Domain Name System (DNS)

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

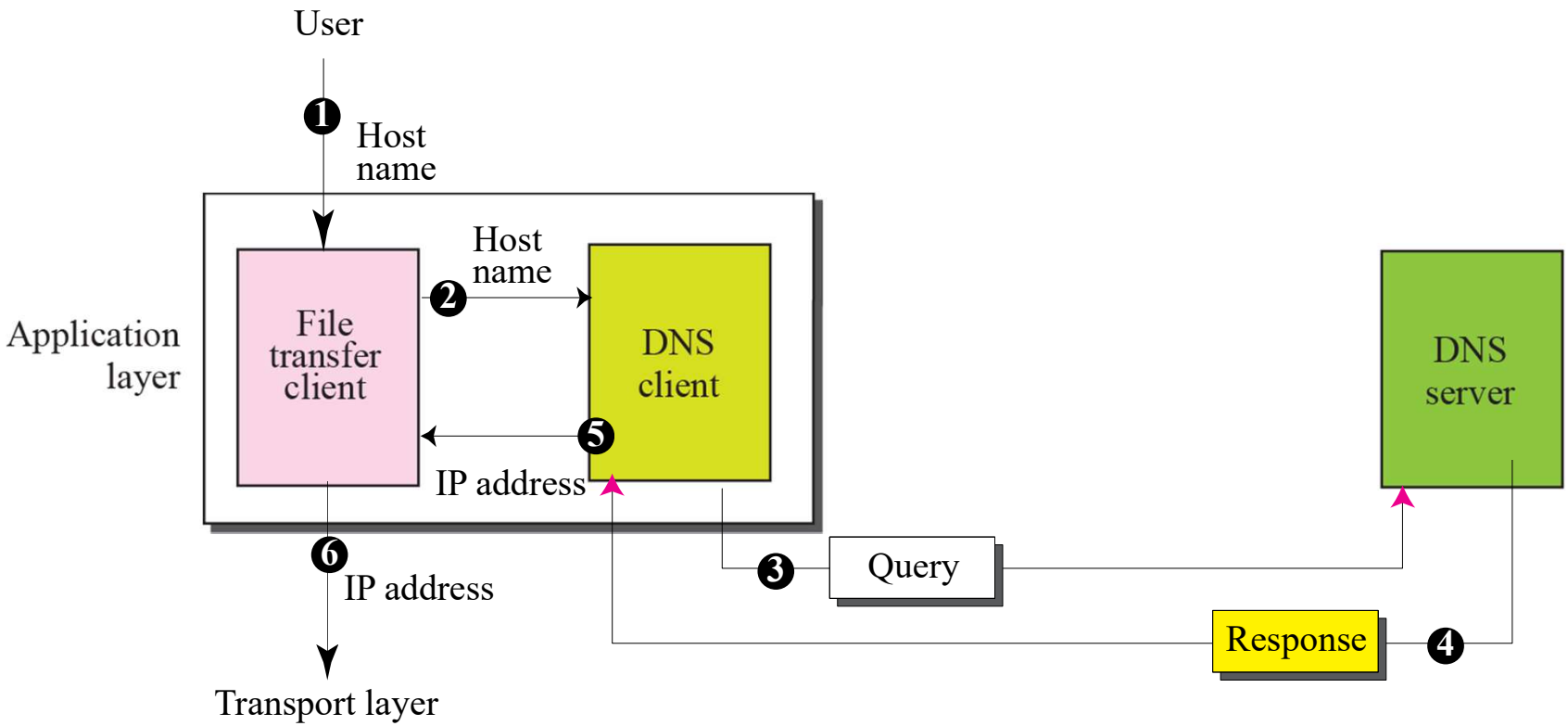
NEED FOR DNS

- To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet.
- However, people prefer to use names instead of numeric addresses.
- Therefore, we need a system that can map a name to an address or an address to a name.

- When the Internet was small, mapping was done using **a host file**. The host file had **only two columns**: **name** and **address**. Every host could store the host file on its disk and update it periodically from a master host file.
- When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.
- Today, however, it **is impossible to have one single host file** to relate every address with a name and vice versa.
- The **host file would be too large to store** in every host. In addition, it would be **impossible to update all** the host files every time **there is a change**.

- One solution would be to store the **entire host file in a single computer and allow access to this centralized information to every computer that needs mapping**. But this would create a **huge amount of traffic on the Internet.**
- Another solution, the one used today, is to **divide this huge amount of information into smaller parts** and **store each part on a different computer**.
- In this method, the host that needs mapping can contact the closest computer (server) holding the needed information. This method is used by the **Domain Name System (DNS)**.

Purpose of DNS



- In above figure a user wants to use a file transfer client to access the corresponding file transfer server running on a remote host. **The user knows only the file transfer server name, such as amazon.com.**
- The following six steps map the host name to an IP address.

1. The user passes the host name to the file transfer client.
2. The file transfer client passes the host name to the DNS client.
3. Each computer, after being booted, knows the address of one DNS server. The DNS client sends a message to a DNS server with a query that gives the file transfer server name using the known IP address of the DNS server.
4. The DNS server responds with the IP address of the desired file transfer server.
5. The DNS client passes the IP address to the file transfer server.
6. The file transfer client now uses the received IP address to access the file transfer server.

NAME SPACE

- To be unambiguous, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.

- In other words, the names must be unique because the addresses are unique.

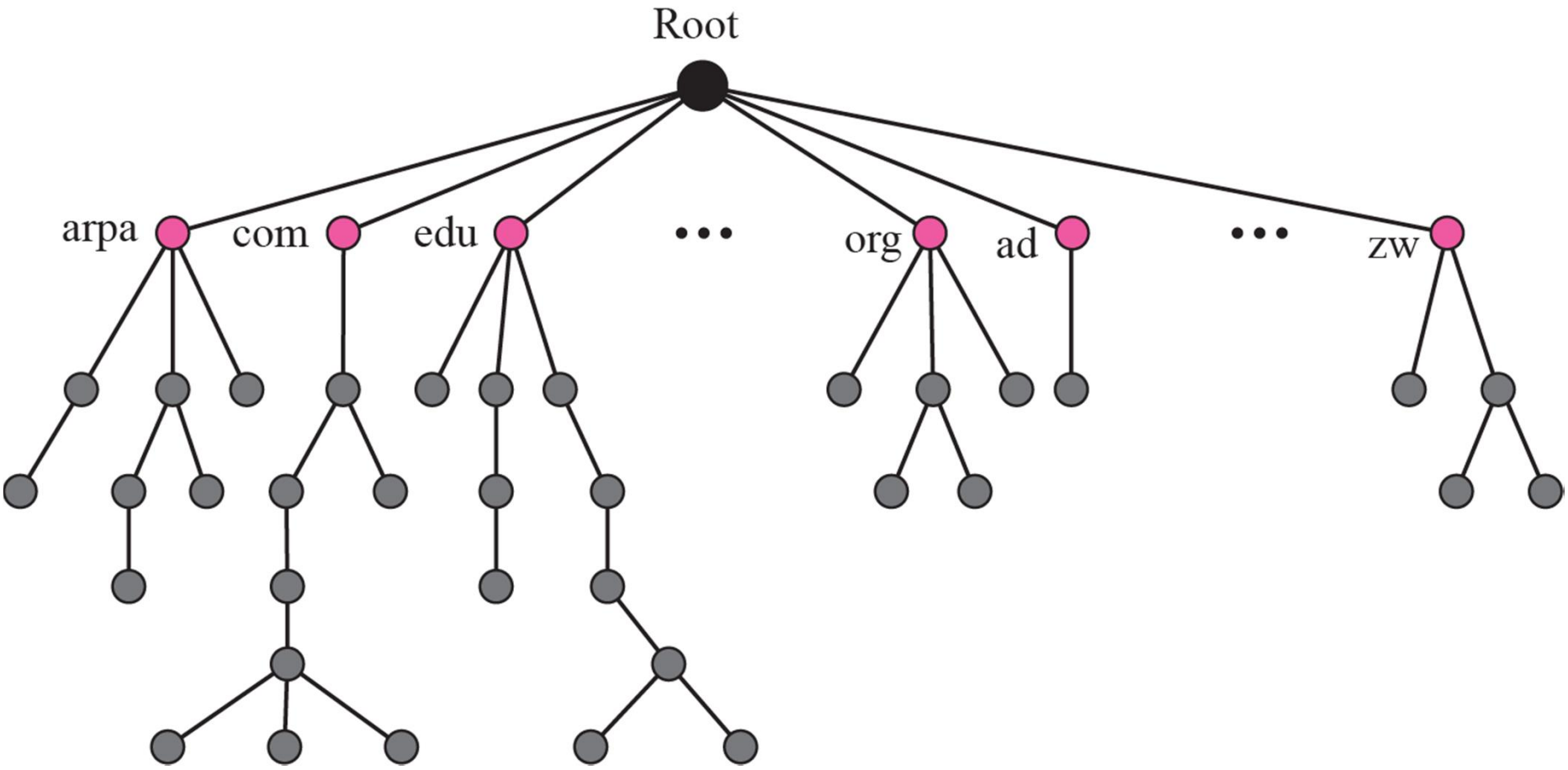
- A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

- In a **hierarchical name space**, each name is made of several parts.
- The **first part can define the nature of the organization**, the second part can **define the name of an organization**, the third part can define **departments in the organization**, and so on.
- In this case, the authority to assign and control the name spaces can be decentralized.
- A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.
- The organization can add suffixes (or prefixes) to the name to define its host or resources.

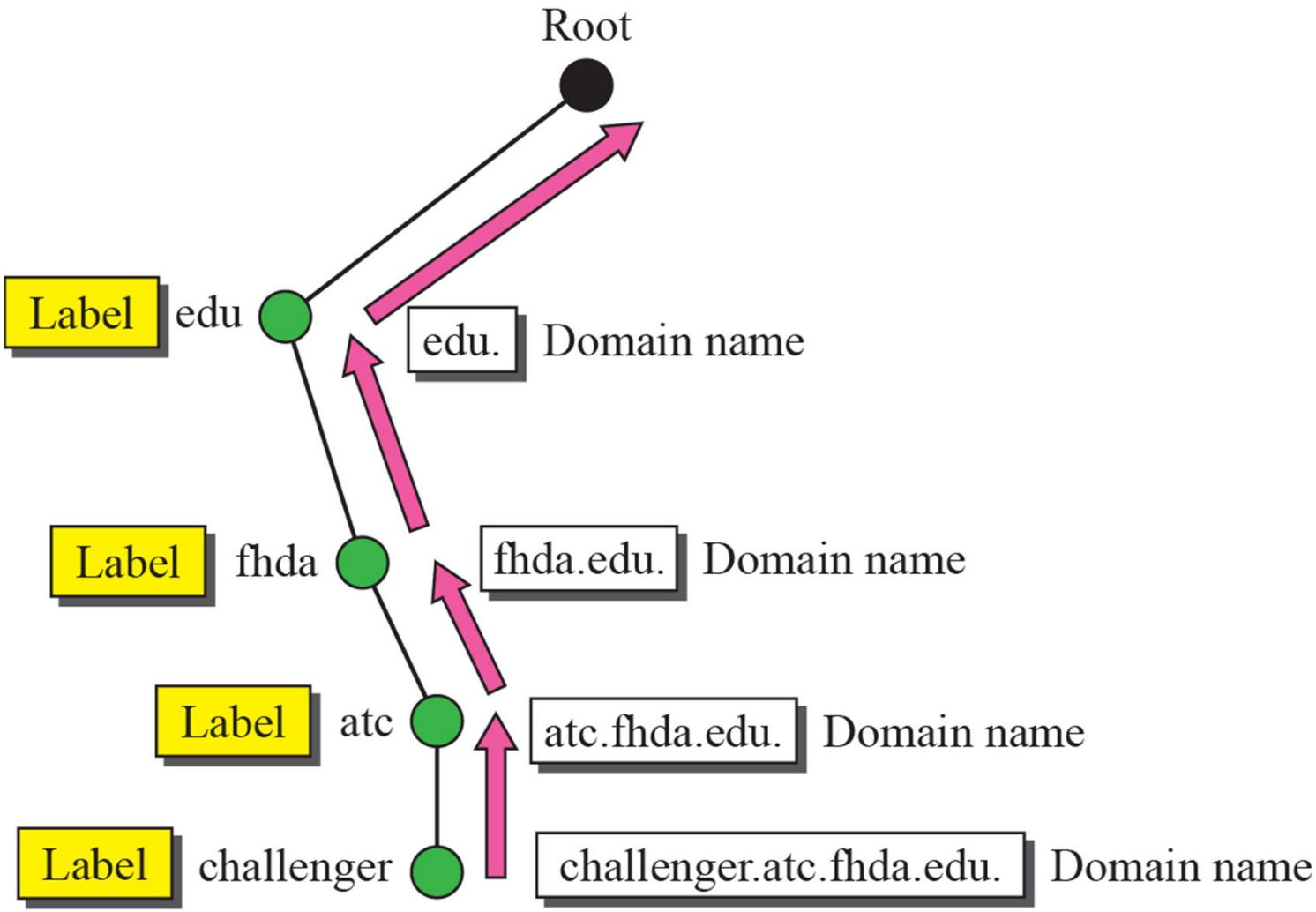
Domain Name Space

- To have a **hierarchical name space**, a domain name space was designed.
- In this design the names are defined in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels: level 0 (root) to level 127.
- Each node in the tree has a label, which is a string with a maximum of 63 characters.
- Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.). The domain names are always read from the node up to the root. The last label is the label of the root (null).

