

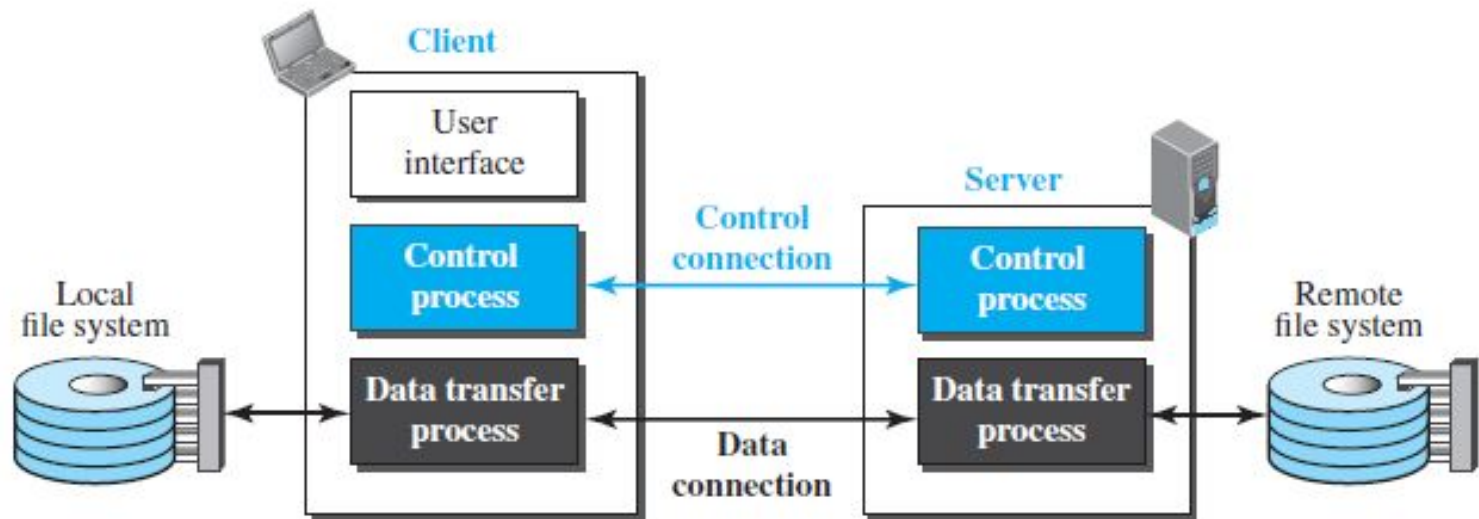
# APPLICATION LAYER

## FTP

- Standard protocol provided by TCP/IP for copying a file from one host to another
- File transfer can be done using HTTP and FTP, but FTP is a better choice to transfer large files or to transfer files using different formats

## Two Connections

The control connection remains connected during the entire interactive FTP session. The data connection is opened and then closed for each file transfer activity.



# APPLICATION LAYER

## FTP

FTP can transfer one of the following file types across the data connection\

- ASCII file
- EBCDIC file
- Image file.

FTP can transfer a file across the data connection using one of the following interpretations of the structure of the data

- *file structure*
- *record structure*
- *page structure*. The

# APPLICATION LAYER

## FTP

### *Transmission Mode*

- *Stream mode* - data delivered from FTP to TCP as a continuous stream of bytes
- *Block mode*- data delivered from FTP to TCP in blocks with block descriptor
- *Compressed mode* – *Compressed data is given to TCP*

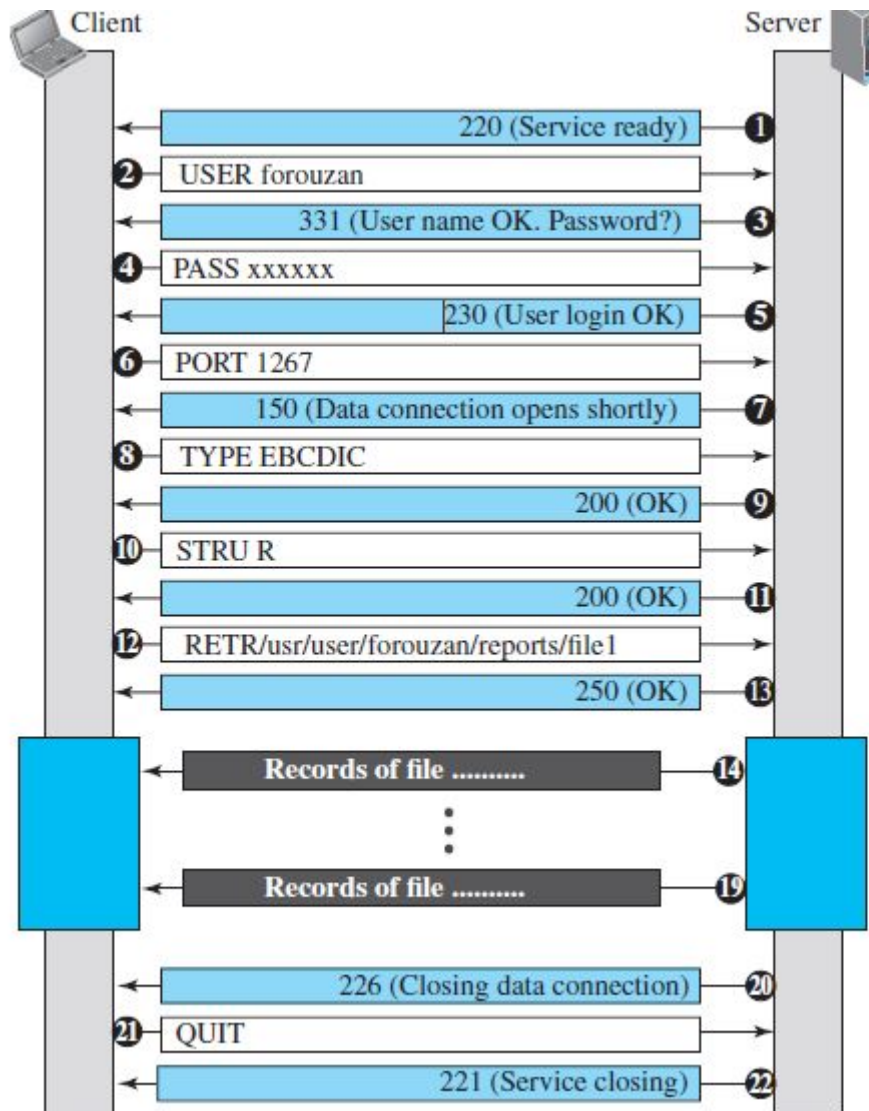
### *File Transfer*

- *Retrieving a file* (server to client)
- *Storing a file* (client to server)
- *Directory listing* (server to client)

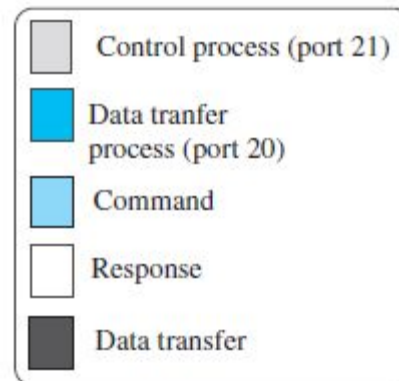
<i>Code</i>	<i>Description</i>	<i>Code</i>	<i>Description</i>
125	Data connection open	250	Request file action OK
150	File status OK	331	User name OK; password is needed
200	Command OK	425	Cannot open data connection
220	Service ready	450	File action not taken; file not available
221	Service closing	452	Action aborted; insufficient storage
225	Data connection open	500	Syntax error; unrecognized command
226	Closing data connection	501	Syntax error in parameters or arguments
230	User login OK	530	User not logged in

# APPLICATION LAYER

## FTP



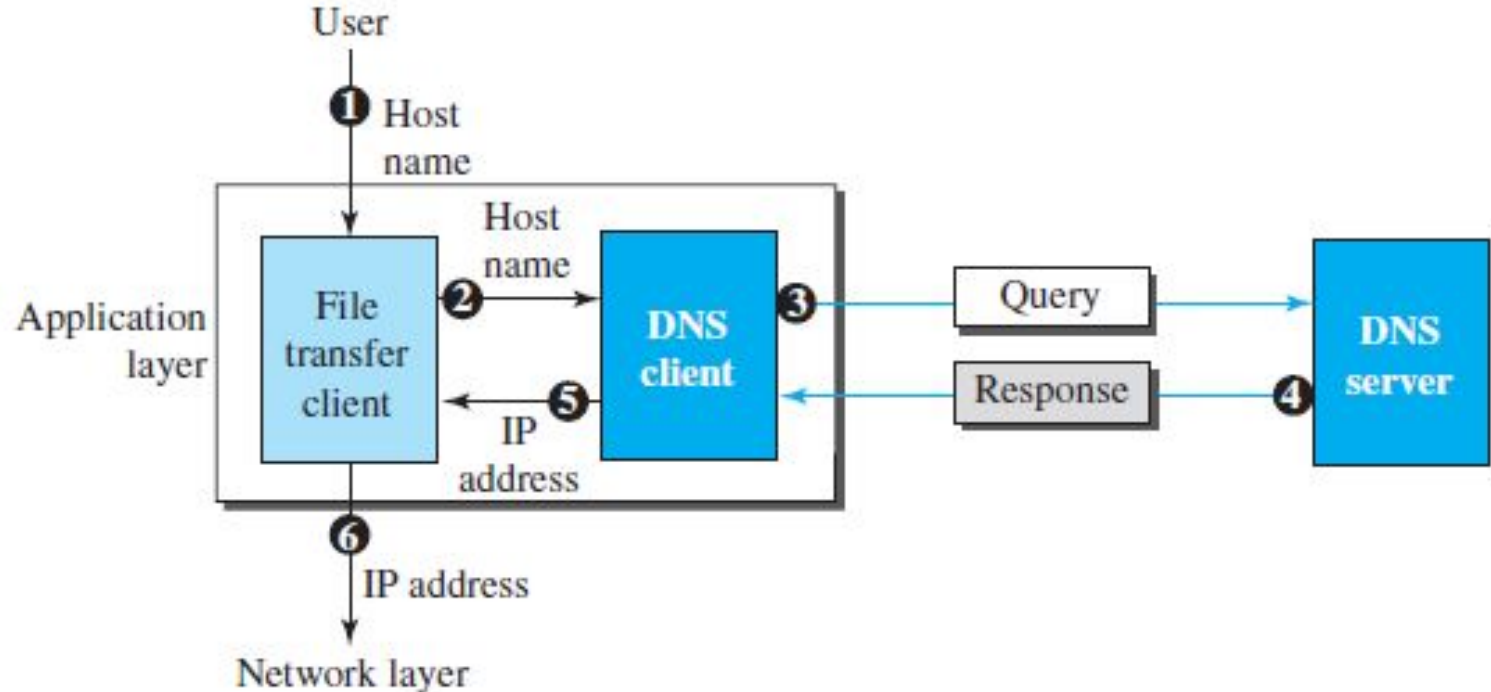
Command	Argument(s)	Description
ABOR		Abort the previous command
CDUP		Change to parent directory
CWD	Directory name	Change to another directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
MKD	Directory name	Create a new directory
PASS	User password	Password
PASV		Server chooses a port
PORT	Port identifier	Client chooses a port
PWD		Display name of current directory
QUIT		Log out of the system
RETR	File name(s)	Retrieve files; files are transferred from server to client
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
STOR	File name(s)	Store files; file(s) are transferred from client to server
STRU	F, R, or P	Define data organization (F: file, R: record, or P: page)
TYPE	A, E, I	Default file type (A: ASCII, E: EBCDIC, I: image)
USER	User ID	User information
MODE	S, B, or C	Define transmission mode (S: stream, B: block, or C: compressed)



