

Capa Aplicación:Correo Electrónico y DNS

ELO322: Redes de Computadores Agustín J. González

Este material está basado en:

- Material de apoyo al texto *Computer Networking: A Top Down Approach Featuring the Internet 3rd* edition. Jim Kurose, Keith Ross Addison-Wesley, 2004.

Capítulo 2: Capa Aplicación

- 2.1 Principios de la aplicaciones de red
- 2.2 Web y HTTP
- 2.3 FTP
- 2.4 Correo Electrónico
 - SMTP, POP3, IMAP
- **2.5 DNS**
- 2.6 P2P
Compartición de archivos
- 2.7 Programación de Socket con TCP
- 2.8 Programación de socket con UDP
- 2.9 Construcción de un servidor WEB

DNS: Domain Name System (Sistema de nombres de dominio)

Personas: muchos identificadores:

- ROL, RUT, name, # pasaporte

Host y router en Internet:

- Dirección IP (32 bit) – usada para direccionar datagramas (ideal para router)
- “nombre”, e.g., www.yahoo.com – son usados por humanos

Q: ¿Quién mapea entre direcciones IP y nombres?

Domain Name System:

- *Base de datos distribuida* implementada en una jerarquía de muchos *servidores de nombres*
- *Protocolo de capa aplicación* permite a host, routers, y servidores de nombre comunicarse para *resolver* nombres (traducción dirección/nombre)
- No está orientado al uso directo de los usuarios, ellos usan nombres.
 - DNS es función central de la Internet implementada como protocolo de capa aplicación
 - La idea de diseño de Internet es dejar la complejidad en la “periferia” de la red.

DNS

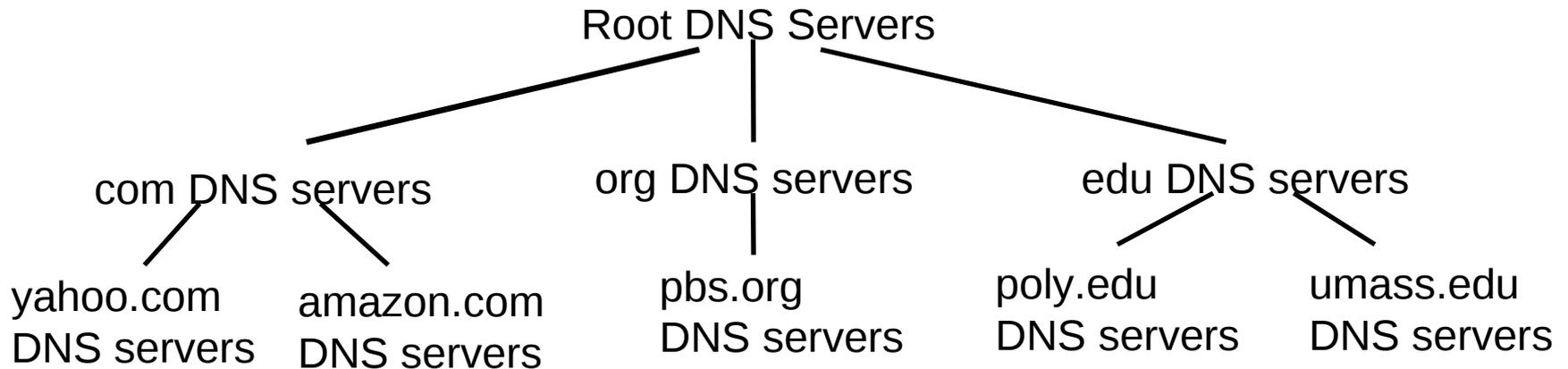
Servicios DNS

- ▣ Traducción de nombre de host a dirección IP
- ▣ Alias para host
 - ▣ Nombre canónico o alias
 - ▣ Nombre canónico: CNAME en RFC 1035
- ▣ Alias para servidor de correo
- ▣ Distribución de carga
 - ▣ Servidores Web replicados: conjunto de direcciones IP para un nombre canónico (e.g. relay1.west-coast.amazon.com), servidores DNS rota entre direcciones IP

¿Por qué no centralizar DNS?