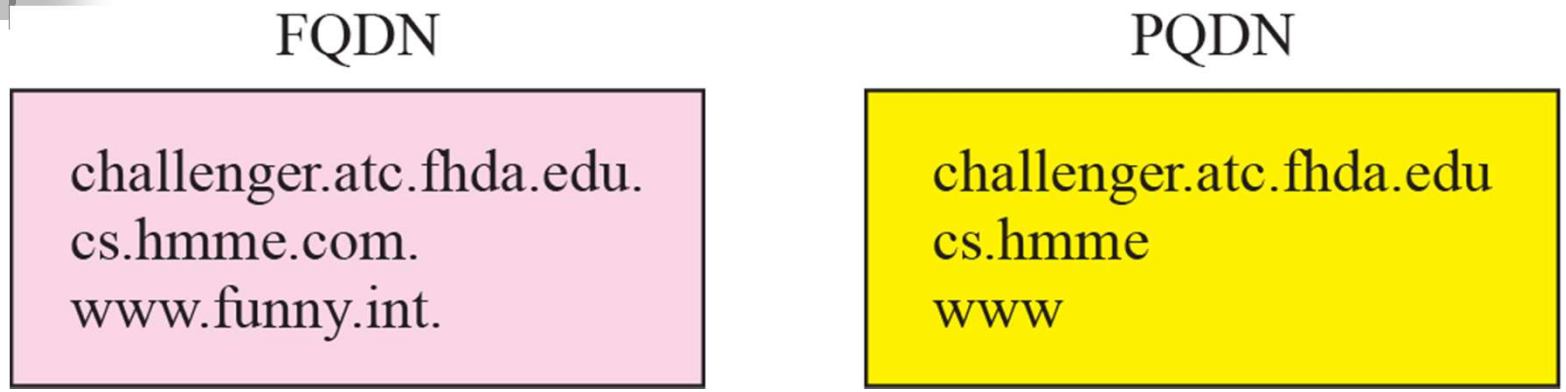
Domain name space



Domain names and labels



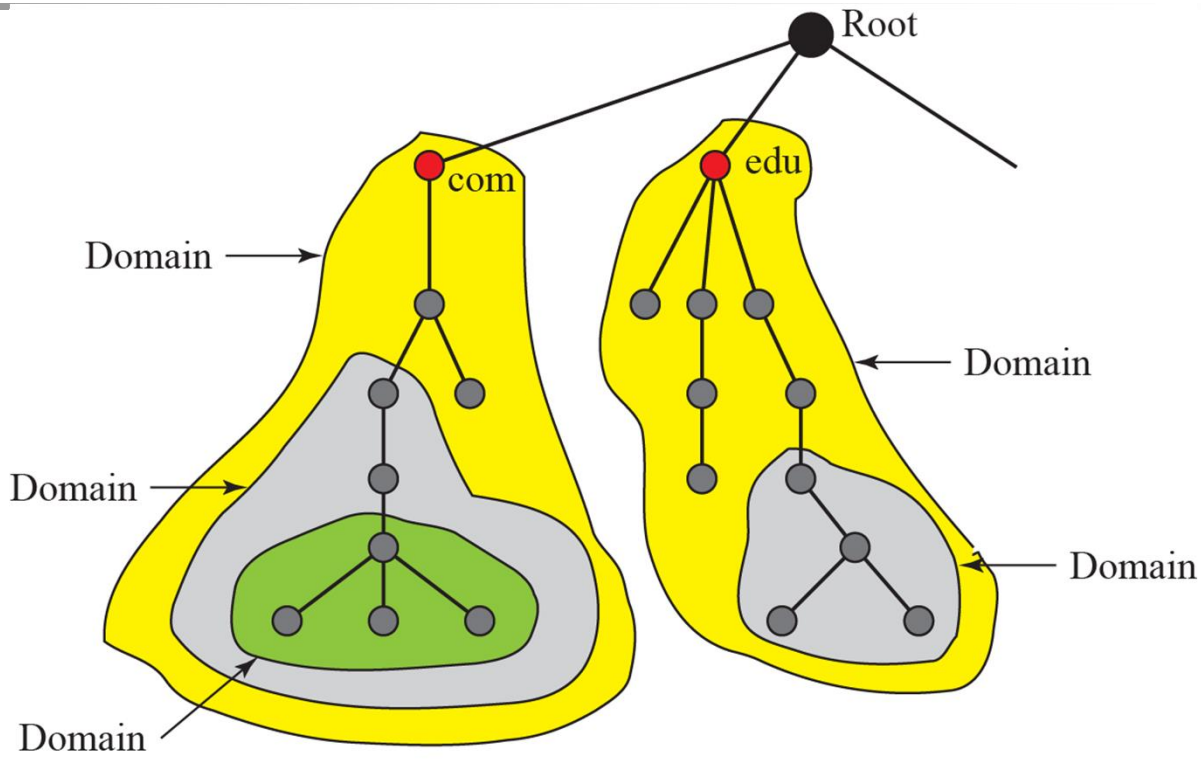
FQDN and PQDN



Fully Qualified Domain Name (FQDN) If a label is terminated by a null string, it is called a fully qualified domain name (FQDN). An FQDN is a domain name that contains the full name of a host.

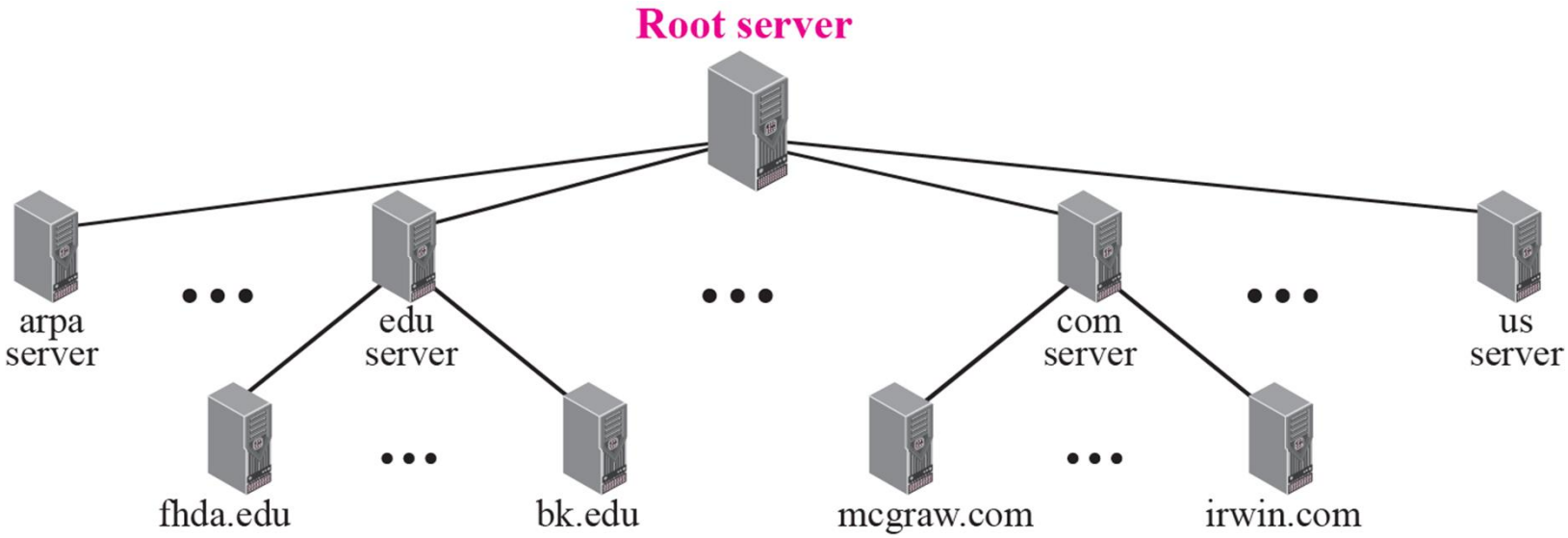
Partially Qualified Domain Name (PQDN) If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN).

Domains



A domain is a subtree of the domain name space.
The name of the domain is the name of the node at the top of the subtree.

Hierarchy of name servers



SNMP

Simple Network Management Protocol

Introduction

- SNMP – Simple Network Management Protocol
  - A set of standards for network management
    - Protocol
    - Database structure specification
    - Data objects
  - A set of standardized tools that
    - Control costs of network management
    - Across various product types
      - End system, bridges, routers, telecommunications, …
- History
  - In 1989
    - SNMP was adopted as TCP/IP-based Internet standards
  - In 1991
    - RMON – Remote network MONitoring
    - Supplement to SNMP to include management of LAN and LAN devices
  - In 1995
    - SNMPv2
      - Functional enhancements to SNMP
      - SNMP on OSI-based networks
    - RMON2
  - In 1998
    - SNMPv3
      - Further enhancements
      - Security capability for SNMP

Requirements of Network Management

- Fault Management
  - Detect, isolate, reconfigurate and repair the abnormal network environment
  - Problem tracking and control
    - Problem is truly resolved and no new ones are introduced
- Accounting Management
  - Track the use of network resources by end user to provide
    - Impropriate usage tracing, charging, statistics
- Configuration and Name Management
  - Startup, shutdown, reconfigure network component when
    - Upgrade, fault recovery or security checks
- Performance Management
  - Capacity utilization, throughput, response time, bottleneck
    - Collect information and assess current situation
- Security Management
  - Information protection and access control

Network Management System

- A collection of tools for
  - Network monitoring
  - Network control

- These tools must be integrated
  - Single operator interface with powerful but user-friendly
  - Support of managed equipments.

Network Management System

- Architecture of NMS
  - NMA
    - Operator interface
  - NME
    - Collect statistics
    - Response to NMA
    - Alert NMA when environment changing

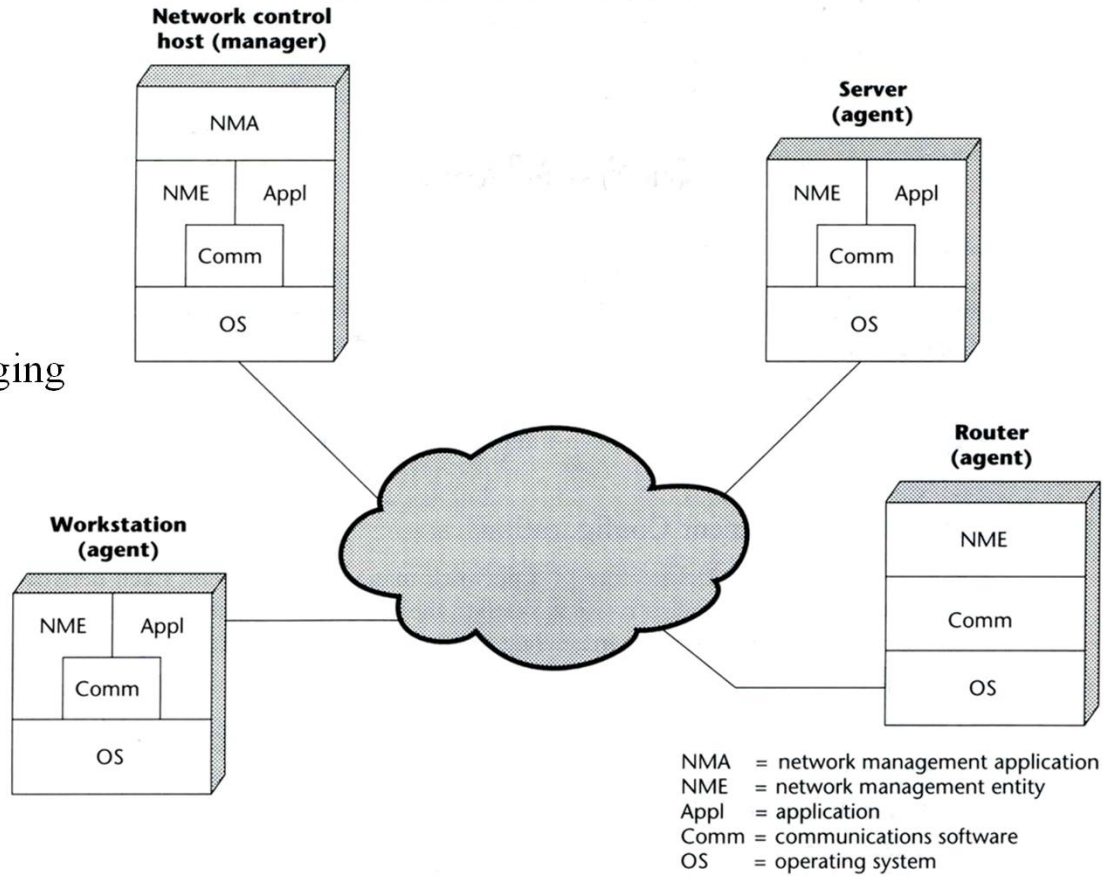
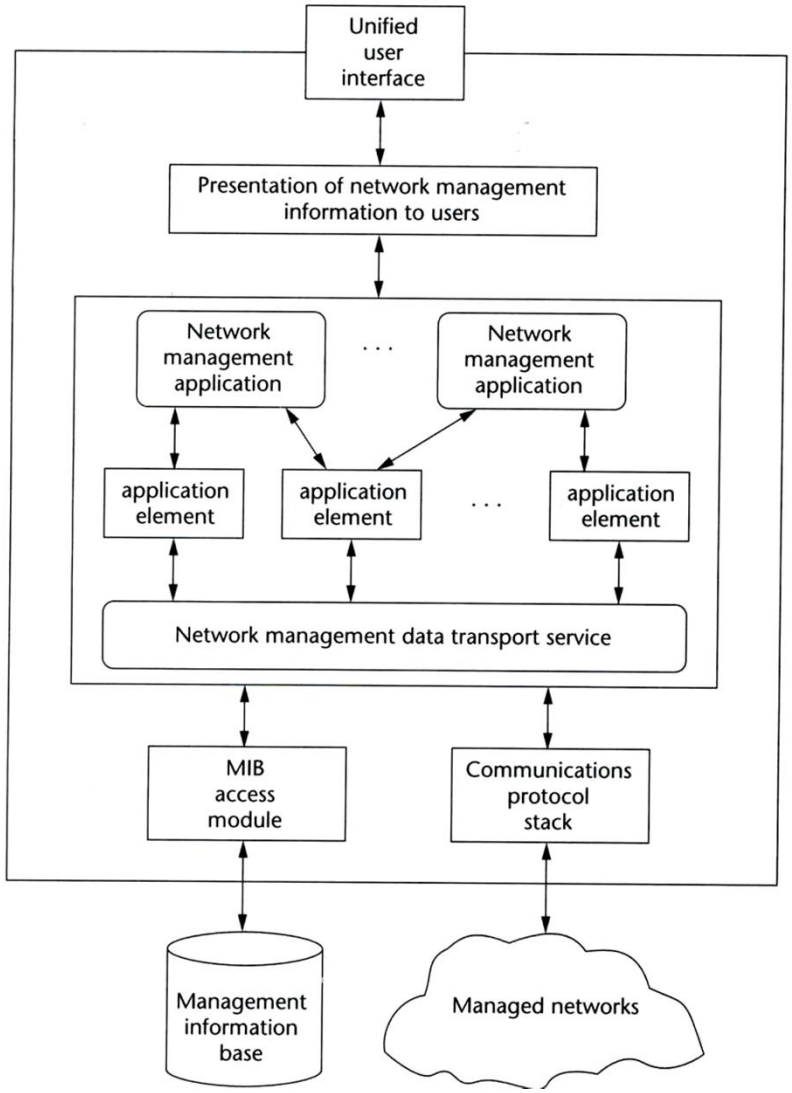