# APPLICATION LAYER

## DNS

- 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 once booted, knows the address of one DNS server and DNS client sends message to DNS server for the address of the file transfer server.
- 4. The DNS server responds with the IP address of the desired file transfer server.
- 5. The DNS server passes the IP address to the file transfer client.
- 6. The file transfer client now uses the received IP address to access the file transfer server.



# APPLICATION LAYER

## DNS- Name Space

It maps each address to a unique name and organized in two ways

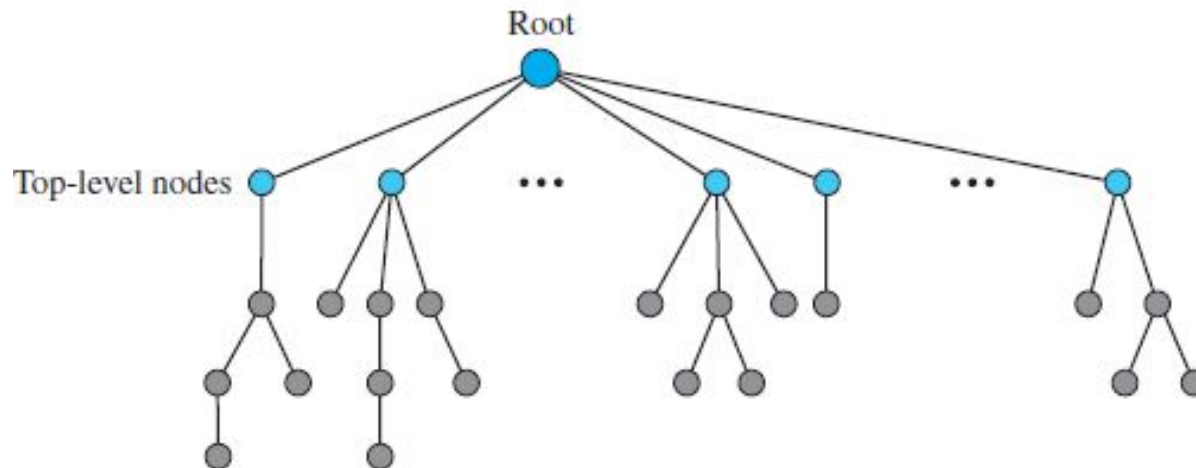
- Flat - name in this space is a sequence of characters without structure
- Hierarchical- each name is made of several parts arranged in an order

Each node in the tree has a domain name consists of sequence of labels separated by dots (.).

Each label is a string with a maximum of 63 characters .

The tree can have only 128 levels: level 0 (root) to level 127

- Fully qualified domain name(FQDN)-reach till root
- Partially qualified domain name (PQDN)-Resolver supplies the suffix

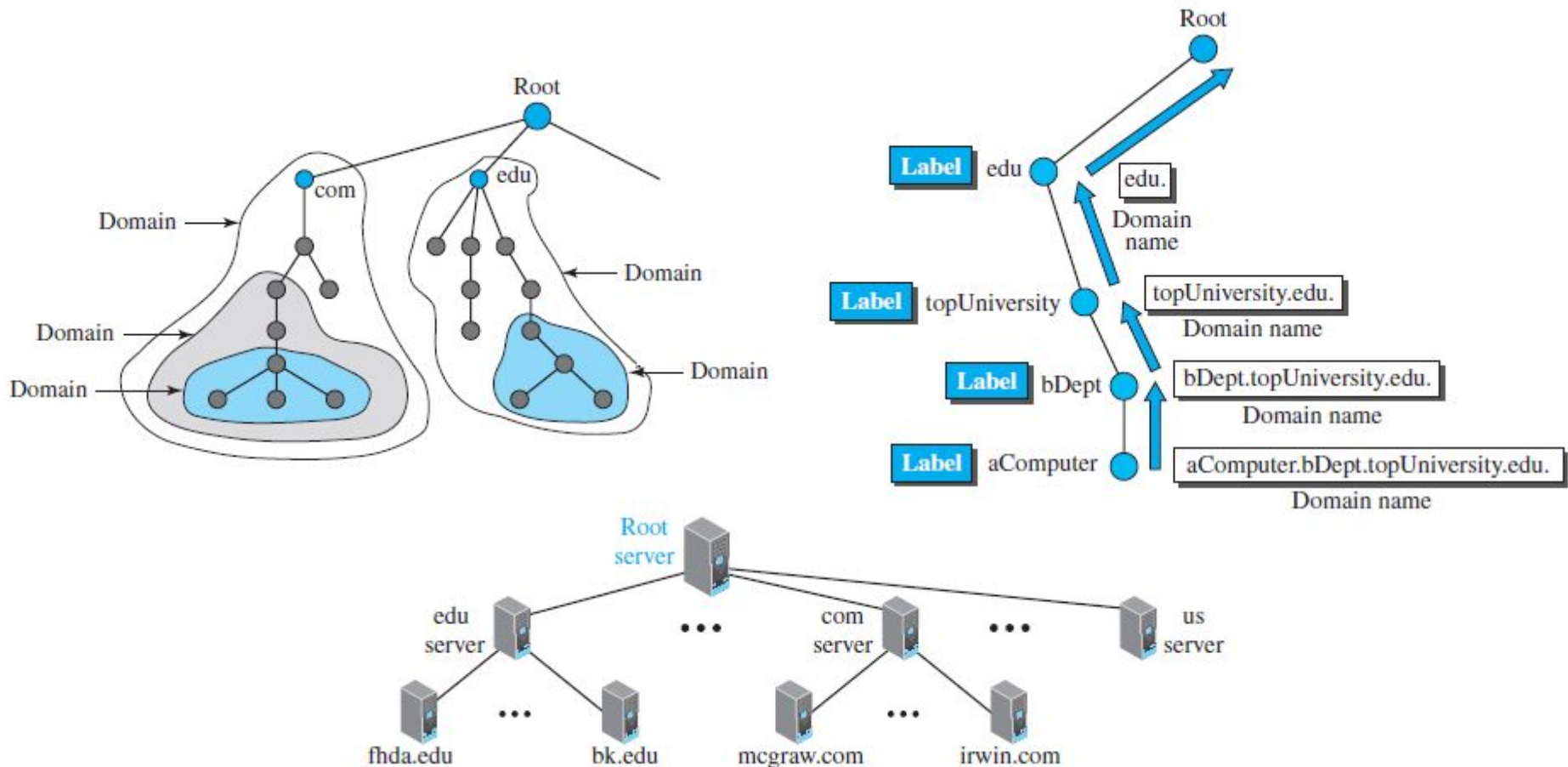




# APPLICATION LAYER

## DNS- Name Space

Domain- subtree of the domain name space



Name server hierarchy

# APPLICATION LAYER

## DNS

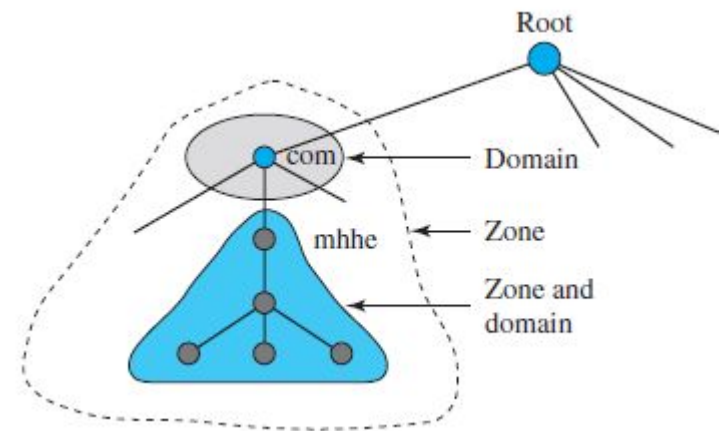
### *Zone*

- Complete domain name hierarchy cannot be stored on a single server and is divided among many servers
- A server is responsible for or has authority over a region is called a *zone*

**Root Servers**-a server whose zone consists of the whole tree (many root servers all over the Internet)

**Primary Servers**-a server that stores a file about the zone in the local disk for which it is the authority and responsible for creating, maintaining, and updating the zone file

**Secondary Servers**- responsible for keeping the primary server data for sharing between zones. It neither creates nor update the zone files



