Bloque II: El nivel de aplicación

Tema 4: Protocolos del nivel de aplicación II

Índice

- Bloque II: El nivel de aplicación
  - Tema 4: Protocolos del nivel de aplicación II
    - DNS
      - Introducción
      - Cliente y servidor DNS
      - Espacio de nombres DNS
      - Funcionamiento DNS
      - DNS: Caché y Forwarding
      - Consultas DNS
    - P2P

- **Lecturas recomendadas:**
  - Capítulo 2, sección 2.5 y 2.6, de “Redes de Computadores: Un enfoque descendente”. James F. Kurose, Keith W. Ross. Addison Wesley.
  - Capítulo 11, secciones 11.1, 11.2, 11.3, 11.4 y 11.5, de “TCP/IP Illustrated, Volume 1: The Protocols”, W. Richard Stevens, Addison Wesley.

DNS: Introducción

- Domain Name System
- Nosotros utilizamos nombres para las máquinas (p.e. www.google.com), pero TCP/IP se comunica utilizando direcciones IP (p.e. 209.85.227.104).
- DNS es el sistema que se encarga de hacer la correspondencia entre nombres de máquinas y direcciones IP.
  - También proporciona información de los servidores de correo.
- Especificaciones: RFC 1034 (conceptos) y RFC 1035 (implementación y especificación). Varias actualizaciones posteriores.
- Modelo cliente-servidor.
- Se implementa sobre UDP (puerto 53), aunque también puede utilizar TCP.
- Antes del DNS, fichero de hosts:
  - Windows: %SystemRoot%\System32\drivers\etc\hosts
  - Linux: /etc/hosts y /etc/nsswitch.conf para orden de consulta
  - Inconvenientes: poco escalable, inconsistente con las copias locales y facilidad para nombres duplicados.
  - Válido como solución simple para redes muy pequeñas sin servidor DNS.

Cliente DNS

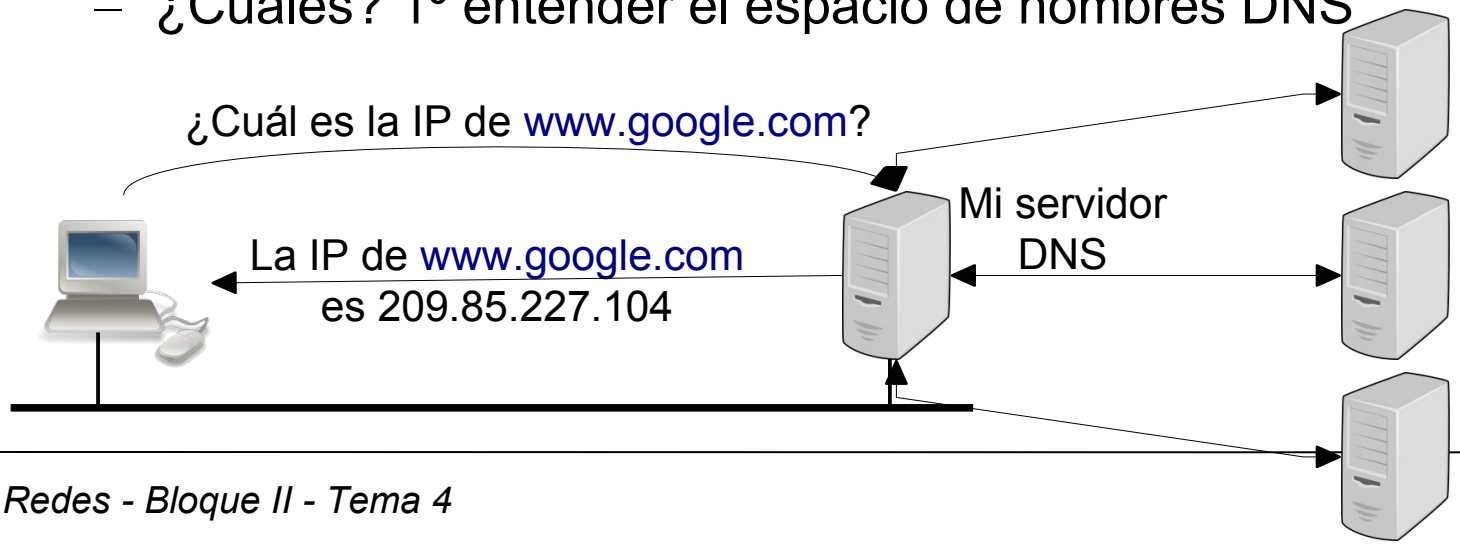
- DNS también es el protocolo que permite a los clientes y servidores comunicarse.
- **Cliente DNS**: cada máquina tiene un cliente DNS (resolver)
  - Cada vez que cualquier aplicación necesita averiguar una dirección IP, le pasa la pregunta al cliente DNS (por ejemplo, InetAddress.getByName()).
  - El cliente DNS envía la consulta a su servidor DNS, cuando obtiene la respuesta, se la pasa a la aplicación.

