Chapter 3

Protocols and Client/Server Applications

By Pavan Poudel

Overview

3.1 Standard Protocols: SMTP, E-Mail Message (RFC22), PGP, IMAP, HTTP, FTP

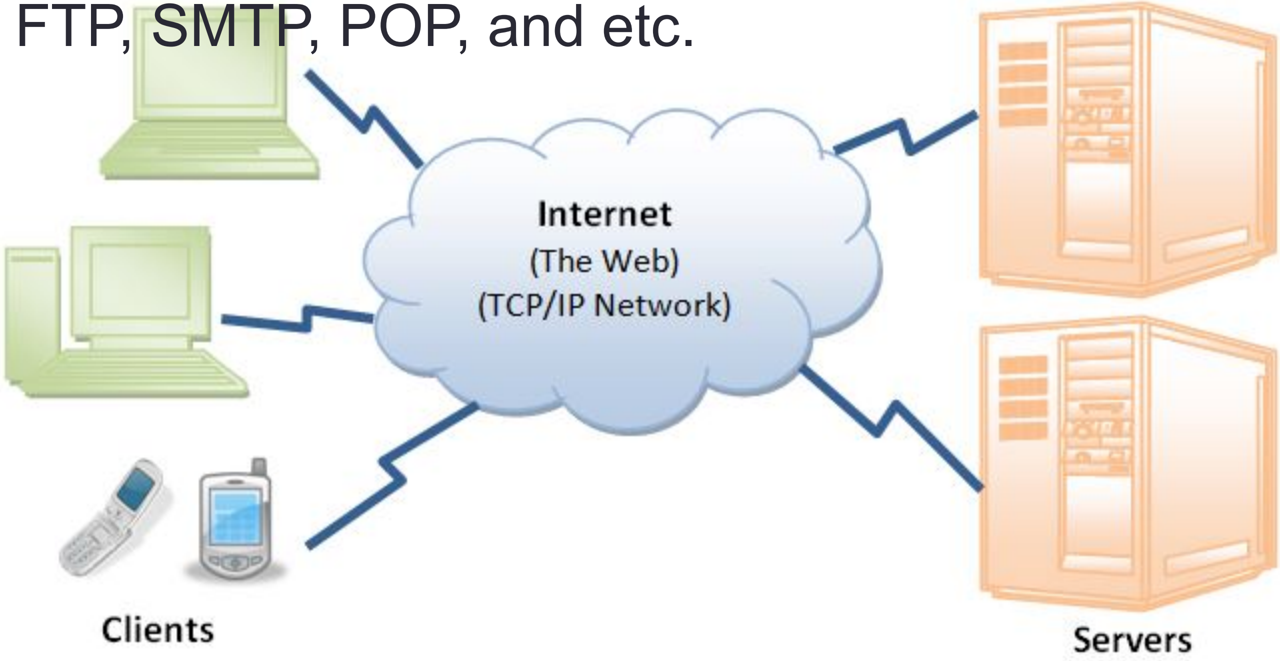
3.2 N-Tier Client Server Architecture

3.3 Universal Internet Browsing

3.4 Multiprotocol Support

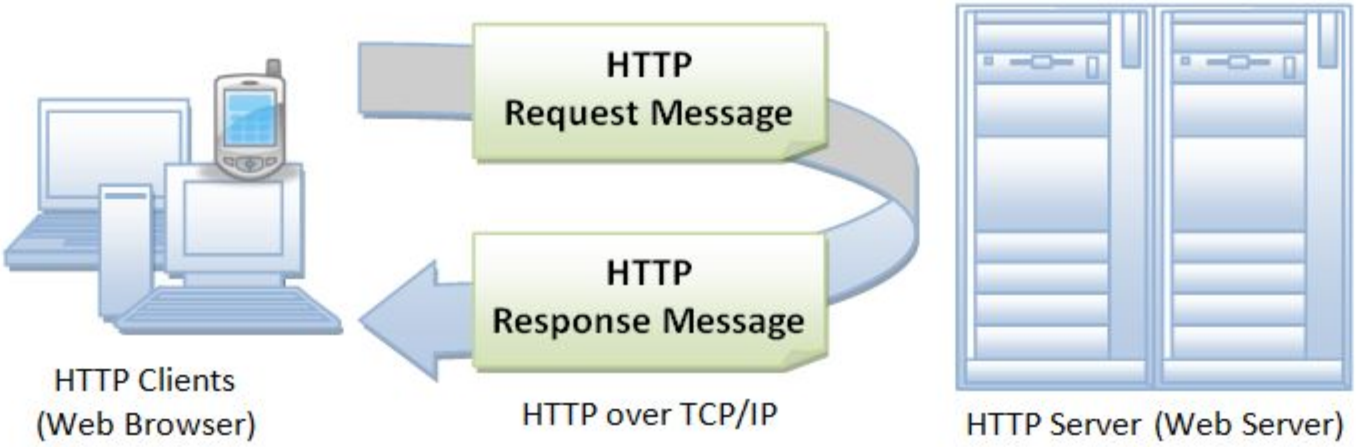
Hyper text Transfer Protocol (HTTP)

- Internet (or The Web) is a distributed client/server information system.
- Many applications are running concurrently over the Web, such as web browsing/surfing, e-mail, file transfer, audio & video streaming, and so on. In order for proper communication to take place between the client and the server, these applications must agree on a specific application-level protocol such as HTTP, FTP, SMTP, POP, and etc.



Hyper text Transfer Protocol (HTTP)

- HTTP (Hypertext Transfer Protocol) is perhaps the most popular application protocol used in the Internet (or The WEB).
- HTTP is an *asymmetric request-response client-server* protocol. An HTTP client sends a request message to an HTTP server. The server, in turn, returns a response message. In other words, HTTP is a *pull protocol*, the client *pulls* information from the server (instead of server *pushes* information down to the client).



Hyper text Transfer Protocol (HTTP)

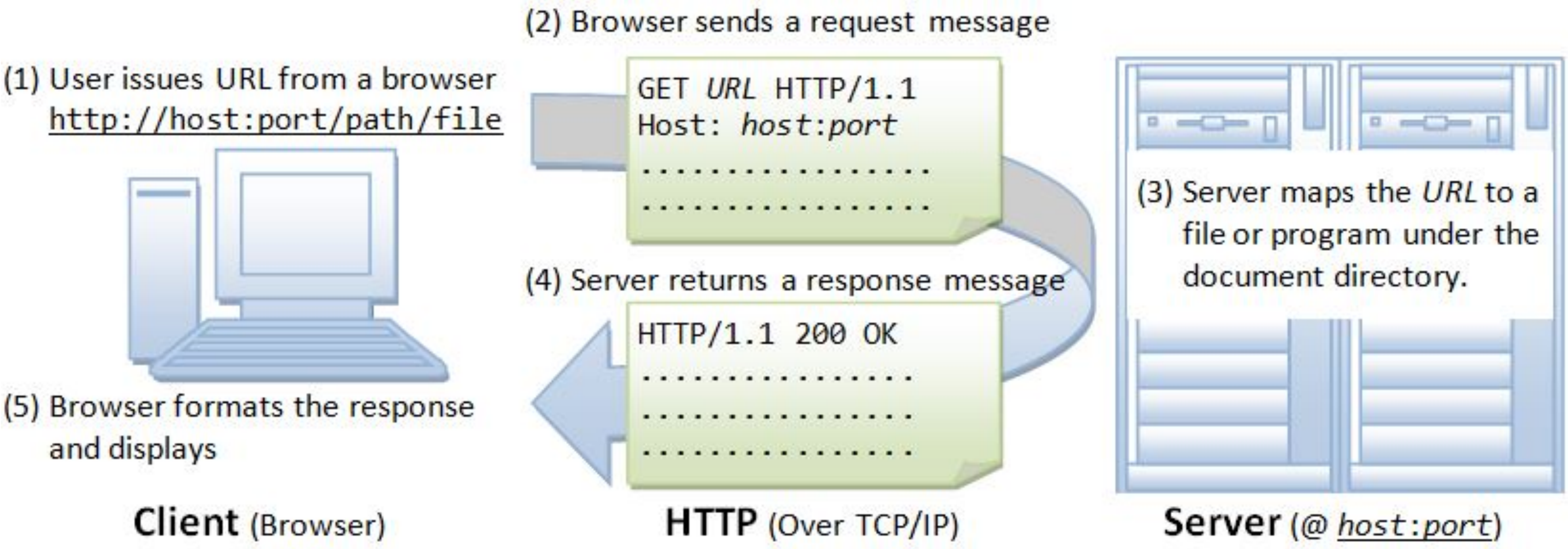
- Web server listen at port number 80.
- HTTP is implemented in two programs: a client program and server program.
- The client program and server programs, executing on different end systems, talk to each other by exchanging HTTP messages.
- HTTP defines the structure of these messages and how the client and server exchange the messages.
- HTTP is a stateless protocol. In other words, the current request does not know what has been done in the previous requests.
- HTTP permits negotiating of data type and representation, so as to allow systems to be built independently of the data being transferred.

HTTP

- A **Web page** (also called a document) consists of objects. An **object** is a simply file -- such as a HTML file, a JPEG image, a GIF image, a Java applet, an audio clip, etc. -- that is addressable by a single URL. Most Web pages consist of a **base HTML file** and several referenced objects.
- For example, if a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images. The base HTML file references the other objects in the page with the objects' URLs.
- Each URL has two components: the host name of the server that houses the object and the object's path name.
- For example, the URL www.someSchool.edu/someDepartment/picture.gif has www.someSchool.edu for a **host name** and /someDepartment/picture.gif for a **path name.**

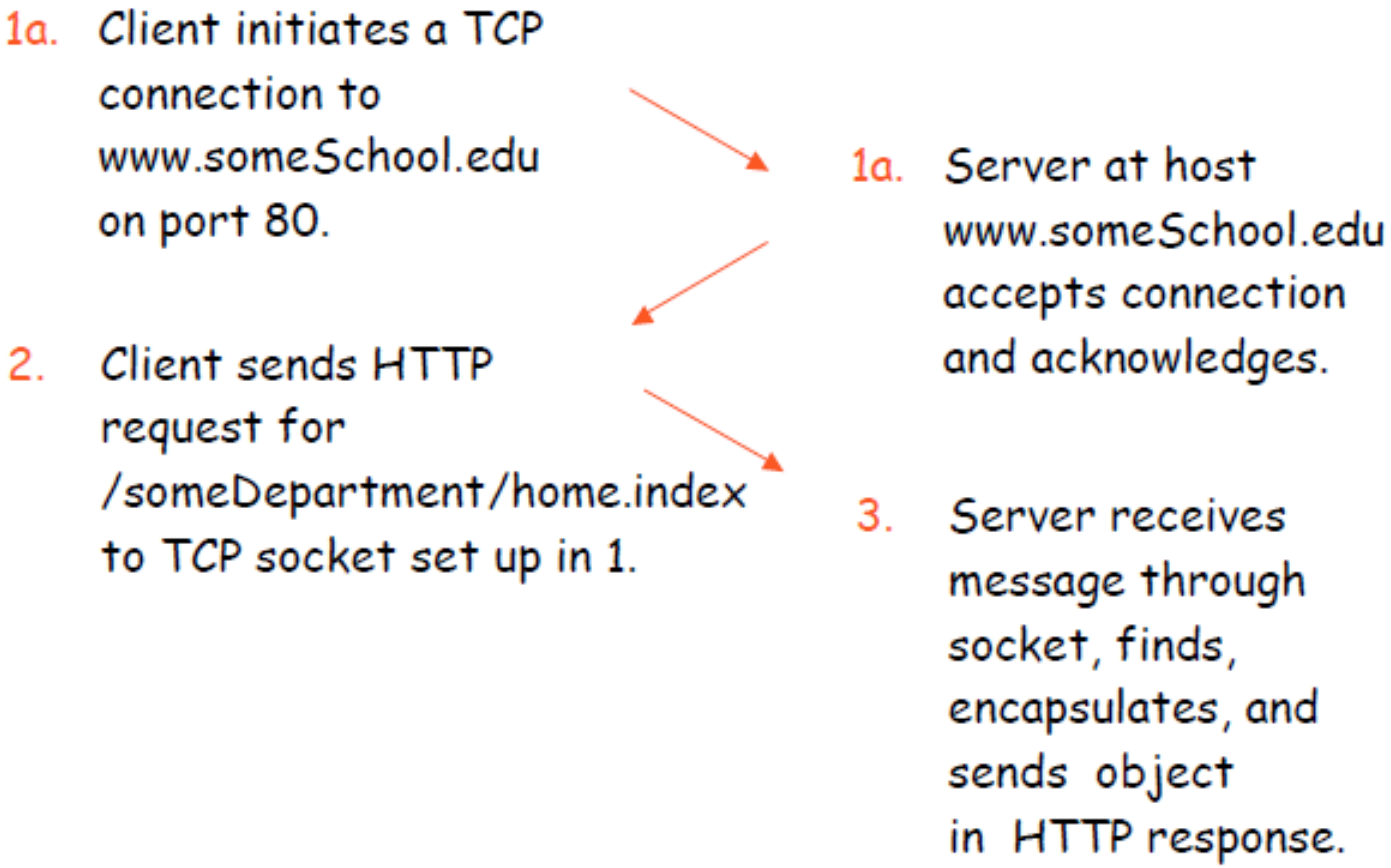
HTTP

- Whenever you issue a URL from your browser to get a web resource using HTTP, e.g. http://www.test101.com/index.html, the browser turns the URL into a *request message* and sends it to the HTTP server. The HTTP server interprets the request message, and returns you an appropriate response message, which is either the resource you requested or an error message. This process is illustrated below:


(2) Browser sends a request message
(1) User issues URL from a browser
http://host:port/path/file
GET URL HTTP/1.1
Host: host:port
....
....
(3) Server maps the URL to a file or program under the document directory.
(4) Server returns a response message
HTTP/1.1 200 OK
....
....
....
(5) Browser formats the response and displays
Client (Browser)
HTTP (Over TCP/IP)
Server (@ host:port)

HTTP/1.0 non-persistent connection

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



1a. Client initiates a TCP connection to www.someSchool.edu on port 80.