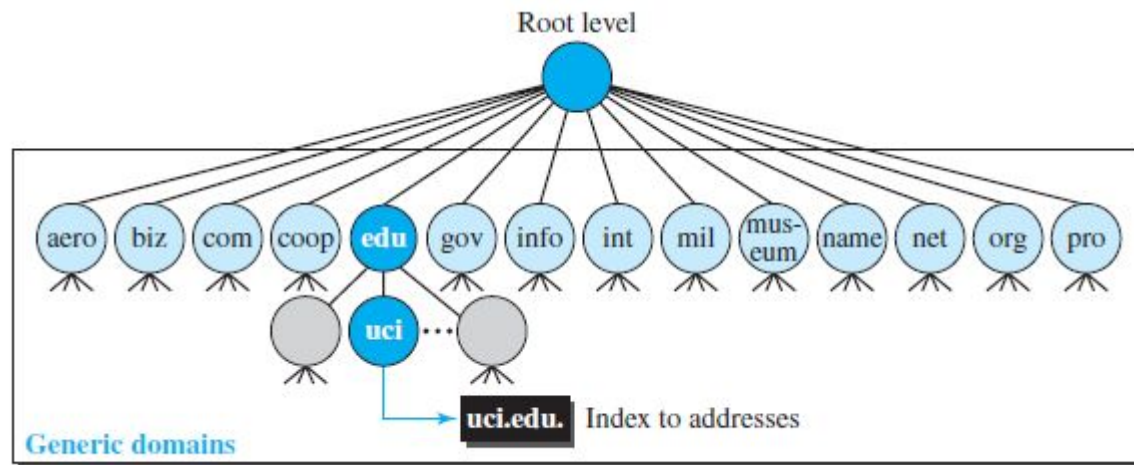


# APPLICATION LAYER

## DNS in the Internet

The domain name space (tree) was originally divided into three different sections:  
Generic domain Country domain and Inverse domain

### Generic domain

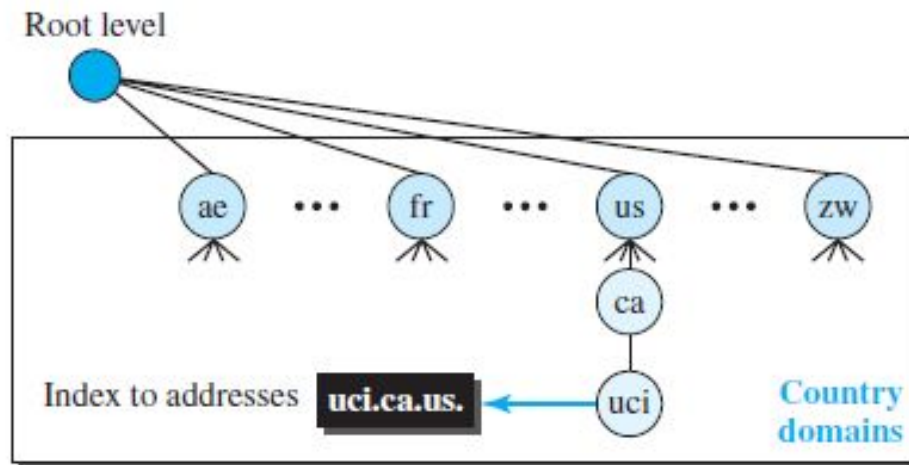


<i>Label</i>	<i>Description</i>	<i>Label</i>	<i>Description</i>
<b>aero</b>	Airlines and aerospace	<b>int</b>	International organizations
<b>biz</b>	Businesses or firms	<b>mil</b>	Military groups
<b>com</b>	Commercial organizations	<b>museum</b>	Museums
<b>coop</b>	Cooperative organizations	<b>name</b>	Personal names (individuals)
<b>edu</b>	Educational institutions	<b>net</b>	Network support centers
<b>gov</b>	Government institutions	<b>org</b>	Nonprofit organizations
<b>info</b>	Information service providers	<b>pro</b>	Professional organizations

# APPLICATION LAYER

## DNS in the Internet

### Country Domains

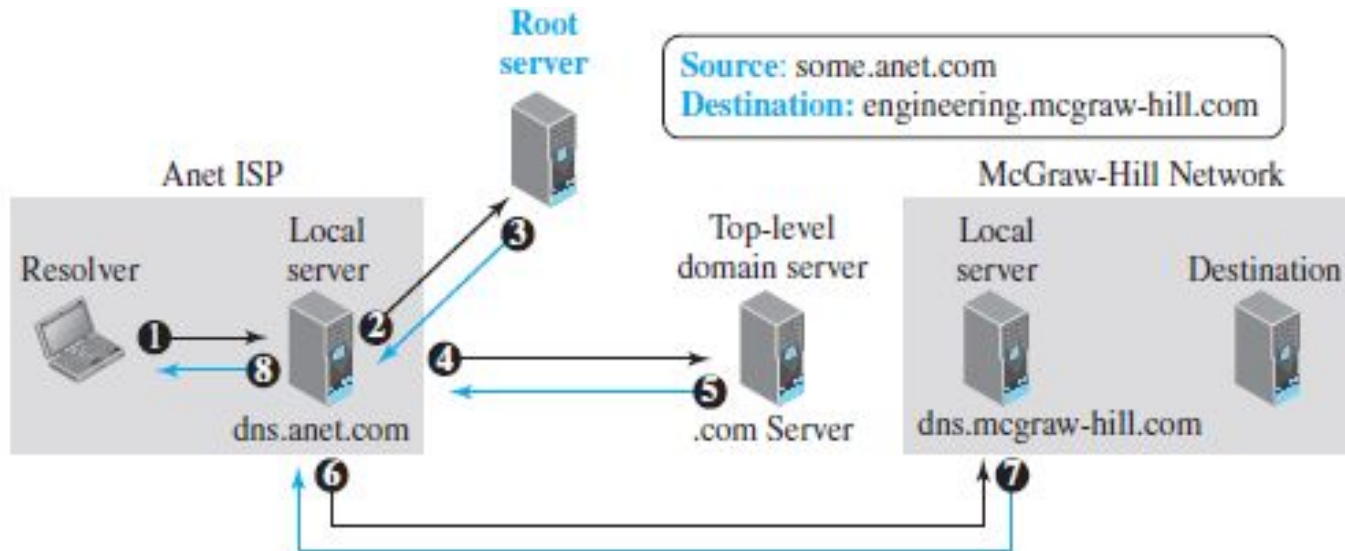
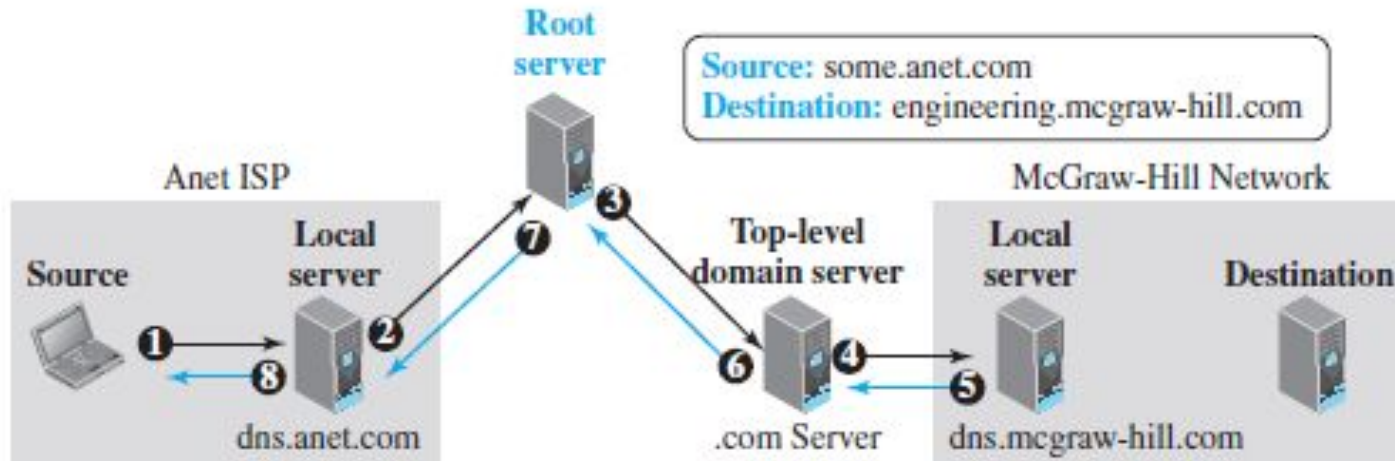


# APPLICATION LAYER

## DNS in the Internet

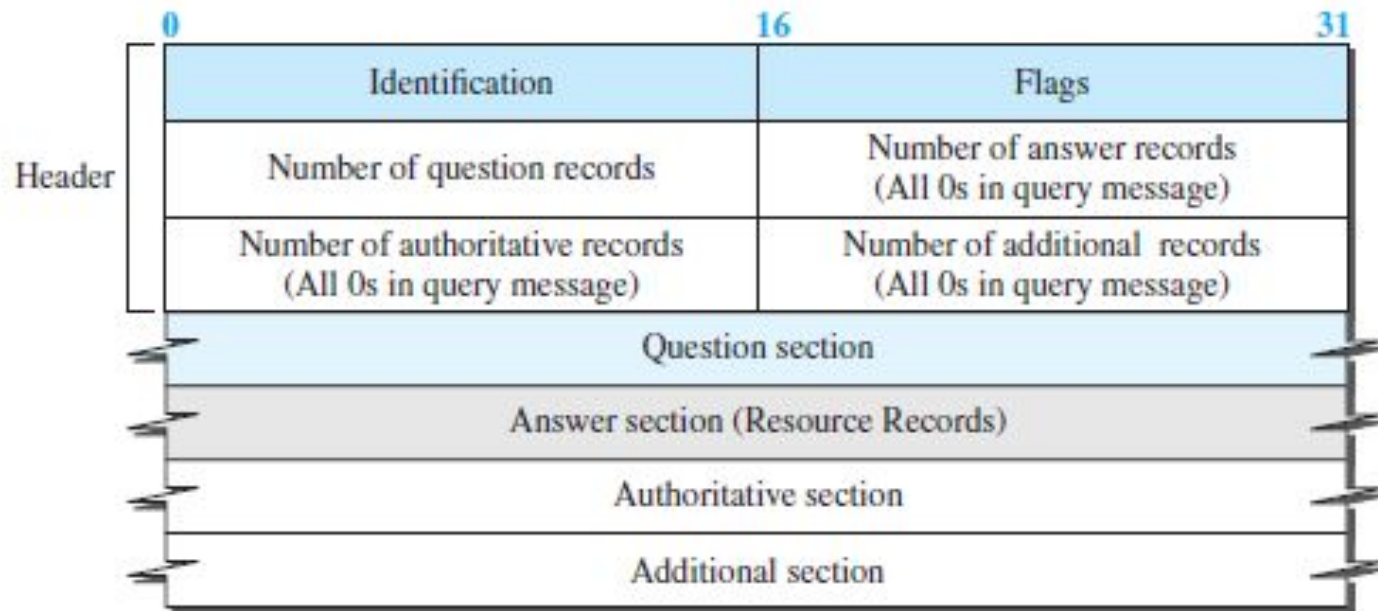
### Resolution Recursive and Iterative

Mapping a name to an address is called *name-address resolution*.



