían la misma máscara y el mismo número máximo de equipos que se podían tener en cada una de ellas.

Pero puede ocurrir que esto no sea la solución óptima para determinados casos (por ejemplo, si una de las subredes se desea volver a segmentar en otras subredes). Para ello, se ha definido las **máscaras de subred de longitud variable** (**VLSM**).

El uso de *VLSM* permite tener subredes con distintas máscaras y distinto número de equipos máximos en cada una de ellas. Pero su uso está condicionado al protocolo de encaminamiento que se esté utilizando y que ha de permitir que contenga la máscara de subred que se esté utilizando en cada ruta.

Ejemplo de uso de VLSM

Para el ejemplo se va a tomar una empresa que tiene asignada la dirección *IP* 180.10.0.0 de una clase *B* y desea segmentar su red entre las oficinas que tiene en Madrid (4), Barcelona (4) y Bilbao (4). Además, en cada oficina hay 8 departamentos que necesitan depósitos de direcciones diferentes y la oficina mayor no necesitará más de 1.500 equipos.

Lo primero que se observa es que ha de tener tres subredes (una para cada ciudad). Para averiguar la máscara de cada subred, fíjese en la fila *Número de redes* de la tabla y busque el valor correspondiente a 3 (o su superior inmediato). Ese valor corresponde a tomar 3 *bits* para la dirección de subred (que permite tener hasta 6 subredes con lo cual se podría aumentar la red en un futuro).

Ahora ya se sabe que la máscara de subred que se va a disponer es 255.255.224.0 (o su notación alternativa que es 180.10.0.0/19).

Al tener que tomar 19 *bits* para la dirección de red y subred, se obtienen 13 *bits* para la dirección de equipo con lo que se pueden tener 2^{13}-2=8.190 ordenadores en cada uno de los segmentos de red (cubre perfectamente las necesidades presentes y futuras).

Finalmente y utilizando el valor de incremento (32) correspondiente a la columna 3, se obtiene la siguiente tabla con todas las direcciones de las 6 subredes posibles:

180.10.32.0	180.10.32.1	180.10.63.254	180.10.63.255
180.10.64.0	180.10.64.1	180.10.95.254	180.10.95.255
180.10.96.0	180.10.96.1	180.10.127.254	180.10.127.255
180.10.128.0	180.10.128.1	180.10.159.254	180.10.159.255
180.10.160.0	180.10.160.1	180.10.191.254	180.10.191.255
180.10.192.0	180.10.192.1	180.10.223.254	180.10.223.255

Ahora ya se sabe que se puede utilizar la dirección de red 180.10.32.0 (junto con su depósito de direcciones correspondiente) para las oficinas de Madrid, la dirección de red 180.10.64.0 (junto con su depósito de direcciones correspondiente) para las oficinas de Barcelona) y la dirección de red 180.10.96.0 (junto con su depósito de direcciones correspondiente) para las oficinas de Bilbao (así mismo quedan tres subredes para un futuro).

Ahora, como en Madrid hay cuatro oficinas se ha de volver a segmentar dicha subred (con lo cual se van a tener 24 segmentos, es decir, 6 subredes por cuatro oficinas en cada una de ellas). Para averiguar la máscara de cada una de ellas, fíjese en la fila *Número de redes* de la tabla y busque el valor correspondiente a 24 (o su superior inmediato). Ese valor corresponde a tomar 5 *bits* para la dirección de subred.

Ahora ya se sabe que la máscara de subred que se va a disponer es 255.255.248.0 (o su notación alternativa que es 180.10.32.0/21).

Al tener que tomar 21 *bits* para la dirección de red y subred, se obtienen 11 *bits* para la dirección de equipo con lo que se pueden tener 2^{11}-2=2.046 ordenadores en cada uno de los segmentos de red (cubre perfectamente las necesidades presentes y futuras).

Finalmente y utilizando el valor de incremento (8) correspondiente a la columna 5, se obtiene la siguiente tabla con todas las direcciones que se podrán utilizar en cada una de las oficinas de Madrid (en el resto de las ciudades se hace de la misma manera):

180.10.32.0	180.10.32.1	180.10.39.254	180.10.39.255
180.10.40.0	180.10.40.1	180.10.47.254	180.10.47.255
180.10.48.0	180.10.48.1	180.10.55.254	180.10.55.255
180.10.56.0	180.10.56.1	180.10.63.254	180.10.63.255

Ahora, como la oficina de Madrid, cuya dirección de red 180.10.32.0 hay 8 departamentos, se ha de volver a segmentar dicha oficina (con lo cual se van a tener 192 segmentos, es decir, 6 subredes por cuatro oficinas en cada una de ellas por ocho departamentos en cada una). Para averiguar la máscara de cada una de ellas, fíjese en la fila *Número de redes* de la tabla y busque el valor correspondiente a 192 (o su superior inmediato). Ese valor corresponde a tomar 8 *bits* para la dirección de subred.

Ahora ya se sabe que la máscara de subred que se va a disponer es 255.255.255.0 (o su notación alternativa que es 180.10.32.0/24).

Al tener que tomar 24 *bits* para la dirección de red y subred, se obtienen 8 *bits* para la dirección de equipo con lo que se pueden tener 2^{8}-2=254 ordenadores en cada departamento.

Finalmente y utilizando el valor de incremento (1) correspondiente a la columna 8, se obtiene la siguiente tabla con todas las direcciones que se podrán utilizar en cada uno de los departamentos de una oficina de Madrid (en el resto se hace de la misma manera):

180.10.32.0	180.10.32.1	180.10.32.254	180.10.32.255
180.10.33.0	180.10.33.1	180.10.33.254	180.10.33.255
180.10.34.0	180.10.34.1	180.10.34.254	180.10.34.255
180.10.35.0	180.10.35.1	180.10.35.254	180.10.35.255
180.10.36.0	180.10.36.1	180.10.36.254	180.10.36.255
180.10.37.0	180.10.37.1	180.10.37.254	180.10.37.255

180.10.38.0	180.10.38.1	180.10.38.254	180.10.38.255
180.10.39.0	180.10.39.1	180.10.39.254	180.10.39.255

Condiciones para implementar VLSM

Para poder utilizar *VLSM* en una red se han de dar las siguientes condiciones:

- El **protocolo de encaminamiento** que se utilice debe soportar el anuncio de la máscara de subred a lo largo de la ruta.

- El **algoritmo de encaminamiento** utilizado debe estar basado en la *ruta mayor proporcionada*. Esto significa que si una dirección se puede representar con varias máscaras de red, deberá tomar siempre la mayor. Por ejemplo, si la dirección *IP* 180.10.35.5 puede pertenecer a las siguientes máscaras: 180.10.0.0/16, 180.10.0.0/19, 180.10.32.0/21 y 180.10.32.0/24. se deberá tomar la última que es la más específica para dicha dirección.

- Los **depósitos de direcciones** deben estar asignados para coincidir con la topología física. Es decir, si un equipo tiene la dirección *IP* 180.10.35.5 y puede pertenecer a las máscaras 180.10.0.0/16, 180.10.0.0/19. 180.10.32.0/21 y 180.10.32.0/24, se deberá tomar aquella máscara que corresponda exactamente con la estructura de segmentación de la red a la que pertenezca el equipo.

EL ENCAMINAMIENTO ENTRE DOMINIOS SIN CLASE

La separación entre las clases dentro del direccionamiento *IP* provoca una gran cantidad de direcciones desaprovechadas. Fíjese que si una compañía de tamaño mediano necesita disponer de 300 direcciones *IP*, deberá utilizar 2 clases *C* (recuerde que cada clase *C* corresponde a 254 direcciones) con el consiguiente desaprovechamiento de 208 direcciones y con el inconveniente de utilizar dos dominios separados dentro de la misma compañía (la otra posibilidad para no utilizar dos dominios en la misma compañía, sería utilizar una clase *B* con el desaprovechamiento de 65.234 direcciones).

Para evitar esto se ha desarrollado un nuevo protocolo denominado **Encaminamiento entre dominios sin clase** (**CIDR**).

Este protocolo de encaminamiento permite utilizar múltiples clases de direcciones pequeñas para funcionar como un dominio único, eliminando los conceptos de direcciones de clase *A*, *B* o *C*. Cada dirección es únicamente una dirección que contiene porciones de red y de equipo y no hay máscara de subred predefinida.

Además, soporta *suma de rutas*. De esta manera, una ruta única puede equivaler a direcciones correspondientes a miles de las rutas actuales.

En la tabla siguiente, se puede ver algunos de los rangos de bloques que pueden utilizarse con *CIDR*:

/12	255.240.0.0	1.048.594	16	4.096
/13	255.248.0.0	524.286	8	2.048
/14	255.252.0.0	262.142	4	1.024
/15	255.254.0.0	131.070	2	512
/16	255.255.0.0	65.534	-	256
/17	255.255.128.0	32.766	-	128
/18	255.255.192.0	16.382	-	64
/19	255.255.224.0	8.190	-	32
/20	255.255.240.0	4.094	-	16
/21	255.255.248.0	2.046	-	8
/22	255.255.252.0	1.022	-	4
/23	255.255.254.0	510	-	2

1er ejemplo. Si se desea disponer de una red con 1.000.000 equipos (equivalente a 16 clases *B* o 4.096 clases *C*) y con la dirección de inicio 197.15.0.1, se podrá llegar hasta la dirección final 197.30.255.254, su máscara será 255.240.0.0 y utilizará 12 *bits* de prefijo de red.

2º ejemplo. Si se desea disponer de una red con 250.000 equipos (equivalente a 4 clases *B* 0 1.024 clases *C*) y con la dirección de inicio 197.16.2.1, se podrá llegar hasta la dirección final 197.19.255.254, su máscara será 255.252.0.0 y utilizará 14 *bits* de prefijo de red.

3er ejemplo. Si se desea disponer de una red con 1.000 equipos (equivalente a 4 clases *C*) y con la dirección de inicio 197.16.15.1, se podrá llegar hasta la dirección final 197.16.18.254, su máscara será 255.255.252.0 y utilizará 22 *bits* de prefijo de red.

4º ejemplo. Si se desea disponer de una red con 4.000 equipos (equivalente a 16 clases *C*) y con la dirección de inicio 194.130.32.1, se podrá llegar hasta la dirección final 194.130.47.254, su máscara será 255.255.240.0 y utilizará 20 *bits* de prefijo de red.

Direccionamiento futuro IPv6

En el futuro, el tamaño de la dirección *IPv6* aumentará de 32 a 128 *bits* para poder soportar un número mayor de nodos direccionables, más niveles de direcciones jerárquicas y una autoconfiguración más sencilla de las direcciones.

Habrá tres formas de representar dichas direcciones:

- La primera forma, que es la más aceptada, consiste en representarla de la manera *x:x:x:x:x:x:x:x*, donde las *x* representan los valores hexadecimales de los ocho bloques de dos octetos cada uno.

  Ejemplos:

  FADB:CA58:96A4:B215:FABC:BA61:7994:1782
  A090:1:0:8:A800:290C:1:817B

  Como puede observarse, no es necesario escribir todos los ceros que hay por delante de un valor hexadecimal en un campo individual, pero se ha de tener por lo menos una cifra en cada campo.

- La segunda forma consiste en suprimir los ceros que se encuentran en medio de las direcciones. La expresión de dos "::" indicaría uno o varios grupos de 16 *bits* iguales a 0. Por ejemplo, la dirección siguiente:

  A123:FF01:0:0:0:0:0:92

  se representaría de la manera siguiente:

  A123:FF01::92

  los "::" sólo pueden aparecer una vez en la dirección.

- Otra forma, a veces más cómoda cuando haya un entorno mixto de nodos con direcciones nuevas y antiguas, es representarla de la manera *x:x:x:x:x:x:d.d.d.d*, donde las *x* son valores hexadecimales (6 grupos de 16 *bits* en la representación futura) y las *d* son valores decimales (4 grupos de 8 *bits* en la representación estándar actual).

  Ejemplos:

  0:0:0:0:0:A234:23.1.67.4
  0:0:0:0:0:1:129.154.52.1

  o con el formato comprimido

  ::A234:23.1.67.4
  ::1:129.154.52.31

IPv6 es una nueva versión de *IP* y representa una fuerte evolución con respecto a *IPv4* (aunque sus principales funciones se conservan en **IPv6**, excepto ciertas funciones poco o nada utilizadas que fueron suprimidas o convertidas en otras opciones) ya que se añadieron ocho grandes características:

- Cuenta con posibilidades extendidas de direccionamiento y encaminamiento. El tamaño de la dirección *IP* aumenta de 32 a 128 *bits* para poder soportar un número más grande de nodos direccionables, más niveles de direcciones jerárquicas y una autoconfiguración más sencilla de las direcciones.