¿Cuál es mi servidor DNS? Ver las propiedades IPv4 (avanzadas) de la conexión en Windows o el fichero /etc/resolv.conf en máquina Linux.

Servidor DNS

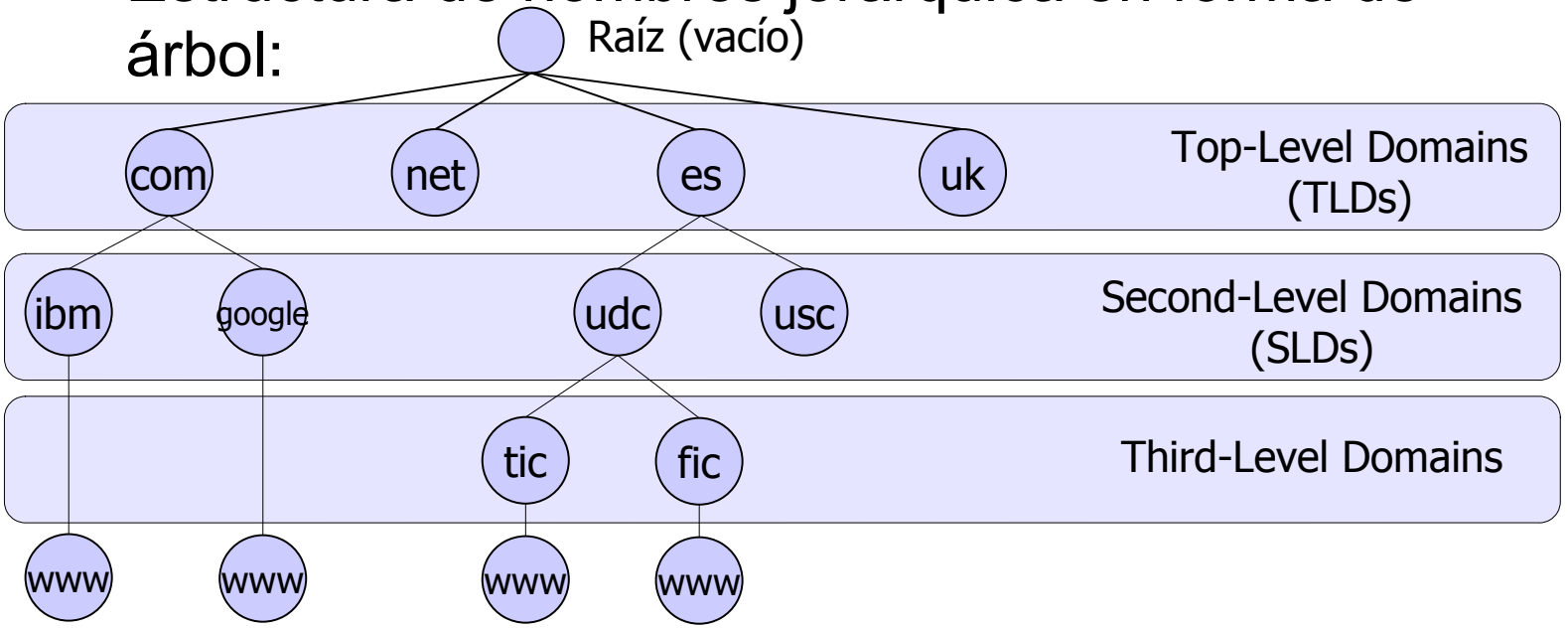
- **Servidor DNS**: cada red (p.e. wifi de la UDC, ISP, …) tiene un servidor DNS.
  - Recibe consultas DNS de clientes, averigua la dirección IP y la envía a los clientes.

