

## E-MAIL (SMTP, POP3, IMAP4); FTP

Dr. Manas Khatua

Assistant Professor

Dept. of CSE, IIT Guwahati

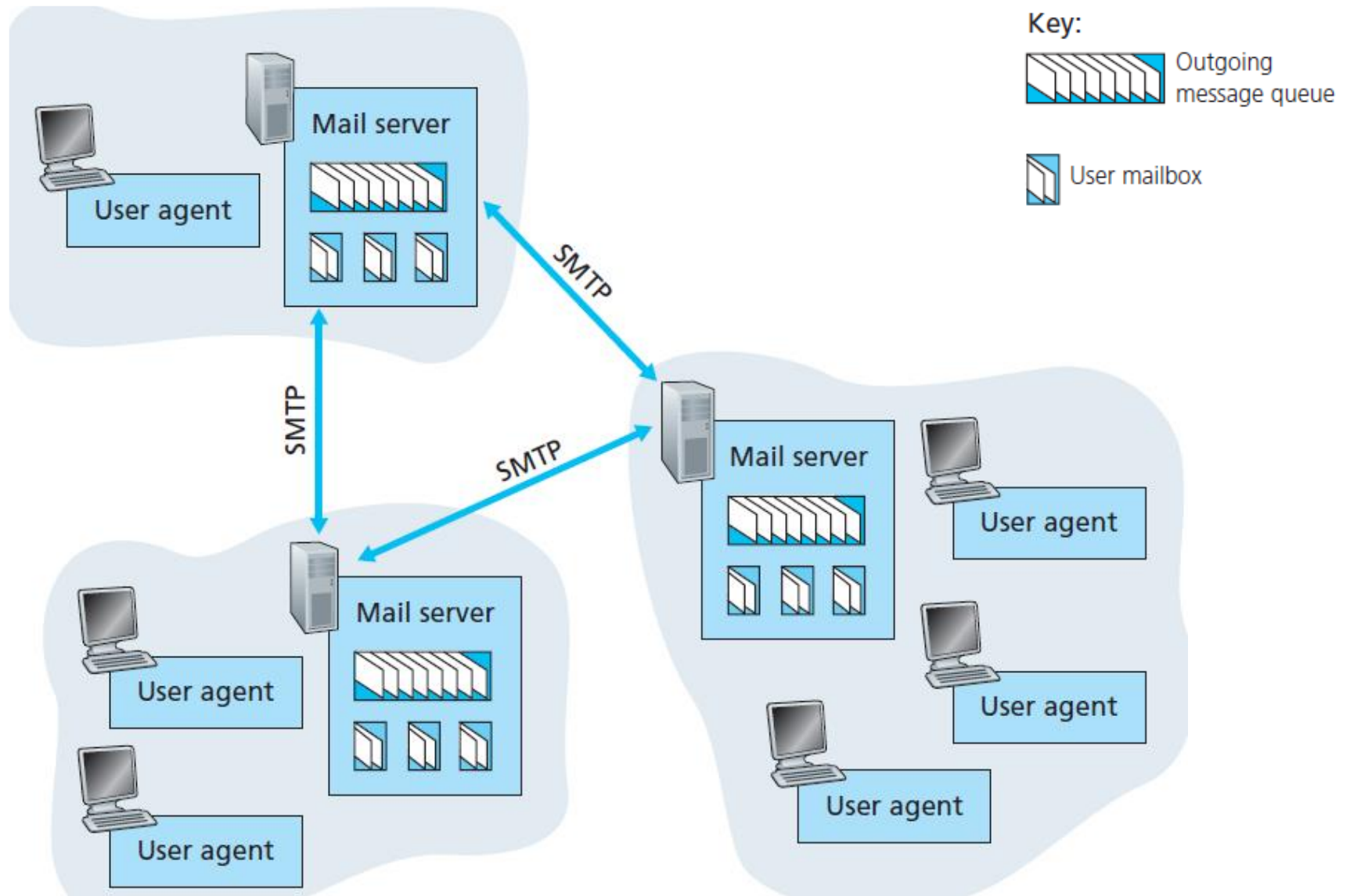
E-mail: [manaskhatua@iitg.ac.in](mailto:manaskhatua@iitg.ac.in)

