BRAC UNIVERSITY

Inspiring Excellence

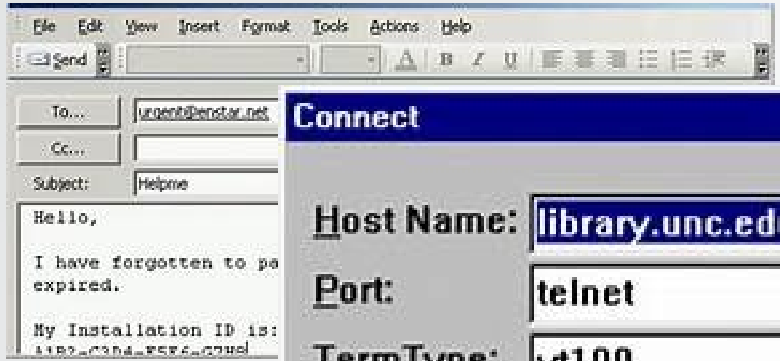
Application Layer (HTTP)

Lecture 3 | Part 1 of 2 | CSE421 – Computer Networks

Department of Computer Science and Engineering
School of Data & Science

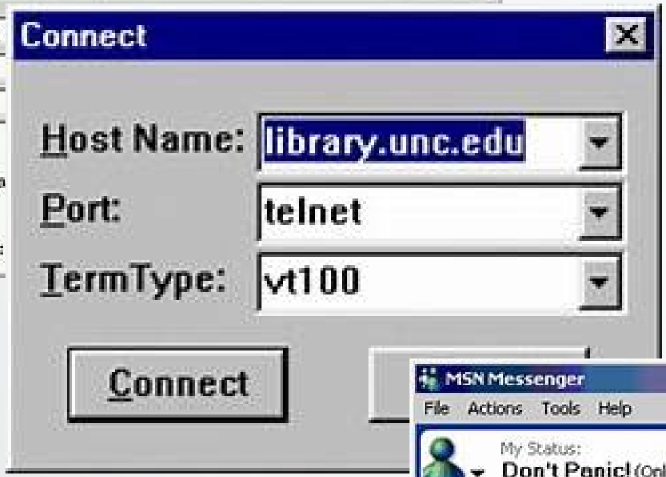
Applications

- 1970-1980
- 1990-2000
- 2000-2020

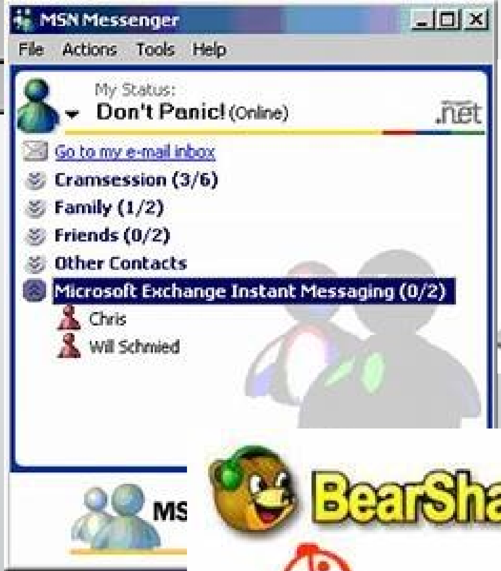


To... urgent@enstar.net
Cc...
Subject: Helpme

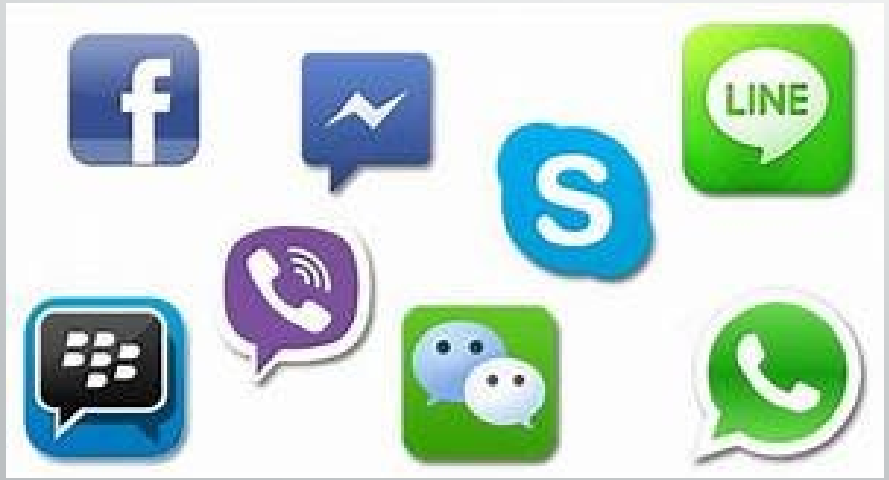
```
Hello,

I have forgotten to pa
expired.

My Installation ID is:
```

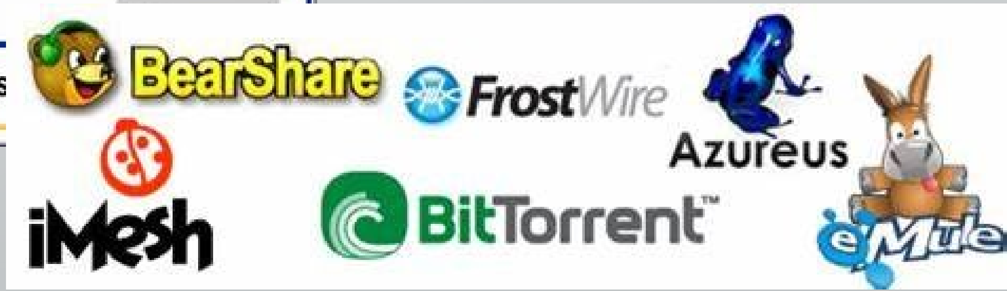


Host Name: library.unc.edu
Port: telnet
TermType: vt100



My Status:
Don't Panic! (Online)
Go to my e-mail inbox
Cramsession (3/6)
Family (1/2)
Friends (0/2)
Other Contacts
Microsoft Exchange Instant Messaging (0/2)
Chris
Will Schmied

