The Application Layer

Chapter 7

Contents

- DNS – The Domain Name System
- The DNS name space
- WWW
- URL
- HTTP

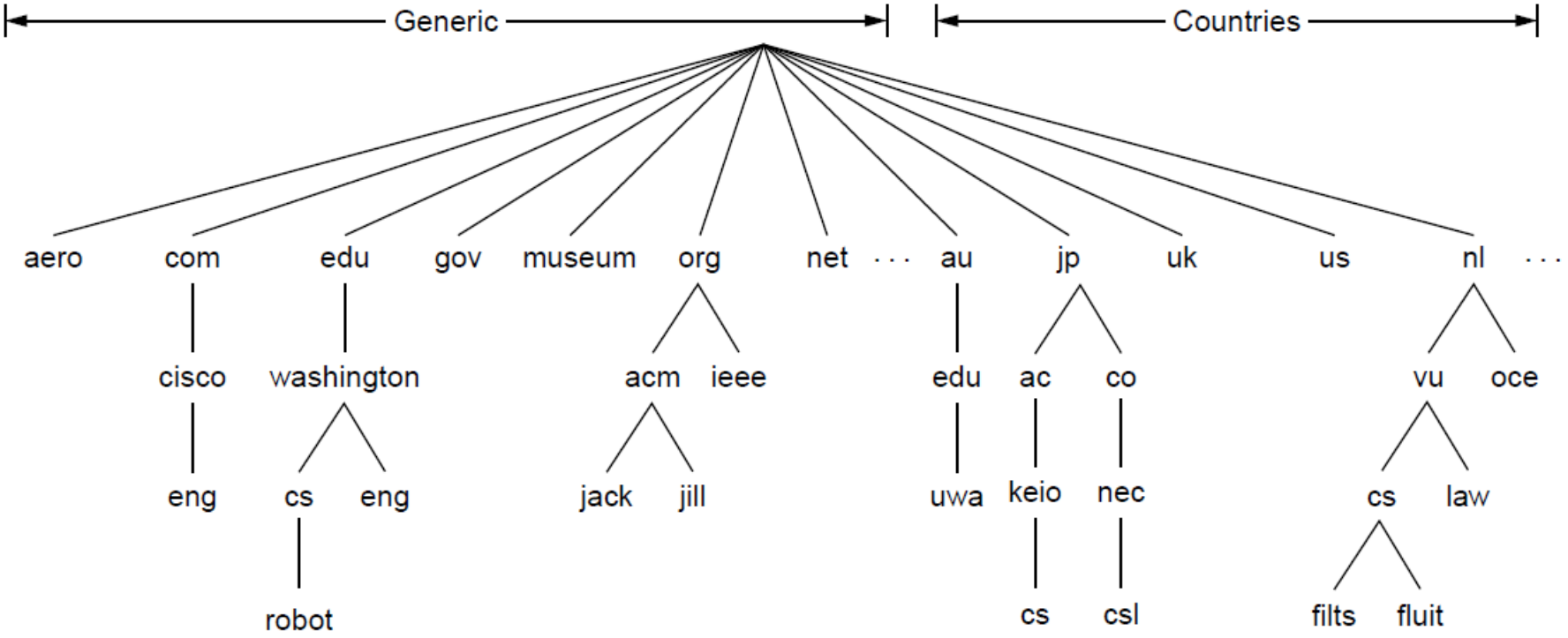
DNS – The Domain Name System

- IP addresses can be used to identify a host machine on the Internet
  - As those machines move around, the addresses need to be changed accordingly as well
- ASCII names have been used to decouple host names and their IPs to provide more flexibility
- The network itself still understands only numerical addresses
- The DNS was invented to manage and resolve host names into IP addresses

DNS: Characteristics

- A file ***host.txt*** listed all the hosts and their IP addresses, but issue some problems:
  - File size, load and latency
  - Host name conflict
- Essence of DNS:
  - Hierarchical
  - Domain-based naming scheme
  - A distributed database system
- To map a name onto an IP address, an application program:
  - Calls a library procedure called the ***resolver***, passing it the name as a parameter
  - The resolver sends a UDP packet to a local DNS server
  - DNS server looks up the name and returns the IP address to the resolver
  - Resolver returns it to the application
  - Armed with the IP address, the program can then establish a TCP connection with the destination or send it UDP packets

The DNS Name Space



A portion of the Internet domain name space.

DNS naming

- ✓ Domain names are case insensitive: edu, Edu, EDU have the same meaning
- ✓ Component name can be up to 63 characters
- ✓ Full path names must not exceed 255 characters
- ✓ Each domain name server manages its own name space. It can create subdomain names without asking for permission from upper server. Examples: hcmut.edu.vn and cse.hcmut.edu.vn

- DNS Name Space is divided into non-overlapping zones
- Each zone has Name Servers holding information about it.