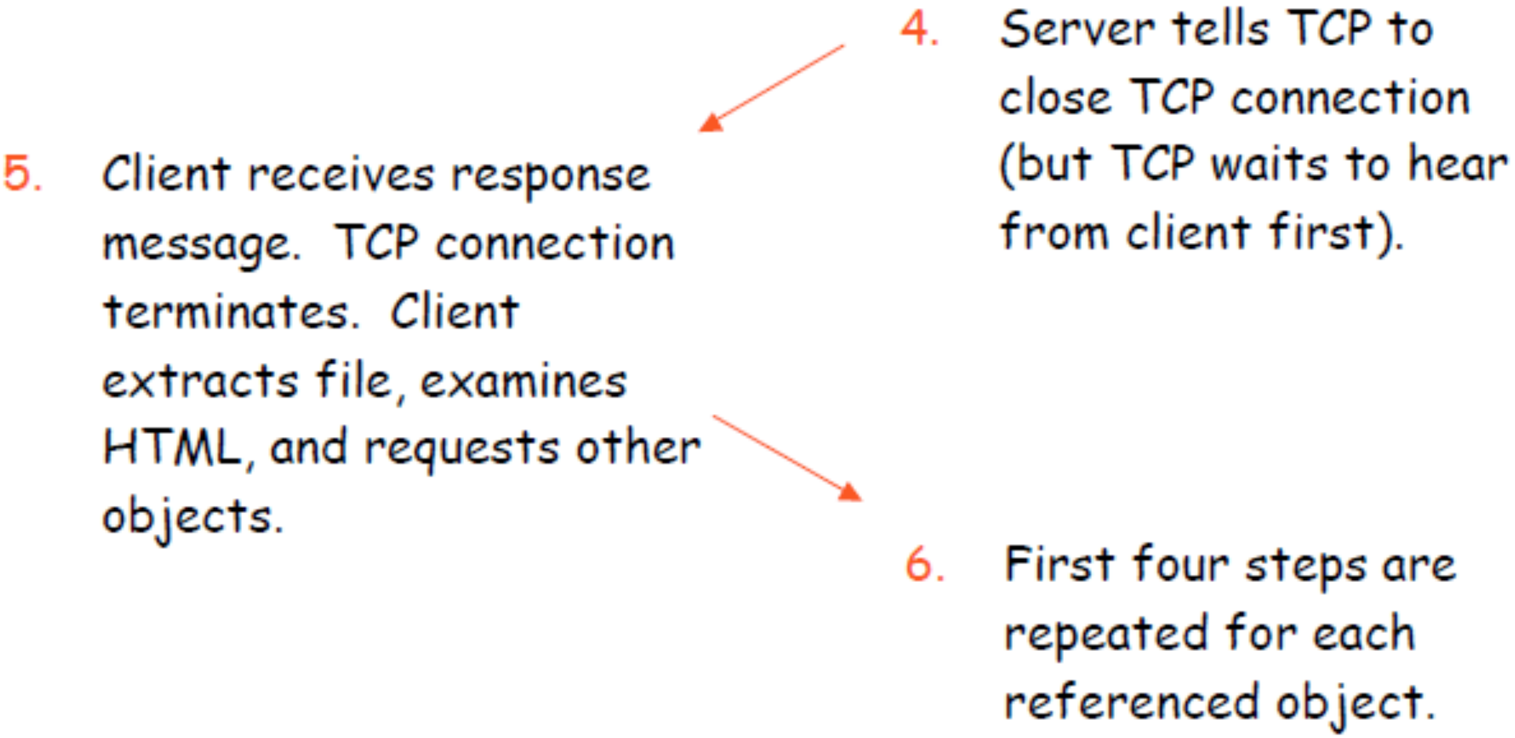
1a. Server at host www.someSchool.edu accepts connection and acknowledges.

2. Client sends HTTP request for /someDepartment/home.index to TCP socket set up in 1.

3. Server receives message through socket, finds, encapsulates, and sends object in HTTP response.

HTTP/1.0 non-persistent connection

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

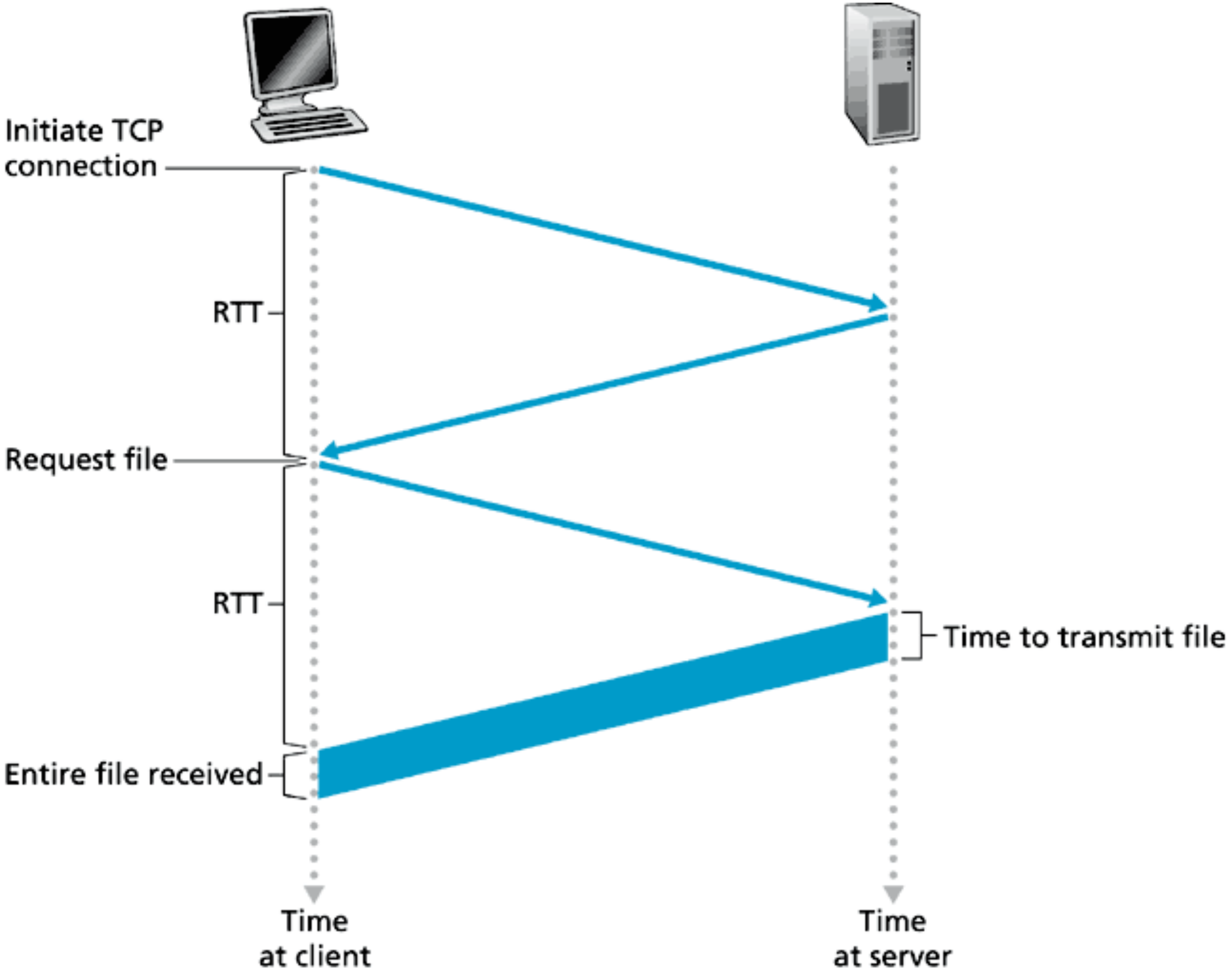


4. Server tells TCP to close TCP connection (but TCP waits to hear from client first).

5. Client receives response message. TCP connection terminates. Client extracts file, examines HTML, and requests other objects.

6. First four steps are repeated for each referenced object.

Timing of a non-persistent connection

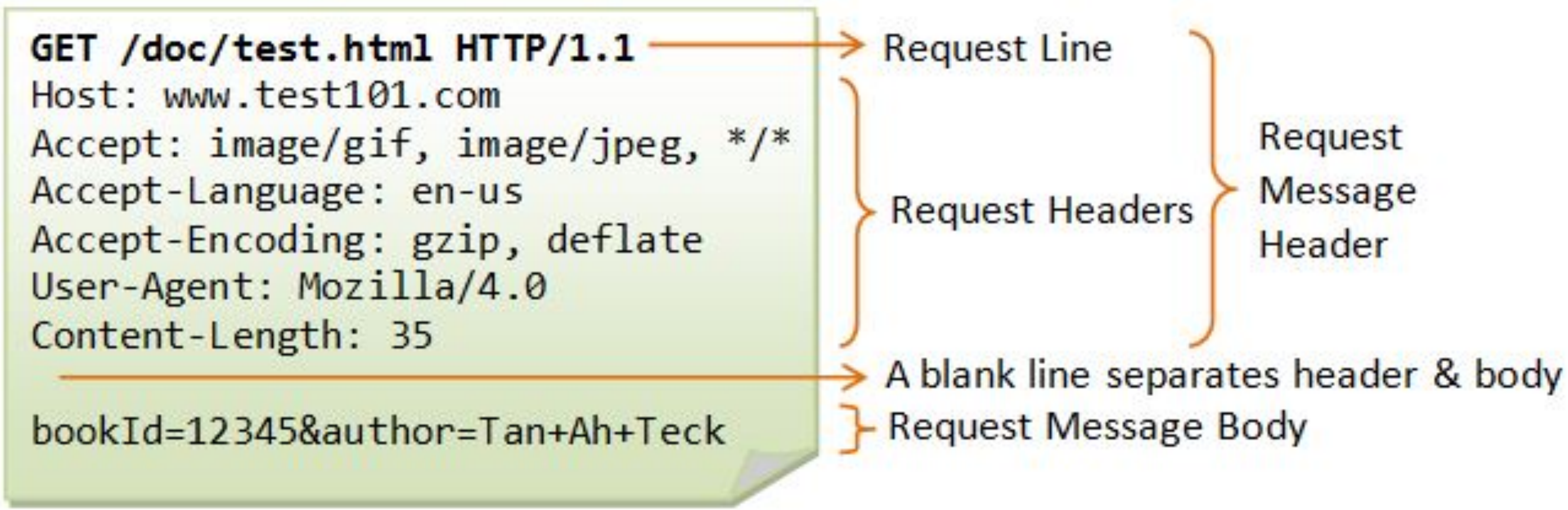


HTTP/1.1 persistent connection

- HTTP/1.0 does not address the issues of proxies, caching, persistent connection, virtual hosts, and range download. These features were provided in HTTP/1.1.
- In an HTTP/1.1 persistent connection, the server leaves the TCP connection open after sending a response.
- Subsequent requests and responses (pipelined or not) between the same client can be sent on the same connection.

HTTP Request Message

- HTTP request message consists of:
  - Request Line
    - *request-method-name request-URI HTTP-version*
  - Request Headers
    - *request-header-name*: *request-header-value1*, *request-header-value2* ...
  - Request Message Body



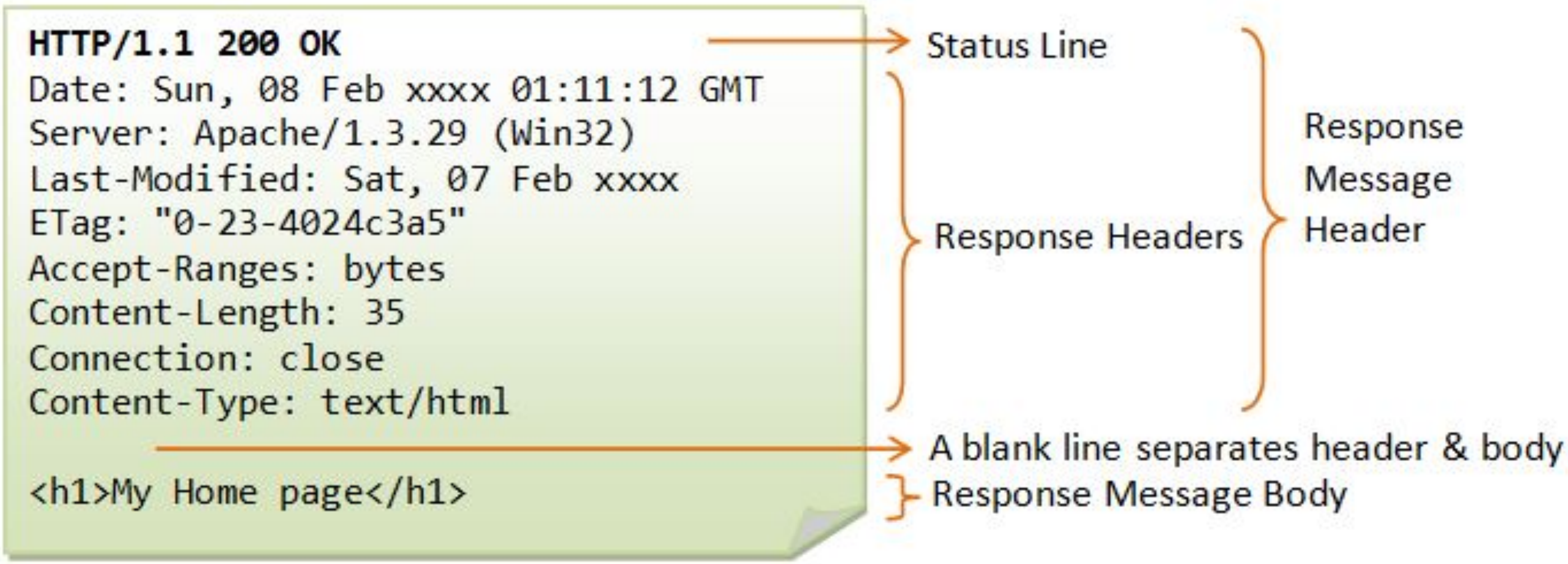
```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

HTTP Request Methods

- HTTP protocol defines a set of request methods. A client can use one of these request methods to send a request message to an HTTP server.
  - **GET**: A client can use the GET request to get a web resource from the server.
  - **HEAD**: A client can use the HEAD request to get the header that a GET request would have obtained. Since the header contains the last-modified date of the data, this can be used to check against the local cache copy.
  - **POST**: Used to post data up to the web server.
  - **PUT**: Ask the server to store the data.
  - **DELETE**: Ask the server to delete the data.
  - **TRACE**: Ask the server to return a diagnostic trace of the actions it takes.
  - **OPTIONS**: Ask the server to return the list of request methods it supports.
  - **CONNECT**: Used to tell a proxy to make a connection to another host and simply reply the content, without attempting to parse or cache it. This is often used to make SSL connection through the proxy.

HTTP Response Message

- HTTP Response message consists of:
  - Status Line
    - *HTTP-version status-code reason-phrase*
  - Response Header
    - *response-header-name*: *response-header-value1*, *response-header-value2*, ...
  - Response Message Body



```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Some HTTP response status codes

- 200 OK
  - request succeeded, requested object in this message
- 301 Moved Permanently
  - requested object moved, new location specified in this message (Location)
- 304 Not Modified
  - In response to the If-Modified-Since conditional GET request, the server notifies that the resource requested has not been modified.
- 400 Bad Request
  - request message not understood by server
- 404 Not Found
  - requested document not found on this server
- 408 Request Timeout
- 505 HTTP Version Not Supported.

The Conditional GET

- You may use additional request header to issue a "conditional request". For example, to ask for the document based on the last-modified date (so as to decide whether to use the local cache copy), or to ask for a portion of the document (or range) instead of the entire document (useful for downloading large documents).
- The conditional request headers include:
  - If-Modified-Since (check for response status code "304 Not Modified").
  - If-Unmodified-Since
  - If-Match
  - If-None-Match
  - If-Range

Conditional GET

- Although Web caching can reduce user-perceived response times, it introduces a new problem -- a copy of an object residing in the cache may be stale. In other words, the object housed in the Web server may have been modified since the copy was cached at the client.
- Fortunately, HTTP has a mechanism that allows the client to employ caching while still ensuring that all objects passed to the browser are up-to-date. This mechanism is called the conditional GET.
- A HTTP request message is a so-called conditional GET message if (i) the request message uses the GET method and (ii) the request message includes an If-Modified-Since:header line.

