MIT-WPU

॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Computer Science and Engineering

TY BTech Trimester-VII

Disclaimer:

a. Information included in these slides came from multiple sources. We have tried our best to cite the sources. Please refer to the references to learn about the sources, when applicable.

b. The slides should be used only for preparing notes, academic purposes (e.g. in teaching a class), and should not be used for commercial purposes.

Unit 4: APPLICATION LAYER

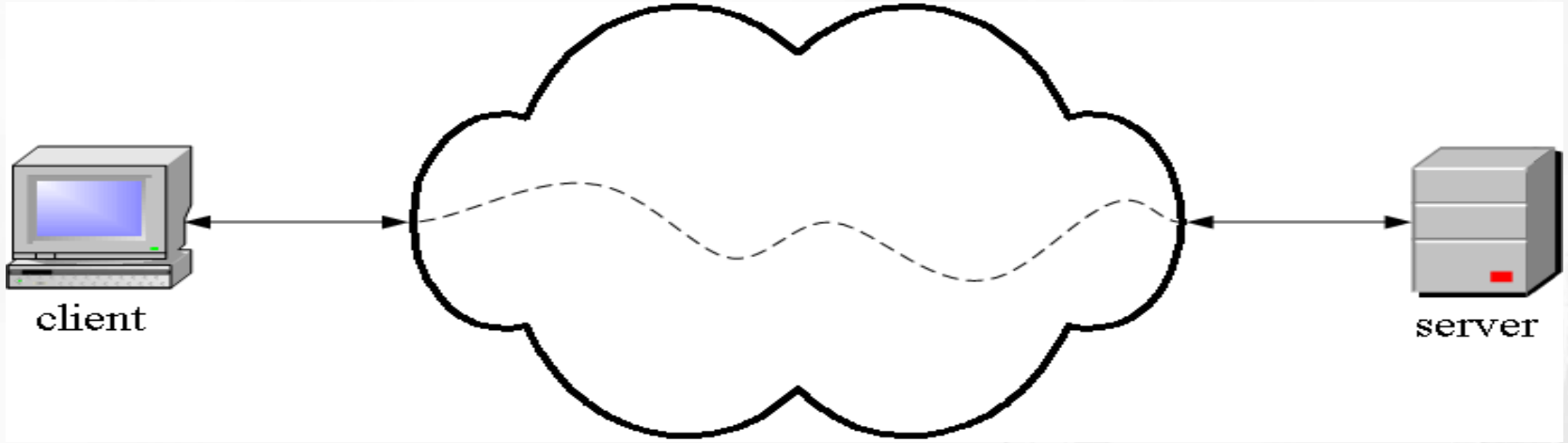
- Hyper Text Transfer Protocol (HTTP)
- Domain Name System (DNS)
- Dynamic Host Control Protocol (DHCP)
- Simple Mail Transfer Protocol: POP3, IMAP, MIME
- File Transfer Protocol (FTP)
- TELNET
- Simple Network Management Protocol (SNMP)

The HyperText Transfer Protocol

- The Hypertext Transfer Protocol (HTTP) is a protocol used mainly to access data on the World Wide Web.

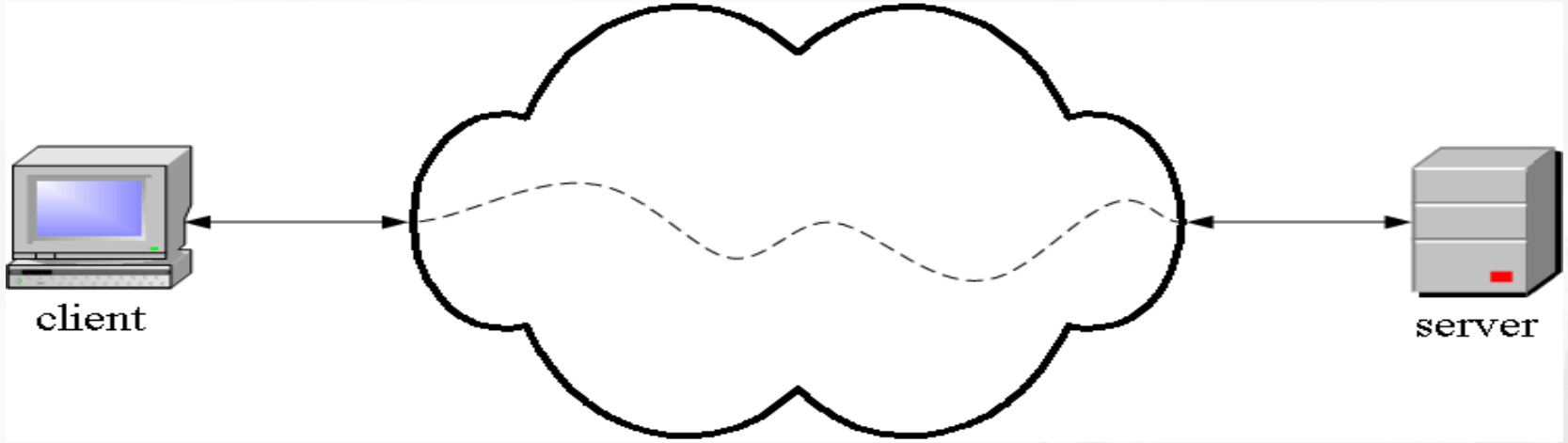
- HTTP layered over bidirectional byte stream
  - Almost always TCP
- Interaction
  - Client sends request to server, followed by response from server to client
  - Requests/responses are encoded in text
- Stateless
  - Server maintains no information about past client requests
  - HTTP uses the services of TCP on well-known port 80

Hypertext Transfer Protocol (HTTP)



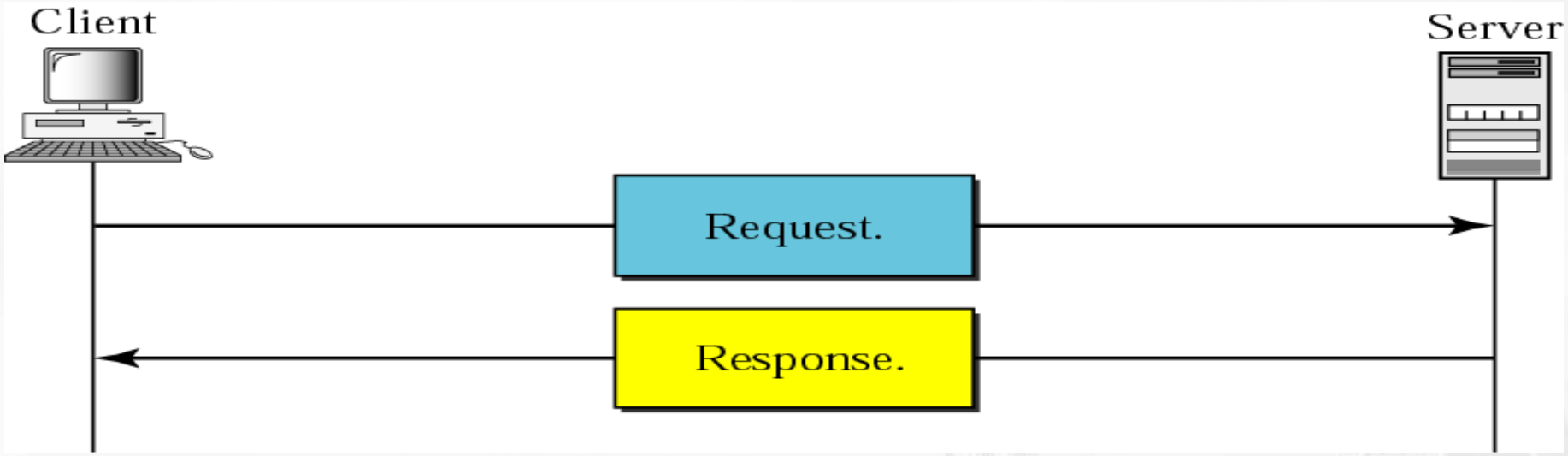
- HTTP is an application layer protocol
- The Web client and the Web server are application programs
- Application layer programs do useful work like retrieving Web pages, sending and receiving email or transferring files

Hypertext Transfer Protocol (HTTP)



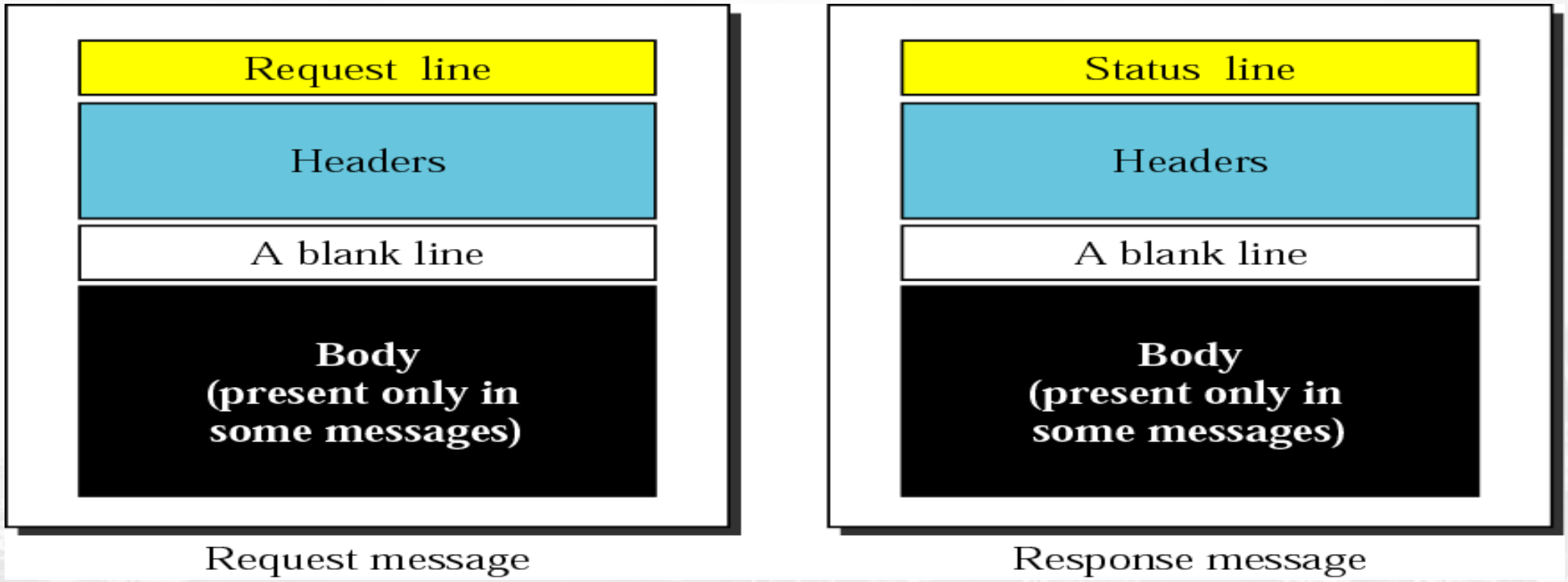
- Lower layers take care of the communication details
- The client and server send messages and data without knowing anything about the communication network

HTTP transaction

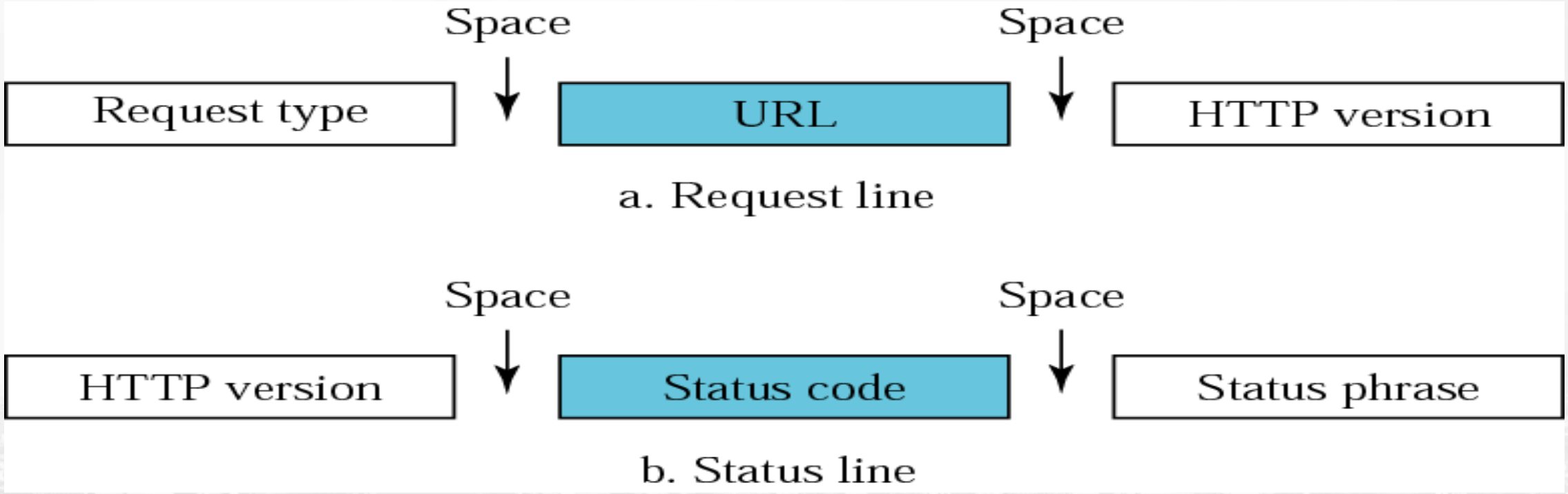


- HTTP uses the services of TCP, HTTP itself is a stateless protocol, which means that the server does not keep information about the client.
- The client initializes the transaction by sending a request. The server replies by sending a response

Request and response messages



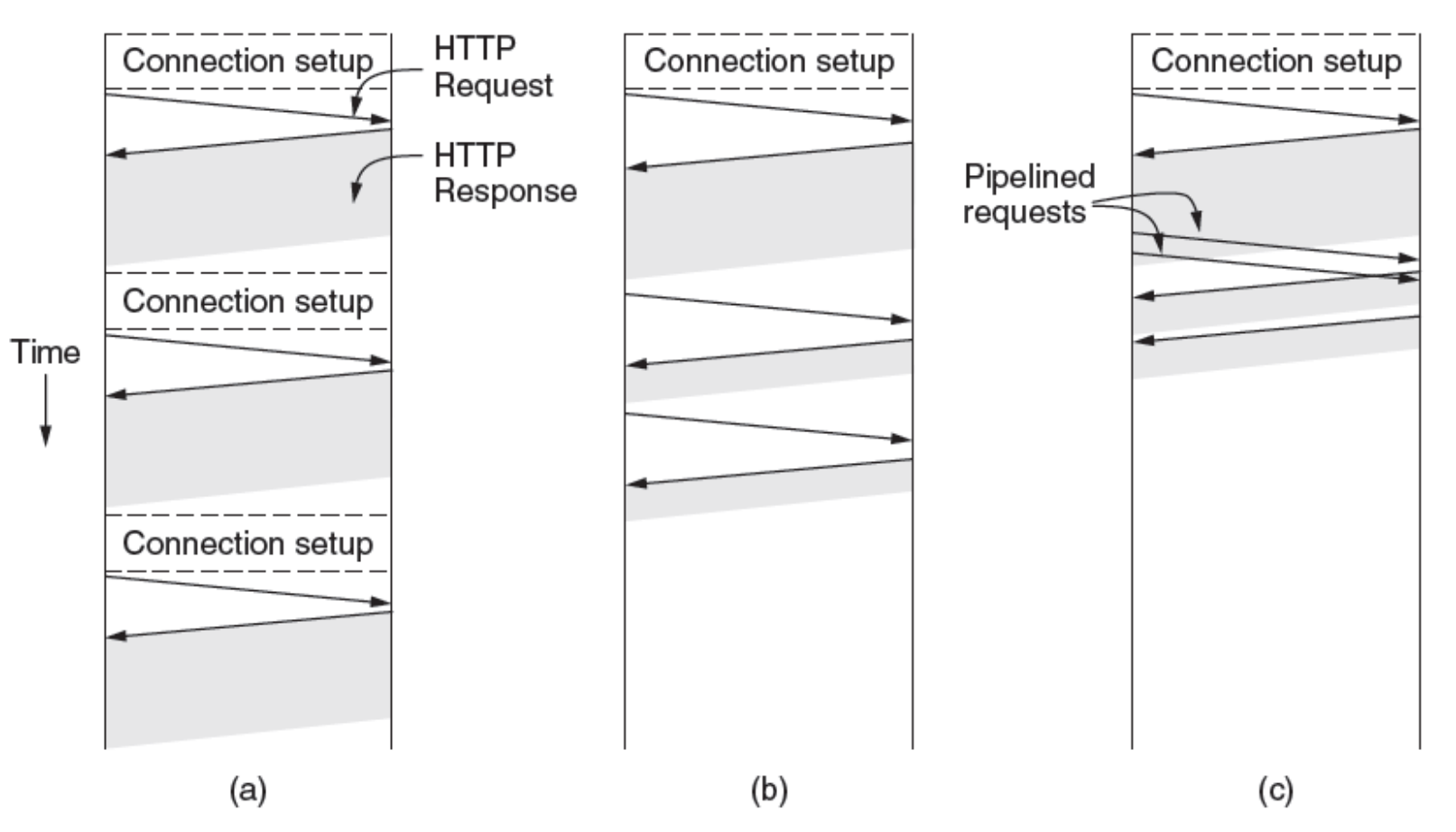
Request and status lines



Difference Between Nonpersistent and *Persistent Connection*

Nonpersistent Connection	***Persistent Connection***
In a nonpersistent connection, one TCP connection is made for each request/response.	The server leaves the connection open for more requests after sending a response.
1. The client opens a TCP connection and sends a request. 2. The server sends the response and closes the connection. 3. The client reads the data until it encounters an end-of-file marker; it then closes the Connection.	HTTP version 1.1 specifies a **persistent connection by default**
	server can close the connection at the request of a client or if a time-out has been reached.

HTTP connections and requests.



HTTP with (a) multiple connections and sequential requests. (b) A persistent connection and sequential requests. (c) A persistent connection and pipelined requests.

The built-in HTTP request methods

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

The built-in HTTP request methods

The status code response groups

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

The status code response groups

HTTP message headers

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
If-Modified-Since	Request	Time and date to check freshness
If-None-Match	Request	Previously sent tags to check freshness
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Referer	Request	The previous URL from which the request came
Cookie	Request	Previously set cookie sent back to the server
Set-Cookie	Response	Cookie for the client to store
Server	Response	Information about the server

. . .

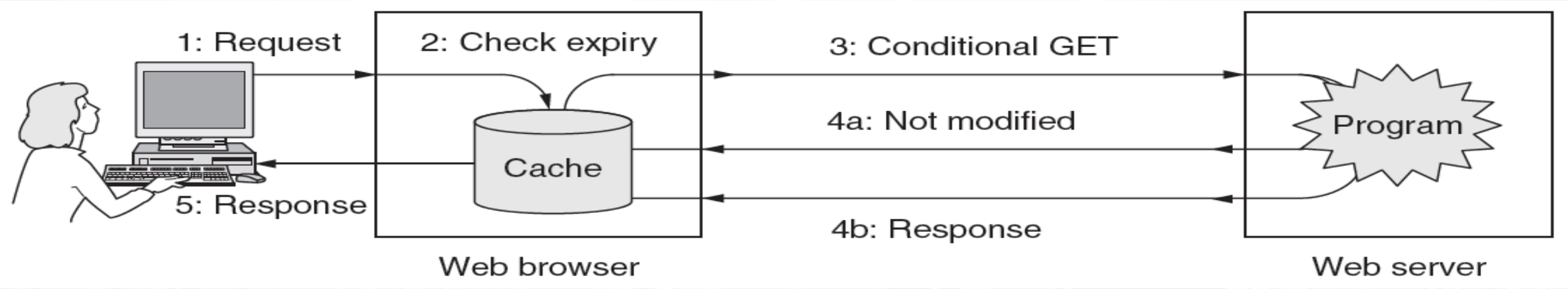
Some HTTP message headers.