FIGURE 1.1 Elements of a network management system

Network Management Software

- Architecture
  - Presentation SW
    - Unified interface and handle information overload
  - Network Management SW
    - NM applications
      - Admin interested tools
      - Fault, security, accounting management
    - Application element
      - Primitive and general-purpose NM functions
      - Generating alarm, summarizing data
  - Communication SW
    - Exchange management information
    - Communication protocol stack
  - Database SW
    - MIB (Management Information Base)
      - Configuration and behavior
      - Operation parameters
    - MIB access modules
      - Convert local MIB to standard form

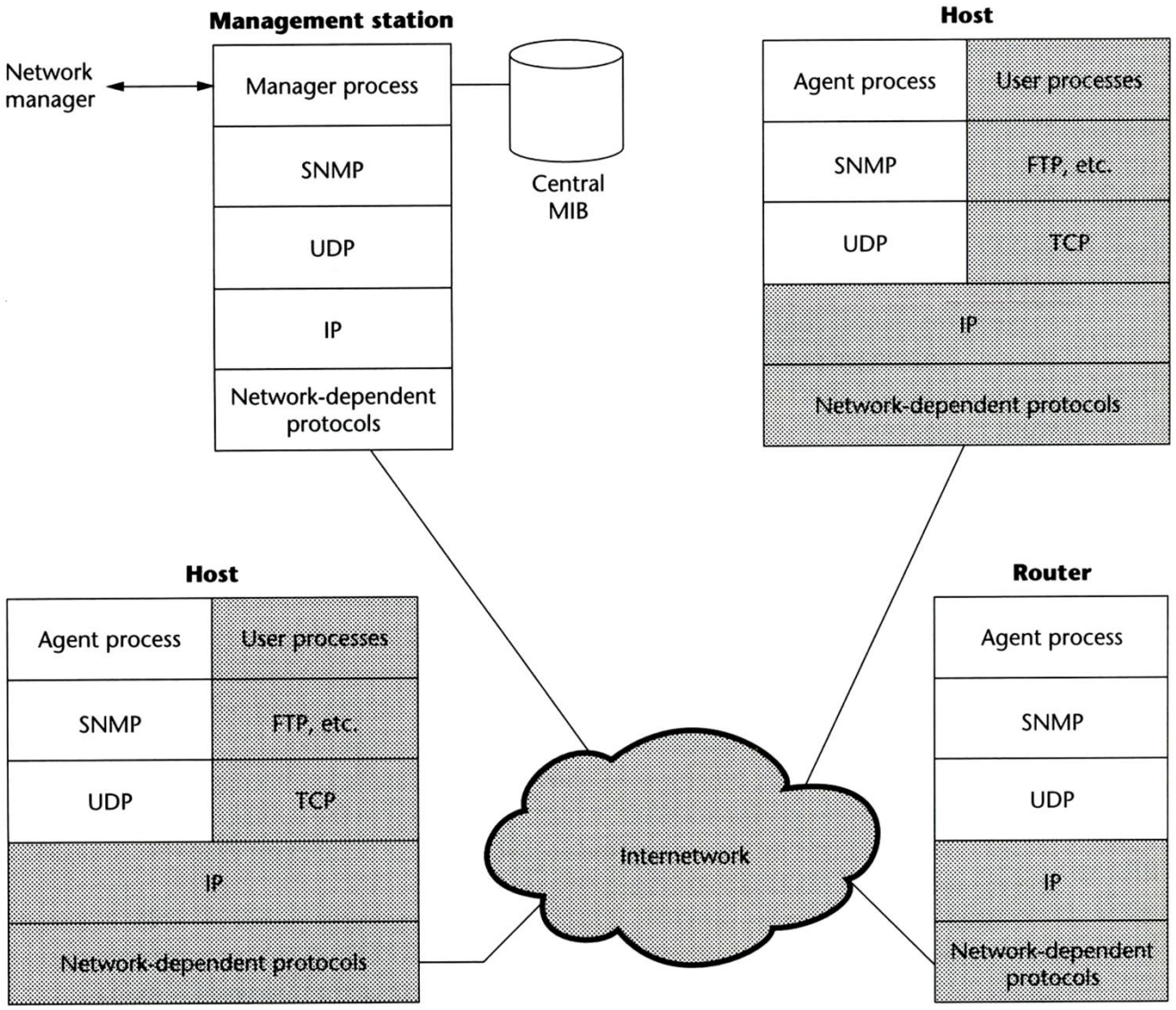


SNMP Network Management Concepts

Network Management Architecture in SNMP

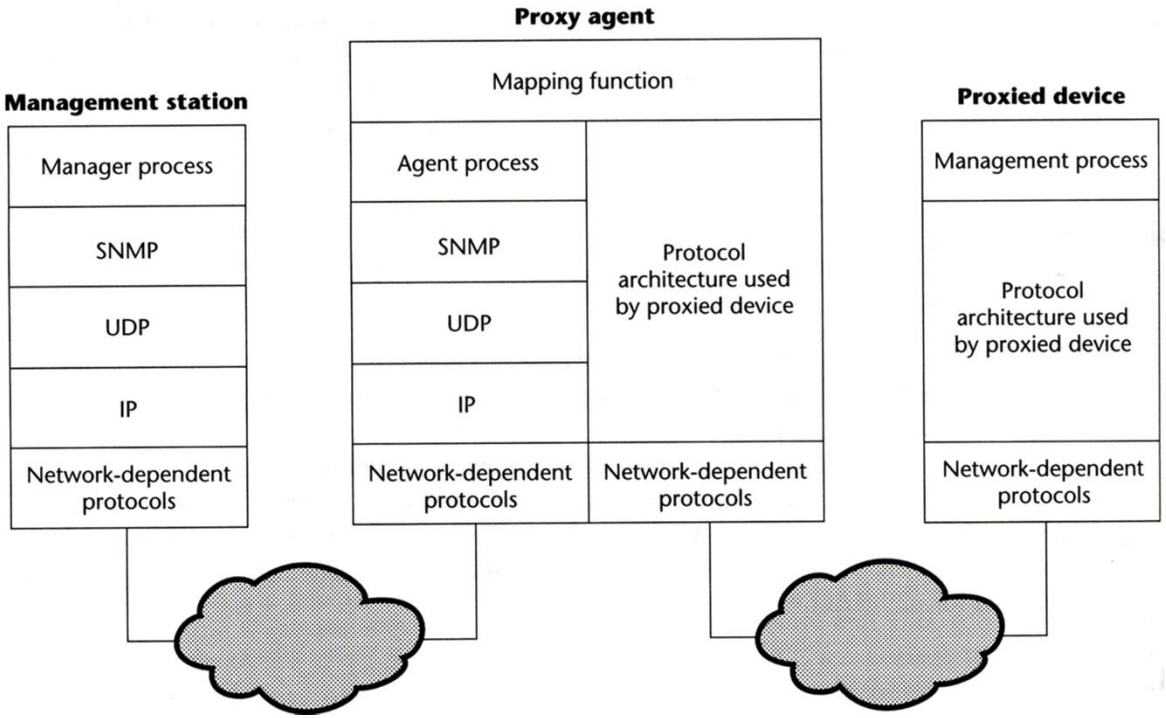
- 4 key elements
  - Management Station
    - Serve as the interface between manager and devices
      - Management applications
      - User-friendly interface
      - Translate manager's requirements into actual monitoring or control operations
      - Database extracted from MIBs of all managed device
  - Management Agent
    - Respond to request from management station
    - Change settings in MIB of managed device
    - Asynchronously report abnormal event (Trap)
  - Management Information Base (MIB)
    - Each resource is represented as an object and
    - MIB is a collection of objects
  - Network Management Protocol
    - get, set, trap

Network Management Architecture in SNMP



Network Management Architecture in SNMP

- SNMP proxy
  - Devices that do not support UDP/IP
    - ex: Bridge, Modem
  - Devices that do not want to add burden of SNMP agent
    - ex: PC, programmable controller



SNMP Message Information

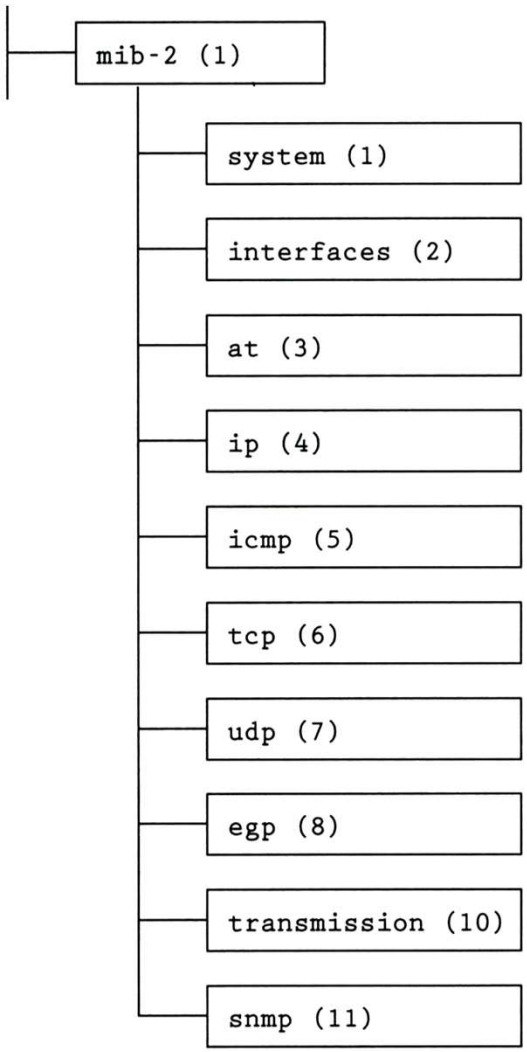
- Message Information Base (MIB)
  - Collection of objects and
  - Each object represents certain resource of managed device
- Interoperability of MIB
  - Object that represents a particular resource should be the same cross various system
    - What objects
    - MIB-I and MIB-II
  - Common representation format
    - SMI (Structure of Management Information)

SNMP Message Information – SMI

- MIB structure
  - Rooted tree
    - The leaves are the actual managed objects
    - Each object has an identifier (OBJECT IDENTIFIER)
      - Number with dot as delimiter
    - The internet node
      - iso -> org -> dod -> internet
      - object identifier of internet node: 1.3.6.1
    - Under internet node
      - directory :OSI X.500 directory
      - **mgmt: used for objects defined in IAB (Internet Activities Board)**
      - experimental: used for internet experiments
      - private: unilaterally usage

MIBs

- RFC1213
  - MIB-I (RFC 1156)
  - MIB-II is a superset of MIB-I with some additional objects and groups



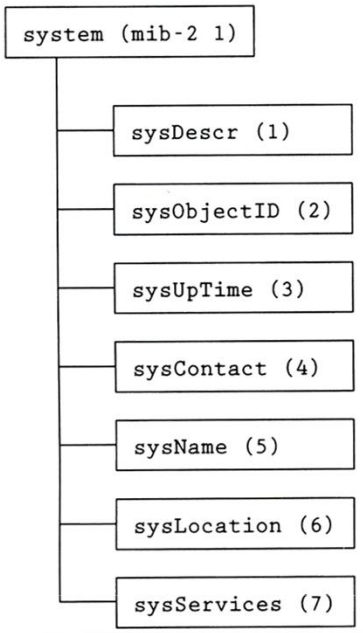
MIB-II

- First layer under mib-2
  - 1.3.6.1.2.1 (iso.org.dod.internet.mgmt.mib-2)

  - system
    - Overall information about the system
  - interfaces
    - Information about each interface
  - at
    - internet-to-subnet address mapping
  - ip, icmp, tcp, udp, egp
  - dot3
    - Transmission schemes and access protocol at each system interface
  - snmp

MIB-II

system group



- sysServices
  - 1 physical (ex: repeater)
  - 2 datalink/subnetwork (ex: bridge)
  - 3 internet (ex: router)
  - 4 end-to-end (ex: IP hosts)
  - 7 applications (ex: mail relays)

Object	Syntax	Access	Description
sysDescr	DisplayString (SIZE (0 . . . 255))	RO	A description of the entity, such as hardware, operating system, etc.
sysObjectID	OBJECT IDENTIFIER	RO	The vendor's authoritative identification of the network management subsystem contained in the entity
sysUpTime	TimeTicks	RO	The time since the network management portion of the system was last reinitialized
sysContact	DisplayString (SIZE (0 . . . 255))	RW	The identification and contact information of the contact person for this managed node
sysName	DisplayString (SIZE (0 . . . 255))	RW	An administratively assigned name for this managed node
sysLocation	DisplayString (SIZE (0 . . . 255))	RW	The physical location of this node
sysServices	INTEGER (0 . . . 127)	RO	A value that indicates the set of services this entity primarily offers

SNMP Protocol

- Supported operations
  - get, set, trap
- Simplicity vs. limitations
  - Not possible to change the structure of MIB by adding or deleting object instances
  - Access is provided only to leaf objects
    - Not possible to access entire table or row in single action

SNMP Protocol – security concern

- In management environment
  - The management station and managed agent
    - One-to-many relationship
    - One station may manage all or a subset of target
  - The managed station and management station
    - One-to-many relationship
    - Each managed agent controls its local MIB and must be able to control the use of that MIB
    - Three aspects
      - Authentication service
      - Access policy
      - Proxy service

SNMP Protocol – communities

- An SNMP community
  - A relationship between an SNMP agent and a set of SNMP managers that defines
    - Authentication, access control and proxy
  - The managed system establishes one community for each combination of authentication, access control and proxy
  - Each community has a unique “community name”
  - Management station use certain community name in all get and set operations

SNMP agent

- snmpconf
  - % man snmpd

  - System Information Setup
    - Location, contact, service
  - Access Control Setup
    - SNMPv3 or SNMPv1 access community
  - Trap Destination
    - Where to send the trap
  - Monitor Various Aspects of the Running Host
    - Process, disk space, load, file
  - Extending the Agent
    - Let snmp agent to return information that yourself define
  - Agent Operating Mode
    - User/group, IP port,...