Conditional GET

- Request

```
GET /fruit/kiwi.gif HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
If-modified-since: Mon, 22 Jun 1998 09:23:24
```

- If not modified, Response

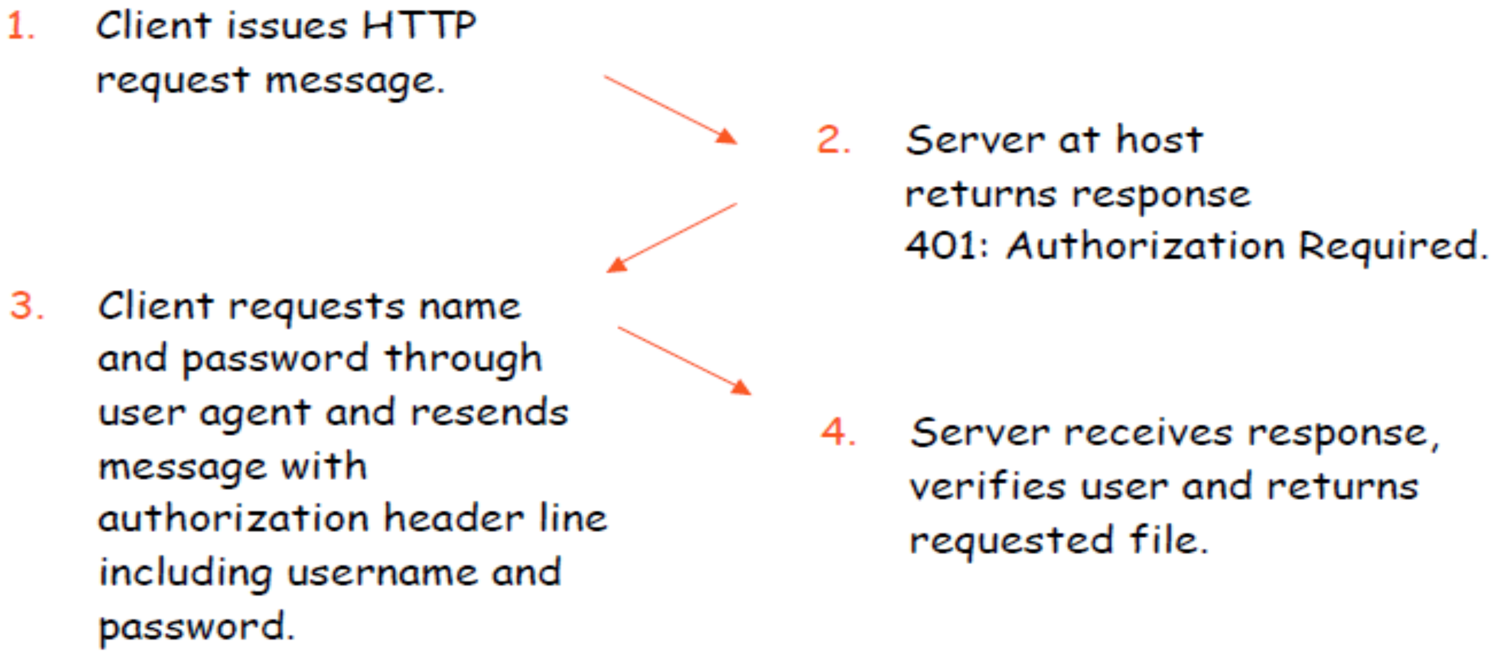
```
HTTP/1.0 304 Not Modified
Date: Wed, 19 Aug 1998 15:39:29
Server: Apache/1.3.0 (Unix)
(empty entity body)
```

User-Server Interaction

- HTTP server is stateless.
- This simplifies server design, and has permitted engineers to develop very high performing Web servers.
- However, it is often desirable for a Web site to identify users, either because the server wishes to restrict user access or because it wants to serve content as a function of the user identity.
- HTTP provides two mechanisms to help a server identify a user: **authentication** and **cookies**.

Authentication

- Many sites require users to provide a username and a password in order to access the documents housed on the server. This requirement is referred to as authentication.
- HTTP provides special status codes and headers to help sites perform authentication.
- Suppose a client requests an object from a server, and the server requires user authorization.

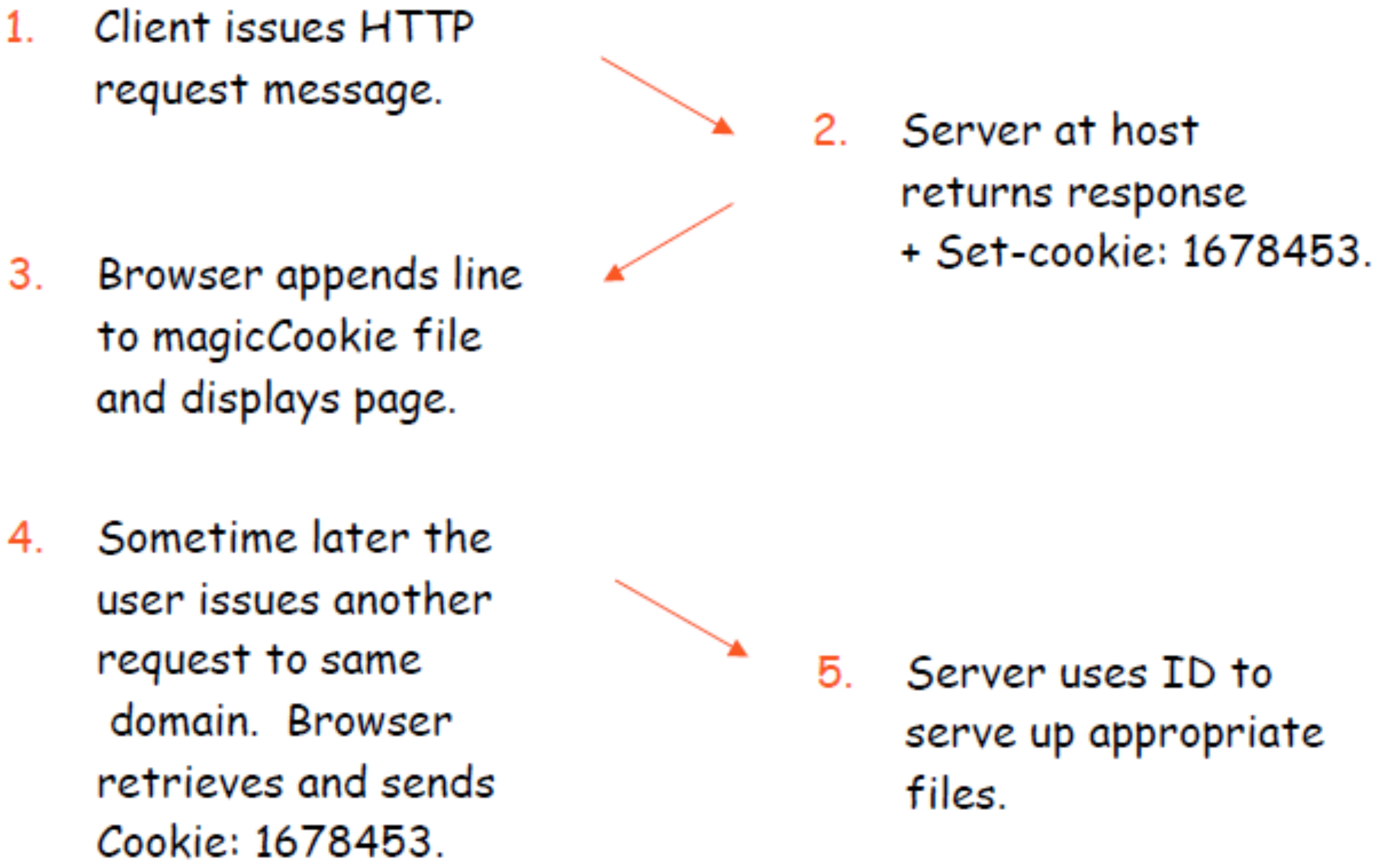


1. Client issues HTTP request message.
2. Server at host returns response 401: Authorization Required.
3. Client requests name and password through user agent and resends message with authorization header line including username and password.
4. Server receives response, verifies user and returns requested file.

Cookies

- A cookie, also known as an HTTP cookie, web cookie, or browser cookie is a small piece of data sent from a website and stored in a user's web browser while the user is browsing that website.
- Every time the user loads the website, the browser sends the cookie back to the server to notify the website of the user's previous activity.
- Cookies were designed to be a reliable mechanism for websites to remember stateful information (such as items in a shopping cart) or to record the user's browsing activity (including clicking particular buttons, logging in, or recording which pages were visited by the user as far back as months or years ago).

Cookies



1. Client issues HTTP request message.

2. Server at host returns response + Set-cookie: 1678453.

3. Browser appends line to magicCookie file and displays page.

4. Sometime later the user issues another request to same domain. Browser retrieves and sends Cookie: 1678453.

5. Server uses ID to serve up appropriate files.

Structure of Cookies

- Browsers are expected to support, at least, cookies with a size of 4KB. It consists of seven components:
  - Name of the cookie
  - Value of the cookie
  - Expiry of the cookie
  - Path the cookie is good for
  - Domain the cookie is good for
  - Need for a secure connection to use the cookie
  - Whether or not the cookie can be accessed through other means than HTTP (i.e., JavaScript)

- The first two components (name and value) are required to be explicitly set.

Types of Cookie

- Session Cookie
- Persistent cookie
- Secure cookie
- HttpOnly cookie
- Third-party cookie
- Supercookie
- Zombie cookie

Types of Cookie

- Session cookie
  - A session cookie, also known as an in-memory cookie or transient cookie, exists only in temporary memory while the user navigates the website. When an expiry date or validity interval is not set at cookie creation time, a session cookie is created. Web browsers normally delete session cookies when the user closes the browser.
- Persistent cookie
  - A persistent cookie outlast user sessions. If a persistent cookie has its Max-Age set to one year (for example), then, during that year, the initial value set in that cookie would be sent back to the server every time the user visited the server. This could be used to record a vital piece of information such as how the user initially came to this website. For this reason, persistent cookies are also called tracking cookies.

Types of Cookie

- Secure cookie
  - A secure cookie has the secure attribute enabled and is only used via HTTPS, ensuring that the cookie is always encrypted when transmitting from client to server. This makes the cookie less likely to be exposed to cookie theft via eavesdropping. In addition to that, all cookies are subject to browser's same-origin policy.

- HttpOnly cookie
  - The HttpOnly attribute is supported by most modern browsers. On a supported browser, an HttpOnly session cookie will be used only when transmitting HTTP (or HTTPS) requests, thus restricting access from other, non-HTTP APIs such as JavaScript. This restriction mitigates but does not eliminate the threat of session cookie theft via cross-site scripting (XSS). This feature applies only to session-management cookies, and not other browser cookies.

Types of Cookie

- Third-party Cookie
  - First-party cookies are cookies that belong to the same domain that is shown in the browser's address bar (or that belong to the sub domain of the domain in the address bar). Third-party cookies are cookies that belong to domains different from the one shown in the address bar. Web pages can feature content from third-party domains (such as banner adverts), which opens up the potential for tracking the user's browsing history. Privacy setting options in most modern browsers allow the blocking of third-party tracking cookies.
- Supercookie
  - A "supercookie" is a cookie with an origin of a Top-Level Domain (such as .com) or a Public Suffix (such as .co.uk). It is important that supercookies are blocked by browsers, due to the security holes they introduce. If unblocked, an attacker in control of a malicious website could set a supercookie and potentially disrupt or impersonate legitimate user requests to another website that shares the same Top-Level Domain or Public Suffix as the malicious website. For example, a supercookie with an origin of .com, could maliciously affect a request made to example.com, even if the cookie did not originate from example.com. This can be used to fake logins or change user information.

Types of Cookie

- Zombie cookie
  - Some cookies are automatically recreated after a user has deleted them; these are called zombie cookies. This is accomplished by a script storing the content of the cookie in some other locations, such as the local storage available to Flash content, HTML5 storages and other client-side mechanisms, and then recreating the cookie from backup stores when the cookie's absence is detected.

Cookies

- Most often cookie is used to store a session ID. The session management is done at the server side, instead of client side.
- For example, the following Java servlet uses cookie for managing session ID:

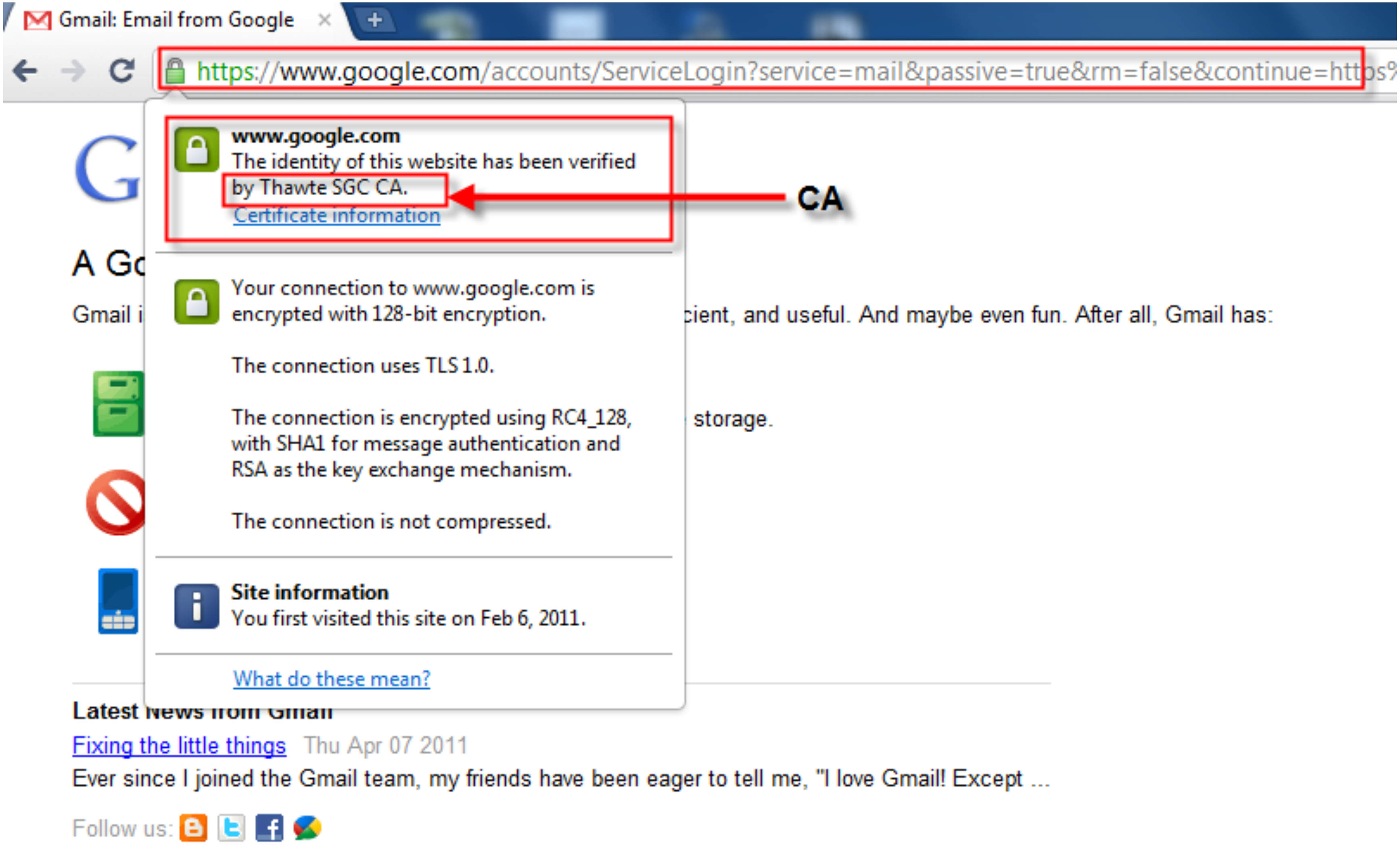
```
HashTable allSessions = new HashTable();
...
String sessionID = getUniqueSessionID();
HashTable sessionData = new HashTable();
allSessions.put(sessionID, sessionData);
Cookie sessionCookie = new Cookie("sessionID", sessionID);
sessionCookie.setPath("/");
response.addCookie(sessionCookie);
```

- The problem on using cookie is some users disable cookie due to the real and perceived privacy concerns over cookies.

HTTPS

- Use TCP at Transport layer.
- Listen at port number 443.
- Hypertext Transfer Protocol Secure (HTTPS) is a widely-used communications protocol for secure communication over a computer network, with especially wide deployment on the Internet.
- Technically, it is not a protocol in itself; rather, it is the result of simply layering the Hypertext Transfer Protocol (HTTP) on top of the SSL/TLS protocol, thus adding the security capabilities of SSL/TLS to standard HTTP communications.
- Need to request the SSL certificate from the Certificate Authority to deploy HTTPS.
- Encryption/Decryption Mechanism is used between client and server for transferring data.
- Like from GoDaddy, VeriSign etc.

HTTPS


Gmail: Email from Google
https://www.google.com/accounts/ServiceLogin?service=mail&passive=true&rm=false&continue=https%
CA

www.google.com
The identity of this website has been verified by Thawte SGC CA.
Certificate information

Your connection to www.google.com is encrypted with 128-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using RC4_128, with SHA1 for message authentication and RSA as the key exchange mechanism.