HTTP message headers

Content-Encoding	Response	How the content is encoded (e.g., *gzip*)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Content-Range	Response	Identifies a portion of the page's content
Last-Modified	Response	Time and date the page was last changed
Expires	Response	Time and date when the page stops being valid
Location	Response	Tells the client where to send its request
Accept-Ranges	Response	Indicates the server will accept byte range requests
Date	Both	Date and time the message was sent
Range	Both	Identifies a portion of a page
Cache-Control	Both	Directives for how to treat caches
ETag	Both	Tag for the contents of the page
Upgrade	Both	The protocol the sender wants to switch to

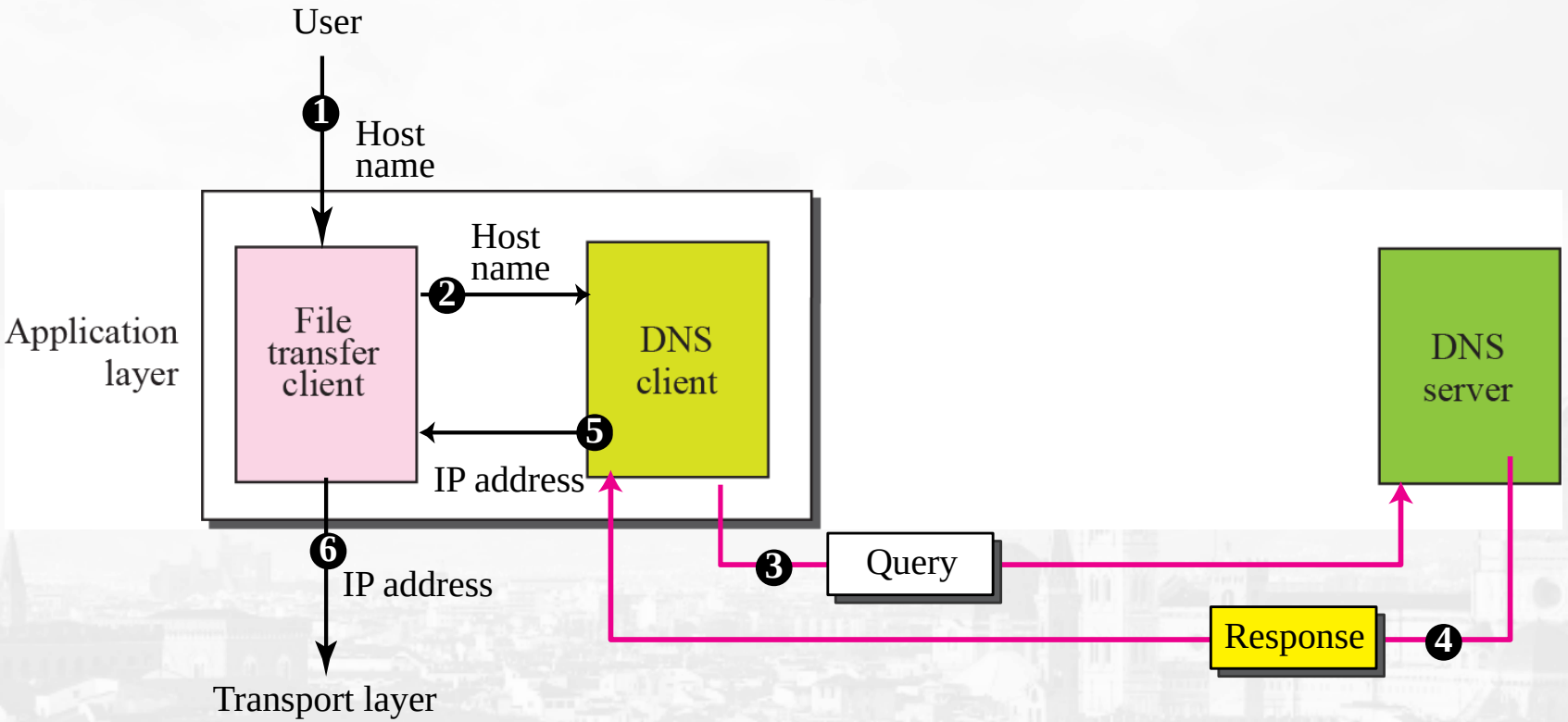
Some HTTP message headers

The HyperText Transfer Protocol Caching



HTTP caching

Domain Name System



Domain Name System (DNS)

Need for DNS:

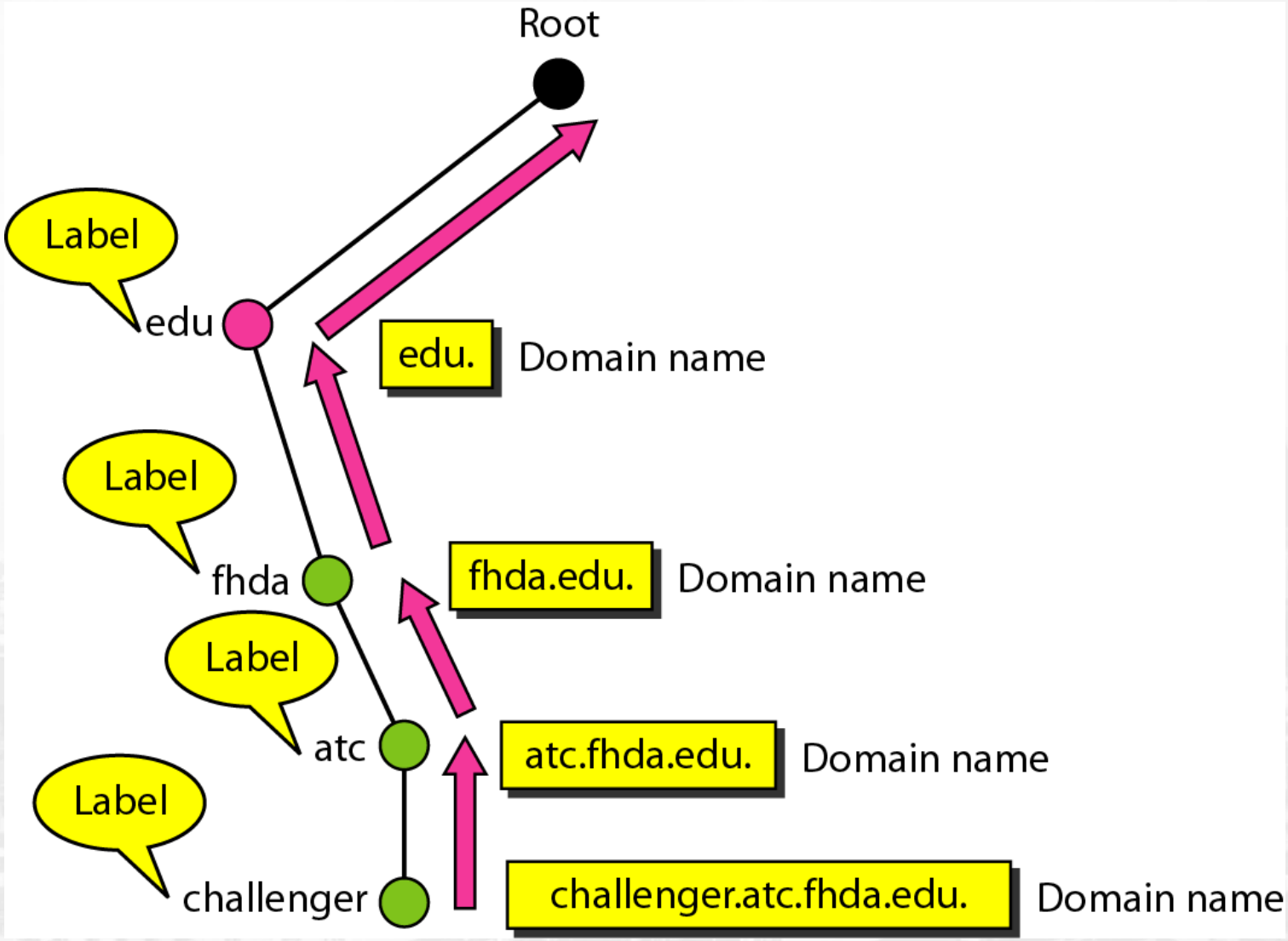
To identify an entity, TCP/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet. However, people prefer to use names instead of numeric addresses. Therefore, we need a system that can map a name to an address or an address to a name.

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.

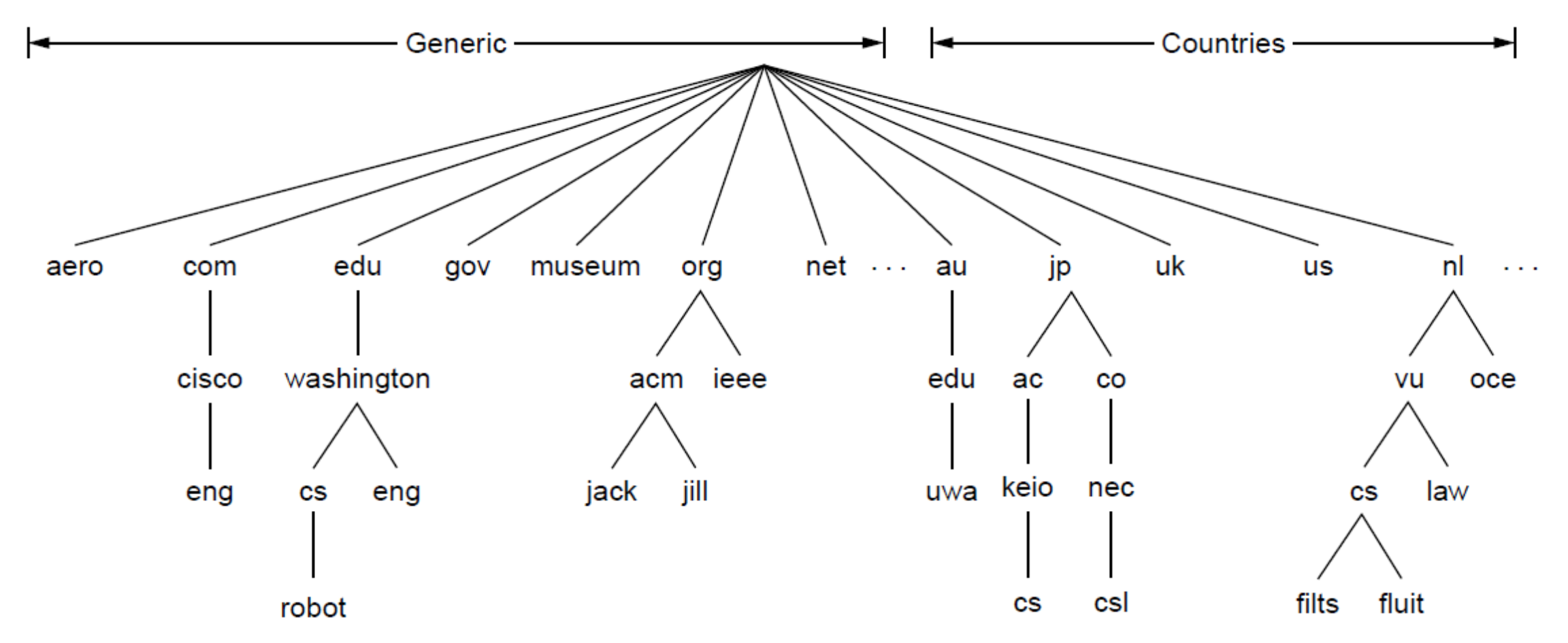
Domain Name System (DNS)

- DNS can use the services of UDP or TCP using the well-known port 53
- The DNS name space
- Domain Resource records
- Name servers

Domain Names and Labels



A portion of the Internet domain name space



Generic top-level domains

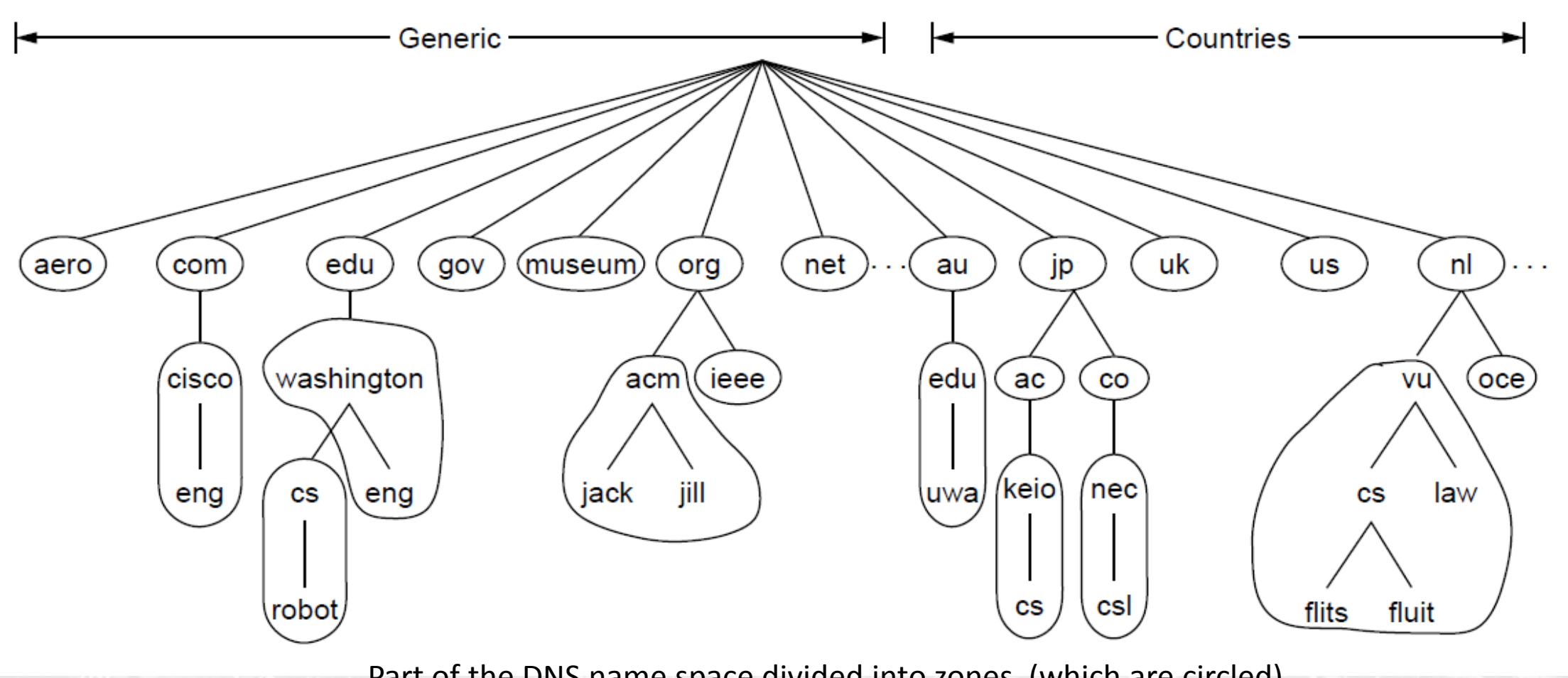
Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Domain Resource Records

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

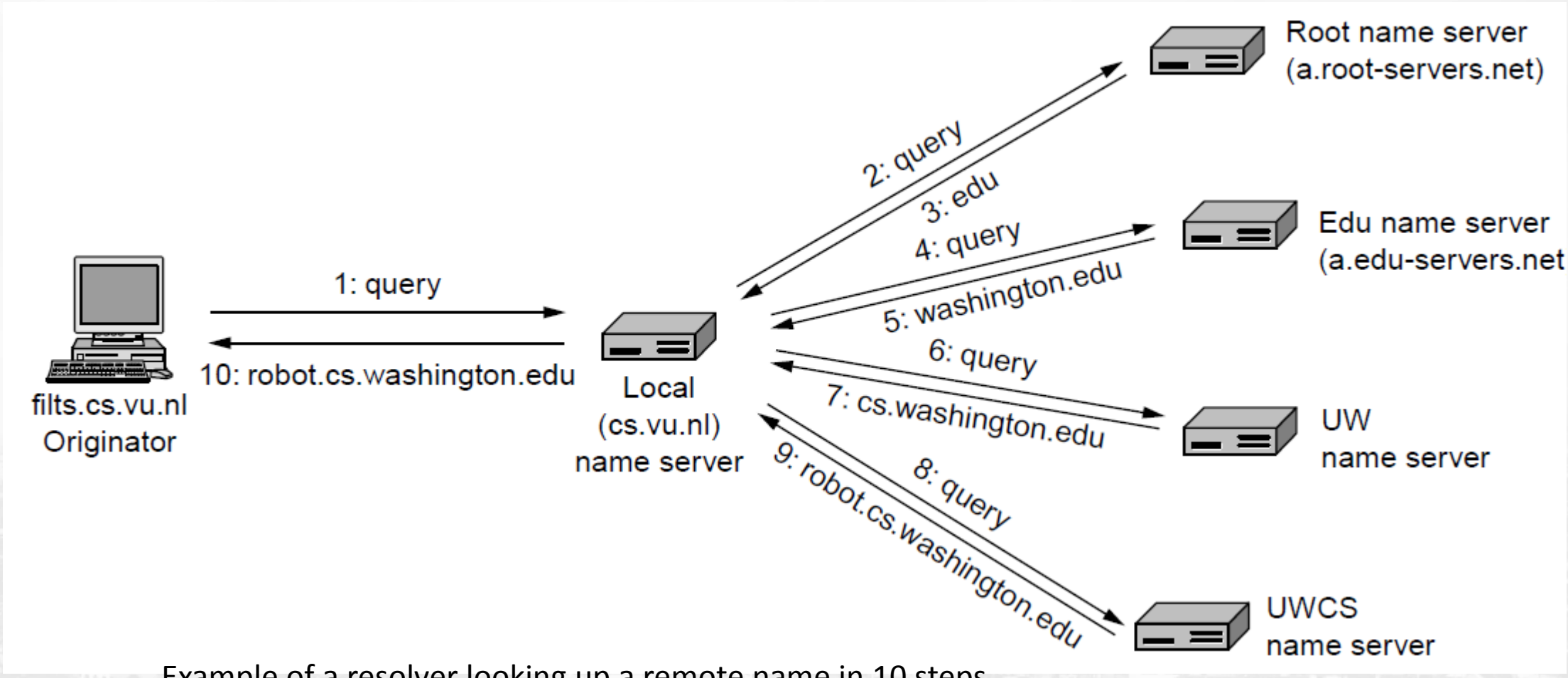
The principal DNS resource record types

Part of the DNS name space divided into zones



Part of the DNS name space divided into zones (which are circled).

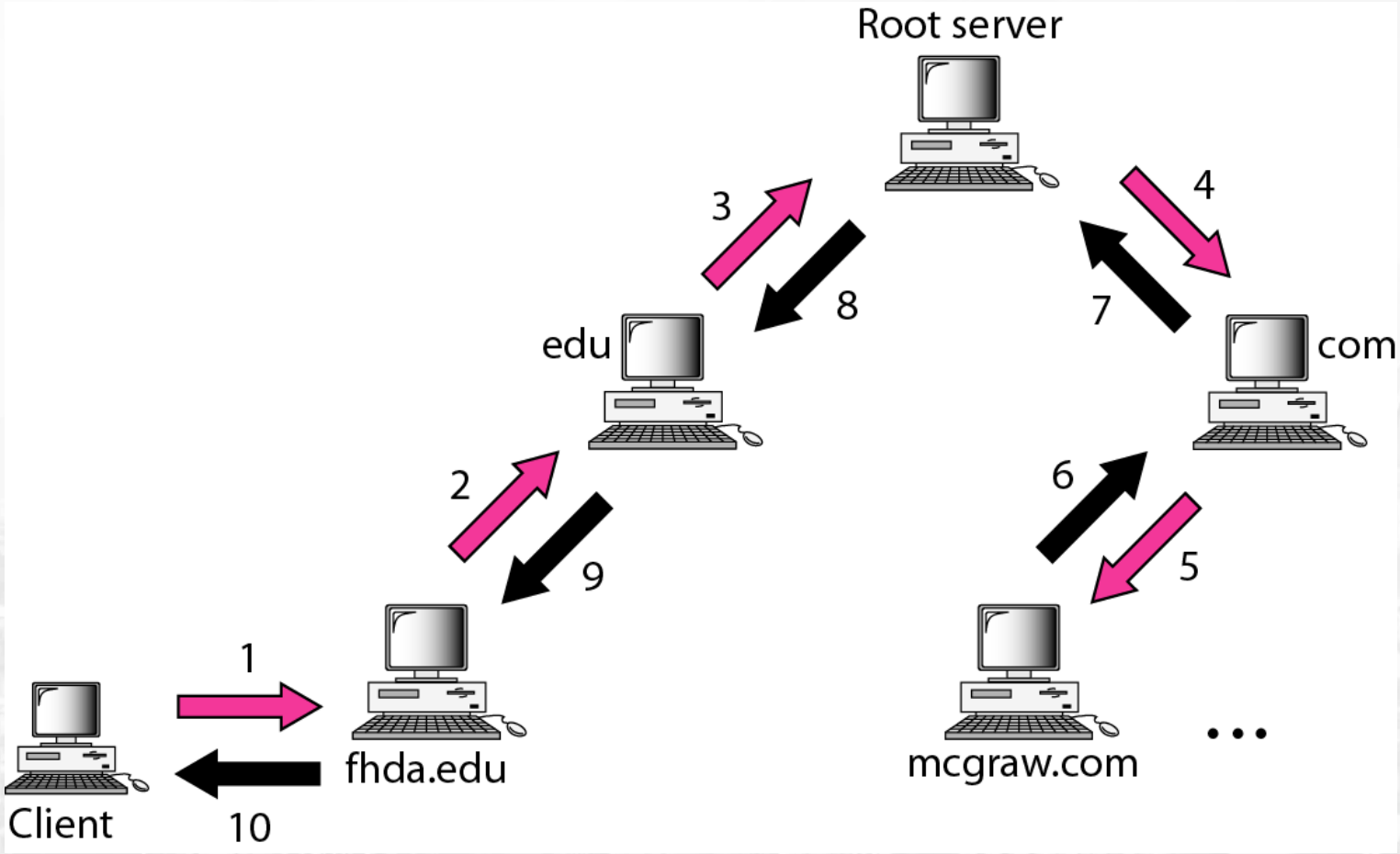
Name Servers (2): Example of a resolver looking up a remote name in 10 steps



