Contenido 9: Capa Aplicación

Lesly Reyes
lareyes2@uc.cl

Álvaro Quintana
alvaroq@uc.cl

Departamento de Ciencias de la Computación
Pontificia Universidad Católica Chile

October 14, 2024

Contenido

Capa Aplicación
- Capa Aplicación
- Sockets
- Programación Sockets
- HTTP
- FTP
- SMTP, POP, IMAP
- DNS
- P2P

Material Adicional

Capa Aplicación

Aplicaciones se construyen a través de procesos en distintos nodos que se comunican

- Web
- Email
- DNS
- Whatsapp
- Torrent
- etc . . .

Capa Aplicación

Protocolos

Aplicaciones se construyen a través de procesos en distintos nodos que se comunican

- DNS, Domain Name System
- HTTP, Hypertext Transfer Protocol
- FTP, File Transfer Protocol
- SSH, Secure Shell
- TELNET, Remote Shell
- SMTP, Simple Mail Transfer Protocol
- POP, Post-Office Protocol
- IMAP, Internet Message Access Protocol
- NTP, Network Time Protocol
- IRC, Internet Relay Chat
- LDAP, Lightweight Directory Access Protocol

Capa Aplicación

Arquitectura Cliente Servidor

Aplicaciones se construyen a través de procesos en distintos nodos que se comunican

- Clientes se comunican con servidor
  - Clientes no se comunican entre ellos
- Servidor posee una dirección fija y conocida
- Servidor puede recibir múltiples solicitude
  - Requerimientos intensivos en infraestructura
  - Capacidad de atención, ancho de banda, . . .
  - Suelen ubicarse en data centers

Web, FTP, Telnet, SSH, email, . . .

Capa Aplicación

Arquitectura P2P

Aplicaciones se construyen en base a comunicación directa entre miembros (pares)

- Requerimientos mínimos en existencia de servidores
- Facilita auto-escalabilidad
    - Miembros añaden capacidades a la red

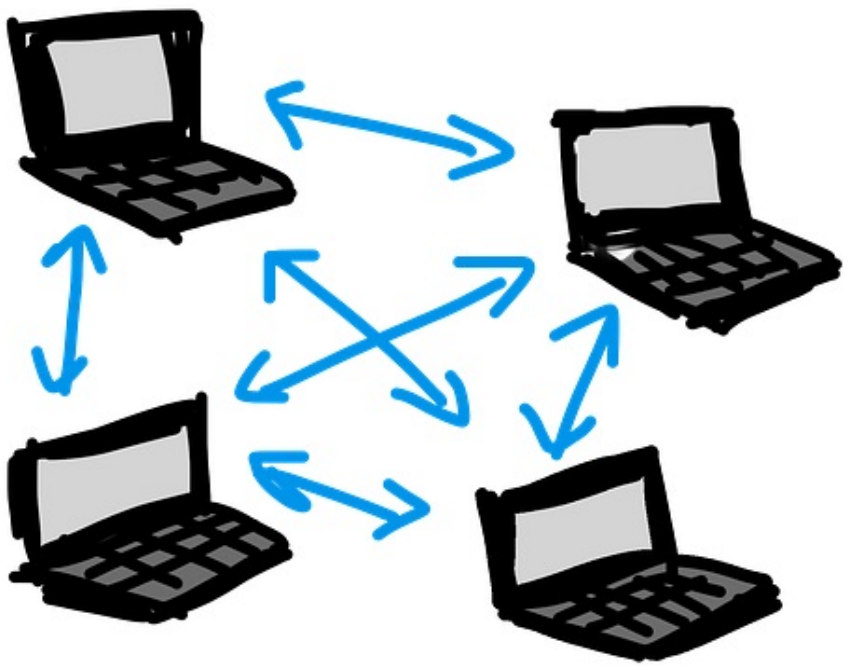
Capa Aplicación

Arquitectura P2P

BitTorrent, download accelerators, telefonía por internet (Skype), IPTV

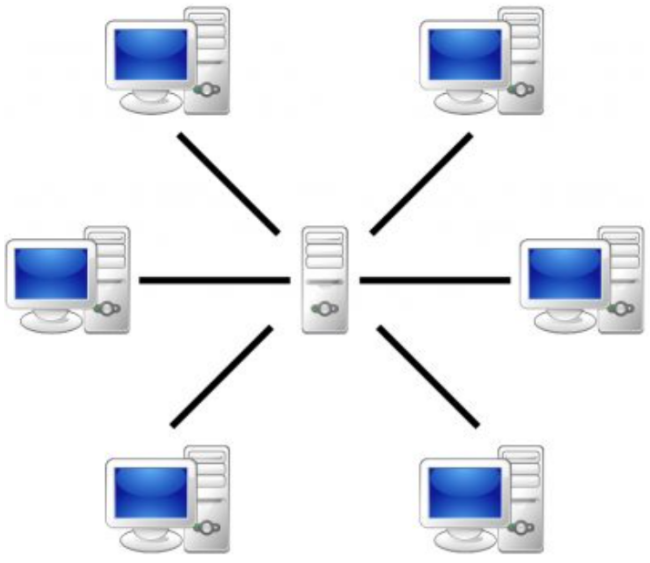
Desafíos

- **ISP Friendly:**Muchas conexiones son asimétricas
- **Seguridad:** Múltiples puntos de vulnerabilidad