- ¿Cómo averigua mi servidor DNS una dirección IP?
  - El DNS es una **base de datos distribuida** → No hay un servidor que conozca todos los nombres y sus IPs.
  - Hay múltiples servidores DNS organizados jerárquicamente → Preguntando a otros servidores.
  - ¿Cuáles? 1º entender el espacio de nombres DNS



Espacio de nombres DNS

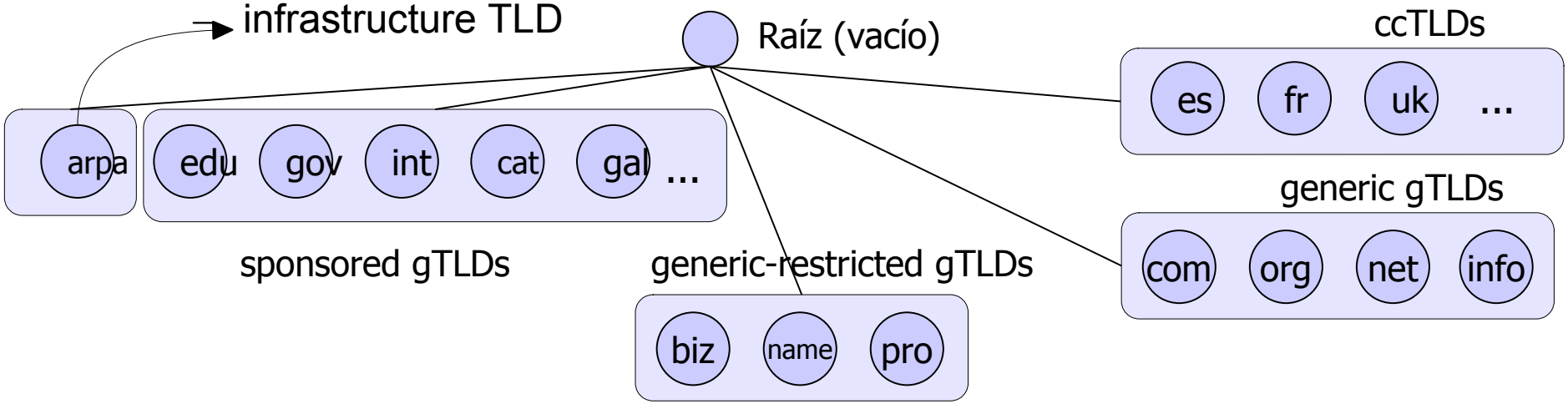
- Estructura de nombres jerárquica en forma de árbol:



- Nombre de dominio: www.fic.udc.es
  - No se distinguen mayúsculas y minúsculas.
- FQDNs (fully qualified domain names): nombre de dominio completo (formalmente acabado en “.”).
  - Si está incompleto → Se “rellena” con nuestro dominio: /etc/resolv.conf

Espacio de nombres DNS: TLDs

- Top-Level Domains (TLDs):
  - ccTLDs: country-codes TLDs
  - gTLDs: generic TLDs. Tres tipos: generic, generic-restricted y sponsored.
  - IDN ccTLDs: internationalized contry-code TLDs
  - infrastructure TLD

