



Nivelul Aplicație

World Wide Web

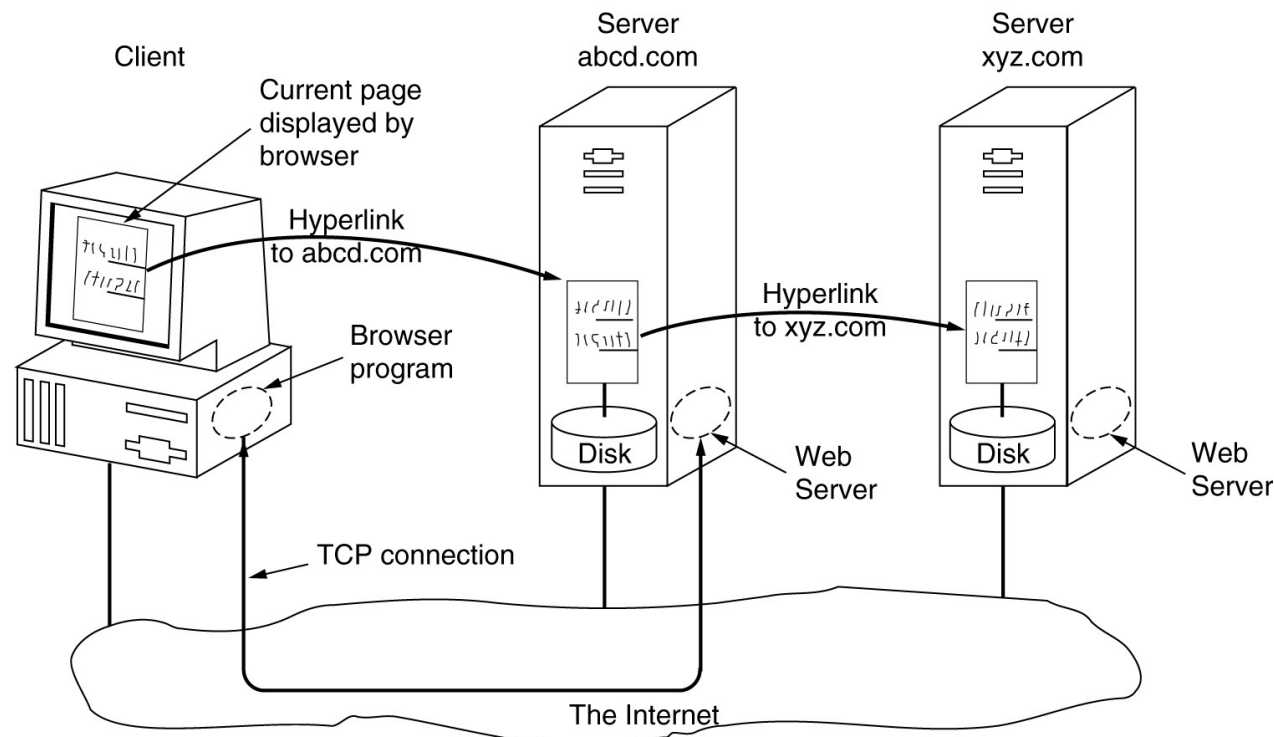


Cuprins

- Funcționarea WWW
- URL
- HTML – marcaje, formulare
- HTTP
- Clientul (Browser)
- Serverul

World Wide Web

- Set de documente (pagini) cu legături între ele (hyperlinks)
- Distribuite pe mașini diferite
- Include o pagină de referință (home page)
 - pagina inițială a unui site Web
 - pagina afișată la pornirea unui browser



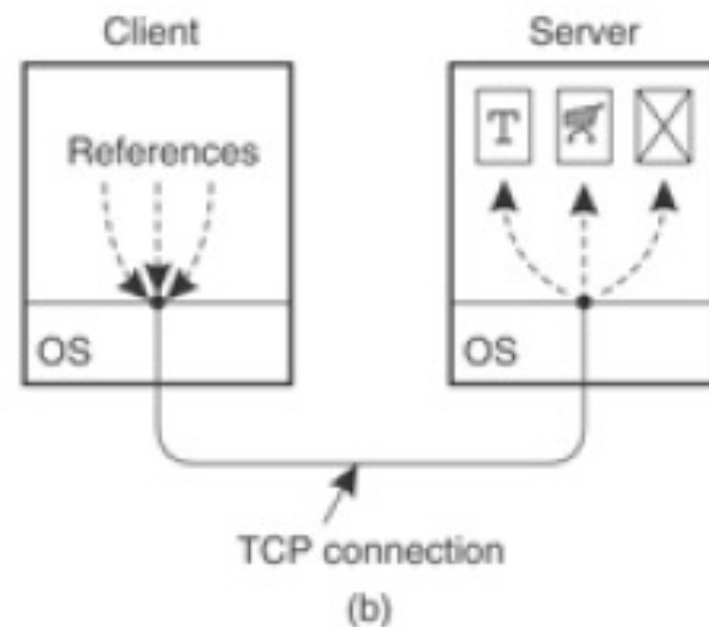
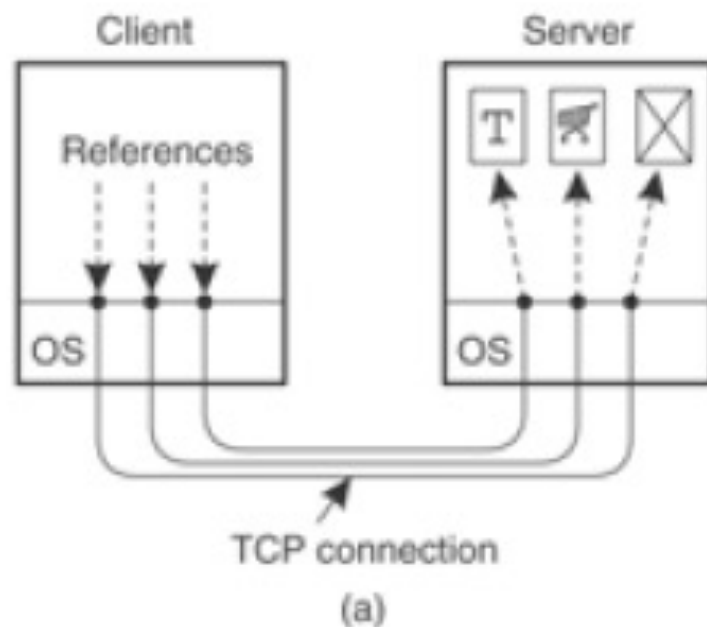


Interacțiunea client – server (reluare!)