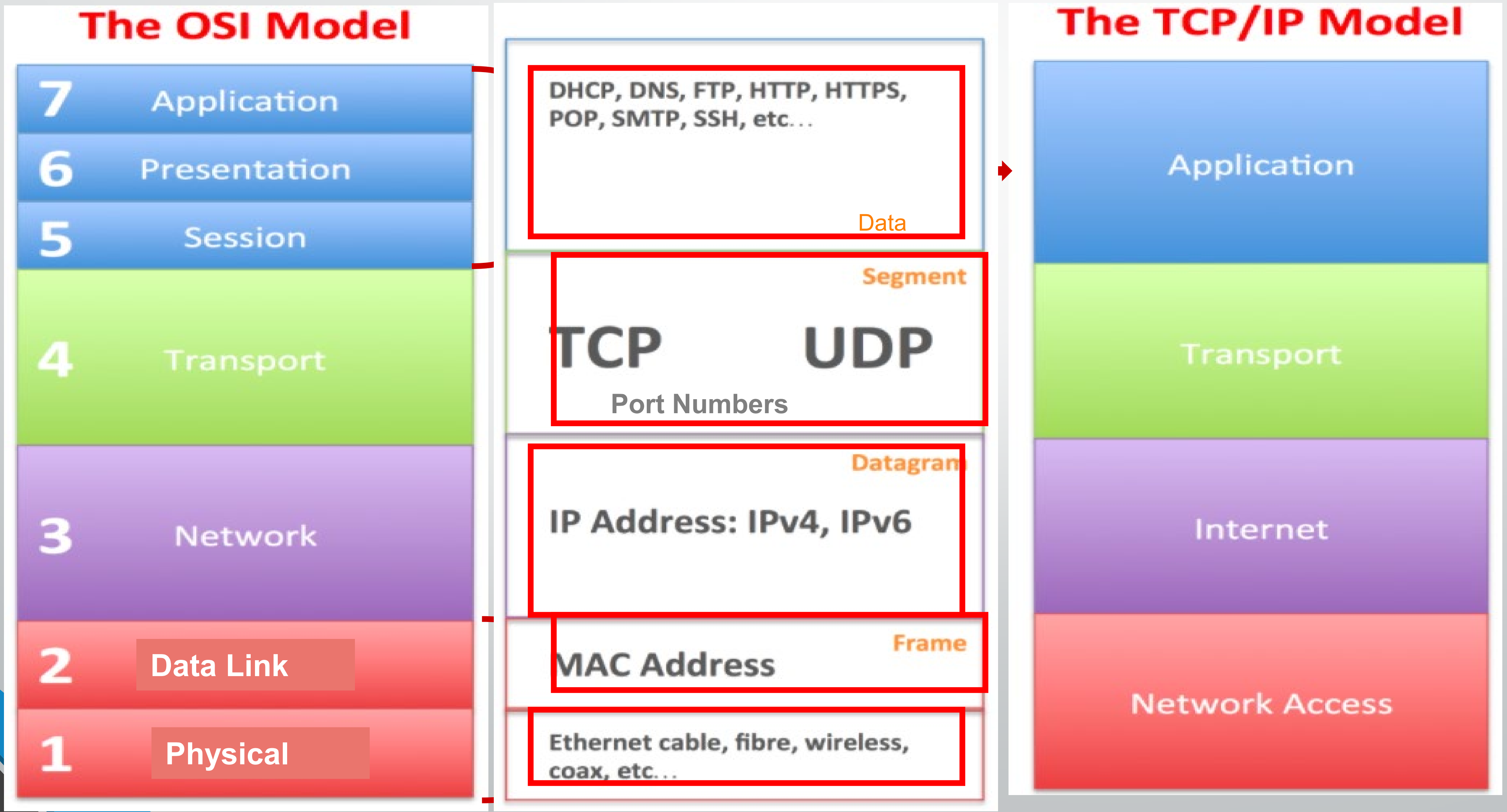




Objectives (Application Layer)

- Principles of network applications
- Web and HTTP
- Electronic Mail (SMTP, POP3, IMAP)
- DNS
- P2P Applications
- Video streaming and content distribution networks
- Socket Programming with UDP and TCP

Network Models

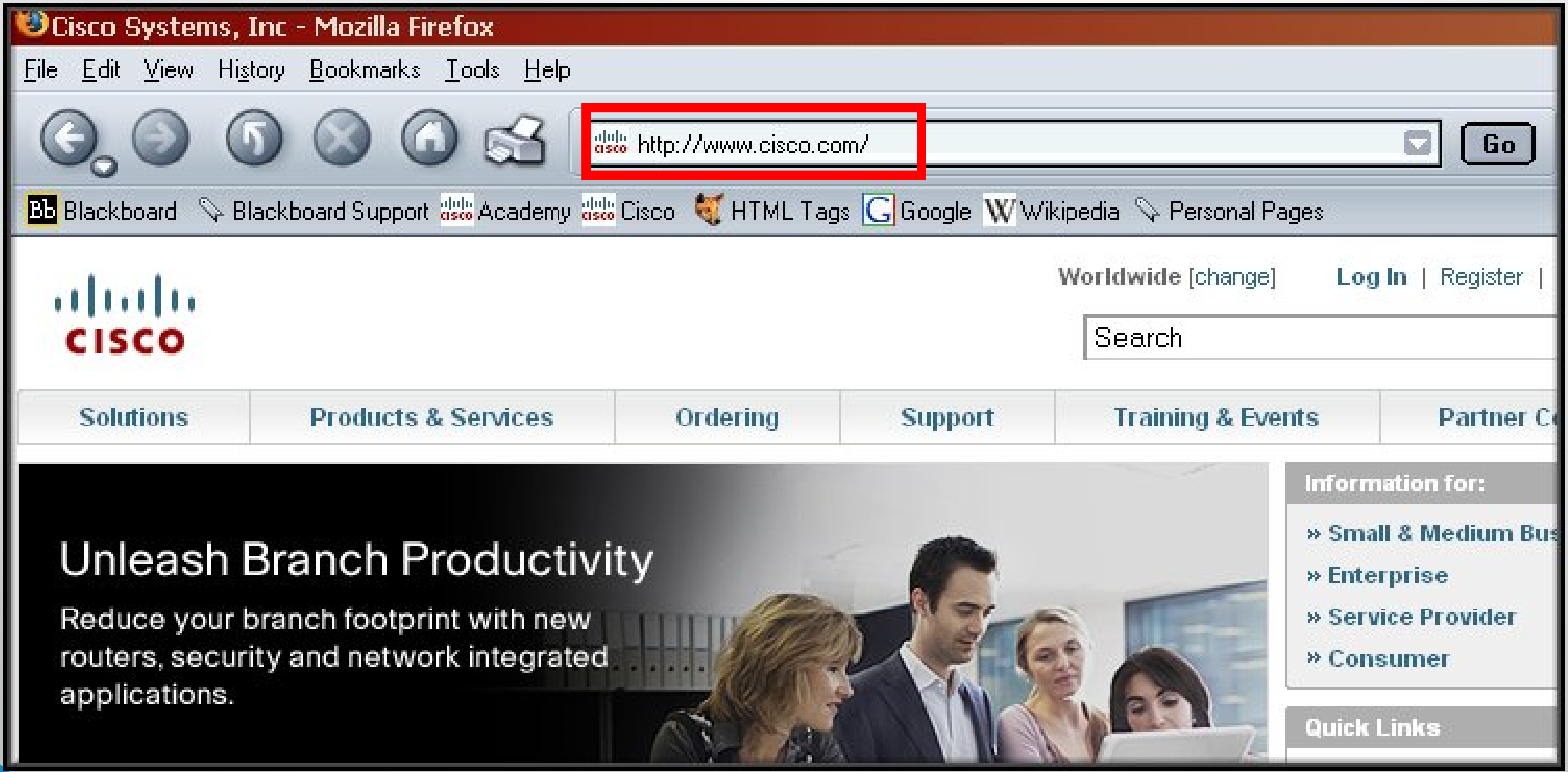


Application Layer

- Application Layer Protocols
  - Provide the rules and formats that govern how data is treated in the application layer.
- Application Software
  - The programs used to communicate over the network.

- For example:
  - When displaying a web page:
    - The Application Layer uses the HTTP(Hyper Text Transfer Protocol) Protocol.
    - The Application Software is your Web browser.

