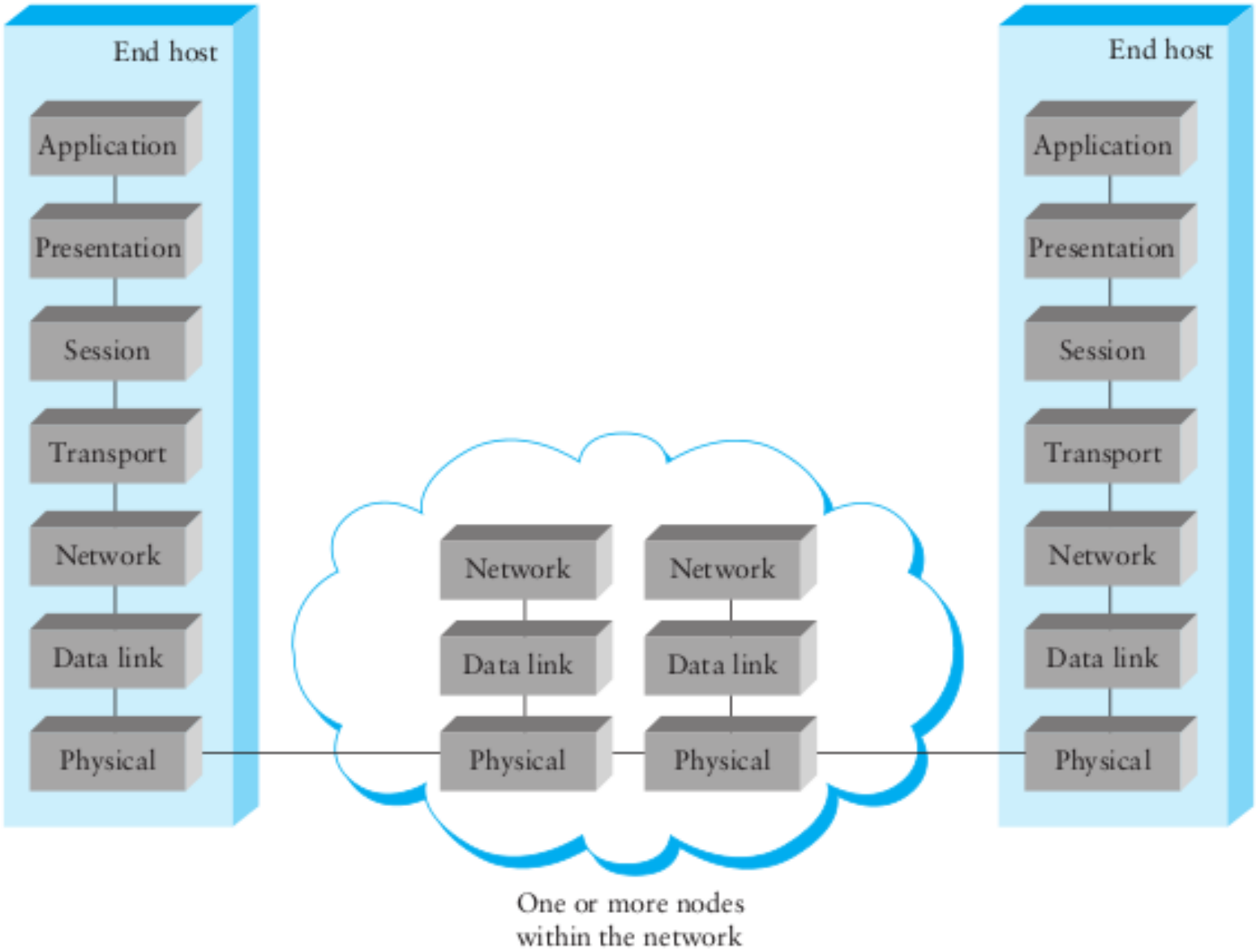
Capa de Aplicación

Pablo Montepagano

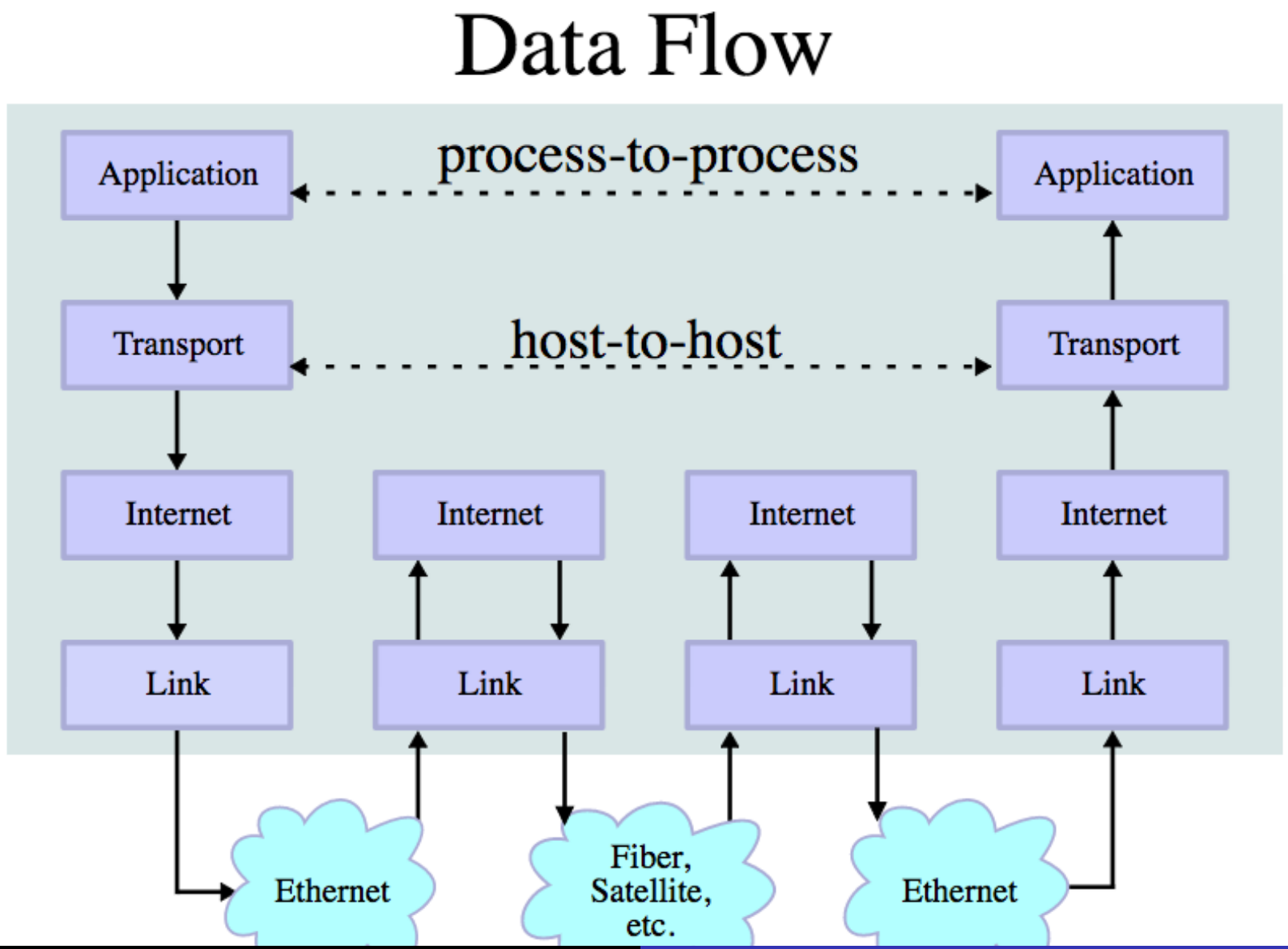
DC - Exactas - UBA

Segundo cuatrimestre 2017

Arquitectura en capas (modelo OSI)



Arquitectura en capas (implementación Internet Protocol)



Objetivos

Específicos de la capa de aplicación

- Aplicaciones son software (Sistemas Operativos, procesos)
- Aspectos conceptuales y abstracciones comunes.
- Tipo de servicio de las capas inferiores.
- Arquitecturas de comunicación.
  - Paradigma cliente - servidor. Ej.: HTTP
  - Paradigma P2P. Ej.: BitTorrent
  - Híbridos.
- Sintaxis y semántica de los mensajes.

Abstracción común más utilizada

Sockets

Requerimientos

- Una aplicación implica diseñar o usar un protocolo existente.
- Para esto es fundamental conocer los diversos protocolos de las diferentes capas para poder elegir el que mejor se atenga a nuestro requerimientos.
- Confiabilidad, seguridad, etc.

Aplicaciones tradicionales

Arquitectura de tipo Request / Reply de mensajes entre cliente y servidor.

Aplicación versus protocolo

Ejemplo: HTTP (Protocolo) ! = Firefox (Aplicación).

DNS: ¿qué hace y para qué sirve?

Dicho mal y pronto: traduce nombres a IPs. Pero también sirve para más cosas.

- Es una base de datos jerárquica y descentralizada para nombrar hosts, servicios y todo tipo de recursos conectados a la Internet o redes privadas.
- Función principal: dar la IP para un hostname. Ej.: `exactas.uba.ar` hoy tiene la IP 157.92.32.35
- El servidor puede cambiar de dirección IP y lo seguimos referenciando con el mismo nombre.
- Es una herramienta fundamental para la Internet.

