

CPSC 471: Computer Communications

Applications

Figures from [Computer Networks: A Systems Approach](#), version 6.02dev
(Larry L. Peterson and Bruce S. Davie)

You may not distribute/post these lecture slides without written permission
from Dr. Mike Turi, ECE Dept., California State University, Fullerton

DNS – Domain Name System

- An infrastructure service
- Translates host names into host addresses
 - Host addresses easy for routers to understand
 - Host names easy for humans to understand

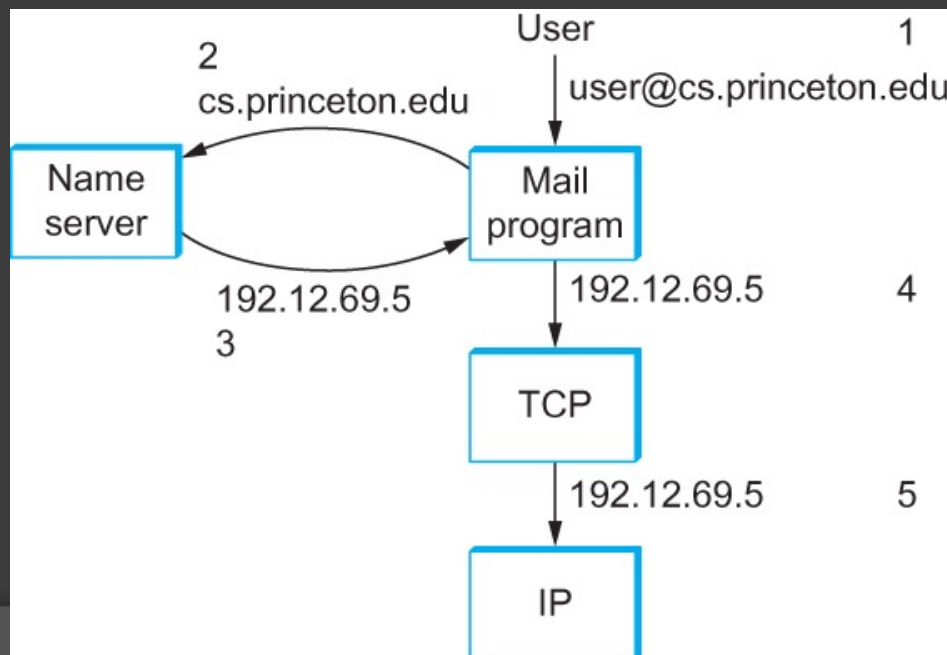


Figure 229

The DNS Name Space (aka Domain Hierarchy)

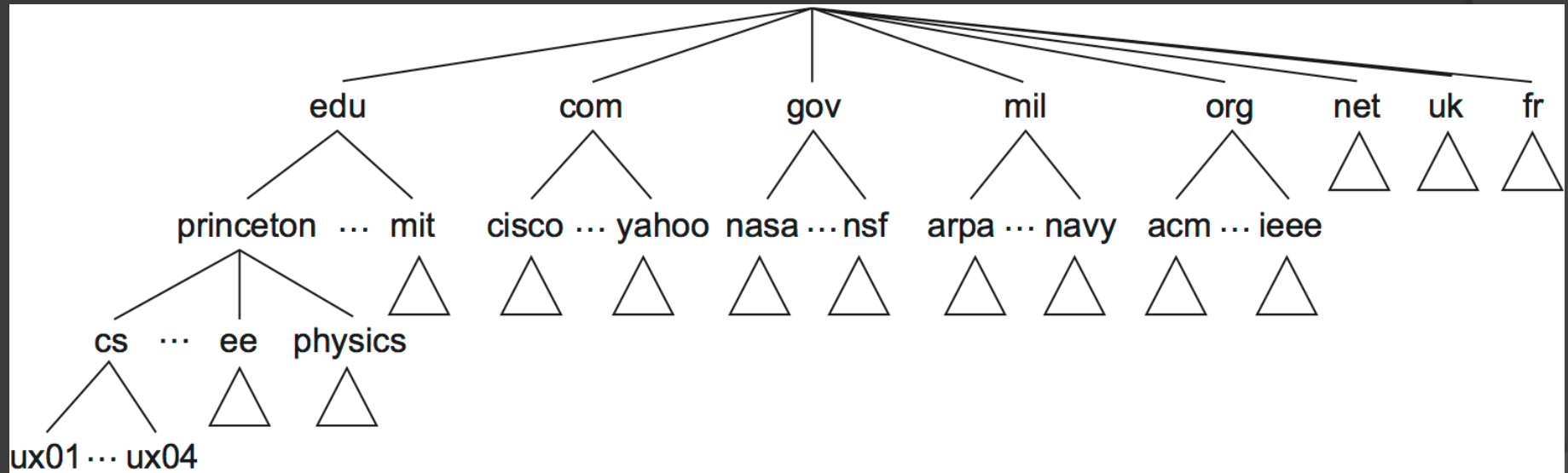


Figure 230

Domain Hierarchy Partitioned into Zones (Name Servers)