- Queda definido un mecanismo adaptable de difusión y un nuevo tipo de direcciones en *cluster*.

- Incorpora un formato de cabecera simplificado. Algunos campos de formato de la cabecera han sido suprimidos o convertidos en opciones, y la cabecera está simplificada y reducida a un tratamiento común en todos los *routers*, lo que disminuye la dificultad de su mantenimiento.

- Cuenta con posibilidades de extensión de las cabeceras y de opciones. Las opciones están contenidas en cabeceras suplementarias colocadas entre la cabecera *IPv6* y la cabecera del paquete de transporte (*T-PDU*, *Transport Protocol Data Unit*). La mayoría de las opciones de las cabeceras de *IPv6* no son examinadas ni tratadas por los *routers* intermedios. Contrariamente a la versión actual, las opciones pueden ser de longitud variable y no existe tamaño limite.

- Define extensiones que permiten la autenticación de los usuarios y la integridad de los datos mediante herramientas de criptografía.

- Contiene varias formas de autoconfiguración como la configuración *Plug and Play* de direcciones de nodos sobre una red aislada gracias a las características ofrecidas por *DHCP*.

- Tiene una función extendida de *Source Routing* gracias a *SRDP* (*Source Demand Routing Protocol*) para difundir el encaminamiento a rutas interdominio e intradominio.

- Una transición de *IPv4* a *IPv6* sencilla y flexible.

CABECERA IPV6

La cabecera *IPv6* se ha simplificado con respecto a *IPv4* (muchos de los campos se han eliminados o hechos opcionales) para disminuir al máximo posible el coste de procesamiento de los paquetes, debido al aumento considerable en el tamaño de las direcciones. Aunque dichas direcciones son cuatro veces mayores que en *IPv4*, las cabeceras son sólo dos veces mayores.

<table>
<tr><td>VERSION</td><td>PRIORITY</td><td colspan="2">FLOW LABEL</td></tr>
<tr><td colspan="2">PAYLOAD LENGTH</td><td>NEXT HEADER</td><td>HOP LIMIT</td></tr>
<tr><td colspan="4">SOURCE ADDRESS</td></tr>
<tr><td colspan="4">DESTINATION ADDRESS</td></tr>
</table>

- **VERSION**. Identifica el número de versión *IP*.
- **PRIORITY**. Se usa para indicar la prioridad del paquete *IP* con relación a otros paquetes.
- **FLOW LABEL**. Se usa para etiquetar aquellos paquetes que requieren un manejo especial por los encaminadores.
- **PAYLOAD LENGTH**. Indica la longitud del resto de los paquetes que siguen a la cabecera (en octetos).
- **NEXT HEADER**. Identifica el tipo de cabecera que sigue inmediatamente a esta cabecera.
- **HOP LIMIT**. Se usa para limitar el impacto de los bucles de encaminamiento.
- **SOURCE ADDRESS**. Indica la dirección inicial del paquete.
- **DESTINATION ADDRESS**. Indica la dirección de destino del paquete (aunque puede no ser la final si incorpora una cabecera de encaminamiento).

CABECERAS SUPLEMENTARIAS

En la nueva versión, ciertas informaciones complementarias son codificadas en cabeceras que deben colocarse en el paquete entre la cabecera *IPv6* y la cabecera del nivel de transporte. Hay un pequeño número de extensiones a la cabecera *IPv6* (cada una de ellas identificada por un valor *Next Header* distinto). Un paquete *IPv6* puede contener ninguna, una o varias cabeceras suplementarias.

Salvo excepciones, las cabeceras suplementarias apenas son examinadas o manipuladas por los nodos alcanzados por el paquete a lo largo de su camino hasta que éste llega al nodo (o a cada grupo de nodos en el caso del *multicast*) identificados por el campo dirección de destino de la cabecera *IPv6*. En este momento se trata la primera cabecera suplementaria, o la cabecera de transporte en el caso de no haber cabeceras suplementarias. El contenido de cada cabecera determinará si es necesario tratar la cabecera siguiente.

La única excepción es la cabecera *Hop-by-Hop* (nodo por nodo), que lleva información que deberá ser examinada por los nodos de la red. Cuando está presente. tiene que seguir inmediatamente a la cabecera *IPv6*.

Cada cabecera suplementaria es de una longitud de un múltiplo de 8 *bits*, para conservar una alineación de 8 *bytes* en las cabeceras suplementarias.

Cuando hay más de una cabecera suplementaria en un mismo paquete, las cabeceras deben aparecer en el orden siguiente:

1. Cabecera **IPv6** (**IPv6 Header**).
2. Cabecera nodo por nodo (**Hop-by-Hop Header**).
3. Cabecera de encaminamiento (**Routing Header**).
4. Cabecera de fragmentación (**Fragment Header**).
5. Cabecera de autentificación (**Authentication Header**).
6. Cabecera de confidencialidad (**Privacy Header**).
7. Cabecera de extremo a extremo (**End-to-End Header**).

Cada tipo de cabecera debe aparecer una sola vez en el paquete (excepto en el caso de una encapsulación *IPv6* en *IPv6*, donde cada cabecera *IPv6* encapsulada debe estar seguida por su propia cabecera suplementaria).

Cabecera nodo por nodo

La cabecera de opciones nodo por nodo (**hop-by-hop**) contiene informaciones analizadas por los distintos nodos del camino seguido por el paquete. Se identifica por el valor del campo *Next Header* igual a 0, y tiene el formato siguiente:

NEXT HEADER	HDR EXT LEN	OPTIONS

- **NEXT HEADER** (8 **bits**): Identifica el tipo de cabecera que sigue inmediatamente a ésta. Sus valores son idénticos al campo **Protocol** de la versión *IPv4*.
- **HDR EXT LEN**: Indica la longitud de la cabecera nodo por nodo en múltiplos de 8 octetos (sin contar los ocho primeros).
- **OPTIONS**: Este campo contiene una o varias opciones codificadas en **TLV** (**Type Length Value**). Es de longitud variable en múltiplos de 8 octetos.

Cabecera de encaminamiento

La cabecera de encaminamiento es utilizada por el emisor *IP* para establecer una lista de nodo(s) intermedio(s) (o topología de *clusters*) que deberá seguir el paquete para llegar a su destino. Esta forma particular de cabecera de encaminamiento está diseñada para soportar el protocolo de encaminamiento a petición del emisor (*Source Demand Routing Protocol, SDRP*).

NEXT HEADER	ROUTING TYPE	MRE	F	RESERVED	SOURCE ROUTE LENGTH
NEXT HOP POINTER	STRICT/LOOSE BIT MASK				
SOURCE ROUTE					

- **NEXT HEADER**: Identifica el tipo de cabecera que sigue inmediatamente a ésta. Sus valores son idénticos a los del campo **Protocol** de la versión *IPv4*.

- **ROUTING TYPE**: Indica el tipo de encaminamiento soportado por esta cabecera. Su valor es 1.

- **MRE (Must Report Errors)**: Si este *bit* está a 1 y un *router* no puede emitir correctamente la lista *Source Route*, el *router* genera un mensaje de error *ICMP*. En el caso de que el *bit* esté a 0, no generará este mensaje de error.

- **F (Failure of Source Route Behavior)**: Si este *bit* está a 1, indicará que si un *router* no puede encaminar más lejos un paquete, como se especifica en el *Source Route*, el *router* fijará el valor del campo *Next Hop Pointer* con el valor del campo *Source Route Length*. De esta forma, el destino siguiente del paquete estará basado únicamente en la dirección del campo *Destination Address* (en el caso de que el *bit* estuviera a 0, el *router* destruirá el paquete).

- **RESERVED**: Inicializado a 0 por el emisor, es ignorado por el receptor.

- **SOURCE ROUTE LENGTH**: Es el número de elementos o nodos que hay en una cabecera de encaminamiento *SDRP*. La longitud de esta cabecera puede calcularse a partir de este valor (*longitud=SrcRouteLen* 16+8*). Este campo no debe exceder el valor 24.

- **NEXT HOP POINTER**: Apunta a los elementos o nodos que hay que alcanzar. Se inicializa a 0 para apuntar al primer elemento o nodo del *Source Route*. Cuando es igual al campo *Source Route Length*, significa que el *Source Route* está terminado.

- **STRICT/LOOSE BIT MASK**: Este campo se utiliza para que un nodo opte por un camino. Si el valor de *Next Hop Pointer* es **N**, significa que el *N-ésimo bit* del *Strict/Loose Bit Mask* está a 1 (indica que el siguiente nodo es un nodo *Strict Source Route Hop*), mientras que si está a 0, el siguiente nodo es un *Loose Route Hop*.

- **SOURCE ROUTE**: Es una lista de direcciones *IPv6* que indica el camino que debe seguir el paquete. Puede contener un conjunto de direcciones de tipo *unicast* y *cluster*.

Cabecera de fragmentación

La cabecera de fragmentación es utilizada por el emisor *IP* para mandar paquetes de un tamaño superior a lo que se puede enviar a los destinatarios. A diferencia de *IPv4*, la fragmentación es ejecutada únicamente por los nodos origen y no por los *routers* que intervengan en el recorrido. La cabecera de fragmentación se

distingue por un valor del campo *Next Header* igual a 44, el cual se encuentra justo después de la cabecera anterior.

NEXT HEADER	RESERVED	FRAGMENT OFFSET	RESERVED	M
IDENTIFICATION				

- **NEXT HEADER**: Identifica el tipo de la cabecera que sigue inmediatamente a ésta. Sus valores son idénticos al campo **Protocol** de *IPv4*.

- **RESERVED**: Inicializado a 0 por el emisor, es ignorado por el receptor.

- **FRAGMENTATION OFFSET**: Indica la posición del primer octeto del datagrama total (es decir, de todo el datagrama, sin fragmentar). El primer fragmento estará en el lugar número 0. El valor de este campo es un múltiplo de 8 octetos.

- **RESERVED**: Inicializado a 0 por el emisor, es ignorado por el receptor.

- **M**: Si este *bit* está a 1, significa que quedan uno o más fragmentos. En caso contrario, indica que no queda ninguno.

- **IDENTIFICATION**: Es un valor asignado al paquete de origen que es diferente de los demás paquetes fragmentados recientemente con la misma dirección fuente, la misma dirección destino y el mismo valor del campo *Next Header*. Permite identificar el datagrama para asegurar el reensamblaje de los paquetes. El número de identificación está contenido en la cabecera de todos los fragmentos.

Cabecera de autentificación

La cabecera de autentificación es utilizada para autentificar y asegurar la integridad de los paquetes. La cabecera de autentificación viene determinada por el valor 51 del campo *Next Header*.

NEXT HEADER	AUTHENTICATION DATA LENGTH	RESERVED
SECURITY ASSOCIATION ID		
AUTHENTICATION DATA		

- **NEXT HEADER**: Identifica el tipo de cabecera que sigue inmediatamente a ésta. Los valores son idénticos a los del campo **Protocol** de *IPv4*.

- **AUTHENTICATION DATA LENGTH**: Es la longitud del campo *Authentication Data* en múltiplos de 8 octetos.

- **RESERVED**: Inicializado a 0 por el emisor, es ignorado por el receptor.

- **SECURITY ASSOCIATION ID**: Se combina con la dirección fuente para identificar al (o a los) destinatario(s) el tipo de seguridad establecido.

- **AUTHENTICATION DATA**: Muestra información sobre el algoritmo que se ha de utilizar para autentificar el origen del paquete y para asegurar su integridad con respecto al tipo de seguridad asociado. La longitud de este campo es variable y múltiplo de 8 octetos.

Cabecera de confidencialidad

La cabecera de confidencialidad se utiliza para evitar el acceso no autorizado al paquete, encriptando los datos y colocándolos en la parte correspondiente de la cabecera de confidencialidad. Dependiendo de las exigencias de seguridad del usuario, se podrá encriptar la trama del nivel de transporte (*UDP* o *TCP*) o el datagrama entero. Este enfoque con encapsulación es necesario para asegurar una confidencialidad completa del datagrama original. Si está presente, la cabecera de confidencialidad es siempre el último campo no encriptado de un paquete.

Esta cabecera funciona entre estaciones, entre una estación y un *firewall* o entre dos *firewalls*.