SMTP, POP3, IMAP

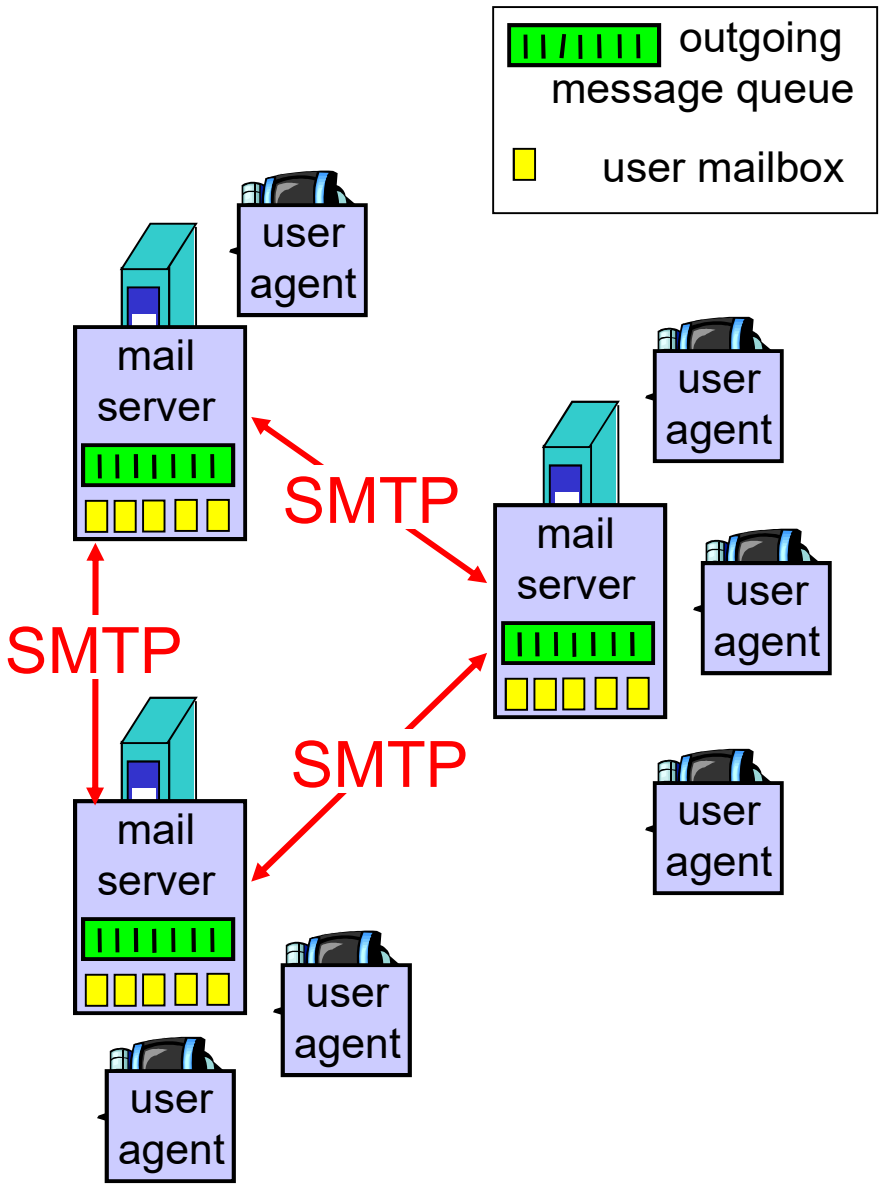
Electronic Mail

Three major components:

- user agents
- mail servers
- simple mail transfer protocol: SMTP

User Agent

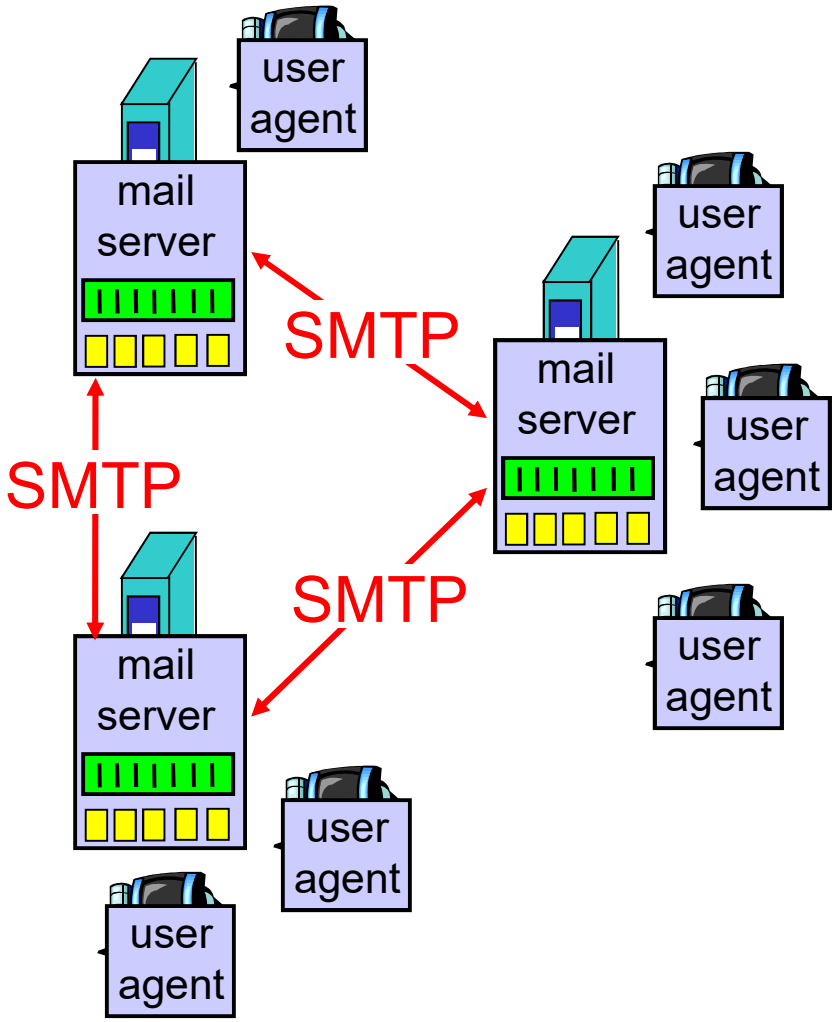
- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Eudora, Outlook, pine, mutt, Thunderbird
- outgoing, incoming messages stored on server



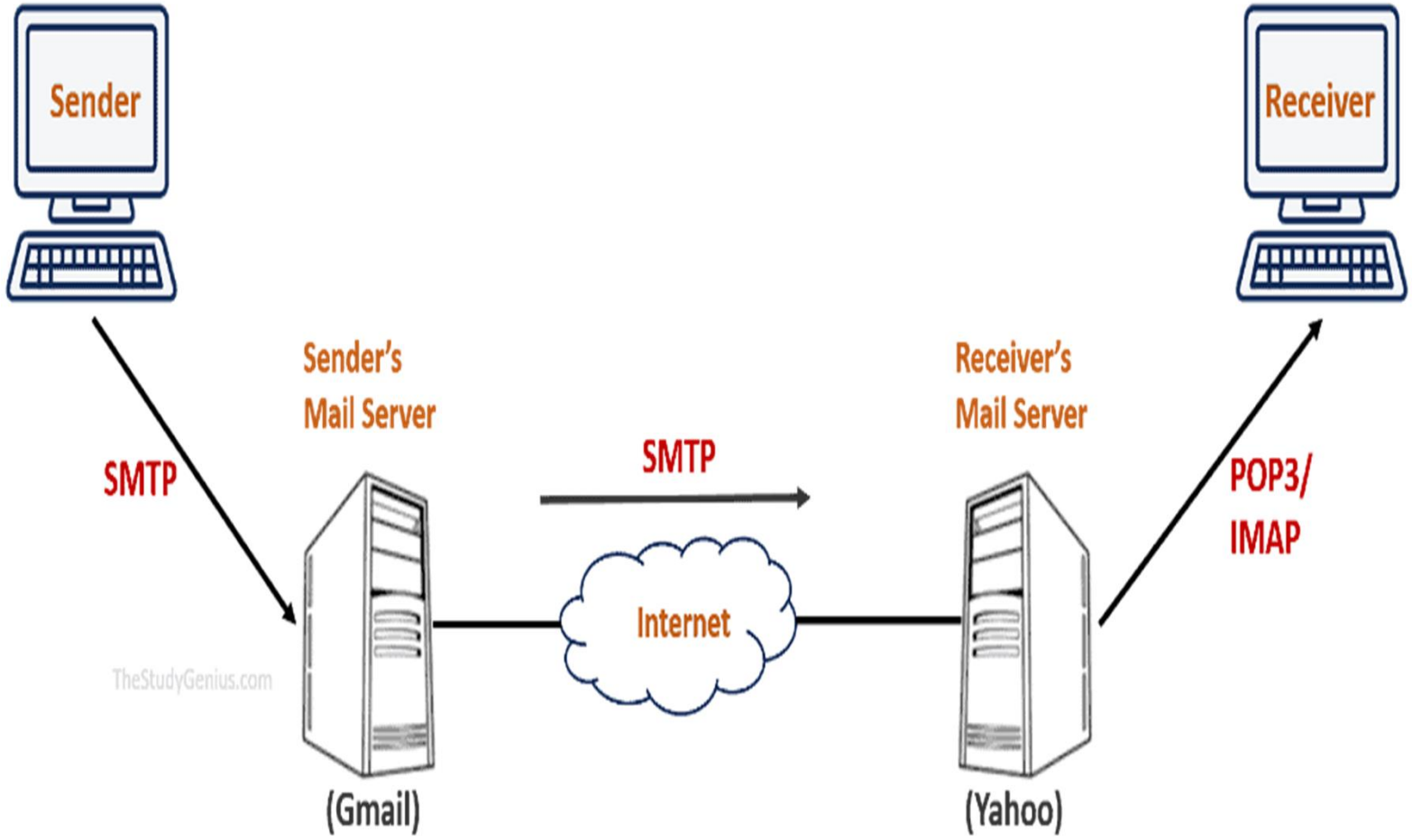
Electronic Mail: mail servers

Mail Servers

- mailbox contains incoming messages for user
- message queue of outgoing (to be sent) mail messages
- SMTP protocol between mail servers to send email messages
  - client: sending mail server
  - server: receiving mail server



Electronic Mail

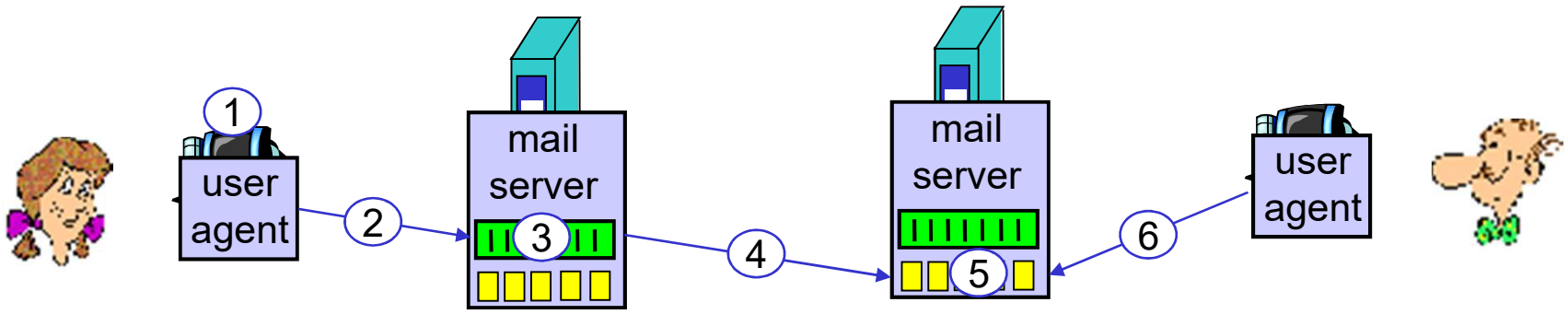


Electronic Mail: SMTP [RFC 2821]

- Uses TCP to reliably transfer email message from client to server, port 25
- Direct transfer: sending server to receiving server
- Three phases of transfer
  - greeting
  - transfer of messages
  - closure
- Command/response interaction
  - commands: ASCII text
  - response: status code and phrase
- Messages must be in 7-bit ASCII

Scenario: A sends message to B

1) A uses UA to compose message and "to" `b@notes.com`
2) A's UA sends message to her mail server; message placed in message queue
3) Client side of SMTP opens TCP connection with B's mail server
4) SMTP client sends A's message over the TCP connection
5) B's mail server places the message in B's mailbox
6) B invokes his user agent to read message



