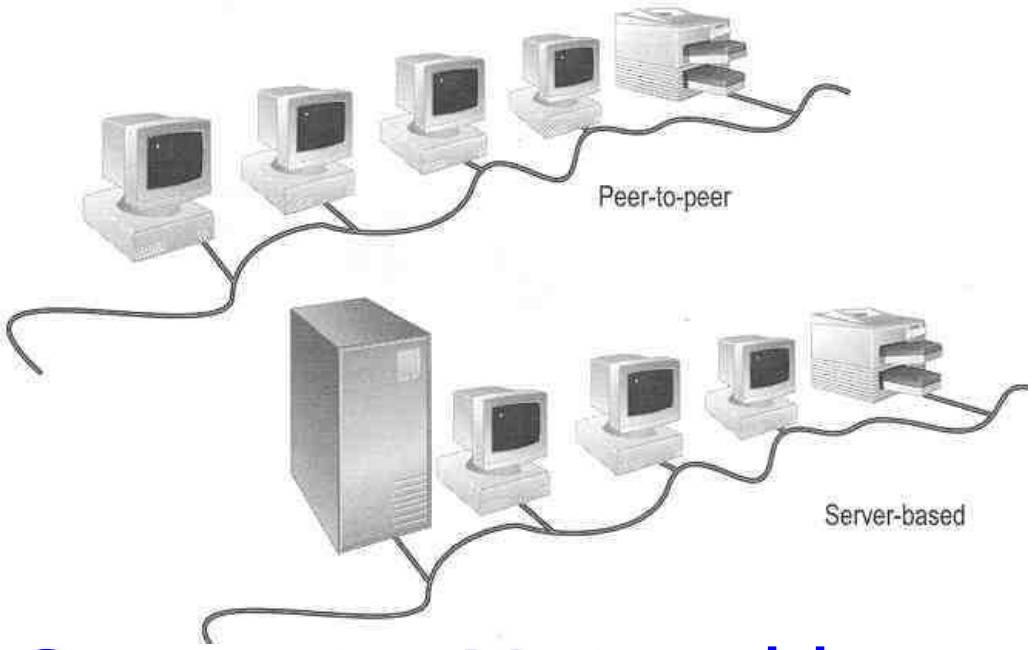


Computer Networking
Lecture 3
Hassan Alamri

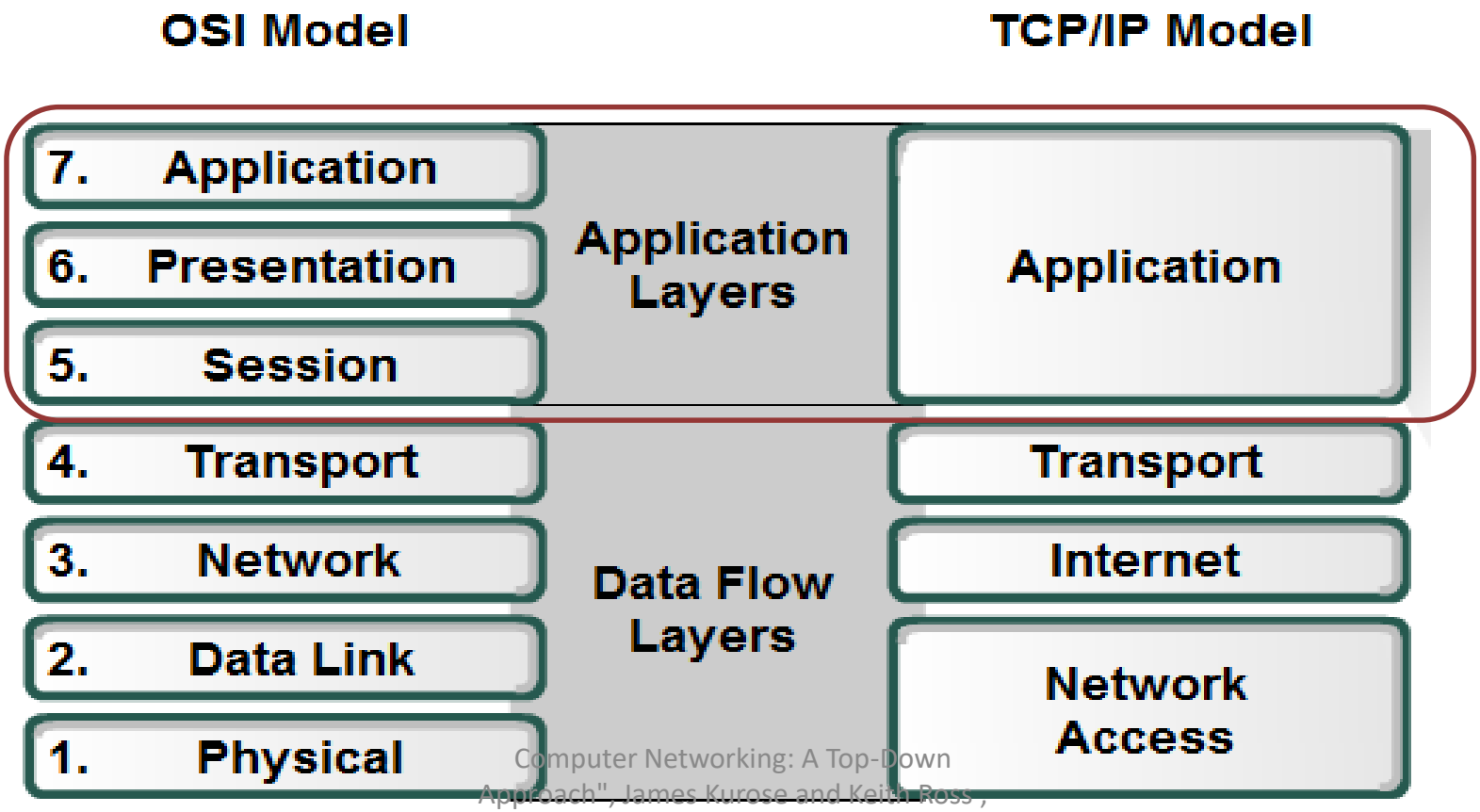
Agenda:

Application Layer (7)

- Application Layer principles
- Application Layer protocols:
  - HTTP
  - DNS
- End

Application (Layer 7)

The application layer is responsible for providing services to the user. It is the interface between applications we use and underlying network.



