

Práctica 4. Implementación de servidores sencillos.

La siguiente práctica consiste en implementar dos servidores básicos para los servicios DHCP (UDP) y HTTP (TCP).

Es muy importante prestar atención extrema a la documentación y seguir a rajatabla el protocolo: los clientes esperan que el servidor así lo haga.

Esta práctica se puede realizar en solitario o en grupos de hasta 3 personas.

Los servidores deben realizar las siguientes tareas:

1. El servidor DHCP debe permitir que un solo cliente adquiera una configuración de red correcta. Responderá a las peticiones DHCPDISCOVER y DHCPREQUEST y ofrecerá la IP 172.16.1.120, con máscara de red 255.255.255.0, router 172.16.1.1 y servidor DNS 8.8.8.8. El tiempo de cesión será de 60 segundos, y el tiempo de renovación será la mitad. El servidor debe tener en su configuración la dirección IP de la interfaz a la que debe conectarse el ServerSocket.
2. El servidor HTTP debe mandar al cliente páginas web compuestas por texto. Solamente atiende a peticiones GET, sin parámetros. El servidor debe identificar el recurso que se está pidiendo, cargarlo y enviarlo. Si el cliente envía una petición de un recurso que el servidor no conoce (un archivo que no forma parte del sitio web), deberá enviar un mensaje con código "204 No Content", las cabeceras y sin contenido. El servidor escuchará peticiones por el puerto 8081. El directorio del sitio web tiene que estar en la raíz del proyecto, de modo que el servidor lo alcance agregando "site" a la ruta del recurso enviada por el cliente.
3. El servidor HTTP es capaz de enviar imágenes (png, jpg y ico para el favicon), archivos de estilos css y de javascript js.
4. El servidor DHCP puede dar servicio a un número arbitrario de clientes, ofrecerá direcciones IP dentro de un rango definido por una IP mínima y una máxima, configurables. El servidor debe comprobar que la petición de renovación de un cliente cae dentro de su ámbito y que no está ya en uso por otro cliente. Si alguna de las condiciones anteriores no se cumple, responderá con DHCPNAK.
5. El servidor DHCP controla los tiempos de cesión y libera las direcciones IP para las cuales el tiempo ha concluido y no se ha pedido una renovación. También se controla que no se ofrezca (en DHCP OFFER) la misma dirección IP a más de un cliente, guardando un registro de las direcciones ofrecidas. Las ofertas no satisfechas han de desaparecer al cabo de un tiempo (puede ser cuando se limpian las cesiones caducadas).

Indicaciones:

- Los parámetros de configuración deben encontrarse todos al principio de la clase principal.
- En el caso del servidor DHCP, los hilos que gestionan las comunicaciones deben ser asignados a un cliente determinado (se pueden distinguir por la dirección MAC), y deben

enviar los mensajes DHCPPOFFER (si procede) y/o DHCPACK. Así pues, cada hilo gestionará la secuencia DHCPDISCOVER - DHCPPOFFER - DHCPREQUEST - DHCPACK o bien DHCPREQUEST - DHCPACK si el cliente pide una renovación. Al enviar el mensaje DHCPACK, el hilo terminará su ejecución.

- Cada hilo puede crear un socket UDP para escribir las respuestas al cliente. Sin embargo, todas las peticiones deben llegar al servidor y ser canalizadas desde ahí a los hilos.
- Como los diferentes hilos no pueden bloquear el socket, pues esto es tarea de la clase principal, la gestión de los mensajes debe implementarse como un consumidor-productor, donde la clase principal es el productor y los hilos son los consumidores.
- El socket del servidor debe abrirse no solamente por el puerto 67, sino por la dirección IP correcta. Debe utilizarse el constructor de Socket que recibe un objeto InetAddress.
- Para probar el servidor DHCP, es recomendable utilizar máquinas virtuales. Una sola máquina virtual puede tener varias interfaces de red, lo cual ayudará a probar el sistema con varios clientes. Es muy importante ejecutar Wireshark para escrutar los diferentes mensajes DHCP y así implementar el servidor de forma que los mensajes que escriba sean formalmente correctos.
- En el caso del servidor HTTP, cada hilo gestiona las comunicaciones con un cliente, y debe morir al enviar el recurso solicitado. Si el cliente cierra la conexión en el proceso, el hilo simplemente debe finalizar su ejecución de forma ordenada, capturando la excepción.
- Para probar el servidor HTTP, en el aula virtual se encuentra un pequeño sitio web de prueba con todos los elementos necesarios.

