FUNDAMENTOS DEL

TCP / IP

Autor : ……. José Manuel Tella Llop

Recopilación : . Agustí Guiu i Ribera

Versión : …..… 1.0

Fecha : …… 30 de Junio del 2001

ACERCA DEL TCP/IP

El TCP/IP es una colección de protocolos estándar de la industria diseñada para intercomunicar grandes redes (WANs = *Wide Area Networks)*.

Las siglas TCP/IP provienen de *Transmission Control Protocol / Internet Protocol.*

Vamos a intentar dar en esta parte, unos conceptos sobre TPC/IP, su terminología y explicar como la *Internet Society* crea el estándar de Internet.

HISTORIA DEL TCP/IP

El TCP/IP fue originado con los experimentos de intercambio de paquetes dirigido por el *U.S. Department of Defense Advanced Research Projects Agency* (DARPA) durante la década de 1960 a 1970.

Hay varios hitos importantes en la historia del TCP/IP:

1970: Los ordenadores de la *Advanced Research Agency Network* (ARPANET) comienzan a utilizar el NCP (*Network Control Protocol)*.

1972: La primera especificación Telnet. *"Ad hoc Telnet Protocol"* se define como una RFC, la 318.

1973: RFC 454. Se introduce el FTP (*File Transport Protocol)*

1974: El TCP (*Transmission Control Program)* se especifica detalladamente.

1981: El estándar IP se publica en la RFC 791

1982: La *Defense Communications Agency'* (DCA) y ARPA establecen a la *'Transmision Control Protcolol (TCP)* y al *Internet Protocol (IP)* como la colección de protocolos TCP/IP.

1983: ARPANET cambia de NCP a TCP/IP

1984: Se define el concepto de DNS (*Domanin Name System*)

EL PROCESO DE ESTANDARIZACION DE INTERNET

Surge un grupo internacional de voluntarios llamado *Internet Society* para administrar la colección de protocolos TCP/IP. Los estándares para el TCP/IP son publicados en una serie de documentos llamados *Request For Comments*, o simplemente **RFC**s. Debemos tener presente que Internet nació como libre y sigue como libre. Por tanto esta no es una organización "propietaria" de Internet o de sus tecnologías. Únicamente son responsables de su dirección.

ISOC

I*nternet* ***SOC****iety* (ISOC) fue creada en 1992 como una organización global responsable de las tecnologías de trabajo en Internet y las aplicaciones de Internet. Su principal propósito es animar al desarrollo y la disponibilidad de Internet.

IAB

La IAB (***I****nternet* ***A****rchitecture* ***B****oard)* es el grupo técnico de la ISOC responsable de las opciones estándar de Internet, publicar las RFCs y vigilar los procesos estándar de Internet.

El IAF dirige la IETF (*Internet Engineering Task Force*), IANA (*Internet Assiged Numbers Authority*) y la IRTF (*Internet Research Task Force*). La IETF desarrolla los estándares y protocolos Internet, y vigila y desarrolla soluciones a problemas técnicos alrededor de Internet. La IANA vigila y coordina la asignación de un identificador único en Internet: las direcciones IP. El grupo IRTF es el responsable de la coordinación de todos los proyectos relacionados con el TCP/IP.

RFCs

Los estándares del TCP/IP se publican en una serie de documentos llamados *Request For Comment* (RFCs). Los RFCs describen todo el trabajo interno en Internet. El estándar TCP/IP es siempre publicado como una RFC, pero no todas las RFCs especifican estándares.

El TCP/IP estándar, no ha sido desarrollado por un comité. Ha sido desarrollado "por consenso". Cualquier miembro de la *Internet Society* puede publicar un documento como una SFC. El documento es revisado por un grupo de expertos y se le asigna una clasificación. Hay cinco clases de clasificaciones en las RFCs:

Required: (requerido) Debe ser implementado en todas las maquinas que ejecuten TCP/IP inclusive los *gateway* y *routers*.

Recommended: (recomendada) Se estimula el que todas las maquinas que ejecuten TCP/IP utilicen esta especificación de la RFC. Las RFC recomendadas, normalmente están siendo utilizadas en todas las maquinas.

Elective: El uso de esta RFC es opcional. No es ampliamente usada.

Limited Use: (uso limitado) No esta pensada para un uso general.

Not recommended: (no recomendada) No está aconsejada su uso.

Si un documento comienza a ser considerado como un estándar, comienza a pasar por los diferentes estados de desarrollo, prueba y aceptación. Durante este proceso, estos procesos son formalmente llamados *'maturity levels'* (niveles de maduración). Hay tres niveles de maduración en los estándares de Internet:

Proposed Standard: (propuesta). Una especificación de propuesta, es generalmente estable, ha resuelto las conocidas alternativas de diseño, está bien comprendida, ha recibido el visto bueno de la comunidad y parece un buen candidato a ser evaluado po la comunidad.

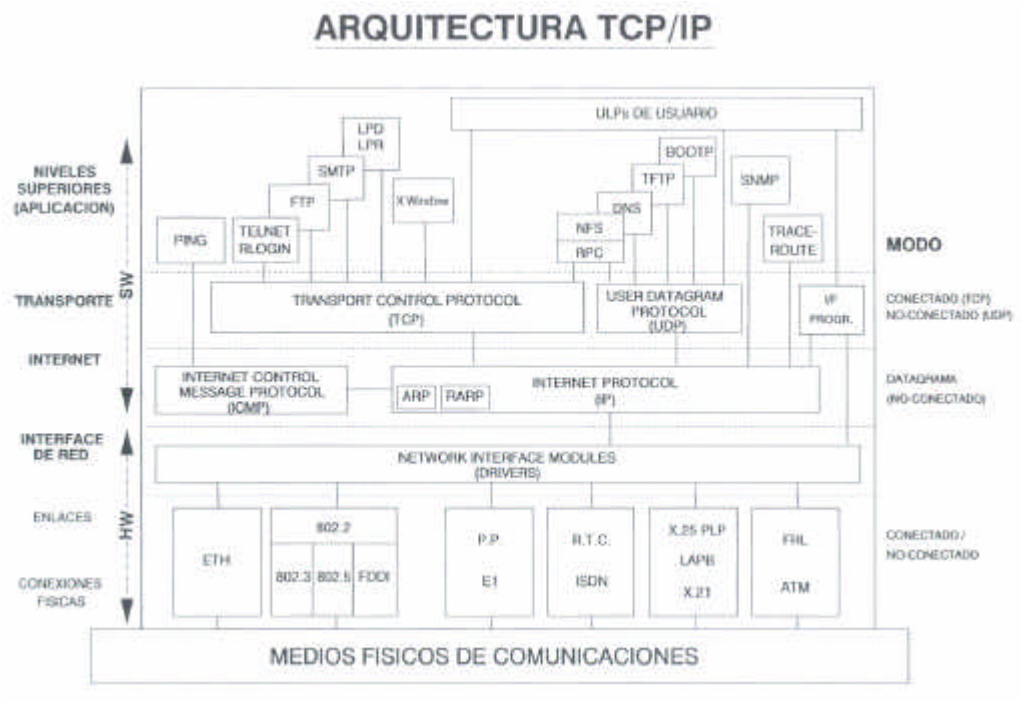
Draft Standard: (borrador). Un borrador, debe ser entendido y reconocido como estable, tanto semánticamente como su base para poder ser desarrollada correctamente.

Internet Standard: El estándar Internet, (muchas veces nos referimos a él como un 'estándar' simplemente) se caracteriza por un alto grado de madurez técnica y generalmente se reconoce como una ayuda al protocolo o al servicio que significa un beneficio para la comunidad Internet.

Cuando se publica un documento, se le asigna un numero de RFC. Este numero original, nunca va a cambiar. Si esta RFC requiera cambios, se publica una nueva RFC con un nuevo numero.

La IAB publica el *'IAB Official Protocol Standard'* trimestralmente.

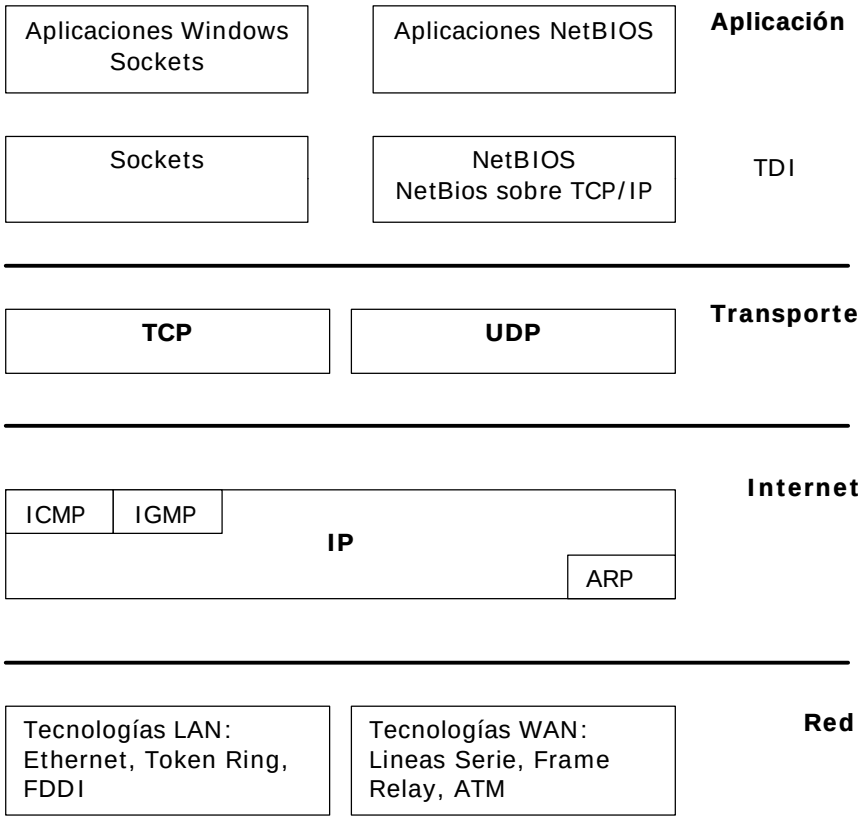
VISION GENERAL DE LA ARQUITECTURA TCP / IP



EL PROTOCOLO TPC/IP

EL MODELO DE 4 CAPAS

Aunque este modelo es general en todas las implementaciones del TCP/IP, a lo largo del presente documento, vamos a ceñirnos a su implementación en Microsoft Windows.



Capa de Red: La base de este modelo de capas, es la capa de *interface* de red. Esta capa es la responsable de enviar y recibir *frames* (estructuras o marcos. Pero prefiero a partir de ahora dejar el termino inglés, ya que es ampliamente aceptado en la terminología informática), los cuales son los paquetes de información que viajan en una red cono una 'unidad simple'. La capa de red, envía *frames* a la red, y recupera *frames* de la red.

Capa de Internet: Este protocolo encapsula paquetes en *datagramas* internet (no es tampoco la palabra *datagrama* una palabra castellana, pero es también aceptada en la terminología informática como 'paquete de datos') y además esta capa ejecuta todos los algoritmos de enrutamiento (*routing*) de paquetes. Los cuatro protocolos Internet son: *Internet Protocol* (IP), *Address Resolution Protocol* (ARP), *Internet Control Message Protocol* (ICMP) y *Internet Group Management Protocol* (IGMP).

- IP es el responsable del envío y enrutamiento de paquetes entre maquinas y redes.
- ARP obtiene las direcciones de hardware de las maquinas situadas en la misma red física. Recordemos que la dirección física de cada tarjeta de red es única en el mundo. Dicha dirección "física" ha sido implementada vía hardware por el fabricante de la tarjeta de red, y dicho fabricante, lo selecciona de un rango de direcciones único asignado a él y garantiza la unicidad de dicha tarjeta. Este caso es el más corriente y es el de las tarjetas de Red Ethernet. Existe para otras topologías de red, igualmente una asignación única hardware de reconocimiento de la tarjeta.
- ICMP envía mensajes e informa de errores en el envío de paquetes.

- IGMP se utiliza para la comunicación entre *routers* (enrutadores de Internet).

Capa de Transporte: La capa de transporte, nos da el nivel de "sesión" en la comunicación. Los dos protocolos posibles de transportes son TCP (*Transmission Control Protocol)* y UDP *(User Datagram Protocol)*. Se puede utilizar uno u otro protocolo dependiendo del método preferido de envío de datos.

El TCP nos da un tipo de conectividad "orientada a conexión". Típicamente se utiliza para transferencias de largas cantidades de datos de una sola vez. Se utiliza también en aplicaciones que requieren un "reconocimiento" o validación (ACK : *acknowledgment*) de los datos recibidos.

El UDP proporciona conexión de comunicación y no garantiza la entrega de paquetes. Las aplicaciones que utilicen UDP normalmente envían pequeñas cantidades de datos de una sola vez. La aplicación que lo utilice, es la responsable en este caso de la integridad de los paquetes y debe establecer sus propios mecanismos para pedir repetición de mensaje, seguimiento, etc, no existiendo ni garantía de entrega ni garantía del orden de entrega en la maquina destino.

Capa de Aplicación: En la cima de este modelo, está la capa de aplicación. Esta es la capa que las aplicaciones utilizan para acceder a la red. Existen muchas utilidades y servicios en la capa de aplicación, por ejemplo: FTP, Telnet, SNMP y DNS.

Otras implementaciones del TCP, por ejemplo Unix' estándar, nos da únicamente en la capa de aplicación los "sockets". Microsoft Windows nos da dos *interfaces* para las aplicaciones de red que usan los servicios del TCP/IP. El primero, llamado Windows Sockets' nos da una *interface* de programación estándar (API) bajo Microsoft Windows para protocolos de transporte como el TCP/IP y el IPX.

La segunda *interface* para las aplicaciones de red es el NetBIOS. Esta *interface* nos da un estándar para el soporte de nombres NetBIOS y servicios de mensajes, usados en TCP/IP y el NetBeui.

TECNOLOGÍAS DE *INTERFACE* DE RED

El IP utiliza la especificación de dispositivos de red (NDIS: *Network Device Interface specification)* para enviar *frames* a capa de Red. IP soporta tecnologías LAN y WAN.

Las tecnologías LAN soportadas por el TCP/IP incluyen Ethernet (Ethenet II y 802.3) Token Ring, ArcNet y tecnologías MAN (*Metropolitan Area Network*) como la *interface* de datos en fibra óptica (FDDI: *Fiber Distributed Data Interface)*.

Usando TCP/IP en un entorno WAN puede requerir los servicios de RAS activados, o incluso hardware adicional. Las dos máximas categorías de las tecnologías WAN soportadas son: líneas serie y *packets-switched*. Las líneas serie incluyen las llamadas telefónicas analógicas. *Packets-switched* incluyen X.25, *frame relay*, y comunicaciones en modo de transferencia asíncrona (ATM: *Asynchronous Transfer Mode)*.

Protocolos sobre líneas serie

El TCP/IP es enviado en las líneas serie encapsulado con los protocolos SLIP (*Serial Line Internet Protocol)* o bien bajo PPP (*Point-to-Point Protocol).* Este ultimo caso es el que normalmente utilizamos al utilizar cualquier módem para conectarnos a Internet.

SLIP es un estándar de la industria desarrollado a mediados de 1980 para soportar TCP/IP en redes de baja velocidad. Utilizando el RAS de Windows NT (o Windows 2000) las maquinas ejecutando Windows, pueden usar TCP/IP y SLIP para comunicarse a maquinas remotas.

- Debemos recordar que Windows NT (o Windows 2000) soportan la funcionalidad de cliente SLIP, no la funcionalidad de servidor SLIP.

PPP fue diseñado como una mejora de la funcionalidad SLIP original. El PPP es un protocolo de enlace de datos que nos da un método estándar de enviar paquetes a la red en un enlace punto a punto. Debido a que PPP nos da mayor seguridad, configuración, y detección de errores que SLIP, es el protocolo recomendado para comunicaciones en líneas serie.

La transmisión de IP sobre líneas serie está descrita en la RFC 1055. El PPP está definido en las RFCs 1547 y 1661.

ARP

Cualquier maquina debe conocer 'siempre' la dirección hardware (física) de otra maquina para poder comunicarse vía red. La resolución de direcciones es el proceso de convertir direcciones IP en direcciones hardware. El ARP (*Address Resolution Protocol),* es parte de las capas del TCP/IP, obtiene direcciones hardware de las maquinas localizadas en la misma red física.

Vamos a introducir aquí un termino inglés *'broadcast'* o *'broadcasting'* que voy a dejar de dicha manera debido a su uso cotidiano en el lenguaje de redes y debido a que su traducción al castellano no tiene todo el sentido que dicha palabra expresa. Su traducción literal es: radiodifusión. Realmente lo que indica es que un mensaje es enviado a la red a todas las maquinas y lo recogerá la maquina que sé de por 'enterada'.

ARP es el responsable de obtener las direcciones hardware de las maquinas TCP/IP en redes basadas en *'broadcasting'*. ARP usa un *broadcast* local de la dirección IP de destino para localizar la dirección hardware de la maquina destino o del *gateway*.

(por *gateway* debemos entender un *router* –enrutador- o cualquier maquina que nos de salida desde nuestra red local o nuestro segmento de red local, a otros segmentos locales de red o bien a otras redes, como por ejemplo Internet).

Una vez que el ARP obtiene la dirección hardware, ambos, la dirección IP y la dirección hardware son almacenadas en una entrada en la *caché* ARP. ARP siempre chequea la *caché* ARP para una dirección IP antes de iniciar una petición mediante *broadcast* a la red.

La resolución de direccionamiento inverso es el proceso de convertir una dirección hardware en una dirección IP. No todas las implementaciones del TCP/IP soportan esto. Por ejemplo, Microsoft Windows no soporta resolución inversa de direcciones.

El ARP está definido en la RFC 826.

Resolviendo una dirección IP local.

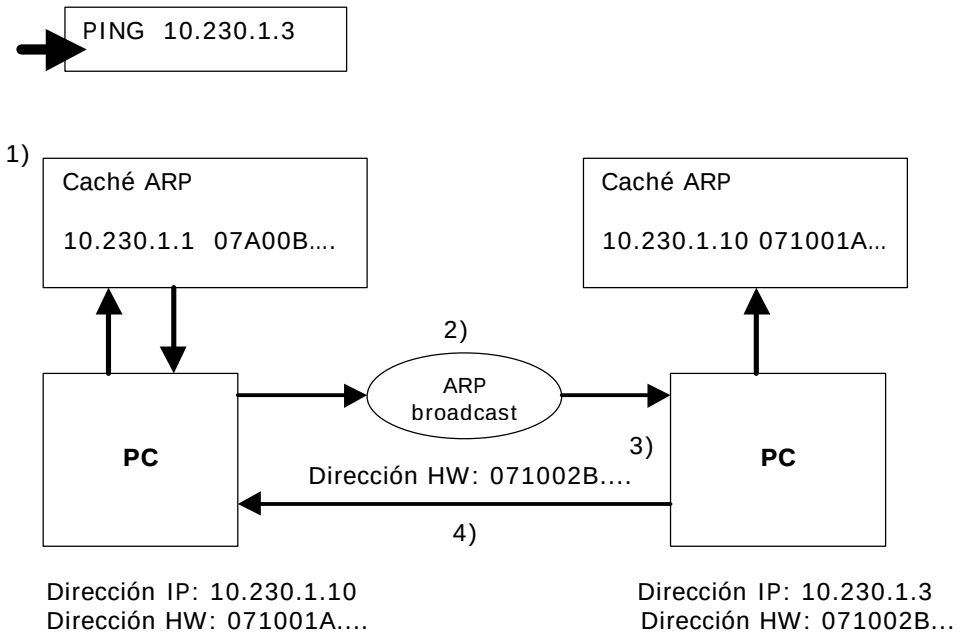
Antes de que la comunicación entre dos maquinas pueda ocurrir, la dirección IP de cada maquina debe ser resuelta como dirección física (hardware). El proceso de resolución de direcciones incluye una petición ARP y una respuesta ARP. Vamos a intentar explicarlo en el siguiente ejemplo:

1) Una petición ARP se inicia en cualquier momento en que una maquina intenta comunicarse con otra. Cuando el IP determina que la dirección IP es en la red local, la maquina origen de la petición chequea su propia *caché* ARP para ver si allí tiene la dirección hardware de la maquina destino.

2) Si no encuentra la dirección en su propia *caché*, ARP envía una petición con una pregunta del tipo "Quien tenga esta dirección IP, que me envíe su dirección hardware". Tanto la dirección IP del origen como su dirección hardware son incluidas en el mensaje de petición. El mensaje de petición es enviado mediante *broadcast* a todas las maquinas en la red local.

3) Cada maquina en la red local recibe el mensaje enviado por *broadcast* y comprueba si se está solicitando su propia dirección IP. Si el mensaje no es para esa maquina, ignora dicha petición.

4) Al final, la maquina de destino comprueba que la petición le coincide con su propia dirección IP y envía una respuesta ARP directamente a la maquina peticionaria ya que en el propio mensaje va la dirección hardware del peticionario. Este además actualiza su propia *caché* ARP con la direccio IP / dirección hardware de la maquina peticionaria. La comunicación se establecerá cuando la maquina origen reciba la respuesta.



Resolviendo una dirección IP remota.

ARP también nos permite que dos maquinas de diferentes redes se comuniquen. En esta situación la petición ARP mediante *broadcasting* es para el *gateway* por defecto y no para la dirección IP de la maquina destino. Es decir la petición *broadcast* es para determinar el *router* que puede enviar los *datagramas* a la maquina destino en la red. Veamos el siguiente ejemplo:

1) Cuando iniciamos la petición, la dirección de destino IP se identifica como perteneciente a una red 'remota'. La maquina origen chequea su tabla de 'rutas' para encontrar un camino a la maquina o a la red destino. Si no encuentra coincidencia, la maquina origen determina la dirección del *gateway* por defecto. Chequea igualmente su *caché* ARP para la dirección IP / dirección hardware del *gateway* por defecto en este caso.

2) Si no encuentra coincidencia para el *gateway,* entonces se envía una petición ARP mediante *broadcast* para la dirección del *gateway* en vez de para la dirección de la maquina destino. El *router* responderá a la maquina origen con 'su' propia dirección hardware. La maquina origen, entonces envía los paquetes de datos al *gateway* para que este a su vez y siguiendo un proceso similar, los reenvíe a la maquina destino.

3) En el *router,* la dirección IP destino también es investigada para ver si es local o remota.Si es local, el *router* usa la técnica ARP (primero en la *caché* y luego por *broadcast)* para obtener su dirección hardware. Si es una dirección remota, el *router* chequea su tabla de rutas para encontrar un *gateway* para esa dirección y entonces usa ARP (*caché* o *boradcast)* para obtener la dirección hardware del siguiente *gateway* hasta el destino. El paquete se envia a la siguiente maquina de detino.

4) Después de que la maquina destino reciba la petición, esta responde con un mensaje de respuesta ICMP. Debido a que la maquina origen está en una red remota, la tabla de rutas local se chequea para encontrar un *gateway* para la dirección de la maquina origen. Cuando encuentra un *gateway*, ARP obtiene su dirección hardware.

5) Si la dirección hardware del *gateway* no está en la *caché* ARP una petición *broadcast* obtendrá esta. Una vez obtenida su dirección hardware, la respuesta ICMP es enviada al *router* que encaminará estos datos a la maquina origen.

Bien. En este caso el grafico de envió y peticiones es algo más complicado que el anterior. Se propone como ejercicio y espero que no tenga mayores complicaciones que la de hacer intervenir una maquina más: un *router.*

La *caché* ARP

Para intentar minimizar el numero de *broadcast* a la red, el ARP mantiene siempre las direcciones de hardware conocidas y que fueron resueltas por primera vez mediante *broadcasting.*

Cada entrada en la *caché* de la ARP tiene un tiempo potencial de vida de 10 minutos. En cada entrada en la ARP, se guarda los datos de fecha / hora (*timestamp*). Si esta entrada no es usada en los dos primeros minutos, se borra de la *caché.* Si se utiliza será borrada después de los 10 primeros minutos. Si la *caché* del ARP alcanza su capacidad máxima antes de que las entradas anotadas en ella expiren, la entrada más antigua será borrada y la nueva será añadida en su lugar.

En algunas implementaciones del TCP/IP cuando una entrada de la *caché* del ARP es utilizada, se le añaden otros 10 minutos de vida. En Microsoft Windows no esta implementada esta característica.

ICMP e IGMP

Mientras que el IP es el protocolo para el envío, ICMP (*Internet Control Message Protocol)* nos informa de errores y mensajes de control en nombre del IP. IP usa IGMP (*Internet Group* Management *Protocol)* para informar a los *routers* que los *hosts* (maquinas) de un grupo especifico están disponibles en una red.

ICMP

ICMP no espera convertir al IP en un protocolo seguro y fiable. Meramente comunica errores e informa de condiciones especificas. Los mensajes ICMP son enviados como *datagramas* IP y son por tanto inseguros en sí mismos.

Por ejemplo, si una maquina está enviando *datagramas* a otra maquina a una velocidad que está saturando *routers* o enlaces entre ellos, el *router* puede enviar un mensaje ICMP del tipo *Source Quench*. Este mensaje le informa al *host* de una petición de disminuir su velocidad de trasmisión.

Dependiendo de la implementación del TCP/IP, este mensaje ICMP puede ser o no enviado al origen. Existen implementaciones que en vez de enviar este mensaje, o bien si no se ha respondido a la petición de este mensaje, simplemente "pierde" o "descarta" los *datagramas* que no puede procesar en el envío sin preocuparse más de ellos.

- Recordemos que el TCP/IP sigue la filosofía de la "patata caliente" con los *datagramas* IP. Si puede enviarlos rápidamente, los envía. Sino simplemente la "patata" se tira. Está caliente y quema por tanto no podemos guardárnosla.

ICMP está definido en la RFC 792

IGMP

Es el equivalente a los mensajes ICMP pero entre los *routers* en vez de entre maquinas en las que interviene un *host* en alguno de sus extremos.

Los mensajes IGMP son enviados como *datagramas* IP y son por tanto inseguros en si mismos.

IGMP está definido en la RFC 1112

IP

IP es el protocolo primario de conexión responsable del envío y enrutamiento de paquetes entre maquinas (*hosts*).

IP no establece una sesión antes de intercambiar datos. IP no es fiable debido a que trabaja sin garantía de entrega. A lo largo del camino, un paquete puede perderse, cambiarse de secuencia, duplicarse, retrasarse, o incluso trocearse.

IP no requiere una comunicación ACK (*acknowledgment)* cuando se reciben los datos. El emisor o el receptor no es informado cuando un paquete se pierde o se envía fuera de secuencia. El ACK de los paquetes es responsabilidad de una capa de más alto nivel de transporte como por ejemplo el TCP.

Si el IP identifica una dirección de destino como una dirección "local", el IP envía el paquete directamente a la maquina. Si el destino es identificado como un destino "remoto", el IP chequea sus tablas de rutas. Si encuentra una ruta, el IP envía el paquete utilizando esa ruta. Si no encuentra una ruta, el IP envía el paquete al *gateway* por defecto (tan bien llamado *router*).

Al *datagrama* se le añaden los campos descritos a continuación a su cabecera cuando se pasa un paquete a la capa de transporte.

- Dirección IP del origen
- Dirección IP del destino
- Protocolo (TCP o UDP)
- *Cheksum* (un numero formado por un sencillo algoritmo matemático que nos garantice la integridad d todo el paquete IP recibido).
- *Time To Live* (TTL) Tiempo de vida. Es el lapso de tiempo en el cual va a vivir el *datagrama* antes de que sea descartado.

IP en el *router.*

Cuando un *router* recibe un paquete, el paquete es pasado a la capa IP la cual realiza lo siguiente:

1) Decrementa el campo TTL (*Time To Live)* al menos en 1. Puede ser disminuido en una cantidad mayor si el *router* estuviese congestionado. Si el TLL alcanza el valor de cero, el paquete será descartado.

2) El IP puede fragmentar el paquete en paquetes más pequeños si el paquete fuese demasiado largo para las líneas de salida del *router.*

3) Si el paquete es fragmentando, el IP crea una nueva cabecera para cada nuevo paquete la cual incluye:

   - Un *flag* (indicador) de que le siguen nuevos fragmentos.
   - Un numero de fragmento (*Fragment ID)* para identificar todos los fragmentos que continúan.
   - Un desplazamiento (*Fragment Offset)* para permitir que la maquina que lo va a recibir sea capaz de reensamblar el paquete.

4) El IP calcula los nuevos *cheksum.*

5) El IP obtiene la dirección hardware del siguiente *router.*

6) Envía el paquete.

En el siguiente *host*, el paquete subirá en el *stack* (pila o capa de protocolos) hasta el TCP o el UDP. Este proceso se repite en cada *router* hasta que el paquete encuentra su destino final. Cuando el paquete llega a su destino final el IP ensamblará las piezas tal y como estaba el paquete original.

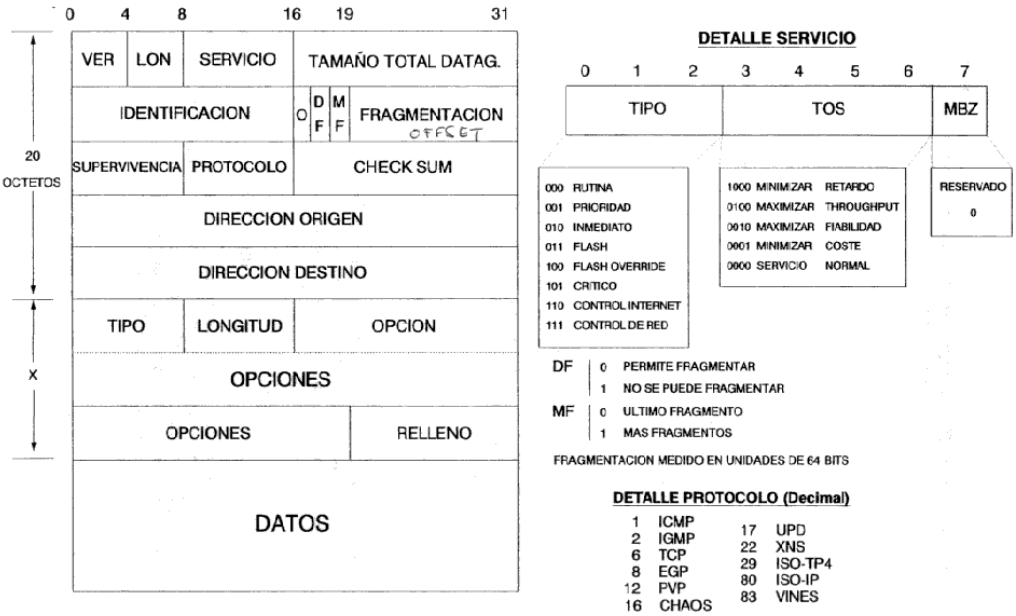
Estructura del paquete IP

Los campos del paquete IP en la versión 4 del TCP/IP (versión actual) son los siguientes:

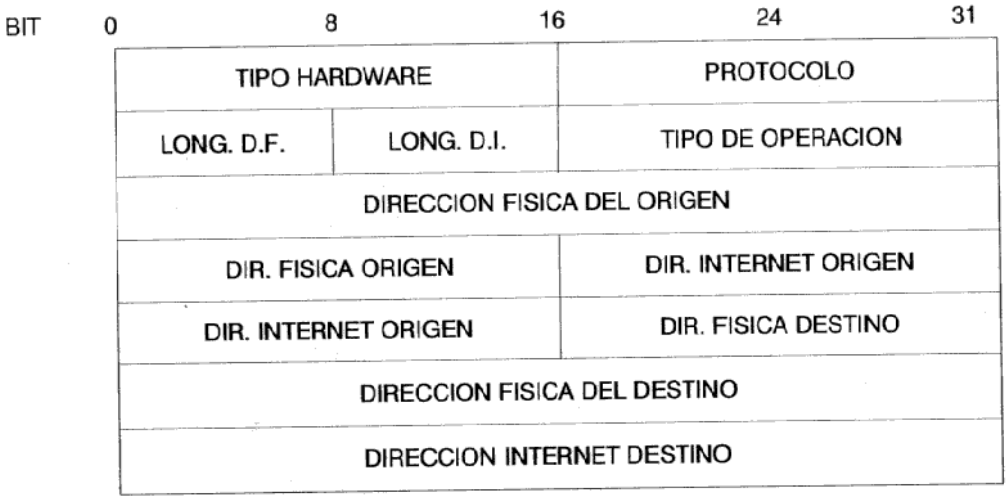
Campo	Función
Versión	Son utilizados 4 bits para indicar la versión dl IP. La versión actual es la versión 4. La siguiente versión del IP va a ser la versión 6.
Longitud de la cabecera	Se utilizan 4 bits que indican el numero de palabras de 32-bits en la cabecera IP. La cabecera IP tiene un mínimo de 20 bytes.
Tipo de servicio	Se utilizan 8 bits para indicar la calidad del servicio esperado por este *datagrama* para entrega a través de los *routers* en la red IP. Especifican procedencia, demora, y tipo de entrega.
Longitud total	16 bits son utilizados para indicar la longitud 'total' incluida cabecera del *datagrama* IP.
Identificación	16 bits son utilizados para identificar este paquete. Si el paquete fuese fragmentado, todos los segmentos que tuviesen esta misma identificación serán usados para reensamblarlos en la maquina destino.
Flags de fragmentación	3 bits son reservados como indicadores del proceso de fragmentación. Sin embargo únicamente 2 bits están definidos para el proceso en curso. Uno de ellos es para indicar cuando el *datagrama* puede ser fragmentado y el otro para indicar que hay más fragmento que lo siguen.
Offset del fragmento.	13 bits se utilizan como un contador del desplazamiento para indicar la posición del fragmento relativo al paquete IP original. Si el paquete no estuviese fragmentado este campo contendrá un cero.
Tiempo de Vida (TTL)	8 bits se utilizan para indicar la cantidad de vida o de 'saltos' que un paquete IP puede realizar antes de ser descartado.
Protocolo	8 bits se utilizan como un identificador del protocolo.
Checksum de la cabecera	16 bits son usados como *checksum* de la cabecera IP unicamente. El cuerpo del mensaje IP no está incluido y deberá ser incluido en él, su propio *checksum* para evitar errores.
Dirección Origen	32 bits que almacenan la dirección IP del origen.
Dirección destino	32 bits que almacenan la dirección IP del destino.
Opciones y relleno	Un múltiplo de 32 bits es utilizado para almacenar las opciones IP. Si las opciones IP no utilizan los 32 bits, se rellenan con bits adicionales a ceros para que la longitud de la cabecera IP sea un numero entero de palabras de 32 bits.

Cabecera IP

CABECERO IP



FORMATO ARP/RARP



TIPO HARDWARE
1 - ETHERNET
2 - EXPERIMENTAL
3 - X.25
4 - PRONET
5 - CHAOS
6 - IEEE 802.X
7 - ARCNEN

TIPO OPERACION
1 - ARP REQUEST
2 - ARP RESPONSE
3 - RARP REQUEST
4 - RARP RESPONSE

TCP

TCP es un servicio de entrega orientado a conexión. Es totalmente fiable.

Los datos TCP se transmiten en segmentos y se establece una sesión antes de que las maquinas puedan intercambiar datos. TCP usa comunicaciones en flujo de bytes, es decir los datos son considerados una secuencia de bytes.

Se consigue la seguridad asignando un numero de secuencia a cada segmento transmitido por el TCP. LA recepción de un ACK nos confirma la llegada correcta de un segmento a la otra maquina. Por cada segmento enviado, el receptor debe devolver un ACK en un periodo de tiempo especificado.

Si el emisor no recibe un ACK, entonces el dato se vuelve a transmitir. Si el segmento se recibe dañado el receptor lo descarta inmediatamente. Debido a que el ACK no se envía en este caso, el emisor retransmitirá el segmento.

El TCP está definido en la RFC 793

PUERTOS

Las aplicaciones *'sockets'* se identifican a si mismas de manera única en una maquina por usar un *'protocol port number'*. Por ejemplo, un servidor FTP usa un determinado puerto TCP para que otras aplicaciones puedan comunicarse con él.

Los puertos pueden usar cualquier numero entre 0 y 65536. Los numeros de puerto de cara a aplicaciones "cliente" son dinámicamente asignados por el sistema operativo cuando se solicita una petición para este servicio. Los numeros de puertos de las aplicaciones servidoras *well-known* (bien conocidos) han sido preasignados por el IANA (*Internet Assigned Numbers Authority)* y no pueden cambiarse.

Los puertos *well-known* están en el rango del 1 al 1024. La lista completa está documentada en la RFC 1700.

SOCKETS

Un *socket* es un concepto similar a un manejador de fichero y este funciona como un punto final de la comunicación de red. Una aplicación crea un *socket* especificando tres ítem: La dirección IP de la maquina, el tipo de servicio (TCP para servicio orientado a conexión, UDP para servicio orientado a datos) y el puerto que la aplicación esta usando.

Una aplicación puede crear un *socket* para enviar trafico orientado a datos a una aplicación remota. Una aplicación también puede crear un *socket* y conectar esta a otra aplicación *socket.* Los datos serán fiables enviados bajo esta conexión.

Puertos TCP

Un puerto TCP nos da una localización para la entrega de mensajes. Los numeros de puertos inferiores a 256 son definidos como los puertos mas corrientemente usados. Por ejemplo podemos fijarnos en los siguientes puertos:

21	FTP
22	Telnet
53	Domain Name System (DNS)
139	Servicio de Sesión NetBIOS

Sesiones TCP

Una sesión TCP se inicia en tres vías. El propósito de estas tres vías es sincronizar el envío y la recepción de segmentos, informar a la otra maquina de la cantidad de datos que es capaz de recibir de una tacada y establecer una conexión virtual.

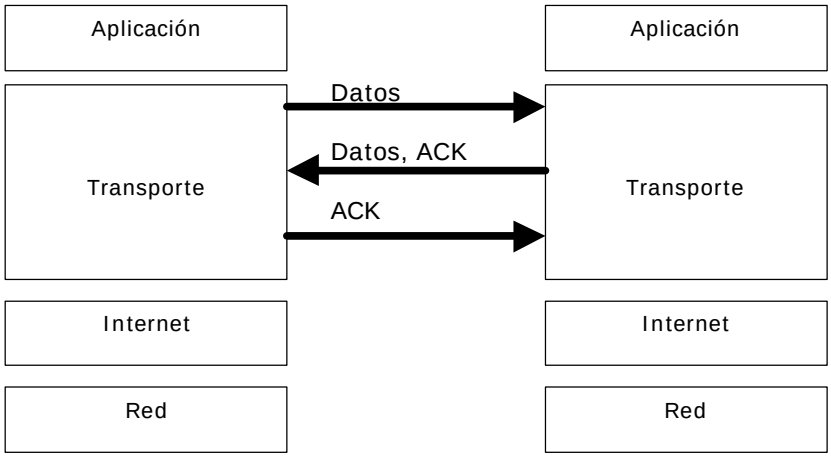
Estos son los pasos seguidos:

1) La maquina que inicia una sesion envia un segmento con el *flag* (indicador) de sincronización (SYN) activado.

2) La maquina receptora envía un ACK a la petición devolviendo un segmento con:

   - El *flag* de sincronización colocado.
   - Un numero de secuencia que indica el byte de comienzo para el segmento que acaba de ser enviado.
   - Un ACK con el numero de secuencia del primer byte del siguiente segmento que espera recibir.

3) El host peticionario vuelve a enviar un segmento con el numero de secuencia ACK. En este momento la conexión queda establecida.



El TCP utiliza un proceso similar para finalizar una conexión. Esto garantiza que ambas maquinas han terminado de transmitir y recibir todos los datos.

