

# Lecture 15

# APPLICATION LAYER

TELNET & EMAIL

# Topics Covered

- TELNET
- Client-Server Model
- Local Login
- Remote Login
- Concept of NVT
- ELECTRONIC MAIL
- Scenarios
- Applications

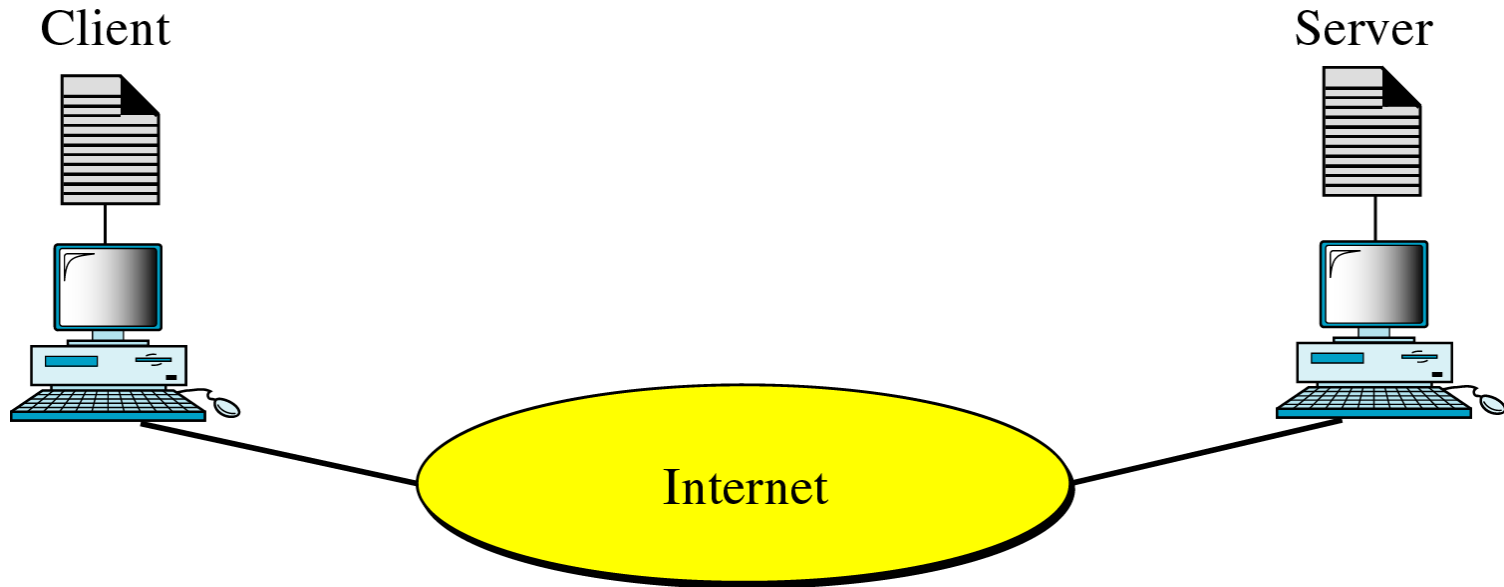
# TELNET

- In the Internet, users may want to run application programs at a remote site & create results that can be transferred to their local site.
- Students may want to connect to their university computer lab from their home to access application programs like major/minor projects.
- **One way** to satisfy that demand is to create a client/server application program for each service, BUT it's not possible to write specific program for each demand.
- **Better way**, A general purpose client/server program that lets a user access any application program on a remote computer
  - **Like Allow the user to log on to a remote computer.**
  - **After logging on user can use the services available on the remote computer & transfer result back to local computer.**