# Electronic mail (E-mail)

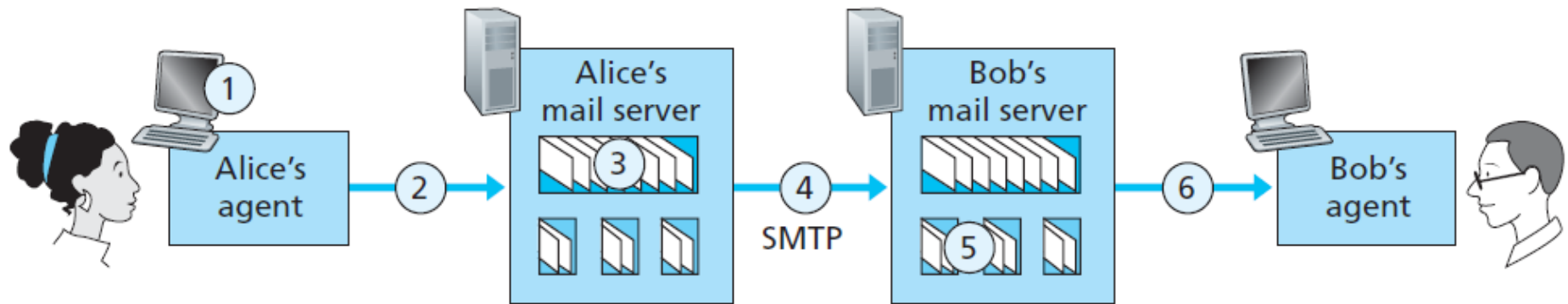


- Allows users to exchange messages.
- It is an asynchronous communication medium.
- In HTTP,
  - the server program is running all the time, waiting for a request from a client.
  - when the request arrives, the server provides the service.
- In E-mail:
  - It is considered as **one-way transaction**.
  - **Sender** may expect a response, but this is not a mandate.
  - it is neither feasible nor logical for the **receiver** to run a server program and wait until someone sends an e-mail to him.
  - the idea of **client/server** programming should be implemented in another way: using **intermediate servers**.
  - both the **end users** run only **client programs** when they want, and the **intermediate servers** apply the client/server paradigm

# High-level view of Internet e-mail system



# Architecture



- **User agent**
  - allows user to read, reply to, forward, save, and compose messages.
  - e.g., Microsoft Outlook, Google Gmail
- **Mail server**
  - form the core of the e-mail infrastructure
- **Mailbox**
  - Each user has a mailbox located in one of the mail servers.
- **Application-layer protocol**
  - transfer mail from the sender's mail server to the recipient's mail server
  - e.g., Simple Mail Transfer Protocol (SMTP)
  - SMTP has two sides: a client side, and a server side

## Journey of a message

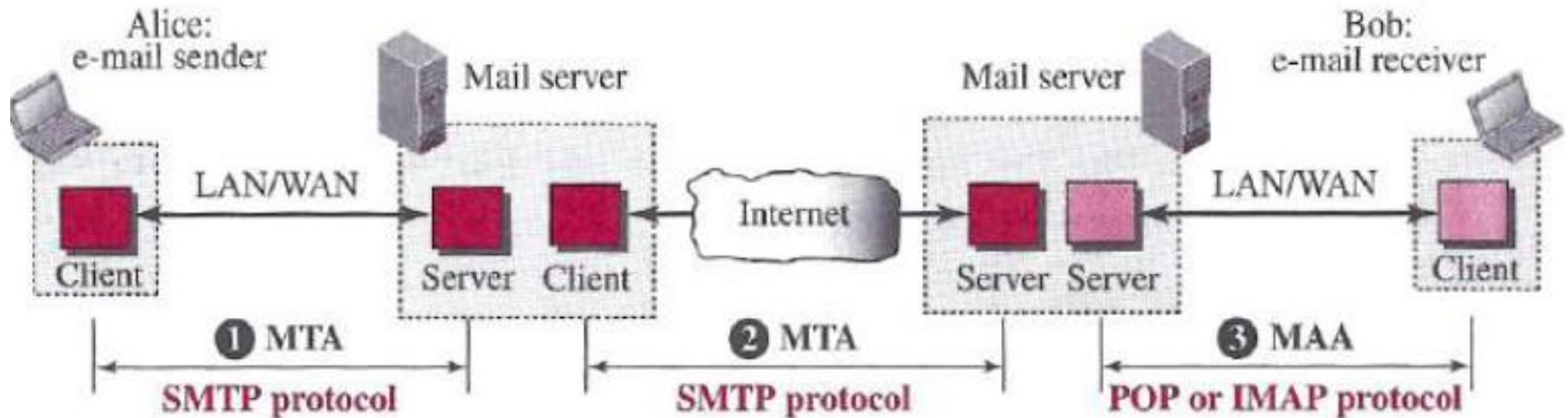
sender's user agent --> sender's mail server --> recipient's mail server --> recipient's user agent

