

The World Wide Web

- See “Computer Networks, 4th Ed.” by Tanenbaum, pages 611—673



1

The World Wide Web (WWW)

- Architectural Framework of Linked Documents
- Documents contained on millions of heterogeneous computers
- Originated at CERN in 1989 due to a need for scientists to share valuable data (reports, blueprints, documents etc) collected internationally from complex experiments
- Popularity, ease of use, and aesthetic quality led to rapid growth
- In 1994, CERN and MIT set up the *World Wide Web Consortium (W3C)* to promote standards and interoperability – it now has many more members

2

Web Documents

- Web Documents are commonly referred to as *Web pages* or *pages*
- Pages have a text based structure which may contain *links* to other documents called *hyperlinks*
- *Hypertext* is notion of pages pointing (i.e. containing links) to other pages (more details later)
- Viewing pages and navigating links is done using a *browser*
- Mosaic was the first browser developed by Mark Andreessen, who later founded Netscape Communications Corporation
- Current layout engines: Gecko (Mozilla Firefox), KHTML/WebKit (Google Chrome, Apple Safari), Trident (Microsoft Internet Explorer), Presto (Opera)

3

Web Browsers

- Web browsers are programs that retrieve requested pages, interpret the content, and present it to the user appropriately
- Browsers can present the data in different formats e.g. text-only, graphics, audio.

```
<html>
<head>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<title>Department of computer science @ University college</title>
<meta name="microsoft border" content="none">
</head>
<!--
-->
```



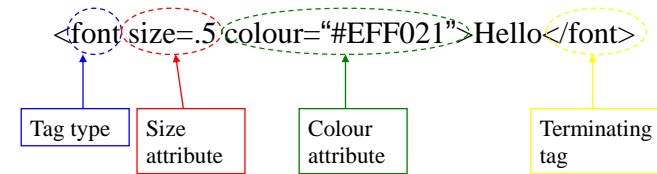
Web Page Structure

- **Q:** How is it possible for many different browsers to open and present these documents in the same manner?
- **Q:** Also, web pages are based on simple text files, so how do they present such rich content e.g. graphics, frames?
- **A:** They use a standardized *Markup Language* that describes these features to the browser and indicates how it should present them to the user
 - There are many mark-up languages e.g. HTML 1.0 to HTML 4.0, XHTML, XML etc
 - *HyperText Markup Language (HTML)*
 - *eXtensible Markup Language (XML)*
- More on HTML at <http://www.w3.org> or <http://www.htmlgoodies.com/primers/basics.html>

5

HTML

- Use *Tags* to describe content
- Tags have a Name or Type and can (optionally) have attributes and most (or all in XHTML and XML) have terminating tag
- Example:



6

Page

```
<html>
  <head>
    <title>Sample page</title>
  </head>

  <body>
    <p>First paragraph
  </body>
```

7

Common HTML Tags

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h n> ... </h n>	Delimits a level n heading
 ... 	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
 ... 	Brackets an unordered (bulleted) list
 ... 	Brackets a numbered list
 ... 	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
	Displays an image here
 ... 	Defines a hyperlink