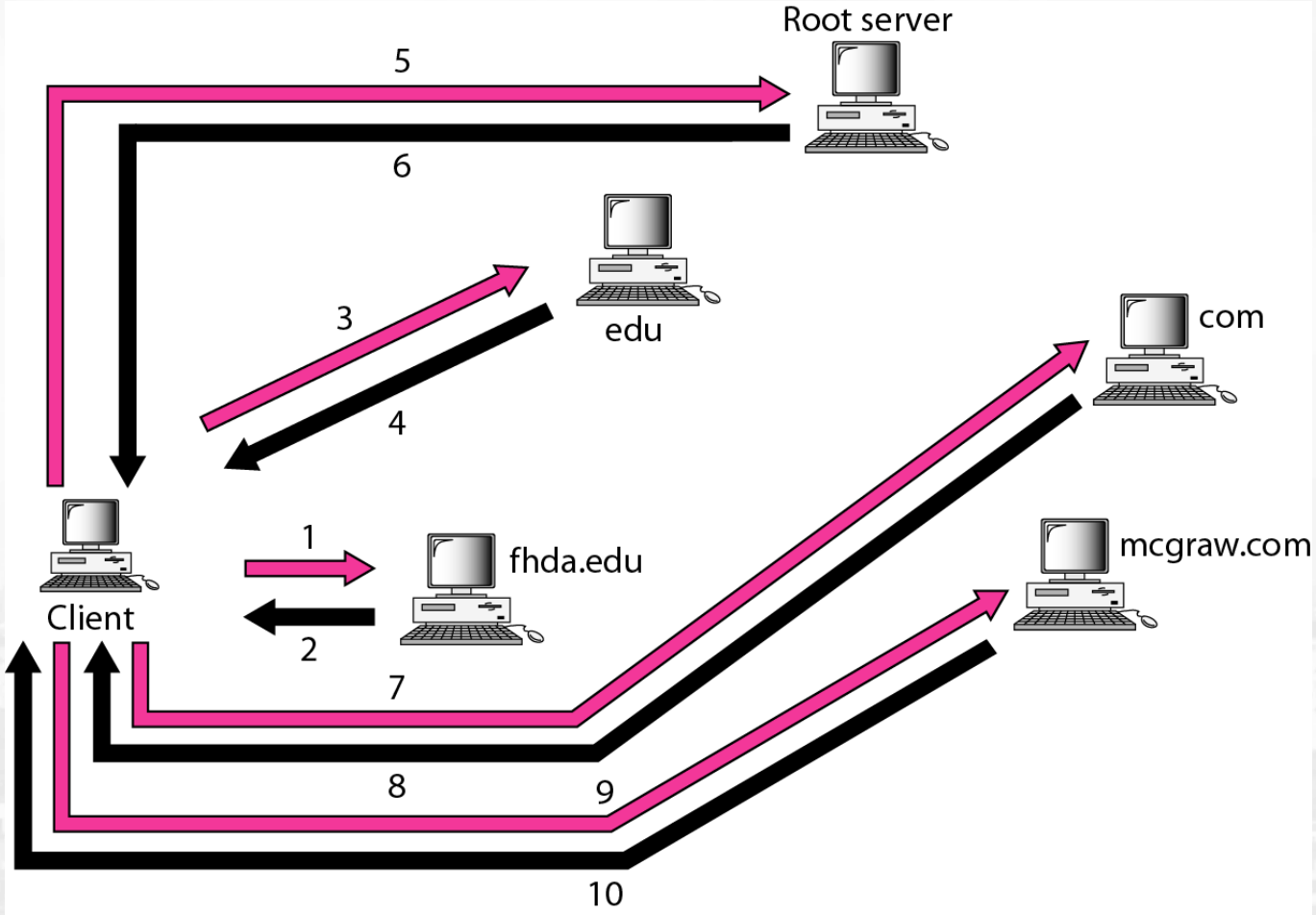
Example of a resolver looking up a remote name in 10 steps

Recursive resolution

- Mapping a name to an address or an address to a name is called name-address resolution



Iterative resolution



DNS MESSAGES

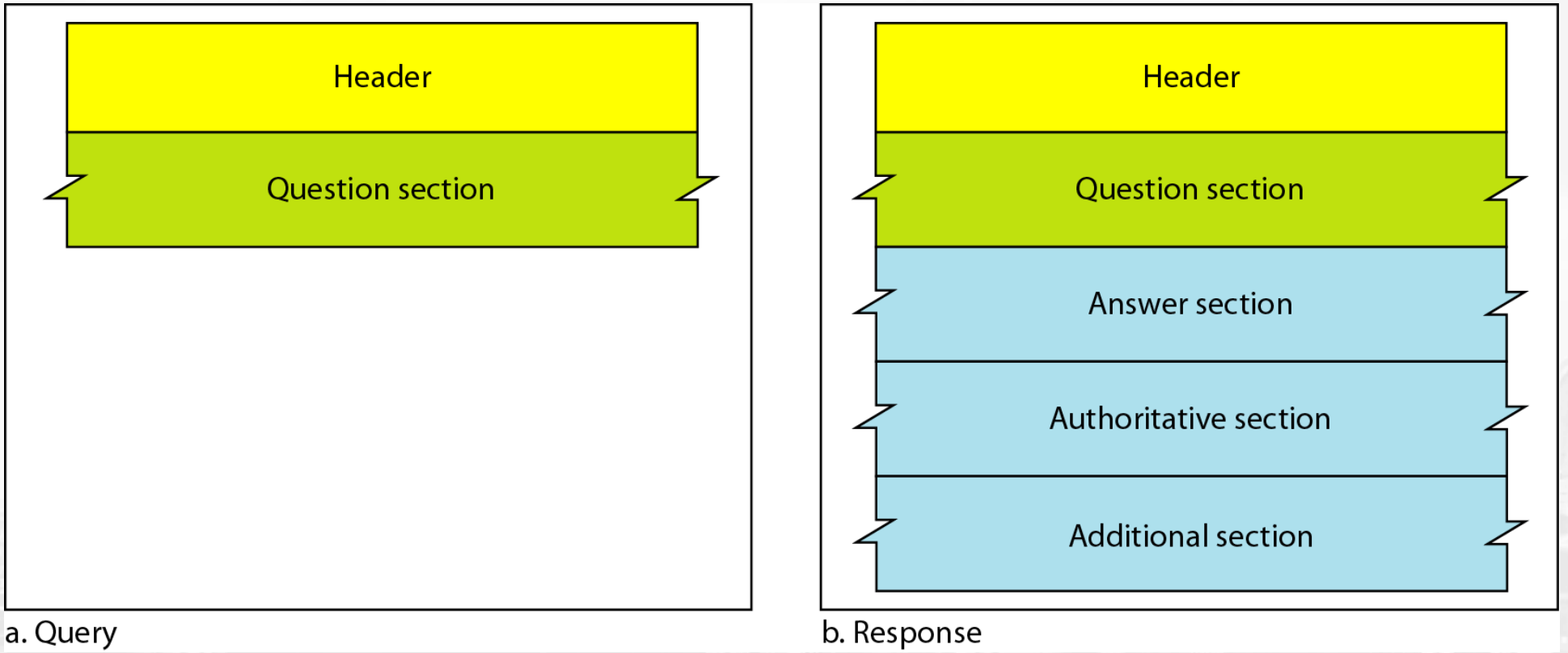


Figure: Query and response messages

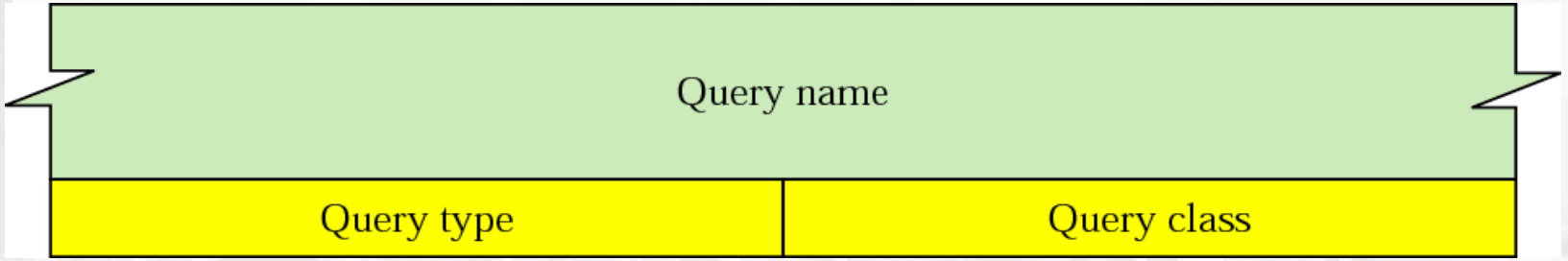
Header format

- Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes

Identification	Flags
Number of question records	Number of answer records (all 0s in query message)
Number of authoritative records (all 0s in query message)	Number of additional records (all 0s in query message)

TYPES OF RECORDS

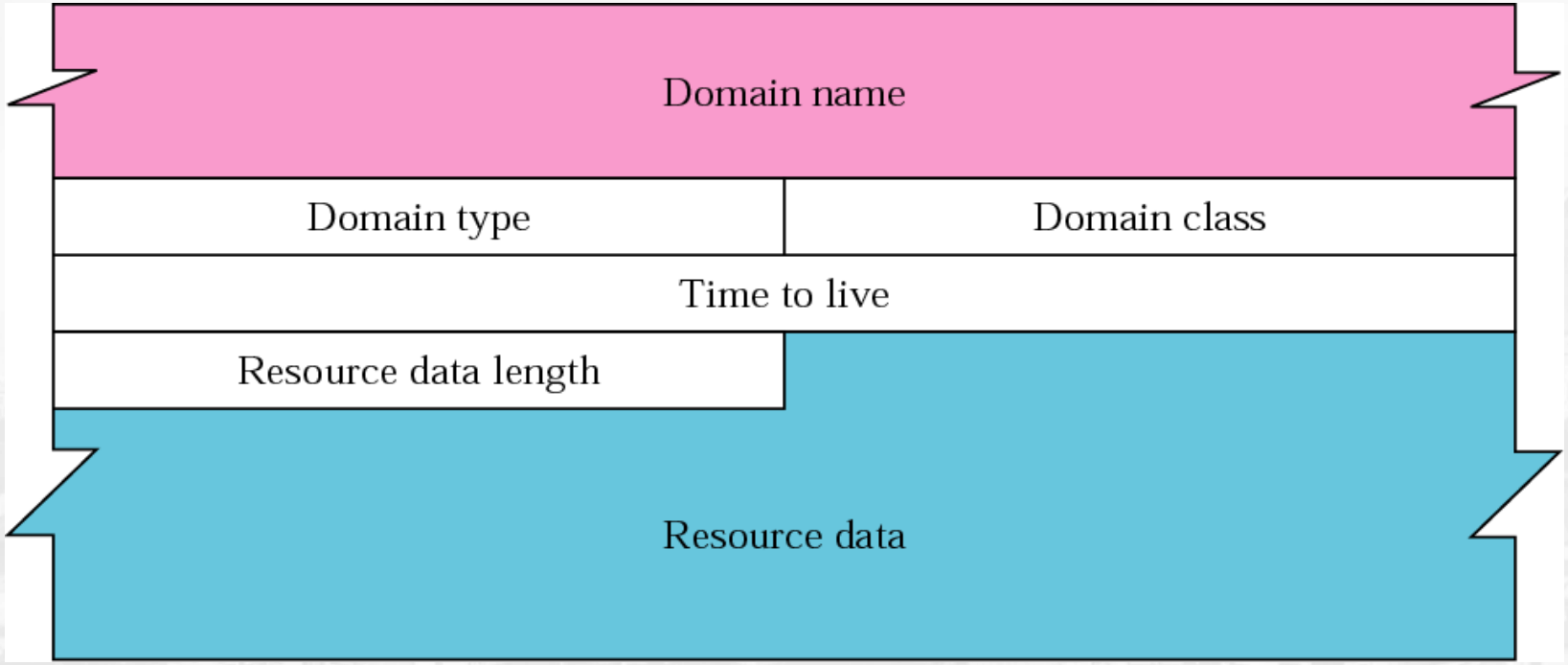
- *Two types of records are used in DNS*
- *The question records are used in the question section of the query and response messages*



Question record format

Resource record format

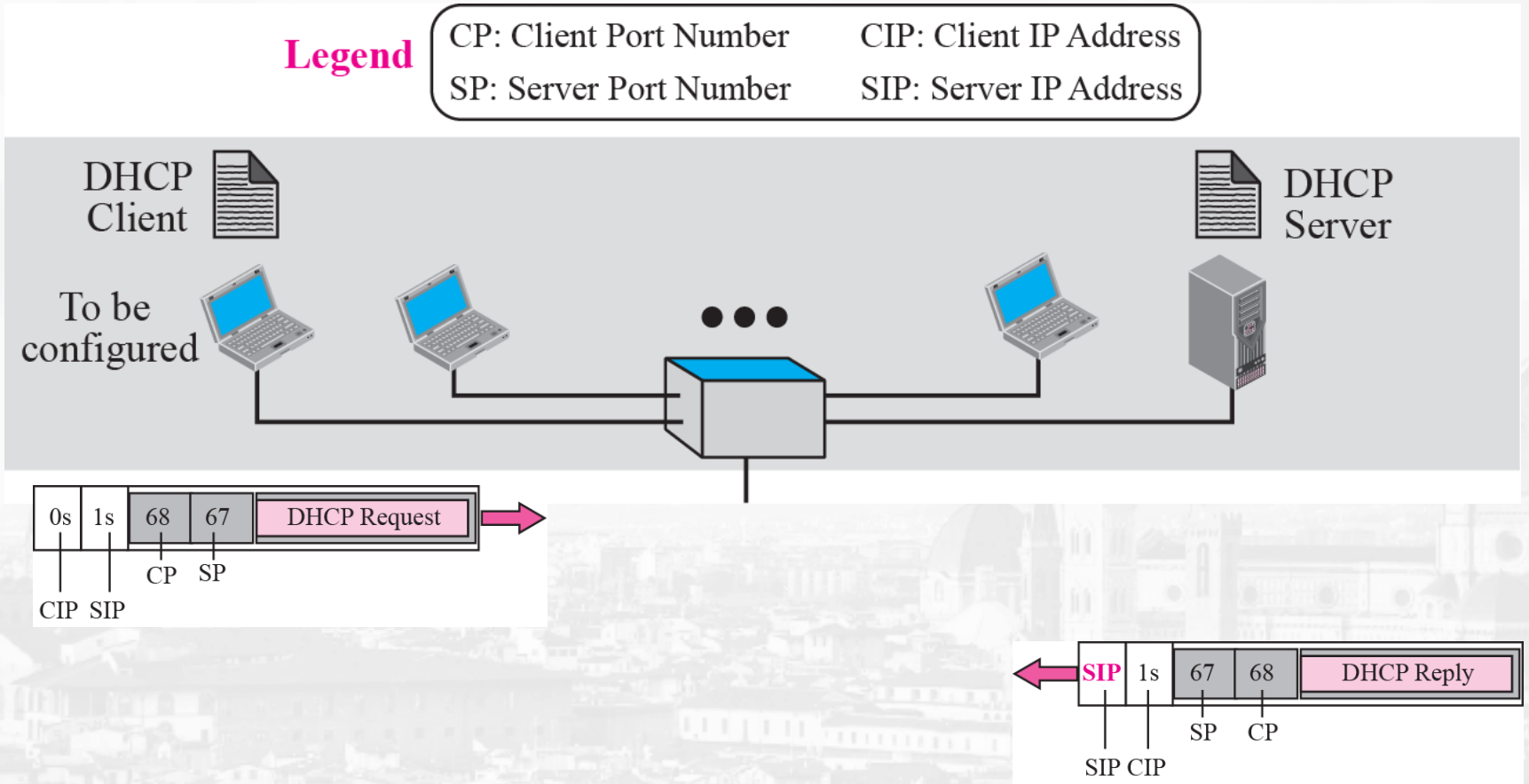
- The resource records are used in the answer, authoritative, and additional information sections of the response message.



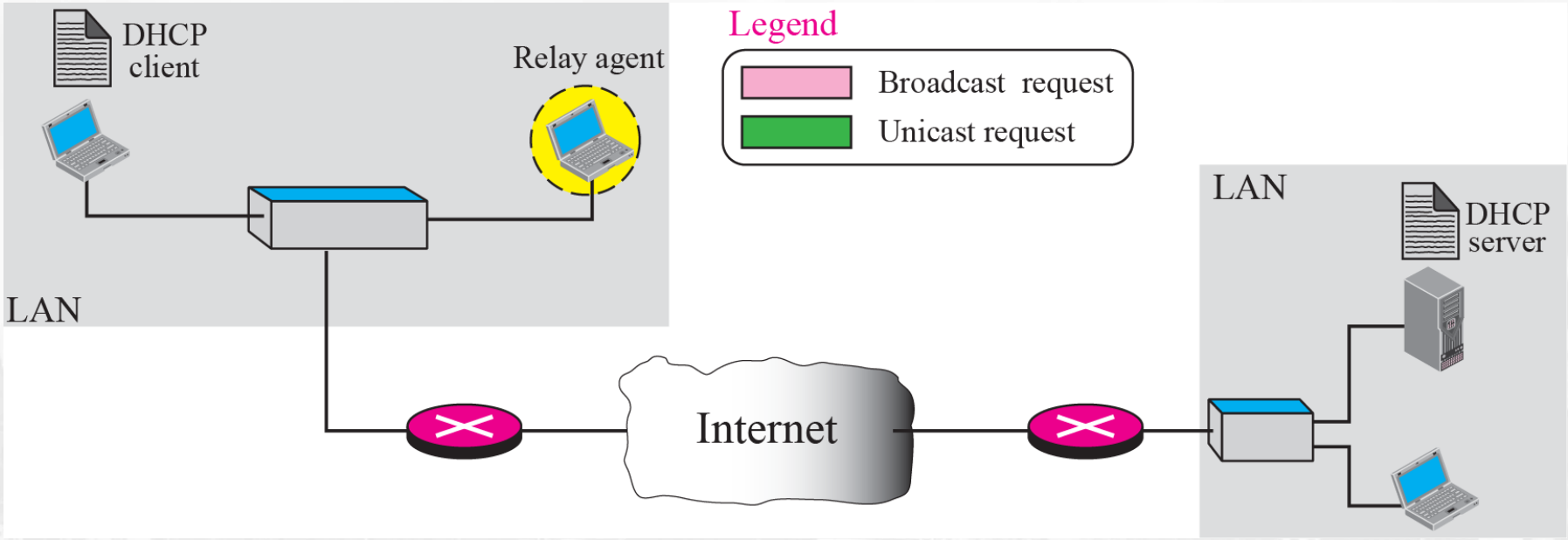
Dynamic Host Control Protocol (DHCP)

- Each computer that uses the TCP/IP protocol suite needs to know its IP address. If the computer uses classless addressing or is a member of a subnet, it also needs to know its subnet mask
- four pieces of information are normally needed: 1. The IP address of the computer 2. The subnet mask of the computer 3. The IP address of a router 4. The IP address of a name server
- These four pieces of information can be stored in a configuration file and accessed by the computer during the bootstrap process
- The DHCP client and server can either be on the same network or on different networks