# SMTP v/s HTTP



- HTTP transfers files (also called **objects**) from a Web **server** to a Web **client** (typically a browser)
- SMTP transfers files (that is, e-mail **messages**) from one mail **server** to another mail **server**.
- Both persistent HTTP and SMTP use persistent connections
- HTTP is mainly a **pull protocol** - someone loads information on a Web server and users use HTTP to pull the information from the server at their convenience
- SMTP is primarily a **push protocol** - the sending mail server pushes the file to the receiving mail server.
- In HTTP, TCP connection is initiated by the machine that wants to receive the file
- In SMTP, the TCP connection is initiated by the machine that wants to send the file
- SMTP requires each message, including the body of each message, to be in 7-bit ASCII format. This restriction made sense in the early 1980s when transmission capacity was scarce.
- HTTP data does not impose this restriction.

# Message Access Protocol: POP,IMAP



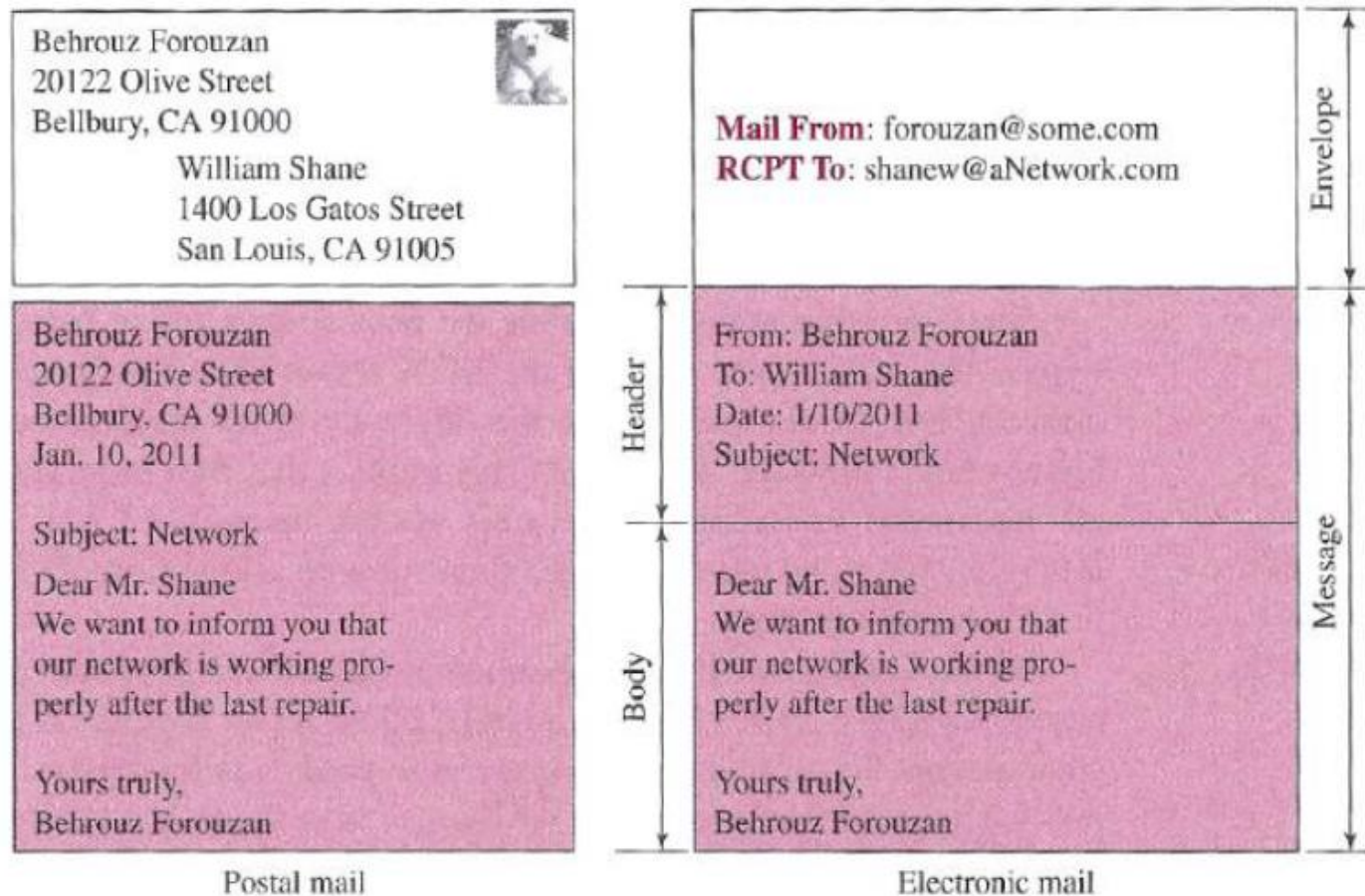
- Once SMTP delivers the message **from** Alice's mail server **to** Bob's mail server, the message is placed in Bob's mailbox.
- Until the early 1990s, Bob used to read his mail by logging onto the server.
- But today, **mail access** uses a **client-server architecture** - typical user reads e-mail with a client that executes on the user's end system
- **Bob's user agent can't use SMTP** to obtain the messages because **SMTP is a push protocol**
- Mail access protocols use by Bob's user agent
  - Post Office Protocol—Version 3 (POP3),
  - Internet Mail Access Protocol (IMAP)

# Mail Transfer Phases

- The process of transferring a mail message occurs in **three phases**:
  - connection establishment,
  - mail transfer,
  - connection termination.
- After a client has made a **TCP connection** to the well-known **port 25**, the SMTP protocol starts its connection phase.