The DNS Name Space (2)

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

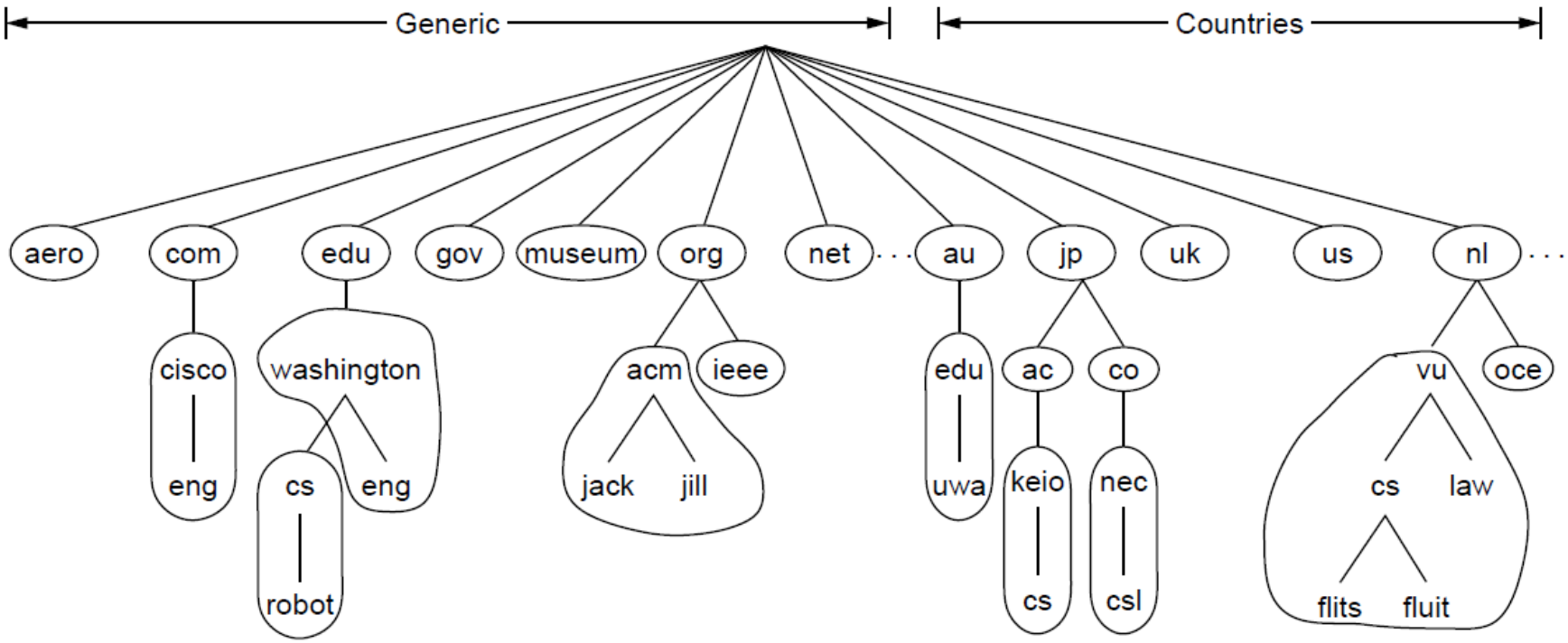
Generic top-level domains

Domain Resource Records (1)

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

The principal DNS resource record types

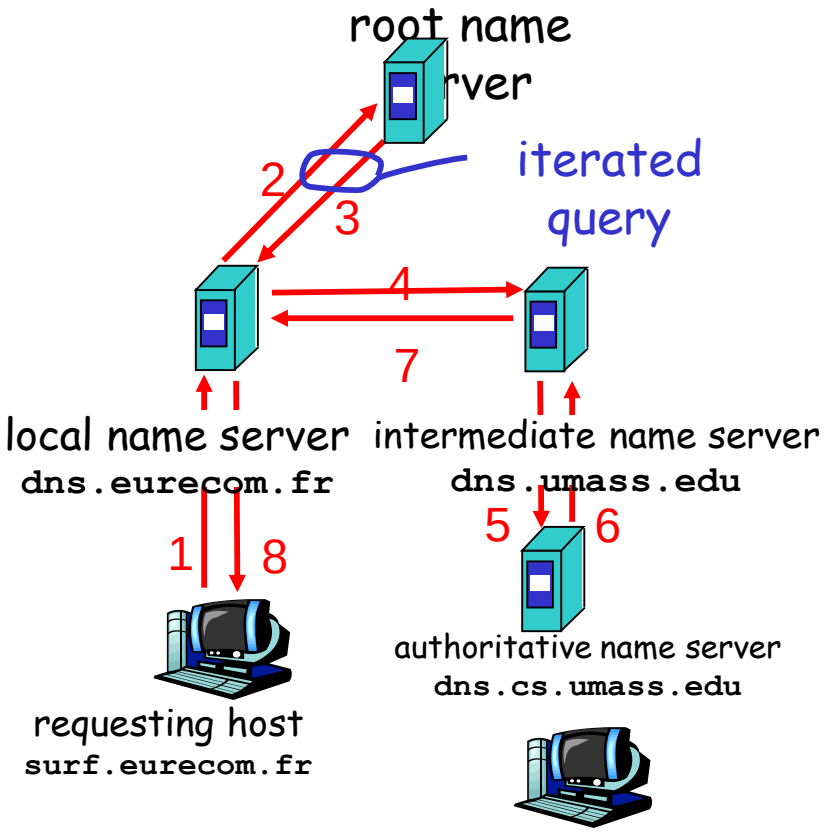
Name Servers (1)