SECURITY ASSOCIATION IDENTIFIER		
INITIALIZATION VECTOR		
NEXT HEADER	LENGTH	RESERVED
PROTECTED DATA		
TRAILER		

- **SECURITY ASSOCIATION IDENTIFIER (SAID)**: Identifica el tipo de seguridad del datagrama. Si no se ha establecido ninguna asociación de seguridad, el valor de este campo será de *0x0000*. Una asociación de seguridad es unilateral, por ello, una comunicación confidencial entre dos estaciones debe tener normalmente dos *SAID* (uno para cada uno de los sentidos). La estación de destino utiliza la combinación del valor del *SAID* y de la dirección fuente para distinguir la asociación correcta.

- **INITIALIZATION VECTOR**: Este campo es opcional, y su valor depende del *SAID* utilizado.

- **NEXT HEADER**: Va encriptado e identifica el tipo de la cabecera que sigue inmediatamente a ésta. Los valores son idénticos a los del campo **Protocol** de *IPv4*.

- **RESERVED**: Va encriptado y es ignorado por el receptor.

- **LENGTH**: Va encriptado e indica la longitud de la cabecera codificada (es un múltiplo de 8 octetos) y no incluye los 8 primeros octetos.

- **PROTECTED DATA**: Va encriptado y puede contener encapsulado un datagrama *IPv6* completo, una secuencia de opciones *IPv6* y, por último, el paquete del nivel de transporte.

- **TRAILER**: Va encriptado y se utiliza para hacer de relleno (necesario en algunos algoritmos) o para registrar datos de autentificación utilizados en un algoritmo de criptografía que proporcione confidencialidad sin autentificación. Este campo está presente únicamente si el algoritmo lo necesita.

Cabecera de extremo a extremo

La cabecera de opciones extremo a extremo (*end-to-end*) da una información opcional que debe ser controlada por el (o los) nodo(s) destinatario(s) del paquete. Es identificada por un valor del campo *Next Header* del *TBD*, que sigue inmediatamente a la cabecera previa, y tiene el mismo formato que la cabecera de opciones nodo por nodo, a excepción de la capacidad de excluir una opción de cálculo de la integridad de autentificación.

Asignación dinámica de direcciones IP

En una red normal cada equipo debe tener asignada una dirección *IP* si utiliza el protocolo *TCP/IP*, pero en una red con un servidor *DHCP*, éstas se asignarán cuando sea necesario.

DHCP (*Dynamic Host Configuration Protocol*) es un sistema desarrollado para asignar direcciones *IP* a los clientes que lo soliciten.

El proceso a seguir por un equipo que quiera conseguir una dirección *IP* es el siguiente:

1. Envía un mensaje al servidor *DHCP* solicitando una dirección *IP*.
2. El servidor *DHCP* responde ofreciendo varias direcciones *IP* que tiene disponibles de las indicadas en la instalación (entre las que están eliminadas aquellas consideradas convenientes).
3. El cliente selecciona una y envía una solicitud de uso de la dirección al servidor *DHCP*.
4. El servidor *DHCP* admite la solicitud y garantiza al cliente la concesión del uso de la dirección.
5. El cliente utiliza la dirección para conectarse a la red.

Las direcciones se conceden por un período de tiempo determinado. Cuando dicho período ha finalizado, el cliente deberá solicitar la renovación de la concesión o la dirección pasará al estado de disponible. Si solicita la renovación y no puede renovársela, se le reasignará otra.

Resolver nombres de ordenadores

Todo ordenador lleva una dirección *IP* y un nombre de equipo. Normalmente se necesitará indicar la dirección *IP* para conectarse a uno de ellos y poder realizar procesos con *TCP/IP*.

Pero también es posible realizar una conexión indicando únicamente el nombre del equipo (que es más sencillo de recordar que su dirección *IP*). Para ello, existen varios métodos:

- Servidor *DNS*
- Archivo *LMHOSTS*
- Servidor *WINS*
- Resolución *NetBIOS* sobre nodos *TCP/IP*

SERVIDOR DNS

DNS (*Domain Name System*) es un sistema que usa servidores distribuidos a lo largo de la red para resolver el nombre de un ordenador (con la estructura de nombre de ordenador, nombre de subdominio y nombre de dominio) en una dirección *IP* (de esta manera, no es necesario tener que recordar y usar su dirección *IP*).

Por tanto, se necesita un archivo que realice dicha conversión (que es lo necesario para establecer la conexión).

En una primera fase se utilizaba un archivo para realizar esta función que recibía el nombre de *HOSTS*. He aquí un ejemplo básico de archivo *HOSTS*:

```
172.16.132.1        principal
172.16.132.30       jlr
171.16.132.31       personal
165.16.132.41       secretaría
```

También era posible utilizarlo indicando el nombre completo del equipo. He aquí un ejemplo este tipo de archivo *HOSTS*:

```
172.16.132.1        principal
172.16.132.1        principal.contabilidad.es
172.16.132.30       jlr
172.16.132.30       jlr.contabilidad.es
172.16.132.31       personal
172.16.132.31       personal.contabilidad.es
```



```
165.16.132.41       secretaría
165.16.132.41       secretaría.contabilidad.es
```

Este archivo tiene que estar situado en el mismo lugar donde se encuentra *TCP/IP* en el equipo y está determinado por el sistema operativo (normalmente se encuentra en un directorio *ETC*).

Otra posibilidad es que el servidor *DNS* cuente con una base de datos para poder resolver el nombre del ordenador.

La información que se encuentra en dicha base de datos se incluye en *registros de recursos* (*RR*).

Entre dichos *registros de recurso* se encuentran:

- **Dirección** *(A)*. Asigna un nombre de un ordenador a una dirección *IP* concreta.
- **Dirección IPv6**. *(AAAA)*. Asigna un nombre de un ordenador a una dirección *IPv6* concreta.
- **Inicio de autoridad** *(SOA)*. Indica el inicio de autoridad para la zona (los servidores de nombres tienen información completa acerca de una parte del dominio llamada *zona*, entonces se dice que tiene autoridad para esa zona).
- **Intercambiador de correo** *(MX)*. Identifica el equipo a que se va a entregar correo en el dominio.
- **Nombre canónico** *(CNAME)*. Se utiliza para asignar un alias al equipo.
- **Puntero de dominio** *(PTR)*. Asigna direcciones *IP* a nombres de equipo.
- **Servidor de nombre** *(NS)*. Asocia un nombre de dominio con un nombre de equipo para un servidor de nombre concreto.
- **Ubicación de servicios** *(SRV)*. Asigna un nombre de dominio *DNS* a una lista especificada de equipos que ofrecen un tipo específico de servicio.

Los servidores *DNS* pueden realizar los siguientes papeles:

- **Servidor de nombre primario**. Este tipo de servidor *DNS* mantiene la base de datos de nombres y direcciones para una zona, guardando información sobre la forma de contactar con servidores de nombre de zonas inferiores y superiores.
- **Servidor de nombre secundario**. Este tipo de servidor *DNS* obtiene información de la zona de un servidor maestro (puede ser de un servidor primario o de otro secundario que tiene una copia del archivo de zona).

Esta información se guarda en un archivo de sólo lectura para aumentar la fiabilidad y descargar trabajo al servidor *DNS* primario. Para mantener actualizada dicha base de datos se realizan *transferencias de zonas*. Estas transferencias de zonas se hacen al arrancar el servidor secundario y cada ez que se detecta una modificación en la base de datos principal.

- **Servidor de nombre maestro**. Este tipo de servidor *DNS* transfiere el archivo de zona a un servidor secundario (puede actuar como servidor maestro tanto un servidor primario como un secundario).

- **Servidor de nombre sólo de caché**. Este tipo de servidor *DNS* no almacena ningún archivo de información de zona. Cuando un equipo solicita a un servidor *DNS*, primario o secundario, la resolución de un nombre de equipo, el servidor de sólo caché guarda la dirección *IP* que devuelve el servidor *DNS* antes de enviarla al equipo que realizó la consulta. En caso de necesitarse resolver otra vez el nombre del equipo anterior, en vez de consultarse al servidor *DNS* primario o secundario, se consultará al servidor de sólo caché.

ARCHIVO LMHOSTS

Otra posibilidad de realizar una conexión indicando únicamente el nombre del equipo es utilizando un archivo *LMHOSTS*.

Éste es un archivo local que asigna direcciones a los nombres de los equipos de una red *Microsoft TCP/IP* para conectarse con otra red exterior.

El funcionamiento de este archivo es muy parecido al del archivo *HOSTS* y también tiene que estar situado en el mismo lugar donde se encuentra *TCP/IP* en el equipo y está determinado por el sistema operativo (en *Windows NT* es *C:\Winnt\system32\Drivers\etc*).

He aquí un ejemplo de archivo *LMHOSTS*:

```
172.16.132.1        principal
172.16.132.30       jlr
171.16.132.31       personal
165.16.132.41       secretaría
```

Pero además puede utilizar palabras claves como:

#PRE. Especifica que cargue el nombre en un caché (normalmente se consulta este archivo cuando ha fallado *WINS* y el mensaje de *B-nodo*. Al realizar la carga en memoria caché se asegura que estará disponible cuando se necesite).

#DOM. Especifica el dominio al que estará asociado el nombre. Debe ir precedido de la palabra clave *#PRE*.

#INCLUDE. Permite cargar entradas desde un archivo remoto. Debe incluir el nombre del equipo, el directorio y el archivo. Debe ir entre las palabras claves **#BEGIN_ALTERNATE** y **#END_ALTERNATE**.

He aquí un ejemplo de archivo *LMHOSTS* con palabras claves:

```
172.16.132.1      principal      #PRE  #DOM:CONTABILIDAD
172.16.132.30     jlr            #PRE

BEGIN_ALTERNATE
#INCLUDE\\PRINCIPAL_1\WINNT\System32\drivers\etc\LMHOSTS
#INCLUDE\\PRINCIPAL_2\WINNT\System32\drivers\etc\LMHOSTS
END_ALTERNATE
```

Este archivo no es necesario si utiliza una red que funciona con *WINS*. Pero si va a conectarse a otra red que no utilice *WINS*, deberá configurar a los clientes para que miren este archivo.

SERVIDOR WINS

WINS (*Windows Internet Name Service*) es un sistema desarrollado por *Microsoft* para convertir los nombres de los equipos *Windows NT* en direcciones *IP* en un entorno encaminado.

De esa manera, cuando un ordenador que tiene una dirección *IP* y un nombre cambia su localización, no será necesario cambiar de dirección *IP*, ya que *WINS* mantiene una base de datos de los nombres con sus respectivas direcciones *IP* (desafortunadamente, *WINS* es un protocolo sólo de *Microsoft* y los clientes exteriores a su red no pueden utilizarlo cuando se conecten a ella).

Un servidor *WINS* se encarga de registrar, consultar y enviar los nombres, solucionando el problema de que utilicen nombres *NetBIOS* en un entorno *TCP/IP* de una forma más flexible que utilizando *DNS*.

En una red que incorpore un servidor *WINS* se pueden dar tres tipos de ordenadores:

- **Servidor WINS**. Puede haber varios servidores *WINS* en una red. Cada uno de ellos puede duplicar las bases de datos de los otros servidores para mejorar el rendimiento de la red.

- **Clientes WINS**. Sólo ordenadores con *Windows NT*, *Windows 3.11 para trabajo en grupo*, *Windows 95* y *Windows 98* pueden interactuar directamente con servidores *WINS*.

- **Clientes NO WINS**. Los clientes que están configurados como *B-nodo*, pueden interactuar con clientes *WINS* en los segmentos locales de su red. Los clientes *WINS* obtienen información del servidor *WINS* en los

segmentos locales y remoto y proporcionan dicha información a los clientes *NO WINS*. De esa manera, los clientes *WINS* funcionan como apoderados (*PROXIES*) de los clientes *NO WINS* (para que esto sea así, por lo menos un ordenador del segmento de red debe ser cliente *WINS*).

Todo el proceso de instalación y configuración de *WINS* tiene que hacerse asignando previamente una dirección *IP* fija a cada servidor *WINS* (no es necesario que sea un cliente *DHCP*).

Aunque es posible que un único servidor *WINS* proporcione servicio a una red entera, lo ideal es que se mantengan, al menos, dos servidores *WINS* para evitar un fallo en el sistema *WINS*.

Si dispone de más de un servidor *WINS*, éstos se comunicarán entre sí para duplicar sus bases de datos, asegurando que cualquier nombre registrado en uno de ellos esté duplicado en todos y cada uno de los servidores *WINS* dentro del conjunto de redes.

Cada servidor *WINS* puede ser *duplicador de inserción* o *duplicador de extracción* con, al menos, otro servidor *WINS*.