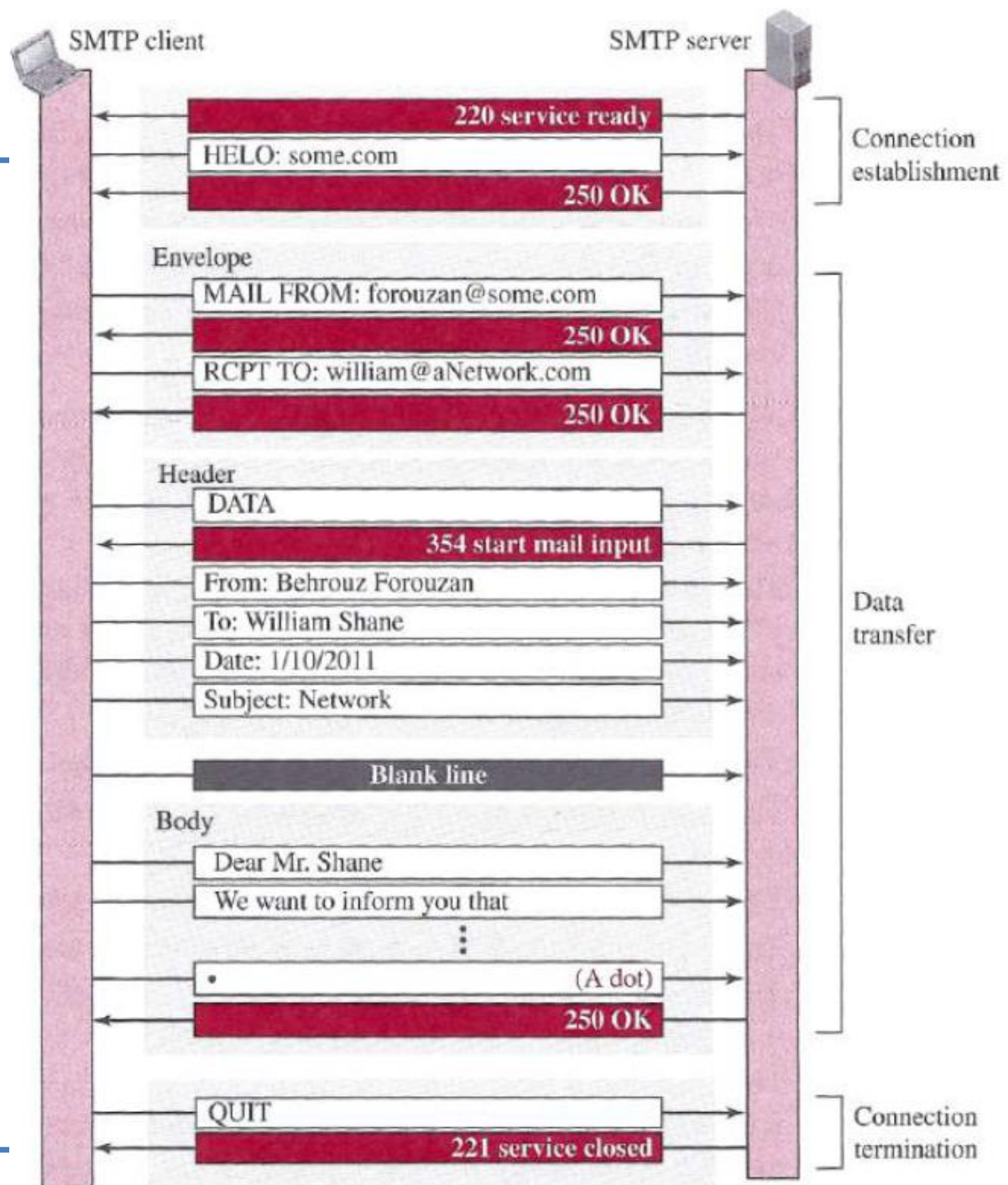
# Cont...

- To **send mail**, the user, through the user agent (UA), creates mail that looks very similar to postal mail.

