Tema 2 - Capa de aplicación

- Creación de una aplicación de red
  - Programas que corren en diferentes hosts
  - se comunican por la red

- **Cliente - servidor**
  - Servidor:
    - host siempre activo
    - IP permanente
    - clusters para mejor escalabilidad
  - Cliente:
    - se comunican con el servidor
    - conexión intermitente
    - IP's dinámicos

- **P-2-P**
  - Sin servidor siempre activo
  - Se comunican directamente
  - Muy escalable, difícil de gestionar

- **Híbrido Cliente - servidor y P2P**
  - ej: Skype, mensajería instantánea
  - El servidor central gestiona las direcciones IP

- **Comunicación de procesos**

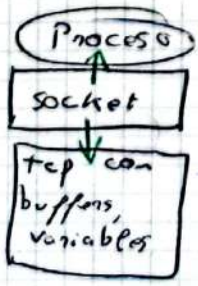
Proceso: programa que corre en un host

- 1 host, 2 procesos se comunican por sistemas definidos por el S.O. (inter-process communication)
- Los procesos en hosts distintos, se comunican por mensajes

- Proceso cliente: El que inicia la comunicación
- Proceso servidor: El que espera a que contacten con el

Sockets

Un proceso envía/recibe mensajes a través de/desde su socket



- El host tiene una IP de 32 bits

Direccionar procesos

- El identificador incluye una IP y el Nº de puerto asociado al proceso de ese host.

  128.119.245.12:80

- Protocolo
  - tipos de mensajes intercambiados (GET, POST)
  - Sintaxis del mensaje (que campos)
  - Semántica de los mensajes (significado contenido campos)
  - Reglas de envío y respuesta de mensajes

  ej:
  - Públicos: HTTP
  - Privados: Skype

¿Que servicios de transporte necesita una app?

- Pérdida de paquetes
  Algunas lo toleran y otras no (TCP, UDP)
- Temporización
  Algunas app requieren bajo retardo (Juegos)
- Tasa de transferencia
  Algunas requieren una tasa mínima (Netflix)
- Seguridad
  Encriptación

Servicios

- TCP
- UDP

- Web y HTTP
  - Una página web consta de objetos
  - Primero se solicita la página base (archivo HTML base)

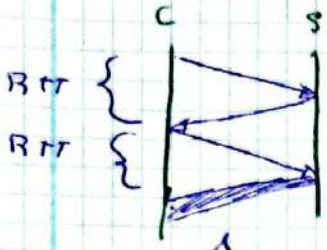
  www.uah.es / notas / foto.jpg

  Nombre host — Ruta

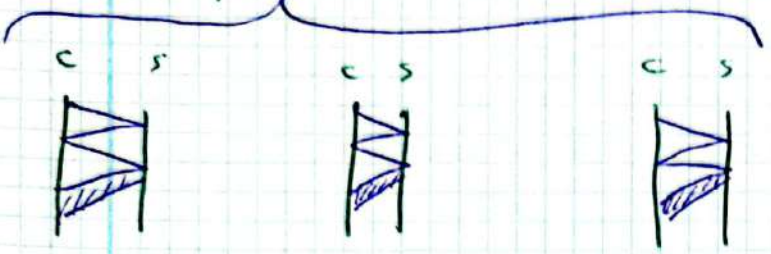
Conexiones HTTP no persistentes
Como máximo, 1 objeto a través de una conexión TCP

Conexión HTTP persistente
Múltiples objetos a través de una conexión TCP

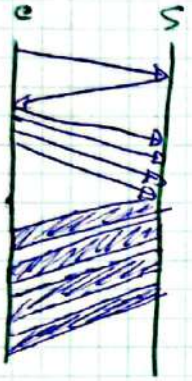
- No persis "En paralelo"



- RTT (tiempo de ida y vuelta)
- Tiempo de respuesta = 2RTT + tiempo de transmisión



Persistente "En cascada"