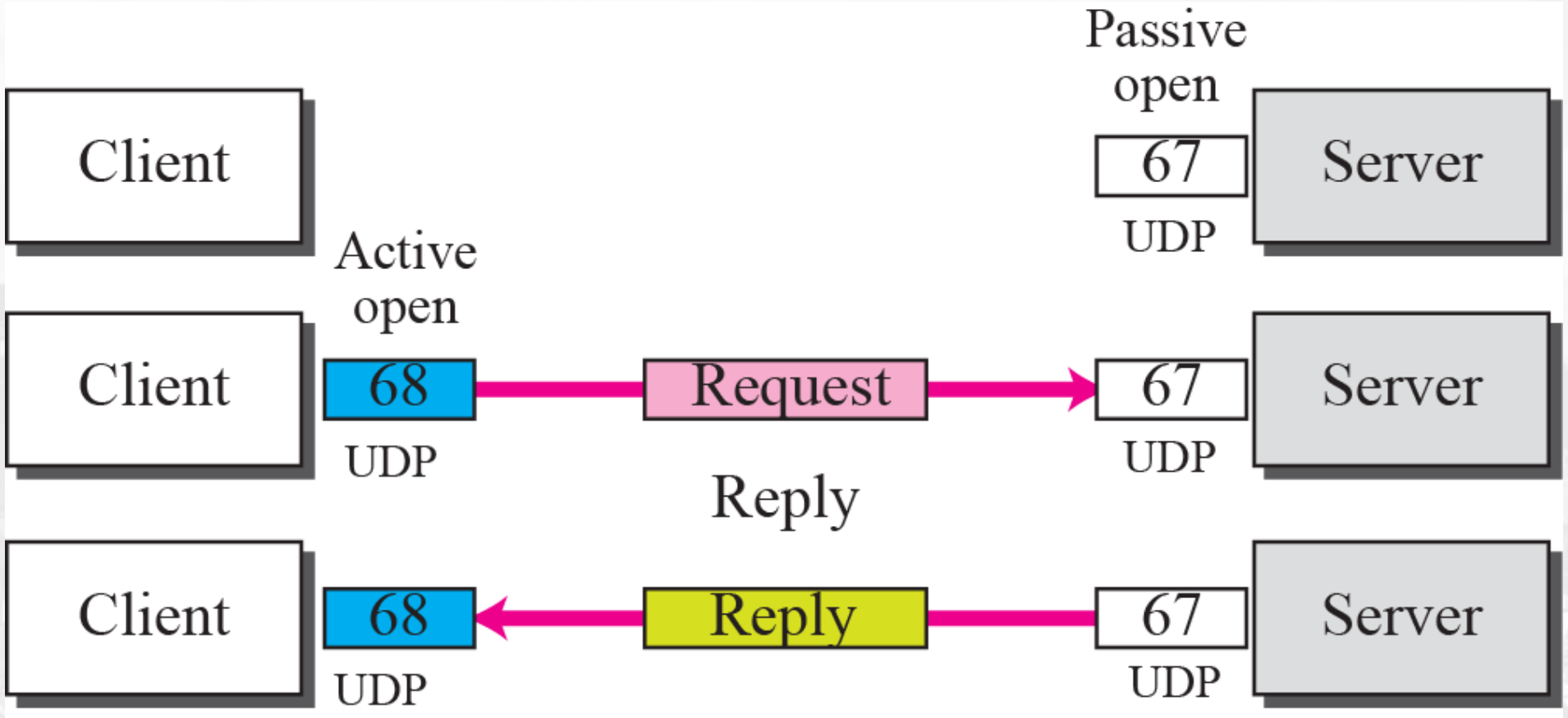
Clent and server on the same network



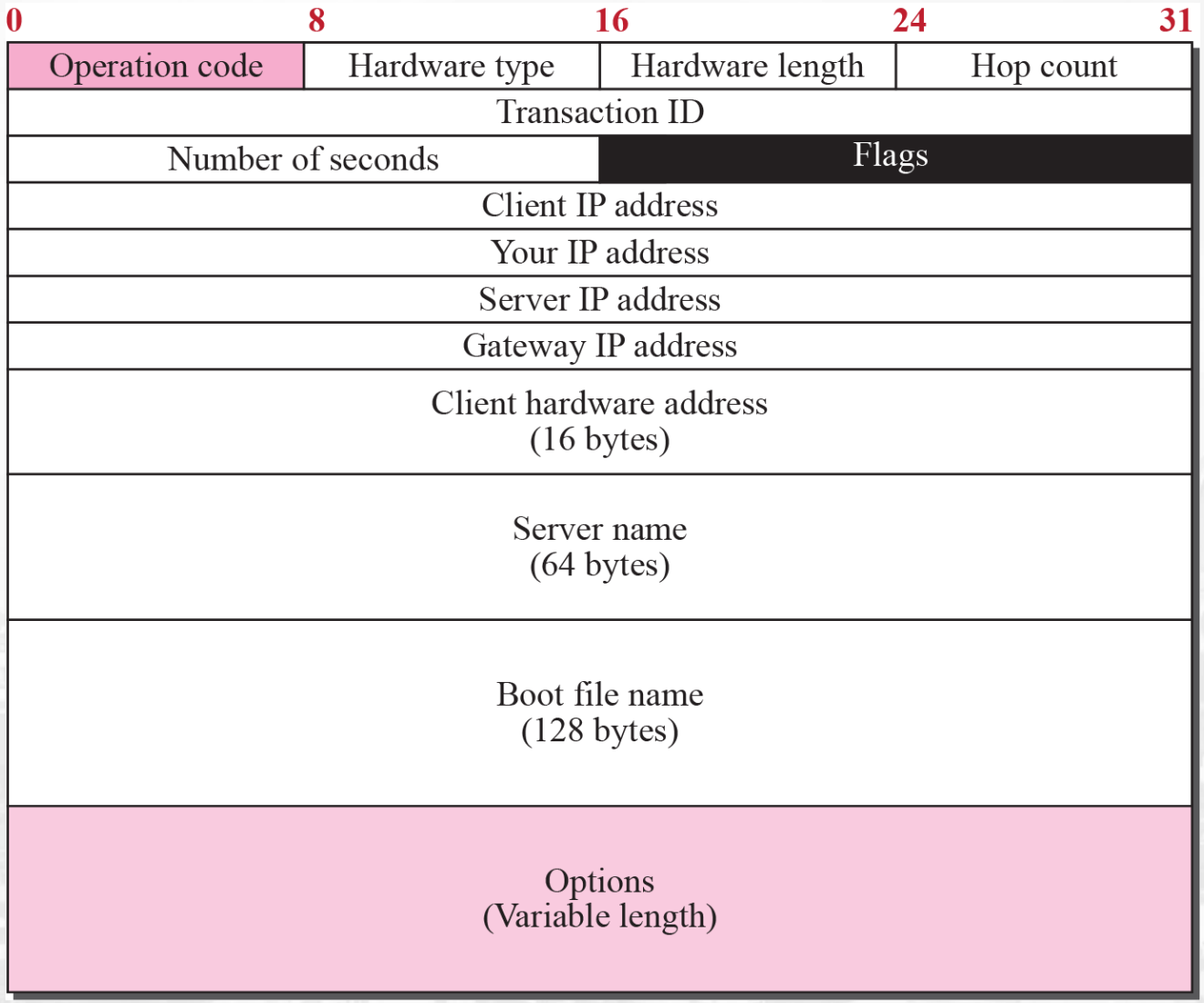
Client and server on two different networks



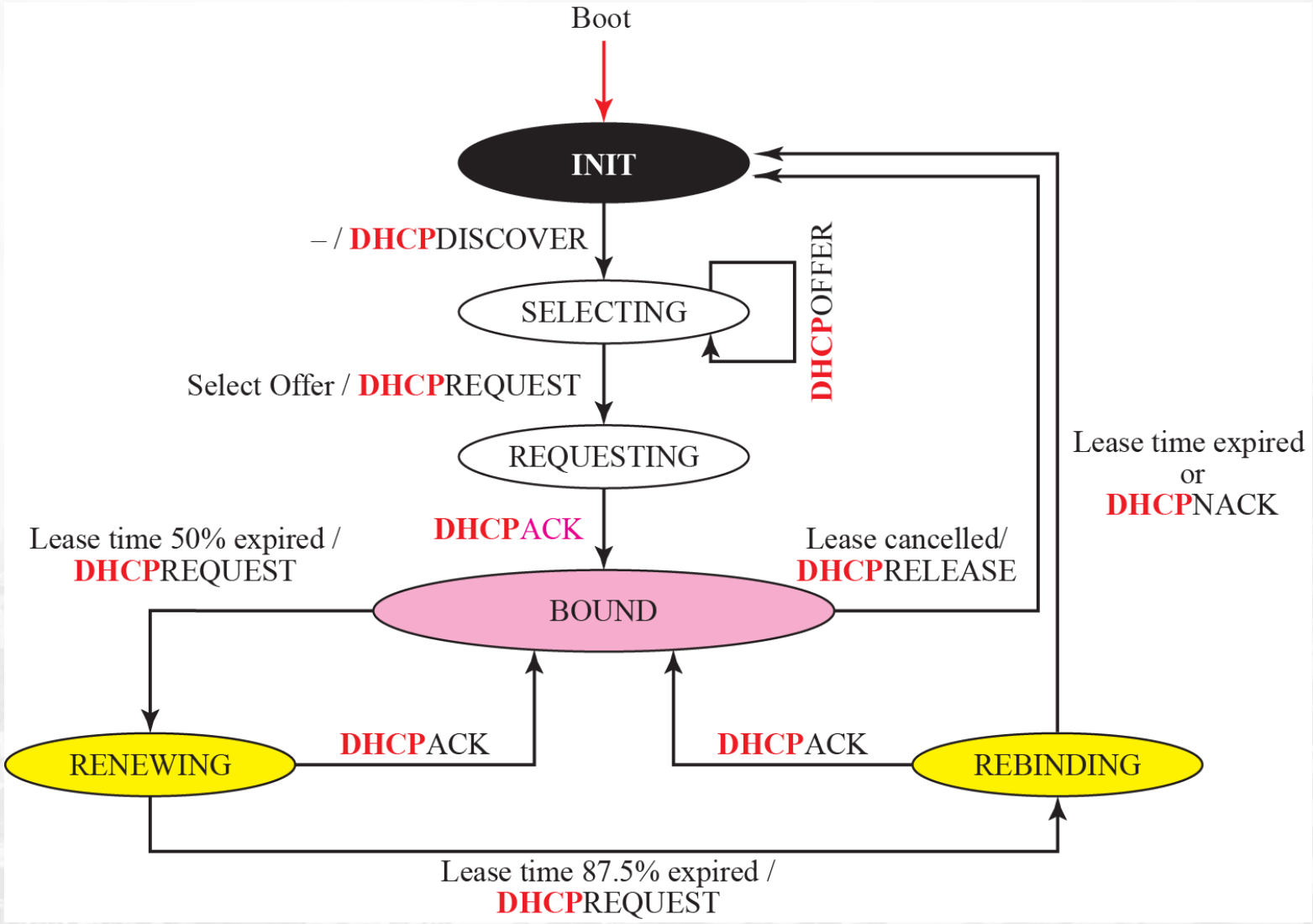
Use of UDP ports



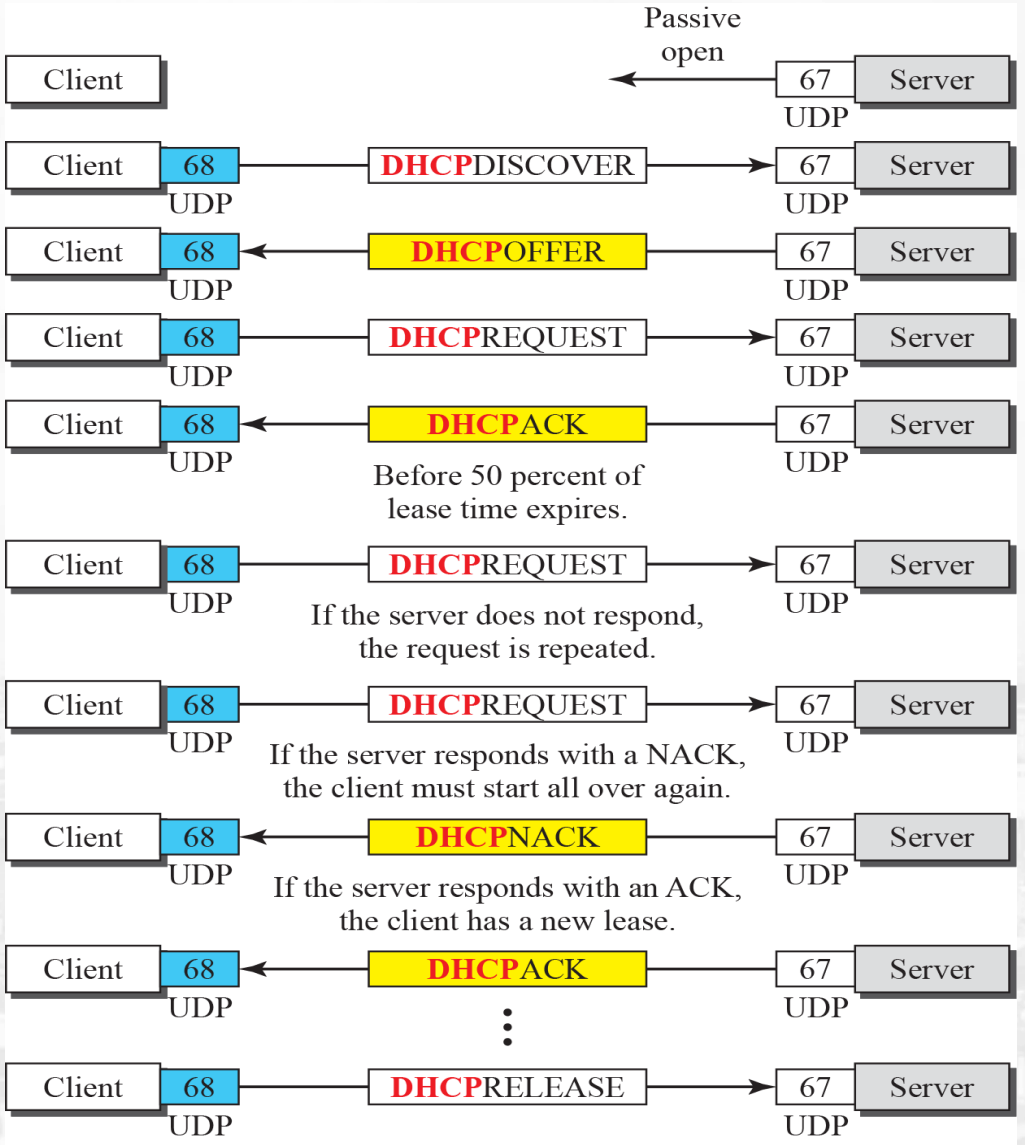
DHCP packet format



DHCP Client Transition Diagram



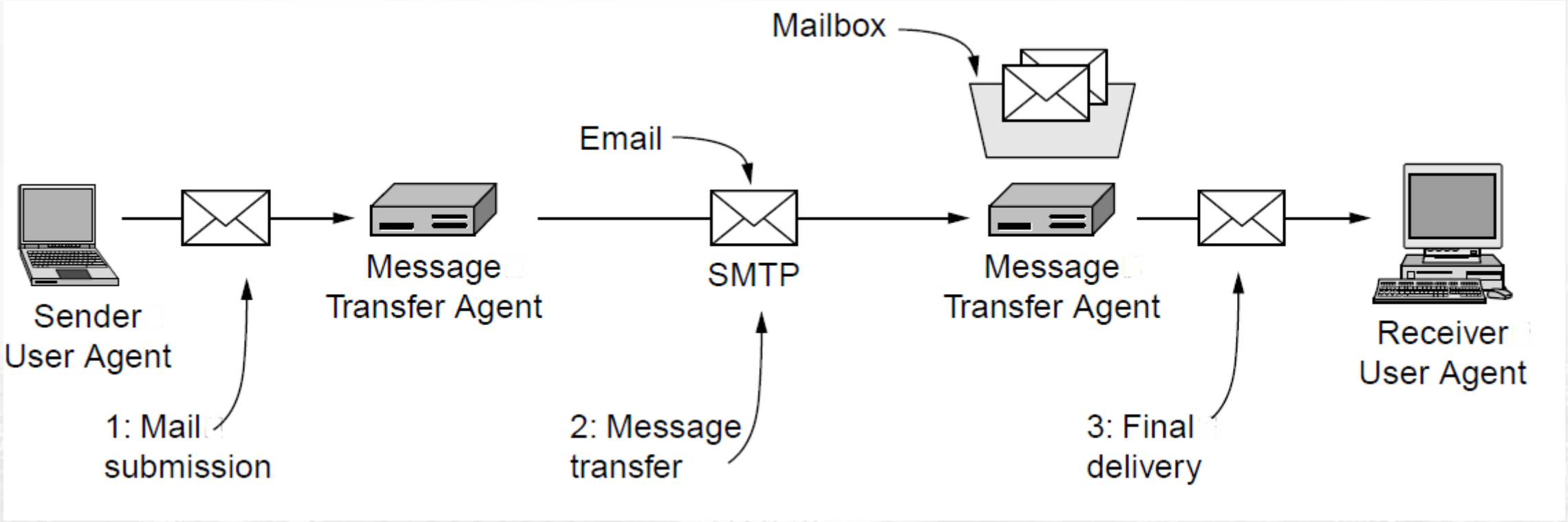
Exchanging messages



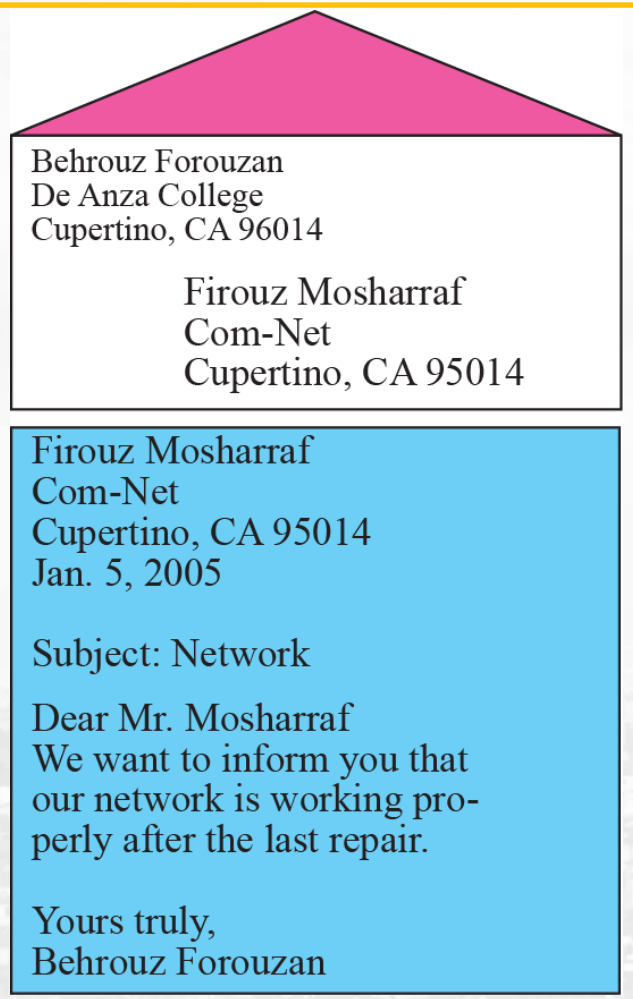
Simple Mail Transfer Protocol: POP3, IMAP, MIME

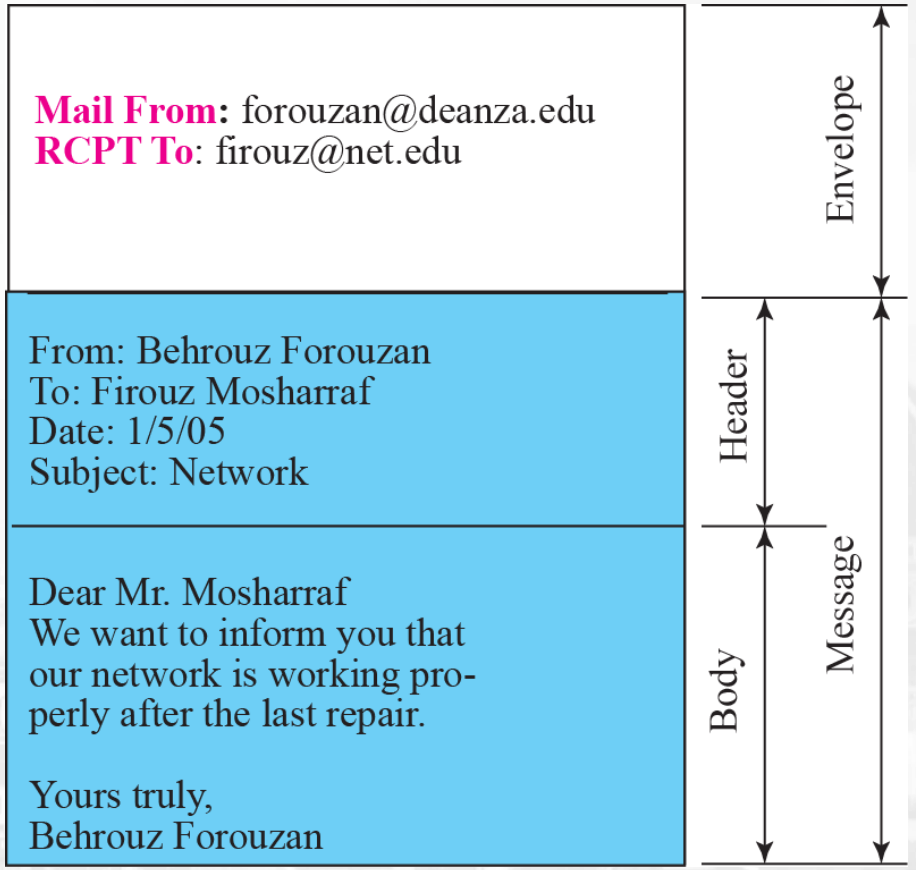
- Architecture and services
- The user agent
- Message formats
- Message transfer
- Final delivery