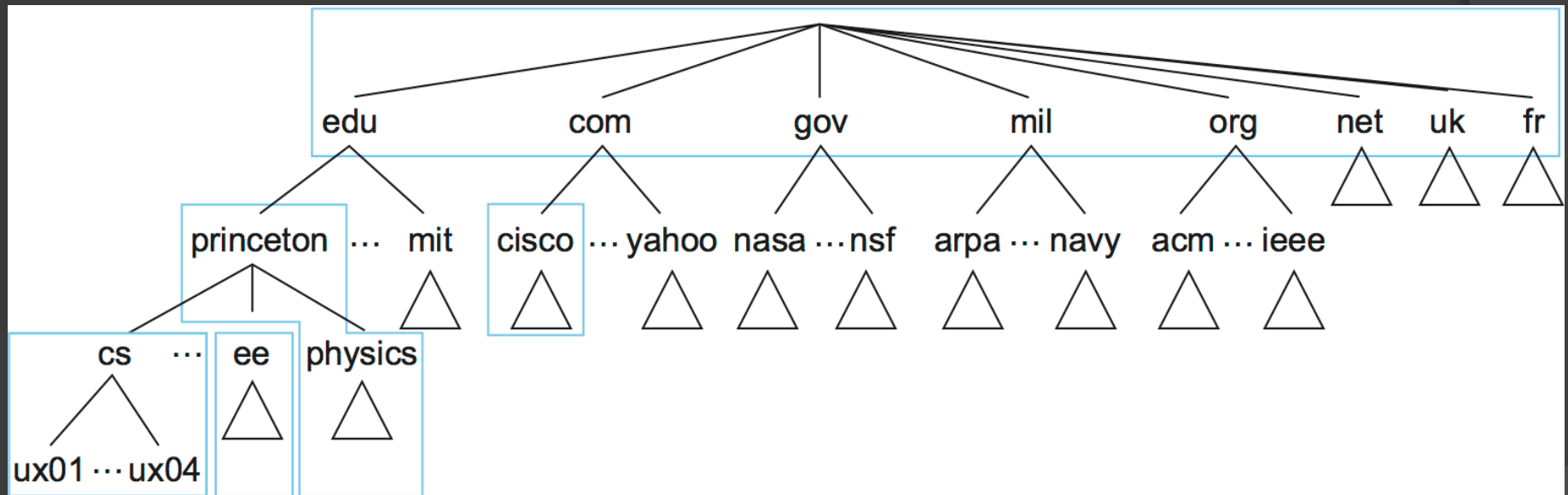


Figure 231

Hierarchy of Name Servers

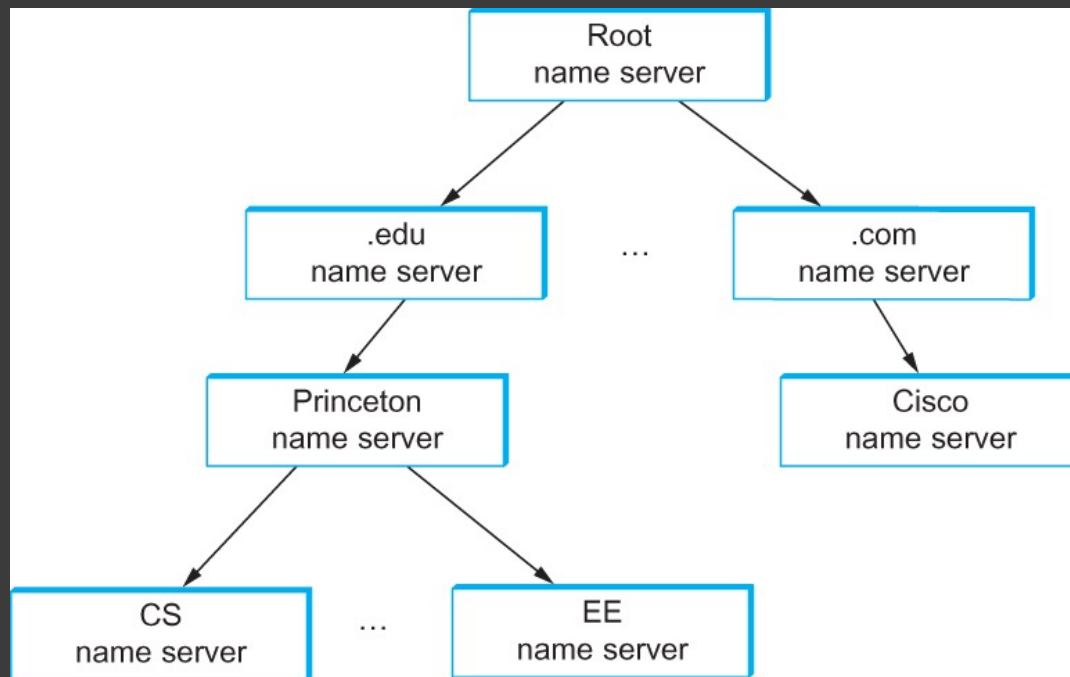


Figure 232

Name Resolution

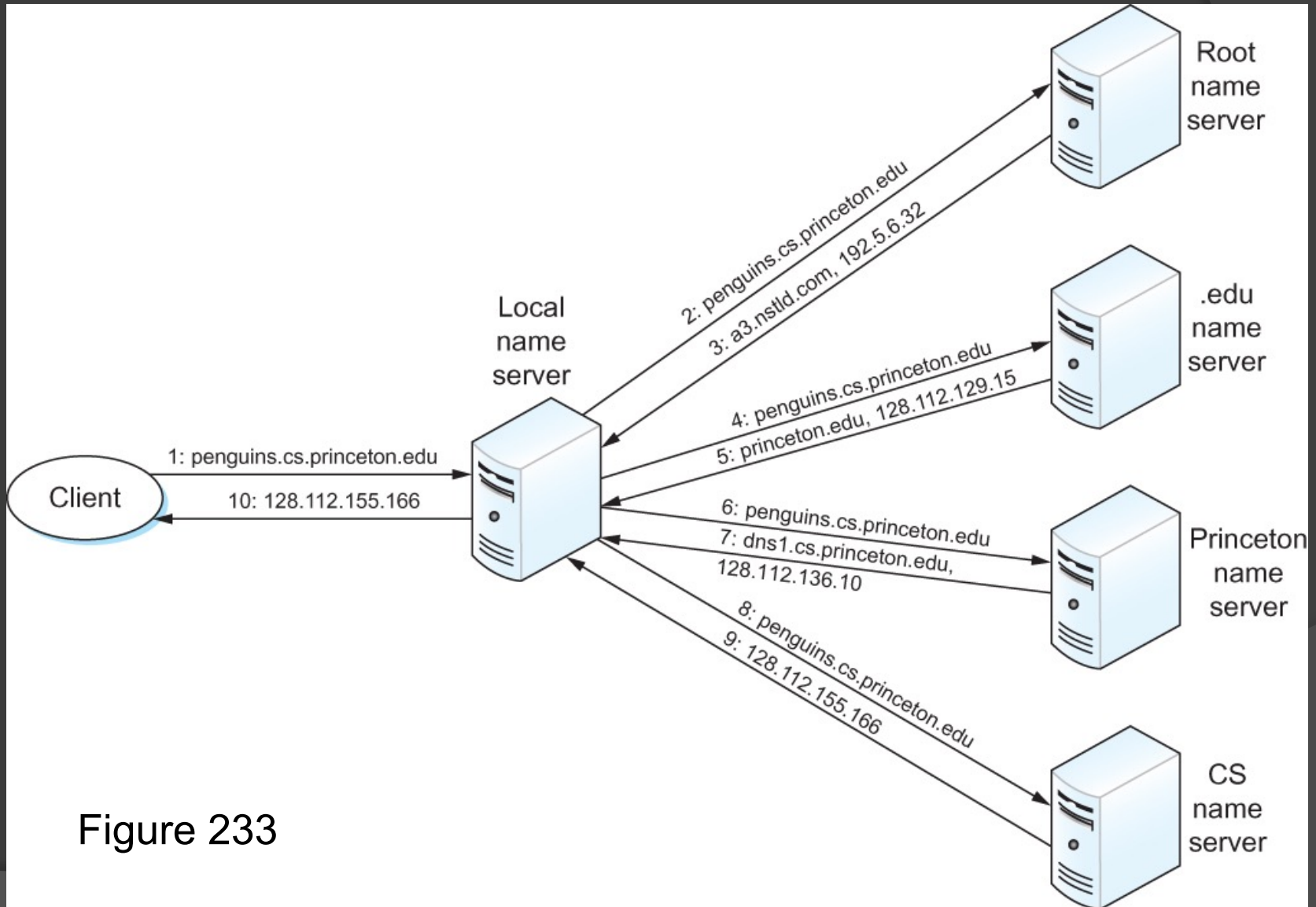


Figure 233

Electronic Mail

- ⦿ Message Formats
- ⦿ MIME
- ⦿ Message Transfer
 - SMTP – Simple Mail Transfer Protocol
- ⦿ Mail Reader
 - POP3 – Post Office Protocol version 3
 - IMAP – Internet Message Access Protocol

MIME – Multipurpose Internet Mail Extensions

⦿ Problems with international languages:

- Languages with accents
(e.g., French, German)
- Languages in non-Latin alphabets