- ▣ Único punto de falla
 - ▣ Volumen de tráfico, muchos necesitan el DNS
 - ▣ Sería una base de datos centralizada distante con grandes retardos de acceso.
 - ▣ Mantenición, es mejor que cada dominio gestione sus nombres
- Porque no sería *escalable!*

Base de datos jerárquica y distribuida



Cliente desea IP de www.amazon.com; 1^{ra} aprox. :

- Cliente consulta al servidor raíz para encontrar servidor DNS de com
- Cliente consulta servidor DNS TLD (Top Level Domain) de com para obtener servidor DNS de amazon.com
- Cliente consulta servidor DNS amazon.com para obtener dirección IP de www.amazon.com

DNS: servidores de nombre en raíz

- ▮ Son contactados por servidor de nombre local cuando no puede resolver un nombre
- ▮ Servidor nombre raíz:
 - ▮ Contacta servidor de nombre autoritario de la zona superior (e.g. com) si mapeo del nombre es desconocido para él
 - ▮ Obtiene mapeo (propio o desde otro servidor raíz)
 - ▮ Retorna mapeo al servidor de nombre local



TLD y Servidores Autoritarios

- ▣ **Top-level domain (TLD) servers:** responsable por com, org, net, edu, etc., y todos los dominios superiores de cada país: uk, fr, ca, jp, cl, etc..
 - ▣ Network solutions mantiene servidores para el TLD de com
 - ▣ Educause para el TLD de edu
 - ▣ Nic (network information center) para el TLD de cl (www.nic.cl)
- ▣ **Servidores DNS autoritarios:** son servidores DNS de las organizaciones y proveen mapeos autoritarios entre hostname e IP (e.g., Web y mail).
 - ▣ Éstos pueden ser mantenidos por la organización o el proveedor de servicio