Application Layer

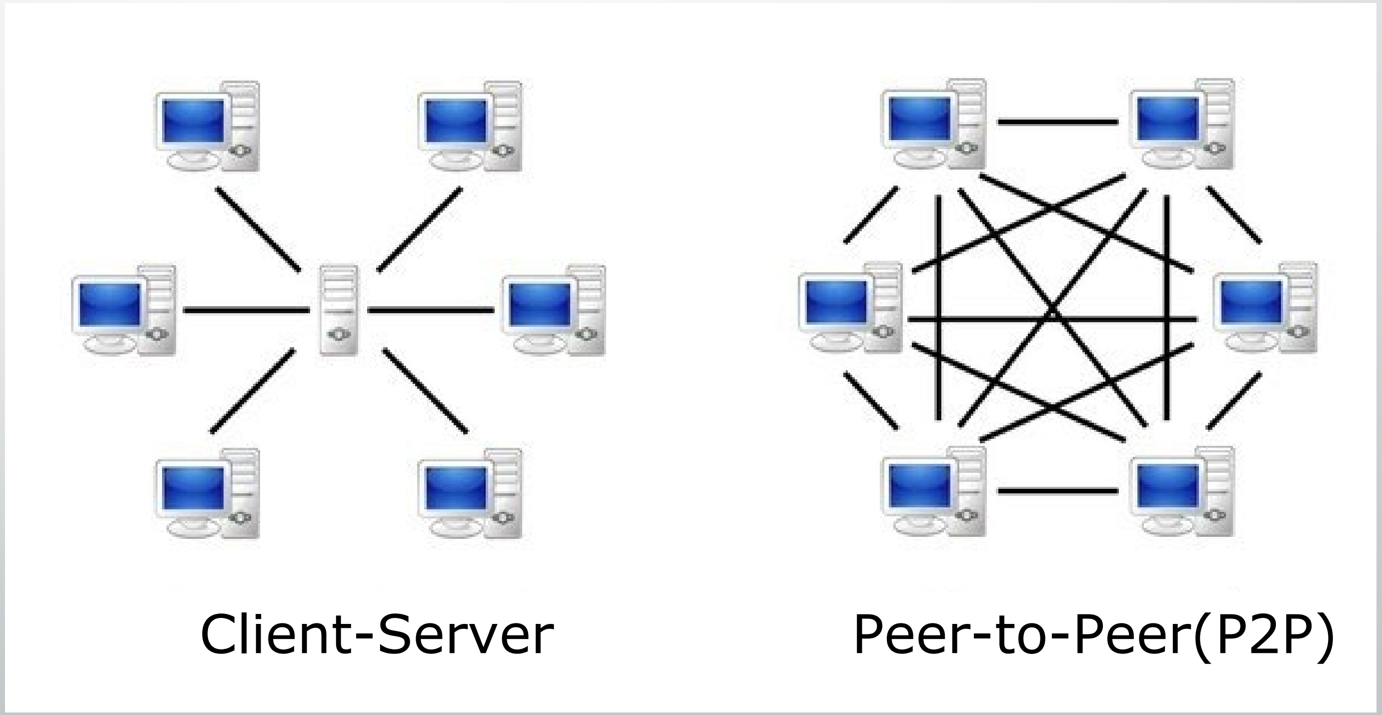


Unleash Branch Productivity

Reduce your branch footprint with new routers, security and network integrated applications.

Application Layer

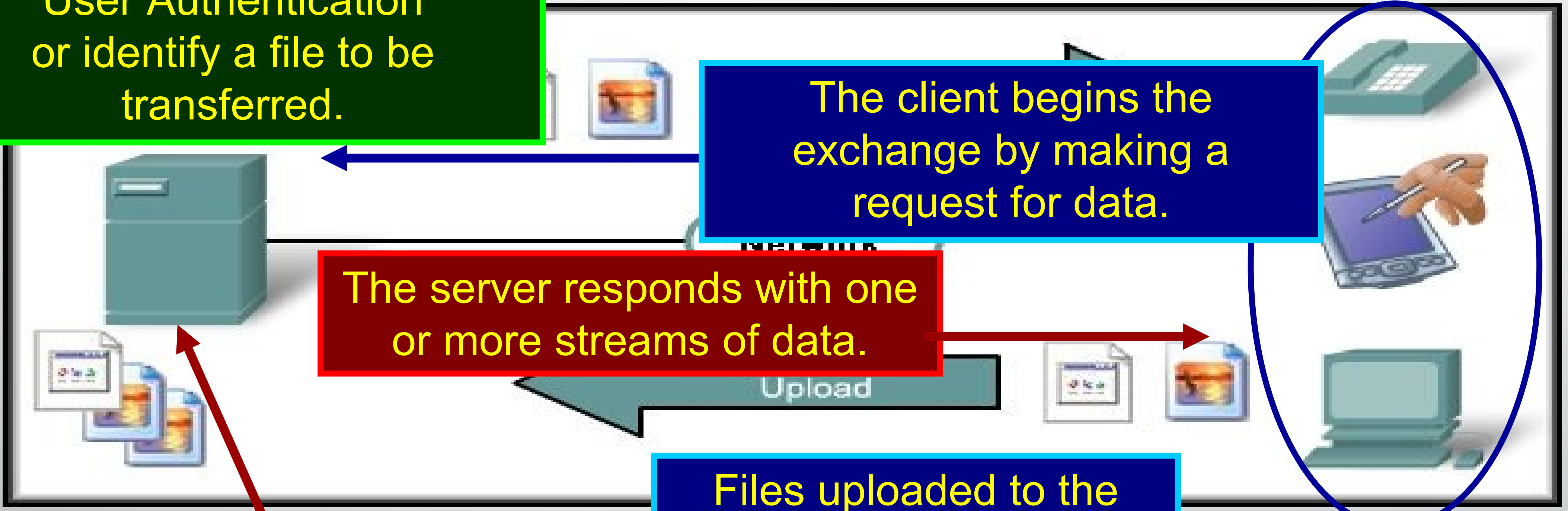
- When accessing information on a device, the data may not be physically stored on that device.
- If that is the case, a request must be made to the device where the data resides.
- **Two methods:**
  - Client/Server
  - Peer-to-Peer (P2P)



Client/Server Model

May also require control information.
User Authentication
or identify a file to be transferred.

Clients – hardware, software combination



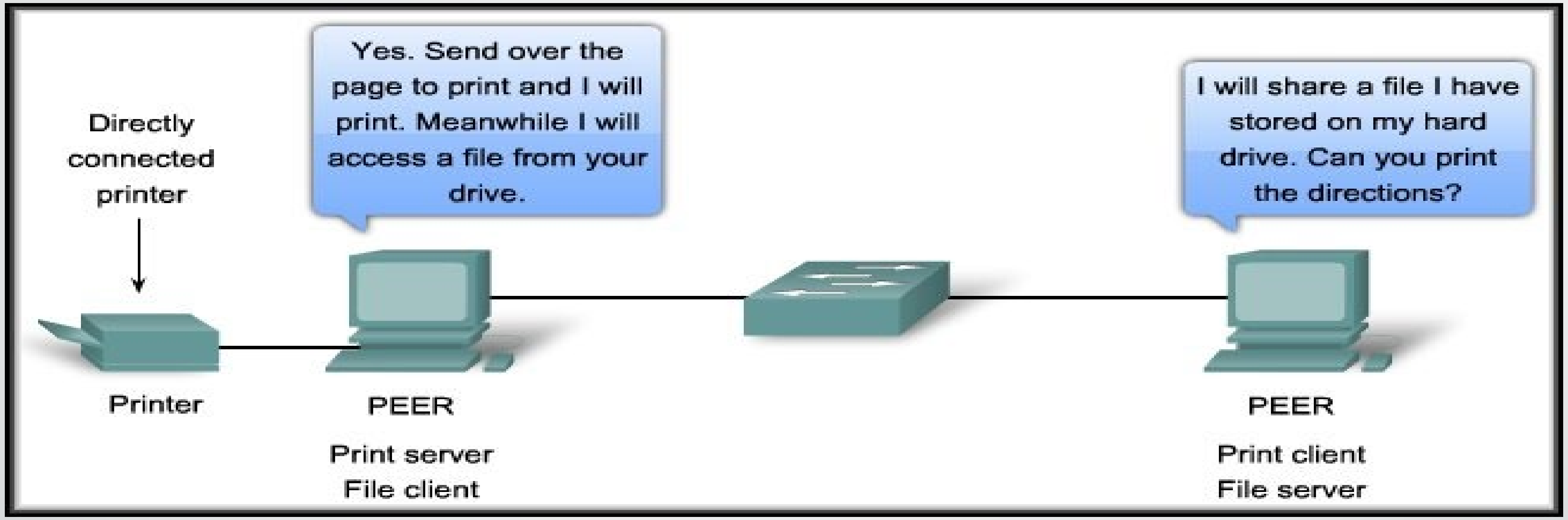
The client begins the exchange by making a request for data.

The server responds with one or more streams of data.

Files uploaded to the server

Resources are stored on the server.

Peer-to-Peer Model


Directly connected printer
Yes. Send over the page to print and I will print. Meanwhile I will access a file from your drive.
I will share a file I have stored on my hard drive. Can you print the directions?
Printer
PEER
Print server
File client
PEER
Print client
File server

- Two or more computers are connected via a network and can share resources (such as printers and files) *without having a dedicated server*.
- End devices (peers) can function as either a *server or client* depending upon the required service.

