

Redes de Datos 1

Capa de Aplicación

Capa de Aplicación

- Es la capa donde se encuentran las aplicaciones que interactúan con el usuario
 - **Son la razón de ser de las redes de datos**
- En Internet, son estas aplicaciones el principal motor de crecimiento

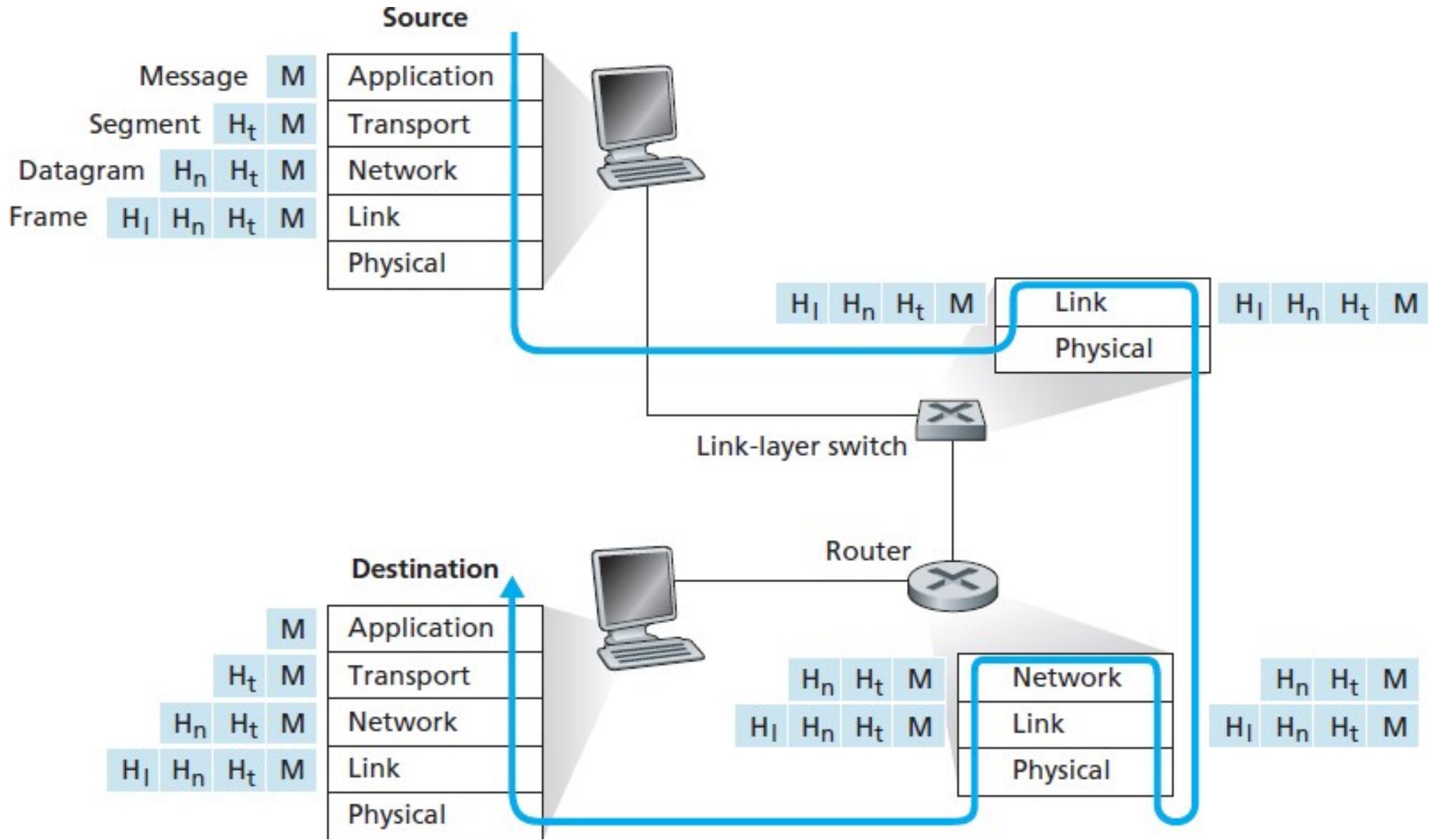
- 197X y 198X – correos de texto y acceso remoto
- 1990 – World Wide Web
- 1998 – Google search engine
- 2003 – skype
- 2005 – Youtube
- 2006 – Facebook
- 2007 – Netflix VoD

- En Internet, se intenta que la “*inteligencia*” se encuentre en las aplicaciones de los extremos.

GLOBAL APP TRAFFIC SHARE		
TOTAL TRAFFIC		
	Category	Total Volume
1	YouTube	14.61%
2	Netflix	9.39%
3	Facebook	7.39%
4	Facebook video	4.20%
5	Tik Tok	4.00%
6	QUIC	3.98%
7	HTTP	3.58%
8	HTTP Media Stream	3.57%
9	BitTorrent	2.91%
10	Google	2.79%

Source: Sandvine Global Internet Phenomena Report, January 2022

Capa de Aplicación



■ Proveer servicios al usuario final:

- Correo electrónico o comunicación instantánea
- Navegación web
- Transferencia de archivos
- Terminal virtual
- audio/video
- Peer to peer
- etc., etc., ...

■ Se necesitan protocolos y servicios de apoyo a las aplicaciones:

- Servicio de nombres (DNS)
- Seguridad (Criptografía/Autenticación)
- Administración y Gestión de redes

Aplicaciones

- Las aplicaciones y servicios disponibles dependen en parte del sistema operativo
- **Desde la perspectiva del desarrollador**, la infraestructura de red “está fija”, solo puede utilizar un conjunto específico de servicios.
- La capa de aplicación utiliza los servicios de las capas inferiores
 - **TCP**: Orientado a conexión, confiable (~**garantía de entrega**)
 - **UDP**: Datagrama, no confiable
- Algunas aplicaciones ofrecen servicios auxiliares a otras aplicaciones
 - Usualmente bibliotecas del sistema operativo
- Aplicaciones interactúan con el usuario

■ ¿Son necesarios?

- Interacción entre aplicaciones de distintos proveedores (web server y web browser)
- Aún en caso de ser un único proveedor tendremos protocolos (Whatsapp: servidor, cliente web, cliente android y cliente ios)
- Intercambio ordenado
- Evitar deadlocks
- Etc...

■ No siempre son complejos

- Muchos protocolos utilizan mensajes de texto en inglés

- **Un protocolo de Capa de Aplicación define:**
 - El tipo de **mensajes** intercambiados
 - La **sintaxis** de los mensajes y los campos de los mensajes
 - La **semántica** de los mensajes
 - Las **reglas** de cuando una aplicación debe enviar un mensaje o responder a un mensaje.

■ Dos grandes arquitecturas:

- **Cliente servidor (CS):** servidor “**always on**”, brinda servicios consumidos por los clientes. Los clientes no dialogan entre si.

- **Ejemplo:** HTTP, correo electrónico, DNS, etc.

- **P2P:** las aplicaciones se comunican entre si en pie de igualdad

- **Ejemplo:** Compartición de archivos (Bittorrent, eMule), telefonía Internet (Skype), etc.

■ Puede haber aplicaciones híbridas:

- Aplicación de mensajería instantánea:

- **CS:** Registro y seguimiento del cliente (control) centralizado

- **P2P:** Dialogo entre pares

¿Cómo se utilizan los servicios de Capa de Transporte?

- Los servicios de capa 4 están accesibles mediante “**primitivas**” o llamadas a bibliotecas del sistema operativo
 - La más usada: biblioteca de “**sockets**”
 - Distintas variantes, misma filosofía pero varían los detalles

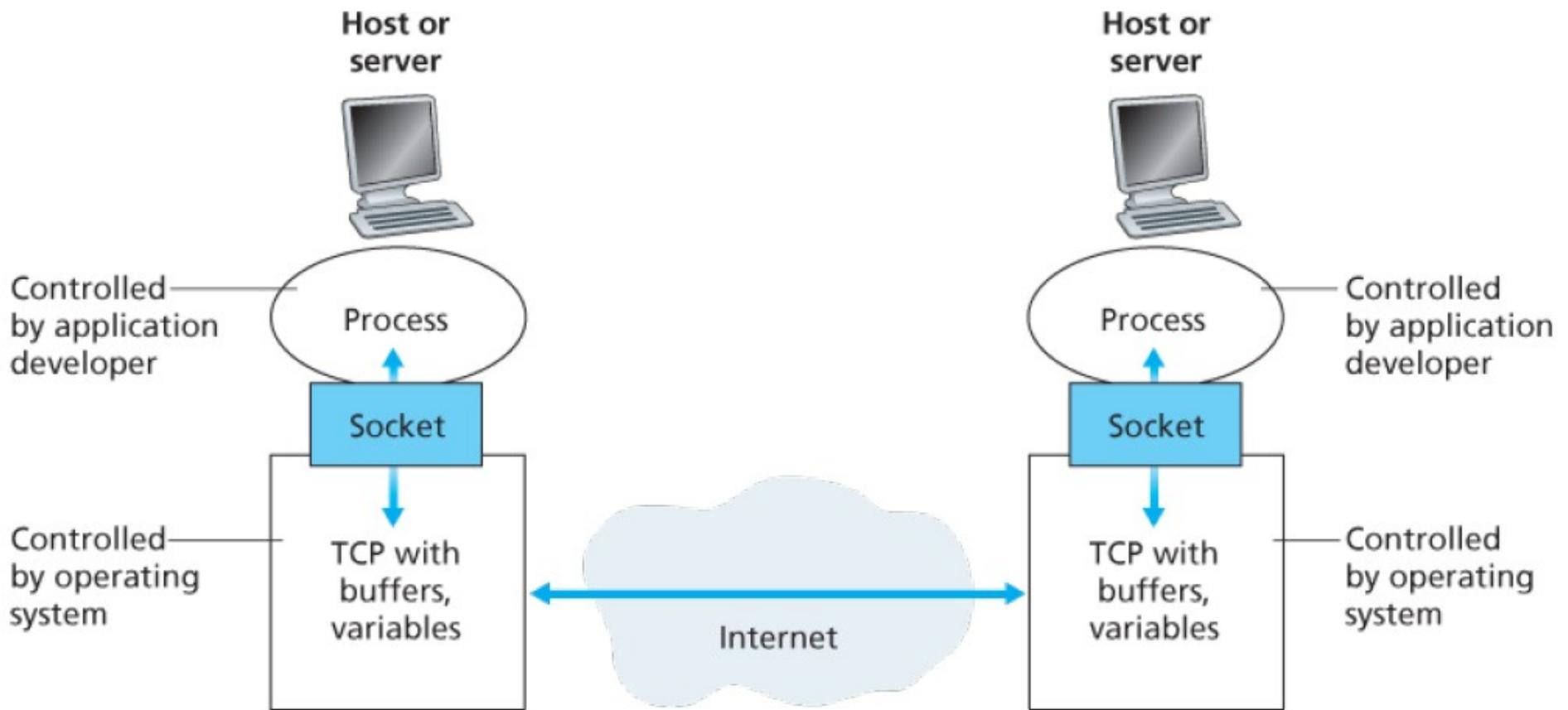


Figure 2.3 Application processes, sockets, and underlying transport protocol

■ Primitivas Socket UDP

- Crear un socket (socket)
- Elegir parámetros (bind)
- Enviar datos (sendto)
- Recibir datos (recvfrom)
- Finalizar y liberar recursos (close)

■ Primitivas Socket en TCP:

- Crear un socket (socket)
- Elegir parámetros (bind)
- **Escuchar conexiones (listen)**
- **Aceptar conexiones (accept)**
- **Iniciar una conexión (connect)**
- Enviar datos (send)
- Recibir datos (recv)
- Finalizar y liberar recursos (close)

Cliente y Servidor TCP

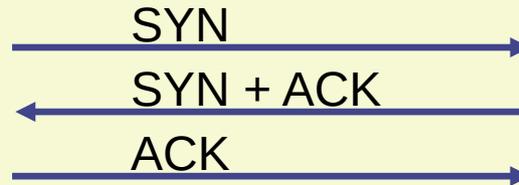
Cliente

Servidor

- socket()
- bind()

- socket()
- bind()
- listen()

- connect()



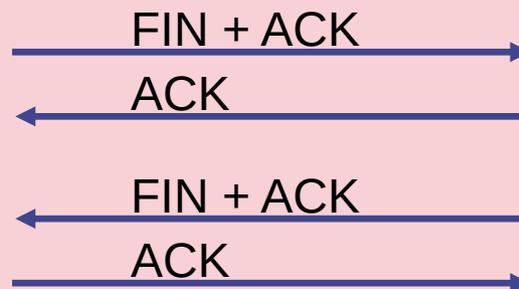
- accept()

- send() / recv()



- send() / recv()

- close()



- close()

■ Transferencia de archivos

- Transferencia de información sin errores
- Capacidad (**throughput**) “alto”
- **Interactividad** no importante

■ Navegación web/correo electrónico

- Transferencia de información **sin errores**
- **Interactividad** aceptable (segundos)
- Capacidad aceptable para una navegación ágil (“muy subjetivo” y evoluciona en el tiempo)

■ Telefonía IP

- Retardo bajo
- Variación de retardo bajo
- No es tan importante el control de errores
- Capacidad “baja” (decenas de kbps)

■ Video IP

- Variación de retardo bajo
- No es tan importante el control de errores
- Capacidad “alta” (cientos/miles de kbps)

■ DNS

- Consultas y respuesta de “un datagrama”

Ejemplos de protocolos populares de capa de aplicación

- **HTTP**: Hyper Text Transfer Protocol, navegación web
- Se basa en “pedidos” (request) por parte del **cliente** y “respuestas” (response) del **servidor**
- El comando más común es el **GET**
- Cuando visitamos **http://iie.fing.edu.uy/index.html**

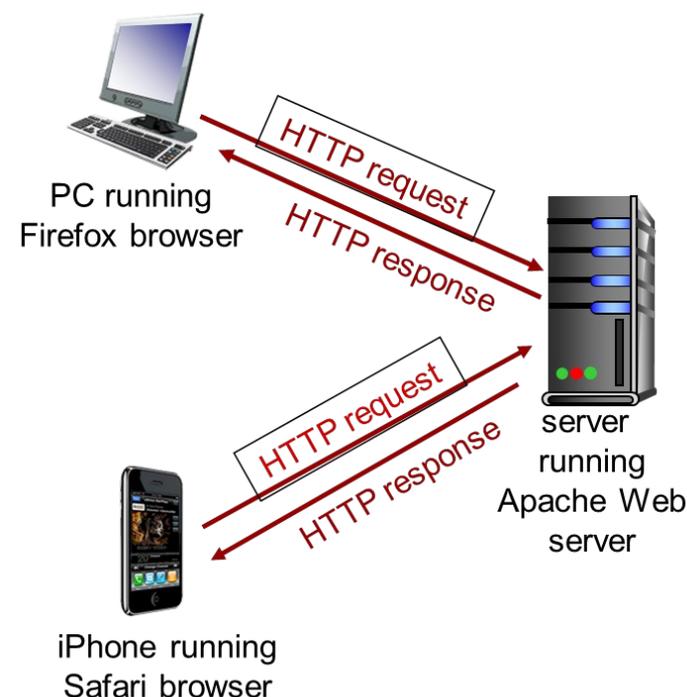
-> GET /index.html HTTP/1.0

-> Host: iie.fing.edu.uy

->

<- “Encabezados”