Architecture and Services



Simple Mail Transfer Protocol: POP3, IMAP, MIME





Format of an email

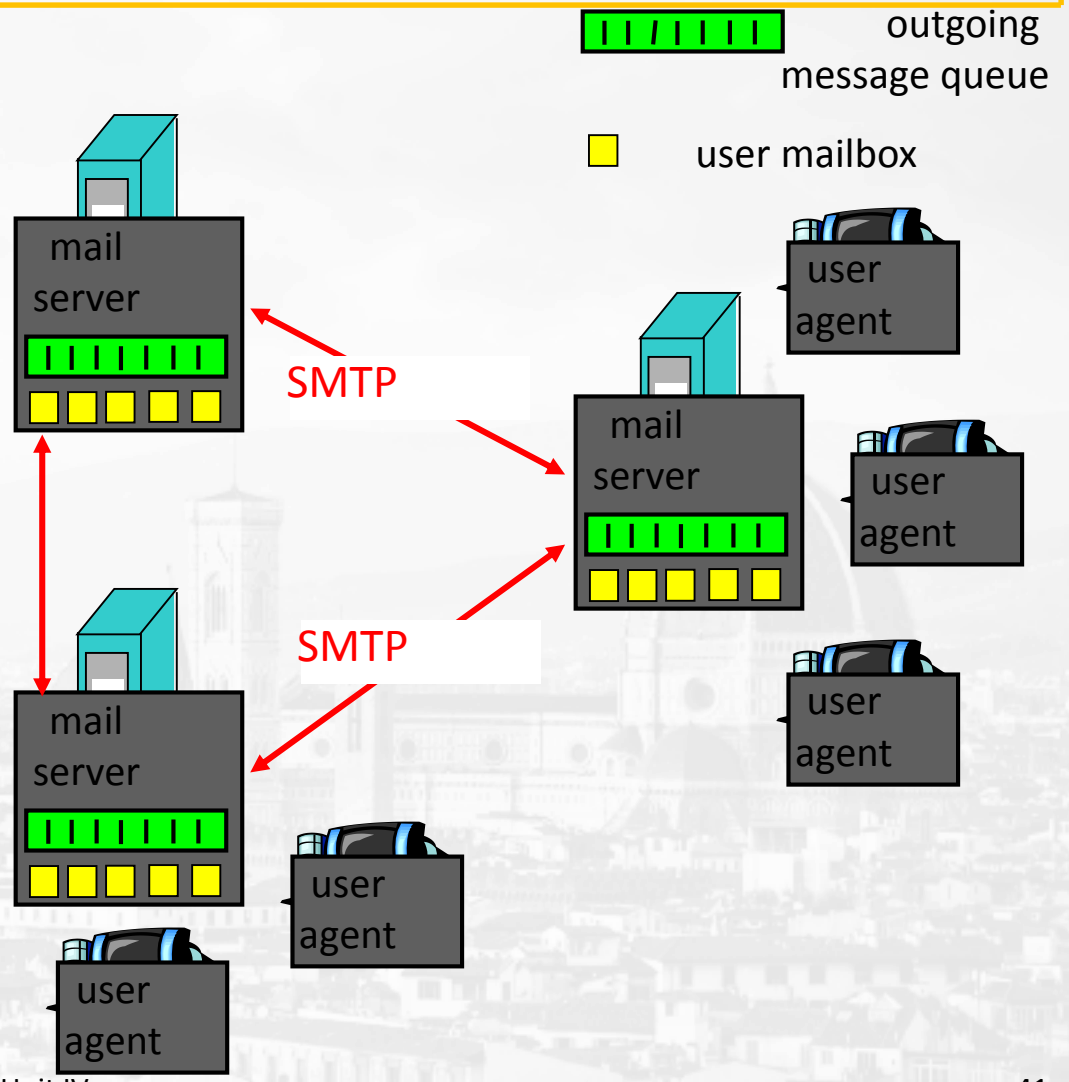
Electronic Mail: SMTP, POP3, IMAP

Three major components:

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

User Agent

- composing, editing, reading mail messages
- e.g., Eudora, Outlook, elm, Mozilla Thunderbird
- outgoing, incoming messages stored on server

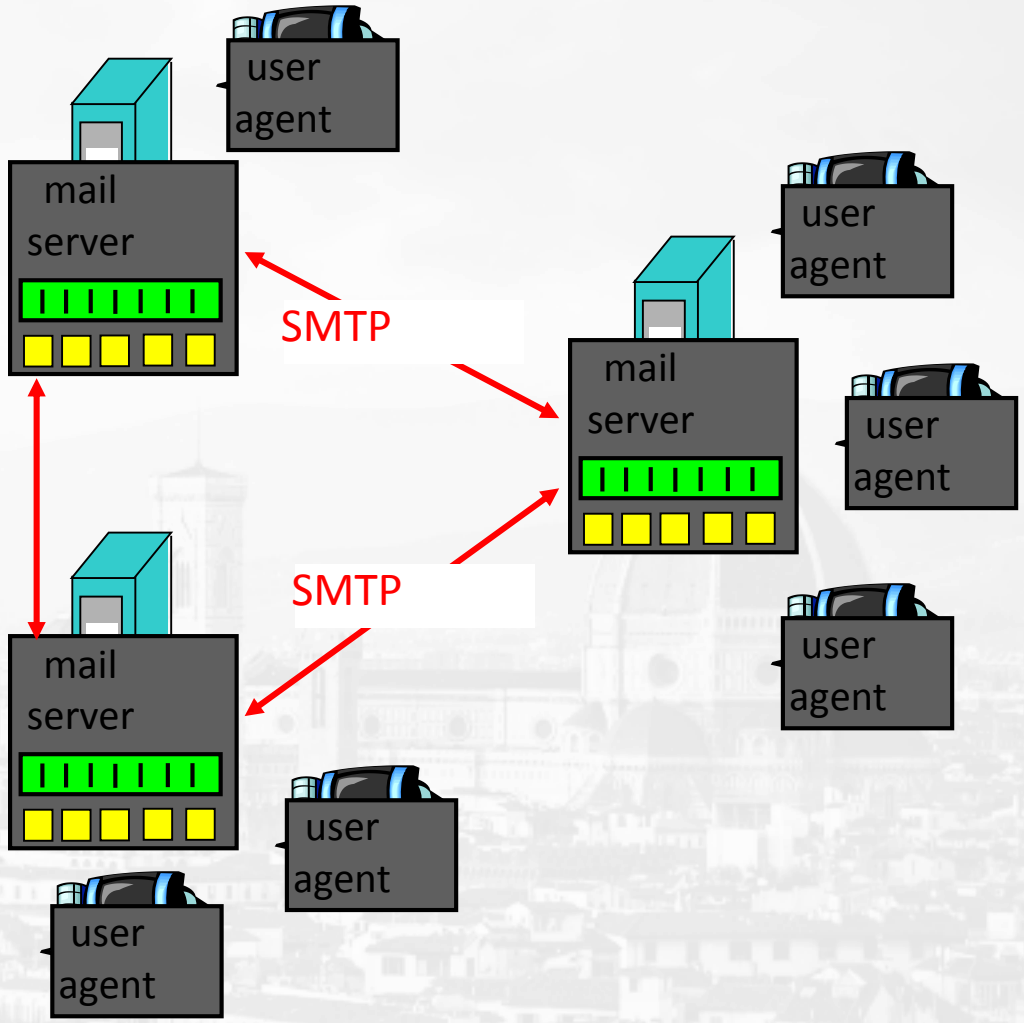


Electronic Mail: mail servers

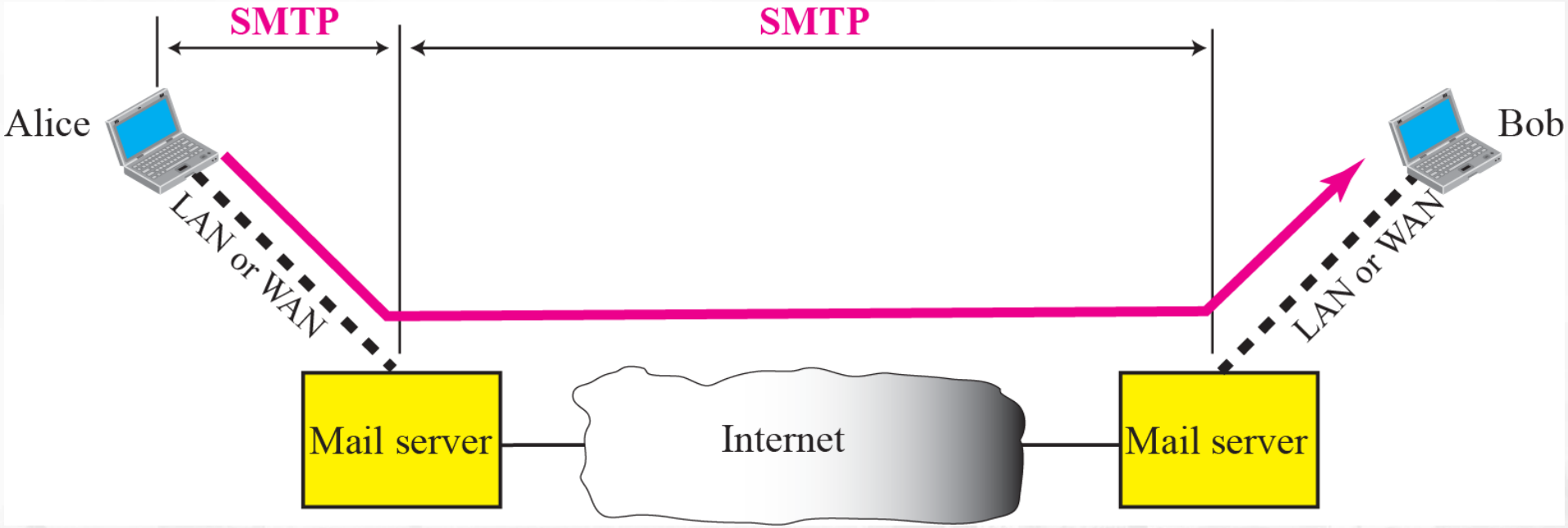
Mail Servers

- 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



SMTP



SMTP is used two times, between the sender and the sender's mail server and between the two mail servers

Electronic Mail: SMTP

- uses TCP to reliably transfer email message from client to server (port 25)
- direct transfer: sending server to receiving server
- three phases of transfer
  - handshaking (greeting)
  - transfer of messages
  - closure
- command/response interaction
  - commands: ASCII text
  - response: status code and phrase
- messages must be in 7-bit ASCII

Scenario: Alice sends message to Bob

1) Alice uses UA to compose message and "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

Comparison with HTTP

- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server uses `CRLF.CRLF` to determine end of message

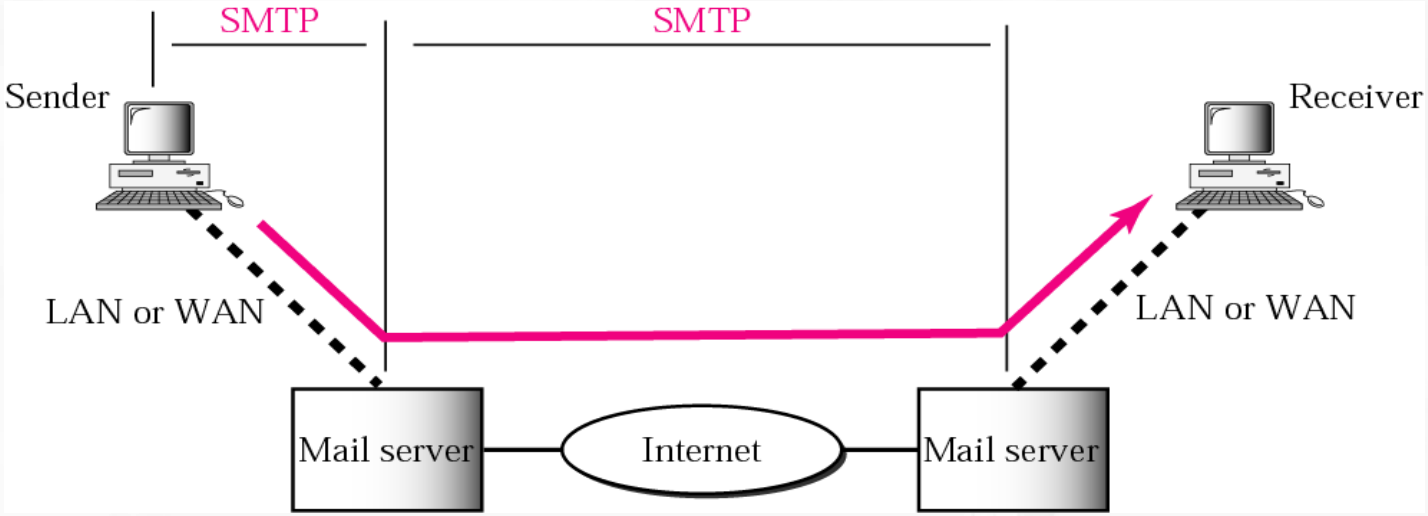
Comparison with HTTP:

- HTTP: pull
- SMTP: push

- both have ASCII command/response interaction, status codes

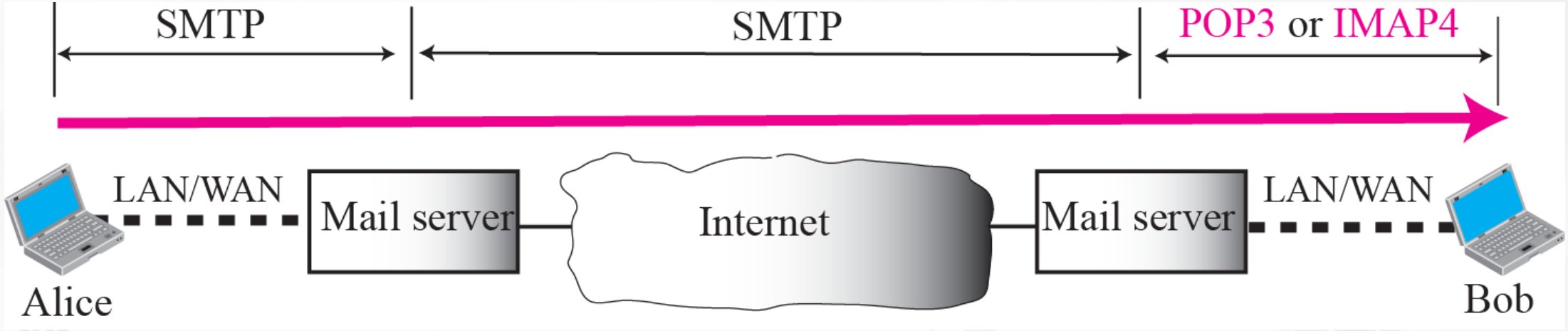
- HTTP: each object encapsulated in its own response message
- SMTP: multiple objects sent in multipart message

SMTP



- 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 direction of the bulk data are from the server to the client.

POP3 and IMAP4



- IMAP4 is similar to POP3, but it has more features;
- IMAP4 is more powerful and more complex.
- POP3 is deficient in several ways. It does not allow the user to organize her mail on the server; the user cannot have different folders on the server.
- In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4 provides the following extra functions

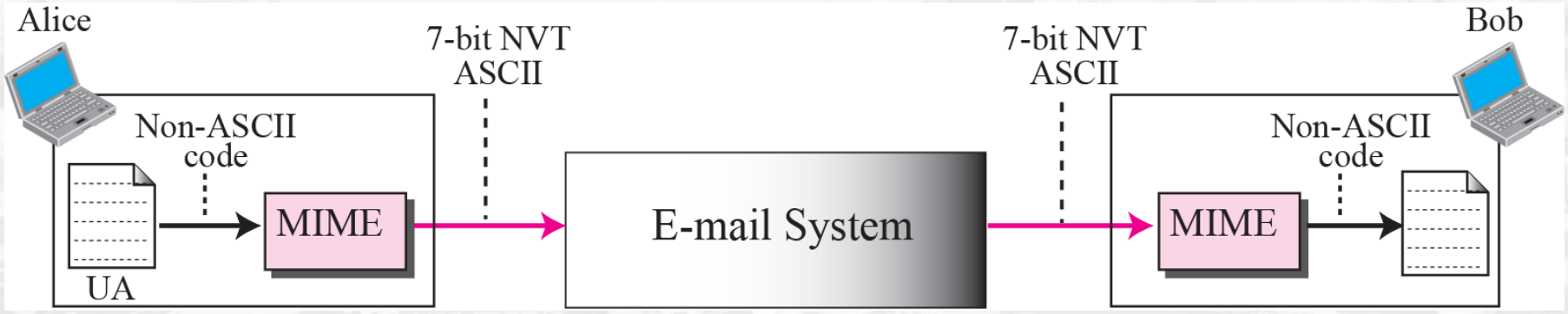
- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth 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.

comparison of POP3 and IMAP

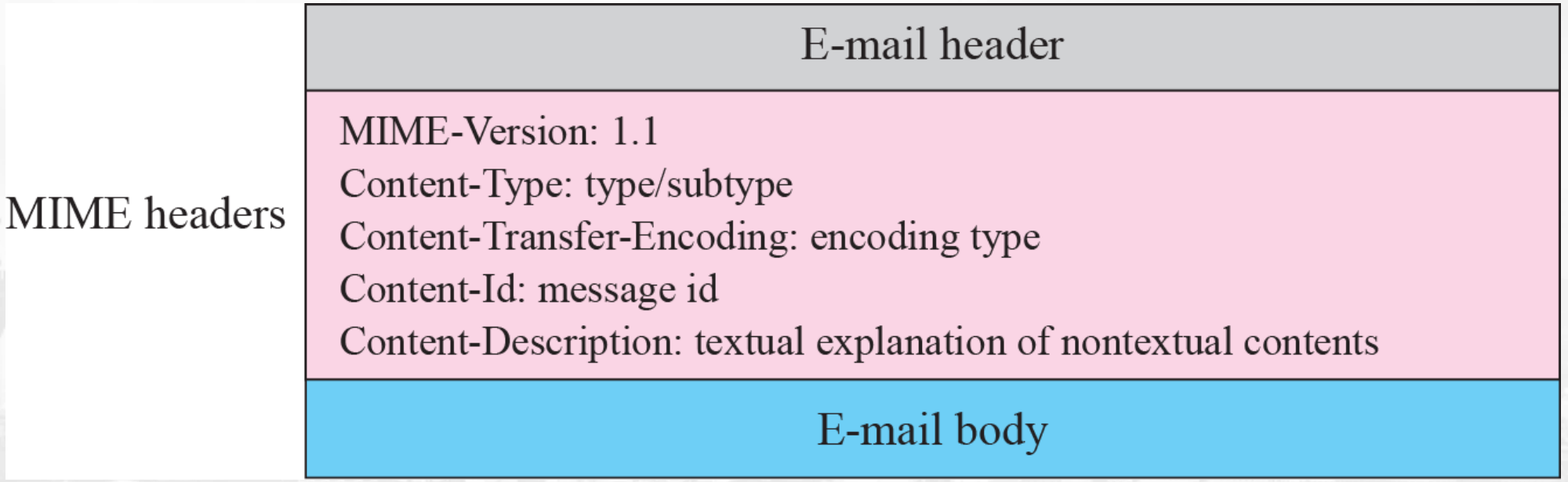
Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is used?	110	143
Where is e-mail stored?	User's PC	Server
Where is e-mail read?	Off-line	On-line
Connect time required?	Little	Much
Use of server resources?	Minimal	Extensive
Multiple mailboxes?	No	Yes
Who backs up mailboxes?	User	ISP
Good for mobile users?	No	Yes
User control over downloading?	Little	Great
Partial message downloads?	No	Yes
Are disk quotas a problem?	No	Could be in time
Simple to implement?	Yes	No
Widespread support?	Yes	Growing

Multipurpose Internet Mail Extensions (MIME)

Electronic mail has a simple structure. Its simplicity, however, comes with a price. It can send messages only in NVT 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 Header


MIME headers
E-mail header
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

Data Types and Subtypes in MIME

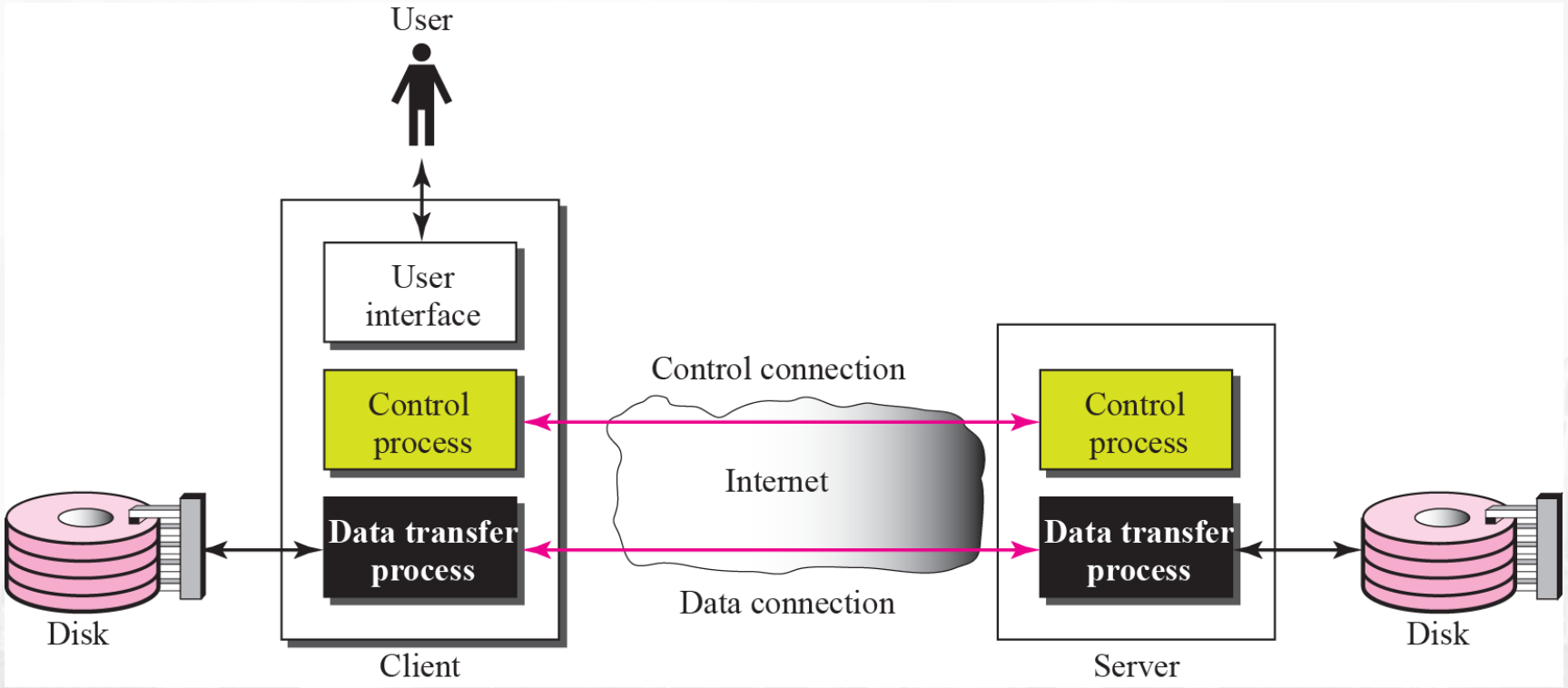
Table 23.3 *Data Types and Subtypes in MIME*

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix E)
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)

File Transfer Protocol (FTP)

- File Transfer Protocol (FTP) is the standard mechanism provided by TCP/IP for copying a file from one host to another
- Although transferring files from one system to another seems simple and straightforward
- some problems must be dealt with first
- For example, two systems may use different file name conventions.
- Two systems may have different ways to represent text and data.
- Two systems may have different directory structures.
- All of these problems have been solved by FTP in a very simple and elegant approach.