(e.g., Hebrew, Russian)
- Languages without alphabets
(e.g., Chinese, Japanese)

⦿ Messages containing:

- Images
- Audio
- Video

Message Transfer

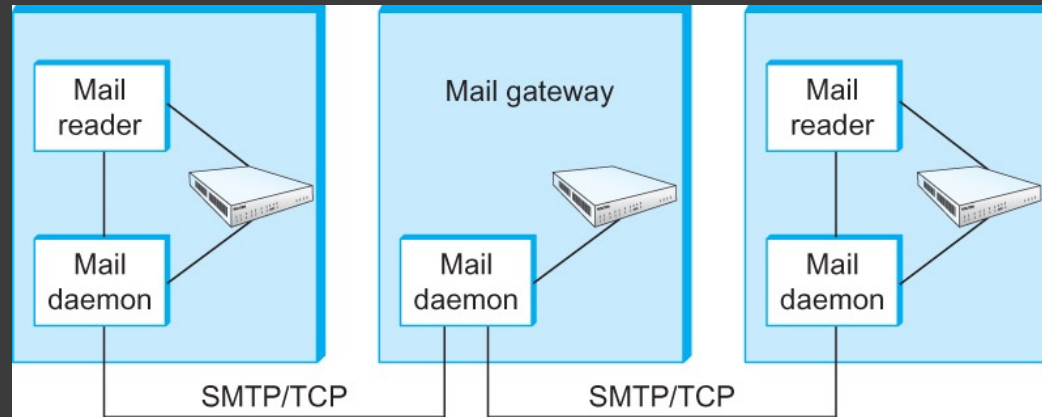


Figure 216

The World Wide Web

- ⦿ Architectural Overview
- ⦿ URLs
- ⦿ HTTP – HyperText Transfer Protocol

Architecture: Client/Server

- ⦿ Client runs a browser as a process
- ⦿ Server may be:
 - Multithreaded
 - One front end for handling requests/replies
 - Multiple processing threads
 - A server farm
 - One machine acts as the front end
 - Separate processing nodes (machines)