DNS

Datasheet

* Puerto 53
* Corre sobre UDP (aunque también sobre TCP si el mensaje es grande)
* `RFC 1034 - Concepts and Facilities`
* `RFC 1035 - Implementation and Specification`
* No hay mucho en el Peterson.

¿Qué se usaba antes de DNS?

HOSTS.TXT (hoy: /etc/hosts)

Diccionario (dominio,IP). Hasta 1983 era actualizado A MANO por una persona. Había que llamar por teléfono y pedir una modificación. Ese archivo era descargado por FTP desde cada máquina conectada a internet. Después de cierto punto, ya se le saturaba el enlace al servidor master...

Hasta que en 1983...

DNS (Domain Name Server)

Establece una relación entre Nombres de Dominio y objetos de la red. El "namespace"de dominios se divide de manera tal que cada organización pueda administrar su propio mappeo. Además, cada organización tiene su propio servidor (no está todo centralizado en un solo lugar).

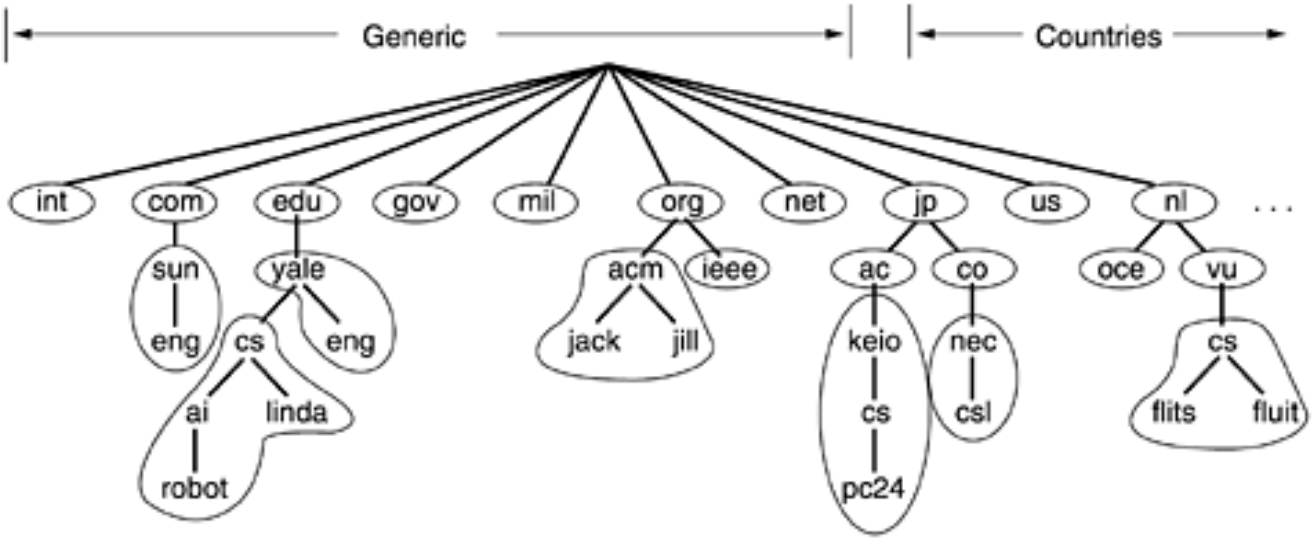
Decisiones de Implementación

- **Arquitectura cliente-servidor**
- **Distribuido**: Cada zona tiene su(s) servidor(es)
- **Jerarquía de servidores de nombres**: Un administrador de zona puede delegar una sub-zona para que otro administre esa porción.

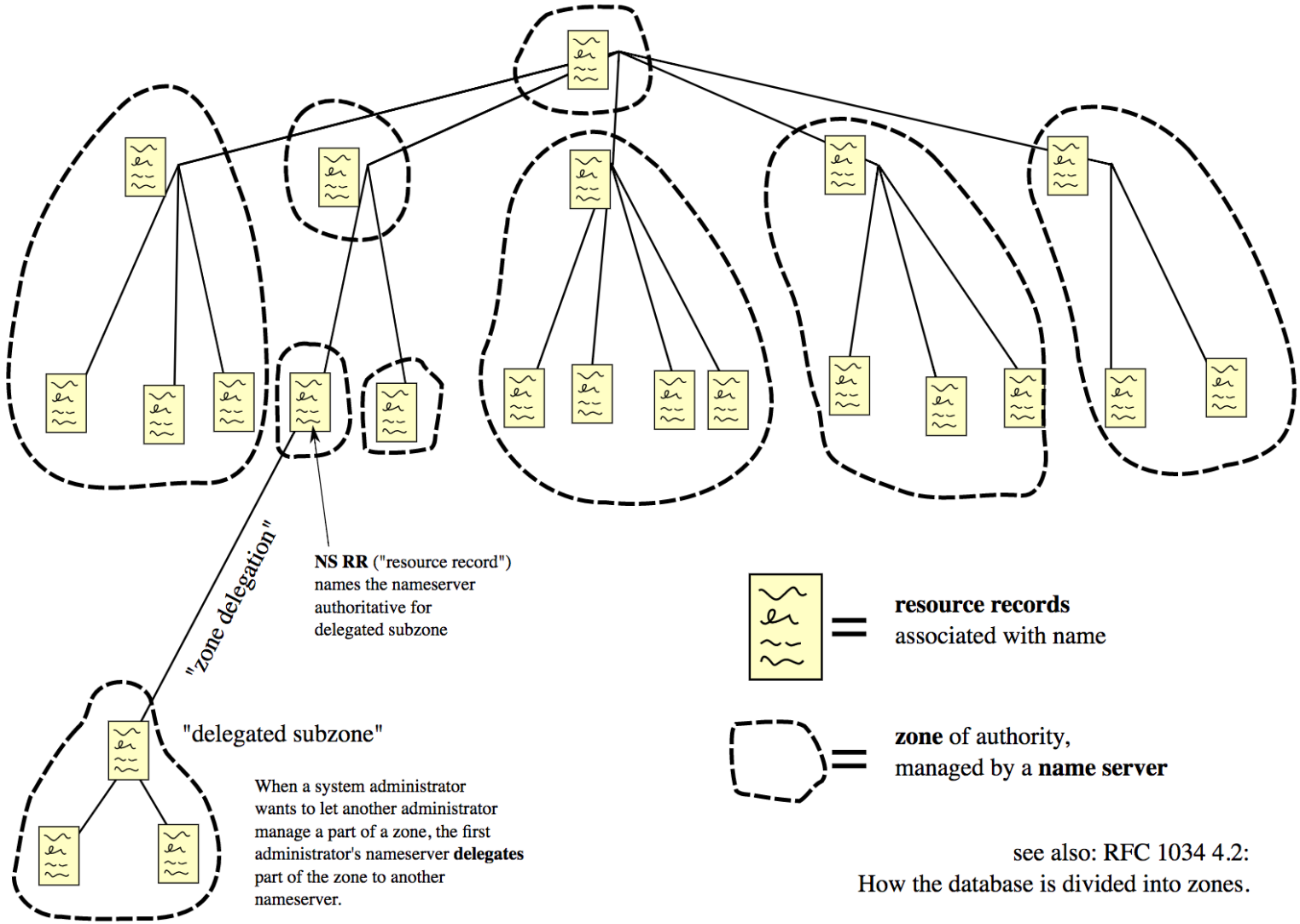
Zonas

La jerarquía de dominios se separa en zonas:

- Agrupan varios dominios.
- No se solapan.
- Cada zona tiene un name server responsable (un name server puede administrar varias zonas).
- Cada zona se administra por su cuenta.



Zonas


"zone delegation"
NS RR ("resource record") names the nameserver authoritative for delegated subzone
"delegated subzone"
When a system administrator wants to let another administrator manage a part of a zone, the first administrator's nameserver delegates part of the zone to another nameserver.
resource records associated with name
zone of authority, managed by a name server
see also: RFC 1034 4.2: How the database is divided into zones.

Nombres de dominios

- Fully Qualified Domain Name (FQDN) vs Non-FQDN
- Siguen jerarquías.
- FQDN
  - Formato. Hostname + Domain
  - Ejemplo de FQDN: mail.dc.uba.ar.
- Non-FQDN
  - Formato. Hostname
  - Se utiliza mucho para redes locales
  - Ejemplo de Non-FQDN: mail

Registros DNS: ¿Que son?

Registros DNS: Mantenidos por los administradores. Entre todos forman la base de datos DNS.
Está formado por los siguientes campos:

- nombre de dominio
- TTL
- clase (en general es IN de internet)
- tipo
- valor

Tipos de registros

Resource records [edit]

Type	Type id. (decimal)	Defining RFC	Description	Function
A	1	RFC 1035[1]	Address record	Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host, but it is also used for DNSBLs, storing subnet masks in RFC 1101, etc.
AAAA	28	RFC 3596[2]	IPv6 address record	Returns a 128-bit IPv6 address, most commonly used to map hostnames to an IP address of the host.
CNAME	5	RFC 1035[1]	Canonical name record	Alias of one name to another: the DNS lookup will continue by retrying the lookup with the new name.
DNSKEY	48	RFC 4034	DNS Key record	The key record used in DNSSEC. Uses the same format as the KEY record.
MX	15	RFC 1035[1] and RFC 7505	Mail exchange record	Maps a domain name to a list of message transfer agents for that domain
NS	2	RFC 1035[1]	Name server record	Delegates a DNS zone to use the given authoritative name servers
PTR	12	RFC 1035[1]	Pointer record	Pointer to a canonical name. Unlike a CNAME, DNS processing stops and just the name is returned. The most common use is for implementing reverse DNS lookups, but other uses include such things as DNS-SD.
SOA	6	RFC 1035[1] and RFC 2308[9]	Start of [a zone of] authority record	Specifies *authoritative* information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.
SRV	33	RFC 2782	Service locator	Generalized service location record, used for newer protocols instead of creating protocol-specific records such as MX.
TXT	16	RFC 1035[1]	Text record	Originally for arbitrary human-readable *text* in a DNS record. Since the early 1990s, however, this record more often carries machine-readable data, such as specified by RFC 1464, opportunistic encryption, Sender Policy Framework, DKIM, DMARC, DNS-SD, etc.

