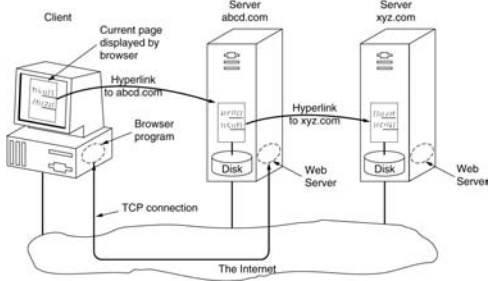


World Wide Web

[Tanenbaum, Distributed Systems 2003]

- Construit pe modelul client-server
- Clienți și servere distribuiți pe mașini diferite
- Comunica prin HTTP

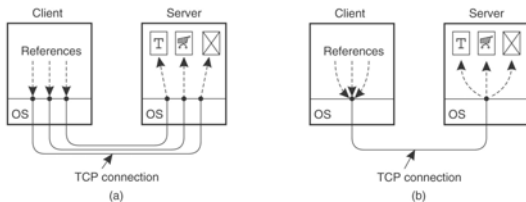


Interacțiunea client - server

- Browser - determina URL
- Browser - cere DNS-ului adresa IP pentru www.w3.org
- DNS - raspunde cu 18.23.0.23
- Browser - deschide o conexiune TCP la port 80 pe 18.23.0.23
- Browser - trimite o comanda
`GET /hypertext/www/TheProject.html`
- Server www.w3.org - trimite fisierul `TheProject.html`
- Conexiunea TCP este inchisa
- Browser - afișează conținutul din `TheProject.html`
- Browser - extrage și afișează toate imaginile din `TheProject.html` (se deschide o noua conexiune TCP pentru fiecare imagine)

Conexiuni persistente

- Disponibile 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)



Comunicarea - 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	Rezervat pentru o utilizare viitoare

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?s=YHOO`

- Cerere HTTP

```
GET /q?s=YHOO HTTP/1.1
Host: financiar.yahoo.com
User-Agent: Mozilla/4.75 [en] (WinNT; U)
```

Raspuns

```
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 - YHO</TITLE></HEAD>
<BODY>
...
</BODY>
</HTML>
```

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)
Content-Encoding	Răspuns	Cum este codat conținutul (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 redirectionarea 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 pastra sau nu conexiunea (ex. Connection: Close)

Replicare - caching

- Tipuri de caching:
 - la client – cache privat
 - la proxy, server – cache-uri partajate
- Control caching – introdus în HTTP/1.1
 - se face de către 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)

- Asigura ca documentul din cache este consistent cu cel din server

- Soluție 1: Folosind comanda **HEAD**
 - clientul transmite **HEAD**
 - primește răspuns și verifică antet **Last-Modified**
 - transmite **GET** dacă document 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)
```

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
```

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
Date: Mon, 05 Feb 2005 04:33:19 GMT
Server: Apache/1.2.5
Last-Modified: Mon, 05 Feb 2005 04:30:28 GMT
Content-Length: 2289
...
```

Soluție pentru performanță

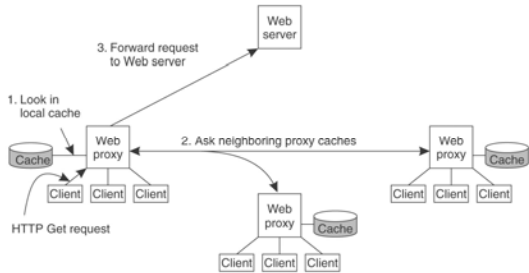
- Clientul nu contactează serverul pentru orice cerere
 - Răspunsul unui server poate include data expirării, care este memorată de client

```
HTTP/1.1 200 OK
Date: Mon, 05 Feb 2005 04:33:20 GMT
Content-Type: image/jpeg
Content-Length: 35782
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 ne-expirată** – folosește intrarea din cache
 - Există expirată** - adaugă la cerere antet **If-Modified-Since**
 - dacă server răspunde cu **304 Not Modified**
 - folosește intrarea din cache
 - altfel folosește pagina primită

Caching cooperativ

- Se aplica cache-urilor partajate
- La o absenta din cache, Web proxy verifica proxy-urile vecine



Autentificare si autorizare

- Mecanisme folosite de server
 - grupuri de adrese IP pentru care accesul este interzis / permis
 - variante de ACL-uri – identificatori+parole pentru care accesul este permis
 - asociate cu "realms" (domenii) – folosite de administratori pentru a organiza resursele protejate in categorii
- Autentificare de baza
 - prin antet de autorizare
 - nume si parola transmise codat Base64 (nu criptat) – atentie HTTPS