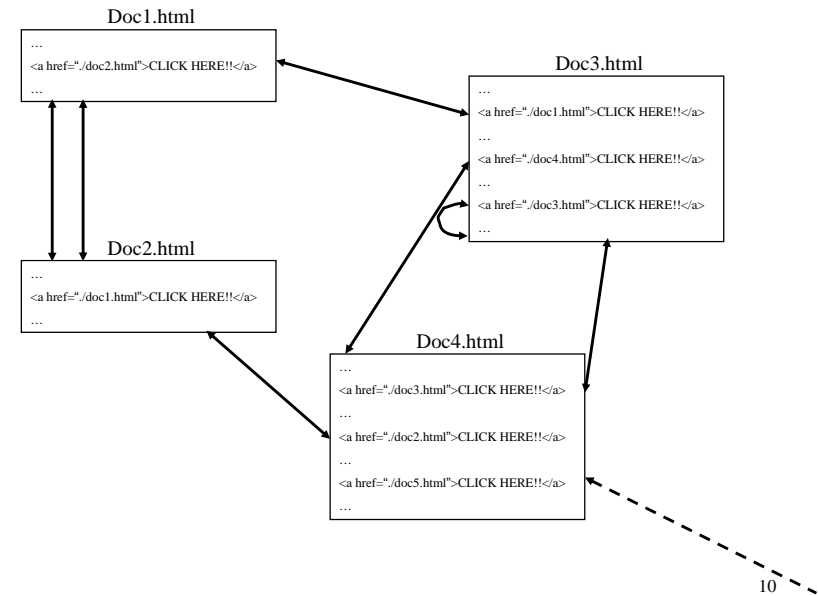
8

Hyperlinks

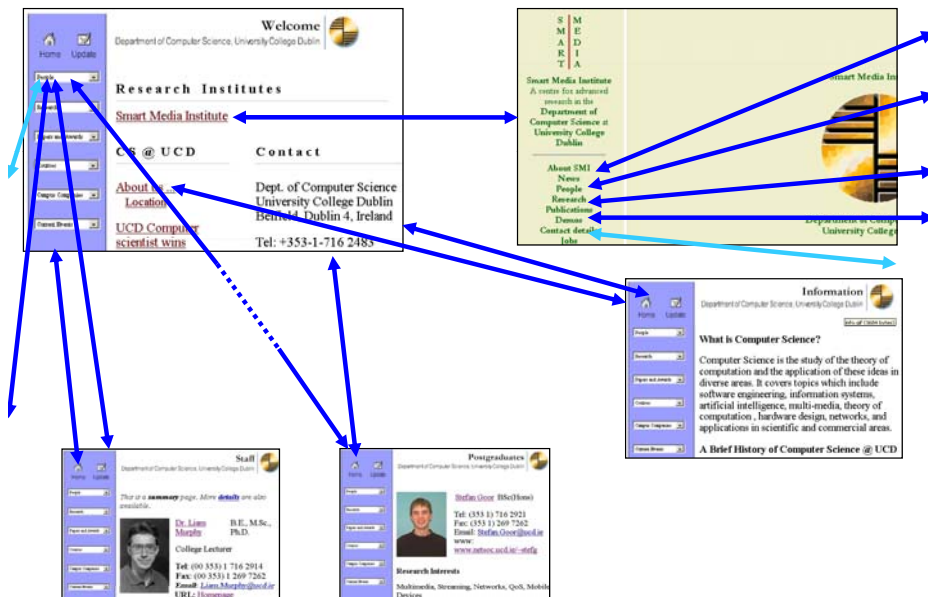
- A hyperlink is used to make a reference between some text, picture, etc. and another Web document
- Browsers usually presents hyperlinks in a different colour and underlined so the user can distinguish them from plain text
- When a hyperlink is clicked the linked document is displayed
- HTML hyperlink example:

```
<a href="http://www.ucd.ie">Click here for UCD</a>
```
- The link specifies the resource to link to by means of a *Uniform Resource Locator (URL)*

