

T1-REDES-anot.pdf



Olmar_eps



Redes de Comunicaciones II



3º Grado en Ingeniería Informática



**Escuela Politécnica Superior
Universidad Autónoma de Madrid**

WUOLAH + BBVA

Hazte **cliente de BBVA y...**
ahórrate 6 meses
de suscripción

BOOM

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

NETFLIX

Spotify

HBOmax

Disney+

PlayStation Plus

DAZN

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

Abre tu cuenta



Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

CAPÍTULO 1: CAPA DE APLICACIÓN

PARTE I: INTRODUCCIÓN

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

PRINCIPIOS

Características de las aplicaciones

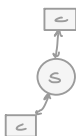
Los programas deben tener las siguientes características:

- Ejecución en varios sistemas.
- Comunicarse a través de la red.

No se necesita escribir software para los elementos de red, ya que éstos no ejecutan aplicaciones de usuario.

Arquitecturas de aplicación

Cliente-Servidor:



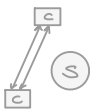
- Siempre existe un host activo (servidor), que da servicio a solicitudes de muchos otros hosts (clientes).
- Los servidores tienen una IP permanente, y ya que un servidor es incapaz de responder a todas las solicitudes de clientes se utilizan las agrupaciones (clusters) de hosts, para crear un servidor de gran capacidad (granjas de servidores).
- Los clientes pueden conectarse intermitentemente, pueden tener IP dinámica y no se comunican entre sí.

conocida por cliente

S no tiene por qué saber IP C

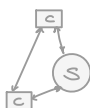
P2P:

TCP o UDP



- Descentralizada: ausencia de servidor central. Los participantes pueden comunicarse entre sí (peers).
- Distribuida: la información no está alojada en un único sitio.
- Carga balanceada: se intenta equilibrar la carga entre todos los participantes.
- Redundancia de información: se duplica información para hacerla más accesible.
- Alta disponibilidad: la caída de un peer no bloquea el servicio.

Híbridas:



- Skype: servidor centralizado permite encontrar la dirección remota con la que se quiere conectar. Conexión directa cliente-cliente.
- Mensajería instantánea: servidor para la localización y las IPs. Comunicación entre usuarios P2P.

Abre tu cuenta



WUOLAH
+ BBVA

WUOLAH

COMUNICACIÓN ENTRE PROCESOS

Son los procesos los que se comunican.

- Mismo sistema informático: se comunican utilizando la comunicación entre procesos (IPC) definida por cada S.O.
- Distinto sistema informático: se comunican utilizando el intercambio de mensajes a través de la red.

Una aplicación consta de una pareja de procesos:

- Cliente: proceso que inicia la comunicación.
- Servidor: proceso que espera peticiones de los clientes.

Las aplicaciones P2P tienen procesos clientes y servidores en todos los sistemas informáticos en los que están funcionando.

Sockets

Interfaz de Programación de Apps
|||

Un proceso envía mensajes a la red y los recibe de la red usando sockets (API), que son análogos a puertos. Oculta detalles de comunicación y está por encima del nivel de transporte.

Direccionamiento de procesos

Los procesos, para que reciban mensajes, tiene un **identificador único**. Un host tiene una **dirección IP** única de 32 bits.

Para identificar a un proceso en un host, no es suficiente con la dirección IP, también hay que incluir un **número de puerto**.

Protocolos

Hay dos tipos de protocolos:

- **De dominio público (abiertos)**: definidos en los RFCs y permiten interoperabilidad. *ej: HTTP, SMTP*
- **Propietarios**: el acceso a ellos está restringido por las compañías propietarias y no permiten interoperabilidad. *ej: SKYPE*

Servicios al transporte en la aplicación

Transferencia de datos confiable: la mayoría requieren transferencia 100% confiable. Audio/video pueden perder paquetes, pero los ficheros NO. La pérdida depende de la densidad de tráfico de la red.