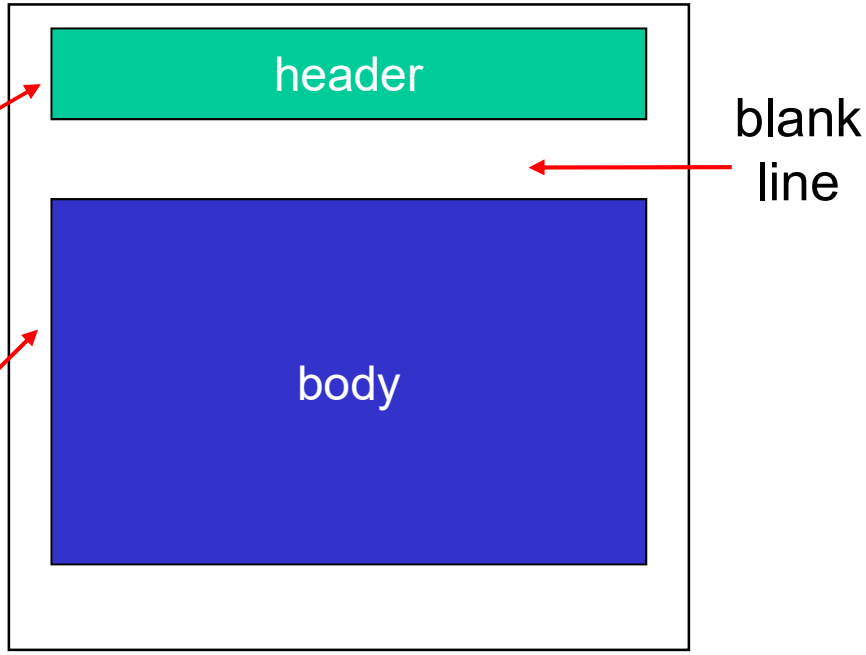
Mail message format

SMTP: protocol for exchanging email msgs

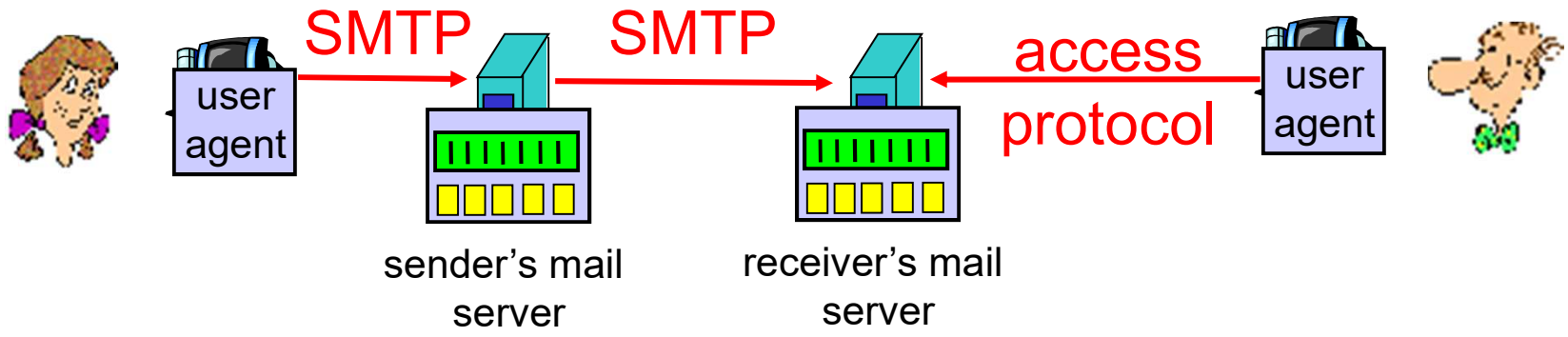
RFC 822: standard for text message format:

- header lines, e.g.,
  - To:
  - From:
  - Subject:

  different from SMTP commands!
- body
  - the "message", ASCII characters only



Mail access protocols



- SMTP: delivery/storage to receiver's server
- Mail access protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <-->server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
  - HTTP: Gmail , Yahoo! Mail, etc.

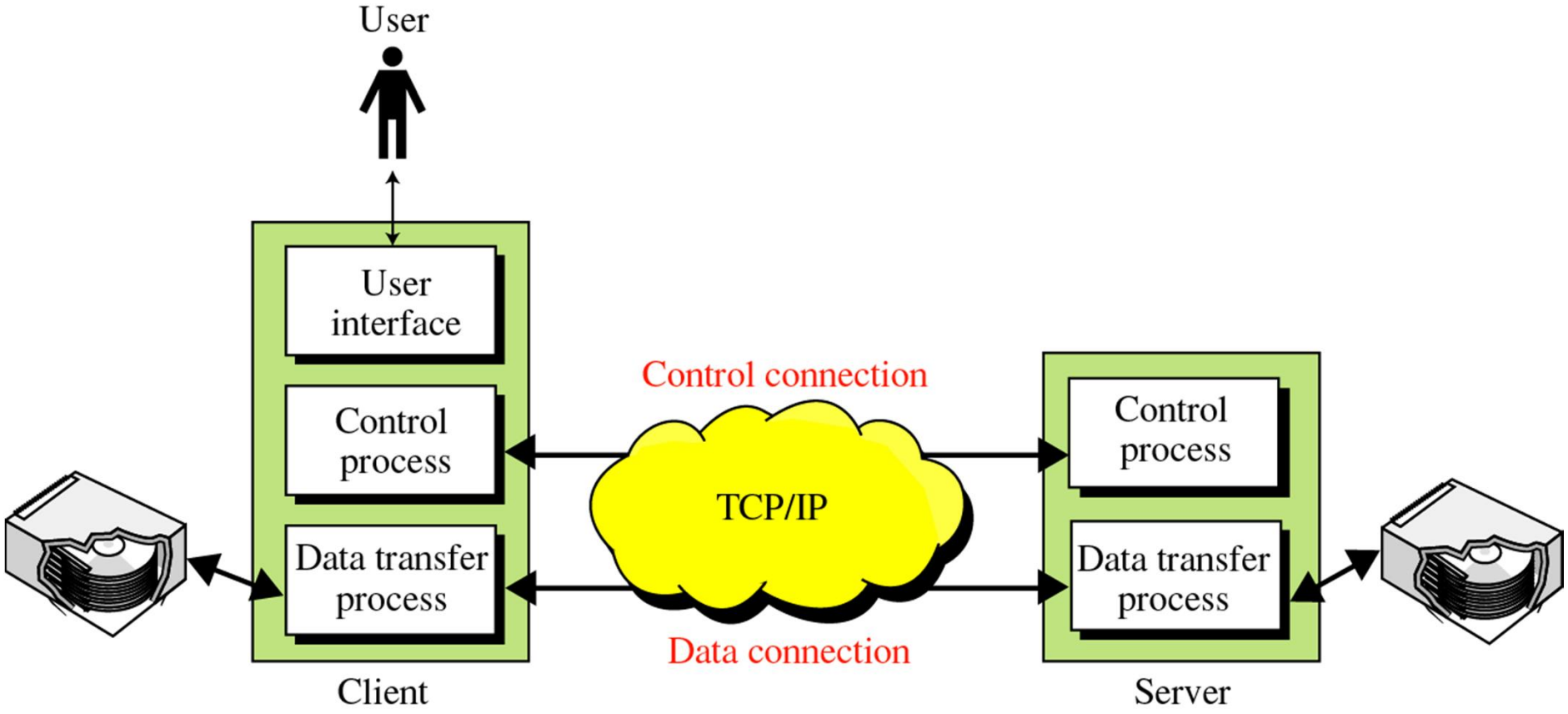
File Transfer Protocol (FTP)

CONTENTS

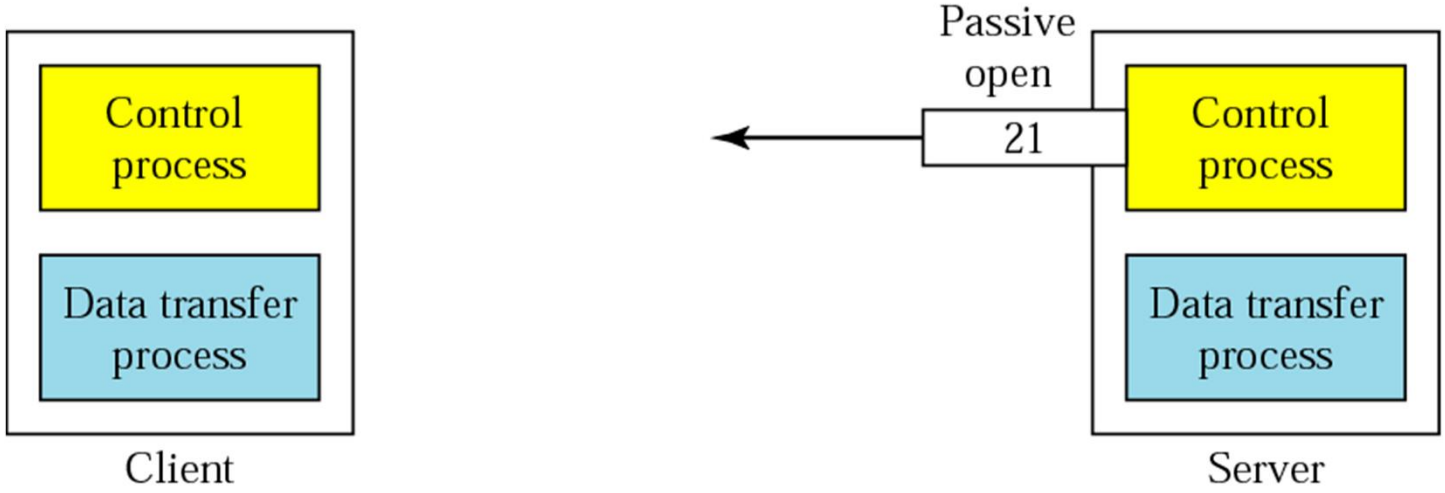
- **CONNECTIONS**
- **COMMUNICATION**
- **COMMAND PROCESSING**
- **FILE TRANSFER**

FTP uses the services of TCP. It needs two TCP connections. The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.

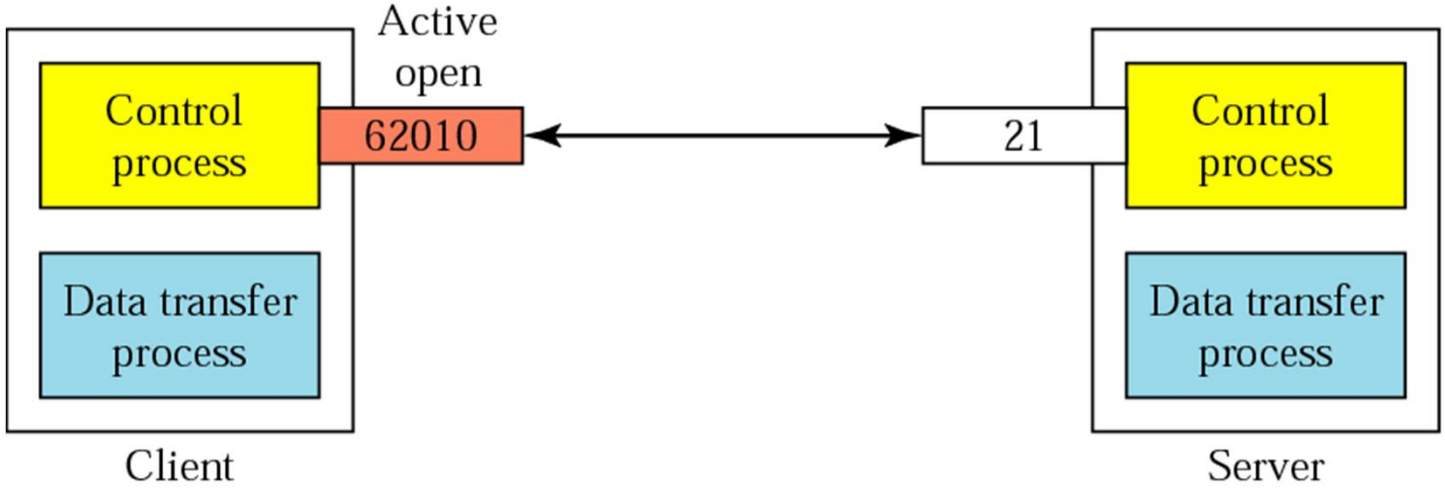
FTP


User
User interface
Control process
Data transfer process
Client
Control connection
TCP/IP
Data connection
Control process
Data transfer process
Server

Connections: The control connection

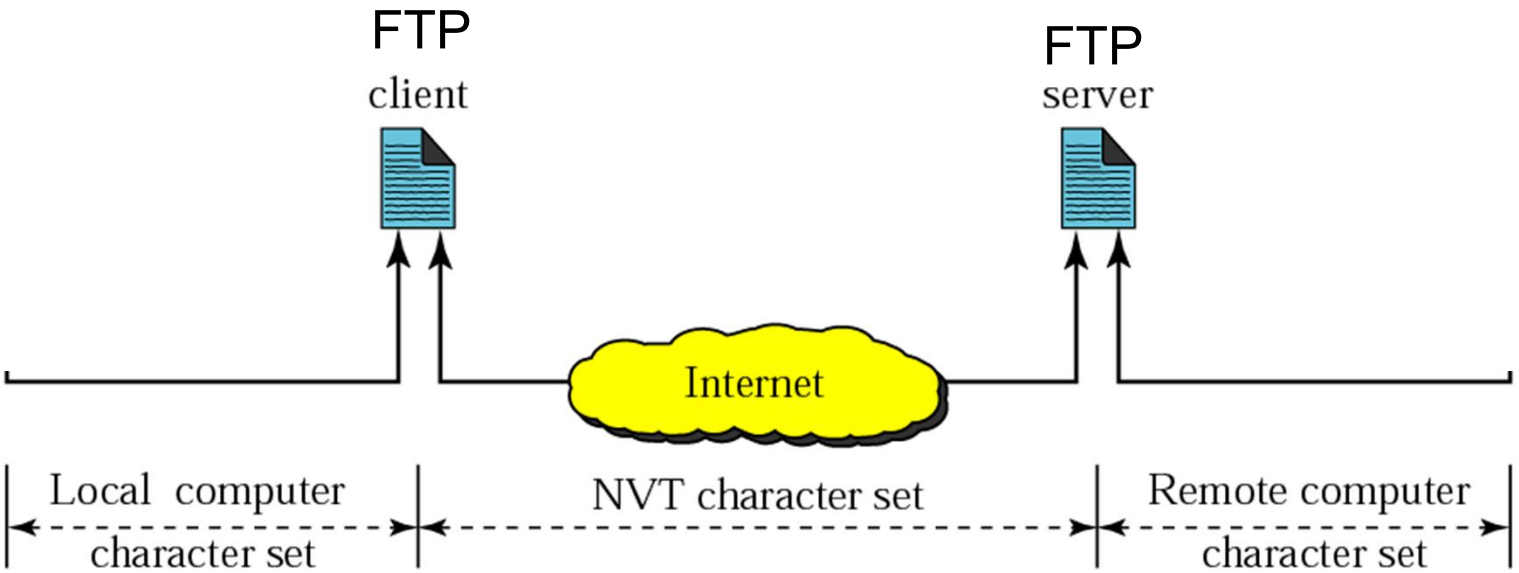


a. Passive open by server

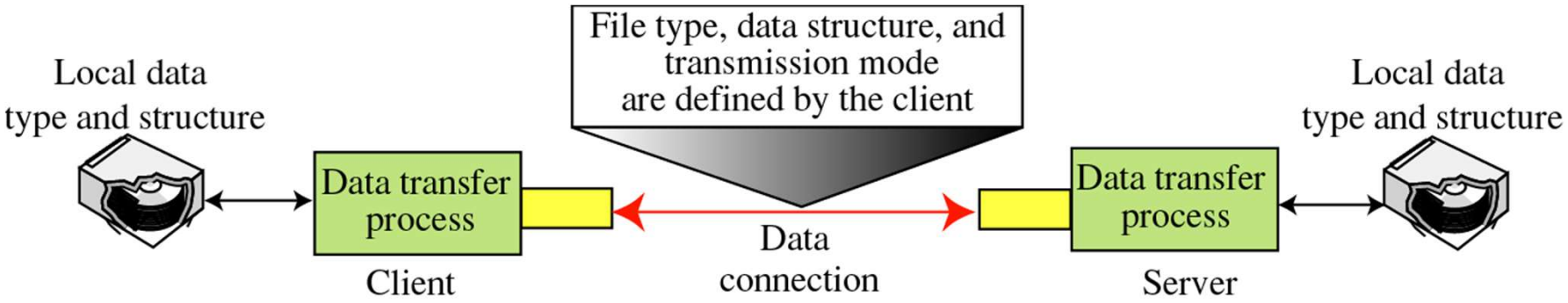


b. Active open by client

NVT


FTP
client
FTP
server
Internet
Local computer
character set
NVT character set
Remote computer
character set