9



10



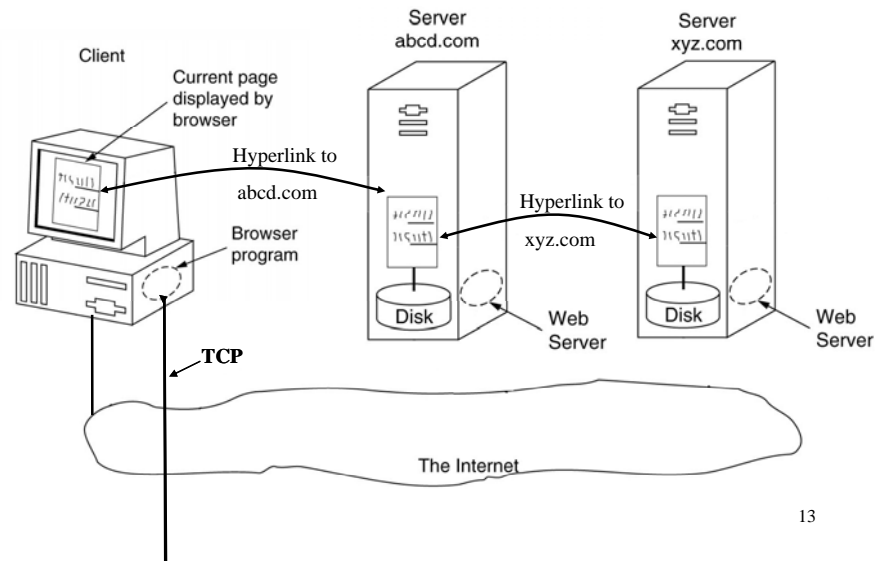
11

Client Side (Browser Side)

- Client Process for Retrieving a Web Page:
 1. Determine **URL** of requested page
 2. Get IP address of URL from **DNS**
 - A DNS (Domain Name System) resolves known URLs to IP addresses
 3. Make a **TCP connection** on **Port 80** to the IP address
 4. Send a request asking for the requested page e.g. /index.html
 5. Receive the page from the server
 6. TCP connection released
 7. Browser interprets and displays the content of the file & then retrieves any images/files contained in the page

12

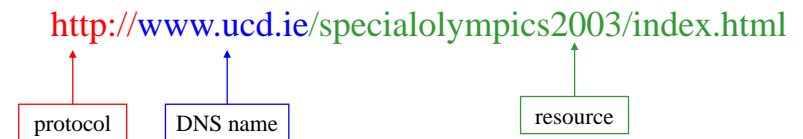
Hyperlink Web Model



13

Uniform Resource Locator

- Documents on the Web are named by URLs
- A URL has 3 essential components:
 - The name of the *Protocol* (http for www documents)
 - The DNS name of the machine where the page resides
 - The resource
- Example:



14

Common Protocols & URLs

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

15

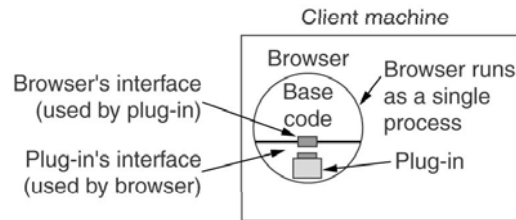
Handling Multiple File Types

- Browser can open more than just HTML files and graphics by use of either *plug-ins* or *helper applications*
- Determining what to do with web documents can be determined using the *MIME type* or file extension
- Plug-ins are extensions to the browser that enable it to interpret and present more content e.g. Flash, QuickTime & Shockwave
- Helper Applications are programs that are not part of the browser but that can be invoked by it whenever a file requires the use of an external program e.g. Adobe Acrobat Reader, MSWord

16

Plug-ins

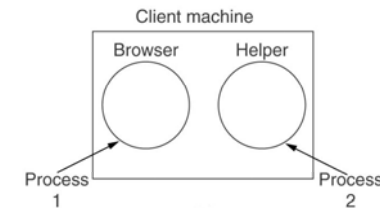
- A plug-in resides within the browser and can be used whenever needed
- When called they can access the content of the page to be viewed and change or control it accordingly
- Plug-ins are only loaded when in use and discarded from memory when finished
- Plug-ins provide an interface for the browser to communicate with it and vice versa.