Tasa de transferencia: hay aplicaciones que son sensibles al ancho de banda, es decir, que tienen requisitos de ancho de banda (*ej: apps multimedia*), y hay otras que son elásticas, que no tienen requisitos de ancho de banda (*ej: FTP, e-mail*).

Retardo: algunas aplicaciones requieren bajo retardo para ser efectivas.

Seguridad: integridad de datos, encriptación, autenticación, etc.



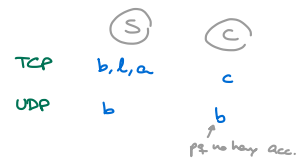
Hazte cliente de BBVA y ... **ahórrate 6 meses** **de suscripción**

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

NETFLIX**HBOmax** **Spotify** **PlayStation.Plus**[Abre tu cuenta](#)

b: bind → registra socket en un puerto
l: listen → disposición a recibir llamadas por un puerto.
a: accept → (bloq. hasta petición) asigna socket a puerto
c: connect → conexión a un servidor

PARTE II: PROGRAMACIÓN DE SOCKETS



CARACTERÍSTICAS

Servicios proporcionados

Se definen diferentes tipos de socket en función de los servicios proporcionados:

- **Stream sockets:** utilizan TCP. Confiable y orientado a conexión. ej: FTP
- **Datagram sockets:** utilizan UDP. No confiable ni orientado a conexión. ej: NFS
- **Raw sockets:** acceso a niveles más bajos de TCP/IP. ej: app pings (para crear nuevos protocolos)

TIPOS DE SERVIDORES

- **Iterativos:** procesan las peticiones de 1 en 1 (no se usa).
- **Concurrentes:** El proceso padre acepta la conexión, que es procesada por un proceso hijo o thread. Crear un proceso/hilo es costoso ⇒ pre-creamos procesos/hilos con un pool.

PARTE III: HTTP (HyperText Transfer Protocol)

INTRODUCCIÓN

Una página web está constituida por objetos y por un fichero HTML base que incluye referencias a esos objetos, dónde cada objeto es direccionable por una URL.

HTTP es el protocolo de la capa de aplicación para la Web. Sigue un modelo Cliente-Servidor:

- Cliente: es un "browser" que solicita, recibe y muestra los objetos Web.
- Servidor: el servidor recibe peticiones, responde y envía objetos.

FUNCIONAMIENTO HTTP

Utiliza **TCP**:

- 1) El cliente inicia la conexión TCP con el servidor (puerto 80).
- 2) El servidor acepta la conexión TCP del cliente.
- 3) Se intercambian mensajes HTTP entre el browser y el servidor.
- 4) Se cierra la conexión TCP.

HTTP **no conserva el estado**, es decir, no mantiene información sobre las peticiones del cliente. (NO MEMORIA)

Tiempo de respuesta RTT

RTT: tiempo ocupado en enviar un paquete pequeño desde el cliente al servidor y su regreso.

Tiempo de respuesta está compuesto por:

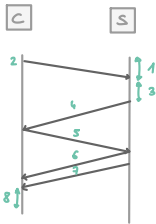
- RTT para iniciar la conexión.
- RTT por requerimiento HTTP y primeros bytes de la respuesta.
- Tiempo de transmisión del archivo.

Tipos de conexiones HTTP

- **HTTP no persistente**: como mucho se envía un objeto por conexión TCP. HTTP/1.0 *X conexiones independientes*
- **HTTP persistente**: pueden enviarse múltiples objetos por conexión TCP entre el cliente y el servidor. HTTP/1.1 *1 conexión que "encapsula" X peticiones y respuestas del mismo servidor*

HTTP no persistente

- 1) El servidor espera conexiones en el puerto 80.
- 2) El cliente HTTP inicia la conexión TCP con el servidor en el puerto 80.
- 3) El servidor acepta la conexión.
- 4) El servidor le comunica al cliente la aceptación.
- 5) El cliente HTTP envía el mensaje de solicitud al socket TCP con la URL, indicando que solicita un objeto.
- 6) El servidor recibe la petición y envía un mensaje de respuesta conteniendo el objeto solicitado y lo envía a su socket.
- 7) El servidor cierra la conexión TCP.
- 8) El cliente recibe el mensaje de respuesta conteniendo el objeto solicitado, lo analiza, lo presenta y encuentra n referencias a otros objetos.