<- “contenido de la página”



HTTP Redirect

■ Cuando visitamos **http://adinet.com.uy/**

-> GET / HTTP/1.0

-> Host: adinet.com.uy

->

<- Location: https://correo.vera.com.uy/

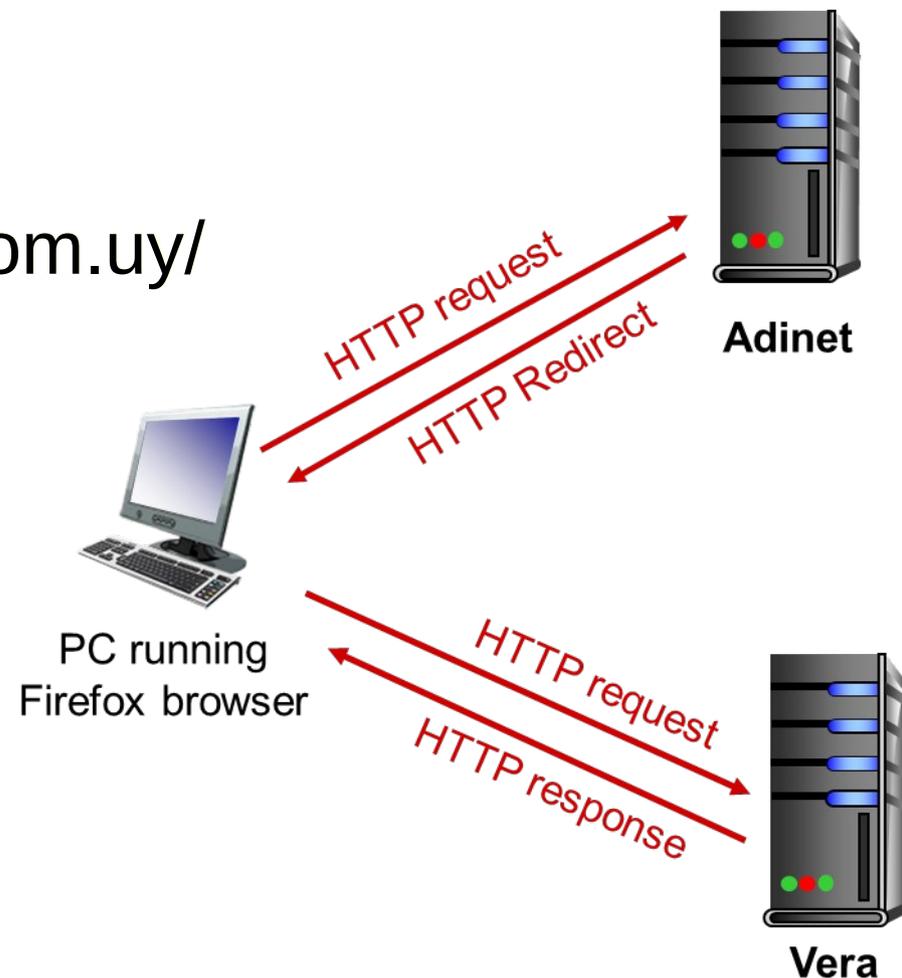
-> GET / HTTP/1.0

-> Host: correo.vera.com.uy

->

<- “Encabezados”

<- “contenido de la página”



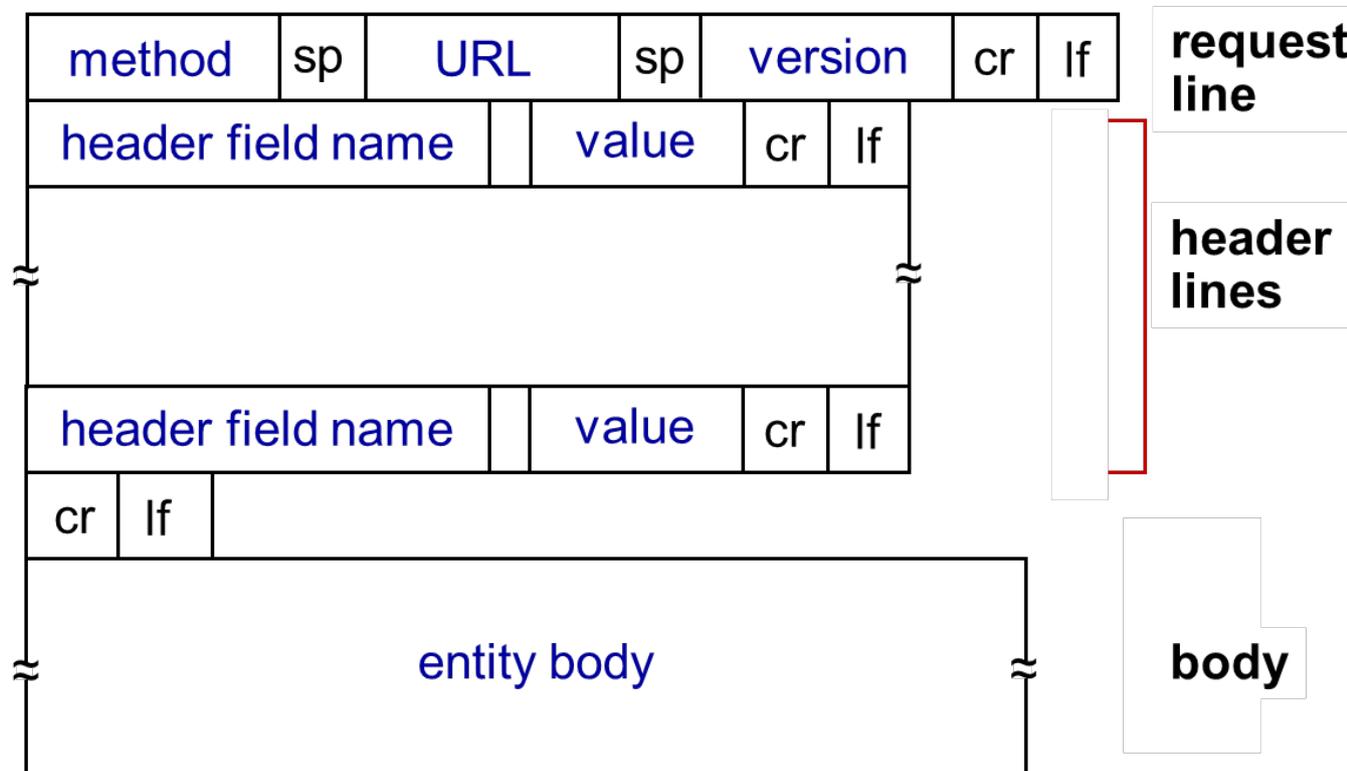
Protocolo de Capa de Aplicación HTTP

■ Otros Comandos (method):

- **HEAD:** solamente devuelve los encabezados
- **POST:** pensado para enviar información al servidor
-

■ Se pueden utilizar otros encabezados en los pedidos por ejemplo **cookies** para seguimiento

■ En los encabezados de la respuesta se indica el tipo de contenido, fecha de modificación, e información varia.



Protocolo de Capa de Aplicación HTTP

- **HTML:** Hyper Text Markup Language
- Una página web se compone de **objetos**: archivo HTML, imagen jpg, un archivo de audio, etc.
- Una página web consiste en un archivo HTML que hace referencia a los otros **objetos**.
- Cada uno de esos **objetos** son alcanzables utilizando una URL (Unified Resource Locator) para encontrarlo
https://www.fing.edu.uy/sites/default/files/cti_flex_logo.png
- HTML es un lenguaje para indicar el contenido y formato de una página.
- Es el browser o navegador que luego interpreta el archivo HTML y lo presenta en pantalla.
- **HTTPS:** HTTP sobre un canal seguro (encriptado)

Otro ejemplo: telefonía IP o VoIP

- De forma simplificada se divide en **control** y **media**.
- **Control:** usualmente H323 o SIP. Corresponde con la señalización de la llamada (establecimiento, corte, servicios adicionales)
 - Requiere un servicio libre de errores, y retardos “razonables”, pero no tiene requerimientos de tiempo real.
- **Media** o Datos: paquetes conteniendo la voz “muestreada” o digitalizada (usualmente RTP)
 - Tiene fuertes requerimientos en cuanto a retardos, y especialmente variación de retardo

RTP (RFC 3550)(Real-time Transport Protocol)

- “Entre capa 4 y 5”
- Generalmente utiliza el servicio de datagramas (UDP) de la capa 4.
- Diseñado para llevar información de audio y video
- **Identificación** de contenido
- **Secuenciamento** (detección de pérdidas)
- **Sincronización** (diferente medios) y cálculo de jitter
- Cada cliente implementa como reaccionar frente a los eventos de pérdidas de datagramas, cambio de retardo, variación de retardo, etc.
- Protocolo binario, contiene una cabecera con varios campos, y luego los datos binarios correspondientes al medio siendo reproducido
- Por ejemplo, en telefonía IP, usualmente se llevan una o varias muestras de voz muestreada a 8000 muestras por segundo (comprimidas o no)



Seguridad

- La seguridad hay que tenerla en cuenta en todas las capas del modelo.
- Como simples clientes **no tenemos control** de las tres capas inferiores.
- Como organización es posible que tengamos control de las capas inferiores, ¿qué sucede cuando el destino pertenece a otra organización?
- Velocidad de implementar cambios.
- Fuerte motivación a implementar estrategias de seguridad extremo a extremo dentro de la capa de aplicación.
- **TLS**: Transport Layer Security, es implementado como una aplicación.
- En las últimas clases del curso se tratará globalmente.

■ ¿Necesidad?

- Red de mi casa: poca o ninguna
 - Proveedor de internet o gran empresa con decenas de puntos de presencia: **esencial**
- La gestión de redes, aún en sus formas más primitivas, debe permitirme una visión de la red que permita operar y tomar decisiones.
- Detectar fallas y corregir fallas.
 - Utilización de recursos, cantidad de clientes.
 - Decisiones de inversión para crecimiento.

Capa de Aplicación y sub-capas

- Formalmente las funciones de las capas inferiores están definidas.
- Por temas de compatibilidad, “**no es posible realizar cambios mayores**” en las capas inferiores.
- **Puedo implementar lo que necesite en capa de aplicación.**
- En muchos casos, parte de estas funciones son comunes a varios escenarios, por lo que existen bibliotecas que implementan estas funciones.
- Estas bibliotecas las podemos “**visualizar/modelar**” como una **sub-capa**.



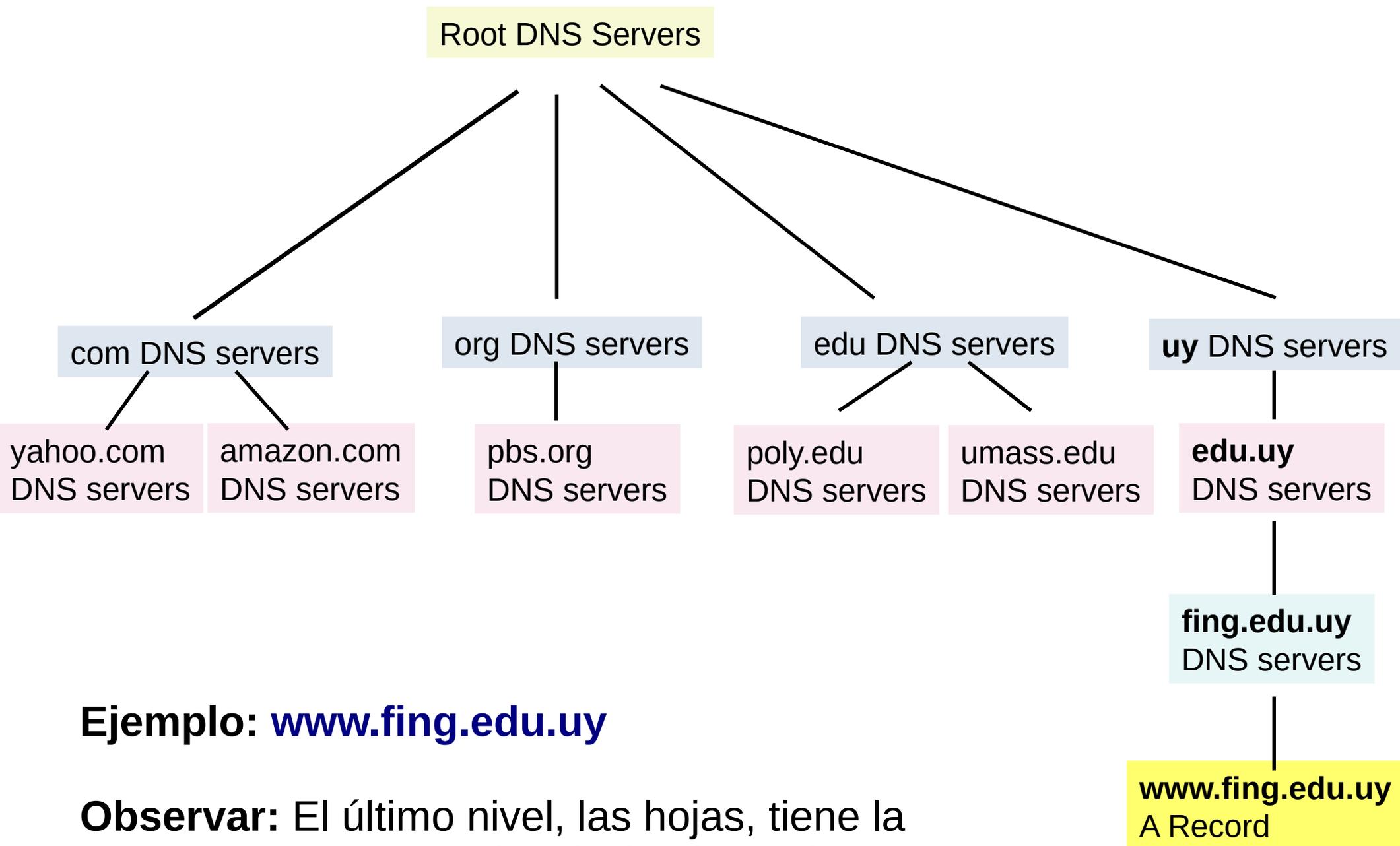
Servicio de Nombre de Dominio: DNS

- Para el usuario es difícil recordar que la información del curso está en la dirección de red (dirección IP) 164.73.32.6
- Es más fácil recordar **eva.fing.edu.uy**
- **Originalmente:** un archivo (hosts.txt) con relación entre dirección capa 3 y nombre.
 - ARPANET 1970
 - Migración de ARPANET a TCP/IP estable 1 de enero 1983
 - 500 computadoras 1983
 - => **No escala** para miles de computadoras
- El servicio DNS resuelve esa correspondencia
- **DNS:** RFC 1034 noviembre 1987

