Protocolos de Capa de aplicación

Protocolos de comunicación
(Correo, Web)

Preparando el Viaje

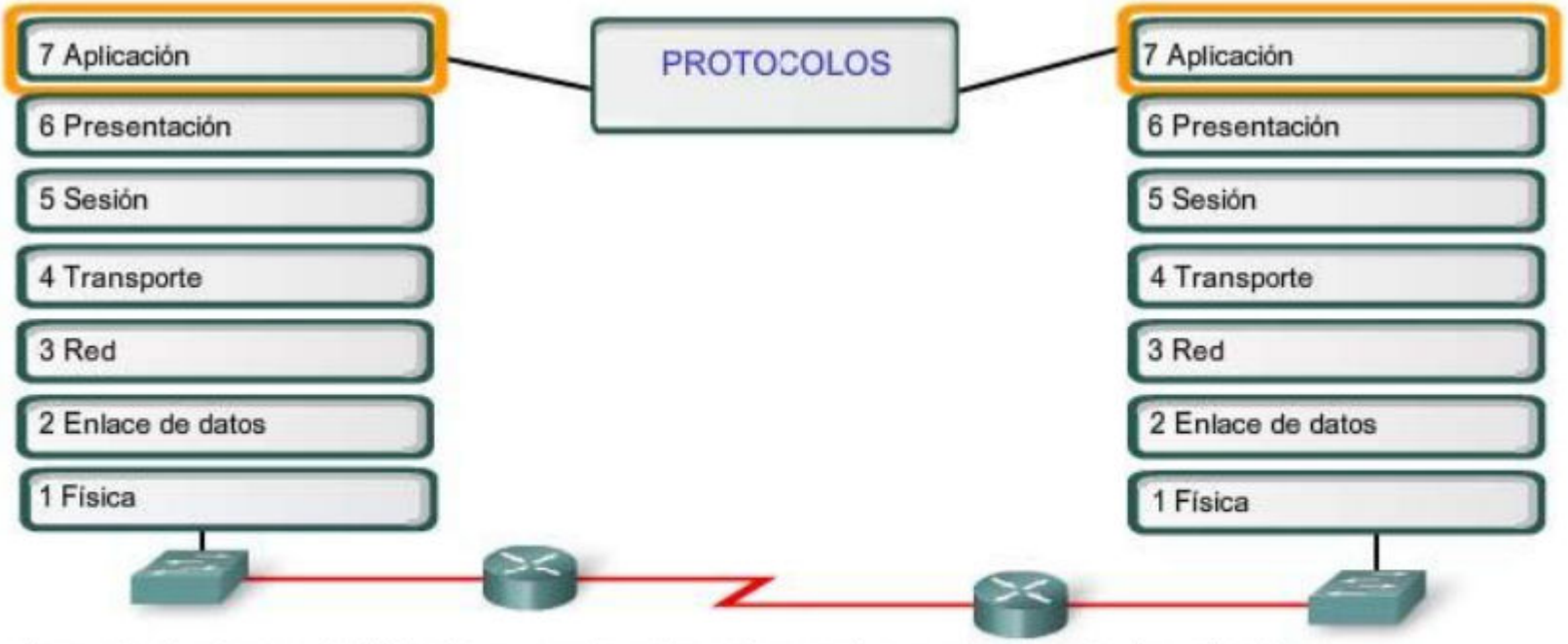
•Llegamos a casa encendemos nuestra computadora personal.
•Buscamos en google un alojamiento.
•Enviamos un mail para pedir una reserva

¿Que protocolos de nivel aplicación participaron para que esto sea posible?

¿Que protocolos se usaron?

- Protocolo Http
- Protocolo SMTP
- Protocolo POP3
- DNS
- DHCP

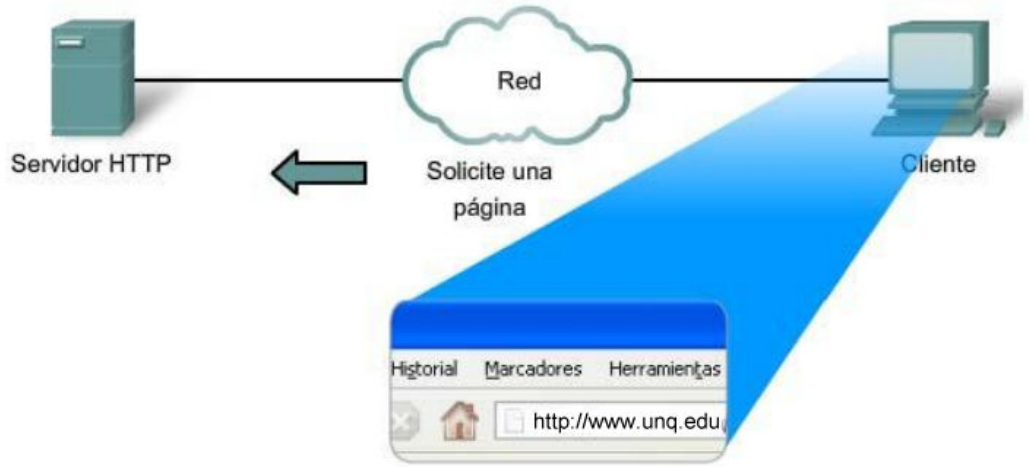
Que se define en los protocolos



Los protocolos de capa de Aplicación proporcionan las reglas para la comunicación entre las aplicaciones.

- Los procesos en cada uno de los extremos de la comunicación.
- Los tipos de mensajes
- La sintaxis de los mensajes.
- El significado de los campos de información.
- La forma en que se envían los mensajes y la respuesta esperada.
- La interacción con la próxima capa inferior.

Modelo utilizado



- Estos protocolos utilizan el modelo Cliente/Servidor

Paso a Paso

- Antes que el navegador pueda establecer la conexión http necesito saber que IP tiene nuestro buscador preferido!
- Lo mismo nos sucedió con nuestro cliente de correo al enviar el MAIL.
- ¿Como lo averigua?

  Gracias al Protocolo de Servicio de DNS

DNS (Domain Name System)