The connection is not compressed.

Site information
You first visited this site on Feb 6, 2011.

What do these mean?

A Go

Gmail i …cient, and useful. And maybe even fun. After all, Gmail has:

storage.

Latest News from Gmail

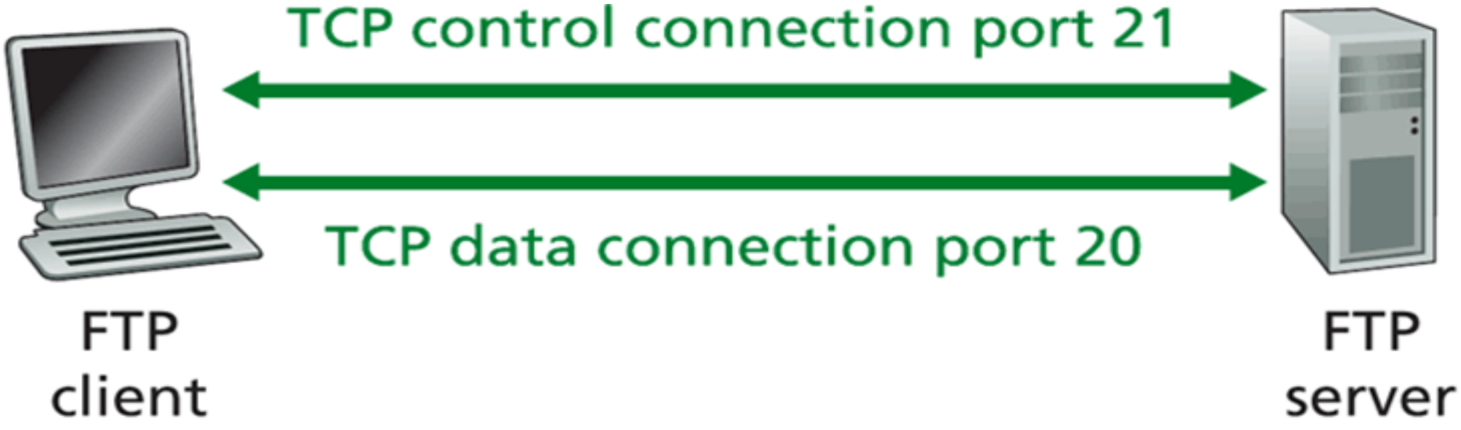
Fixing the little things Thu Apr 07 2011

Ever since I joined the Gmail team, my friends have been eager to tell me, "I love Gmail! Except ...

Follow us:

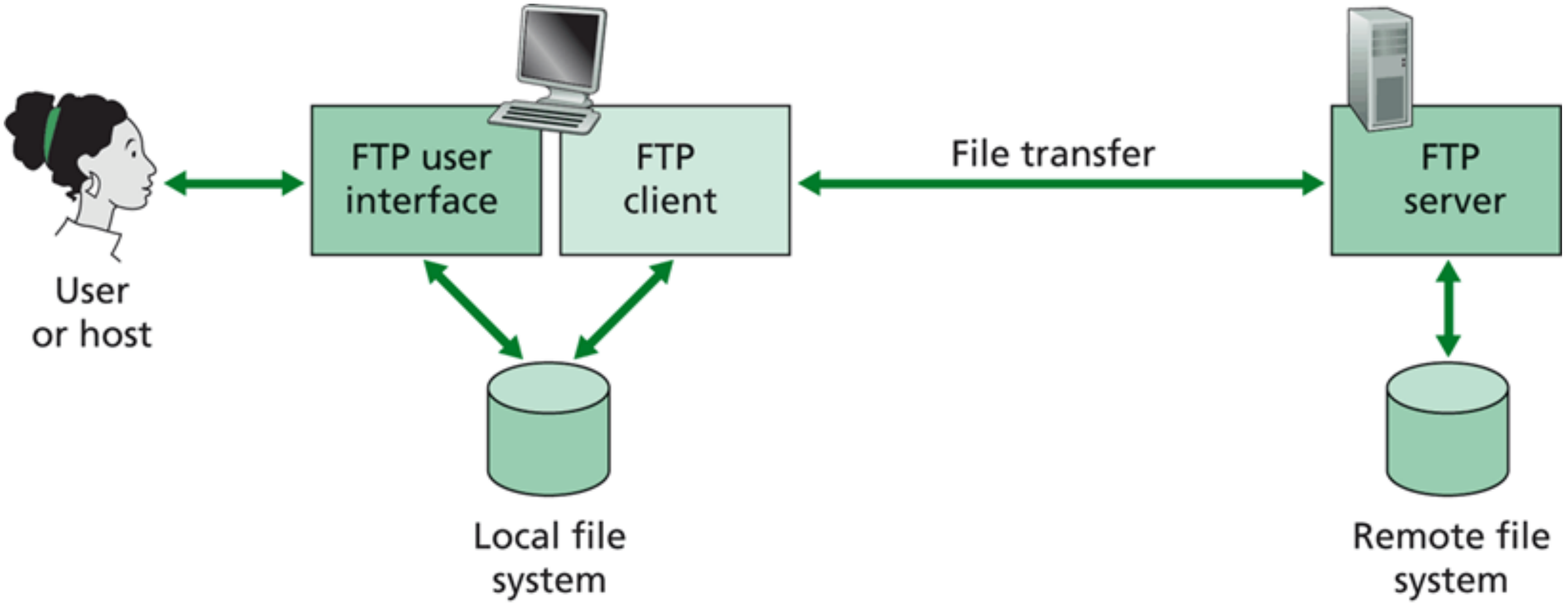
File Transfer Protocol (FTP)

- FTP, like HTTP, runs on top of TCP.
- However, unlike HTTP, FTP uses two parallel TCP connections to transfer a file, a control connection (port #21) and a data connection (port #20). We say FTP sends its control information out-of band.
- Also unlike HTTP, FTP maintains state. In particular, FTP remembers the current directory and earlier authentication.



◆ Control and data connections

File Transfer Protocol (FTP)

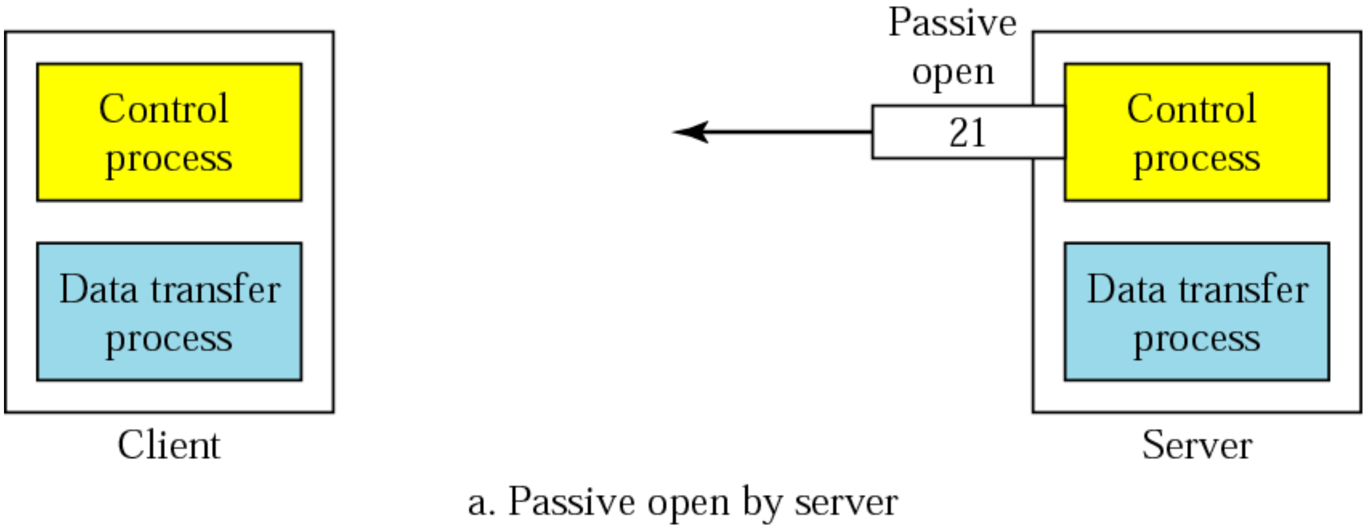


- FTP moves files between local and remote file systems

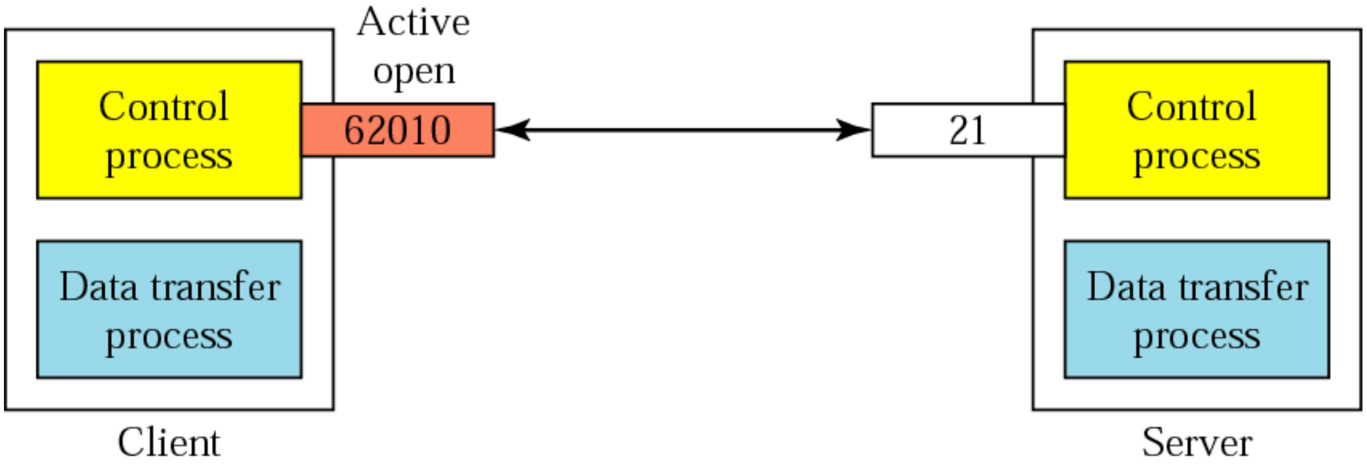
File Transfer Protocol (FTP)

- FTP establishes two connections between the client and server.
- One is for data transfer and the other is for the control information.
- The control connection uses simple rules of communication. Only one line of command or a line of response is transferred at a time.
- But the data connection uses more complex rules due to variety of data types being transferred.
- FTP uses port 21 for control connection and port 20 for the data connection.
- Control connection is maintained during the entire FTP session.
- The data connection is first opened, file is transferred and connection is closed. This is done for transferring each file.

The Control Connection



a. Passive open by server

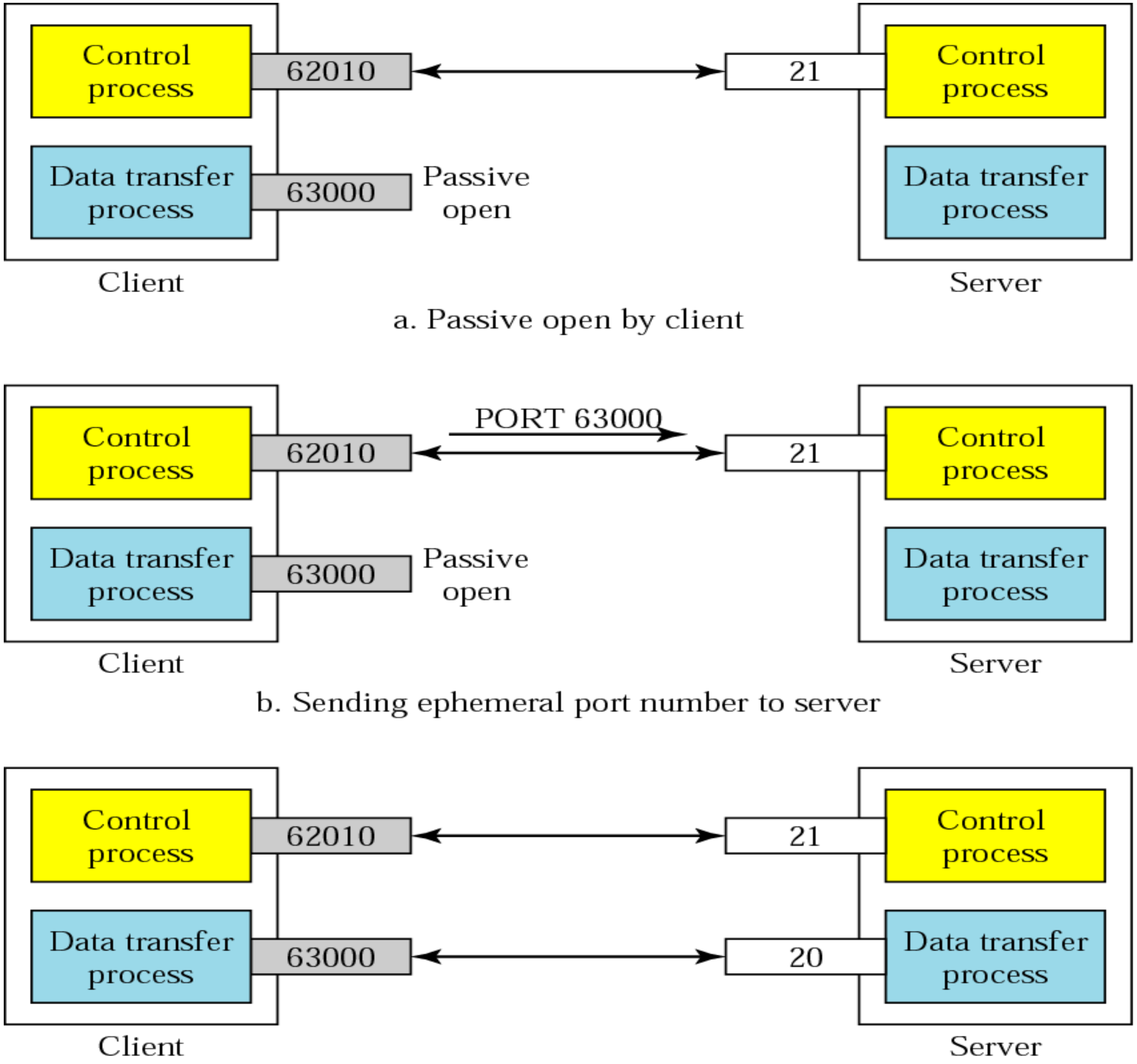


b. Active open by client

The Data Connection

- Uses Server’s well-known port 20.
  - Client issues a passive open on an ephemeral port, say *x*.
  - Client uses PORT command to tell the server about the port number *x*.
  - Server issues an active open from port 20 to port *x*.
  - Server creates a child server/ephemeral port number to serve the client