- Browser - determină URL
- Browser - cere DNS-ului adresa IP pentru `www.w3.org`
 - DNS - răspunde cu `18.23.0.23`
- Browser - deschide o conexiune TCP la port 80 pe `18.23.0.23`
- Browser - trimite o comandă
`GET /hypertext/www/TheProject.html`
 - Server `www.w3.org` - trimite fișierul `TheProject.html`
- Conexiunea TCP este închisă
- Browser - afișează conținutul din `TheProject.html`
- Browser - extrage și afișează toate imaginile din `TheProject.html` (se deschide o nouă conexiune TCP pentru fiecare imagine)

Conexiuni persistente

- Introduse în HTTP 1.1
- O singură conexiune persistentă poate fi folosită pentru mai multe cereri-răspunsuri
- Cererile pot fi transmise și în “pipeline” (fără a aștepta răspunsurile)





Trei elemente de bază

- O schemă de adresare a documentelor în Internet (**URL – Uniform Resource Locator**)
- Un limbaj de formatare a documentelor (**HTML – HyperText Markup Language**)
- Un protocol pentru transportul mesajelor specializate prin rețea (**HTTP – HyperText Transfer Protocol**)



URL – Uniform Resource Locator

scheme://host[:port#]/path/.../[/;url-params][?query-string][#anchor]

scheme	protocol (http, ftp etc.)
host	nume / adresa IP a serverului Web
port#	număr port server Web (80 pentru http)
path	calea de la rădăcina serverului la document
url-params	pentru identificarea sesiunii
query-string	valori din formular HTML
anchor	referință la un marcaj pozițional din document

exemplu

<http://www.situlmeu.ro/cv/test;id=8079?name=valentin&x=true#aici>



Câteva URL-uri obișnuite

Schema	Utilizat pentru	Exemple
http	Hipertext (HTML)	http://www.cs.vu.nl/~ast
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
File	Fișier local	file:///usr/suzanne/prog.c
news	Grup de știri	news:comp.os.minix
news	Articol de știri	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/libraries
mailto	Trimitere de poșta electronică	mailto:JohnUser@acm.org
telnet	Conectare la distanță	telnet://www.w3.org:80



HTML - HyperText Markup Language

Structura unei pagini

```
<html>
  <head>
    <title>
      Prima incercare
    </title>
  </head>
  <body>
    Prima incercare: Nu este greu sa
    construisti un text urat in html,
    mai complicat este sa construisti
    un text care sa arate bine.
  </body>
</html>
```

Ce afișează browser-ul:

**Prima încercare: Nu este greu să
construiești un text urât în html, mai
complicat este să construiești unul care
să arate bine.**



O selecție de marcaje uzuale

Marcaj	Descriere
<code><html> ... </html></code>	Delimitează textul scris în HTML
<code><head> ... </head></code>	Delimitează zona de antet
<code><title> ... </title></code>	Definește titlul (nu este afișat de programul de navigare)
<code><body> ... </body></code>	Delimitează zona de corp
<code><hn> ... </hn></code>	Delimitează un titlu de nivel n
<code> ... </code>	Text îngroșat
<code><i> ...</i></code>	Text cursiv
<code><center> ... </center></code>	Centrat pe orizontală
<code>
</code>	Trecere la linie nouă
<code><p></code>	Început de paragraf
<code> ... </code>	Delimitează o listă neordonată
<code> ... </code>	Delimitează o listă ordonată (numerotată)
<code> ... </code>	Delimitează un element într-o listă ordonată sau neordonată
<code><hr></code>	Linie orizontală
<code></code>	Afișează o imagine în acest loc (sau text-ul specificat)
<code>text</code>	Hiper-legătură la o pagină
<code>text</code>	Declară o ancoră într-un document

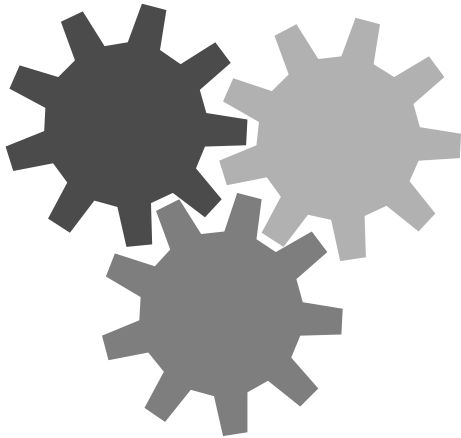


HTML – un exemplu

```
<html>
<head> <title> AMALGAMATED WIDGET, INC. </title></head>
<body> <h1> Welcome to AWI's Home Page </h1>
<img SRC="http://www.widget.com/images/logo.gif" ALT="AWI Logo"> <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's</b>
home page. We hope <i> you </i> will find all the information you need here.
<p> Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by FAX. </p>
<hr>
<h2> Product Information </h2>
<ul>
  <li> <a href="http://widget.com/products/big" > Big widgets </a>
  <li> <a href="http://widget.com/products/little" > Little widgets </a>
</ul>
<h2> Telephone Numbers </h2>
<ul>
<li> 1-800-WIDGETS
<li> 1-415-765-4321
</ul>
</body>
</html>
```

Pagina formatată

Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope *you* will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

Product Information

- [Big widgets](#)
- [Little widgets](#)

Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321



Formulare – marcaje specifice

element HTML	Parametri	Semnificație
<INPUT>, TYPE=text	NAME, SIZE, MAXLENGTH	câmp de intrare (tipul implicit)
<INPUT>, TYPE=radio	NAME, VALUE	buton radio
<INPUT>, TYPE=checkbox	NAME, CHECKED	casetă de selecție
<INPUT>, TYPE=password	NAME, SIZE, MAXLENGTH	câmp de parolă
<INPUT>, TYPE=reset sau submit		buton de acțiune
<INPUT>, TYPE=image	NAME, ALIGN, SRC	hartă (image) activă
<INPUT>, TYPE=hidden	NAME,	element ascuns
<SELECT>	NAME, OPTION, MULTIPLE	listă de selecție
<TEXTAREA>	NAME, COLS, ROWS, WRAP	zonă de editare