Ejemplos de registros

nombre de dominio	TTL	clase	tipo	valor
www.dc.uba.ar.	600	IN	CNAME	www-1.dc.uba.ar.
www-1.dc.uba.ar.	600	IN	CNAME	dc.uba.ar.
dc.uba.ar.	600	IN	A	157.92.27.127
dc.uba.ar.	600	IN	NS	ns1.uba.ar.
dc.uba.ar.	600	IN	NS	ns-1.dc.uba.ar.
dc.uba.ar.	600	IN	NS	ns-2.dc.uba.ar.
dc.uba.ar.	600	IN	NS	ns2.uba.ar.
ns1.uba.ar.	7200	IN	A	157.92.1.1
ns1.uba.ar.	7200	IN	AAAA	2001:1318:100c:1::1
ns2.uba.ar.	7200	IN	A	157.92.4.1
ns2.uba.ar.	7200	IN	AAAA	2001:1318:100c:4::1
ns-2.dc.uba.ar.	600	IN	A	157.92.27.253
dc.uba.ar.	600	IN	MX	10 mta1.exactas.uba.ar.
dc.uba.ar.	600	IN	MX	5 mta0.exactas.uba.ar.

Servidores master y slaves

Puede haber varios servidores slaves para back-up y balance de carga. Estos periódicamente actualizan la información del servidor master.

El registro SOA indica el nombre del servidor master y brinda información sobre los updates.

- Serial: Sirve para versionar los registros de la zona.
- Refresh: Determina cuán seguido se actualiza la zona desde el servidor primario.
- Retry: Determina cuánto tiempo esperar para volver intentar en caso que falle el pedido del SOA al primario.
- Expire: Indica cuánto tarda en expirar la zona en caso de no obtener respuesta del primario.
- TTL: Tiempo máximo en que un resolver puede cachear una respuesta negativa. RFC 2308

Registros DNS: Ejemplo de SOA

```
dc.uba.ar IN SOA ns-2.dc.uba.ar. admin.dc.uba.ar. (
     2017102000 ; serial
     3600       ; refresh (1 hour)
     900        ; retry (15 minutes)
     700000     ; expire (1 week 1 day 2 hours 26 minutes 40 seconds)
     600        ; minimum (10 minutes)
     )
```

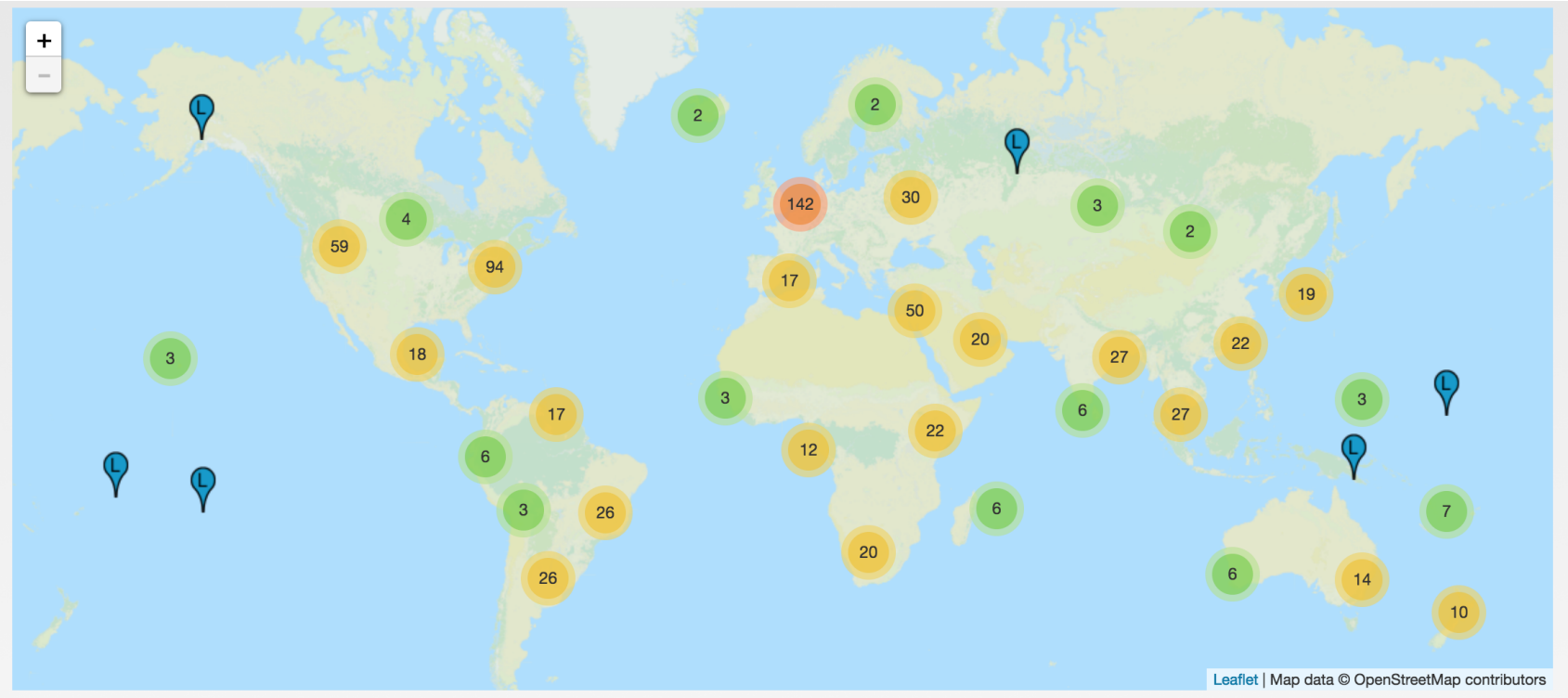
- ns-2.dc.uba.ar : servidor DNS primario
- admin.dc.uba.ar : mail de contacto con lxs responsables (usa . en vez de @)
- 2017102000 es el serial. Hay que incrementarlo cada vez que se modifica la zona, si no, no se transfieren los cambios de la zona. (hay wraparound en 32 bits, ver RFC 1982)
- El resto establece parámetros de configuración de tiempo para el traspaso de los archivos entre distintos servidores DNS.

Registros DNS: Ejemplos de MX

```
IN      MX      5 mta0.fcen.uba.ar.
IN      MX      10 mta1.fcen.uba.ar.
```

- cuando mandamos mail a qwerty@dc.uba.ar, el servidor final al que se debería mandar el mail es uno de los mencionados arriba.
- Notar que tienen un número antes del nombre, indicando prioridades. A menor el número, es mayor la prioridad.
- Por eso primero se intentaría enviar el mail a mta0.fcen.uba.ar, caso contrario a mta1.fcen.uba.ar