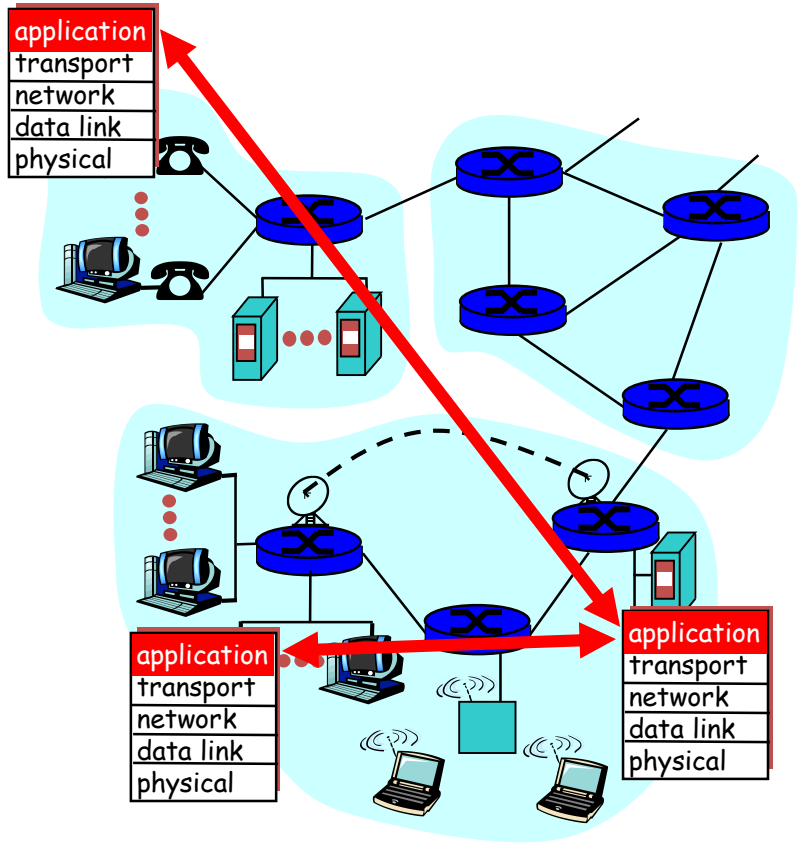
Applications and application-layer protocols*

Application: communicating, distributed processes

- running in network hosts in "user space"
- exchange messages to implement app
- e.g., email, file transfer, the Web

Application-layer protocols

- one "piece" of an app
- define messages exchanged by apps and actions taken
- user services provided by lower layer protocols



Network applications: some jargon**

- A **process** is a program that is running within a host.
- Within the same host, two processes communicate with **interposes communication** defined by the OS.
- Processes running in different hosts communicate with an **application-layer protocol**

- A **user agent** is an interface between the user and the network application.
  - Web: browser
  - E-mail: mail reader
  - streaming audio/video: media player

Q: how does a process "identify" the other process with which it wants to communicate?

- IP address of host running other process
- "port number" - allows receiving host to determine to which local process the message should be delivered

Client-server paradigm

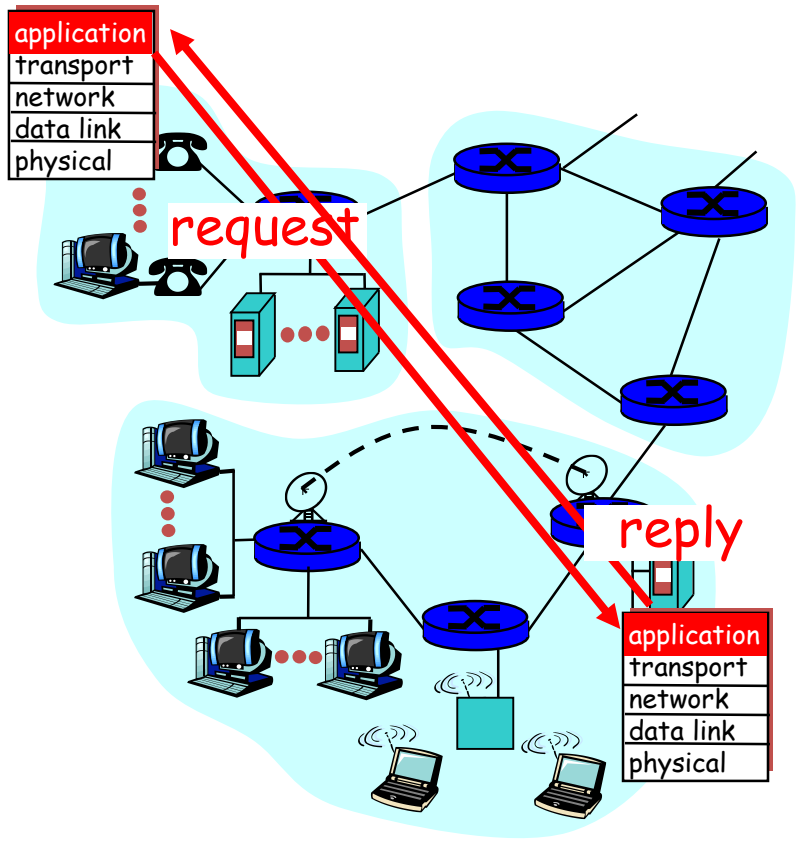
Typical network app has two pieces:
client and *server*

Client:

- initiates contact with server ("speaks first")
- typically requests service from server,
- for Web, client is implemented in browser; for e-mail, in mail reader

Server:

- provides requested service to client
- e.g., Web server sends requested Web page, mail server delivers e-mail



The Web: some jargon*

- Web page:
  - ✓ consists of “objects”
  - ✓ addressed by a URL
- Most Web pages consist of:
  - ✓ base HTML page, and
  - ✓ several referenced objects.
- URL has two components: host name and path name:

- User agent for Web is called a browser:
  - ✓ MS Internet Explorer
  - ✓ Netscape Communicator
- Server for Web is called Web server:
  - ✓ Apache (public domain)
  - ✓ MS Internet Information Server