# APPLICATION LAYER

## DNS Messages



### Note:

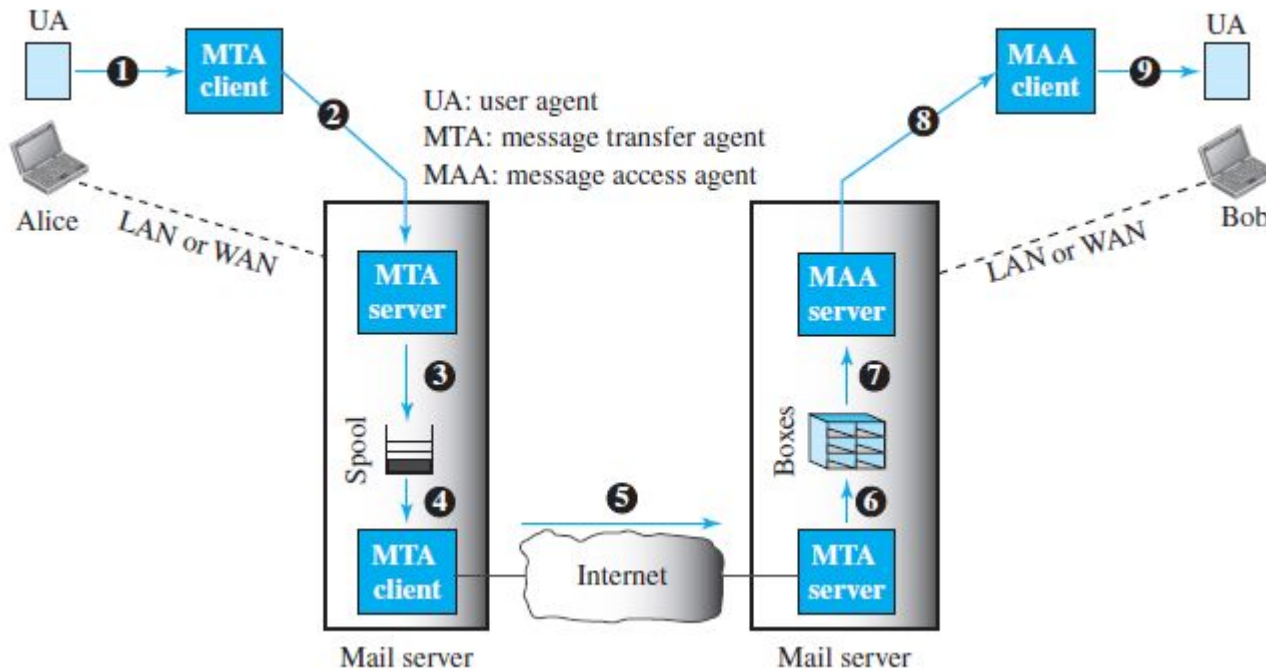
The query message contains only the question section.  
The response message includes the question section,  
the answer section, and possibly two other sections.

# APPLICATION LAYER

## E-mail

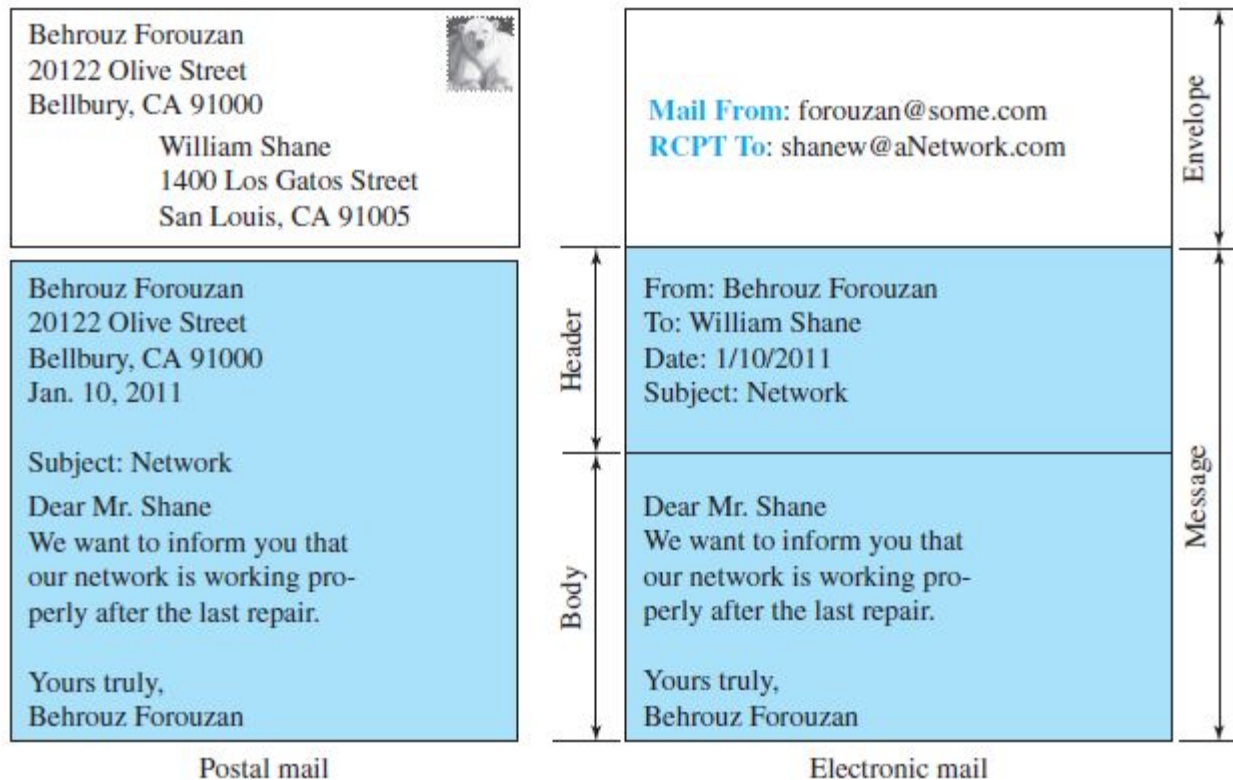
The electronic mail system needs two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server)

UA is a software package (program) that composes, reads, replies to, and forwards Messages through local mailboxes on the user computers (Command-driven and GUI-based).



# APPLICATION LAYER

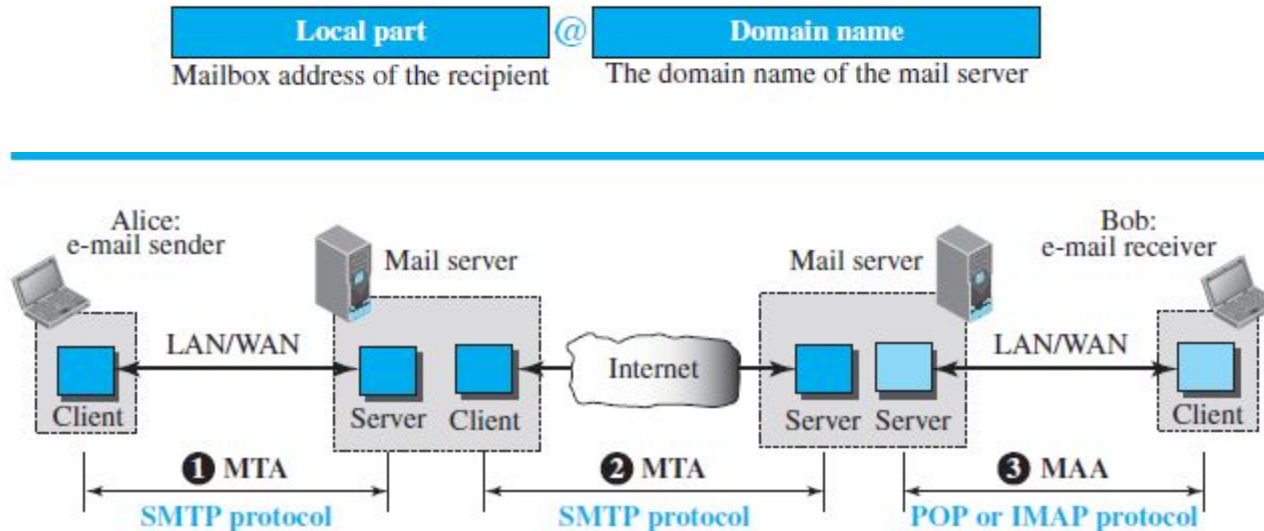
## Postal Mail Vs E-mail





# APPLICATION LAYER

## Protocols in E-mail



SMTP is used two times, between the sender and the sender's mail server and between the two mail servers( Commands and Responses)

# APPLICATION LAYER

## SMTP Commands

<i>Keyword</i>	<i>Argument(s)</i>	<i>Description</i>
HELO	Sender's host name	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message
RSET		Aborts the current mail transaction
VERFY	Name of recipient	Verifies the address of the recipient
NOOP		Checks the status of the recipient
TURN		Switches the sender and the recipient
EXPN	Mailing list	Asks the recipient to expand the mailing list
HELP	Command name	Asks the recipient to send information about the command sent as the argument
SEND FROM	Intended recipient	Specifies that the mail be delivered only to the terminal of the recipient, and not to the mailbox
SMOL FROM	Intended recipient	Specifies that the mail be delivered to the terminal <i>or</i> the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal <i>and</i> the mailbox of the recipient