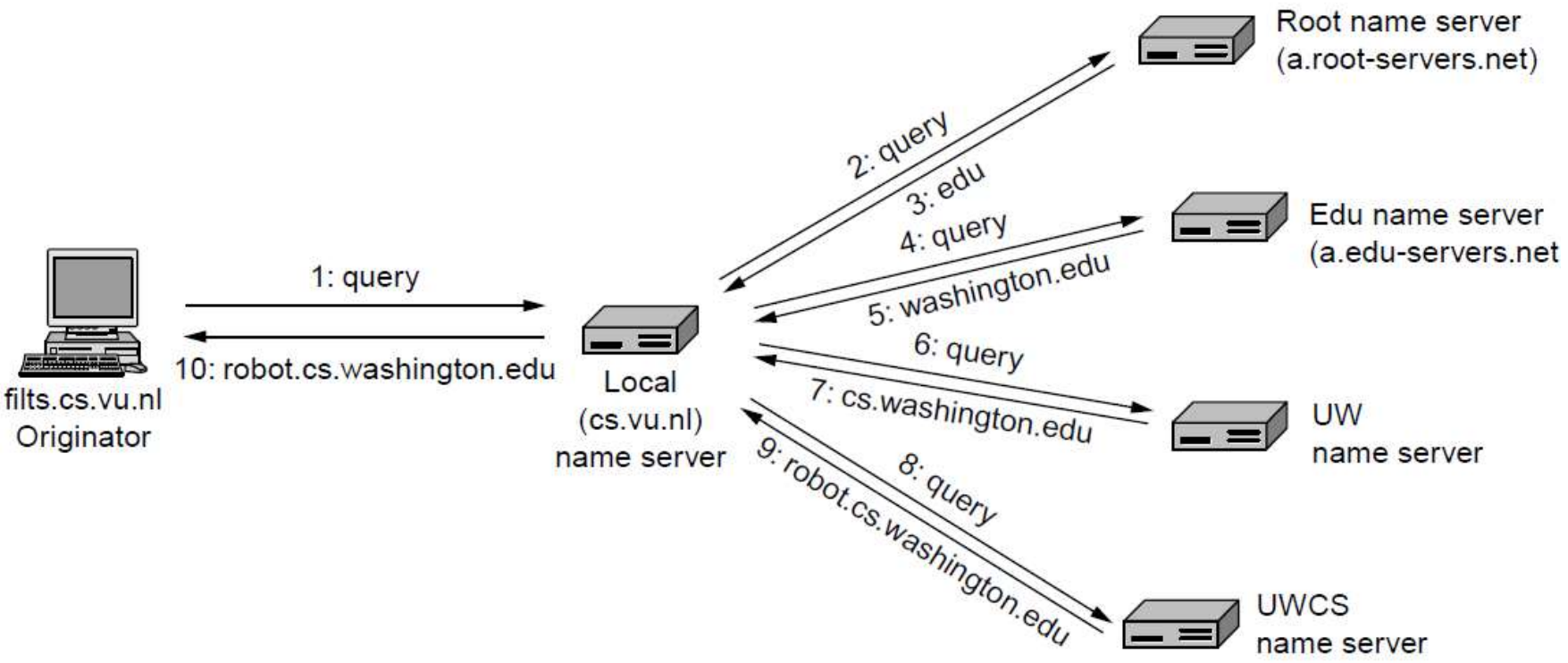
Part of the DNS name space divided into zones (which are circled).

DNS - Query

- recursive query
  - puts burden of name resolution on contacted name server.
- iterated query
  - contacted server replies with name of server to contact.



Name Servers



Example of a resolver looking up a remote name in 10 steps.