Formulare – un exemplu

```
<html>
```

```
<head><title> AWI CUSTOMER ORDERING FORM </title></head>
```

```
<body>
```

```
➔ <h1> Widget Order Form </h1>
```

```
<form ACTION="http://widget.com/cgi-bin/widgetorder" method=POST>
```

```
➔ <p> Name <input name="customer" size=46> </p>
```

```
➔ <p> Street Address <input name="address" size=40> </p>
```

```
➔ <p> City <input name="city" size=20> State <input name="state" size=4> Country  
    <input name="country" size=10> </p>
```

```
➔ <p> Credit card # <input name="cardno" size=10> expires <input name="expires"  
    size=4> M/C <input name="cc" type=radio value="mastercard"> VISA <input  
    name="cc" type=radio value="visacard"> </p>
```

```
➔ <p> Widget size Big <input name="product" type=radio value="expensive"> Little  
    <input name="product" type=radio value="cheap"> Ship by express courier  
    <input name="express" type=checkbox> </p>
```

```
➔ <p> <input type=submit value="Submit order"> </p>
```

```
Thank you for ordering an AWI widget, the best widget money can buy!
```

```
</form>
```

```
</body>
```

```
</html>
```



Widget Order Form



Name



Street address



City State Country



Credit card # Expires M/C ☐ Visa ☐



Widget size Big ☐ Little ☐ Ship by express courier ☐



Thank you for ordering an AWI widget, the best widget money can buy!

Formulare

Un text cu informațiile completate de utilizator

Widget Order Form

Name

Street address

City State Country

Credit card # Expires M/C ☒ Visa ☐

Widget size Big ☐ Little ☒ Ship by express courier ☒

Thank you for ordering an AWI widget, the best widget money can buy!

customer=John+Doe&address=100+Main+St.&city=White+Plain&
state=NY&country=USA&cardno=1234567890&expires6/98&cc=mastercard&
product=cheap&express=on

(împărțit aici în trei linii din motive de aliniere in pagină)



HTTP

- Protocol “**stateless**”
- Folosește paradigma **request/response**
 - clientul și serverul comunică direct sau prin proxy-uri
 - structura mesajelor:
 - linia de comandă / răspuns
 - linii de antet
 - linie blank
 - corp mesaj

Structura mesaj **request**

```
METHOD /path-to-resource  
      HTTP/version-number  
Header-name-1: value  
Header-name-2: value  
...  
[ optional request body ]
```

Exemplu

```
GET /sj/index.html HTTP/1.1  
Host: www.mywebsite.com
```

Structura mesaj **response**

```
HTTP/version-number status-code message  
Header-name-1: value  
Header-name-2: value  
...  
[ response body ]
```

Exemplu

```
HTTP/1.1 200 OK  
Content-Type: text/html  
Content-Length: 9934  
...  
<HTML>  
<HEAD> ... </HEAD> ...  
... </HTML>
```




Metode HTTP

Metoda	Descriere
GET	Cerere de citire a unei pagini Web
HEAD	Cerere de citire a antetului unei pagini de Web
POST	Adăugarea la resursa specificată (de exemplu o pagină de Web)
PUT	Cerere de memorare a unei pagini de Web
DELETE	Ștergerea unei pagini de Web
TRACE	Transmite în ecou cererea care a sosit
OPTIONS	Interogarea anumitor opțiuni
CONNECT	Folosit pt. conectare prin proxy server pe conexiune tunel



Exemplu GET

- Formular HTML

<HTML>

<HEAD><TITLE>Formular
simplu</TITLE></HEAD>

<BODY>

<H2>Formular simplu</H2>

<FORM ACTION="http://financiar.yahoo.com/q"
METHOD="get">

Ticker: <INPUT SIZE="25" NAME="s">

<INPUT TYPE="submit" VALUE="Get Quote">

</FORM>

</BODY>

</HTML>

Formular simplu

Ticker:

URL construit de browser pentru intrarea
YHOO

http://financiar.yahoo.com/q?

Cerere HTTP

GET /q?s=YHOO HTTP/1.1

Host: financiar.yahoo.com

User-Agent: Mozilla/4.75 [en]



Răspuns

HTTP/1.1 200 OK

Date: Sat. 03 May 2005 17:48:35 GMT

Connection: close

Content-Type: text/html

Set-Cookie: B=dfaosiu534qjnfretk&b=2;expires=Thu, 15
Aug 2011 20:00:00 GMT; path=/; domain=.yahoo.com

<HTML>

<HEAD><TITLE>Yahoo! financiar - YHOO</TITLE></HEAD>

<BODY>

...

</BODY>

</HTML>



Exemplu POST

- Aceeași cerere, formulată cu metoda POST

```
POST /q HTTP/1.1
```

```
Host: financiar.yahoo.com
```

```
User-Agent: Mozilla/4.75 [en] (WinNT; U)
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 6
```

```
s=YHOO
```

- Răspunsul este identic



Exemplu HEAD

Cerere

HEAD http://www.cs.pub.ro/~ionescu/ HTTP/1.1

Host: www.cs.pub.ro

User-Agent: Mozilla/4.75 [en] (WinNT; U)

Răspuns

HTTP/1.1 200 OK

Date: Mon, 05 Feb 2005 04:33:19 GMT

Server: Apache/1.2.5

Last-Modified: Mon, 05 Feb 2005 04:30:19 GMT

Content-Length: 2234

Content-Type: text/html

Coduri de stare

Cod	Semnificație	Exemple
1xx	Informație	100 = serverul acceptă continuarea tratării cererii de la client (asociat cu un antet Expect din cerere)
2xx	Succes	200 = cerere reușită; 204 = nu există conținut
3xx	Redirectare	301 = pagină mutată definitiv; 302 = pagină mutată temporar; 304 = pagina din memoria ascunsă este încă validă
4xx	Eroare la client	400 = cerere incorectă; 401 = ne-autorizat 403 = interzis 404 = pagina nu a fost găsită
5xx	Eroare la server	500 = eroare internă la server; 501 = ne-implementat 503 = încearcă mai târziu