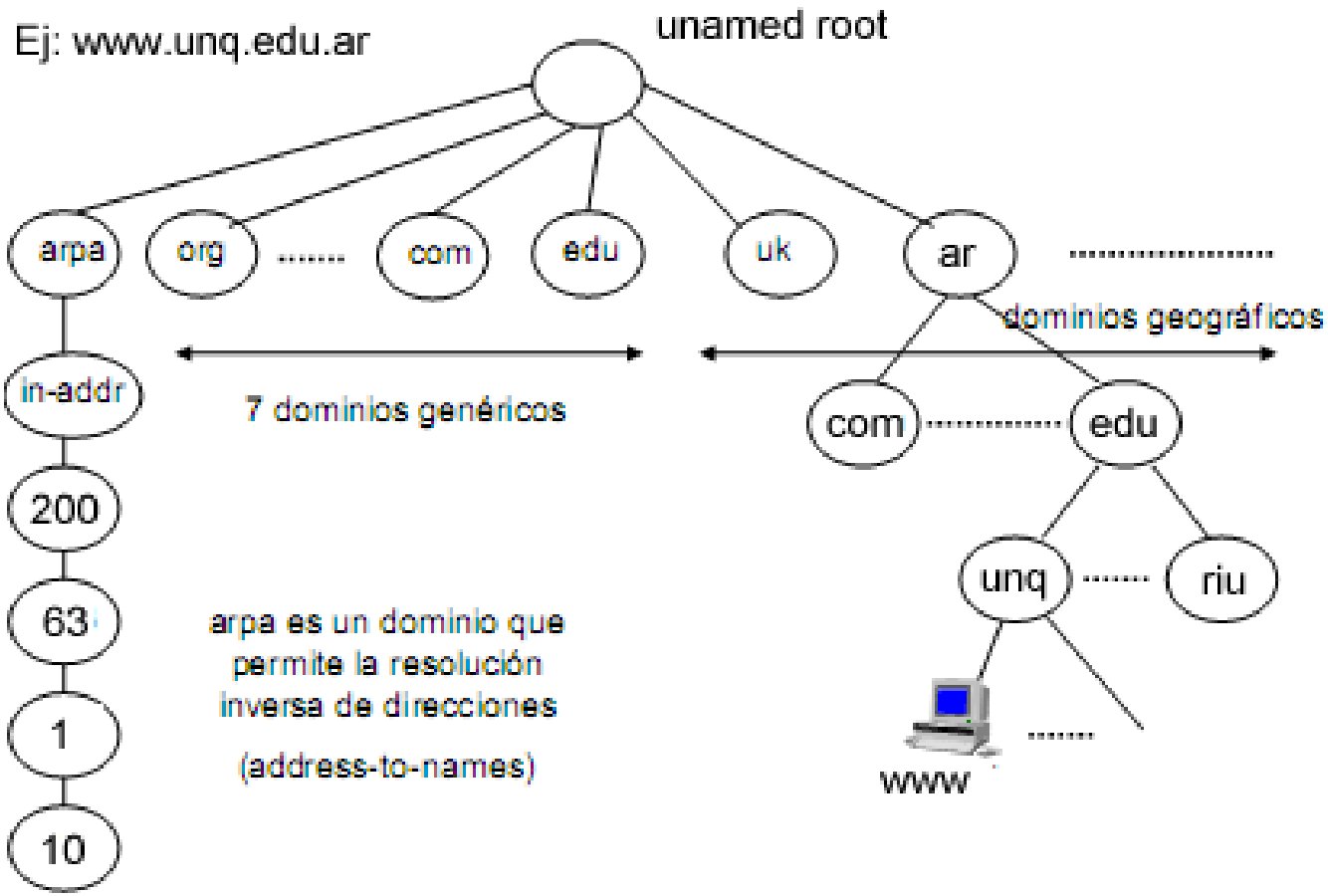
- Sistema usado en Internet para traducir nombres de nodos de red en direcciones IP.
- A cada máquina se le asigna una dirección IP (200.63.1.10) y un nombre (www.unq.edu.ar)
- DNS es una base de datos distribuida que permite realizar un mapeo entre nombres de máquinas y direcciones IP
- De esta manera cuando queramos acceder a una máquina (Web, Ftp, SMTP, Telnet, ...) en vez de recordar la IP, basta recordar el nombre

Asignación de Dominios

- Un dominio es un mecanismo de identificación utilizado en Internet
- El ICANN es la responsable que los nombres de las máquinas sean únicos
- Consta de varias palabras separadas por un punto: xxx.xxx.com
- Las primeras palabras xxx.xxx indican un conjunto de nombres que identifican a la empresa u organización
- Las empresas deben registrar su nombre para que pase a ser su marca en Internet (problemas con marcas ya registradas)
- El resto del dominio está organizado en:
  - Dominios genérico: .com, .org, .net, .edu, ...
  - Dominios geográficos: .es, .fr, .uk, .it, ...
- Propuesta (de CORE) para ampliar el número de dominios genericos. .firm, .shop, .info, .web, .nom, .arts, .rec

Jerárquia de nombres

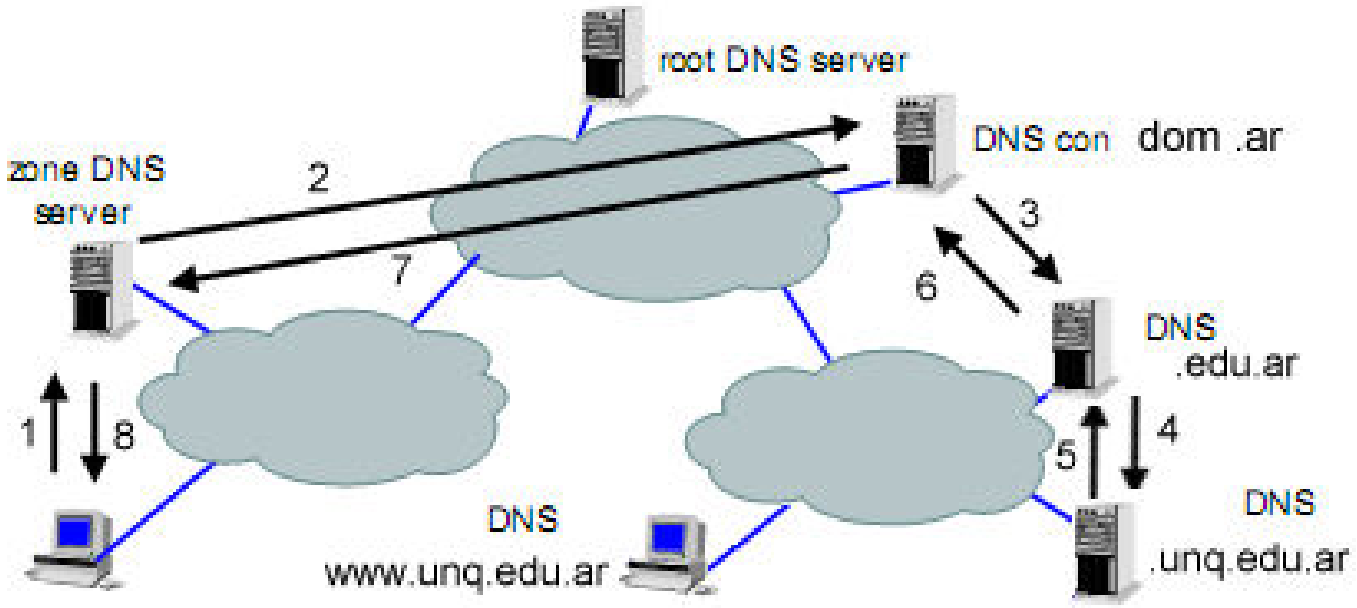
- DNS basa su funcionamiento en un espacio jerárquico de nombres



Caracteristicas del DNS

- DNS se basa en servidores de nombres distribuidos geográficamente
- Puerto 53 (tanto UDP como TCP)
- Cada administrador de sistemas de una zona es responsable de mantener Un servidor de DNS primario (disk file), tiene la información de una zona, y la autoridad sobre ella.
- Uno o varios servidores de DNS secundarios (backups) independientes del primario pero que obtienen la información a partir del primario (normalmente se actualizan cada 3 horas)
- Cuando se conecta un nuevo equipo hay que añadirlo al primario (IP y nombre), mientras que los secundarios lo obtendrán ("zone transfer") del primario (hacen "querys" del primario cada 3 horas)
- Si la información no está en el DNS de la zona, este servidor debe acceder a uno de los servidores "root" para obtenerla (DNS servers deben conocer la IP de los roots)
- Los servidores DNS disponen de caches para resolver nombres que han consultado recientemente.
- En los Hosts corren clientes DNS los cuales dan soporte a las aplicaciones que lo requieran.