Ventanas de apertura en el TCP

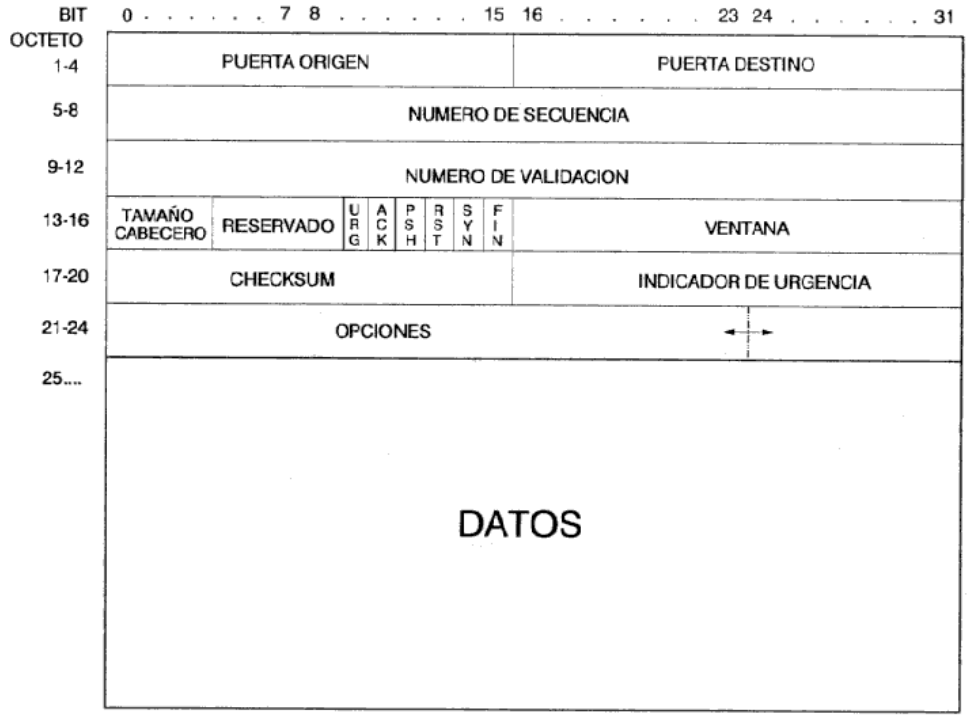
Los búferes TCP para transmisión entre dos maquinas se realiza utilizando ventanas. Cada maquina TCP mantiene dos ventanas: una para recibir datos y otra para enviar datos. El tamaño de las ventanas, indica la cantidad de datos que pueden mantenerse en los búferes en una de las maquinas.

Estructura de los paquetes TCP

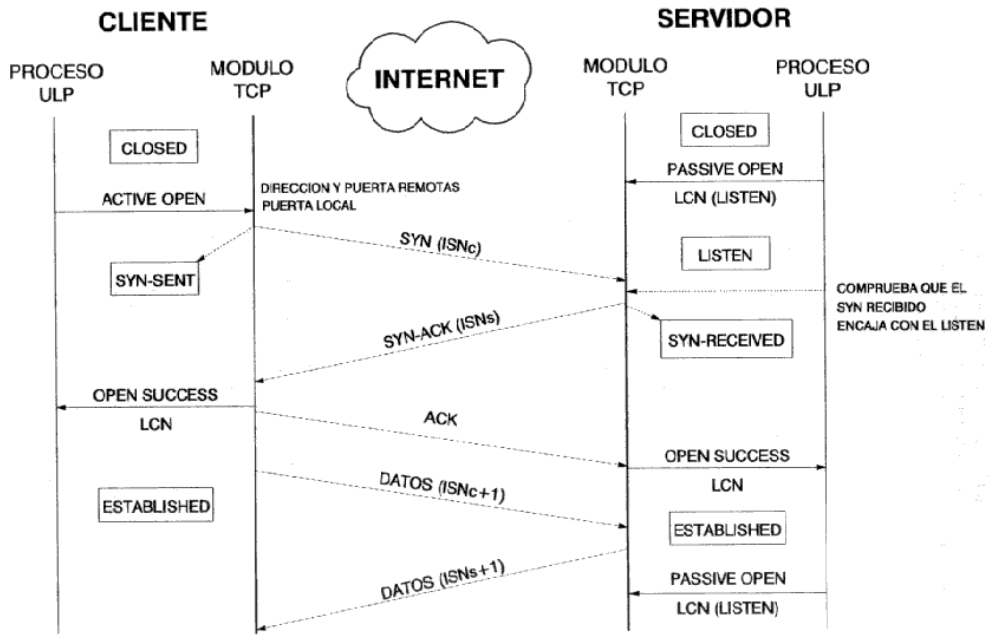
Todos los segmentos TCP tienen dos partes: datos y cabecera. La siguiente tabla lista los campos de la cabecera del TCP:

Campo	**Función**
Puerto Origen	Puerto TCP de la maquina 'emisora' de datos
Puerto destino	Puerto TCP de la maquina destino.
Numero de ACK	El numero de secuencia del próximo byte que se espera recibir.
Longitud de datos	Longitud del segmento TCP
Reservado	Reservado para uso futuro
Flags	Este campo especifica cual es el contenido del segmento.
Ventana	Cuanto espacio queda disponible en la ventana TCP
Checksum	Numero de control para verificar que la cabecera es correcta.
Apuntador 'urgente'	Cuando se están enviando datos 'urgentes')especificados así en el campo *flags*) este campo apunta al final de los datos urgentes en el segmento.

CABECERO TCP



ESTABLECIMIENTO DE CONEXION TCP



UDP

***U**ser **D**atagram **P**rotocol* ' UDP es un servicio de envío de *datagramas* sin garantía de entrega. A este método se le denomina ho conectado ' al contrario que el TCP que al establecer una sesión, se le denomina 'conectado '. Por tanto, la llegada al destino de un *datagrama* o la secuencia correcta de entrega no está garantizada.

UDP se utiliza en las aplicaciones que no requieren un ACK (*acknowledgement*) de acuse de recibo de recepción de datos. Las aplicaciones que lo utilizan son típicamente las aplicaciones que transmiten pequeñas cantidades de datos a la vez. Por ejmplo, aplicaciones que lo utilizan son, el servicio de nombres NetBIOS y el SNMP (un protocolo de control de redes. No confundirlo con el SMTP de correo electrónico).

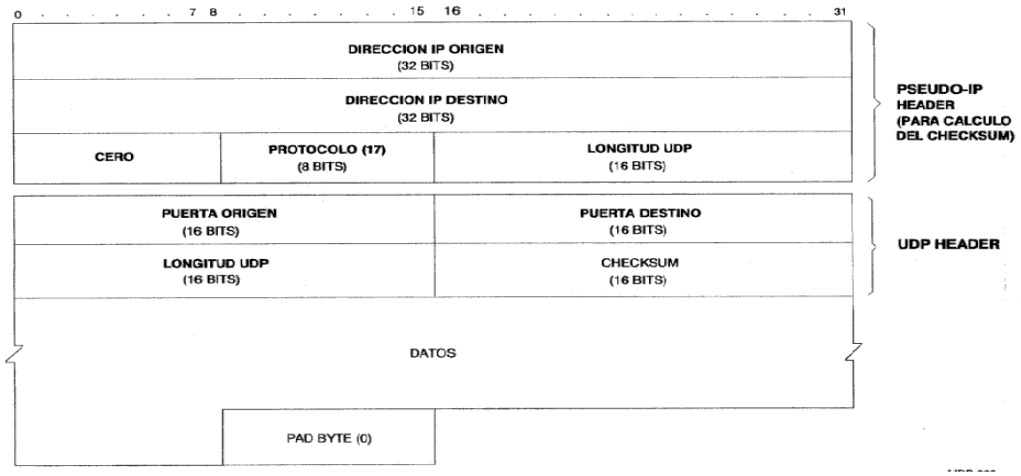
Puertos UDP

Para utilizar UDP, una aplicación debe dar una dirección IP y un numero de puerto de la aplicación destino. Un puerto, funciona como una cola de mensajes multiplexados que puede recibir múltiples mensajes al tiempo. Es importante resaltar que los puertos que vamos a mencionar en la siguiente tabla son puertos UDP y son distintos de los puertos TCP aún cuando algunos de ellos puedan tener el mismo numero.

15	NETSTAT	Estado de la red
53	DOMAIN	DNS (*Domain Name Server)*
69	TFTP	*Trivial File Transfer Protocol*
137	NETBIOS-NS	Servicio de nombres NETBIOS
138	NETBIOS-DGM	Servicio de datagramas NETBIOS
161	SNMP	Monitor de red SNMP

El UDP está definido en la RFC 768

FORMATO UDP



DIRECCIONAMIENTO IP

LA DIRECCIÓN IP

La dirección IP identifica la localización de un sistema en la red. Equivale a una dirección de una calle y número de portal. Es decir, es única. No pueden existir en la misma ciudad dos calles con el mismo nombre y números de portal.

Cada dirección IP tiene dos partes. Una de ellas, identifica a la RED y la otra identifica a la maquina dentro de esa red. Todas las maquinas que pertenecen a la misma red requieren el mismo numero de RED el cual debe ser además único en Internet.

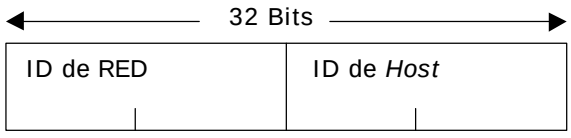
El número de maquina, identifica a una *workstation*, servidor, *router* o cualquier otra maquina TCP/IP dentro de la red. El número de maquina (número de *host)* debe ser único para esa red. Cada *host* TCP/IP, por tanto, queda identificado por una dirección IP que debe ser única

Identificación de RED e identificación de *Host*

Hay dos formatos para referirnos a una dirección IP, formato binario y formato decimal con puntos. Cada dirección IP es de 32 bits de longitud y está compuesto por 4 campos de 8 bits, llamados bytes u octetos. Estos octetos están separados por puntos y cada uno de ellos representa un numero decimal entre cero y 255. Los 32 bits de una dirección IP contienen tanto la Identificación de RED como la Identificación de *Hosts* dentro de la RED.

La manera mas fácil de "leer" para los humanos una dirección IP es mediante la notación decimal con puntos. Vamos a ver a continuación un ejemplo de una dirección IP en binario y decimal con puntos:

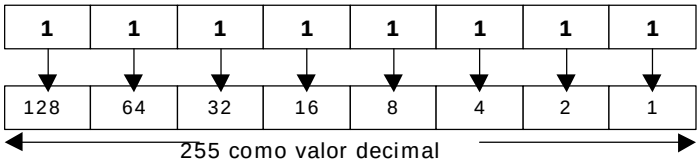
10011001110111000011010100001111 153.220.53.15



w.x.y.z

Convirtiendo direcciones IP de binario a decimal.

Para convertir las direcciones de binario a decimal recordemos que cada bit de un octeto tiene asignado un valor decimal. Cuando convertimos cada bit a formato decimal, el mayor valor de un octeto es 255. Cada octeto se convierte separadamente.



Una manera rápida de convertir de binario a decimal y al revés, es mediante la calculadora de Windows.

CLASES DE DIRECCIONES

Hay dos diferentes clases de direcciones IP. Cada clase define la parte de la dirección IP que identifica a la RED y la parte que identifica al número de *hosts* dentro de esa red.

La comunidad Internet ha definido 5 clases de direcciones para poder acomodar redes de diferentes tamaños. El TCP/IP de Microsoft soporta las clases A, B y C. Estas clases, definen que bits son usados para la red y cuales son usados para identificar el número de *host* dentro de la red.

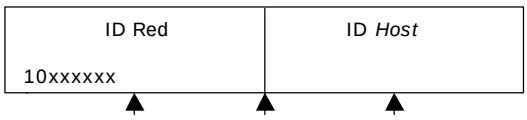
Se puede identificar la clase de dirección por el número del primer octeto. Recordemos que por ser un numero de 32 bits la dirección IP, teóricamente podrían existir 2 elevado a la 32 direcciones diferentes IP.

La clase A, son direcciones del tipo w.x.y.z en donde 'w' representa la RED y x.y.z el número de *host* dentro de la red. En el siguiente cuadro podemos ver las clases A, B y C.

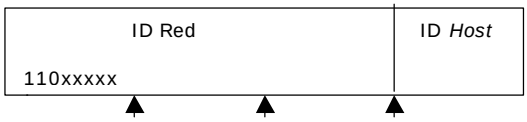
Clase	**Dirección IP**	**ID de Red**	**ID de *Host***
A	w.x.y.z	w	x.y.z
B	w.x.y.z	w.x	y.z
C	w.x.y.z	w.x.y	z

Clase A:

Clase B:



Clase C:



Clase A

Las direcciones de Clase A son asignadas a redes con un elevado numero de *hosts*. El bit de mayor orden en una dirección de clase A siempre es un cero. Los siguiente 7 bits que completan el primer octeto es la identificación de RED. Los restantes 24 bits (los 3 últimos octetos) representan el número de *host.* Esto permite en total 126 redes y aproximadamente 17 millones de *host* por cada red.

Clase B

Las direcciones de clase B son asignadas a redes de tamaño mediano / grande. Los dos primeros bits del primer octeto de las direcciones de clase B son siempre 1 0. Los siguientes 14 bits que completan los dos primeros octetos son la identificación de la RED. Los restantes 16 bits de los dos últimos octetos representan la Identificación del *host.* Esto supone 16.384 redes y aproximadamente 65.000 *hosts* en cada red.

Clase C

La clase C se utiliza para pequeñas LANs (redes de área local). Los tres primeros bits del primer octeto son siempre 1 1 0. Los siguientes 21 bits que completan los 3 primeros octetos representan la Identificación de una red en Clase C. Los últimos 8 bits (ultimo octeto) representa la Identificación del *host.* Esto permite aproximadamente 2 millones de redes y 254 *hosts* en cada red.

Clase D

Las direcciones de clase D son usadas para uso de grupos *multicast.* Un grupo *multicast* puede estar formado por uno o más *hosts* o por ninguno de ellos. Los 4 bits de mayor orden en el primer octeto en una clase D son siempre 1 1 1 0. El resto de bits designan el grupo especifico en el cual participa el cliente. No hay redes o Identificaciones de *hosts* de las operaciones de *multicast.* Los paquetes son pasados a una colección de *hosts* en una red. Solo los *hosts* registrados con una direccion *multicasr* van a recibir esos paquetes. Microsoft soporta las direcciones de clase D para las aplicaciones de datos en *multicasting* (radiodifusión) a los *hosts* en un segmento de trabajo Internet. Esto incluye WINS y Microsoft NetShow.

Clase E

La clase E son direcciones experimentales que no están disponibles para uso general y que se reservan para uso futuro. Los 4 bits del byte de mayor orden en una clase E están colocados a 1 1 1 1.

PRINCIPIOS DE DIRECCIONAMIENTO

No existen reglas para asignar direcciones IP. Por tanto se deben seguir ciertos principios para asegurarse que se está asignando un numero válido de Identificación de RED y de *host*.

Vamos a ver como asignar direcciones IP en un entorno de RED.

Hay varios principios que se deben seguir para asignar una Identificación de red y las Identificaciones de *hosts.*

- El ID de RED no debe ser 127. Esta Identificación está reservada para *loopback* (lazo ' para simular una red dentro de un único PC) y para funciones de diagnóstico.
- La identificación de RED y el número de *host* no pueden estar todos a '1'. Si todos los bits están colocados a '1', la dirección se interpreta como una dirección de *broadcast* en vez de una dirección de un *host.*
- La identificación de RED y el número de *host* no pueden estar todos a 0. Si todos los bits están colocados a 0, la dirección se interpreta como 'esta red únicamente'.
- EL número de *host* debe ser único para la Identificación de RED.

Asignando Identificaciones de RED.

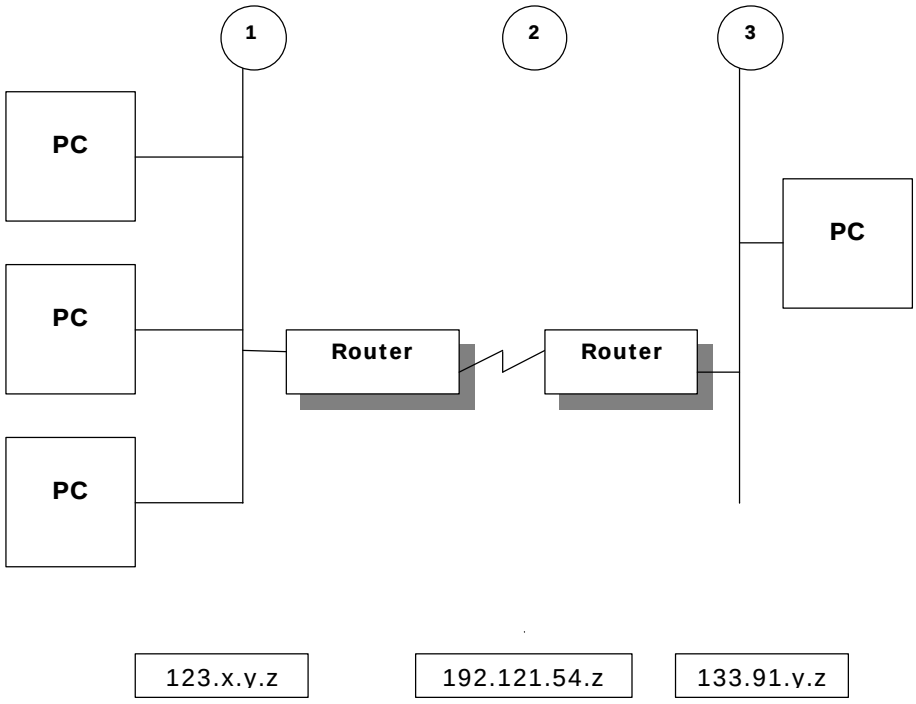
Es necesario un número único de RED para cada red y conexiones de área ancha (*wide area)*. Si nos estamos conectando públicamente a Internet, deberemos obtener una identificación de red del *'Internet Network Information Center' (InterNIC)*. Si no planeamos conectarnos públicamente a Internet, podemos seleccionar cualquier numero o ID de red válido según las premisas anteriores.

La ID de RED identifica los *hosts* que están localizados en la misma RED física. Todos los *hosts* en la misma red física deben tener el mismo número de RED para poder comunicarse unos con otros.

Si nuestra red está conectada por *routers*, un número de RED único es necesario para cada conexión de área ancha (*wide area)*.

Por ejemplo, en el siguiente dibujo:

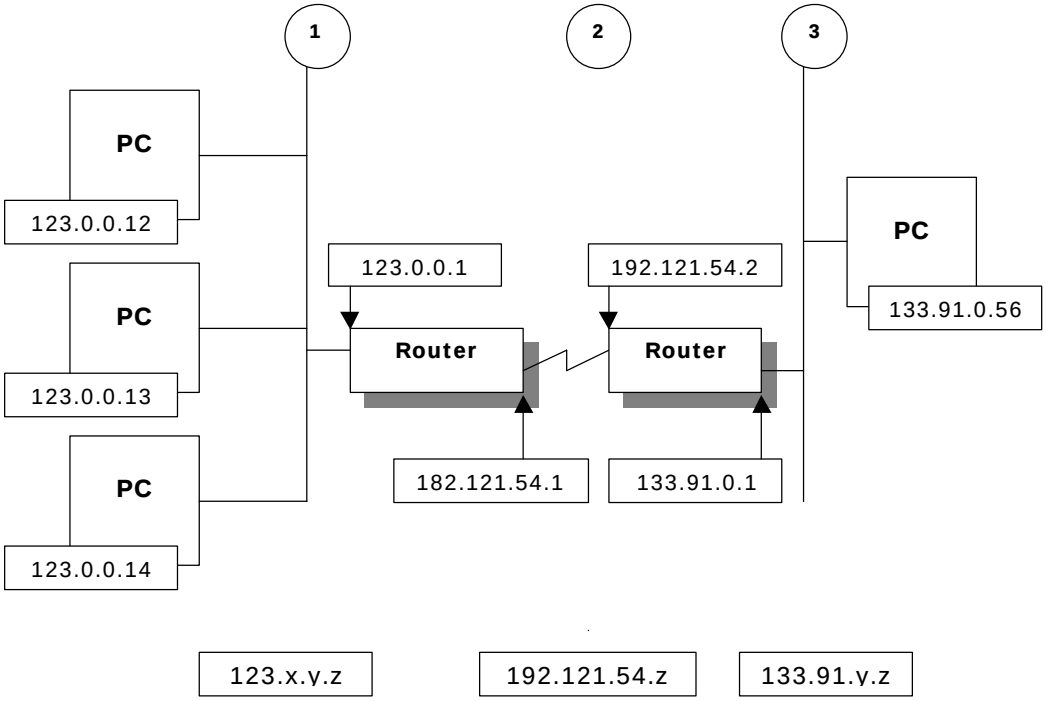
- Redes 1 y 3, representas dos redes conectadas – encaminadas: *routed.*
- Red 2 representa la conexión WAN entre los *routers.*
- La Red 2, requiere una identificación de RED que haga de *interface* entre los dos *routers*.



Nota: Si planeamos conectar nuestra red 'directamente' a Internet, debemos obtener un número de RED único. Para saber como registrar nombres de Dominio y direcciones IP, podemos visitar el registro online de InterNIC en http://internic.net

La asignación de direcciones IP para redes privadas está definida en la RFC 1918.

Asignando ID a los *hosts.*



Un número de *host (ID host)* identifica un *host* TCP/IP en una RED y debe ser único para esa Identificación de RED. Todos los *hosts* TCP, incluyendo las *interfaces* a los *routers* requieren una única ID. La ID del *router* es la dirección IP configurada como una *workstation default gateway* (pasarela por defecto).

En el ejemplo anterior, para el *host* 123.0.0.13 su pasarela por defecto (*default gateway)* sería el 123.0.0.1.

Identificaciones de *hosts* válidas.

La siguiente tabla lista los rangos válidos para ID de *hosts* en una red provada:

Clase	Comienza	Finaliza
Clase A	w.0.0.1	w.255.255.254
Clase B	w.x.0.1	w.x.255.254
Clase C	w.x.y.1	w.x.y.254

Sugerencias para asignar números de *hosts.*

No existen reglas de cómo asignar una dirección IP válida. Se pueden por ejemplo, numerar todos los *hosts* consecutivamente o se puede asignar un numero que pueda ser fácilmente identificado. Por ejemplo:

- Asignar los ID de los *hosts* en grupos basados en el tipo o en las características de su servidor.

MASCARA DE RED Y DIRECCION IP

Cada *host* en una red TCP/IP requiere una mascara de red (*subnet mask*). Vamos a ver el propósito de una mascara de red y como esta, forma parte del proceso que el IP usa para enviar paquetes.

Una mascara de red es una dirección de 32 bits usada para 'enmascarar' una parte de la dirección IP para distinguir el ID de red del ID de *host*. Esto es necesario para que el TCP/IP pueda determinar cuando una dirección IP pertenece a la red local o a una red remota.

Cada maquina en una red TCP/IP requiere una mascara de red, bien una mascara de red por defecto usada cuando una red no está dividida en subredes, o una mascara 'personalizada' cuando la red está dividida en segmentos.

Mascaras de red por defecto.

Una mascara de red por defecto se usa en las redes TCP/IP cuando estas no están divididas en subredes. Todos los *hosts* TCP/IP requieren esta mascara aunque estén en un solo segmento de red. La mascara por defecto que podemos utilizar, depende de la 'clase' de dirección.

En la mascara de red, todos los bits que corresponden a un ID de red están colocados a 1. El valor decimal de un octeto con todos unos, es 255. Todos los bits que corresponden al ID *host* estarán colocados a cero.

Clase	Bits usados por la mascara de red	Valor decimal
Clase A	11111111 00000000 00000000 00000000	255.0.0.0
Clase B	11111111 11111111 00000000 00000000	255.255.0.0
Clase C	11111111 11111111 11111111 00000000	255.255.255.0

Ejemplo en clase B

Dirección IP	123.298.32.200
Máscara	255.255.0.0
ID de Red	123.298.y.z
ID de *host*	w.x.32.200

Determinando el destino de un paquete.

Una suma binaria (**AND**) es el proceso interno que el IP utiliza para determinar cuando un paquete está destinado para un *host* local (en la propia red local) o remoto (en una red remota). Debido a que el **AND** es usado internamente por el IP, normalmente no necesitaremos realizar esta tarea.

Cuando se inicializa el TCP/IP, la dirección IP del *host* es sumada (AND) con la mascara de red. Antes de que un paquete sea enviado, la dirección IP del destino en sumada (AND) también con la misma mascara. Si el resultado de ambas sumas es idéntico, el IP sabe que debe enviarlo a la red local. Si este resultado no coincide el paquete será enviado a la dirección de un *router* o *gateway* por defecto (*default gateway).*

Para sumar una dirección IP con la mascara de red, el TCP/IP compara cada bit en la dirección IP con el correspondiente bit de la mascara de red. Si ambos bit están colocados a 1, el resultado es 1. En cualquier otro caso, el resultado es cero. Podemos verlo en la siguiente tabla:

Combinaciones de bit	Resultado
1 AND 1	1
1 AND 0	0
0 AND 1	0
0 AND 0	0

Como ejemplo:

Dirección de Red:	10010110	11010000	00001011	11100010
Máscara:	11111111	11111111	00000000	00000000
Resultado:	10010110	11010000	00000000	00000000

DIRECCIONES IP CON LA VERSIÓN 6.0

Bajo el actual direccionamiento de 32-bits implementado en la versión 4.0 (Ipv4),las identificaciones de red (ID de red) son escasas. Vamos a ver un poco cual es el futuro de las direcciones IP.

La actual cabecera de un paquete IP (visto anteriormente), no ha sifo modificado desde 1970. Este es el tributo que estamos pagando al diseño inicial. Por desgracia, el diseño inicial no esperaba el crecimiento de Internet y la posibilidad de que se gastasen todas las direcciones IP.

Sin embargo, una nueva versión del TCP/IP llamada Ipv6 ha sido desarrollada. Esta nueva versión, llamada la siguiente generación de IP'(*IP-The Next Generation)* **Ipng** incorpora las ideas de varios de los métodos propuestos para crear una nueva versión del protocolo IP.

Ipv6 ha sido creado para solucionar los problemas de direccionamiento en las redes actuales y nos da una amplia solución al ampliar completamente el espacio de direcciones IP. Ipv6 utiliza 16 octetos (frente a 4). Al escribirlo, está dividido en 8 pares de octetos separados por puntos y comas. Los octetos se representan en hexadecimal.

Ipv6 en una nueva estructura de paquetes que es incompatible con los sistemas Ipv4, pero que nos da muchos beneficios como un espacio de direcciones extendido, una cabecera simplificada, soporte para el tráfico dependiente del tiempo, y la posibilidad de añadir nuevas características.

El espacio de direcciones extendido es una de las principales características del Ipv6. Ipv6 tiene 128 bits como direcciones origen y direcciones destino (cuatro veces mayor que Ipv4). 128 bits pueden expresar cantidades del orden de 3 * 10 elevado a 38 direcciones. En Ipv6, una dirección puede ser del tipo:

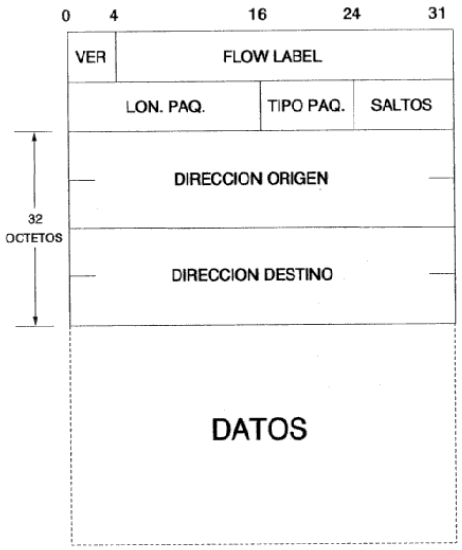
4b3e:23ed:f234:452a:aec4:32e2:78ea:ff34

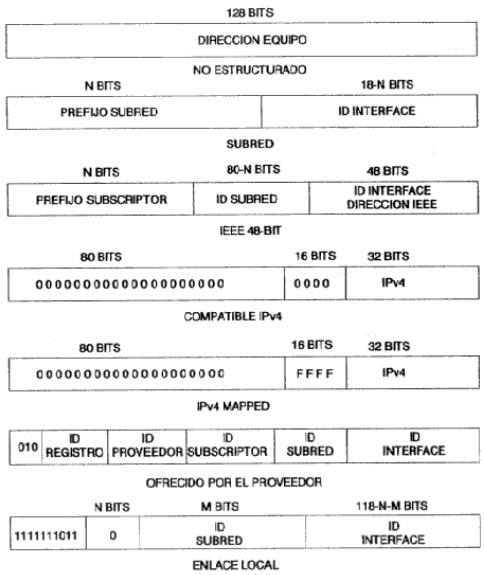
Las cabeceras IP están diseñadas para contener unicamente un minimo de datos, moviendo los campos no esenciales y los campos de opciones a las extensiones de la cabecera que están situadas a continuación de la propia cabecera. Cualquier cosa no incluida en la base de la cabecera Ipv6 puede ser añadido en las extensiones de ella.

Un nuevo campo en la cabecera Ipv6 permite la permite la preasignación de los recursos de la red a lo largo del camino, como son los servicios urgentes o dependientes del tiempo, como la voz y el video y garantizan un ancho de banda solicitado con unos retrasos prefijados máximos (indispensable para la transmisión de sonido e imagen).

Existe la posibilidad de encapsulamiento del IPv4 para solventar temas de incompatibilidades.

CABECERO Y DIRECCIONAMIENTO IPv6





SUB-REDES

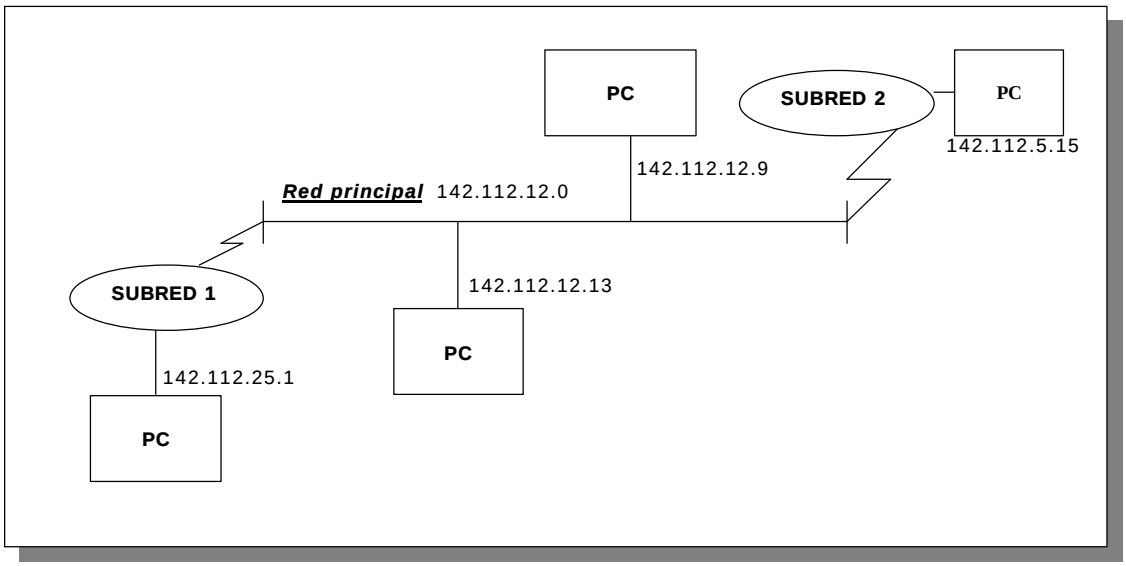
Vamos a ver como podemos asignar direcciones IP a múltiples redes TCP/IP con un simple identificativo de numero de red (un único ID).

Veremos conceptos fundamentales y procedimientos para implementar las subredes y supraredes. Igualmente veremos cuando es necesario hacer una subred, y como y cuándo usar una submascara de red por defecto, como definir una submascara personalizada y como crear un rango de direcciones IP para cada subred.

INTRODUCCIÓN A LAS SUBREDES

Una subred es un segmento físico del entorno TCP/IP que utiliza una direccion IP derivada de un único **ID** de red. Recordemos que una empresa o una organización tiene un **ID** de red que le es asignado por el comité ***InterNIC***.

Dividiendo la red en sub-redes, requiere que cada segmento use un diferente **ID** de red, o en un diferente **ID** de subred.



Como vemos en el ejemplo anterior un único ID de subred está utilizado para cada segmento simplemente haciendo que el ID de red forme parte de ID de subred. Una parte la utilizamos para identificar el segmento como una única red, y la otra parte es la usada para identificar los PCs (*hosts*). Esto es lo que llamamos subredes. No es necesario utilizarlo en una red privada, pero también es conveniente en ella por labores administrativas y de mantenimiento.

Existe beneficios claros al hacer subredes:

- Mezclar diferentes topologías de red, como por ejemplo *Ethernet* y *Token Ring.*
- Superar limitaciones de las actuales tecnologías, como exceder el máximo numero de *hosts* por segmento.
- Reducir la congestión de red redireccionando el trafico y reduciendo el *broadcasting.*

Nota: El tema de subredes está definido en la RFC 950.

Implementando las subredes.

Antes de implementar las subredes, debemos determinar las necesidades actuales y planear los requerimientos futuros. Esta pequeña guía puede orientarnos:

1) Determinar el número de segmentos físicos en nuestra red.

2) Determinar el número de direcciones *hosts* en cada segmento físico de la red. Cada *host* TCP/IP requiere al menos una dirección IP.

3) Basado en nuestras necesidades, definir:

   - Una mascara de red para TODA la red
   - Una única ID de subred para segmento físico.
   - Un rango de ID de *host* para cada subred.

Mascaras de *bits* en las subredes.

Antes de definir una mascara de subred, debemos determinar el número de segmentos y *host* por segmento que vamos a necesitar en el futuro.

Cuantos mas bits utilicemos en las mascara de subred, más subredes estarán disponibles. Por ejemplo, los siguientes ejemplos en clase B muestran la correlación entre el número de bits y el número de subredes y *hosts.*

3 bits = 6 subredes = 8000 *hosts* por subred (aproximadamente)
8 bits = 254 subredes = 254 *hosts* por subred.

Usando mas bits de los necesarios, nos permitirá aumentar el número de subredes pero nos va a limitar la cantidad de *hosts* en cada subred. Si se utilizan los bits necesarios para las subredes actuales, nos permitirá aumentar el numero de *hosts* pero estaremos limitados al número de subredes definidas inicialmente.

DEFINIENDO UNA MASCARA DE SUBRED

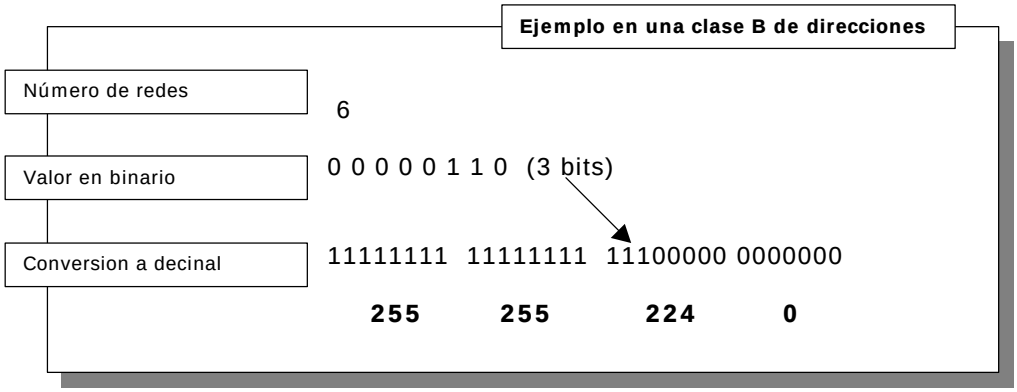
El definir una mascara de subred es un proceso de tres pasos. Vamos a ver esos tres pasos y a realizar unos ejemplos para definir las subredes.

Definir una mascara de subred es necesario si estamos dividiendo nuestra red en subredes. Vamos a seguir para ello los tres pasos siguientes:

1) Una vez que hayamos determinado el número de segmentos en nuestra red, convertimos dicho número a formato binario.

2) Contamos el número de bits necesarios para representar el número de segmentos físicos en binario. Por ejemplo, si necesitásemos 6 subredes, el valor binario es 1 1 0. Para representar 6 en binario, requerimos **tres** bits.

3) Convertir ese número de bits a formato decimal de izquierda a derecha. Por ejemplo si son necesarios 3 bits, utilizaremos los tres primeros bits del ID de *host* como el ID de subred. Es decir: 11100000. Su valor decimal (podemos utilzar para las conversiones la calculadora de Windows) es 224. La mascara de subred es por tanto: 255.255.224.0 en nuestro ejemplo de clase B.



Mascara de bits contiguos.

Debido a que las subredes quedan definidas por la mascara de la subred, el administrador no está obligado a seleccionar los bits de orden mas alto para la mascara de la subred. Cuando el tema de subredes fue inicialmente definido en la RFC 950, se recomendaba que se utilizasen los bits de orden mas alto como identificación de la subred. Hoy día, sin embargo, algunos vendedores de *routers* soportan el uso de los bits de orden más bajo o incluso sin ordenar en las identificaciones (IDs) de la subred.

Tablas de Conversión

La siguiente tabla lista mascaras de subred ya convertidas a decimal usando un octeto para las redes de clase A.

Número de subredes	número de bits	mascara subred	Nºhosts por subred
0	1	Inválido	Inválido
2	2	255.192.0.0	4.194.302
6	3	255.224.0.0	2.097.150
14	4	255.240.0.0	1.048.574
30	5	255.248.0.0	524.286
62	6	255.252.0.0	262.142
126	7	255.254.0.0	131.070
254	8	255.255.0.0	65.534

La siguiente tabla lista mascaras de subred ya convertidas a decimal usando un octeto para las redes de clase B.

Número de subredes	número de bits	mascara subred	Nºhosts por subred
0	1	Inválido	Inválido
2	2	255.255.192.0	16.382
6	3	255.255.224.0	8.190
14	4	255.255.240.0	4.094
30	5	255.255.248.0	2.046
62	6	255.255.252.0	1.022
126	7	255.255.254.0	510
254	8	255.255.255.0	254

La siguiente tabla lista mascaras de subred ya convertidas a decimal usando un octeto para las redes de clase C.

Número de subredes	número de bits	mascara subred	Nºhosts por subred
0	1	Inválido	Inválido
2	2	255.255.255.192	62
6	3	255.255.255.224	30
14	4	255.255.255.240	14
30	5	255.255.255.248	6
62	6	255.255.255.252	2
126	7	Inválido	Inválido
254	8	Inválido	Inválido

Subredes utilizando más de un octeto.

Hasta el momento, hemos trabajado con un octeto para definir la mascara de subred. En algún momento, puede ser necesario (y ventajoso), realizar la descomposición en subredes, utilizando más de un octeto. Esto puede permitirnos una mayor flexibilidad en el rango de direcciones.

Por ejemplo, supongamos que estamos configurando una Intranet para una gran compañía. Esta empresa, planea conectar internamente con sus distribuidores en toda Europa, América y Asia. En total unos 25 puntos geográficos con cerca de 1000 subredes y con una media de 750 *hosts* por subred.

Esto puede ser posible utilizando varias clases B como ID de red. Los requerimientos de los *hosts* en una clase B, necesitamos una mascara de subred de 255.255.252.0 (ver tablas anteriores). Añadiendo los requerimientos de las subredes, necesitaríamos en total 16 direcciones de clase B.