File Transfer Protocol



- FTP uses two well-known TCP ports: Port 21 is used for the control connection, and port 20 is used for the data connection

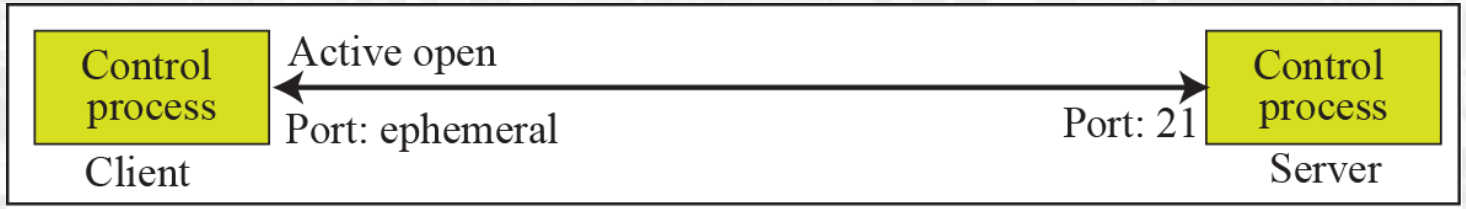
Opening the control connection

The control connection is created in the same way as other application programs described so far. There are two steps:

1. The server issues a passive open on the well-known port 21 and waits for a client.

2. The client uses an ephemeral port and issues an active open.

a. First, passive open by server



b. Later, active open by client

Creating the data connection

The **data connection uses the well-known port 20 at the server site. However, the creation** of a data connection is different from what we have seen so far. The following shows how FTP creates a data connection:

1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.

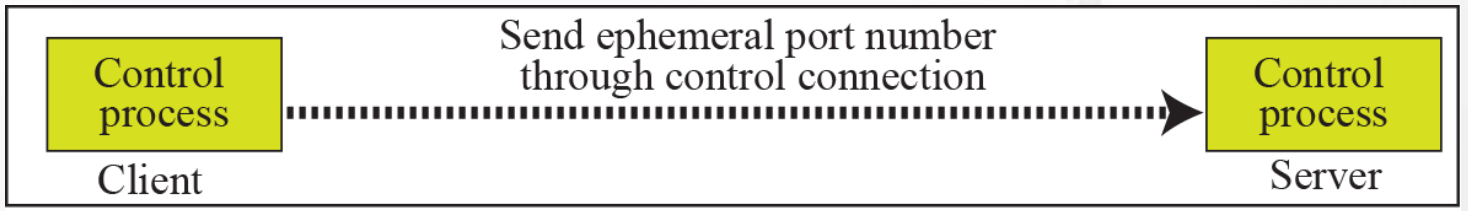
2. The client sends this port number to the server using the PORT command

3. The server receives the port number and issues an active open using the wellknown port 20 and the received ephemeral port number.

Creating the data connection

1. The client, not the server, issues a passive open using an ephemeral port. This must be done by the client because it is the client that issues the commands for transferring files.
2. The client sends this port number to the server using the PORT command
3. The server receives the port number and issues an active open using the wellknown port 20 and the received ephemeral port number.

a. First, passive open by client



b. Second, sending of ephemeral port

c. Third, active open by server

File Managemnet commands

Table 21.2 *File management commands*

Command	*Argument(s)*	*Description*
CWD	Directory name	Change to another directory
CDUP		Change to parent directory
DELE	File name	Delete a file
LIST	Directory name	List subdirectories or files
NLIST	Directory name	List subdirectories or files without attributes
MKD	Directory name	Create a new directory
PWD		Display name of current directory
RMD	Directory name	Delete a directory
RNFR	File name (old)	Identify a file to be renamed
RNTO	File name (new)	Rename the file
SMNT	File system name	Mount a file system

File transfer commands

Table 21.5 *File transfer commands*

Command	*Argument(s)*	*Description*
RETR	File name(s)	Retrieve files; file(s) are transferred from server to client
STOR	File name(s)	Store files; file(s) are transferred from client to server
APPE	File name(s)	Similar to STOR, but if file exists, data must be appended to it
STOU	File name(s)	Same as STOR, but file name will be unique in the directory
ALLO	File name(s)	Allocate storage space for files at the server
REST	File name(s)	Position file marker at a specified data point
STAT	File name(s)	Return status of files

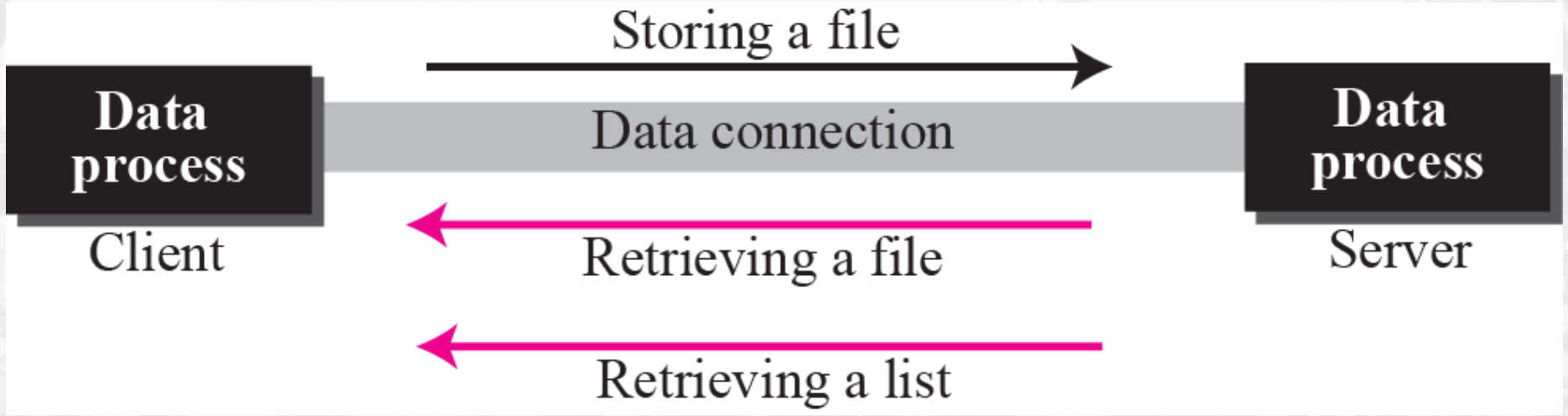
Responses

Table 21.7 *Responses*

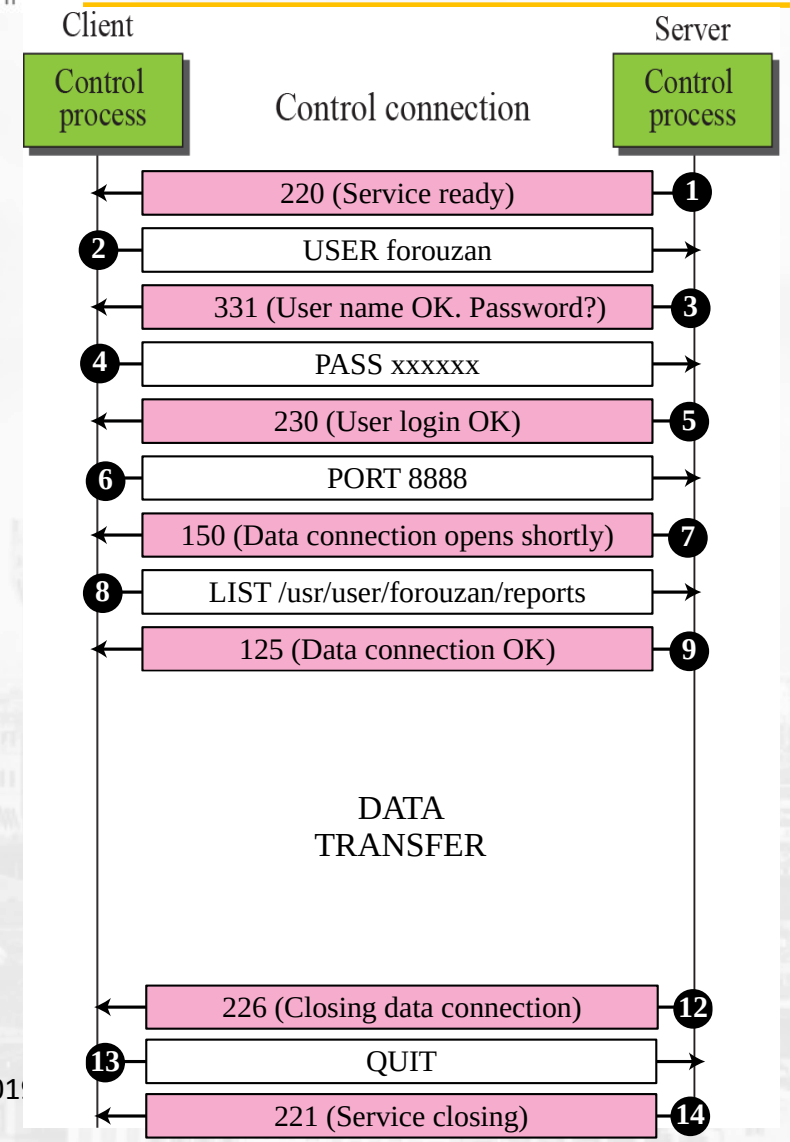
Code	*Description*
Positive Preliminary Reply	
120	Service will be ready shortly
125	Data connection open; data transfer will start shortly
150	File status is OK; data connection will be open shortly
Positive Completion Reply	
200	Command OK
211	System status or help reply
212	Directory status
213	File status
214	Help message
215	Naming the system type (operating system)
220	Service ready
221	Service closing
225	Data connection open
226	Closing data connection
227	Entering passive mode; server sends its IP address and port number
230	User login OK
250	Request file action OK
Positive Intermediate Reply	
331	User name OK; password is needed
332	Need account for logging
350	The file action is pending; more information needed

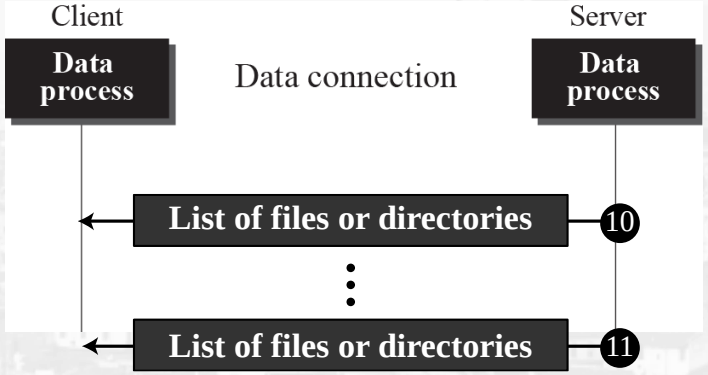
File Transfer

1. A file is to be copied from the server to the client (download). This is called retrieving a file. It is done under the supervision of the RETR command.
2. A file is to be copied from the client to the server (upload). This is called storing a file. It is done under the supervision of the STOR command.
3. A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command. Note that FTP treats a list of directory or file names as a file. It is sent over the data connection.



Example of using FTP for retrieving a list of items in a directory.





Using FTP for retrieving a list of items in a directory

1. **After the control connection to port 21 is created, the FTP server sends the 220 (service** ready) response on the control connection.
2. **The client sends the USER command.**
3. **The server responds with 331 (user name is OK, password is required).**
4. **The client sends the PASS command.**
5. **The server responds with 230 (user login is OK).**
6. **The client issues a passive open on an ephemeral port for the data connection and sends the** PORT command (over the control connection) to give this port number to the server.
7. **The server does not open the connection at this time, but it prepares itself for issuing an** active open on the data connection between port 20 (server side) and the ephemeral port received from the client. It sends response 150 (data connection will open shortly).

Using FTP for retrieving a list of items in a directory

8. The client sends the LIST message.

9. Now the server responds with 125 and opens the data connection.

10. The server then sends the list of the files or directories (as a file) on the data connection.

When the whole list (file) is sent, the server responds with 226 (closing data connection) over the control connection.

11. The client now has two choices. It can use the QUIT command to request the closing of the control connection or it can send another command to start another activity (and eventually open another data connection). In our example, the client sends a QUIT command.

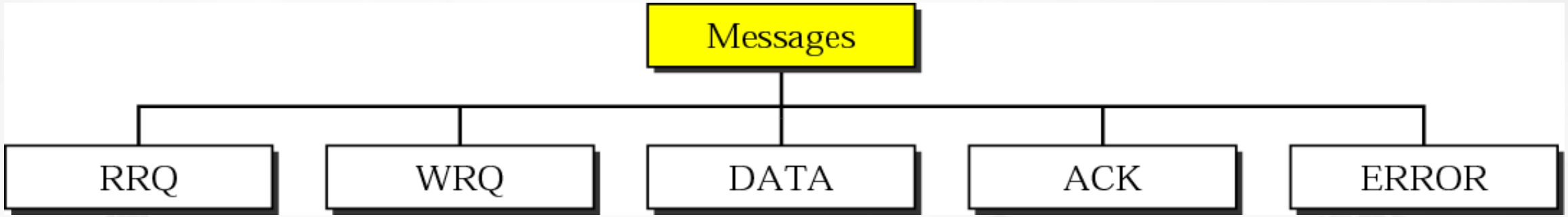
12. After receiving the QUIT command, the server responds with 221 (service closing) and then closes the control connection.

Trivial File Transfer Protocol (TFTP)

- There are occasions when we need to simply copy a file without the need for all of the features of the FTP protocol. For example, when a diskless workstation or a router is booted, we need to download the bootstrap and configuration files. Here we do not need all of the sophistication provided in FTP. We just need a protocol that quickly copies the files.
- **Trivial File Transfer Protocol (TFTP) is designed for these types of file transfer.**
- It is so simple that the software package can fit into the read-only memory of a diskless workstation.

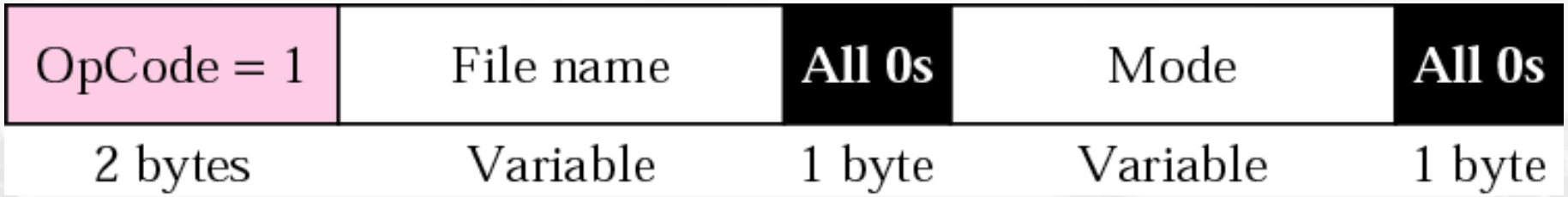
- *TFTP is good for simple file transfers, such as during boot time.*
- *It uses UDP and basic IP and can operate out of ROM.*
- *It uses the well-known port 69.*
- TFTP can read or write a file for the client. *Reading means copying a file from the server site to the client* site. *Writing means copying a file from the client site to the server site.*

Message categories



RRQ – read request: used to establish a connection for reading data from a server
WRQ – write request
DATA – used to send data blocks

RRQ format



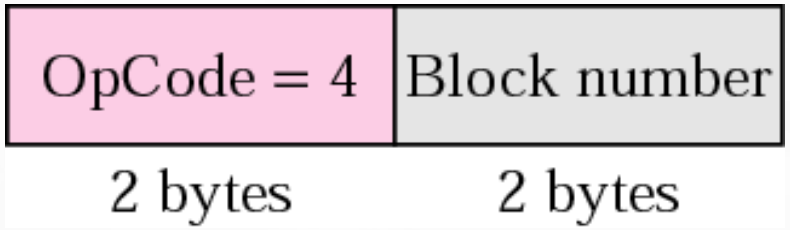
Mode: defines the type of the file transferred
"netascii" for an ASCII file; "octet" for a binary file

WRQ format and Data format

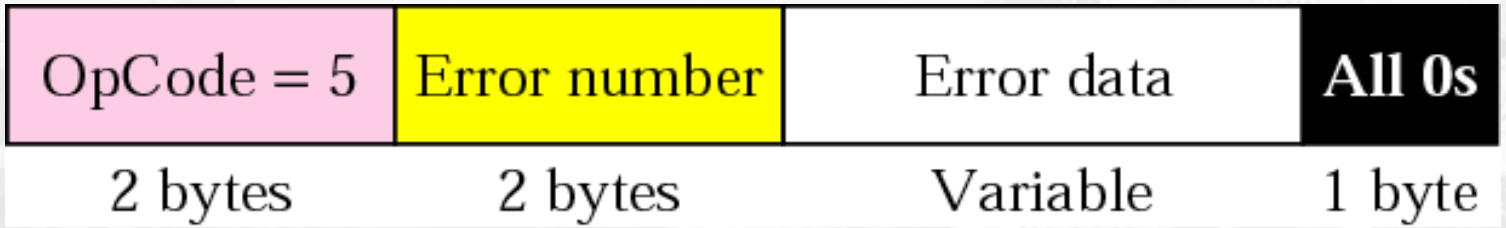
WRQ format

Data format

ACK and Error format



ACK and Error format



ACK and Error format

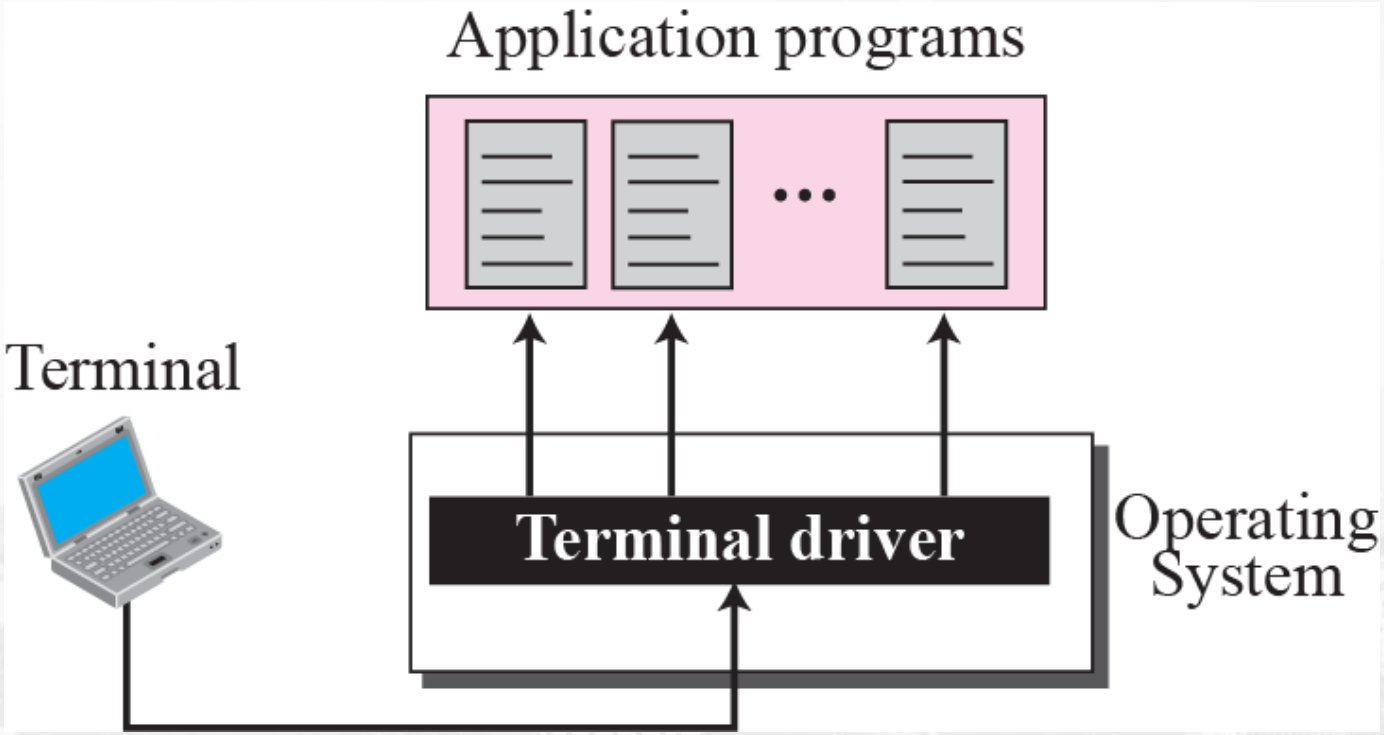
Difference Between FTP and TFTP

FTP	TFTP
FTP uses two connection	TFTP uses one conection
Provide many commands	Provide only five commands
Uses TCP	Uses UDP
Client must login to server	No login procedure
Allow for user authentication	TFTP does not allow for user authentication
FTP provide reliable service	Unreliable
21- control, 20-data	Port 69