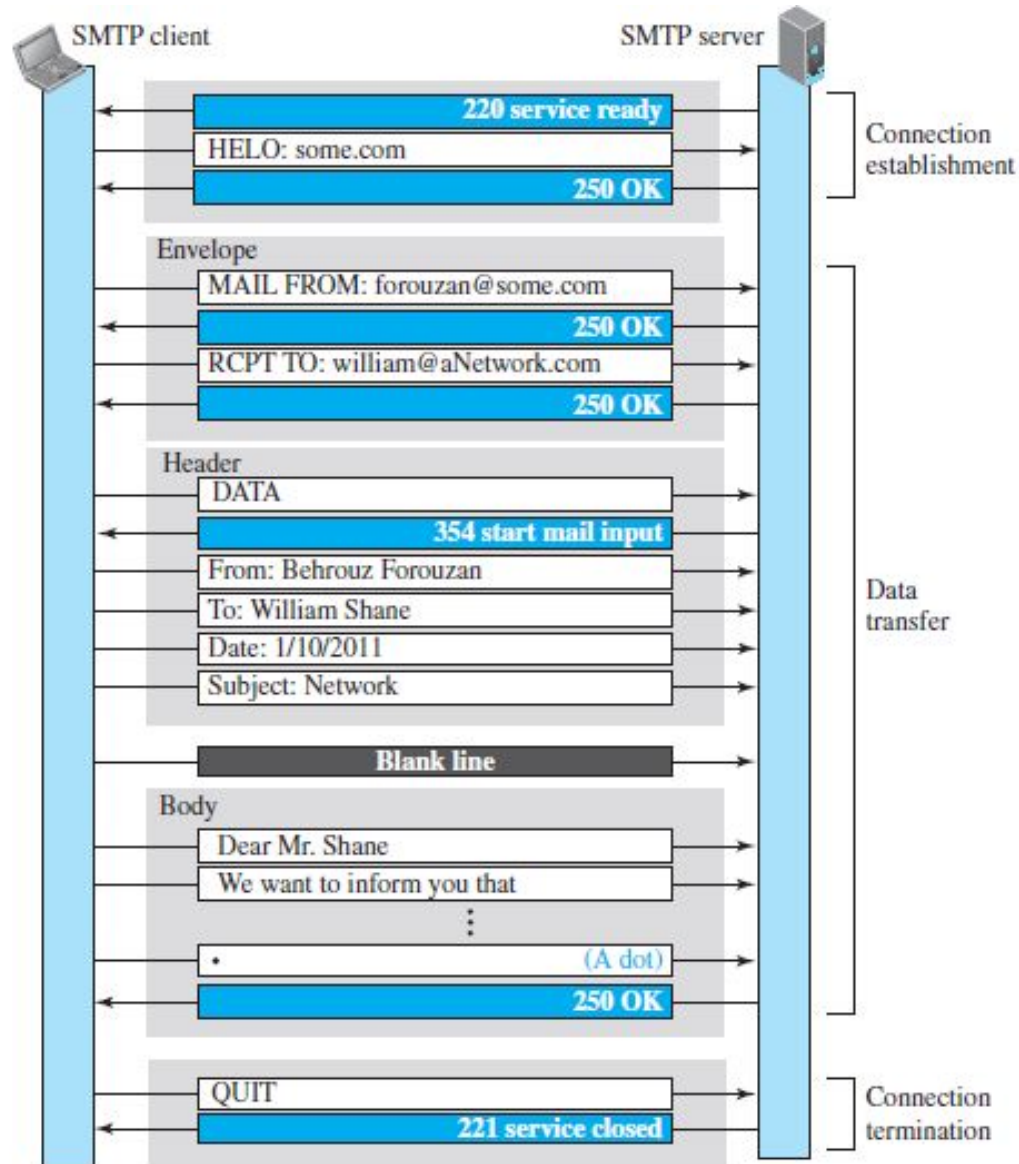
# APPLICATION LAYER

## SMTP Responses

<i>Code</i>	<i>Description</i>
<b>Positive Completion Reply</b>	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
<b>Positive Intermediate Reply</b>	
354	Start mail input
<b>Transient Negative Completion Reply</b>	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
<b>Permanent Negative Completion Reply</b>	
500	Syntax error; unrecognized command

# APPLICATION LAYER

## Mail Transfer Phases

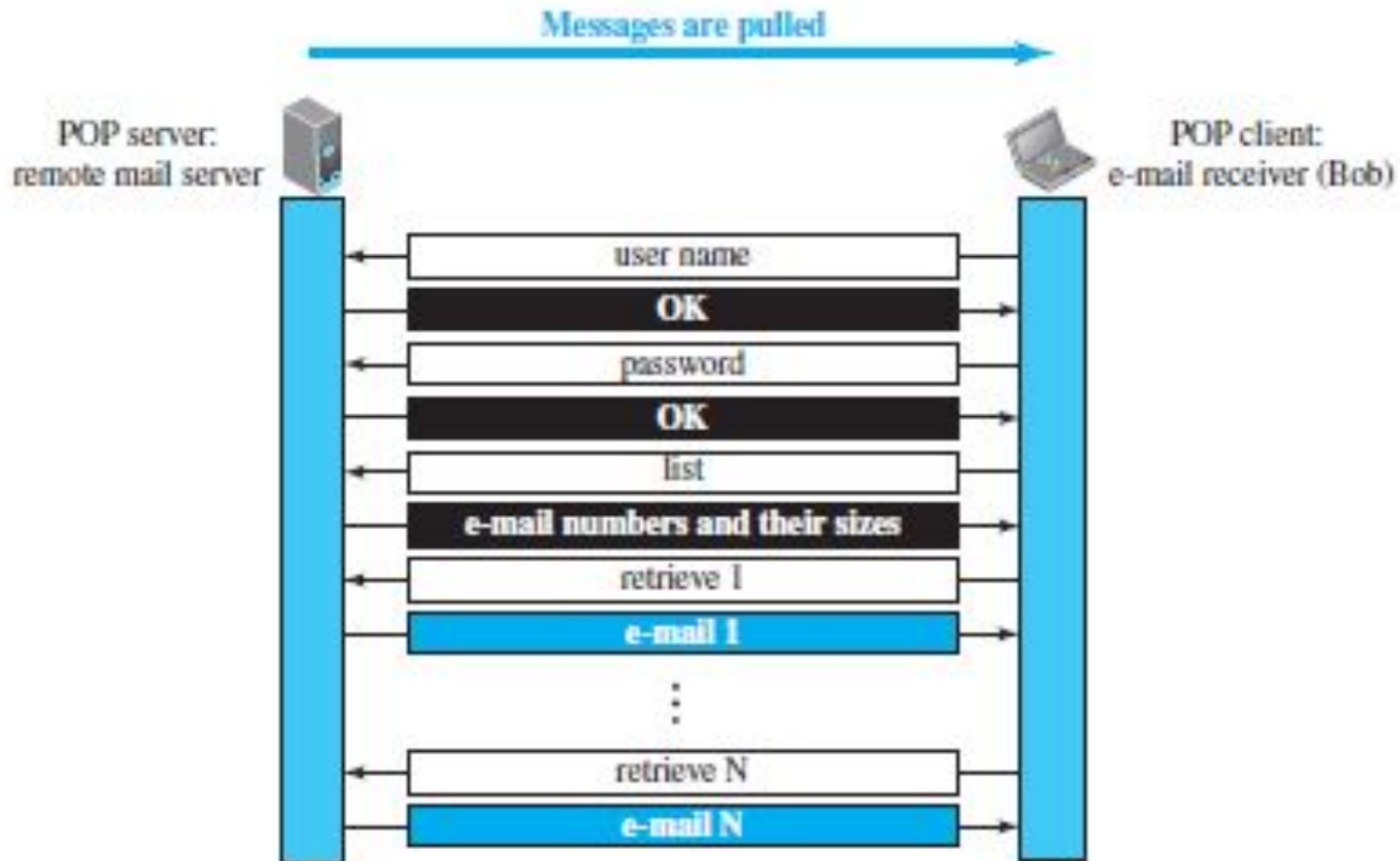




# APPLICATION LAYER

## Message Access Agent: POP and IMAP

SMTP is a *push* protocol POP and IMAP are *pull* protocols



# APPLICATION LAYER

## **MAA: POP (Post Office Protocol) and IMAP (Internet Mail Access Protocol)**

IMAP4 provides the following extra functions:

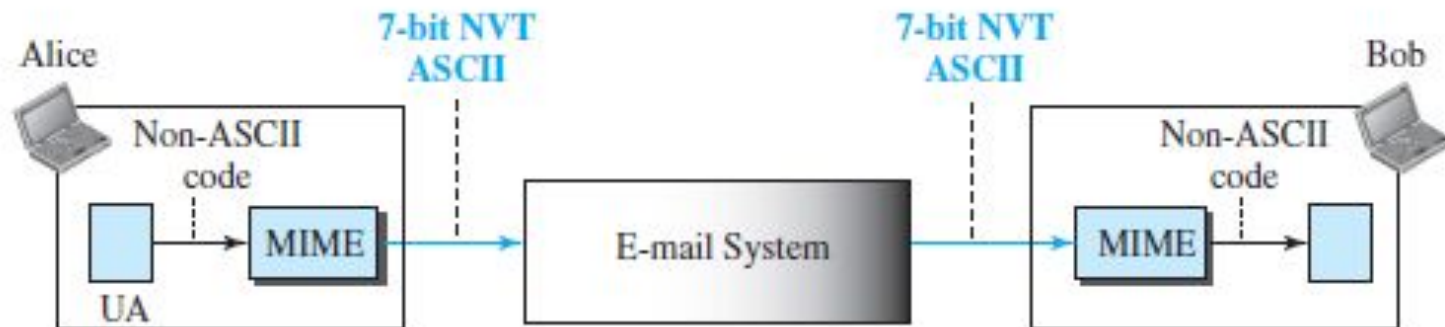
- ❑ A user can check the e-mail header prior to downloading.
- ❑ A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- ❑ A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- ❑ A user can create, delete, or rename mailboxes on the mail server.
- ❑ A user can create a hierarchy of mailboxes in a folder for e-mail storage



# APPLICATION LAYER

## MIME

**Multipurpose Internet Mail Extensions (MIME)** is a supplementary protocol that allows non-ASCII data to be sent through e-mail