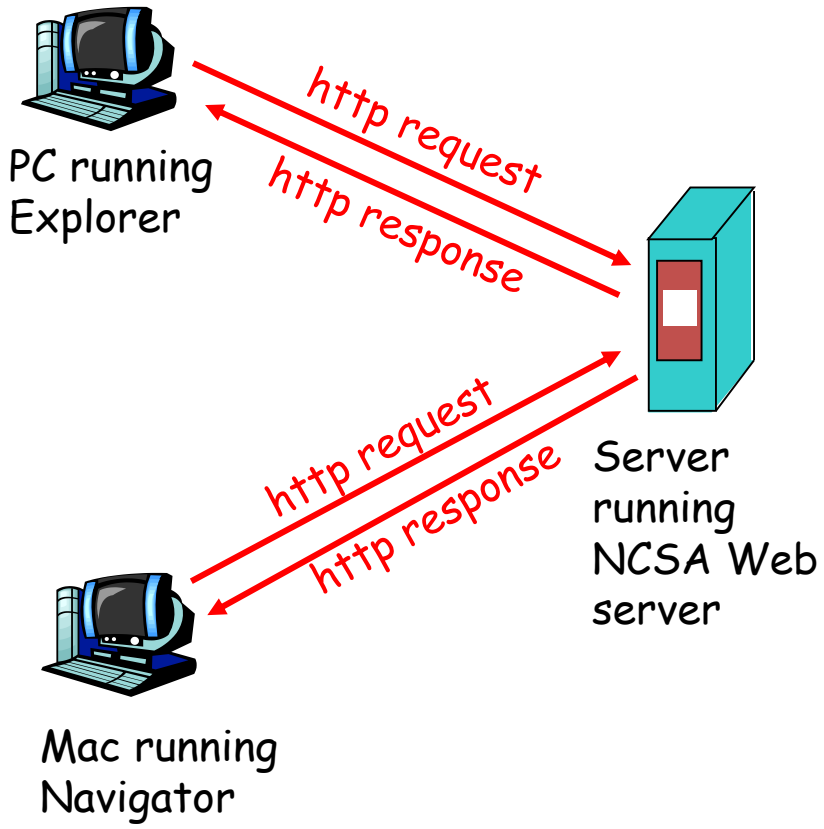
www.someSchool.edu/someDept/pic.gif

Application Layer protocol

HTTP protocol

http: hypertext transfer protocol

- Web's application layer protocol
- client/server model
  - *client:* browser that requests, receives, "displays" Web objects
  - *server:* Web server sends objects in response to requests
- http1.0: RFC 1945
- http1.1: RFC 2068



The http protocol: more

http: TCP transport service:

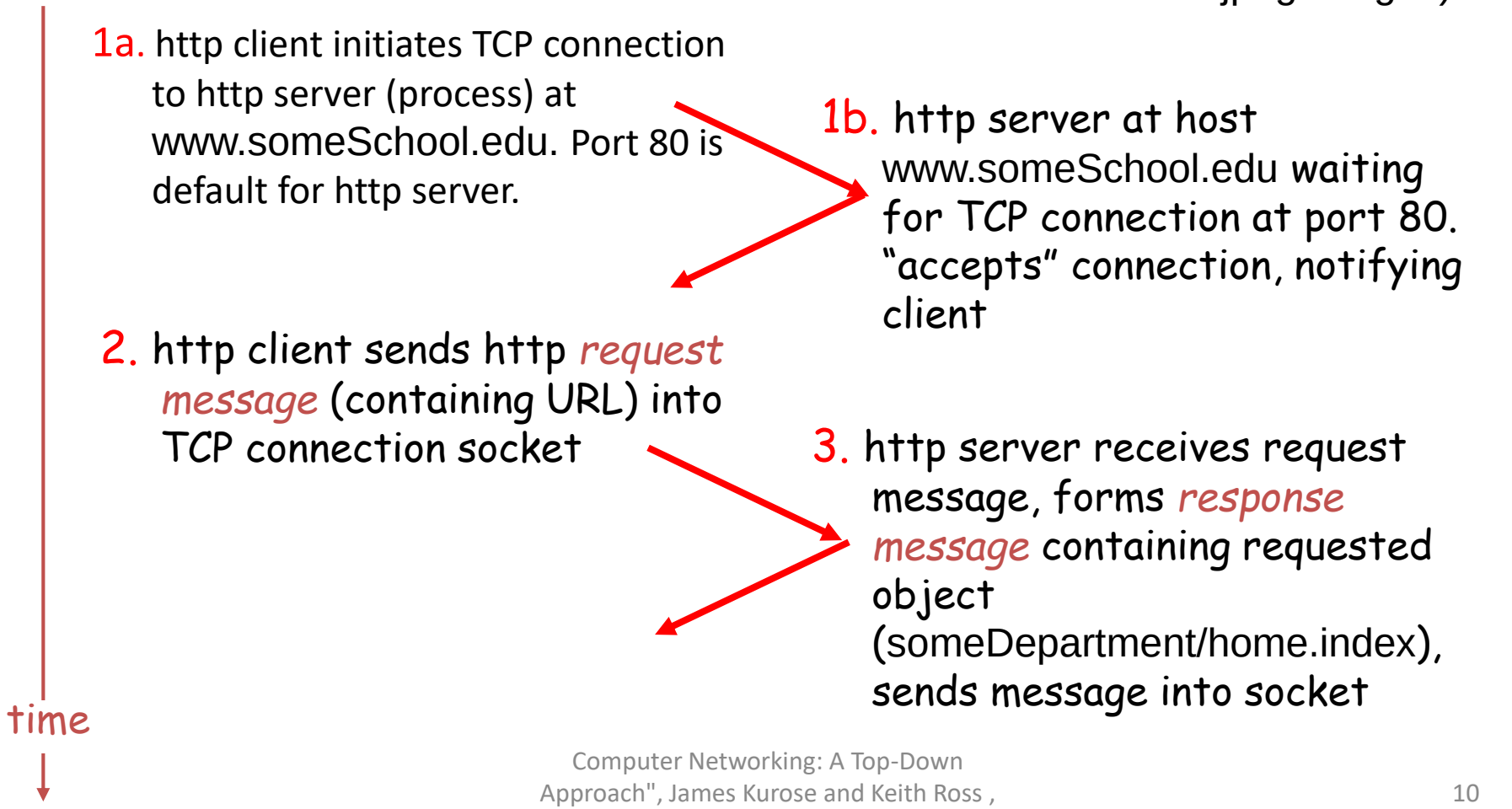
- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- http messages (application-layer protocol messages) exchanged between browser (http client) and Web server (http server)
- TCP connection closed

http is “stateless”

- server maintains no information about past client requests

http example

Suppose user enters URL www.someSchool.edu/someDepartment/home.index (contains text, references to 10 jpeg images)



http example (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 and persistent connections**