Con las referencias encontradas el cliente repite todo el proceso hasta que las n referencias a objetos estén resueltas.

Problemas:

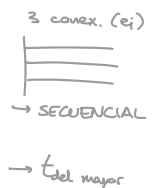
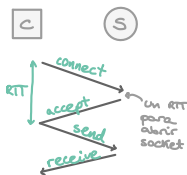
- Requiere 2 RTTs por objeto.
- El SO debe trabajar y dedicar recursos para cada conexión TCP.
- El navegador abre conexiones paralelas para traer objetos referenciados.

HTTP persistente

El servidor deja las conexiones abiertas después de enviar la respuesta. Los siguientes mensajes entre los mismos cliente/servidor son enviados por la conexión abierta.

Dos tipos:

- **Sin "pipelining"**: el cliente envía nuevo requerimiento sólo cuando el previo ha sido recibido. Un RTT por cada objeto referenciado.
- **Con "pipelining"**: el cliente envía requerimientos tan pronto éste se encuentra un objeto referenciado. Un RTT por todas las referencias.



Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

PETICIÓN HTTP

- **GET:** obtiene un recurso del servidor. Los datos son visibles en los logs del servidor Web. Puede codificar parámetros en la URL. No tiene cuerpo.
- **POST:** manda datos al servidor. Usado para el envío de datos de formularios Web. Los datos se codifican en el cuerpo, y suelen ser procesados por un script en el servidor. Tiene cuerpo
- **HEAD:** obtiene sólo las cabeceras de un recurso. Usado para obtener información de un recurso sin descargarlo. No tiene cuerpo.

Códigos de Respuesta

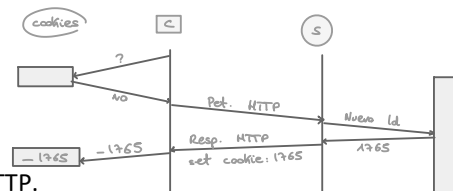
1__: información, 2__: éxito, 3__: redirección, 4__: error cliente, 5__: error servidor

COOKIES

Permiten a los sitios web seguir la pista a los usuarios.

Utiliza 4 componentes:

- Línea de cabecera cookie en el mensaje respuesta HTTP.
- Línea de cabecera cookie en el mensaje de petición HTTP.
- Archivo cookie almacenado en la máquina del usuario y administrada por su navegador.
- Base de datos sitio Web.



WEB CACHÉ (SERVIDORES PROXY)

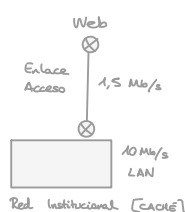
Servidor Proxy, o Caché Web entidad que satisface solicitudes HTTP en nombre de un servidor web de origen. Actúa como cliente y servidor.

Utilidades:

- Almacena copia de los objetos recientes.
- Proceso: el cliente solicita un objeto, el proxy mira por ese objeto, si lo tiene lo manda como respuesta, si no el proxy pregunta al servidor de origen. Cuando lo obtiene lo envía al cliente.
- Reduce el tiempo de respuesta.
- Reduce tráfico en los servidores de origen.
- **GET condicional:** utiliza este mecanismo para saber si los objetos están actualizados. NO enviar un objeto si el Web-Caché está actualizado.

