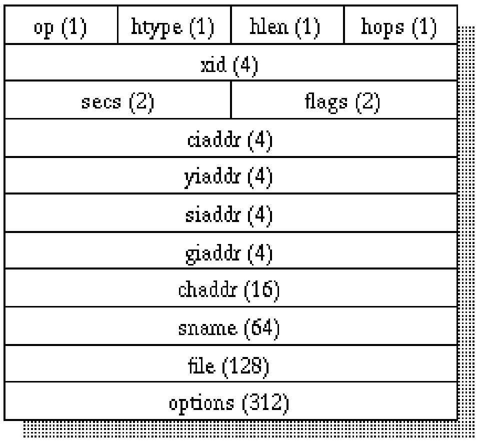
Presentación de DHCP

Los PC´s han reducido enormemente su tamaño posibilitando la movilidad sin pérdida de capacidad de trabajo. Además las tarjetas PCMCIA permiten las conexión a red de estos PC´s. Para la conexión temporal de un ordenador portátil, principal uso de DHCP aunque no el único, a una red IP necesitamos ciertos parámetros como máscara de subred, pasarela por defecto… Existe la posibilidad de configurar manualmente todo ello pero no es la mejor manera para usuarios noveles. Así, un método para obtener direcciones IP temporales es necesario para solucionar estos problemas. Otra dificultad que hace más necesario este servicio es la saturación de direcciones IP que no se relajará hasta la aparición del IPv6 .

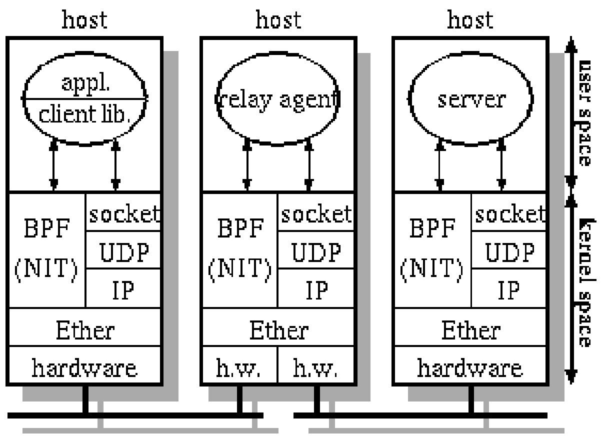
Por ello se propuso DHCP, que no es más que una extensión del servicio BOOTP. Tanto BOOTP como RARP podían proporcionar el servicio de configuración automática, pero no la asignación de direcciones dinámica y temporalmente. Ambos usaban un sistema sencillo el cliente solicita y el servidor responde. RARP se usaba en Ethernet y relacionaba las direcciones de Ethernet con las IP. BOOTP, sin embargo utiliza mensajes UDP.

DHCP también usa el modelo cliente-servidor y está construido sobre UDP con un agente *relay* como BOOTP. Incluso el paquete de DHCP y BOOTP son compatibles. Así, el servidor DHCP puede responder a un cliente BOOTP y viceversa.