Root zone



http://www.root-servers.org/
https://www.internic.net/zones/named.root

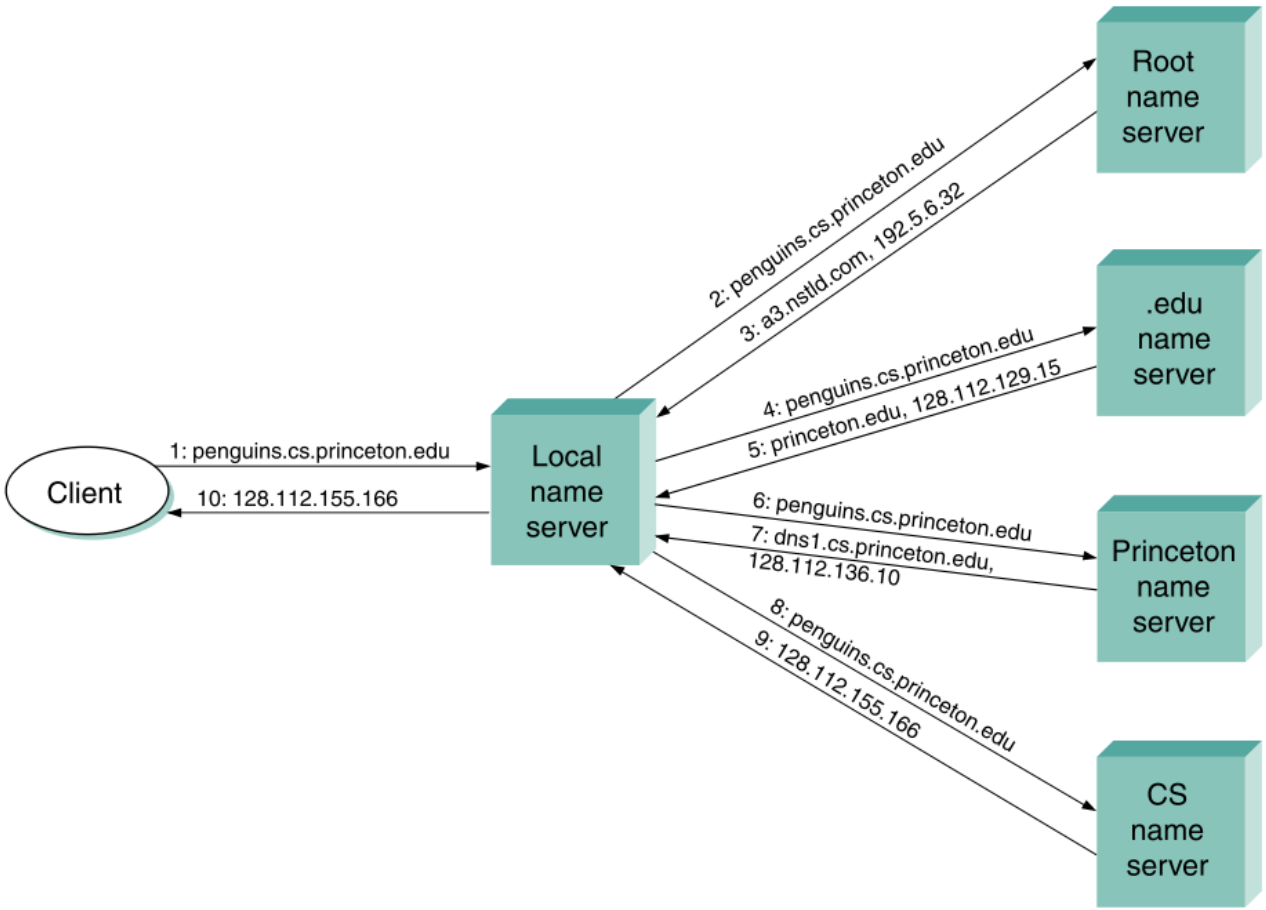
Roles en DNS

- **Servidores DNS (master y slaves):** Servidores Autoritativos que tienen los datos originales.
- **Resolvers (servidores no autoritativos):** Dan respuesta a consultas (habitualmente en redes locales).
- **Stub resolvers (clientes DNS):** Es el componente que hace consultas a los resolvers (normalmente implementado por el Sistema Operativo de un host). (ver /etc/resolv.conf en Linux)

La implementación más usada (BIND) puede cumplir los dos primeros roles, y por eso a veces algunos las confunden.

Consultas: Iterativa

Iterativa: cuando un servidor no tiene la dirección responde la dirección del NS que sigue y es responsabilidad del origen hacer la consulta siguiente. Con este metodo se puede cachear los NS que se acceden en el camino.



Consultas: Recursiva

Recursiva: cuando un servidor no tiene la dirección pedida delega la consulta al próximo NS en el camino y cuando recibe la respuesta la envía al origen.

- Los ejemplos antes mostrados son en el peor caso. Es decir, cuando mi resolver local (que podría ser un DNS autoritativo) no conoce el registro buscado ni conoce los servidores autoritativos intermedios.
- Por ejemplo, si mi resolver conocía directmante a los NS de princeton.edu. entonces la iteración empezaría desde ahí y no desde el root.

Respuestas Autoritativas y no Autoritativas

- Si la información viene de un servidor master o slave es autoritativa, es decir, está respaldada por el administrador de la zona. En cambio si es información cacheada no se la considera autoritativa y en caso de que no sea correcta se puede recurrir a un servidor de esa zona para obtener una respuesta confiable
- ¿Qué pasa cuando el servidor de una zona esta dentro de esa misma zona? No se puede acceder porque se genera un loop. Solución: Glue records, además del registro NS se agrega el registro A de ese host para que el servidor pueda devolver la IP del NS en cuestión

Ejercicio

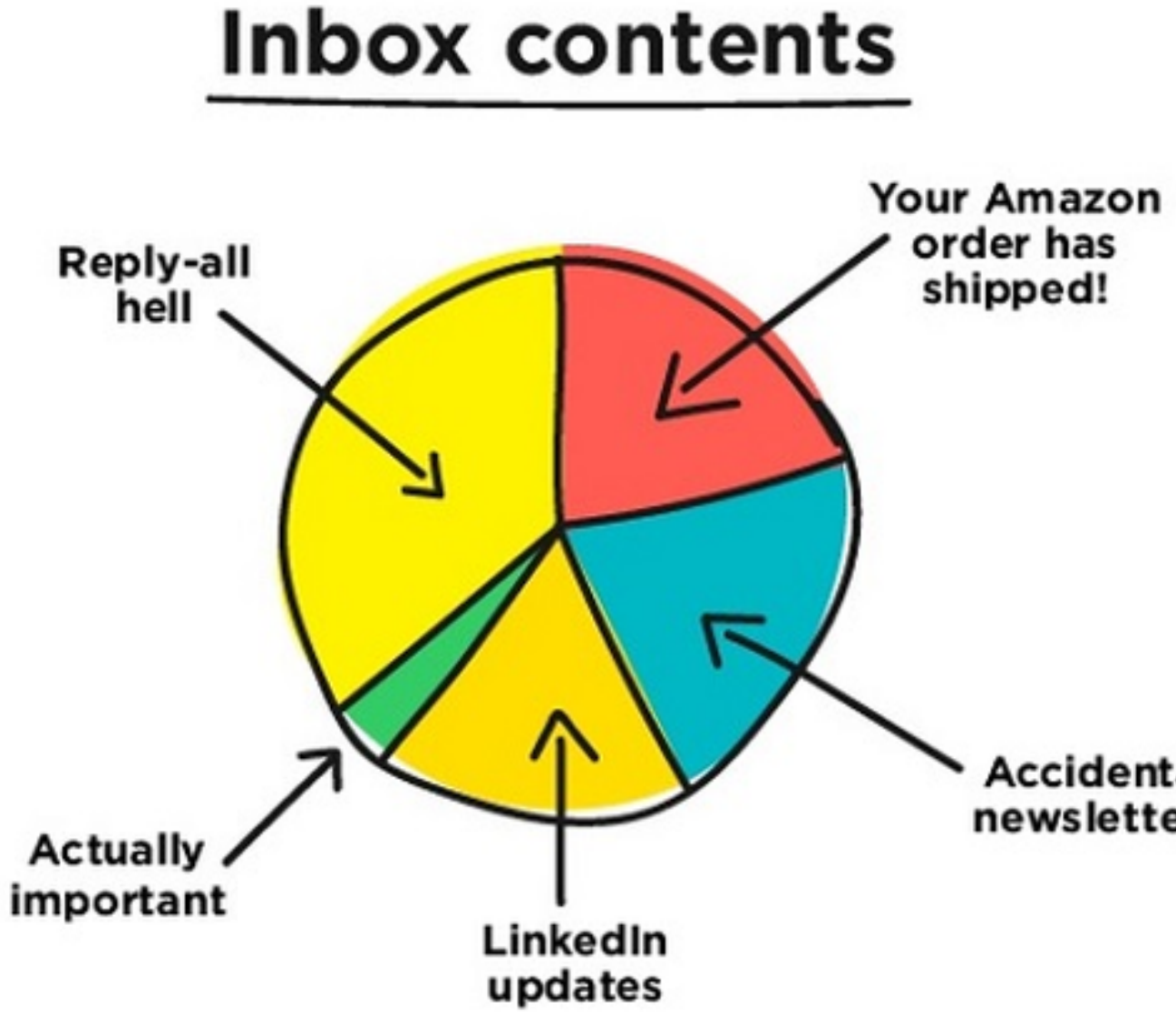
`drill` y `dig` son dos comandos que permiten realizar consultas DNS (son muy similares).

a. Utilíce alguno para consultar registros del tipo `A`, `CNAME` y `MX` para los siguientes dominios:
   - `www.dc.uba.ar`
   - `dc.uba.ar`
   - `uba.ar`
   - `ar`
   - `.`

b. ¿Cómo se reconocen los servidores autoritativos de una determinada zona?

c. ¿Cómo se distinguen los servidores secundarios del servidor primario?

SMTP, POP3, IMAP, Webmail, MIME



Simplicidad del correo

Estrategia

- Los protocolos de correo electrónico son considerablemente simples debido en gran parte a que delegan la interpretación de los datos en otros protocolos o convenciones.
- En los mensajes se transfiere solo texto en formato ASCII.
- La interpretación queda del lado del RFC822 y las extensiones MIME (RFC 2045).

Formato de mensaje estándar

- Header (Comandos Separados por CRLF).
  - From:.
  - To:, Cc:, etc.
- Body (ASCII plano).

Extensiones al formato de mensaje

- ¿Por qué?
  - Quiero poder mandar más que solo texto plano.
- ¿Qué permite?
  - Transferir contenido multimedia sin modificar los protocolos básicos.
- ¿Qué se agrega para poder implementarlo?
  - Nuevos campos en el header (MIME-Version:, Content-Type:).
  - Definiciones de tipos y subtipos (Imagenes, Videos, Documentos).
- ¿Cómo hago que todo sea ASCII?
  - Encodings para codificar los distintos tipos de datos en ASCII (base64).
- ¿Cómo hago para enviar más de un tipo de archivo?
  - Uso delimitadores como tecnica de framing.