# Introduction: TELNET

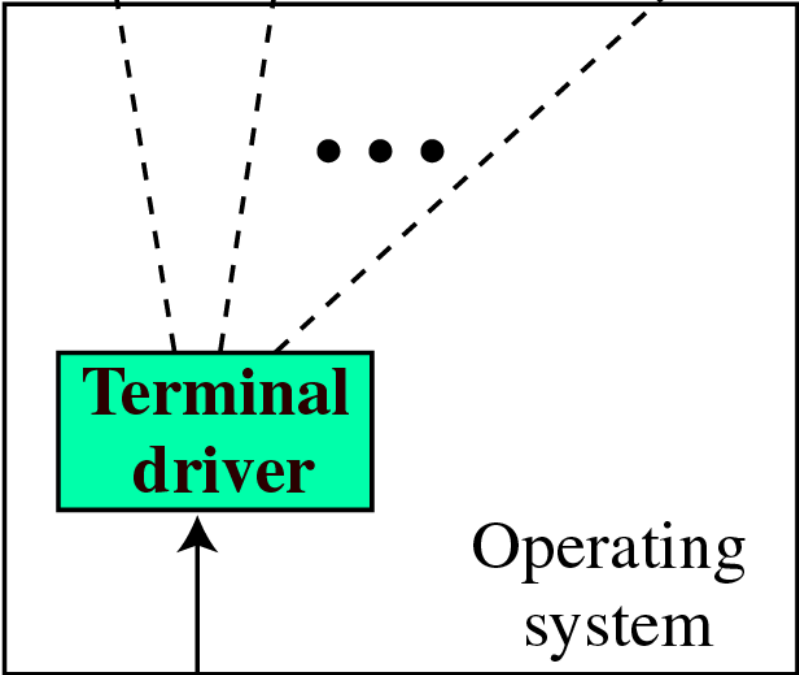
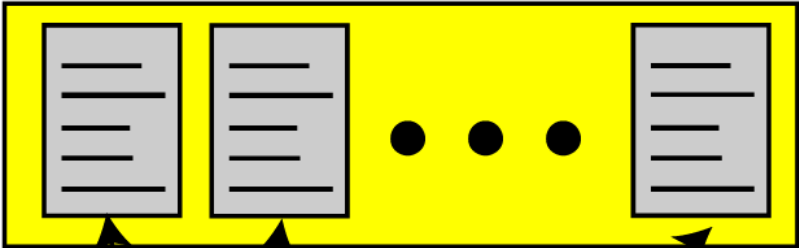
- Telnet is a popular general purpose client-server application program.
- TELNET is an abbreviation for **Terminal Network**.
- TELNET enables the **establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system.**
- Interaction between a user and a the computer occurs through a terminal
- User has an identification and password.
- User logs into the system with a user id and password to prevent unauthorized user from accessing the resources