Autentificare si autorizare

- Secventa de actiuni
 - Browser cere resursa restrictionata
 - Server raspunde 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="Capitol3"
 - Browser 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
 - Server verifica credentialele de autorizare si satisface cererea (sau refuza cu 403)
 - Browser foloseste credentiale si in viitoarele cereri la URL dependente

<http://cs.pub.ro/~popescu/clase/depinde> de <http://cs.pub.ro/~popescu/>

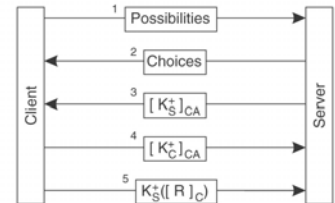
Securitatea comunicatiei client-server

- SSL – Secure Socket Layer -> HTTPS (Secure HTTP)
- TLS – Transport Layer Security
 - permite autentificarea mutuala

$[K_s^*]_{CA}$ – certificat Server

$[K_c^*]_{CA}$ – certificat Client

$K_s^*([R]_c)$ – numar aleator R criptat cu cheia secreta a clientului C si cu cheia publica a serverului K_s^*



Suportul sesiunii

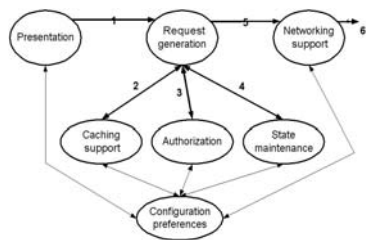
- HTTP protocol "stateless" – serverul nu face asocieri intre diversele comenzi
- Mentinerea starii intr-o sesiune – serverul sa poata asocia logic cereri, ca provenind de la acelasi client
- Mecanismul = cookie
- Principiul
 - cookie = pereche "cheie = valoare"
 - transmis de server si memorat de browser
 - inclus de browser in cereri legate logic in aceeasi sesiune
- Doi parametri importanti
 - domeniul – nume gazda (dom1.cs.pub.ro) sau suffix de nume (.pub.ro)
 - calea – cel mai inalt nivel al unei cai URL pentru care se aplica cookie (/carte/capitol3/ se transmite cookie pentru toate caile care au acest prefix)

Suportul sesiunii - cookie

- Intelegerea initiata 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 de browser
 - path, domain identifica cererile care trebuie insotite de cookie
 - secure browser-ul trebuie sa transmita info pe legatura securizata
 - Cand e nevoie, browserul trimite cookie prin antet Cookie
 - Cookie: <nume>=<valoare>
 - Inclus de browser pentru cererile referitoare la URL in care domeniul si calea se incadreaza in cele din Set-Cookie
- Exemplu:
- ```
HTTP/1.1 200 OK
Set-Cookie: client=Ion; path=/carte/capitol3/; domain=.pub.edu

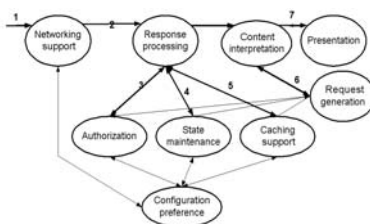
GET /carte/capitol3/index.html HTTP/1.1
Host: www.cs.pub.edu
Cookie: client=Ion
```

## Generarea cererii



1. Identifica evenimentul si paseaza legatura
2. Verifica daca o copie a resursei este in cache
3. Verifica daca sunt necesare credentiale
4. Verifica daca trebuie incluse antete Cookie
5. Cererea este pasata modulului de retea
6. Transmite prin retea

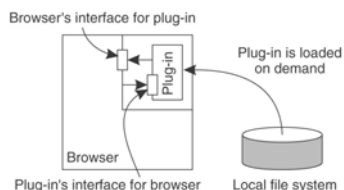
## Procesarea raspunsului



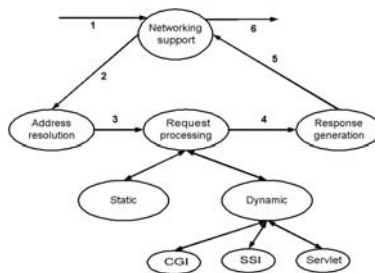
1. Primeste raspuns
2. Paseaza raspuns
3. Cererea a fost rejectata – verifica daca pot fi folosite credentiale
4. Daca se primeste info cookie, contacteaza modulul management stare
5. Contacteaza suport caching pentru memorarea raspunsului; apoi paseaza raspuns la interpretare continut
6. Decodifica corp raspuns, proceseaza diferite tipuri MIME si paseaza continut ptr determinare resurse aditionale necesare (in raspuns sunt referinte la alte resurse)
7. Continut pasat la modul prezentare