Ejemplo de mail

```
MIME-Version: 1.0
Received: by 10.140.93.165 with HTTP; Fri, 20 Oct 2017 12:16:25 -0700 (PDT)
From: Patricio Inzaghi <pa*******@gmail.com>
Date: Fri, 20 Oct 2017 16:16:25 -0300
Message-ID: <CAHjFU62VtJvZi+t6tesmHf=H9M0sm3EfiaMop5invHcBzreSqA@mail.gmail.com
To: tdc-alu <tdc-alu@dc.uba.ar>
Content-Type: multipart/mixed; boundary="001a11407326275e36055bff4f9b"
Subject: [Tdc-alu] Enunciado TP2 y slides de la clase
List-Id: "Teor&#237;a de las Comunicaciones - Alumnos" <tdc-alu.dc.uba.ar>
Sender: "Tdc-alu" <tdc-alu-bounces@dc.uba.ar>

--001a11407326275e36055bff4f9b
Content-Type: text/plain; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

Estimados/as,

Les env=C3=ADo adjuntos tanto el enunciado del TP2 como las slides de la
clase de ICMP.
```

Ejemplo de mail (cont.)

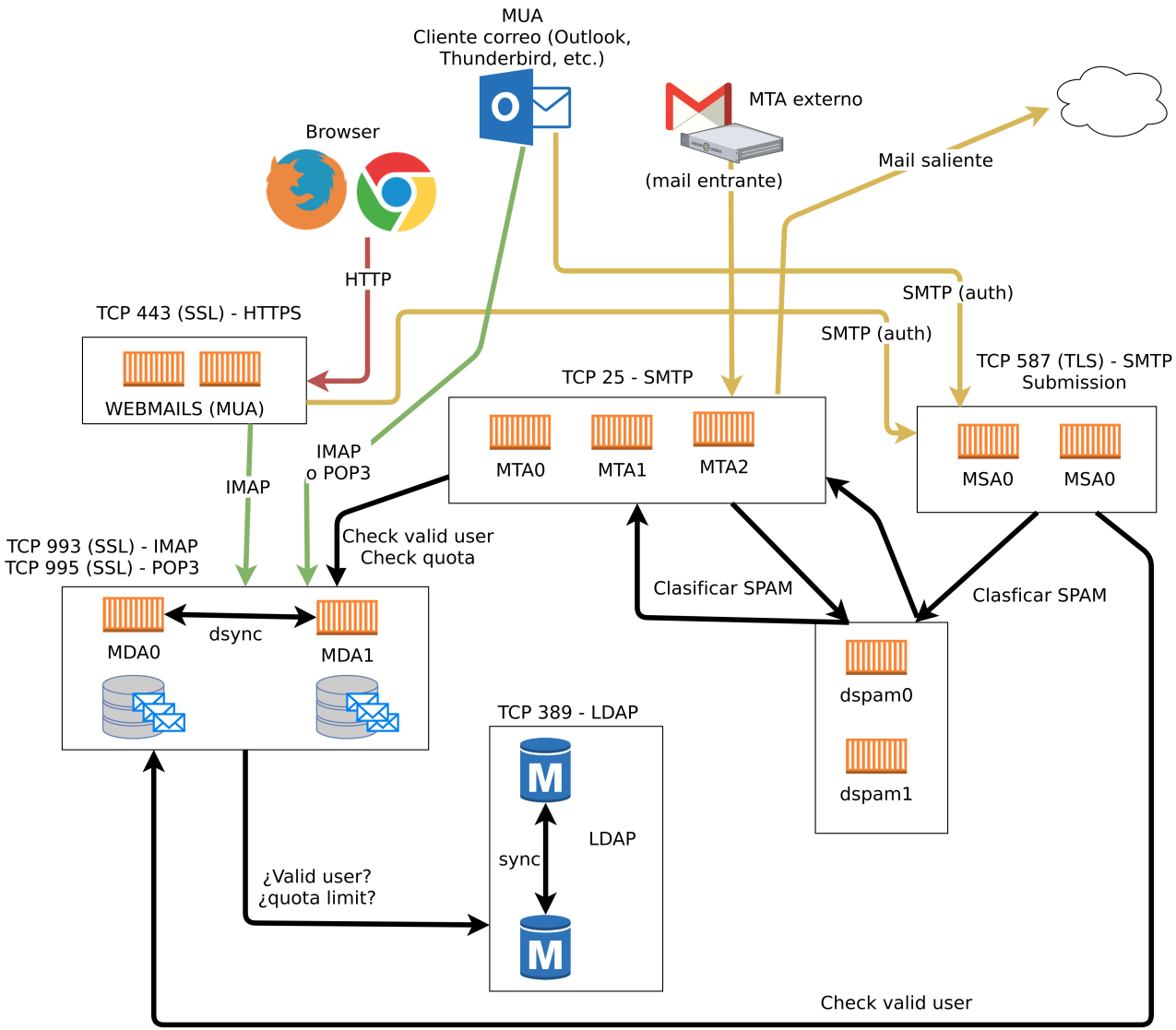
```
En breve estar=C3=A1n subidas a la p=C3=A1gina de la materia.

Saludos.

--001a11407326275e36055bff4f9b
Content-Type: application/pdf; name="enunciado.pdf"
Content-Disposition: attachment; filename="enunciado.pdf"
Content-Transfer-Encoding: base64
X-Attachment-Id: f\_j90aa3pw0

JVBERi0xLjUKJdDUxdgKMyAwIG9iago8PAovTGVuZ3RoIDIyNTUgICAgICAKL0ZpbHRlciAvRmxh
dGVEZWNvZGUKPj4Kc3RyZWFtCnjapVhLk9y2Eb7vr5gjp6KBCIBP32RFSjnlKIozyWWtA4bEaqni
EGOQdHn169ONbnAeosvjSm3VDohHo59fdyPdfN6km789pH/w+/3+4fX7TG9kKZTMs83+aaNgnNX1
psQ5pTb7dvOY7D+q77Y7lafJT/NkRhragX5/GCbrBzttP+3/TtSkqPNcIbVc1Gm+2alKVLVkWtsS
zjqP2+E2kaXFJhVpXtPyz6rIwlIqtKo3u8ulVElDd7aWfvvIzFt3nIeuMU3nBjsurFSiLoog1w5Y
KYCe1KLOKiL4V3syfjJHO0zumi6QO4GkTRdYkakWFVADXrL8hk1gsMyIz7jmBloppcjh8qsr3799
94Gu2NHPf77f6jx5s6a8nRIaLLFTWqi6oPOyEjIVKpUlnyg2MhM6K+iErGpYl5tdpkVelXxGbHcy
TYOl/FZWiWvnhkWrU1GkKJks1a1kUsi6QsmWNZIs8FmDOlURLi0zkRVwp1JCFcznO3AOncnEjpOl
(.... sigue)
```

SMTP

Algunos datos

- La implementación más ~~conocida~~ vieja es sendmail. (80 % en 1996, 5 % hoy). Otras más populares hoy: Microsoft Exchange, Postfix, Exim.
- Nos podríamos conectar directo con el host donde reside la casilla del destinatario y enviarle el correo con el protocolo SMTP.
- En general nos conectamos con nuestro proveedor y luego el correo pasa por varios gateways antes de llegar al destino.
- STORE & FORWARD
- Protocolo tipo PUSH
- Primero formalizado en RFC 821 (1982). Última revisión RFC 5321 (2008).

Arquitectura de mails (ejemplo del mundo real)



Demo SMTP

```
220 mta1.exactas.uba.ar ESMTP Postfix
EHLO pablo.fcen.uba.ar
250-mta1.exactas.uba.ar
250-PIPELINING
250-SIZE 35882577
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM:<pmontepagano@fcen.uba.ar>
250 2.1.0 Ok
RCPT TO:<pmontepagano@fcen.uba.ar>
454 4.7.1 <pmontepagano@fcen.uba.ar>: Relay access denied
RCPT TO:<pmontepagano@dc.uba.ar>
250 2.1.5 Ok
RCPT TO:<pinzaghi@dc.uba.ar>
552 5.2.2 <pinzaghi@dc.uba.ar>: Recipient address rejected: Mailbox is full
RCPT TO:<postmaster@dc.uba.ar>
250 2.1.5 Ok
DATA
```

Demo SMTP (cont)