HTTP Behavior

HTTP 1.0

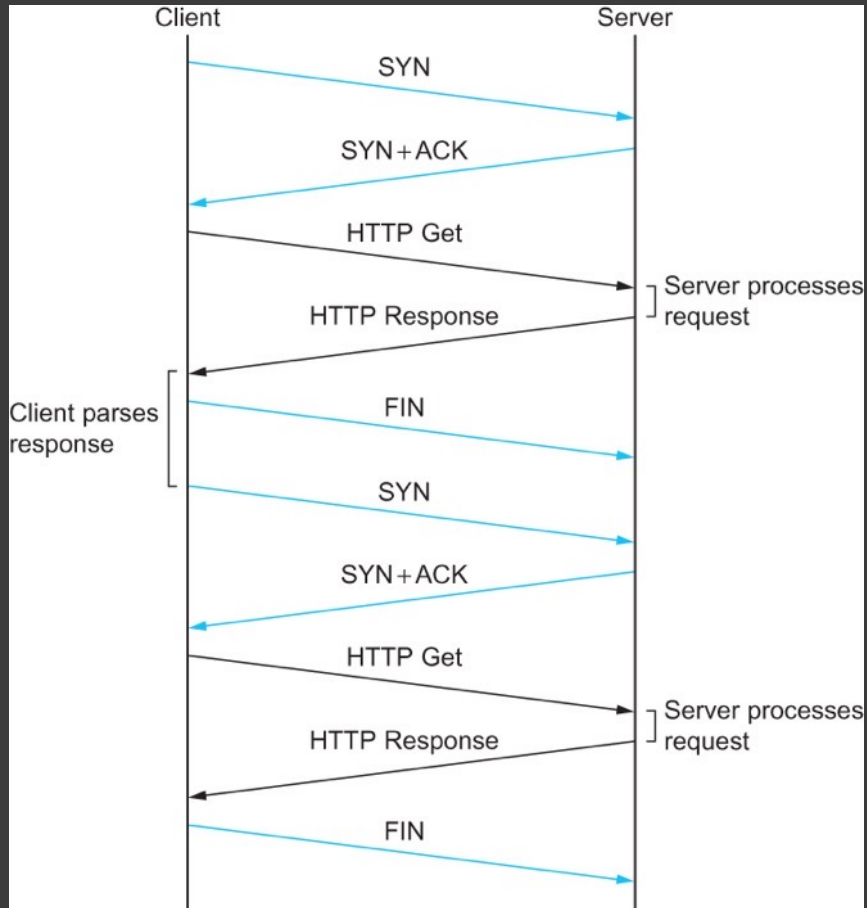


Figure 219

HTTP 1.1

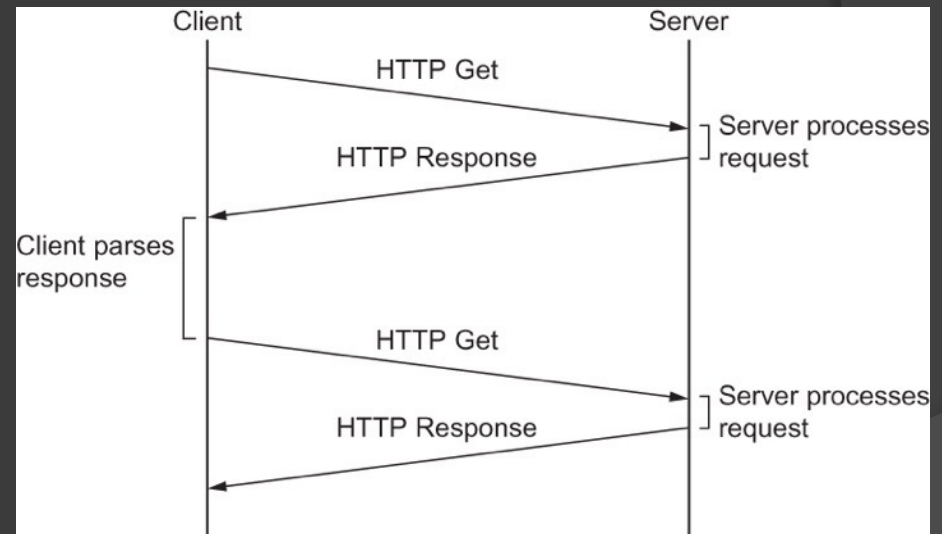
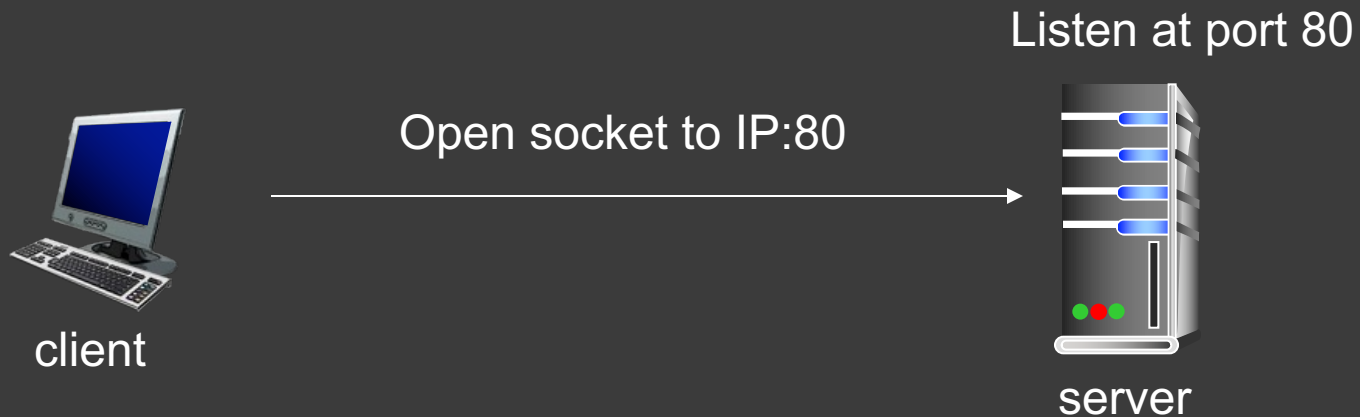