Método - URL - Versión

ejm

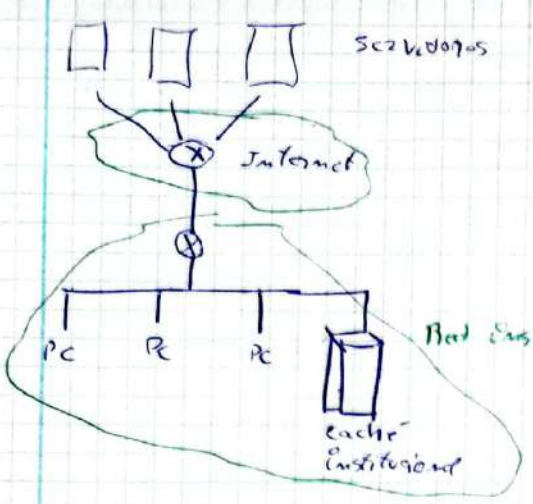
Get /fruit/kiwi.gif HTTP/1.1

(5.3) Cookies

- Componentes
  1. Línea de cabecera de la cookie en la respuesta HTTP
  2. Línea de cabecera de la cookie en la solicitud HTTP
  3. El archivo de cookie está en el host del usuario, gestionado por su navegador
  4. Base de datos de respaldo (back-end) en el sitio web
- Aporta
  - Autorización
  - Carritos de compra
  - Recomendaciones
- Pero
  - Permiten a los sitios saber mucho de ti
  - Proporcionan nombres y direcciones de e-mail

Caché web

- También denominado servidor proxy
- Cumple las solicitudes del cliente sin enviarlas al servidor de origen
- Normalmente son los ISP
- Reduce el tiempo de consulta de un cliente
- Reduce el tráfico



- El navegador envía todas sus solicitudes a la caché web
  - Acierta: devuelve el objeto
  - Fallo: se lo pide al servidor y se lo reenvía al cliente

Get condicional

- Verifica si los objetos de la caché web están actualizados
- El mensaje get contiene como cabecera
  If-modified-since: <Fecha>

Ejm
Get /fruit/kiwi.gif HTTP/1.1
Host: www.exo.com
If-modified-since: Wed, 4 Jul 2007 09:34:27

- El servidor devuelve el objeto únicamente si ha sido modificado

Ejm
HTTP/1.1 304 Not Modified

FTP

File Transfer Protocol

- Transferir un archivo desde un host a otro
  - El usuario proporciona el nombre del host remoto
  - se establece una conexión TCP con el host remoto
  - El usuario proporciona su identificación y contraseña, se envían y una vez verificados y autorizados.
  - El usuario copia 1 o más archivos locales.
- HTTP, FTP son protocolos FTP: RFC 959 FTP: puerto 21
- se utilizan 2 conexiones TCP
  - Una conexión de control
  - Una conexión de datos
- La conexión de control está "fuera de banda"

FTP Comandos, respuestas

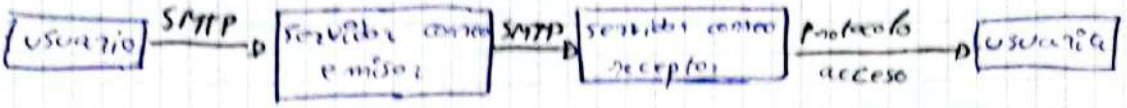
Ejemplos comando	Ejemplos código
• Se envían como texto ASCII por el canal de control	
- USER usuario	- 331 USERNAME OK, Password required
- PASS contraseña	
- LIST devuelve lista de archivos en el directorio	- 125 data connection already open
- RETR Recupera un archivo (get)	- 425 Can't open data connection
- STOR Almacena un archivo (PUT) en el host remoto	- 452 Error writting file

Correo electrónico

- Medio de comunicación asíncrono
- 3 componentes principales
  - Agentes de usuario
  - Servidores de correo
  - Protocolo simple de transferencia de correo (SMTP)
- Los agentes de usuario permiten a los usuarios leer, responder, ... (outlook)
- Los servidores de correo tienen un buzón de correo, una cola de mensajes salientes y un SMTP entre servidores para enviar mensajes.
- Se realiza una transferencia directa entre emisor y receptor
- Utiliza el puesto 25
- 3 fases de transferencia
  - 'handshaking' (establecimiento de conexión)
  - Transferencia de mensajes
  - Cierre
- Formato
  - cabecera
    - To
    - From
    - subject
  - cuerpo Mensaje solo caracter ascii