Antete Mesaje HTTP

Antet	Tip	Descriere
User-Agent	Cerere	Informație asupra programului de navigare și a platformei
Accept	Cerere	Tipul de pagini pe care clientul le poate trata
Accept-Charset	Cerere	Seturile de caractere care sunt acceptabile la client
Accept-Encoding	Cerere	Codificările de pagini pe care clientul le poate trata
Accept-Language	Cerere	Limbajele naturale pe care clientul le poate trata
Host	Cerere	Numele DNS al serverului (folosit pentru virtual hosting)
Authorization	Cerere	O listă a drepturilor clientului
Cookie	Cerere	Trimite (la server) un cookie setat anterior
Set-Cookie	Răspuns	Serverul vrea să salveze un cookie la client
Server	Răspuns	Informație despre server (ex. Server: Apache/1.2.5)



Antete Mesaje HTTP (2)

Antet	Tip	Descriere
Content-Encoding	Răspuns	Cum este codat conținutului (de exemplu, gzip)
Content-Length	Răspuns	Lungimea paginii în octeți
Content-Type	Răspuns	Tipul/subtipul MIME al paginii
Last-Modified	Răspuns	Ora și data la care pagina a fost ultima dată modificată
Location	Răspuns	O indicație pentru client pentru redirectarea cererii
Accept-Ranges	Răspuns	Serverul va accepta cereri în anumite limite de octeți
Date	Ambele	Data și ora la care mesajul a fost trimis
Connection	Ambele	Intenția de a păstra sau nu conexiunea (ex. Connection: Close)



Antete referitoare la tipul conținutului

- Sistem de tipuri împrumutat din MIME (Multipurpose Internet Mail Extensions)
- Două niveluri (reprezentate de două antete în răspuns)
 - Content-Encoding
 - gzip (GNU zip)
 - compress (UNIX)
 - deflate (zlib format definit în RFC 1950 și 1951)
 - Content-Type
 - Tip, subtip și (opțional) perechi *atribut = valoare*
 - *Exemple*
`Content-Type: text/plain; charset = 'us-ascii'`
`Content-Type: text/xml`
`Content-Type: application/pdf`
`Content-Type: video/x-mpeg`



Exemplu mesaje multipart

Cerere

```
GET /cgi-bin/doit.cgi HTTP/1.1
Host: cgi-bin.netscape.com
Date: Sun, 18 Feb 2004 06:22:33 GMT
```

Răspuns

```
HTTP/1.1 200 OK
Server: Netscape-Enterprise-3.6 SP1
Date: Sun, 18 Feb 2004 06:22:35 GMT
Content-Type: multipart/x-mixed-replace;
    boundary="ThisRandomString"
```

```
--ThisRandomString
Content-Type: image/gif
```

...

```
--ThisRandomString
Content-Type: image/gif
```

...

```
--ThisRandomString
Content-Type: image/gif
```

...



Antete pentru control caching

Trei tipuri de caching:

- la client – cache privat
- la proxy, server – cache-uri partajate

Control caching – introdus în HTTP/1.1

- se face de server prin antet **Cache-Control** cu valorile
 - **public** - nici o restricție pentru caching
 - **private** – nu în *shared caches*
 - **no-cache** – nici în browser, nici în proxy

Exemplu

HTTP/1.1 200 OK

Date: Mon, 05 Feb 2005 04:33:19 GMT

Server: Apache/1.2.5

Last-Modified: Mon, 05 Feb 2005 04:30:28 GMT

Cache-Control: private

Content-Length: 2289

...



Consistența cache-urilor (1)

- Asigură că documentul din cache este același cu cel din server
- **Soluție 1:** Folosind comanda **HEAD**
 - clientul transmite **HEAD**
 - primește răspuns și verifică antet **Last-Modified**
 - transmite **GET** dacă document din server este mai nou decât copia din cache

- **Cerere**

HEAD http://www.cs.pub.ro/~ionescu/ HTTP/1.1

Host: www.cs.pub.ro

User-Agent: Mozilla/4.75 [en] (WinNT; U)

- **Raspuns**

HTTP/1.1 200 OK

Date: Mon, 05 Feb 2005 04:33:19 GMT

Server: Apache/1.2.5

Last-Modified: Mon, 05 Feb 2005 04:30:19 GMT

. . .



Consistența cache-urilor (2)

- **Soluție 2:** Folosind comanda GET cu antet **If-Modified-Since**
GET /~ionescu/ HTTP/1.1
Host: www.cs.pub.ro
If-Modified-Since: Mon, 04 Feb 2005 04:30:28 GMT
- serverul transmite
HTTP/1.1 304 **Not Modified**
Date: Mon, 05 Feb 2005 04:33:19 GMT
Server: Apache/1.2.5
- sau
HTTP/1.1 200 OK
. . .
Last-Modified: Mon, 05 Feb 2005 04:30:28 GMT
Content-Length: 2289



Soluție pentru performanță

- Clientul nu contactează serverul pentru orice cerere
 - Răspunsul unui server poate include **data expirării**, care este memorată de client

```
HTTP/1.1 200 OK
Date: Mon, 05 Feb 2005 04:33:20 GMT
. . .
Cache-Control: private
Expires: Tue, 06 Feb 2005 04:33:20 GMT
Last-Modified: Mon, 05 Feb 2005 04:33:18 GMT
```
 - Clientul verifică existența paginii în cache
 - **Nu există** – cere resursa necondiționat
 - **Există expirată** - adaugă la cerere antet If-Modified-Since
 - dacă serverul răspunde cu **304 Not Modified**, folosește intrarea din cache
 - **Există ne-expirată** – folosește intrarea din cache



Autentificare și autorizare

- Autentificare de bază
 - permite accesul la **pagini protejate**
 - prin antet **de autorizare**
 - **nume** și **parolă** transmise **codat Base64** (nu criptat)
 - atenție, trebuie folosit HTTPS
- **Secvența de acțiuni**
 - Clientul cere resursa restricționată
 - Serverul răspunde cu 401