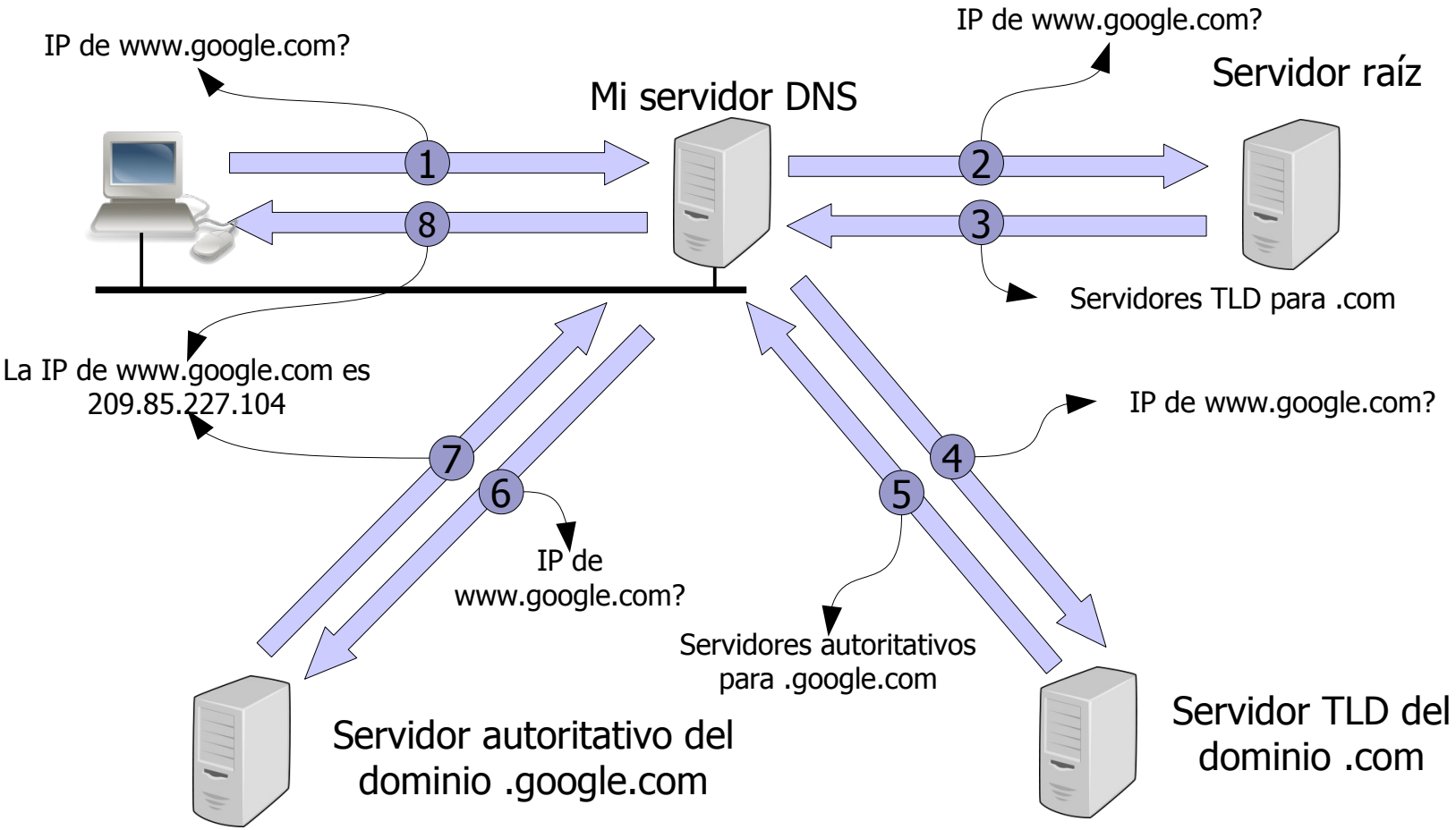


- Listado completo de los TLDs: http://www.iana.org/domains/root/db/
- Los nuevos gTLDs: http://newgtlds.icann.org/en/program-status/delegated-strings

