WEB TECHNOLOGY

UTTAM K. ROY

Dept. of Information Technology,
Jadavpur University, Kolkata

Agenda

- *Background*
- *HTTP Protocol*
- *Domain Name System (DNS)*
- *Simple Mail Transfer Protocol (SMTP)*
- *HyperText Markup Language(HTML)*
- *JavaScript*
- *XML*
- *JSP*

HyperText Transfer Protocol (HTTP)

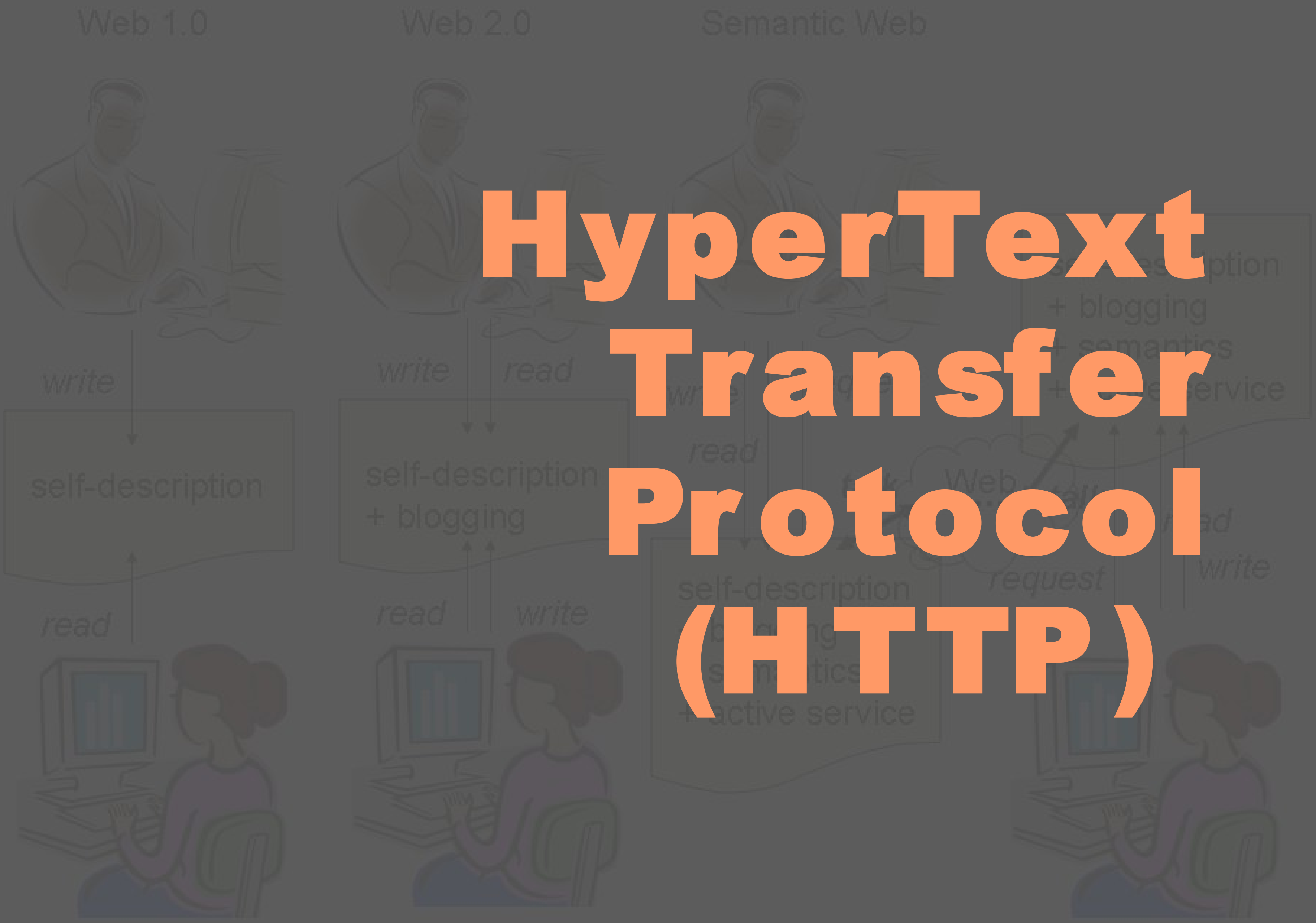


Image from Thinking Space by Yihong Ding

WWW

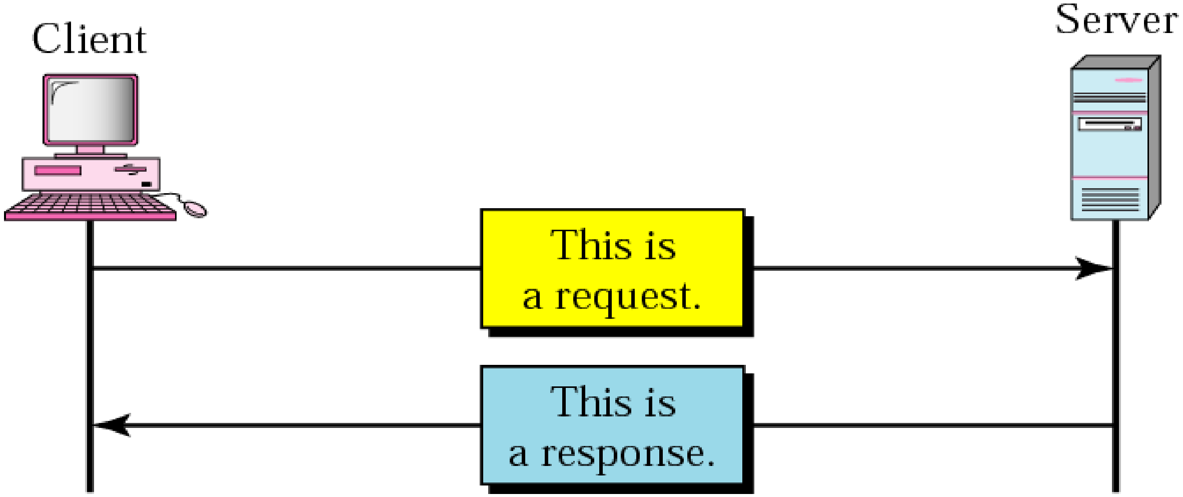
- World Wide Web—**a repository of Information**
- Introduced in 1991
- Originated from the CERN High-Energy Physics laboratory in Geneva, Switzerland.
- Purpose—**create a system to handle distributed resource**
- A client-server service
- Service provider—**called *website***

The Web: Some Jargon

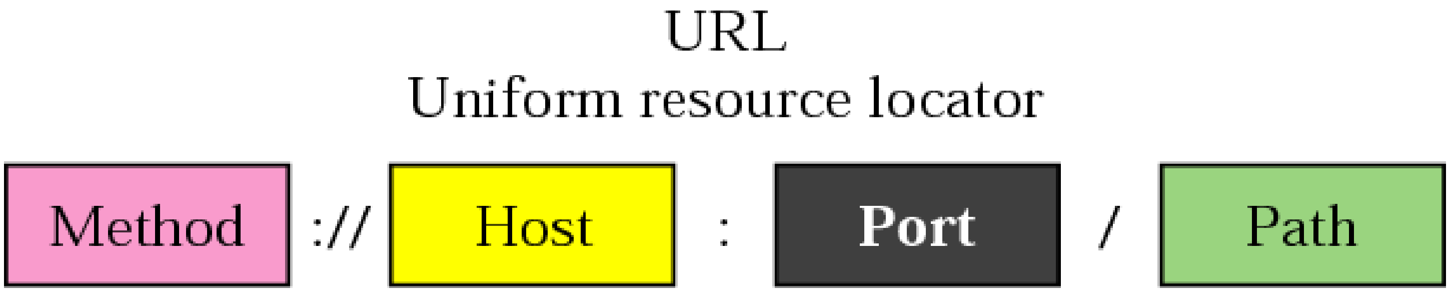
- Web page
  - consists of objects (HTML file, JPEG image, GIF image…)
  - addressed by URL
- Most Web pages consist of
  - base HTML page
  - several referenced objects—Hypertext and Hepermedia
- URL
  - A standard way of specifying the location of an object, typically a web page, on the Internet