HTTP/1.1 401 Authenticate

Date: Mon, 05 Feb 2005 04:33:19 GMT

Server: Apache/1.2.5

WWW-Authenticate: Basic realm="Capitol13"

- **realm** definește domeniul protejat



Autentificare și autorizare (2)

- Browserul retrimite cererea cu antet suplimentar de autorizare

`GET /carte/capitol3/index.html HTTP/1.1`

`Date: Mon, 05 Feb 2005 04:33:20 GMT`

`Host: www.cs.pub.ro`

`Authorization: Basic eNCoDEd-userID:PaSSwoRd`

- Serverul verifică prudențialele de autorizare și satisface cererea (sau refuză cu 403)
- Odată trimise prudențialele, browserul retrimite automat antetul de autorizare și credențialele în viitoarele cereri la **URL dependente** (fișiere din **subdirectoare**)

Ex. `http://cs.pub.ro/~popescu/clase/`

depinde de `http://cs.pub.ro/~popescu/`

care este un **prefix** al primului



Suport sesiune

Cookie este un mecanism ce permite transmiterea unor informații de stare prin mesaje HTTP

- ex. info stare - identificatorul unei sesiuni
- Înțelegerea este inițiată de server prin antet Set-Cookie

Set-Cookie: <nume>=<valoare>[; expires=<data>]

[;path=<cale>] [;domain=<nume_domeniu>][; secure]

<nume>=<valoare> pereche atribut/valoare de trimis înapoi de browser

path, domain identifică cererile care sunt calificate

- pentru domeniul **.pub.edu** domeniile **calificate** au forma ***.pub.edu**
- pentru calea **/test/** caile **calificate** sunt de forma **/test/***

secure browser-ul trebuie să transmită info pe legătură securizată



Suport sesiune (2)

- Înțelegerea este acceptată de client prin **antet** Cookie

Cookie: <nume>=<valoare>

inclus de browser în cererile referitoare la URL în care **domeniul** și / sau **calea** sunt calificate

Exemplu:

HTTP/1.1 200 OK

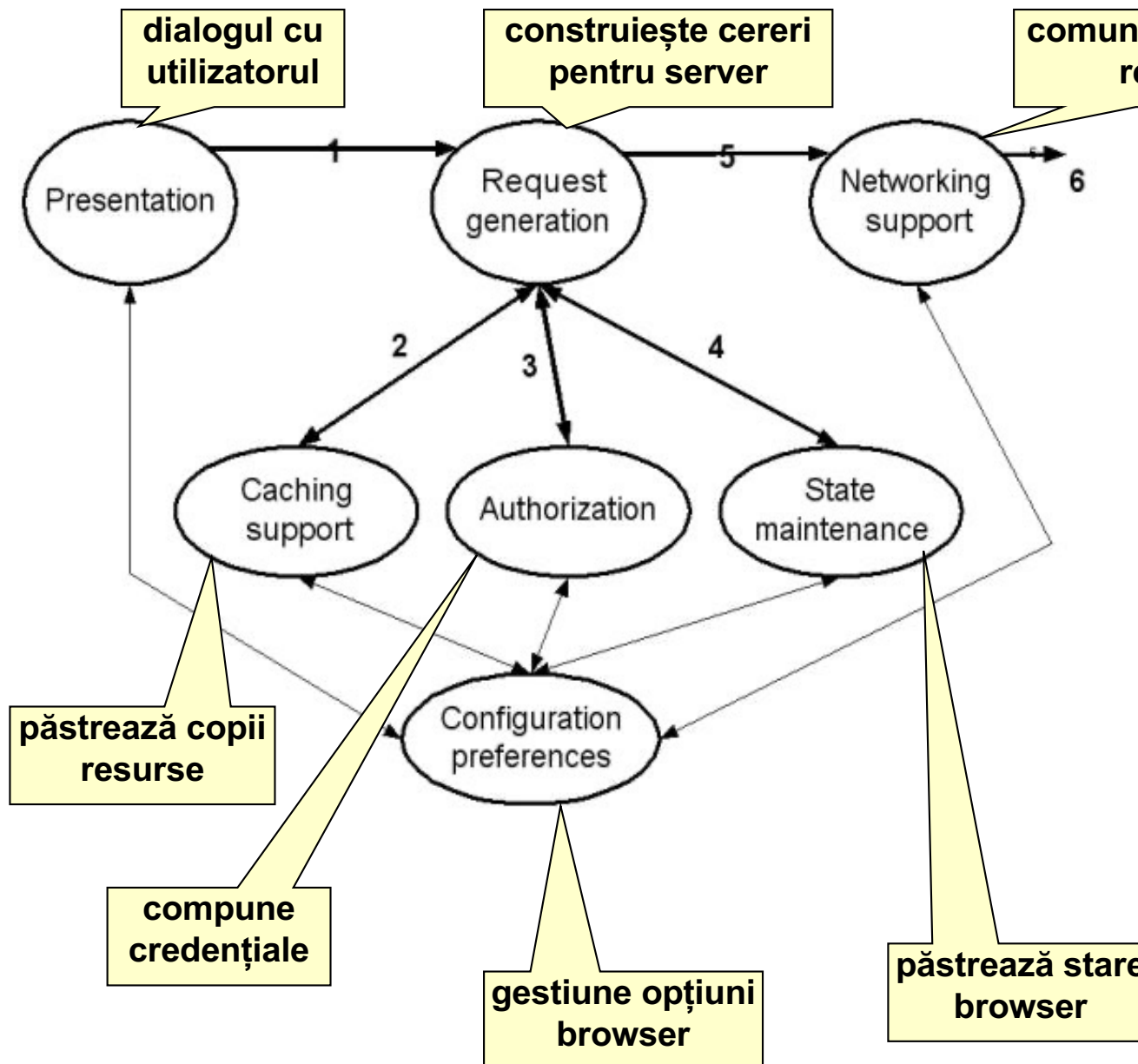
Set-Cookie: client=lon; path=/carte/capitol3/; domain=.pub.edu