Retardo total = ret. Internet + ret. acceso + ret. LAN

Enviar solo si la que tengo NO es actualizada
↳ mirando LastModified



$$r_{\text{acceso}} = \frac{\text{tam. pet.}}{\text{Mb/s}} = (s)$$

→ Si hay caché, las pet. no saldrán a Internet ⇒ ↓ tráfico acceso

Nuevo r_{acceso} con menos pet/s.

$$r_{\text{total}} = 0.6 (r_{\text{int}} + r_{\text{acceso}}) + 0.4 (r_{\text{lan}})$$

cache 40%

↳ SUBRED

WUOLAH

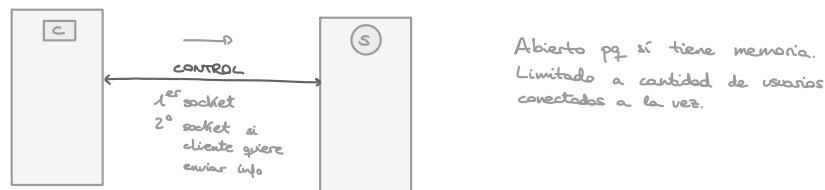
PARTE IV: FTP (File Transfer Protocol)

CARACTERÍSTICAS

FTP transfiere ficheros entre dos sistemas. Es un servicio tanto de perfil público (entrando con user anonymous), como de perfil privado (usando user y password).

ARQUITECTURA

El cliente está conectado al servidor mediante una conexión de control. El cliente envía comandos contestados por el servidor. Estos comandos permiten trabajar con ficheros.



Si el cliente pide un directorio o una transferencia de archivo, se establece una nueva conexión de datos. Finalizada la transferencia, la conexión de datos se cierra.



MODOS DE TRANSMISIÓN

- **Modo stream:** los datos se envían tal como están en el fichero.
- **Modo bloque:** se incluyen cabeceras con cada bloque.
- **Modo comprimido:** se realizan operaciones de compresión y descompresión de los datos antes y después de la transferencia. Poco utilizado.

CÓDIGOS DE RESPUESTA

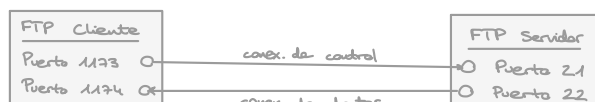
- 1XX: Inicio de acción con éxito.
- 2XX: Comando realizado con éxito.
- 3XX: Respuesta intermedia con éxito.
- 4XX: Errores pasajeros.
- 5XX: Errores permanentes.

APERTURA DE CONEXIÓN

Se comunica que el cliente aceptará una conexión por el puerto 1174. Cuando se hace a conexión de control se abren dos puertos.



Se realiza la conexión de datos por parte del servidor.



COMANDO PASV

Por seguridad, no es conveniente que un cliente FTP acepte solicitudes de conexiones de datos.

podir IP del cliente y puerto

En vez de utilizar PORT, el cliente envía un comando PASV al servidor, que le contesta con un puerto del servidor para la conexión de datos.



MODOS DE CONEXIÓN

- **Modo activo:** (por defecto) el servidor inicia la conexión de datos. → *Problemas firewall cliente*
- **Modo pasivo:** el cliente inicia las conexiones de control y datos. → *Abrir rango puertos servidor*

SERVICIO DE REINICIO

El FTP emisor transmite bloques con marcadores de reinicio en puntos adecuados de la transferencia de datos. Cada marcador es una cadena de texto. Cuando el receptor recibe un marcador, almacena los datos en el disco con la posición del marcador.

Si el receptor es el cliente, notifica la recepción del marcador a la aplicación FTP.

Si el receptor es el servidor, se notifica al cliente por la conexión de control la recepción del marcador.

Si se produce un fallo, se puede reiniciar indicando el valor del marcador adecuado.