- User agent for Web is called a browser
  - Windows
    - MS Internet Explorer
  - Linux
    - Netscape Navigator
    - Mozzila
    - Konquor
- Server for Web is called a Web server

HyperText Transfer Protocol

- Web’s application layer protocol
  - Used to access data on the World Wide Web
  - Rapid jump from one document to another
- Client-server model
  - client: browser that requests, receives, “displays” web objects
  - server: Web server sends objects in response to request
- uses TCP connection on the well-known port 80



URL



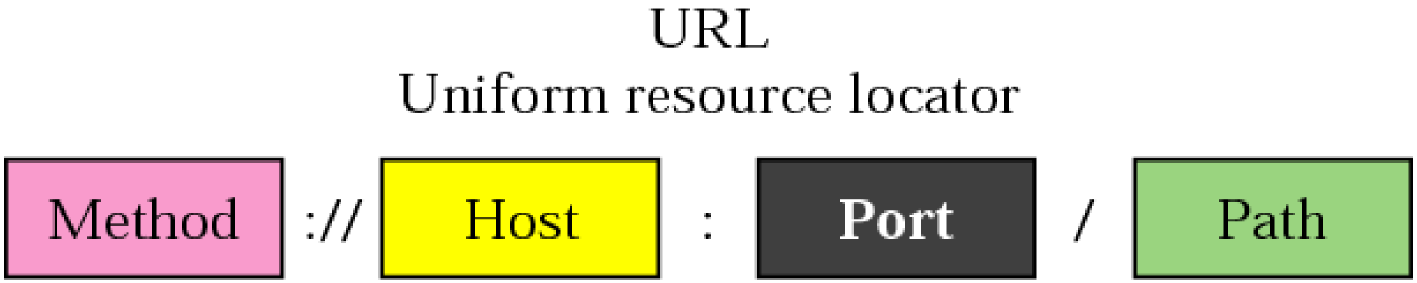
URL

Uniform resource locator

Method :// Host : Port / Path

- An address of the web page or other information on the Internet
- Example
  - http://www.yahoo.com/
  - http://www.jusl.ac.in/images/sitemap.gif
  - http://www.foldoc.org/?Uniform+Resource+Locator
  - http://mail.jusl.ac.in/
  - http://www.itd.jusl.ac.in:8080/jsp/test.jsp
  - ftp://wuarchive.wustl.edu/mirrors/msdos/graphics/gifkit.zip