# Client-Server Model



# Local Login

Application programs



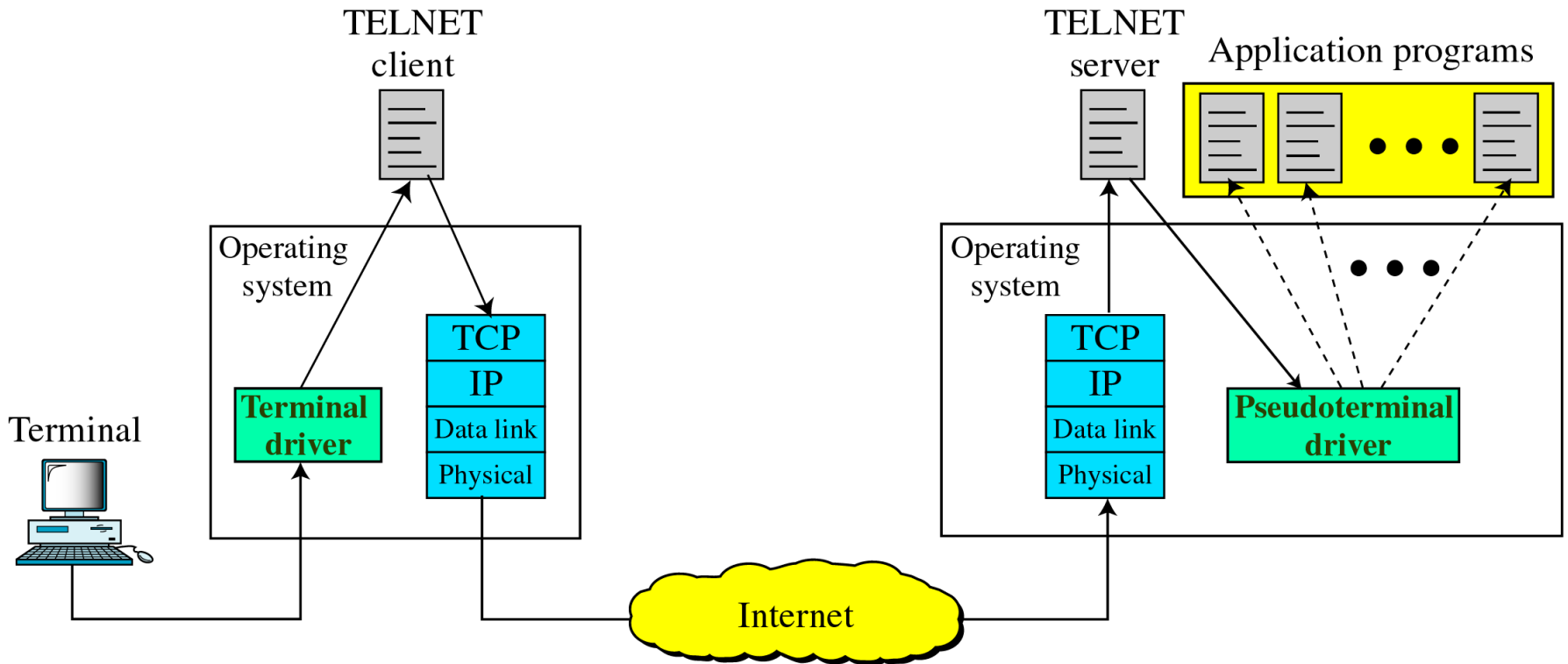
Terminal



# Local Login

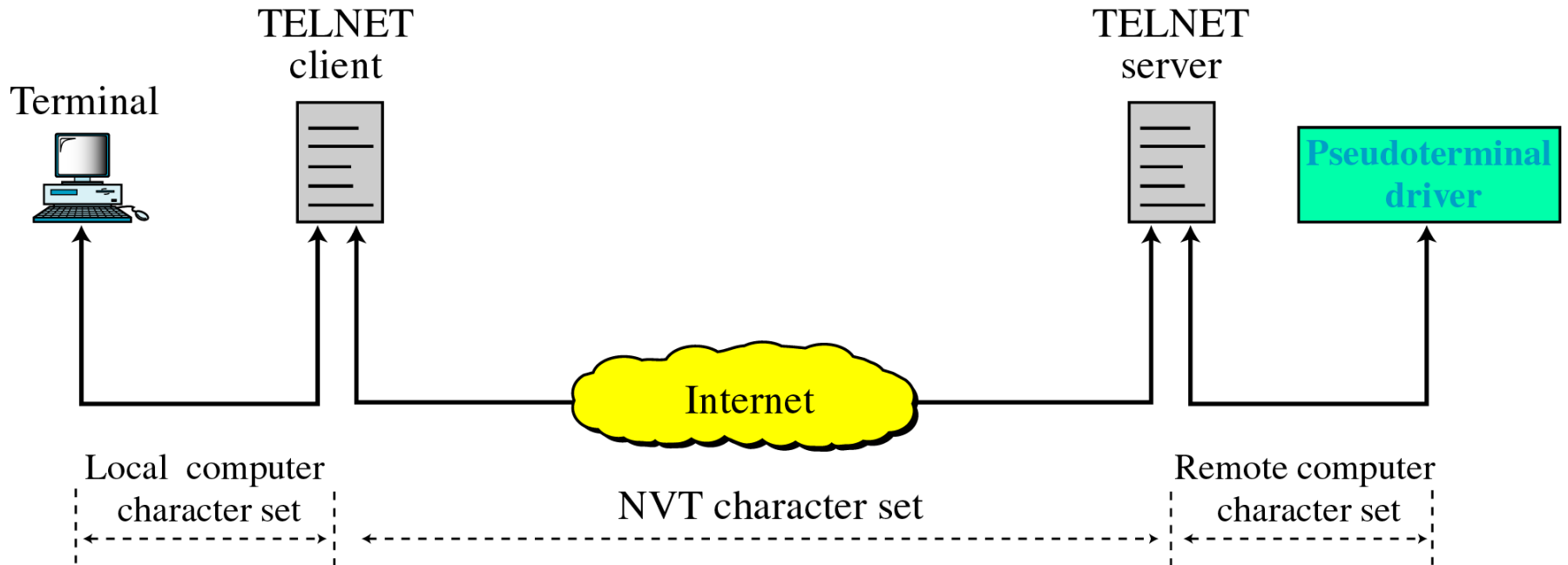
- When a user logs into a local time-sharing system, it is called Local Login.
- As a user types at a terminal, the keystrokes are accepted by the terminal driver.
- The terminal driver passes the characters to the O/S
- The O/S, in turn, interprets the combination of characters & invokes the desired application program or utility.

# Remote Login





# Concept of NVT



# REMOTE LOGIN

- When a **user wants to access an application program** or utility located on a remote machine, he or she performs remote login. Here the **TELNET client and server programs** come into use.
- The **user sends the keystrokes** to the terminal driver where the local O/S accepts the characters but doesn't interpret them.
- The **characters are sent to the TELNET client**, which transforms the characters to a universal character set called **network virtual terminal characters** and **delivers them to the local TCP/IP**
- The commands or text in **NVT form travel through the internet** and **arrive at the TCP/IP** at the remote machine.

# REMOTE LOGIN

- **Characters are delivered to the OS and passed to the telnet server** to the corresponding characters understandable by the remote computer
- **Characters cannot be passed directly to the operating system** because the remote Os is not designed to receive the characters from a TELNET server.
- **It can receive from terminal driver.** So, a **pseudo terminal server** is added which acts like as if characters are coming from local terminal.
- **OS passes the characters to the appropriate application program.**

# Network Virtual Terminal (NVT)

- Mechanism to access remote computer is complex.
- Every computer and its OS accepts special combination of characters.
  - Ex: end of file in DOS is ctrl+Z and in UNIX is ctrl+d
- If we want to access any remote computer in the world, we first must know to what type of computer we will be connected, and we also must install the specific terminal emulator used by that computer.
- TELNET solves this problem by defining a universal interface called the Network Virtual Terminal (NVT) character set.
- Via this interface, the client TELNET translates characters (data or command) that come from the local terminal into NVT form & delivers them to the network.
- The server TELNET, on the other hand, translates data & commands from NVT form into the form acceptable by the remote computer.