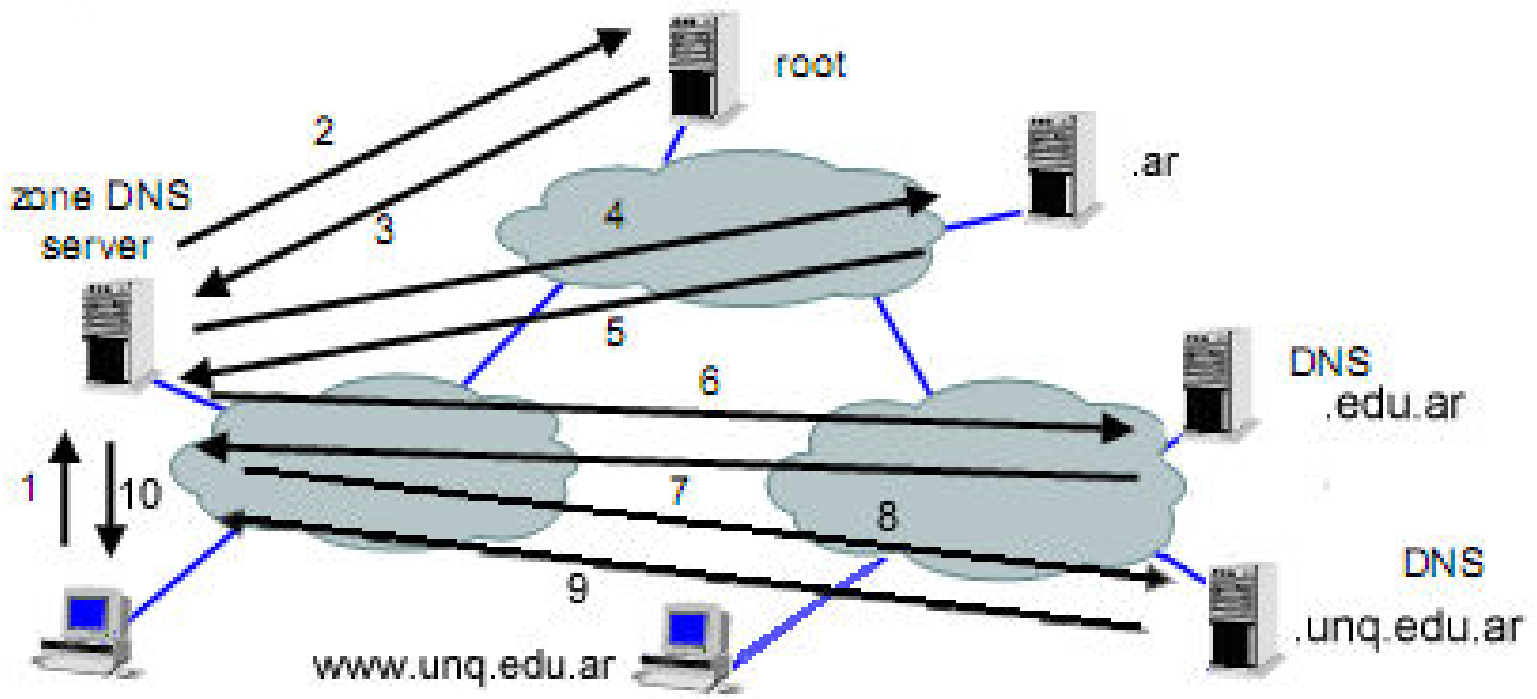
Ejemplo de una "query recursiva"



1. Host pregunta por www.unq.edu.ar a su DNS
2. Zone DNS pregunta al DNS server con dominio .ar
3. DNS con dominio .ar pregunta a DNS con dominio .edu.ar
4. . DNS con dominio . edu.ar pregunta a DNS con dominio .unq.edu.ar
5. DNS .unq.edu.ar le devuelve la IP del servidor www.unq.edu.ar al dominio .edu.ar
6, 7, 8. la IP del servidor ftp.upc.es vuelve al cliente

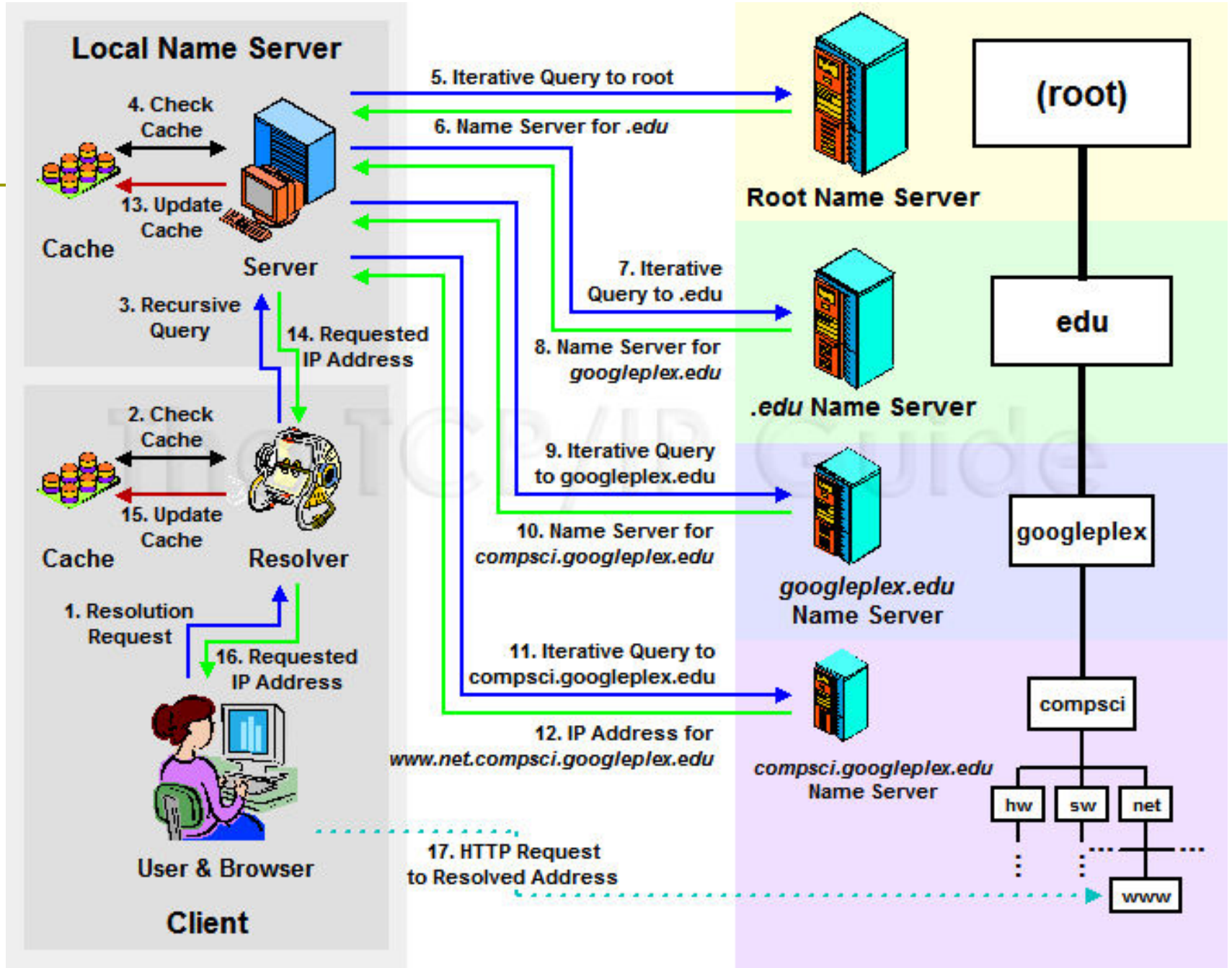
*Cada DNS guarda en su cache las IP un tiempo indicado por TTL