Servidor de nombre local

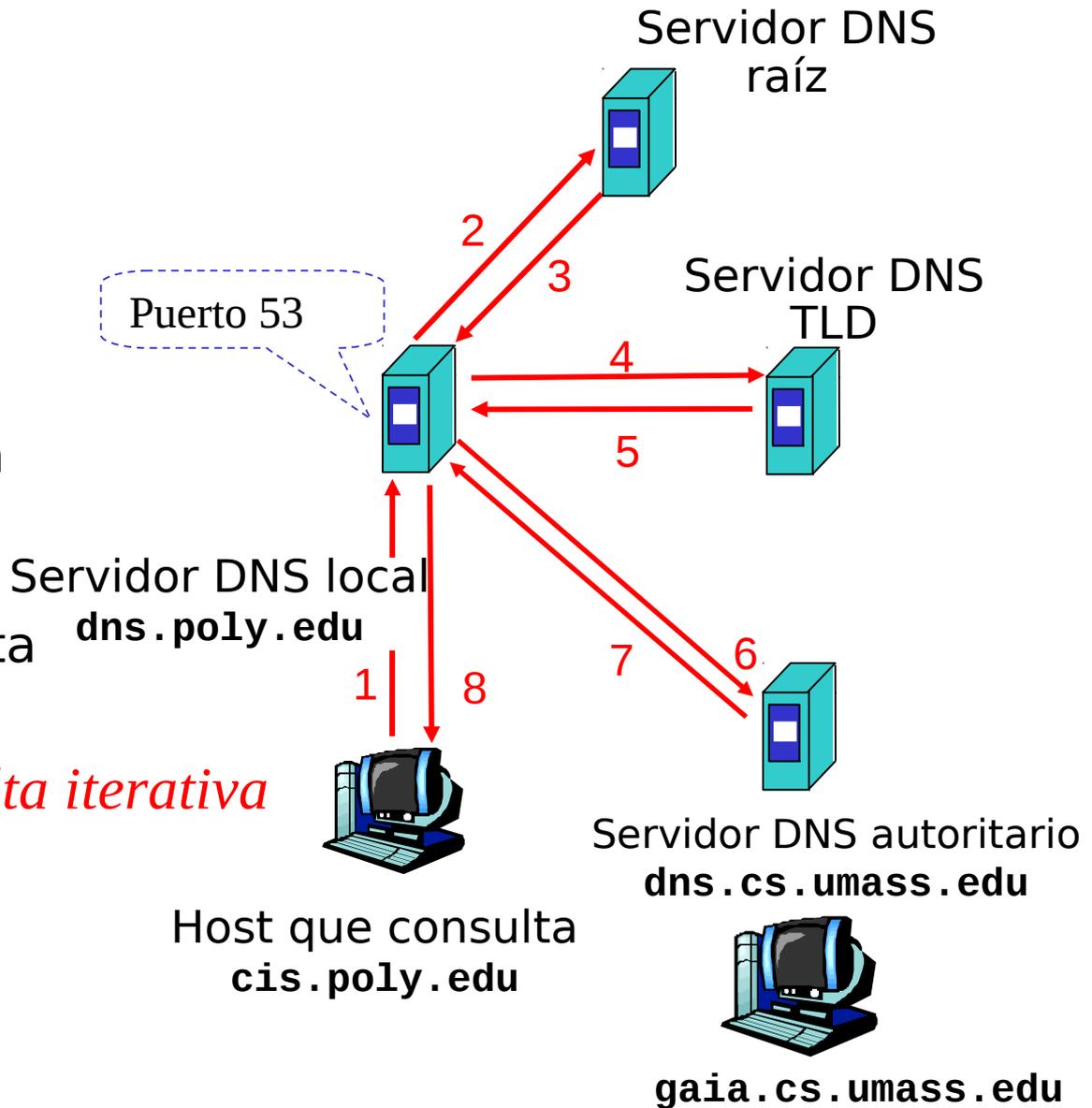
- No pertenece estrictamente a la jerarquía
- Cada ISP (ISP residencial, compañía, universidad) tiene uno.
 - También son llamados “servidor de nombre por omisión” (default name server)
- Cuando un host hace una consulta DNS, ésta es enviada a su servidor DNS local
 - Actúa como proxy, re-envía consulta dentro de la jerarquía.

Ejemplo 1

Consulta iterativa:

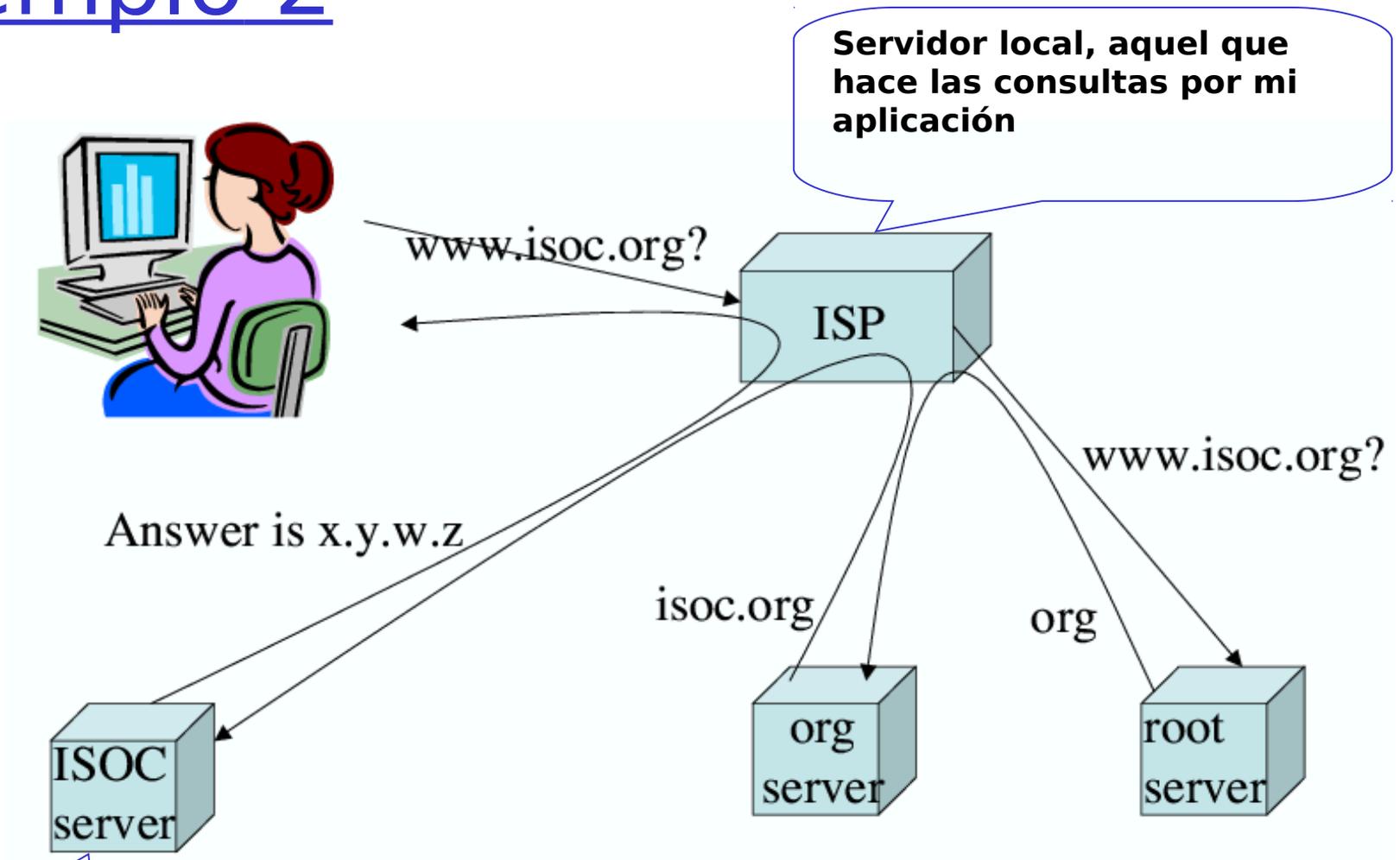
- Host en cis.poly.edu quiere la dirección IP de gaia.cs.umass.edu
- Servidor contactado responde con el nombre del servidor a contactar
- “Yo no conozco este nombre, pero pregunta a este servidor”