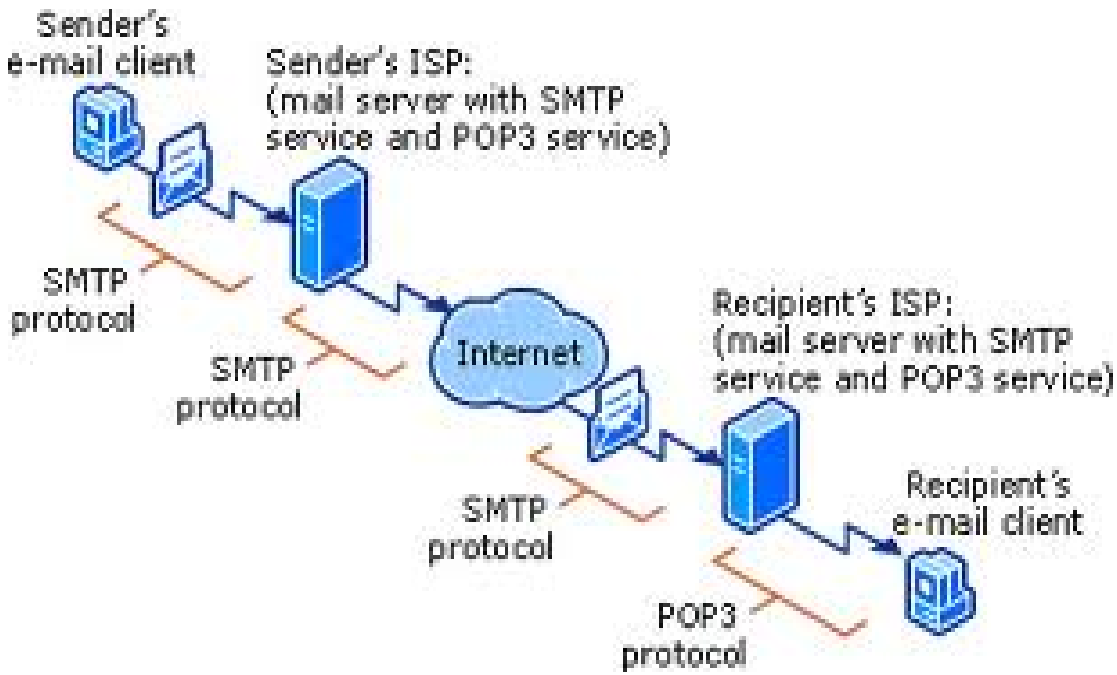
```
354 End data with <CR><LF>.<CR><LF>
From: Pablo Montepagano <pmontepagano@fcen.uba.ar>
To: Pablo DC <pmontepagano@dc.uba.ar>
Cc: Postmaster <postmaster@dc.uba.ar>
Subject: Spoofing

Hola. Esto es una prueba.

.
250 2.0.0 Ok: queued as B73CE3001C7
QUIT
221 2.0.0 Bye
```

POP3

Client oriented

- Se descargan los correos y administran localmente.
- Solo permite acceder a la bandeja de entrada.
- No hace falta estar conectado para acceder a los correos.
- El espacio de alamacenamiento lo define el usuario.
- Contenido local, ¿Más inseguro?.
- Responsabilidad de backup del lado del usuario.
- Sencillo, fácil de usar y configurar.
- Para más información, RFC 1939.

Ejemplo


Sender's
e-mail client
Sender's ISP:
(mail server with SMTP
service and POP3 service)
SMTP
protocol
SMTP
protocol
Internet
Recipient's ISP:
(mail server with SMTP
service and POP3 service)
SMTP
protocol
Recipient's
e-mail client
POP3
protocol

POP3

Mensajes

- LIST.
- STAT.
- DELE n.
- QUIT.
- RETR n.
- Sencillo, fácil de usar y configurar.
- Para más información, RFC 1939.

Ejemplo

```
S: <wait for connection on TCP port 110>
C: <open connection>
S:    +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C:    APOP mrose c4c9334bac560ecc979e58001b3e22fb
S:    +OK mrose's maildrop has 2 messages (320 octets)
C:    STAT
S:    +OK 2 320
C:    LIST
S:    +OK 2 messages (320 octets)
S:    1 120
S:    2 200
S:    .
C:    RETR 1
S:    +OK 120 octets
S:    <the POP3 server sends message 1>
S:    .
C:    DELE 1
S:    +OK message 1 deleted
C:    RETR 2
S:    +OK 200 octets
S:    <the POP3 server sends message 2>
S:    .
C:    DELE 2
S:    +OK message 2 deleted
C:    QUIT
S:    +OK dewey POP3 server signing off (maildrop empty)
C:  <close connection>
S:  <wait for next connection>
```

IMAP

Server oriented

- Es el protocolo de acceso más complejo y potente.
- Sirve cuando se accede a la misma casilla desde distintos hosts.
- Permite administrar carpetas del lado del servidor.
- Permite guardar estado.
- Busqueda del lado del servidor.
- Webmails y celulares lo utilizan.
- ¿Más lento?
- Para más información, RFC 3501.

Ejemplo

The initial `telnet: >` symbolises your shell prompt.

```
telnet: > telnet imap.example.com imap
telnet: Trying 192.0.2.2...
telnet: Connected to imap.example.com.
telnet: Escape character is '^]'.
server: * OK Dovecot ready.
client: a1 LOGIN MyUsername MyPassword
server: a1 OK Logged in.
client: a2 LIST "" "*"
server: * LIST (\HasNoChildren) "." "INBOX"
server: a2 OK List completed.
client: a3 EXAMINE INBOX
server: * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
server: * OK [PERMANENTFLAGS ()] Read-only mailbox.
server: * 1 EXISTS
server: * 1 RECENT
server: * OK [UNSEEN 1] First unseen.
server: * OK [UIDVALIDITY 1257842737] UIDs valid
server: * OK [UIDNEXT 2] Predicted next UID
server: a3 OK [READ-ONLY] Select completed.
client: a4 FETCH 1 BODY[]
server: * 1 FETCH (BODY[] {405}
server: Return-Path: sender@example.com
server: Received: from client.example.com ([192.0.2.1])
server:         by mx1.example.com with ESMTP
server:         id <20040120203404.CCCC18555.mx1.example.com@client.example.com>
server:         for <recipient@example.com>; Tue, 20 Jan 2004 22:34:24 +0200
server: From: sender@example.com
server: Subject: Test message
server: To: recipient@example.com
server: Message-Id: <20040120203404.CCCC18555.mx1.example.com@client.example.com>
server:
server: This is a test message.
server: )
server: a4 OK Fetch completed.
client: a5 LOGOUT
server: * BYE Logging out
server: a5 OK Logout completed.
```

Ejercicio

SMTP — IMAP vs Webmail vs POP3

- ¿Por qué no se puede recibir correo usando SMTP?
- ¿Qué diferencias hay entre el Webmail, POP3 e IMAP, como soluciones para revisar el correo?

Envío de mail

Alice, desde su Webmail le manda un correo a Bob quien lo lee desde su servidor de mail POP3. Cómo y usando qué protocolos de aplicación se transmite el mensaje desde la máquina de Alice hasta la máquina de Bob?

HTTP

Generales

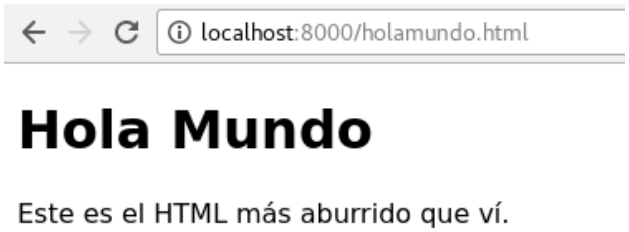
- Usa TCP (puerto 80).
- Un archivo índice con referencia a los diversos objetos.
- Browser y servidor se intercambian mensajes HTTP (HEADER:BODY).
- Cada objeto se referencia por un identificador único (URL).
- HTTP **no** mantiene estado.

Versiones

- HTTP 1.0 es NO persistente.
- HTTP 1.1 lo es.
  - Muchos objetos pueden ser transmitidos por una única conexión TCP.
  - Permite pipelinening
  - Tiene el problema conocido como head-of-line blocking
- HTTP 2.0
  - No introduce cambios grandes en sintaxis (metodos, codigos, headers, etc.)
  - Server push
  - Multiplexación de requests en una misma conexión TCP para mejorar performance.

Ejemplo HTML

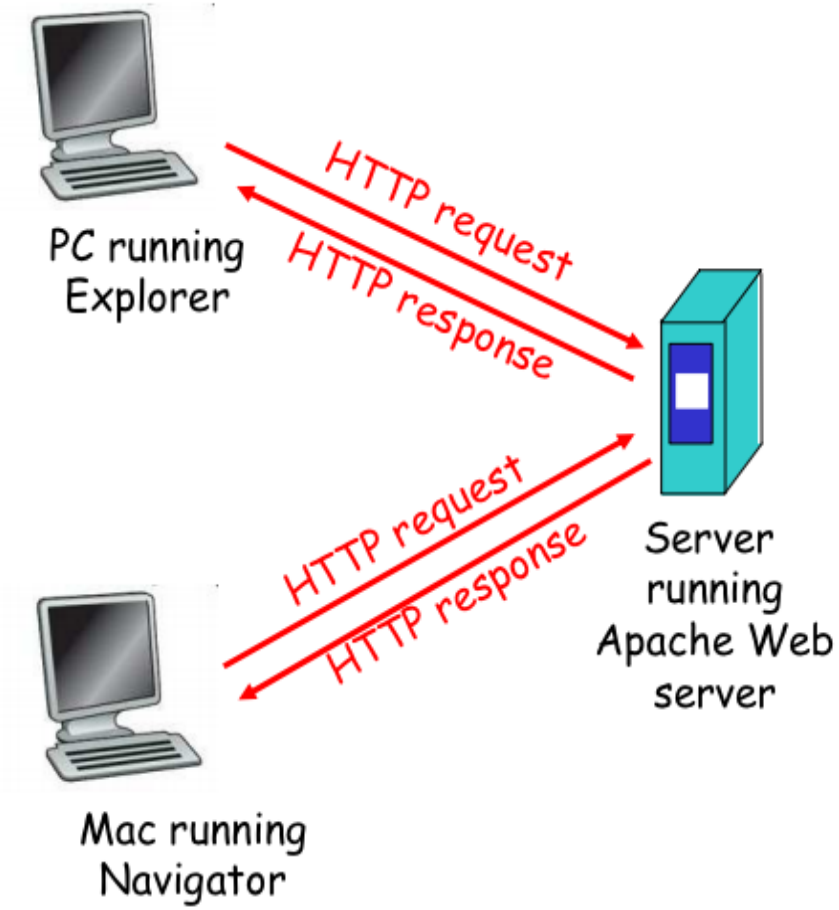
```
<!DOCTYPE html>
<html>
<head>
<title>HOLA MUNDO</title>
</head>
  <body>
    <h1>Hola Mundo</h1>
    <p>Este es el HTML m&aacute;s aburrido que v&iacute;.</p>
  </body>
</html>
```



Modelo HTTP

HTTP: hypertext transfer protocol.