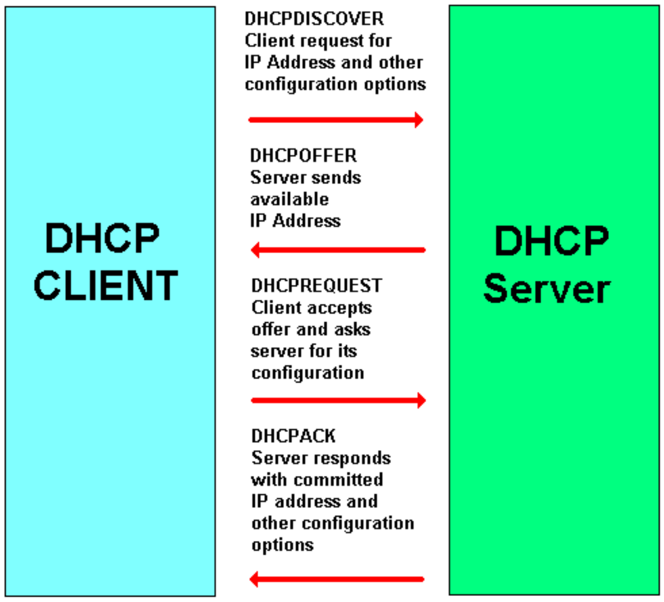


field	description
op	Message type. BootRequest/BootReply
xid	Transaction ID, a random number chosen by the client, used by the client and server to associate messages and responses between a client and a server.
ciaddr	Client IP address; only filled in if client can respond to ARP requests.
yiaddr	'your' (client) IP address.
giaddr	Relay agent IP address, used in booting via a relay-agent.
sname	Optional server host name.
file	Boot file name.
options	Optional parameters field.

Las capas del protocolo, basándonos en una red local Ethernet, se puede especificar así (la capa BPF (NIT) es la forma de implementarlo de las máquinas Sun para salvar las dificultades de trabajar con *sockets*):



El funcionamiento del modelo cliente-servidor en DHCP se puede representar esquemáticamente así:



DHCPdiscover lo manda el cliente por la dirección de broadcast, pudiendo sugerir la dirección IP a asignarle. El servidor responde con DHCPoffer que incluye la dirección IP disponible y otro parámetros tales como la máscara de subred. DHCPrequest incluye un identificador de servidor que indica que acpeta su oferta y rechaza la de otros posibles servidores que le hayan respondido. DHCPpack contiene los parámetros de servidor aceptado.

Falta saber cómo el servidor DHCP descubre desde qué subred ha venido la petición. Los mesajes del cliente se mandana ...., que son, a menudo, parte de un routr IP. El agente DHCP *relay* graba la subred desde la que se ha recibido el mensaje en la cabecera del mensaje que se pasará al servidor DHCP.

<u>Funcionamiento general del servidor DHCP (Linux)</u>

El protocolo DHCP permite a un host desconocido al administrador de red que le sea asignada una nueva dirección IP de un grupo de direcciones IP de su red.

Al comenzar *dhcpd* lee el archivo *dhcpd.conf* y almacena una lista de direcciones disponibles para cada subred. Cuando un cliente pide una dirección *dhcp* le asigna una dirección y una 'licencia' que expira después de una determinada cantidad de tiempo elegida por el administrador.

Para mantener la pista de las *licencias* cuando el sistema rebota y el servidor rearranca, *dhcpd* mantiene una lidta de *licencias* que ha asignado en el fichero *dhcpd.leases*, archivo que se rellena antes de confiramr la *licencia* al cliente.

Cualquier camnio que se realice el el fichero *dhcpd.conf* exige el relanzamiento del servicio *dhcpd*. Para hacerlo primero debemos matar el proceso presente, que tiene un *pid* almacenado en el fichero */var/run/dhcpd.pid*

Los nombres de las interfaces de red que el servicio debe escuchar una petición broadcast se debe especificar en la línea de comandos, aunque ciertos sistemas ya tienen la capacidad de detectar estas interfaces de broadcast. Si no se indica el sistema hará lo posible por identificar éstas y escuchará en todas y cada una de ellos. Si, además el *dhcpd* debe escuchar en otro puerto que el estándar (67), tenemos que usar el flag *'-p´* e indicar el puerto.

Para corre el servicio en *foreground* , en vez de cómo demonio en *background*, debemos especificar el *flag '-f'*, esto es sobre todo útil para depuraciones

<u>Configuración del servidor 'dhcpd.conf'</u>

Como en la mayoría de archivos de configuración los elementos que se especifican se dividen en parámetros y declaraciones. Los parámetros indican cómo hacer, si hay que hacerlo o qué parámetros proporcionar al cliente. Las declaraciones son necesarias para la descripción de la topología de la red, es decir cómo organizar las direcciones con las que trabajará el servicio.