Consulta iterativa



gaia.cs.umass.edu

Ejemplo 2



Consulta iterativa

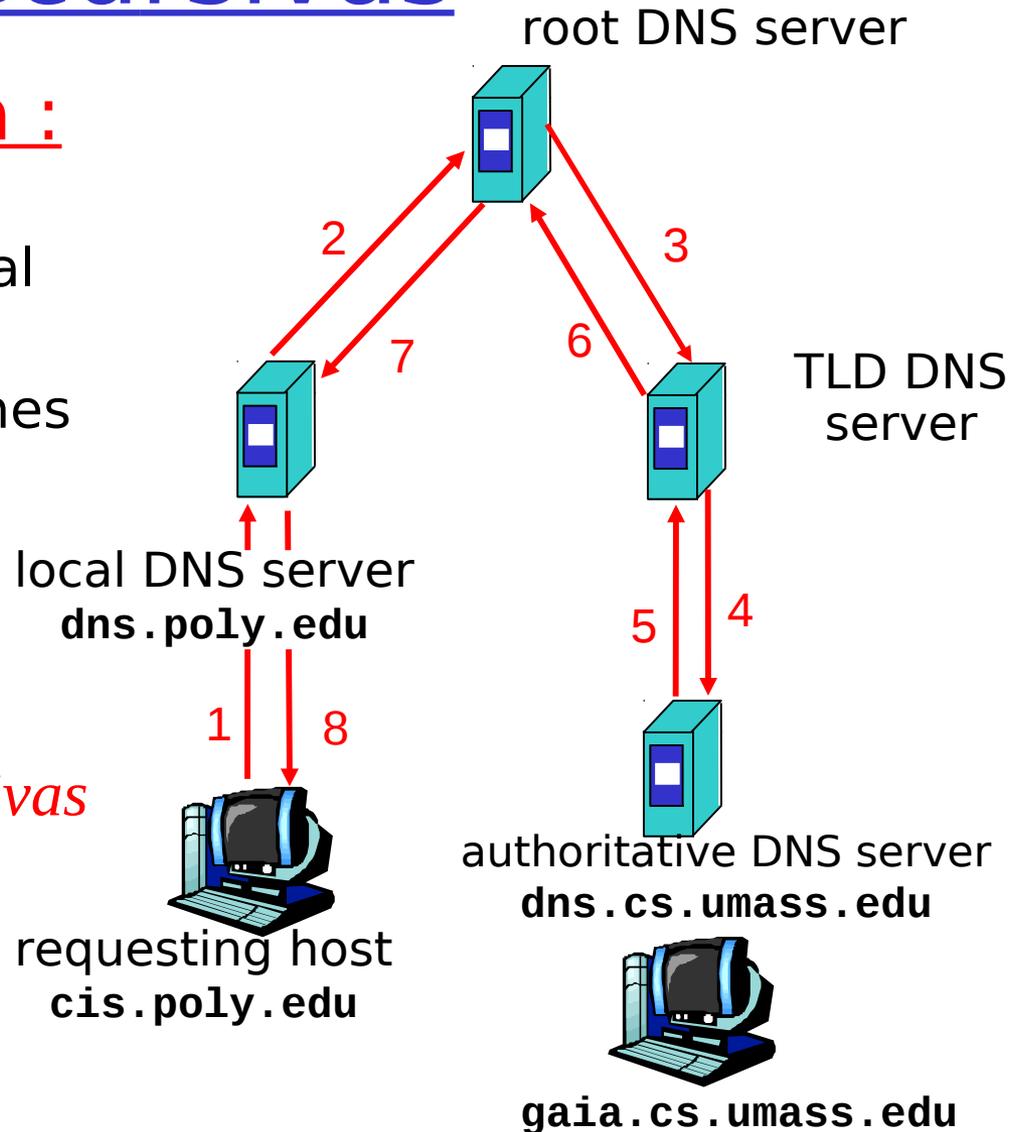
Servidor autoritario, aquel que define el mapeo nombre <-> IP

Consultas Recursivas

Consulta recursiva :

- Pone la carga de la resolución de nombre al servidor contactado.
- ¿Qué pasa en situaciones de alta carga?

Consultas Recursivas



Ejemplo

- Hacer algo del tipo:
\$ nslookup www.elo.utfsm.cl
- Luego:
\$ nslookup 200.1.17.5
- Finalmente:
\$ nslookup www.google.com
- Y
\$ nslookup 64.233.161.99
- Estando en aragorn hacer:
\$ nslookup 200.1.17.195

Es común que las máquinas tengan asignados alias; por ejemplo www.elo.utfsm.cl es un alias para vega.elo.utfsm.cl. ¿Expliqué por qué no conviene usar www.elo.utfsm.cl como nombre canónico de la máquina?