## MIME Header

MIME headers

E-mail header	
MIME headers	MIME-Version: 1.1
	Content-Type: type/subtype
	Content-Transfer-Encoding: encoding type
	Content-ID: message ID
	Content-Description: textual explanation of nontextual contents
E-mail body	

# APPLICATION LAYER

## MIME

Current MIME version is 1.1

<i>Type</i>	<i>Subtype</i>	<i>Description</i>
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order'
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

<i>Type</i>	<i>Description</i>
7-bit	NVT ASCII characters with each line less than 1000 characters
8-bit	Non-ASCII characters with each line less than 1000 characters
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equal sign plus an ASCII code

# APPLICATION LAYER

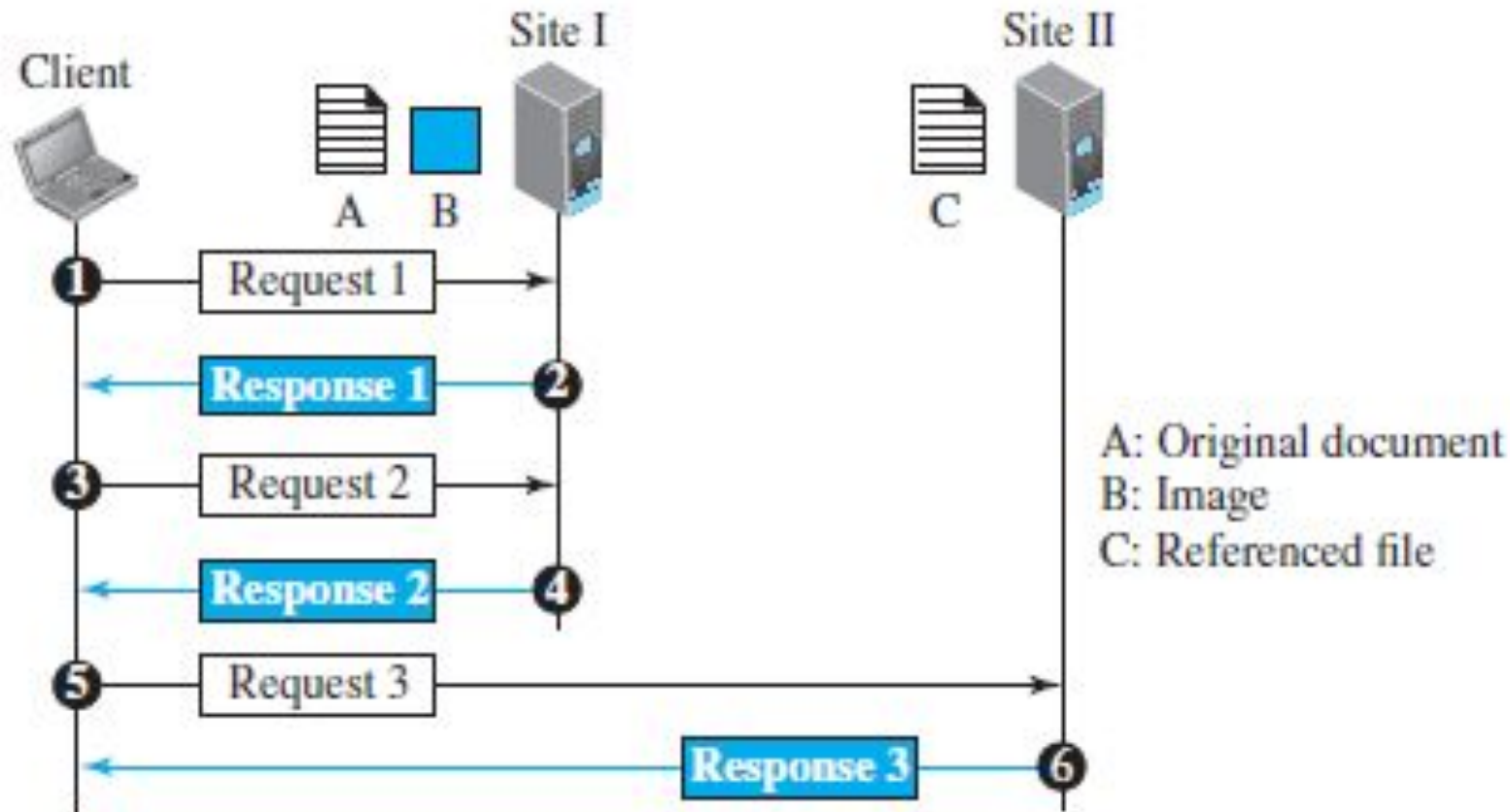
## **WORLD WIDE WEB**

- The WWW today is a distributed client-server service, in which a client using a browser can access a service using a server
- HyperText Transfer Protocol (HTTP)** is used to define how the client-server programs can be written to retrieve web pages from the Web
- The web page is stored at the server. Each time a request arrives, the corresponding document is sent to the client
- simple web page has no links to other web pages; a composite web page has one or more links to other web pages.
- Each web page is a file with a name and address.

# APPLICATION LAYER

## WORLD WIDE WEB

### Data Extraction



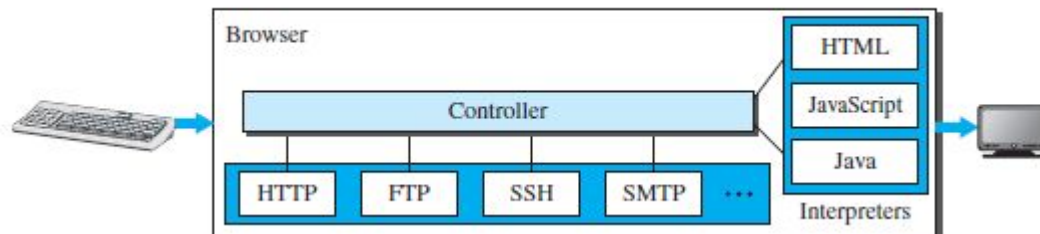
# APPLICATION LAYER

## WORLD WIDE WEB

### *Web Client (Browser)*

A variety of vendors offer commercial **browsers** that interpret and display a web page, and all of them use nearly the same architecture.

Each browser usually consists of three parts: a controller, client protocols, and interpreters.



### *Uniform Resource Locator (URL)*

**Protocol-** client-server program that we need in order to access the web page

**Host-** IP address of the server or the unique name given to the server(DNS)

**Path-** identifies the location and the name of the file in the underlying operating system.

# APPLICATION LAYER

## WORLD WIDE WEB

### COOKIES

**Helps in accessing the documents easily**

#### *Web Documents*

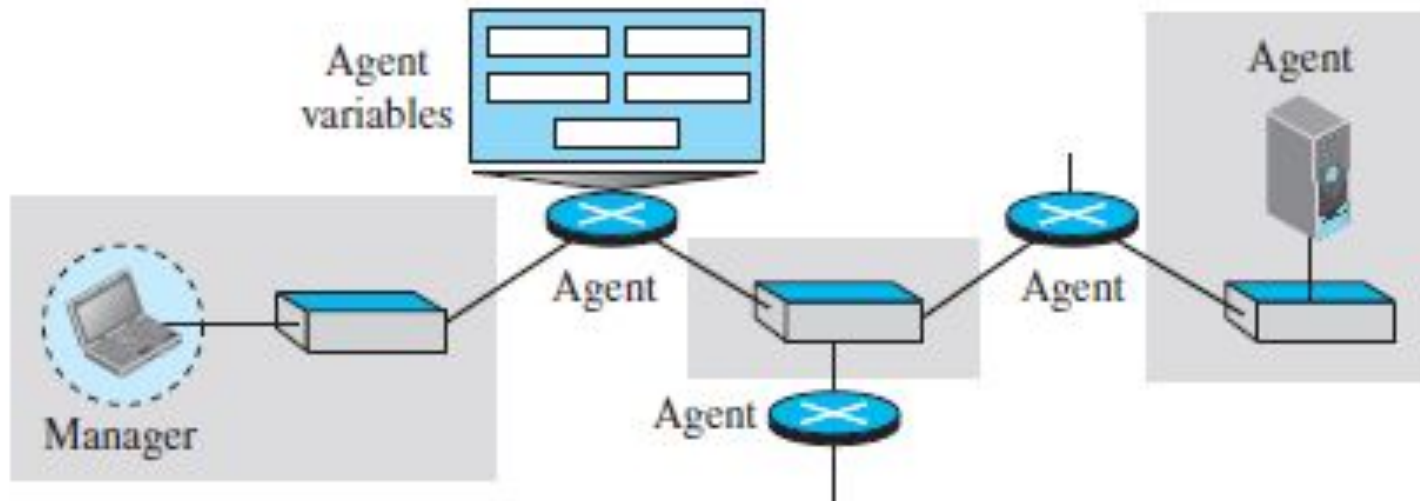
- *Static Documents*- fixed-content documents that are created and stored in a server
- *Dynamic Documents*- When a request arrives, the web server runs an application program or a script that creates the dynamic document that varies based on request
- *Active Documents* - When a browser requests an active document, the server sends a copy of the document or a script. The document is then run at the client (browser) site



# APPLICATION LAYER

## SNMP

SNMP is a framework for managing devices in an internet using the TCP/IP protocol suite. SNMP is an application-level protocol in which a few manager stations( runs the client program) control a set of agents(runs the server program).



1. A manager checks an agent by requesting information that reflects the behavior of the agent
2. A manager forces an agent to perform a task by resetting values in the agent database.
3. An agent contributes to the management process by warning the manager of an unusual situation.