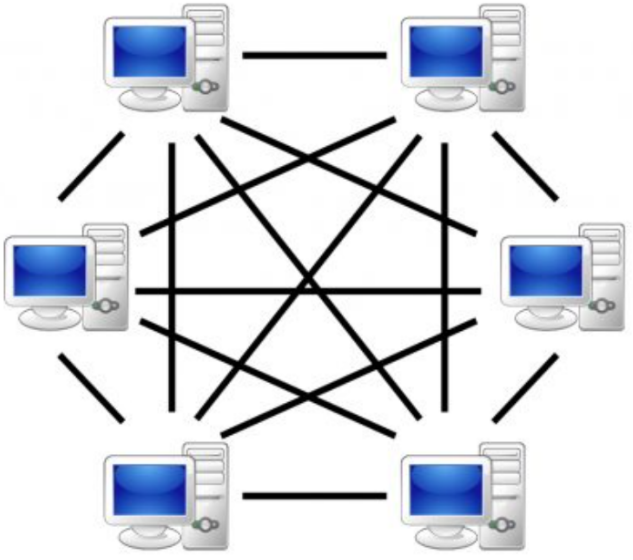


Capa Aplicación

Arquitectura P2P vs Cliente-Servidor



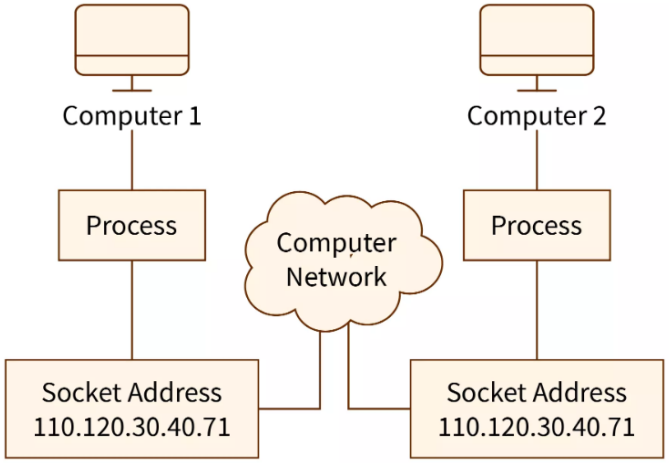
Server-based



P2P-network

Sockets

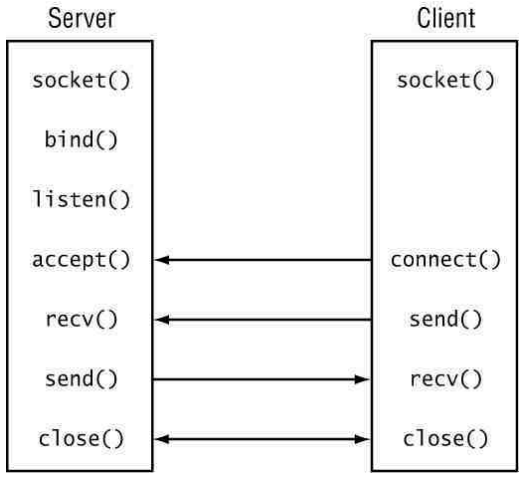
Socket: elemento de software que interactúa con la red



- Socket es una puerta de entrada/salida al sistema
- Socket utiliza servicios de transporte (capa inferior)
- Socket **es una interfaz** para entre el programador y la red
- Programador puede controlar parámetros del socket
  - Tipo de servicio de transporte (TCP / UDP)
  - Direccionamiento del host remoto (IP + puerto)

Programación Sockets

Una vez que ambos extremos, cliente y servidor, poseen un socket, es posible mandar mensajes.



Socket requiere definición de:

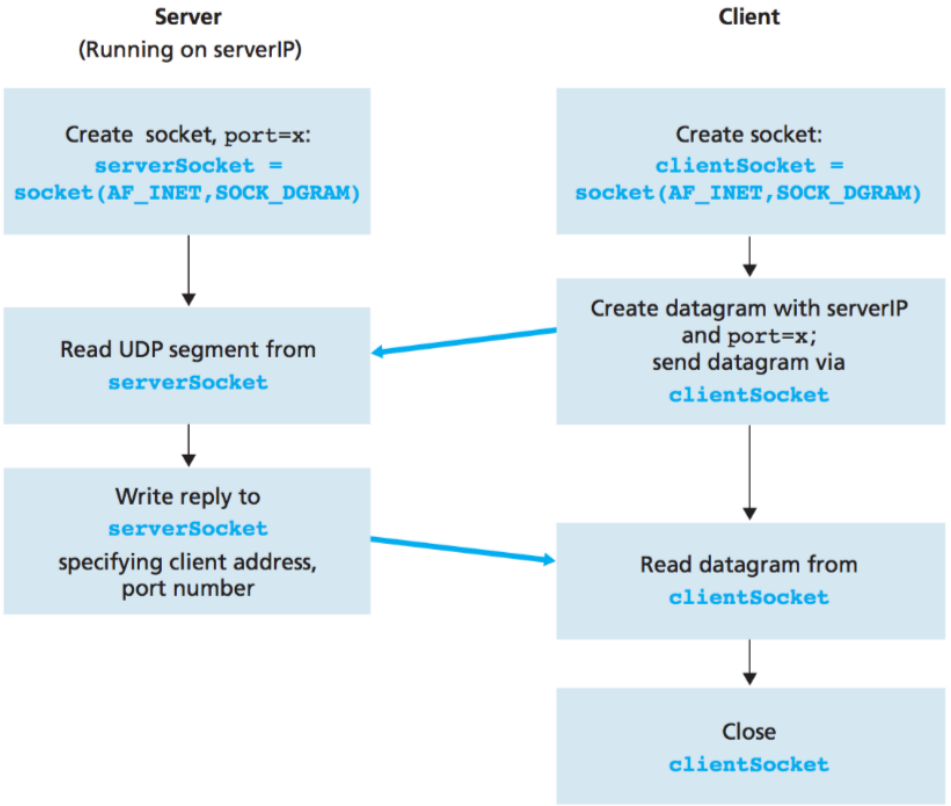
- Tipo de servicio de transporte
  - UDP: comunicación rápida, no confiable
  - TCP: comunicación confiable

- Cliente lee una línea desde la entrada estándar (stdin)
- Cliente envía línea al servidor
- Servidor recibe línea
- Servidor convierte caracteres a mayúsculas (uppercase)
- Servidor envía línea modificada al cliente2
- Cliente2 recibe línea
- Cliente2 escribe la línea en la salida estándar (stdout)

Creación de socket con AF_INET, SOCK_DGRAM

- **int socket(int domain, int type, int protocol);**
  - **domain:** Familia de protocolos. Ej: AF_LOCAL, AF_INET, AF_PACKET, etc.
  - **Tipo de socket** Ej: SOCK_DGRAM, SOCK_STREAM
  - **Protocolo:** Protocolo. Default: 0. Se suele determinar según domain y type

Creación de socket con AF_INET, SOCK_DGRAM



Envío de mensaje UDP

- ssize_t **sendto**(int sockfd, const void *buf, size_t len, int flags, const struct sockaddr *dest_addr, socklen_t addrlen);

Retorno: cantidad de bytes enviados

sockfd. Socket obtenido de socket

buf. Buffer con datos a enviar

len. Longitud (byte) de los datos a enviar

flags. Flags.

dest_addr. struct con datos de destino.

addrlen. Tamaño de dest_addr

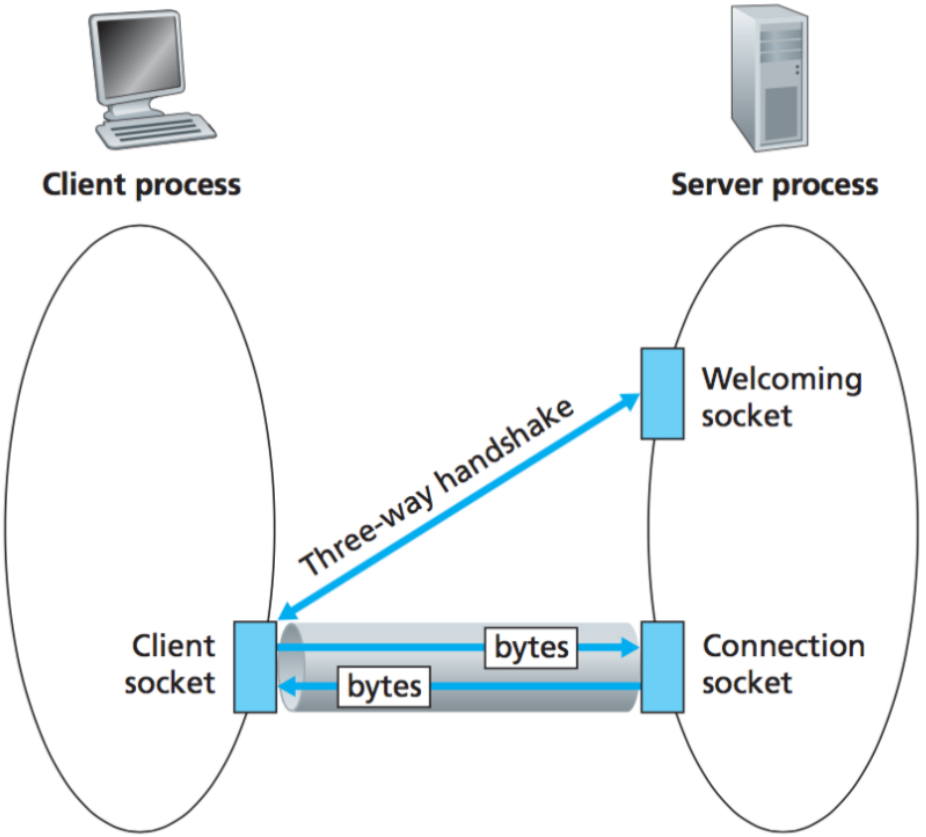
Recepción de mensaje UDP

- ssize_t **recvfrom**(int sockfd, void *buf, size_t len, int flags, struct sockaddr *src_addr, socklen_t *addrlen);