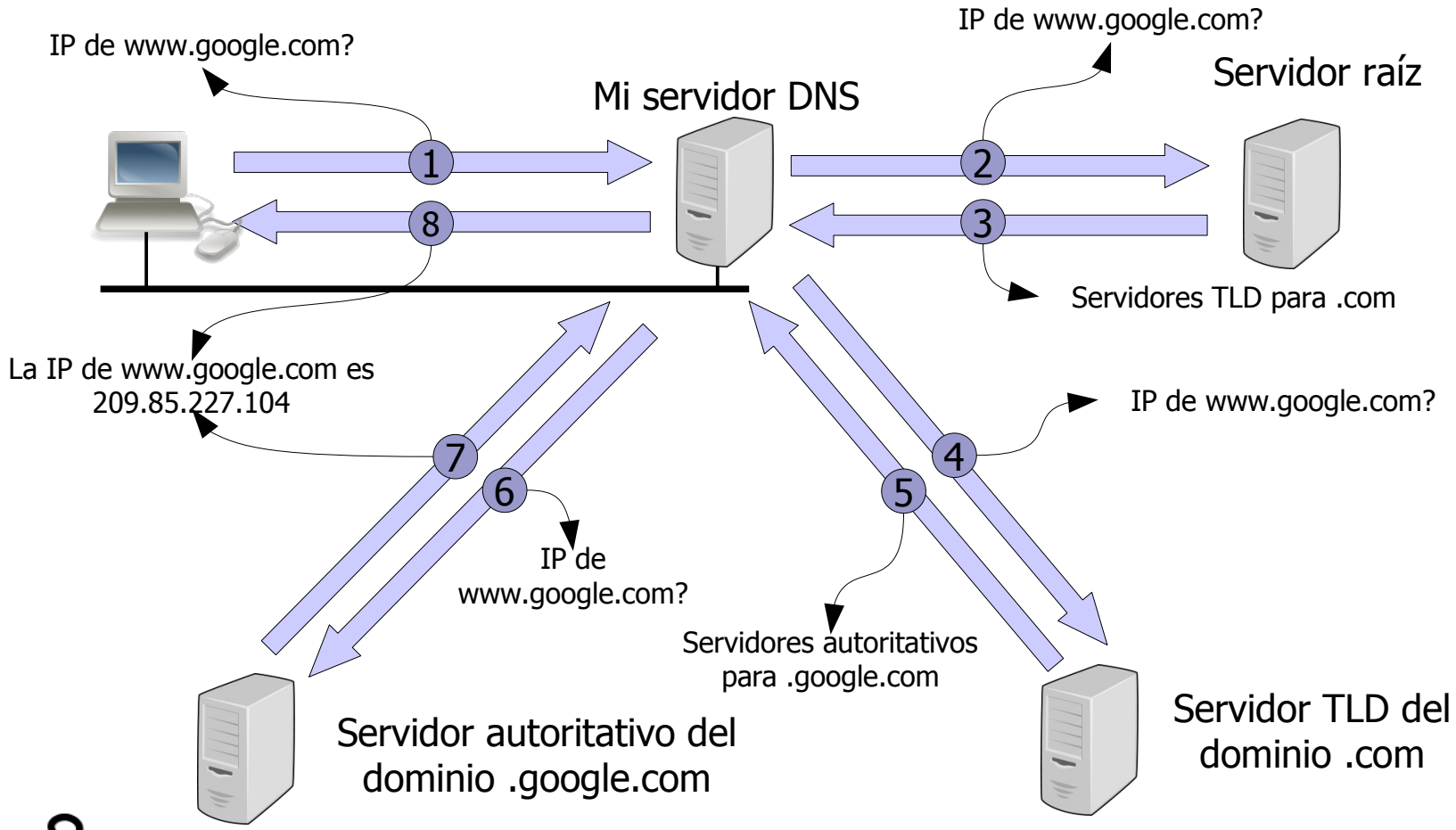
Servidores de nombres

- Hay servidores DNS en cada nivel de la jerarquía de los nombres de dominio:
  - **Distribuir** la carga entre los servidores de nombres.
  - **Delegación** de la administración de los servidores de nombres
- Servidores **raíz**: http://www.root-servers.org/
  - Existen 13 servidores raíz (A-M), replicados por seguridad y fiabilidad → Son críticos.
  - Conocen a todos los TLDs y delegan en ellos.
- Servidores TLD:
  - Cada dominio de 1er nivel tiene su servidor TLD asociado.
  - Delegan en servidores de 2º nivel la gestión de los sub-dominios.
- Servidores DNS inferiores:
  - Conocen a todos los equipos de su dominio.
  - Conocen a los servidores DNS raíz.
  - Ante una consulta, si no conoce una IP, le pregunta a un servidor raíz.

Funcionamiento DNS



Funcionamiento DNS



¿Qué pasa si falla mi servidor DNS?

¿Puedo usar otro?

Funcionamiento DNS

- Consultas recursivas:
  - El servidor DNS hará todo el trabajo necesario para devolver la respuesta completa a la petición.
  - Puede implicar múltiples transacciones del servidor con otros servidores DNS.
  - No es obligatorio que los servidores DNS soporten este tipo de consultas.
  - Mi servidor, normalmente, será recursivo.
- Consultas iterativas (no recursivas):
  - Si el servidor DNS tiene la respuesta, entonces la devuelve.
  - Si el servidor DNS no tiene la respuesta, devolverá información útil, pero no hará peticiones adicionales a otros servidores DNS.
  - Los servidores raíz y TLD son no recursivos.

Caché DNS

- Para reducir los mensajes DNS → Cachés.
- Los servidores DNS disponen de una caché:
  - Cada par dirección IP – nombre que se resuelve se almacena en la caché.
  - Tº de vida (TTL) de varios días.
  - Negative caching: almacenar también las peticiones incorrectas.
- **Respuesta autoritativa**: responde directamente el servidor DNS que "conoce" la información → **Servidor autoritativo**

- ¿Caché también en el cliente o sólo en el servidor?