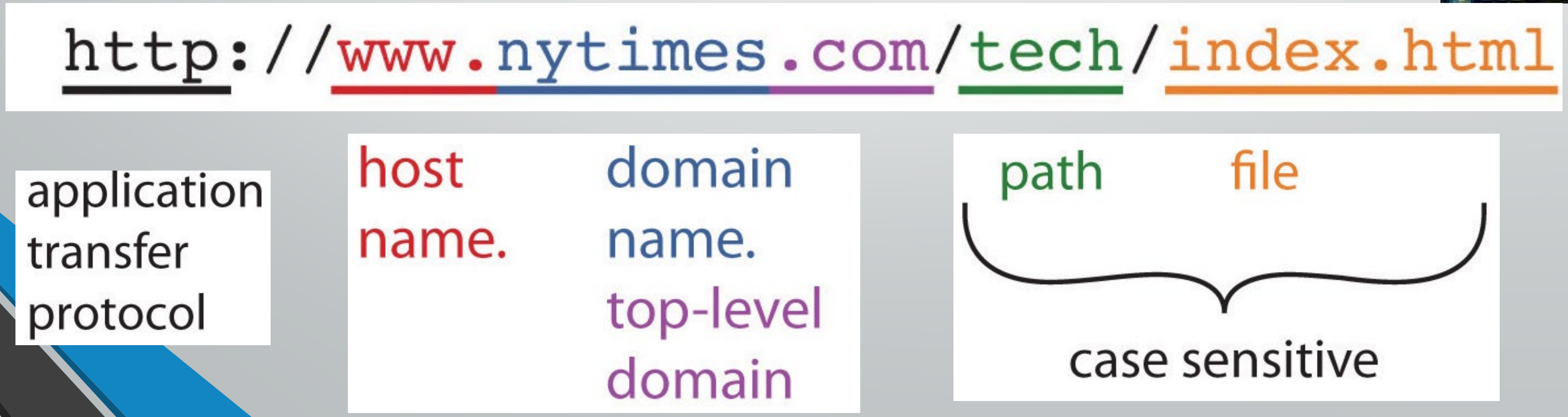
Web and HTTP

Objectives (HTTP-Part 1)

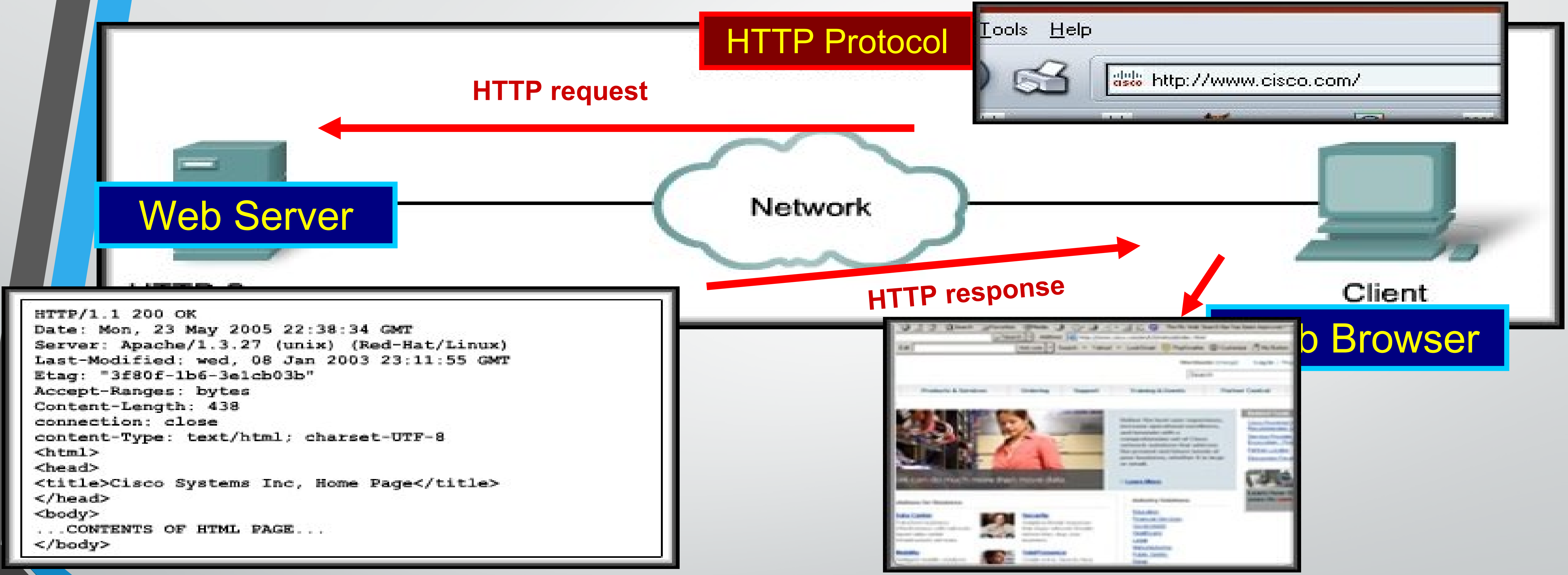
- WWW – The Web
- HTTP
- HTTP Connections
- Persistent HTTP Connections
- Non Persistent HTTP Connections

WWW- The Web

- *Web page* consists of *objects*
- Object can be HTML file, JPEG image, Java applet, audio file,…
- Web page consists of *base HTML-file* which includes *several referenced objects*
- Each object is addressable by a *URL Uniform Resource Locater*, e.g.,


http://www.nytimes.com/tech/index.html
application transfer protocol
host name.
domain name.
top-level domain
path
file
case sensitive

HTTP



```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Server: Apache/1.3.27 (unix) (Red-Hat/Linux)
Last-Modified: wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Content-Length: 438
connection: close
content-Type: text/html; charset-UTF-8
<html>
<head>
<title>Cisco Systems Inc, Home Page</title>
</head>
<body>
...CONTENTS OF HTML PAGE...
</body>
```

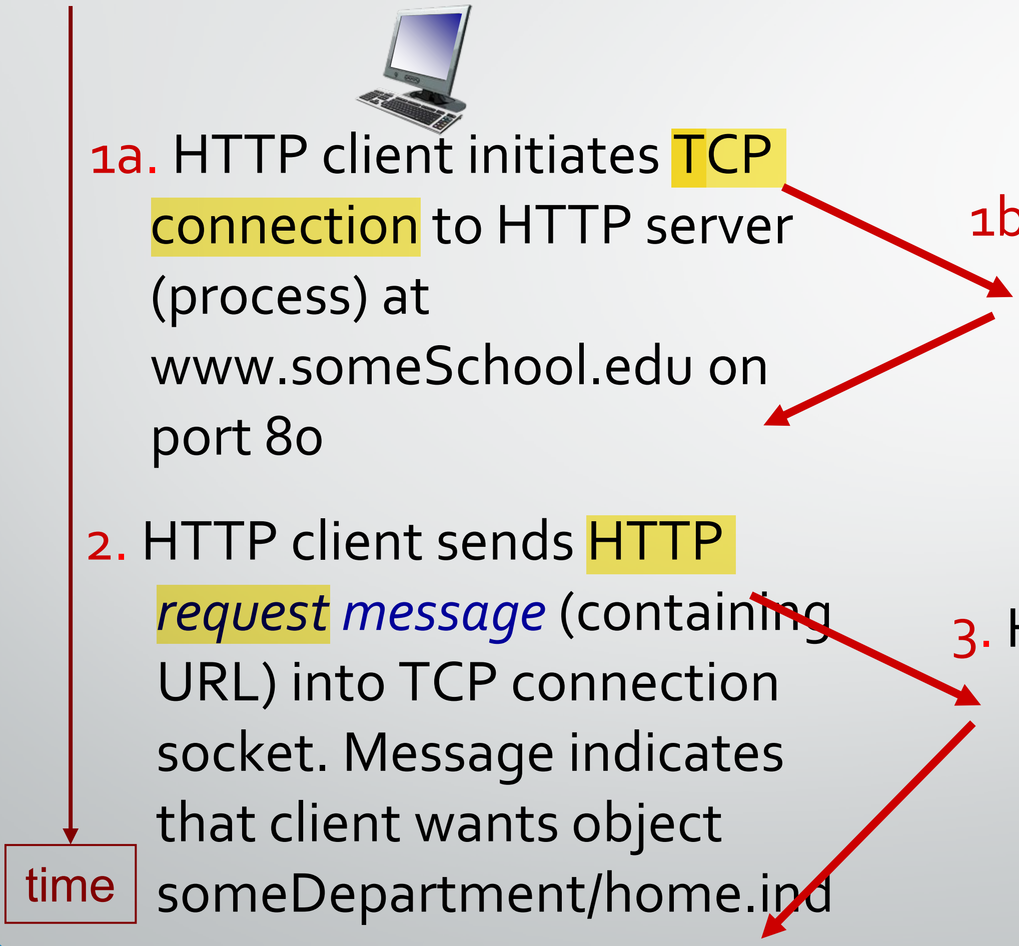
- Web browsers are the client applications used to interpret the HTTP application protocol received from a web server.