Ejemplo de una “query iterativa”



1. Host pregunta por www.unq.edu.ar a su DNS
2. Zone DNS pregunta a su root DNS
3. Root DNS le devuelve la IP del DNS con dominio .ar al zone DNS
4. Zone DNS pregunta al DNS .ar
5. DNS .es devuelve la IP de DNS .edu.ar
6. Zone DNS pregunta a DNS .edu.ar
7. DNS .edu.ar devuelve IP de DNS unq .edu.ar
8. Zone DNS pregunta a DNS .unq.edu.ar
9. DNS .unq.edu.ar devuelve IP del Server www.unq.edu.ar
10 . Zone DNS devuelve la IP del server al host

*Cada DNS guarda en su cache las IP un tiempo indicado por TTL

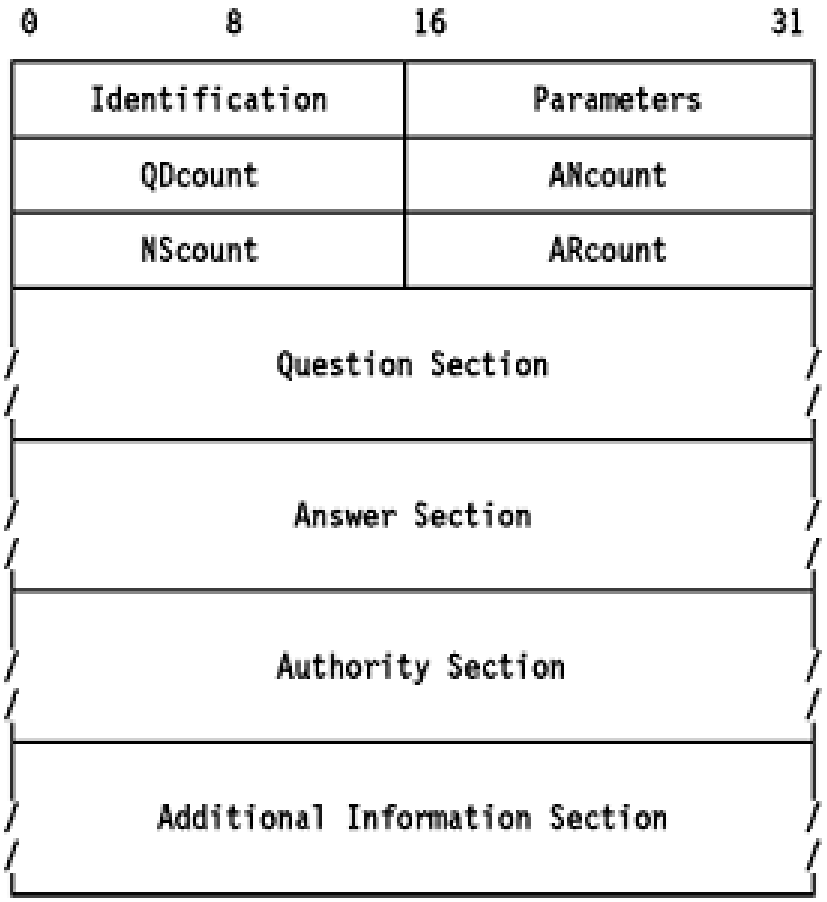

Local Name Server
4. Check Cache
13. Update Cache
Cache
Server
3. Recursive Query
14. Requested IP Address
2. Check Cache
15. Update Cache
Cache
Resolver
1. Resolution Request
16. Requested IP Address
User & Browser
Client
5. Iterative Query to root
6. Name Server for .edu
7. Iterative Query to .edu
8. Name Server for googleplex.edu
9. Iterative Query to googleplex.edu
10. Name Server for compsci.googleplex.edu
11. Iterative Query to compsci.googleplex.edu
12. IP Address for www.net.compsci.googleplex.edu
17. HTTP Request to Resolved Address
Root Name Server
.edu Name Server
googleplex.edu Name Server
compsci.googleplex.edu Name Server
(root)
edu
googleplex
compsci
hw
sw
net
www

Formato del mensaje DNS

Dns usa el mismo formato para:

- Todos los tipos de consulta de clientes y respuestas del servidor.
- Mensajes de error.
- La transferencia de información de registros de recursos entre servidores.