Hay una vía más fácil. Debido a que los ordenadores que estamos utilizando están en una Intranet, podemos utilizar una red privada. Por tanto si decidimos en este caso asignar una clase A del tipo de red privada 10.0.0.0 podríamos planear igualmente el crecimiento de las necesidades de la empresa. Obviamente, realizando una subred únicamente con el segundo octeto no satisfaría nuestro requerimientos del orden de unas 1000 subredes. Si utilizásemos el segundo octeto y parte del tercer octeto podríamos satisfacer todas nuestras necesidades.

ID de red	Mascara de subred	(en binario)
10.0.0.0	255.255.248.0	11111111 11111111 11111000 00000000

Usando 13 bits para una mascara de subred en clase A, podríamos utilizar 8.190 subredes cada una de ellas de 2.046 *hosts.*

DEFINIENDO IDs DE SUBRED

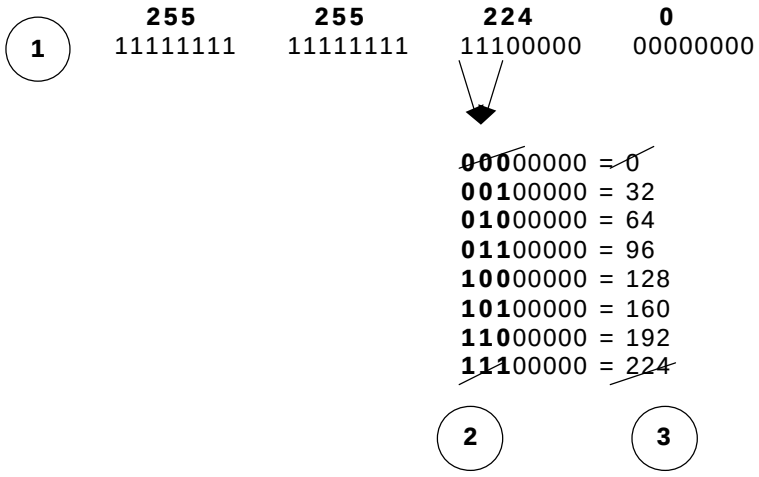
Los identificadores (IDs) de subred se definen usando el mismo número de bits que se usan para definir la mascar de subred. Existen dos métodos diferentes para definir un rango de IDs de subred para una red de trabajo en Internet.

Podemos definir el ID de subred usando el mismo numero de bits que hemos utilizado para la mascara de subred. Estas posibles combinaciones de bits vamos a evaluarlas y convertirlas a formato decimal. Los siguientes pasos muestran como definir un rango de IDs de subred para una red de trabajo en Internet.

1) Usando el mismo número de bits que son usados para el caclulo de la mascara de subred, listamos todas las posibles combinaciones.

2) Eliminamos todos los valores que su contenido son todos ceros o unos. Todos los ceros o unos son direcciones IP inválidas, debido a que todo ceros, indica "esta red unicamente" y todos a unos, coincide con la mascara de subred.

3) Convertir a decimal los valores para cada subred. Cada valor decimal representa una única subred. Este valor será usado para definir el rango de *hosts* para esa subred.



Un caso especial de direcciones de Subred

Los IDs de subred con todo a ceros o todo a unos son llamados los "casos especiales de direcciones de subred". Una subred con todo a unos indica una subred de *broadcast* (radiodifusión), y una subred con todo a ceros indica "esta subred". Al construir las subredes, no está recomendado utilizar estas direcciones. Sin embargo es posible usar estas direcciones especiales si estan soportadas por todos los *routers* y hardware de nuestra red. La RFC 950 discute las limitaciones impuestas cuando usamos estas direcciones especiales.

DEFINIENDO IDs DE *HOSTS* EN UNA SUBRED

Podemos seguir un pequeño procedimiento para determinar el número de *hosts* por subred. De hecho, si hemos definido los IDs de subred, entonces hemos definido ya los IDs de los *hosts* de cada subred.

El resultado de cada valor incremental que hemos visto anteriormente, indica el comienzo de un rango de IDs de *host*. Sigamos con el ejemplo:

IDs de subred	Rango de IDs de *host*		
00000000 = 0	Inválida		
00100000 = 32	x.y.**32.1**	-	x.y.**63.254**
01000000 = 64	x.y.**64.1**	-	x.y.**95.254**
01100000 = 96	x.y.**96.1**	-	x.y.**127.254**
10000000 = 128	x.y.**128.1**	-	x.y.**159.254**
10100000 = 160	x.y.**160.1**	-	x.y.**191.254**
11000000 = 192	x.y.**192.1**	-	x.y.**223.254**
11100000 = 224	Inválida		

Como determinar el número de *hosts* por subred.

1) Calcular el número de bits disponibles para la ID del *host*. Por ejemplo, si estamos en una direccion de clase B, que usa 16 bits para la ID de red y 2 bits para la ID de subred, nos quedan 14 bits para el ID de *host*.

2) Convertir el valor binario de los bits del ID de *host* a decimal. Por ejemplo 11111111111111 en binario (14 bits) es 16.383 en formato decimal.

3) Restarle 1.

IMPLEMENTANDO *ROUTING* DE IP

Routing (encaminar) es el proceso de escoger el camino bajo el cual van a ser enviados los paquetes. El *routing* sucede cuando enviamos los paquetes a través de un *router* debido a que el *host* destino no está en nuestra red. Un *router* es una maquina o un dispositivo que reenvía los paquetes desde una red física a otra. A los *routers* muchas veces se les llama *gateways*.

Recordemos, de los primeros capítulos, la secuencia por la que el TCP/IP envía los paquetes:

1) Se hace una suma lógica (AND) entre la dirección IP origen (nuestra maquina) y la mascara de IP.

2) Se hace una suma lógica (AND) entre la dirección IP destino y la mascara de IP.

3) Si coinciden, pertenece a nuestra red, por lo que se localizará la dirección física del destino, primero mirando en la caché ARP de nuestra maquina, y si no existe, se localizará la maquina destino mediante *broadcasting* de ARP. Una vez localizada se enviará el paquete IP al destino y se guardará la dirección física de ese destino en la caché ARP.

4) Si no coinciden, el paquete se envía al *router* o *gateway* por defecto que tengamos definido en nuestro *hosts*.

Pero, antes de enviar el paquete, se debe tomar la decisión de por donde hay que enviarlo. Esta decisión deben tomarla todos los *hosts* bien sea nuestro propio *host* o cualquier *router* por los que el paquete atraviese. Para tomar la decisión de enrutamiento, la capa IP consulta una tabla de rutas que está almacenada en memoria. Una tabla de rutas, contiene entradas que relacionan la dirección IP buscada y la *interface* a utilizar.

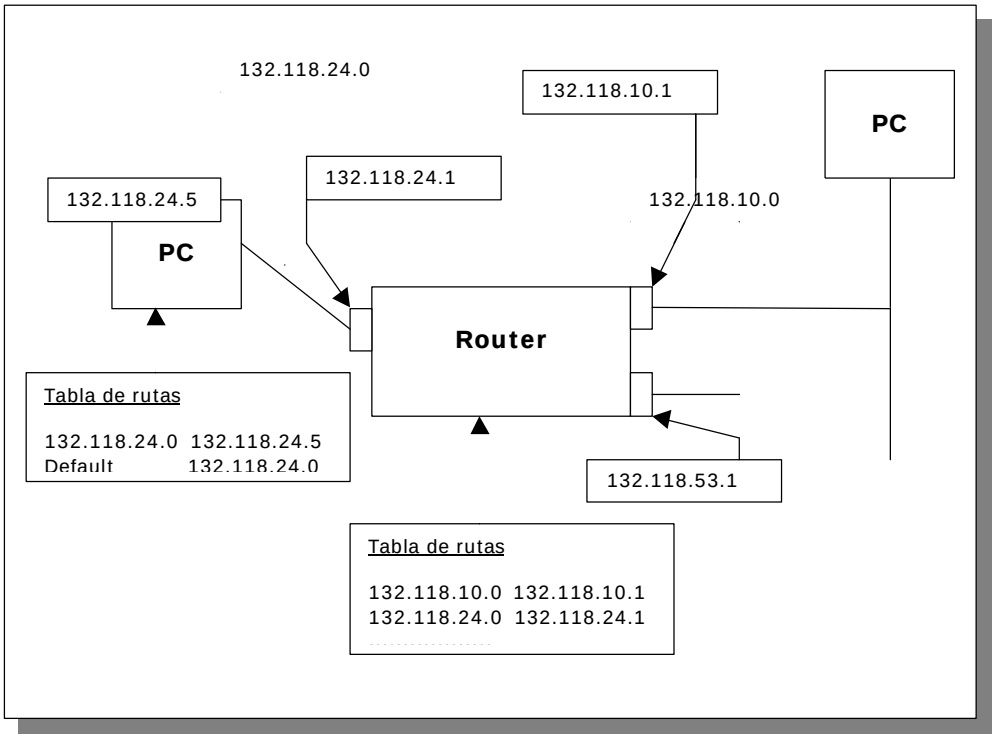
Pensemos que nuestra maquina, puede tener más de un adaptador de red. Este es el caso de los *routers* e incluso el caso de un PC domestico, con tarjeta de red y con módem (el cual es un adaptador más). Antes de enviar el paquete a la red, se debe tomar la decisión de þor donde 'enviarlo.

1) Cuando un *host* espera comunicar con otro *host*, el IP determina primero si el destino está en la red local o en otra red.

2) Si el destino es un *host* remoto (está en otra red), el IP busca en la tabla de rutas una posible ruta para localizar el *host* destino en la red remota.

3) Si no hay una ruta explicita, IP utiliza el *gateway* por defecto para enviar el paquete al *router*.

4) En el *router* otra vez, es consultada su tabla de rutas, para seguir buscando un camino del *host* remoto o de la red. Si no existe un camino explicito, el *router* reenviará otra vez el paquete a su propio *gateway* por defecto para que continuar la cadena y que sea este siguiente *router* el encargado de repetir el ciclo.



Según vamos encintrando cada *router* el paquete se envía al siguiente *router*. Esto sel le llama un "salto". Finalmente el paquete es entregado en el *host* destino. Si alguna ruta no se encuentra se envía un mensaje de error al *hosts* origen.

Detección de un *Gateway* muerto. *(dead gateway)*

El TCP/IP puede detectar el fallo del *gateway* por defecto e intentar hacer los necesarios ajustes en las tablas de rutas para intentar utilizar otro *gateway* por defecto. El TCP/IP envía un paquete al *gateway* por defecto hasta que recibe un ACK (*acknowledgment*). Si el tiempo medio del parámetro de configuración del TCP/IP *TcpMaxDataRetransmissions* se excede y existen varios *gateways* comfigurados en ese ordenador, el TCP/IP solicita que el IP cambie al siguiente *gateway* por defecto.

Cuando configuramos un ordenador que está ejecutando Windows con las direcciones IP de multiples *gateways* la detección de la muerte de uno de ellos está colocada en **'on'** (si).

Nota: la detección de un *'dead gateway'*, los reintentos del TCP y el metodo de reselección está descrito en la RFC 816.

Encaminamiento (*routing)* de IP. Estático *versus* Dinámico.

Como los *routers* obtienen información depende de si los *routers* permiten encaminamiento de IP estático o dinámico.

Los *routers* estáticos necesitan que las tablas de rutas sean construidas y actualizadas manualmente. Si una ruta cambia, los *routers* estáticos no informan a nadie de esta cambio, es decir los *routers* estáticos con intercambian información con los *routers* dinámicos.

El encaminamiento dinámico es una función de los protocolos de *routing,* como por ejemplo el *Routing Informatiuon Protocol* (RIP) y el *Open Shortest Parh First* (OSPF). Los protocolos de *routing* periódicamente intercambian rutas a redes conocidas a lo largo de los *routers* dinámicos. Si una ruta cambia, todos los *routers* dinámicos son informados de dicho cambio.

Windows NT Server y Windows 2000 Server pueden funcionar como un *router* estático o dinámico. Un ordenador ejecutando Windows NT o Windows 2000 puede ser configurado con multiples adaptadores de red y rutas entre ellos.. Este tipo de sistema, que es ideal para pequeñas Intranet, se le llama *'multihomed computer'*.

Windows NT o Windows 2000, nos da la capacidad de funcionar como un *router* RIP que soporta manejo dinámico de las tablas de rutas de IP. El protocolo RIP elimina la necesidad de establecer tablas de rutas estáticas.

Nota: Microsoft da soporte a los protocolos *inter-routing* en Windows Server. El RIP está definido en la RFC 1723.

Resumen

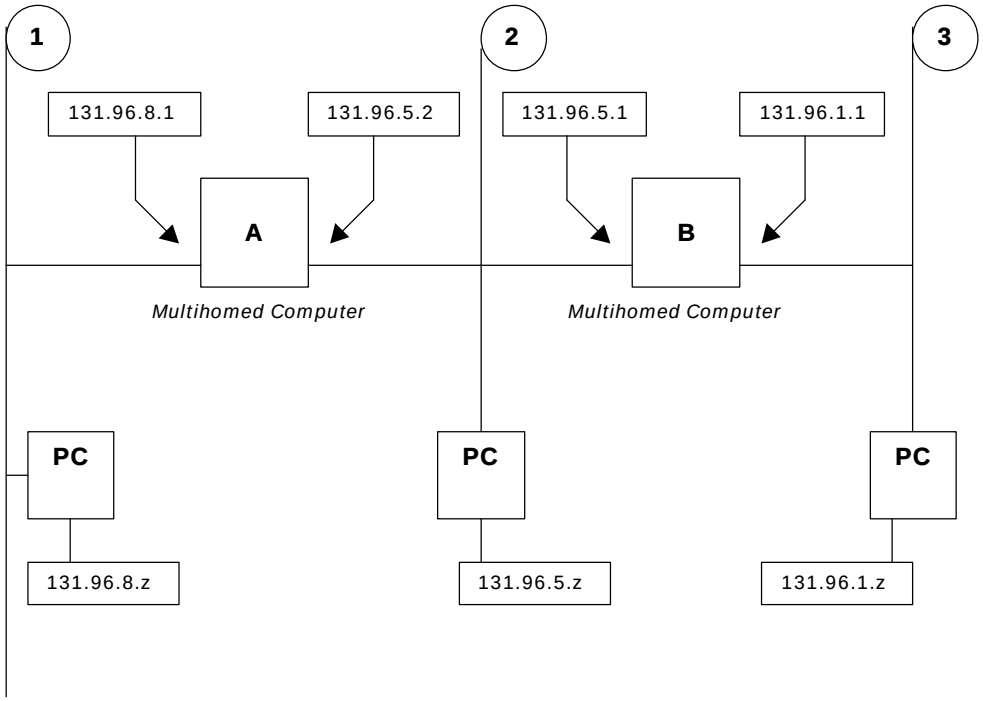
Los *routers* reenvían paquetes desde una red física a otra. La capa IP consulta la tabla de rutas que está almacenada en memoria. Una tabla de rutas contiene entradas con las direcciones IP de las *interfaces* a otras redes. Los *routers* estáticos requieren que las tablas de rutas sean hechas y actualizadas manualmente. Cuando existe encaminamiento dinámico, si una ruta cambia, el resto de *routers* son informados de dichos cambios.

- Podemos ver la tabla de rutas de nuestro ordenador ejecutando el comando *route print.*
- Dicho comando también nos permitirá modificarla, añadiendo entradas o cambiándolas.

ENRUTAMIENTO ESTATICO DE IP

Para enviar paquetes IP a otras redes, debemos configurar cada uno de los *routers* estáticos de nuestra red. Debemos entrar en la configuración de cada *router* y modificar la tabla de rutas para cada red o subred de nuestra red total de trabajo.

Veamos el siguiente ejemplo:

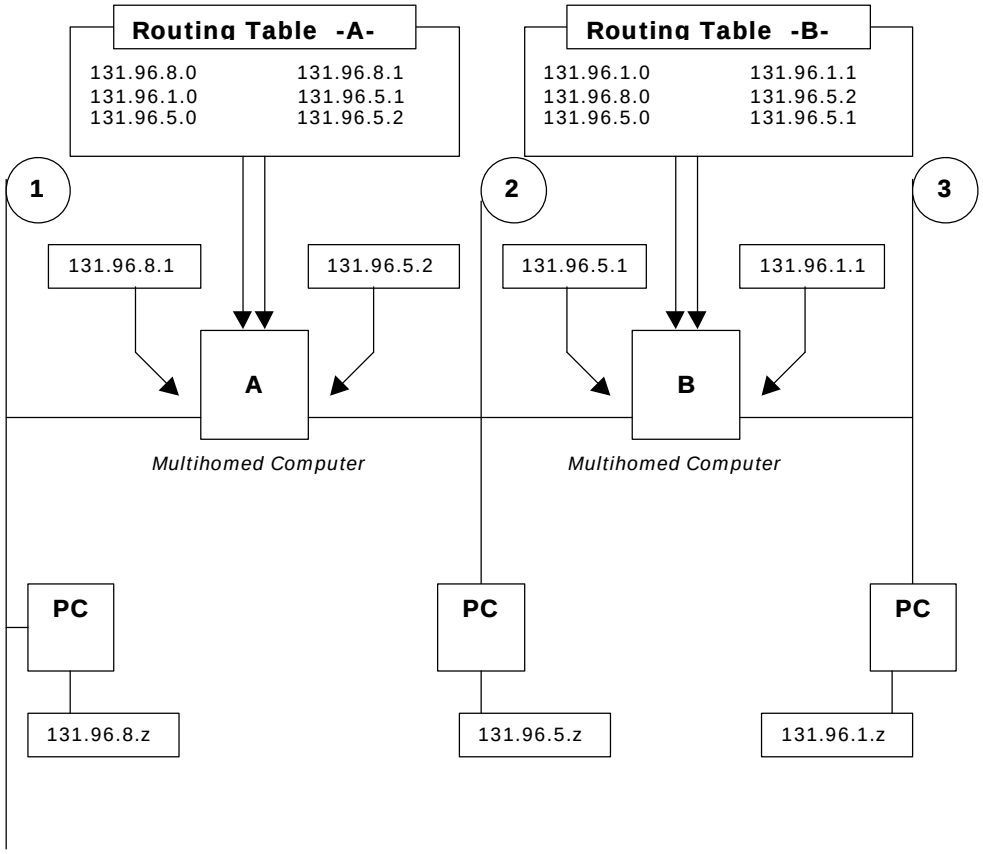


- El ordenador **A** tiene únicamente una conexión local a las redes **1** y **2**. De esta manera los *hosts* de la red **1** pueden comunicarse con los *hosts* de la red **2**, pero no pueden comunicarse con los *hosts* de la red **3**.
- El ordenador **B** puede únicamente conectar las redes **2** y **3**. Los *hosts* de la red **3** pueden comunicar con los *hosts* de la red **2**, pero no pueden comunicar con los *hosts* en la red **1**.

Configurando los *routers* estáticos.

En una red de trabajo con al menos un *router* estático, necesitamos configurar la entrada de la tabla de rutas (*routing table*) de cada *router* para mostrarle 'todas las redes conocidas.

Vamos a ceñirnos al ejemplo anterior y veamos como debemos configurar cada uno de los *routers* **A** y **B**.



- Creamos una entrada en la tabla estática de rutas en el ordenador **A**. La entrada contiene la identificación de red (*network ID*) de la red **3** y la dirección IP (131.96.5.1) de la *interface* que el ordenador **A** necesita para enviar paquetes (*route)* desde la red **1** a la red **3**.

- Creamos una entrada en la tabla estática de rutas en el ordenador **B**. La entrada contiene la identificación de red (*network ID)* de la red **1**. Estea entrada también contiene la direccion IP (131.96.5.2) de la *interface* que el ordenador **B** necesita para enviar paquetes desde la red **3** a la red **1**.

Si nuestra red tuviese más de dos *routers*, y al menos uno de ellos es un router estático, necesitaremos configurar la tabla de rutas de cada uno de los *'multihomed computers'.*

Para que un *host* pueda comunicar con otros *hosts* en una red de trabajo, la dirección de su *gateway* por defecto debe estar configurada con la dirección IP de la *interface* del *router* local.

Usando la dirección del *Gateway* por defecto.

Uno de los métodos de configurar un *router* estático sin añadir manualmente rutas a la tabla de rutas, es configurar cada *'multihomed computer'* la dirección del *gateway* por defecto como la *interface* local de otro *'multihomed computer'* en la red común. Este método solo trabaja correctamente con dos *routers* estáticos.

Construyendo una tabla de rutas.

Podemos añadir información a la tabla de rutas, utilizando el comando **route**. El comando **route print** se puede utilizar para ver las entradas por defecto en las tablas de rutas. Una entrada estática debe añadirse a los *routers* estáticos de todas las redes en los cuales no esté configurada una nueva *interface*. Una entrada estática incluye a lo siguiente:

- Dirección de red. El ID de red o el nombre de red de la red de destino. Si un nombre de red es usado para definir el destino, este debe encontrarse en el fichero Networks'. (veremos estos temas de resolución de nombres en capítulos posteriores).

- Mascara de red. La mascara de subred para esa dirección de red.

- Dirección del *Gateway*. La dirección IP o el nombre del *host* de la *interface* de destino de red. Si utilizamos un nombre para esta *gateway*, debe ser encontrado en le fichero Hosts'. (veremos estos temas de resolución de nombres en capítulos posteriores).

i especificamos un nombre de red o un nombre de *host* en la tabla de rutas, el nombre debe estar configurado en los ficheros al respecto. Ambos ficheros están en el directorio *systemroot*\System32\Drivers\Etc en Windows NT o Windows 2000 y en el directorio de Windows en Windows 95 / Win98.

Tanto Windows 95 como Windows 98 no pueden hacer *routing* por defecto. Y con las herramientas estándar de Microsoft no pueden ser configurados para realizar *routing.* Existe software de terceros fabricantes que sí lo soportan.

Entradas por defecto en la tabla de rutas.

La taba de rutas que mantiene Windows con las entradas por defecto lo podemos ver en la siguiente tabla.

Dirección	Descripción
0.0.0.0	La dirección usada como ruta por defecto para cualquier no especificada en la tabla de rutas.
Subnet Broadcast	La dirección usada para *broadcasting* en la subred local.
Network Broadcast	La dirección usada para *broadcasting* a la red.
Local loopback	La dirección usada para pruebas de configuración de IP y conexiones.
Local network	La dirección usada para enviar paquetes a los *hosts* en a red local.
Local host	La dirección del ordenador local (del propio ordenador). Esta dirección referencia a la dirección de *loopback*.

Añadiendo entradas estáticas.

Podemos utilizar el comando **route** para añadir entradas a la tabla de rutas.

Para añadir o modificar ruta.	Función
route add [red] **mask** [mascara] [gateway]	Añade una ruta
route –p add [red] **mask** [mascara] [gateway]	Añade una ruta persistente
route delete [red] [gateway]	Borra una ruta.
route change [red] [gateway]	Modifica una ruta
route print	Muestra la tabla de rutas
route –f	Borra todas las rutas.

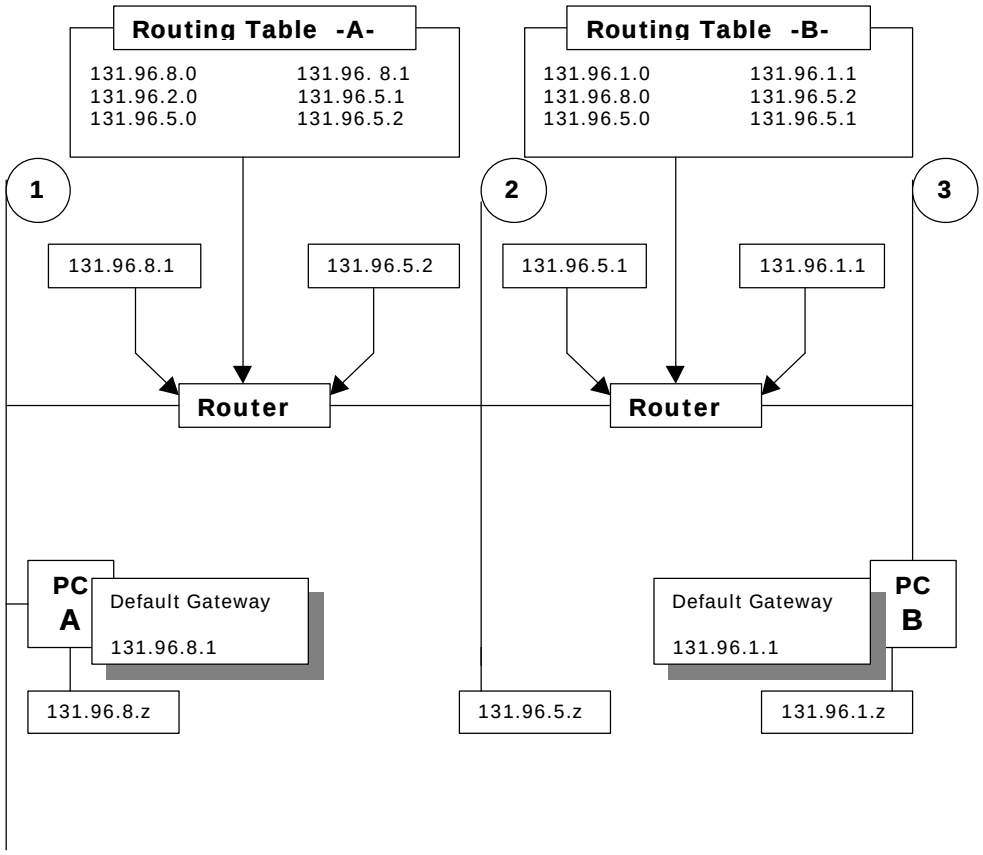
Nota: Las tablas de rutas estáticas se almacenan únicamente en memoria a no ser que especifiquemos el parámetro **–p**. Las rutas persistentes se almacenan en el registro de Windows. Si apagásemos el ordenador necesitaremos volver a crear todas aquellas rutas que no hayan sido definidas como persistentes.

ENCAMINAMIENTO DINAMICO DE IP

Con encaminamiento dinámico, los *routers* automáticamente intercambian los caminos conocidos para ir de una a otra red. Si el camino cambia, los protocolos de *routing* automáticamente actualizan las tablas de rutas e informan a los otros *routers* de estos cambios. En las grandes redes (y en Internet), las tablas de rutas dinámicas, juegan un papel importante en las comunicaciones de la red.

El *routing* dinámico se implementa típicamente en las grandes redes debido a que necesita una mínima configuración y mantenimiento por los administradores de la red. El *routing* dinámico requiere un protocolo de *routing* como RIP u OSPF.

Para que un *hosts* se comunique con otros *hosts* en una red, la dirección del *default gateway* debe contener la dirección IP del *router*. No es necesaria otro tipo de configuración.



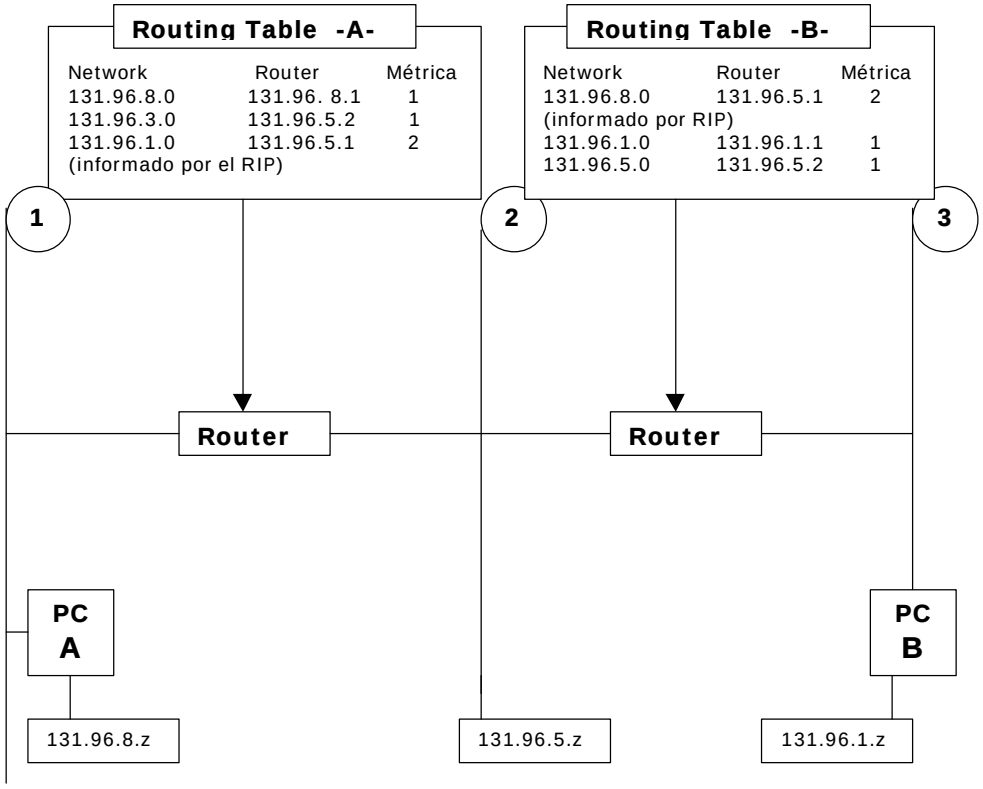
Tal y como vemos en la figura anterior el ordenador **A** necesita un *gateway* por defecto con la dirección 131.96.8.1 (la dirección IP de la tarjeta de red del *router* que está conectada a nuestra subred). Igualmente, el ordenador **B** tiene configurado su *gateway* por defecto con 131.96.1.1 la cual es la direccion IP de la tarjeta de red del *router* que está conectado a nuestra subred.

RIP

El protocolo RIP (***R**outing **I**nformation **P**rotocol*) para el IP facilita el intercambio de información de encaminamiento en una red IP. Todos los mensajes RIP se envían bajo el puerto 520 de UPD.

RIP permite a los *routers* intercambiar las informaciones de las direcciones IP de las redes (las direcciones 'alcanzables' por el *router*), y la 'distancia' de estas redes. RIP utiliza un campo de contador de saltos, o también llamado 'métrica', en la tabla de rutas para indicar la distancia a la identificación (ID) de red. El contador de saltos, es el número de *routers* que deben ser cruzados para alcanzar la red de destino. El máximo numero de saltos para un contador RIP, es 15. Las redes que necesiten 16 o más saltos son consideradas 'inalcanzables'. El contador de saltos puede ajustarse para indicar redes lentas o congestionadas. Si existen varias entraas para una identificación de red en la tabla de *routing*, el *router* seleccionará la ruta con el número más bajo de saltos (la métrica mas pequeña).

Nota: Un *router* que recibe mensajes de *broadcast* RIP pero no envía ningún mensaje RIP se le llama *Silent router RIP* (un encaminador RIP silencioso).



La siguiente red muestra tres subredes conectadas con dos *routers* (que perfectamente pudieran ser dos ordenadores ejecutando NT server) con el protocolo RIP de encaminamiento activado. Cada *router* está configurado con el intervalo de actualización por defecto: es decir 30 segundos. El *router* A envía *broadcast* a la red 2, y todos los *routers* RIP activos en la red 2 informan de esto a la red 1. Al recibir el propio *broadcast* el *router* B, además de responder, actualiza su propia tabla de rutas. Los propios *routers* actualizan también la tabla de saltos si encontrasen un ruta más corta.

Problemas que pueden existir con RIP

A pesar de ser sencillo y estar muy bien soportado por la industria, el intercambio RIP sufre algunos problemas inherentes al diseño de las originales redes LAN (*Local Area Network* o Red de Área Local). Estos problemas, hacen que el RIP sea una buena solución únicamente en las pequeñas redes con un pequeño número de *routers*.

Con RIP, la tabla de rutas de cada *router* tiene una lista completa de todas las identificaciones de red (*network ID)* y todas las posibles vías para alcanzar dichas identificaciones. La tabla de rutas puede tener cientos o incluso miles de entradas en una red IP con multiples caminos. A causa de que el tamaño de un unico paquete RIP es 512 bytes, con tablas de rutas largas se deben enviar por tanto múltiples paquetes RIP.

Los *routers* RIP envían el contenido de sus tablas de rutas a la red cada 30 segundos. Las redes con multiples *routers* y caminos saturan la red con mensajes RIP para intentar intercambiar sus tablas. Esto puede ser especialmente problemático en redes WAN (*Wide Area Network)*, ya que un elevado ancho de banda de la red, se utilizará únicamente para el intercambio de mensajes RIP.

Cada entrada 'automática' en la tabla de rutas (es decir 'aprendida' y añadida automáticamente por el *router*) tiene una vida de 3 minutos después de haber sido recibida en un mensaje RIP. Cuando existe la caída de un *router*, pueden pasar varios minutos hasta que los cambios se propagan en la red. Esto es conocido como *slow convergence problem* o el problema de la lenta convergencia.

INTEGRANDO *ROUTING* DINAMICO Y ESTATICO

Un *router* estático no intercambia información de sus tablas de rutas con los *routers* dinámicos. Para encaminar desde un *router* estático a uno dinámico, necesitamos añadir una ruta estática en ambos *routers*.

Nota: Algunas implementaciones del RIP no propagan las tablas de rutas estáticas. En este caso, es necesario configurar estáticamente todos los *routers* de la red.

IMPLEMENTANDO WINDOWS NT COMO ROUTER

El *routing* estático puede trabajar bien para redes pequeñas, pero en un gran red, la sobrecarga de trabajo de mantener manualmente las tablas de rutas es muy elevada. Vamos a ver y a intentar comprender lo que se requiere para implementar un Windows NT como *router.*

Activando RIP, Windows NT Server puede funcionar como un *router* dinámico. El RIP en Windows NT elimina la necesidad de configurar manualmente las tablas de rutas. Es una solución aceptable y deseable en redes de tamaño medio, pero no es aconsejable para grandes redes a causa de la cantidad de trafico de *broadcast* que genera.

Para Implementar un *router* Windows NT.

1) Instalar varias tarjetas adaptadoras y sus apropiados controladores, o configurar múltiples direcciones IP en un único adaptador de red.

2) Configurar la(s) tarjeta(s) adaptadora(s) de red con direcciones IP válidas y sus mascaras de red.

3) En la pestaña de 'propiedades' del cuadro de dialogo **Microsoft TCP/IP properties** 'seleccionar: **Enable IP Fordwarding**.

4) Dependiendo de que versión de Windows NT se está ejecutando:

   - En el **Panel de Control->Network**, en la pestaña **Services** añadir el protocolo RIP.
   - O bien, añadir rutas estáticas a la tabla de rutas para todas las redes conocidas a las que el ordenador no tiene acceso directo mediante su *interface* de red, para que se comporte como un *router* estático.

LA UTILIDAD 'TRACERT'

La utilidad TRACERT verifica la ruta que un paquete debe realizar para alcanzar un destino. Puede ser exitosa para determinar si está fallando un router. Si el comando no es exitoso podemos determinar la ruta fallada y posiblemente nos indique el *router* o el enlace WAN con problemas.

TRACERT es también una buena herramienta para determinar los *routers* lentos. El tiempo de respuesta lo vemos en la salida de este comando. Esta información puede ser comparda para otra ruta con el mismo destino.

Resumen

Windows NT Server puede funcionar como un *router* dinámico de IP activando el protocolo RIP. Esto elimina la configuración manual de las tablas de rutas. La utilidad TRACERT es una buen herramienta para determinar si un *router* está fallando o bien si hay *routers* lentos en nuestra red.

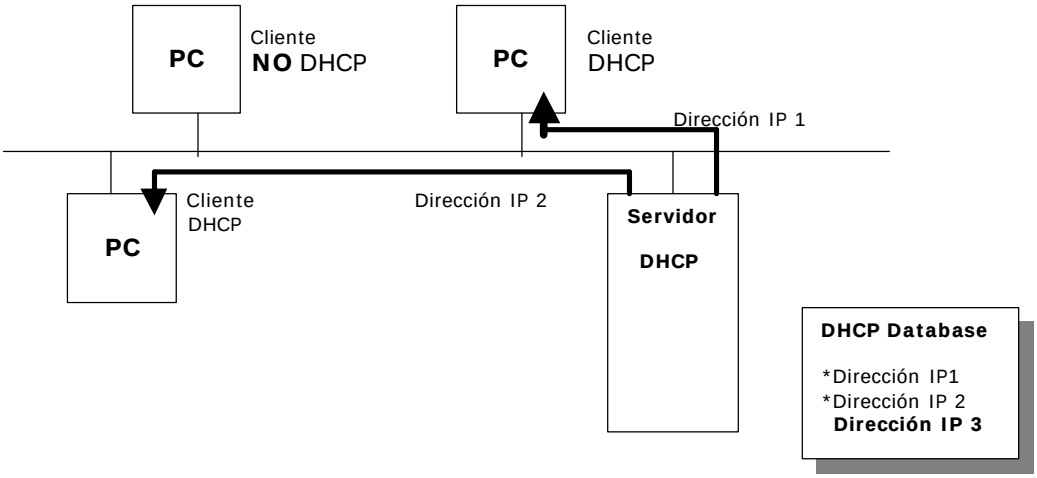
DHCP – DYNAMIC HOST CONFIGURATION PROTOCOL

ACERCA DEL DHCP

EL DHCP (*Dynamic Host Configuration Protocol*) asigna automáticamente direcciones IP a los ordenadores. El DHCP nos facilita el no tener que configurar manualmente cada *host* TCP/IP.

DHCP es un extensión del protocolo BOOTP. DHCP centraliza y maneja la asignación de configuraciones TCP/IP asignando automáticamente una dirección IP a los ordenadores configurados para usar DHCP. Implementando el DHCP eliminamos los posibles problemas asociados con la configuración manual como por ejemplo, la posibilidad de asignar IPs repetidas a las maquinas de nuestra red.

Como vemos en el siguiente gráfico, cada vez que un cliente DHCP arranca, pide una dirección IP a un servidor DHCP. Incluyendo la dirección IP, la mascara de red y otros valores opcionales. Los valores opcionales, pueden incluir un *default gateway,* un dirección DNS (*Domain Name Server)* y un servidor de direcciones NetBIOS.



Cuando el servidor DHCP recibe una petición, selecciona una dirección IP desde un *pool* de direcciones definido en su base de datos y se la ofrece al cliente DHCP. Si el cliente acepta esta oferta, la dirección IP es 'prestada' al cliente durante un periodo específico de tiempo. Si no hay direccion IP disponibles en el servidor, el cliente no podrá iniciar el TCP/IP.

Nota: Windows NT 4 con el SP2 está ya preparado para soportar la peticiones BOOTP de los clientes.

El protocolo BOOTP está definido en la RFC 1532. DHCP está definido en la RFC 1533, 1534, 1541 y 1542.

CONFIGURACIÓN MANUAL *versus* AUTOMATICA

Para comprender porqué el beneficioso el DHCP para configurar a los clientes TCP/IP, vamos a contrastar el método manual de configuración del TCP/IP con el método automático usando DHCP.

Configurando TCP/IP manualmente.

La configuración manual de TCP/IP permite que los usuarios puedan fácilmente seleccionar una dirección IP aleatoria en lugar de una dirección IP válida. El uso de direcciones incorrectas puede causar problemas en la red y puede ser muy dificultoso el descubrir su fuente.

Además, permitiendo seleccionar una dirección IP, mascara de subred o *gateway* por defecto puede causar problemas generales en la red si el *gateway* por defecto es invalido o la mascara de la red es incorrecta y los problemas asociados con una dirección IP duplicada.

Otra limitación de configurar manualmente el TCP/IP es la carga administrativa en grandes redes en donde por ejemplo, los ordenadores son movidos físicamente de unos puestos de trabajo a otros. Por ejemplo, si un ordenador lo cambiamos a una diferente subred, la dirección IP y el *gateway* por defecto deben cambiarse para que el ordenador pueda comunicarse desde su nueva localización.