Creating The Data Connection



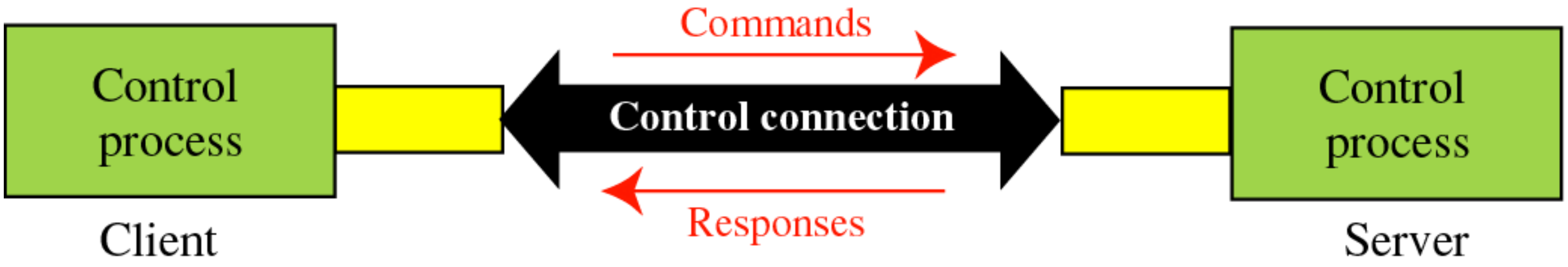
a. Passive open by client

b. Sending ephemeral port number to server

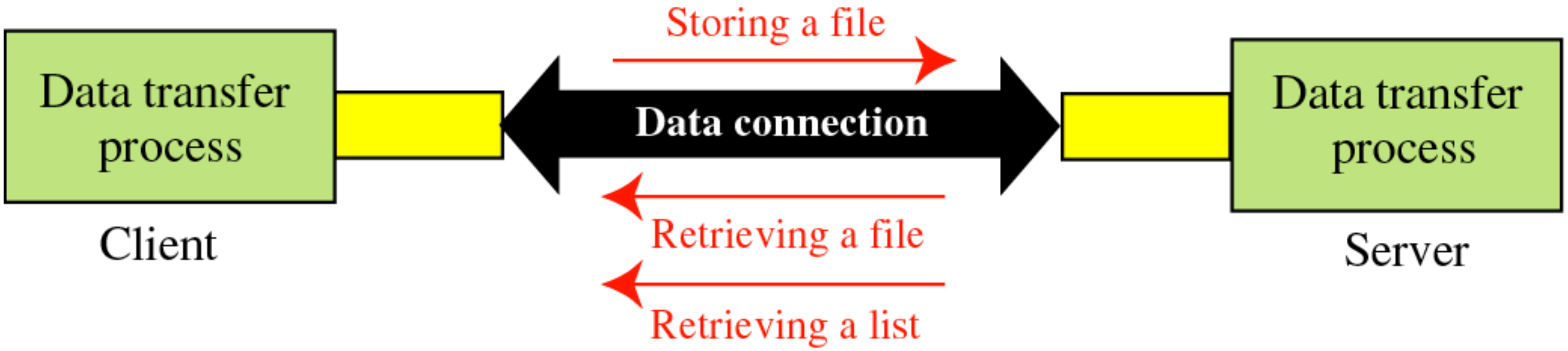
c. Active open by server

Command Processing

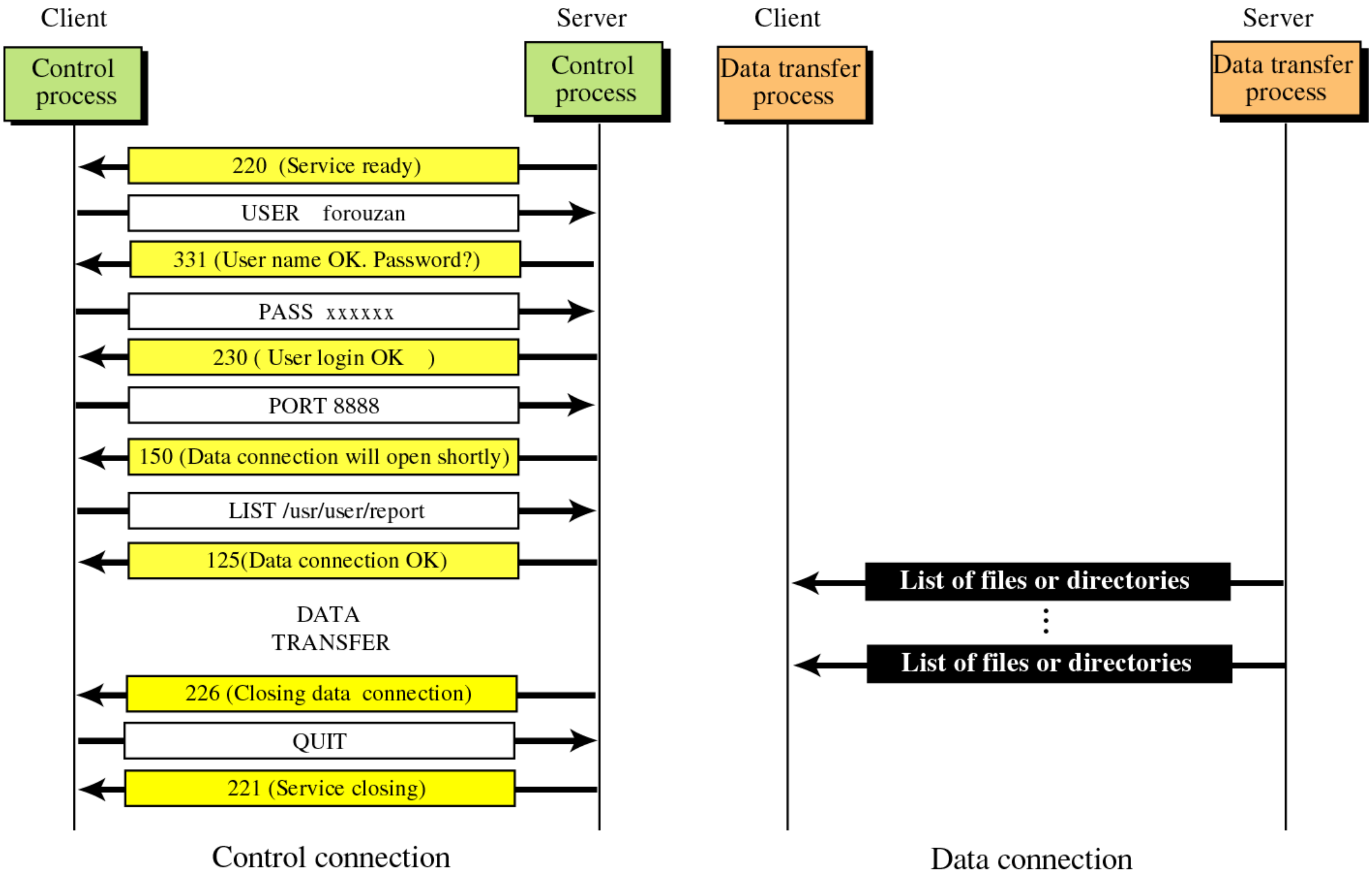
- Access Commands
- File Management
- Data Formatting
- Port defining
- File transfer
- Miscellaneous



File Transfer



FTP Example


Client
Control process
Server
Control process
Client
Data transfer process
Server
Data transfer process
220 (Service ready)
USER forouzan
331 (User name OK. Password?)
PASS xxxxxx
230 (User login OK)
PORT 8888
150 (Data connection will open shortly)
LIST /usr/user/report
125(Data connection OK)
DATA TRANSFER
226 (Closing data connection)
QUIT
221 (Service closing)
List of files or directories
⋮
List of files or directories
Control connection
Data connection

FTP Commands, Responses

- **Sample commands:**

  (sent as ASCII text over control channel)

  - USER username
  - PASS password
  - LIST return list of file in current directory
  - RETR filename retrieves (gets) file
  - STOR filename stores (puts) file onto remote
  - host

- **Sample return codes:**

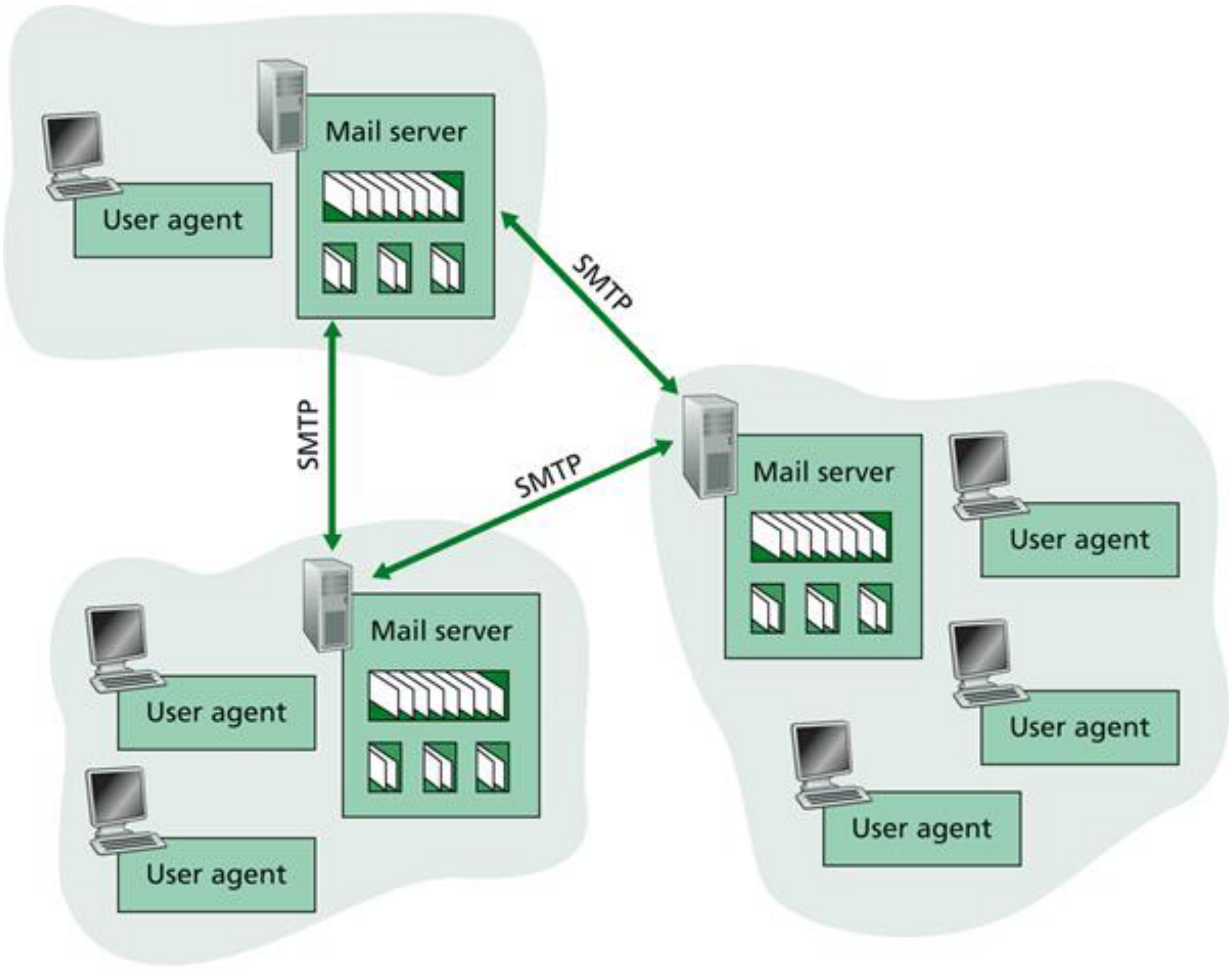
  (status code and phrase, as in HTTP)

  - 331 Username OK, password required
  - 125 data connection already open; transfer starting
  - 425 Can’t open data connection
  - 452 Error writing file

Electronic Mail

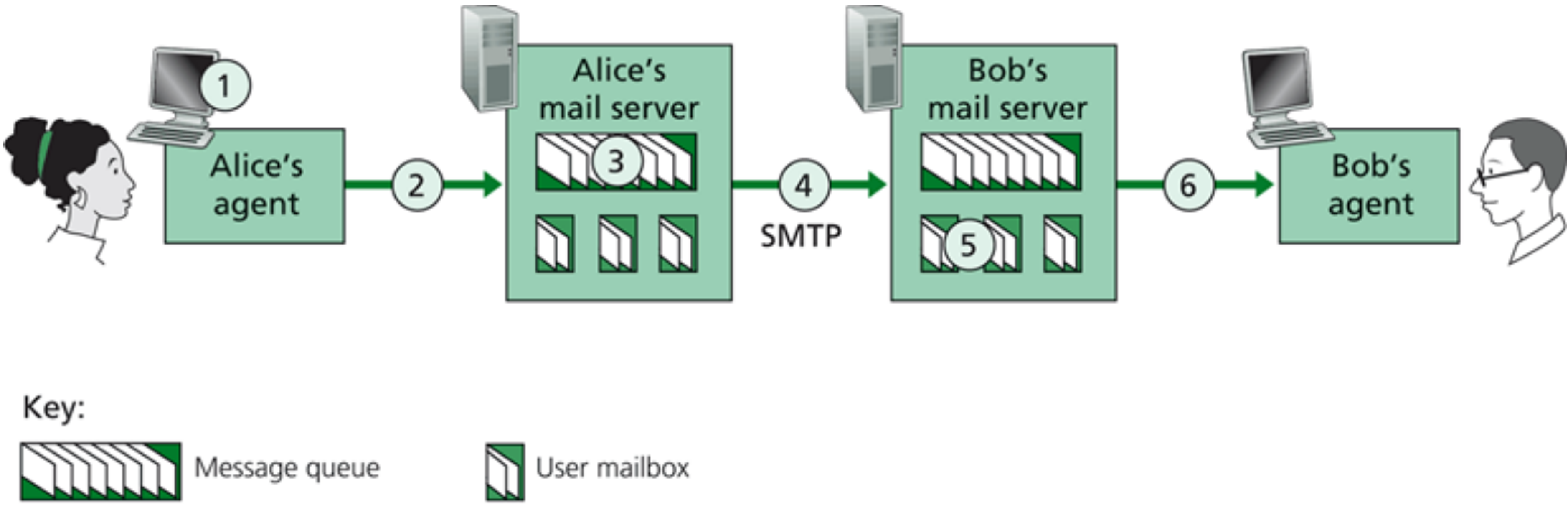
- Mail Send Protocol
  - Simple Mail Transfer Protocol(SMTP)
    - Port Number 25/TCP
    - Relay Mail From One domain to another or within same domain
- Mail Access Protocol
  - Post Office Protocol v3 (POP3)
    - Port number 110/TCP
    - Access mail from Mail server
  - Internet Mail Access Protocol (IMAP)
    - Web Based mail access protocol
    - Port Number 143/TCP

High Level View of E-mail System



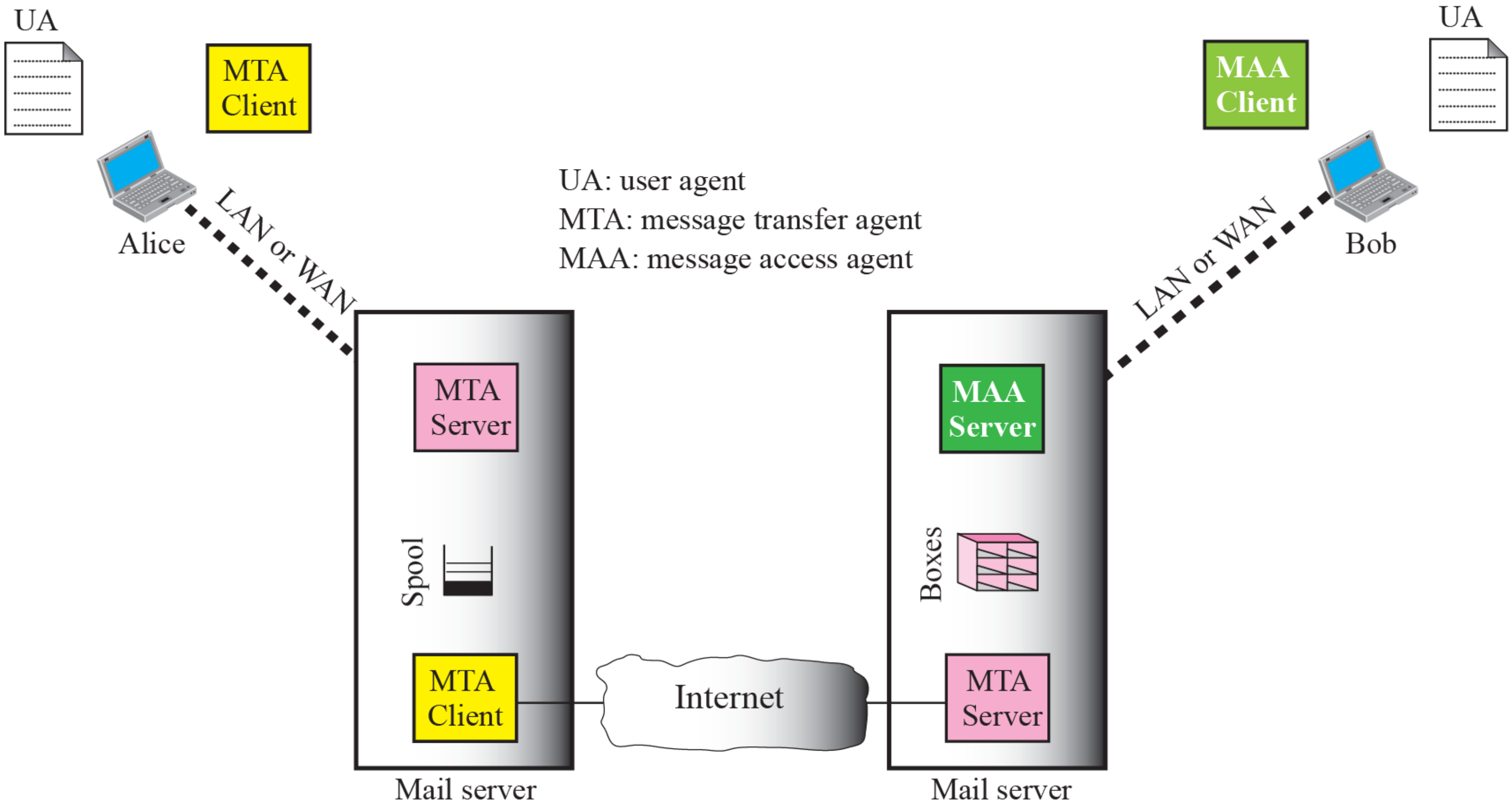
♦ A high-level view of the Internet e-mail system

Send Email



- Alice sends a message to Bob

E-mail System



When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today.