## Extensibilitate browser

- ☐ Folosita pentru a suporta diverse tipuri de documente
- ☐ Mecanismul: **plug-in**
  - incarcat dinamic de browser
  - are interfata standard pentru browser



## Operatii Server



1. serverul primeste o cerere
2. Paseaza la modulul de rezolutie a adresei care (a) determina target-ul; (b) determina daca cerere continut static / dinamic; (c) examineaza credentiale autorizare.
3. Paseaza la modul procesare cerere, care apeleaza sub-module necesare
4. Rezultat pasat generatorului de raspuns
5. Pasat modulului suport retea
6. Transmite clientului

## Functionare server

Server HTTP = set de thread-uri care proceseaza cererile clientilor

- **Fisier configurare fizica (Apache pentru Windows)**
  - ServerName demo
  - ServerRoot "C:/Program Files/Apache Group/Apache"
  - ServerType Standalone
  - Port 80
  - KeepAlive On
  - MaxKeepAliveRequest 100
  - KeepAliveTimeout 15
  - MaxRequestsPerChild 200
  - Timeout 300
- procesul ramane continuu in executie
- asculta la portul 80
- serverul poate suporta conexiuni persistente
- cu cel mult 100 cereri in asteptare per conexiune
- cu 15 sec timeout de asteptare cerere noua
- numar maxim de 200 cereri procesate fara repornire thread
- timp maxim de procesare a unei cereri de 300 sec

## Clustere server

- ☐ Mecanismul de "TCP handoff"
- ☐ Switch de nivel transport
  - accepta cererea de conectare TCP de la un client
  - identifica serverul potrivit
  - paseaza cererea catre acel server
- ☐ Serverul
  - transmite confirmarea cererii de conectare TCP
  - in campul sursa la pachetului IP pune adresa switch-ului (pentru a "pacali" clientul)

## Clustere server

Functionare Front end:

- la nivel transport

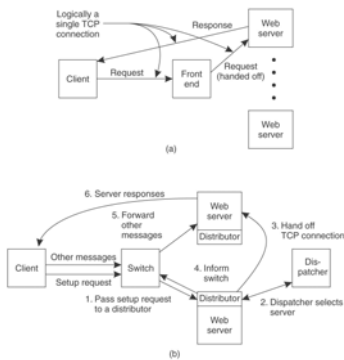
- la nivel aplicatie

a) cererea este pasata unui server, in functie de

- incarcare
- continut

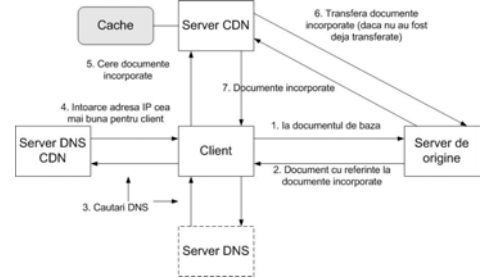
raspunsul este trimis direct clientului

b) functia front end impartita intre un switch de nivel transport, distribuitori si dispatcher



## Replicare – CDN – Content Delivery Network

- Presupunere: o pagina include multe alte documente (imagini, video etc.) care se schimba in timp – sunt replicate in servere Akamai CDN
- URL modificat contine:
  - virtual ghost – o referinta la doc incorporat + nume server DNS CDN
  - numele serverului de origine (care furnizeaza referintele la doc. incorporate)



## Solutia Akamai

- URL-ul (modif) unui document CDN refera un **ghost virtual**
- include numele unui server CDN DNS (rezolvat de un DNS obisnuit) + nume server de origine
- serverul CDN DNS intoarce adresa unui server CDN obisnuit care**
  - este apropiat geografic de client
  - nu este incarcat
- cererile de resurse incorporate in pagina sunt dirijate la acest server CDN
- daca serverul CDN nu are inca resursele, le descarca folosind adresa serverului de origine
- Problema cheie:** gasirea CDN DNS apropiat
  - in Akamai: se cauta dupa adresa IP a clientului intr-o baza de date care mentine o harta a Internet-ului
  - alternativa: se da aceeasi adresa IP mai multor servere CDN, urmand ca cel mai apropiat sa fie gasit prin politica de rutare "calea cea mai scurta"