World Wide Web

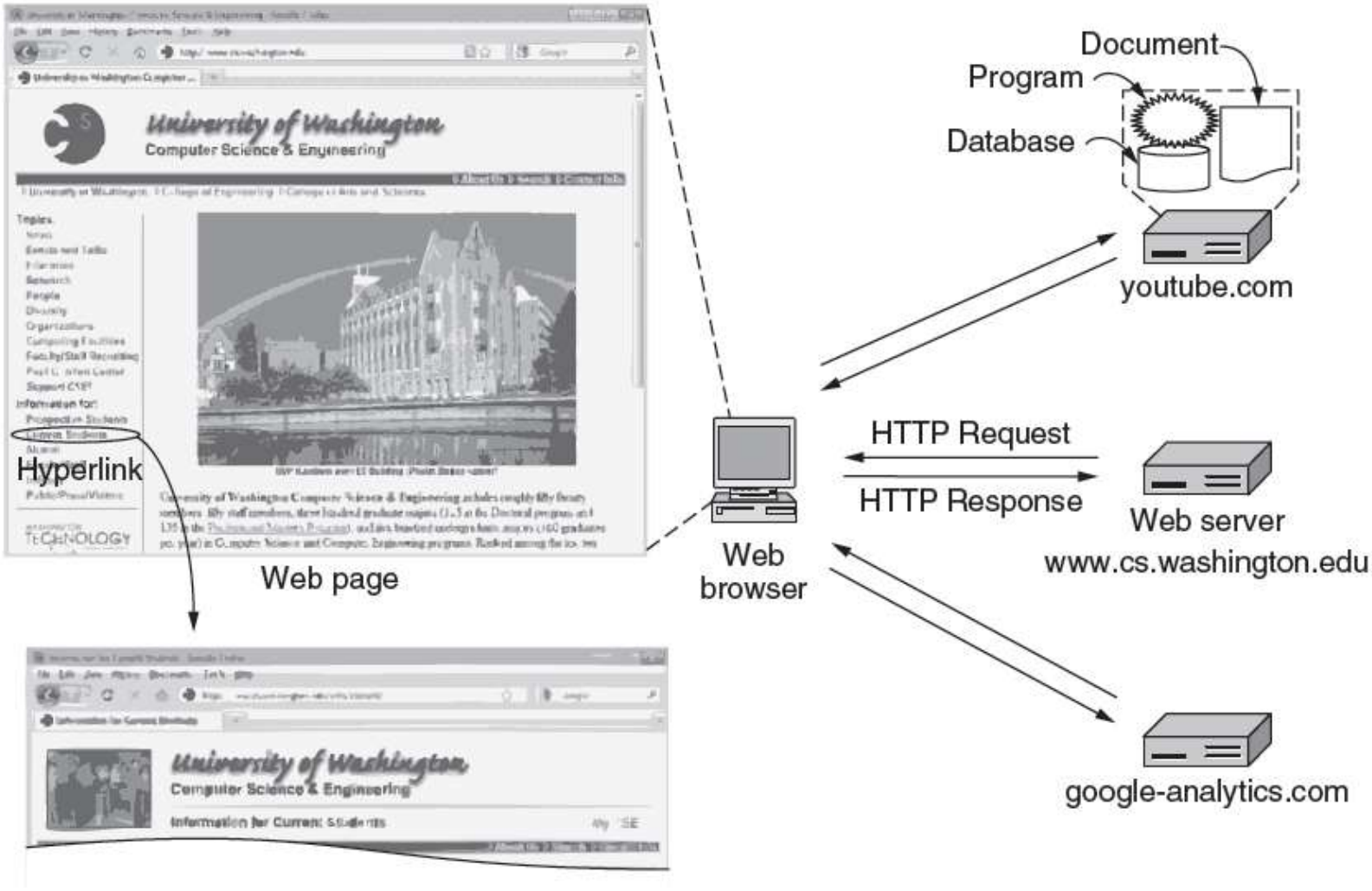
- Began in 1989 at CERN (Switzerland) by Tim Berners-Lee
- An architectural framework for accessing linked documents spread out over millions of machines all over the Internet
- Web consists of a vast, worldwide collection of documents, called Web pages
- Hypertext: Each page may contain links to other pages anywhere in the world, and repeated indefinitely
- Pages are viewed with a program called a browser

Architectural Overview

Three questions had to be answered before a selected page could be displayed:

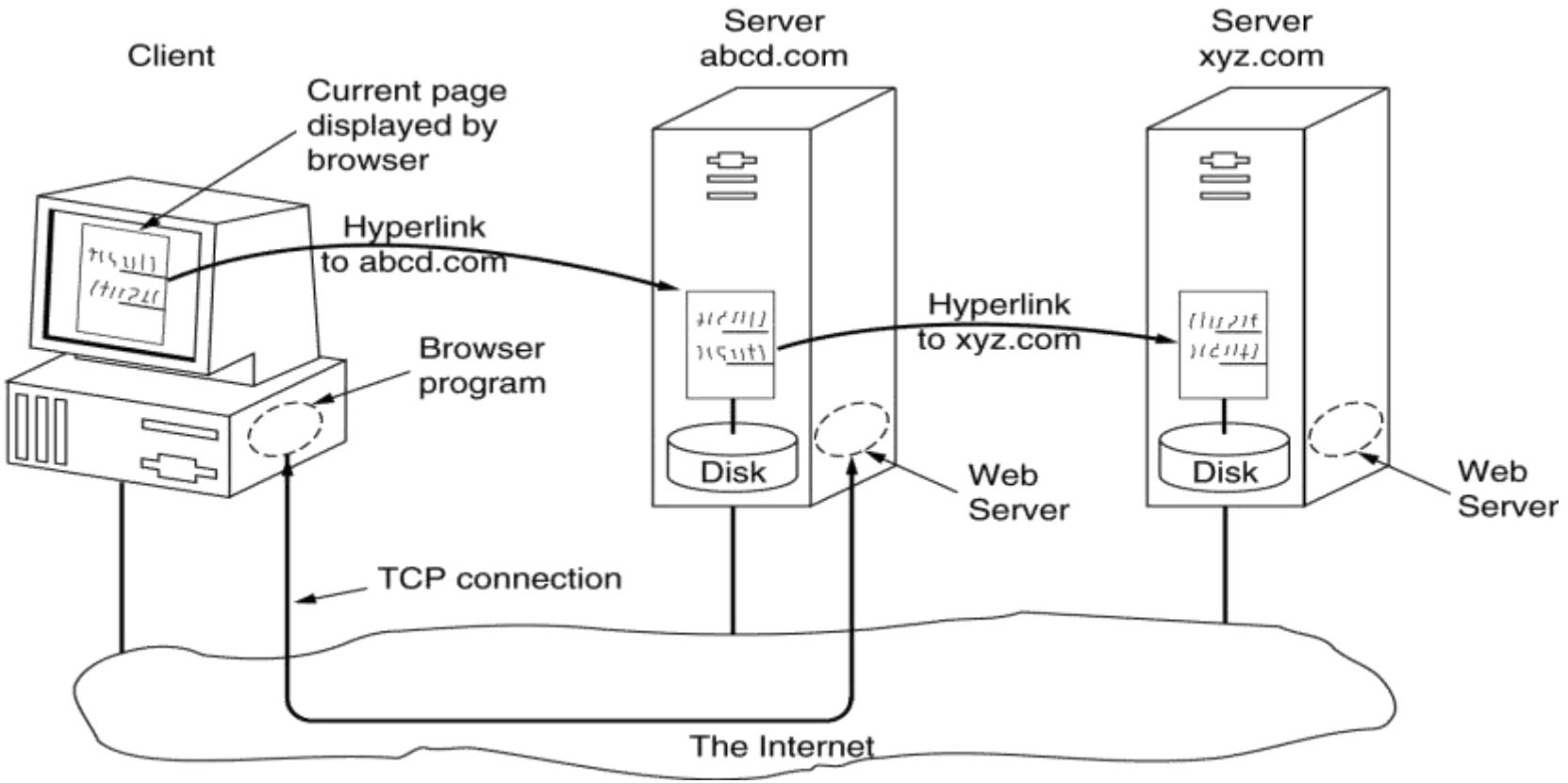
1. What is the page called?
2. Where is the page located?
3. How can the page be accessed?