Components of Email System

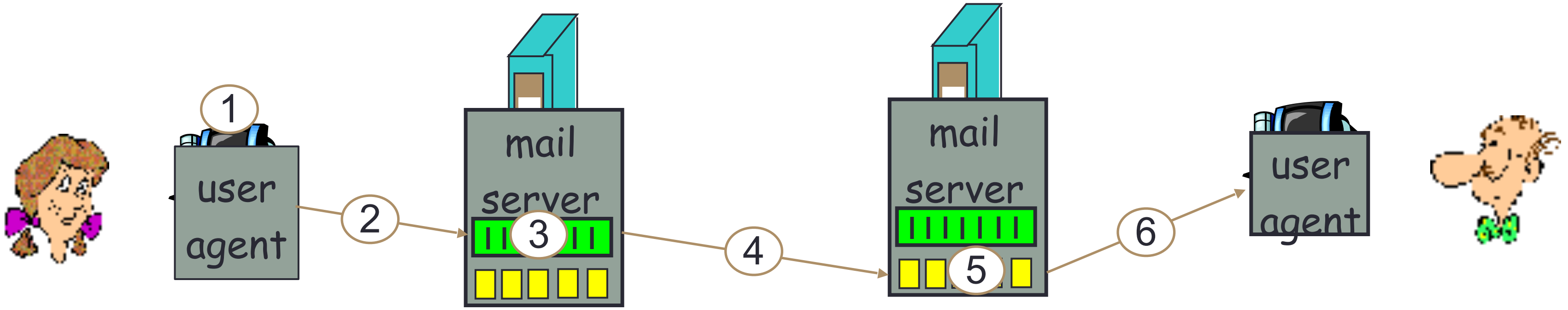
- User Agent
  - The first component of an electronic mail system is the user agent (UA). It provides service to the user to make the process of sending and receiving a message easier.
- Message Transfer Agent
  - The actual mail transfer is done through message transfer agents (MTAs). To send mail, a system must have the client MTA, and to receive mail, a system must have a server MTA. The formal protocol that defines the MTA client and server in the Internet is called Simple Mail Transfer Protocol (SMTP). Two pairs of MTA client-server programs are used in the most common situation.
- Message Access Agents
  - The first and the second stages of mail delivery use SMTP. However, SMTP is not involved in the third stage because SMTP is a push protocol; it pushes the message from the client to the server. On the other hand, the third stage needs a pull protocol; the client must pull messages from the server. The third stage uses a message access agent.

SMTP(Simple Mail Transfer Protocol)

- SMTP is a principle application-layer protocol for Internet e-mail.
- It uses the reliable transfer service of TCP, uses port 25
- Direct transfer: sending server to receiving server
- Three phases of transfer:
  - 1) handshaking (greeting) 2) transfer of messages 3) closure
- Messages must be in 7-bit ASCII
- 7-bit ASCII restriction is a bit of pain:- requires binary multimedia data to be encoded to ASCII before being sent over SMTP

Scenario: Alice sends message to Bob

1) Alice uses UA to compose message to: `bob@someschool.edu`
2) Alice's UA sends message to her mail server; message placed in message queue
3) Client side of SMTP opens TCP connection with Bob's mail server
4) SMTP client sends Alice's message over the TCP connection
5) Bob's mail server places the message in Bob's mailbox
6) Bob invokes his user agent to read message

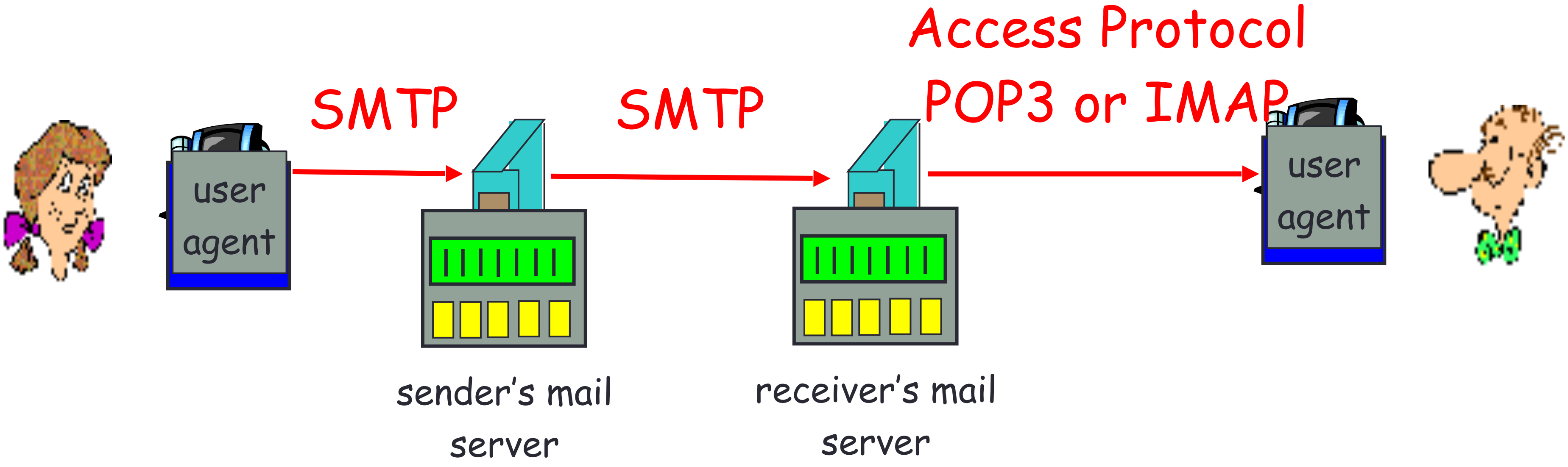


Mail Access Protocols (Pull Protocols)

- SMTP was a Mail Transfer Protocol or push Protocol and used to push the mail message up to the receiver's mail server.

- Mail Access Protocol: retrieval from server
  - **HTTP** is also used to Compose and retrieve Emails.
  - Also called Web based email.
  - Eg. Hotmail, Yahoo Mail Etc.
- POP: Post Office Protocol (POP3)
  - authorization (agent <-->server) and download (deleted from server)
  - TCP Port no. 110
- IMAP: Internet Mail Access Protocol
  - more features (more complex).
  - TCP Port no. 143
  - Remote manipulation of stored messages on server

POP3 and IMAP



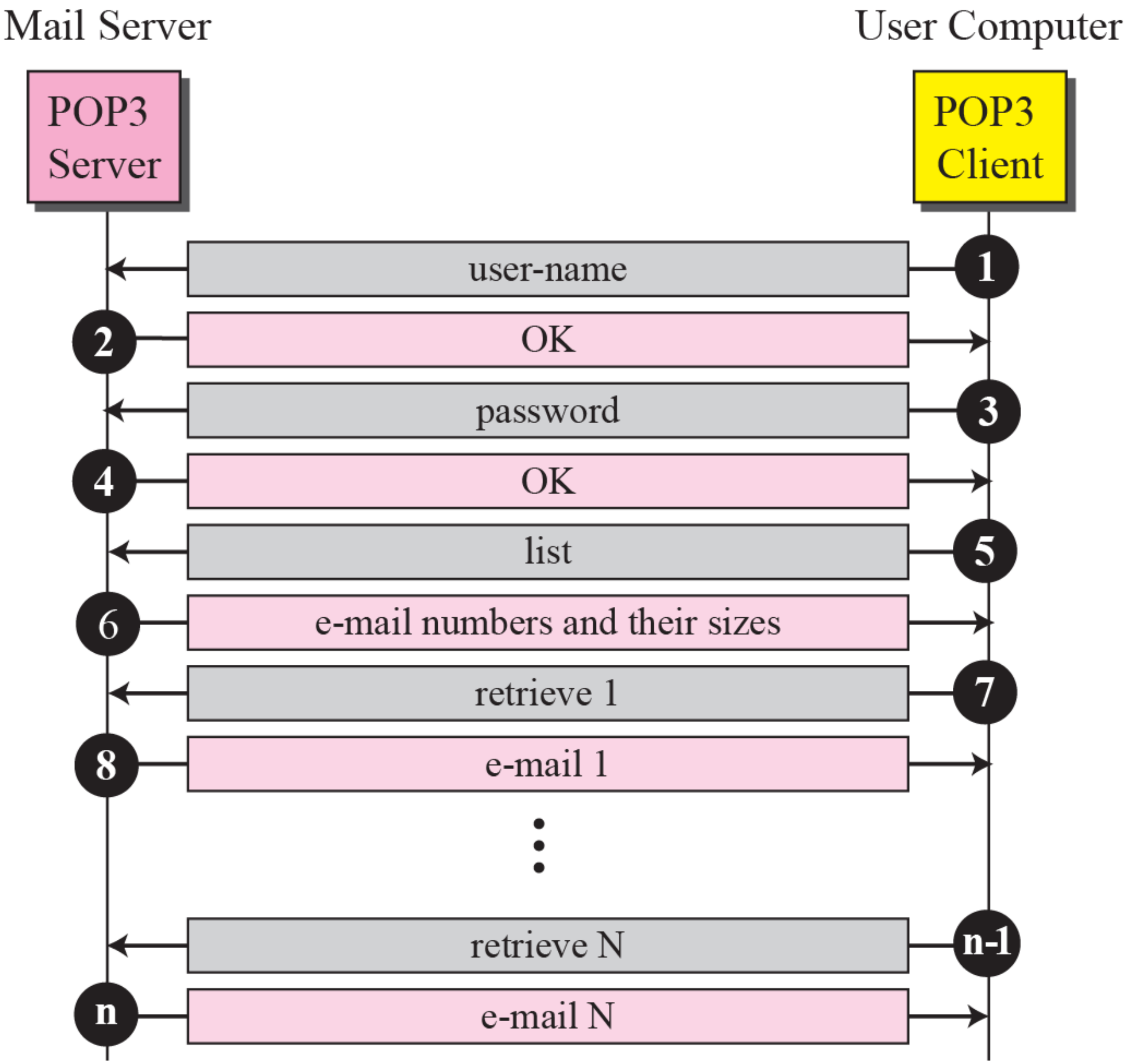
POP3 (Post Office Protocol)

- POP3 begins when user agent opens a TCP connection to the mail server on port 110.
- Then POP3 progresses through 3 phases :
  - Authorization : user agent sends a username and password to authenticate the user
  - Transaction : user agent retrieves message, user-agent can mark messages for deletion, remove deletion mark
  - Update : occurs after client has issued the quit command, ending POP3 session, at this time mail server delete the marked messages

POP3

- POP3 is extremely simple mail access protocol and functionality is limited.
- POP3 can be configured to “download and delete”(default configuration) or “download and keep”.
- A problem with download and delete mode – user may want to access his mail messages form multiple machines.
- During POP3 session, POP3 server maintains some state information about which user messages have been marked for deletion.
- However, POP3 server does not carry state information across POP3 session.

POP3


Mail Server
User Computer
POP3 Server
POP3 Client
1
user-name
2
OK
3
password
4
OK
5
list
6
e-mail numbers and their sizes
7
retrieve 1
8
e-mail 1
n-1
retrieve N
n
e-mail N

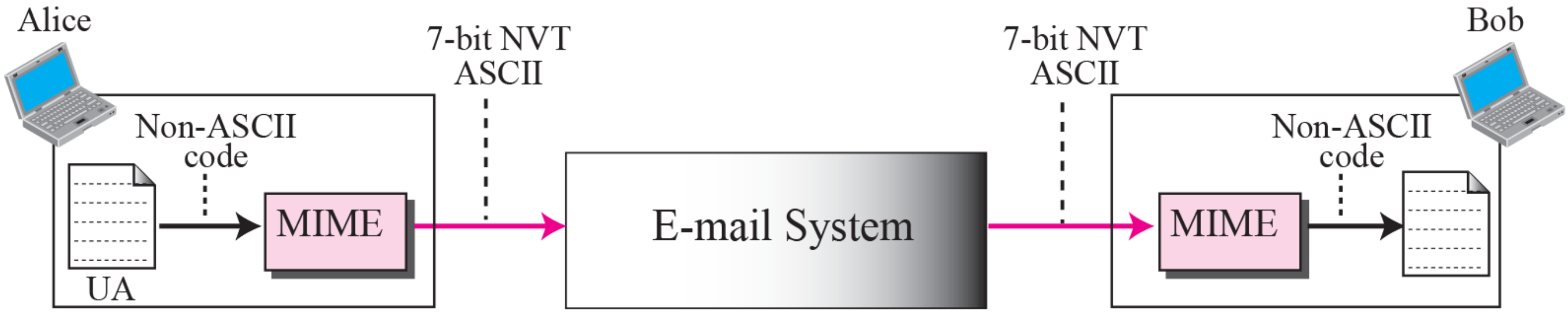
IMAP (Internet MAP)

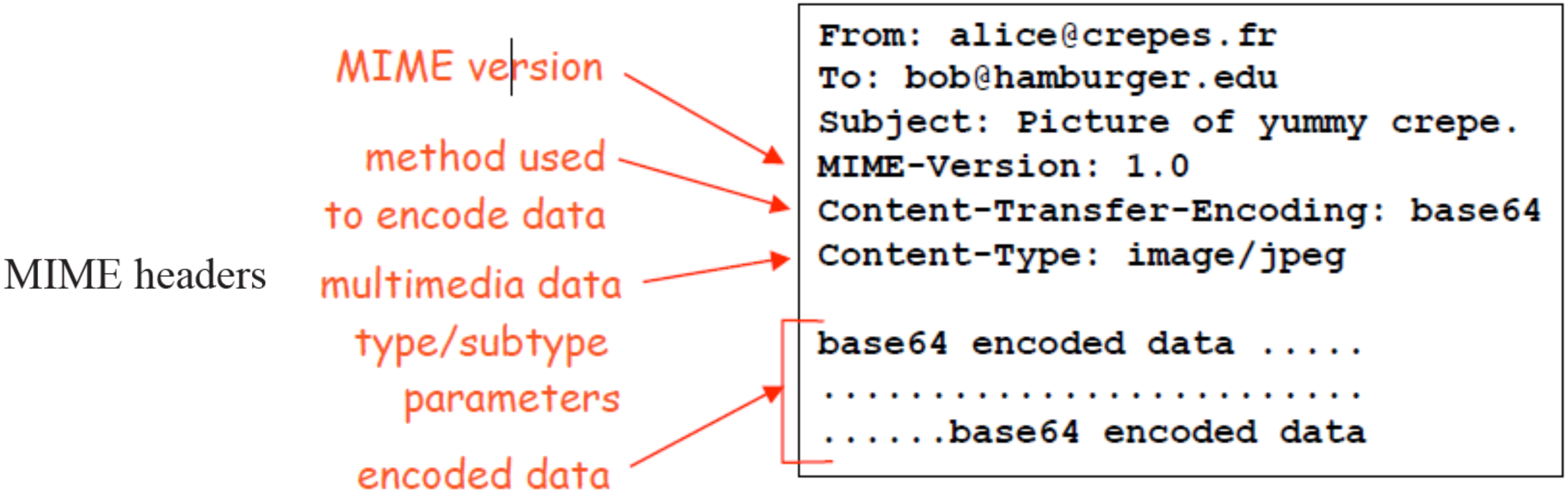
- User prefer to maintain a folder hierarchy on remote server that can be accessed from any computer.
- This is not possible with POP3 however IMAP protocol is defined to solve the issue.
- IMAP has more feature along with more complexity.
- When message first arrives, it is associated with recipient’s INBOX folder.
- The recipient can then move message from one folder to another.
- Unlike POP3, IMAP maintains state information across IMAP session,- for example, names of folders and which messages are associated with which folders.
- IMAP enable to obtain components of messages. (useful for low bandwidth).