DNS - Domain Name System

- Definido en las RFC 1034 y RFC 1035
 - Múltiples agregados y actualizaciones.
- **Define:**
 - Un espacio de nombres jerárquico basado en dominios
 - Un sistema de base de datos distribuido para implementarlo (guardar el espacio de nombre)
 - Un protocolo para consultarlo
- **Es un servicio crítico de Internet**
- **Espacio de nombres de DNS**
 - Jerárquico en forma de árbol invertido
 - Algunas limitaciones:
 - Cada etiqueta menos de 64 caracteres
 - Nombre completo menos de 255 caracteres
 - **Ejemplo:** *www.fing.edu.uy*

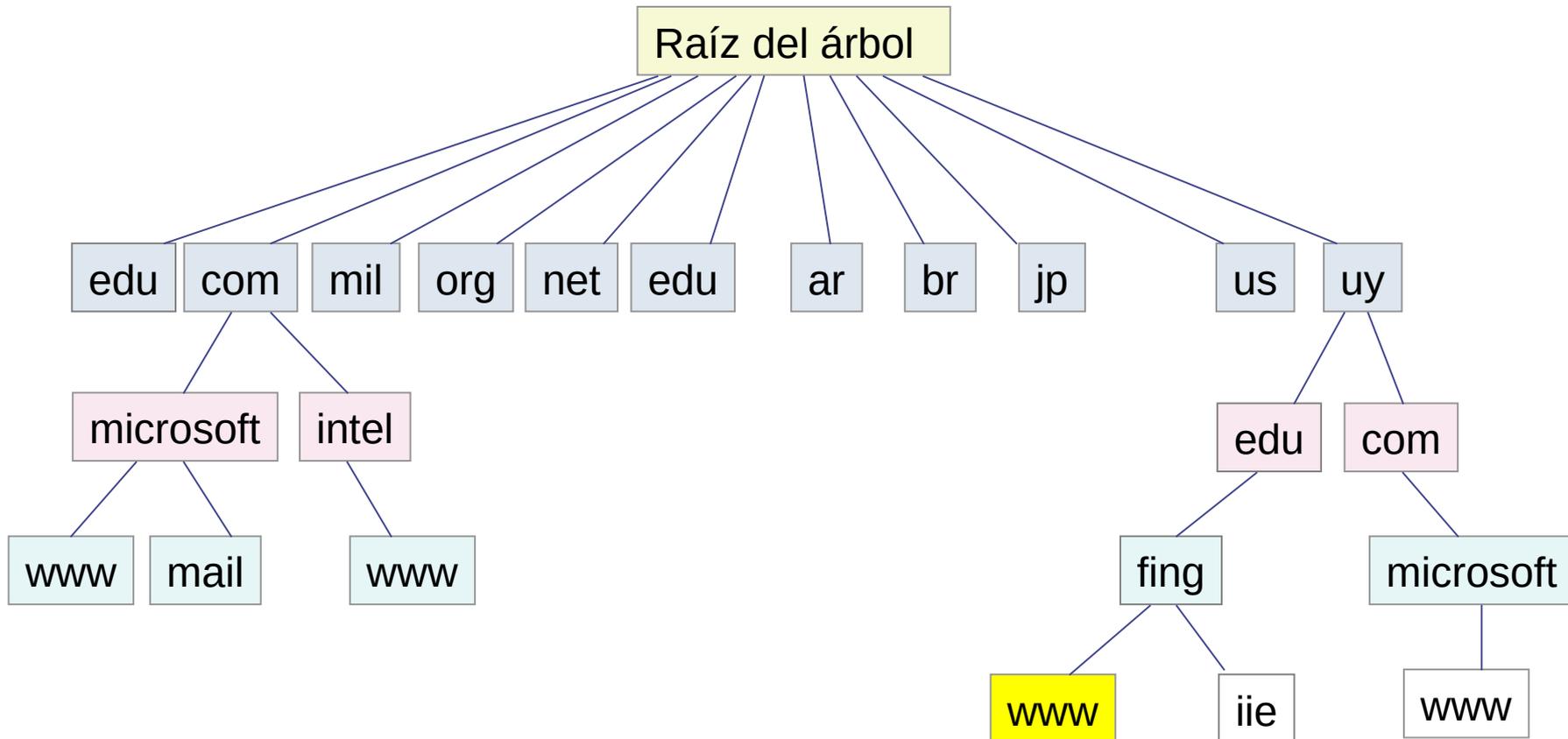
Árbol de nombre de Dominio



Ejemplo: www.fing.edu.uy

Observar: El último nivel, las hojas, tiene la correspondencia (registro A).

Árbol de Nombre de Dominio II



- Otra forma de representar el árbol.
- El nombre de dominio se construye concatenando hacia la raíz.
- Las “hojas” tienen las correspondencias(registros).

Árbol de Nombres de Dominio – DNS Root Server

- Raíz coordinada por la IANA (Internet Assigned Numbers Authority)
- Raíz: “única”
 - Convencionalmente se representa con un punto
 - 13 direcciones de servidores
 - Múltiples copias distribuidas en el mundo (anycast).

- Raíces alternativas:
 - nadie me obliga a utilizar la raíz “oficial” de la IANA
 - Pero lo usual es usarla



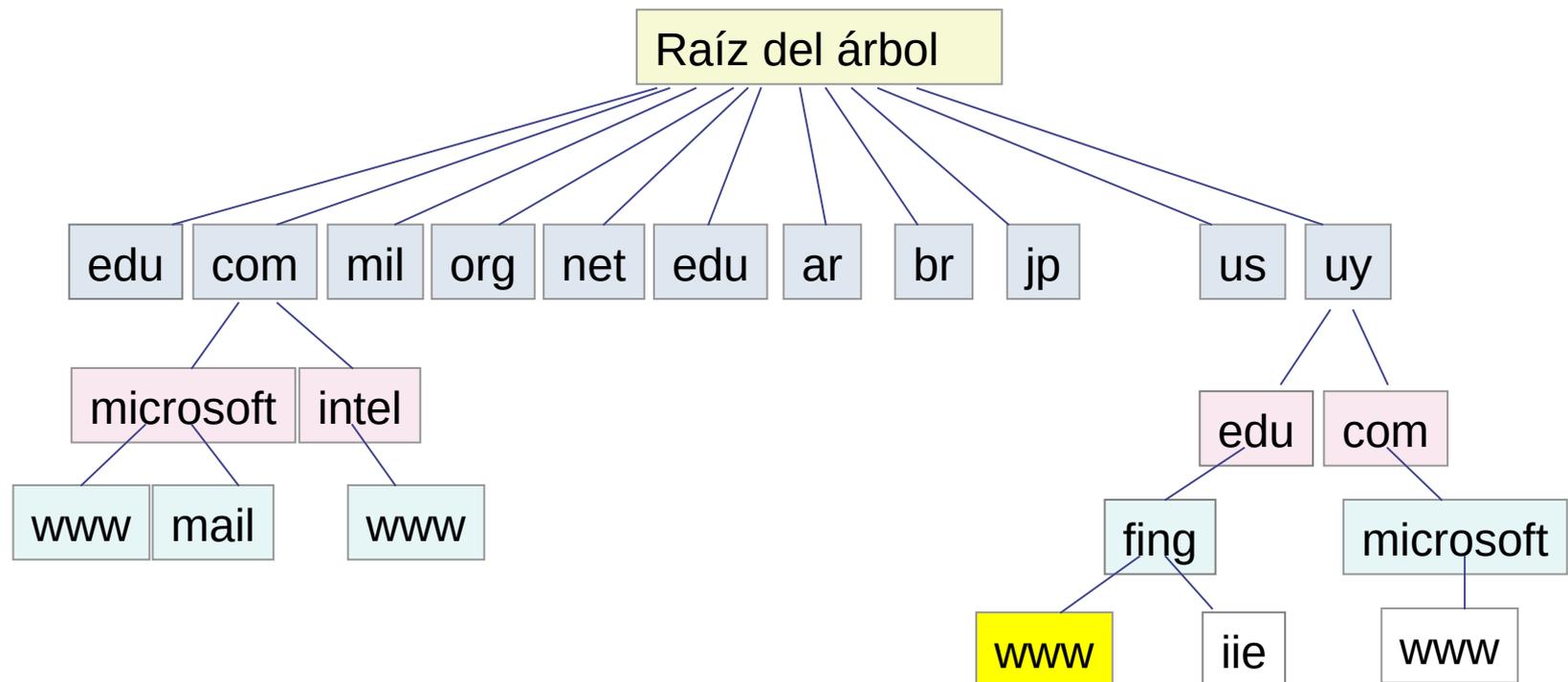
DNS – Niveles dentro del Árbol

- **Dominios de primer nivel:** TLDs (Top Level Domains)
- **Dominios genéricos:** org, net, com, mil, int, edu, info, biz, etc.
- Existen otros TLDs agregados con posterioridad
- **Dominios de países:** Código definido en ISO 3166
- Cada **TLD** define la estructura de las ramas bajo **su responsabilidad**
- En los dominios genéricos, es sencillo registrar un dominio de 2º nivel
- Para los TLD de países, **cada país define la estructura de su rama**
 - Por ejemplo en Uruguay tradicionalmente se replicó la estructura genérica (com, net, edu, etc.)
 - Actualmente es posible el registro de dominios de 2º nivel (empresa.uy)

Nombre de Dominio

- Las **etiquetas** se leen desde “las hojas” hacia la raíz
- Se concatenan uniéndolas con el carácter “.” (punto)
- Por ejemplo, la etiqueta **www** bajo el dominio **fing**, bajo **edu**, bajo **uy**, tiene el “nombre completo”:

www.fing.edu.uy.



- La información se guarda en registros de recursos, RR (Resource Records)
- Un **RR** consta de **5 campos**:
 - **Nombre**: cualquier etiqueta del árbol
 - **Clase**: actualmente solo “IN” (internet)
 - **Tipo**: el tipo de registro
 - **Valor**: el contenido del registro
 - **TTL**: tiempo de validez (en segundos) de la información (para manejo de cache)
- **Ejemplo de sintaxis**:
`www.fing.edu.uy. 1200 IN A 164.73.32.5`

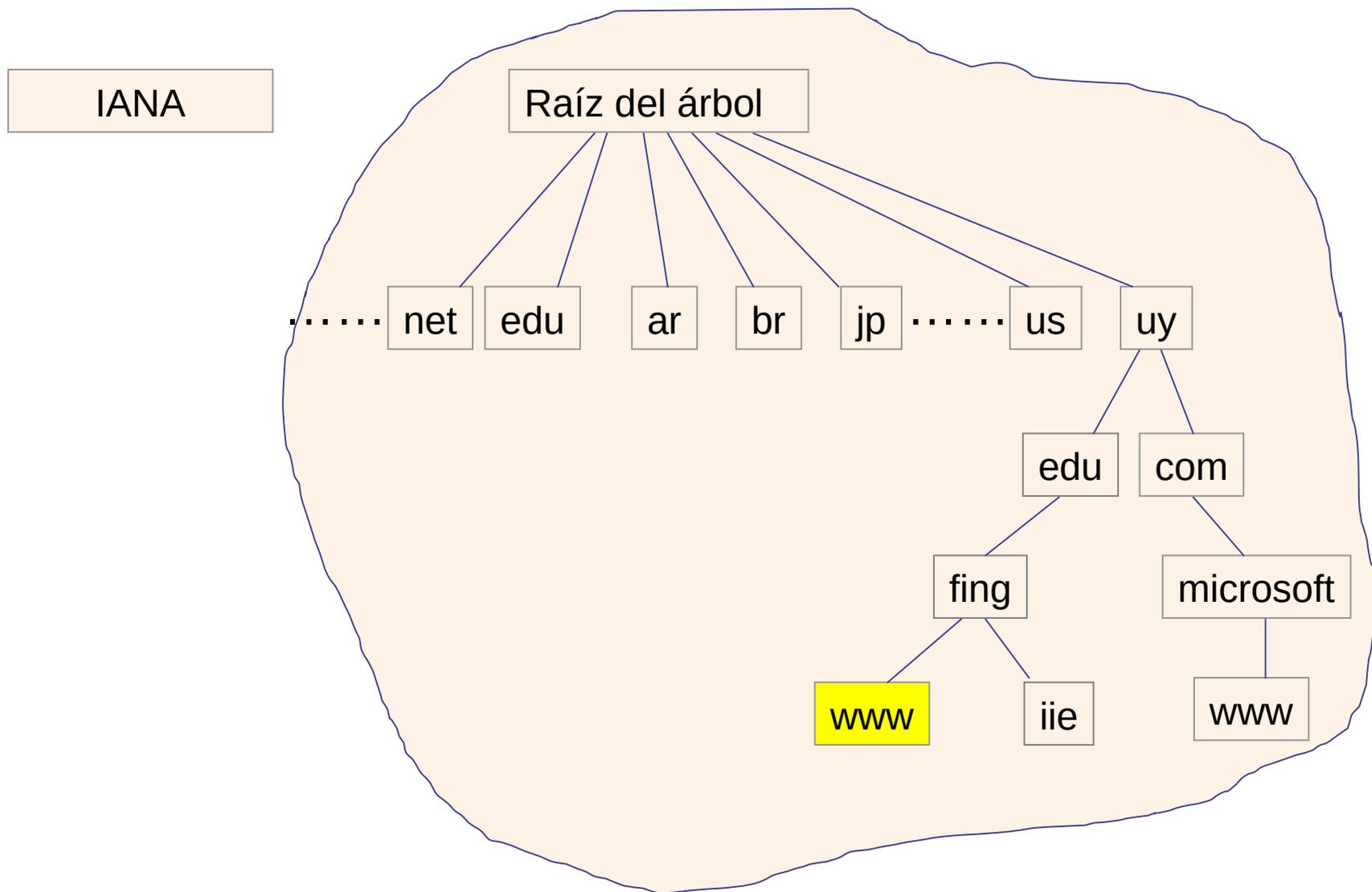
Tipos de Registros DNS (los más comunes)

Tipo	Significado	Valores	Uso
SOA	Inicio de autoridad	Parámetros de zona	Nombre del servidor de nombre de dominio, tiempo de vida y correo del administrador
A	Dirección IPv4	Entero de 32 bits expresado en notación decimal con punto	Mapea un nombre hacia una dirección IPv4
MX	Servidor de correos	Prioridad y nombre del servidor que acepta correos para ese dominio	Nombre de máquina que recibe correos para el dominio
NS	Servidor de Nombres	Nombre de máquina	Nombre de una máquina servidor de nombres para ese dominio
CNAME	Nombre Cánico	Nombre de máquina	Permite crear un alias al nombre de otra máquina
PTR	Puntero	Nombre asociado a una dirección IP	Mapea una dirección IP al nombre de una máquina
AAAA	Dirección IPv6	Entero de 128 bits expresado en notación hexadecimal con dos puntos	Mapea el nombre hacia una dirección IPv6