Architectural Overview



Architecture of the Web.

Architectural Overview



Architectural Overview

Steps that occur when link is selected:

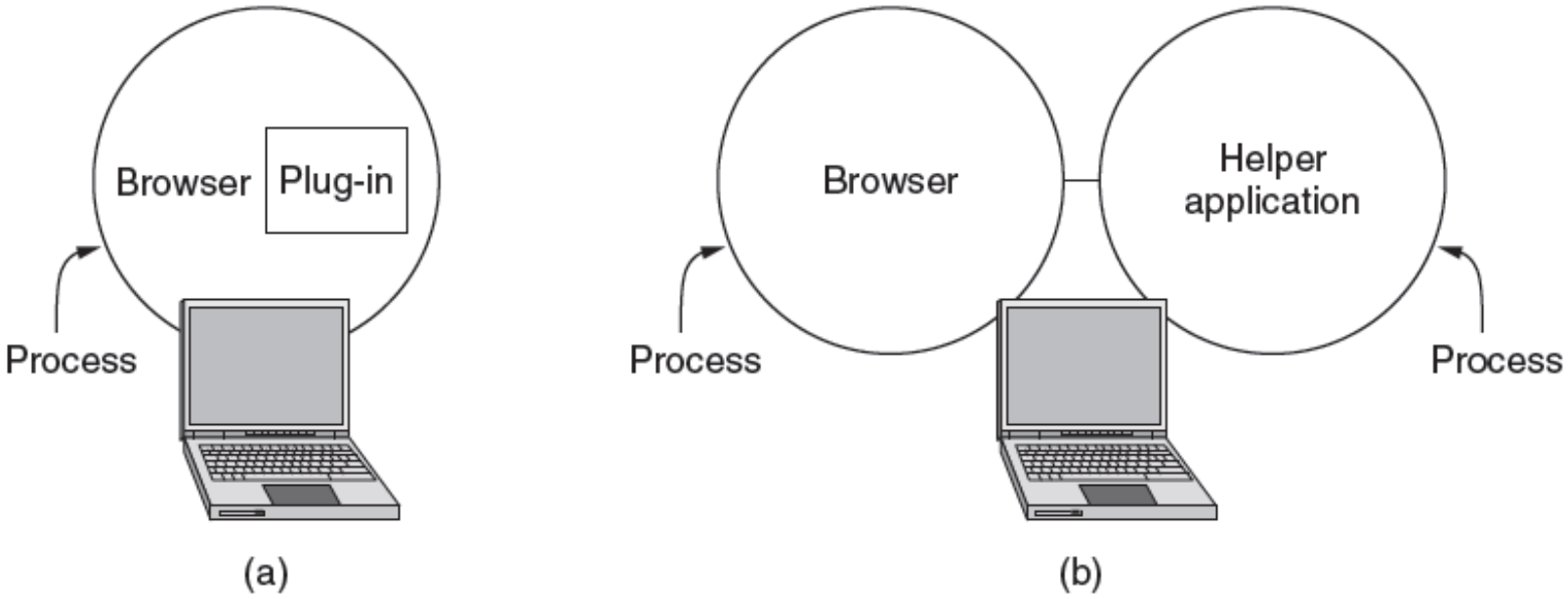
1. Browser determines the URL
2. Browser asks DNS for the IP address of the server
3. DNS replies
4. The browser makes a TCP connection
5. Sends HTTP request for the page
6. Server sends the page as HTTP response
7. Browser fetches other URLs as needed
8. The browser displays the page
9. The TCP connections are released

Architectural Overview

Name	Used for	Example
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hypertext with security	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
mailto	Sending email	mailto:JohnUser@acm.org
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg
sip	Multimedia calls	sip:eve@adversary.com
about	Browser information	about:plugins

Some common URL schemes.