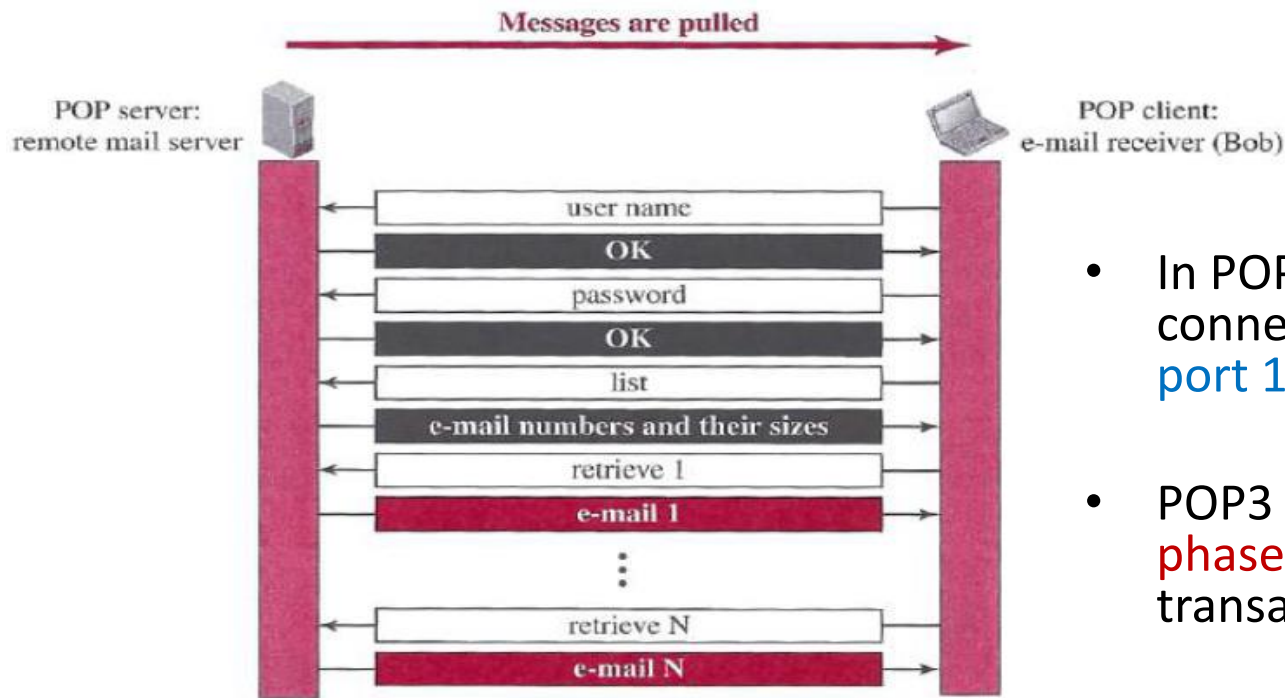




# Cont...



# POP3



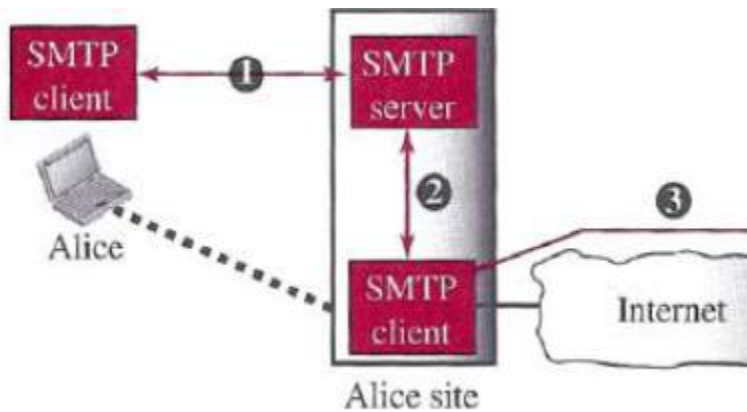
- In POP3, user agent opens a TCP connection to the mail server on **port 110**.
- POP3 progresses through **three phases**: authorization, transaction, and update

- In POP3 transaction, the user agent (UA) issues **commands**, and the server replies to each command with a **response**. Two possible responses: +OK; and -ERR
- User agent can be configured to two modes:
  - “**download and delete**” or “**download and keep**.”
- The download-and-delete mode partitions Bob’s mail messages over the machines – downloaded to the accessing PC, and removed from the mail server