Zonas de Autoridad y delegación de autoridad

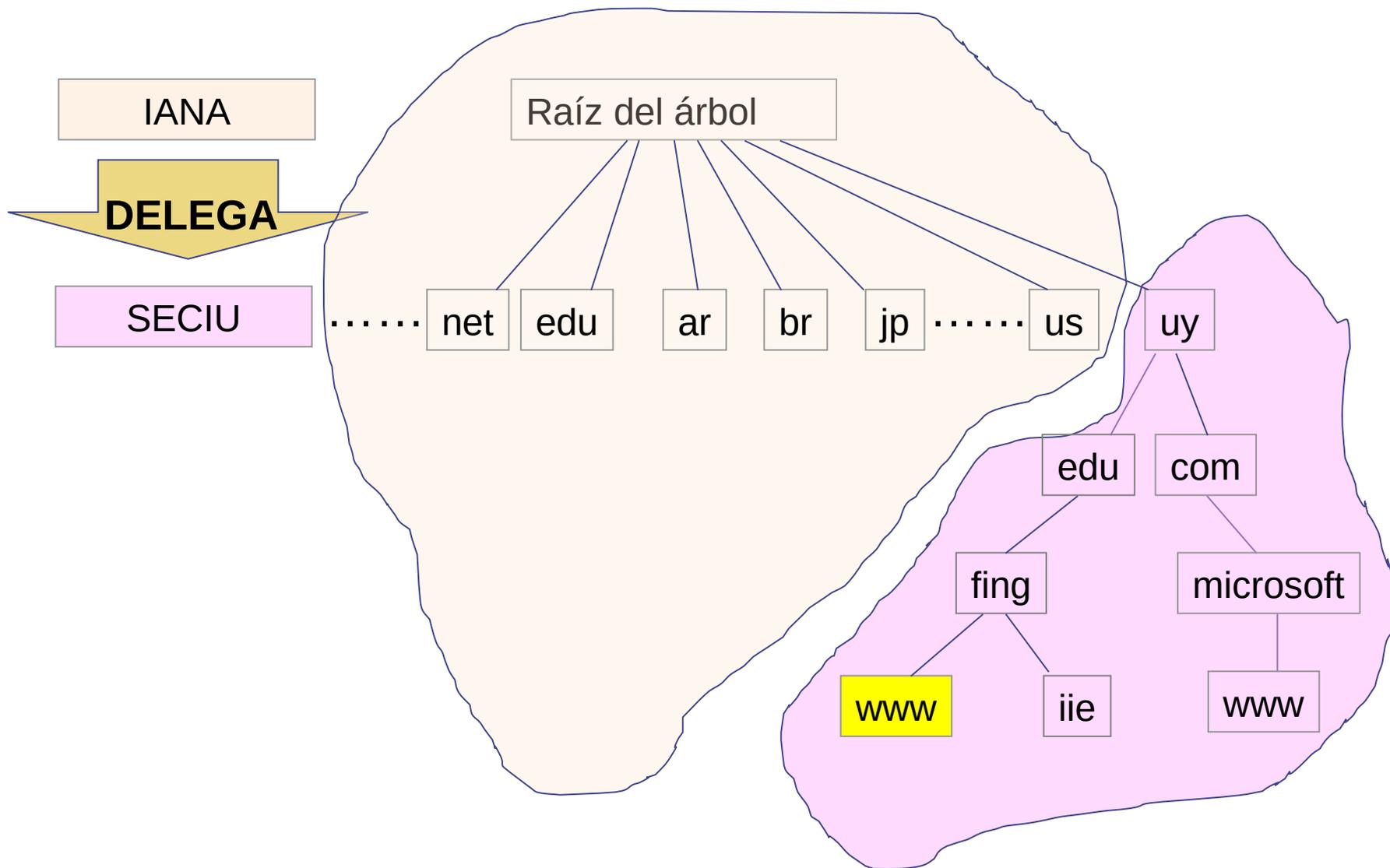
- **Inicialmente**, todo el árbol de DNS se encuentra bajo el control de quienes administran la raíz
- Para hacer el **sistema escalable**, la autoridad de distintas ramas se puede **delegar** a otras instituciones
- El administrador de cada zona puede a su vez **delegar** partes del árbol bajo su responsabilidad a otros
- La zona “**uy**” está **delegada** al SeCIU (Servicio Central de Informática de la Universidad) por IANA
- A su vez, SeCIU **delega** en ANTEL la administración del dominio “**com.uy**”
- ANTEL puede delegar a quien decida cualquier dominio bajo com.uy, ej. **miempresa.com.uy**, sin consultar a SeCIU
- SeCIU puede seguir administrando algunos de los subdominios, por ejemplo “**edu.uy**”

Delegación de Dominios - paso I

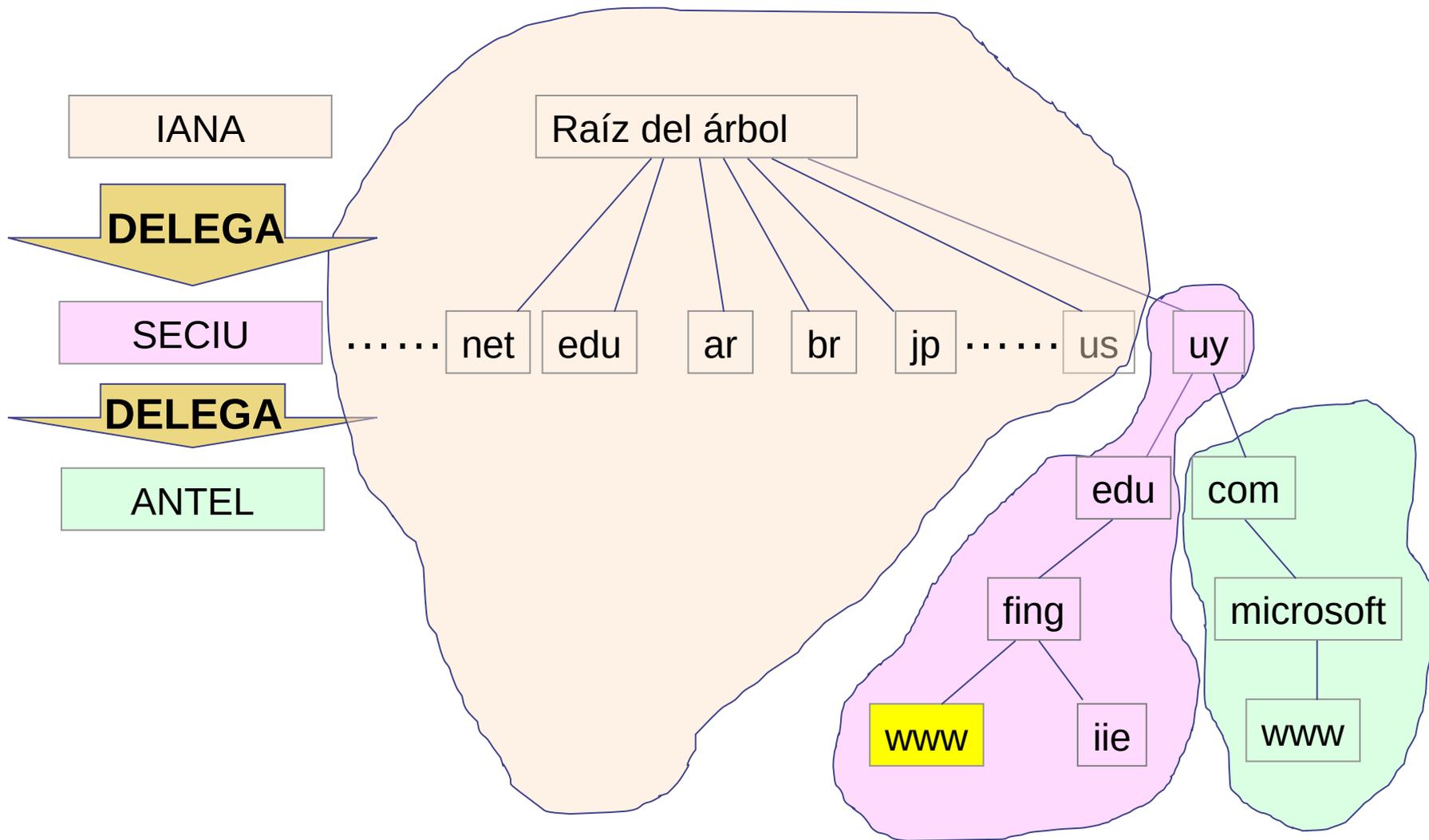


Zona de Autoridad: Porción del árbol de DNS para la cual “**estoy habilitado y debo**” responder por los **Registros** de DNS de dicha zona.

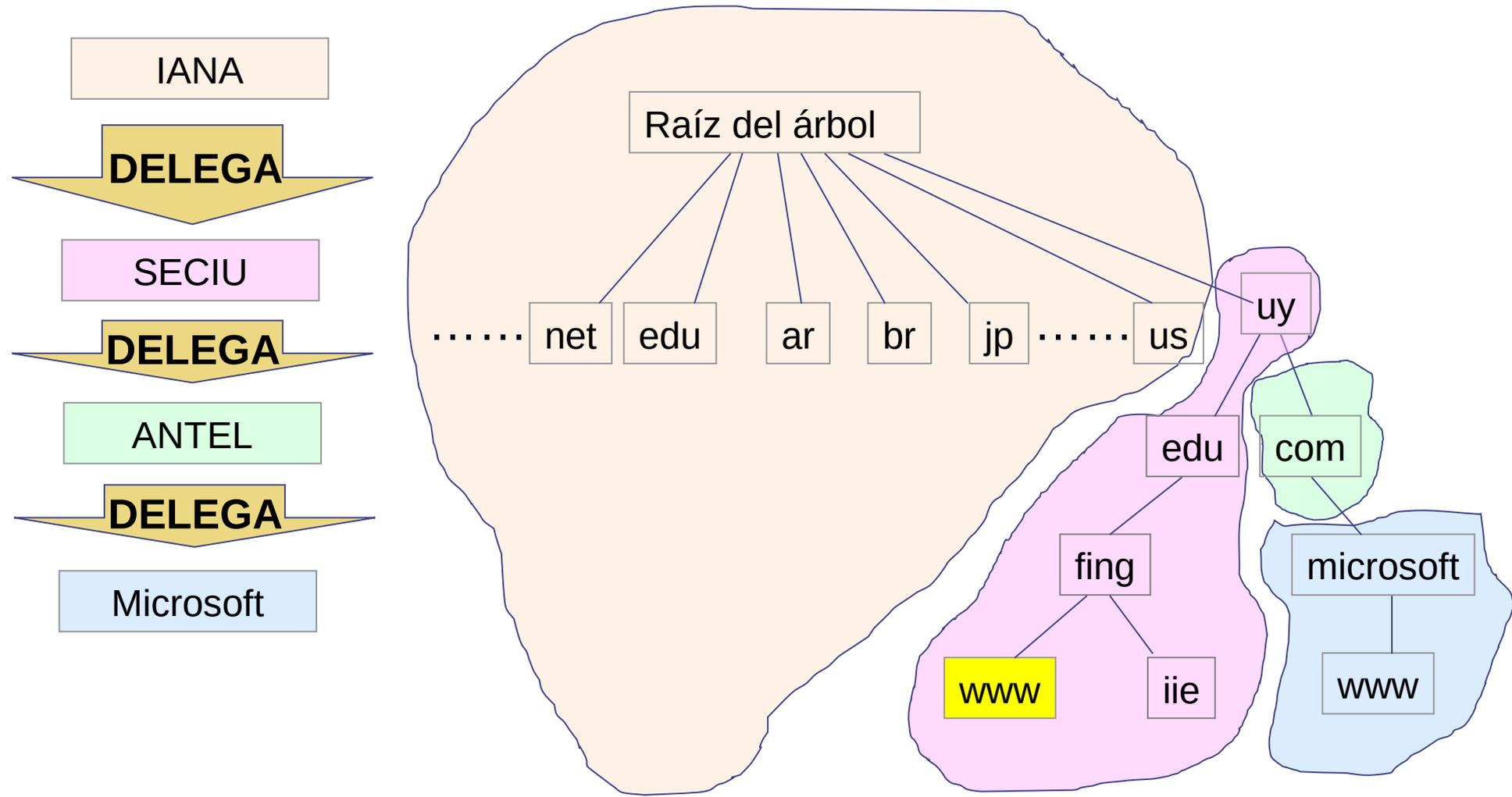
Delegación de Dominios - paso II



Delegación de Dominios – paso III



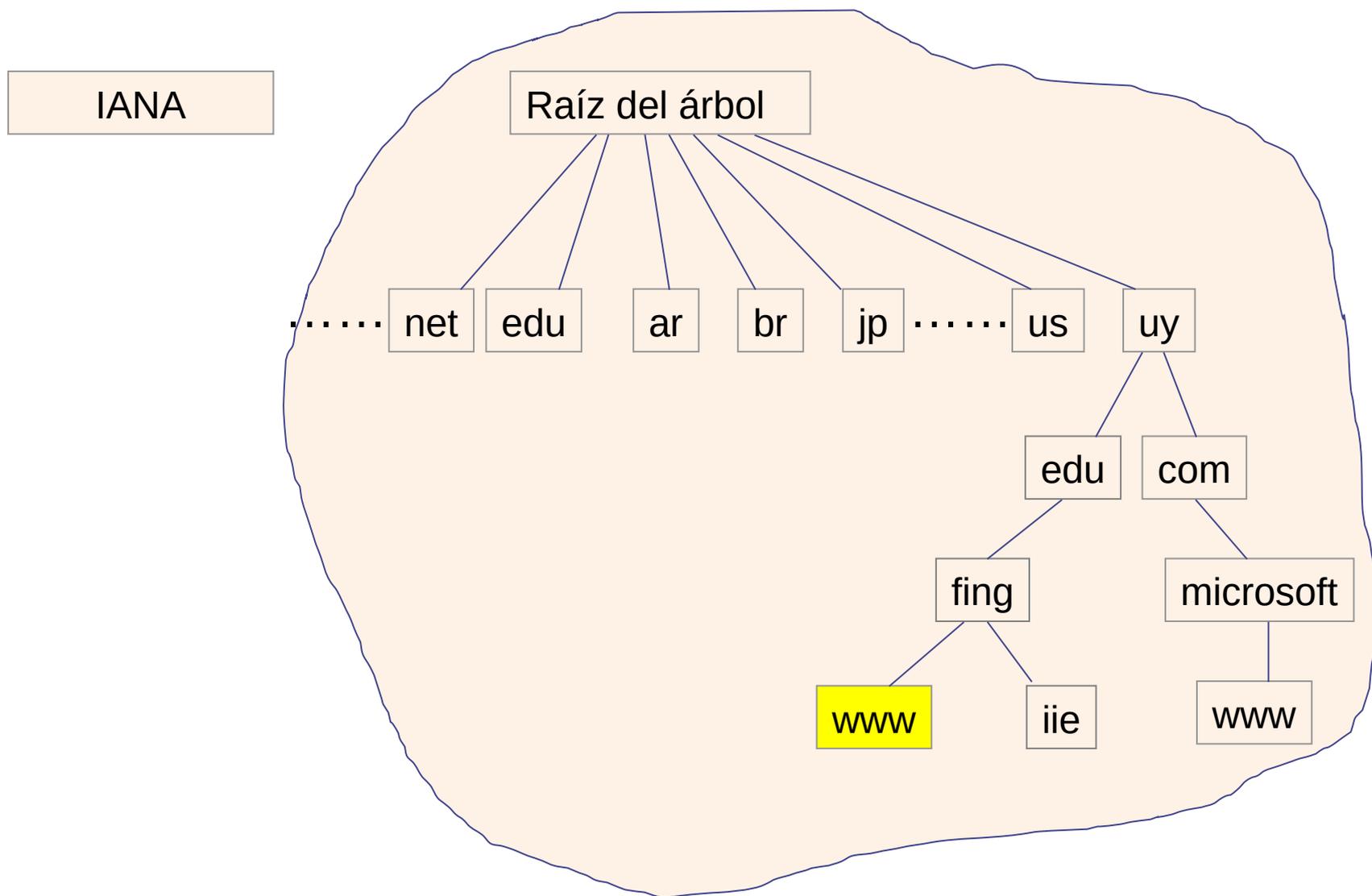
Delegación de Dominios – paso IV



Delegación de Dominios y registro NS

- Los servidores de la raíz tendrán uno o más registros **NS** indicando los nombres de los servidores de nombre de “**uy**” (ejemplo: a.nic.uy, b.nic.uy, ns.dns.br, etc)
- Los servidores de la zona “**uy**” tendrán registros **NS** indicando los nombres de los servidores de nombre de ANTEL para la zona “**com.uy**”
- Los servidores de ANTEL tienen registros **NS** indicando los servidores de nombre de cada uno de los dominios delegados
- **La Delegación de autoridad se realiza utilizando registros NS**
- La Delegación es lo que nos permite que sea **distribuido**