PARTE V: SMTP (Simple Mail Transfer Protocol)

COMPONENTES

- **Agente de usuario:** es el lector de correo. Escritura, edición y lectura de mensajes de correos. Mensajes de salida y entrada son almacenados en el servidor.
- **Servidor de correo:** Compuesto por: **casillero** (o **buzón de entrada**) que contiene mensajes de entrada del usuario y una **cola de mensajes** de los mensajes de salida.
- **Protocolo SMTP:**
 - o Usa TCP en el puerto 25.
 - o Transferencia directa: servidor envía correos al servidor receptor.
 - o Tres fases: Handshaking, Transferencia de mensajes y Cierre.
 - o Interacción comandos/respuestas: los **Comandos** son texto ASCII, mientras que las **Respuestas** son códigos de estado y frase.
 - o Mensajes enviados en ASCII de 7 bits. (modo texto 7 bits, binario 8 bits)
 - o CR/LF para terminar el mensaje.

COMPARACIÓN CON HTTP

HTTP pull (saca contenido desde servidor) – SMTP push (mete contenido en servidor).

HTTP cada objeto es encapsulado en su mensaje – SMTP múltiples objetos enviados en un mensaje.

Ambos tienen interacción comando/respuesta en ASCII y códigos estatus.

PROTOCOLOS DE RECEPCIÓN DE CORREO ELECTRÓNICO

- Puerto 110* - **POP** (Post Office Protocol): POP3:
 - o No mantiene el estado de una sesión a otra (stateless).
- Puerto 143* - **IMAP** (Internet Mail Access Protocol):
 - o Permite manipulación de los mensajes almacenados en el servidor.
 - o Mantiene todos los mensajes en el servidor.
 - o Mantiene el estado de un usuario de una sesión a otra.
- **HTTP:** Analiza el servidor localmente y presenta los resultados al usuario vía web.
- **MIME:** Añade al SMTP mejoras como incluir objetos distintos a texto ASCII.

Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

PARTE VI: DNS (Domain Name System)

CARACTERÍSTICAS

Utilizado por aplicaciones TCP-IP para obtener la dirección IP a partir del nombre del host.

La aplicación cliente que se encarga de la resolución de nombres se denomina **resolver**. El resolver obtiene la información consultando un fichero *hosts* en la máquina y contactando con un servidor de nombres. (*gethostbyname* y *gethostbyaddress*)

→ ref. IP → ref. nombre dom.

Este servicio opera tanto sobre UDP como TCP en el puerto 53.

Una organización debe disponer de un Servidor Local de Nombres donde se encuentra la base de datos con los nombres de las máquinas y sus direcciones IP.

Cualquier consulta desde un cliente de la organización sobre una máquina interna será resuelta por el servidor local. En cambio, cualquier consulta sobre una máquina externa será resuelta por el servidor local que contactará con el servidor del *host* externo. La respuesta será enviada al cliente y almacenada en la caché del servidor local.

El servidor, antes de contactar con un servidor externo, consulta si tiene dicha información en la caché.

ORGANIZACIÓN

Estructura DNS

Estructura jerárquica en forma de árbol. Cada **nodo** tiene una etiqueta (máx. 63 caracteres) excepto el root. Cada nodo se asigna a una **autoridad** que es responsable del nodo.

El **nombre de dominio** se forma añadiendo etiquetas separadas por un punto, desde el nodo terminal hasta el root. *ej: il.uam.es*

El **FQDN** (Full Qualified Domain Name) de un host se forma añadiendo el nombre de la máquina al dominio al que pertenece. *ej: delbod.il.uam.es*

En el nivel superior del árbol se encuentran:

- **Nodo raíz:** dominio utilizado para resolver direcciones a nombres.
- **Dominios genéricos asociados a actividades:** com, edu, net, etc. -> gTLD (generic) no administra ICAANN
- **Dominios geográficos de 2 caracteres:** es, it, fr, etc. -> ccTLD (country code) no administra PAIS

Los dominios de nivel superior son administrados por el NIC.

DNS LOCAL → ¿IP de www.uam.es?