Campos del mensaje DNS

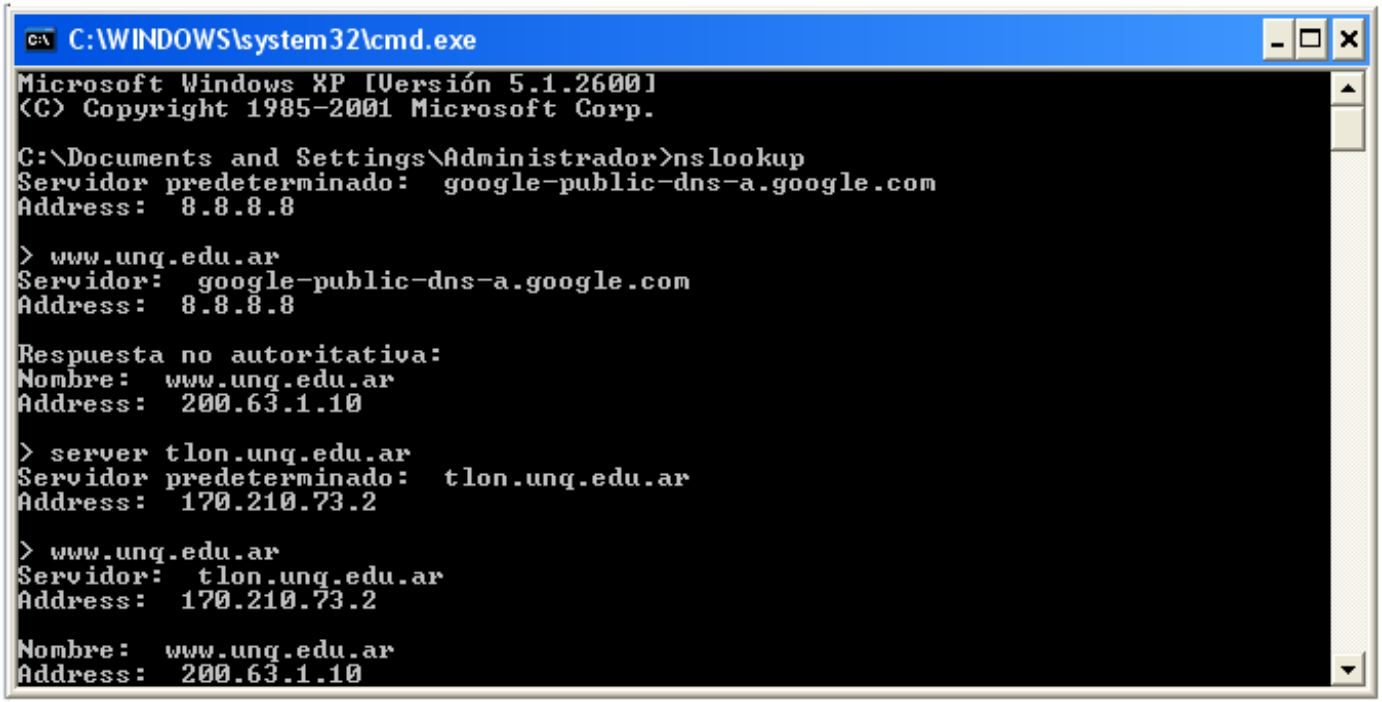


- Identification: permite mapear respuestas con peticiones (activado por el cliente y retornado por el servidor)
- 16-bit flags: dividido en multiples campos (e.g. 1-bit QR: Queries=0, Responses=1; 1-bit RD for iterative/recursive queries, etc)
- Los siguientes 4 campos indican cuantas peticiones y respuestas hay en el resto del mensaje
- Question contiene las consultas al servidor de nombres.
- Answer, Autority, Additional Information contienen un número variable de recursos como TTL, etc...

Comando “nslookup”

Permite averiguar un nombre o una dirección IP

nslookup [-option ...] [host-to-find | -[server]]



Registros DNS

Registros SOA: (Start Of Authority) Se forma con una serie de parametros a tener en cuenta.

Host Origen: Host donde se mantiene el archivo.

Correo electrónico: Del responsable de la BD. La arroba (@) se sustituye por un punto (.), debido a que @ representa el dominio raiz de la zona.

Numero de serie: La versión de ese archivo. Aumenta cada vez que el archivo cambia.

Tiempo de actualización: Tiempo que espera un servidor de nombres secundario para ver si el archivo ha cambiado, y por lo tanto pedir una **transferencia de zona**.

Tiempo de reintento: Tiempo que espera un servidor de nombres secundario para iniciar una nueva transferencia de zona en el caso de que falle este procedimiento.

Tiempo de caducidad: Tiempo que el servidor de nombres secundario intentará descargar una zona. Cuando pase, se rechaza la información antigua.

Tiempo de vida: Tiempo en el que el servidor de nombres mantiene la caché cualquier registro del recurso de este archivo en base de datos.

Registros DNS

- **Registros NS:** El Registro NS. (siglas de **Name Server**), contiene los servidores de nombre de ese dominio, lo que permite que otros servidores de nombres vean los nombres de su dominio.
- **Registros A:** Los registros de dirección A, (**Adress**) asocian nombres de host a direcciones IP dentro de una zona. Son los más numerosos dentro del archivo.
- **Registros CNAME:** Estos registros son llamados también **alias**, si bien son conocidos como entradas de *nombre canónico* (**CNAME, Canonical Name**). Su uso más común es utilizar para apuntar a un único host más de un nombre.
- **Registros MX:** El registro MX es el registro de Intercambio de correo (**Mail eXchange**). Indica que host se encarga del procesamiento del correo electrónico de ese dominio.

Resolución inversas

- Conozco la IP y quiero el nombre
- Cada servidor gestiona una rama que comienza con la etiqueta "in-addr" de la que cuelgan las direcciones en sentido numérico inverso.
- Es decir, las direcciones cuelgan del árbol en orden descendiente:

  e.g.; la IP 200.63.1.10 estaría como 1.63.200.in-addr.arpa (Registro PRT)

Gestión en la asignación de Dominios

- Historia: Hasta hace poco tiempo lo gestionaba IANA (Internet Assigned Numbers Authority: http://www.iana.org) - organización del gobierno USA
- Ahora es privado (aunque con una organización sin ánimo de lucro) que recogio las actividades del IANA. Esta organización se llama ICANN (Internet Corporation for Assigned Names and Numbers: http://www.icann.org) y que se encarga de dar tanto dominios genéricos como IP
- Los dominios genéricos son registrados por compañías a las que ICANN da el derecho a que actúen como tales bajo ciertas restricciones (Accredited Registrars)

Gestión en la asignación de Dominios

El dominio territorial de Argentina (.ar) los asigna el Ministerio de Relaciones Exteriores, Comercio Internacional y Culto (http://www.nic.ar)

El registro no tiene costo y es necesario renovarlo anualmente.
El dominio edu.ar los administra el consorcio ARIU (www.riu.edu.ar)

Tenemos nuestro IP

- Ya con el soporte del servicio DNS tenemos el IP de destino.
- Ahora **¿Qué es HTTP?**

¿Qué es HTTP?

- HTTP significa **Hypertext Transfer Protocol.**
- HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999
- HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web para comunicarse.

  (Clientes, Servidores, Proxys)

Elementos de una comunicación HTTP

- Cliente que efectúa la petición "user agent" (navegador)
- Servidor http el cual posee el puerto 80 tcp en escucha de una petición.
- La información transmitida es el recurso identificada por un URL
- Está basado en el envío de comandos y respuestas en texto ASCII.
- El contenido enviado (archivos HTML, imágenes, PDF, etc.) se enviará tal cual está almacenado en el servidor.(En formato MIME)
- HTTP es un protocolo sin estado, no guarda ninguna información sobre conexiones anteriores.
- Si fuese necesario mantener estado se utilizan las cookies.
- Las Cookies es información que un servidor puede almacenar en el sistema cliente por tiempo indeterminado.
- Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios.

Los estándares HTTP

- HTTP/0.9 Pre-93
- RFC 1945 (HTTP/1.0, 1996)
- RFC 2616 (HTTP/1.1, 1999)
- RFC 2774 (HTTP/1.2, 2000)

*www.**rfc-editor**.org (Request for Comments)*

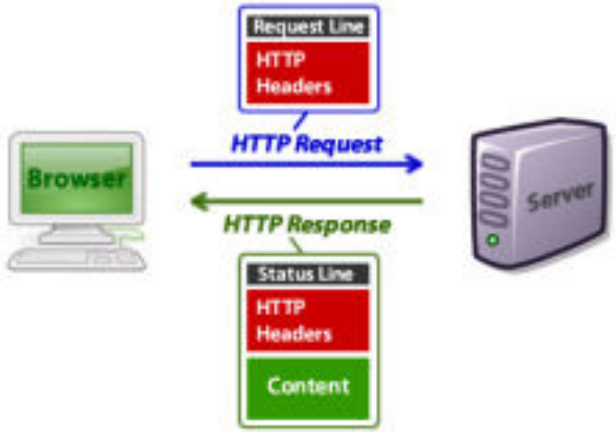
Características de las versiones

HTTP/0.9: Primer versión del protocolo Http, cuya implementación fue realizada con requerimientos muy simples. Solo acepta el método GET
HTTP/1.0: Esta es la primera revisión del protocolo que especifica su versión en las comunicaciones, y todavía se usa ampliamente, sobre todo en servidores proxy. Soporta la mayoría de los métodos de comunicación. Esta como la primer versión por cada elemento que trae de la web realiza una conexión tdc/ip
HTTP/1.1: Versión actual; las conexiones persistentes están activadas por defecto y funcionan bien con los proxies. También permite al cliente enviar múltiples peticiones a la vez (pipelining) lo que hace posible eliminar el tiempo de Round-Trip delay por cada petición.Soporta **name-based virtual hosting**.
HTTP/1.2: El RFC 2774 (experimental), *HTTP Extension Framework*, incluye en gran medida a PEP(Protocol extension protocol). Se publicó en febrero de 2000.