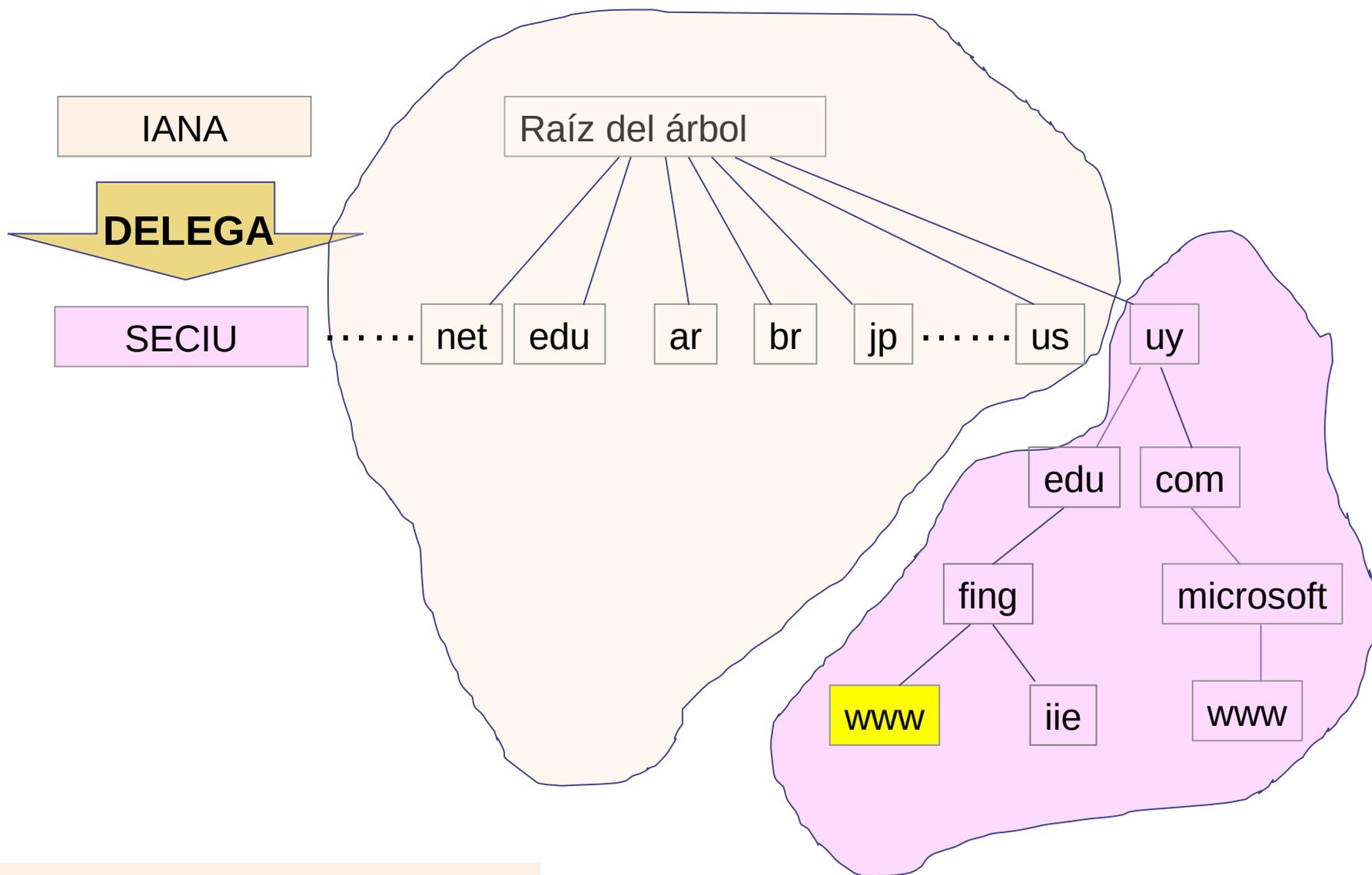
Delegación de Dominios y registro NS- paso I



DELEGAR acción en “zona padre”)

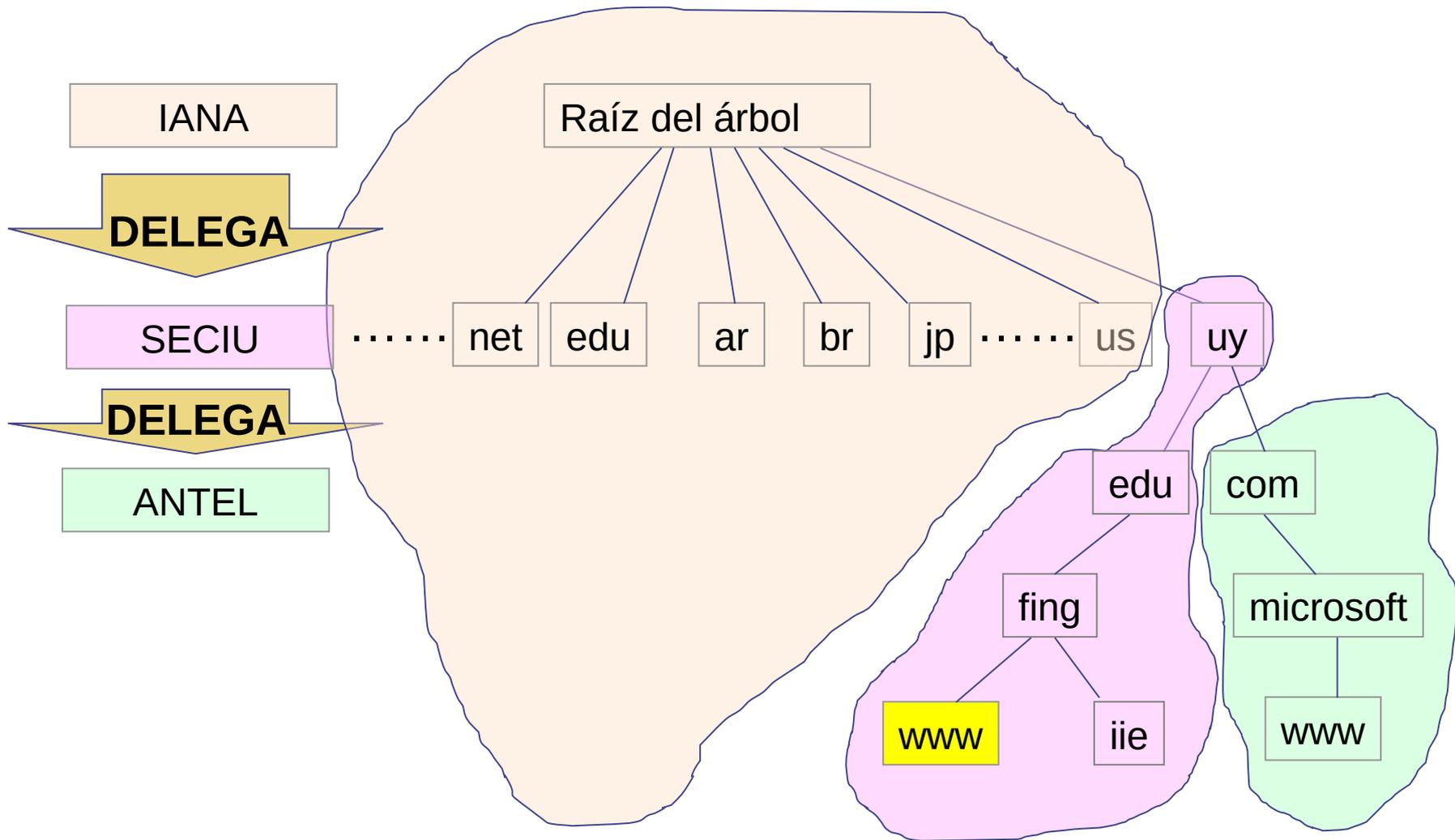
Registro DNS - “zona hija” TTL IN NS “name server zona hija”

Delegación de Dominios y registro NS - paso II



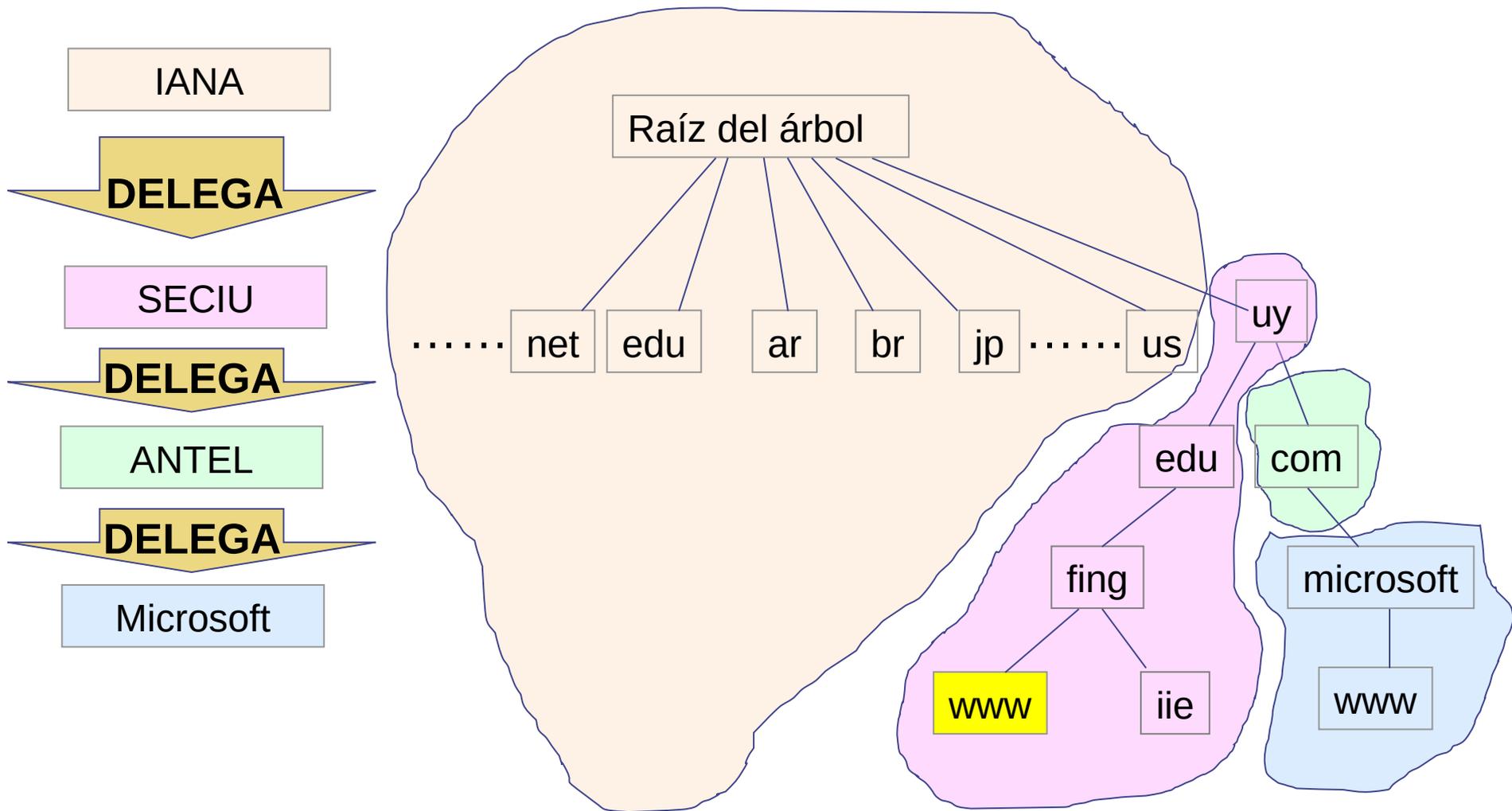
uy. 86400 IN NS a.nic.uy.

Delegación de Dominios y registro NS – paso III



com.uy. 86400 IN NS ns1.anteldata.com.uy.

Delegación de Dominios y Registro NS – paso IV



microsoft.com.uy. 86400 IN NS ns1.msft.net.