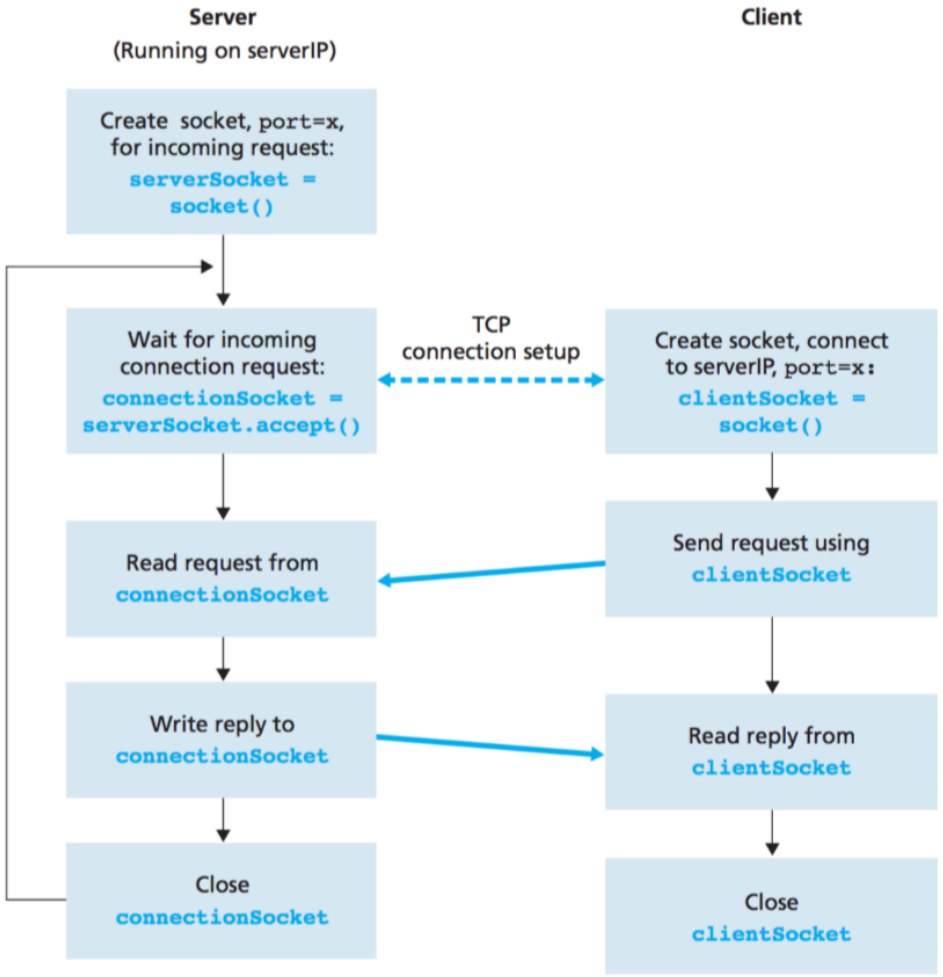
Establecimiento de conexión: TCP Handshake

- Requiere establecer conexión antes de enviar
- Cliente inicia solicitud de conexión en un puerto conocido del servidor
- Servidor debe tener un socket en ese puerto
- Servidor debe aceptar la conexión
- Al aceptar la conexión, servidor asigna nuevo socket para la conexión

Establecimiento de conexión: TCP Handshake



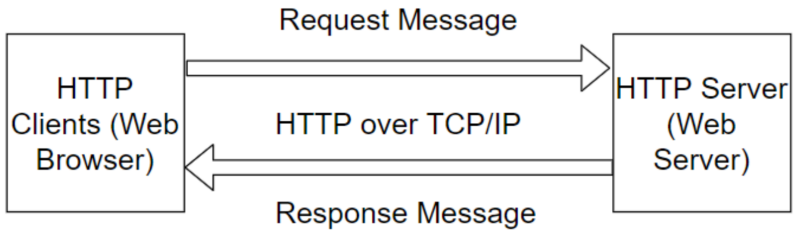
Creación de socket con AF_INET, SOCK_STREAM



HTTP

Hypertext Transfer Protocol

Hypertext Transfer Protocol: es un protocolo de la capa de aplicación para la transmisión de documentos como HTML. Es quizás el protocolo de aplicación más popular utilizado en Internet (o la Web).