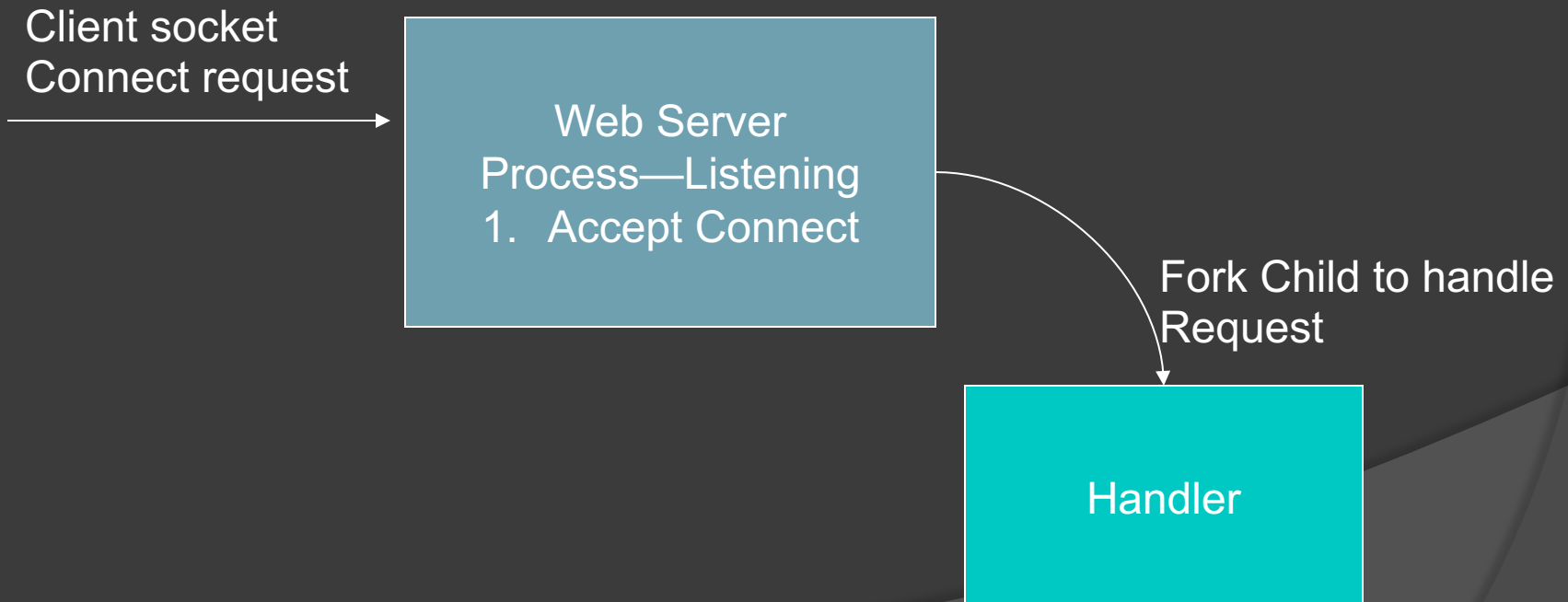
Figure 220

Anatomy of a Web Request

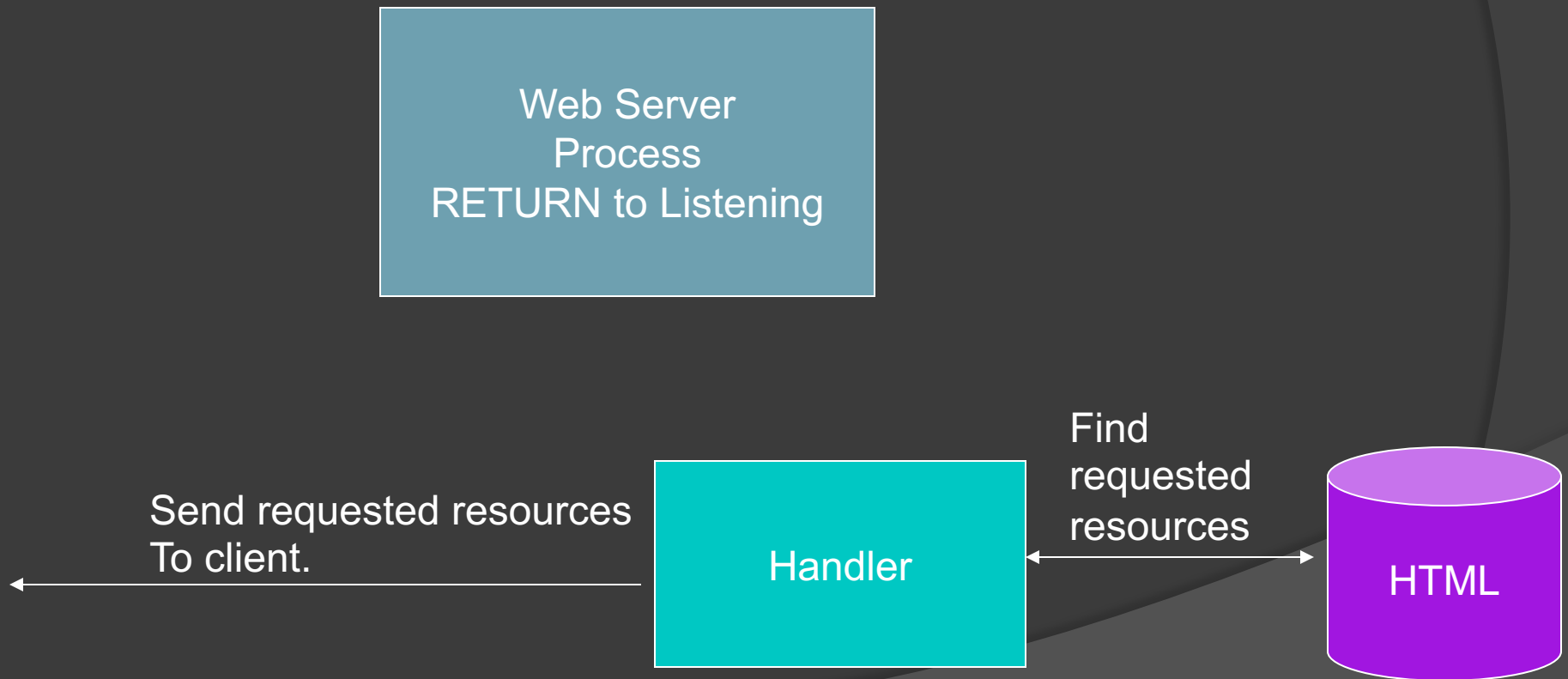


The remainder of this presentation is from Dr. Victor Valgenti,
Washington State University, CptS 455, Fall 2012

Anatomy of a Web Request (cont.)



Anatomy of a Web Request (cont.)



Anatomy of a Web Request (cont.)