Un **duplicador de inserción** es un servidor *WINS* que envía mensajes de notificación de actualización, a su servidor asociado, cuando su base de datos ha sido modificada. Cuando su asociado responda a la notificación, le enviará una copia de su base de datos actual. Ese otro servidor asociado es un **duplicador de extracción**.

También se puede activar la duplicación de un servidor *WINS*, en el caso de que el servidor alcance el umbral definido por el administrador, si se activa manualmente o si se activa a una hora concreta.

Para que las bases de datos de ambos servidores *WINS* sean coherentes, tienen que ser duplicadores de inserción y extracción a la vez, los unos con los otros.

Es posible que haya varios duplicadores de inserción y varios de extracción en un conjunto de redes.

RESOLUCIÓN NETBIOS SOBRE NODOS

Existen cuatro tipos de nodos que pueden usarse para resolver los nombres utilizando un servidor *NetBIOS*. Éstos son:

- **B-nodo** (*broadcast*). El cliente utiliza mensajes en forma de datagramas de *UDP* para resolver y registrar su nombre. Esto funciona bien en redes pequeñas pero incrementa intensamente su tráfico, por lo que no es aconsejable para redes grandes.

- **P-nodo** (*punto a punto*). El cliente no crea ni responde mensajes. Los equipos registrados utilizan comunicaciones punto a punto para registrarse. Si el servidor no está disponible, no podrá hacerlo.

- **M-nodo** (*mixto*). El cliente primero intenta conectarse en modo *B-nodo*, si lo consigue intenta cambiarse a modo *P-nodo* (la conexión inicial en *B-nodo* genera un tráfico elevado de mensajes en la red sin conseguir el objetivo de conectarse a un servidor).

- **H-nodo** (*híbrido*). Es el modo más reciente de todos. El cliente intenta primero conectarse en modo *P-nodo*, y si no lo consigue, intenta registrarse en modo *B-nodo*.

Protocolos TCP/IP

TCP/IP es una familia de protocolos desarrollados para permitir la comunicación entre ordenadores de cualquier tipo de red o fabricante, respetando los protocolos de cada red individual.

Los protocolos *TCP/IP* se estructuran en 5 niveles funcionales:

APLICACIÓN
TRANSPORTE INTERNET
RED
FÍSICO

- El **nivel físico** corresponde al *hardware*. Puede ser un cable coaxial, un cable par trenzado, cable de fibra óptica o una línea telefónica. *TCP/IP* no considera oficialmente el nivel físico como componente específico de su modelo y tiende a agrupar el nivel físico con el nivel de red. Los protocolos principales de este nivel son: *ARP* y *RARP*.

- El **nivel de red**. Independientemente del medio físico que se utilice, necesitará una tarjeta de red específica que, a su vez, dependerá de un *software* llamado controlador de dispositivo proporcionado por el sistema operativo o por el fabricante. Proporciona fiabilidad (aunque no necesariamente) en la distribución de datos que pueden adoptar diferentes formatos. Aunque *TCP/IP* no especifica ningún protocolo para este nivel, los protocolos más notables son: *SLIP*, *PPP* y *PPTP*.

- El nivel **Internet** se superpone a la red física creando un servicio de red virtual independiente de aquélla. No es fiable ni orientado a conexión. Se encarga del direccionamiento y encaminamiento de los datos hasta la estación receptora. El protocolo específico de este nivel es *IP*.

- El **nivel de transporte** suministra a las aplicaciones servicios de comunicaciones desde la estación emisora a la receptora. Utiliza dos tipos de protocolos: *TCP* que es fiable y orientado a conexión y *UDP* que es no fiable y no orientado a conexión.

- El **nivel de aplicación** corresponde a las aplicaciones disponibles para los usuarios como pueden ser: *FTP*, *SNMP*, *TELNET*, etc.

Estos niveles se corresponden con los niveles del modelo de referencia *OSI* de la siguiente manera:

TCP/IP	OSI
	APLICACIÓN
	PRESENTACIÓN
APLICACIÓN	SESIÓN
TRANSPORTE	TRANSPORTE
INTERNET	RED
RED	ENLACE DE DATOS
FÍSICO	FÍSICO

Esta correspondencia es teórica porque, como los protocolos *TCP/IP* fueron desarrollados antes que el modelo de referencia *OSI*, existen sustanciales diferencias, como son:

- **El concepto de jerarquía en relación con los niveles**. Indica que una tarea de comunicaciones se divide en entidades que pueden comunicar con otras entidades pares en otro sistema. Una entidad dentro de un sistema proporciona servicios a otras entidades y, a su vez, utiliza los servicios de otras. Estas entidades deben tener una relación jerárquica, de manera que una entidad sólo utilice los servicios de las entidades jerárquicamente inferiores. La diferencia entre ambos modelos es consecuencia del pragmatismo con el que se desarrollaron los protocolos *TCP/IP*, ya que éstos proporcionan a los diseñadores mayor grado de libertad para la utilización de uno u otro; mientras que *OSI* es más prescriptivo, ya que dicta los protocolos de un nivel determinado que deben realizar unas funciones específicas.
- **La interoperación de redes**, ya que los protocolos *TCP/IP* se han concebido para interconectar sistemas no conectados a la misma red.
- **La fiabilidad extremo a extremo**. El protocolo *IP* no es fiable, es decir, no garantiza que los paquetes entregados sean correctos y que conserven la secuencia con que fueron emitidos, ya que supone que son los protocolos de transporte los que deben garantizarlo.
- **Los servicios no orientados a conexión**. El protocolo *IP* tampoco es orientado a conexión, ya que ésta debe proporcionarse en niveles superiores.
- **La gestión de red**. En los primeros documentos del modelo *OSI* no se contemplaban las funciones de gestión y, aunque actualmente ya se contemplan, no alcanzan el nivel de aceptación de los de *TCP/IP*.

PROTOCOLOS DEL NIVEL FÍSICO

Aunque *TCP/IP* no considera oficialmente el nivel físico como componente específico de su modelo y tiende a agrupar el nivel físico con el nivel de red, se van a describir en este apartado los protocolos *ARP* y *RARP*.

ARP

ARP (*Address Resolution Protocol*) es un protocolo que se utiliza para convertir las direcciones *IP* en direcciones físicas que puedan ser utilizadas por los manejadores.

Para poder realizar esta conversión, existe en cada ordenador un módulo *ARP* que utiliza una **tabla de direcciones ARP**, que en la mayoría de los ordenadores trata como si fuera una memoria intermedia (*caché*), de forma que la información que lleva mucho tiempo sin utilizarse se borra.

Si encuentra la correspondencia entre la dirección *IP* y la dirección física se procede a la transmisión.

Si no la encuentra en la tabla, se genera una petición *ARP* que se difunde por toda la red. Si alguno de los ordenadores de la red reconoce su propia dirección *IP* en la petición *ARP*, envía un mensaje de respuesta indicando su dirección física y se graba en la **tabla de Direcciones ARP**.

RARP

RARP (*Reverse Address Resolution Protocol*) se utiliza cuando, al producirse el arranque inicial, los ordenadores no conocen su dirección *IP*.

Requiere que exista en la red, al menos, un servidor *RARP*. Cuando un ordenador desea conocer su dirección *IP*, envía un paquete que contiene su propia dirección física.

El servidor *RARP*, al recibir el paquete, busca en su tabla *RARP* la dirección *IP* correspondiente a la dirección física inicial indicada en el paquete y envía un paquete al ordenador origen con esta información.

A diferencia del protocolo *ARP* que se incorpora normalmente en todos los productos *TCP/IP*, el protocolo *RARP* sólo se incorpora en unos pocos productos.

PROTOCOLOS DEL NIVEL DE RED

Aunque *TCP/IP* no especifica ningún protocolo para este nivel, se van a describir los protocolos *SLIP* y *PPP*.

SLIP

SLIP (*Serial-Line Internet Protocol*) es, históricamente, el primero desarrollado para satisfacer la necesidad de establecer una conexión *TCP/IP* empleando únicamente una línea serie.

La utilización actual más común de este protocolo es la conexión (a *Internet*, por ejemplo) a través de una línea telefónica aunque también puede utilizarse para conectar dos ordenadores próximos mediante un cable serie.

Se describe en el *RFC 1055* y es un mecanismo muy sencillo de transmisión de paquetes. De hecho, su único cometido es el envío de datagramas en formato *IP* a través de una línea serie.

Sus inconvenientes más significativos son los siguientes:

- Carece de métodos de corrección de errores, delegando estas funciones en las capas superiores del *software*.
- Es incapaz de realizar tareas de gestión del enlace.
- Carece de métodos de autentificación.
- Es incapaz de negociar parámetros fundamentales de la comunicación (direcciones de red, tamaño de los paquetes o el empleo de algoritmos de compresión de datos). Todas estas características deben establecerse antes de efectuar la conexión, y deben coincidir en ambos extremos del enlace.

Existe una versión de *SLIP* denominada *CSLIP* que, según las recomendaciones del *RFC 1144* es capaz de comprimir las cabeceras *TCP*, con el aumento de eficiencia que esto supone por la menor cantidad de datos que es preciso transmitir.

PPP

PPP (*Point-to-Point Protocol*) es un protocolo *SLIP* mejorado con control y recuperación de errores.

Funcionalmente, es mucho más completo y robusto que *SLIP*. Además de asumir todas sus funciones, incorpora múltiples mejoras:

- Es posible negociar el tamaño máximo de los paquetes entre los dos extremos o la utilización de técnicas de compresión.
- Existe posibilidad de autentificación.
- Puede monitorizarse la calidad del enlace.

Una característica que hace a *PPP* aún más interesante es la posibilidad de conexión a través de *RDSI*.

Existe una gran cantidad de software comercial y de dominio público disponible para PPP.

PPTP

Aunque **PPTP** (*Point to Point Tunneling Protocol*) no es un protocolo propio de *TCP/IP*, se va a describir en esta sección, ya que es un nuevo protocolo de red, incorporado en *Windows*, que utiliza redes privadas multiprotocolo para permitir a los usuarios remotos tener acceso de forma segura, a través de *Internet*, a redes de empresas (*Extranet*).

Ofrece las siguientes ventajas:

- **Costes de transmisión más bajos**. Ya que usa *Internet* para la conexión en lugar de una llamada normal a través de la línea telefónica.
- **Menores costos de hardware**. Ya que permite separar los módems y las tarjetas *RDSI*, así como colocarse en un servidor de comunicaciones.
- **Mayor nivel de seguridad**. Funciona con cifrado de datos y actúa con cualquier protocolo.

Los datos enviados con *PPTP* se encapsulan en paquetes *PPP* cifrados que se envían a través de *Internet*. Pero también, puede ser usado para transportar el tráfico de acceso remoto *IPX* y *NetBEUI*.

PROTOCOLOS DEL NIVEL INTERNET

En este nivel se encuentran los protocolos *ICMP* e *IP*.

ICMP

ICMP (*Internet Control Message Protocol*) es un protocolo de mantenimiento/gestión de red que ayuda a supervisar la red.

Se utiliza para poder encontrar una ruta a través de la cual los datagramas viajen por la red y alcancen su destino.

El objetivo principal de *ICMP* es proporcionar la información de error o control entre nodos. La implementación de *ICMP* es obligatoria como un subconjunto lógico del protocolo *IP*.

Los mensajes de error de este protocolo normalmente los genera y los procesa *TCP/IP* y no el usuario.

Existen cuatro tipos de mensajes *ICMP*:

Un *socket* está compuesto por un par de números que identifican de manera única a cada aplicación. Cada *socket* se compone de dos campos:

1. La **dirección IP** del ordenador en el que se está ejecutando la aplicación.
2. El **puerto** a través del cual la aplicación se comunica con *TCP/IP*.

UDP

UDP (*User Datagram Protocol*) es un protocolo que se basa en el intercambio de datagramas. *UDP* permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

El inconveniente de esta forma de actuación es que no hay confirmación de recepción ni de haber recibido los datagramas en el orden adecuado, debiendo ser la aplicación la que se encargue de controlarlo.

Al igual que el protocolo *TCP*, utiliza puertos y *sockets* y, también, permite la *multiplexación*.

PROTOCOLOS DEL NIVEL DE APLICACIÓN

Todas las aplicaciones *TCP/IP* utilizan el modelo cliente/servidor.

En este nivel se encuentran un buen número de protocolos de los cuales se van a describir los siguientes: *FTP*, *HTTP*, *NFS*, *NTP*, *RPC*, *SMTP*, *SNMP*, *TELNET* y *TFTP*.

FTP

FTP (*File Transfer Protocol*) es el más utilizado de todos los protocolos de aplicación y uno de los más antiguos.