GET /carte/capitol3/index.html HTTP/1.1

Host: www.cs.pub.edu

Cookie: client=lon

Schema browser - Generarea unei cereri



Funcționarea **modul prezentare** este bazată pe **evenimente** (ex. selecție URL afișat)

1. Identifică evenimentul și pasează legătura (URL)

2. Verifică dacă o copie a resursei este în cache

3. Verifică dacă sunt necesare credențiale

4. Verifică dacă trebuie incluse antete Cookie

5. Cererea este pasată modulului de rețea

6. Transmite prin rețea



Funcțiile modulelor din browser

Interfața utilizator

- **Afișează** pe ecran rezultatul primit de la modulul **Interpretare conținut** (din grupul de procesare a răspunsului)
- **Execută acțiunile** inițiate de utilizator, prin meniu, taste speciale etc.
 - Selectare/introducere URL
 - Completare formulare
 - Activare butoane de navigare (ex. Back)
 - Vizualizare sursa paginii, info resurse etc.
 - Setare opțiuni **configurare**
 - **Nu descarcă imagini referite în pagina HTML**
 - **Rejectează cookies**
- Pasează informația de cerere la **Generator cereri**



Generator cereri

- Primește legătura (URL) la pagina care va fi cerută
- URL poate fi:
 - **absolut** (ex. introdus manual) – este complet <http://domeniu/cale> → nu trebuie prelucrat
 - **relativ** (ex. preluat din pagina curentă de la modulul Interpretare conținut) → trebuie rezolvat!

Sunt **două cazuri**:

(1) - URL **relativ** la **directorul curent** al paginii afișate (calea din HREF **nu începe cu /**)

Ex:

URL curent: <http://www.myserver.com/mydirectory/index.html>

Link in pagina: `...`

Rezolvat la: <http://www.myserver.com/mydirectory/altdirector/pag2.html>



Generator cereri (2)

(2) - URL **relativ** la directorul **rădăcină al serverului Web** cu numele din URL-ul paginii curente (calea din HREF **începe cu /**)

Ex:

URL curent: <http://www.myserver.com/mydirectory/index.html>

Link în pagină: `...`

Rezolvat la: <http://www.myserver.com/rootdirector/homepage.html>



- Construiește linia de cerere, care are 3 componente

METODA

Implicit (la [activare hyperlink](#)) GET

În formular (specificat explicit) GET sau POST

/cale-resursa

Numai calea în HTTP/1.1

Tot URL în HTTP/1.0

HTTP/versiune



– Construiește antetele de bază

Host: www.cs.pub.ro

User-Agent: Mozilla/4.75 [en] (WinNT; U)

Referer: <http://www.cs.pub.ro/~ionescu/index.html>

Accept: text/html, text/plain, type/subtype

Accept-Charset: ISO-8859-1

...

Content-Type: mime-type/mime-subtype

Content-Length: xxx

Date:

Referer – pagina în care se află link-ul activat de utilizator



- Întreabă **Suport caching** dacă există intrare în cache
 - Nu există – cere resursa necondiționat
 - Există expirată - adaugă la cerere antet **If-Modified-Since**
 - dacă server răspunde cu **304 Not Modified**
 - » pasează intrarea din cache la **Interpretare conținut**
 - Există ne-expirată – întoarce intrarea din cache
- Întreabă **Autorizare** dacă e nevoie de autorizare pentru **domain/path**
 - Există credențele – adaugă antet **Authorization**
- Întreabă **Management stare** despre cookies (**domain/path**)
 - Da – adaugă antet **Cookie**
- Construiește corp cerere (vezi slide-uri următoare)
- Pasează întreaga cerere la **Suport rețea**
- Preferințele utilizatorului (**Configurare**) pot modifica fluxul cererii
 - nu se cer imaginile referite în pagină
 - nu se includ Cookies



- Construiește corp cerere
 - se aplică pentru POST, PUT
 - POST
 - parametrii din formulare în corp comandă

Content-Type: application/x-www-form-urlencoded

Content-Length: 6

s=YHOO



- PUT sau POST
 - folosind MIME

```
Content-Type: multipart/multipart_subtype;  
boundary="ThisRandomString"
```

```
--ThisRandomString
```

```
Content-Type: tip/subtip partea 1
```

```
Content-Transfer-Encoding: schema codificare partea 1
```

```
continut partea 1
```

```
--ThisRandomString
```

```
Content-Type: tip/subtip partea 2
```

```
Content-Transfer-Encoding: schema codificare partea 2
```