Criterios de calificación:

- Para obtener un aprobado en solitario, los servidores deben poder cumplir los puntos 1 y 2 de la práctica.
- Para obtener un notable en solitario o un aprobado en grupo, se debe cumplir hasta el punto 3 de la práctica.
- Para obtener un sobresaliente en solitario o un notable en grupo, se debe cumplir hasta el punto 4 de la práctica.
- Para obtener un sobresaliente en grupo, se debe cumplir hasta el punto 5.
- Las prácticas entregadas deben poder compilarse sin errores. Si existen errores de sintaxis la práctica será calificada como 0.
- Si se producen errores en tiempo de ejecución, la calificación dependerá de la importancia del error y de bajo qué circunstancias se produce.
- Se tendrá en cuenta el grado de desarrollo de cada uno de los ítems de que consta la práctica, de cara a su calificación.
- La calidad del código también será objeto de escrutinio. Aunque funcione, hay formas mejores y peores de escribir el código y por consiguiente los trabajos con mejores prácticas de programación serán calificadas con una mayor puntuación.

La existencia de las siguientes circunstancias o elementos será objeto de penalización:

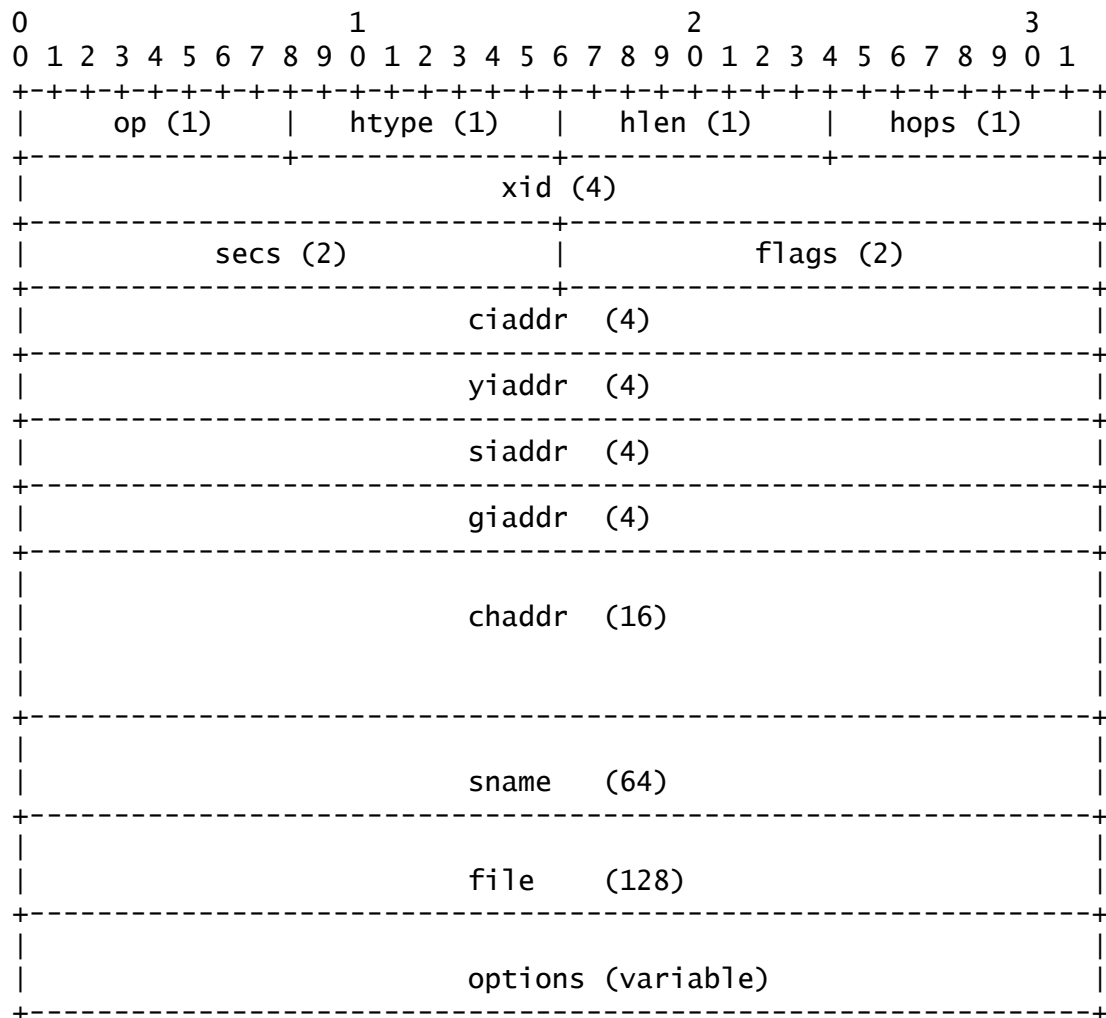
- Insuficiente descomposición en métodos. Generalmente, un método debe caber en la pantalla (hasta -1 punto).