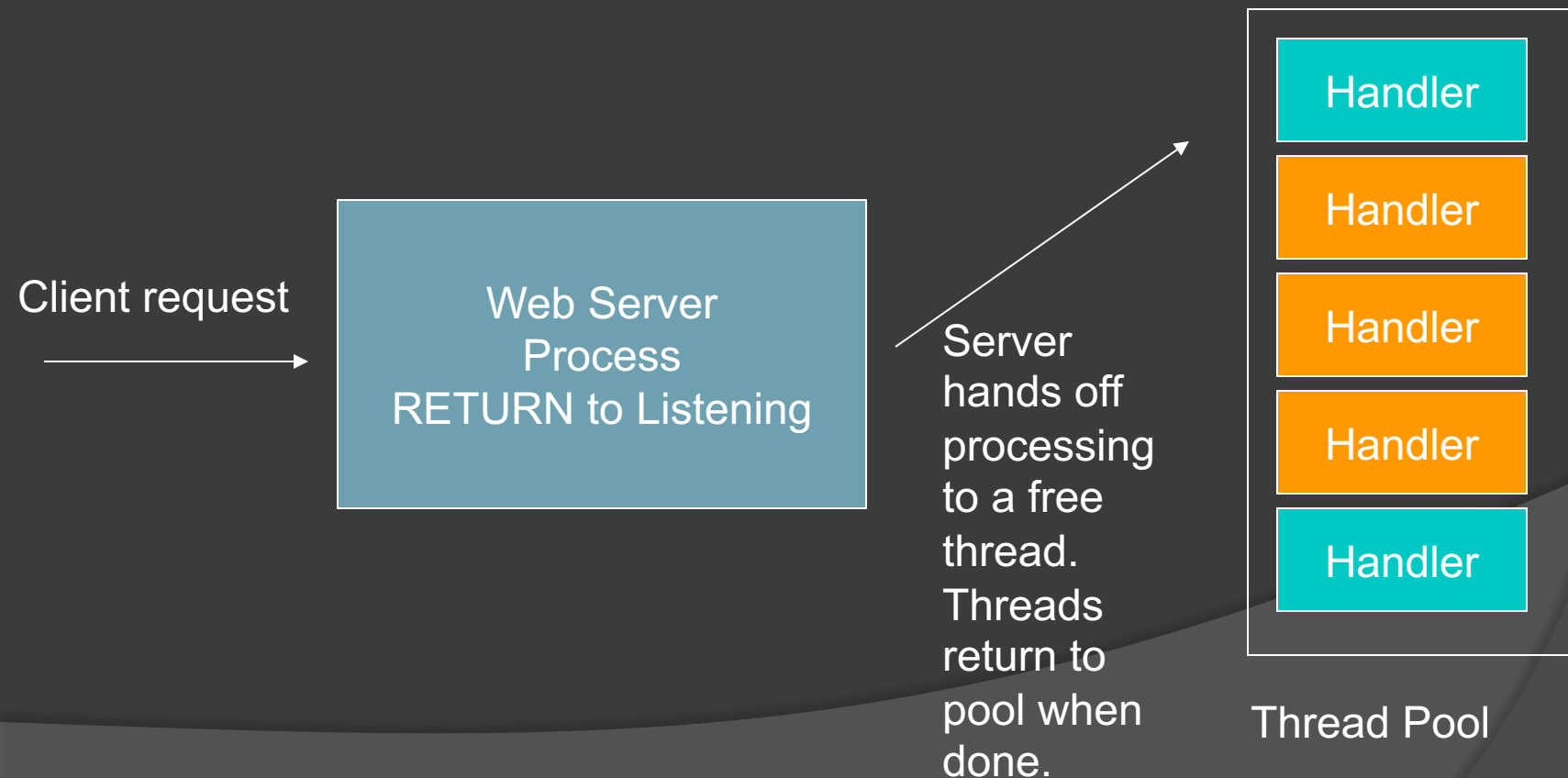
Web Server
Process
RETURN to Listening

The diagram illustrates the lifecycle of a web request. A light blue box at the top represents the 'Web Server Process' which has 'RETURN to Listening'. Below it, a cyan box represents