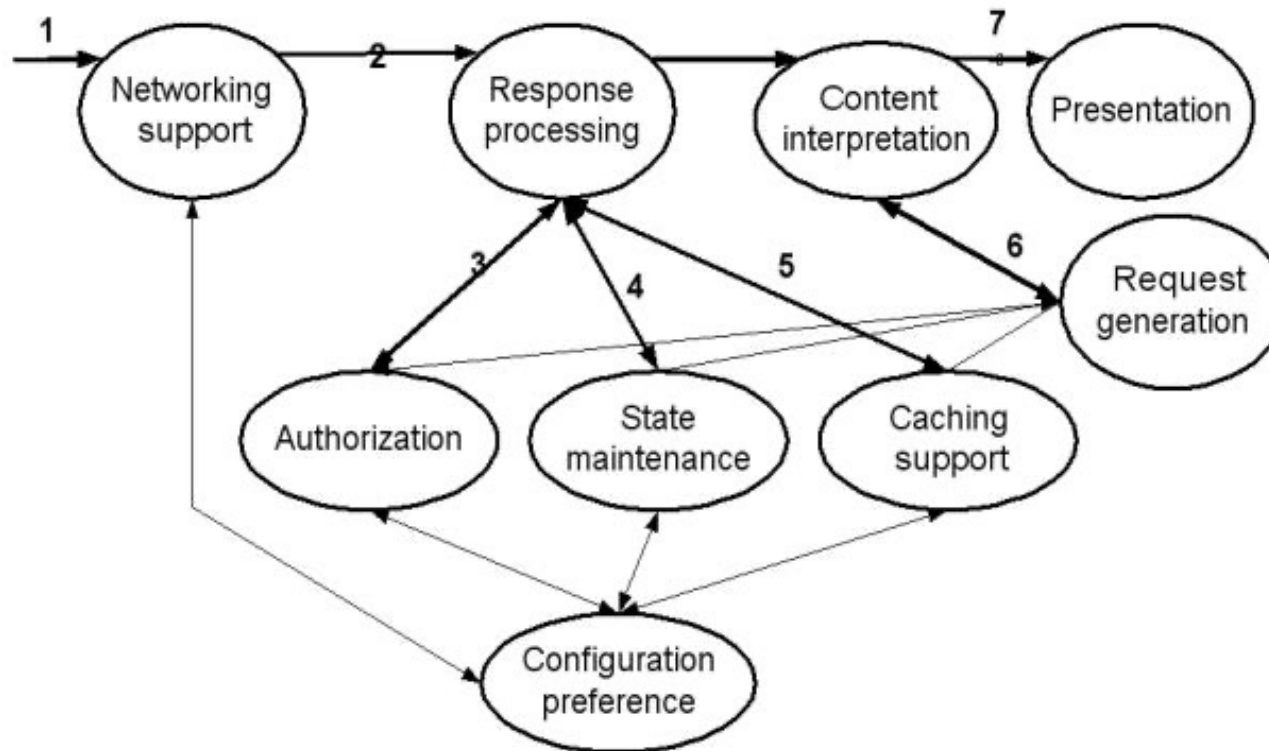
```
continut partea 2
```



Suport rețea

- **Transmite cererea**
 - Primește cereri de la **Generator cereri** și le pune în coada transmisie
 - Întreabă **Configurare** pt. a determina dacă ținta este un proxy și alte opțiuni rețea
 - Deschide socket pentru a transmite cereri din coadă
 - transmite mai multe cereri la o conectare
- **Tratează răspuns**
 - Așteaptă răspunsuri la cereri
 - Pasează la **Procesare răspuns**

Procesarea răspunsului



1. Primește răspuns
2. Pasează răspuns la modulul de procesare
3. Cererea a fost **rejectată** – verifică dacă pot fi folosite **credențele** – verifică **redirectare**
4. Dacă se cere info **cookie**, contactează modulul management stare

5. Contactează **suport caching** pentru memorarea răspunsului; apoi pasează răspuns la **interpretare conținut**
6. **Decodifică** corp răspuns, **procesează** diferite tipuri MIME și **parsează** conținut pt. eventuale referințe la resurse adiționale
7. Conținut pasat la modul prezentare



Procesare răspuns

- Verifică stare 401 (ne-autorizat)
 - Cere modulului de **Autorizare** credențele pt. domeniul din antet **WWW-Authenticate**
 - Există – retransmite cerere cu credențele adăugate
 - Nu – cere credențele de la utilizator (prin **Interfața utilizator**) și retransmite
 - Credențialele sunt memorate pe durata unei sesiuni



– Verifică stare **redirectare** (301/302/307)

- Dacă

HTTP/1.1 301 Moved Permanently

Location: <http://www.alta-locatie.com/pagina.html>

- Retransmite cerere la URL din antet **Location**

GET /pagina.html HTTP/1.1

Host: www.alta-locatie.com

- Dacă 301, memorează în ***persistent lookup table*** pentru redirectare automată a cererilor următoare
- Dacă 302 (pagina mutată temporar) - nu memorează



- Verifică antet **Set-Cookie**
 - Cere **Management stare** să memoreze cookie în browser
 - Memorarea: pe sesiune / pentru o durată specificată
- Verifică **opțiuni caching** și transmite cerere la **Suport caching** pentru a memora resursele obținute
 - răspunsul poate include data expirării
 - HTTP/1.1 200 OK**
 - . . .**
 - Cache-Control: private**
 - Expires: Tue, 06 Feb 2005 04:33:20 GMT**
 - Last-Modified: Mon, 05 Feb 2005 04:33:18 GMT**
 - ...**
- Pasează rezultat la **Interpretare conținut**