HTTP envía archivos de cliente web a otro
SMTP envía archivos de un servidor de correo a otro | Protocolo pull / Push

• Protocolos de acceso para correo



• Existen 2 alternativas para que el usuario final reciba los correos

- Protocolo de oficina de correos versión 3 (POP3)

- El usuario establece una conexión TCP con el servidor : 110
- POP3 tendrá 3 fases

  • Autorización: El usuario envía su nombre de usuario y contraseña
  • Transacción: El usuario recupera los mensajes
  • Actualización: Cuando el cliente hace close, termina la sesión

- IMAP

• Carpetas y mensajes se guardan en servidor (POP3 en local)
• Asocia cada mensaje a una carpeta
• Manipulación de mensajes en el propio servidor

- HTTP (gmail, hotmail, ...)

• Se conecta a una página web y utilizarán el mismo HTTP en lugar de IMAP, POP3.

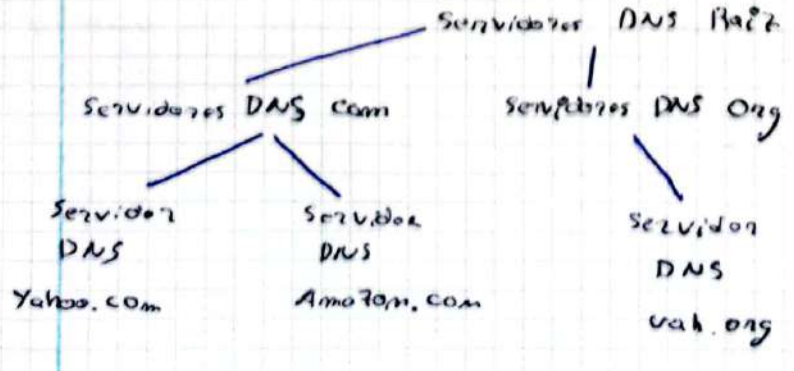
DNS (Domain Name System)

• base de datos distribuida implementada en jerarquía de servidores de nombres

• Traduce los nombres a direcciones IP
• Distribuye la carga de los servidores web

¿Por qué no centralizar DNS?

- Punto críticos para fallos
- Volumen de tráfico
- Base de datos centralizada lejana



Ej:

- Cliente quiere IP de www.Amazon.com
  - Cliente pregunta a servidor raíz por servidor com
  - " " " " com " " amazon.com
  - " " " " amazon.com por la IP

TLD (Top-Level-Domain)

- (Servidores de nivel superior)
  - Responsables de los dominios com, org, edu

- Cada ISP tiene un servidor de DNS, (servidor, local)
  - Primero se consulta este y si no lo encuentra, va al raíz

DNS cacheado

- Se almacena la dirección en la caché, si se vuelve a consultar, no hace falta consultar fuera del ordenador

- **Registro DNS**
  - Formato RR: (nombre, valor, tipo, ttl)
    - Tipo = A
      - nombre = nombre de host
      - Valor = su IP
    - Tipo = NS
      - nombre = su dominio (uah.com)
      - Valor = Nombre del servidor autoritativo para ese dominio
    - Tipo = CNAME
      - nombre = alias de un nombre "canónico" (el real)
        - www.uah.com —NS→ Servereast.backup2.uah.com
      - Valor = nombre canónico
    - Tipo = MX
      - Valor = Nombre de un servidor de correo asociado con nombre

ejem

- "Registrar "chumerías alcalá"
  - Registra el nombre chumerias-alcala.com en un registrador DNS
  - Crear registros en el servidor autoritativo

ejem. Visita m servidores (DNS) → funciona con UDP

Pag web es un objeto pequeño

¿ tiempo en obtenerlo (T_{http})?

$$T_{http} = m \cdot RTT_{DNS} + \underbrace{RTT_{TCP}}_{\text{establecer conexión}} + \underbrace{RTT_{web}}_{\text{obtener objeto}}$$