the 'Handler', which is crossed out with a large grey 'X', indicating its termination. To the right of the handler box, text states 'Communication complete. Child dies.'.

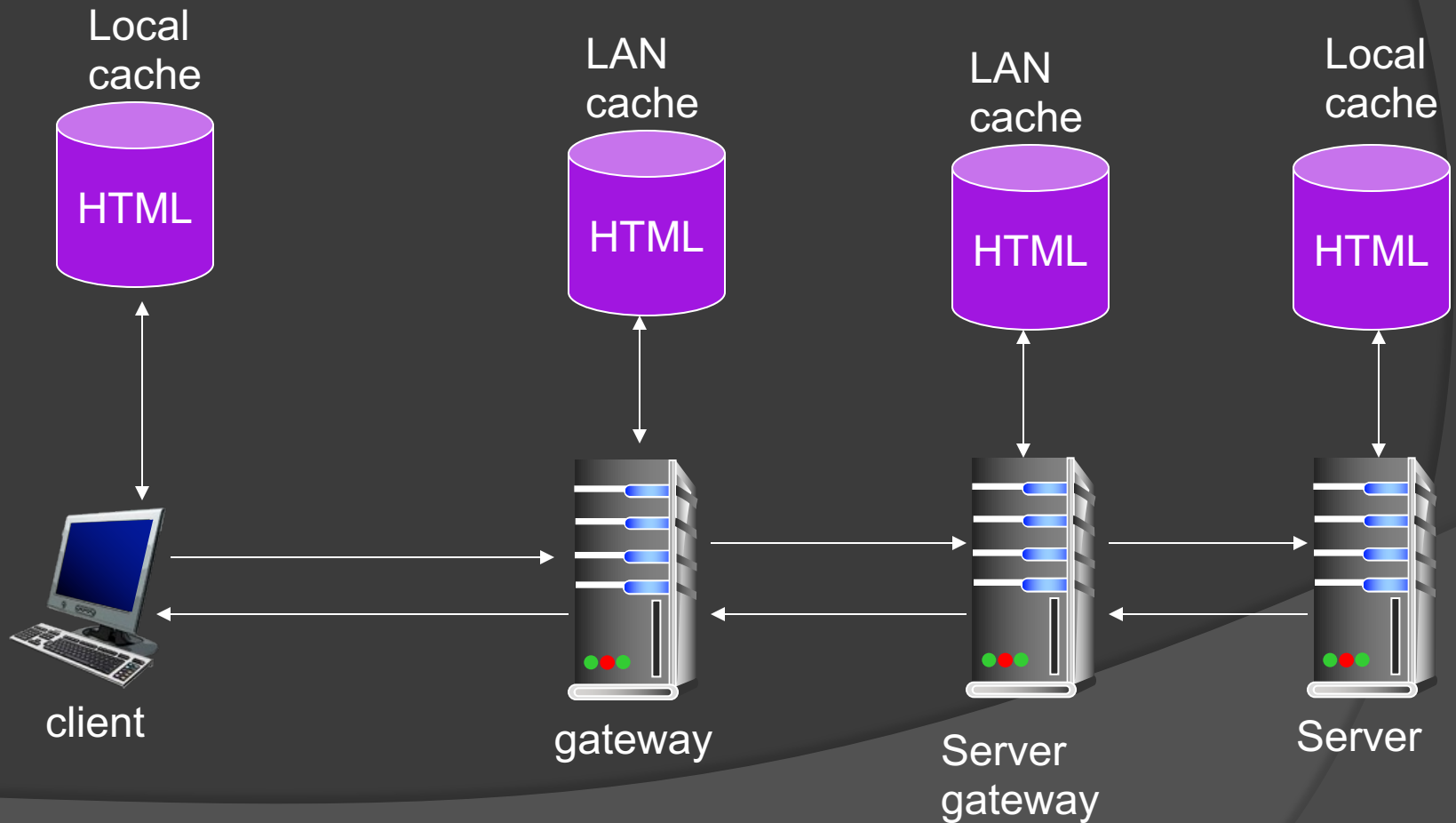
Handler

Communication complete.
Child dies.