Diálogo HTTP

- Como mencionamos, este protocolo se basa en envíos y respuestas en ASCII.
- Para realizar esto establece una conexión TCP al puerto 80 y una vez establecida comienza el dialogo.
- Entonces podemos mediante una consola terminal de telnet establecer manualmente un dialogo HTTP

Diálogo HTTP



- Iniciamos la conexión, Request Line/Status Line

  (telnet www.unq.edu.ar 80)

  Envío de HTTP Request Header

  (GET / HTTP/1.1)

  Recibimos
  - HTTP Response Headers (HTTP/1.1 200 OK)
  - HTTP Response Body (Html, jpg, objeto requerido)

Solicitud HTTP

- **Una línea de solicitud**: es una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio:
  - el método
  - la dirección URL
  - la versión del protocolo utilizada por el cliente (por lo general, *HTTP/1.0*) Ejemplo: *GET / HTTP/1.1*
- **Los campos del encabezado de solicitud**: es un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

  User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.2b) Gecko/20021016 Accept-Language: es-es, en-us;q=0.66, en;q=0.33

- **El cuerpo de la solicitud**: es un conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

Metodos y Comandos

GET	Solicita el recurso ubicado en la URL especificada
HEAD	Solicita el encabezado del recurso ubicado en la URL especificada
POST	Envía datos al programa ubicado en la URL especificada
PUT	Envía datos a la URL especificada
DELETE	Borra el recurso ubicado en la URL especificada

Accept	Tipo de contenido aceptado por el navegador (por ejemplo, texto/html). Consulte Tipos de MIME
Accept-Charset	Juego de caracteres que el navegador espera
Accept-Encoding	Codificación de datos que el navegador acepta
Accept-Language	Idioma que el navegador espera (de forma predeterminada, inglés)
Authorization	Identificación del navegador en el servidor
Content-Encoding	Tipo de codificación para el cuerpo de la solicitud
Content-Language	Tipo de idioma en el cuerpo de la solicitud
Content-Length	Extensión del cuerpo de la solicitud
Content-Type	Tipo de contenido del cuerpo de la solicitud (por ejemplo, texto/html). Consulte Tipos de MIME
Date	Fecha en que comienza la transferencia de datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
From	Permite especificar la dirección de correo electrónico del cliente
From	Permite especificar que debe enviarse el documento si ha sido modificado desde una fecha en particular
Link	Vínculo entre dos direcciones URL
Orig-URL	Dirección URL donde se originó la solicitud
Referer	Dirección URL desde la cual se realizó la solicitud
User-Agent	Cadena con información sobre el cliente, por ejemplo, el nombre y la versión del navegador y el sistema operativo

Respuesta HTTP

- **Una línea de estado**: es una línea que especifica la versión del protocolo utilizada y el estado de la solicitud en proceso mediante un texto explicativo y un código. La línea está compuesta por tres elementos que deben estar separados por un espacio: La línea está formada por tres elementos que deben estar separados por un espacio:
  - la versión del protocolo utilizada
  - el código de estado
  - el significado del código

  HTTP/1.1 200 OK

- **Los campos del encabezado de respuesta**: es un conjunto de líneas opcionales que permiten aportar información adicional sobre la respuesta y/o el servidor. Cada una de estas líneas está compuesta por un nombre que califica el tipo de encabezado, seguido por dos puntos (:) y por el valor del encabezado Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

  Date: Fri, 31 Dec 2003 23:59:59 GMT
  Server: Apache/2.2.9 (Debian) Last-Modified: Wed, 10 Dec 2008 14:36:40 GMT
  ETag: "27956d-5de-45db23022ee00"
  Content-Type: text/html Content-Length: 1221

- **El cuerpo de la respuesta**: contiene el documento solicitado. (html, .jpg, .pdf, xml, etc..)