17

Helper Applications

- Another program residing on the same computer but requiring the running of a distinct process from that of the browser
- When a browser receives content it (or its plug-ins) cannot interpret, it will look at the MIME type or file extension.
- If another program is available on the computer to handle the file, it is invoked



18

Web Servers

- A Web Server is a computer connected to the Internet that contains web documents and supports the HTTP protocol for document requests and delivery
- Servers are identified by an IP address (e.g. 137.43.1.49) or hostname (e.g. www.ucd.ie) specified in the URL

19

Server Side

- In basic terms a server operates as follows
 1. Accept a TCP connection from client
 2. Look at name of the resource requested
 3. Retrieve/construct the resource
 4. Serve / deliver the resource back to the client
 5. Release the TCP Connection
- However, this a very naïve view of a server's responsibilities – it tends to be more complicated in practice

20

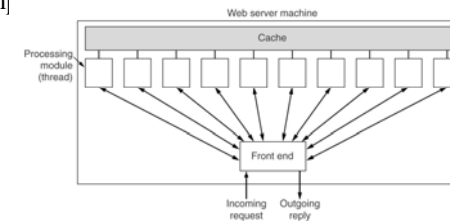
Server Side

- More Realistic Server Transaction
 1. Resolve name of Web page requested
 2. Verify the client that requested the page
 3. Check that the client has valid access
 4. Ensure that the file is permitted for viewing
 5. Check if the page resides in cache (memory)
 6. Get the page from the disk (only if necessary)
 7. Determine the MIME type and include it in the response
 8. Miscellaneous other tasks e.g. stats, profiles etc
 9. Respond with this data to the client
 10. Log the transaction in the *Server Log*

21

Server Issues

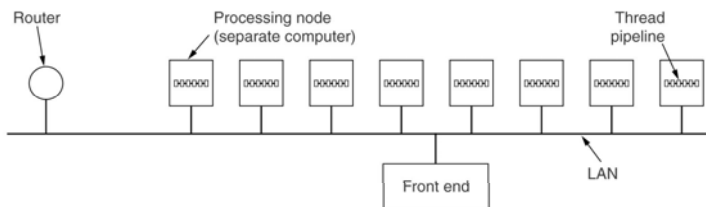
- Problem:
 - Every request consumes computational resources (CPU, memory, disk) so the server is restricted to a maximum number of accesses per second, and therefore limited to the number of clients it can serve
- Solution(s):
 - Use large amounts of memory to cache the pages so there is no (or minimal) disk access
 - Use Multi-threading, so each request has its own thread to handle its response
 - Emj are different!!)



22

Server Issues (cont'd)

- Server Farm
 - Even high spec servers may get overloaded at peak times or during special events
 - Many server processes on many processors (or computers)
 - Attempts to reduce load on each computer
 - *Problem* – no shared cache between processors
 - *Solution* – Use front end to monitor what processors have which parts of the web site



3

Server Replication

- a.k.a. Mirroring
- Same content held on numerous different servers to provide load sharing and geographical enhanced quality
- Often user is given a list of links to choose the most appropriate server (*static*)
- This can lead to *flash crowds*
- To avoid this, dynamic replicas can be used to manage load on servers
- Problem with URLs in replication
- Proposed Solution: URNs (Universal Resource Names, see RFC 2141)

24

Statelessness and Cookies

- The Web is stateless
- No concept of sessions or login, simple request and receive process
- Not appropriate for some Web purposes e.g. online stores, subscription services
- **Cookies** can be employed to aid these problems
- Cookies are files that reside on the client's computer and can be used by the server to gather information about the user
- Cookies can be *persistent* or *nonpersistent*

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

Criticisms of Cookies

- They are simply static files or strings so don't contain viruses, etc. but...
- Many users on a PC – wrong cookies used
- Intrusive and hidden Cookies
- Can be hacked and used maliciously
- Now many browser include security options when cookies are created

26

Static Web Documents

- Static Web documents are simply files that are sitting on a server and can be retrieved as-is
- HTML is used to create static Web pages, although they can have dynamic content specified within them

27

Collection of Data

- [HTML v2.0+](#) allows use of *Forms*
- Forms provide a method of collecting data from the user such as names, addresses, answers to questions, etc.
- They can accept response in the form of text input, radio buttons, etc.
- Commonly they have a submit button which sends the input to an interface on the server that processes the entries and responds appropriately

28

Data / Information Extraction

- HTML provides information about **formatting** but not **what** information is contained in the page
- Consider a pages that give information about peoples' record collections
- Some pages may present it in tables, others in lists, etc.
- If we need to be able to determine the song name, the artist, etc, we might write some program to parse the HTML – but it will have to consider each layout individually
- How could this be done generically to parse all possible presentation formats?