Se utiliza para la transferencia de archivos proporcionando acceso interactivo, especificaciones de formato y control de autentificación (aunque es posible conectarse como el usuario *anonymous* que no necesita contraseña).

HTTP

HTTP (*HyperText Transfer Protocol*) es uno de los protocolos más recientes. Se utiliza para manejar la consulta de hipertexto y el acceso de datos con *World Wide Web* (*WWW*). El tráfico generado por este protocolo ha pasado, debido a la influencia de *Internet*, a ser muy grande.

NFS

NFS (*Network File System*) ha sido desarrollado por *Sun Microsystems Incorporated* y autoriza a los usuarios el acceso en línea a archivos que se encuentran

en sistemas remotos (accede a un archivo remoto como si se tratara de un archivo local). La mayoría del tráfico *NFS* es ahora un caso especial del protocolo *RPC*.

NTP

NTP (*Network Time Protocol*) permite que todos los sistemas sincronicen su hora con un sistema designado como servidor horario.

RPC

RPC (*Remote Procedure Call*) es una llamada a un procedimiento que se ejecuta en un sistema diferente del que realiza la llamada.

El proceso cliente envía un mensaje al proceso servidor y espera una respuesta. Éste, al recibir la llamada, estudia los procedimientos del proceso llamado, obtiene los resultados y los envía de vuelta al proceso cliente mediante un mensaje de respuesta.

Existen dos tipos de servidores:

1. El **servidor iterativo** que recibe una llamada proporciona el servicio y vuelve al estado de espera.
2. El **servidor concurrente** que recibe la llamada contesta al mensaje enviando al cliente un número de puerta, arranca un proceso paralelo para prestar el servicio requerido por el cliente y vuelve al estado de espera. Cuando el proceso paralelo haya finalizado el servicio requerido, acaba su ejecución.

SMTP

SMTP (*Simple Mail Transfer Protocol*) es un protocolo de correo electrónico. Especifica el formato exacto de los mensajes que un cliente debe enviar desde un ordenador al servidor de otro, pero no especifica cómo debe almacenarse el correo ni con qué frecuencia se debe intentar el envío de los mensajes.

SNMP

SNMP (*Simple Network Management Protocol*) sirve para administrar los sistemas de forma remota. También se puede utilizar para supervisar el tráfico de la red desde una o varias estaciones de trabajo llamadas administradores *SNMP*.

Los elementos de la red que puede administrar y monitorizar son dispositivos como ordenadores, puertas de enlace (*gateways*), encaminadores (*routers*), *mainframes*, miniordenadores, *hubs*, etc.

SNMP minimiza el número y la complejidad de las funciones realizadas por el administrador y cuenta con las siguientes ventajas:

- Reduce el coste de desarrollo del *software* del agente de administración necesario para soportar este protocolo.

- Aumenta el grado de las funciones de administración utilizadas de forma remota, permitiendo un uso completo de los recursos de *Internet* en dichas tareas.

- Permite que las funciones de administración sean de fácil comprensión y uso por parte de los desarrolladores de herramientas de administración de la red.

Utiliza una arquitectura distribuida que consiste en agentes y sistemas de administración.

- Un **agente** es un ordenador que ejecuta el *software* de agente *SNMP* o un encaminador.

  La obligación principal de un agente es ejecutar las tareas iniciadas por los comandos *SNMP* que han sido requeridas por un sistema de administración.

  Los comandos *SNMP* que se utilizan pertenecen a los tipos siguientes:

  - **GetRequest**: Éste es el comando que usa el sistema de administración para solicitar información a un agente.

  - **GetNextRequest**: También es empleado por el sistema de administración para solicitar información al agente y se utiliza si la información deseada se encuentra en forma de tabla o matriz (se usa de forma repetitiva hasta que se hayan conseguido todos los datos de la matriz).

  - **GetResponse**: El agente consultado utiliza este comando para contestar una solicitud hecha por el sistema de administración.

  - **SetRequest**: El sistema de administración lo utiliza para cambiar el valor de un parámetro del *MIB (Management Information Base).*

  - **Trap**: Este comando lo utiliza un agente para informar al sistema de administración de un suceso determinado que se ha producido.

- Un **sistema de administración** es un ordenador que ejecuta un *software* de administración *SNMP*. Puede iniciar las operaciones de los comandos *GetRequest*, *GetNextRequest* y *SetRequest*.

Un agente únicamente puede iniciar el comando *Trap* para informar al sistema de administración de un suceso extraordinario y contestar al sistema de administración con el comando *GetResponse*.

La forma en que actúa el protocolo *SNMP* es la siguiente:

1. El sistema de administración envía primero una solicitud al agente para obtener el valor de una variable de *MIB*.

2. El agente contesta a la solicitud en función del nombre de la comunidad que acompaña a la solicitud.

   Una comunidad comprende un grupo de ordenadores que ejecutan el servicio *SNMP*. El uso de un nombre de comunidad proporciona una seguridad mínima para los agentes que reciben solicitudes e inician capturas (*traps*) así como para las tareas iniciadas por los sistemas de administración.

   Un agente no responderá a una solicitud de un sistema de administración distinto a aquellos que tenga configurados (un agente puede pertenecer a varias comunidades a la vez).

MIB describe los objetos que están incluidos en la base de datos de un agente *SNMP*.

Los objetos que haya en *MIB* deben estar definidos para que los desarrolladores de *software* para la administración de las estaciones de trabajo los conozcan, así como sus valores respectivos.

MIB registra y almacena información sobre el ordenador en el que se está ejecutando. Un administrador *SNMP* puede solicitar y recoger información de un agente *MIB* así como revisar o alterar los objetos que contiene.

TELNET

TELNET permite que un usuario, desde un terminal, acceda a los recursos y aplicaciones de otros ordenadores.

Una vez que la conexión queda establecida, actúa de intermediario entre ambos ordenadores.

Se fundamenta en tres principios:

- El **concepto de terminal virtual de red** (*NVT*). Corresponde a la definición de cómo han de ser los datos, caracteres de control y las secuencias de los mandatos que han de circular por la red para permitir una heterogeneidad de los sistemas.

- La **simetría entre terminales y procesos**. La comunicación puede ocurrir entre dos terminales, dos procesos o entre un terminal y un proceso.

- **Permite que el cliente y el servidor negocien sus opciones**. La conexión comienza con una fase de negociación de opciones en la que se utilizan cuatro mandatos: *WILL*, *WONT*, *DO*, y *DONT*.

  WILL se envía para mostrar el deseo de comenzar una opción (que se ha de indicar) y se contesta con *DO* (respuesta positiva) o *DONT* (respuesta negativa).

  WONT se envía para mostrar el deseo de no comenzar una opción (que se ha de indicar) y se contesta con *DONT* (mostrando el acuerdo de no utilización).

  DO se envía para indicar que comience a utilizar una opción (que se ha de indicar) y se contesta con *WILL* (respuesta positiva) o *WONT* (respuesta negativa).

  DONT se envía para indicar que no comience a utilizar una opción (que se ha de indicar) y se contesta con *WONT* (mostrando el acuerdo de no utilización).

TFTP

TFTP es un protocolo destinado a la transferencia de archivos aunque sin permitir tanta interacción entre cliente y servidor como la que existe en *FTP*.

Además, existe otra diferencia. En lugar de utilizar el protocolo *TCP*, utiliza *UDP*.

Sus reglas son muy sencillas. En el envío del primer paquete se establece una interacción entre el cliente y el servidor. Se empieza una numeración de los bloques (empezando desde 1). Cada paquete de datos contiene una cabecera que especifica el bloque que contiene. Un bloque de menos de 512 *bytes* indica que es el último y corresponde al final del archivo.

Enviando paquetes en la subred local

Una de las responsabilidades de *IP* es determinar si un paquete debe ser enviado a la subred local o bien debe ser encaminado a otra subred.

Se siguen los pasos siguientes:

- *IP* recibe una trama de *TCP* que está dirigida a una dirección *IP* determinada.

- *IP* compara el identificativo de la subred de la dirección recibida con el identificativo de la subred local. Si ambos coinciden, la trama se envía localmente.

- Antes de proceder al envío local, *IP* debe determinar la dirección de la red física que corresponde a la dirección *IP* destino. Para ello, utiliza *ARP*.

- *IP* añade la siguiente información a la trama:

  - La dirección *IP* origen.
  - La dirección física de la red origen.
  - La dirección *IP* destino.
  - La dirección física de la red destino.

- *IP* pasa el paquete con las direcciones añadidas al protocolo de nivel inferior que lo lleva a su destino.

Enviando paquetes a la subred remota

Si en el punto 2 del apartado anterior se determina que las dos subredes (origen y destino) son distintas, es una indicación clara de que el paquete se tiene que encaminar hacia una subred remota.

Cuando un ordenador está conectado con el exterior, debe estar configurado con una dirección *IP* de un *gateway* (*puerta de enlace*, *router o encaminador*) por defecto.

Como el paquete tiene que dirigirse hacia el exterior, *IP* dirige el paquete al *gateway* por defecto.

Allí se realizan distintos algoritmos de encaminamiento (que pueden ser simples o complejos) hasta que se identifica el *gateway* de la subred a la que tiene que enviar el paquete. Entonces, lo envía a dicho *gateway* que, por fin, lo hace llegar al ordenador destino.

Para identificar el *gateway* de la subred remota, es necesario consultar las tablas de encaminamiento que están disponibles en el *gateway* por defecto.

Esta tabla puede ser:

- **Estática**. Este tipo de tabla lo tiene que crear manualmente el administrador de la red y no se actualiza automáticamente cuando se producen cambios en la red. Debe contener, por lo menos, los datos siguientes:

  - **Direcciones de red**. Indica las direcciones *IP* remotas a las que va tener acceso.

  - **Máscara de subred**. Indica la máscara correspondiente a la subred de la dirección *IP* remota.

- **Dirección del gateway**. Indica la dirección del *gateway* remoto que se usará para enviar un paquete a una dirección *IP* remota.

- **Dirección física de la red**. Indica la dirección física de la red remota.

- **Dinámicas**. Este tipo de tablas se actualizan automáticamente cuando se produce algún tipo de cambio en la red.

  Para ello, se utilizan varios algoritmos de encaminamiento. Entre ellos, se encuentran:

  - **OSPF** (*Open Shortest Path First*). Este algoritmo está basado en el estado de enlaces. Cada *gateway* envía, de forma periódica, paquetes de estado de enlaces que describen sus conexiones a sus vecinos. Usando estas comunicaciones, los *gateways* vecinos elaboran una base de datos de estado de enlaces que utilizan para identificar encaminamientos.

    Este algoritmo detecta bucles en la transmisión de ruta que evitan el problema de que dos rutas se llamen entre ellas y puede estar emitiendo los mensajes de estado de enlaces indefinidamente.

    También obligan a la autentificación de los intercambios que evitan que una persona ajena pueda recibir la información de ruta.

  - **RIP** (*Routing Information Protocol*). Este algoritmo está basado en el vector/distancia.

    Si un *gateway* conoce varias rutas para llegar a un destino, asigna un coste a la ruta en función de los saltos que deba realizar (cuanto más *routers* tenga que cruzar más saltos tendrá que realizar).

    Cada 30 segundos envía un mensaje con su tabla de encaminamiento a los demás, que actualizan sus propias tablas con los datos recibidos (esto origina un aumento considerable del tráfico de la red).

    Este algoritmo no detecta bucles en la transmisión de ruta, por lo que se daría el problema de que dos rutas que se llamen entre ellas, estarían emitiendo sus tablas de encaminamiento indefinidamente.

    Tampoco obligan a la autentificación de los intercambios, por lo que una persona ajena podría recibir la información de rutas enviadas por los *gateways*.

    Está disponible en dos versiones: *RIP I* y *RIP II* (soporta el uso de máscaras de subred).

Multicast

La mayoría de los protocolos de alto nivel de una red (como los protocolos de transporte *ISO*, *TCP* o *UDP*) sólo proporcionan un servicio de transmisión *unicast*. Es decir, los nodos de una red sólo son capaces de enviar paquetes de uno a otro en un momento dado.

Si un nodo desea enviar la misma información a muchos destinos por medio de un servicio de transporte *unicast*, deberá llevar a cabo un *unicast replicado* para después enviar distintas copias de los datos a cada uno de los destinos aunque con la limitación del ancho de banda de que se dispone.

Una manera mejor de transmitir datos desde un origen a varios destinos es proporcionar un servicio de transporte *multicast*. De esta forma, un nodo puede enviar datos a varios destinos haciendo simplemente una llamada al servicio de transporte que enviará dichos datos al grupo de ordenadores seleccionado y que estará definido por sus correspondientes direcciones *IP*.