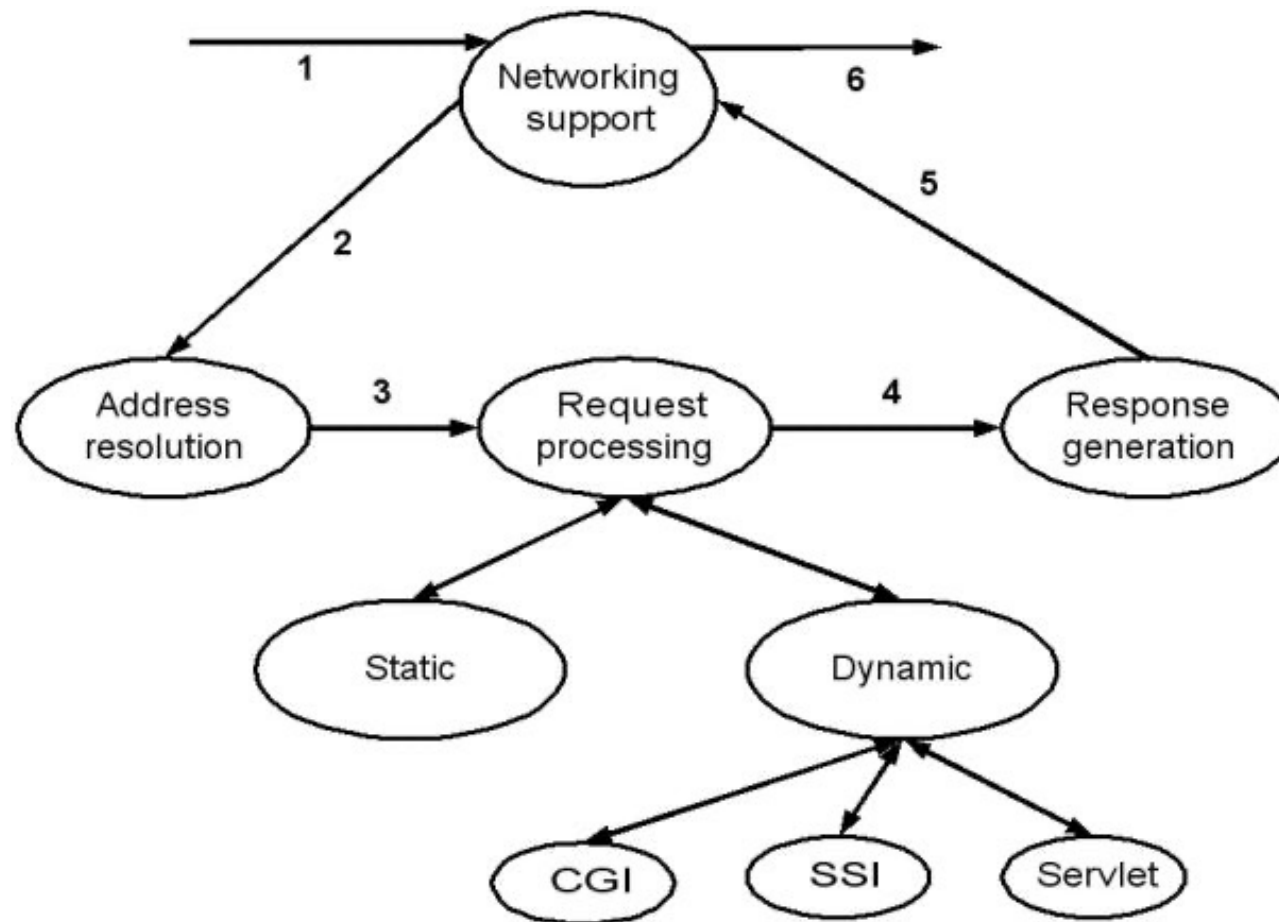
- Interpretare conținut

- Primește conținut de la Procesare răspuns (Uneori de la Suport caching)
- Examinează antete codificare și, dacă e cazul, decodifică conținut
 - Content-Transfer-Encoding: chunked
 - Content-Encoding: compress | gzip
- Pasează conținut decodificat la module specifice tipului MIME pe baza antet Content-Type
- Dacă link-uri la alte resurse, pasează URL la Generator cereri
- Pasează fiecare modul prelucrat la Interfața utilizator

- Configurare

- Folosește Interfața utilizator pentru setări preferințe
- Primește cereri de la alte module, pentru a determina acțiunile în funcție de preferințele utilizatorilor

Operații Server



1. Serverul primește o cerere
2. Pasează la modulul de **rezoluție a adresei** care
 - (a) determină **server-ul**;
 - (b) determină dacă cerere necesită conținut **static / dinamic**;
 - (c) examinează **credențele** autorizare.
3. Pasează la modul **procesare cerere**, care apelează sub-module necesare

4. Rezultat pasat **generatorului de răspuns**
5. Pasat modulului **suport rețea**
6. Transmite clientului



Rezolvarea adresei

- selectează **virtual host**
 - nu există antet Host: -> eroare 400 Bad request
 - există -> determină domeniul
 - > determină **parametrii de config. logică** (proprii virtual host)

Obs. fragment din fișier configurare Apache

<VirtualHost www.ceva.com>

ServerAdmin webmaster@calculatoare.com

Alias /test /servlet/test

Alias /images /static/images

DocumentRoot /www/docs/ceva

ServerName www.ceva.com

ErrorLog logs/ceva-error-log

CustomLog logs/ceva-access-log common

</VirtualHost>



- rezolvă alias-uri folosind info de configurare logică

Alias	/test	/servlet/test
Alias	/images	/static/images

– <http://www.ceva.com/test?a=1&b=2>

/test -> /servlet/test

– <http://www.ceva.com/images/nou.gif>

/images -> /static/images



- **mapează adresa**
 - **pagină statică** – adaugă calea la rădăcina serverului la calea din URL
 - URL **http://www.ceva.com/pagini/cucu.html**
 - în configurare **DocumentRoot /www/docs/ceva**
 - calea devine **-> /www/docs/ceva/pagini/cucu.html**
 - **pagină dinamică** – este creată de un program
 - mecanismul folosit se determină pe baza:
 - **prefix cale URL** **/servlet/** target = servlet Java
 - /cgi-bin/** target = CGI script
 - **sufix nume** **.cgi** **.php**
- **verifică autentificare**
 - **cod eroare dacă resursa cerută este protejată**



- **Procesare cerere**

- regăsește conținut (sau primește de la programul care-l generează)
- setează tipul MIME conform configurare server

text/css css

text/html html htm

text/plain asc txt

text/xml xml

video/mpeg mpeg mpg mpe

- setează alte antete (Content-Length, Last-Modified etc.)
- antet transfer pe bucăți (chunked)

HTTP/1.1 200 OK

Content-Type text/plain

Content-Transfer-Encoding: chunked



- Conexiunea persistentă
 - cozi de cereri și de răspunsuri – păstrează **ordinea**:
 - **răspunsurile** din coada de răspunsuri păstrează ordinea **cererilor** din coada de cereri



Funcționare server

- server HTTP = set de thread-uri care procesează cererile clientilor
- **Fișier configurare fizică (exemplu din Apache pentru Windows)**

ServerName demo

ServerRoot "C:/Program Files/Apache Group/Apache"

ServerType Standalone păstrat continuu în execuție

Port 80

KeepAlive On suportă conexiuni persistente

MaxKeepAliveRequest 100 nr maxim cereri în așteptare

KeepAliveTimeout 15 taie conex. dacă nu cerere nouă în 15 sec

MaxRequestsPerChild 200 nr max cereri procesate fără repornire child

Timeout 300 timp maxim de procesare a unei cereri



Studiu individual

A. S. Tanenbaum Rețele de calculatoare, ed 4-a, BYBLOS 2003

7.3 WORLD WIDE WEB

7.3.1 Aspecte arhitecturale

7.3.2 Documente Web statice

7.3.3 Documente Web dinamice

7.3.4 HTTP – HyperText Transfer Protocol

A. S. Tanenbaum Computer networks, 5-th ed. PEARSON 2011

7.3 THE WORLD WIDE WEB

7.3.1 Architectural Overview

7.3.2 Static Web Pages

7.3.3 Dynamic Web Pages and Web Applications

7.3.4 HTTP—The HyperText Transfer Protocol