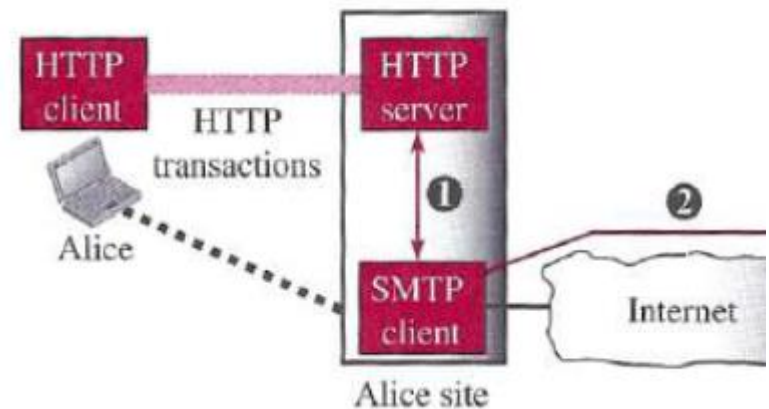
- IMAP4 is similar to POP3, but **it has more features**; IMAP4 is more powerful and more complex.
- IMAP4 provides the **following** extra functions:
  - A user can **check the e-mail header** prior to downloading.
  - A user can **search the contents** of the e-mail for a specific string of characters prior to downloading.
  - A user can **partially download e-mail**. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
  - An IMAP server will associate each message with a **folder**.
  - A user can **create, delete, or rename mailboxes (i.e. folders)** on the mail server.
  - A user can **create a hierarchy of mailboxes** in a folder for e-mail storage.