# ELECTRONIC MAIL

---

# How Email *Appears* to Work



Sender's Outbox



Recipient's Inbox

# How Email *Really* Works



Sender's Mail Client (MUA)



Sender's MDA/MTA

A

B

C

Company Network

Sender's Mail Server (MTA)



D

Router

The Internet

Router

Router

E



F



Recipient's Mail Client (MUA)



Recipient's MDA/MTA

Company Network

Recipient's Mail Server(s) (MTA)



## Step A: Sender creates and sends an email

- The originating sender creates an email in their [Mail User Agent \(MUA\)](#) and clicks 'Send'.
- The MUA is the application the originating sender uses to compose and read email, such as Eudora, Outlook, etc.

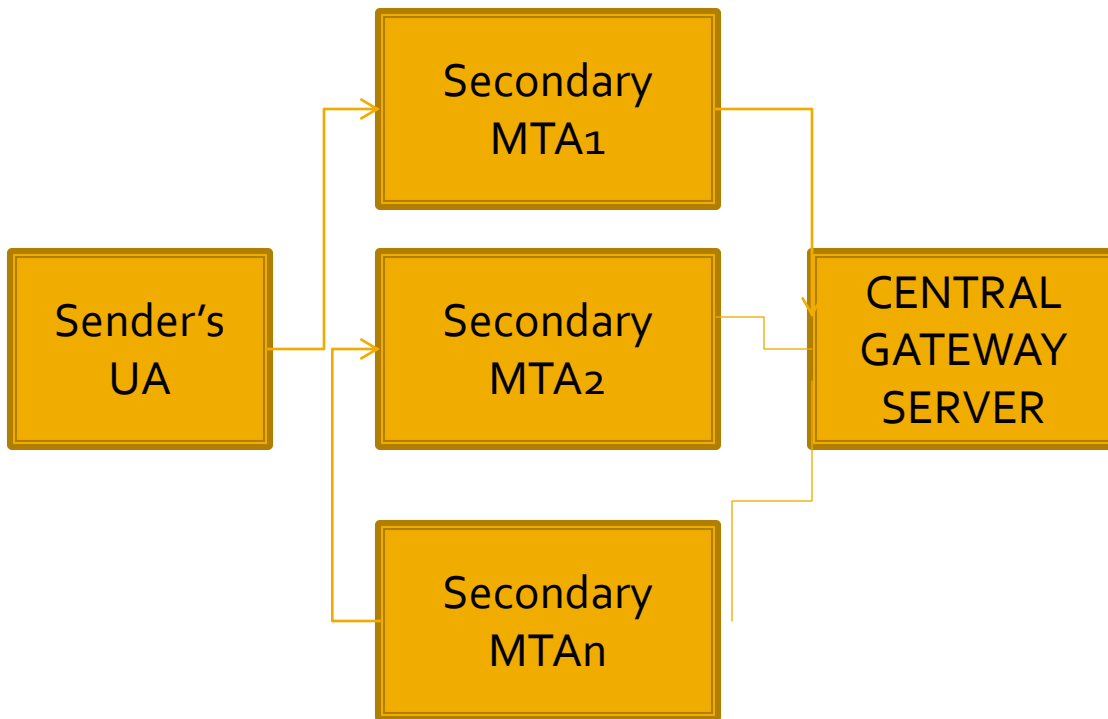
## Step B: Sender's MDA/MTA routes the email

- Sender's MTA accepts the email, then routes it to local mailboxes or forwards it if it isn't locally addressed.
- It forwards and enters the first of a series of "network clouds," labelled as a "Company Network" cloud.
- When we send an email, the email client finds the mail server using the domain name system and contacts it using the simple mail transfer protocol



# Step C: Network cloud

- An email can encounter a network cloud within a large company or the largest network cloud in existence: the Internet.
- **The network cloud may encompass a multitude of mail servers, DNS servers, routers.**
- Email service providers and other companies that process a large volume of email often have their own, **private network clouds.**
- These organizations commonly have multiple mail servers, and route all email through a central gateway server (i.e., mail hub) that redistributes mail to whichever MTA is available.
- Email on these secondary MTAs must usually wait for the primary MTA (i.e., the designated host for that domain) to become available, at which time the secondary mail server will transfer its messages to the primary MTA.



Actual mail transfer is done through mail transfer agent

# ARCHITECTURE

To explain the architecture of email, we give four scenarios. We begin with the simplest situation and add complexity as we proceed. The fourth scenario is the most common in the exchange of email.