Configurando TCP/IP usando DHCP.

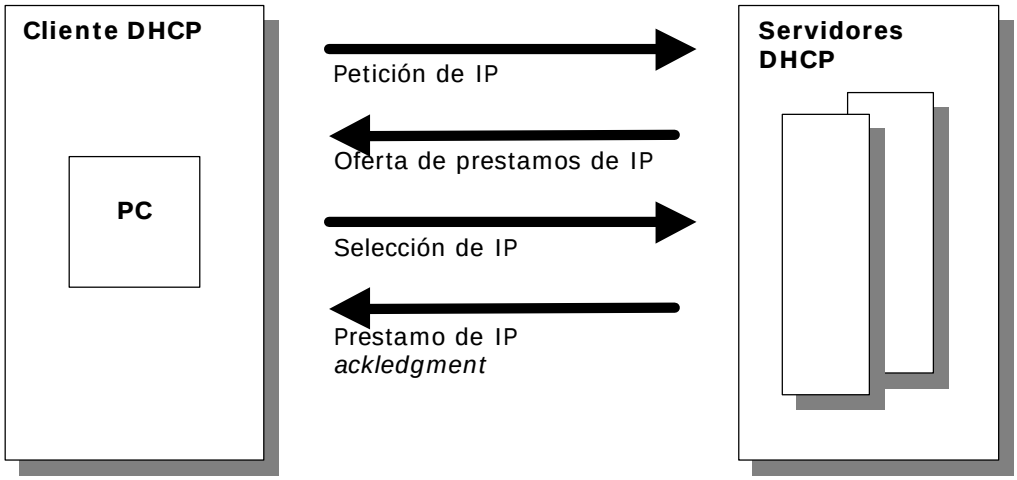
Usando DHCP para configurar automáticamente la información de la dirección IP permite que los usuarios no necesiten adquirir información de un administrador para configurar el TCP/IP. El servidor de DHCP suministra toda la configuración necesaria a los clientes DHCP. Algunos de los problemas más dificultosos de verificar mensajes y caminos en una red son eliminados al utilizar DHCP.

Como trabaja el DHCP

El DHCP utiliza un proceso de 4 fases para configurar un cliente DHCP como vamos a ver en la siguiente tabla y en el correspondiente gráfico. Si un ordenador tiene múltiples adaptadores de red, el proceso de DHCP ocurre exactamente igual en cada adaptador de red. Una única dirección IP válida será asignada a cada adaptador. Todas las comunicaciones DHCP se hacen a través de los puertos UDP 67 y 68.

Algunos mensajes DHCP son enviados mediante *broadcast*. Para que los clientes DHCP puedan comunicar con un servidor DHCP en una red remota, los *routers* IP deben soportar *fordwarding DHCP broadcast*.

Fase	Descripción
Petición IP	El cliente inicializa un versión limitada de TCP/IP y emite una petición *broadcast* para localizar un servidor DHCP.
Ofrecimiento de IP	Todos los servidores DHCP que tienen información de direcciones válidas envían una oferta al cliente.
Selección de IP	El cliente selección la dirección IP de la primera oferta que recibe y envía una petición *broadcast* para pedir prestada la dirección IP de la oferta.
Préstamo de IP	El servidor DHCP que ha hecho la oferta responde al mensaje y el resto de servidores DCHP liberan su oferta. La información de la dirección IP es asignada al cliente y un ACK (*acknodledgment*) se envía. El cliente termina inicializando y enlazando el protocolo TCP/IP. Una vez que el proceso de configuración automático se completa, el cliente puede comunicarse y conectar con otras direcciones IP.



Petición de préstamo y oferta.

En las primeras dos fases el cliente pide un préstamo al servidor DHCP y los servidores DHCP ofrecen direcciones IP al cliente.

Petición de préstamo de IP

En el primer momento en que el cliente se inicializa, pide un préstamo de dirección IP mediante *broadcasting* a los servidores DHCP. Debido a que el cliente, todavía no tiene dirección IP y desconoce la dirección IP de los servidores, utiliza la dirección 0.0.0.0 como origen y la dirección 155.155.155.155 como dirección de destino.

La petición de prestamos es enviada en un mensaje DHCPDISCOVER. El mensaje también contiene la dirección hardware y el nombre del ordenador del cliente, con lo cual los servidores DHCP conocen quien es el cliente que ha enviado la petición.

El proceso de préstamo de IP se utiliza si ocurre uno de los siguientes casos:

- El TCP/IP se inicializa por primera vez como un cliente DHCP.
- El cliente solicita una determinada dirección IP y le es denegada, posiblemente a causa de que el servidor DHCP ha perdido este préstamo.
- El cliente previamente ha tomado prestada esa dirección IP, pero libera este prestamo y ahora solicita un nuevo prestamo.

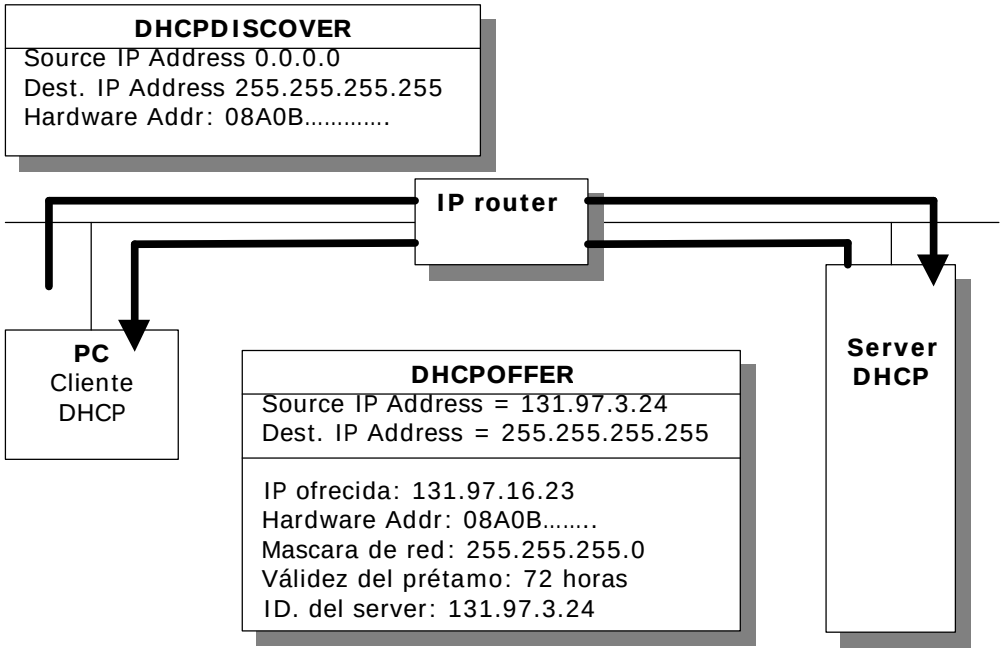
Oferta de préstamo IP

Todos los servidores DHCP que reciben la petición y tienen una configuración válida para el cliente, emiten una oferta con la siguiente información.

- La dirección hardware del cliente.
- Una oferta de dirección IP.
- La mascara de red.
- Validez del préstamo.
- Un identificador del propio servidor (la dirección IP del servidor DHCP).

Se utiliza una petición *broadcast* debido a que el cliente no tiene una dirección IP válida. En el gráfico que veremos a continuación veremos que la oferta se envía como un mensaje DHCPOFFER.

El servidor DHCP reserva esa dirección IP y por tanto, no le será ofrecida a otro posible cliente DHCP. El cliente DHCP selecciona la primera dirección IP que recibe.



Cuando no hay servidores DHCP on-line.

El cliente DHCP espera un segundo para una oferta. Si no se recibe una oferta el cliente no va a ser capaz de inicializarse correctamente y por tanto vuelve a emitir otra petición *broadcast* tres veces (con 9, 13 y 16 segundos de intervalo más un periodo de tiempo aleatorio entre 0 y 1000 milisegundos). Si no existe oferta para ninguna de las cuatro peticiones, el cliente lo reintentará cada 5 minutos.

Selección de la IP prestada y ACK

En las últimas dos fases, el cliente selecciona una oferta y el servidor envía un mensaje de confirmación – ACK.

Selección de la IP.

Después de que el cliente reciba una oferta de al menos un servidor DHCP, envia un mensaje *broadcast* a todos los servidores informando que ha seleccionado una oferta aceptándola.

El mensaje *broadcast* es enviado como un mensaje DHCPREQUEST e incluye la dirección IP del servidor cuya oferta ha sido aceptada. El resto de servidores DHCP retiran sus ofertas y esas direcciones IP quedan libres para ofrecérselas a otros posibles clientes.

Mensaje de confirmación del préstamo – ACK. (CORRECTO)

El servidor DHCP del cual ha sido aceptada la oferta envía un mensaje de confirmación 'exitosa' mediante un mensaje DHCPACK. Este mensaje contiene un préstamo válido para una dirección IP y posiblemente otra información complementaria.

Cuando el cliente DHCP recibe esta confirmación, el TCP/IP está completamente inicializado.

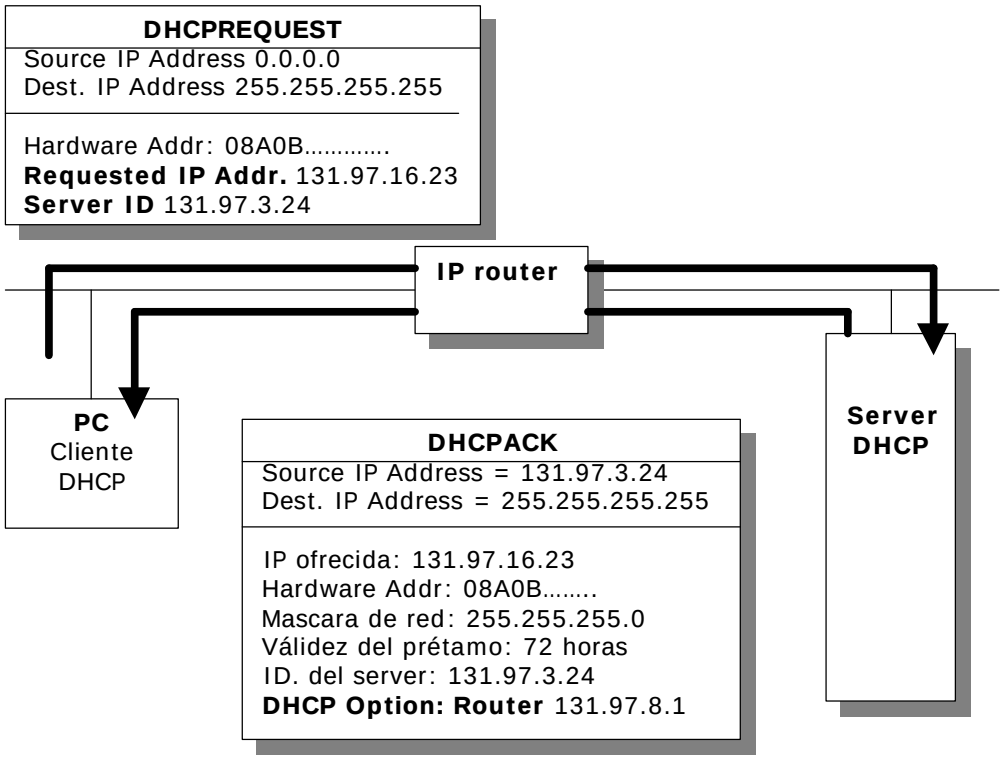
El cliente almacena esta dirección IP, mascara de red y el resto de información en el registro bajo la siguiente clave:

HKEY_LOCAL_MACHINE
SYSTEM
CurrentControlSet
Services
Adapter
Parameters
Tcpip

Mensaje de confirmación del préstamo – ACK. (INCORRECTO)

Se envía un mensaje DHCPNACK si el cliente está intentando utilizar un dirección IP previa y esta direccion IP ya no está disponible. Igualmente en el caso de que la dirección IP fuese inválida, por ejemplo su el ordenador se ha movido físicamente a una subred diferente.

Como vemos en el siguiente gráfico, cuando un cliente recibe un INCORRECTO ACK se vuelve al proceso de petición de préstamo de IP.



Renovación del préstamo de IP

Solicitud de renovación inicial.

Todos los clientes DHCP intentan renovar su préstamo cuando se ha cumplido el 50% del tiempo del préstamo. Para renovar este préstamo, un cliente DHCP envía un mensaje DHCPREQUEST directamente al servidor DHCP del cual obtuvo su préstamo.

Si el servidor DHCP está disponible. Renueva el préstamo y envía al cliente un mensaje de ACK CORRECTO (DCHPACK) con el nuevo plazo de validez (tiempo de vida) y cualquier otro parámetro de configuración actualizado.

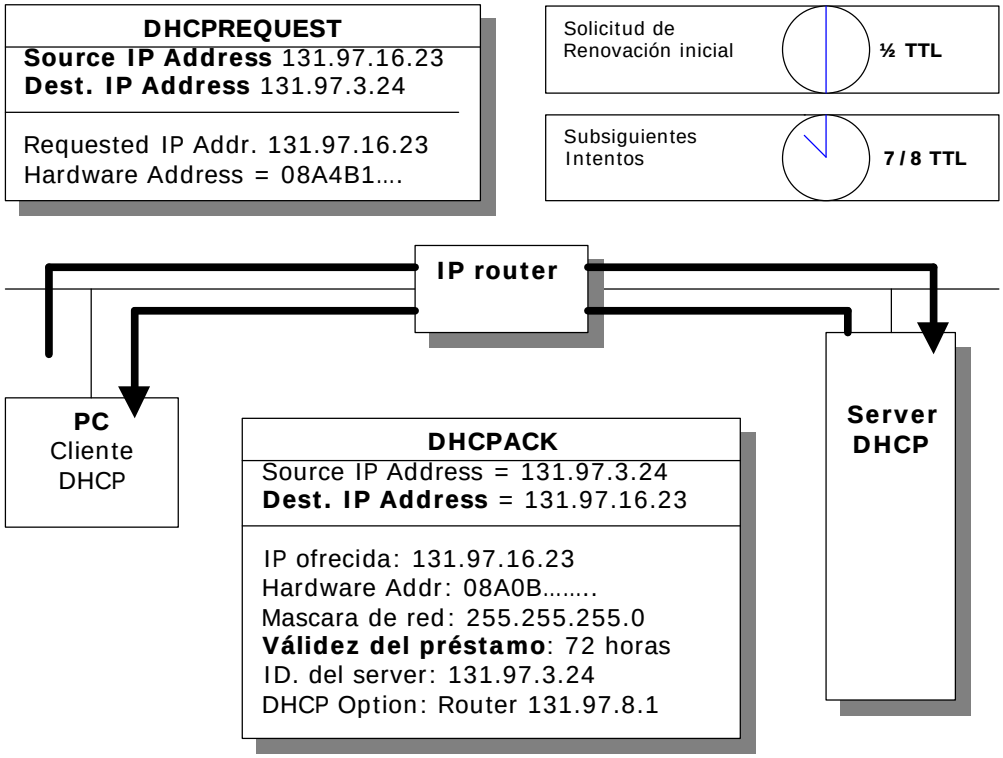
Cuando el cliente recibe este mensaje de ACK, actualiza su propia configuración. Si un cliente solicita renovar su préstamo de IP pero no es capaz de contactar con el servidor DHCP original, el cliente

recibe un mensaje indicando que el préstamo no ha sido renovado. El cliente, todavía puede seguir usando la dirección debido a que todavía tiene disponible el 50% del tiempo de vida del préstamo original.

Cuando reiniciamos la maquina configurada como un cliente DHCP, esta, intenta siempre solicitar una renovación del préstamo de IP que tenía anteriormente del servidor DHCP original. Sino pudiese hacerlo, envía mediante *broadcasting* un mensaje DHCPREQUEST especificando la última IP prestada. Si no consigue respuesta y todavía le queda tiempo de vida del préstamo, el cliente DHCP continúa usando la misma dirección IP durante el tiempo de préstamo restante que le quede.

Subsiguientes intentos de renovación.

Si el préstamo no puede ser renovado por el servidor DHCP original cuando se ha cumplido el 50% de tiempo de vida, el cliente intentará contacta con cualquier servidor DHCP cuando se haya cumplido el 87,5% del tiempo. Como vemos en el siguiente gráfico, el cliente envía un mensaje DHCPREQUEST mediante *broadcasting*. Cualquier servidor DHCP puede responder mediante un mensaje DHCPACK (renovando el préstamo) o un mensaje DHCPNACK, forzando en este caso al cliente DHCP a reiniciar la secuencia de obtención de préstamo para una nueva dirección IP).



Si la licencia expira o se recibe un mensaje un mensaje DHCPNACK, el cliente DHCP debe dejar de utilizar esa dirección IP inmediatamente. El cliente DHCP vuelve entonces a comenzar de nuevo su proceso de solicitud de un nuevo préstamo de dirección IP.

Si expira la licencia del préstamo de dirección IP y el cliente no puede obtener uno nuevo, la comunicación vía TCP/IP se detiene hasta que una nueva dirección IP pueda ser asignada al cliente. Ocurren en este caso errores de red para cualquier aplicación que espere comunicarse bajo esta situación (*invalid TCP/IP protocol stack interface).*

PROGRAMA DE UTILIDAD *IPCONFIG*

Para verificar la configuración IP de un ordenador tenemos la utilidad *Ipconfig* que además puede ser utilizada para renovar las opciones y tiempo de vida de una licencia de préstamo IP, así como volver a solicitar un nuevo préstamo de direccion IP.

En la ventana de comandos, podemos teclear el siguiente comando para verificar la dirección IP del ordenador, la mascara de red y el *gateway* por defecto:

ipconfig

Igualmente en la ventana de comandos para verificar la configuración IP del ordenador y el adaptador de red, podemos teclear:

ipconfig /all

Usando el parámetro **/all** nos suministra la siguiente información de la configuración:

- Nombre del *host* del ordenador local.
- Dirección IP de los servidores de DNS.
- Tipo de nodo NetBIOS: *broadcast,* híbrido, *peer-peer*, y mixto.
- ID NetBIOS.
- Si está permitido en encaminamiento (*routing)* de IP.
- Si la resolución WINS está o no activa.
- Si la resolución NetBIOS usa DNS.

Con el parámetro **/all** obtenemos la siguiente información del adaptador de red:

- Descripción de la tarjeta adaptadora.
- Dirección física de la tarjeta adaptadora.
- Si tenemos activado el DHCP.
- La dirección IP del ordenador.
- Mascara de red.
- *Gateway* por defecto.
- *D*irección IP de los servidores WINS primario y secundario.

Actualizando una licencia (préstamo de IP).

El parámetro **/renew** causa el envío de un mensaje DHCPREQUEST al servidor DHCP para obtener nuevamente las opciones de actualización y un nuevo tiempo de vida. Si el servidor DHCP no está disponible el cliente puede continuar usando la configuración enviada por ultima vez desde el servidor DHCP. En la pantalla de comandos, debemos teclear:

ipconfig /renew

Liberando una licencia (un préstamo de IP).

El parámetro **/release** provoca el envío de un mensaje DHCPRELEASE al servidor DHCP para liberar la licencia. Después de que este comando ha sido enviado, las comunicaciones TCP/IP se paran. Debemos teclear en la pantalla de comandos:

ipconfig /release

Los clientes DHCP de Microsoft no inician un mensaje DHCPRELEASE cuando se apagan. Si un cliente permanece apagado durante el tiempo de vida de la licencia, el servidor DHCP puede asignar esa dirección IP a otro posible cliente IP que le solicite una licencia. Al no enviar el cliente al cerrar la sesión un mensaje DHCPRELEASE, el cliente tiene muchas probabilidades de recibir la misma direccion IP al volver a iniciarse.

INSTALANDO Y CONFIGURANDO UN SERVIDOR DHCP

Antes de instalar un servidor DHCP debemos considerar varias cuestiones acerca de nuestra instalación.

Planteémonos primero las siguientes cuestiones:

- ¿Van a funcionar todos los ordenadores como clientes DHCP? Si no, debemos considerar que los clientes que no sean DHCP deberán tener una IP fija y por tanto esas IP fijas y estáticas deberán ser excluidas del rango de direcciones a asignar por los servidores DHCP. Si un cliente necesita una dirección IP específica esta direccion IP deberá ser una dirección reservada en todos los servidores DHCP.

- ¿Van a suministrar los servidores DHCP direcciones IP a múltiples subredes? Si es así, debemos considerar que cualquier *router* conectando subredes deben actuar como un agente de relevo DHCP (DHCP *relay agent)*. Si nuestros *routers* no tienen activo el DHCP *relay agent* al menos un servidor DHCP será requerido en cada subred en la que existan clientes DHCP.

- ¿Cuántos servidores DHCP necesitamos? Debemos recordar que un servidor DHCP no comparte información con otros servidores DHCP. Esto implica que deberemos crear un rango de direcciones IP diferente para cada servidor y que pueda por tanto dar de su 'rango a los clientes DHCP.

- ¿Al recibir la dirección IP, que mas opciones pueden obtener los clientes desde un servidor DHCP? Las opciones y el resto de información recibida, pueden ser:

    - *Router*
    - Servidor de DNS
    - Resolución de nobres NetBIOS sobre TCP/IP
    - Servidor WINS
    - Alcance (rango) del ID NetBIOS.

  Estas opciones se determinan cuando se configura el servidor DHCP.

Implementando múltiples servidores DHCP

Si nuestra red requiere múltiples servidores DHCP es necesario crear un ámbito o rango de direcciones para cada subred. Este rango, serán las únicas direcciones IP que cada servidor puede dar a sus clientes. Debido a que los servidores DHCP no comparten este tipo de información entre ellos, debemos prestar atención especial al rango asignado a cada servidor y que estos rangos no se crucen entre ellos.

Para asegurarse que los clientes pueden obtener una licencia de préstamo IP, es importante tener distribuidos múltiples rangos para cada subred a lo largo de los servidores DHCP. Por ejemplo:

- Cada servidor DHCP debe tener un rango de aproximadamente el 75% de las direcciones IP para la subred local.

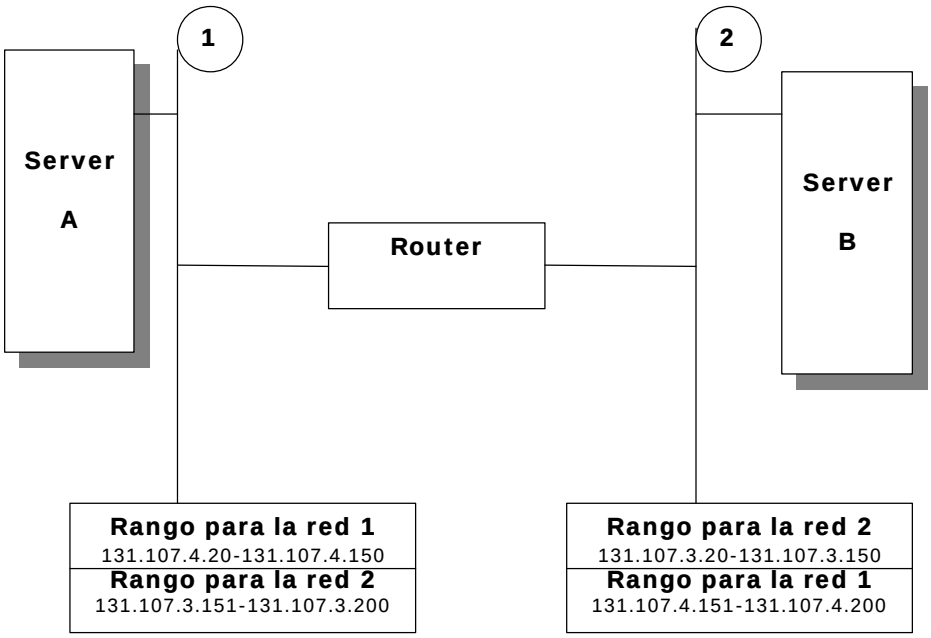
- Cada servidor DHCP debe tener un rango de aproximadamente el 25% de las direcciones posibles de cada subred remota.

Cuando un servidor DHCP está indisponible en una subred, el cliente puede recibir la licencia de préstamo de IP desde otro servidor DHCP en una subred diferente (asumiendo que el *router* entre subredes funciona como un agente DHCP *relay).*

Como vemos en la siguiente figura, el servidor **A** tiene un rango de IP desde 131.107.4.20 hasta 131.107.4.150 en la subred local, y el servidor **B** tiene un rango de direcciones IP desde 131.107.3.20 hasta 131.107.3.150. Cada servidor puede prestar estas licencias de IP en su propia subred.

Adicionalmente, cada servidor tiene un pequeño rango de direcciones IP para las subredes remotas. Por ejemplo, el servidor **A** tiene un rango para la **subred 2** desde 131.107.3.151 hasta 131.107.3.200. El servidor **B** tiene un rango para la **subred 1** de direcciones IP desde 131.107.4.151 hasta 131.107.4.200.

Cuando un cliente en la **subred 1** no puede obtener licencia del servidor **A**, puede obtener la licencia de préstamo de IP desde el servidor **B** y viceversa.



Requerimiento del DHCP

Para implementar DHCP, ambos, el servidor y el cliente requieren que sean configurados. Todos los *routers* que conectan subredes con servidores DHCP deben soportar la RFC 1542 y deben actuar como BOOTP *relay agents.* (agentes de relevo BOOTP).

Un servidor DHCP requiere:

- El servicio de servidor DHCP configurado al menos en un ordenador en la red TCP/IP ejecutando Windows NT Server o Windows 2000 (si no existe un controlador de dominio) y que los *routers* soporten la RFC 1542. En otro caso necesitaremos un servidor DHCP en cada subred.

- El servidor DHCP debe estar configurado con una IP estática, mascara de subred y *default gateway* además del resto de parámetros del TCP/IP (es decir, no puede ser además un cliente DHCP).

- El rango (alcance) del servidor DHCP debe estar definido. Un rango de DHCP consiste en un ámbito de direcciones IP que el servidor puede asignar (o prestar) a los clientes DHCP. Por ejemplo: 131.107.3.51 hasta 131.107.3.200.

Todos los clientes DHCP deben tener un sistema operativo que soporte que el DHCP pueda activarse. En la familia Windows pueden ser:

- Windows NT 4 Server o Windows 2000 Server
- Windows NT Workstation o Windows 2000 Professional.
- Windows 95
- Windows 98
- Windows Millennium
- Windows 3.11 ejecutando TCP/IP-32 (se suministra este ultimo en los discos del Server NT).
- Cliente para redes Microsoft 3.0 para MS-DOS con los drivers de TCP/IP en modo real (suministrados en los CD de Windows NT Server).
- LAN Manager 2.2c incluido en los CD de Windows NT Server. (LAN Manager para 2.2c para OS/2 no está soportado).

INSTALANDO Y CONFIGURANDO UN SERVIDOR DHCP

El servicio de servidor DHCP debe estar ejecutándose para comunicarse con los clientes DHCP. Una vez que el servidor DHCP está instalado y arrancado, algunas opciones deben ser configuradas. Vamos a ver los siguientes pasos para instalar y configurar DHCP:

- Instalar el servicio de Servidor DHCP.
- Un rango o *pool* de direcciones deben estar definidas antes que el servidor DHCP pueda dar una licencia de préstamo de IP a los clientes DHCP.
- Opciones globales y de rango de clientes pueden ser configuradas para un cliente particular de DHCP.
- El servidor DHCP puede ser configurado para que asigne siempre la misma dirección IP al mismo cliente DHCP.

Nota: El servidor DHCP no puede ser a su vez cliente DHCP. Debe por tanto tener, dirección IP, mascara de red y *gateway* por defecto perfectamente definidos.

ACTIVANDO EL AGENTE DE *RELAY* DEL DHCP

Windows NT Server y Windows 2000 cumplen con la RFC 1542 para poder actuar como un DHCP *relay agent.* Este agente, cuando está usado en conjunción con *routers* estáticos o dinámicos, permite los mensajes DHCP entre los clientes DHCP y los servidores DHCP en diferentes redes IP.
Si los *routers* separan los clientes y servidores DHCP, debemos configurar el *router* (por ejemplo un Windows NT Server funcionando como *router*) como un DHCP *relay agent*. Este agente va a interceptar las llamadas *broadcast* DHCP y reenviar los paquetes al servidor DHCP, a traves de los routers IP. Podemos añadir este agente en un Windows NT Server en el programa de Red en el Panel de Control de Windows NT.

MANEJANDO LA BASE DE DATOS DEL DHCP (*DHCP DATABASE*)

La base de datos del DHCP realiza un backup automático propio cada 60 minutos. Si Windows NT o Windows 2000 detectan un base de datos corrompida, automáticamente recupera de la ultima copia de backup.

Realizando un backup de la base de datos del DHCP

Las copias de backup son almacenadas en:

systemroot\System32\Dhcp\Backup\Jet

El intervalo de backup por defecto puede cambiarse colocando el valor de **BackupInterval** con los minutos que deseemos y rearrancando del servicio de Servidor de DHCP. El parámetro **BackupInterval** está en el registro en la siguiente clave:

```
HKEY_LOCAL_MACHINE
   System
      CurrentControlSet
         DHCPServer
            Parameters
               BackupInterval
```

Una copia de esta subclave de registro está almacenada como DHCPCFG en el directorio:

systemroot\System32\Dhcp\Backup

Recuperando la base de datos del DHCP

La base de datos del DHCP puede ser restaurada automática o manualmente. Para el proceso de restauración podemos utilizar cualquiera de los siguientes métodos:

- Rearrancar el servicio de Servidor de DHCP. Si el Servidor de DHCP detecta una base de datos corrupta, este, automáticamente recupera de la copia previa de backup anterior.

- Colocar el valor **RestoreFlag** a **1** y reiniciar el servicio de Servidor de DHCP. El parámetro **RestoreFlag** está en el registro en la clave siguiente:

  HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DHCPServer\
  Parameters.

  Una vez que la base de datos ha sido recuperada el servidor cambia automáticamente este valor por el valor de **0**.

- Copiar el contenido de *systemroot*\System32\Dhcp\Backup\Jet al directorio de *systemroot*\System32\Dhcp y reiniciar el servicio de Servidor DHCP.

COMPACTANDO LA BASE DE DATOS DEL DHCP

Windows NT Server y Windows 2000 estan diseñados para compactar automáticamente la base de datos del DHCP por lo que normalmente no necesitaremos realizar estos procedimientos de mantenimiento. Sin embargo, si todavía usamos Windows NT Server versión 3,51 o anterior deberemos compactar manualmente su base de datos cada vez que esta alcance un tamaño aproximado de 30 megabytes.

Para compactar la base de datos del DHCP podemos usar la utilidad JetPack. Esta utilizada se lanza desde la consola de comandos.

Para compactar la base de datos:

- Para el servicio del Servidor DHCP. Podemos hacerlos desde el Panel de Control, Servicios, Microsoft DHCP Server o bien desde la linea de comandos. Para parar el servicio en la linea de comandos, debemos teclear:

  net stop dhcpserver

- En la línea de comandos, ir al directorio *systemroot*\System32\Dhcp y ejecutar la utilidad JetPack usando la siguiente sintaxis:

  jetpack dhcp.mdb nombre_temporal**.mdb**

  El contenido de dhcp.mdb será compactado en la nueva base de datos especificada en el 'nombre_temporal', esta será copiada al final sobre la original y será borrada.

- Rearrancar el servicio del Servidor DHCP desde el Panel de Control, Servicios, Mcrosfot DHCP Server, o desde la linea de comandos. Para rearrancar desde la linea de comandos, debemos ejecutar:

  net start dhcpserver

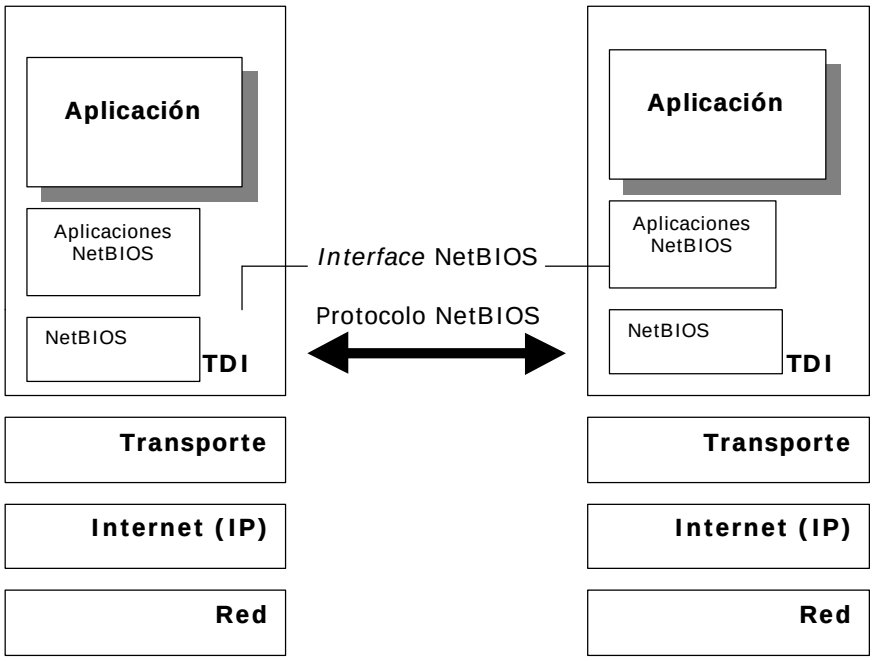
NETBIOS SOBRE TCP / IP

En los capítulos anteriores como es necesario resolver una dirección IP a la dirección hardware para poder comunicarse. Vamos a ver ahora acerca de la resolución de nombres NetBIOS, sus conceptos y los métodos. Vamos a intentar clarificar como se resuelve un nombre NetBIOS en una dirección IP usando *broadcast,* el fichero LMHOSTS, un servidor de nombres NetBIOS, un Servidor de nombres de dominio (DNS) y el fichero HOSTS. Vamos primero a aprender y a utilizar el fichero LMHOSTS.

NOMBRES NETBIOS

El nombre NetBIOS es el nombre asignado a nuestro ordenador. Vamos a explicar como en nombre NetBIOS se utiliza por la familia de productos Windows para comunicarse con otros ordenadores basados en NetBIOS.

El NetBIOS fue desarrollado por IBM en 1983 por *Sytek Corporation* para permitir a las aplicaciones comunicarse bajo una red. Como podemos ver en el siguiente gráfico NetBIOS define dos entidades: un nivel de sesión de *Interface* y una sesión de transporte (protocolo) de manejo y datos.



La *interface* NetBIOS es una API estándar para que las aplicaciones de usuario puedan enviar a la red peticiones de I/O (entrada / salida) y directivas de control bajo el software de red. Un programa de aplicación utiliza el API de *interface* NetBIOS sobre cualquier protocolo que soporte encapsulamiento NetBIOS.

NetBIOS también define un protocolo que funciona al nivel de sesión / transporte. Este está implementado por debajo del protocolo como un NBFP (NetBEUI) o NetBT para permitir que las entradas / salidas a la red se acomoden a la *interface* NetBIOS. NetBT o NetBIOS sobre TCP/IP es una sesión, es decir una capa en el servicio de red.

NetBIOS nos da comandos y soporte a los siguientes servicios:

- Registrar el nombre de red y su verificación.
- Establecimiento y finalización de sesión.
- Verdadera sesión orientada a la transferencia de datos.
- Transferencia de datos orientada a *datagramas.*

Nombres NetBIOS

Un nombre NetBIOS es un nombre único de 16 bytes de longitud usado para identificar un recursos NetBIOS en la red. Este nombre puede ser único (exclusivo) o de grupo (no exclusivo). Los nombres únicos se utilizan para el envío de comunicaciones a la red a un único proceso especifico en un ordenador. Los nombres de grupo, son usados para enviar información a múltiples ordenadores a un tiempo.

Podemos usar el comando **nbtstat –n** para ver nuestro nombre NetBIOS de nuestro ordenador. Un ejemplo de un proceso usando el nombre NetBIOS es el servicio de Servidor en un ordenador ejecutando Windows NT o Windows 2000. Cuando nuestro ordenador arranca, el servicio de servidor registra un único nombre NetBIOS basado en el nombre de nuestro ordenador. El nombre exacto usado por el servidor son los 15 primeros caracteres del nombre de nuestro ordenador más un decimosexto carácter con un contenido hexadecimal de 20. Otros servicios de red, también utilizan el nombre del ordenador para construir sus nombres NetBIOS utilizando el carácter decimosexto para cada servicio específico como por ejemplo: Redirector, Servidor, o servicios de Mensajería.

Cuando intentamos conectarnos con un ordenador ejecutando Windows NT, Windows 2000 o Windows 95 / 98 con el comando **net use**, el nombre NetBIOS para el servicio de servidor es buscado con una petición *Name Query*. Cuando el nombre es encontrado se establece la comunicación.

Todos los servicios registrados en una red Windows, son nombres NetBIOS. Todos los comandos Windows (Windows Explorer, Administrador de archivos y comandos **net**) usan nombres NetBIOS para acceder a dichos servicios.

Los nombres NetBIOS se utilizan también por otros ordenadores que están basados en NetBIOS como Windows para trabajo en grupo, LAN Manager, y LAN Manager en *hosts* UNIX.

Nombres NetBIOS comunes.

Muchas veces, con solo ver los nombres registrados nos puede ayudar para determinar los servicios que se están ejecutando en nuestro ordenador. La tabla siguiente describe en nombre común NetBIOS que podemos ver en la base de datos WINS (*Windows Internet Name Service*). Mas adelante, cuando veamos la implementación de WINS revisaremos estos temas.

Nombre registrado	Descripción
\\computer_name[00h]	El nombre registrado para el servicio de *Workstation* en el cliente WINS.
\\computer_name[03h]	El nombre registrado para el servicio de mensajería en el cliente WINS.
\\computer_name[20h]	El nombre registrado para el servicio de Servicio de servidor para el cliente WINS.
\\username[03h]	El nombre del usuario que actualmente está conectado (*logged on*) en el ordenador. El nombre usuario queda registrado en el servicio de mensajería para que el usuario pueda recibir mensajes enviados con el comando **net send** a su nombre de usuario. Si más de un usuario se conecta con el mismo nombre de usuario, solo el primer ordenador al cual se han conectado, va a registrar el nombre.
\\domain_name[1Bh]	El nombre del dominio registrado por el servidor Windows NT o Windows 2000 como controlador primario de dominio (PDC – *Primary Domain Controller*).

Registro del nombre NetBIOS, localización y liberalización.

Todos los nodos NetBIOS sobre TCP/IP utilizan el registro de nombres (*name registration)*, localización de nombres (*name discovery)* y liberar dichos nombres (*name release*) para interactuar con los *hosts* NetBIOS, como por ejemplo con un ordenador cuyo sistema operativo es Windows.

Name Registration

Cuando es inicia el NetBIOS sobre TCP/IP , registra el nombre NetBIOS usando una petición de registro de nombre (*name registration request)*. Este registro puede hacerse usando un *broadcast* o enviando un mensaje directo al servidor de nombres NetBIOS.

Si otro *hosts* está registrado con el mismo nombre NetBIOS, cualquier otro *hosts* o el servidor de nombres NetBIOS responde con un mensaje negativo de registro de nombre (*negative name registration response*). El *hosts* que está arrancando recibe un error de inicalización.

Name Discovery

La localización de nombres en una red local es manejada por peticiones locales *broadcast* o por un servidor de nombres. Cuando Windows quiere comunicarse con otro *host* TCP/IP, un mensaje NetBIOS *name query request* conteniendo el nombre de destino NetBIOS es enviado mediante *broadcasting* a la red local o enviado al nombre del Servidor de nombres NetBIOS para intentar resolverlo.