DNS RAÍZ → ¿DNS para .es?

DNS TLD → ¿DNS para uam.es?

DNS PRIMARIO → ¿IP uam.es?

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.



Abre tu cuenta



WUOLAH
+ BBVA

WUOLAH

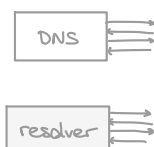
Servidor raíz

La administración de una zona, rama del árbol DNS, se delega en una organización. Un servidor de nombres es autoridad para dicha zona, debiendo tener en su base de datos los nombres de máquinas y direcciones IP de la zona. Una zona puede englobar a varios nodos del árbol DNS.

Además del servidor primario de nombres, pueden existir servidores secundarios que son autoridad para la zona. Su información es redundante y actúan en caso de fallo del servidor primario.

Cada servidor de nombres conoce la dirección IP del servidor de nombres del nivel superior y del nivel inferior del árbol DNS.

Tipos de consulta a servidor DNS



- Recursiva: el servidor de nombres dará la respuesta completa haciendo las subconsultas necesarias.
- Iterativa: el servidor de nombres da una respuesta parcial y el cliente “sigue preguntando”.
- Inversa: preguntan por dominio sabiendo IP. Se resuelven a través de consultas iterativas o recursivas.

Cachés y TTL

Las respuestas almacenadas en caché tienen un **tiempo de vida, TTL**, transcurrido el cual, si no es utilizada de nuevo, la respuesta se elimina de la caché.

Objetivo: evitar desbordamiento cachés

FORMATO DEL PROTOCOLO

Pregunta inversa

Solicitud de un nombre a partir de su dirección IP.

Cada organización debe mantener un servidor de nombres para resolver estas preguntas. Estos servidores se encuentran en la organización DNS por debajo del dominio arpa.in-addr.

El campo nombre en estas preguntas debe indicarse con la dirección IP en orden inverso. ej: para conocer el nombre de la máquina con IP 150.244.28.254 debe indicarse 254.28.244.150.in-addr.arpa.

Validación por respuesta inversa

El servidor se asegura al cliente que se quiere conectar viendo que las IPs y los nombres coinciden.

NS: Servidor de nombre
A: IP
PTR: Nombre asignado a IP
MX: Intercambiador de correo
↳ comprueba servidor receptor

→ no puede cambiar puerta por defecto HTTP ni asignarle IP variable.
→ permite asignar IPs a determinados ordenadores identificados por su MAC.

PARTE VII: DHCP (Dynamic Host Configuration Protocol)

CARACTERÍSTICAS

Información suministrada por DHCP para configurar TCP-IP: Dirección IP, Máscara de subred, Router por defecto.

Ventajas: facilita la administración y reduce el número total de direcciones IP.

Inconveniente: información del DNS no actualizada, por lo que hay que usar Dynamic DNS (DDNS).

Usa UDP en el puerto 67.

DHCP soporta tres mecanismos para asignar direcciones:

- **Automática**: la dirección IP asignada es permanente para el cliente.
- **Dinámica**: la dirección IP asignada está limitada a un periodo de tiempo.
- **Manual**: la dirección IP se configura manualmente en el cliente.