HTTP Connections

Suppose user enters URL:

`www.someSchool.edu/someDepartment/home.index`

(contains text, references to 10 jpeg images)



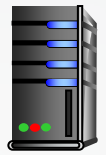
1a. HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. HTTP client sends HTTP *request message* (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request message, forms *response message* containing requested object, and sends message into its socket

HTTP Connections (cont.)



4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

time

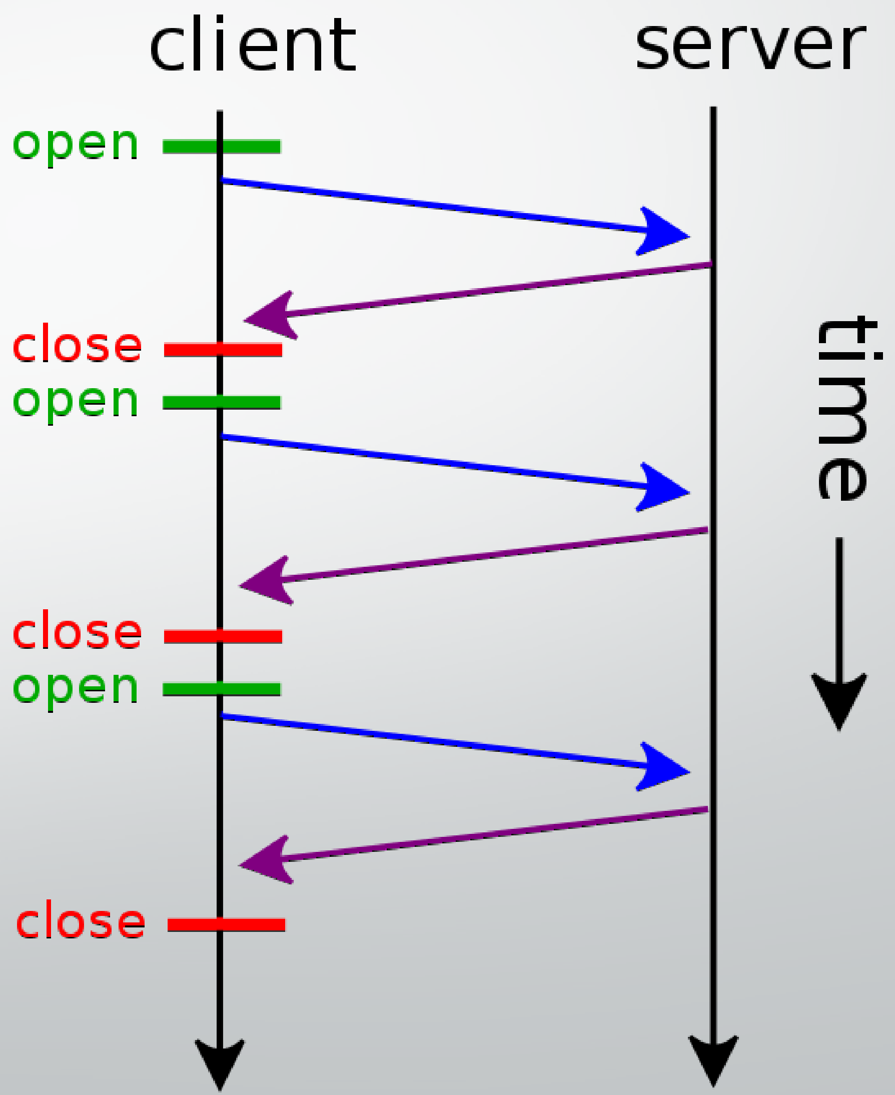
Non-persistent HTTP

HTTP connections

Non-persistent HTTP

- At most one object sent over TCP connection
- Connection is then closed
- Downloading multiple objects required multiple connections