Los parámetros tienen que ser especificados antes de las declaraciones, las cuales dependen de los parámetros especificados. Así al principio del fichero hay un lugar para parámetros globales, los más habituales son el nombre del dominio de la organización, las direcciones del servidor de nombres,por ejemplo:

option domain-name "acme"
option domain-name-servers m4.d4.acme, m4.d2.acme

A continuación tenemos el campo *shared-network* al que recurriremos en el caso de que tengamos varias subredes IP que compartan una misma red física, con unos parámetros generales a esta red física y unos parámetros particulares para cada subred que designemos dentro de ella. Este campo, y sus parámetros aplicados, será utilizado al lanzar clientes en esa subred a menos que otros parámetros se especifiquen en el campo de subred o en el *host* mismo. A la hora de asignar direcciones no se distinguirá a que subred pertenece el *host*.

Ela campo *subnet* se utiliza para proporcionar al servicio DHCP información acerca de si cierta dirección IP está en esa subred o no. Para definir las direcciones que se le podrán asignar dinámicamente a *hosts* de esa subred tenemos el campo *range* en el que se especifica la primera y la útlima posibles, si tiene la in dicación *dynamic-bootp* sabemos que la dirección se podrá asignar tanto a clientes de BOOTP como de DHCP. El *subnet-number* indica la dirección que describe a la red completa. La *netmask* que debe darse con cada campo *subnet*. Con estos dos parámetros podemos saber si una máquina pertenece a esta subred.

La subredes IP que formen por sí mismas una red física deben especificarse com *subnet* fuera del los campos *shared-network*.

La declaración *host* es necesaria al menos para cada cliente BOOTP que se va a servir, también se pueden especificar para clientes DHCP, aunque no es necesario a menos que el arranque sólo esté permitido para hosts conocidos. Es recomendable, sin embargo, poder lanzar un cliente DHCP o BOOTP en más de una subred con direcciones fijas, lo que se puede especificar en el parámetro *fixed address*. Si no queremos que se arranque con una dirección fija no especificamos este parámetro.

La declaración *host* se identifica con un cliente DHCP o BOOTP por el campo dhcp-client-identifier indicado en el host o por el parámetro *hardware* proporcionada del cliente (necesario en BOOTP).

La declaración *group* se usa simplemente para aplicar unas determinadas características a más de una declaración. Puede agrupar hosts, shared-networks, subredes o, incluso, grupos.

Existen otros tipos de declaraciones, menos asociadas a la topología de la red, y más al comportamiento de DHCP ante ciertas peticiones. (*deny* and *allow* statements *).*

Comportamiento ante *unkown clients*, les asiganmos direción o no.

Comportamiento ante una petición BOOTP ¿ debe responder ?, por defecto sí lo hace.

No responder a cierto host. Debe encontrarse dentro de declaración *host.*

También debemos especificar otros parámetros hasta ahora no explicados:

Default-lease-time Indica la cantidad de tiempo dado a un cliente DHCP si no pide un tiempo específico. El tiempo que le permite el uso de una dirección asignada dinámicamente El número dado es en segundos.

Max-lease-time Indica el máximo tiempo que se la dará a un cliente aunque solicite más.

Filename especifica el fichero que se cargará por el cliente. Debe ser un nombre reconocible para cualquier protocolo que el cliente pueda usar para cargar el fichero.

Server-name simple inforamción al cliente de quién le ha servido.

Next-server aporta la dirección del servidor desde donde se tiene que cargar el archivo especificado en *filename*. El nombre tiene que ser o una dirección numérica o un nombre de dominio.

Dynamic-bootp-lease-length Para establecer la duración de los

Server-identifier

Configuración del cliente 'dhclient.conf'

Este fichero es usado para la configuración del comportamiento del cliente en temes tales com: temporización del protocolo, información pedida desde el servidor, pedida al servidor, para establecer valores por defecto si no los especifica el servidor, corregir parámetros generales del servidor…

Para la temporización tenemos la sentencia *timeout*. Ésta determina la cantidad de tiempo que debe transcurrir entre la petición para conocer su dirección y la decisión de que no va a recibir respuesta de ningún servidor. Por defecto es de 60 seg.