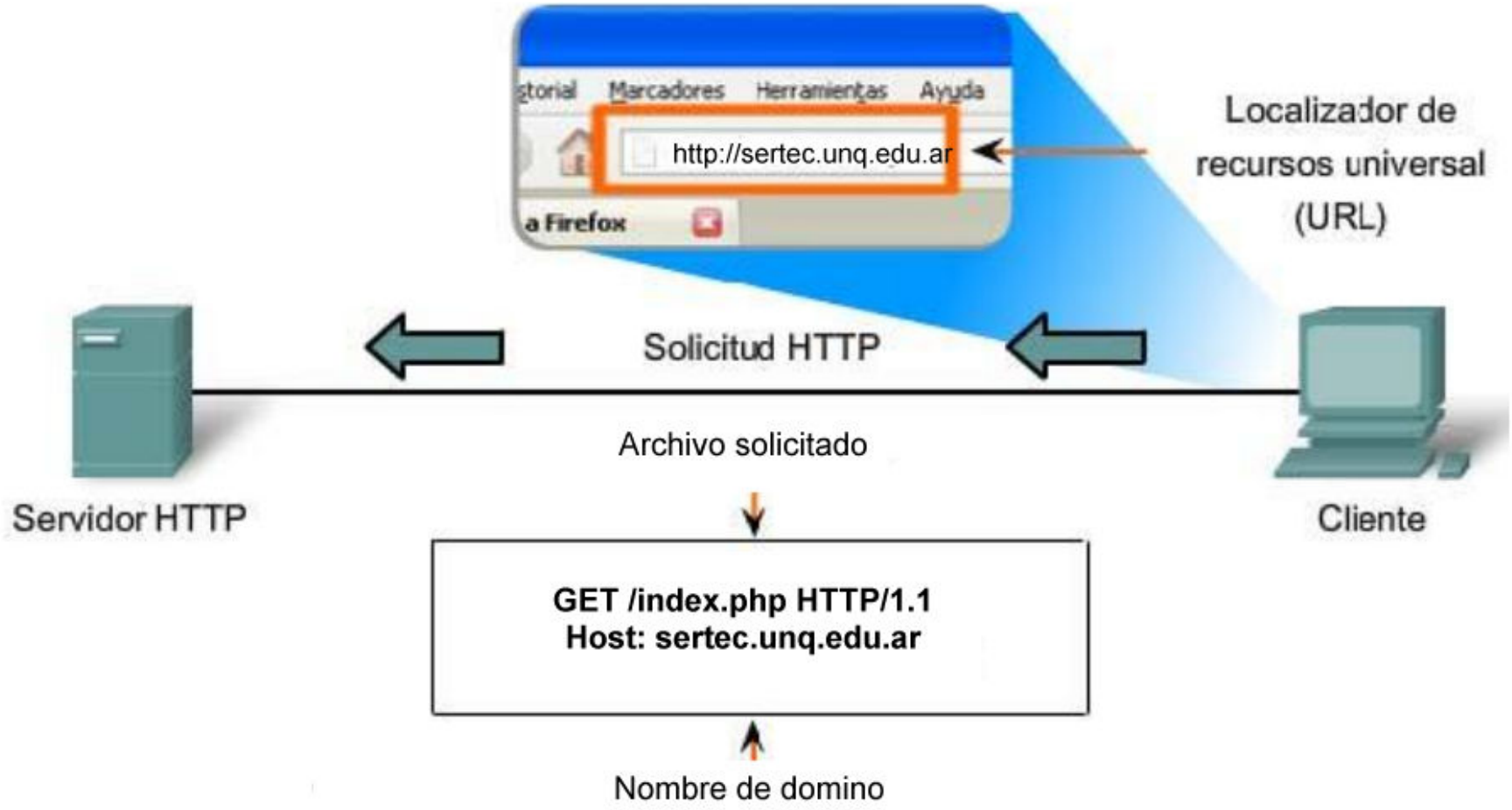
Códigos de respuesta HTTP

- **1xx Mensajes** Conexión rechazada
- **2xx Operación exitosa**
- **3xx Redirección hacia otro URL**
- **4xx Error por parte del cliente**
- **5xx Error por parte del servidor**

Conectandonos a un Server

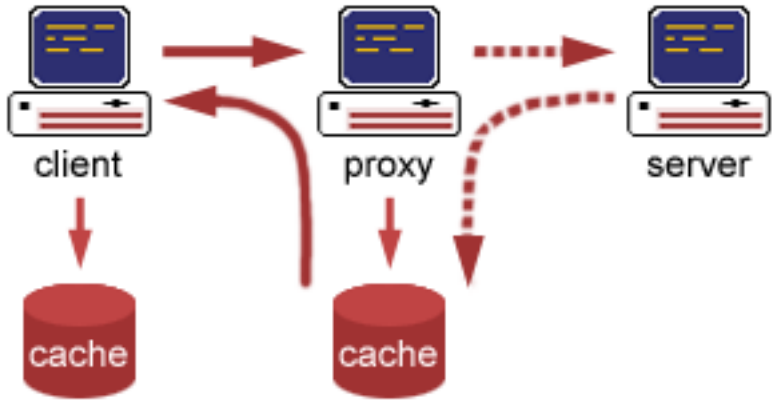
```
GET /index.php HTTP/1.1
Host: sertec.unq.edu.ar
User-Agent: Mozilla/5.0
^M
HTTP/1.1 302 Found
Date: Mon, 17 May 2010 07:12:08 GMT
Server: Apache
X-Powered-By: PHP/5.2.6-1+lenny8
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: http://sertec.unq.edu.ar/modules/news/
Vary: Accept-Encoding
Content-Type: text/html
Content-Length: 0
Connection: Keep-Alive
Set-Cookie: PHPSESSID=88166dfa5c34a4a5662516a7a7875177; path=/
```

Mediante un Navegador



Proxys Http

Este protocolo admite la posibilidad de la existencia de servidores proxy intermedios entre el cliente y la Web



- **Para el manejo del cache del cliente como del Proxy se utiliza información de la respuesta del servidor http**

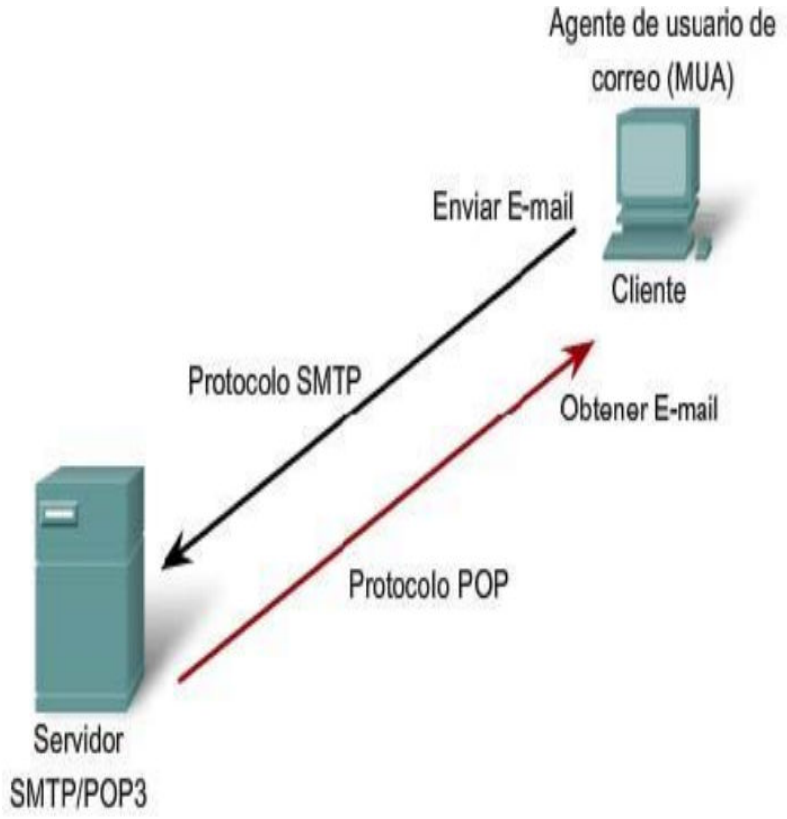
  Last-Modified: Wed, 10 Dec 2008 14:36:40 GMT
  ETag: "27956d-5de-45db23022ee00"

- **El server informa en el Etag el objeto que estamos manejando y la fecha de la ultima modificación, también el server puede informar que esta transacción no deberá ser cacheada (Pragma: no-cache).**

Protocolo SMTP

¿ y ahora como enviamos el correo?

El primer salto



- **El Cliente solo puede conectarse por POP3 y SMTP a su servidor**
- **Y este es el que se encarga de enviar a destino**

Message User Agent (MUA)
(Clientes de correo)