OPERACIÓN DHCP

origen: 0.0.0.0.
destino: 255.255.255.255

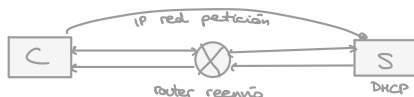
- 1) El cliente manda en difusión un mensaje **DISCOVER** que puede incluir como opciones una posible dirección IP y el tiempo de validez.
- 2) Cada servidor puede responder con un mensaje **OFFER** que incluye una dirección IP y otras opciones de configuración. Se envía con MAC del cliente y con IP de difusión. Los servidores recuerdan las direcciones ofertadas, evitando ofrecerlas en posteriores mensajes OFFER.
- 3) El cliente elige una de las direcciones IP y manda en difusión un **REQUEST** en el que indica el servidor y la dirección IP elegida. La recepción de este mensaje por los servidores no elegidos, sirven de notificación de que el cliente no ha aceptado su oferta.
- 4) El servidor elegido manda un mensaje **PACK** confirmando la dirección IP y otros parámetros de configuración, así como dos tiempos T1 y T2 relacionados con el tiempo máximo T en el que la dirección IP es válida.
- 5) Cuando el temporizador T1 expira, el cliente manda un REQUEST (unicast) al servidor solicitando extender el tiempo de validez de la configuración. Si el servidor está de acuerdo, manda un mensaje PACK con los nuevos valores de T1 y de T2.
- 6) Si no llega el mensaje PACK, cuando el temporizador T2 expira, el cliente manda un mensaje REQUEST (difusión). Esta solicitud puede ser contestada por alguno de los servidores con un mensaje PACK.
- 7) Si el cliente no recibe PACK y vence el tiempo T, el cliente debe dejar de usar la configuración IP actual e iniciar el proceso con un mensaje DISCOVER.
- 8) Un cliente puede decidir dejar de usar la configuración IP asignada enviando un mensaje **RELEASE** al servidor.

AGENTE ROUTER DE REENVÍO

Se emplea en el caso de que el servidor DHCP no esté en la misma subred que el cliente.

Cuando el agente detecta una solicitud introduce la dirección IP de la red por la que llegó la petición en el campo Router Relay y lo reenvía al servidor remoto DHCP.

La respuesta desde el servidor se hace a través del agente **Router de Reenvío**.



PARTE VII: P2P

ARQUITECTURA P2P PURA

No tiene necesidad de servidores. Se comunican dos sistemas finales. Los pares se conectan intermitentemente y cambian sus IPs.

DISTRIBUCIÓN

Cliente-Servidor vs P2P

¿Cuánto se tarda en distribuir un fichero en ambas arquitecturas?

us: Ancho de banda de subida en el servidor *Mbps no velocidad*

ui: Ancho de banda de subida del par i

di: Ancho de banda de bajada del par i

▪ **Tiempo de distribución Cliente-Servidor**

El servidor manda N copias de un fichero de tamaño F:

$$\boxed{\frac{NF}{u_s}} = t_{\text{min servidor para subir datos}}$$

El cliente i tarda en descargar el fichero:

$$\boxed{\frac{F}{d_i}} = t_{\text{min cada cliente para descargar fichero}}$$

Por tanto, el tiempo T que tarda el servidor en distribuir el fichero a los N clientes es:

$$\boxed{d_{cs} \geq \max \left(\frac{NF}{u_s}, \frac{F}{\min(d_i)} \right)}$$

que para grandes N crece linealmente.

Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

■ Tiempo de distribución P2P

Gran ventaja P2P: ESCALABILIDAD (transferencias aceptables, aunque crezca el número de clientes) y no hay punto de fallo.

El servidor tiene que mandar la copia inicial:

$$\frac{F}{u_s}$$

El más lento de los nodos tardará en recibir F:

$$\frac{F}{\min(d_i)}$$

El ancho de banda de la red es:

$$u_{total} = u_s + u_1 + \dots + u_N$$

Por otro lado, el sistema debe transmitir en total $N \cdot F$ bytes, que no puede hacerse nunca a una tasa de transferencia mayor a $u_s + \sum_{i=1}^N u_i$. Por tanto, el tiempo mínimo de distribución es:

$$\frac{N \cdot F}{u_s + \sum_{i=1}^N u_i}$$

Por tanto, el tiempo total es:

$$d_{P2P} \geq \max \left(\frac{F}{u_s}, \frac{F}{\min(d_i)}, \frac{N \cdot F}{u_s + \sum_{i=1}^N u_i} \right)$$

dominante cuando $N \rightarrow \infty$

Conclusión: P2P es mejor.