Servidores de DNS Autoritativos

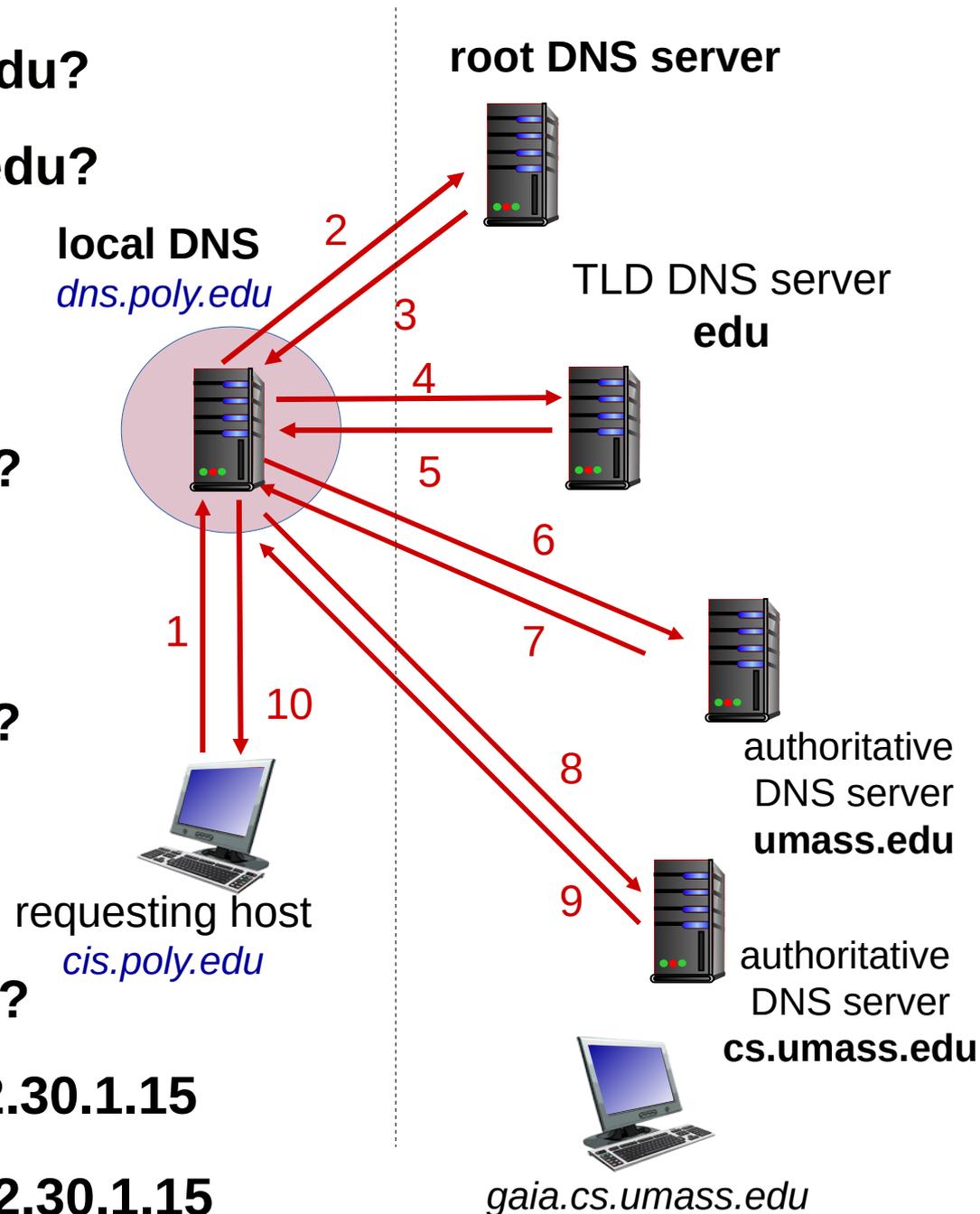
- Un servidor de DNS se dice **autoritativo** para una zona, si contiene los registros correspondientes a la misma
- Para ser de utilidad, deben poder ser encontrados siguiendo la **delegación de autoridad**
- La delegación se realiza en la “**zona padre**”. Se utiliza para ello los registros de tipo “**NS**” (name server) indicando los nombres de los servidores de nombre para la “**zona hija**”
- **Las búsquedas DNS se inician en la raíz y siguen las delegaciones.**

Consultas DNS

- **Servidor de DNS Local:** Para simplificar el software en los clientes y aumentar la eficiencia, usualmente los equipos finales no son los que realizan la búsqueda, sino que se configura un servidor (o más de uno) localmente para realizar las búsquedas
 - Por ejemplo, cada proveedor de Internet tiene un conjunto de servidores para realizar las búsquedas de sus clientes
- En el equipo final solo debo configurar dicho servidor
 - Muchas veces automáticamente
- Cuando un servidor DNS realiza una búsqueda en “**nombre de otro equipo**”, se dice que está respondiendo la consulta de forma **recursiva**
- **No todos los servidores están configurados para responder a peticiones de consultas recursivas**
 - Los servidores de **DNS locales** deben responder recursivamente
 - Se recomienda que los **servidores autoritativos** no lo hagan

Consultas DNS iterativas

- 1 – DNS Local - IN A **gaia.cs.umass.edu**?
- 2 – Elijo un root, IN A **gaia.cs.umass.edu**?
- 3 – Lista de NS de edu y **direcciones IP**
a.edu-servers.net 192.5.6.30
- 4 – Elijo uno, IN A **gaia.cs.umass.edu**?
- 5 – Lista de NS de umass.edu
ns1.umass.edu
- 6 – Elijo uno, IN A **gaia.cs.umass.edu**?
- 7 – Lista de NS de cs.umass.edu
unix1.cs.umass.edu
- 8 – Elijo uno, IN A **gaia.cs.umass.edu**?
- 9 – **gaia.cs.umass.edu** IN A 3600 **172.30.1.15**
- 10 – **gaia.cs.umass.edu** IN A 3600 **172.30.1.15**

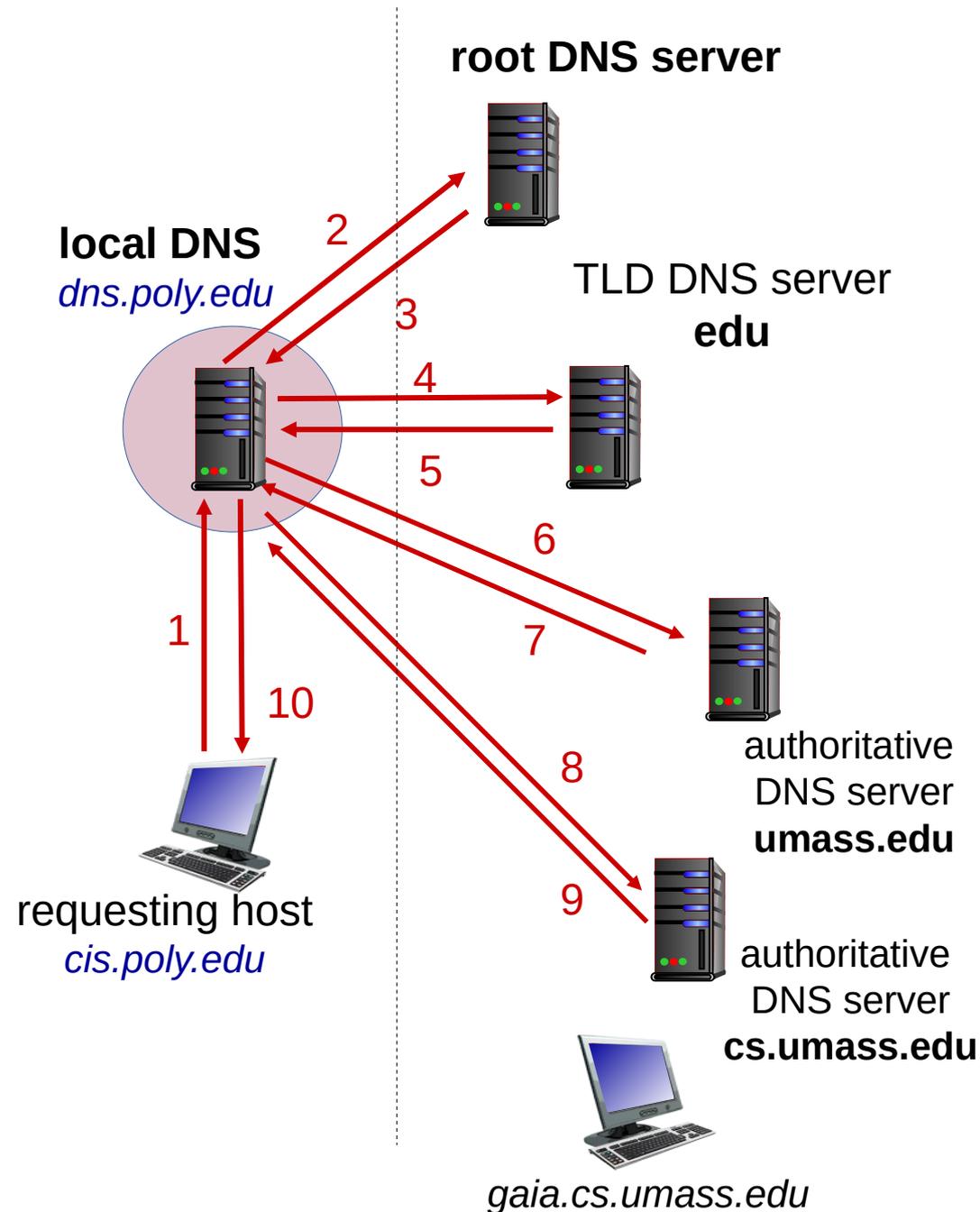


Consultas DNS iterativas

¿Porqué en los pasos 1, 2, 4, 6 y 8 consulto **siempre** por **gaia.cs.umass.edu**?

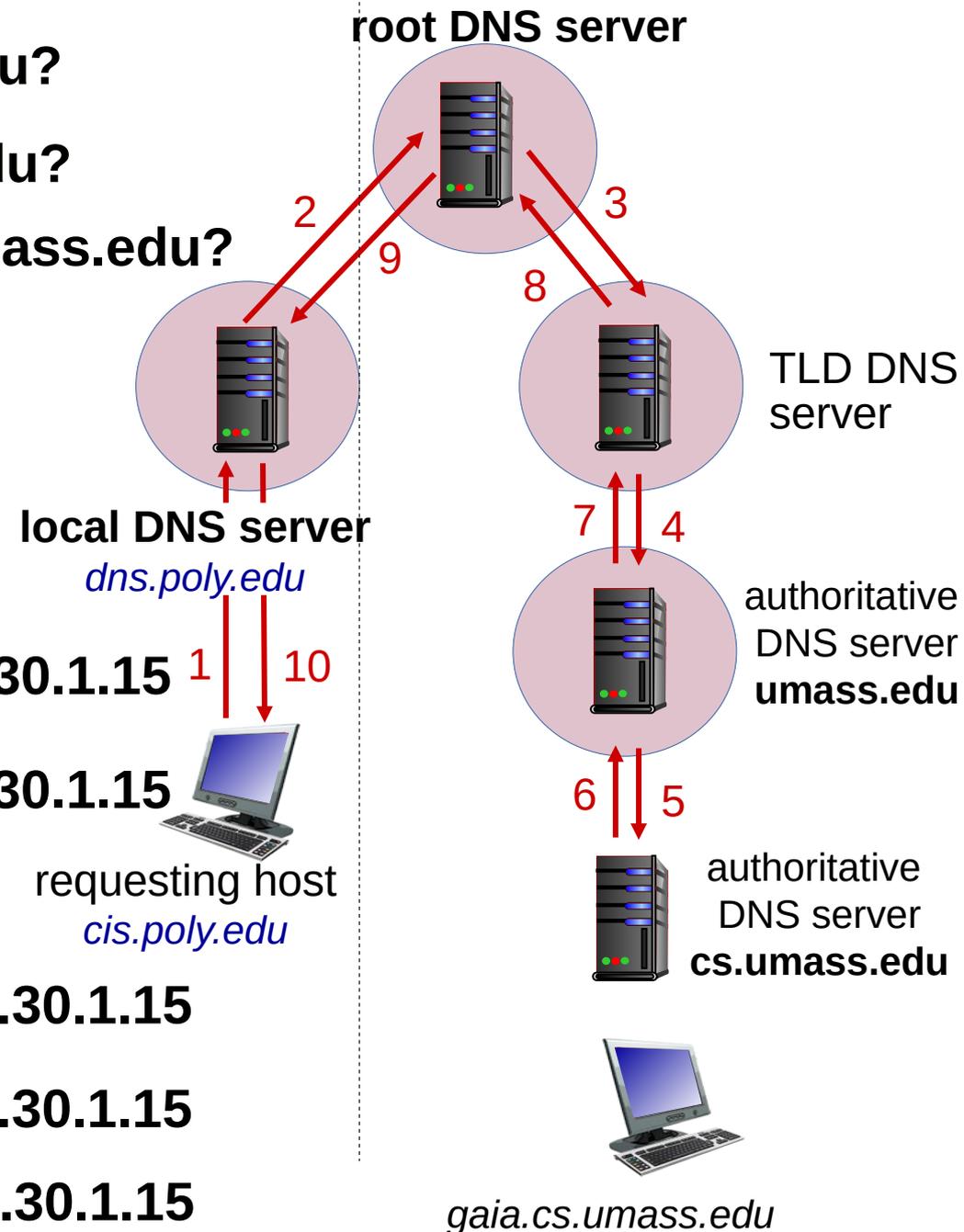
Podría solo consultar por los **NS** y avanzar en la búsqueda iterativa

¿Cómo cambia la secuencia de las consultas si el DNS Autoritativo de **umass.edu** también es Autoritativo del dominio **cs.umass.edu**?



Consultas DNS Recursivas

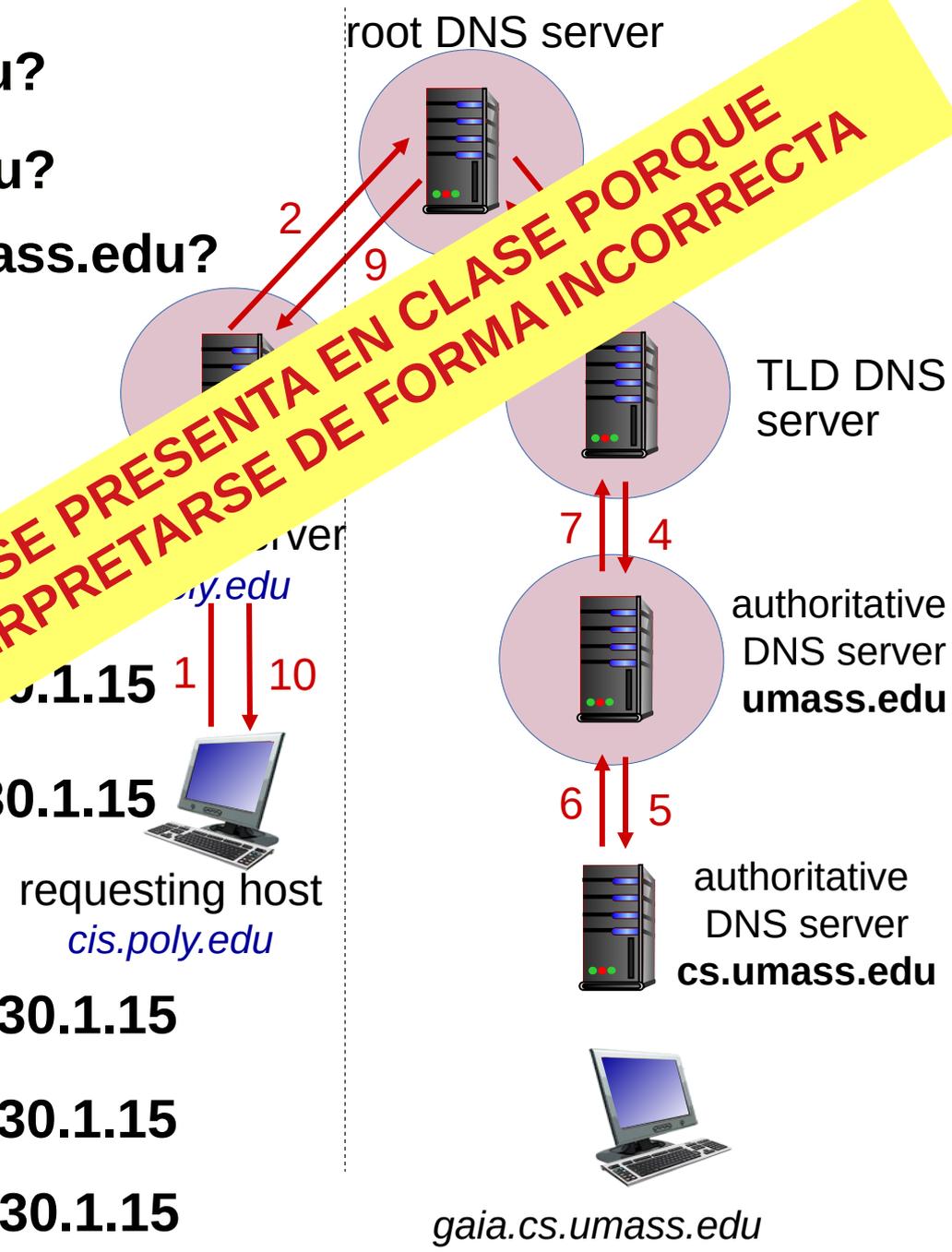
- 1 – DNS Local - IN A **gaia.cs.umass.edu**?
- 2 – Elijo un root, IN A **gaia.cs.umass.edu**?
- 3 – Elijo un NS de edu, IN A **gaia.cs.umass.edu**?
- 4 – Elijo un NS de umass.edu, IN A **gaia.cs.umass.edu**?
- 5 – Elijo un NS de cs.umass.edu, IN A **gaia.cs.umass.edu**?
- 6 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 7 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 8 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 9 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 10 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15