- Presencia de código repetido (hasta -1 punto).
- Presencia de variables o métodos innecesarios (hasta -0.5 puntos).
- Uso de atributos public en métodos o variables que no lo requieren (-0.25 puntos).
- Presencia de código innecesario o inalcanzable (hasta -0.5 puntos).
- Nombres de variables o métodos inapropiados (-0.25 puntos).
- Salida de bucles mediante break (-0.5 puntos).
- Iteraciones innecesarias en un bucle (-0.5 puntos).
- Código no comentado (-0.5 puntos).
- Código mal indentado o desordenado (-0.25 puntos).

Estructura básica del protocolo DHCP

- El servidor escucha siempre por el puerto 67, el cliente escucha siempre por el puerto 68.
- De acuerdo a la documentación, todo cliente debe reservar al menos 576 bytes para cada mensaje del servidor. El servidor debe hacer lo propio.
- La fuente fundamental de la siguiente información es <https://datatracker.ietf.org/doc/html/rfc2131>.

Estructura de un mensaje DHCP:



Podemos decir que el mensaje tiene dos partes: una de cabeceras que cubre hasta el campo file incluido, y una de opciones. El servidor debe tratar de forma diferente ambas partes, sobre todo porque la primera sigue rígidamente la estructura anterior, mientras que la parte de opciones tiene un contenido de longitud variable, con solamente algunas de las opciones posibles y dispuestas sin un orden predefinido.

Los campos de la cabecera son los siguientes:

Campo	Bytes	Descripción
op	1	Message type. 1 = BOOTREQUEST (de cliente a servidor). 2 = BOOTREPLY (de servidor a cliente).
htype	1	Hardware type. Se puede usar simplemente el valor 1, que hace referencia a una red ethernet de 10mb.
hlen	1	Hardware address length. Usar el valor 6.
hops	1	Utilizado por relay agents. Usar el valor 0.
xid	4	Transaction ID. Número aleatorio generado por el cliente que identifica de forma (probablemente) única cada intercambio de mensajes entre cliente y servidor. El servidor debe responder con el mismo número que el cliente envía.
secs	2	Rellenado por el cliente. El servidor puede poner 0.
flags	2	Solamente debe tomar los valores 0 y 32768 (10000000000000000). 0 = el mensaje no es broadcast (DHCPACK y DHCPNAK) 32768 = el mensaje es broadcast (DHCPOFFER).
ciaddr	4	La dirección IP del cliente. El servidor puede rellenarla con ceros.
yiaddr	4	La dirección IP que el servidor ofrece al cliente.
siaddr	4	La dirección IP del siguiente servidor DHCP. El servidor puede rellenarla con ceros.
giaddr	4	La dirección IP del relay agent. El servidor puede rellenarla con ceros.
chaddr	16	La dirección MAC del cliente.
sname	64	Server host name. Opcional. Si se rellena (que no es necesario), debe contener un carácter /0 al final.
file	128	Boot file name. Opcional. Puede dejarse en 128 ceros.
options	variable	Parámetros opcionales. Muchos de ellos son necesarios.