Non-persistent

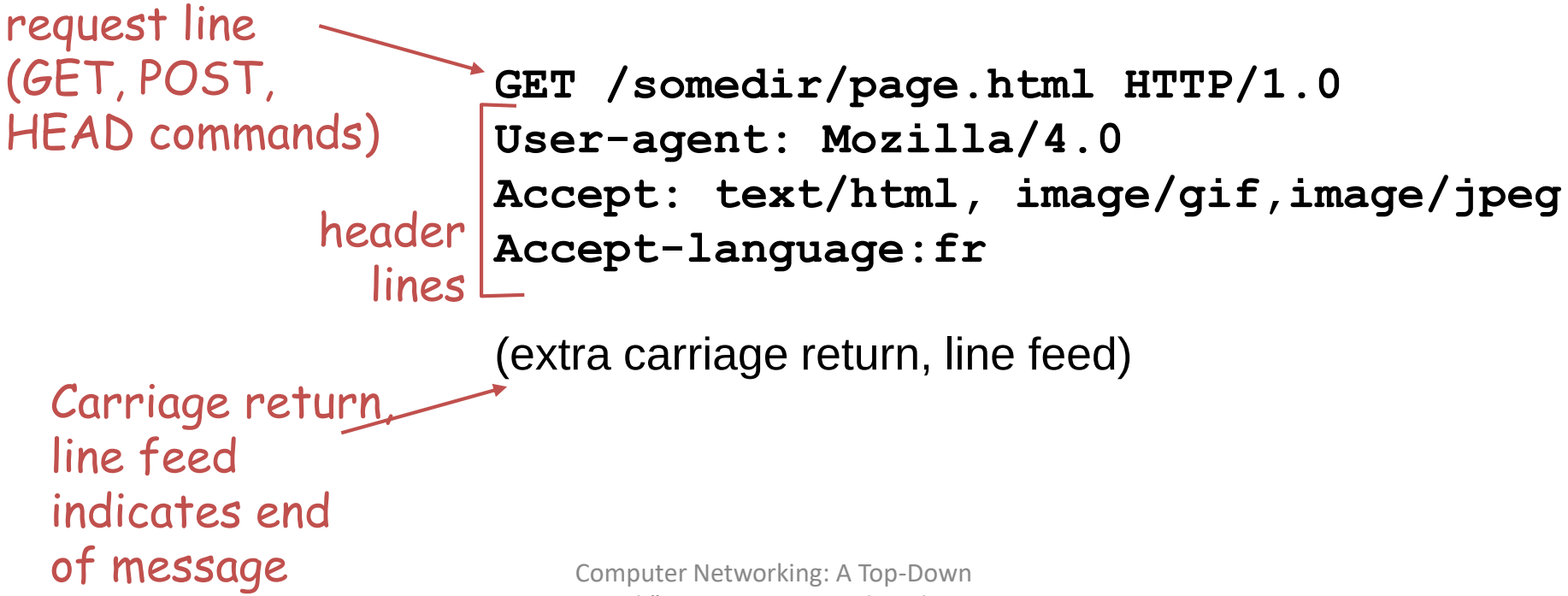
- HTTP/1.0
- server parses request, responds, and closes TCP connection
- 2 RTTs to fetch each object
- Each object transfer suffers from slow start

Persistent

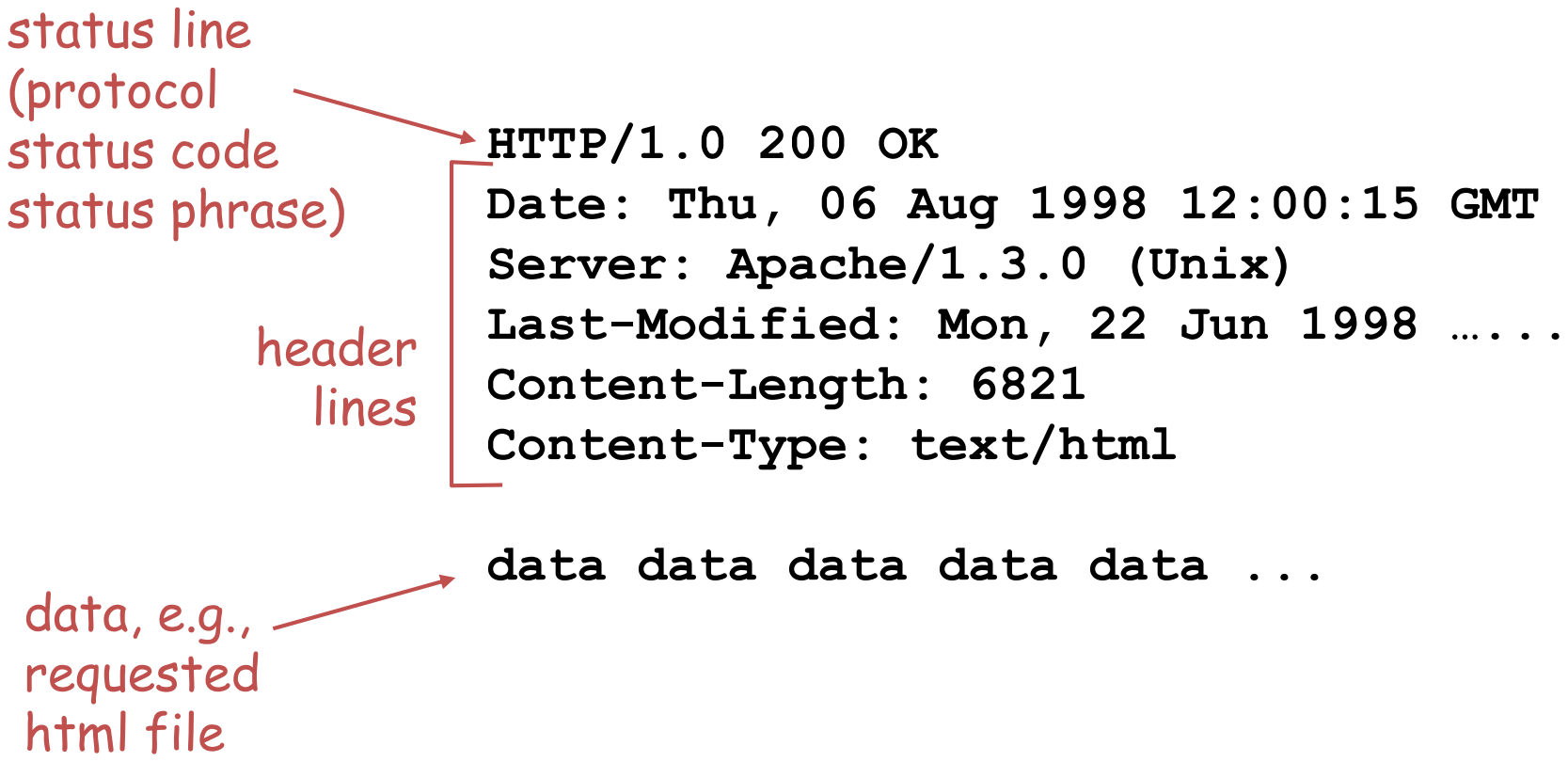
- default for HTTP/1.1
- on same TCP connection: server, parses request, responds, parses new request,..
- Client sends requests for all referenced objects as soon as it receives base HTML.
- Fewer RTTs and less slow start.

http message format: request*

- two types of http messages: *request, response*
- http request message:
  - ASCII (human-readable format)



http message format: respone*



http response status codes

In first line in server->client response message.