Multicast (multidifusión) es aconsejable porque permite la construcción de aplicaciones verdaderamente distribuidas y una importante optimización del rendimiento sobre las transmisiones *unicast*.

Para definir las direcciones de los ordenadores que van a recibir los datos, se forma un grupo *multicast* utilizando las direcciones *IPv4* de la clase *D* que van desde 224.0.0.0 hasta 239.255.255.255, estando reservadas las direcciones que van desde 224.0.0.0 hasta 224.0.0.255 a protocolos de encaminamiento y a otros protocolos necesarios para mantener la topología de la red.

En *IPv6* el formato de la dirección *IP* es diferente debido a su cabecera nueva y al incremento de la longitud de la dirección. Se utiliza un campo de 8 *bits* con todos los valores puestos a uno, un campo de 4 *bits*, otro campo de 4 *bits* y el resto de 112 *bits* se utiliza para la identificación del grupo. El primer campo de 4 *bits* se utiliza para indicar si se trata de una dirección permanente o no (si el cuarto *bit* de dicho campo es cero, la dirección es permanente y si es uno, la dirección es temporal y se reutilizará al finalizar la sesión). El segundo campo de 4 *bits* se utiliza para determinar el alcance del ámbito de la dirección (es decir, si es un ámbito global, un nodo local, un enlace local, etc.).

Existen dos posibilidades de funcionamiento del servicio *multicast*:

- Basado en el emisor.
- Basado en el receptor.

En un servicio *multicast* basado en el emisor, los ordenadores destino envían los requerimientos de datos al ordenador emisor y éste envía una copia de dichos datos. De esta forma, el ordenador origen conoce los ordenadores destino que desean obtener una copia de los datos.

En un servicio *multicast* basado en el receptor, los ordenadores destino envían los requerimientos de datos al *router* y éste es el que envía una solicitud al ordenador origen para que le envíe una copia de los datos. De esta forma, el ordenador origen no conoce los ordenadores destino que desean obtener una copia de los datos, pero a cambio, se reduce el tráfico dentro de la red.

La forma de trabajar del servicio *multicast* es la siguiente:

- Desde el ordenador origen se envía una única copia de los datos.
- Cuando llega al primer *router* se hacen tantas copias de los datos como caminos distintos se producen para llegar a cada uno de los ordenadores destino.
- Cada vez que llega a un nuevo *router* repite el proceso anterior hasta que, al final del proceso, hay tantas copias de los datos como ordenadores destino.
- Al final, cada ordenador destino recibe una copia de los datos enviados.

Como se puede observar, los responsables de la duplicación de los datos son los encaminadores correspondientes que se encuentran en el camino.

PROTOCOLOS MULTICAST

Dentro de los protocolos *multicast* se encuentran los siguientes:

ST II

ST-II (*Stream Protocol Version II*) es un protocolo basado en el emisor, correspondiente al nivel de red, que ofrece soporte a envíos continuos de datos.

A diferencia de *IP*, está orientado a conexión y define un enlace virtual entre el emisor de los datos y su receptor (una parte de este procedimiento consiste en reservar recursos de la red como *buffers* de cada *router* por el que se van a dirigir los datos y ancho de banda de la red).

Este enlace virtual va a definir una ruta fija por la que se van a transmitir todos los datos entre ambos ordenadores manteniendo, así mismo, otro enlace para establecer un canal de retorno para el control de mensajes enviados desde el receptor al emisor.

XTP

XTP (*Xpress Transport Protocol*) es un protocolo basado en el emisor, correspondiente a los niveles de red y de transporte propios, que no excluye el uso de otros protocolos *TCP/IP* en los mismos dispositivos de red, ya que está diseñado para coexistir con ellos.

Está preparado para soportar una gran variedad de aplicaciones como programas multimedias en redes o sistemas de tiempo real.

Este protocolo no incluye ninguna administración de direcciones dependiendo de *IP* para su manejo.

Tampoco requiere ninguna preparación en la comunicación entre el emisor y el receptor, pero sí necesita que se establezca un acuse de recibo de los datos entre ambos cuando se unen a un grupo *multicast*.

Incorpora un sistema de prioridades para definir la importancia de los datos así *XTP* transmitirá primero los datos que se consideran más importantes.

MTP

MTP (*Multicast Transport Protocol*) es un protocolo basado en el emisor, correspondiente al nivel de transporte, que ofrece a los programas de aplicaciones garantías de atomicidad y fiabilidad; pero, por su propia estructura de funcionamiento, implica una sobrecarga del tráfico de la red.

Para su funcionamiento requiere tres tipos distintos de ordenadores:

- **Maestro**. Es el ordenador responsable de ordenar los mensajes de todos los miembros del grupo *multicast* y de controlar las transmisiones de los datos. Cuando un miembro del grupo no mantiene un mínimo flujo de datos, le requiere para que abandone el grupo.
- **Productores**. Son aquellos ordenadores que van a enviar las copias de datos (cada *productor* cuenta con datos distintos a los de los demás) al resto de ordenadores miembros del grupo *multicast*. Previamente al envío de las copias, deben enviar una solicitud al *ordenador maestro* para que les autorice a realizar el envío.
- **Consumidores**. Es el resto de ordenadores del grupo *multicast* y son los destinatarios de las copias de datos enviadas. En el caso de no recibir bien los paquetes de datos, pueden solicitar a los ordenadores *productores* que les sean reenviados los datos incompletos o erróneos.

El protocolo *MTP* es apropiado para aplicaciones como bases de datos distribuidas, que necesitan que todos los miembros de un grupo *multicast* reciban los mismos paquetes de datos.

IP MULTICAST

IP Multicast es un protocolo basado en el receptor que sirve para transmitir datagramas *IP* desde un origen a varios destinos en redes de área local o extensa.

El servicio básico proporcionado por *IP Multicast* es un servicio *multicast* de datagramas no fiable (es decir, no hay ninguna garantía de que un paquete dado haya alcanzado todos los nodos destinatarios a los que iba destinado), pero permite reducir el tráfico de la red, y, así mismo, el ordenador emisor no necesita formar parte del grupo *multicast*.

Este servicio es apropiado para ciertas aplicaciones que se centran más en el rendimiento que en la fiabilidad.

Seguridad de TCP/IP

La seguridad es una preocupación importante siempre que se conecta una red al exterior.

El *software* básico de *TCP/IP* no cifra los datos por sí mismo, debe realizarlo la aplicación. Si no se cifran los datos, la contraseña se enviará en texto *ASCII* y podrá ser leída fácilmente durante el trayecto por personas que disponga de medios y conocimientos. Es importante que los datos (sobre todo la contraseña) vayan cifrados para evitar que sean examinados por personas ajenas.

Otra opción es mantener alejadas del sistema a todas aquellas personas ajenas. Para ello, lo mejor es instalar un cortafuegos (*firewall*).

Su función es filtrar los intentos de establecimiento de conexión de forma que se pueda detectar e impedir el acceso al sistema a posibles intrusos sin que ni siquiera se haya llegado a establecer un enlace directo entre ellos.

El cortafuegos puede ser configurado para permitir que sólo determinadas direcciones, origen y destino, puedan acceder a su red (o desde ella).

Las funciones de cortafuegos se pueden realizar por:

- Ordenadores dedicados exclusivamente al filtrado de paquetes (servidor *proxy*).
- Encaminadores de red (*routers)* configurados para esta tarea.
- Programas de *software* para distintos sistemas operativos.
- Cualquier otro dispositivo intercalado entre la red y el exterior que soporte el filtrado de paquetes según unos parámetros previamente definidos.

Entre los posibles beneficios de utilizar cortafuegos se encuentran:

- Acceso controlado a la red.
- Protección para servicios de *Internet* que sean vulnerables.

- Administración de seguridad centralizada.
- Estadísticas de las conexiones a la red.
- Filtrado sofisticado de paquetes. Los filtros de paquetes controlan qué tipos de paquetes *IP* pueden acceder a los servicios de la red interna. Así, puede denegar paquetes, bloquear paquetes de un ordenador determinado de *Internet*, rechazar direcciones fantasmas, evitar ataques *FRAG* (un ataque *FRAG* se produce cuando se provoca un fallo en el algoritmo de reensamblado de los paquetes *IP* que se reciben debido al envío de fragmentos de paquetes trucados) o evitar ataques *SYN* (un ataque *SYN* se produce cuando se inunda un servidor con requerimientos de conexiones falsas que evitan el procesamiento de requerimientos verdaderos).
- Configuración desde un sistema de *hardware* independiente que no dependa de ningún otro sistema de *hardware* y *software*.

Entre las posibles razones para no utilizar un cortafuegos se encuentran:

- El acceso a los servicios deseados puede llegar a ser más complejo de lo normal.
- El peligro de acceso por una puerta trasera a la red, se incrementa si no se tiene previsto su inutilización.
- Es necesario una administración suplementaria de la red.
- El coste económico es mayor.
- La configuración se hace demasiado compleja para realizarla de forma adecuada.

Comandos TCP/IP

Hay distintos comandos *TCP/IP* que permiten realizar distintas acciones y que no son iguales en todas las versiones.

A continuación, se hace una relación alfabética de todos los comandos *TCP/IP* disponibles en *Windows 2000* de una forma detallada (los valores a sustituir en un comando van encerrados entre paréntesis angulares "< >", pero dichos símbolos no se han de teclear):

ARP

Muestra o modifica las tablas de traducción de las direcciones *IP* a direcciones *Ethernet* o a direcciones *Token Ring* para que sean utilizadas por el protocolo de resolución de direcciones *ARP*.

La forma de escribir este comando es:

ARP <-opción> <dirección_IP> <dirección_ETHERNET> <dirección_interfaz>
<dirección_IP> indica la dirección *IP* en notación decimal con puntos.

<dirección_ETHERNET> corresponde a la dirección física y se ha de indicar como 6 *bytes* hexadecimales separados por guiones.

<dirección_interfaz> indica la dirección *IP* de la interfaz cuya tabla de conversión de direcciones se desea modificar (si no se indica ninguna, se usará la primera disponible).

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
a	Muestra las entradas actuales de *ARP*. Si se especifica *<dirección_IP>*, únicamente mostrará las direcciones *IP* y físicas del equipo al que corresponde dicha dirección *IP*.
d	Elimina de la tabla la entrada indicada en *<dirección_IP>*.
N	Se usa únicamente con *<dirección_interfaz>*. Muestra las entradas de la tabla para la interfaz de red especificada.
s	Añade una entrada en la tabla para asociar la *<dirección_IP>* con la *<dirección_ERTHERNET>*.

Ejemplos:

Para añadir a la tabla de traducción de direcciones la dirección *IP* 172.16.132.1 asociada con la dirección física 00C0DFA00CBE, introduzca:

ARP -s 172.16.132.1 00-C0-DF-A0-0C-BE

Para ver el contenido de la tabla de traducción de direcciones, introduzca:

ARP -a

FINGER

Muestra información sobre un usuario conectado a un equipo que está ejecutando el servicio *Finger*.

La forma de escribir este comando es:

FINGER <-opción> <usuario>@<equipo>

<Usuario> corresponde al usuario del que se desea obtener información (si no se indica ninguno, mostrará información de todos los usuarios del equipo especificado).

<Equipo> corresponde al equipo al que pertenecen los usuarios de los que se quiere obtener información.

Opciones:

Este comando tiene la opción siguiente:

OPCIÓN	SIGNIFICADO
I	Muestra la información en formato ancho.

Ejemplos:

Para que muestre información sobre todos los usuarios conectados en el equipo *PRINCIPAL*, introduzca:

FINGER @PRINCIPAL

Para que muestre información en formato ancho, sobre el usuario *ENRIQUE* conectado en el equipo *PRINCIPAL*, introduzca:

FINGER -I ENRIQUE@PRINCIPAL

FTP

Transfiere archivos entre una estación de trabajo y un servidor *FTP*, y viceversa (pueden ejecutar sistemas operativos distintos).

Cuando se ejecuta en forma de sesión interactiva, es necesario utilizar las ordenes *FTP*. Entre las más utilizadas se encuentran las siguientes:

ORDEN	SIGNIFICADO
! <comando>	Ejecuta el comando indicado en el equipo local (si no se especifica ningún comando, mostrará el símbolo del sistema. Hay que escribir *exit* para volver a *FTP*).