BIT TORRENT

Para transferencia de archivos grandes

Arquitectura

Tracker: rastrea a los participantes en Torrent.

Torrent: grupo de pares intercambiando fragmentos (chunks) de ficheros.

Características

Recogiendo fragmentos:

En cualquier momento, diferentes pares tienen subcolecciones de fragmentos del mismo fichero diferentes. Periódicamente, un par les pide a sus vecinos la lista de fragmentos. Este par solicita los fragmentos que no tiene a sus vecinos (siempre se pide primero el menos reproducido, el más raro).

Enviando fragmentos (toma y daca, o tit-fot-tat):

Mejores planos para el intercambio. Permite descargar el fichero de forma + rápida

Un determinado par envía sus fragmentos a los cuatro vecinos que le envían fragmentos a mayor ritmo. Se vuelve a evaluar el top 4 cada 10s. Cada 30s se selecciona otro par e inicia envío de fragmentos. EL nuevo par siempre estará en el top 4 en esa ronda. De forma optimista deshace atascos de pares parados.

DHT (Distributed Hash Table)

Descripción

DHT es la base de datos distribuida de P2P.

Esta base de datos contiene pares (**clave, valor**), donde la clave es el identificador de usuario (nombre) y el valor es el identificador de contenido (dirección IP).

Los pares consultan la base de datos con la clave y la base de datos responde con el valor. Los pares solo pueden insertar pares (clave, valor) en la base de datos.

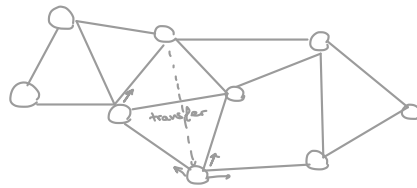
Identificadores DHT

Asigna un id a cada par (peer) en el rango $[0, 2^n - 1]$. Todas las claves tienen que estar en el mismo rango. Si la clave es un literal, hay que crear un "hash" de éste.

Primeros esquemas P2P basados en flooding

Ventajas: resistencia a censura, escalabilidad, encontrar contenido popular.

Inconvenientes: encontrar contenido "poco" popular, vulnerabilidades como envenenamiento de datos y tracking, no hay garantías de privacidad.



Pregunta a los conectados, si no lo tienen, estas preguntan a sus conectados.
Funciona bien con pocos nodos.

DHT CIRCULAR

Protocolo Chord

Los nodos se distribuyen en anillo. No hay redundancia, si se cae el peer pierdes la información.

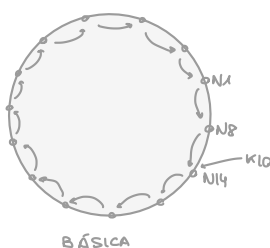
Cada clave k se almacena en el sucesor del nodo cuyo *nodeid* es mayor o igual a *keyid*.

▪ Búsqueda básica

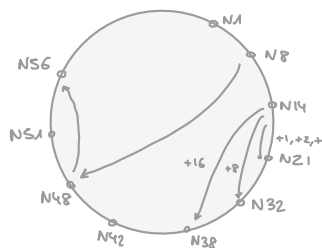
No inunda la red pero las cadenas de búsqueda son largas. Rendimiento pobre, en el caso peor, todos los nodos deben transmitir un mensaje.

▪ Búsqueda eficiente: uso de *fingers* (atajos) → para reducir tiempo de búsqueda de un contenido en la red

Para equilibrar número de fingers con longitud de cadenas el coste es $\log(n)$ y para ello hago uso de tablas de rutas.



BÁSICA



TABLA

Índice	Nodo
$N14 + 1$	N21
$N14 + 2$	N21
$N14 + 4$	N21
$N14 + 8$	N32
$N14 + 16$	N38

$$\text{finger}(i) = \text{sucesor}((id + 2^i) \bmod 2^{160})$$