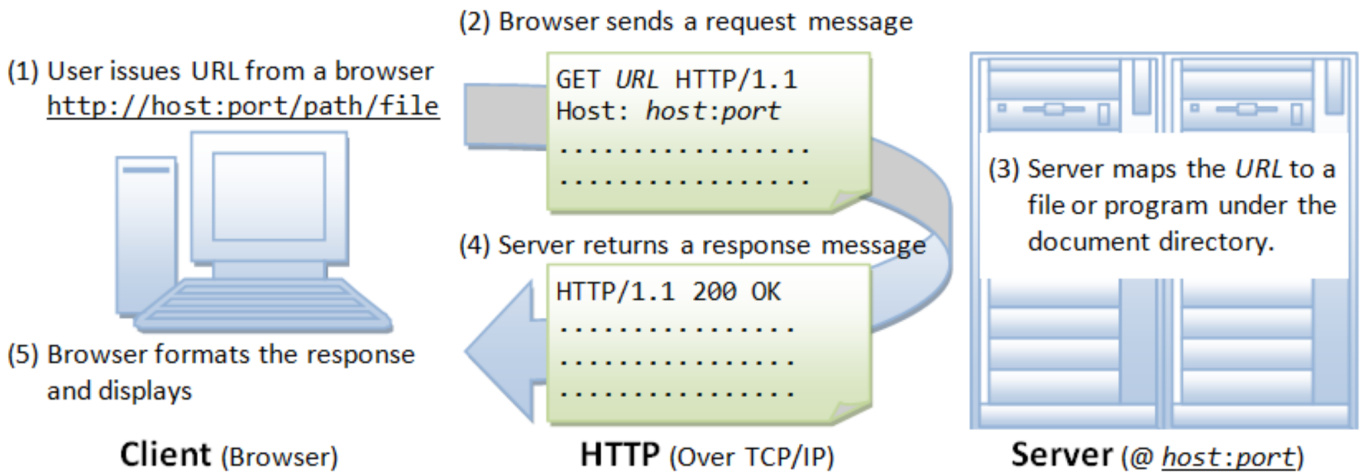
- HTTP es un protocolo **cliente-servidor asimétrico**. Un cliente HTTP envía un mensaje de solicitud a un servidor HTTP. El servidor, a su vez, devuelve un mensaje de respuesta.
- HTTP es **un protocolo de extracción**, el cliente extrae información del servidor (en lugar de que el servidor envíe información al cliente).
- HTTP es un protocolo **sin estado**.

HTTP

Hypertext Transfer Protocol

Request

Siempre que pidas una URL desde el navegador para obtener un recurso web mediante HTTP. El navegador convierte la URL en un mensaje de solicitud y lo envía al servidor HTTP.



- A URL (Uniform Resource Locator) is used to uniquely identify a resource over the web.
- URL syntax: protocol://hostname:port/path-and-file-name

Códigos

- 1xx. Mensaje de información.
- 2xx. Mensaje de éxito
  - 200 OK
  - 206 Partial Content
- 3xx. Mensaje de redirección
  - 301 Moved Permanently
  - 307 Temporary Redirect
- 4xx. Mensaje de error del cliente
  - 400 Bad Request
  - 403 Forbidden
  - 404 Not Found
  - 405 Method Not Allowed
- 5xx. Mensaje de error del servidor
  - 500 Internal Server Error
  - 501 Not Implemented
  - 502 Bad Gateway
  - 503 Server Not Available

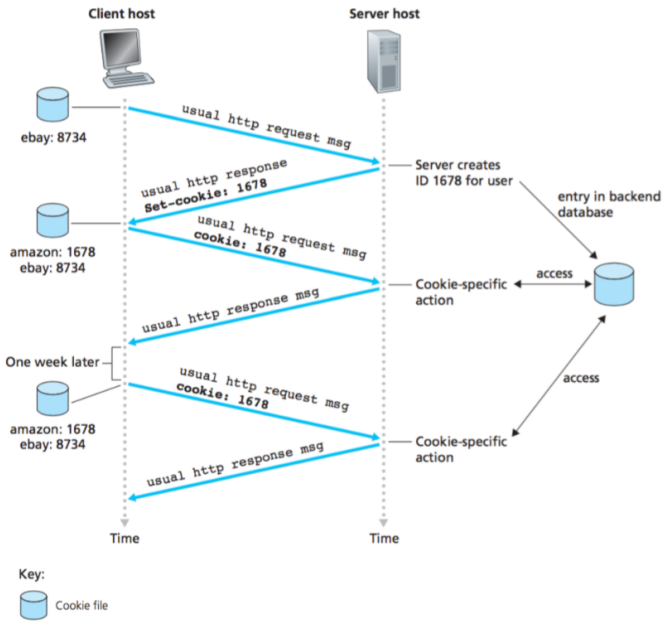
HTTP

Sin Estado

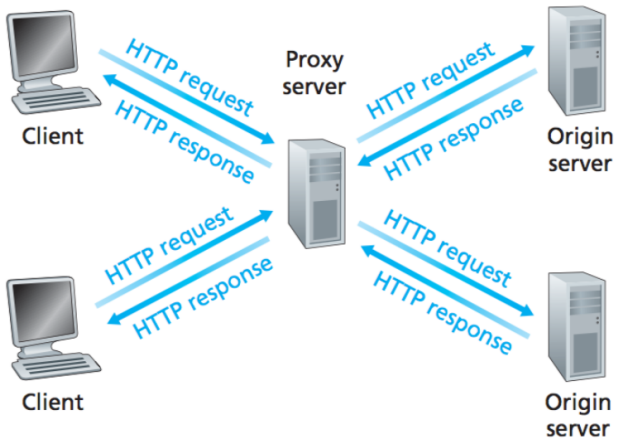
Stateless... Si se guardan cookies.