- Protocolo de la capa aplicación de la Web.
- Modelo cliente/servidor
  - *cliente:* browser que requiere, recibe, "despliega" objetos Web.
  - *servidor:* Servidor Web envía objetos en respuesta a requerimientos.
- HTTP 1.0: RFC 1945.
- HTTP 1.1: RFC 2616.



HTTP persistente

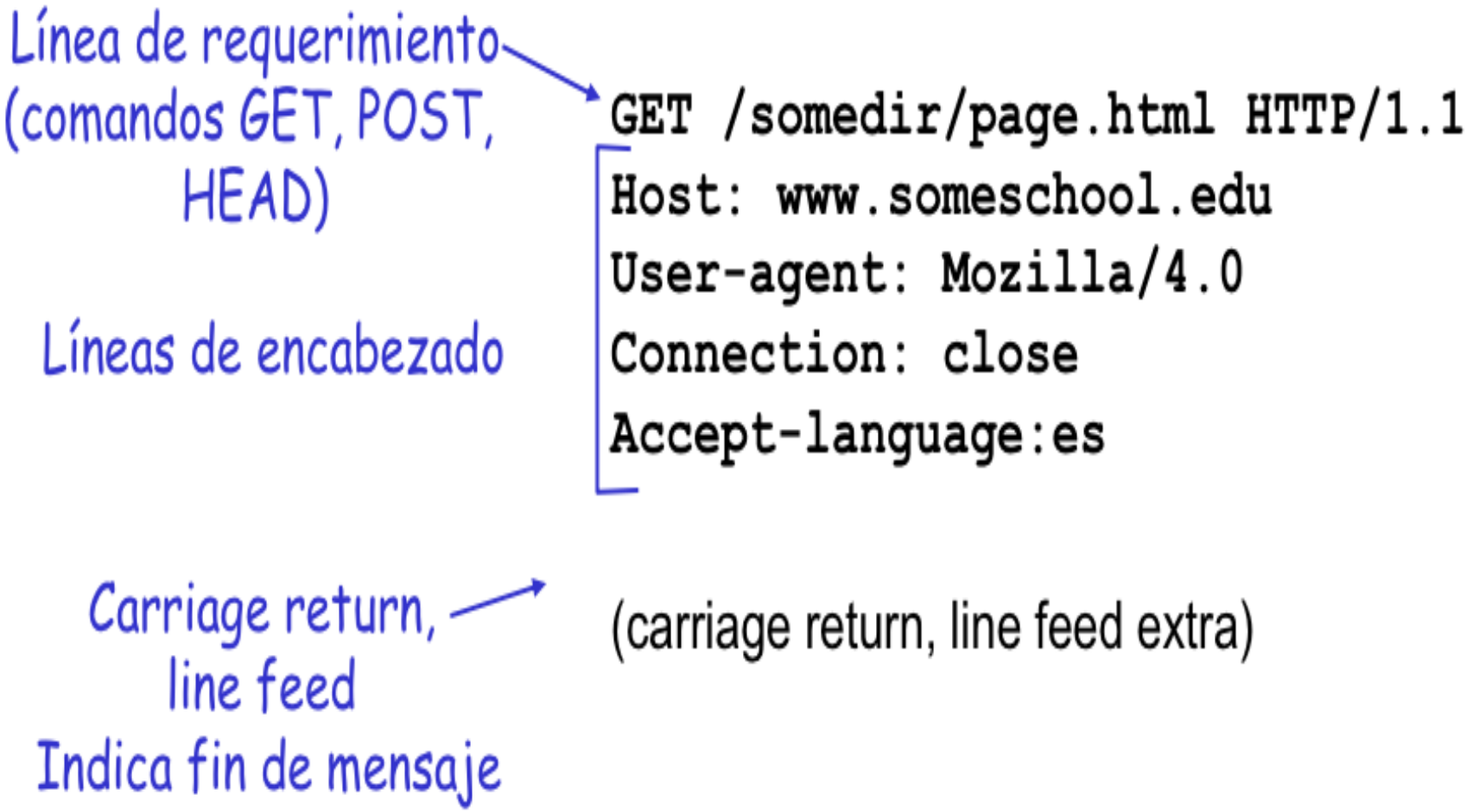
¿Por qué?

- $2 * RTT * Objeto$.
- Muchas conexiones, sobrecarga OS.
- Congestion Window.

Pipelining

- Espera completar los pedidos antes de envíar nuevos requests.
- Con PL, envía request apenas encuentra referencias (default en 1.1).

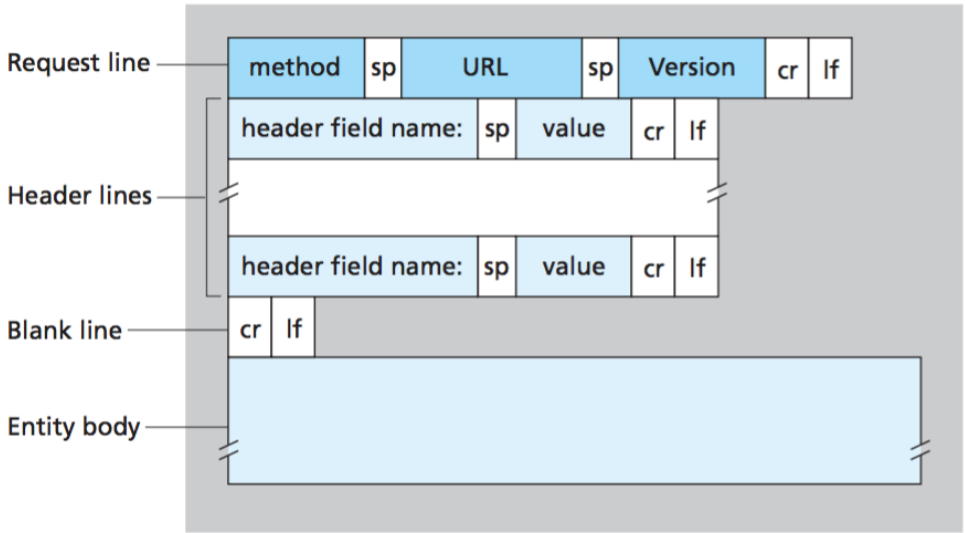
Petición HTTP



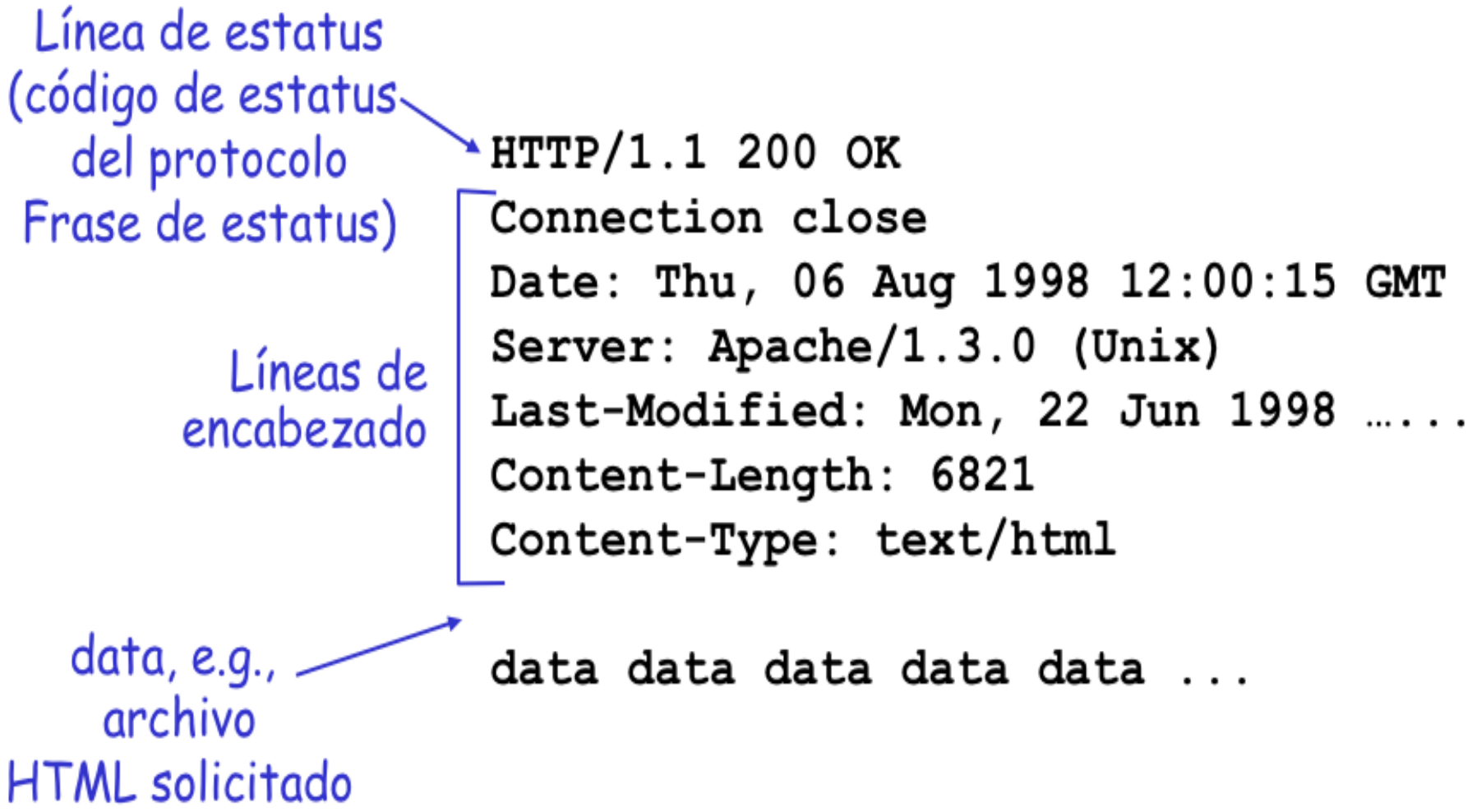
Petición HTTP

Tipos

- GET.
- HEAD.
- POST.
- PUT.
- DELETE.