Consultas DNS Recursivas

- 1 – DNS Local - IN A **gaia.cs.umass.edu**?
- 2 – Elijo un root, IN A **gaia.cs.umass.edu**?
- 3 – Elijo un NS de edu, IN A **gaia.cs.umass.edu**?
- 4 – Elijo un NS de umass.edu, IN A **gaia.cs.umass.edu**?
- 5 – Elijo un NS de cs.umass.edu, IN A **gaia.cs.umass.edu**?
- 6 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 7 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 8 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 9 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15
- 10 – **gaia.cs.umass.edu** IN A 3600 172.30.1.15

ESTO NO SUCEDE EN LA REALIDAD, SE PRESENTA EN CLASE PORQUE FIGURA EN LOS LIBROS Y SUELE INTERPRETARSE DE FORMA INCORRECTA



Búsquedas DNS

- **(1)** En el ejemplo, el cliente consulta por el registro tipo **A** de **gaia.cs.umass.edu** al servidor local
- **(2)** El servidor local, si no tiene información de “**edu**”, realiza la consulta a la raíz (consulta por el registro tipo **A** de **gaia.cs.umass.edu**)
- **(3)** El **servidor raíz** responderá con los nombres de los servidores autoritativos para “**edu**” (registros NS)
 - **Con sus direcciones IP**
- **(4)** Servidor local consulta servidor de “**edu**”

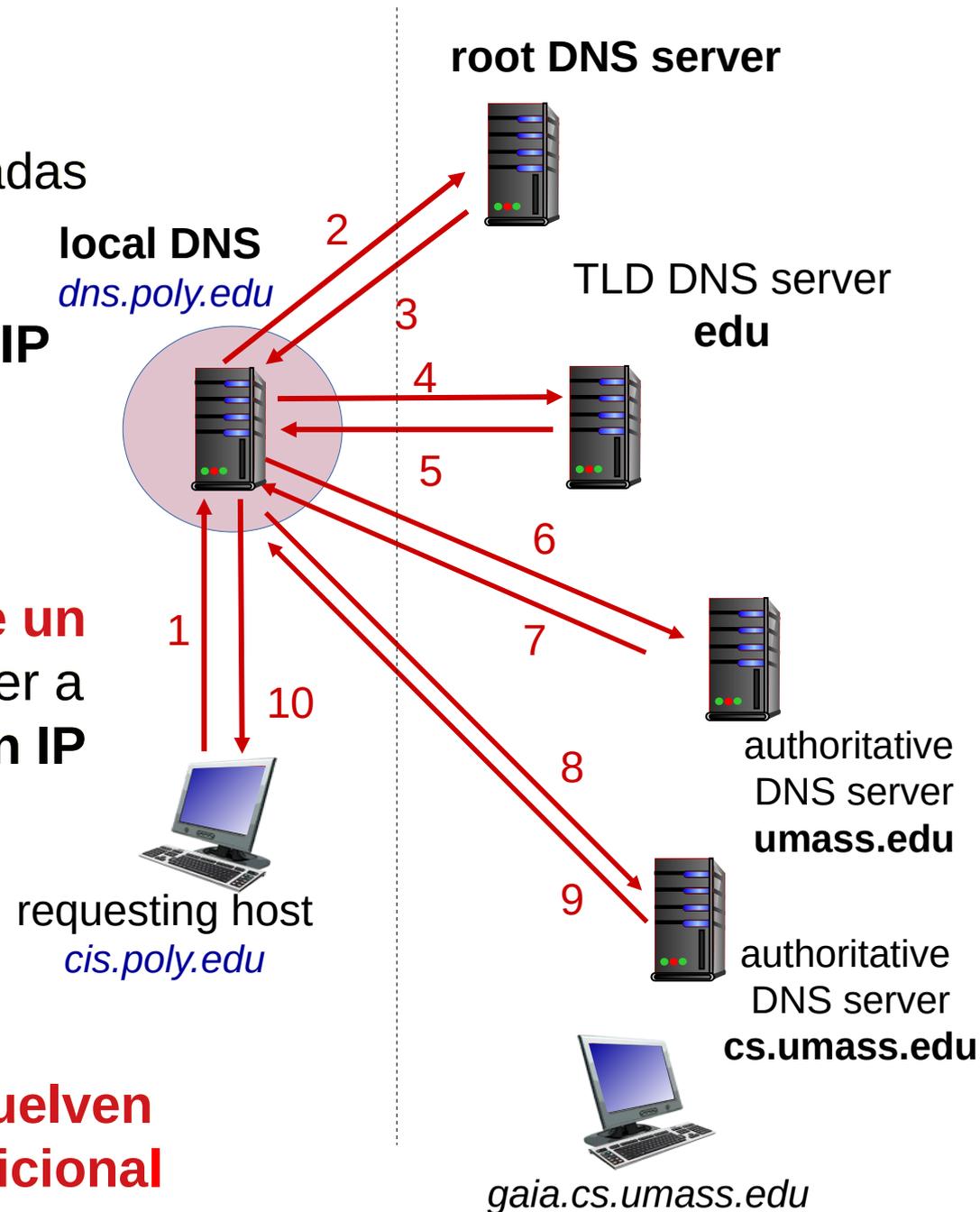
- Para poder comenzar las búsquedas, el servidor local debe conocer las direcciones de los servidores del dominio raíz
- Si el nombre del servidor pertenece a la zona de autoridad, no podríamos resolver la consulta. Para solucionar el problema, el servidor de la “**zona padre**” envía también el registro **A** correspondiente al nombre del servidor (en Additional Section)
- A esto se le llama “**glue record**”

Consultas DNS – ¿Cómo avanzo?

- 1 – DNS Local configurado en PC
- 2 – Lista de 13 direcciones Ips configuradas desde archivo.
- 3 – Lista de NS de edu y direcciones IP
a.edu-servers.net 192.5.6.30

Recordar que el **registro NS devuelve un nombre de dominio**, pero para acceder a un equipo remoto necesito la **dirección IP**

Los root DNS server SIEMPRE devuelven la dirección IP como un registro adicional



Consultas DNS – ¿Cómo avanzo?

7 – Lista de NS de cs.umass.edu

unix1.cs.umass.edu

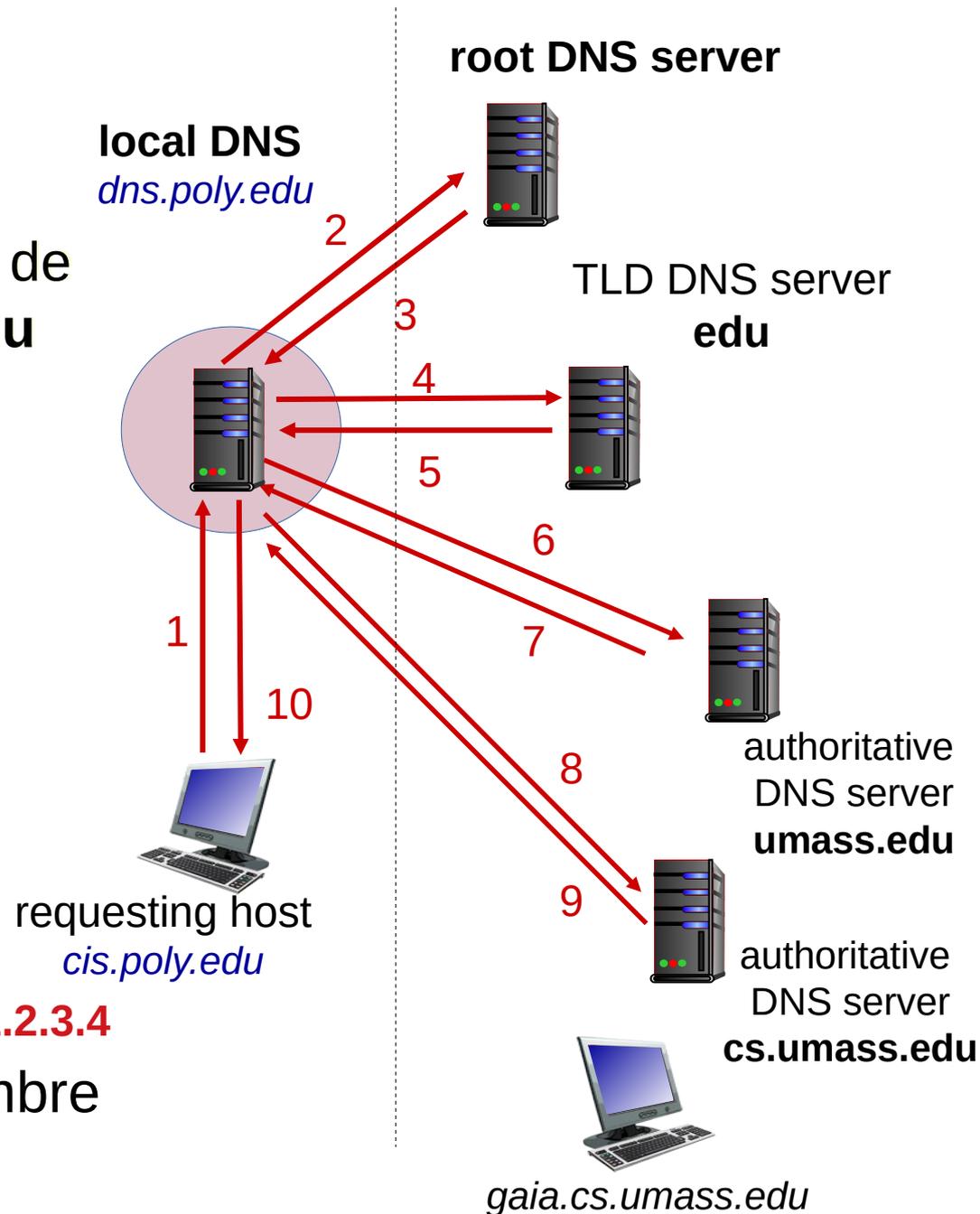
El servidor de nombre que responde de forma autoritativa para **cs.umass.edu** se llama **unix1.cs.umass.edu**

¿Cómo obtengo la dirección IP el servidor de nombre autoritativo **unix1.cs.umass.edu**?

Cuando el nombre del servidor está dentro de mi zona de autoridad **debe de agregarse el registro A “glue record” en Additional Section.**

unix1.cs.umass.edu TTL IN A 1.2.3.4

¿Qué sucede si el servidor de nombre es **unix1.cs.umass.edu.uk**?



- Para hacer eficiente al DNS, tanto los clientes como los servidores recursivos pueden guardar la información obtenida
- Para determinar el tiempo de validez de dicha información, se utiliza el campo **TTL** (tiempo de vida) de los registros
- Los servidores no solo guardan el resultado final, sino también la información de los servidores intermedios

Cache DNS

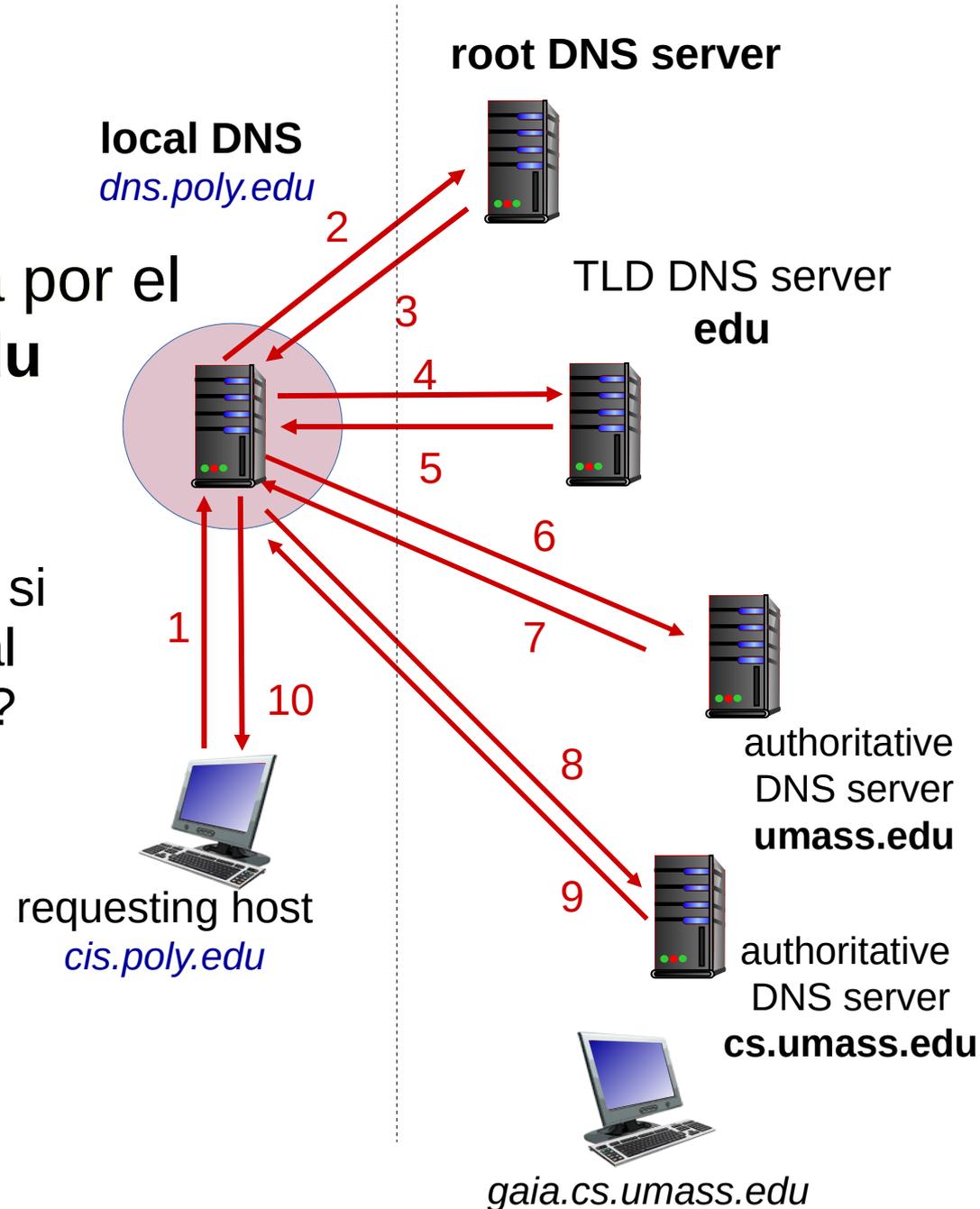
Inmediatamente después que consulto y obtengo la respuesta por el registro A de **gaia.cs.umass.edu**

¿Cuál sería la secuencia de consultas si otro cliente consulta inmediatamente al DNS local por el registro A asociado a?