A few 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

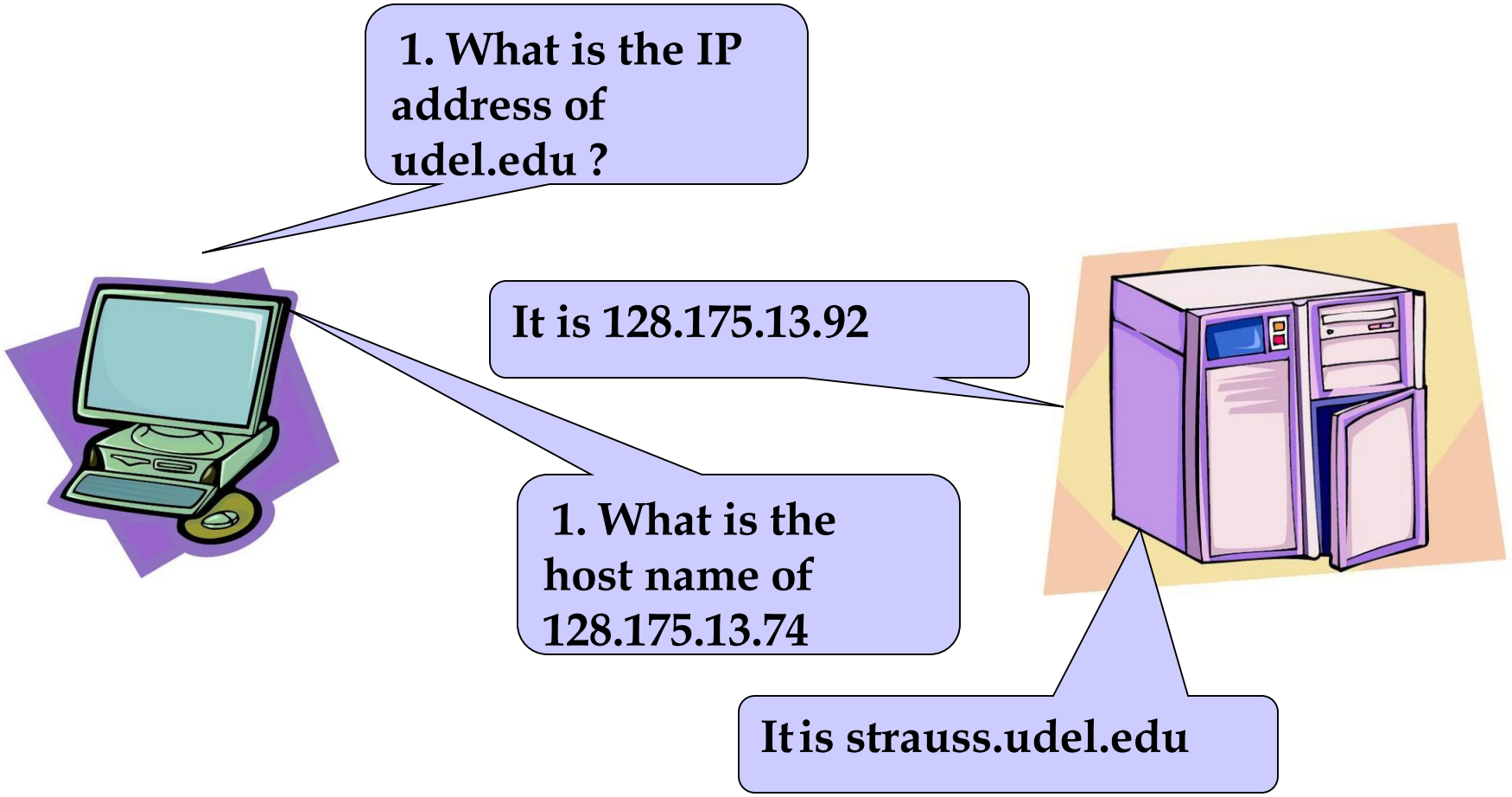
404 Not Found

- requested document not found on this server

505 HTTP Version Not Supported

Application Layer protocol

DNS protocol



DNS Components*

DNS is an application layer protocol that translates IP addresses to names and vice versa.

There are 3 components:

- Name Space:

  Specifications for a structured name space and data associated with the names

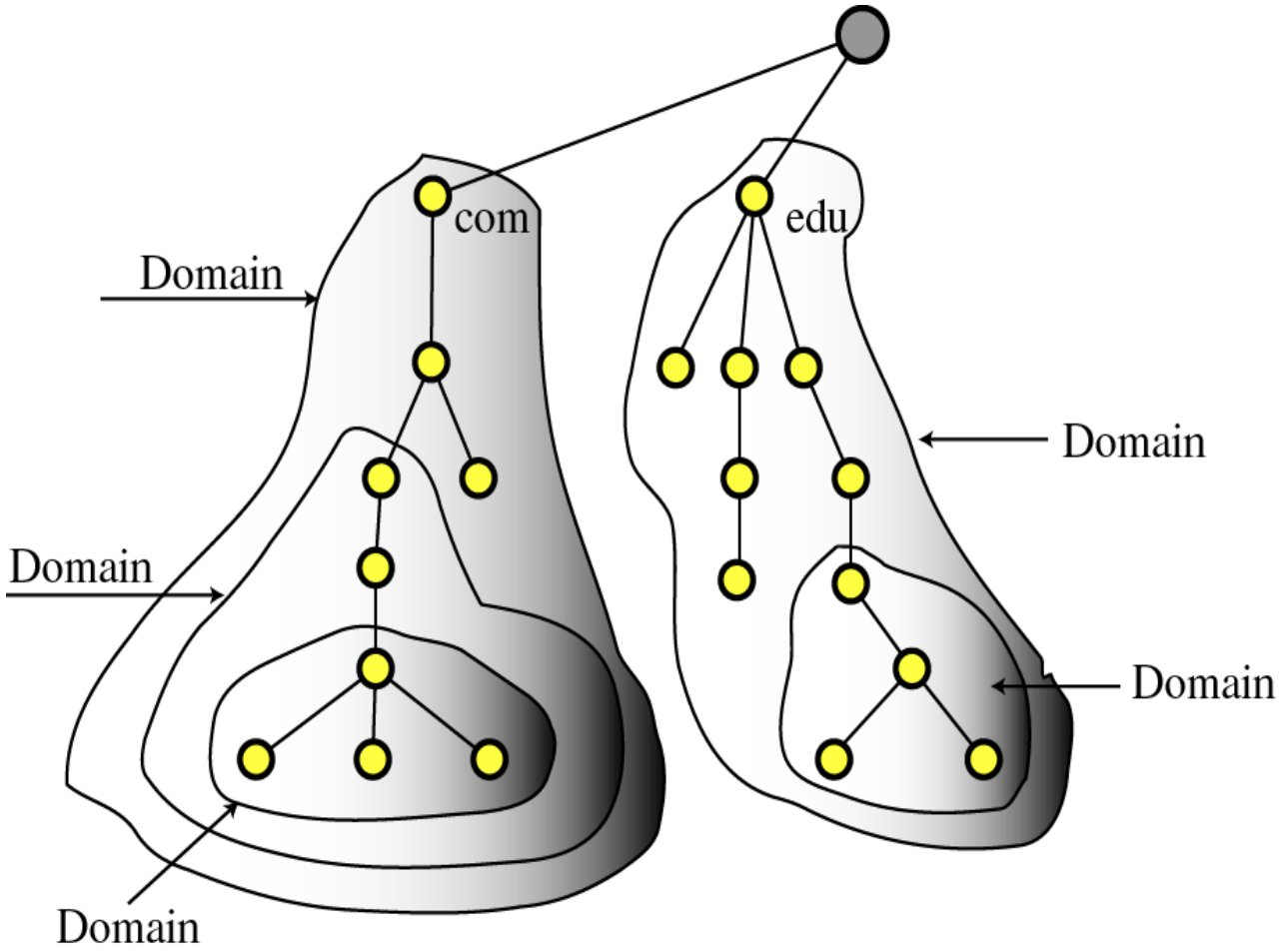
- Resolvers:

  Client programs that extract information from Name Servers.

- Name Servers:

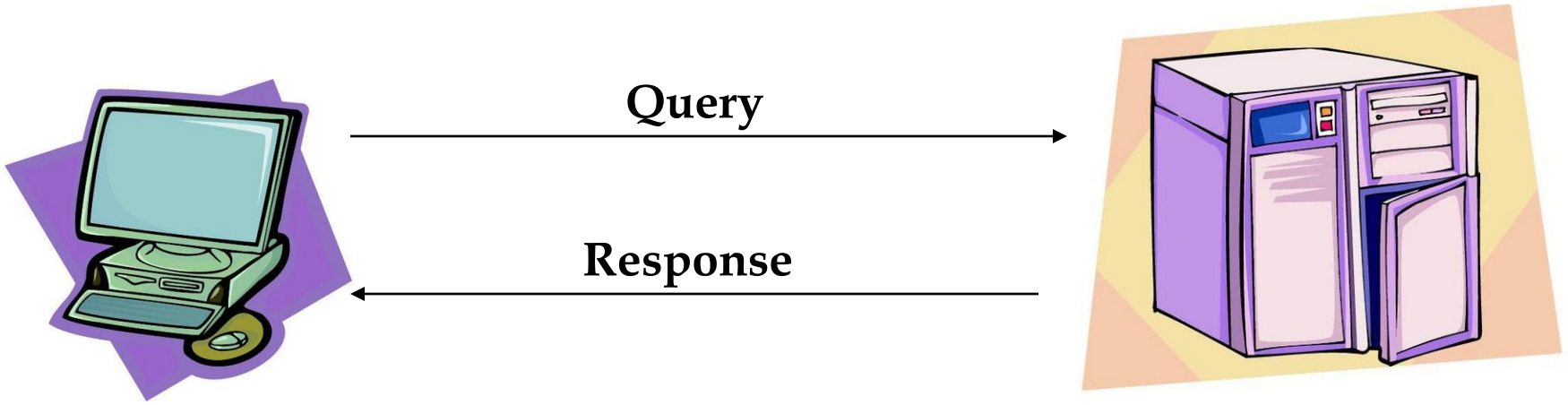
  Server programs which hold information about the structure and the names.