ORDEN	SIGNIFICADO
? <orden>	Presenta una descripción de la orden indicada (si no se escribe ninguna, mostrará la lista de órdenes que reconoce).
APPEND <arch1> <arch2>	Añade el archivo de la estación de trabajo indicado primero al archivo del servidor *FTP* especificado en segunda posición.
ASCII	Establece el tipo de transferencia de archivos a formato *ASCII* (es el valor por defecto).
BELL	Activa o desactiva un tono de aviso al finalizar la ejecución de un comando de transferencia de archivos.
BINARY	Establece el tipo de transmisión a binario.
BYE	Finaliza la sesión *FTP* con el servidor y sale de *FTP*.
CD <directorio>	Cambia el directorio actual del servidor *FTP*.
CLOSE	Finaliza la sesión *FTP* con el servidor pero no sale de *FTP*.
DEBUG	Activa o desactiva el modo de depuración (por defecto, está desactivado.
DELETE <archivo>	Borra el archivo indicado del servidor *FTP*.
DIR <camino> <archivo>	Muestra una lista de todos los archivos del servidor *FTP* que se encuentran en el camino indicado. Si se especifica un nombre de archivo, la salida se guardará en dicho archivo dentro del propio ordenador (en caso contrario, se mostrará en pantalla).
DISCONNECT	Se desconecta del servidor *FTP* pero no sale de *FTP*.
GET <arch1> <arch2>	Copia el archivo (indicado en primer lugar) del servidor *FTP* a la estación de trabajo con el nombre indicado en segunda posición (si no se especifica ningún nombre, se copiará con el mismo nombre que tenía en el servidor *FTP*).
GLOB	Activa o desactiva la posibilidad de utilizar comodines en los nombres de archivos o de rutas de acceso locales (por defecto, está activado el uso de comodines).

ORDEN	SIGNIFICADO
HASH	Activa o desactiva la impresión del signo # para cada bloque de datos transferido (por defecto, no se imprime el signo).
HELP <comando>	Lista todos los comandos *FTP* o muestra información sobre el comando indicado.
LCD <directorio>	Cambia el directorio actual de la estación de trabajo (por defecto, el directorio de trabajo es en el que se ha iniciado *FTP*).
LITERAL <argumento>	Envía el argumento especificado al servidor *FTP*.
LS <camino> <archivo>	Muestra una lista resumida de todos los archivos y subdirectorios del servidor *FTP* que se encuentran en el camino indicado. Si se especifica un nombre de archivo, la salida se guardará en dicho archivo dentro del propio ordenador (en caso contrario, se mostrará en pantalla).
MDELETE <archivos>	Borra los archivos indicados del servidor *FTP* del directorio actual. Admite caracteres comodín.
MDIR <camino> <archivo>	Muestra una lista de los archivos y subdirectorios del servidor *FTP* que se encuentran en el camino indicado. Si se especifica un nombre de archivo, la salida se guardará en dicho archivo dentro del propio ordenador (en caso contrario, se mostrará en pantalla). Admite caracteres comodín.
MGET <archivos>	Copia los archivos indicados del servidor *FTP* al directorio actual de la estación de trabajo. Admite caracteres comodín.
MKDIR <directorio>	Crea el directorio indicado en el servidor *FTP*.
MLS <camino> <archivo>	Muestra una lista resumida de los archivos y subdirectorios del servidor *FTP* que se encuentran en el camino indicado. Si se especifica un nombre de archivo, la salida se guardará en dicho archivo dentro del propio ordenador (en caso contrario, se mostrará en pantalla). Admite caracteres comodín.

ORDEN	SIGNIFICADO
MPUT <archivos>	Copia los archivos indicados de la estación de trabajo al directorio actual del servidor *FTP*. Admite caracteres comodín.
OPEN <servidor> <puerto>	Abre una sesión en el servidor *FTP* indicado. Se puede indicar mediante una dirección *IP* o su nombre de equipo (en este caso, se deberá disponer de un fichero *HOSTS* o *DNS* en el que conste el equipo). En *<puerto>* se puede indicar el número de puerto que se desea utilizar para conectarse.
PROMPT	Activa o desactiva la aparición de mensajes de confirmación durante la transferencia de archivos (por defecto, está activada).
PUT <archivo1> <archivo2>	Copia el archivo (indicado en primer lugar) de la estación de trabajo, al servidor *FTP* con el nombre indicado en segunda posición (si no se especifica ningún nombre, se copiará con el mismo nombre que tenía en la estación).
PWD	Muestra el nombre del directorio actual del servidor *FTP*.
QUIT	Termina la sesión *FTP* con el servidor y sale de *FTP*.
QUOTE <argumento>	Envía el argumento especificado al servidor *FTP*. Actúa igual que *LITERAL*.
RECV <arch1> <arch2>	Copia el archivo (indicado en primer lugar) del servidor *FTP* a la estación de trabajo con el nombre indicado en segunda posición (si no se especifica ningún nombre, se copiará con el mismo nombre que tenía en el servidor *FTP*). Actúa igual que *GET*.
REMOTEHELP <comando>	Lista todos los comandos *FTP* remotos o muestra información sobre el comando indicado.
RENAME <arch1> <arch2>	Cambia el nombre (indicado en primer lugar) del archivo del servidor *FTP* al nombre indicado en segunda posición.

ORDEN	SIGNIFICADO
RMDIR <archivo>	Elimina el directorio indicado del servidor *FTP*.
SEND <archivo1> <archivo2>	Copia el archivo (indicado en primer lugar) de la estación de trabajo, al servidor *FTP* con el nombre indicado en segunda posición (si no se especifica ningún nombre, se copiará con el mismo nombre que tenía en la estación). Actúa igual que *PUT*.
STATUS	Muestra el estado actual de las conexiones y comandos *FTP*.
TRACE	Activa o desactiva el seguimiento de los paquetes.
TYPE <tipo>	Establece o muestra el tipo de transferencia de los archivos. En *<tipo>* se tendrá que indicar: *ASCII* o *BINARY* (si no especifica ningún tipo, se mostrará el valor actual).
USER <nombre>	Inicia una sesión en el servidor *FTP* con el nombre de usuario indicado (pedirá la contraseña si es necesaria).
VERBOSE	Activa o desactiva el modo detallado (es decir, mostrando todas las respuestas de *FTP*). Por defecto, está activado.

La forma de escribir este comando es:

FTP <-opción> <servidor_FTP>

<servidor_FTP> indica el nombre del equipo o la dirección *IP* del equipo remoto con el que se va a establecer la conexión.

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
a	Hace que se utilice cualquier interfaz local al realizar una conexión *FTP*.
d	Hace que se active la depuración y muestre las órdenes *FTP* introducidas en la pantalla.

OPCIÓN	SIGNIFICADO
g	Desactiva el uso de comodines. Actúa igual que la orden *GLOB*.
i	Desactiva las peticiones interactivas durante la transferencia de archivos múltiples.
n	Suprime el inicio de sesión automático cuando se establece la conexión.
s <archivo>	Especifica un archivo de texto que contiene una secuencia de órdenes *FTP* que se ejecutarán automáticamente cuando se inicie *FTP*.
v	Suprime la visualización de las respuestas del servidor *FTP*.
w <tamaño>	Indica el tamaño del búfer de transferencia.

Ejemplo:

Para que se realice una transferencia de archivos entre una estación de trabajo y el servidor *RED*, introduzca:

FTP RED

Entrará dentro de una sesión *FTP* y deberá utilizar las órdenes *FTP* específicas. Cuando finalice, introduzca *QUIT* y pulse [**Intro**] para volver al sistema operativo.

HOSTNAME

Indica el nombre del equipo actual.

La forma de escribir este comando es:

HOSTNAME

Opciones:

Este comando no tiene ninguna opción.

Ejemplo:

Para ver el nombre del equipo actual, introduzca:

HOSTNAME

IPCONFIG

Muestra todos los valores actuales de la configuración *TCP/IP*. Es especialmente útil en los sistemas que ejecutan *DHCP* ya que permite averiguar las direcciones *IP* que se han adjudicado.

La forma de escribir este comando es:

IPCONFIG </opción>

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
all	Muestra la presentación completa de datos (sin esta opción, únicamente mostrará la dirección *IP*, la máscara de subred y la dirección *IP* de la puerta de enlace predeterminada para cada tarjeta de red).
release <adaptador>	Libera la configuración actual de *DHCP*, desactivando *TCP/IP* (en *<adaptador>* hay que poner el nombre que aparece cuando se utiliza *IPCONFIG* sin ninguna opción).
renew <adaptador>	Renueva los parámetros de configuración de *DHCP* (en *<adaptador>* hay que poner el nombre que aparece cuando se utiliza *IPCONFIG* sin ninguna opción).

Ejemplo:

Para ver distintos valores *TCP/IP*, introduzca:

IPCONFIG

LPQ

Muestra el estado de la cola de impresión indicada en un equipo que ejecuta el servidor *LPD*.

La forma de escribir este comando es:

LPQ <S servidor> <-P impresora> <-opción>

Opciones:

Este comando tiene la siguiente opción:

OPCIÓN	SIGNIFICADO
i	Muestra información detallada.

Ejemplo:

Para ver el estado de la cola de impresión de la impresora *IMPRE403*, introduzca:

LPQ -P IMPRE403

LPR

Envía un archivo a una impresora de un equipo que ejecute un servidor *LPD*. La forma de escribir este comando es:

LPR <-S servidor> <-P impresora> <-C clase> <-J nombre trabajo> <-O opción> <archivo>

<clase> especifica el contenido de la página de encabezado.

<nombre trabajo> especifica el nombre del trabajo.

<opción> especifica el tipo de archivo que se va a imprimir (por defecto, es un archivo de texto).

Opciones:

Este comando no tiene ninguna opción.

Ejemplo:

Para imprimir el archivo *CONFIG.SYS* que está en el directorio raíz, a la impresora *IMPRE403*, introduzca:

LPR -P IMPRE403 C:\CONFIG.SYS

NBTSTAT

Muestra las estadísticas de protocolo y las conexiones *TCP/IP* actuales que utilizan *NBT* (*NetBIOS sobre TCP/IP*).

La forma de escribir este comando es:

NBTSTAT <-a nombre> <-A dirección_IP> <-opción> <intervalo>

<nombre> especifica la tabla de nombres del equipo remoto utilizando su nombre.

<dirección_IP> especifica la tabla de nombres del equipo remoto utilizando su dirección *IP*.

<intervalo> indica el tiempo (en segundos) de pausa que esperará antes de volver a mostrar las estadísticas (deberá pulsar **[Ctrl]+[C]** para interrumpir la presentación de las estadísticas).

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
c	Presenta el contenido del caché de nombres *NetBIOS* indicando la dirección *IP* de cada nombre.
n	Presenta una lista de los nombres *NetBIOS* locales.
r	Presenta las estadísticas de resolución de nombres de red de *Windows*. Si utiliza *WINS*, devolverá el número de nombres resueltos y registrados mediante **WINS** o mediante difusión amplia.
R	Vuelve a cargar el archivo *LMHOSTS* después de limpiar la memoria caché de nombres *NetBIOS*.
s	Presenta las sesiones de cliente y servidor intentando convertir la dirección *IP* del equipo remoto en un nombre utilizando el archivo *HOSTS*.
S	Presenta las sesiones de cliente y servidor indicando los equipos remotos mediante su dirección *IP*.

Ejemplo:

Para ver las estadísticas de protocolo y conexiones *TCP/IP* actuales que usan *NBT* del equipo remoto con la dirección *IP* 172.54.23.4 cada 30 segundos, introduzca:

NBTSTAT -A 172.54.23.4 30

NETSTAT

Muestra las estadísticas de protocolo y las conexiones actuales de la red *TCP/IP*.

La forma de escribir este comando es:

NETSTAT <-opción> <intervalo>

<intervalo> indica el tiempo (en segundos) de pausa que esperará antes de volver a mostrar las estadísticas (deberá pulsar **[Ctrl]+[C]** para interrumpir la presentación de las estadísticas).

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
a	Presenta todas las conexiones y puertos de escucha.
e	Presenta estadísticas relativas a *Ethernet*.
n	Presenta las direcciones y los números de puerto en formato numérico.
p <protocolo>	Muestra las conexiones del protocolo indicado que puede ser *TCP* o *UDP* (si se utiliza junto a la opción **s**, el protocolo podrá ser *ICMP*, *IP*, *TCP* o *UDP*).
r	Presenta el contenido de la tabla de enrutamiento.
s	Presenta estadísticas de cada protocolo (*ICMP*, *IP*, *TCP* y *UDP*). Si se utiliza junto a la opción **p**, presentará un subconjunto de dichas estadísticas.