Estructura del campo options:

- Los primeros 4 bytes son siempre 99 130 83 99 (se conocen como "magic cookie").
- A continuación se suceden las opciones, que tienen la siguiente estructura:
 - 1 byte que señala el tipo de mensaje.
 - 1 byte que señala la longitud de los datos que van encapsulados.
 - Los datos.
- La última opción, obligatoria, es la opción END, la cual consta exclusivamente de 1 byte de valor 255.

El orden en el que las opciones se escriben en el mensaje es irrelevante, salvo por la opción END.

Opciones que es necesario que el servidor incluya en sus mensajes o atienda de los recibidos por el cliente:

Opción	Código	Longitud	Datos
Tipo de mensaje	53	1	1 = DHCPDiscover 2 = DHCPOffer 3 = DHCPRequest 4 = DHCPDecline 5 = DHCPACK 6 = DHCPNAK 7 = DHCPRelease 8 = DHCPInform
Máscara	1	4	La máscara de subred
Servidor DNS	6	4/8	Las direcciones IP de los servidores DNS (1 o 2)
Router	3	4/8	Las direcciones IP de los gateways (1 o 2)
Requested IP	50	4	La IP solicitada por el cliente en un mensaje DHCPRequest.
Tiempo de cesión	51	4	El tiempo de cesión en segundos
Tiempo de renovación	58	4	El tiempo de renovación en segundos (típicamente, la mitad del tiempo de cesión).
Identificador del servidor	54	4	La IP del servidor. Debe incluirse siempre en los mensajes DHCPOffer y DHCPACK.
END			Un byte con valor 255

Detalles importantes:

- Cuando un cliente todavía no tiene IP, la dirección que solicita en el mensaje DHCPRequest va en la parte de opciones (código 50). Sin embargo, cuando el cliente solicita renovación, dicha IP va en la parte de la cabecera, en los bytes 12 a 15.
- El mensaje DHCPACK que el servidor envía al cliente cuando éste todavía no tiene IP ha de ser enviado en broadcast. Los mensajes DHCPACK que responden a una petición de renovación deben ir dirigidos directamente al cliente.

Estructura básica del protocolo HTTP

El mensaje tiene tres partes:

1. Línea de petición. Tiene la siguiente estructura:

`<funcion> /<archivo solicitado> <protocolo>/r/n`

Por ejemplo:

`GET /functions.js HTTP/1.1`

`GET / HTTP/1.1`

2. Cabeceras. Hay muchísimos tipos de cabeceras, y pueden contener mucha información útil para el navegador o para las aplicaciones web que se ejecuten a instancias del cliente. No obstante, para esta práctica solamente son necesarias dos cabeceras:

Content-Length. Contiene la longitud del contenido.

Content-Type. Son los tipos MIME. Se puede encontrar una lista completa en <https://www.iana.org/assignments/media-types/media-types.xhtml>

Por ejemplo, si el contenido tiene 3554 bytes, la cabecera tendría la siguiente forma:

`Content-Length: 3554\r\n`

3. Contenido. El propio archivo solicitado por el cliente. Aunque el archivo puede ser de texto (html) o binario (png), debe convertirse siempre a bytes antes de ser enviado.

Detalles importantes:

- Entre las cabeceras y el contenido tiene que existir siempre una línea vacía (solamente `\r\n`).
- Para enviar un mensaje al cliente, es mejor utilizar `OutputStream` y enviar chorros de bytes. Sin embargo, para leer las peticiones del cliente podemos utilizar `BufferedReader` porque no es necesario preparar al servidor para recibir archivos (binarios o no) en el contenido.
- Los hilos del servidor deben morir en cada petición del cliente, no se reutilizan en diferentes peticiones.
- Para identificar el tipo MIME de cada archivo servido hay que atender a la extensión del mismo. En caso de utilizar la función `split`, y separar el nombre del archivo de su extensión aprovechando que están separados por un punto '.', es importante recordar que hay que poner `split("\\.")`, porque escribir `split(".")` no tendrá el resultado deseado.