open ---—TCP Connection Request
close ---— TCP Termination Request

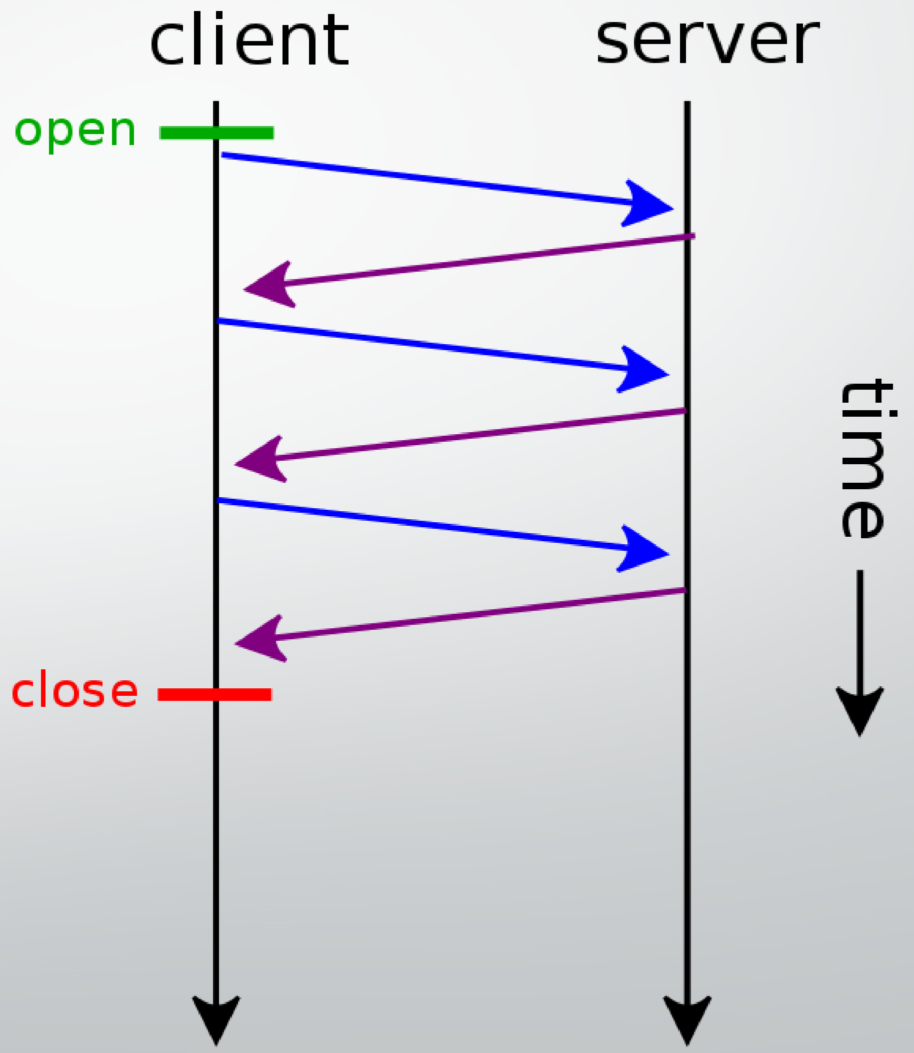


HTTP connections

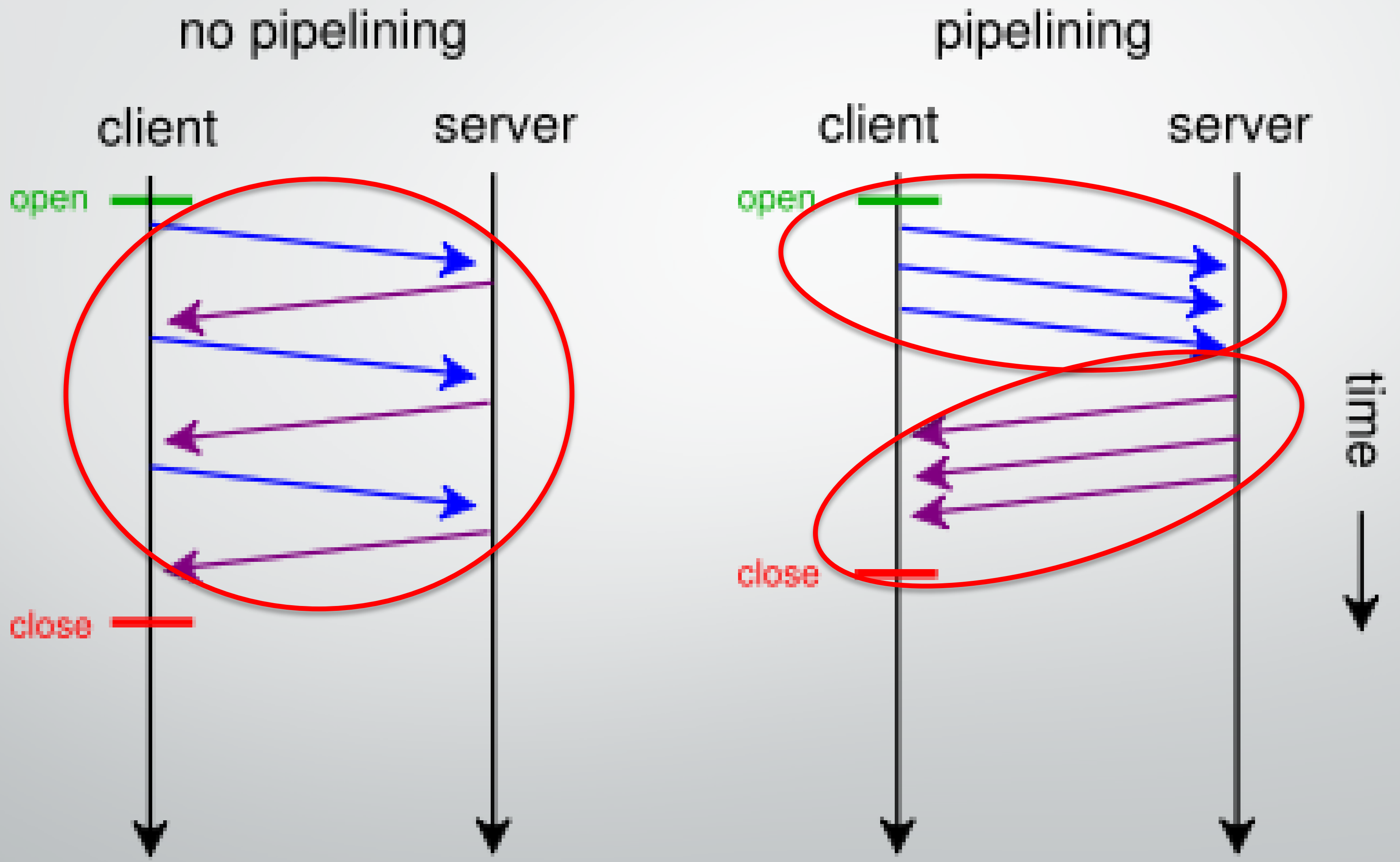
Persistent HTTP

- Multiple objects can be sent over single TCP connection between client, server

open ---–TCP Connection Request
close ---– TCP Termination Request



HTTP connections



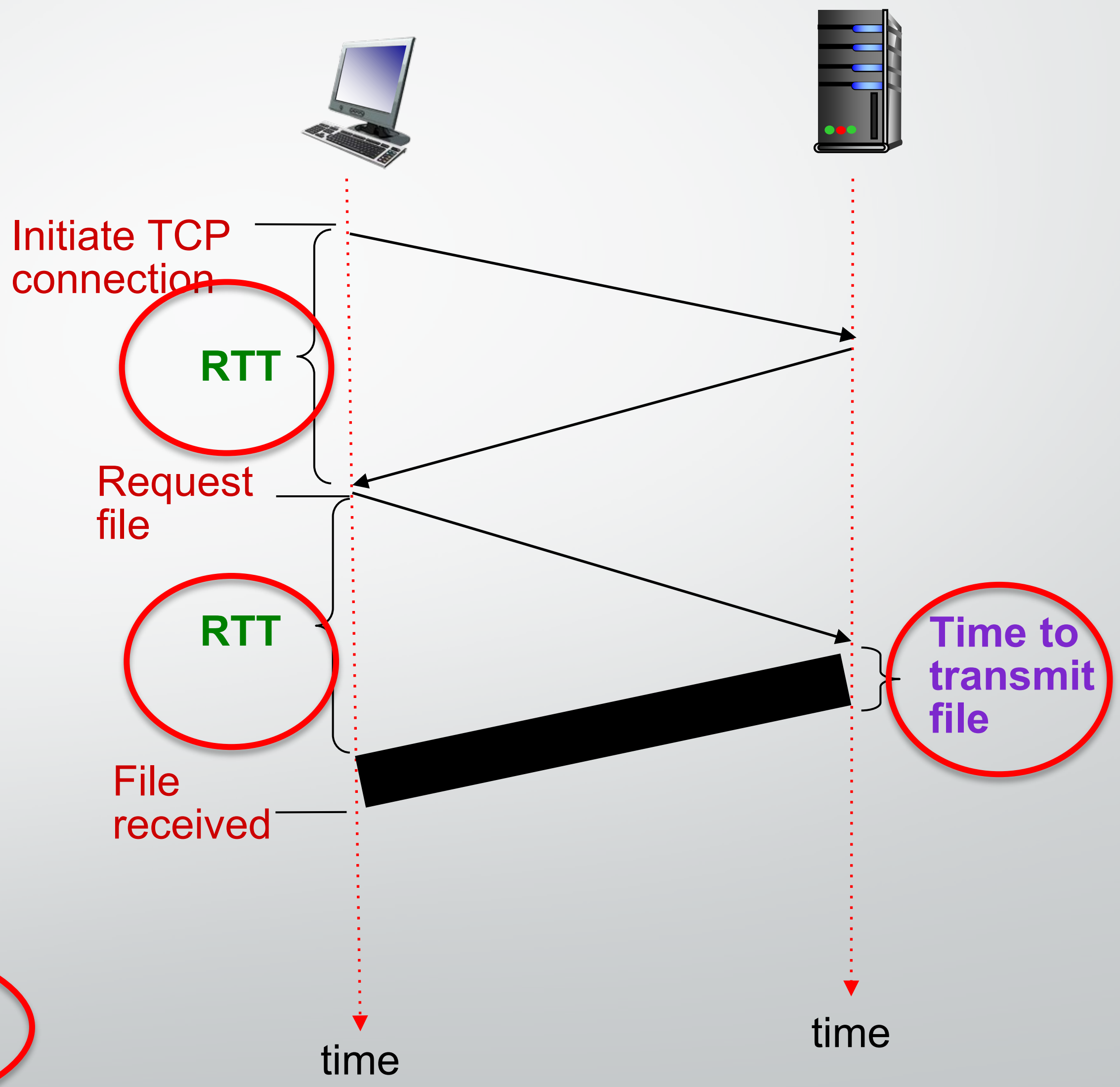
Non-persistent HTTP: response time

RTT (Round Trip Time): time for a small packet to travel from client to server and back

HTTP response time:

- One RTT to initiate TCP connection
- One RTT for HTTP request and first few bytes of HTTP response to return
- File transmission time
- Non-persistent HTTP response time =