La sentencia *retry* indica el período que transcurrirá entre que el cliente determina que no hay servidor DHCP y reintenta una nueva petición.

Select-timeout indica el tiempo de espera que ejercerá el cliente tras la recepción de una oferta de tratamiento de un servidor DHCP para ver si llega una que le sea preferente. Por defecto este tiempo de espera es nulo.

Backoff-cutoff: El cliente puede usar un algoritmo de *backoff* con cierta aleatoriedad, para que si muchos clientes intentan configurase por sí mismos a la vez , que no tengan sus peticiones bloqueadas.

Initial-interval indica la cantidad de tiempo entre el primer intento en alcanzar un servidor y el segundo intento. Cada vez que se manda una petición, el intervalo entre mensajes se incrementa dos veces el intervalo actual multipplicado por un número aleatorio entre cero y uno. Así hasta llegar al valor definido en *backoff-cutoff* , donde se establece a ese valor.

Existe otro grupo de opciones a indicar en este fichero que son el tratamiento de peticiones cliente<->servidor. La sentencia *request* indica a los servidores que le responden que le manden los datos especificados a continuación. Por ejemplo: *domain-name*, *host-name*...

Require Las ofertas de servidores que no respondadn a las opciones pedidas no se aceptarán como validas.

Send Hace que el cliente mande las opciones especificadas al servidor con los valores indicados, las opciones que ya se envían por efecto del protocolo DHCP no tienen que indicarse aquí.

Cuando las opciones envíadas por el servidor no son válidas, o no las envía simplemente se recurre a lo indicado en la sentencia *default* (valores por defecto) y la sentencia *supersede* (valores a utilizar obligatoriamente).

La sentencia *script* especifica el path del scritp de configuración del cliente para la configuración previa a la petición a un servido DHCP.

Agente de retransmisiones 'dhcprelay'

Facilita la retransmisión de peticiones *DHCP* y *BOOTP* de una subred que no tiene conexión con un servidor *DHCP* directamente a uno o más servidores en otras subredes. Este agente escucha en todas las interfaces unidas a un *host*. A menos que se hayan especificado determinadas interfaces.

Cuando se recibe una solicitud, la redirige a la lista de servidores *DHCP* especificados en la línea de comandos. Cuando se recibe la respuesta, la retransmite a la subred que realizó la consulta, bien por broadcast o por unicast cuando sea posible.

El nombre de las interfaces que debe escuchar se debe especificar desde la línea de comandos usando la opción *'-i'*. Si no se especifica escuchará en todas las interfaces excepto en aquellas que no sean de *broadcast*. Si, además, debe escuchar en puerto distinto del estándar (el 67), tenemos que especificarlo con la opción *'- p'*.

Dhcprelay funcionará normalmente en *foreground* hasta que haya configurado una interfaz, entonces pasará a funcionar en *background*.

<u>Elección del tiempo de licencia</u>

Un factor muy importante para esto es que el cliente empieza a intentar renovar la *licencia* cunado está por la mitad de ella. Otro factor es que mientras más larga sea más tiempo toma ver, o propagarse, los cambios en configuración del cliente DHCP. Más preguntas a responder son:

¿Tienes más usuarios que direcciones? Si es así tienes que hacer el tiempo pequeño para no paralizar a nuevos solicitantes. Por ejemplo si tú tienes 20 usuarios para 10 direcciones una *licencia* de 2 días no es práctica.

¿Tienes usuarios móviles? Aquí debes situarte de forma idéntica a la situación anterior.

¿Cómo de rápidas serán las comunicaciones sobre las líneas que usan los paquetes DHCP? A más corto el período mayor será la carga en el servidor. En general una licencia de, al menos, 2 horas es lo suficientemente larga como para que la carga de miles de clientes sea despreciable.

¿Tengo usuarios que quieren comunicar su dirección IP a otros usuarios? En este caso el período debe ser bastante largo para que es dirección se mantenga.

La decisión, por tanto, depende de la respuesta que demos a estas preguntas.