# Web-Based Mail

- Some websites today provide this service to anyone who accesses the site. E.g., [Hotmail](#), [Yahoo](#), and [Google](#) mail.



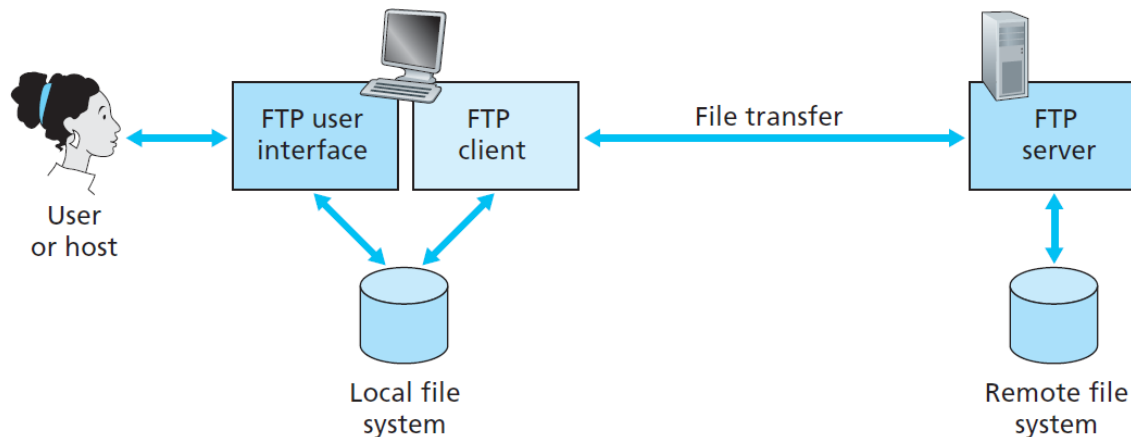
User agent uses SMTP



User agent uses HTTP

- The protocol discussed so far **does not provide any security** provisions per se.
- e-mail exchanges can be secured **using two application-layer securities** designed in particular for e-mail systems
  - Pretty Good Privacy (PGP)
  - Secure/Multipurpose Internet Mail Extensions (S/MIME)

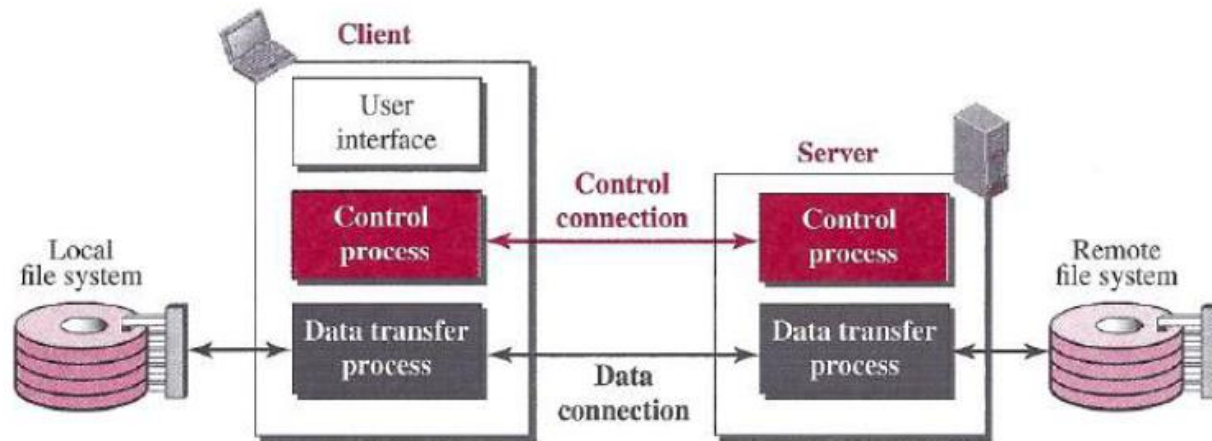
- File Transfer Protocol (**FTP**) is the standard protocol
- In a typical FTP session, the user is sitting in front of one host (the **local host**) and wants to **transfer files to or from a remote host**
- user accessing the remote account
  - user must provide a user **identification** and a **password**.