# APPLICATION LAYER

## Management Components

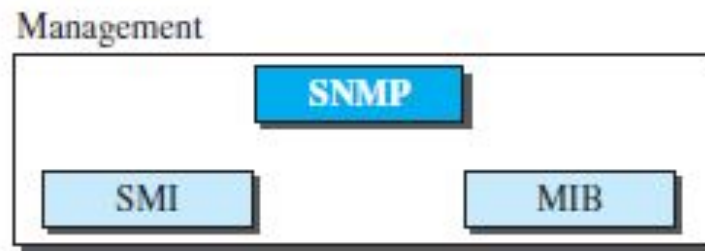
**SNMP defines the format of packets exchanged between a manager and an agent.**

**It reads and changes the status of objects (values of variables) in SNMP packets.**

**SNMP works with**

**SMI-Structure of Management Information**

**MIB- Management Information Base**

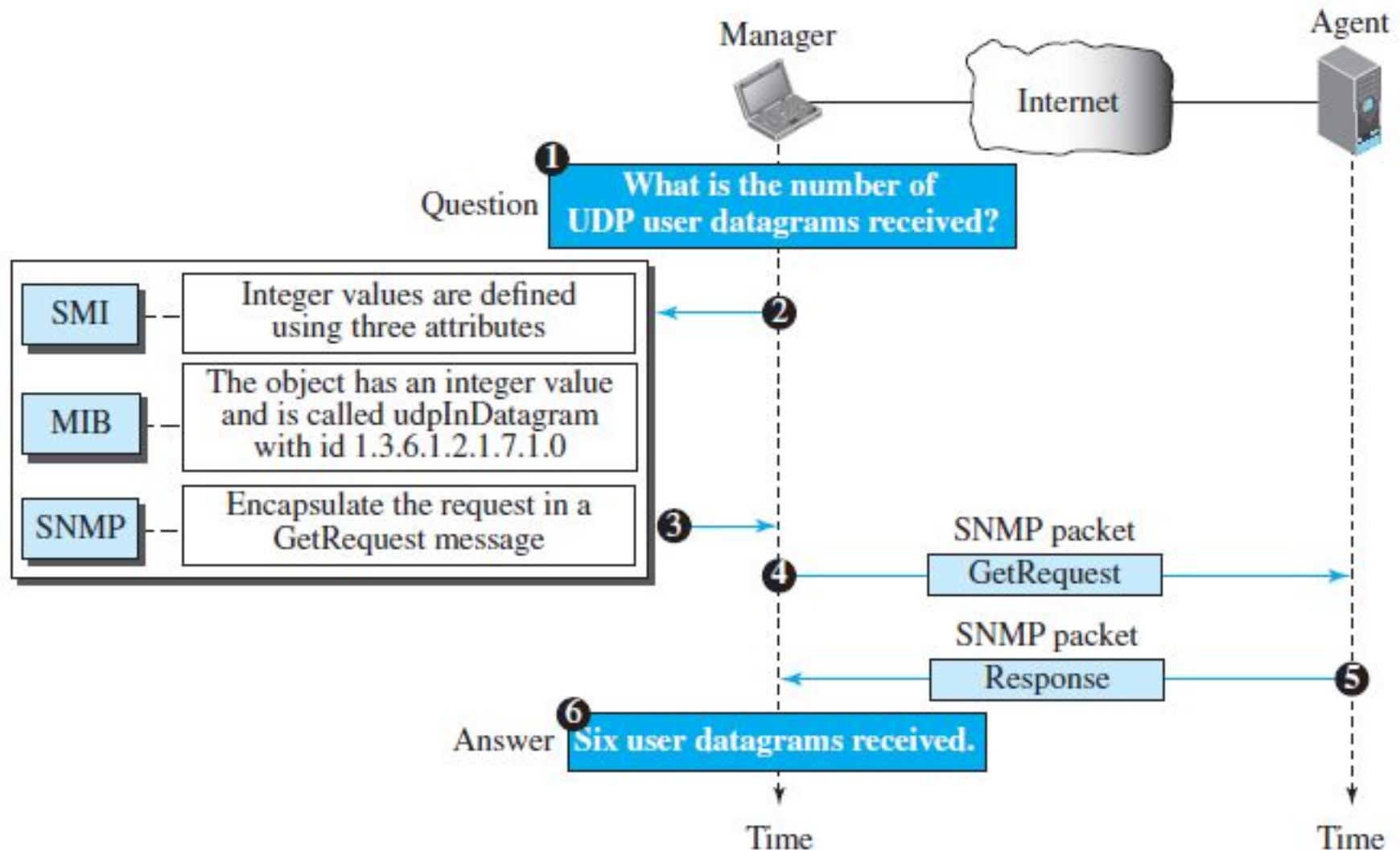


**SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values.**

**MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.**

# APPLICATION LAYER

## Management Overview

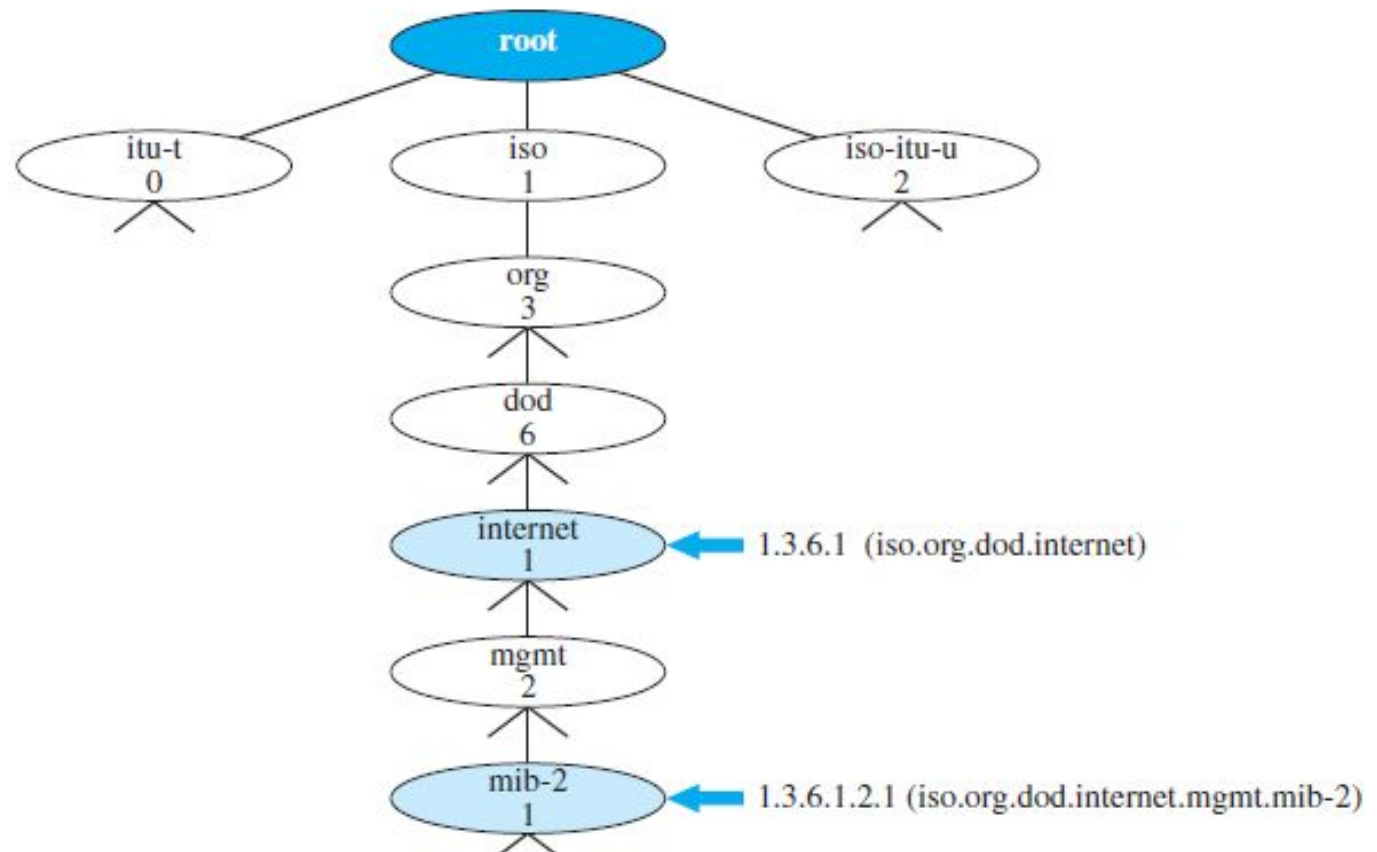


# APPLICATION LAYER

## SMI Functions

- To name objects.
- To define the type of data that can be stored in an object.
- To show how to encode data for transmission over the network

SMI uses an **object identifier**, which is a hierarchical identifier based on a tree structure to name objects globally.



# APPLICATION LAYER

## SMI Functions Contd.

SMI uses **Abstract Syntax Notation One (ASN.1)** to define the type of objects

SMI has two broad categories of data types: *simple* and *structured*.

Type	Size	Description
INTEGER	4 bytes	An integer with a value between $-2^{31}$ and $2^{31}-1$
Integer32	4 bytes	Same as INTEGER
Unsigned32	4 bytes	Unsigned with a value between 0 and $2^{32}-1$
OCTET STRING	Variable	Byte-string up to 65,535 bytes long
OBJECT IDENTIFIER	Variable	An object identifier
IPAddress	4 bytes	An IP address made of four integers
Counter32	4 bytes	An integer whose value can be incremented from zero to $2^{32}$ ; when it reaches its maximum value it wraps back to zero

Type	Size	Description
Counter64	8 bytes	64-bit counter
Gauge32	4 bytes	Same as Counter32, but when it reaches its maximum value, it does not wrap; it remains there until it is reset
TimeTicks	4 bytes	A counting value that records time in 1/100ths of a second
BITS		A string of bits
Opaque	Variable	Uninterpreted string

# APPLICATION LAYER

## SMI Functions Contd.

### *Structured Data Types*

***Sequence.*** A *sequence* data type is a combination of simple data types, not necessarily of the same type similar to a *struct* in C.