Respuesta HTTP

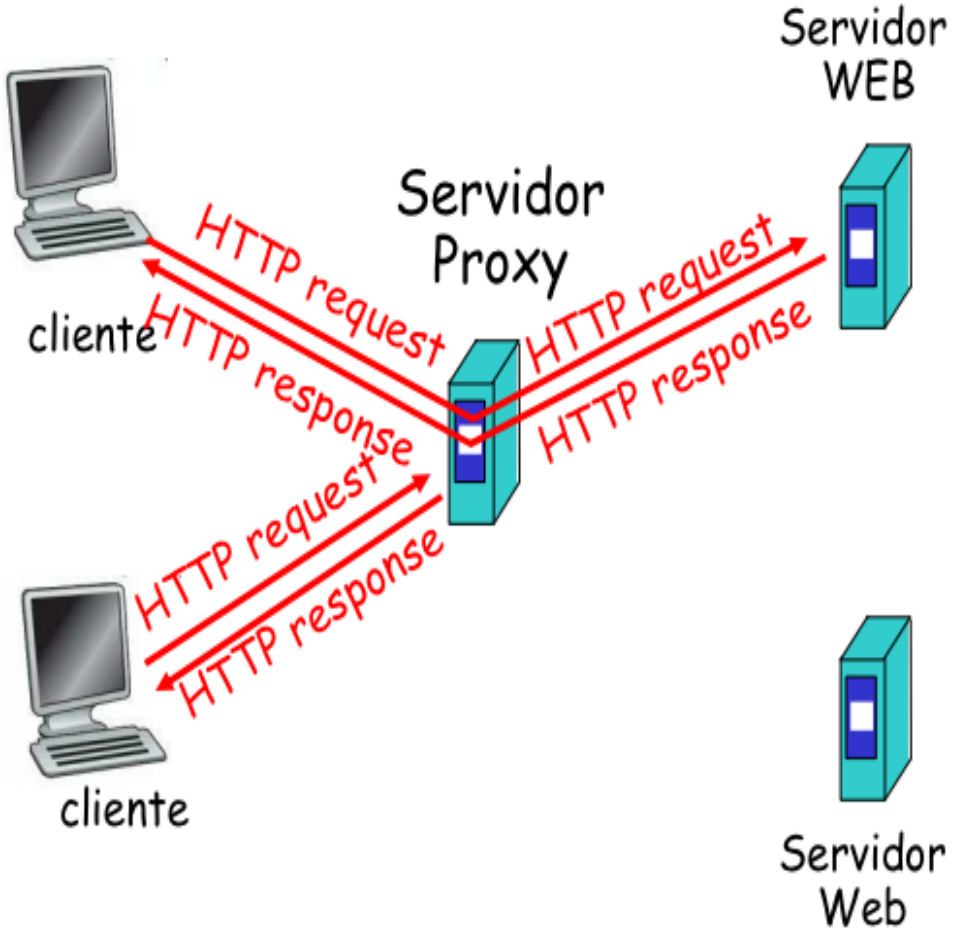


Algunos campos del encabezado

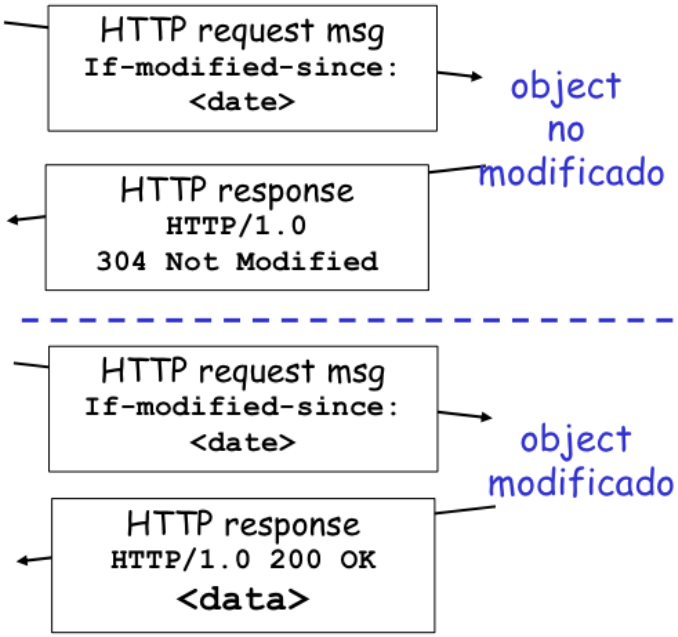
Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

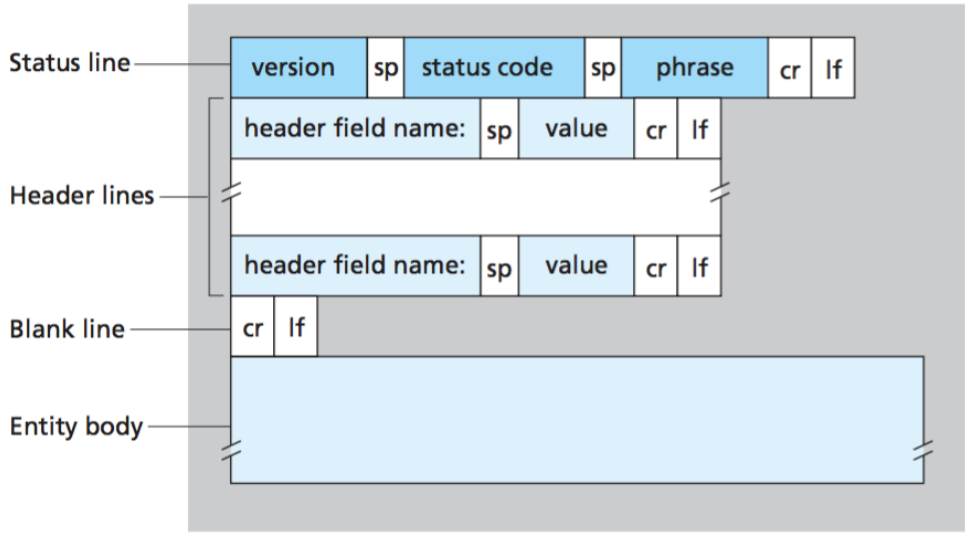
Técnicas de cacheo

- Usuario configura el browser: Acceso Web vía cache.
- Browser envía todos los requerimientos HTTP al cache:
  - Si objeto está en cache: cache retorna objeto.
  - Sino cache requiere los objetos desde el servidor Web, y retorna el objeto al cliente.



Cacheo HTTP y formato de respuesta





Registros DNS

Dada la siguiente base de registros de un servidor DNS de la facultad de exactas.

Base de registros DNS

```
exactas.uba.ar. 1w    IN    SOA      exactas.uba.ar
                                     jperez.exactas.uba.ar (
                                     2005091900 3h 1h 1w 1h)
exactas.uba.ar.       IN    NS       ns.exactas.uba.ar
exactas.uba.ar.       IN    MX       mailserver.exactas.uba.ar
mailserver            IN    CNAME    proxy.exactas.uba.ar
www                   IN    A        208.190.1.20
ads                   IN    A        208.190.1.21
proxy                 IN    A        208.190.1.22
proxy                 IN    A        208.190.1.23
proxy                 IN    A        208.190.1.24
ns                    IN    A        208.190.1.26
zorzal                IN    A        208.190.1.21
pc1                   IN    A        208.190.1.30
```

Consulta DNS

Además de la base de registros, se muestra una salida por pantalla de una consulta DNS:

Consulta parcial

```
;; flags: qr rd ra;

;; QUESTION SECTION:
;exactas.uba.ar. IN MX

;; ANSWER SECTION:
....

;; Query time: 3 msec
;; SERVER: 208.190.1.26#53(208.190.1.26)
;; WHEN: Sat Feb 31 11:06:56 2046
;; MSG SIZE  rcvd: 328
```

Enunciado

a. Explique en qué consiste la consulta realizada y complete la sección answer de la consulta con una respuesta válida que permita el acceso al servicio solicitado.

b. Detalle los mensajes HTTP (*Requests* y *Responses*) y DNS (*Consultas* y *Respuestas*) que desencadena un *Request* GET por el recurso `https://www.pagina12.com.ar/secciones/sociedad` por parte de la `pc1` al servidor proxy hasta que le llega el respectivo *Response*.

¿Repasando...?

¿Dudas?,

¿Preguntas?