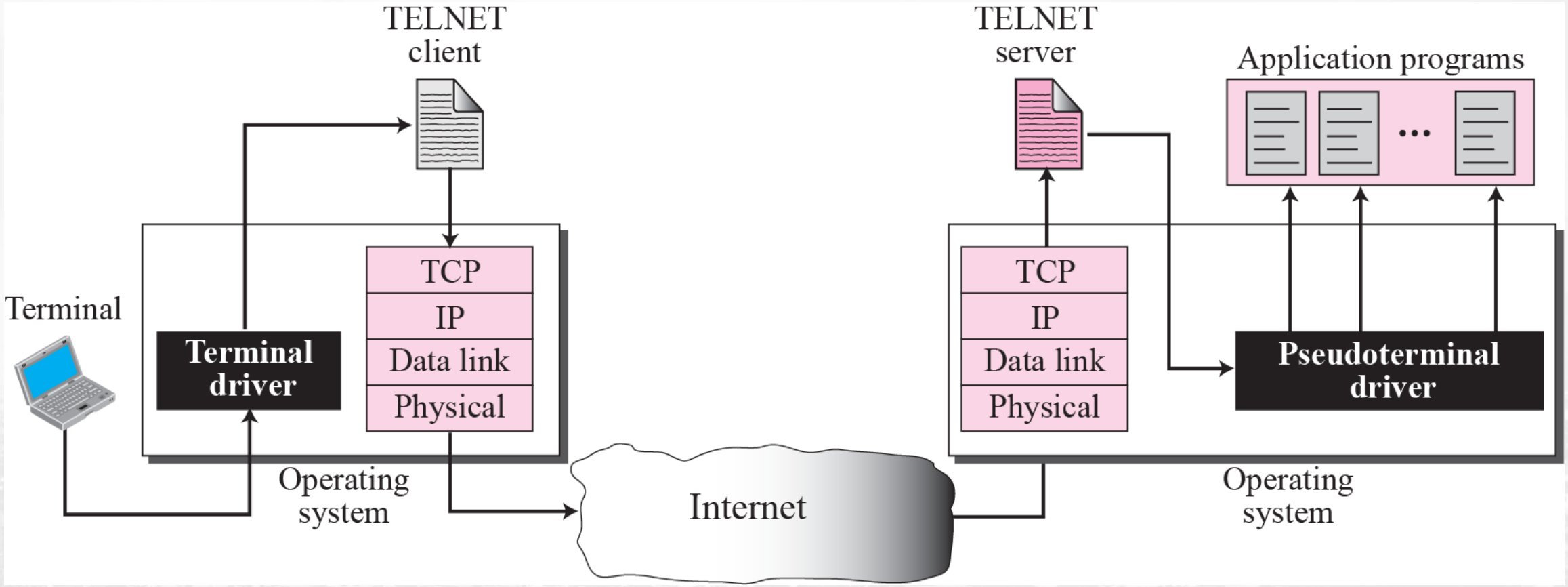
TELNET

- TELNET is an abbreviation for TErminaL NETwork
- TELNET is a *protocol* that provides “a general, bi-directional, eight-bit byte oriented communications facility”.
- **Telnet** is a *program* that supports the TELNET protocol over TCP.
- Many application protocols are built upon the TELNET protocol.
- TELNET enables the establishment of a connection to a remote system

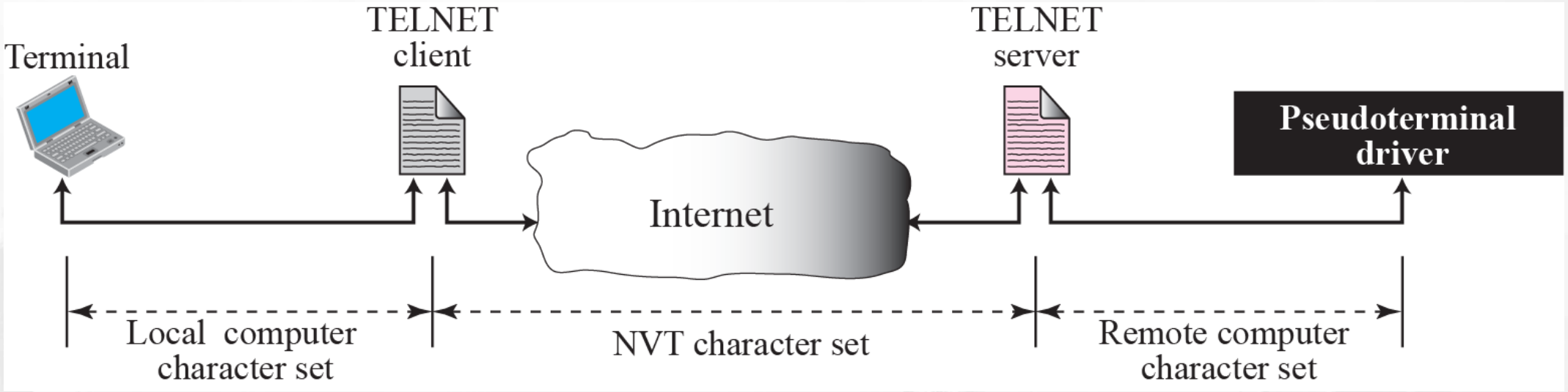
Local Login


Application programs
Terminal
Terminal driver
Operating System

Remote Login



Concept of Network Virtual Terminal



- intermediate representation of a generic terminal
- provides a standard language for communication of terminal control functions

TELNET

- Reference: RFC 854

- TCP connection
- data and control over the same connection.

Command Structure

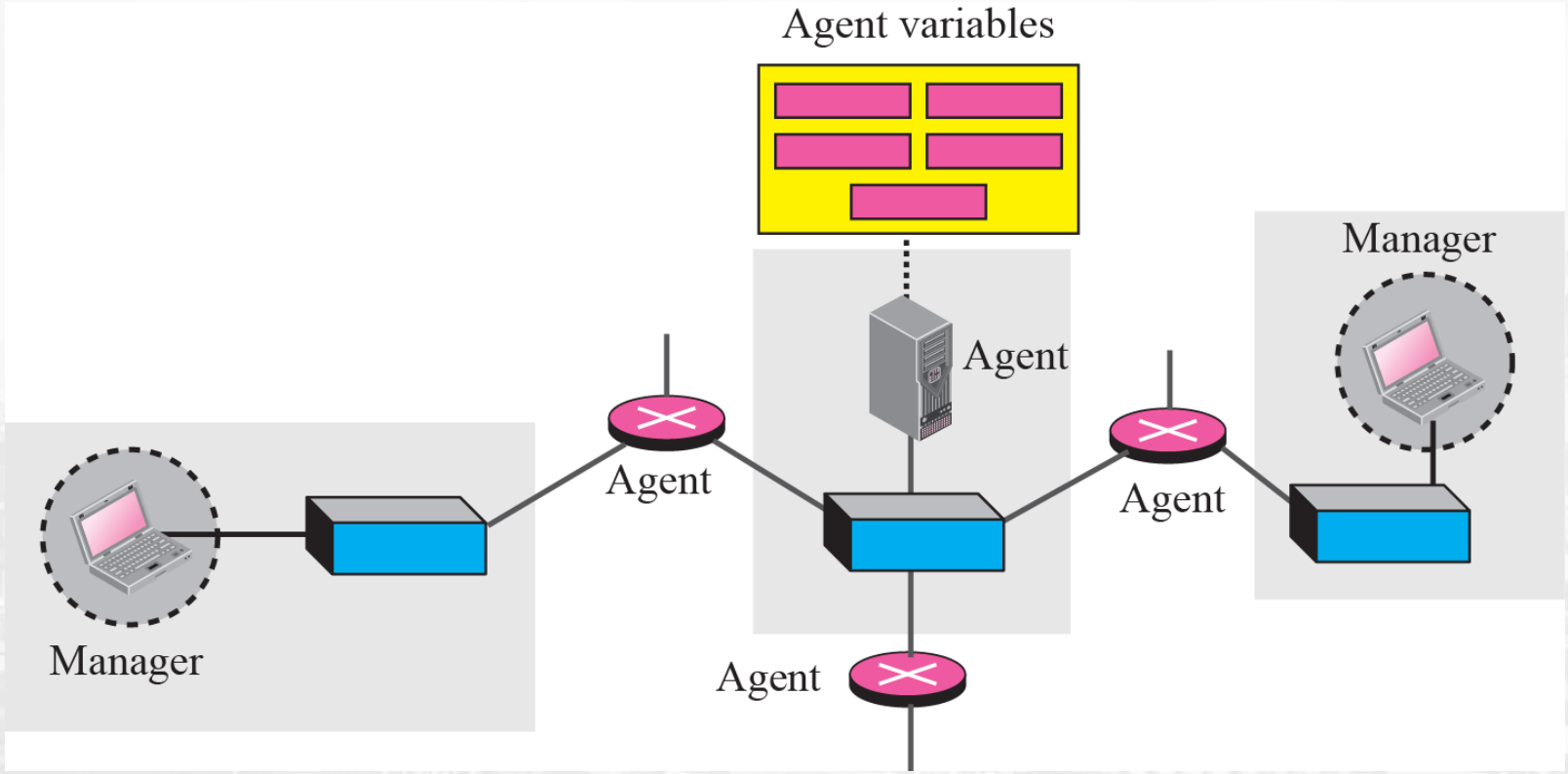
- All TELNET commands and data flow through the same TCP connection.
- Commands start with a special character called the Interpret as Command *escape* character
  - The IAC code is 255.
  - If a 255 is sent as data - it must be followed by another 255.
- If IAC is found and the next byte is IAC
  - a single byte is presented to application/terminal
- If IAC is followed by any other code
  - the TELNET layer interprets this as a command.

Playing with TELNET

- You can use the **`telnet`** program to play with the TELNET protocol.
- **`telnet`** is a *generic* TCP client.
  - Sends whatever you type to the TCP socket.
  - Prints whatever comes back through the TCP socket
  - Useful for testing TCP servers (ASCII based protocols).
- Many Unix systems have these servers running (by default):
  - **`echo`** port 7 **`discard`** port 9
  - **`daytime`** port 13 **`chargen`** port 19

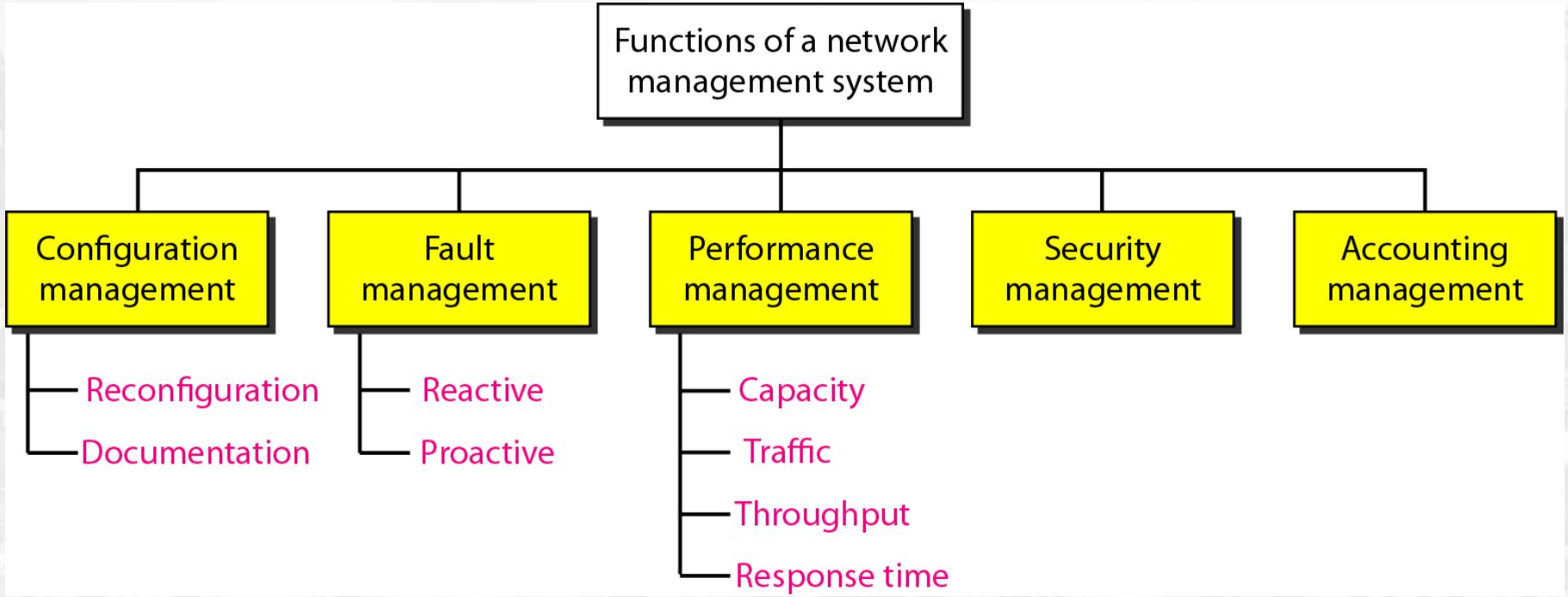
SNMP concept

- The Simple Network Management Protocol (SNMP) is a framework for managing devices in an internet using the TCP/IP protocol suite.
- It provides a set of fundamental operations for monitoring and maintaining an internet.



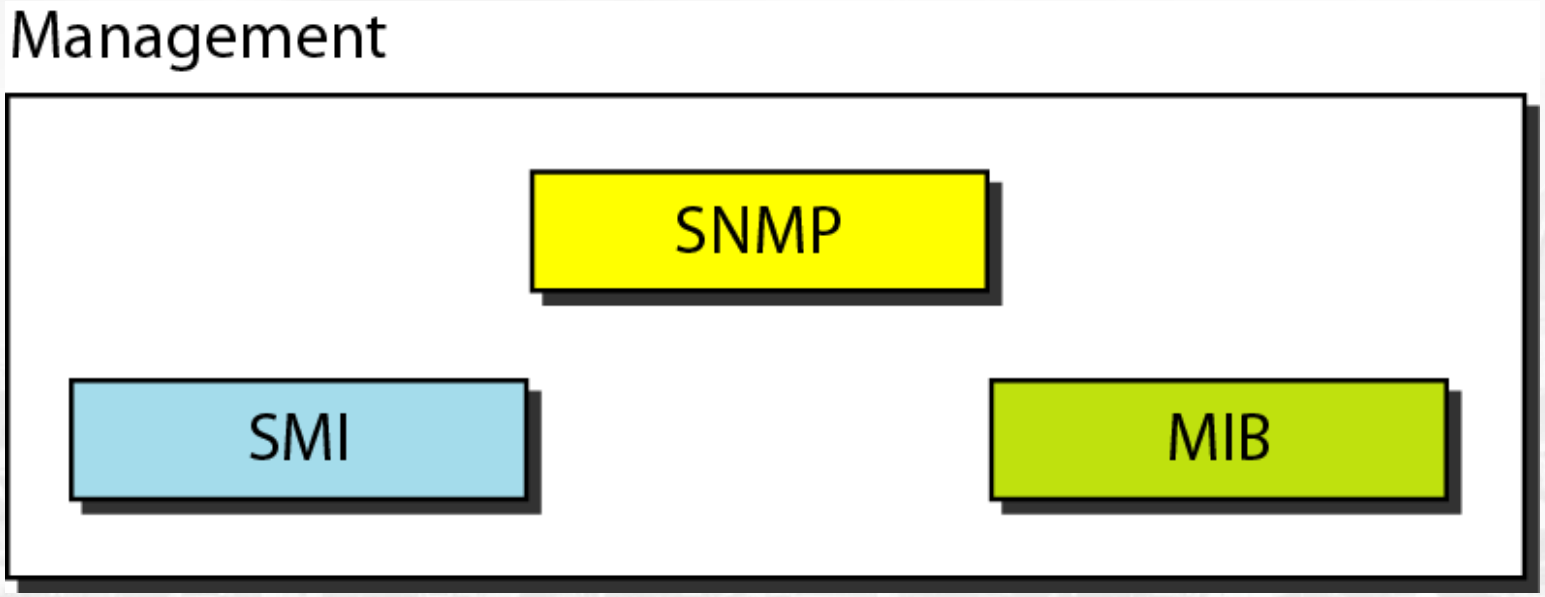
Simple Network Management Protocol (SNMP)

- Functions performed by a network management system can be divided into five broad categories: configuration management, fault management, performance management, security management, and accounting management.



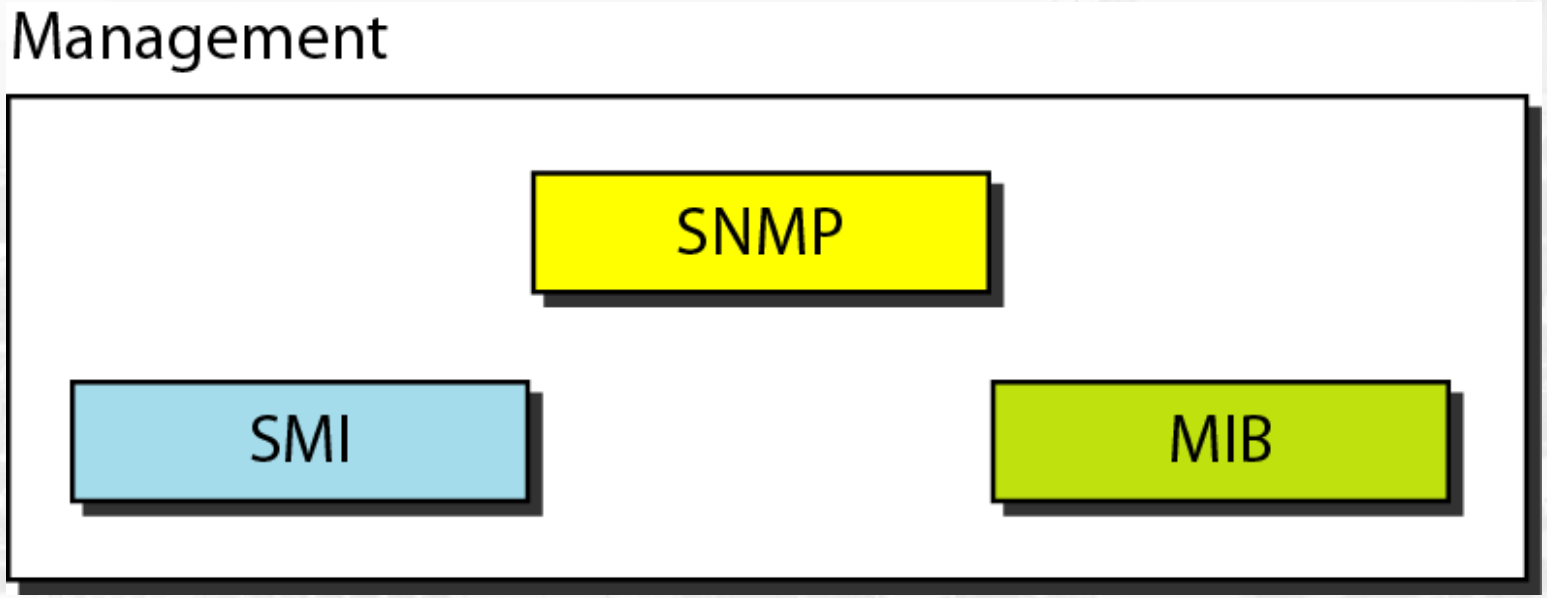
Components of network management on the Internet

- **Structure of Management Information (SMI)**
- **and Management Information Base (MIB)**



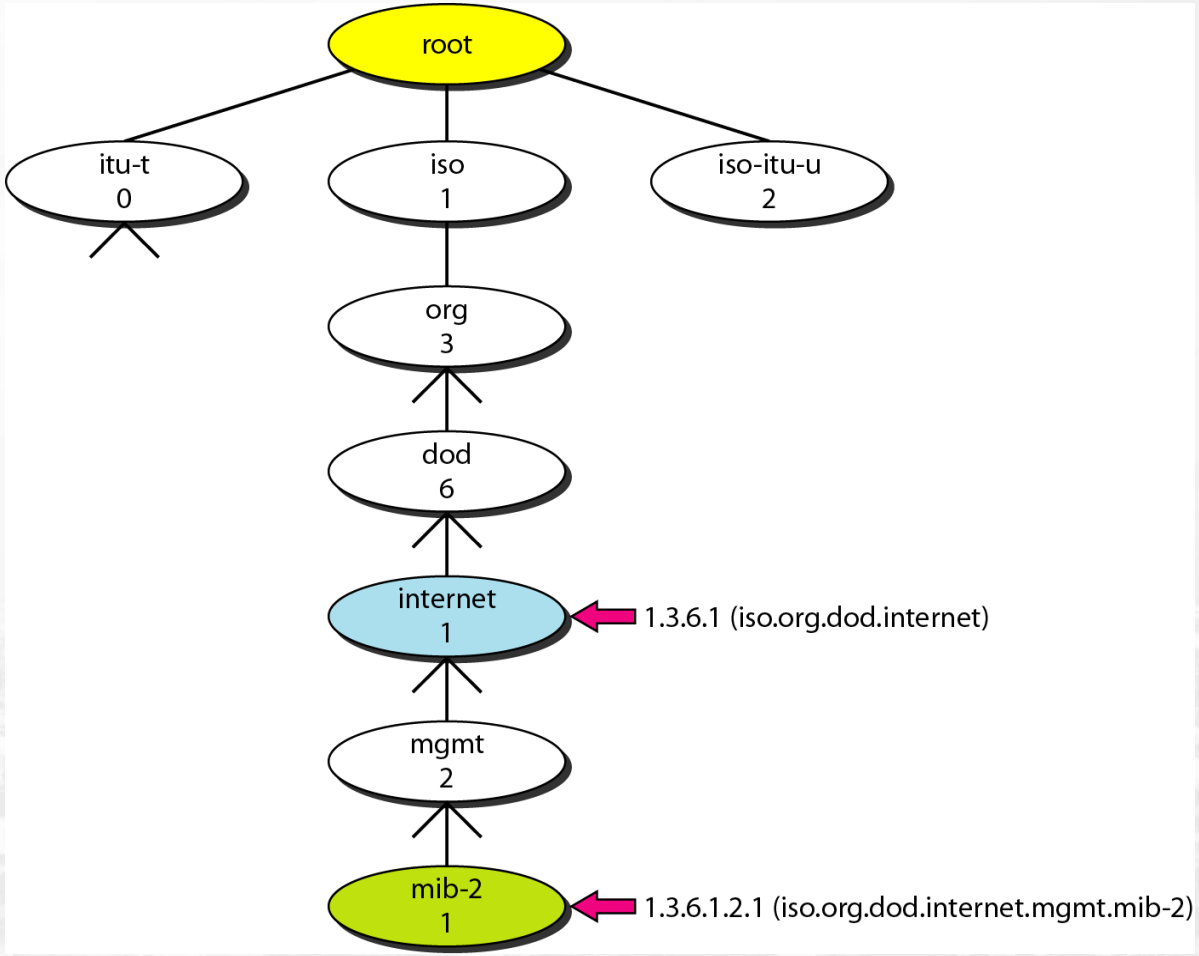
SMI and MIB

- SMI defines the general rules for naming objects, defining object types (including range and length), and showing how to encode objects and values
- SMI does not define the number of objects an entity should manage or name the objects to be managed or define the association between the objects and their values.
- MIB creates a collection of named objects, their types, and their relationships to each other in an entity to be managed.