- Cookies se almacenan en el cliente
- Cookies se envían en conexiones posteriores.
- Servidor utiliza cookies para recordar estado del cliente

.



Web Caching Contenido que no cambia (estático) puede ser almacenado en servidores intermedio: web proxy

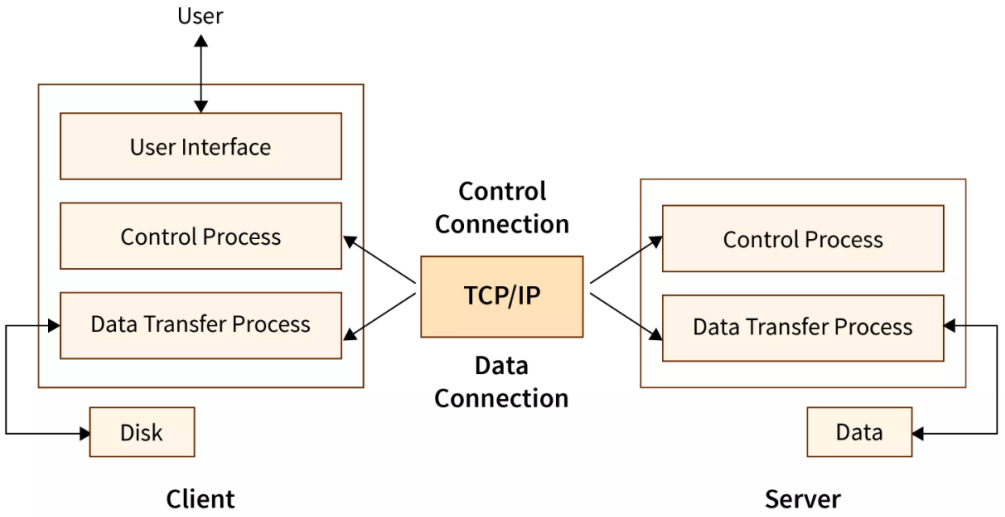


CDN: Content Delivery Network. Usan esta idea a nivel de redes de rápido acceso, cercanas al cliente.

FTP

File Transport Protocol

Transferencia de archivos entre sistema local y remoto
Transfiere archivos de texto y binarios a través de Internet. Pero se utiliza principalmente para transferir los archivos de la página web de su creador a la computadora que actúa como servidor para otras computadoras en Internet.



FTP

File Transport Protocol

- FTP usa TCP en la capa de transporte.
- FTP crea dos conexiones entre las computadoras, una conexión para los comandos y respuestas llamada conexión de control y una segunda conexión para transferencias de datos llamada conexión de datos.

SMTP, POP, IMAP

Correo electrónico

Hay 3 protocolos de correo electrónico: **SMTP, POP3 e IMAP**. Cada uno de ellos funciona en números de puerto específicos y funciona de manera diferente.

Diferencias

- **SMTP**: es un servidor de salida, encargado de enviar los correos, distribuirlos, y entregarlos a su destino.
- **IMAP Y POP3**: son servidores de entrada, se encargan de almacenar y organizar los correos, una vez recibidos.

SMTP, POP, IMAP

Correo electrónico

- **IMAP:** tus emails serán visualizados desde tu gestor de correo, pero físicamente estarán almacenados en el servidor, ocupan espacio en tu hosting. (Ej: Ustedes pueden ver sus correos desde cualqueira de sus dispositivos)
- **POP3:** los correos serán descargados directamente desde el servidor a tu equipo, por lo que no ocuparán espacio en tu hosting. (Solo un dispositivo)

SMTP, POP, IMAP

Correo electrónico

Protocolo de envío: SMTP (RFC5321)

- Simple Mail Transfer Protocol
- Puerto 25

Protocolo de lectura: POP3 (RFC1939)

- Post Office Protocol
- Puerto 110
- No mantiene estado

Protocolo de lectura: IMAP (RFC3501)

- Internet Mail Access Protocol
- Permite creación de directorios en servidor
- Mantiene estado del lector

DNS

Domain Name System

Antes de iniciar la conexión TCP para el protocolo HTTP a URL: http://iic2333.ing.puc.cl/slides/5-disk.html