Name Space*

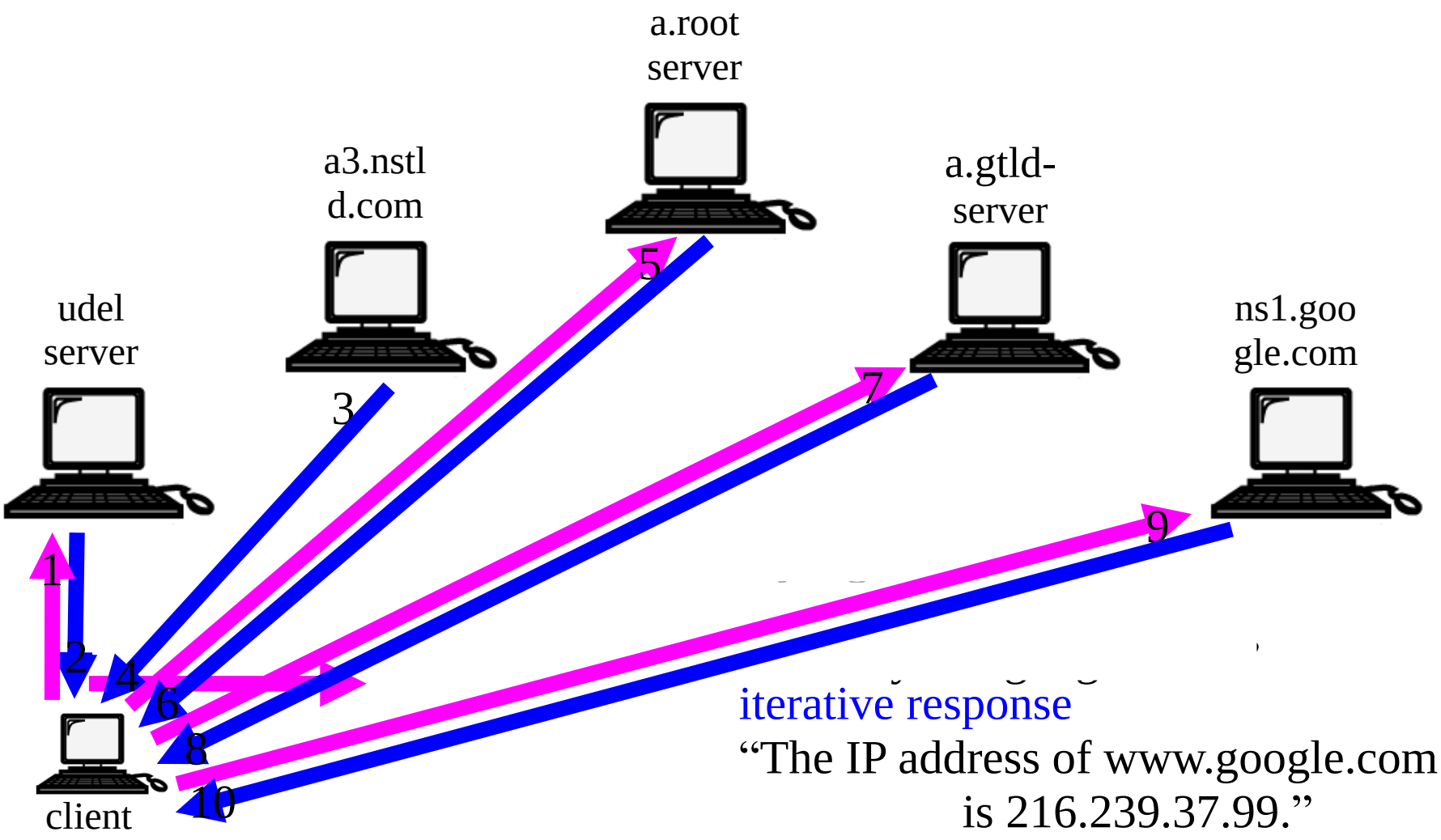


Resolvers*

A Resolver maps a name to an address and vice versa.



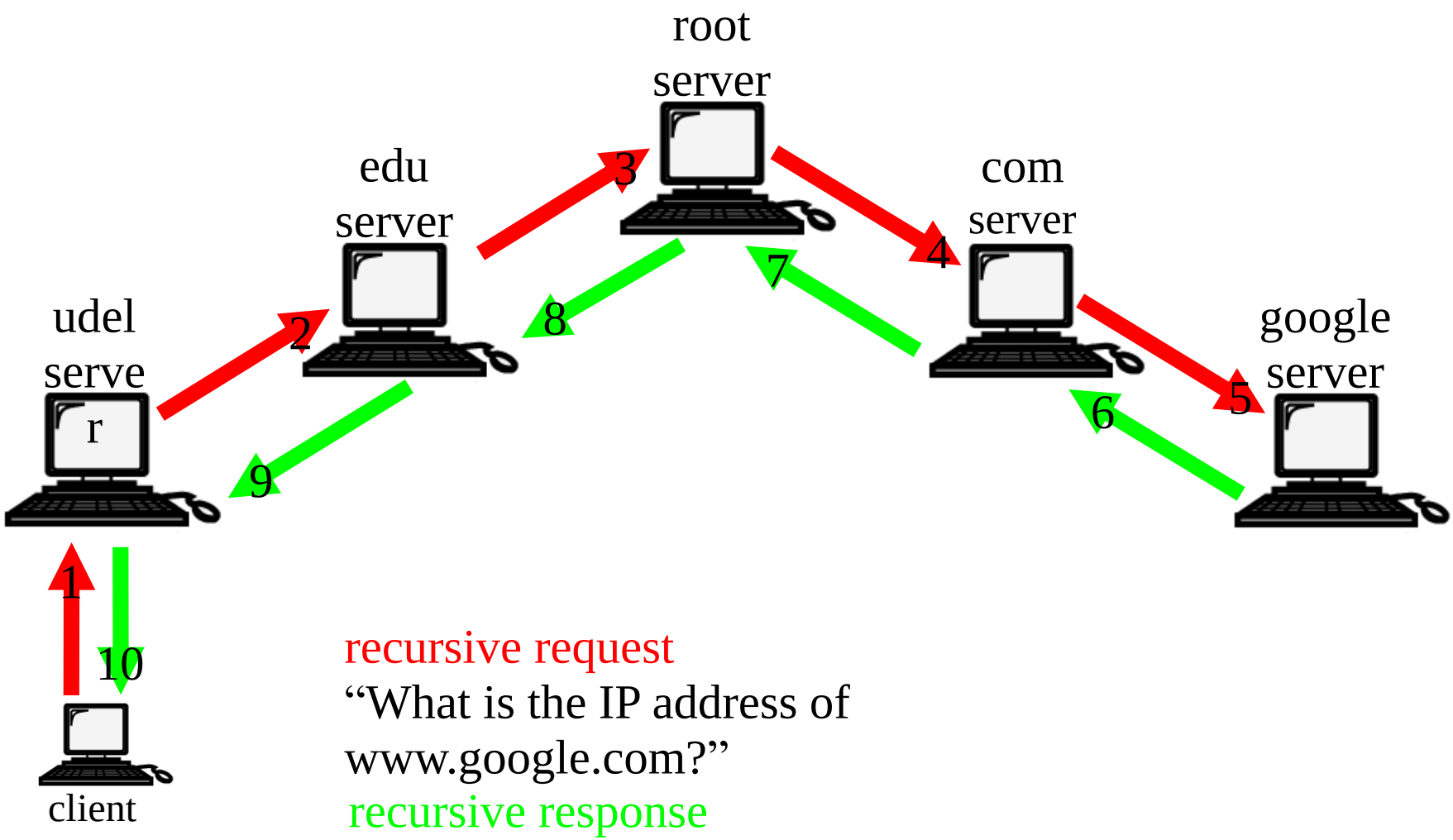
Iterative Resolution*



iterative response
"The IP address of www.google.com is 216.239.37.99."

iterative request
"What is the IP address of www.google.com?"