Using the data connection



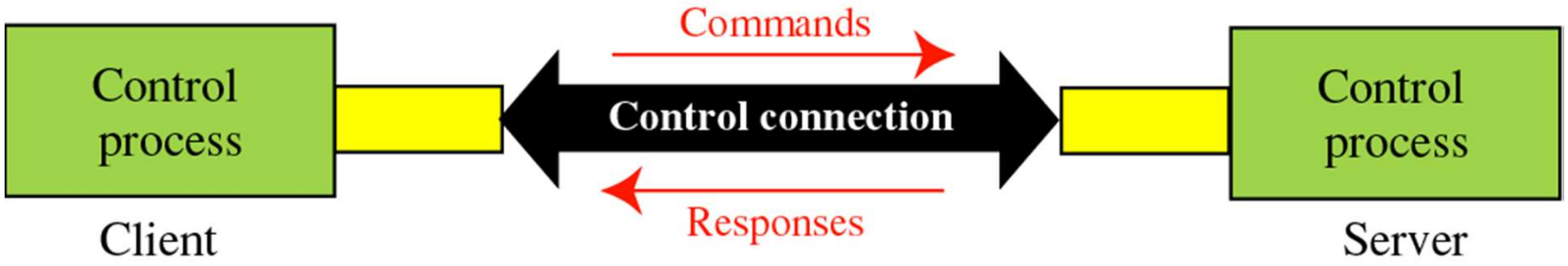
Data Structure

- File Structure
- Record Structure
- Page Structure

Transmission Mode

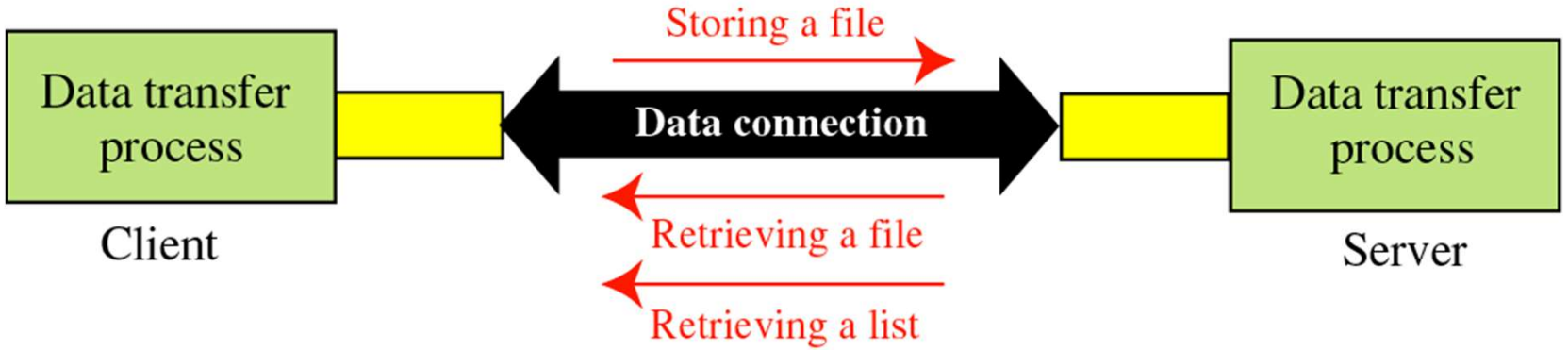
- Stream mode
- Block mode
- Compressed mode

Command processing

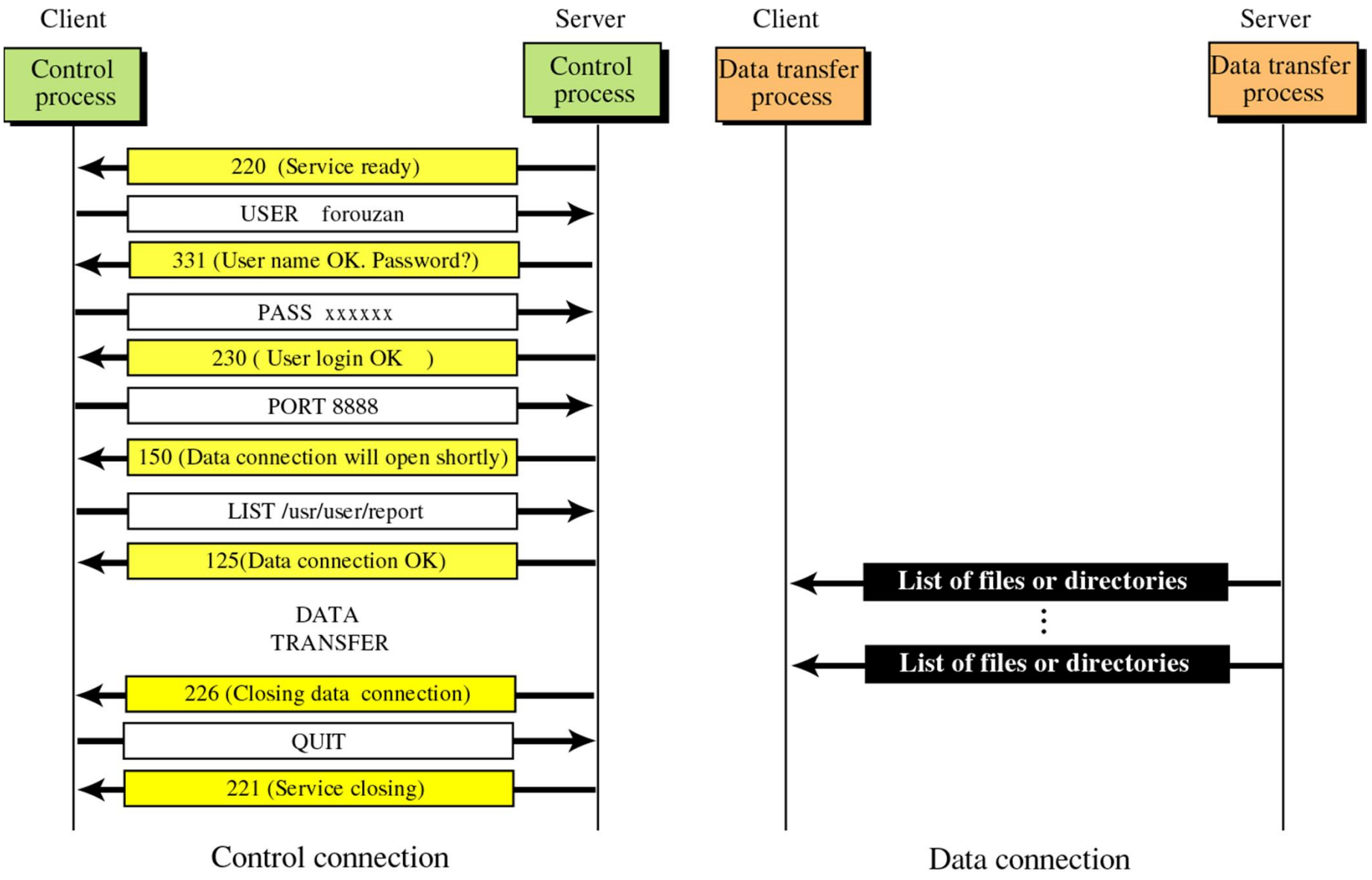


- Access Commands
- File Management
- Data Formatting
- Port defining
- File transfer
- Miscellaneous

File transfer



Example


Client
Server
Client
Server
Control process
Control process
Data transfer process
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
List of files or directories
List of files or directories
226 (Closing data connection)
QUIT
221 (Service closing)
Control connection
Data connection

TELNET

TELNET is an abbreviation for TErminaL NETwork. It is the standard TCP/IP protocol for virtual terminal service as proposed by ISO. 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.

OBJECTIVES:

- To introduce the TELNET protocol and show how it implements local and remote login.
- To discuss options and sub-options used in TELNET and how they are negotiated.
- To define out-of-band signaling in TELNET.
- To define different modes of operations in TELNET.
- To introduce SSH as an alternative to TELNET.

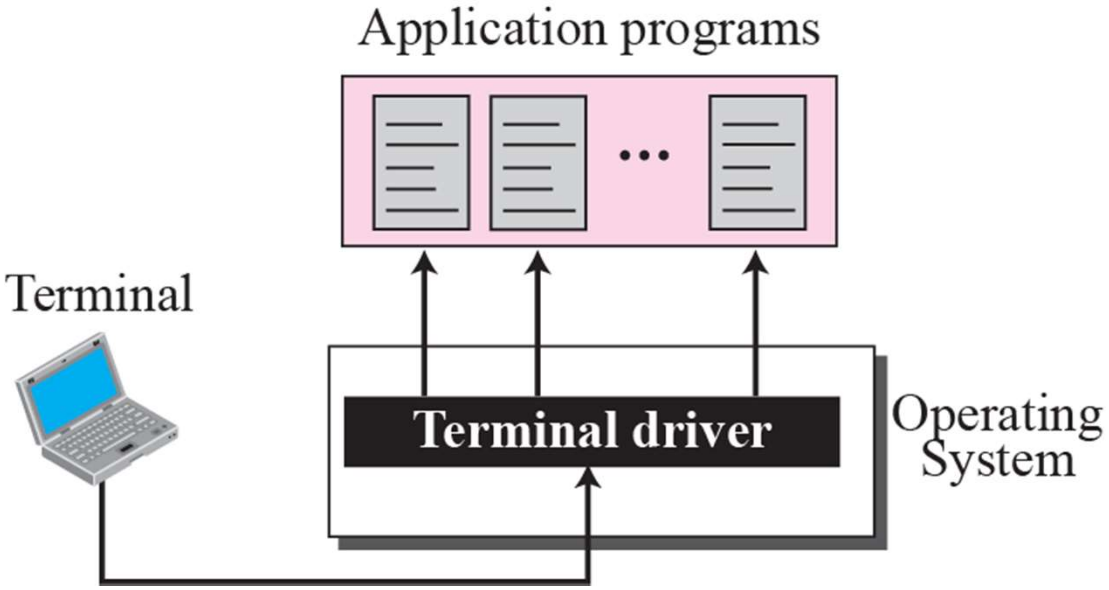
Topics Discussed in the Section

- ✓ Concepts
- ✓ Time-Sharing Environment
- ✓ Network Virtual Terminal (NVT)
- ✓ Embedding
- ✓ Options and Suboption Negotiation
- ✓ Controlling the Server
- ✓ Out-of-Band Signaling
- ✓ Escape Character
- ✓ Modes of Operation
- ✓ User Interface
- ✓ Security Issue

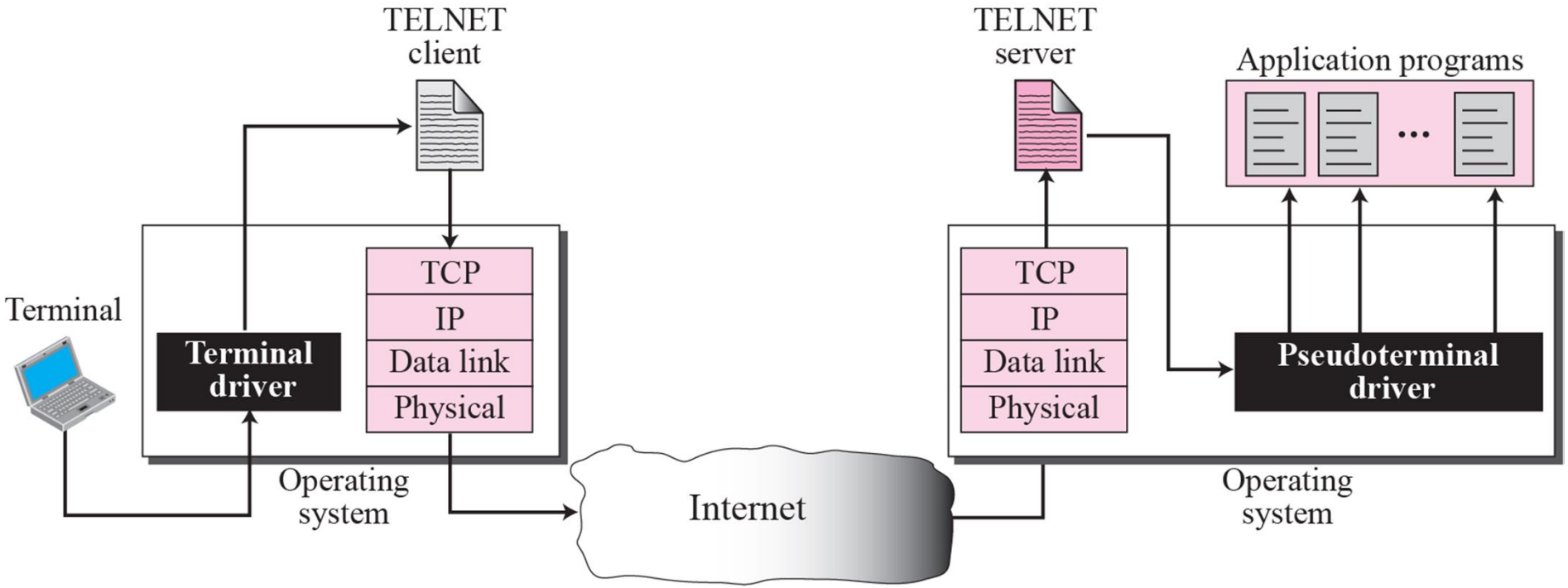
Note

TELNET is a general-purpose client-server application program.