*The topics discussed in this section include:*

*First Scenario*

*Second Scenario*

*Third Scenario*

*Fourth Scenario*

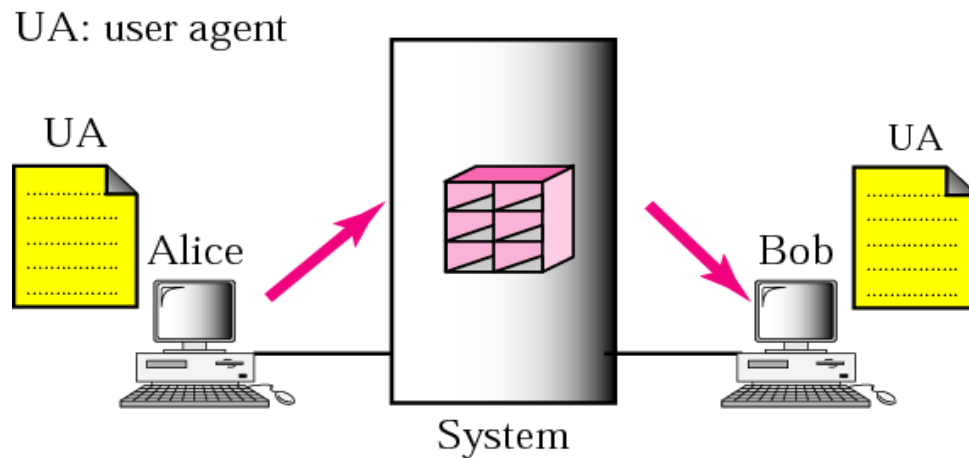


*When the sender and the receiver of an email are on the same system, we need only two user agents.*

# FIRST SCENARIO

- In the first scenario, the sender and the receiver of the e-mail are users (or application programs) on the same system;
- they are directly connected to a shared system.
- The administrator has created one mailbox for each user where the received messages are stored.
- A *mailbox* is part of a local hard drive, a special file with permission restrictions . Only the owner of the mailbox has access to it.

## *First scenario*



When the sender and the receiver of an e-mail are on the same system, we need only two user agents



*When the sender and the receiver of an email are on different systems, we need two UAs and a pair of MTAs (client and server).*

# SECOND SCENARIO

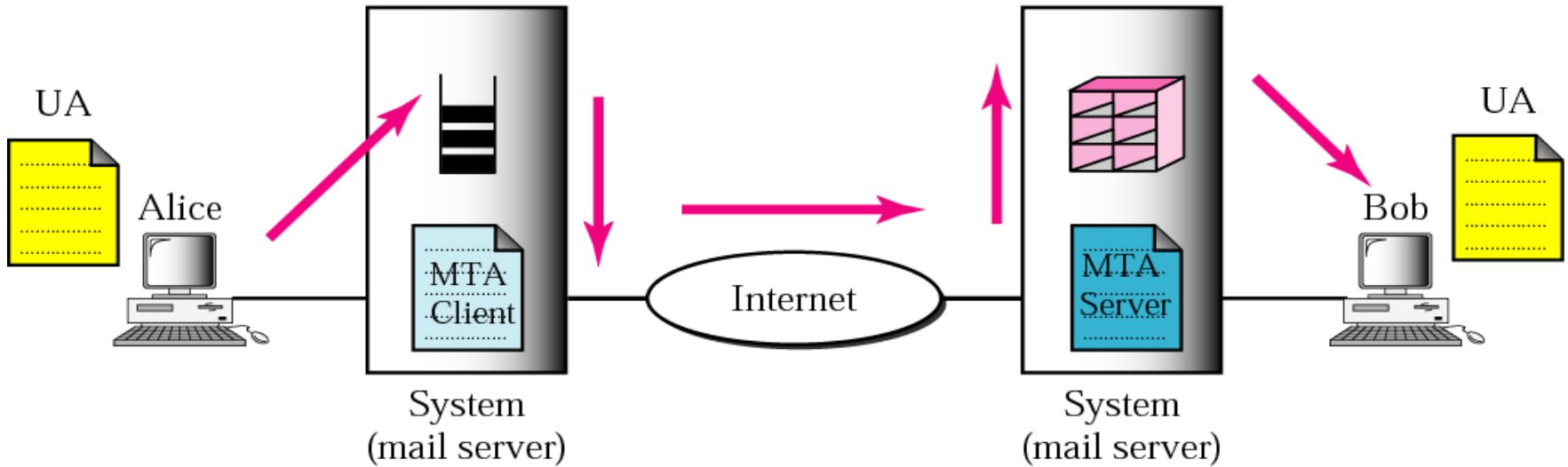
- In the second scenario, the sender and the receiver of the e-mail are users on two different systems .
- The message needs to be sent over the Internet. Here we need user agents (VAs) and message transfer agents (MTAs)