29

eXtensible Markup Language

- XML provides a method of identifying information by use of tags
- These can be nested and defined to suit particular requirements

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="b5.xsl"?>
<book_list>
  <book>
    <title> Computer Networks, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2003 </year>
  </book>
  <book>
    <title> Modern Operating Systems, 2/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2001 </year>
  </book>
  <book>
    <title> Structured Computer Organization, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 1999 </year>
  </book>
</book_list>
```

30

XML Continued

- Structuring like this makes parsing of data very easy, as every bit of data is identified as a specific piece of information
- BUT...
- If we only have XML and we use a browser, all we get is a mass of structured text
- So, what about all the nice colours, graphics and formatting that's available in HTML?

31

eXtensible Style Language

- XSL provides a solution to the previous problem
- It defines how to represent the information contained in an XML document in a manner similar to HTML Web pages
- Allows designers to specify browser formatting more accurately e.g. can set font-size, colour,... instead of <h1> etc

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/">
    <html>
      <body>
        <table border="2">
          <tr>
            <th> Title</th>
            <th> Author</th>
            <th> Year </th>
          </tr>
          <xsl:for-each select="book_list/book">
            <tr>
              <td> <xsl:value-of select="title"/> </td>
              <td> <xsl:value-of select="author"/> </td>
              <td> <xsl:value-of select="year"/> </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

32

XML + XSL

- Used together XML and XSL provide a method to present and deliver information in a query compatible manner
- XML and XSL are customisable to requirements e.g. CML – Cheese Markup Language used for trading cheese between suppliers, distributors
- Implementation of this is widespread but mainly between businesses or associated communities
- HTML has ruled for a long time and won't be overthrown easily because of its familiarity
- W3C and Web design communities dispute the two
- As a compromise *XHTML* has been established

33

XHTML

- Differences between HTML 4.0 & XHTML 1.0
 1. Standard is *strict* and all pages must obey it
 2. All tags and attributes must be lowercase e.g. <HTML> or <Html> becomes <html>
 3. All tags require a closing tag, even previously unclosed tags e.g.
 becomes <br ... />
 4. All attribute values must be within quotes e.g. becomes
 5. Tags must be properly nested e.g. <center>blah</center> is not valid
 6. Every document must specify its [document type](#)

34

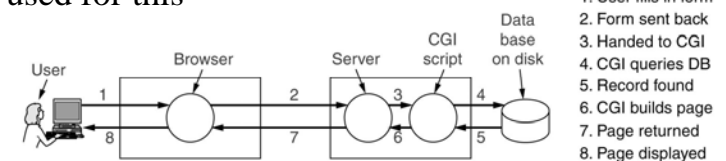
Dynamic Web Documents

- Pages generated on demand, rather than just retrieved from disk or memory
- Pages can be generated either on client or server side
- Provide greater functionality of Web documents

35