Si el objeto hace referencia a 8 objetos

a) Conexión NO persistente, En paralelo

$$T_{HTTP} = m \cdot RTT_{DNS} + 2 \cdot (8+1)\, RTT_{web}$$

b) No Persis, 5 conex. paralelas

$$T_{HTTP} = m \cdot RTT_{DNS} + 2\,RTT_{\substack{Pag \\ Base}} + \underbrace{2\,RTT_{web}}_{5\,obj} + \underbrace{2\,RTT_{web}}_{5\,obj}$$

c) Persistente, Paralelo

$$T_{HTTP} = m \cdot RTT_{DNS} + RTT_{TCP} + \underbrace{9 \cdot RTT_{web}}_{\substack{1\,pag \\ 8\,obj}}$$

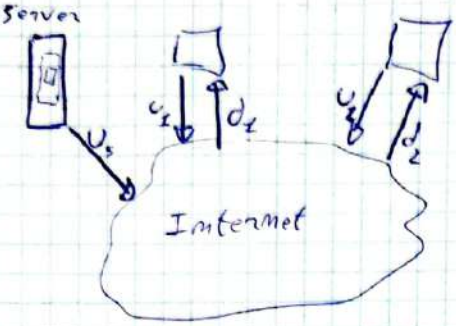
Aplicaciones P2P

3 cuestiones

- Distribución de archivos
- Búsqueda de información
- Caso Skype

Distribución archivos

C-S



- Tiempo en distribuir

F a N clientes = d_{cs} =

$$= \max\left\{\frac{NF}{u_s}, \frac{F}{\min(d_i)}\right\}$$

P-2-P

$$d_{P2P} = \max\left\{\frac{F}{u_s}, \frac{F}{\min(d_i)}, \frac{NF}{(u_s + \sum u_i)}\right\}$$

- Tracker: Registra pares que participan en un torrente
- Torrente: Grupo de pares que intercambian trozos de un archivo

- Los archivos se dividen en trozos (chunks) de 256KB
- Un nuevo par, se conecta a un torrente
  - No tiene trozos, por ahora
  - Se registra en el tracker para obtener lista de pares (vecinos)
- Mientras descarga, el par carga trozos a otros pares

- Extraer trozos
  - Periodicamente, cada par pide a cada vecino la lista de los que tienen
  - Entonces solicita los trozos que le faltan (el menos común primero)
- Envío de trozos: ojo por ojo
  - Un par envía trozos a los 4 vecinos que le envían a él a la velocidad más alta ("top 4" cada 10 seg)
  - Cada 30 seg, elige aleatoriamente otro par y se le empieza a enviar
    - Si entra en el "top 4" o no

• Distributed Hash Table (DHT)

- DHT: base de datos P2P distribuida
- La base tiene duplas (clave, valor) (DNI, Nombre)
- Los pares consultan la BD con la clave
  - ↳ esta devuelve los valores que coincidan con la clave
- Los pares también pueden insertar

• Se asigna un ID a cada par $[0, 2^m-1]$

• Para obtener claves enteras, hacer hash de la original

ej

clave = h ("Led Zeppelin IV")

¿ como asignar claves a pares?

• Asignar clave al par que tenga el ID más cercano

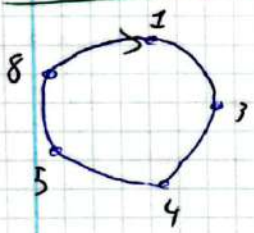
ej

m=4 Pares 1, 3, 4, 5, 8, 10, 12, 14

clave = 13 ⟶ Sucesor par 14

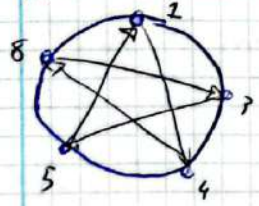
clave = 15 ⟶ Sucesor par 1

DHT circular



• Cada par sabe conoce al siguiente

DHT circular con atajos



• Para manejar el abandono de un par, estos tienen que saber la IP de sus 2 sucesores

• se hace periodicamente un ping a su 2 sucesor para saber si sigue

• Caso Skype

• Deducido por ingeniería inversa

• ↳ jerarquía de su par-pares

↳ se elije un retransmisor para conectar a 2 pares