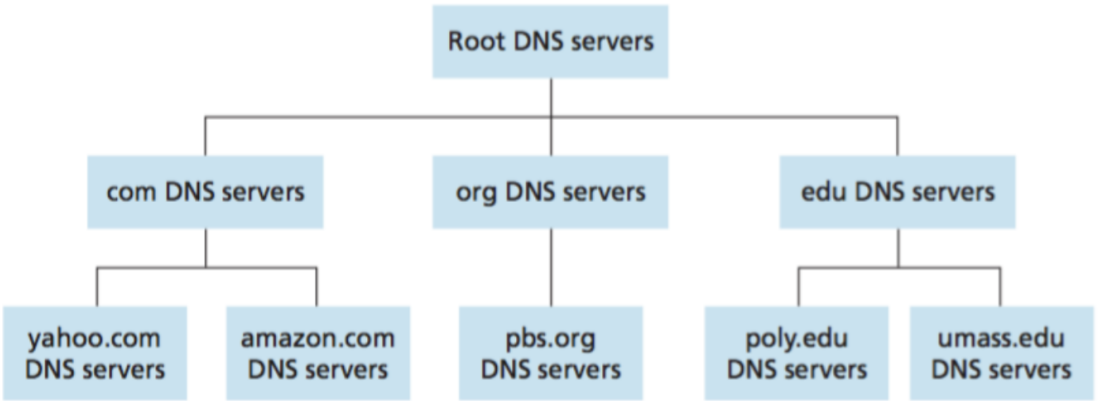
1. Cliente ejecuta proceso cliente DNS
2. Browser extrae iic2333.ing.puc.cl y lo pasa al cliente DNS
3. Cliente DNS envía query por iic2333.ing.puc.cl a servidor DNS
4. Cliente DNS recibe respuesta de servidor DNS, que incluye IP de iic2333.ing.puc.cl
5. Cliente DNS pasa respuesta a browser
6. Cliente HTTP inicia conexión TCP a la IP de iic2333.ing.puc.cl

DNS utiliza una estructura jerárquica de dominos y servidores

DNS

Domain Name System

Base de datos distribuida y jerárquica

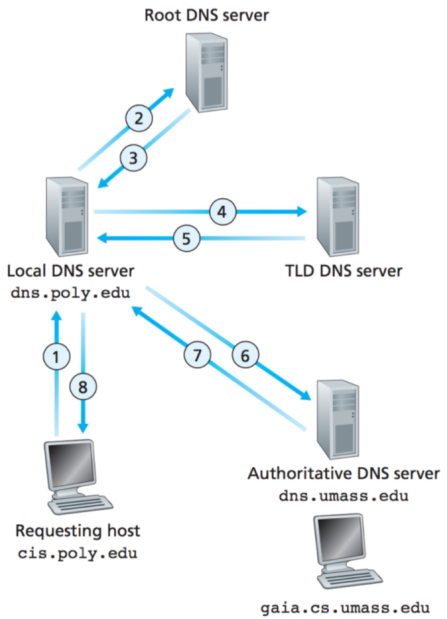


Servidores almacenan mappings

- Si no conoce un mapping, lo consulta a un servidor superior en la jerarquía
- Eventualmente puede llegar a servidor raíz
- 13 servidores raíz en el mundo: A,B,C,D,E,F,G,H,I,J,K,L,M
  - Con múltiples instancias de cada uno (cuatro de ellos replicados en Chile)
  - http://www.root-servers.org
- Servidores que administran el mapping son servidores autoritativos
  - Servidores autoritativos administran el mapping (al nivel de institución)
  - Servidores no-autoritativos mantienen temporalmente el mapping

Tipo de consulta: Iterativa o recursiva

- **Consultas iterativas**: retornan la respuesta. Ejemplo el servidor (1,2) de la imagen envían respuesta (completa o parcial) al origen
- **Consulta Recursiva**: no tienen la respuesta entonces reenvían la consulta. Ejemplo (4,6) reenvían la consulta.