El ordenador propietario de ese nombre NetBIOS, o bien el servidor de nombres NetBIOS, responden enviando un mensaje *positive name query response.*

Name Release

La liberalización del nombre, sucede cuando una aplicación NetBIOS o un servicio se detiene. Por ejemplo, cuando el servicio de *Workstation* en un ordenador se detiene, el ordenador envia una respuesta negativa al servicio de nombres cuando algun otro ordenador intenta usar este nombre. El nombre NetBIOS se dice que ha sido 'liberado' y disponible para el uso por parte de otro ordenador.

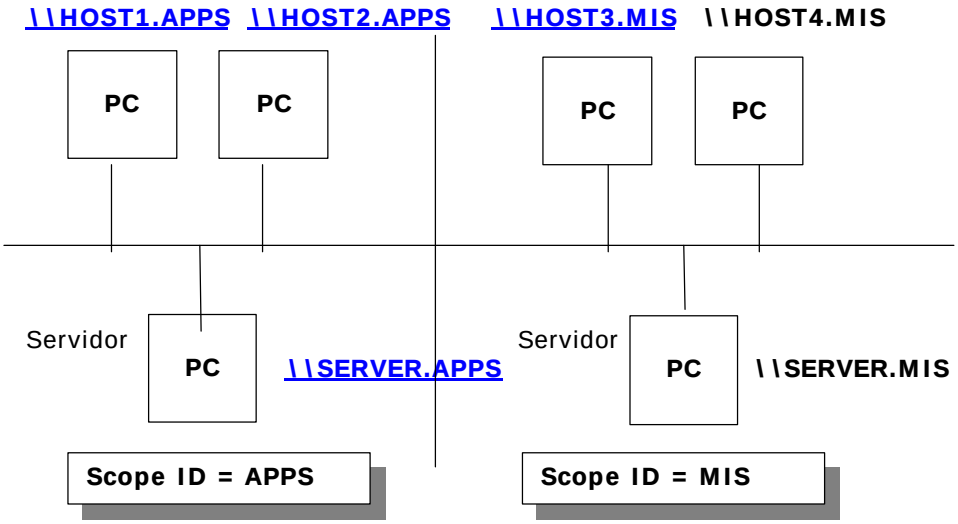
Segmentando los nombres NetBIOS.

Otro parámetro muy utilizado es el *scope ID* (alcance del nombre). Este alcance (*scope)* se utiliza para segmentar el espacio de nombres NetBIOS. Utilizando estas técnicas, no aumentaremos el rendimiento de la red, pero si se reducirán el número de paquetes que serán aceptados y evaluados por el *host.*

El '*scope ID*' NetBIOS es una cadena de caracteres que se añade al nombre NetBIOS. Es usado para segmentar los 16 caracteres del nombre NetBIOS. Sin '*scopes*' un nombre NetBIOS debe ser único en todos los recursos de la red. Con '*scopes*' un nombre NetBIOS es único solo con un *scope* en particular, no en todo el espacio de nombres de una red.

Los recursos NetBIOS utilizando *scope* se aislan de todos los demas recursos NetBIOS fuera de ese *scope.* El *scope ID* NetBIOS debe coincidir en dos *hosts* para que sean capaces de comunicarse.

Como vemos en el siguiente grafico, dos *scopes* NetBIOS estan siendo utilizados: APPS y MIS :



- HOST1.APPS y HOST2.APPS serán capaces de comunicarse con SERVER.APPS, pero no podrán comunicarse con el resto.
- El alcance (*scope*) NetBIOS permite que los ordenadores puedan utilizar el mismo nombre NetBIOS (teniendo un diferente '*scope*'). Debido a que el *scope* NetBIOS forma parte del nombre NetBIOS, esta combinación formará un nombre único.

Nota: El *scope* ID NetBIOS está definido en la RFC 1001

RESOLUCIÓN DE NOMBRES NETBIOS

El resolver el nombre NetBIOS de un ordenador a su dirección IP es lo que se llama 'resolución de nombres NetBIOS'.

La resolución de nombres NetBIOS es el proceso de localizar correctamente la dirección IP a través del nombre NetBIOS del ordenador. Antes de que una dirección pueda resolverse en un dirección hardware el nombre NetBIOS de un ordenador debe estar resuelto en una dirección IP.

El TCP/IP de Microsoft utiliza diversos métodos para resolver lo nombres NetBIOS. El tipo de método a utilizar, depende de que el *host* sea local o remoto.

Métodos estándar de resolución	**Descripción**
Caché de nombres NetBIOS	El caché local conteniendo los nombres NetBIOS que el ordenador local ha resuelto recientemente.
Servidor de nombres NetBIOS	(NBNS) Un servidor que bajo las normas de las RFC 1001 y 1002 nos da resolución de nombres NetBIOS. La implementación por parte de Microsoft de este servidor es el WINS.
Broadcast local.	Un *broadcast* en la red local para la dirección IP del nombre NetBIOS de destino.
Fichero LMHOSTS	Un fichero de texto que *'mapea'* direcciones IP en nombres NetBIOS de ordenador para ordenadores en una red Microsoft en redes remotas.
Fichero HOSTS	Un fichero de texto en el mismo formato que el fichero 4.3 *Berkeley Software Distribution* (BSD) UNÍS\Etc\Hosts. Este fichero relaciona nombres *hosts* con direcciones IP. Este fichero se utiliza fundamentalmente las utilidades TCP/IP para resolver nombres de *hosts.*
Domain Name Server (DNS)	Un servidor que mantiene una base de datos de direcciones IP / nombres de ordenador (*host name)*.

Resolviendo nombres NetBIOS locales usando un *broadcast.*

Cuando el *host* destino está en la red local, la resolución de nombres NetBIOS es usando un *broadcast.* Los siguientes pasos y el gráfico muestran los procesos:

1) Cuando un usuario o el propio sistema utiliza un comando **net use**, se chequea el caché de nombres para encontrar la dirección IP que corresponde al nombre NetBIOS del *host* de destino. Esto elimina *broadcast* extraños en la red. Si el nombre ha sido resuelto recientemente se encontrará la direccion en la caché de nombres y no se emitirá el *broadcast.*

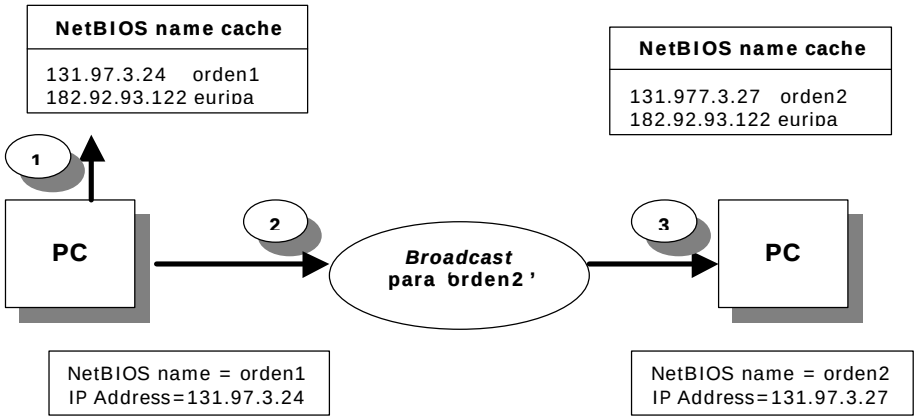
2) Si el nombre NetBIOS no se resuelve en el caché de nombres, el ordenador origen de la búsqueda emitirá un mensaje *'name query'* mediante *broadcast* a la red local con el nombre NetBIOS del destino.

3) Cada ordenador de la red local recibe el *broadcast* y chequea en su propia tabla local NetBIOS para ver si es propietario del nombre pedido.

   El ordenador propietario del nombre emite un mensaje de respuesta *'name query response'*. Antes de que la respuesta sea emitida, se utiliza ARP (mediante caché o *broadcast*) para obtener la dirección hardware del ordenador origen. Cuando se ha obtenido la dirección hardware se envía la respuesta.

   Cuando el ordenador origen recibe este mensaje, la sesion **net use** queda establecida.

```
net use x: \\orden2\public
```



Limitaciones de los *broadcast*.

No todos los *routers* pueden dejar pasar los mensajes *broadcast*. Además, lo mas normal es que tengan deshabilitada esta característica debido a que reenviando los paquetes de *broadcast* se incrementa el trafico entre redes lo cual afecta negativamente al rendimiento de la red. Por tanto, los *broadcast* se reducen al ámbito de la red local.

Nota: Para que un *router* reenvíe los mensajes *broadcast*, los puertos UDP 137 y 138 deben estar activos para permitir reenvío de paquetes en el *router.*

Resolviendo nombres con un Servidor de Nombres NETBIOS

Un método común de resolver nombres NetBIOS a direcciones IP es con un servidor de nombres NetBIOS. El proceso de resolución es de la siguiente forma:

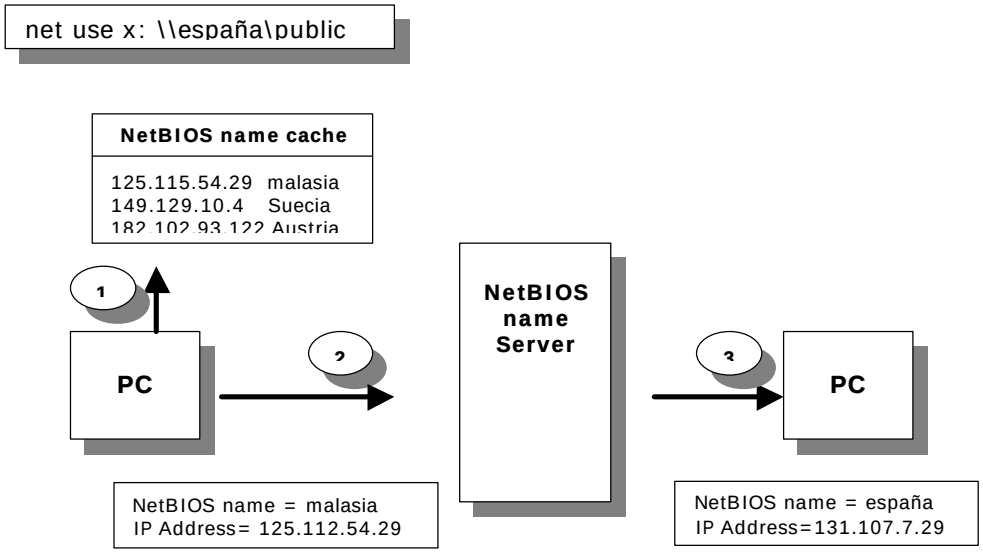
1) Cuando un usuario o el propio sistema emite un comando como **net use** comienza el proceso de resolución de nombres. Se busca primero en la caché de nombres NetBIOS. Si este nombre no se localiza en la caché, el cliente Windows va a intentar determinar la dirección IP usando otros métodos.

2) Si el nombre no puede ser resuelto usando la caché de nombres NetBIOS, el nombre NetBIOS del *host* de destino se envía al servidor de nombres NetBIOS que está definido en la configuración del ordenador origen. Cuando el nombre NetBIOS está resuelto a una dirección IP, esta es devuelta al ordenador peticionario.

   Por defecto, el cliente Windows espera localiza el servidor WINS primario tres veces. Si no obtiene respuesta, intenta contactar con el servidor secundario. Sin embargo, si el servidor WINS primario notifica que no posee la dirección buscada del destino, Windows acepta esta respuesta y no buscará en el servidor secundario.

3) Después de que haya sido resuelto el nombre NetBIOS, el ordenador origen usa un mensaje ARP para resolver la dirección IP a la dirección hardware.

4)



Métodos Microsoft para resolver nombres NetBIOS

La resolución de nombres NetBIOS puede ser resuelta utilizando una combinación de los métodos soportados por Microsoft. Windows puede ser configurado para resolver nombres NetBIOS utilizando el fichero LMHOSTS, el fichero HOSTS y un DNS, además de *broadcasting* y la utilización del servidor de nombres NetBIOS. Si uno de estos métodos falla se intenta el método siguiente para continuar la búsqueda. En el siguiente ejemplo, vemos como se combinan estos métodos:

1) Cuando un usuario o el propio sistema utiliza un comando como **net use**, se investiga en la caché de nombres NetBIOS. Si el nombre se encuetra, será resuelto inmediatamente sin generar actividad de red.

2) Si el nombre no es resuelto por el caché de nombres NetBIOS, se realizan 3 intentos para contactar con el servidor de nombres NetBIOS (si está configurado así). Si se resuelve el nombre se devuelva ya la direccion IP al ordenador origen de la petición.

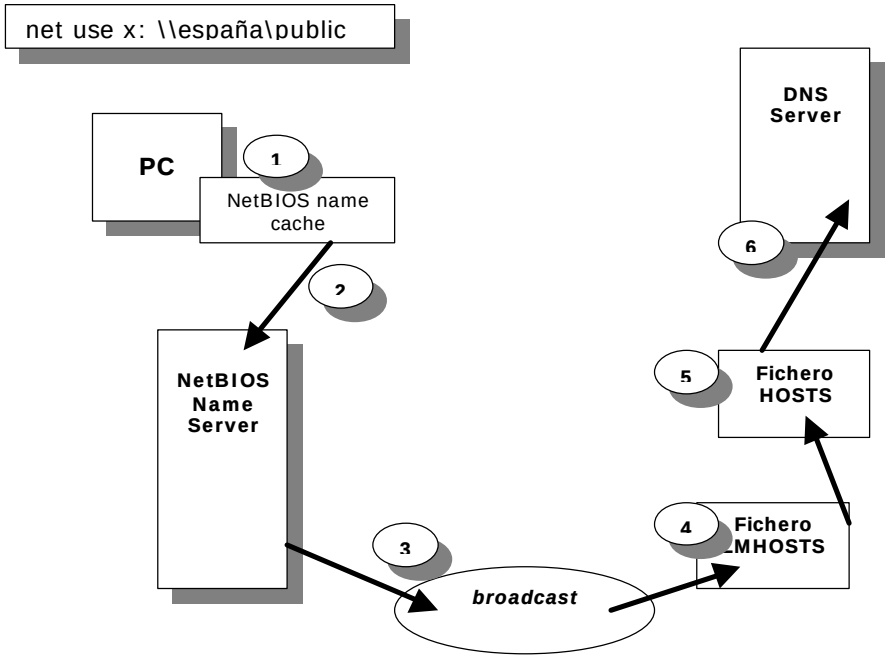
3) Si el nombre no se resuelve por el servidor de nombres NetBIOS, el cliente genera tres *broadcast* a la red local. Si el nombre NetBIOS se encuentra en la red local, ya estará resuelta su dirección.

4) Si el nombre NetBIOS no se puede resolver usando *broadcast* se busca en el fichero local LMHOSTS. Si el nombre es localizado en este fichero, ya estará resuelta su dirección.

5) Si el nombre NetBIOS no está resuelto en el fichero LMHOSTS, Windows intenta resolver el nombre mediante las siguientes técnicas: sí tenemos activo **Enable DNS for Windows resolution**, el primer paso es chequear si existe el fichero HOSTS y si en él se encuentra la resolución del nombre. Si lo localizamos, ya estará resuelta su dirección.

6) Si el nombre no es localizado en el fichero HOSTS o este no existe, el ordenador origen envía una petición al servidor de DNS. Si el nombre del host es encontrado por el servidor DNS, será ya resulta su dirección IP.

   Si el servidor de DNS no responde a la petición, se realizan intentos adicionales en intervalos de 5, 10, 20 y 40 segundos.



Si ninguno de estos metodos resuelve el nombre NetBIOS, el comando Windows devolverá un error al usuario indicando que el ordenador o el recursos buscado no puede ser encontrado.

Resolución de Nombres de Nodos en NetBIOS sobre TCP/IP

Windows da soporte a todos los nodos NetBIOS sobre TCP/IP que están definidos en las RFC 1001 y 1002. Cada nodo NetBIOS resuelve los nombres de diferente forma.

Nodo	Descripción
B-node	(*broadcast*) B-node utilizan *broadcast* (datagramas UDP) para el registro y resolución de nombres. B-node tiene dos grandes problemas: (1) En una gran red los *broadcast* incrementan la carga de la red y (2) los *routers* normalmente no están configurados para reenviar *broadcast* y por tanto solo responderán los ordenadores en la red local.
P-node	(*peer-peer*) P-node utiliza un servidor de nombres NetBIOS (NBNS) como por ejemplo WINS para resolver los nombres NetBIOS. P-node no utiliza *broadcast*: pregunta por el nombre del servidor directamente. Debido a que no se utilizan *broadcast* las peticiones pueden sobrepasar a los *routers*. Los problemas más significativos con P-node es que todos los ordenadores deben estar configurados con la dirección IP del servidor NBNS, y si el servidor NBNS se cae, los ordenadores no serán capaces de comunicarse ni tan siquiera en la red local.
M-node	(*mixed* – mixto) M-node es una combinación de B-node y P-node. Por defecto un M-node funciona como un B-node. Si no es capaz de resolver el nombre por *broadcast* utiliza el servidor NBNS de P-node.
H-node	*(hybrid)* H-node es una combinación de P-node y B-node. Por defecto un H-node funciona como un P-node. Si no es capaz de resolver el nombre por el servidor de NetBIOS, utilizará *broadcast* para resolver el nombre.
B-node(enh)	(*Microsoft enhanced B-node*) Microsoft utiliza un B-node extendido para resolver el nombre NetBIOS de ordenadores remostos. El fichero LMHOSTS es un fichero estático que convierte nombres NetBIOS a direcciones IP.

Las entradas en el fichero LMHOSTS que están marcadas con **#PRE** se llevan a la caché cuando se inicializa en TCP/IP. Antes de enviar un *broadcast* se chequea en la caché. Si allí no se encuentra se inicia el *broadcast.* Si tampoco es existoso, entonces se lee completo el fichero LMHOSTS para intentar resolver el nombre.

Nota: Los nodos NetBIOS sobre TCP/IP están definidos en las RFC 1001 y 1002.

Configurando tipos de Nodo

Podemos configurar el método de resolución de nombres que va a utilizar **NetBT** en la siguiente entrada del registro:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Netbt\Parameters

Nota: El sistema por defecto es *Microsoft Enhanced B-node* si no hay servidores WINS configurados. Si hemos configurado al menos un servidor WINS, el sistema por defecto será H-node.

Utilidad NBTSTAT

La utilidad NBTSTAT nos comprueba el estado de las conexiones actuales NetBIOS sobre TCP/IP, actualiza la caché LMHOSTS y determina nuestro nombre registrdo y el alcance (*scope ID*).

Este programa puede resultarnos de utilidad para la resolución de problemas y para precarga la caché de nombres NetBIOS.

Comando	Descripción
nbtstat –n	Lista los nombres NetBIOS registrados en el cliente.
nbtstat –c	Nos muestra la caché de nombres NetBIOS.
nbtstat –R	Manualmente recarga la caché de nombres NetBIOS usando las entradas que están en el fichero LMHOSTS marcadas con **#PRE**.

WINDOWS INTERNET NAME SERVICE – WINS

En los capítulos anteriores hemos visto los diferentes métodos de resolver nombres NetBIOS. En este capítulo vamos a ver como podemos implementar WINS y vamos a ver como con WINS se reduce el tráfico *broadcast* asociado con la implementación B-node de NetBIOS sobre TCP/IP.

WINS (*Windows Internet Name Service*) elimina la necesidad de *broadcast* para resolver nombres de ordenador en direcciones IP y además nos da una base de datos dinámica que mantiene relaciones entre nombres de ordenador y direcciones IP.

WINS es un servidor de nombres NetBIOS mejorado (NBNS) diseñado por Microsoft para eliminar el trafico *broadcast* asociado con la implementación B-node. Se utiliza para registrar nombres de ordenadores NetBIOS y resolver estos a direcciones IP tanto para *hosts* locales como remotos.

Existen varias ventajas para la utilización de WINS. La primera ventaja es que las peticiones de los clientes para resolver un nombre de ordenador son enviadas directamente al servidor WINS. Si el servidor WINS puede resolver el nombre, este enviará la dirección IP directamente al cliente. De esta manera no se necesita un *broadcast* y por tanto se reduce el tráfico de red. Si el servidor WINS no estuviese disponible entonces el cliente WINS utilizaría la técnica de *broadcast* vista anteriormente para intentar resolver el nombre.

Otra ventaja de usar WINS es que la base de datos de WINS se actualiza dinámicamente y por tanto siempre tiene datos actualizados. Esto elimina la necesidad de tener un fichero LMHOSTS. Además WINS nos da capacidades de ver y resolver redes e interdominios.

Antes de que dos ordenadores NetBIOS puedan comunicarse, el nombre del destino debe ser resulto como una dirección IP. Esto es necesario debido a que el TCP/IP requiere una dirección IP en vez de un nombre de ordenador. La resolución usa el siguiente proceso:

1) En un entorno WINS, cada vez que el cliente WINS arranca, registra el nombre / dirección IP NetBIOS del servidor WINS.

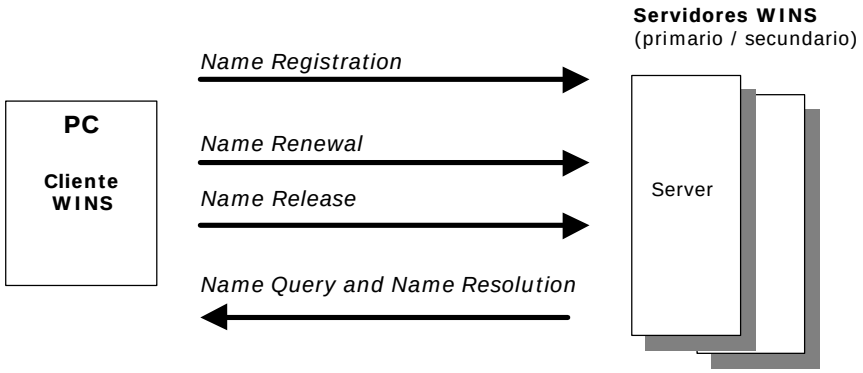
2) Cuando un cliente WINS inicia un comando Windows para comunicarse con otro *host*, una petición *'name query '*se envía directamente al servidor WINS en lugar de un *broadcasting* a la red local.

3) Si el servidor WINS encuentra el nombre NetBIOS en su base de datos, devuelve la dirección IP del ordenador destino. Debido a que el servidor WINS obtiene los nombres de NetBIOS y direcciones IP dinámicamente, siempre estará actualizado con los últimos datos de la red.

Proceso de resolución WINS

WINS utiliza para resolver y mantener nombres NetBIOS un proceso similar a la implementación B-node. El método usado para renovar un nombre NetBIOS es único en los tipos de nodos que utilizan un servidor de nombres NetBIOS. WINS es una extensión de la RFC 1001 y 1002.

El siguiente gráfico muestra el proceso de resolución de nombres NetBIOS:



Name Registration

Cada cliente WINS está configurado con la dirección IP de un servidor WINS primario y opcionalmente de uno secundario. Cuando el cliente arranca, registra su nombre y dirección IP en los servidores WINS. El servidor WINS almacena el nombre NetBIOS y su dirección IP es su base de datos.

Name Renewal

Todos los nombres NetBIOS están registrados en una base temporal, de esta manera el mismo nombre NetBIOS puede ser usado posteriormente por un ordenador diferente si el original deja de usar ese nombre.

Name Release

Cada cliente WINS es responsable de mantener su nombre registrado. Cuando el nombre ya no va a ser usado, por ejemplo cuando apagamos un ordenador, el cliente WINS envia un mensaje al servidor WINS para liberar ese nombre.

Name Query and Name Resolution

Despues de que un cliente WINS ha registrado su nombre NetBIOS y su dirección IP en un servidor WINS, puede comunicarse con otros *hosts* obteniendo la dirección IP de otros ordenadores NetBIOS desde el servidor WINS.

Todas las comunicaciones WINS se hacen usando *datagramas* UDP a través del puerto 137 (*Netbios Name Service*).

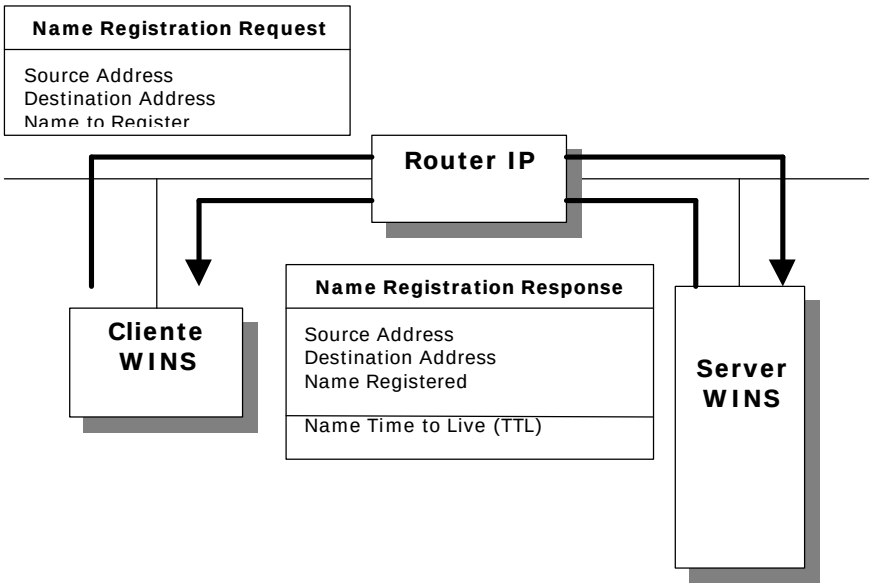
Vamos a ver en detalle cada una de estos 4 puntos:

Name Registration.

Al contrario que en la implementación B-node de NetBIOS sobre TCP/IP con *broadcast* para el registro de los nombres, los clientes WINS registran su nombre NetBIOS en los servidores WINS directamente.

Cuando se inicia un cliente WINS, registra su nombre NetBIOS enviando un mensaje de petición de registro de nombre directamente al servidor WINS que tiene configurado. Los nombres NetBIOS se registran cuando se arrancan los servicios o las aplicaciones, como por ejemplo *Workstation, Server* y *Messenger.*

Si el servidor WINS está disponible y el nombre no está registrado por otro cliente WINS, se devuelve al cliente un mensaje de registro correcto del nombre. Este mensaje contiene el tiempo de vida (TTL) en el cual el nombre va a estar disponible en el servidor.



Cuando se encuentra un nombre duplicado

Si hay un nombre duplicado en la base de datos de WINS, el servidor WINS intenta comunicarse con el propietario del nombre actualmente registrado. El servidor WINS reintenta este proceso hasta 3 veces con 500 milisegundos de intervalo.

Si el ordenador registrado con ese nombre es un ordenador con varios adaptadores de red (*multihomed computer*) el servidor WINS intenta el proceso en cada dirección IP hasta que recibe respuesta o hasta que ha agotado todas las direcciones IP.

Si en este proceso, el propietario registrado responde correctamente al servidor WINS, este enviará una respuesta negativa al registro de nombre del cliente que ha intentado registrar el mismo nombre. Si el propietario del nombre no responde al servidor WINS, el servidor envía una respuesta correcta a la petición de registro y registra correctamente ese nombre en su base de datos.

Cuando el servidor WINS no está disponible.

Un cliente WINS realiza tres intentos (usando ARP) para encontrar el servidor WINS primario. Si falla estos tres intentos se envía la petición al servidor WINS secundario si el cliente estuviese configurado así. Si no hubiese ningún servidor disponible, el cliente iniciaría un *broadcast* para registrar su nombre tal y como hemos visto en los capítulos anteriores.

Name Renewal

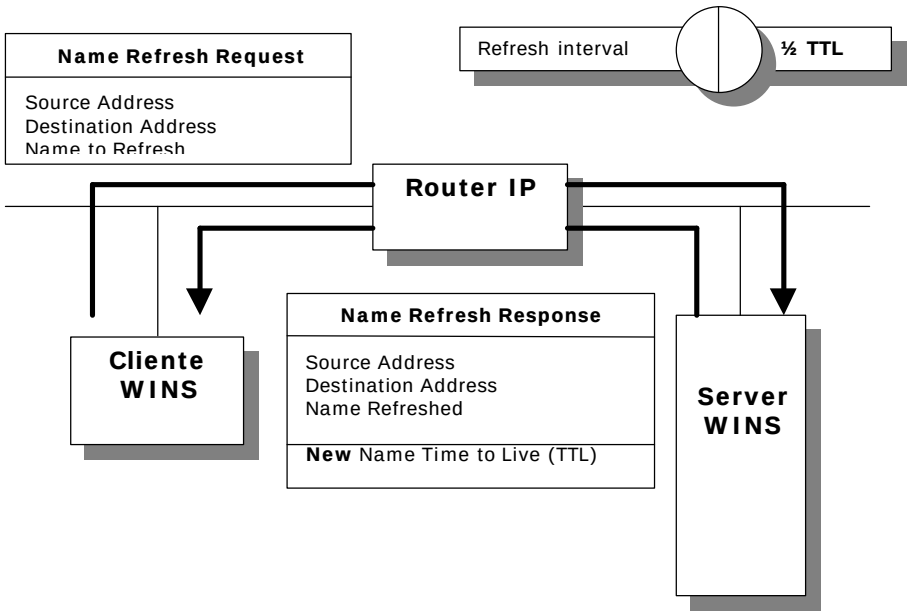
Para poder seguir usando su nombre, el cliente debe renovar el mismo antes de que el tiempo asignado (tiempo de vida) por el servidor expire. Si el cliente no renueva este nombre, el servidor WINS lo dejará disponible para otro posible cliente WINS.

Petición de refresco de nombre (*Name Refresh Request*)

Un cliente WINS intenta primero "refrescar" su nombre después de agotar un octavo de su tiempo de vida. Si el cliente WINS no recibe una respuesta correcta, intentará refrescar su nombre cada 2 minutos hasta que la mitad de su tiempo de vida (TTL) se haya cumplido.

En ese momento el cliente WINS intentará refrescar su nombre en el servidor WINS secundario si estuviese configurado así el cliente (con servidor WINS secundario).

Después de que el cliente haya "refrescado" su registro de nombre una primera vez, las siguientes peticiones de refresco no se realizaran hasta que se haya completado la mitad del tiempo de vida.



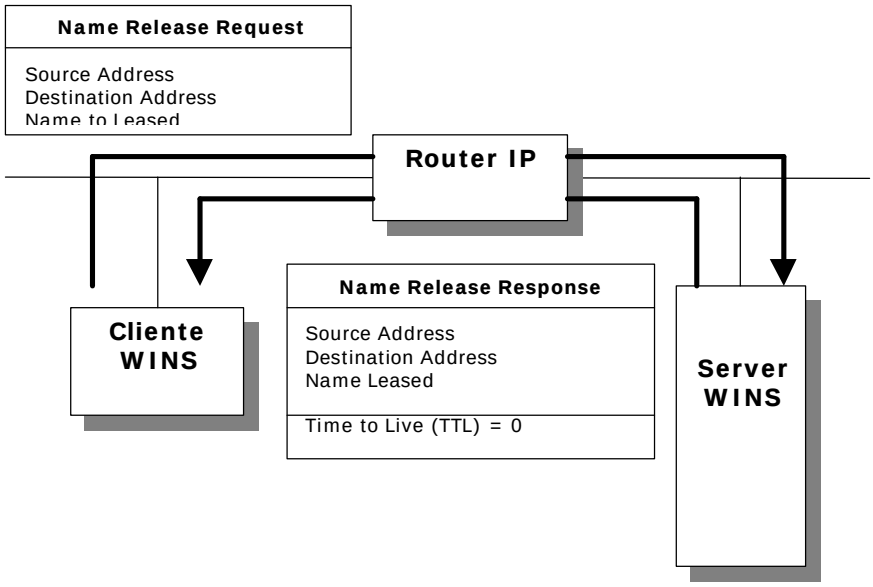
Respuesta de petición de refresco de nombre (*Name Refresh Response*)

Cuando un servidor WINS recibe una petición de refresco de nombre envía al cliente una respuesta asignándole el nuevo tiempo de vida (TTL).

Name Release

Name Release Request

Cuando un cliente WINS se desconecta de la red o se apaga su ordenador de una manera normal, envía una petición para liberar el nombre al servidor WINS por cada nombre que tenga registrado. La petición de liberar nombre incluye la dirección IP y el nombre NetBIOS del cliente que solicita ser removido de la base de datos WINS. Esto dejará ese nombre disponible para otros clientes WINS tal y como vemos en el siguiente gráfico:



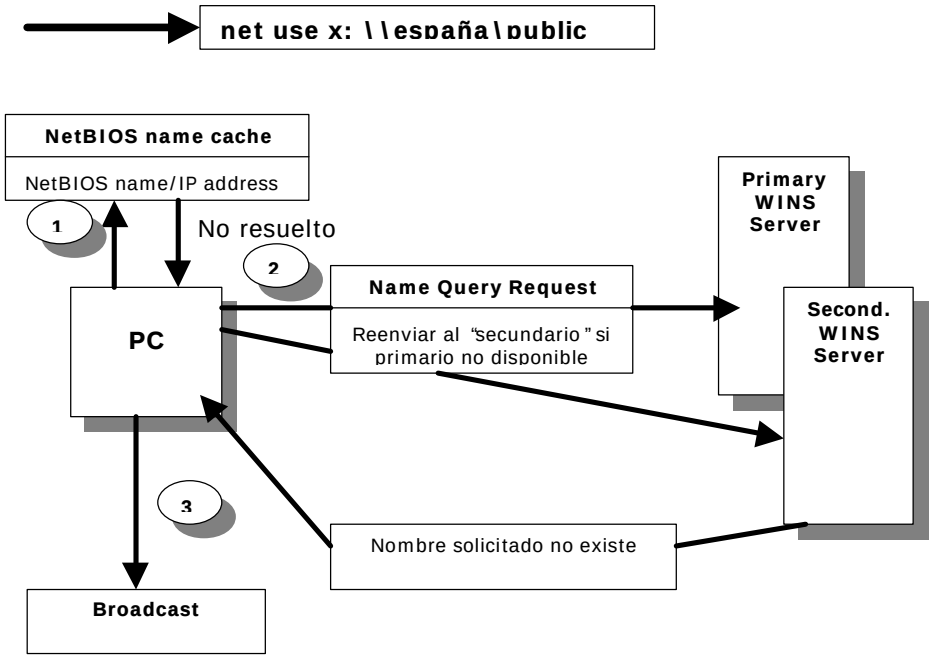
Name Release Response

Cuando el servidor WINS recibe una petición de liberar un nombre, chequea su base de datos para ese nombre. Si WINS encontrase un error en su base de datos o una direccion IP diferente, enviará una respuesta negativa a la petición del cliente.

En otro caso, el servidor WINS envia una respuesta positiva a la liberación de ese nombre y deja inactivo el nombre en su base de datos. El mensaje de respuesta contiene el nombre NetBIOS liberado y un TTL igual a cero.

Name Query and Name Response

Un método común de resolver nombres NetBIOS en direcciones IP es con un servidor de nombres NetBIOS como por ejemplo WINS. Cuando configuramos un cliente WINS, por defecto queda configurado como un nodo de tipo H-node de NetBIOS sobre TCP/IP. Siempre se chequeará para localizar un nombre NetBIOS / dirección IP al servidor de nombres NetBIOS antes de iniciar un *broadcast*. Los siguientes pasos nos ilustran el proceso:



1) Cuando un usuario inicia un comando como por ejemplo **net use** (o indirectamente lo inicia el sistema por ejemplo para ver los ordenadores en una red al abrir el "entorno de red"), el nombre NetBIOS se intenta localizar en la caché de NetBIOS / direcciones IP del propio ordenador.

2) Si el nombre no puede resolverse en la caché, una petición *'name query request'* es enviada directamente por el cliente al servidor WINS primario. Si el servidor primario no estuviese disponible, el cliente reenvía la petición dos veces más y posteriormente repite el ciclo con el servidor WINS secundario. Si el nombre puede resolverse el servidor reenvía un mensaje con la dirección IP solicitada al cliente.

3) Si los servidores WINS no pueden resolver el nombre, el cliente PC enviará una petición *broadcast* a la red para intentar resolver la dirección buscada.

Recordemos que si el nombre no es resuelto por un servidor WINS, o por un *broadcast* el nombre podría quedar resuelto mirando en los ficheros LMHOSTS o HOSTS, o bien usando un DNS (*Domain Name System)*.

ENTORNO DE RED y FUNCIONES DE DOMINIO

En capítulos previos hemos visto la resolución de nombres NetBIOS usando el fichero LMHOSTS y WINS. Vamos a ver ahora como pueden "verse" (*browsing*) los recursos NetBIOS en una red TCP/IP. En este capítulo vamos a ver los siguiente procesos: visualización de recursos NetBIOS, dominio de *logon*, cambio en la cuenta de usuario y sincronización de dominios.

ACERCA DE LA VISUALIZACION (*browsing*)

Para compartir recursos eficientemente en una red. Los usuarios deben ser capaces de encontrar que recursos están disponibles. Por ejemplo en Windows NT tenemos el servicio de *Computer Borwser* y en Windows 2000, Windows 95 y Windows 98, tenemos el icono de "Entorno de Red" en el escritorio.

Los servicios citados anteriormente, son una serie de listas de los recursos disponibles de red. Estas listas son distribuidas a ordenadores especialmente designados al efecto y permiten ver los servicios al resto de los ordenadores de una red.

Los ordenadores designados como *browsers* eliminan la necesidad de que 'todos' los ordenadores mantengan una lista de 'todos' los servicios compartidos en una red. Asignando el papel de *browser* a ordenadores específicos, el servicio de *Computer Borwser*, o bien, el "Entorno de Red", es capaz de resolver estas peticiones y se minimiza la cantidad de tráfico requerido para mantener una lista de todos los recursos compartidos.

Los tipos de *browsers* difieren de acuerdo con sus papeles:

Papel del ordenador	Función
Master Browser	(examinador principal). Es el ordenador encargado de coleccionar una lista de todos los servidores en el dominio o en el grupo de trabajo y la lista de otros dominios y grupos de trabajo. Tambien es encargado de la distribución de esta lista (*browse list*) a los servidores de backup (*backup browsers*).
Backup Browser	Es el ordenador que recibe una copia de la lista generada por el *Master Browser*. Es capaz de distribuir esta lista a los clientes mediante petición.
Domain Master Br.	El *Domain Master Browser* 'tiene un papel adicional. Si hay otros *'master browser'* para este dominio en redes remotas este sincronizará la *browse list* en todos los *master browser* del dominio.