Anatomy of a Web Request (cont.)



Web Caching



HTTP Header Request

request line
(GET, POST,
HEAD commands)

header
lines

carriage return,
line feed at start
of line indicates
end of header lines

carriage return character
line-feed character

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

HTTP Header Response

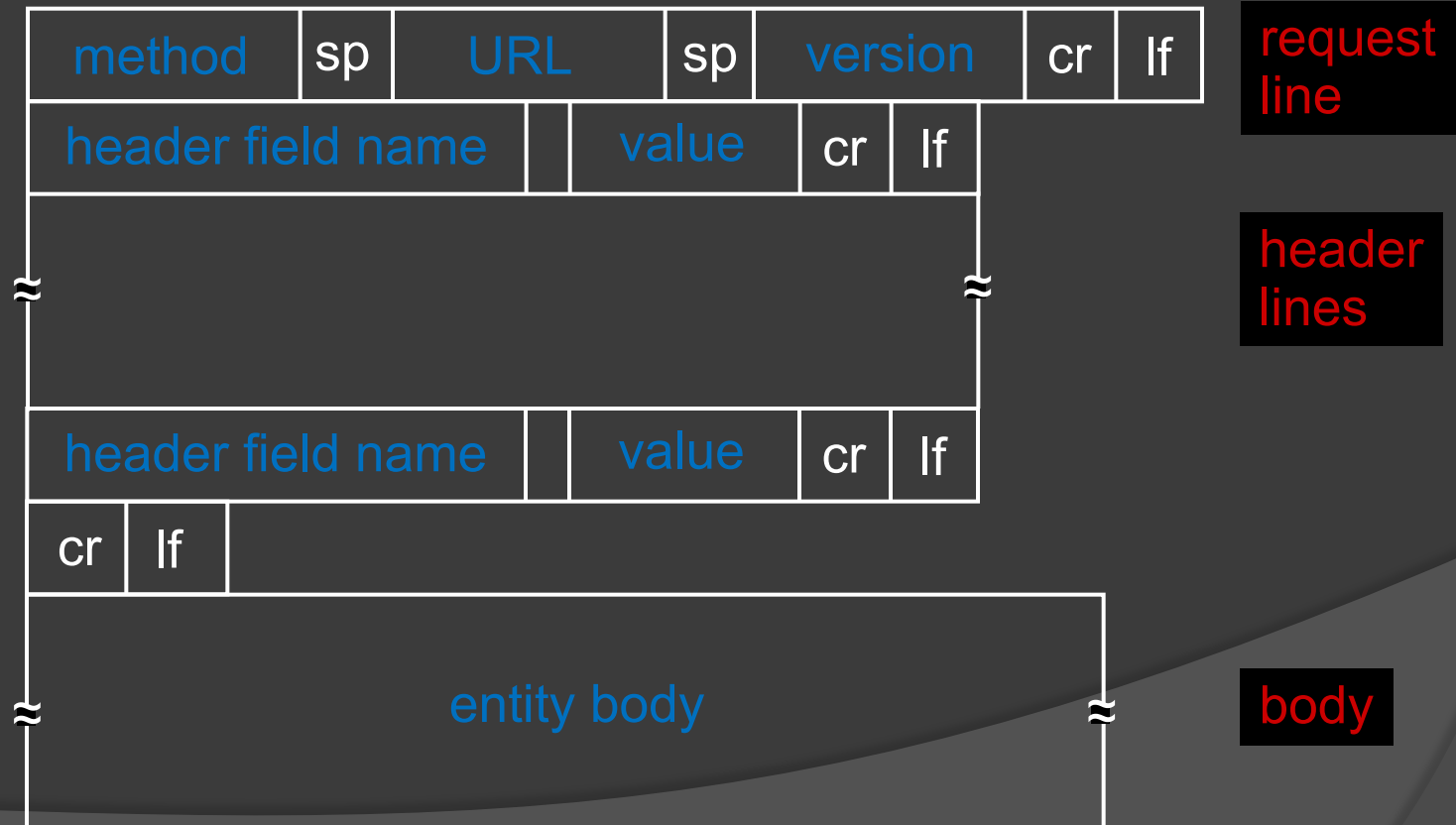
status line
(protocol
status code
status phrase)

header
lines

data, e.g.,
requested
HTML file

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
      GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
      1\r\n
\r\n
data data data data data ...
```

HTTP Message format



HTTP Functions And Replies

⦿ Functions

- GET
 - url ? query
- POST
 - url
 - Post data
- HEAD
 - Request headers
- PUT
 - Move data to server

⦿ Reply Codes

- 100
 - informational
- 200
 - OK
- 300
 - Redirection
- 400
 - Request Error
- 500
 - Server Error