Architectural Overview



(a) A browser plug-in. (b) A helper application.

Plugin and Helper

- Plug-in is a code module
- The browser fetches from a special directory on the disk and installs as an extension to itself
- After the plug-in has done its job, it is removed from the browser's memory

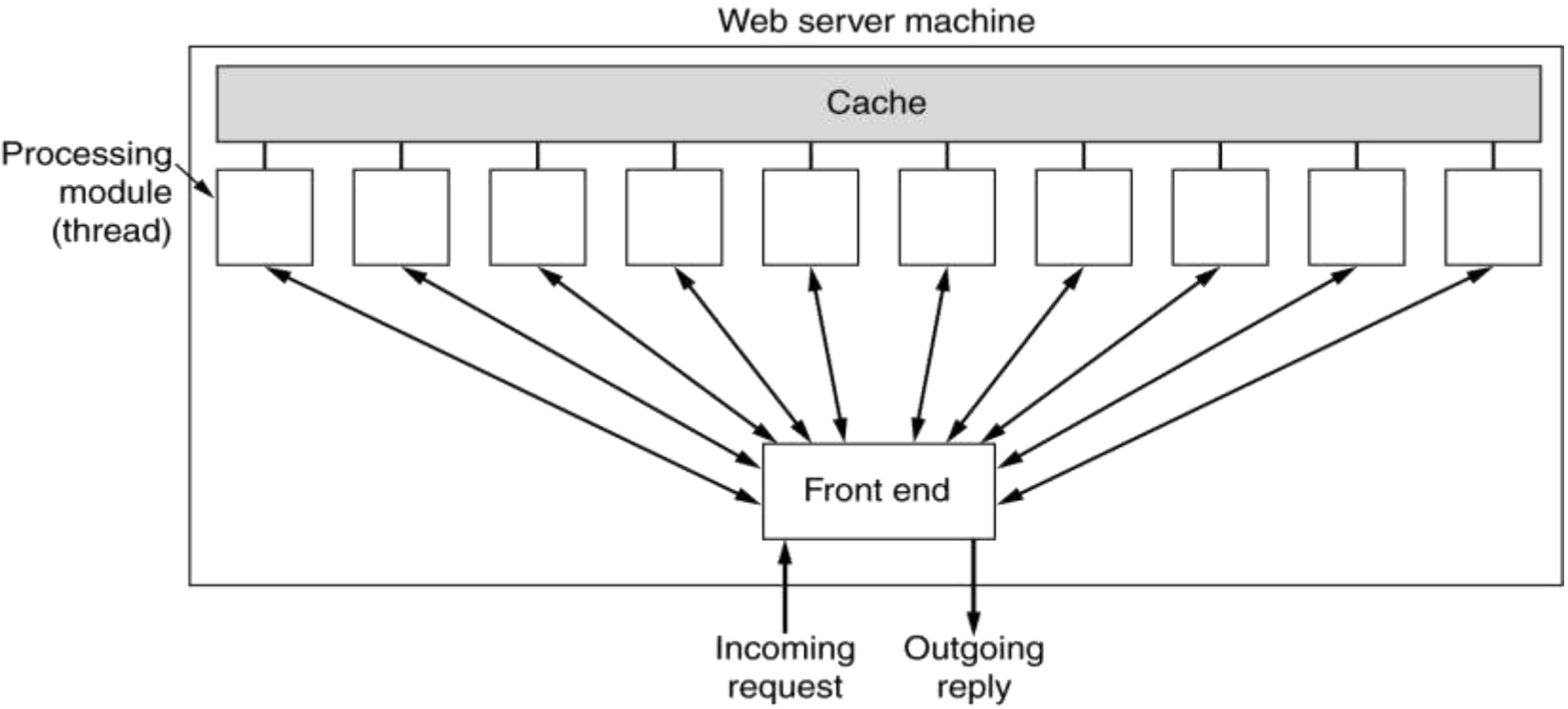
Helper application is a complete program, running as a separate process

Accepts the name of a scratch file

Opens the file, and displays the contents

Server Side: Multithreads

A multithreaded Web server with a front end and processing modules.