MIME

- Electronic mail has a simple structure. Its simplicity, however, comes with a price. It can send messages only in NVT(Network Virtual Terminal) 7-bit ASCII format. In other words, it has some limitations.
- Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data.

MIME





MIME headers

MIME version

method used to encode data

multimedia data type/subtype parameters

encoded data

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data
```

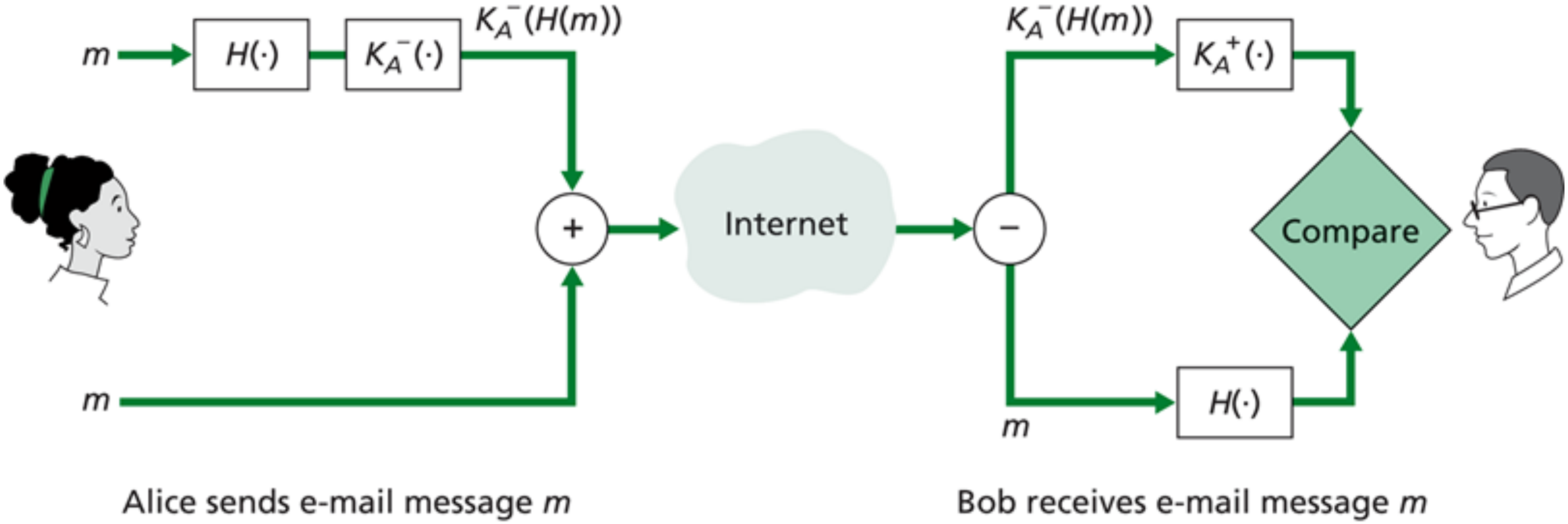
Common MIME types

- Content-Type: type/subtype; parameters
  - Text
    - plain; html
  - Image
    - jpeg; gif
  - Audio
    - basic; 32kadpcm
  - Video
    - quicktime
  - Application
    - msword; octet-stream

Securing Email Using Pretty Good Privacy (PGP)

- pretty good privacy (PGP) is e-mail an encryption scheme that has become a de-facto standard, with thousands of users all over the globe.
- Depending on the version, the PGP software uses MD5 or SHA for calculating the message digest; CAST, Triple-DES or IDEA for symmetric key encryption; and RSA for the public key encryption.
- In addition, PGP provides data compression.
- When PGP is installed, the software creates a public key pair for the user.
- The public key can be posted on the user's Web site or placed in a public key server.
- The private key is protected by the use of a password. The password has to be entered every time the user accesses the private key.
- PGP gives the user the option of digitally signing the message, encrypting the message, or both digitally signing and encrypting.

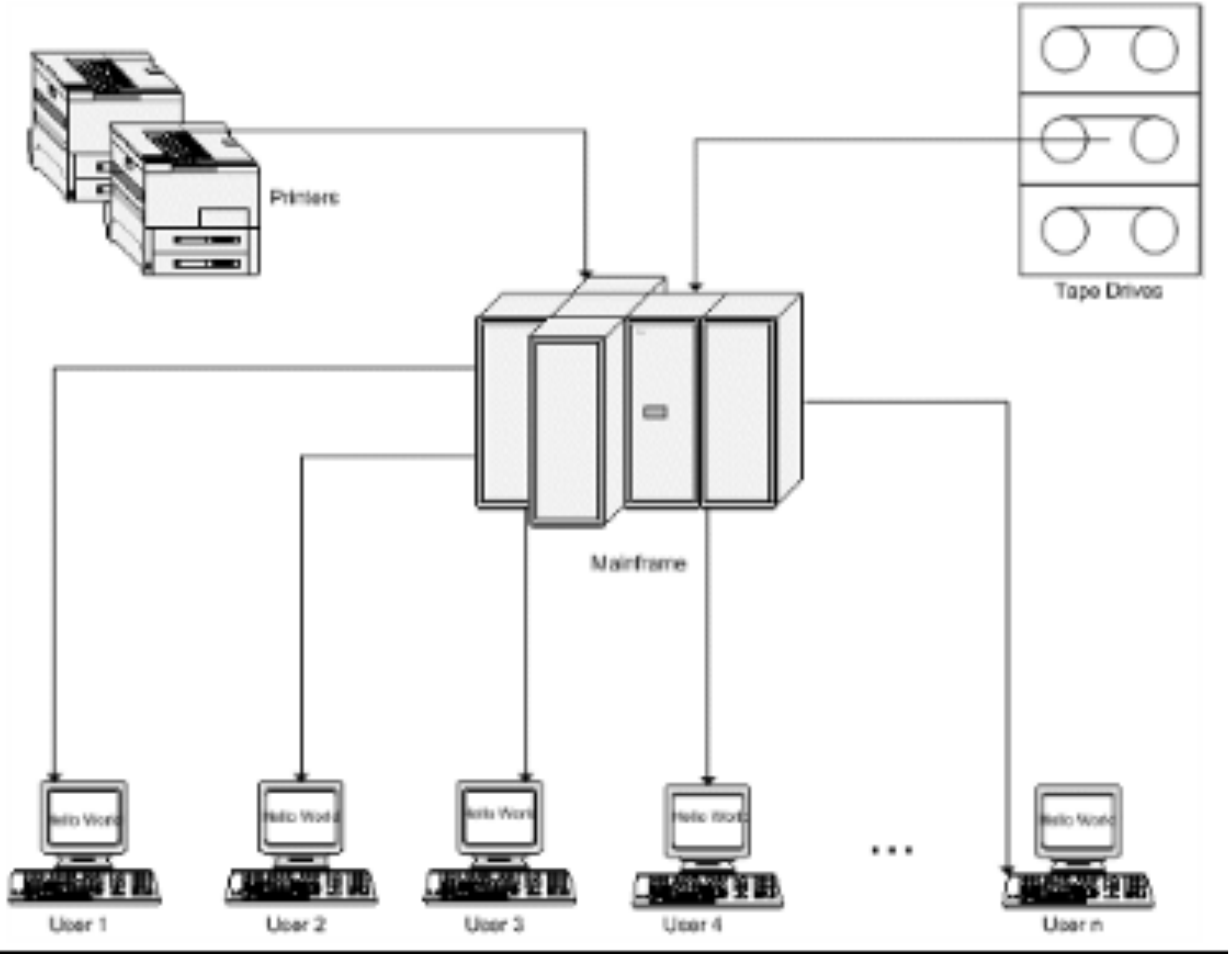
PGP



- Using hash functions and digital signatures to provide sender authentication and message integrity

Tiered Architecture

- 1- Tier Architecture

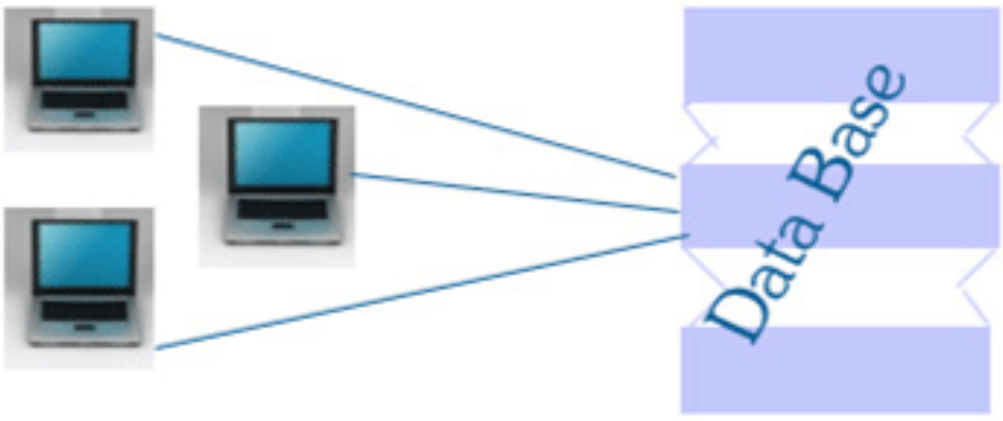


1-Tier Architecture

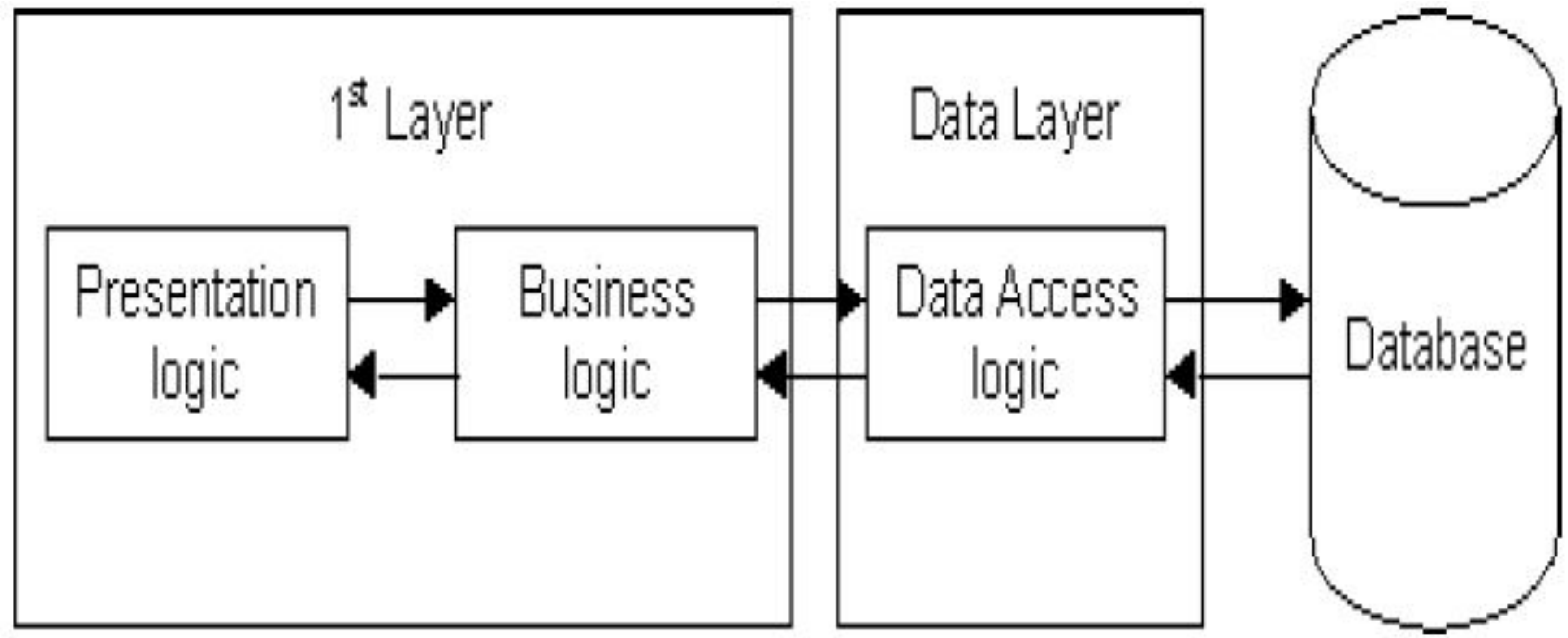
- Time of Huge Mainframe.
- All processing in a single computer.
- All resources attached to the same computer.
- Access via dumb terminals.
- In the context of traditional computer terminals that communicate over a serial RS-232 connection, dumb terminals are those that can interpret a limited number of control codes (CR Carriage return , LF Line Feed, etc.) but do not have the ability to process special escape sequences that perform functions such as clearing a line, clearing the screen, or controlling cursor position.
- In this context dumb terminals are sometimes dubbed glass Teletypes, for they essentially have the same limited functionality as does a mechanical Teletype.

2-Tier Architecture

- The two-tier architecture is like client server application.
- The direct communication takes place between client and server.
- There is no intermediate between client and server.
- The two tiers of two-tier architecture is:
  - Database (Data tier)
  - Client Application (Client tier)
- Business Logic Implemented at client side.



2-Tier Architecture



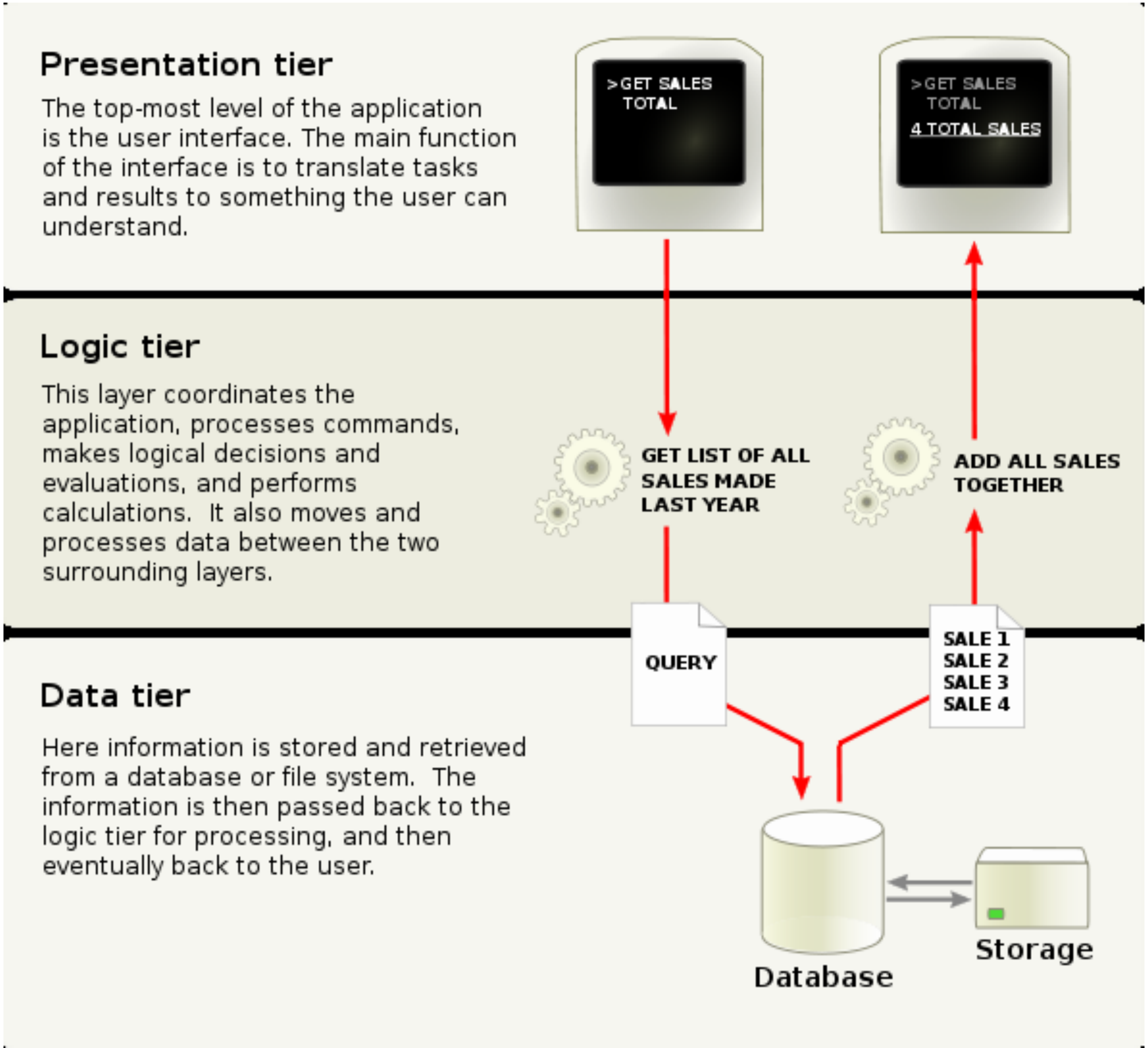
2-Tier Architecture

- Advantages:
  - Understanding and maintenances is easier.
- Disadvantages:
  - Business-logic is implemented on the PC
  - Increased network traffic
  - Application logic can’t be reused
  - Must design/implement protocol for communication between client and server.
  - Performance will be reduced when there are more users.

3-Tier Architecture

- A three way Interaction in a client/Server environment
  - The user interface is stored in the client
  - The bulk of the business application logic is stored in one or more servers
  - The data is stored in database server
- **Presentation tier** This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/ client tier and all other tiers in the network.
- **Application tier** (business logic, logic tier, data access tier, or middle tier) The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing.
- **Data tier** This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

3-Tier Architecture



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

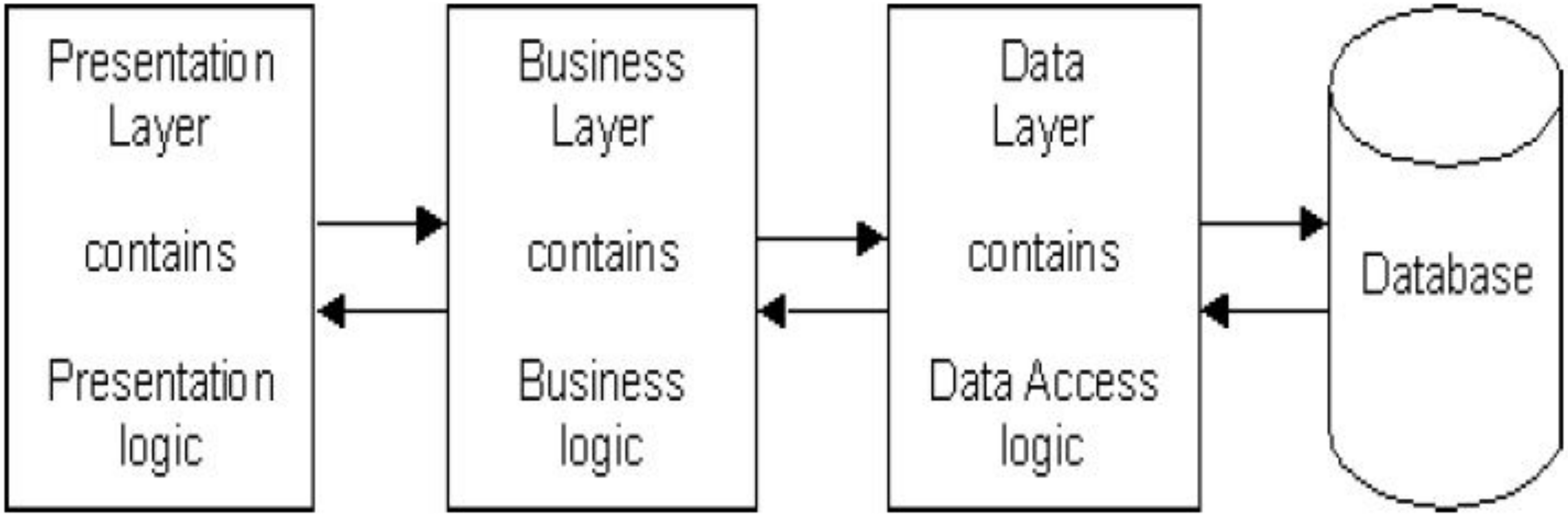
Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

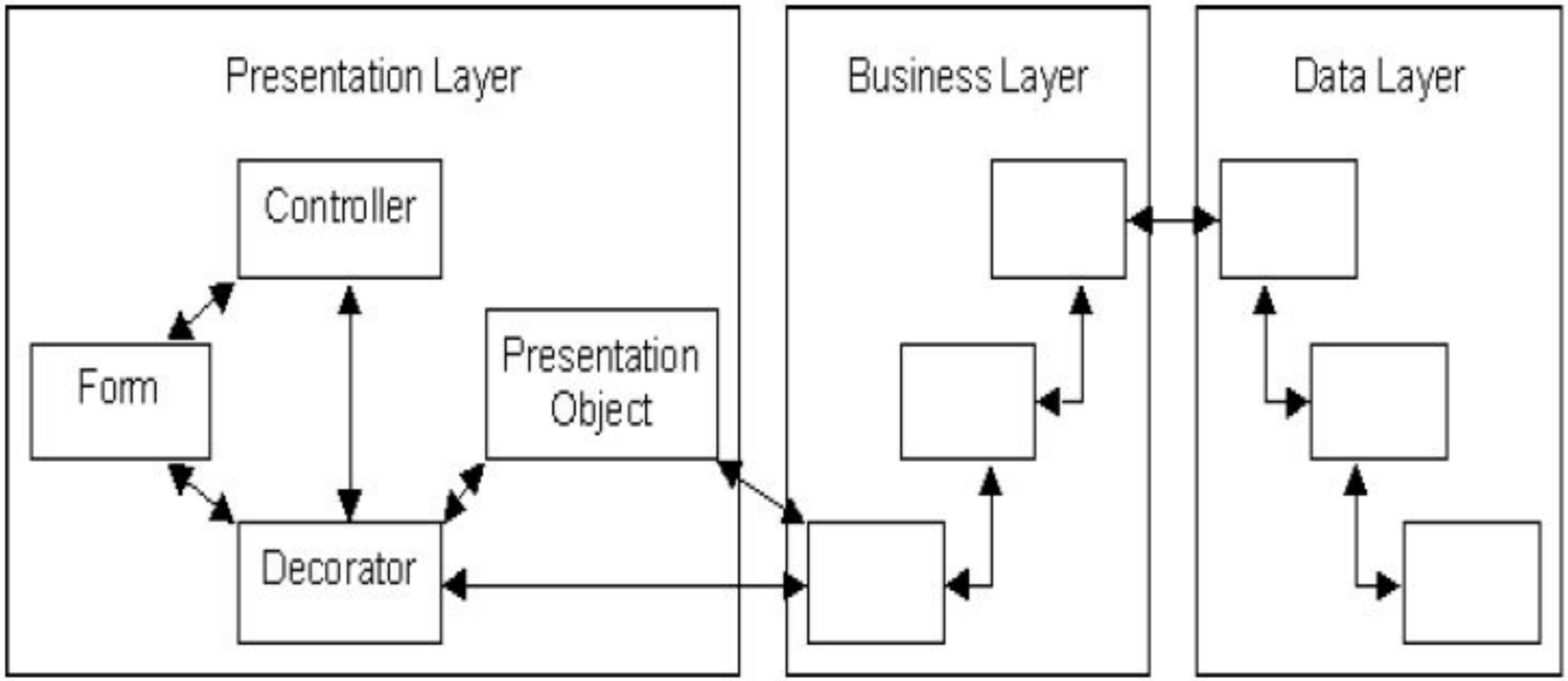
3-Tier Architecture



- **Advantages**
  - Clear separation of user-interface-control and data presentation from application-logic.
  - Change in business logic wont need change in other layers.
  - Dynamic load balancing by use of multiple servers
  - Fast communication
  - Performance will be good in three tier architecture.