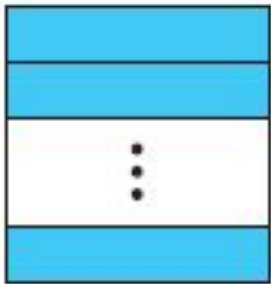
***Sequence of.*** A *sequence of* data type is a combination of simple data types all of the same type like array



a. Simple variable



b. Sequence



c. Sequence of



d. Sequence of (sequences)

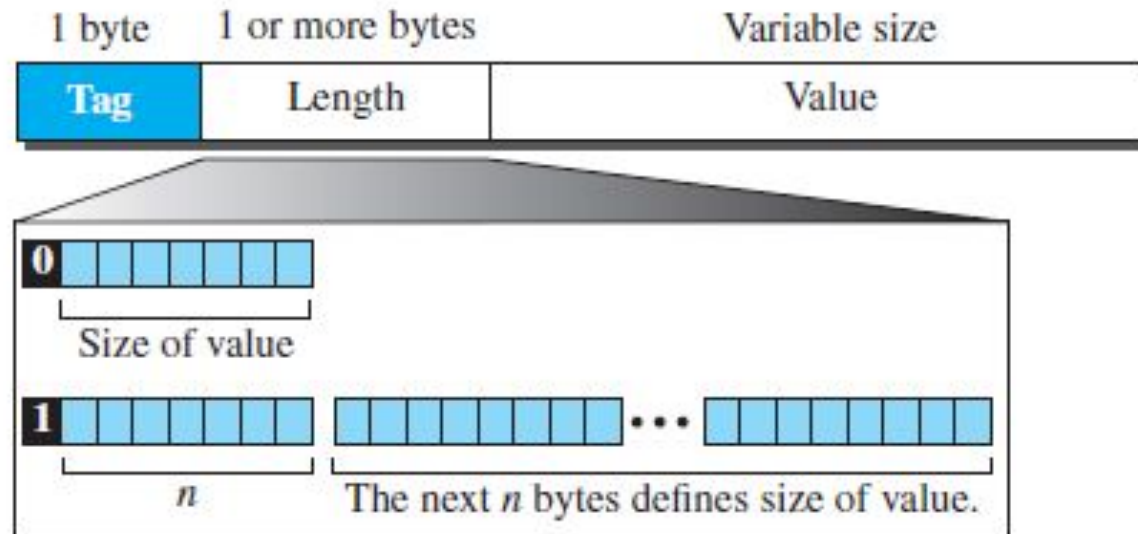


# APPLICATION LAYER

## SMI Functions Contd.

SMI uses **Basic Encoding Rules (BER)**, to encode data to be transmitted over the network in triplet form: tag, length, and value (TLV)

- The tag is a 1-byte field that defines the type of data
- The length field is 1 or more bytes.
- If it is 1 byte, the most significant bit must be 0 and other 7 bits defines length
- If it is more than 1 byte, the most significant bit of the first byte must be 1 and the other 7 bits of the first byte specify the number of bytes



# APPLICATION LAYER

## SMI Functions Contd.

Codes used to specify data types

<i>Data Type</i>	<i>Tag (Hex)</i>	<i>Data Type</i>	<i>Tag (Hex)</i>
INTEGER	02	IPAddress	40
OCTET STRING	04	Counter	41
OBJECT IDENTIFIER	06	Gauge	42
NULL	05	TimeTicks	43
SEQUENCE, SEQUENCE OF	30	Opaque	44

### Example : INT 14

0x02	0x04	0x00	0x00	0x00	0x0E
Tag (integer)	Length (4 bytes)	Value (14)			

### Example : String HI

0x04	0x02	0x48	0x49
Tag (String)	Length (2 bytes)	Value (H)	Value (I)

### Object Identifier 1.3.6.1

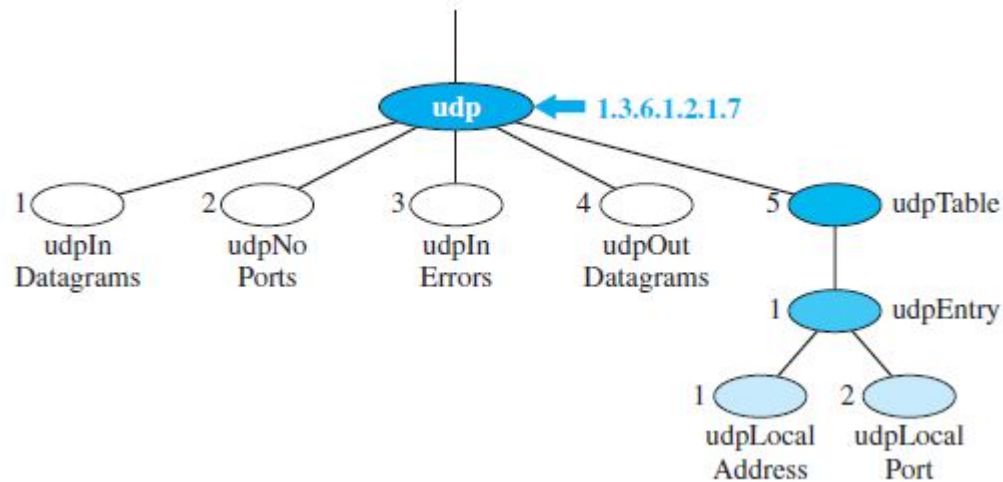
0x06	0x04	0x01	0x03	0x06	0x01
Tag (ObjectId)	Length (4 bytes)	Value (1)	Value (3)	Value (6)	Value (1)
← 1.3.6.1 (iso.org.dod.internet) →					

# APPLICATION LAYER

## MIB

### *Accessing MIB Variables*

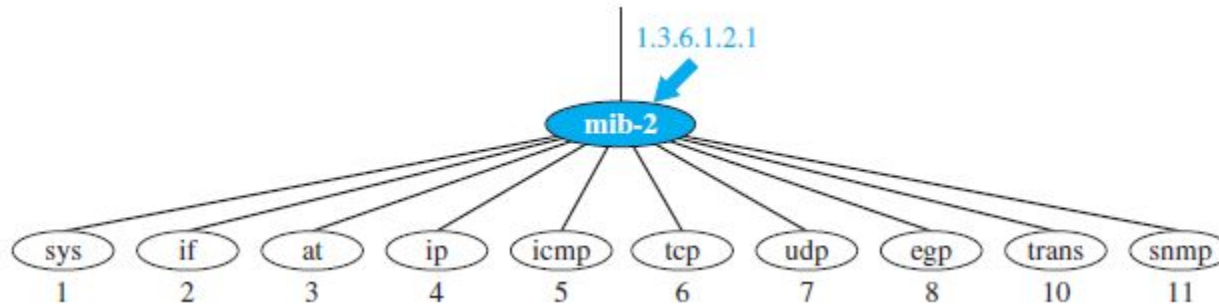
udpInDatagrams	→	1.3.6.1.2.1.7.1
udpNoPorts	→	1.3.6.1.2.1.7.2
udpInErrors	→	1.3.6.1.2.1.7.3
udpOutDatagrams	→	1.3.6.1.2.1.7.4



# APPLICATION LAYER

## MIB

Each agent has its own MIB2, which is a collection of all the objects that the manager can manage



- **Sys** - System details
- **if** - Interface details
- **at** - ARP table information
- **ip** - info related to routing table and the IP address.
- **icmp** info related to ICMP, such as the number of packets sent and received including errors
- **tcp** - information related to TCP, such as the connection table, time-out value, number of ports, and number of packets sent and received.
- **udp** This object defines general information related to UDP, such as the number of ports and number of packets sent and received.
- **egp** These objects are related to the operation of EGP.
- **trans** These objects are related to the specific method of transmission (future use).
- **snmp** This object defines general information related to SNMP itself.

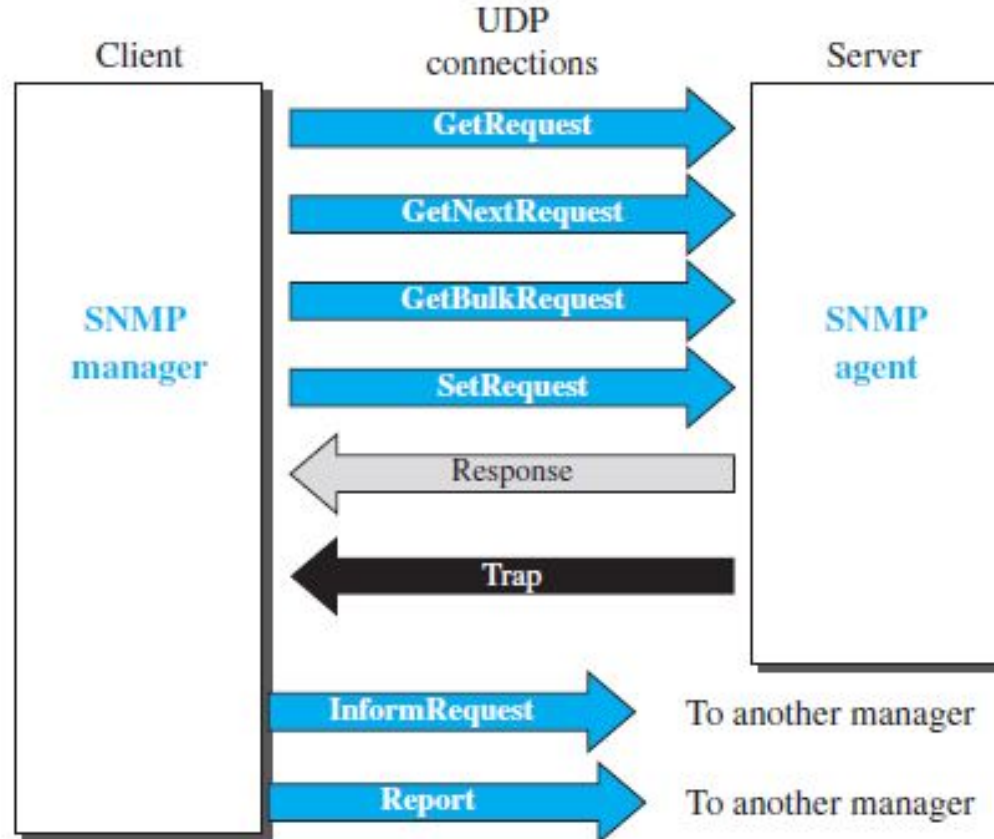
# APPLICATION LAYER

## MIB

SNMPv3 defines eight types of protocol data units (or PDUs): *GetRequest*, *GetNextRequest*, *GetBulkRequest*, *SetRequest*, *Response*, *Trap*, *InformRequest*, and *Report*

Trap is to report an event.

For example, if the agent is rebooted, it informs the manager and reports the time of rebooting



# APPLICATION LAYER

## Port Numbers for SNMP

SNMP uses the services of UDP on two well-known ports, 161 and 162. The wellknown port 161 is used by the server (agent), and the well-known port 162 is used by the client (manager). SNMPv3 has added two new features to the previous version: security and remote administration.