Object Identifier

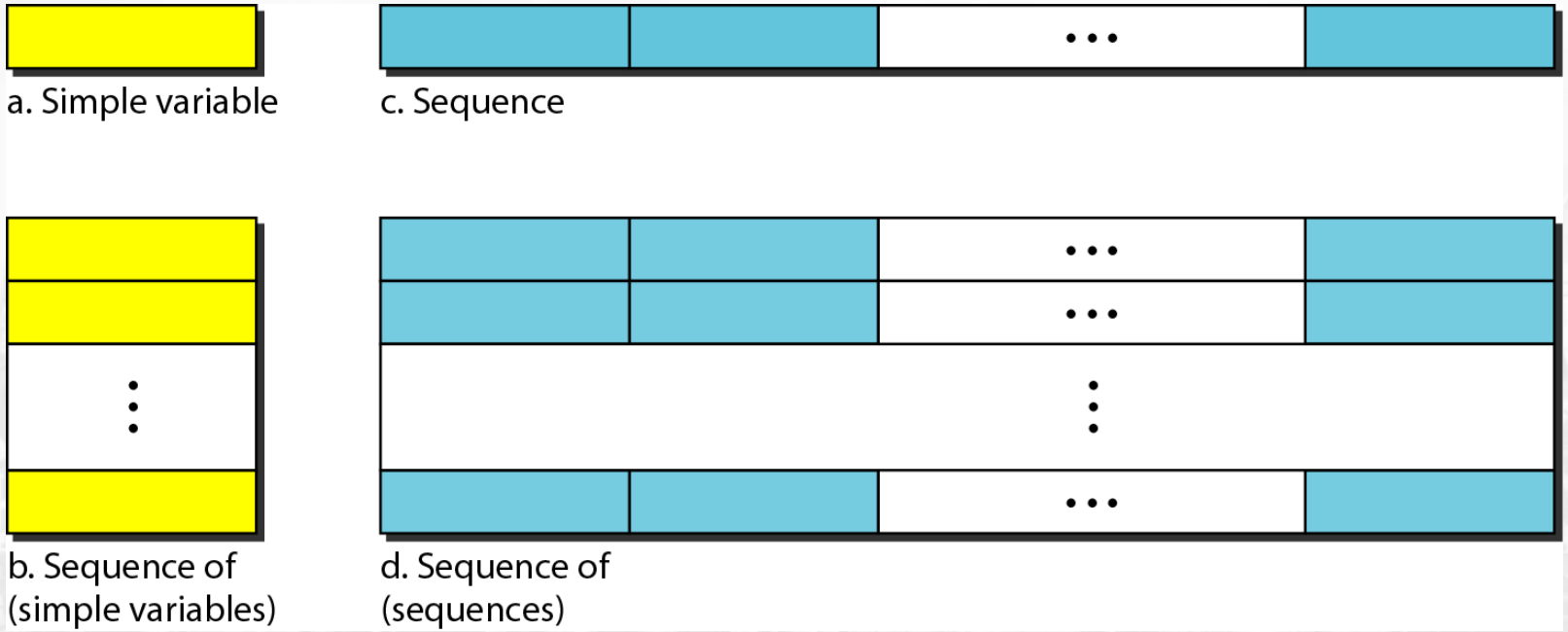
- SMI uses an **object identifier,** which is a hierarchical identifier based on a tree structure
- All objects managed by SNMP are given an object identifier.
- The object identifier always starts with 1.3.6.1.2.1.



Conceptual data types

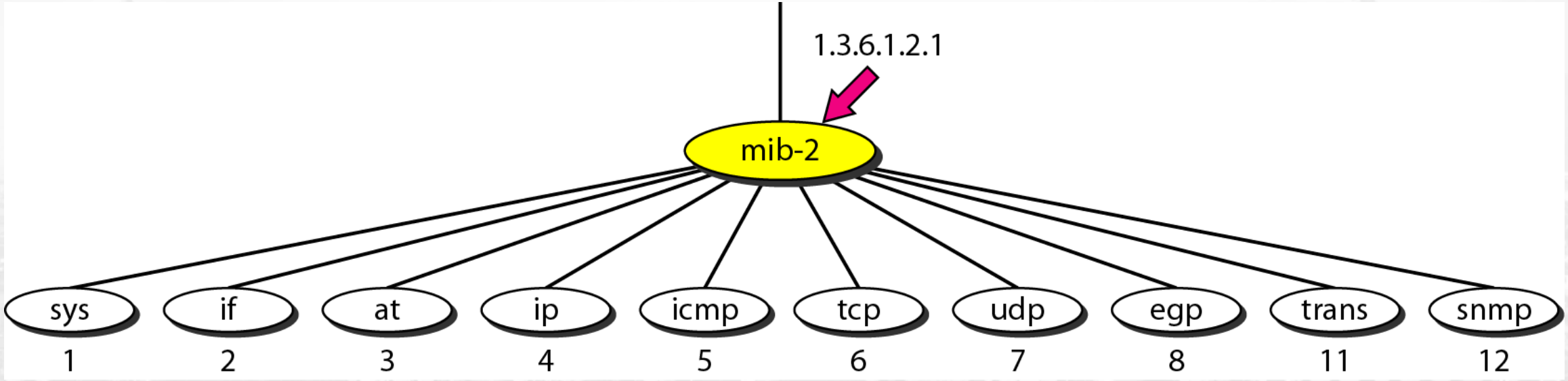
SMI defines two structured data types: sequence and sequence of

- A sequence data type is a combination of simple data types, not necessarily of the same type
- sequence of data type is a combination of simple data types all of the same type

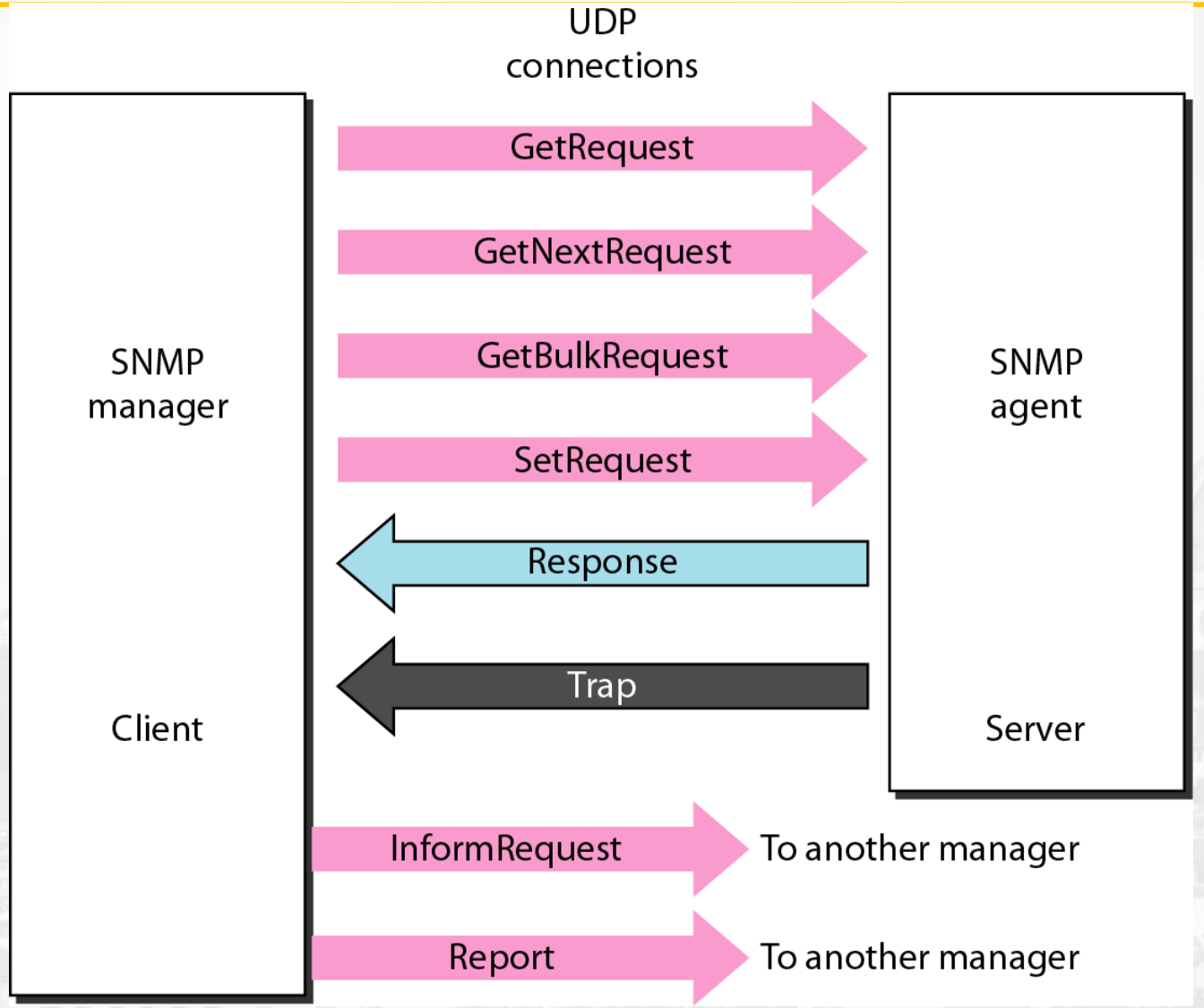


Management Information Base, version 2 (MIB2)

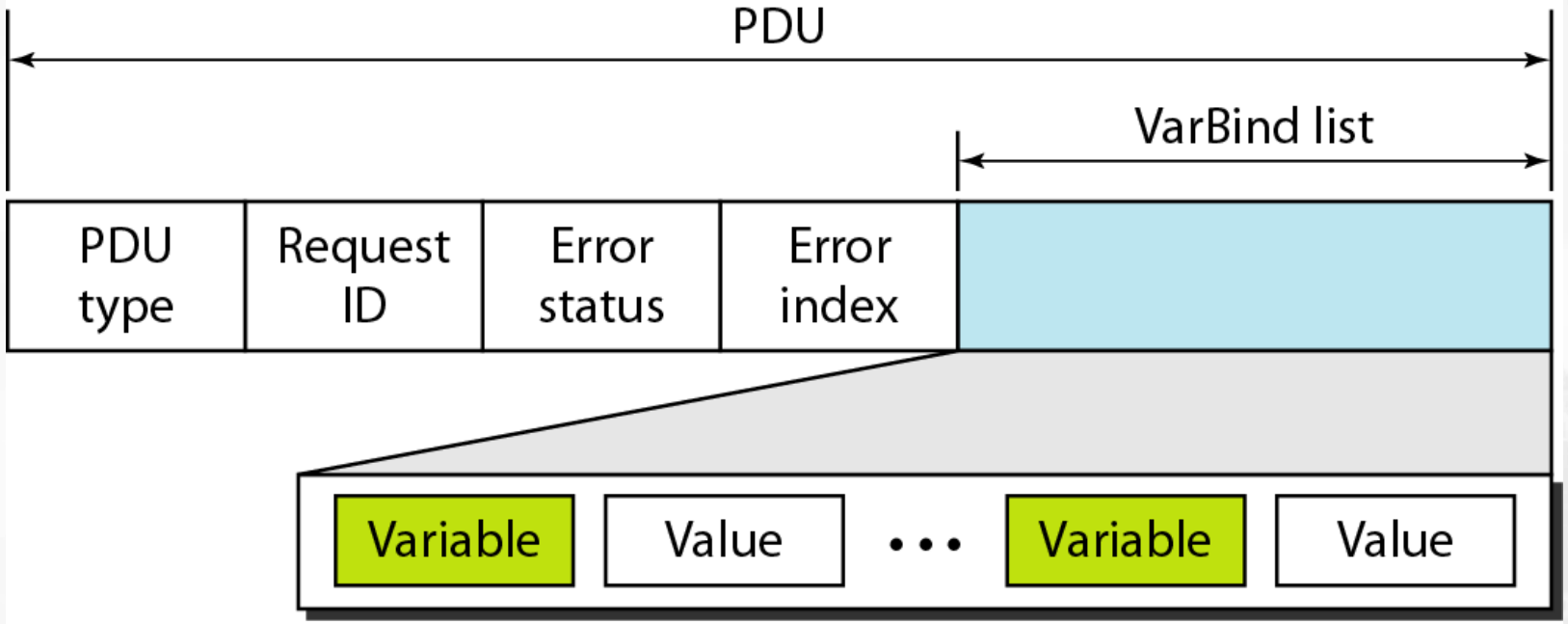
- The Management Information Base, version 2 (MIB2) is the second component used in network management.
- Each agent has its own MIB2, which is a collection of all the objects that the manager can manage.



SNMP PDUs


UDP
connections
SNMP
manager
SNMP
agent
GetRequest
GetNextRequest
GetBulkRequest
SetRequest
Response
Trap
Client
Server
InformRequest
To another manager
Report
To another manager

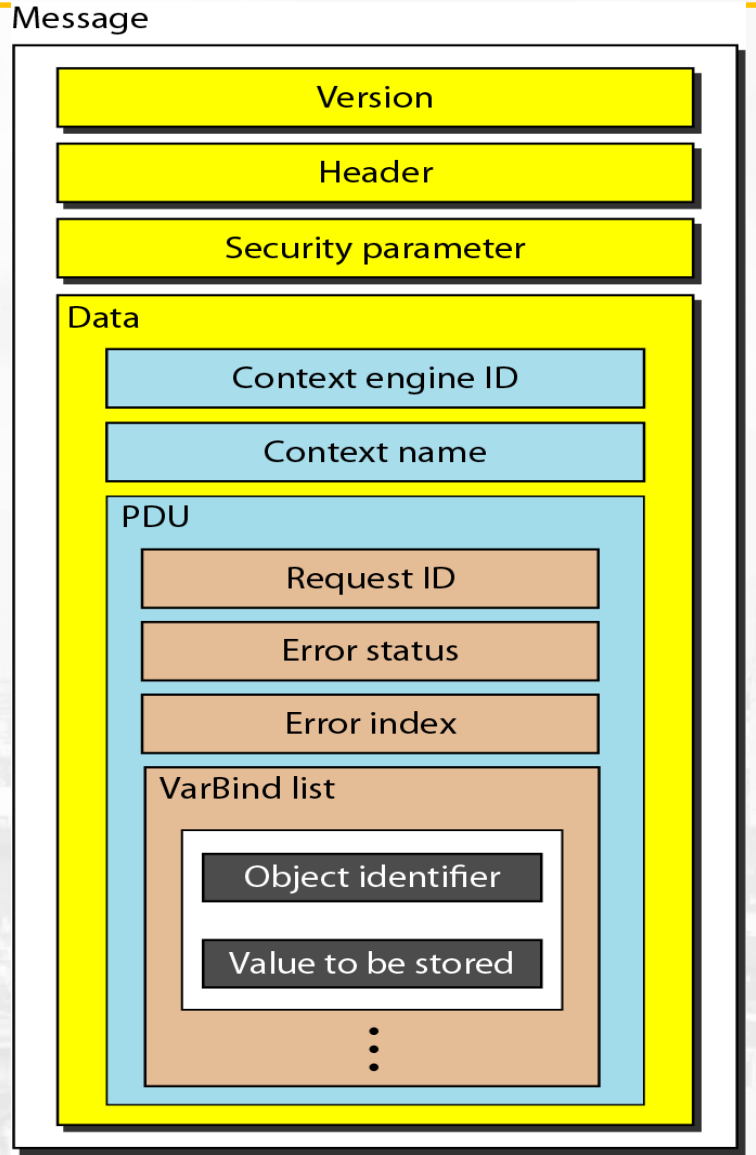
SNMP PDU format



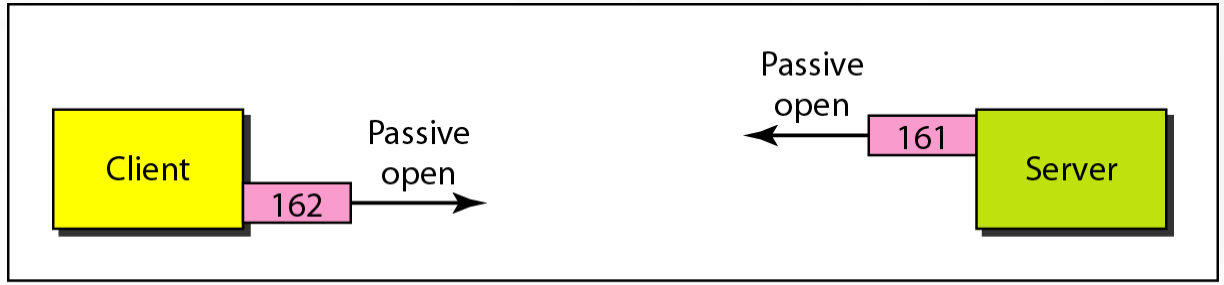
Differences:

1. Error status and error index values are zeros for all request messages except GetBulkRequest.
2. Error status field is replaced by nonrepeater field and error index field is replaced by max-repetitions field in GetBulkRequest.

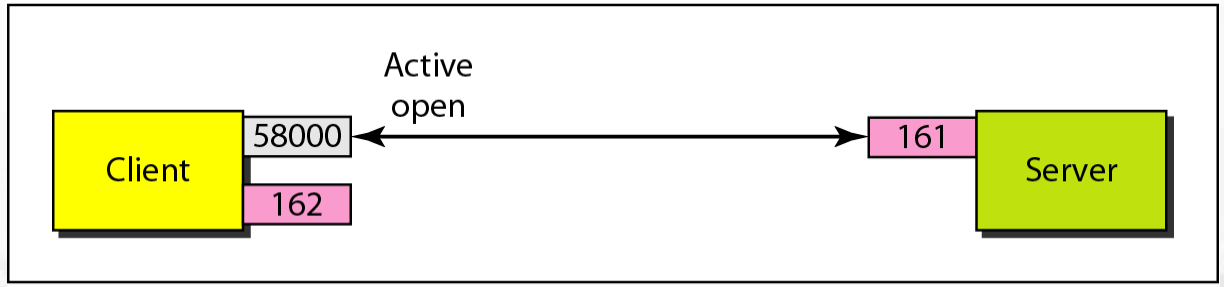
SNMP message



Port numbers for SNMP



a. Passive open by both client and server



b. Exchange of request and response messages

c. Server sends trap message

Questions

1. What is HTTP?
2. What are the basic Features of HTTP?
3. What are request methods in HTTP ?
4. Differentiate between persistent and non persistent HTTP connection ?
5. Explain HTTP caching ?
6. Define and Explain DNS ?
7. Explain DNS Query and response messages ?
8. Explain how name resolution happens in DNS. Enlist all the resource records and its function ?
9. DNS uses UDP instead of TCP. If a DNS packet is lost, there is no automatic recovery. Does this cause a problem, and if so, how is it solved?
10. Can a computer have two DNS names that fall in different top-level domains? If so, give a plausible example. If not, explain why not ?
11. Explain DHCP.
12. Explain DHCP packet format

Questions

13. Explain the function of E-mail system.
14. Explain E-mail system along with SMTP, POP3 and IMAP 4 protocol
15. When web pages are sent out, they are prefixed by MIME headers. Why?
16. List the similarities and differences between POP3 and IMAP.
17. Explain File Transfer Protocol.
18. Explain data connection and control connection in FTP
19. Compare between FTP and TFTP.
20. Why does it need an PRQ or WRQ message in TFTP but not in FTP
21. Explain TELNET
22. Enlist functions of Network Management System
23. Explain Structure of Management Information (SMI)
24. Write a note on Management Information Base