A – gaia.cs.umass.edu

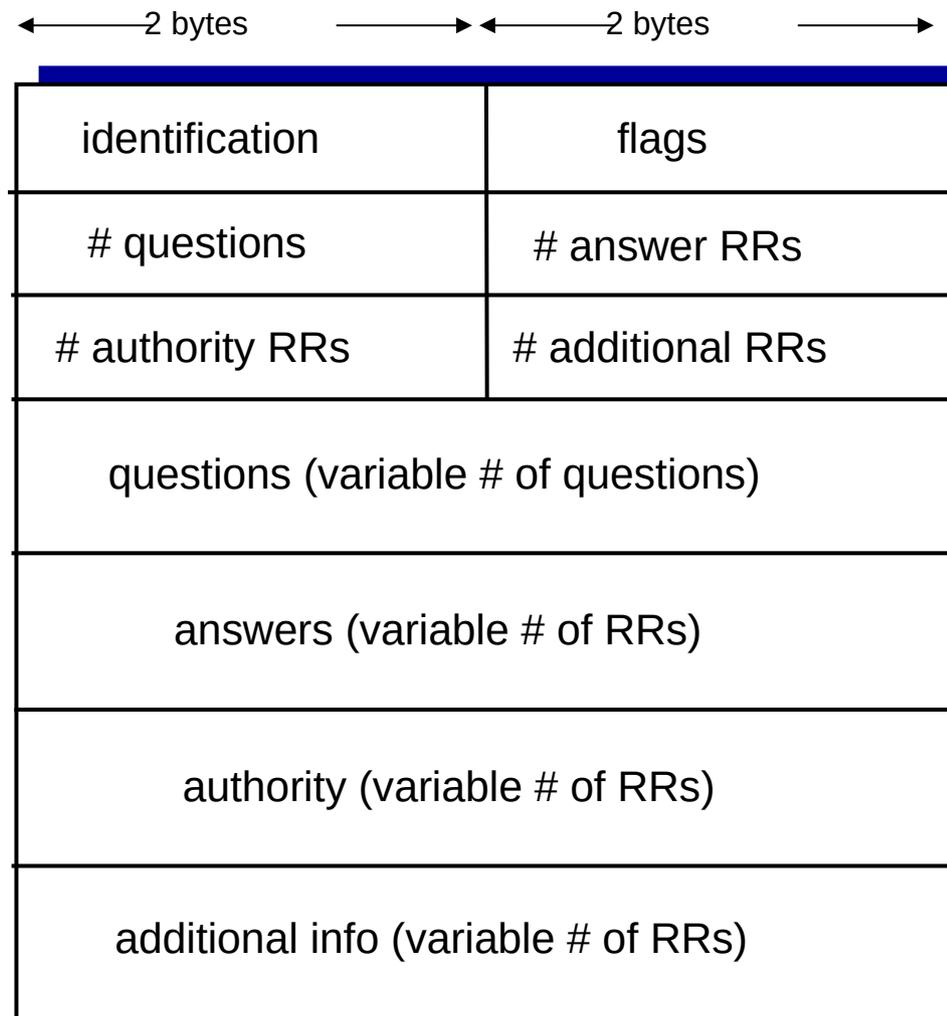
B – volta.umass.edu

C – www.stanford.edu



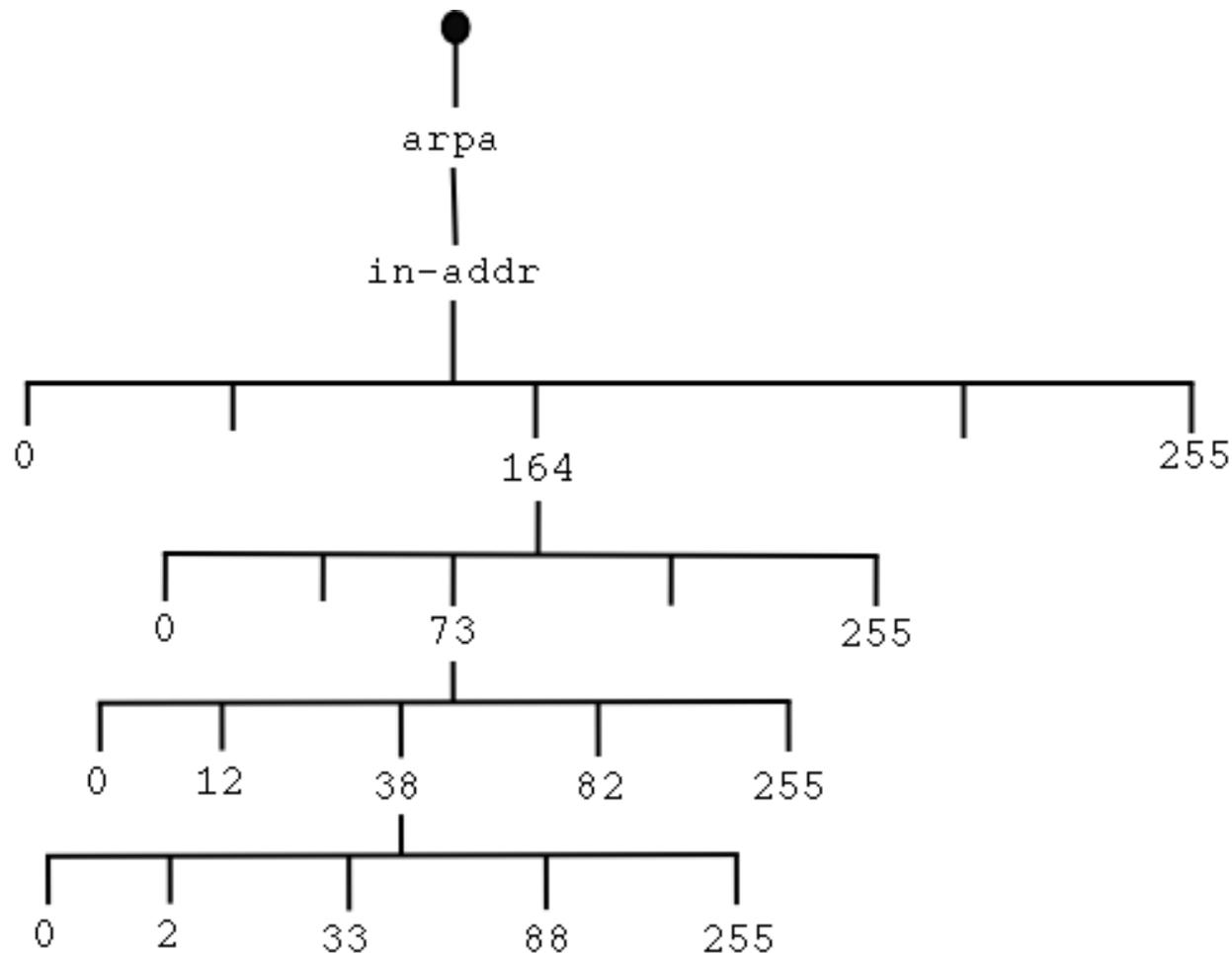
Formato de Mensaje DNS

- Consultas y respuestas tienen el mismo formato
- Hay un **flag** que indica si es consulta o respuesta
- Hay un **flag** que indica si deseamos que la consulta sea recursiva, y otro **flag** que indica si es o no permitida (respuesta).
- El campo de pregunta lleva el nombre, clase y tipo consultado. Se repite en mensaje de respuesta
- El campo Respuestas trae 0 o más registros correspondientes a lo consultado
- En el campo Autoridad se listan los registros NS de los servidores autoritativos
- El campo de información adicional trae información como las direcciones IP de los servidores autoritativos, etc.



Búsqueda del nombre a partir de la IP – Búsqueda Inversa

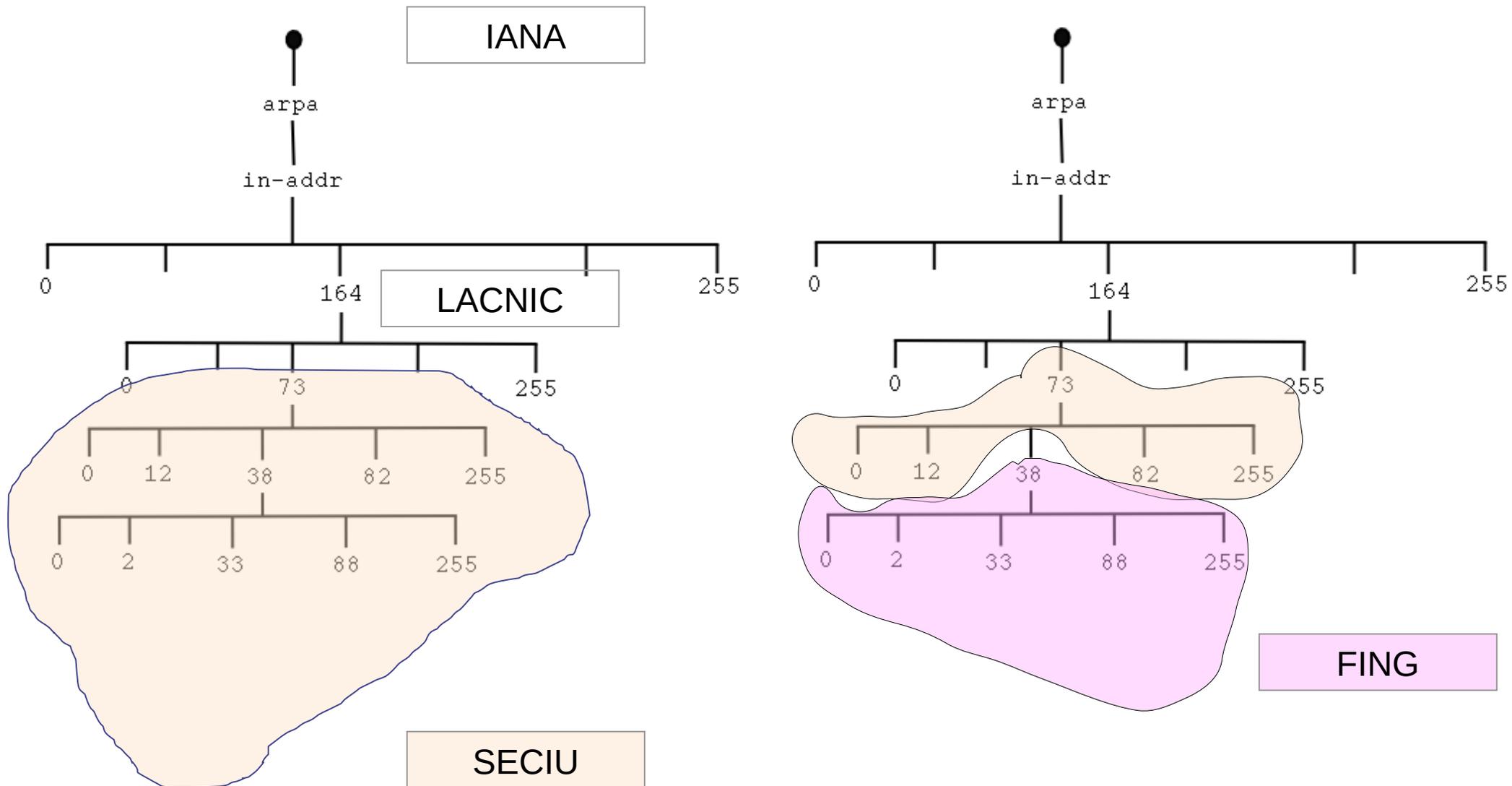
- Se define una nueva rama, **“in-addr.arpa”**
- Se utiliza un nuevo tipo de registro: **PTR**



- Para buscar el nombre correspondiente a la dirección **A.B.C.D**, se consulta por el registro PTR correspondiente a **D.C.B.A.in-addr.arpa**

- Las direcciones **se asignan en bloques** (potencias de 2), y una asignación puede ser partida y reasignada
- Si por ejemplo SECIU recibió el bloque 164.73.0.0 – 164.73.255.255, recibirá (se le delega) la rama **73.164.in-addr.arpa**
- Si delega la subred 164.73.38.0/24 (rango 164.73.38.0 a 164.73.38.255) a otra institución (FING), puede delegarle la rama **38.73.164.in-addr.arpa**

Delegación de árbol inverso/reverso



Los reversos son una exigencia del **RIR** (Regional Internet Registry) para asignar un bloque de direcciones IP a una empresa.

DNS – Agrupación lógica

- Las direcciones **se asignan en bloques** 164.73.0.0 – 164.73.255.255)
- El “**ecaminamiento**” o “**forwarding**” de paquetes se realiza basado en el rango de direcciones IP, los rangos (o bloques) “normalmente” tienen asociación geográfica.

- Si mi empresa es multipaís, puedo hacer que

www.empresa.com. TTL IN A IP_USA

www1.empresa.com. TTL IN A IP_ALEMANIA

Agrupación lógica en vez de geográfica.

- Las empresas “no suelen” tener direccionamiento IP propio, depende de su proveedor de Internet.

¿Qué sucede si cambio de proveedor? Si todo esta don nombres DNS, simplemente cambio:

Antiguo) **www.empresa.com. TTL IN A IP_proveedor_antiguo**

Nuevo) **www.empresa.com. TTL IN A IP_proveedor_nuevo**

La empresa no es “dueña” de las direcciones IPs pero si puede serlo de sus nombres de dominio.

IPv6 e impactos en capa de Aplicación

- Llamada a primitivas (sock)
- Consultar registro AAAA y usar dirección IPv6.
- Permisos asociados a direcciones IPv6.
- Protocolos de capa de aplicación que envíen direcciones IPv6 de los interlocutores (SIP, opciones de HTTP)
- Aplicaciones que asignan direcciones RADIUS/DIAMETER o DHCP
- Largo de campos en Bases de Datos o logs de eventos