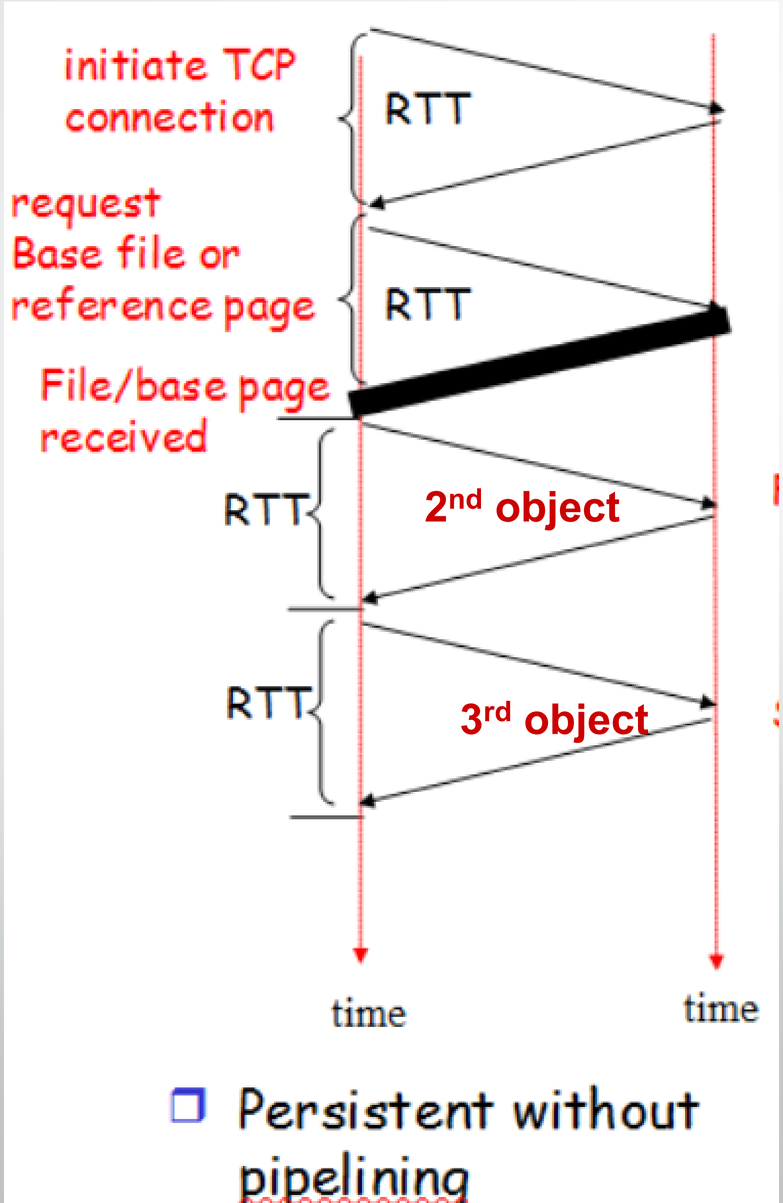
  2*RTT+ file transmission time



Persistent HTTP

Non-persistent HTTP issues:

- Requires 2 RTTs per object
- OS overhead for *each* TCP connection
- Browsers often open parallel TCP connections to fetch referenced objects



❒ Persistent without pipelining

Persistent HTTP:

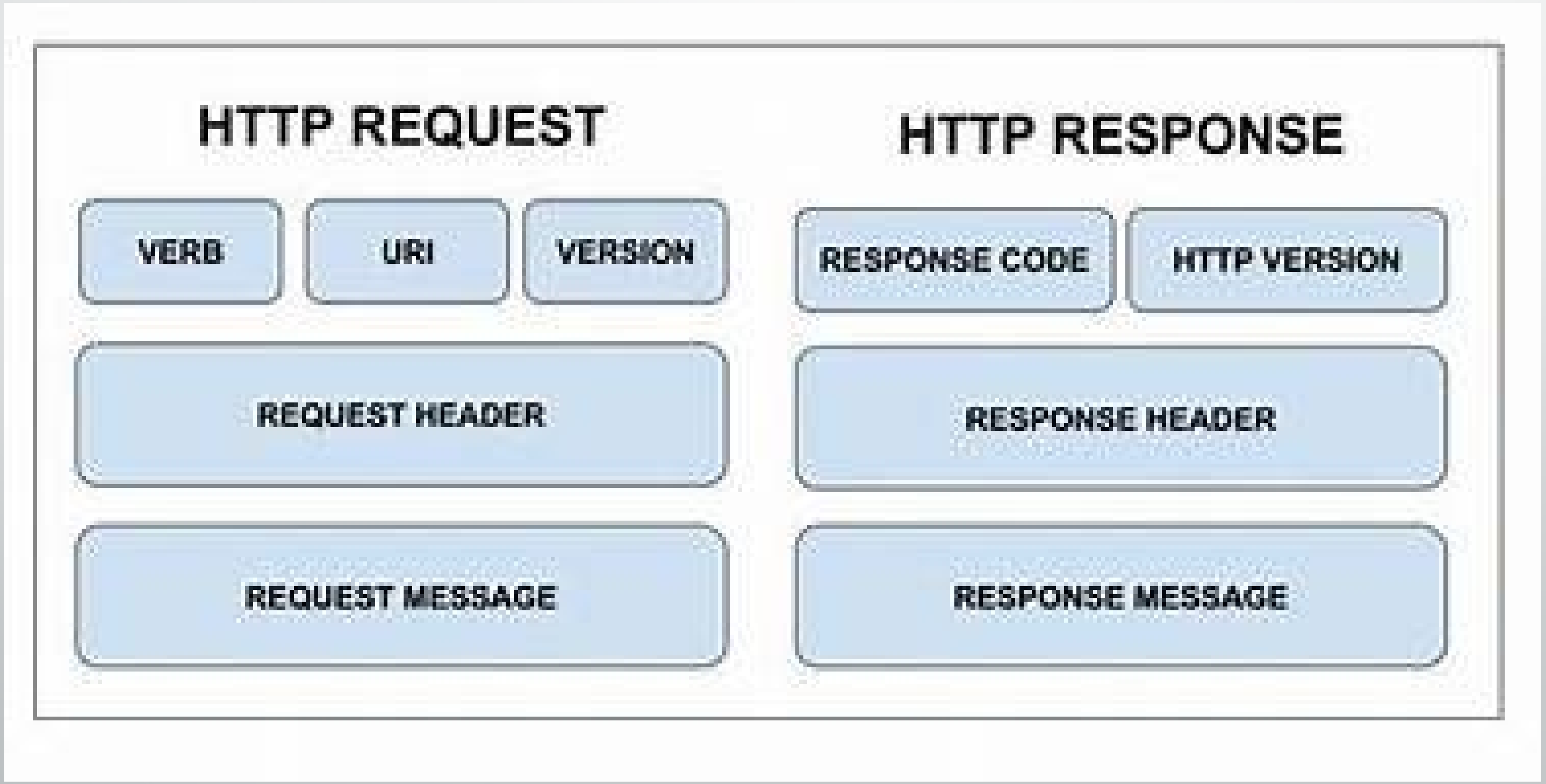
- Server leaves connection open after sending response
- Subsequent HTTP messages between same client/server sent over open connection
- Client sends requests as soon as it encounters a referenced object
- As little as one RTT for all the referenced objects

Objectives – Part 3

- HTTP Message Formats
- HTTP Request Message
- HTTP Methods
- HTTP Response Message

HTTP messages

- Two types of HTTP messages:
- *Request* and *Response*



HTTP request message

- **HTTP request message:**
  - ASCII (human-readable format)

carriage return character

line-feed character

request line (GET, POST, HEAD,PUT,DELETE commands)

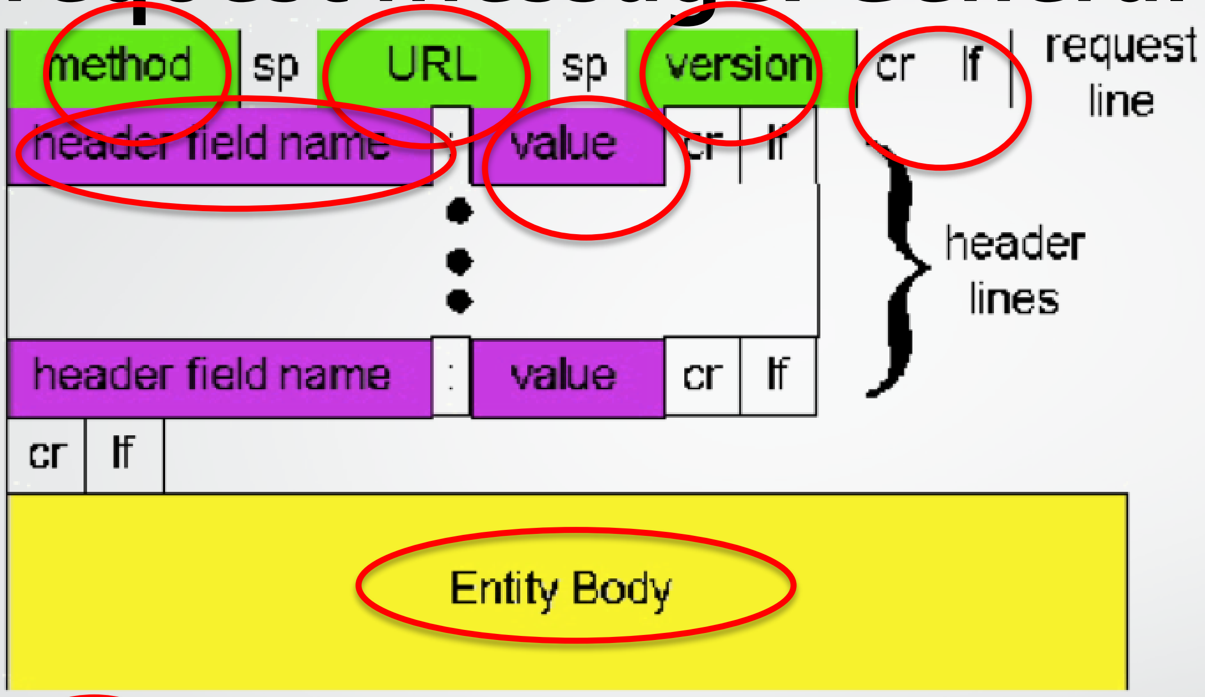
header lines

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

```
GET /somedir/index.html HTTP/1.1\r\n
Host: www.someschool.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,image/gif, image
  jpeg\r\n
Accept-Language: fr \r\n
Connection: close\r\n

\r\n
```