Procesos del servidor de e-mail

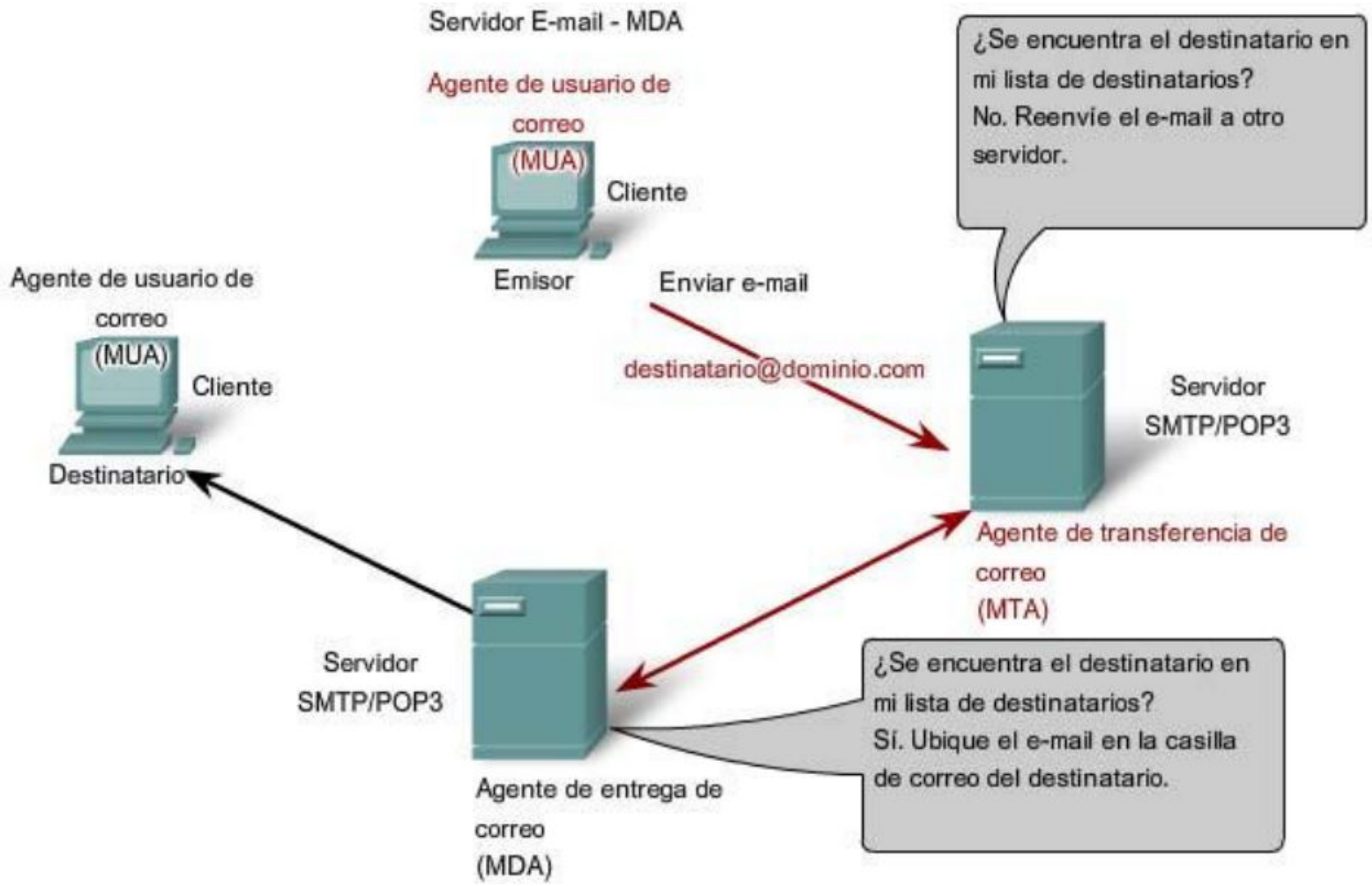
El servidor de e-mail ejecuta dos procesos individuales:

Agente de transferencia de correo (MTA, Mail Transfer Agent).

Agente de entrega de correo (MDA, Mail Delivery Agent).

Agente de Acceso al correo (MAA, Mail Access Agent)

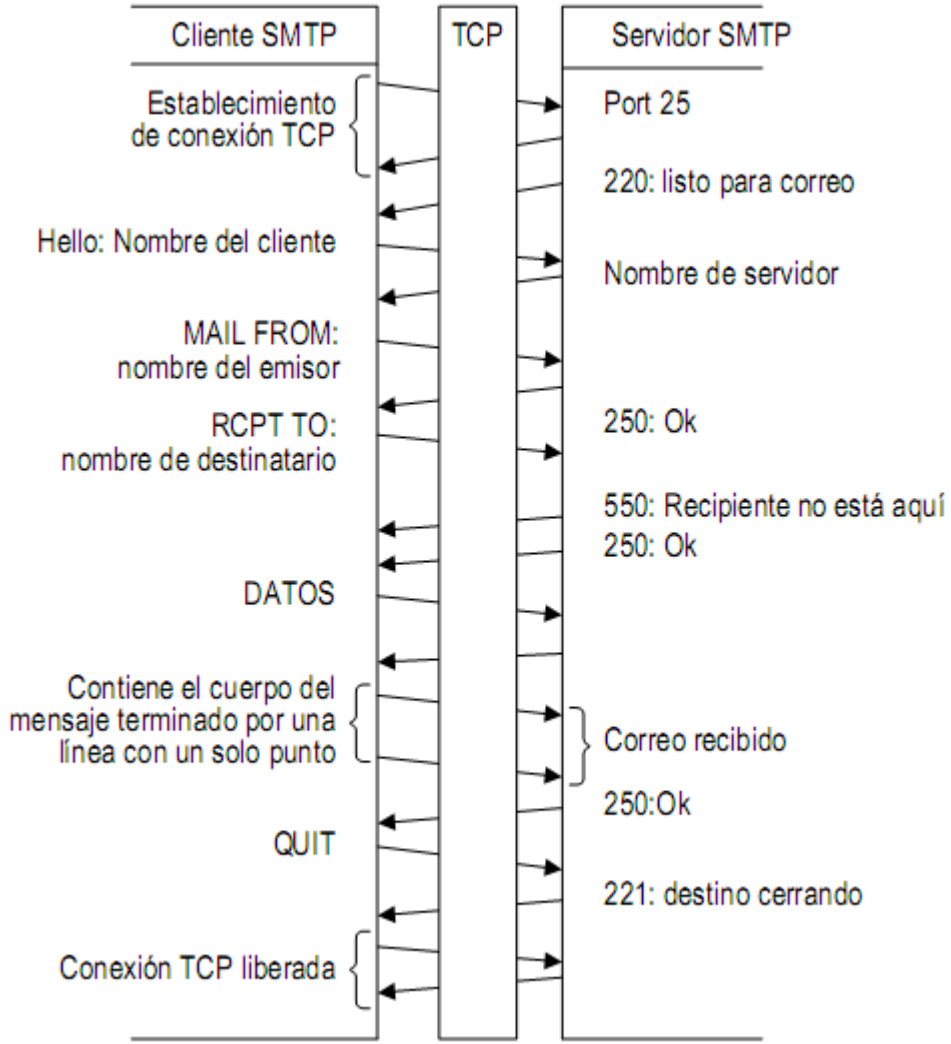
Proceso de entrega de correo



Conexión SMTP

- Este protocolo también hace uso un socket TCP pero en el puerto 25.
- Una vez establecida la conexión también realiza su dialogo en ASCII
- Podemos emular un envío utilizando telnet

Diagrama de conexión SMTP



Conexión SMTP

telnet mail.monitortools.com 25

220 listas.unq.edu.ar ESMTP MDaemon 10.1.1; Mon, 17 May 2010 05:32:32 -0300
helo unq
250 listas.unq.edu.ar Hello 190-49-172-115.speedy.com.ar, pleased to meet you
MAIL FROM: cesar@unq.edu.ar
250 <cesar@unq.edu.ar>, Sender ok
rcpt to: cesar@unq.edu.ar
250 <cesar@unq.edu.ar>, Recipient ok
data
354 Enter mail, end with <CRLF>.<CRLF>
Este es un mensaje de prueba
[ENTER].[ENTER]
250 Ok
quit
221 Bye

Muchas Gracias