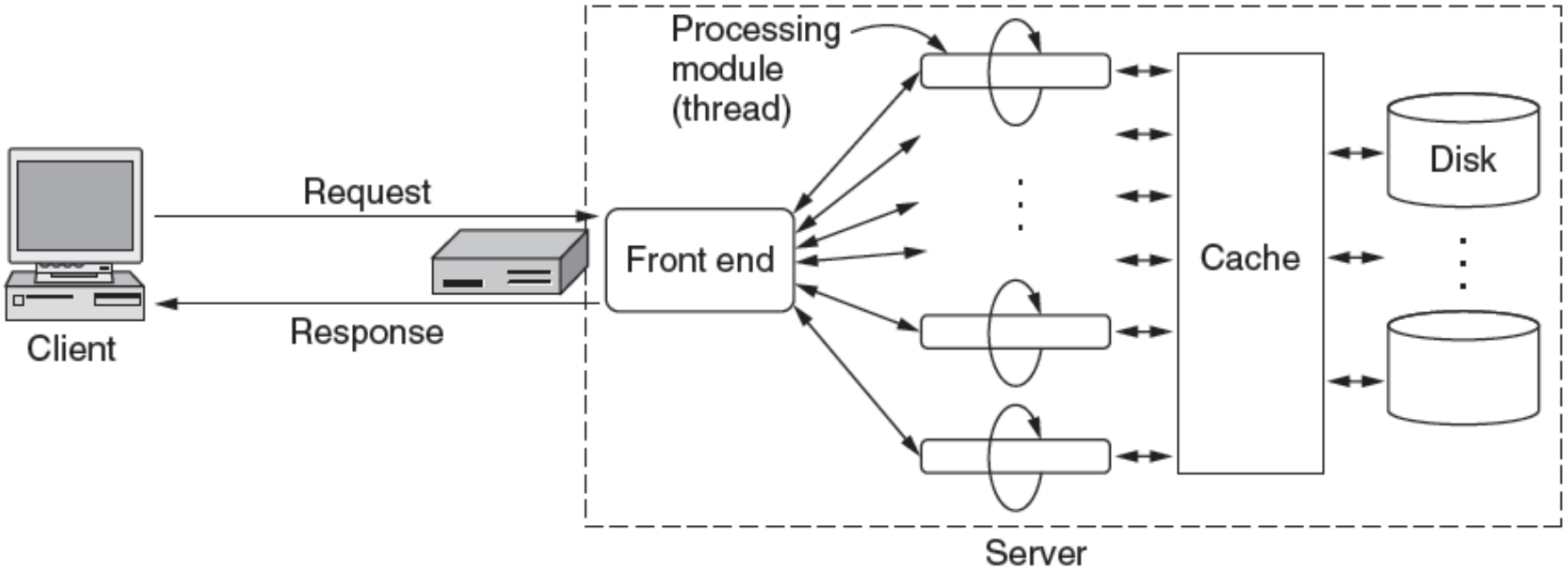


Architectural Overview

Steps server performs in main loop

1. Accept a TCP connection from client
2. Get path to page, name of file requested.
3. Get the file (from disk).
4. Send contents of the file to the client.
5. Release the TCP connection.

Architectural Overview



A multithreaded Web server with a front end and processing modules.

Architectural Overview

A processing module performs a series of steps:

1. Resolve name of Web page requested.
2. Perform access control on the Web page.
3. Check the cache.
4. Fetch requested page from disk or run program
5. Determine the rest of the response
6. Return the response to the client.
7. Make an entry in the server log.