Los ordenadores ejecutando Windows 2000, Windows NT, Windows 95 / 98 Windows para trabajo en Grupo, pueden actuar como *master browser* o como *backup browser*. Solo un ordenador con Windows 2000 Server o Windows NT Server actuando como PDC (*Primary Domain Controller)* puede asumir el papel de *domain master browser.*

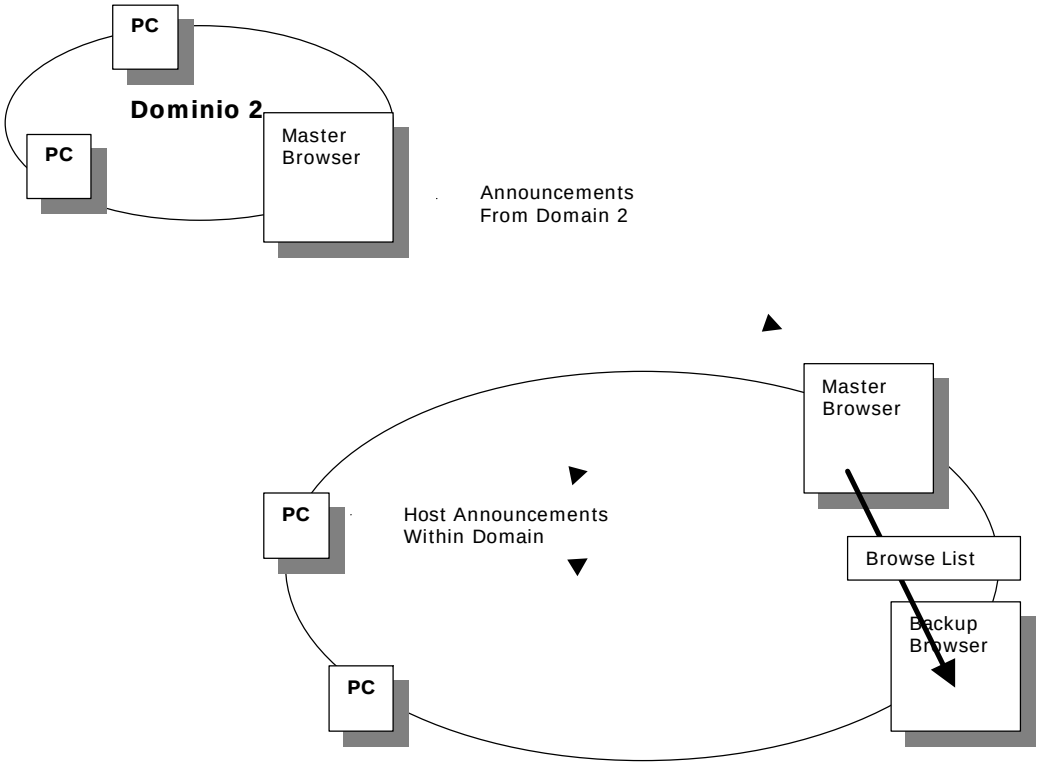
Browsing Collection y su distribución.

Los servicios de visualización (*browsing)* en Windows, pueden entenderse en tres claves de procesos:

- Coleccionando la información de visualización
- Distribuyendo la información de visualización.
- Sirviendo a las peticiones de visualización de los clientes.

El proceso de colección de información de visualización.

El proceso de colección de la información es realizado por el ordenador que actúa como *master browser.* El *master browser* (examinador principal) colecciona información en su lista de visualización (*browse list*) tal y como podemos ver en los gráficos siguientes. Esta información incluye una lista de servidores con su dominio o grupo de trabajo y una lista de otros dominios o grupos de trabajo.



El proceso de distribución.

El proceso de distribución ocurre cuando la lista de visualización (*browse list)* recuperada en el proceso de colección de información se distribuye a los clientes bajo petición. El proceso de distribución, conlleva también:

- *Master Browser Announcement.*

  Periódicamente el examinador principal (*master browser)* envía un anuncio mediante *broadcast* a la red de su existencia mediante un paquete *'master browser announcement'*. Este paquete informa a los *backup browsers* que el *master browser* todavía existe. Si el *master browser* dejase de responder, se inicia un proceso de elección para seleccionar un nuevo *master browser.*

- *Browse List Pull Operation* desde el examinador principal al examinador de backup.

  Periódicamente, cada examinador de backup (*backup browser)* contacta con el examinador principal (*master browser)* en el dominio y descarga la lista (*browse list)* que está guardada en el examinador principal.

Atendiendo a las peticiones de los clientes.

Una vez que la lista está construida en el examinador principal (*master browser)* y distribuida a los examinadores de backup (*backup browser)*, está todo preparado para comenzar el servicio a las peticiones de los clientes. El proceso es el siguiente:

1) Cuando un cliente intenta acceder a un dominio o grupo de trabajo desde el Explorador de Windows NT o desde en Entorno de Red', contacta con el *master browser* del dominio o grupo de trabajo que quiere visualizar.

2) El *master browser* envía al ordenador peticionario una lista de tres *backup browsers.*

3) El cliente entonces, solicita la lista de recursos de red a uno de los *backup browsers.*

4) El *backup browser* responde a la petición del cliente con una lista de los servidores (NT, 2000, o 95 / 98) en ese dominio o grupo de trabajo.

5) El cliente selecciona un servidor y le solicita la lista de los servicios compartidos en ese servidor (puede ser otro Windows NT, 2000 o 95 / 98).

Browsing EN UNA RED IP

El servicio de *browser* (o desde el Entorno de Red en w95 / w98) utiliza *broadcast* NetBIOS para obtener las listas de recursos de red.

Debido a que las peticiones *broadcast* NetBIOS no pueden atravesar un *router* es importante que los *hosts* estén configurados para utilizar WINS o un fichero LMHOSTS para poder ver la actividad en un dominio o en las subredes. Podemos resolver los problemas de *browser* usando WINS o un fichero LMHOSTS. Sin embargo si nuestro router puede reenviar peticiones *broadcast* de nombres NetBIOS (no confundir con *routing* puro de IP), no será necesario usar WINS o el fichero LMHOSTS.

Soluciones de *router* de IP

Algunos *routers* pueden ser configurados para enviar *broadcast* desde una subred IP a otra. Si el *router* está configurado para reenviar estas petición *broadcast* NetBIOS, el servicio de *browsing* trabajará siempre –como si todos los dominios o grupos de trabajo estuviesen localizados en la misma subred. Todos los examinadores principales (*master browser)* serán capaces de ver todos los servicios en sus dominios o grupos de trabajo y en todos los otros dominios o grupos de trabajo y todos los clientes podrán realizar correctamente sus peticiones.

Sin embargo el reenvio de paquetes *broadcast (forwarding broadcast)* no está recomendado a causa de que propaga todo el trafico NetBIOS sobre TCP/IP alrededor de toda la red disminuyendo por tanto el rendimiento de todos los nodos de la red.

Soluciones Windows NT - 2000

Típicamente los *routers* IP no están configurados para reenviar peticiones NetBIOS. Por ello, para ver, coleccionar, distribuir y el servicio de peticiones del cliente son bajo trafico directo IP en vez de bajo trafico de *broadcast* (difusión). Hay dos vias para solucionar el problema:

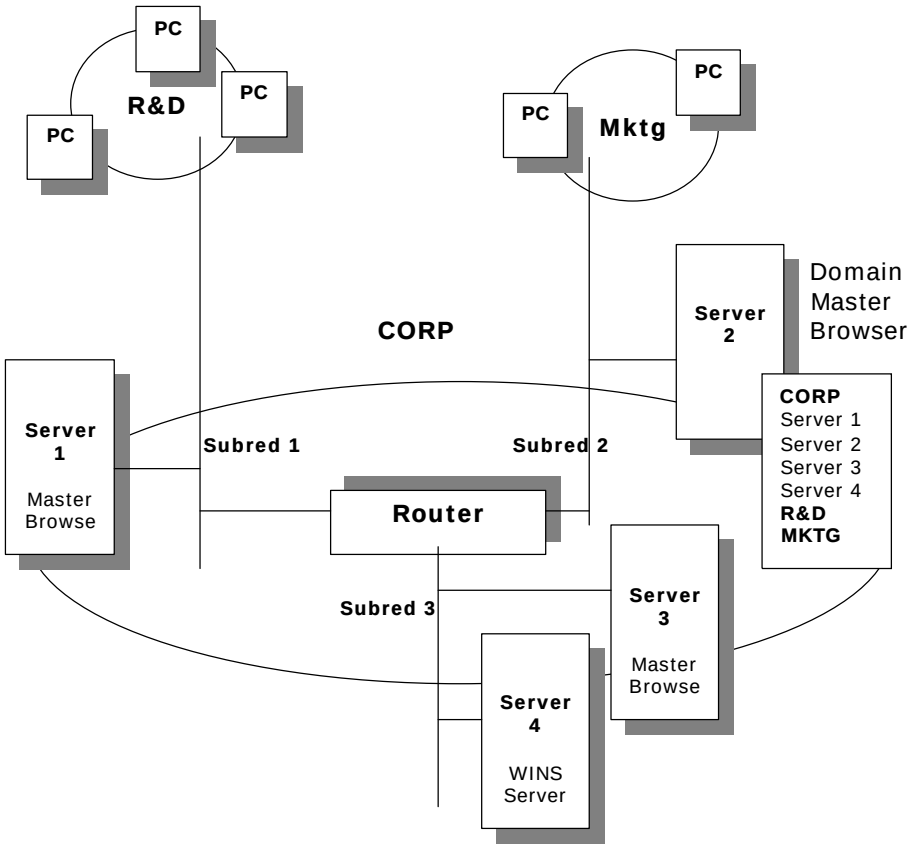
- WINS. Usado para coleccionar la lista de servicios de visualización bajo peticiones de los clientes.
- LMHOSTS. Las entradas especiales en el fichero LMHOSTS van a ayudar a facilitar la distribución de información de visualizacion y el servicio de *browsing* de los clientes.

Browsing con WINS

WINS soluciona los problemas de peticiones *broadcast* NetBIOS registrando dinámicamente los nombres NetBIOS y sus direcciones IP de los ordenadores en dicha subred y almacenándolos en una base de datos WINS. Cuando los clientes WINS comunican bajo TCP/IP hacia las subredes, el IP del *host* de destino se recupera de la base de datos WINS en lugar de utilizar *broadcast.*

Una mejora que WINS añade a este mecanismo de coleccionar nombres de dominios o grupos de trabajo es que un examinador principal de visualización (*doamin master browser)* ejecutándose como un cliente WINS va a preguntar periódicamente al servidor WINS por la lista de todos los dominios listados en la base de datos WINS.

La ventaja de *browsing* bajo WINS es que el examinador principal (*doamin master browser)* para un determinado dominio, ahora posee una lista de todos los dominios incluyendo las que estén en redes remotas.

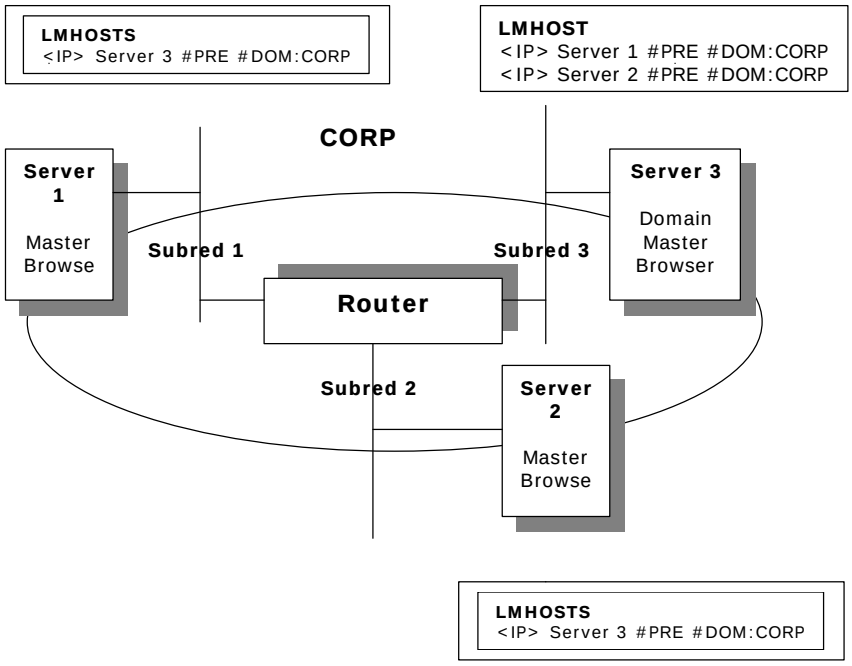


Nota: La lista de dominios obtenida a través de una pregunta a WINS contiene solo los nombres de dominios y sus correspondientes direcciones IP, pero no incluye los nombres de los examinadores principales que 'anuncian 'estos dominios.

Browsing Usando el fichero LMHOSTS

Para implementar comunicaciones directas entre subredes, clientes no-WINS pueden usar *broadcast* para el registro de nombres NetBIOS y su resolución requiere el fichero LMHOSTS. El fichero debe ser configurado con la dirección IP y el nombre NetBIOS de los controladores de dominio localizados en otras subredes.

Para comunicación directa entre los examinadores principales (*master browser)* en redes remotas y el *domain master browser* el fichero LMHOSTS debe estar configurado con los nombres y direcciones IP de los examinadores principales de las otras redes tal y como vemos en el gráfico adjunto:



Master Browser

Para los ordenadores ejecutando Windows, el fichero LMHOSTS de cada examinador principal en cada subred debe contener la siguiente información:

- Dirección IP y nombre del ordenador que es el *domain master browser.*
- El nombre del dominio precedido por **#PRE #DOM:**

Por ejemplo:

126.20.3.81 <*domain master_browser*> #PRE #DOM:<*domain_name*>

Domain Master Browsers

El fichero LMHOSTS del examinador principal del dominio debe estar configurado con las entradas de cada uno de los examinadores principales de las redes remotas.

Cada examinador principal debe tener una entrada **#DOM** para cada uno de los otros examinadores principales en el dominio. De esta manera si un examinador principal es promocionado a examinador principal del dominio los ficheros LMHOSTS no necesitaran ser cambiados en los otros examinadores principales.

Cuando existen múltiples entradas en el fichero LMHOSTS para el mismo nombre de dominio, el examinador principal determina que entradas corresponden al examinador principal del dominio enviando un pregunta a la dirección IP de cada entrada. Únicamente el examinador principal del dominio responderá. El examinador principal que contacta con el examinador principal del dominio procederá a intercambiar las listas de *browsing.*

RESOLUCIÓN DE NOMBRES (*Host Name Resolution)*

Esquemas de nombres en TCP/IP

Aún cuando los *hosts* TCP/IP requieren una dirección IP para comunicarse, los *hosts* pueden ser referenciados por un nombre en lugar de su dirección IP.

Existen varios esquemas diferentes de nombres usados por *hosts* Windows y UNIX. Un *host* Windows puede tener asignado un nombre *host*, pero este nombre *host* se utiliza solo con utilidades TCP/IP. Los *hosts* UNÍS requieren solo una dirección IP. El usar un nombre *host* o nombre de dominio para comunicarse, es opcional.

Antes de que la comunicación tenga lugar, es necesario siempre tener la dirección IP de cada *host* TCP/IP que interviene en la comunicación. Por tanto, el esquema de nombres afecta a la vía en la cual el *host* es referenciado. Por ejemplo:

- Para poder usar un comando **net use** entre dos ordenadores ejecutando Windows, el usuario, siempre especifica su nombre NetBIOS en lugar de una dirección IP, por ejemplo:

  net use x: \\nombre_de_ordenador

  El nombre NetBIOS debe poderse resolver como una dirección IP antes de que el ARP pueda resolver la dirección IP en una dirección hardware.

- Para referenciar un *host* UNIX ejecutando TCP/IP, el usuario especifica una dirección IP, un nombre de *host* o un nombre de dominio. Si se utiliza el nombre de *host* o el nombre de dominio, el nombre debe resolverse como una dirección IP. Si se utiliza una dirección IP, la resolución de nombres no es necesaria y la dirección IP se resuelve a dirección hardware.

La principal diferencia en la vía de referenciar los dos tipos de *host* es que debemos comunicar siempre usando nombres NetBIOS con los comandos de la red Microsoft y no direcciones IP. Usando utilidades TCP/IP para referenciar los *hosts* UNIX se permite usar la dirección IP.

Nota: Windows permite conectar con otro ordenador ejecutando también Windows utilizando la dirección IP. Por ejemplo:
net use x: \\137.121.2.213\nombre_compartido

En resumen: Windows y UNIX utilizan diferentes esquemas de nombres. Windows y otros sistemas basados en redes Microsoft requieren un nombre NetBIOS para comunicarse con otros ordenadores Windows.

Nombres de *HOST*

Un nombre de *host* simplifica la madera de referenciar a una maquina debido a que los nombres son más fáciles de recordar que las direcciones IP. Los nombres de *host* se utilizan en prácticamente todos los entornos TCP/IP. Vamos a ver en esta parte, como funciona la resolución de nombres.

Un nombre de *host* es una alias asignado a un ordenador por un administrador del sistema para identificar un *host* TCP/IP. El nombre *host* no tiene por qué coincidir con el nombre NetBIOS del ordenador, y puede ser cualquier cadena de caracteres de hasta 256 de longitud. Múltiples nombres de *host* pueden ser asignados a la misma maquina.

Un nombre de *host* simplifica la manera en que un usuario referencia otros *host* TCP/IP. Los nombres de *host* son mas fáciles de recordar que las direcciones IP. De hecho, un nombre de *host* puede ser usado en vez de una dirección IP cuando utilizamos PING u otras utilidades TCP/IP.

Un nombre de *host* siempre corresponde a una dirección IP que está almacenada en el fichero HOSTS o en una base de datos en un DNS o servidor de nombres NetBIOS. Windows utiliza además el fichero LMHOSTS para *'mapear '*nombres de *hosts* en direcciones IP.

En Windows NT (o en Windows 2000) la utilidad HOSTNAME nos mostrará el nombre asignado a nuestro sistema. El nombre *host* es el nombre del ordenador cuando estamos ejecutando Windows.

Resolución de Nombres *Host*

La resolución de nombres *host* es el proceso de traducir un nombre *host* en una dirección IP. Antes de que la dirección IP pueda ser resuelta a dirección hardware, el nombre de *host* debe ser resuelto a dirección IP.

Windows puede resolver los nombres de *host* usando varios metodos. Estos metodos los estuvimos anticipando en el capítulo anteriormente visto sobre "NetBIOS sobre TCP/IP".

El TCP/IP de Microsoft puede utilizar cualquiera de los metodos mostrados en las siguientes tablas para resolver nombres de *host.* Los metodos que Windows (NT o 2000) pueden usar para resolver un nombre de *host* son configurales.

Métodos estándar de resolución	**Descripción**
Local host name	El nombre de *host* configurado para esa maquina. Este nombre se compara siempre con el nombre de destino.
Fichero HOSTS	Un fichero de texto local en el mismo formato que el fichero UNIX\Etc\Host *4.3 Berkeley Software Distribution (BSD)*. Este fichero convierte nombres *host* en direccion IP. Este fichero se utiliza fundamentalmente por las utilidades TCP/IP.
Domain Name System (DNS)	Un servidor que mantiene una base de datos de direcciones IP / nombres *host*.

Métodos Windows de Resolución	**Descripción.**
NetBIOS *Name Server* (NBNS)	Un servidor que cumpliendo las definiciones dadas en las RFCs 1001 y 1002 dan la resolución de nombres NetBIOS. La implementación de este servidor por Microsoft, es WINS.
Local Broadcast	Llamada mediante *broadcasting* a la subred local para la localización de la dirección IP del nombre NetBIOS de destino.
Fichero LMHOSTS	Un fichero de texto en local que traduce las direcciones IP a nombres NetBIOS en las redes Microsoft Windows.

Resolviendo nombres mediante un fichero HOSTS.

Al contrario que el fichero LMHOSTS al cual es utilizado únicamente para resolver *hosts* remotos, el fichero HOSTS convierte nombres de *hosts* tanto locales como remotos en sus direcciones IP. Tal y como vemos en el siguiente gráfico, el proceso es el siguiente:

1) La resolución de nombres comienza cuando un usuario utiliza un comando usando el nombre de *host* para el ordenador destino.

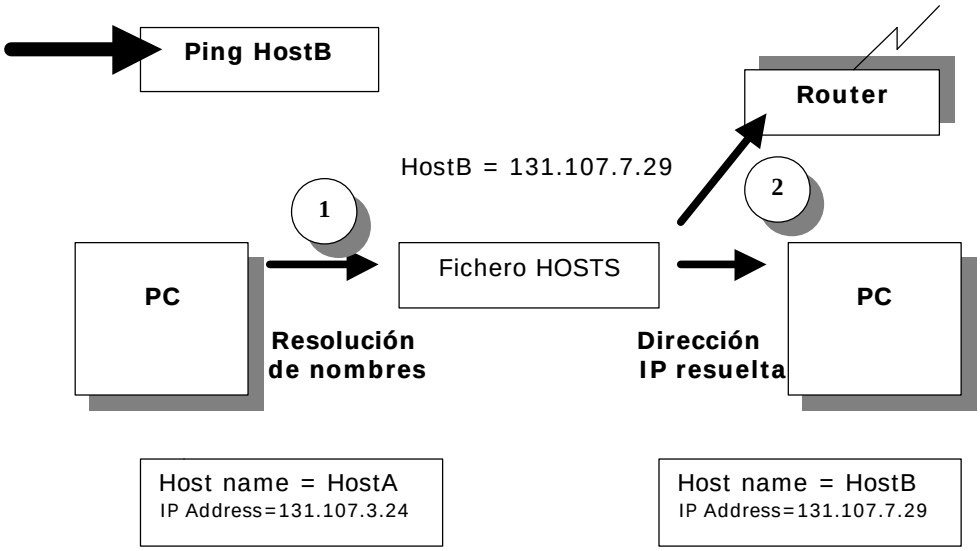
   Windows comprueba primero si el nombre de *host* es el mismo que el nombre del propio ordenador local. Si los nombres no son iguales, intenta localizar la existencia de un fichero HOSTS. Si el nombre buscado está en dicho fichero, automáticamente se tomará de él la dirección IP.

   Si el nombre de *host* no puede ser resuelto y no existen otros métodos de resolución, como DNS o un servidor de nombres NetBIOS o un fichero LMHOSTS, el proceso se detiene y el usuario recibe un mensaje de error.

2) Después de haber resuelto el nombre *host* en una dirección IP, se debe resolver el IP del destino en una dirección hardware.

   Si el *host* destino está en la red local, mediante ARP se obtiene su dirección hardware, bien consultando la caché ARP o bien mediante *broadcasting* de la dirección IP del destino.

   Si el *host* destino está en una red remota, el ARP obtiene la dirección hardware de un *router* y la petición es encaminada hacia el *host* destino.



Resolviendo nombres con un servidor DNS.

Un servidor *Domain Name System* (DNS) es una base de datos online centralizada que se utiliza en entornos UNIX para resolver un nombres de dominios completamente calificados (*Fully Qualified Domain Names:* FQDNs) y otros nombres de *hosts* en direcciones IP. Windows puede utilizar un servidor DNS y además Windows NT 4 y Windows 2000 pueden darnos los servicios de servidor DNS. La resolución de nombres de dominio utilizando un servidor DNS es muy similar a utilizar un fichero HOSTS.

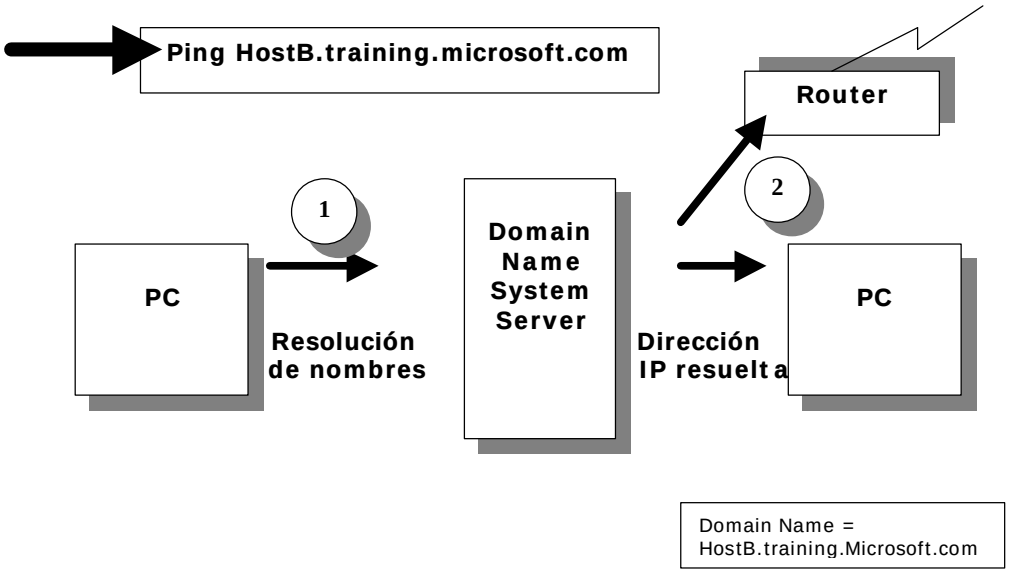
Si Windows está configurado para resolver nombres de *host* usando un servidor DNS, utiliza dos pasos para resolver el nombre, tal y como vemos en el siguiente proceso:

1) Cuando un usuario usa un comando usando un FQDN o un nombre de *host,* el servidor de DNS busca el nombre en su base de datos y resuelve este a una dirección IP.

   Si el servidor DNS no responde a la petición, se realizan intentos adicionales en intervalos de 5, 10, 20, 40, 5, 10 y 20 segundos. Si el servidor DNS no responde a ninguno de estos intentos y no existen otros metodos de resolución configurados, como servidor de nombres NetBIOS o fichero LMHOSTS, el proceso se detiene y nos informa de un error.

2) Después de haber sido resuelto el nombre de *host*, mediante ARP se obtiene la dirección hardware. Si el *host* destino está en la red local, ARP obtiene su dirección hardware bien mediante consulta a la caché ARP o bien mediante *broadcasting* a la red local. Si el *host* destino está en una red remota, ARP obtiene la dirección hardware de un *router* que pueda reenviar la petición.

   Si el servidor DNS está en una red remota, ARP deberá obtener la dirección hardware de un *router* antes de que el nombre pueda ser resuelto.



Métodos Microsoft de resolución de nombres de *hosts.*

Windows puede ser configurado para resolver nombres de *host* utilizando un servidor de nombres NetBIOS, *broadcast*, y LMHOSTS además del fichero HOSTS y el servidor de DNS. Si uno de estos métodos falla se utiliza el siguiente metod como si fuese un backup, tal y como mostramos en el siguiente ejemplo:

Si NBND y LMHOSTS están configurados, el orden de resolución es el siguiente:

1) Cuando un usuario utiliza un comando usando un nombre de *host*, Windows mira primeramente a ver si el nombre de *host* es el mismo que el nombre del ordenador local. Si son los mismos, el nombre está resuelto y el comando es ejecutado sin generar actividad de red.

2) Si el nombre de *hosts* y el nombre local no son los mismos, se intenta localizar el fichero HOSTS y resolver en él la dirección IP del destino. Si el nombre del *host* se encuentra en el fichero HOSTS estará resuelta su dirección IP. El fichero HOSTS debe residir en el ordenador local.

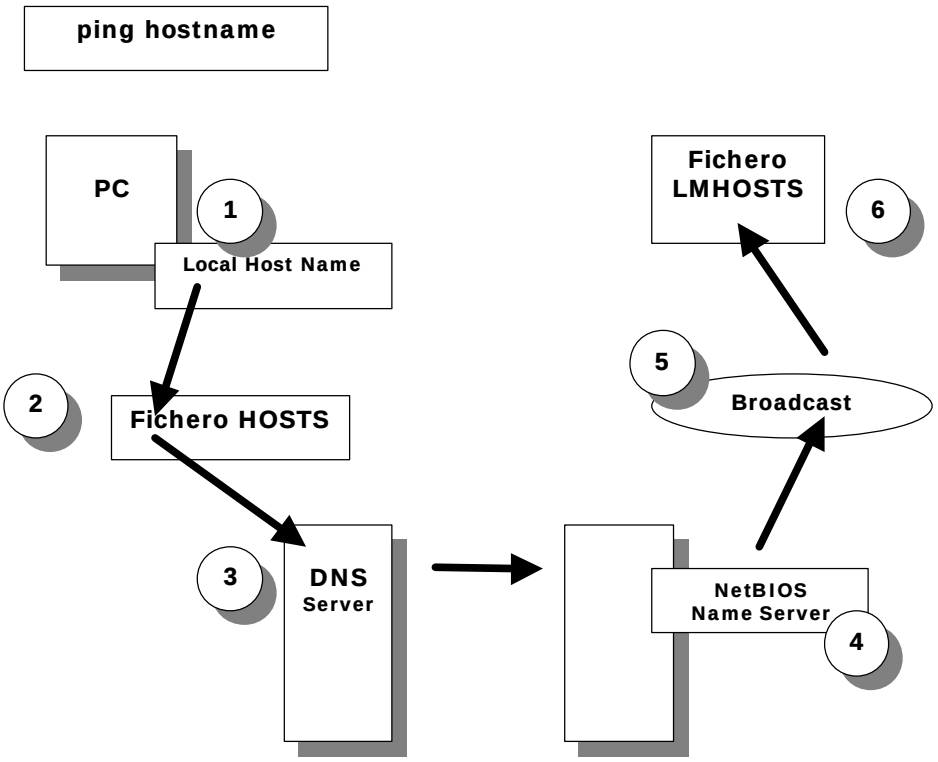
3) Si el nombre de *host* no puede ser resuelto utilizando el fichero HOSTS, se envia una petición al servidor de DNS. Si el nombre del destino se encuentra en un servidor DNS se resuelve a su dirección IP y la resolución de direcciones ha resultado correcta.

   Si el servidor DNS no responde a la petición, se realizan intentos adicionales en intervalos de 5, 10, 20, 40, 5, 10 y 20 segundos.

4) Si el servidor de DNS no puede resolver el nombre de *host*, el ordenador local mira a ver en la caché de nombres NetBIOS antes de realizar 3 intentos de contactar con el servidor de nombres NetBIOS que tenga configurado. Si el nombre del destino se encuentra en la caché de nombres NetBIOS o es localizado por un servidor de nombres NetBIOS, se resolverá a una dirección IP, y el proceso de resolución ha finalizado.
5) Si el nombre de *host* no es resuelto por el servidor de nombres NetBIOS, el ordenador origen envía 3 mensajes *broadcast* a la red local. Si el nombre del destino se encuentra en la red local, se resolverá a una dirección IP y el proceso de resolución ha finalizado.

6) Si el nombre del *host* no se resuelve utilizando *broadcast*, se intenta localizar el fichero LMHOSTS. Si el nombre del destino se encuentra en el fichero LMHOSTS, se resolverá a una dirección IP y el proceso de resolución ha finalizado.

Si ninguno de estos métodos resuelve el nombre de *host* la única manera de comunicarse con el otro *host* es especificando su dirección IP.



Resumen

Un nombre de *host* se utiliza para identificar un *host* TCP/IP o un *default gateway*. La resolución de nombres *host* es el proceso de convertir el nombre *host* en una dirección IP. Esto es necesario antes de que el ARP pueda resolver la dirección IP en una dirección hardware.

El Fichero HOSTS

Acabamos de ver como los nombres de *hosts* son convertidos a direcciones IP utilizando diferentes métodos. Vamos a ver el fichero HOSTS.

El fichero HOSTS es un fichero estático que si existe debe estar en el directorio directorio de Windows (en Windows 95 o 98) o bien en el caso de Windows NT o Windows 2000 en *systemroot*\Systen32\Drivers\Etc. Este fichero nos suministra la compatibilidad con el fichero HOSTS de UNIX. El fichero HOSTS es usado por por las utilidades estándar TCP/IP como por ejemplo el PING que necesitan resolver un nombre de *host* en una dirección IP tanto en redes locales como remotas. El fichero HOSTS también puede ser usado para resolver nombres NetBIOS (específico del TCP/IP-32 de Microsoft).

Un fichero HOSTS puede residir en cada ordenador. Una entrada sencilla consiste en una entrada IP y su correspondiente nombre o nombres de *hosts*. Por defecto el nombre de *host* localhost ' es una entrada estándar en dicho fichero.

El fichero HOSTS es utilizado cuando un nombre de *host* es utilizado o referenciado. Los nombres se leen secuencialmente por lo que en ficheros grandes, los nombres normalmente más utilizados deben estar al comienzo de dicho fichero.

Nota: El fichero HOSTS puede ser editado. Está localizado en el directorio de Windows en el caso de Windows 95 o Windows 98, y en el directorio *Systemroot*\System32\Drivers\Etc en el caso de Windows NT o Windows 2000.

Cada entrada *host* está limitada a 255 caracteres y las entradas en dicho ficero no distinguen mayúsculas de minúsculas, no son *'case sensitive'*.

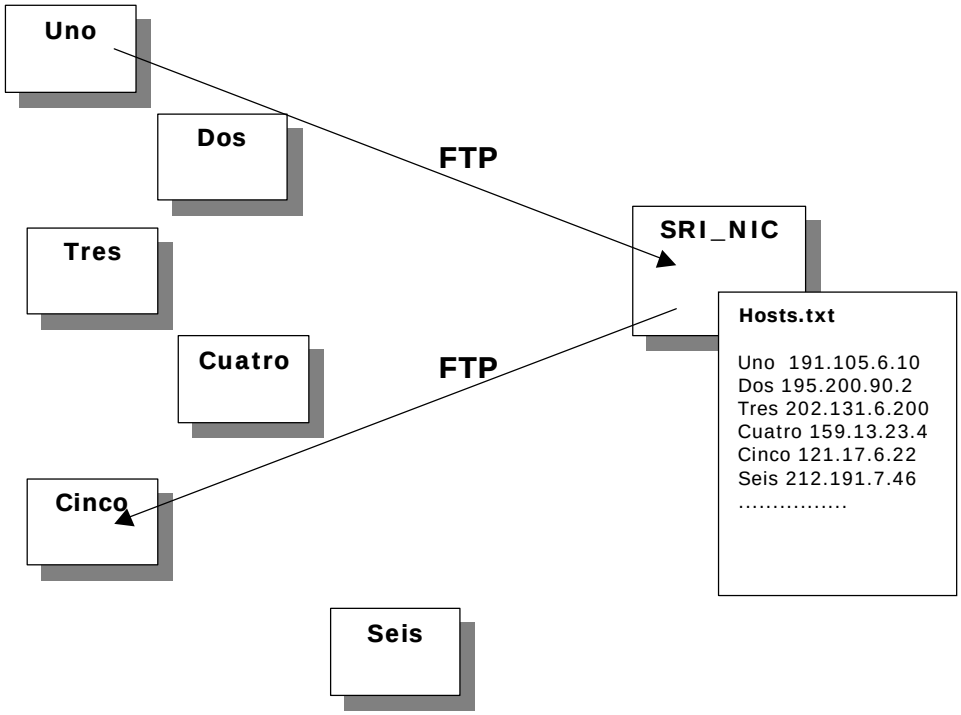
DOMAIN NAME SYSTEM (DNS)

En este capítulo vamos a ver la estructura y componentes del DNS: *Domain Name System*, incluyendo como resolver direcciones TCP/IP, como configurar los archivos del DNS, y como registrar un servidor DNS con el dominio.

Domain Name System (DNS)

El DNS es similar a un listín telefónico.

Antes de 1980, la red ARPANET tenía unicamente unos pocos cientos de ordenadores. El nombre de ordenador y su direccion estaba contenido en un simple fichero llamado Hosts.txt. Este fichero estaba almacenado en el ordenador de *Stanford Research Institute´s Network Information Center* (SRI-NIC) en *Menlo Park, California.* Tal y como vemos en el siguiente dibujo, el resto de ordenadores de ARPANET copiaban el fichero Hosts.txt desde el SRI-NIC a los sitios en que fuese necesario.



Inicialmente este esquema funcionaba bien debido a que la lista necesitaba actualizarse solamente una o dos veces a la semana. Si embargo en unos pocos años surgieron problemas debido al aumento de tamaño de ARAPNET. Estos problemas eran:

- El fichero Hosts.txt empezó a ser demasiado extenso.
- Empezó a ser necesario de más de una actualización diaria.
- Debido al tráfico de red hacia y desde el SRI-NIC, el mantenimiento del fichero Hosts.txt empezó a ser un cuello de botella en la propia red.
- El tráfico de red en el SRI-NIC se volvió inmanejable.
- El fichero Hosts.txt, utilizaba la estructura þplana'de nombres (*name space*). Esto implicaba que el nombre del ordenador debía ser único en toda la red.

Estos y otros problemas de otro tipo de envergadura obligaron a ARPANET a intentar encontrar otras soluciones al mecanismo que giraba alrededor del fichero Hosts.txt. La decisión tomada fue la creación del *Domain Name System* (DNS), el cual es una base de datos distribuida usando un estructura de nombres 'jerárquicos' (*hierarchical name space*).

Nota: El *Domain Name System* está dewscrito en las RFCs 1034 y 1035.

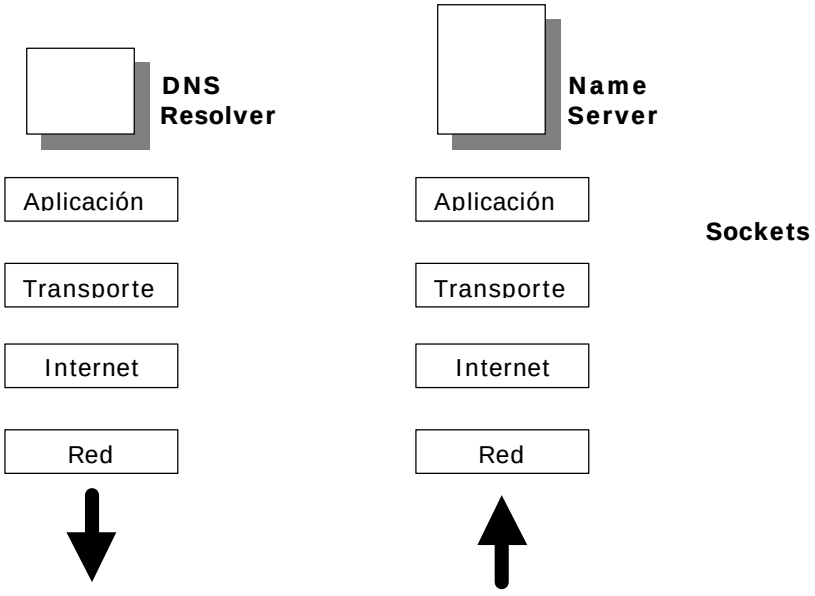
Como trabaja el DNS.

El DNS trabaja utilizando tres componentes principales: resoluctores (*resolvers)*, servidores de nombres (*name servers*) y el espacio de nombres de dominios (*domain name space*).

Con la comunicación básica DNS, un cliente DNS, o *resolver*, envía peticiones a un servidor de nombres. El servidor de nombres devuelve la información solicitada, o apunta a otro servidor de nombres, o bien un mensaje de error si la petición no puede ser resuelta.

El *Domain Name System* es una base de datos jerárquica basada en el estructura cliente / servidor. El DNS proyecta las direcciones a la capa de aplicación y utiliza UDP y TCP como protocolos de base.

El propósito de la base de datos del DNS es convertir nombres de ordenador en direcciones IP tal y como vemos en el siguiente gráfico. A nivel de DNS, los clientes son llamados *'resolvers'* y los servidores son llamados *'name servers'*.



El *Domain Name System* es similar a un listín telefónico. El usuario mira el nombre de la persona u organización que está buscando y las referencias cruzadas del nombre al número de teléfono.

Los *resolvers* primero envían peticiones UDP a los servidores para incrementar el rendimiento y reenvían únicamente en TCP si ocurre un truncamiento de datos en los datos recibidos.