Ejemplo:

Para ver las estadísticas de protocolo y conexiones actuales *TCP/IP* de todos los protocolos cada 30 segundos, introduzca:

NETSTAT -s 30

NSLOOKUP

Muestra información de los servidores de nombres *DNS*.

La forma de escribir este comando es:

NSLOOKUP <-opción> <equipo> <servidor>

<equipo> indica la dirección *IP* o el nombre del equipo a buscar. Si se escribe un guión en lugar de un valor, se pasa al modo interactivo (se distingue porque el indicador del sistema es el signo >).

<servidor> indica el servidor *DNS* que se desea utilizar en lugar del predeterminado.

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
cl=<valor>	Cambia el grupo de protocolo de la información. El valor puede ser: *ANY* (cualquiera de los siguientes), *CHAOS* (corresponde a la clase *Chaos*), *HESIOD* (corresponde a la clase *MIT Athena Hesiod*) o *IN* (corresponde a la clase *Internet*). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
d2	Activa el modo de depuración exhaustiva y se podrán ver todos los campos de cada paquete (poniendo *nod2* se desactiva). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
deb	Activa el modo de depuración y así podrá ver mayor información sobre el paquete enviado al servidor y la información resultante (poniendo *nodeb* se desactiva). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
def	Anexa el nombre de dominio *DNS* predeterminado a una solicitud de búsqueda de componente que no puede contener puntos (poniendo *nodef* se desanexa). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.

OPCIÓN	SIGNIFICADO
ro=<nombre>	Cambia el nombre del servidor raíz. Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
root	Cambia el servidor predeterminado al servidor de la raíz del espacio de nombres del dominio *DNS* (puede cambiar el nombre del servidor raíz con *ro=<nombre>*). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
sea	Si la solicitud de búsqueda contiene por lo menos un punto pero no finaliza con un punto, activa que se anexen los nombres de dominio *DNS* que hay en la lista de búsqueda de dominio *DNS* a la solicitud hasta que se recibe una respuesta (poniendo *nosea*, se desactiva). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
server <dominio>	Cambia el servidor predeterminado al dominio *DNS* indicado (utiliza el servidor predeterminado actual para buscar información sobre el dominio *DNS*).
set all	Se utiliza en el modo interactivo y muestra los valores actuales de la configuración.
srchl <n1>/<n2>/../<n6>	Cambia el nombre de dominio *DNS* y la lista de búsqueda predeterminada. Puede incluir hasta seis nombres. Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
ti=<número>	Establece el número de segundos inicial de espera por una solicitud. Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
ty=<valor>	Cambia el tipo de consulta de la información (actúa igual que *q=<valor>*). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.

OPCIÓN	SIGNIFICADO
v	Activa que siempre utilice un circuito virtual al enviar solicitudes al servidor (poniendo *nov*, se desactiva). Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.
view <archivo>	Ordena y muestra una lista de la salida de las opciones que están en el archivo. Si se pone delante la palabra *set*, se cambia el valor en el modo interactivo.

Ejemplos:

Para ver información del servidor de nombres *DNS* cuya dirección *IP* sea 172.16.132.8 con tiempo inicial de espera de 3 segundos y 5 reintentos por vez, introduzca:

NSLOOKUP -ti=3 -ret=5 172.16.132.8

Para establecer una conexión interactiva, introduzca:

NSLOOKUP -

Una vez establecida una conexión interactiva, para ver los parámetros de configuración *NSLOOKUP*, introduzca:

set all

Para salir de la conexión interactiva, introduzca:

exit

PING

Envía una llamada a un equipo remoto e informa si se puede establecer conexión o no con él. También muestra determinadas estadísticas sobre el estado de la conexión establecida.

La forma de escribir este comando es:

PING <-opción> <dirección_IP>

Si en vez de una dirección *IP* indica un nombre de equipo, deberá encontrarse en el archivo *HOSTS* local o en el servidor *DNS*.

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
a	Resuelve las direcciones *IP* en nombres de equipos.
f	Envía un indicador para que las puertas de enlace no fragmenten el paquete.
i <número>	Espera respuesta del equipo el tiempo dado.
j <equipos>	Encamina los paquetes mediante la lista de equipos indicada (el número máximo de equipos que se pueden indicar es 9). Los equipos consecutivos pueden separarse por puertas de enlace intermedias.
k <equipos>	Encamina los paquetes mediante la lista de equipos indicada (el número máximo de equipos que se pueden indicar es 9). Los equipos consecutivos no pueden separarse por puertas de enlace intermedias.
l <número>	Pone el número de *bytes* de datos al indicado.
n <número>	Hace los reintentos de comunicación el número indicado de veces y se para.
r <número>	Registra el camino del paquete de salida y el paquete de vuelta en el apartado "Enrutamiento de registro" (*<número>* ha de estar entre 1 y 9).
s <número>	Indica la marca de hora para el número de saltos indicado.
T	Envía continuamente la señal al equipo, esperando cada vez la respuesta. Para salir de la orden, pulse **[Ctrl]+[C]**.
v <servicio>	Manda la señal con el tipo de servicio indicado.
w <tiempo>	Especifica el intervalo de tiempo de espera en milisegundos.

Ejemplo:

Para ver si está activo el equipo *PC109.RED*, introduzca:

PING PC109.RED

RCP

Realiza una copia de archivos entre equipos *Windows 2000* y un sistema que ejecuta un *shell* remoto.

Es imprescindible que en el equipo remoto estén especificados los nombres del equipo y del usuario en el archivo *.RHOSTS*, dentro del subdirectorio privado de cada usuario, para poder establecer un acceso remoto.

También es necesario que el usuario que ha iniciado la sesión en *Windows 2000* tenga cuenta en el equipo remoto para poder establecer la conexión ya que no pide una contraseña para realizar la copia de archivos.

La forma de escribir este comando es:

RCP <-opción> <equipo_origen.usuario:archivo> <equipo_destino.usuario:archivo>

Si se omite *<equipo_origen>* o *<equipo_destino>* se entiende que es el equipo local.

Si se omite *<usuario>* se entiende que es el nombre de usuario que ha iniciado la sesión en *Windows 2000*.

En *<archivo>* habrá que indicar la ruta completa junto al nombre del archivo (en *UNIX* los directorios se indican con la barra */* y en *Windows 2000* con la barra \).

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
a	Indica que el archivo a transferir es *ASCII*.
b	Indica que el archivo a transferir es binario.
h	Transfiere los archivos marcados con el atributo de oculto en *Windows 2000*.
r	Indica que se va a realizar una copia recursiva, es decir que va a incluir los subdirectorios que dependen del subdirectorio origen.

Ejemplo:

Para copiar el archivo local *TCP.BAT*, en formato *ASCII*, al subdirectorio *TMP* del servidor *RED* que trabaja con el sistema operativo *UNIX*, para el usuario *RODRIGUEZJL*, introduzca:

RCP -a TCP.BAT RED.RODRIGUEZJL:/TMP

REXEC

Permite ejecutar órdenes del sistema operativo en otro equipo que esté ejecutando el servicio *REXEC*.

Es imprescindible que en el equipo remoto estén especificados los nombres del equipo y del usuario en el archivo *.RHOSTS*, dentro del subdirectorio privado de cada usuario, para poder establecer un acceso remoto.

La forma de escribir este comando es:

REXEC <equipo> <-l usuario> <-opción> <comando>

Si se omite el nombre de usuario, se usa el nombre de usuario con el que se ha iniciado la sesión.

El equipo remoto le pedirá la contraseña de acceso para el usuario indicado.

Opciones:

Este comando tiene la siguiente opción:

OPCIÓN	SIGNIFICADO
n	Redirige la entrada a *null*.

Ejemplo:

Para ver los archivos cuyo nombre empieza por *TCP*, del usuario *RODRIGUEZJL*, que se encuentran en el subdirectorio *TMP* del servidor *RED*, que trabaja con el sistema operativo *UNIX*, introduzca:

REXEC RED -l RODRIGUEZJL la /TMP/TCP*

ROUTE

Controla las tablas de enrutamiento de la red.

La forma de escribir este comando es:

ROUTE <-opción> <comando> <destino> <máscara> <puerta> <métrica>

<*destino*> indica el equipo al que se enviará el comando.

<*máscara*> especifica una máscara de red que se va a asociar con el camino (si no se indica se tomará 255.255.255.255).

<*puerta*> especifica una puerta de enlace.

<*métrica*> asigna una medida de costo para calcular las rutas más rápidas.

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
f	Borra las tablas de enrutamiento de todas las entradas de puerta de enlace (si se utiliza junto a algún comando, se borrarán antes de la ejecución de éste).
p	Se puede utilizar con el comando *ADD* (establecerá una ruta permanente para todos los inicios del sistema) o *PRINT* (mostrará la lista de rutas permanentes registradas).

Los posibles comandos son:

COMANDO	SIGNIFICADO
ADD	Agrega un camino.
CHANGE	Modifica el camino existente.
DELETE	Elimina un camino.
PRINT	Imprime un camino

Ejemplo:

Para borrar todas las entradas de las tablas de enrutamiento, introduzca:

ROUTE –f

RSH

Permite ejecutar comandos del sistema operativo en otros equipos que ejecuten el servicio *RSH*.

Es imprescindible que en el equipo remoto estén especificados los nombres del equipo y del usuario en el archivo *.RHOSTS*, dentro del subdirectorio privado de cada usuario, para poder establecer un acceso remoto.

También es necesario que el usuario que ha iniciado la sesión en *Windows 2000* tenga cuenta en el equipo remoto para poder establecer la conexión ya que no pide contraseña para ejecutar el comando indicado.

La forma de escribir este comando es:

RSH <equipo> <-l usuario> <-opción> <comando>

Si se omite el nombre de usuario, se usa el nombre de usuario con el que se ha iniciado la sesión.

Opciones:

Este comando tiene la siguiente opción:

OPCIÓN	SIGNIFICADO
n	Redirige la entrada a *null*.

Ejemplo:

Para ver los archivos cuyos nombres empiezan por *TCP*, del usuario *RODRIGUEZJL*, que se encuentran en el subdirectorio *TMP* del servidor *RED*, que trabaja con el sistema operativo *UNIX*, introduzca:

RSH RED -l RODRIGUEZJL ls /TMP/TCP*

TFTP

Transfiere archivos entre un equipo local y un equipo remoto que está ejecutando el servicio *TFTP* sin necesidad de autentificación del usuario.

La forma de escribir este comando es:

TFTP <-opción> <equipo> <operación> <archivo_local> <archivo_remoto>

<equipo> especifica el nombre del equipo remoto.

<operación> se refiere a las órdenes de traer (*get*) y llevar (*put*) archivos del comando *FTP*.

Opciones:

Este comando tiene la siguiente opción:

OPCIÓN	SIGNIFICADO
i	Pone el modo de transferencia a binario (si se omite esta opción, el modo de transferencia es *ASCII*).

Ejemplos:

Para traer el archivo *MANUAL.TXT* del servidor *RED*, al equipo local y dejarle con el nombre *MANUAL2.TXT*, introduzca:

TFTP GET RED MANUAL2.TXT MANUAL.TXT

Para llevar del equipo local el archivo *MANUAL.TXT* al equipo remoto *PRINCIPAL* y dejarle con el nombre *MANUAL2.LST*, introduzca:

TFTP PUT PRINCIPAL MANUAL.TXT MANUAL2.LST

TRACERT

Determina el camino tomado hacia un destino enviando paquetes del protocolo *ICMP* con valores variables de *Tiempo de duración* (*TTL*) para el destino. Cada enrutador disminuirá *TTL* al menos en una unidad antes de reenviarlo. Cuando *TTL* llegue a cero, el enrutador devolverá al sistema de origen un mensaje de tiempo excedido *ICMP*.

Para determinar la ruta, se enviará el paquete con *TTL* de valor uno e irá aumentando dicho valor en una unidad hasta que responda el destino o se llegue al máximo número de saltos indicado.

La forma de escribir este comando es:

TFTP <-opción> <destino>

<destino> especifica el nombre del equipo destino.

Opciones:

Este comando tiene las siguientes opciones:

OPCIÓN	SIGNIFICADO
D	Indica que las direcciones no se deben resolver en nombres de equipos.
h <saltos>	Especifica el número máximo de saltos que se han de dar para buscar el destino.
j <equipos>	Indica los posibles caminos a través de los equipos que se indican.
w <tiempo>	Espera cada respuesta el número de milisegundos indicado.

Ejemplo:

Para determinar el camino hasta el equipo *PRINCIPAL* con un número máximo de 9 saltos y con un tiempo de espera de 10 milisegundos, introduzca:

TRACERT -h 9 -w 10 PRINCIPAL