**Figure 2.14** ♦ FTP moves files between local and remote file systems

# Basic Model of FTP

- FTP must address the following:
  - two systems may use **different file name** conventions
  - two systems may have **different ways to represent** data
  - two systems may have **different directory structures**



- The **client** has **three components**:
  - the user interface, the client control process, and the client data transfer process.
- The **server** has **two components**:
  - the server control process and the server data transfer process.
- There are **two connects**:
  - control and data connection

# Cont...



- The two connections in FTP have **different lifetimes**.
  - The control connection **remains connected** during the entire interactive FTP session.
  - The data connection is **opened and then closed** for each file transfer activity
- FTP server uses two well-known TCP ports:
  - **port 21** is used for the **control connection**,
  - **port 20** is used for the **data connection**.
- **Benefits** for having two separate connections:
  - You can have **multiple data transfers** running at a time without having to establish multiple control connections.
  - No need for **complicated framing** on the control connection.
  - Handling special cases, like **cancelling a data connection**, is simpler.



# Control Connection

- Control communication is achieved through **commands** and **responses**.
- During this control connection, **commands are sent** from the **client to the server** and **responses are sent** from the **server to the client**.
- Every FTP command generates at least one response
- A **response** has two parts:
  - Three-digit number** : defines the code
  - Text** : defines needed parameters or further explanations

**Table 26.5** *Some responses in FTP*

Code	Description
125	Data connection open
150	File status OK
200	Command OK
220	Service ready
221	Service closing

**Table 26.4** *Some FTP commands*

Command	Argument(s)	Description
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password

# Data Connection

- the creation of a data connection is different from the control connection.
- **Data connection steps:**
  - The client, not the server, issues a **passive open** using an **ephemeral port (>1023)**.
  - Using the **PORT command** the client sends this port number to the server.
  - The server receives the port number and issues an **active open** using the **well-known port 20** and the received ephemeral port number.
- If the Client initiates the Data connection the FTP connection is **passive**. Only the server is required to open up ports for incoming traffic.
- If the data connection is initiated by the Server, the FTP connection is **active**

# Communication over Data Connection



- We prepare for data transmission through the control connection.
- The **heterogeneity** problem is resolved by defining three attributes of communication:
  - **file type**: *ASCII, EBCDIC, or image* file.
  - **data structure**: *file, record, or page* structure
  - **transmission mode**: *stream, block, or compressed* mode
- The **file structure** format (used by default) has no structure. It is a continuous stream of bytes.
- In the **record structure**, the file is divided into *records*. This can be used only with text files.
- In the **page structure**, the file is divided into pages, with each page having a page number and a page header.

# HTTP v/s FTP

- HTTP and FTP
  - Are both **application layer protocols**
  - are both **file transfer** protocols
  - they both run on top of TCP
  - FTP uses **two parallel TCP connections** to transfer a file, a **control connection** and a **data connection**.
  - HTTP sends request and response header lines into the same TCP connection that carries the transferred file itself
  - FTP is said to send its control information **out-of-band**
  - HTTP is said to send its control information **in-band**
  - with FTP, the **control connection remains open** throughout the duration of the user session, but a new data connection is created for each file transferred within a session (that is, the **data connections are non-persistent**)
  - Throughout a session, the FTP server must **maintain state** about the user.
  - HTTP, on the other hand, is **stateless** —it does not have to keep track of any user state.



# Security for FTP

- The FTP protocol was designed when security was not a big issue.
- Although FTP requires a password, **the password is sent in plaintext** (unencrypted)
- To be secure, one can add a **Secure Socket Layer** between the FTP application layer and the TCP layer.
- In this case FTP is called SSL-FTP

# Thanks!