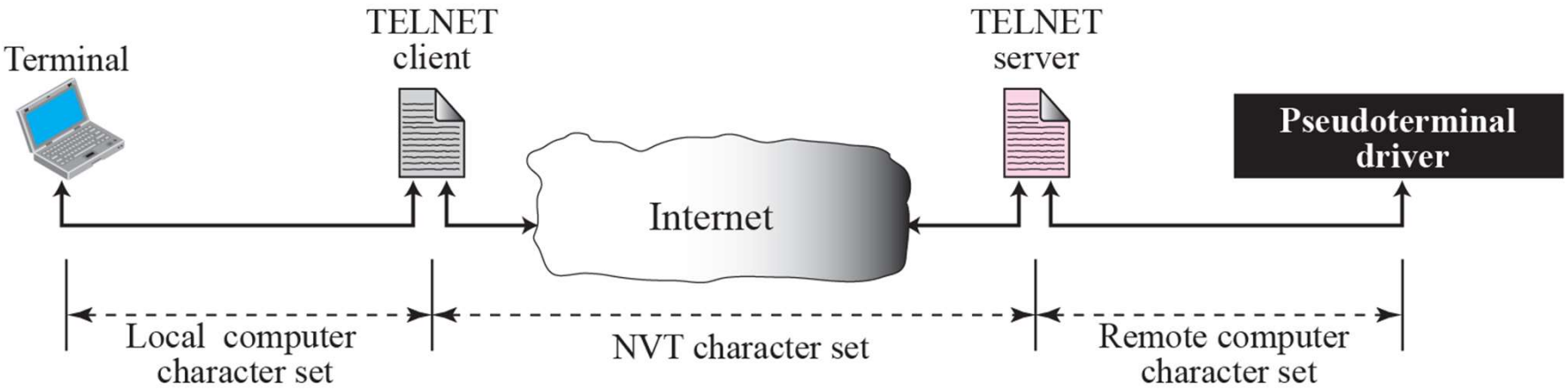
Local login


Application programs
Terminal
Terminal driver
Operating System

Remote login



Concept of NVT



Format of data and control characters

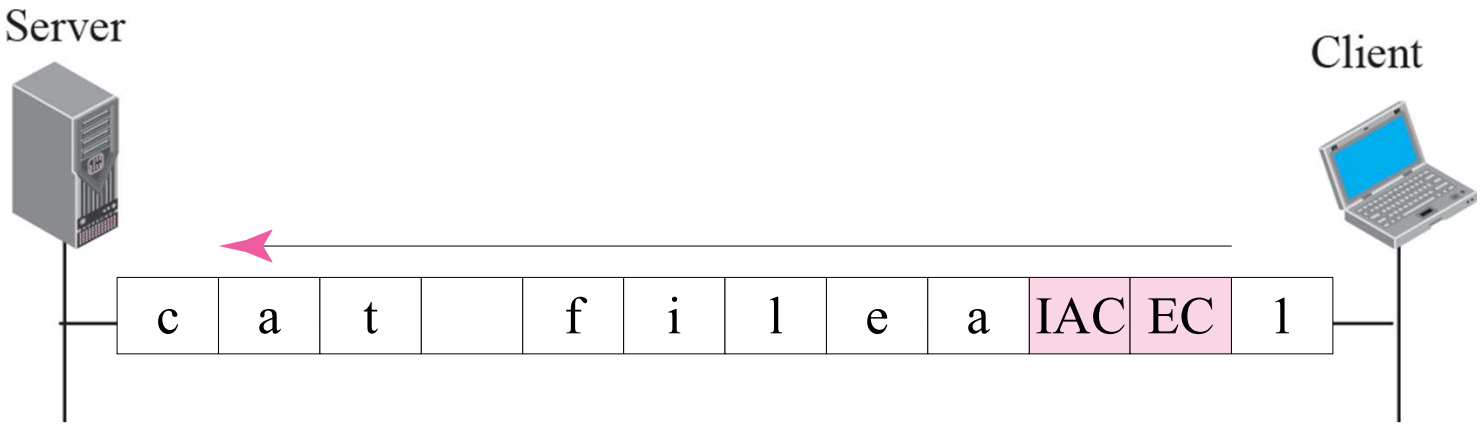
a. Data Character

b. Control Character

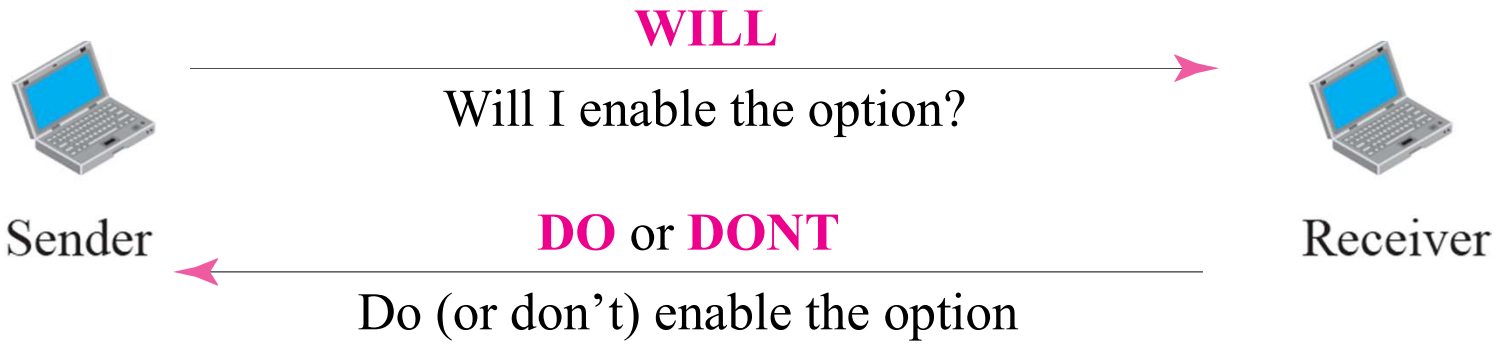
Table 20.1 *Some NVT control characters*

Character	*Decimal*	*Binary*	*Meaning*
EOF	236	11101100	End of file
EOR	239	11101111	End of record
SE	240	11110000	Suboption end
NOP	241	11110001	No operation
DM	242	11110010	Data mark
BRK	243	11110011	Break
IP	244	11110100	Interrupt process
AO	245	11110101	Abort output
AYT	246	11110110	Are you there?
EC	247	11110111	Erase character
EL	248	11111000	Erase line
GA	249	11111001	Go ahead
SB	250	11111010	Suboption begin
WILL	251	11111011	Agreement to enable option
WONT	252	11111100	Refusal to enable option
DO	253	11111101	Approval to option request
DONT	254	11111110	Denial of option request
IAC	255	11111111	Interpret (the next character) as control