Resolvers

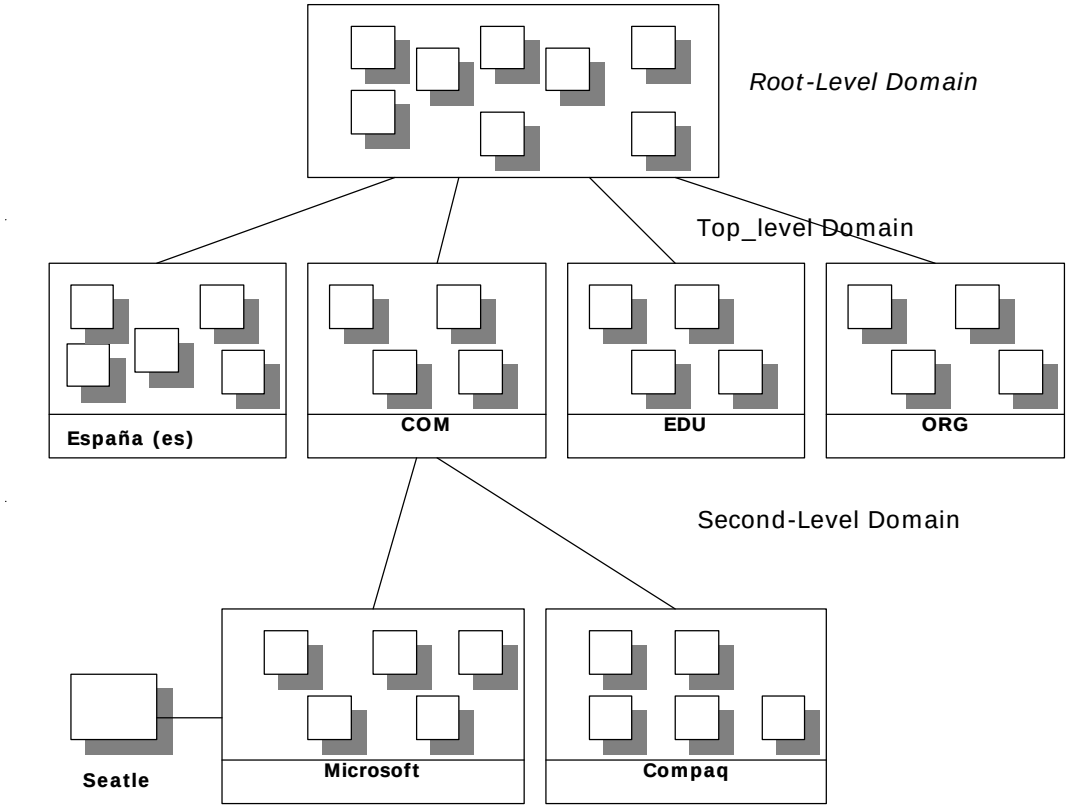
La función de los *resolvers* es pasar las peticiones de nombre entre las aplicaciones y los servidores de nombres (*name servers*). La petición de nombre contiene una pregunta. Por ejemplo, la pregunta puede interrogar sobre la dirección IP de un sitio Web. El *resolver* puede estar incorporado dentro de la aplicación, o lo que es más normal, estar ejecutándose en el ordenador como una rutina del sistema.

Name Servers

El *Name Server* acepta las peticiones de nombres desde los *resolvers* y resuelven el nombre de ordenador (o de dominio) a direcciones IP. Si el *name server* no es capaz de resolver la petición, puede reenviar la petición a un *name server* que sea capaz de resolverla. Los *name servers* están agrupados en diferentes niveles que son llamados **dominios**.

Domain Name Space

El *domain name space* es la agrupación jerárquica de nombre en forma de camino invertido tal y como podemos ver en la siguiente ilustración:



Root-Level Domains

Los dominios definen diferentes niveles en una estructura jerárquica. El punto más alto de la jerarquía es el llamado dominio *root*. El dominio *root* utiliza una etiqueta nula, pero las referencias a dicho dominio pueden ser expresadas por un punto (**.**).

Top-Level Domains

Los descritos a continuación están presentes en los *top-level domains*.

- **com** Organizaciones comerciales.
- **edu** Instituciones educativas y universidades.
- **org** Organizaciones sin animo de lucro.
- **net** Redes
- **gov** Organizaciones gubernamentales no militares.
- **mil** Organizaciones gubernamentales militares.
- **num** Listines telefónicos.

- **arpa** DNS reservados
- ***xx*** Dos letras del código de país.

Los dominios de primer nivel o *top-level* pueden contener dominios de segundo nivel o *second-level* así como *hosts.*

Nota: La *Internet Society* está considerando otros varios adicionales dominios de primer nivel como por ejemplo: **.firm** y **.web**.

Second-Level Domains.

Los niveles de segundo nivel (*second-level*) pueden contener ambos: *hosts* y otros dominios llamados sub-dominios. Por ejemplo, el dominio Microsoft (microsoft.com) puede contener ordenadores como ftp.microsof.com y subdominios como dev.microsoft.com. El sub-dominio dev.microsoft.com puede contener *hosts* como por ejemplo: ntserver.dev.microsoft.com.

Nombres de *hosts.*

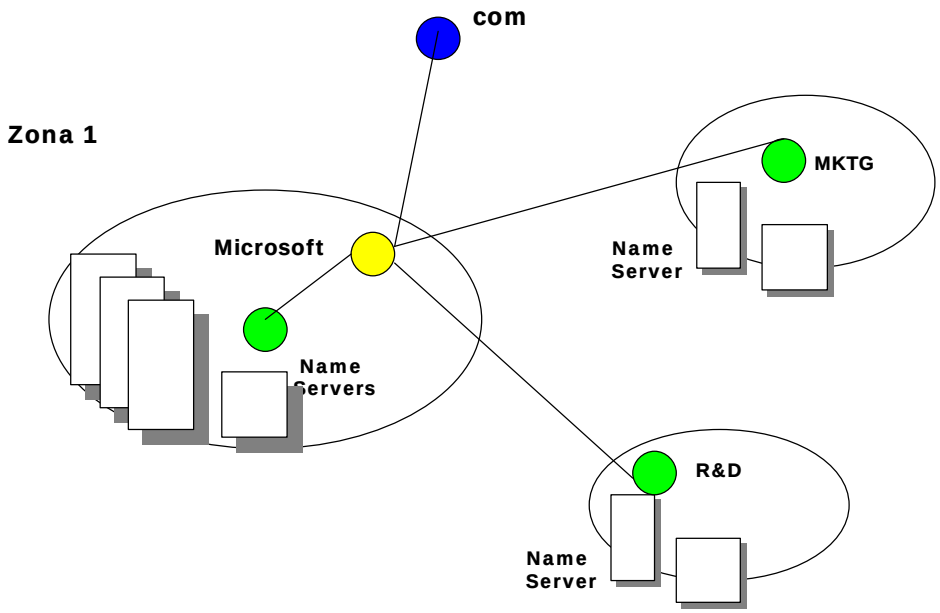
Los nombres de *hosts* dentro de los dominios, son añadidos al comienzo del nombre del dominio y esto constituye su nombre completo (*Full Qualified Doamin Name* o FQDN). Por ejemplo, un *host* llamado fileserver ' en el dominio microsoft.com debe tener el nombre cualificado (FQDN) de fileserver.microsoft.com.

Zonas de Autoridad.

Una zona de autoridad (*zone of authority*) es la porción del *Domain Name Space* de la que es responsable un determinado servidor de nombres (*name server*). El servidor de nombres almacena todas las direcciones y sus direcciones IP para la zona correspondiente a ese *domain name space* y responde a las preguntas de los clientes ante estos nombres.

La zona de autoridad del servidor de nombres abarca al menos un dominio. A este dominio se le referencia como la ¿zona del dominio principal 'La zona de autoridad también puede incluir subdominios. Por tanto, una zona puede no contener necesariamente todos los subdominios bajo la zona del dominio principal.

En el siguiente ejemplo, microsoft.com es un dominio, pero el dominio entero no está controlado por un fichero de zona. Parte del dominio está localizado en en un fichero separado de zona para dev.microsoft.com. Multiples ficheros de zona pueden ser necesarios para la distribución y el control del dominio en diferentes grupos o por eficiencia en replicación de datos.



Un único DNS puede ser configurado para manejar una o múltiples zonas. Cada zona está imbuida en un dominio especifico llamado el ¡domino de zona principal'.

Papeles del *Name Server*

Los servidores de nombres (DNS) pueden ser configurados para diferentes papeles. Los DNS pueden almacenar y mantener sus bases de datos de nombres por medio de diferentes vías. Cada uno de los siguientes papeles describe una diferente vía en la cual un servidor de nombres puede ser configurado para almacenar sus datos de zona.

Primary Name Servers

El servidor de nombres primario obtiene los datos de la zona desde ficheros locales. Cambios en la zona, como añadir dominios o *hosts*, se realizan en el nivel del servidor de nombres primario.

Secondary Name Servers

Un servidor de nombres secundario obtiene los datos para las zonas desde otro servidor de nombres de red que tenga autoridad para esa zona. El obtener esta información de zona en la red lo denominamos transferencia de zona (*zone transfer*).

Hay tres razones para tener servidores de nombres secundarios:

- Redundancia. Se necesita al menos un servidor primario y uno secundario para cada zona. Los ordenadores que lo contengan deben ser tan independientes como sea posible.
- Acceso más rápido para localizaciones remotas. Si tenemos un número de clientes en localizaciones remotas, teniendo servidores de nombres secundario (u otro primario para los subdominios) nos impide que estos clientes se comuniquen lentamente a través de enlaces para la resolución de nombres.
- Reducción de carga. Los servidores secundarios de nombres reducen la carga del primario.

Debido a que la información de cada zona se almacena en ficheros separados, esta definición de primario o secundario es definida a nivel de zona. En otras palabras, un servidor de nombres particular puede ser servidor de nombres primario para ciertas zonas y servidor de nombres secundario para otras zonas.

Master Name Servers

Cuando definimos una zona en un servidor de nombres como una zona secundaria, debemos designar otro servidor de nombres desde el cual obtener la información de zona. La fuente de la información de zona para un servidor de nombres secundario en un DNS jerárquico es denominada *master name server.* Un servidor de nombres maestro (*master name server*) puede ser servidor de nombres primario o secundario para la zona referenciada. Cuando un servidor de nombres secundario arranca, este contacta con el servidor de nombres maestro e inicia una transferencia de zona con este servidor.

Caching-Only Servers

Son los servidores DNS que las preguntas que tienen resueltas están en caché. *Caching-Only Servers* son servidores de nombres DNS que solo permiten preguntas, buscan la respuesta en la caché, y devuelven los resultados. En otras palabras: no están autorizados para ningún dominio (los datos de zona no están guardados localmente) y solo contienen información que tiene en memoria para resolver las preguntas.

Cuando intentamos determinar cuanto uso tiene un servidor, debemos recordar que cuando el servidor es arrancado inicialmente no tiene información en caché y debe construir esta información en el momento en que existe una petición de servicio.

RESOLUCIÓN DE NOMBRES (*Name Resolution*)

Hay tres tipos de preguntas que un cliente (*resolver*) puede hacer a un servidor DNS: recursiva, iterativa e inversa.

Preguntas recursivas.

En una pregunta recursiva, se le solicita una respuesta al servidor de nombres interrogando con los datos pedidos o con un error indicando que los datos de la petición no existen o que el nombre del dominio no existe. El servidor de nombre no puede en este caso diferir la petición a otro diferente servidor de nombres y debe intentar por todos los medios resolverla.

Preguntas iterativas.

En una pregunta iterativa, el servidor de nombres envía la mejor respuesta que puede o que conoce en ese momento al peticionario. Esta respuesta puede ser la resolución del nombre, o puede referirse a otro servidor de nombres que sea capaz de responder al cliente original de la petición.

La siguiente ilustración muestra un ejemplo de ambas preguntas: recursiva e iterativa. En este ejemplo, un cliente en una corporación, está preguntando al servidor de DNS por la dirección IP de www.whitehouse.gov.

1) El *resolver* envía una pregunta DNS recursiva al servidor local DNS preguntando por la dirección IP de www.whitehouse.gov. El servidor local de nombres, es responsable de resolver el nombre y no puede referirse a otro servidor de nombres para resolverlo.

2) El servidor de nombres local chequea sus zonas, y no encuentra zonas correspondientes a la petición de dicho dominio. Este entonces, envía una pregunta iterativa para www.whitehouse.gov al servidor principal de nombres (*root name server*).

3) El servidor principal de nombres, tiene autoridad para el dominio principal (*root domain*) y va a responder con la dirección IP del servidor de nombres para el dominio de mas alto nivel .gov.

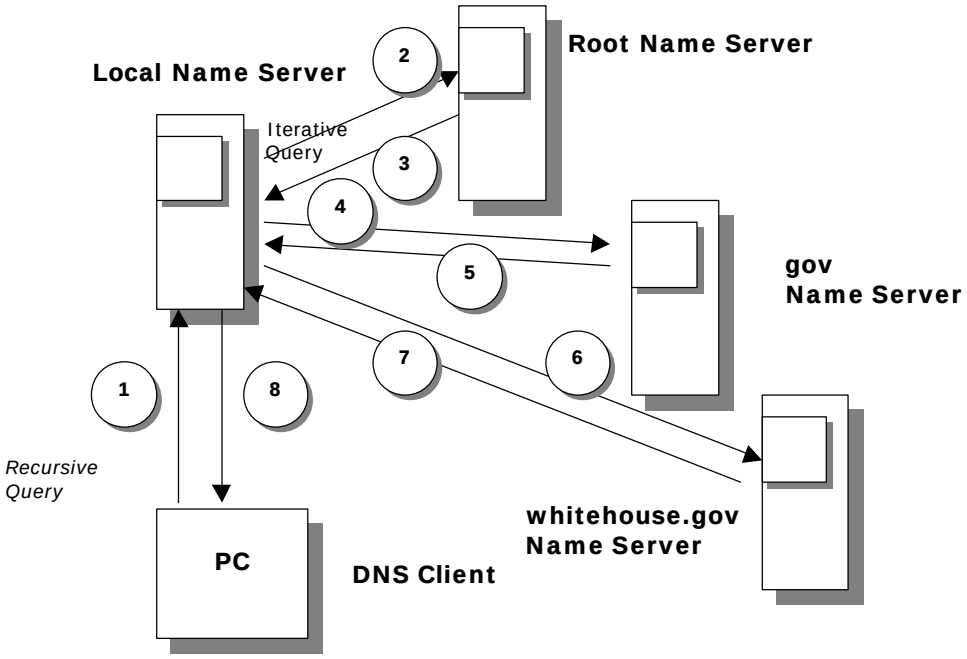
4) El servidor de nombres local envía una pregunta iterativa para www.whitehouse.gov al servidor de nombres .gov.

5) El servidor de nombres .gov devuelve la dirección IP del servidor de nombres que está dando servicio al dominio whitehouse.gov.

6) El servidor de nombres local, envía una pregunta iterativa para www.whitehouse.gov al servidor de nombres whitehouse.gov.

7) El servidor de nombres whitehouse.gov devuelve la dirección IP correspondiente a www.whitehouse.gov.

8) El servidor de nombres local envía la dirección IP de www.whitehouse.gov al cliente (*resolver*) original.



Preguntas inversas.

En una pregunta inversa, el *resolver* envía envía una petición al servidor de nombres para resolver el nombre de *host* asociado a una determinada dirección IP. No hay correlación entre direcciones IP y nombres de *hosts* en el espacio de nombres DNS. Por tanto, solo una búsqueda en todos los dominios garantizará una respuesta correcta.

Para prevenir una búsqueda exhaustiva en todos los dominios en una pregunta inversa, ha sido creado un dominio especial llamado 'in-addr.arpa'. Los nodos en el dominio 'in-addr.arpa' están nombrados después de la numeración (con puntos, en representación decimal) de la dirección IP. Debido a que la numeración IP es más específica leyéndola de izquierda a derecha y los nombres de dominios son menos específicos de izquierda a derecha, el orden de los octetos de la dirección IP deben ser invertidos cuando se construye el dominio 'in-addr.arpa'. De esta manera, la administración en un nivel más bajo puede ser delegada a organizaciones que tienen asignadas sus clases de direcciones IP: A, B o C.

Una vez que el dominio 'in-addr.arpa' está construido, unos registros especiales llamados *pointer records* (PTR) son añadidos para asociar las direcciones IP con el correspondiente nombre de *host*. Por ejemplo, para encontrar el nombre correspondiente a la dirección IP 157.55.200.51, el *resolver* pregunta al servidor 'in-addr-arpa' de DNS por un *pointer record* para 51.200.55.157. El PTR encontrado contiene el nombre de *host* y su correspondiente dirección IP 157.55.200.51. Esta información se envía al *resolver*.

Caching y TTL

Cuando un servidor de nombres está procesando una pregunta recursiva, puede ser necesario enviar varias preguntas para encontrar la respuesta. El servidor de nombre, lleva en caché toda la información durantes el proceso durante un tiempo que está especificado en los datos devueltos. Esta cantidad de tiempo de vida la denominamos *Time to Live* (TTL). El administrador del servidor de nombres de la zona que contiene el dato es el que decide el TTL para ese dato. Valores pequeños del TTL, asegurarán que el dato en el dominio será más consistente en la red si este dato cambiase. Esto incrementa la carga en los servidores locales.

Una vez que el dato está en la caché del DNS, este debe estar decrementando el valor del TTL desde su valor original para decidir cuando el dato es inválido y borrarlo de la caché. Si una pregunta de un cliente puede ser satisfecha con el dato en caché, el TTL que se devuelve con el dato contiene el valor que le resta en ese servidor de nombres. Los clientes (*resolvers*) también tienen caché de datos y verifican el valor del TTL para conocer cuando ese dato es válido.

Configurando los ficheros DNS

Hay cuatro ficheros de configuración para un servidor de nombres típico de DNS.

Un servidor de nombres DNS típico, tiene una base de datos (*database file*), fichero inverso (*reverse lockup file),* un fichero de caché (*cache file*) u fichero de inicio (*boot file*). Estos ficheros de configuración permiten una variedad de funciones en el servidor.

Database File

El fichero de base de datos (Zone.dns) almacena los registros de recursos para un dominio. Por ejemplo, si nuestra zona es 'microsoft.com ', este fichero tendrá el nombre de 'microsoft.com.dns '.

Windows NT 4, nos da un fichero de ejemplo llamado Place.dns ' como una plantilla con la cual podemos trabajar. Este fichero puede ser editado y renombrado antes de usarlo como un fichero de producción en un servidor DNS. Es en general una buena idea que el nombre de este fichero sea el mismo que la zona que representa. Este es el fichero que va a ser replicado entre el *master name server* y el *secondary name server.*

Hay varios tipos de recursos de registros definidos en un DNS. La RFC 1034 define los tipos de registro: SOA, A, NS, PTR, CNAME, MX y HINFO. Microsoft ha añadido los registros específicos WINS y WINS-R.

Registro de Autoridad.

El primer registro en cualquier base de datos debe ser el ***S****tart* ***O****f* ***A****uthority* (SOA). El SOA define los parámetros generales para la zona DNS. Este es un ejemplo de un registro SOA:

```
@ IN SOA nameserver1.microsoft.com.  glennwo.microsoft.com.  (
            1           ; número de serie
            10800       ; refresco (3 horas)
            3600        ; reintentos (1 hora)
            604800      ; expiración (7 días)
            86400 )     ; Tiempo de vida. TTL. (1 día)
```

Las siguientes reglas se aplican a los registros SOA:

- El símbolo arroba (@) en un fichero de base de datos indica: "este servidor ".
- IN indica un registro Internet.
- Cualquier nombre de host no terminado con un punto (**.**) será añadido en el dominio principal.
- El símbolo @ es reemplazado por un punto (**.**) en la dirección de e-mail del administrador.
- Deben utilizarse paréntesis **()** para encerrar las líneas que ocupan más de una línea física.

Name Server Record

El registro del servidor de nombres (NS) lista los adicionales servidores de nombres. Un fichero de case de datos puede contener mas de un registro NS. Lo siguiente es un ejemplo de registro:

```
@ IN NS nameserver2.microsoft.com
```

Host Record

Un registro *Host* (A) asocial estáticamente un nombre de *host* con su dirección IP. Los siguientes son ejemplos de registros *hosts*:

```
rhino      IN  A  157.55.200.143
localhost  IN  A  127.0.0.1
```

CNAME *record*

Un registro de nombre cacónico 'Canonical Name' (CNAME) permite asociar mas de un *host* con una dirección IP. Esto algunas veces se le denomina 'alias'. El siguiente es un ejemplo del registro CNAME:

```
FileServer 1      CNAME   rhino
www               CNAME   rhino
ftp               CNAME   rhino
```

Nota: Los tipos de registro de la base de datos, están definidos en las RFCs 1034, 1035 y 1183.

Reverse Lockup File

El fichero de direcciones inversas (z.y.x.w.in-addr.arpa) permite a un *resolver* el dar una dirección IP y pedir el nombre del *host* que posee esa dirección. Un fichero de *reverse lockup* se le llama igual que a un fichero de zona en la zona *in-addr.arpa*

Por ejemplo, para dar el nombre para la dirección IP 157.57.28.0 se crea un fichero de *reverse lockup* con el nombre del fichero 57.175.in-addr.arpa. Este fichero contiene registros SOA y *name server* similares a los otros ficheros de base de datos de DNS, así como registros PTR.

La capacidad de *reverse-lockup* del DNS es importante debido a que algunas aplicaciones nos dan las capacidades para implementar seguridad basado en los nombres de *hosts* que pueden conectarse. Por ejemplo, un cliente intenta un enlace a un volumen NFS (*Network File System*) que tiene montado el mecanismo de seguridad. El servidor NFS debe contartar con el servidor DNS y realizar un *reverse lockup* del nombre para la dirección IP que quiere conectarse. Si el nombre del *host* devuelto por el DNS no está en la lista de accesos del volumen NFS o si el nombre del *hosts* no es localizado por el DNS, el NFS denegará la conexión.

El *Pointer Record*

Los registros de tipo apuntador (PTR) nos dan una conversión 'dirección' a 'hombre' en una zona de *reverse lockup*. Los números IP están escritos en orden inverso y además, 'in-addr.arpa' debe ser añadido para crear este registro apuntador. Como un ejemplo, la dirección 157.200.200.51 requiere un apuntador para el nombre '51.200.55.157.in-addr.arpa'. Es decir:

```
51.200.55.157.in-addr.arpa.   IN  PTR   mailserver1.Microsoft.com.
```

El fichero Caché

El fichero Cache.dns contiene los registros de los servidores del dominio principal (*root*). El fichero caché es esencialmente el mismo en todos los servidores de nombres y debe estar presente. Cuando el servidor de nombres recibe una pregunta de fuera de su zona, este comienza la resolución con esos servidores de dominio principal.

Este es un ejemplo de contenido de dicho fichero:

```
.                          3600000  IN    NS    A.ROOT.SERVERS.NET.
A.ROOT.SERVERS.NET         3600000  A           198.41.0.4
```

El fichero caché contiene información que es necesaria para resolver nombres fuera de los dominios autorizados. Contiene nombres y direcciones de los servidores de dominio principal (*root name servers*). El fichero suministrado por defecto en Windows NT 4 Server contiene todos los registros de los *root servers* de Internet. Para instalaciones no conectadas a Internet, este fichero debe ser reemplazado para contener el nombre de los servidores *root* de la red privada.

Nota: para obtener un fichero actualizado de caché, puede descargarse desde ftp://rs.internic.net/domain/named.cache

---

El fichero *Boot*

El fichero *boot* es el fichero de configuración de arranque en un servidor DNS *Berkeley Internet Name Deamon* (BIND). Este fichero contiene la información necesaria para resolver nombres fuera de los dominios autorizados. Este fichero no está definido en ninguna RFC y no es necesario por tanto cumplir ninguna RFC. El servidor de DNS de Windows NT 4 puede ser configurado para usar un fichero *boot* en vez de utilizar el administrador de DNS.

El fichero *boot* controla el arranque del servidor DNS. Los comandos deben comenzar al principio de la línea y no pueden ser precedidos por ningún espacio. Los comandos reconocidos son: **directory**, **cache**, **primary** y **secondary**.

La sintaxis del fichero *boot* se muestra en la siguiente tabla:

Comando	Descripción
Directory	Especifica el directorio en donde van a estar los otros ficheros de configuración.
Cache	Especifica el fichero usado para ayudar al DNS a encontrar los nombres de los servidores del dominio principal. Este comando y el fichero al cual se refieren, deben estar presentes.
Primary	Especifica un dominio para el cual este servidor de nombres está autorizado y un fichero de base de datos que contiene los registros de recursos para el dominio (es decir, la ona del fichero). Pueden existir múltiple registros *'primary'* en el fichero *boot*.
Secondary	Especifica un dominio para el cual este servidor de nombres está autorizado y una lista de servidores maestros de direcciones IP desde los cuales se espera descargar la información de zona, en vez de leerlo desde este fichero. Este tambien define el nombre del fichero local para mantener el caché de esta zona. Pueden existir múltiple registros *'secondary'* en el fichero de *boot*.

La siguiente tabla muestra ejemplos de los comandos en el fichero de *boot*:

Sintaxis	Ejemplo
directory [directorio]	directory c:\winnt\system32\dns
cache. [nombre_fichero]	cache.cache
primary [dominio] [nombre_fichero]	primary microsoft.com.microsoft.dns primary dev.microsoft.com dev.dns
secondary [dominio] [hostlist] [local_file_name]	secondary test.microsoft.com 157.55.200.100 test.dns

Resumen

Cuatro ficheros de configuración son utilizados por un servidor DNS típico. El fichero de bases de datos, almacena registros de recursos para un dominio. Para que el servidor de nombres pueda resolver preguntas inversas, un fichero de *reverse lockup* es necesario. El fichero caché contiene los nombres y direcciones de los servidores de nombres que mantienen el dominio principal (*root*). El fichero *boot* es el fichero de configuración de arranque en un servidor DNS *Berkeley Internet Name Deamond.*

Planificando una implementación del DNS

La configuración de los servidores DNS depende de factores como por ejemplo, el tamaño de nuestra organización, la localización física de la organización, y los requerimientos de tolerancia a fallos (*fault tolerance*).

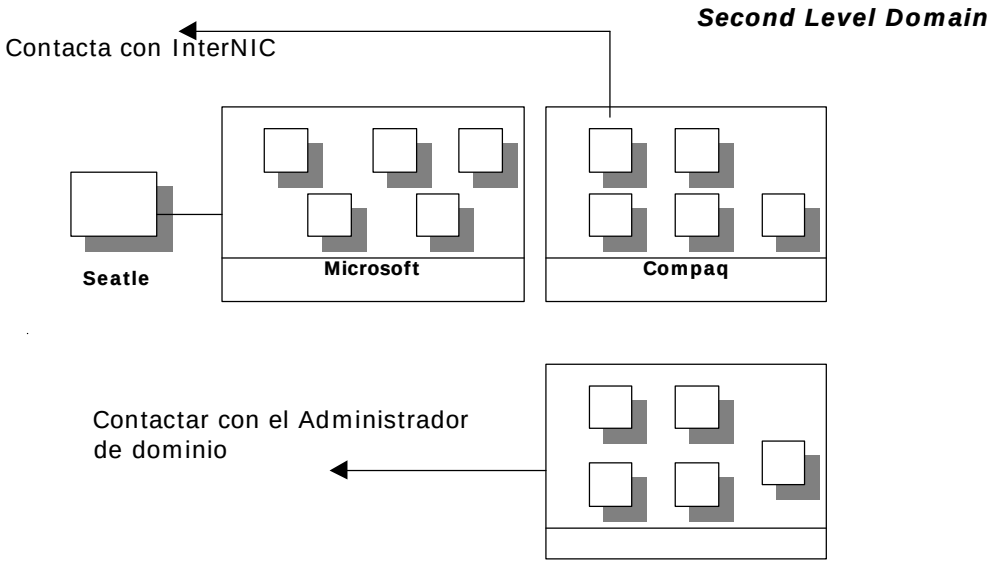
Entes que mantener un servidor DNS, una organización con una pequeña red puede encontrar más simple y más eficiente tener clientes DNS que pregunten al servidor de nombres DNS mantenido por un IPS. Algunos ISP pueden mantenernos información de nuestro dominio gratis o por poco costo. Las organizaciones que quieren controlar su dominio deben mantener sus propios DNS.

Si una organización, mirando su tamaño, quiere conectar en Internet como un dominio de segundo nivel el InterNIC debe ser informado del nombre del dominio de la organización y la dirección IP de al menos dos servidores DNS que den servicio al dominio. Una organización, igualmente puede tener activos servidores DNS en si mismo, independientemente de Internet.

Por redundancia y facilidad, se recomienda que sean configurados al menos dos servidores de DNS por dominio: un servidor de nombres primario y uno secundario. El servidor de nombres primario mantiene la base de datos de información, la cual es replicada en el servidor de nombres secundario. Esta replicación permite dar el servicio de respuestas a las peticiones de nombre, aunque uno de los servidores estuviese indisponible. La planificación de la replicación, puede ser configurada dependiendo de como y cuando cambien los nombres en el dominio. La replicación debe ser frecuente para que los cambios sean conocidos por ambos servidores. Sin embargo, una replicación excesiva puede cargar la red y los servidores de nombres innecesariamente.

Registrando con el dominio padre.

Una vez que tenemos el servidor o servidores DNS configurados e instalados, necesitamos registrar con el servidor de DNS que esté por encima en la estructura jerárquica de nombres. El siguiente gráfico nos muestra un ejemplo de registro de nuestro DNS con el nivel de dominio que esté por encima de él. El padre, en el sistema, necesita el nombre y la dirección de nuestros servidores de nombres y puede requerir algún otro tipo de información, como por ejemplo, la fecha en la cual el dominio va a estar disponible y los nombres y direcciones de correo de las personas de contacto.



Nota: Si queremos registrar como un subdominio o mayor, es conveniente visitar las paginas online de los servios de InterNIC en http://internic.net

Implementado el DNS

Vamos a ver como instalar y configurar un servidor DNS, integrar el DNS y WINS y usar NSLOOKUP (la herramienta de diagnostico del DNS). En este capítulo vamos a ver y configurar un *Domain Name System*, configurar los archivos DNS y usar los servidores DNS para resolver los nombres de *host* en direcciones IP.

El servidor DNS de Microsoft.

Windows NT 4 y Windows 2000 incluyen un servicio estándar de DNS. Vamos a ver la implementación de Microsoft del servidor DNS.

El DNS de Microsoft cumple con la definición dada en las RFC sobre los servidores DNS, por tanto, crea y utiliza los ficheros de zona estándares del DNS y soporta todos los tipos de registros de recursos. Puede ínter operar con otros servidores DNS e incluye una utilidad de diagnóstico del DNS, el NSLOOKUP. El servidor DNS de Microsoft está íntimamente integrado con WINS y puede ser administrado en una utilidad gráfica de administración llamada: ***DNS Manager***.

Instalando el servidor de DNS de Microsoft

Antes de instalar el servicio de servidor de DNS de Windows NT, es importante verificar que el TCP/IP del servidor esté configurado correctamente. El servicio de servidor de DNS obtiene las opciones por defecto del nombre del *host* y nombre del dominio de la caja de dialogo de las propiedades del TCP/IP. El servicio del servidor DNS va a crear los registro por defecto SOA, A, y NS basándose en el nombre especificado allí del dominio y del *host*. Si el nombre del *host* y del dominio no están especificados, únicamente se creará el registro SOA.

Práctica

En esta prática vamos a intalar el servicio de servidor de DNS.

- Para configurar el servicio de servidor de DNS:

  1) Conectarse como Administrador.
  2) En la consola del sistema, teclear: **ipconfig**
  3) Apuntar la dirección IP de nuestro ordenador
  4) In a la caja de dialogo de la propiedades del TCP/IP de Microsoft y pulsar la pestaña DNS
  5) En la caja de dialogo de **dominio**, teclear el nombre de nuestro dominio (por ejemplo **dominio1**)
  6) En **DNS Service Search Order**, pulsar **Add.**
  7) En la caja **DNS Server**, teclear nuestra direccion IP y pulsar **Add.**
  8) Pulsar **OK**. (aparecerá la caja de diálogo **Network**)

- Para instalar el servicio de DNS:

  1) En el Panel de Control, pulsamos un doble-click en el icono de red (*Network)* y entoces pinchamos en **Services**.
  2) Pulsamos **Add**. Aparecerá la caja de dialogo: **Select Network Service**.
  3) En la lista de **Network Service**, pinchamos en **Microsodft DNS Server**, y posteriormente **OK**. En este momento, Windows NT nos mostrará una caja de dialogo preguntándonos por el camino completo de los ficheros de distribución de Windows NT.
  4) Tecleamos el camino de los ficheros de distribución de Windows NT y pulsamos **Continue**. Todos los ficheros necesarios, incluyendo los ficheros de ejemplo seran copiados a nuestro disco duro.
  5) En la caja de diálogo de **Network**, pinchar **Close**.
  6) Cuando nos lo solicite, deberemos reinicar nuestro ordenador.

Solucionando problemas del DNS con NSLOOKUP.

La utilidad NSLOOKUP, la primera herramienta de diagnostico para el DNS, permite a los usuarios interactuar con el servidor DNS. NSLOOKUP puede utilizarse para ver el registro de recursos en un servidor DNS, incluyendo implementaciones DNS de UNIX. NSLOOKUP se instala cuando instalamos el protocolo TCP/IP.

Modos de funcionamiento de NSLOOKUP

NSLOOKUP tiene dos modos de funcionamiento: interactivo y no interactivo. Si únicamente necesitamos un dato, podemos utilizar el modo no interactivo o modo línea de comando'. Si necesitamos más de un dato, debemos usar el modo interactivo.

Sintaxis de NSLOOKUP

nslookup [-opción ...] [ordenador_a_buscar | - [server]]

Sintaxis	**Descripción**
-opción ...	Especifica uno o mas comandos NSLOKKUP. Para ver la lista de comandos podemos usar la opción **Help** dentro de NSLOOKUP.
ordenador_a_buscar	Si el ordenador a busacar en una dirección IP, y el tipo de pregunta es A o PTR, será devuelto el nombre del ordenador.Si el nombre a buscar es un 'nombre' y este no tiene un punto (**.**) intermedio, el nombre del dominio DNS será añadido al nombre. Para buscar un ordenador fuera del dominio del DNS actual, debemos añadir un punto al final del nombre.
server	Utiliza ese servidor como el servidor de nombres DNS. Si este servidor se omite, el servidor que actualmente esté configurado por defecto, será el que se utilice.

- Para utilizar NSLOOKUP en modo de línea de comandos:

  1) En la ventana de comandos, modificar sus propiedades para tener al menos un buffer de pantalla de 50 lineas.

  2) Pulsar el siguiente comando:

     nslookup *hostx*

     en donde *hostx* es un nombre de *host* en nuestro dominio.

     NSLOOKUP nos devolverá la dirección IP de dicho ordenador (si la información estuviese almacenada en la base de datos del DNS).

- Para usar NSLOOKUP en modo interactivo.

  1) En la pantalla de comandos, teclear **nslookup** y pulsar Intro.

     Aparecerá un *'prompt'*.

  2) Teclear **set all** en dicho *prompt*. Este commando lista todos los valores actuales de las opciones de NSLOOKUP.

  3) Utilizando el **Help de Windows NT** y el comando **set** procedermos a cambiar el **time-out** a **1** segundo y el número de reintentos a 7. Utilizar **set all** para verificar los valores por defecto que van a ser cambiados.

     Set ti=1

     Set ret=7

  4) Ir al **DNS Manager** y apuntar el número de *host* en nuestro dominio.

  5) Volver a la pantalla de comandos.

6) Teclear los nombres de los otros ordenadores, uno cada vez, en el símbolo del *prompt*. Debmos pulsar Intro después de cada nombre.

7) Ir al **DNS Manager** y pulsar F5.
Todos los nombres de ordenador que han podido ser resueltos serán añadidos a la base de datos de la zona.

8) Teclear *Exit* (salir) en la ventana de comandos.

Administrando el servidor de DNS.

Existen dos vías para administrar el servidor de DNS: utilizar el *DNS Manager* ' o manualmente editar los ficheros de configuración del DNS.

Configurando las propiedades del servidor de DNS.

Podemos utilizar el *DNS Manager* para configurar el servidor de Windows NT. Debido a que el servidor de DNS no tiene información inicial acerca de nuestra red, el servidor DNS instala un servidor de nombres *caching-only* para Internet. El fichero de la configuración inicial contiene únicamente información de los servidores principales (*root*) de Internet. Para algunas configuraciones de los servidores DNS debemos dar información adicional.

Las propiedades y opciones que tenemos en el *DNS Manager* son las siguientes:

Propiedad	Descripción
Interfaces	Especifica con que adaptadores de red operará el DNS en un ordenador con varios adaptadores de red (*multihomed*). Por defecto se usarán todos los adaptadores.
Forwarders	Configura nuestro servidor para utilizar otro servidor de nombres como un reexpedidor. El servidor de nombres, puede también ser configurado como un esclavo del reexpedidor (*forwarder*).
Boot method	Nos informa del método para iniciarse que el servidor de nombres está utilizando (desde el registro o desde ficheros de datos).

Configurando manualmente el DNS.

El servidor DNS puede ser configurado manualmente editando los ficheros que por defento, en la instalación están en *system_root*\System32\Dns. La administración es idéntica a una administración tradicional de un DNS. Estos ficheros pueden ser modificados utilizando un editor de texto. El servicio de DNS cuando lo estemos editando, debe pararse y rearrancarse.

Añadiendo Dominios y Zonas al DNS.

El primer paso en la configuración del servidor de DNS es determinar la jerarquia de nuestras Zonas y Dominios. Una vez que la información de Zona y Dominio ha sido determinada, esta información debe ser incorporada en la configuración del DNS utilizando el *DNS Manager*.

Añadiendo zonas primarias y secundarias.

Podemos añadir Zonas primarias y secundarias a través del *DNS Manager*. Después de introducir la información de zona, el *DNS Manager* va a crear un fichero con el nombre por defecto de la zona. Si el fichero de zona ya existe en el directorio del DNS, el *DNS Manager* va a importar directamente esos regsitros.

Una zona primaria almacena nombres y direcciones locales. Cuando configuramos una zona primaria, no necesitamos otra información que el nombre de la zona.

Las zonas secundarias, obtienen la traslación de nombres – direcciones IP desde un servidor maestro. Cuando configuramos una zona secundaria, debemos dar el nombre de la zona y un servidor maestro de nombres.

Nota: Windows NT crea un fichero llamado *zonename*.dns que es diferente a los creados por otros servidores DNS que suelen utilizar el nombre: Db.zone.

Añadiendo subdominios.

Una vez que todas las zonas han sido añadidas al servidor, debemos añadir los subdominios por debajo de dichas zonas. Para añadir un subdominio, debemos pinchar en el *DNS Manager* la zona y seleccionamos ***New Domain***. Allí definiremos el nombre del nuevo subdominio.

Si son necesarios múltiple niveles de subdominios, podemos crear cada subdominio de la misma forma que la definida anteriormente, pero colgando justo del anterior nuevo dominio.

Estas claves definidas para cada zona, se escriben en el registro bajo la clave:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Zones

Cada zona tiene su propia clave y la clave contiene el nombre del fichero de base de datos (*database file*), el cual indica cuando el servidor de DNS es un servidor de nombres primario o secundario. Por ejemplo, para la zona 'dev.volcano.com 'se crearía la clave de registro:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Zones\dev.volcano.com

Configurando las propiedades de zona.

Propiedad	Descripción
General	Configura el fichero de zona en el cual van a ser almacenados los registros y especifica cuando el servidor de zona va a ser primario o secundario.
SOA Record	Configura la información de transferencia de zona y el correo del administrador del servidor de nombres.
Notify	Especifica los servidores secundarios que van a ser avisados cuando cambie el contenido de la base de datos del servidor primario. También puede aplicarse seguridad adicional al servidor de nombres especificando que únicamente los servidores secundarios definidos, pueden contactar con este servidor.
WINS lookup	Activa al servidor de nombres para resolver los nombres interrogando al servidor WINS. Una lista de servidores WINS puede ocnfigurarse en ese cuadro de dialogo.

Añadiendo registros de Recursos.

Una vez que las zonas y subdominios han sido configurados, podemos añadir los registros de recursos. Para añadir un registro de recursos, seleccionamos una zona o subdominio y pinchamos **DNS-New Host** o seleccionamos **New Record** en la barra de Menú.