N-Tier Architecture

- Also known as the “layered” architecture.
- N usually denotes 3 or more tiers (layers).
- Can be used to model both a web-based application and a desktop application.



N-Tier Architecture

- Advantages
  - **Easy to change**: you can decide to switch from desktop applications to web based applications by just changing the UI layer (a small part of the application). The same thing with the database system.
  - **Easy to manage**: if each layer has its own functionality, when something needs to be changed you will know what to change
  - **Easy to reuse:** if another application is developed for the same domain, it can use a big part of the business layer
  - **Easy to develop:** each layer can be developed by separate teams, and focus only on theirs specific problems (you don’t have to know HTML, ASP, OO design and SQL at the same time)

Universal Internet Browsing

- A web browser is a software application for retrieving, presenting and traversing information resources on the world wide web (www). An information resource is identified by a Uniform Resource Identifier (URI) and may be a webpage, image, video, or other piece od content.
- Hyperlinks present in resources enable users easily to navigate their browsers to related resources.
- A web browser can also be defined as an application software or a program designed to enable users to access, retrieve and view documents and other resources on the Internet.
- Although browsers are primarily intended to access the www, they can also be used to access information provided by web servers in networks or files in file systems.

Multiprotocol Support

- **Example: MPLS**
- Multiprotocol Label Switching (MPLS) is a mechanism in high-performance telecommunications networks that directs data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table.
- The labels identify virtual links (paths) between distant nodes rather than endpoints.
- MPLS can encapsulate packets of various network protocols.
- The MPLS working group is responsible for standardizing a base technology for using label switching and for the implementation of label-switched paths over various packet based link-level technologies, such as Packet-over-Sonnet, Frame Relay, ATM and LAN technologies(e.g. All forms of Ethernet, Token Ring, etc.).
- This includes procedures and protocols for the distribution of labels between routers and encapsulation.

MPLS

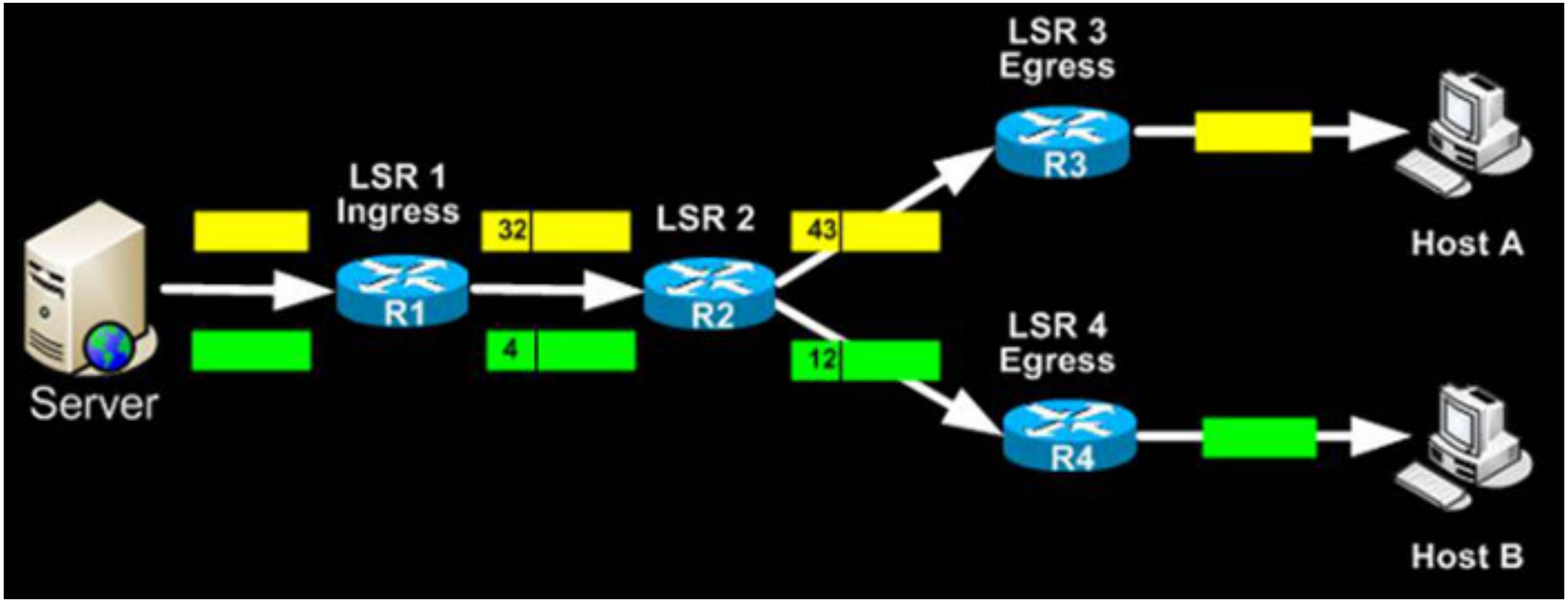
- MPLS allows most packets to be forwarded at Layer 2 (the switching level) rather than having to be passed up to Layer 3 (the routing level).
- Each packet gets labelled on entry into the service provider's network by the ingress router.
- All the subsequent routing switches perform packet forwarding based only on those labels—they never look as far as the IP header.
- Finally, the egress router removes the label(s) and forwards the original IP packet toward its final destination.
- The label determines which pre-determined path the packet will follow.
- The paths, which are called label-switched paths (LSPs), allow service providers to decide ahead of time what will be the best way for certain types of traffic to flow within a private or public network.

MPLS

- Service providers can use MPLS to improve quality of service (QoS) by defining LSPs that can meet specific service level agreements (SLAs) on traffic latency, jitter, packet loss and downtime.
- For example, a network might have three service levels -- one level for voice, one level for time-sensitive traffic and one level for “best effort” traffic.
- MPLS also supports traffic separation and the creation of virtual private networks (VPNs) virtual private LAN services (VPLS) and virtual leased lines (VLLs).
- MPLS got its name because it works with the Internet Protocol (IP), Asynchronous Transport Mode (ATM) and frame relay network protocols; any of these protocols can be used to create an LSP.
- It was created in the late 1990s to avoid having routers waste time by having to stop and look up routing tables.

MPLS

- A common misconception is that MPLS is only used on private networks, but the protocol is used for all service provider networks -- including Internet backbones.
- Today, Generalized Multi-Protocol Label Switching (GMPLS) extends MPLS to manage time division multiplexing (TDM), lambda switching and other classes of switching technologies beyond packet switching.



Thank You !!!