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

HTTP request message: General Format



```
GET /somedir/index.html HTTP/1.1\r\n
Host: www.someschool.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,image/gif, image jpeg\r\n
Accept-Language: fr \r\n
Connection: close\r\n
\r\n
```

Uploading form input

POST method:

- Web page often includes form input
- Input is uploaded to server in entity body

URL method:

- Uses GET method
- Input is uploaded in URL field of request line:

```
www.somesite.com/animalsearch?monkeys&banana
```

Method types

HTTP/1.0:

- GET
  - Primarily gets information only
  - URL Method of data insertion
- POST
  - Creating new data
- HEAD
  - Asks server to leave requested object out of response

HTTP/1.1:

- GET, POST, HEAD
- PUT
  - Uploads file in entity body to path specified in URL field
  - Replaces existing objects
- DELETE
  - Deletes file specified in the URL field

HTTP response message

Status line (protocol status code status phrase)

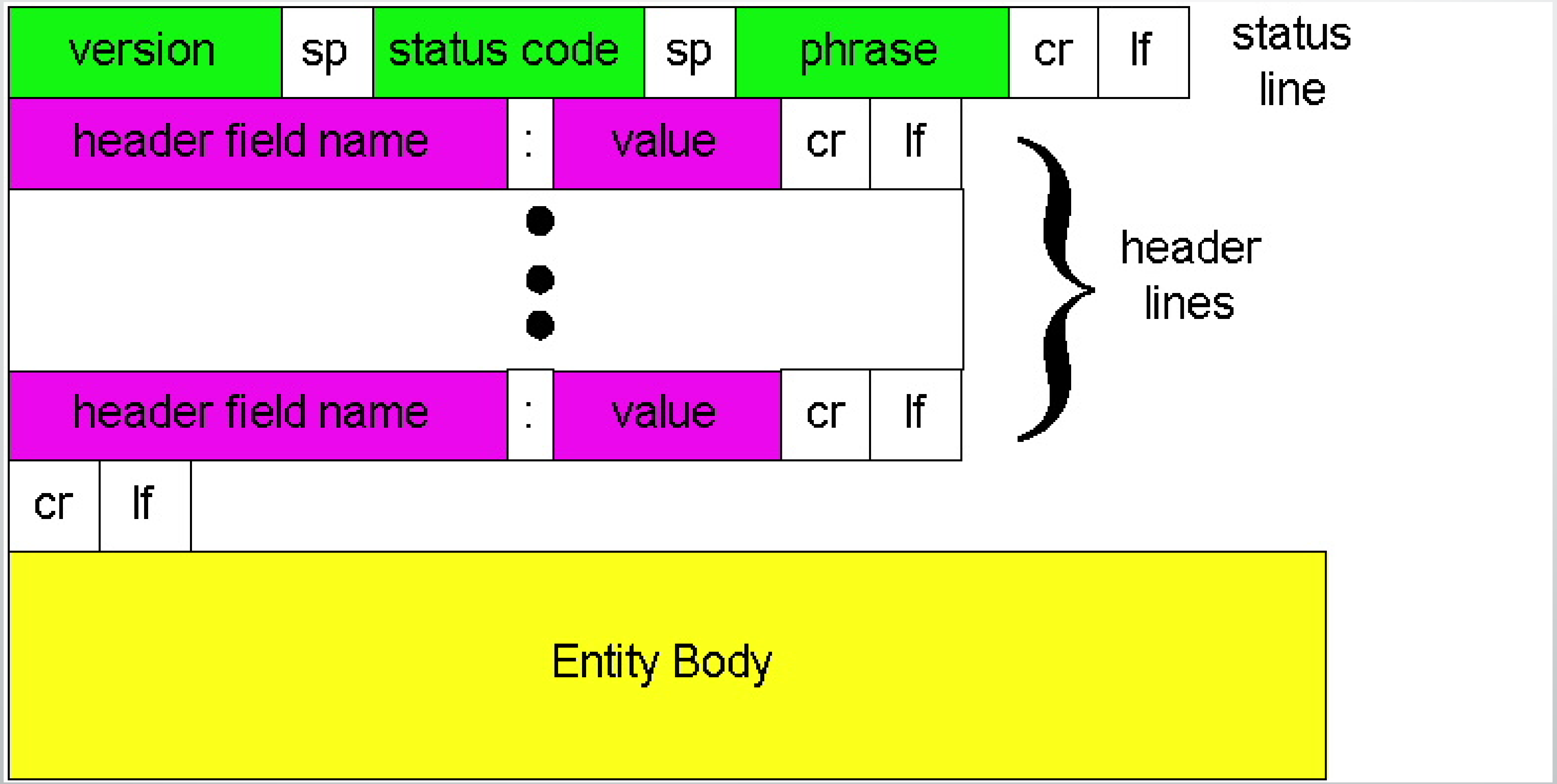
Header lines

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: close\r\n
Content-Type: text/html; charset=ISO-8859-
   1\r\n
\r\n
data data data data data ...
```

* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

HTTP response message: General Format



HTTP response status codes

- Status code appears in 1st line in server-to-client response message.
- Some sample codes:

200 OK

- request succeeded, requested object later in this message

301 Moved Permanently

- requested object moved, new location specified later in this message (Location:)

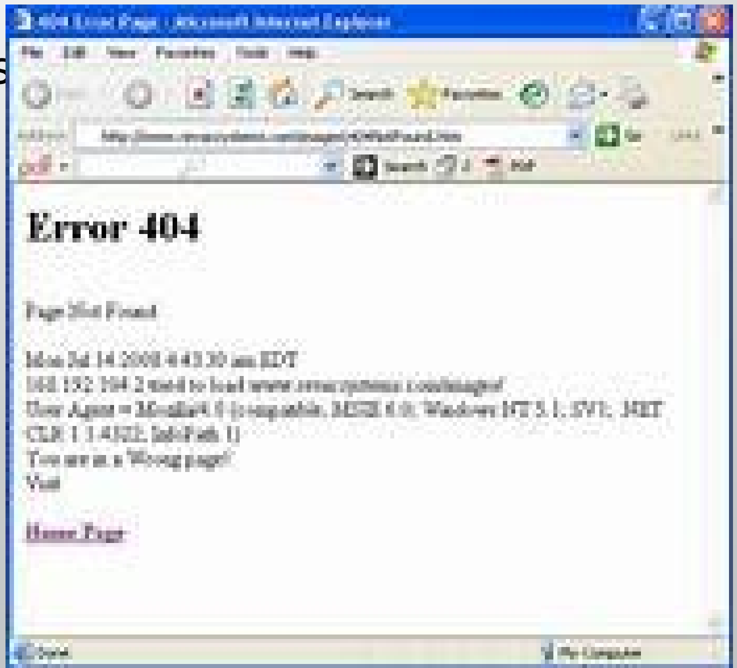
400 Bad Request

- request message not understood by server

404 Not Found

- requested document not found on this server

505 HTTP Version Not Supported


404 Error Page - Microsoft Internet Explorer

Error 404

Page Not Found

Mon Jul 14 2008 4:43:30 am EDT
[illegible] tried to load www.[illegible]
User Agent = Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; InfoPath.1)
You are at a Wrong page!
Visit

Home Page