## Second scenario

UA: user agent

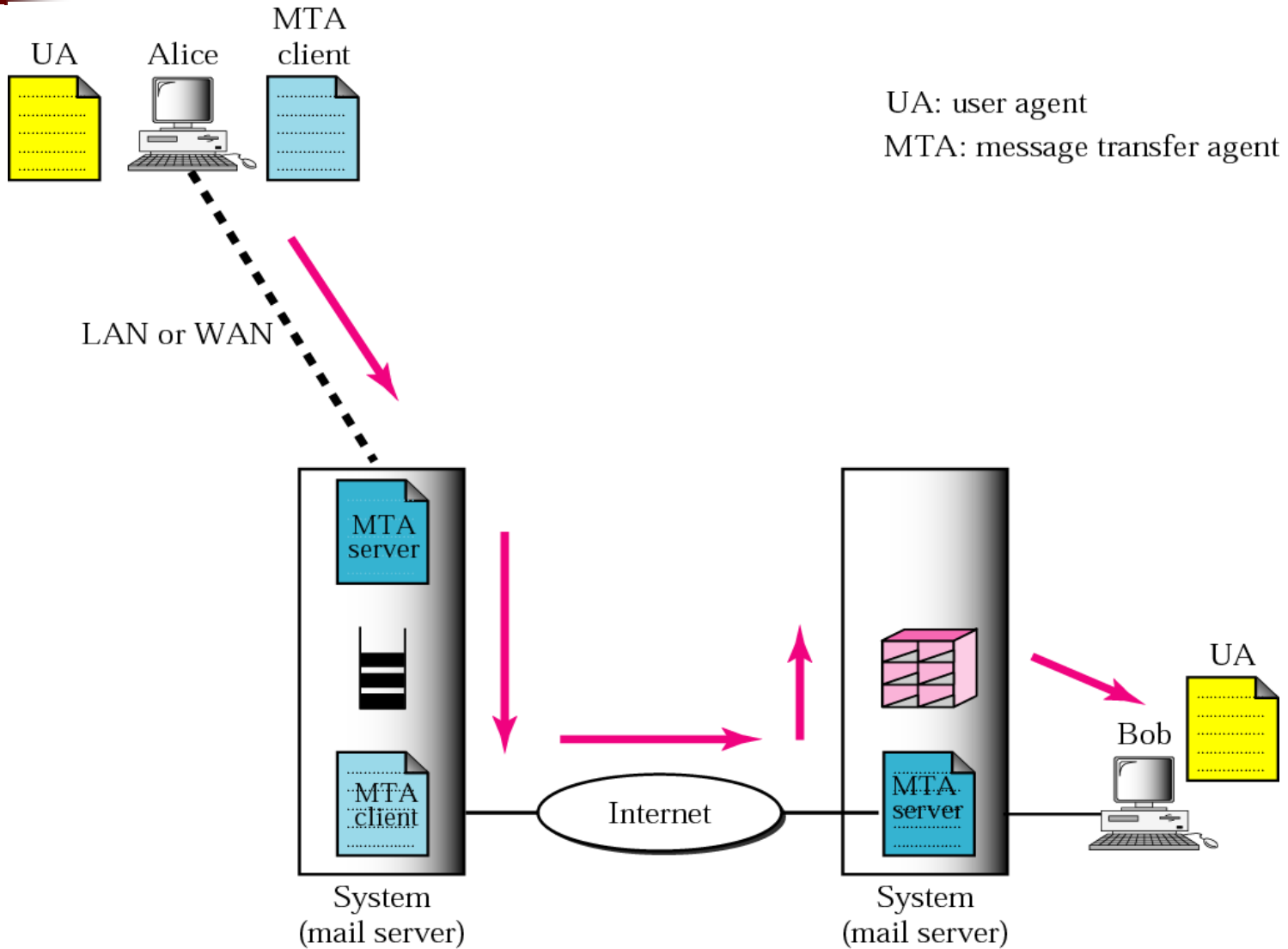
MTA: message transfer agent





*When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server).*

# Third scenario



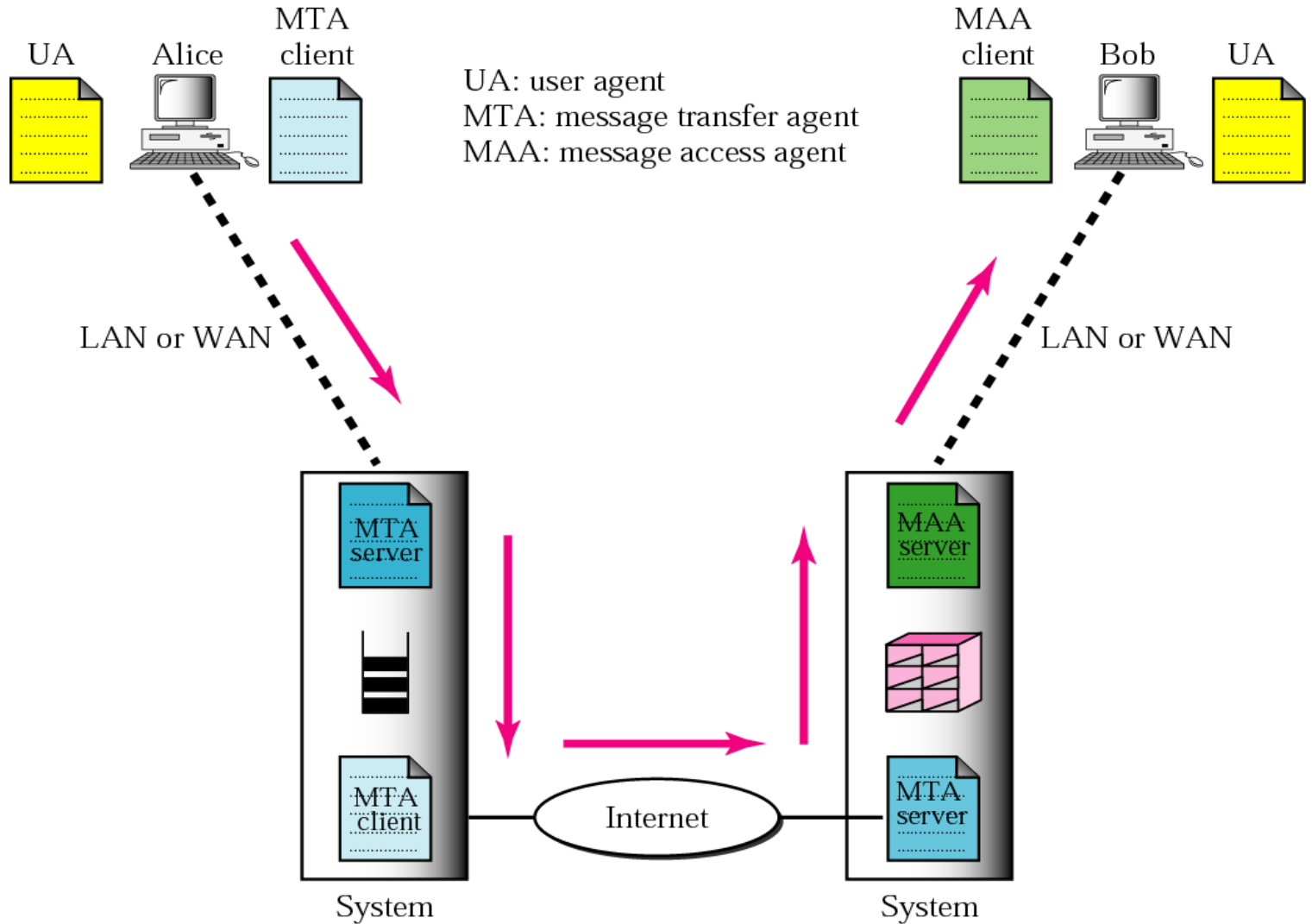
# THIRD SCENARIO

- In the third scenario B is directly connected to his system A is connected to the system via a point-to-point WAN, such as a dial-up modem, a DSL, or a cable modem; or is connected to a LAN in an organization that uses one mail server for handling e-mails-all users need to send their messages to this mail server.
- A still needs a user agent to prepare message. And needs to send the message through the LAN or WAN. Whenever A has a message to send, calls the user agent which, in tern, calls the MTA client. The MTA client establishes a connection with the MTA server on the system, which is running all the time.



*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.*

# Fourth scenario



## **At the A's side**

- When we want to send the email we create a message using user agent.
- When we send an email, the email client finds the mail server using the domain name system and contacts it using the simple mail transfer protocol
- Once the mail server has been contacted, the message is sent and the email client then proceeds to send any further pending messages

## **At the B's side**

- To collect mail a similar procedure is followed
- Using the Post Office Protocol (POP), with the email client sending an account name & password to tell the mail server who's mail to collect before collecting waiting messages, which are then collected one at a time.

## *Format of an email*

Behrouz Forouzan  
De Anza College  
Cupertino, CA 96014

Sophia Fegan  
Com-Net  
Cupertino, CA 95014

Sophia Fegan  
Com-Net  
Cupertino, CA 95014  
Jan. 5, 2005

Subject: Network

Dear Ms. Fegan:  
We want to inform you that  
our network is working properly  
after the last repair.