Server-Side Page Generation

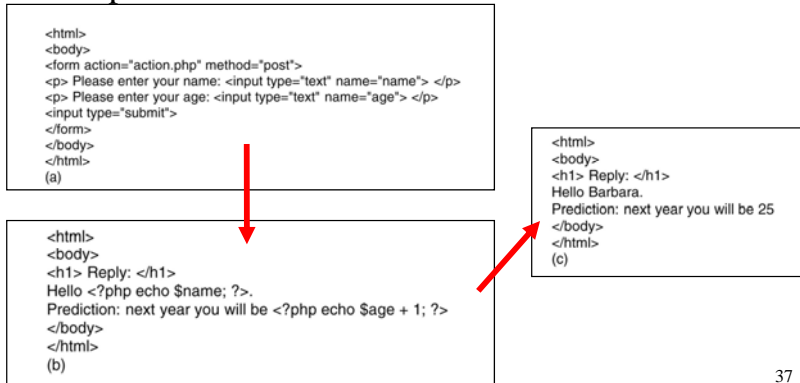
- When a form is used on a web page the server has to do something with the input
- Usually when the 'Submit' button is pressed the values are passed to a program on the server that takes them as arguments and deals with them appropriately
- Traditionally *CGI – Common Gateway Interface* is used for this



36

PHP Scripts

- **PHP**: Hypertext Pre-processor
- Same as HTML but includes some tags that require processing on the server before sending a response to the client



37

Other Scripts

- **JSP** – *Java Server Pages* by Sun/Oracle based on the Java Programming Language
- **ASP** – *Active Server Pages* by Microsoft
- Both of these are proprietary, while **PHP** is open source
- They all provide a similar type of service

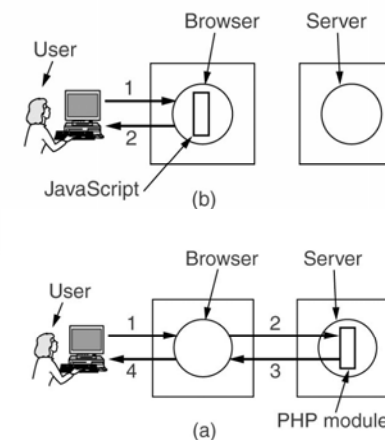
38

Client Side Page Generation

- Similar to the idea of PHP, except that the processing of the script happens on the client side
- **JavaScript** – not the same as Java!!
- **Applets** – Java RE based
- Allow more complicated features to be displayed in a Web page e.g. drop down boxes, dynamic images or graphs
- Again they are simply inserted in the HTML

39

Client vs Server Page Generation



40

HTTP Protocol

- HyperText Transfer Protocol
- Specifies what message a client can send to a server and what type of response they should get
- Use **TCP Connection on Port 80** so no loss, duplication or acknowledgement handling requirement on clients and servers – these issues are handled at a lower level
- **Persistent Connection**

HTTP Methods

- Methods are used so that more than just page retrieval is possible
- Method example: GET filename HTTP/1.1

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

HTTP Responses

- Each client request is replied to with a status line that contains a 3-digit response code (and possibly more information)

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

HTTP Headers

- Headers are like arguments that can be sent in requests or responses: **Request Headers** and **Response Headers**

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

Web Performance Enhancements

- Caching
- Proxy
- Server Replication
- CDN – Content Delivery Network

45

The Wireless Web

- WAP
- I-Mode
- Second Generation Wireless Web

46

WAP

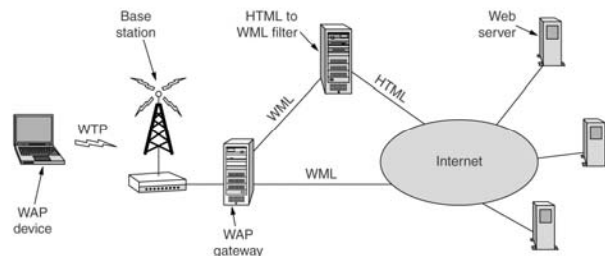
- *Wireless Application Protocol*

Combination of mobile devices and the Internet

Optimised for low bandwidth connections, slow CPUs, little memory and small screens

Uses WML (Wireless Markup Language, XML application) not HTML!

WAP was expensive, over-hyped and small screens meant it was not a great success



47

2nd-Generation Wireless Web

- Most future networks are expected to be packet switched e.g. GPRS, 3G, CDMA
- WAP is compatible with both packet and circuit switched networks
- WAP 2.0 supports the old protocol stack and also the standard Internet stack, an altered version of TCP and HTTP/1.1
- WAP 2.0 supports XHTML Basic, which NTT DoCoMo has also agreed to support

48