  Ahora la tendencia es poner también una caché en el cliente:
  - Windows ya la incorpora.
  - En Linux: dnsmasq → servidor DNS local para caching (y forwarding)
    - Deshabilitar en /etc/NetworkManager/NetworkManager.conf

Servidor DNS de Forwarding

- Servidores DNS de Forwarding:
  - No es responsable de ninguna zona → No almacena información en disco.
  - Sólo reenvía las consultas a otros servidor DNS → Consultas recursivas.
  - Almacena las respuestas en caché → Respuesta rápida para consultas frecuentes.

- Un router inalámbrico, lo normal es que incorpore un servidor DNS de forwarding:
  - Reenvía las consultas al servidor DNS de mi ISP.
  - Las consultas en caché se resuelven en mi LAN → Evito accesos a la red del ISP.

Consultas DNS

- Consulta A (Estándar): nombre → IP
  - *dig www.google.com*
  - Online: http://www.kloth.net/services/dig.php
- Consulta CNAME: alias de un nombre
  - dig -t CNAME www.lavoz.com
- Consulta PTR (Consulta inversa – Pointer)
  - Un cliente DNS necesita conocer el nombre de dominio asociado a la dirección IP 88.221.32.170 → Consulta inversa 170.32.221.88.in-addr.arpa.
  - Se basa en el TLD arpa.
  - Es necesario invertir la dirección IP, ya que los nombres de dominio son más genéricos por la derecha (al contrario que las direcciones IP).
  - *dig -x 88.221.32.170*

Consultas DNS

Cuando se envía un e-mail (p.e. a john.doe@gmail.com), ¿cómo sabe mi servidor de correo cuál es el servidor de correo SMTP del dominio gmail.com?

- Consulta MX (Mail Exchanger):
  - El servidor de correo origen, envía una consulta MX a su servidor DNS preguntando por el dominio de destino (p.e. gmail.com).
  - La respuesta incluye un listado con los servidores de correo disponibles.
    - Menor preferencia → Primero
  - *dig -t MX gmail.com*

DNS: Comandos

- Comandos nslookup y dig:
  - Envía peticiones DNS al servidor DNS por defecto
  - Por defecto, envían peticiones estándar.
  - Permiten especificar otros tipos de peticiones.
- Comando bind:
  - Berkeley Internet Name Domain
  - Servidor DNS más utilizado en Internet.
  - Incluido en Linux

- Servidor DNS público de Google:
  - 8.8.8.8 y 8.8.4.4
  - http://code.google.com/speed/public-dns/

P2P

- Los protocolos anteriores se basaban en el modelo cliente-servidor: el servidor proporciona un servicio y el cliente consume ese servicio.
- El modelo P2P (**Peer To Peer**) está compuesto por pares (peers) que realizan ambas funciones: consumir y proporcionar un servicio.
- Se basa en equipos de usuarios:
  - No son propiedad de un proveedor de servicio.
  - Conectados intermitentemente.
  - Proporcionan acceso a una parte de sus recursos (disco, cpu, ancho de banda)
- Ventajas:
  - Compartición de recursos (cuantos más, mejor).
  - Gran tolerancia a fallos.
- Inconvenientes:
  - Seguridad: acceso a los recursos de un equipo → Aumento de las medidas de seguridad en los últimos años.
  - Gran uso de ancho de banda → A veces restringidos por los ISPs.

P2P: Ejemplos y tipos

- Distribución de archivos: BitTorrent, Napster,
- Voz sobre IP: Skype
- Instant messaging (IM)
- Préstamo masivo de CPU
  - Proyecto SETI@Home
  - Plataforma BOINC (Berkeley Open Infrastructure for Network Computing)
- Blockchain, bitcoin: basados en redes P2P.

- P2P estructurado: los nodos se organizan en una topología específica.
- P2P sin estructura: los nodos se conectan entre sí de forma aleatoria.
  - P2P puro: todos los nodos son iguales.
  - P2P centralizado: dispone de un nodo central que funciona como un servidor de directorio. El resto de los nodos no están organizados.
  - P2P híbrido: existe nodos especiales (supernodos) que realizan algunas tareas del servidor de directorio.