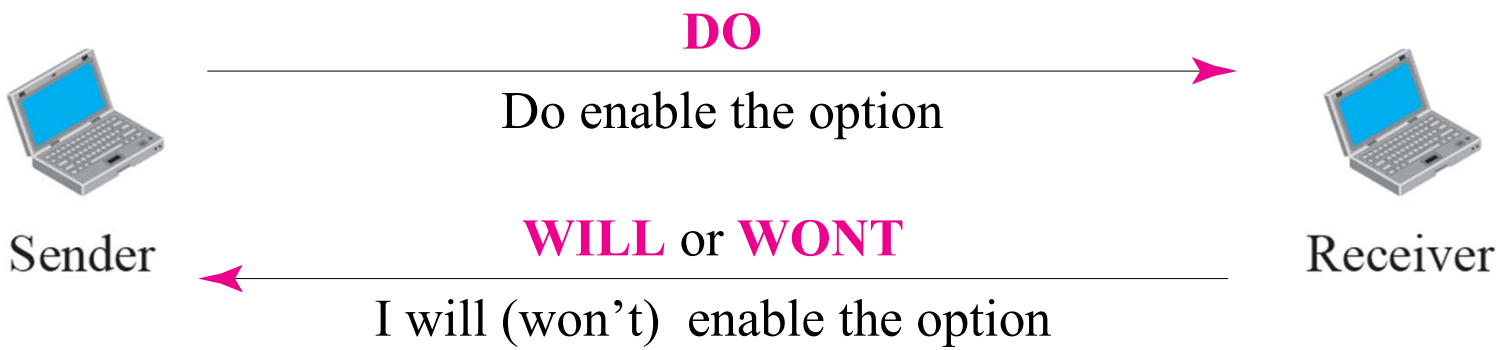
An example of embedding



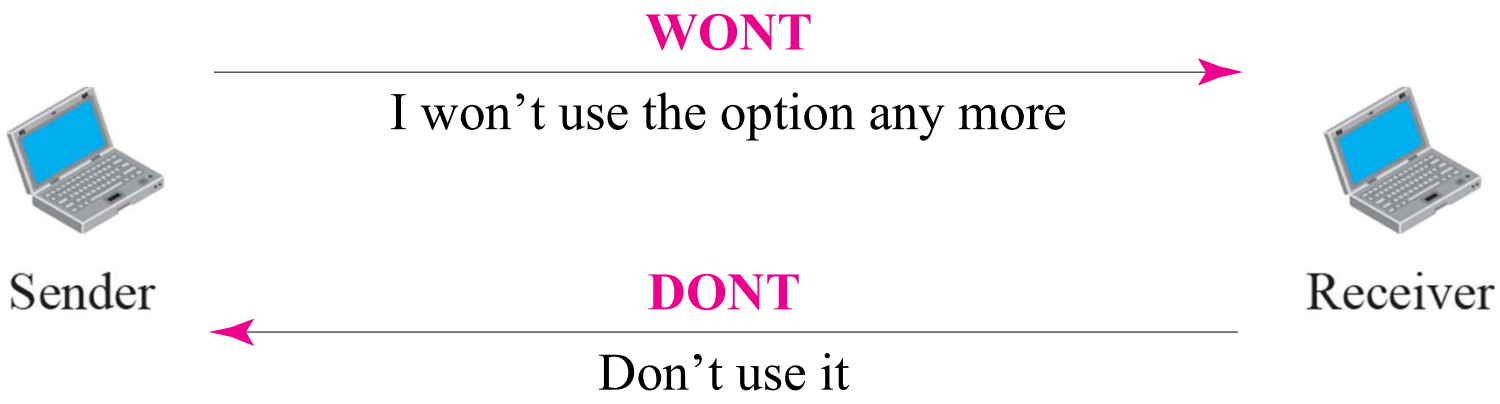
Offer to enable an option



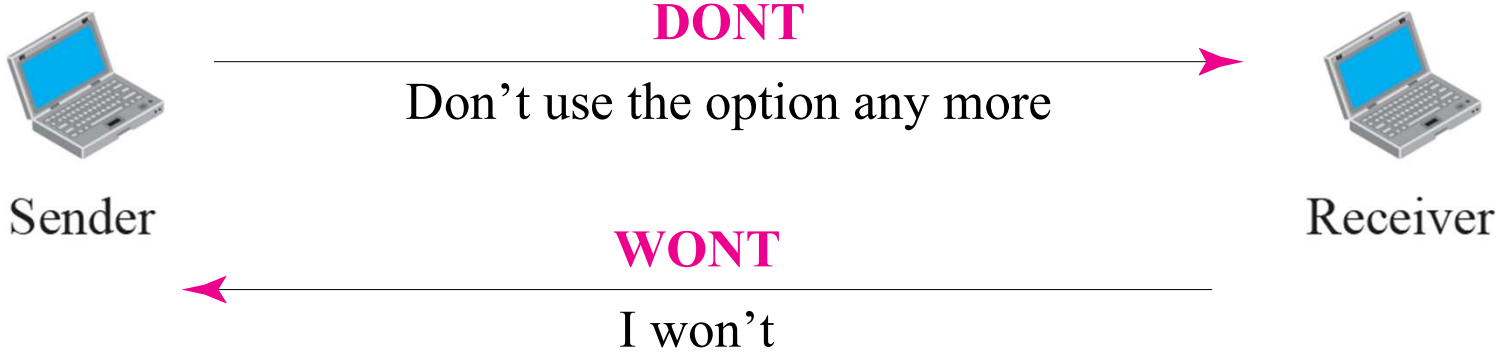
Request to enable an option



Offer to disable an option



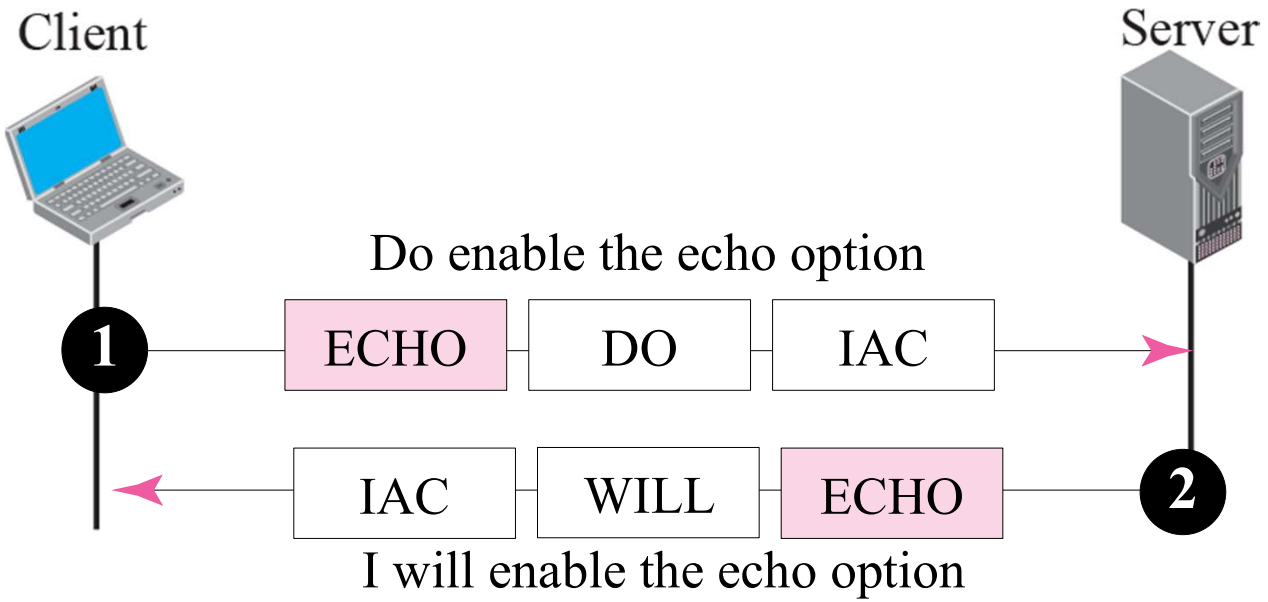
Request to disable an option



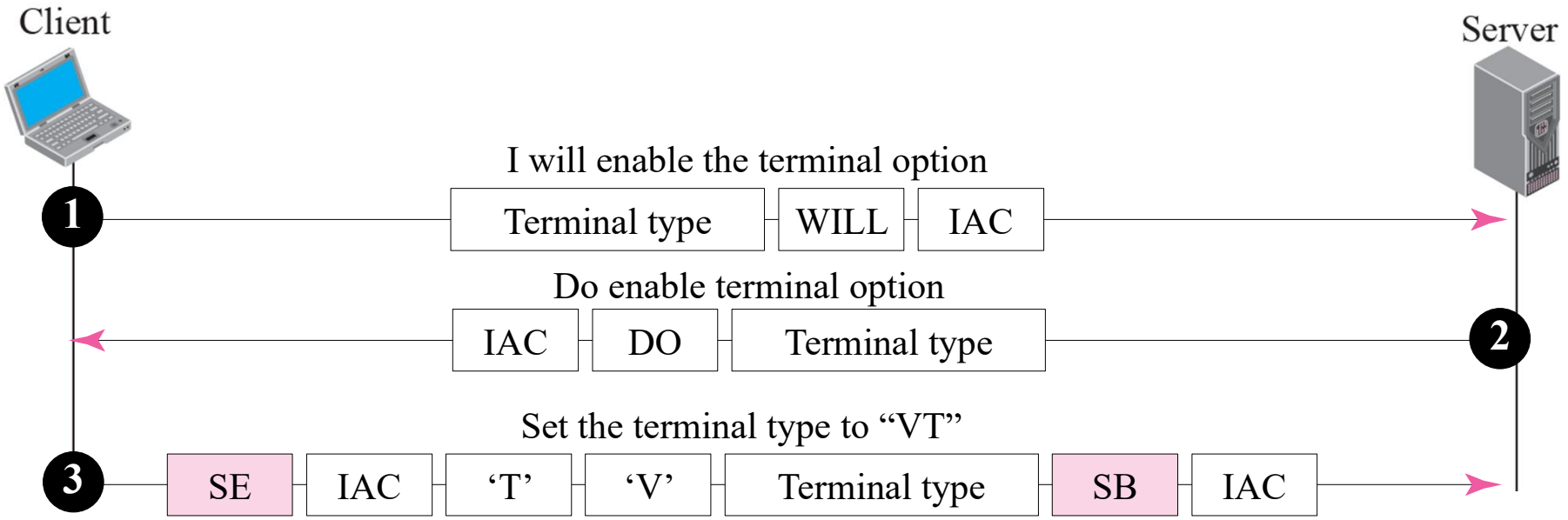
Example

Figure-1 shows an example of option negotiation. In this example, the client wants the server to echo each character sent to the server. In other words, when a character is typed at the user keyboard terminal, it goes to the server and is sent back to the screen of the user before being processed. The echo option is enabled by the server because it is the server that sends the characters back to the user terminal. Therefore, the client should request from the server the enabling of the option using DO. The request consists of three characters: IAC, DO, and ECHO. The server accepts the request and enables the option. It informs the client by sending the three-character approval: IAC, WILL, and ECHO.

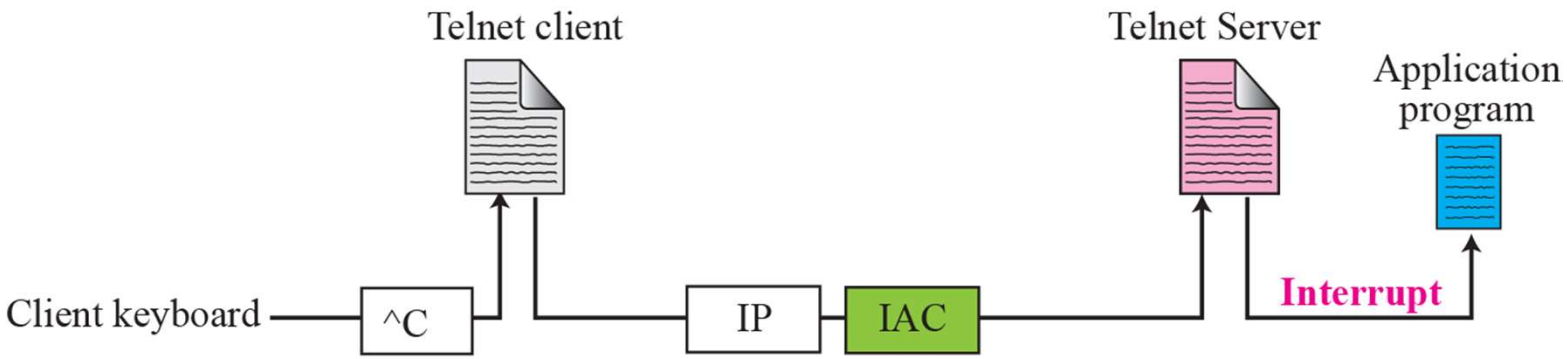
Figure 1 *Example 20.1: Echo option*



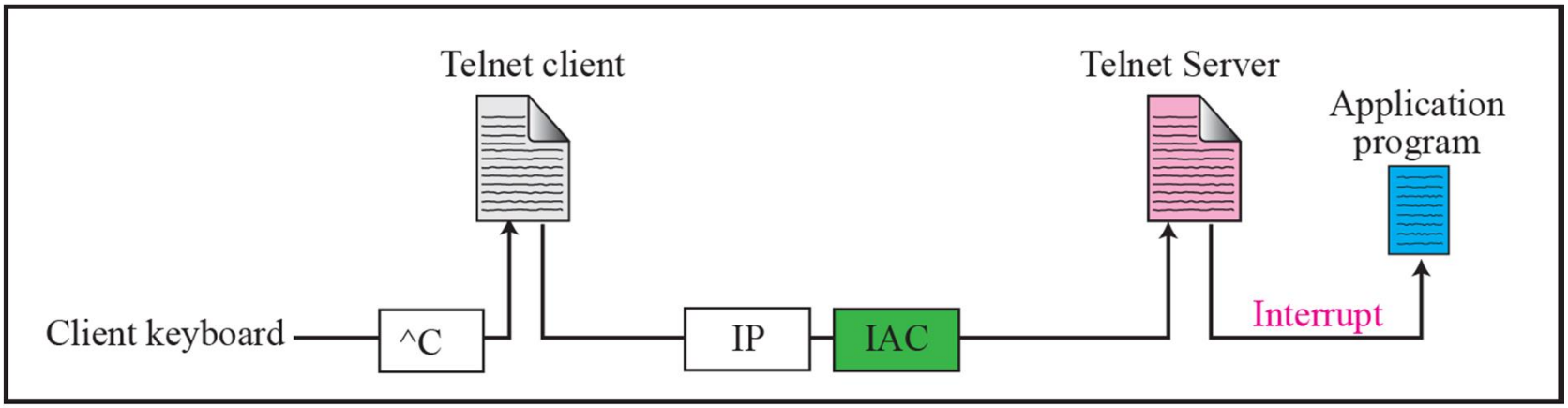
Example of sub-option negotiation



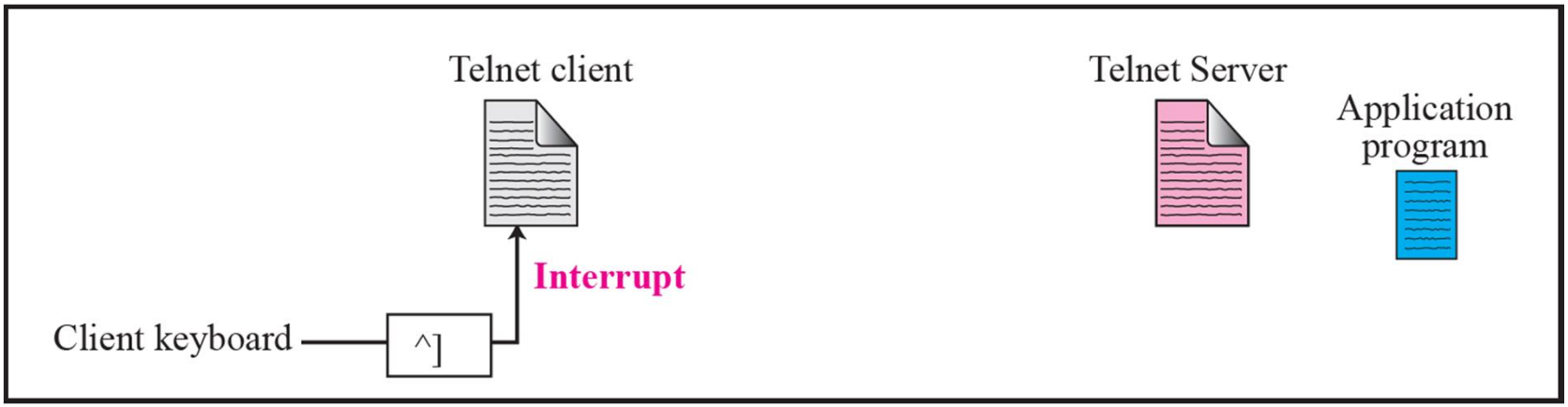
Example of interrupting an application program



Two different interruptions

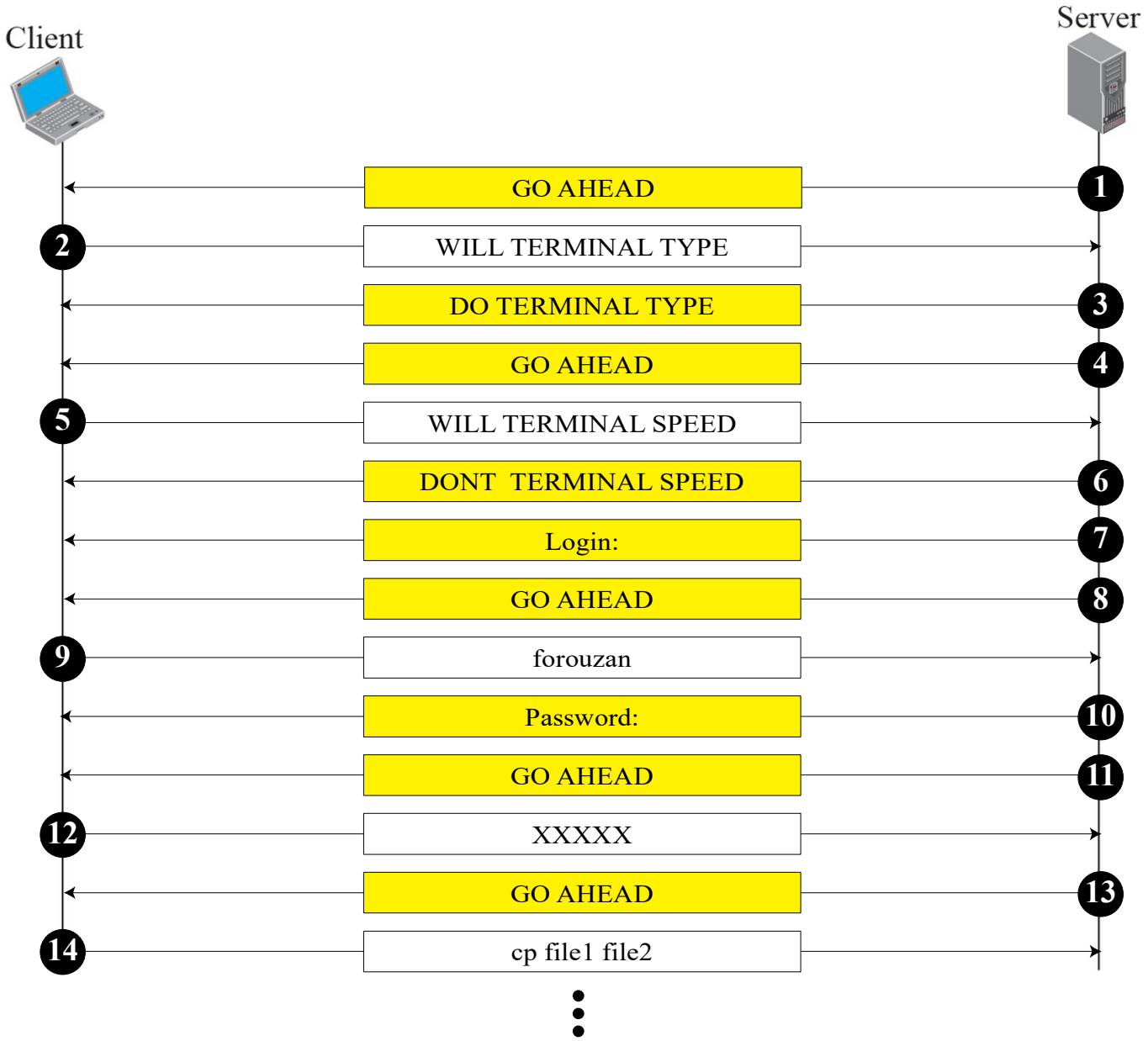


a. Interrupting the application program



b. Interrupting the client

Example


Client
Server
GO AHEAD
1
2
WILL TERMINAL TYPE
DO TERMINAL TYPE
3
GO AHEAD
4
5
WILL TERMINAL SPEED
DONT TERMINAL SPEED
6
Login:
7
GO AHEAD
8
9
forouzan
Password:
10
GO AHEAD
11
12
XXXXX
GO AHEAD
13
14
cp file1 file2