- Usando www.elo.utfsm.cl como alias es posible configurar otra máquina para reemplazar el servidor WEB y una vez que esté lista, sólo hacemos el cambios del alias en el DNS para que los accesos futuros se dirijan al nuevo servidor. Esta operación resulta transparente para los usuarios. Si fuera nombre canónico, mientras configuramos el nuevo servidor WEB, no podríamos tener dos máquinas con igual nombre.

DNS: Cache y actualización de registros

- Una vez que un servidor de nombre conoce un mapeo, éste *guarda* (caches) el mapeo
 - Las entradas del cache expiran (desaparecen) después de algún tiempo
 - Servidores TLD típicamente están en cache de los servidores de nombre locales
 - Así los servidores de nombre raíz no son visitados con frecuencia
- Mecanismos de Actualización/notificación están bajo diseño por el IETF (Internet Engineering Task Force)
 - RFC 2136
 - <http://www.ietf.org/html.charters/dnsind-charter.html>

Registros DNS

DNS: es una base de datos distribuida que almacena registros de recursos (resource records, **RR**)

Formato RR: (name, value, type, ttl)

□ Type=A

- **name** es un hostname (nombre real o canónico)
- **value** es una dirección IP

□ Type=NS

- **name** es un dominio (e.g. foo.com)
- **value** es la dirección IP (nombre) del servidor autoritario que sabe cómo obtener las direcciones IP de este dominio.

□ Type=CNAME

- **name** es un alias para algún nombre real (indicado en type A)
www.ibm.com es realmente
servereast.backup2.ibm.com

value es el nombre real (canónico)

□ Type=MX

value es el nombre del servidor de correo asociado con name

Inserción de registros en DNS

- Ejemplo: Recién se crea una empresa “Network Utopia”
- Debemos registrar el nombre networkutopia.com en un **administrador de dominio** (e.g., Network Solutions)
 - Necesitamos proveer el nombre y la dirección IP de nuestro servidor de nombre autoritario (primario y secundario)
 - Administrador del dominio inserta dos RRs en el servidor TLD com:
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
- Incorporar en el servidor autoritario un registro Tipo A para www.networkutopia.com y un registro Tipo MX para networkutopia.com
- En Chile debemos acceder a NIC Chile para arrendar un nombre de dominio.

Explique por qué los resultados de varios PING a `www.youtube.com` muestran direcciones IPs distintas:

```
agustin@agustin-laptop:~$ ping www.youtube.com
PING youtube-ui.l.google.com (74.125.224.76) 56(84) bytes of data.
64 bytes from 74.125.224.76: icmp_seq=1 ttl=52 time=162 ms
```

.... Luego:

```
agustin@agustin-laptop:~$ ping www.youtube.com
PING youtube-ui.l.google.com (74.125.224.42) 56(84) bytes of data.
64 bytes from 74.125.224.42: icmp_seq=1 ttl=52 time=160 ms
```

.... Luego:

```
agustin@agustin-laptop:~$ ping www.youtube.com
PING youtube-ui.l.google.com (74.125.224.79) 56(84) bytes of data.
64 bytes from 74.125.224.79: icmp_seq=1 ttl=52 time=175 ms
```

...

Explique cómo esto se hace posible.

- El ping a un mismo nombre lógico condujo a tres máquinas distintas por ello tres IPs distintas. Esto se explica porque el servicio de youtube es atendido por un conjunto de máquinas para poner servir a más usuarios a la vez.
- Esto es posible gracias al servidor DNS. Cuando el ping consulta por la IP de `www.youtube.com`, el servidor que maneja este nombre identifica la máquina adecuada de entre el conjunto para atender la petición y retorna esa dirección IP.



Capítulo 2: Capa Aplicación

- 2.1 Principios de la aplicaciones de red
- 2.2 Web y HTTP
- 2.3 FTP
- 2.4 Correo Electrónico
 - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 P2P
Compartición de archivos
- 2.7 Programación de Socket con TCP
- 2.8 Programación de socket con UDP
- 2.9 Construcción de un servidor WEB