URLs – Uniform Resource Locators

- URLs are used to refer to resources in the Internet, such as a web page

Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80

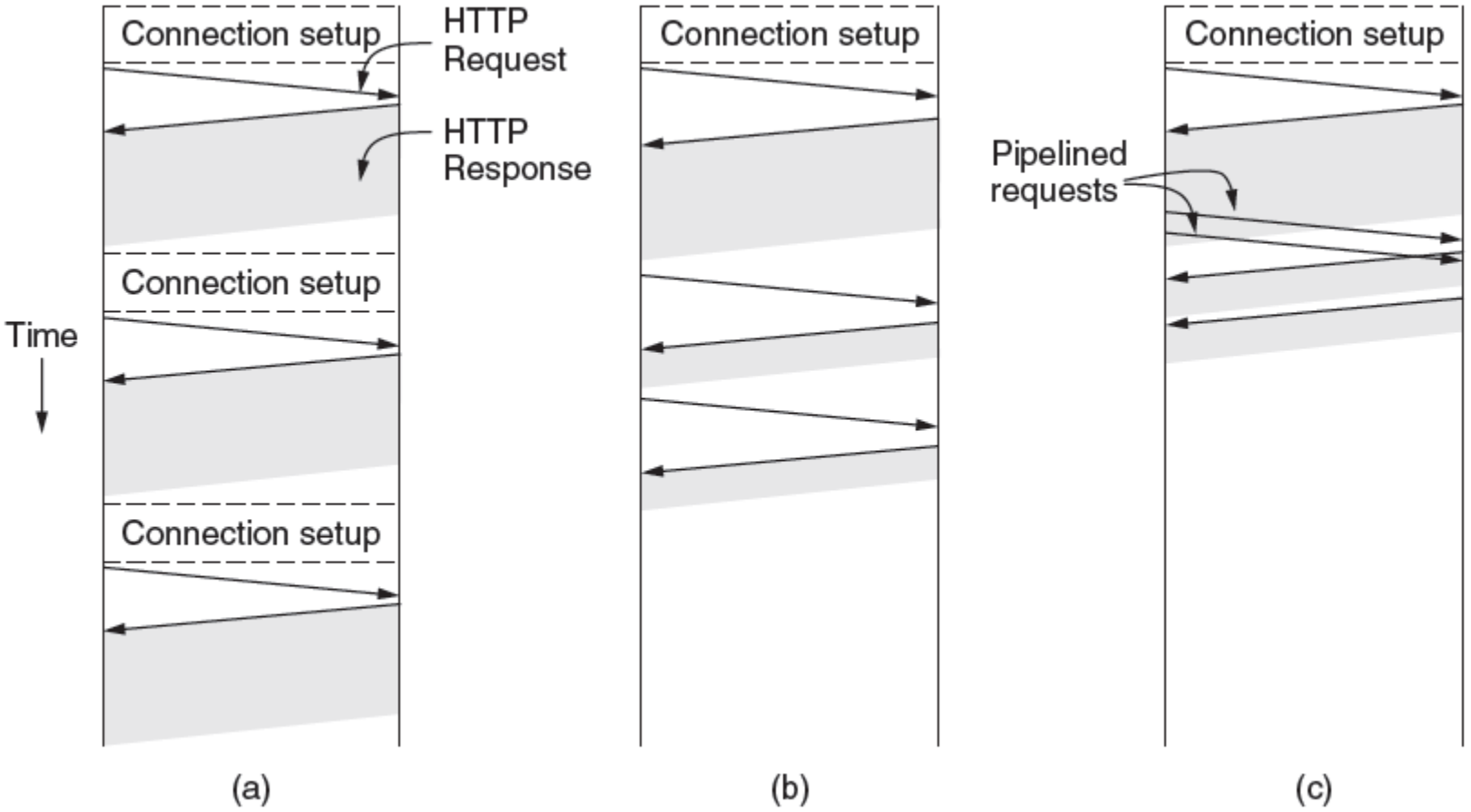
Cookies

a) Cookies - When a client requests a Web page, the server can supply additional information along with the requested page. This information may include a **cookie**, which is a small (at most 4 KB) file (or string)

b) The name derives from ancient programmer slang in which a program calls a procedure and gets something back that it may need to present later to get some work done.

c) RFC 2109

The HyperText Transfer Protocol



HTTP with (a) multiple connections and sequential requests.
(b) A persistent connection and sequential requests.
(c) A persistent connection and pipelined requests.

The HyperText Transfer Protocol

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

The built-in HTTP request methods.

The HyperText Transfer Protocol

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

The status code response groups

The HyperText Transfer Protocol

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
If-Modified-Since	Request	Time and date to check freshness
If-None-Match	Request	Previously sent tags to check freshness
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Referer	Request	The previous URL from which the request came
Cookie	Request	Previously set cookie sent back to the server
Set-Cookie	Response	Cookie for the client to store
Server	Response	Information about the server

...

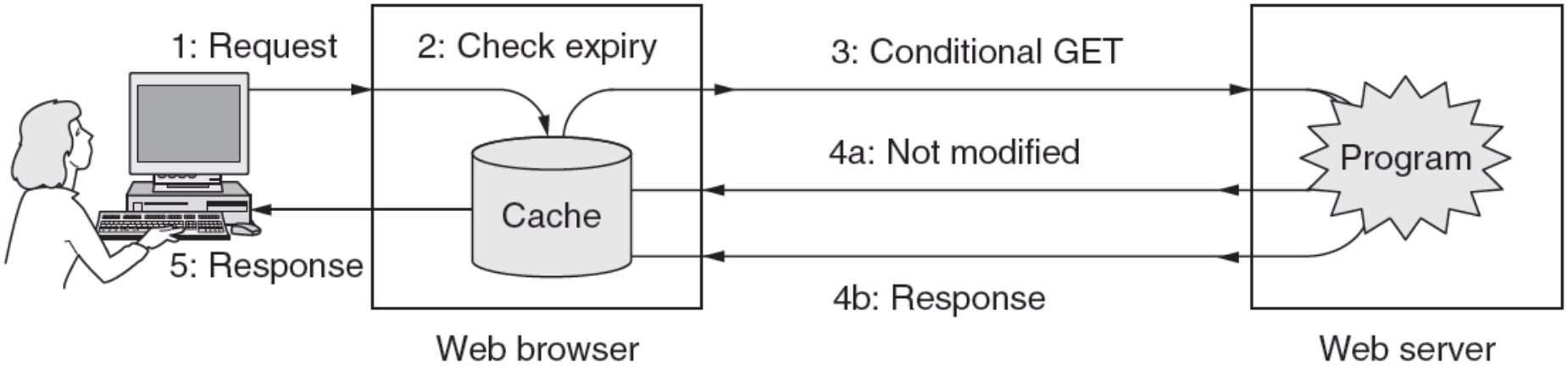
Some HTTP message headers.

The HyperText Transfer Protocol

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
Content-Range	Response	Identifies a portion of the page's content
Last-Modified	Response	Time and date the page was last changed
Expires	Response	Time and date when the page stops being valid
Location	Response	Tells the client where to send its request
Accept-Ranges	Response	Indicates the server will accept byte range requests
Date	Both	Date and time the message was sent
Range	Both	Identifies a portion of a page
Cache-Control	Both	Directives for how to treat caches
ETag	Both	Tag for the contents of the page
Upgrade	Both	The protocol the sender wants to switch to

Some HTTP message headers.

The Hypertext Transfer Protocol



HTTP caching.

End

Chapter 7