New Host

Para crear un nuevo *host*, debemos teclear la direccion IP, y podemos además seleccionar **Create Asociated PTR Record** en el dominio asociado para el *reverse lookup*.

New Record

Para crear un nuevo registro, debemos seleccionar que tipo de registro de recursos vamos a crear. Una caja de dialogo, nos muestra varios campos específicos al tipo de registro. El campo TTL (*Time to Live*)

mostrado es el valor por defecto del TTL del registro SOA para esa zona. Un valor del TTL será almacenado en el registro solo si cambiamos el defecto.

Configurando Reverse Lookup.

Para encontrar un nombre de *host* dando una dirección IP debemos crear una zona de *reverse lookup*. Añadir una zona de *reverse lookup* es proceduralmente idéntico a añadir otro tipo de zona, excepto en el nombre de zona.

Por ejemplo, si un *host* tiene una dirección 198.231.25.89, va a ser representado en el dominio **in-addr.arpa** como 89.25.231.198.in-addr.arpa. Por tanto, debemos añadir una zona al DNS para 25,231,198.in-addr.arpa.

Posteriormente todos los registros PTR para la red 198.231.25.0 deberán ser añadidos en esta zona de *reverse lookup.*

Resumen

El primer paso configurando un servidor de DNS en Windows NT, es determinar la jerarquía de nuestros dominios y zonas DNS. Una vez que las zonas y subdominios están configuradas, podemos añadir los registros de recursos. Para encontrar un nombre de *host* dando su dirección IP, debemos crear una zona de *reverse lookup*.

Integrando DNS y WINS.

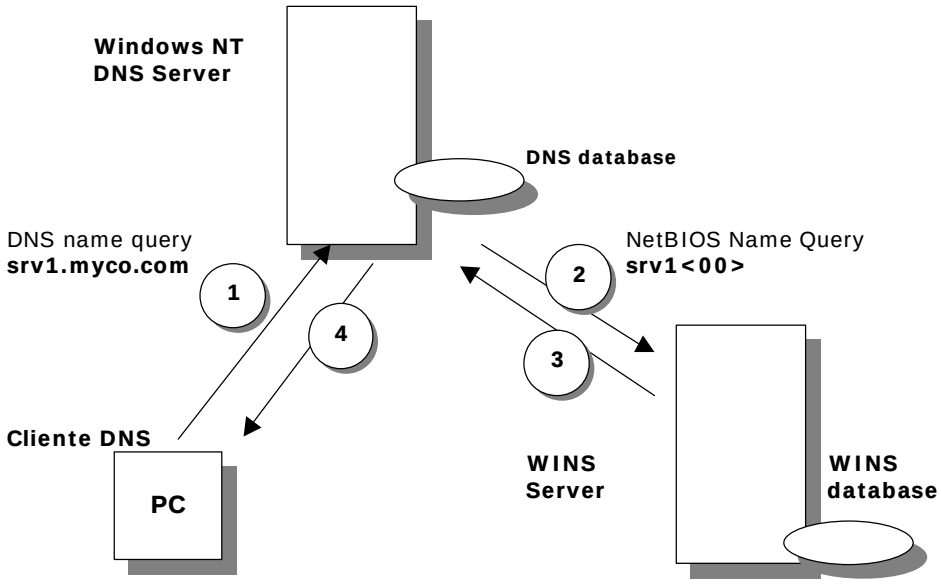
WINS requiere menos administración que el DNS debido a que automáticamente registra los nombres – direcciones IP.

El DNS es una base de datos estática de relaciones entre nombres y direcciones IP que debe ser manualmente actualizada. DNS implementa un modelo jerárquico, el cual permite administración y replicación de la base de datos en las zonas.

WINS, y otros manejadores, permite a las maquinas registrar dinámicamente su nombre y su dirección IP y por tanto necesita menos administración. WINS es un espacio de nombres 'plano' (frente al DNS jerárquico) y requiere a cada servidor WINS el mantener una base de datos completa.

El registro WINS.

Un nuevo registro de datos WINS está definido como parte de la base de datos y es único en el Servidor DNS de Microsoft. Debe ser definido en la zona del dominio principal (*root domain)* colocando un registro de tipo WINS en la base de datos del DNS. Si la relación nombre – dirección IP no está definido en la base de datos del DNS, este DNS preguntará al la base de datos WINS. Por ejemplo:



1) Un cliente contacta con el servidor de DNS y le pide la dirección IP de otro *host*.

   El servidor de nombres busca este en su base de datos y no encuentra un registro de dirección para ese *host*.

2) Debido a que la base de datos contiene un registro WINS, el servidor de DNS convierte la porción *host* del nombre en un nombre NetBIOS y envia la petición para este nombre NetBIOS al servidor WINS.

3) Si el servidor WINS es capaz de resolver el nombre, este devuelve la dirección IP al servidor de DNS.

4) El servidor de DNS devuelve la dirección IP al cliente peticionario.

Nota: Si la zona está configurada con resolución WINS, todos los servidores DNS que tienen autorización para esa zona deben ser configurados con resolución WINS.

Activando WINS *lookup*

Activando WINS *Lookup*, el DNS puede ser configurado para enviar preguntas a un servidor WINS cuando el nombre – dirección IP no puede ser resuelta por el servidor DNS.

Podemos activar WINS *Lookup* en el administrador de DNS (*DNS Manager)*, seleccionado la zona, abriendo el menú **Shorcut** y seleccionando **Properties**. Pulsando entonces en la pestaña **WINS Lookup**, marcamos **Use WINS Resolution** y entramos la dirección IP de los servidores WINS preferidos.

WINS *Reverse Lookup*

La presencia de un registro WINS-R en la zona, indica al servidor de DNS a usar un nodo NetBIOS para *lookup*. Este *lookup* es para cualquier petición de *reverse lookup* para una dirección IP que no haya sido estáticamente definida con un registro PTR en el servidor de DNS.

Con el administrador de DNS, podemos activar WINS *Reverse lookup* obteniendo las propiedades en la zona apropiada (in-addr.arpa) y seleccionando la propiedad de **WINS Reverse Lookup**.

WINS *Time to Live*

El TLL de WINS puede ser configurado desde la caja de diálogo **Advanced** en la pagina de propiedades de **WINS Lookup** de las propiedades de la zona. Cuando una relación nombre – dirección IP es resuelta por el servidor WINS, la dirección se guarda en la caché durante un tiempo llamado **Cache Timeout Value**. Por defecto este valor es de 10 minutos. Si esta dirección es reenviada a otro servidor DNS, el TTL es también reenviado.

CONECTIVIDAD EN ENTORNOS HETEROGÉNEOS

En esta parte vamos a ver la conectividad con *hosts* basado en NetBIOS y con *hosts* 'extraños'. Vamos a ver las diferentes utilidades de conectividad que Windows nos suministra.

Conectividad en entornos Heterogéneos.

Uno de los principales beneficios del TCP/IP es que nos da la posibilidad de conectar e interoperar con diferentes tipos de *hosts*, como por ejemplo con *hosts* UNIX. Vamos a ver los diferentes requerimientos para conectar con otros tipos de *host* y conectar e interoperar con *host* basados en NetBIOS (que cumplan las correspondientes RFCs).

El TCP/IP de Microsoft permite conectividad con otros tipos de *host* debido a que es un protoclo común de red usado por casi todos los sistemas. Para comunicar con cualquier ordenador, como por ejemplo, OS/2, UNIX, Solaris o VMS necesitamos un protocolo común de red, como por ejemplo el TCP/IP. También necesitamos aplicaciones (normalmente cliente / servidor) en ambos extremos de la comunicación.

Conectando a un *host* remoto con la red Microsoft.

Para utilizar los comandos de red estándar Microsoft y sus funciones (como por ejemplo **net use**, el Explorador o el Administrador de Archivos), para conectar a un *host* remoto, debemos tener presente los siguientes requerimientos:

- *Transport Driver Connectivity*: Ambos ordenadores deben ser capaces de comunicarse usnado el mismo controlador de transporte, como TCP/IP, NBF, o IPX.
- *SMB Connectivity*: El servicio de estación de trabajao comunica con un proceso servidor SMB en el *host* remoto. SMB es un protocolo de compartición de archivos usado en todos los productos de red Microsoft.

Nota: Si el parámetro de ámbito NetBIOS está configurado en el *host* remoto, el *scope ID* debe coincidir con el *scope ID* de nuestros clientes Microsoft o el sistema no será capaz de comunicarse mediante NetBIOS.

Algunos fabricantes han implementado NetBIOS sobre TCP/IP y servidores SMB en sus sistemas operativos. Ejemplos de estos fabricantes son: *Digital Equpment Corporation's* PATHWORKS en VMS, IBM LAN Server en OS/2, y LAN Manager para UNIX.

Conectando a Windows desde un *host* remoto.

Windows nos suministra servicio de ficheros en ordenadores personales a través delk protocolo ***S**erver **M**essage **B**lock* (SMB). El servicio de ficheros para los clientes UNIX está disponible a través del protocolo de ***N**etwork **F**ile **S**ystem* (NFS), el servicio FTP o bien instlando un cliente basado en SMB.

Los servidores de NFS de terceras partes están disponibles para Windows NT. Igualmente Microsoft ha sacado un paquete llamado ***S**ervices **F**or **U**nix* (SFU) que nos suminitr un servidor y cliente NFS. Estos servidores permiten a Windows NT Server suministrar servicio de ficheros a los ordenadores personales, estaciones UNIX u otros sistemas actuando como clientes NFS. Este protocolo dá soporte a las particiones nativas de disco NT (NTFS), a particiones estandar (FAT), y a sistemas de manejo de CD-ROM (CDFS), así como al sistema de archivos HPSF.

Utilidades Microsoft TCP/IP

Las siguientes utilidades del TCP/IP de Microsoft nos permiten multitud de opciones para conectar a *hosts* TCP/IP de terceros, usando *Windows Sockets*.

Utilidad TCP/IP	Función
REXEC	Ejecuta un proceso en un *host* remoto que esté ejecutando el servidor de REXEC. Este mecanismo nos da protección de seguridad mediante password.
RSH (*remote shell*)	Permite la ejecución de commandos en un servidor remoto de RSH sin conectar mediante *logon*. Este sistema no da protección mediante password.
Telnet	Nos da emulación de terminal (DEC VT 100, DEC VT 520 y TTY). Es necesario autentificación mediante password.
RCP (*remoty copy*)	Copia ficheros entre un ordenador ejecutando Windows y un servidor ejecutando el *deamon* de RCP si realizar *logon*, es decir no suministrando autentificación de usuario.
FTP	Nos permite transferencia bidireccional de ficheros entre un ordenador ejecutando Windows y cualquier *host* TCP/IP que esté ejecutando el software de servidor de FTP. Es necesario autentificación de usuario / password.
TFTP	Es una subcolección del FTP que utilza UDP (*User Datagram Protocol*) en lugar de TCP. Nos permite transferencia bidireccional de ficheros entre un ordenador ejecutando Windows y cualquier *host* ejecutando el software de servidor de TFTP. No utiliza autentificación de usuario.
Web Browser	Los navegadores, acceden a documentos almacenados en un servidor WWW, y pueden utilizar autentificación de usuario / password.
LPD	Peticiones de sercvicios LPR y el envio de trabajos de impresión a undispositivo de impresión. Nos suministra autentificación de usuario / password.
LPR	Nos da la capacidad de enviar un trabajo de impresión a una impresora conectada a un servidor ejecutando el servicio LPD. Nos suministra autentificación de usuario / password.
LPQ	Nos da la capacidad de ver la cola de impresión de un servidor LPD previa autentificación de usuario / password.

Utilidades de Ejecución Remota.

Varias utilidades TCP/IP nos dan la posibilidad de conectar a *hosts* remotos. Vamos a ver los requerimientos para su uso e cada una de las utilidades de ejecución remota.

REXEC

Remote Execution (REXEC) nos permite la facilidad de ejecución remota con autentificación basada en nombre de usuario y password. Cuando ejecutamos el comando **rexec**, nos pregunta por un usuario y una password para el *host* remoto. Después de conectar al usuario, la password es verificada en dicho *host*. Si la password es válida se ejecutará el comando especificado. REXEC normalmente termina cuando el comando remoto termina. La sintaxis de REXEC es:

rexec *tcpiphost command*

RSH

Remote Shell (RSH) se utiliza para ejecutar comandos en un servidor remoto que esté ejecutando el *daemon* RSH (en la mayoría de los casos, un *host* UNIX). RSH es util para la compilación de programas. Un usuario no tiene que conectarse (autentificarse o hacer *logon*) en el *host* UNIX para ejecutar este comando. La unica seguridad es que el nombre de usuario debe estar configurado en el fichero .rhosts del ordenador UNIX. RSH no pregunta por una password. La sintaxis de RSH es:

rsh *unixhost command*

Telnet

Telnet es un protocolo de emulación de terminal remoto originario de los terminales VT100, VT52 y TTY de *Digital Equipment Corporation*. Telnet utiliza un servicio del TCP orientado a conexión. Cualquier programa o comando que nosotros estemos procesando, lo serán por el servidor Telnet y no por el *host* local.

Para ejecutar Telnet, el sistema remoto debe estar ejecutando un programa servidor de Telnet, tambien llamado *deamon*. Windows NT original no da este programa servidor. Es necesario montar el paquete SFU (*Services for Unix*) de Microsoft para poder tener un servidor de Telnet en Windows NT. Windows 2000 nos da en nativo este servidor. Debemos tener una cuenta de usario y password en el servidor remoto para poder ejecutar este programa.

El ordenador cliente debe estar configurado con un software de cliente Telnet (este software se suministra por Microsoft en todos sus productos Windows) y una cuenta de usuario en el servidor Telnet.

Utilidades de Transferencia de Datos.

TCP/IP nos da varias utilidades de transferencia de datos, incluida la usada mayoritariamente que es el FTP. Vamos a ver los requerimientos y el uso de estas utilidades de transferencia de datos:

RCP

Al igual que el RSH, el *Remote Copy Protocol* (RCP) no requiere que el usuario esté conectado en un servidor ejecutando el *daemon* RCP (en la mayoría de los casos, un *host* UNIX). Sin embargo, el nombre del usuario debe haber sido configurado en el fichero .rhosts del ordenador UNIX y debe tener privilegios de ejecutción remota. RCP se utiliza para copiar ficheros entre un ordenador local y un UNIX remoto o entre dos *hosts* remotos. RCP no pregunta por ninguna password. Un ejemplo de la sintaxis de RCP es:

rcp *host1.user1:source host2.user2:destination*

FTP

La utilidad FTP, la cual utiliza el TCP como transporte, es una de las utilidades mayoritariamente utilizadas. Nos permite tranferencia de dicheros tipo texto y binarios desde / hacia un servidor FTP. El servidor FTP debe estar ejecutándose en el *host* UNIX o en el Windows NT remoto. FTP es usado normalmente para transferir ficheros a través de Internet.

Se requiere una cuenta de usuario en el servidor FTP, a no ser que el servidor FTP estñe configurado para permitir conexiones anónimas (*anonymous*). La mayoria de los servidores FTP de internet perminten conexiones anónimas. La sintaxis del FTP es:

ftp [*options*] *host command*

El servidor debe estar configurado con el *daemon* del servidor FTP (suministrado en Windows NT y Windows 2000), y se debe tener una cuenta de usuario definida en el servidor Windows NT.

El ordenador cliente debe estar configurado con un software FTP de cliente (suministrado en todos los productos Windows, y en particular, el propio navegador de Internet es capaza de realizar transferencias FTP), y una cuenta de usuario en el servidor FTP a no ser que este admita conexiones anónimas.

Comandos FTP

Un comando FTP lo podemos teclear en una linea o a través de un intérprete de comandos. Si el comando lo introducimos en una linea, el FTP inmediatamente intenta realizar la conexión al servidor FTP. Si no lo introducimos en una linea, el FTP se abre en modo intérprete de comados en el cual el usuario puede teclear cuanlquiera de los comandos FTP.

Algunos de los comandos FTP los vemos en el siguiente cuadro:

Commando	Función
binary	Cambia la transferencia de modo a tipo binario.
get	Copia un fichero remoto al *host* local.
put	Copia un fichero local al *host* remoto.
!	Temporalmente volvemos al *command prompt* hasta que en él tecleamos *'exit'*.
quit o **bye**	Sale del FTP.

TFTP

Trivial File Transfer Protocol (TFTP) se utiliza para la tranferencia de ficheros desde / hasta un remoto o un *host* local. El TFTP utiliza los servicios de conexión via UDP. El TFTP no soporta ninguna autentificación de usuario. Unicamente los ficheros de destino deben tener los atributos de lectura y escritura (para el 'mundo') en el entorno UNIX en el sistema remoto.

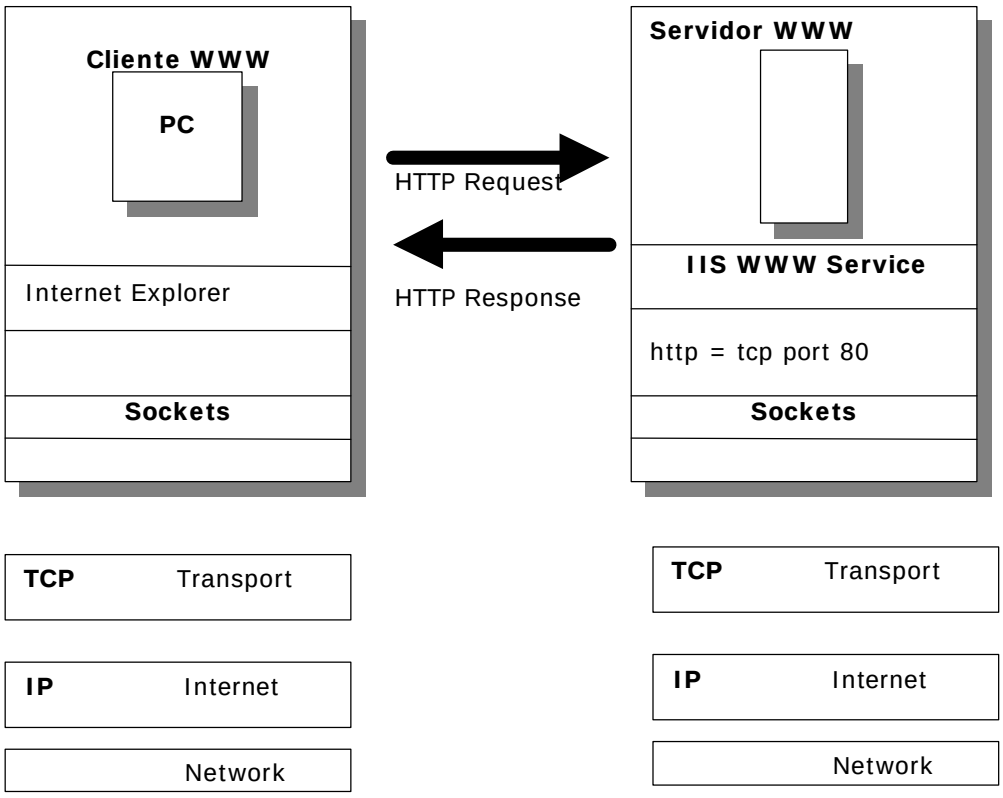
Microsoft nos da unicamente software cliente TFTP. Si queremos utilizar el servicio de servidor TFTP debemos usar software de terceros. Un ejemplo de sintaxis TFTP es:

tftp *–i host get file-one file-two*

Nota: El FTP está definido en la RFC 959. El TFTP está definido en la RFC 1350.

Web Browsers

La ***W**ord **W**ide **W**eb* (WWW) se ha convertido en la vía más popular de transferencia de datos en Internet. Los 'navegadores' acceden a documentos almacenados en un servidor de *Word Wide Web*. WWW sigue un modelo cliente / servidor y utiliza el *Hypertext Transfer Protocol* (HTTP) entre el cliente y el servidor tal y como podemos ver en la siguiente ilustración:



El cliente debe estar configurado con un navegador de Web. Existen en el mercado varios clientes Web disponibles, la mayoría de los cuales pueden ser descargados libremente desde Internet. El servidor, debe estar configurado con el servicio *Word Wide Web*.

El servidor responde con el *status* de la transacción. Correcta o fallada y el dato pedido. Después de que el dato ha sido enviado, la conexión se cierra y su estado de conversación no es retenido por el servidor. Cada objeto en un documento http requiere una conexión separada.

Los navegadores del Web, nos dan tipos tipos de beneficios en la transferencia de datos. Primero, los navegadores, soportan muchos tipos de datos. Un navegador puede automáticamente descargar y mostrar ficheros de texto y graficos, ejecutar video y sonido y lanzar aplicaciones de ayuda para tipos conocidos de ficheros.

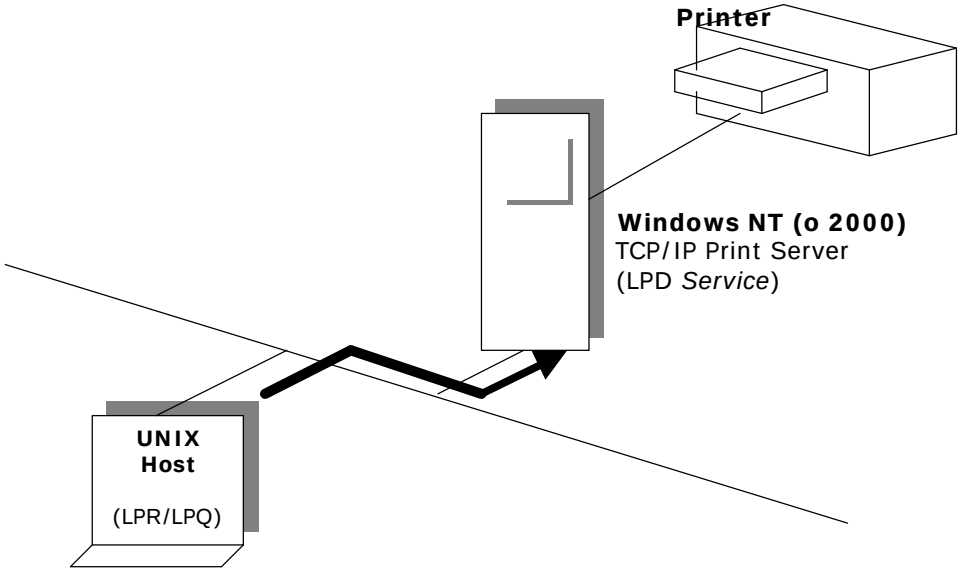
El segundo beneficio de los navegadores del Web es su soporte a varios protocolos de transferencia de datos, incluyendo FTP, Gopher, http y NNTP (*Network News Transfer Protocol*).

UTILIDADES DE IMPRESIÓN

Una vez que se ha instalado y configurado el soporte TCP/IP a impresoras, podemos contactar con la impresora usando el Administrador de Impresión o mediante comandos LPR dependiendo de sí la impresora está conectada a un ordenador ejecutando Windows NT o a un *host* UNIX. En esta parte, vamos a hablar sobre el soporte de impresión TCP/IP.

LPR y LPQ son aplicaciones clientes que comunican con LPD en el servidor, tal y como podemos ver en la siguiente imagen, Estas 3 aplicaciones, nos dan las siguientes funciones:

- LPD rueda como un servicio en un ordenador ejecutando Windows NT (LPDSVC) y parmote a cualquier ordenador con TCP/IP y LPR enviar trabajos de impresión al servidor Windows NT.
- LPR es la aplicación de impresión del cliente, y permite imprimir en cualquier *host* que utilice LPD.
- LPQ puede ser utilizado para preguntar a una impresora por la situación de los trabajos de impresión enviados.



Nota: El soporte de impresión TCP/IP de Microsoft cumple con la RFC 1179.

Usando el Servidor de Impresión TCP/IP (LPD)

Para que Windows NT acepte trabajos de impresión desde clientes LPR, el servicio del Servidor de Impresión TCP/IP (LPDSVC) necesita estar instalado y en ejecución. El servicio del Servidor de Impresión TCP/IP, debe ser iniciado desde el Panel de Control -> Servicios, desde la propia consola o bien desde el Administrador del Servidor (*Server Manager*).

Se recomienda configurar dicho servicio para que se inicie automáticamente al arrancar el Servidor.

Entradas en el Registro del TCP/IP *Print Server*

Los parámetros de configuración del servidor de impresión TCP/IP están localizadas en el registro bajo la clave:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LPDSVC\Paramteres

Usando LPR y LPQ.

Enviando trabajos de impresión (LPR)

El método para imprimir en una impresora basada en TCP/IP varía de acuerdo con el entorno en que estemos imprimiendo.

- Para aplicaciones basadas en Windows, usar el Administrador de Impresión.
- Para situaciones de linea de comandos, o cuando imprimimos desde un *host* UNIX, usar la utilidad PLR (lpr.exe).

La utilidad LPR envía ficheros a impresión al servicio LPD en un servidor Windows NT o en un *host* UNIX con la siguiente sintaxis:

lpr –*S**ip_address*** **–P***printer_file_name filename*

para enviar el trabajo de impresión, LPR realiza una conexión al servicio LPD usando los puertos 512 a 1023.

Chequeando el estado de la impresora (LPQ).

Una vez que el fichero ha sido enviado a la impresora usando LPR, podemos usar la utilidad LPQ (lpq.exe) para ver el estado de la cola de impresión. La sintaxis es la siguiente:

lpq –S*ip_address* **–P***printer_name* **–l**

Nota: Debemos fijarnos que **–S** y **–P** deben teclearse en mayusculas. En cambio **–l** (la letra 'ele') puede ser tecleada en mayusculas o minusculas.

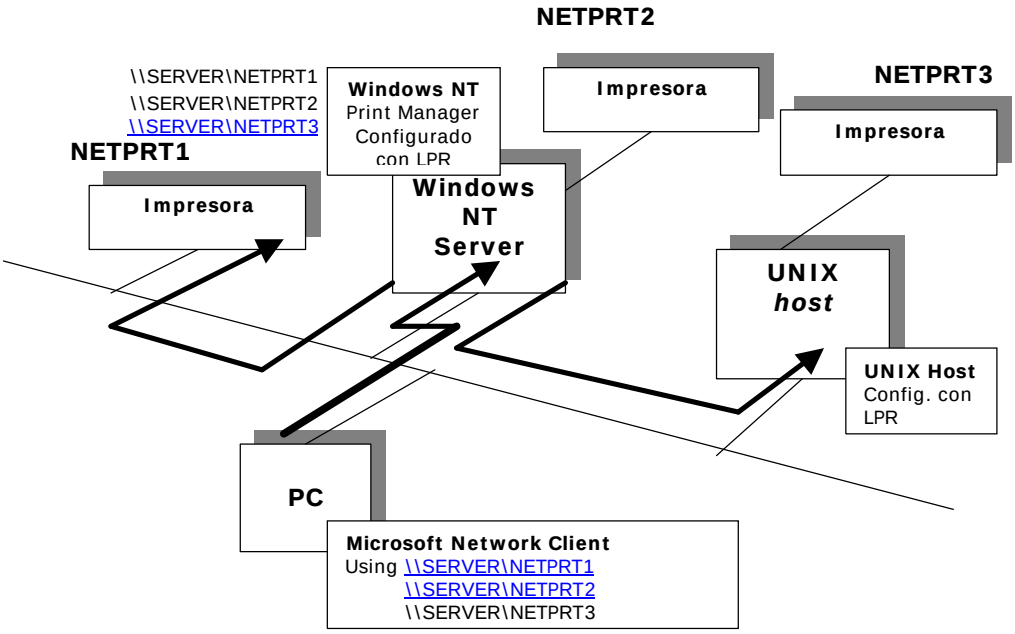
Configurando *Print Manager* con LPR *Print Monitor.*

Para configurar Windows NT para utilizar un servidor de impresión LPD, debemos añadir el soporte de impresión TCP/IP de Microsoft y configurar una impresora para que utilice LPR *Print Monitor*'.

Debe realizarse en el 'Asistente' de "Añadir Impresora" y en el botón '**Add Port**' donde seleccionamos LPR Port'.

Si no está instalado el soporte para impresoras TCP/IP de Microsoft, dicha opción no nos aparecerá.

UTILIZANDO WINDOWS NT COMO UN *PRINT GATEWAY*:



Un ordenador ejecutando Windows NT con los servicios de Impresión TCP/IP (LPD) puede realizar dos funciones de *gateway* tal y como hemos intentado expresar en el dibujo superior. Primero el ordenador ejecutando Windows NT puede recibir trabajos de impresión desde los clientes Microsoft y reenviar los paquetes automáticamente a otro servidor basado en TCP/IP que esté ejecutando LPD. En este caso, el cliente no requiere LPR o incluso no requiere TCP/IP (puede ser una comunicación NetBeui).

Además, el ordenador ejecutando Windows NT puede recibir trabajos de impresión desde cualquier cliente LPR y reenviar estos a cualquier impresora visible al ordenador que esté ejecutando Windows NT.

SOLUCIONANDO PROBLEMAS EN EL TCP/IP

En este capítulo (como colofón a estos Articulos), vamos a revisar unas guías para la solución de problemas en una red IP. Vamos a revisar los problemas más comunes en TCP/IP, sus síntomas y sus posibles causas.

HERRAMIENTAS DE DIAGNÓSTICO.

Existe un proceso ordenado para intentar solucionar los problemas en TCP/IP. Vamos a ver el proceso y a recordar las utilidades de Windows para los problemas en TCP/IP.

Solucionar un problema es facil cuando podemos identificar su origen. Los problemas basados en TCP/IP pueden ser agrupados en las categorías que vamos a listar en la siguiente tabla:

Origen del Problema	Características comunes
Configuración	El *host* no quiere inicializarse o uno de los servicios no quiere arrancar.
Direccionamiento IP	No es capaz de comunicar con otros *hosts*. El *hosts* puede para de responder.
Subnetting	Podemos hacer un *ping* a las estaciones de trabajo, pero no somos capaces de acceder a *hosts* remotos ni locales.
Resolución de direcciones	Podemos hacer *ping* a nuestro PC. Pero no a otros *hosts*.
Resolución NetBIOS	Podemos acceder al *host* por dirección IP, pero no podemos establecer una conexión con el comando **net**.
Resolución *Host name*	Podemos acceder a un *host* por su dirección IP, pero no por su nombre de *host*.

Utilidades TCP/IP

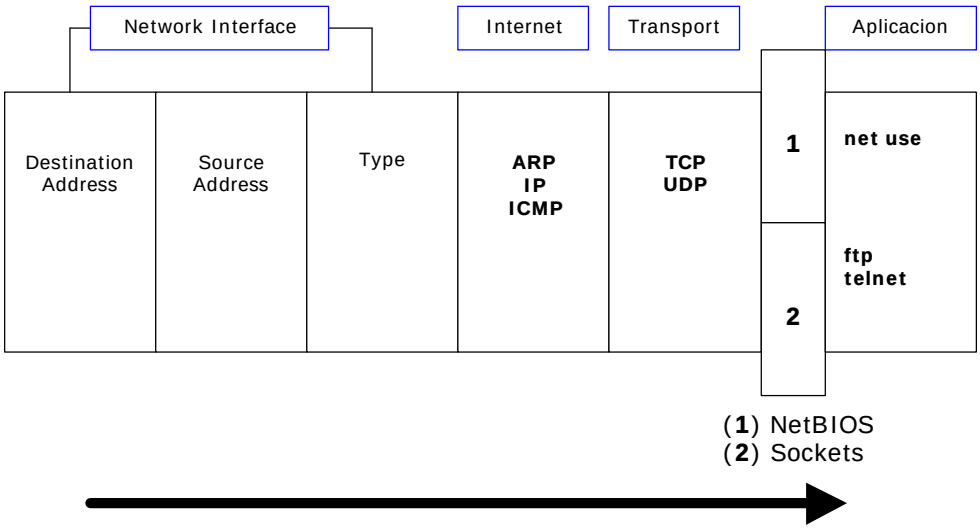
Windows incluye varias utilidades que nos pueden ayudar para la solución de problemas TCP/IP:

Usar esta herramienta	Para
PING	Verificar que el TCP/IP está correctamente configurado y que los otros hosts están disponibles.
ARP	Ver la caché ARP para detectar entradas inválidas.
NETSTAT	Nos muestra estadísticas de los protocolos y el estado actual de las conexiones TCP/IP.
NBTSTAT	Ver en estado de la conexiones actuales NetBIOS sobre TCP/IP, actualizar el caché LMHOSTS, o determinar nuestro nombre registrado y *'scope'* (alcance).
IPCONFIG	Verificar la confioguración TCP/IP, incluyendo las direcciones de los servidores DHCP y WINS.
TRACERT	Verificar el camino a un *host* remoto.
ROUTE	Ver o modificar la actual tabla de rutas.
NSLOOKUP	Ver información desde los servidores de nombres DNS.
Servicio SNMP	Dar información estadística para la administración de sistemas SNMP.
Event Log	Visor de sucesos (en Windows NT).

Performance Monitor	Analizar rendimientos (en Windows NT).
Network Monitor	Capturar paquetes de entrada y salida para analizar un problema (en Windows NT).
Editro de Registro	Ver y editar los parámetros de configuración.

Guía de solución de problemas.

Cuando solucionamos problemas en TCP/IP, es recomendable intentar hacerlo desde la capa más baja de la colección de protocolos Internet hasta la capa más alta, tal y como no muestra la siguiente ilustración. El objetivo es verificar que protocolos en cada capa pueden comunicar con protocolos en la capa por encima y por debajo de ella.



Hay dos pasos en la solución de problemas. Debemos asegurarnos que podemos:

1) Realizar correctamente un PING.

   Si podemos realizar correctamente un PING, acabamos de verificar que las comunicaciones IP entra la capa de Interface de Red (*Network Interface*) y la capa de Internet son correctas. PING utiliza el ARP para resolver la dirección IP a dirección hardware.

2) Establecer una sesión con un *host*.

   Si podemos establecer una sesión, acabamos de veridicar la comunicación entra la capa de Inerface de Red hasta la capa de aplicación.

Nota: Si somos incapaces de resolver un problema, puede que necesitemos un analizador de IP (como por ejemplo el *Microsoft Network Monitor*) para ver la actividad de red en cada capa.

Verificando comunicaciones IP

El primer intento para solucionar problemas es asegurarse que podemos ejecutar correctamente un PING a una dirección IP. Esto verifica las comunicaciones entre la Interface de Red y la capa de Internet. Usar el PING utilizando un nombre de *host* solo puede ser existoso si previamente podemos llegar con el PING a su dirección. El siguiente procedimiento y dibujo, muestra como solucionar problemas de conexión usando PING.

1) PING a la dirección de *loopback* (127.0.0.1) para verificar que el TCP/IP ha sido instalado y cargado correctamente. Si este paso no es correct, volver a verificar el sistema después de instalar de nuevo y configurar el TCP/IP.

2) PING a nuestra dirección IP para verificar que está configurado correctamente. Si este paso no es correcto:

    - Ver la configuración de la Red en el Panel de Control para verificar que la dirección IP está correctamente definida.

    - Verificar que la dirección IP es válida y que sigue las reglas dadas para las direcciones.

3) Ping a la dirección IP del *gateway* por defecto para verificar que el *gateway* está funcionando y configurado correctamente y que la comunicación está operativa en la red local. Si este paso no es correcto, verificar que estamos utilizando una dirección IP correcta y mascara de red también correcta.

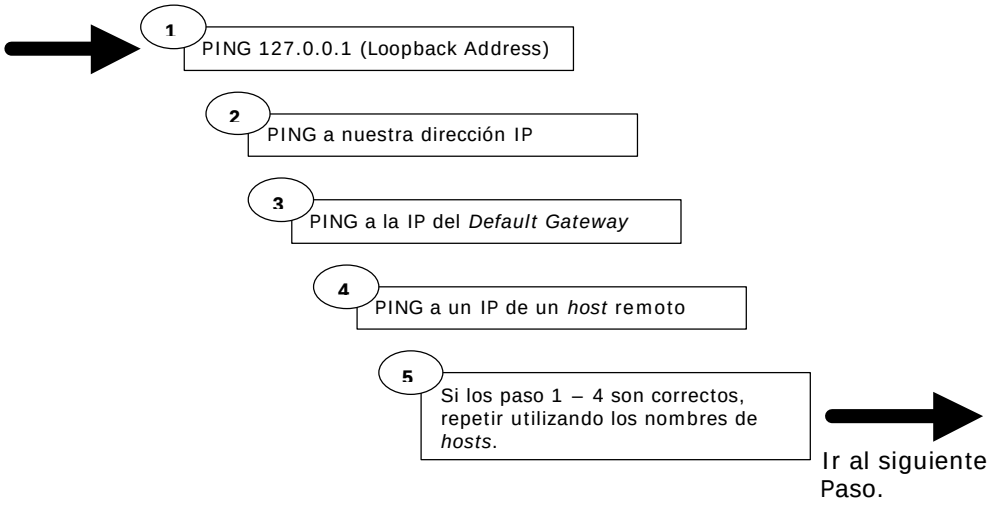
4) PING a la dirección IP de un *host* remoto para verificar la conexión en la WAN. Si este paso no es correcto:

    - Verificar que la dirección IP del *gateway* por defecto sea correcta.

    - Asegurarse de que el *host* remoto está operativo.

    - Verificar que el enlace entre los *routers* está operativo.

5) Después que podamos realizar un PING a la dirección IP, debemos realizar un PING al nombre del *host* para verificar que el nombre ha sido correctamente configurado en el fichero HOSTS.



Verificando la sesión de comunicación TCP/IP.

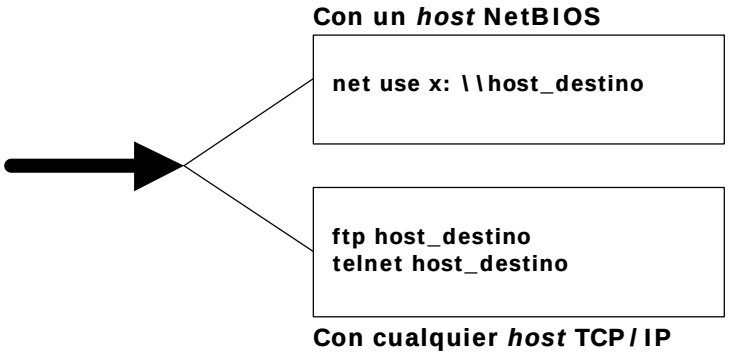
El punto siguiente durante la solución de problemas es verificar las comunicaciones desde la capa Internet a la capa de aplicación mediante el establecimiento correcto de una sesión. Usar uno de los siguientes metodos para verificar las comunicaciones entre la capa de Interface de Red y la capa de Aplicación tal y como vemos en la siguiente ilustración:

Para establecer una sesión NetBIOS sobre TCP/IP en un ordenador ejecutando Windows o un *host* que cumpla las especificaciones NetBIOS, debemos realizar una conexión con el comando **net use** o el comando **net view**. Si este paso no es correcto:

- Verificar que el *host* destino tiene instalado y funcionando el NetBIOS.
- Confirmar que el *scope* ID' en el *host* destino coincide con el del origen.
- Verificar que estamos utilizando un nombre NetBIOS correcto.
- Si el *host* destino, está en una red remota, verificar que la conversión nombre – dirección IP está disponible en un servidor WINS o en el archivo LMHOSTS y que su definicón es correcta.

Para establecer una sesión Windows Sockets con un *host* IP, usar las utilidades Telnet o FTP para realizar la conexión. Si este paso no es correcto:

- Verificar que el *host* destino está configurado con un servidor Telnet o con un servidor FTP.
- Confirmar que tenemos los permisos y autorizaciones correctas en el *host* destino.
- Comprobar el fichero HOSTS o el servidor DNS para verificar que tengan una entrada vákida si estamos utilizando un nombre de *host*.



INDICE