Yours truly,  
Behrouz Forouzan

Mail From: forouzan@deanza.edu  
RCPT To: fegan@comnet.com

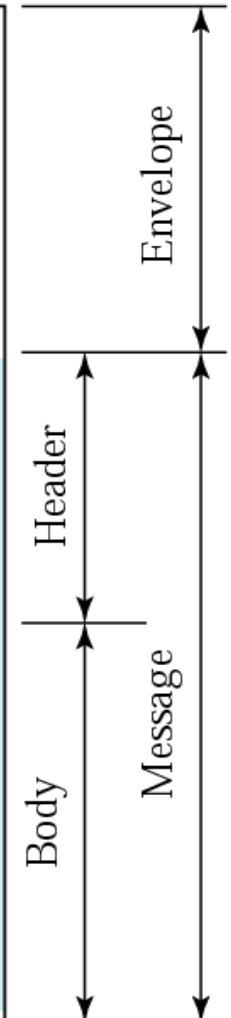
---

From: Behrouz Forouzan  
To: Sophia Fegan  
Date: 1/5/05  
Subject: Network

---

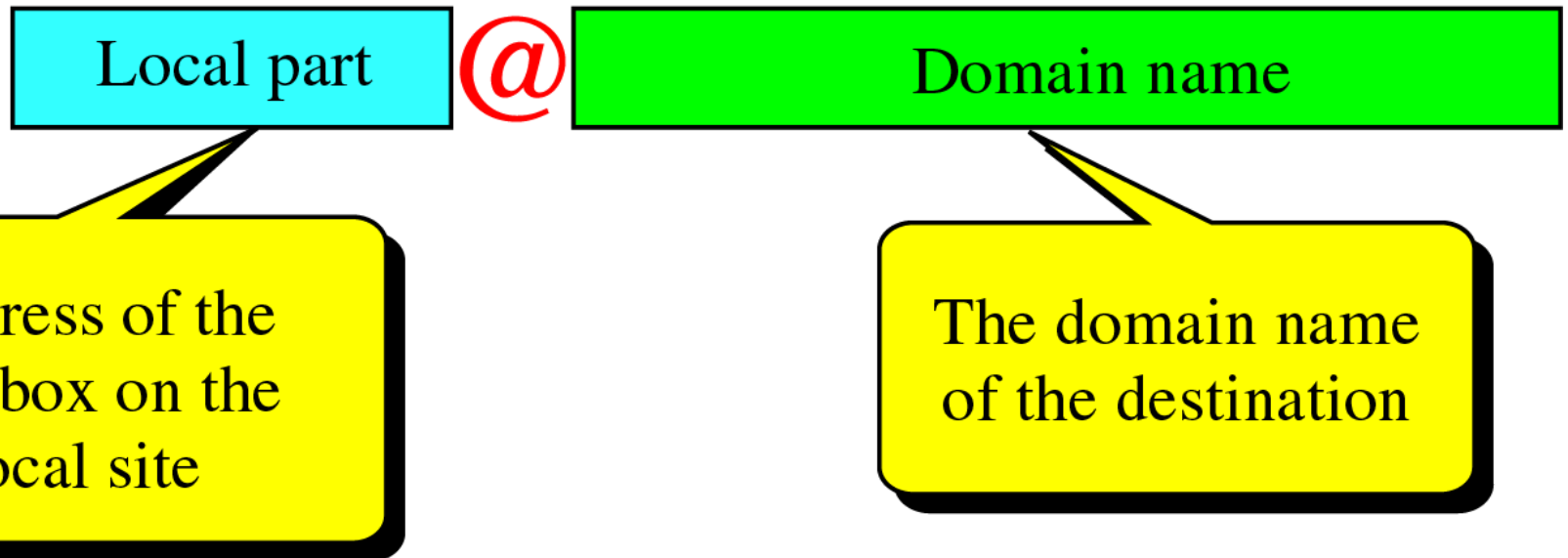
Dear Ms. Fegan:  
We want to inform you that  
our network is working properly  
after the last repair.

Yours truly,  
Behrouz Forouzan





# E-mail Address



# USER AGENT

*The user agent (UA) provides service to the user to make the process of sending and receiving a message easier.*

*The topics discussed in this section include:*

*Services Provided by a User Agent*

*User Agent Types*

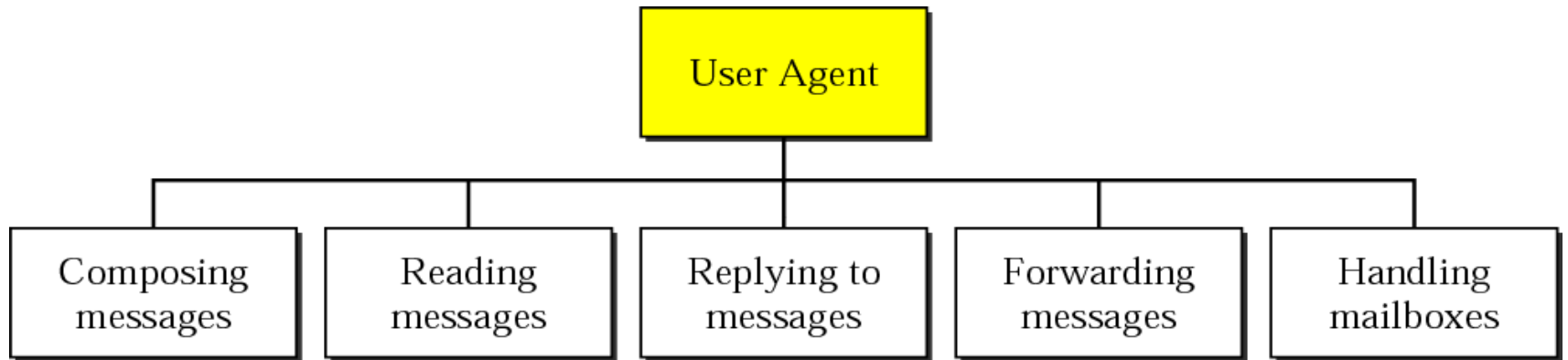
*Sending Mail*

*Receiving Mail*

*Addresses*

*Mailing List*

*MIME*





*Some examples of command-driven user agents are **mail**, **pine**, and **elm***

**Pine** :Program for Internet News and E-mail they provide License to mail server organizations .  
**Elm** : Electronic mail . its basically written in C language.



*Some examples of GUI-based user agents are **Eudora**, **Outlook**, and **Netscape**.*

# Applications

- TELNET is used in remote login and remote computing
- Email is used for formal and informal communication.