Recursive Resolution*

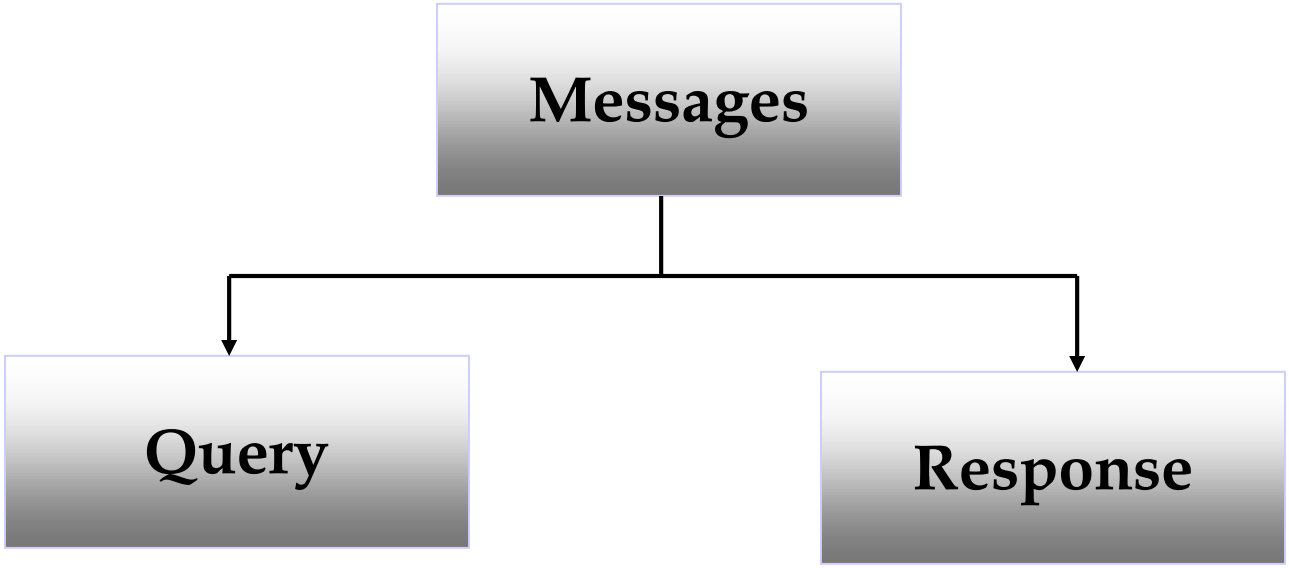


recursive request
"What is the IP address of www.google.com?"
recursive response
"The IP address of www.google.com is 216.239.37.99."

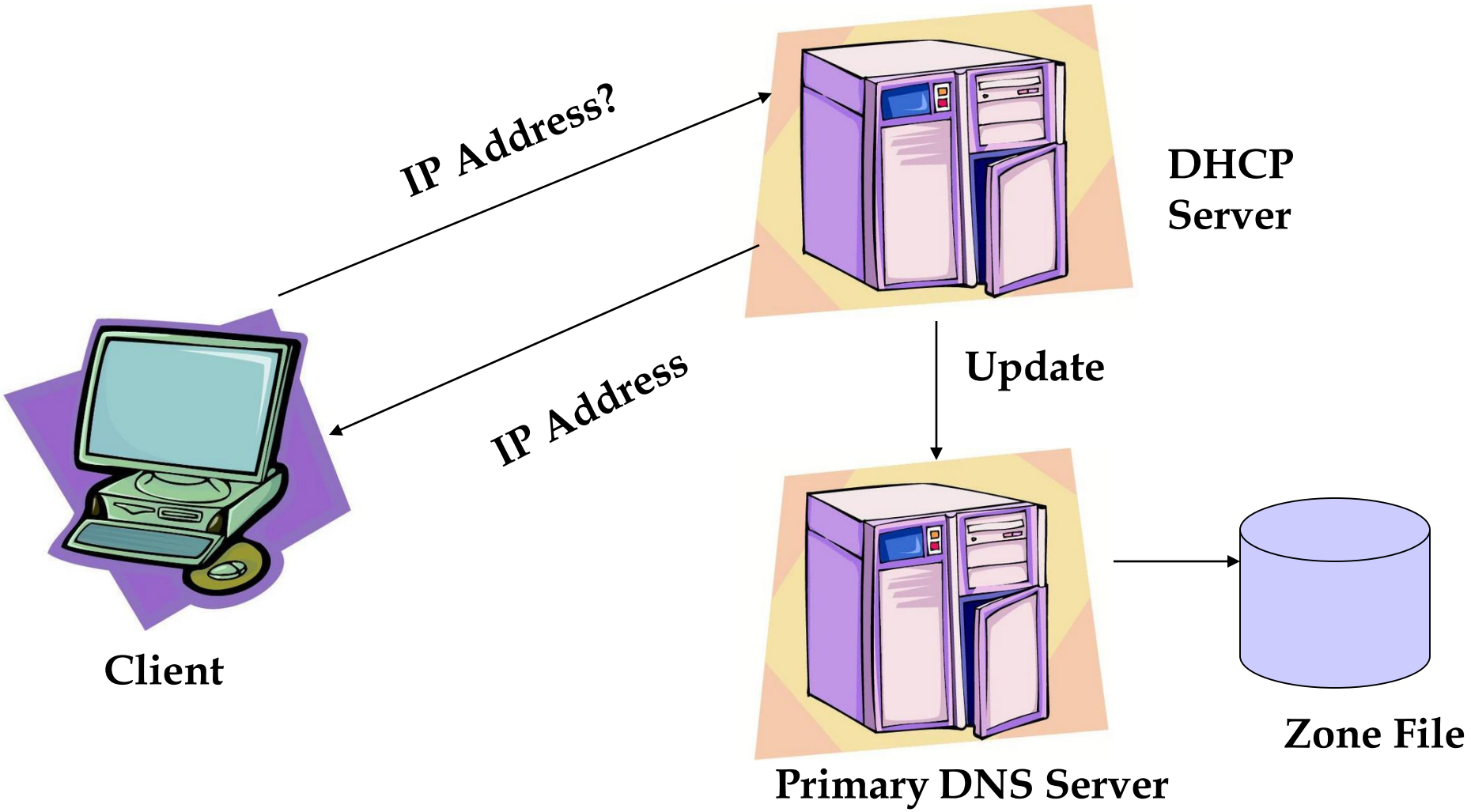
Name Server*

A **name server** is a computer hardware or software **server** that implements a network service for providing responses to queries against a directory service.

DNS Messages



Dynamic DNS*


IP Address?
IP Address
DHCP Server
Update
Client
Primary DNS Server
Zone File

Reference

- Computer Networking: A Top-Down Approach", James Kurose and Keith Ross , 5th edition