DNS

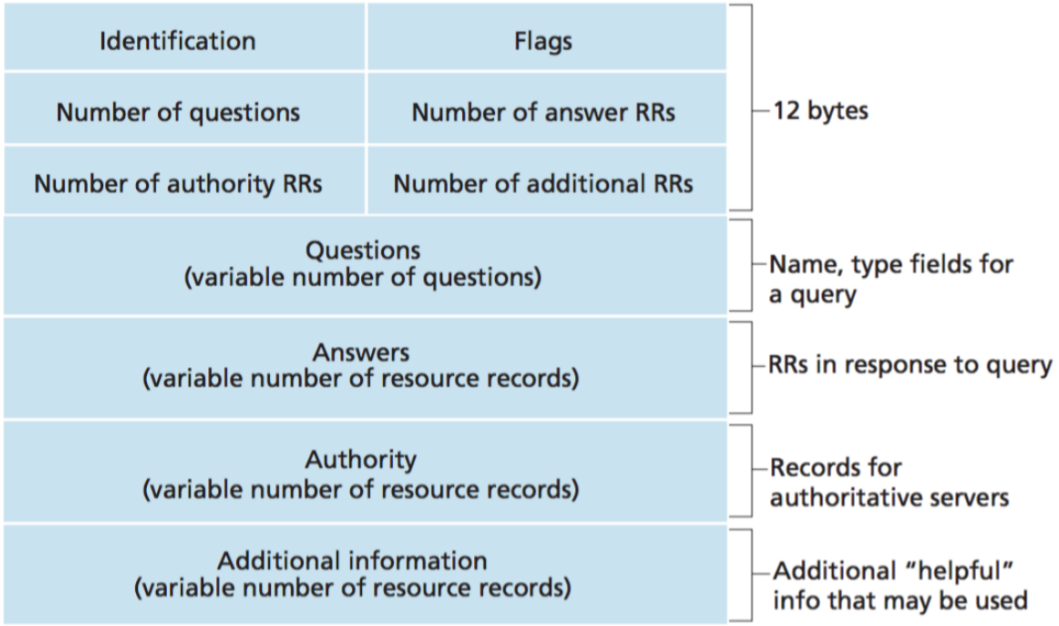
En la vida real

Los servidores raíz no son tan consultados porque ya están "guardadas" o cacheadas las respuestas en servidores de más abajo de la jerarquía

- Caching de respuestas permite reducir el tráfico
- Respuestas cached posee un timeout

DNS

Argumentos DNS



(nombre, valor, tipo, ttl)

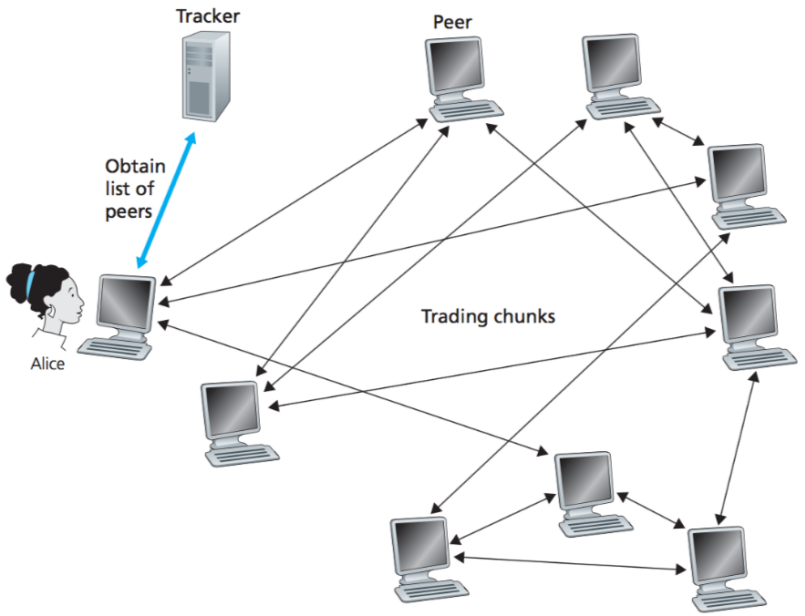
- Tipo A: nombre es nombre del host; valor es dirección IP
  - (relay1.bar.foo.com, 145.37.93.126, A)
- Tipo NS: nombre es dominio; valor es IP de host autoritativo
  - (foo.com, dns.foo.com, NS)
- Tipo CNAME: alias, nombre es alias; valor es nombre real
  - (foo.com, relay1.bar.foo.com, CNAME)
- Tipo MX: valor es el nombre de un mail server
  - (foo.com, mail.bar.foo.com, MX)

Redes P2P

PEER-TO-PEER

Redes entre pares
No dependen de un servidor

- Mejor escalabilidad
- Deben maneja pertenencia y abandono de miembros

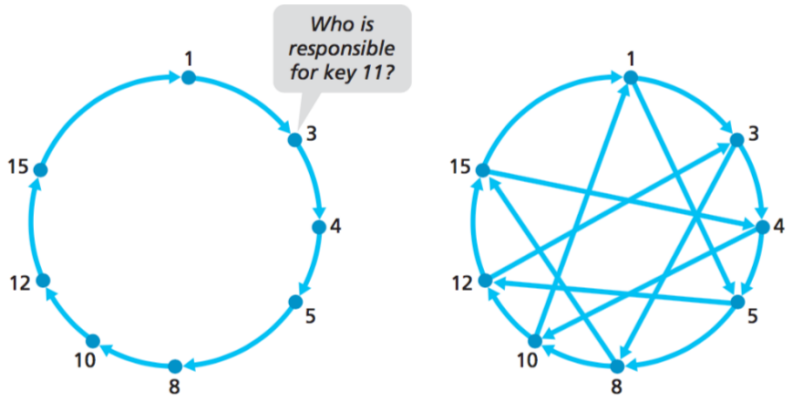


Redes P2P

DISTRIBUTED HASH TABLE

Método de direccionamiento para redes P2P estructuradas
Cada miembro posee una identificación única

- Se construye una overlay network entre miembros
- Miembros se encarga de enrutar mensajes



Material Adicional

- Video explicativo Capa Aplicación: Link Video

Próximo contenido a revisar será Capa Transporte