URL - continued



URL

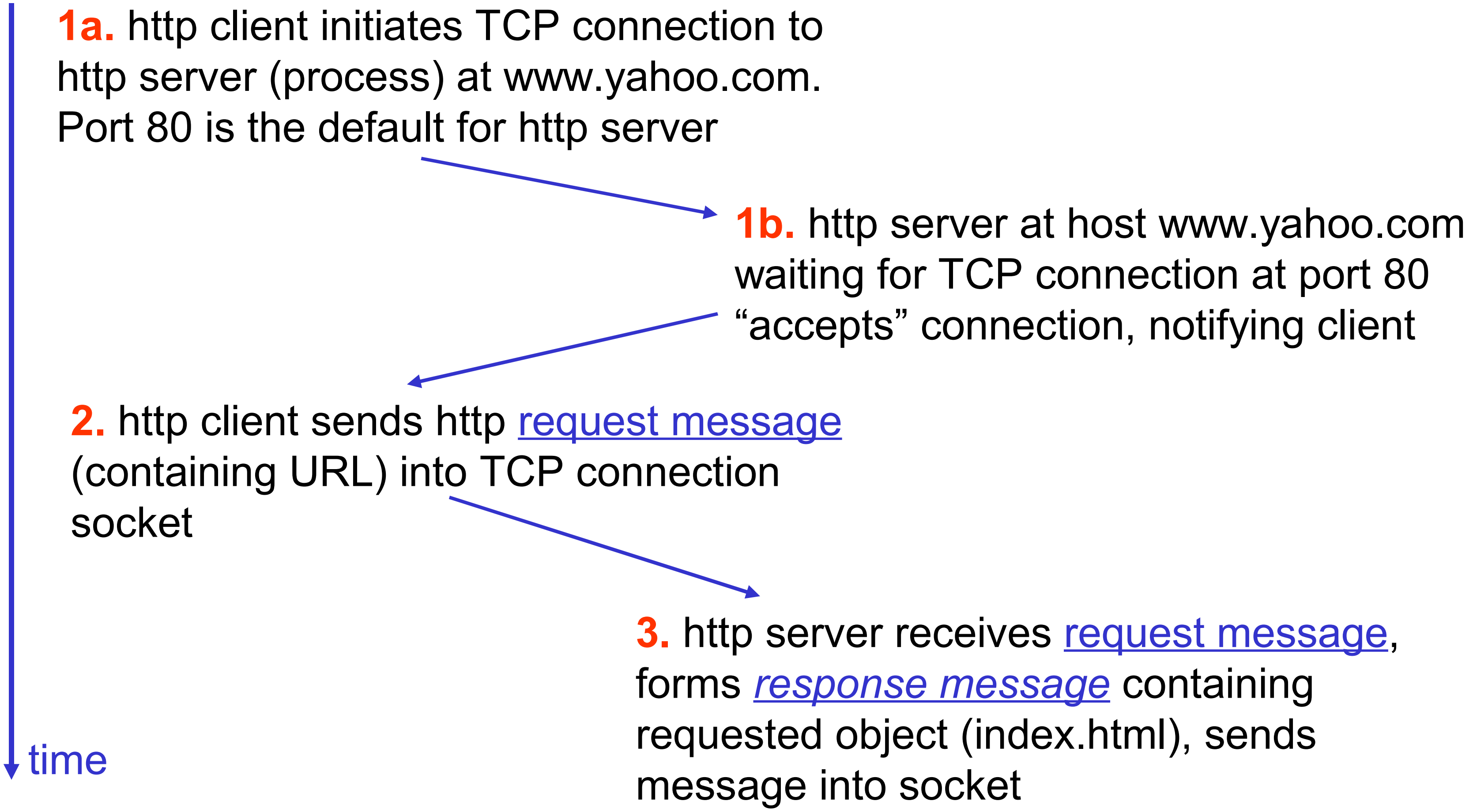
Uniform resource locator

Method :// Host : Port / Path

- **Method**
  - protocol used to retrieve the document (FTP, HTTP, …)
- **Host**
  - a computer where the info is located
  - the name/IP address of the computer can be an alias (not necessary www)
- **Port**
  - optional port # of the server (default is 80)
- **Path**
  - the path name of the file where the info is located

HTTP - example

- Suppose user enters URL www.yahoo.com/index.html



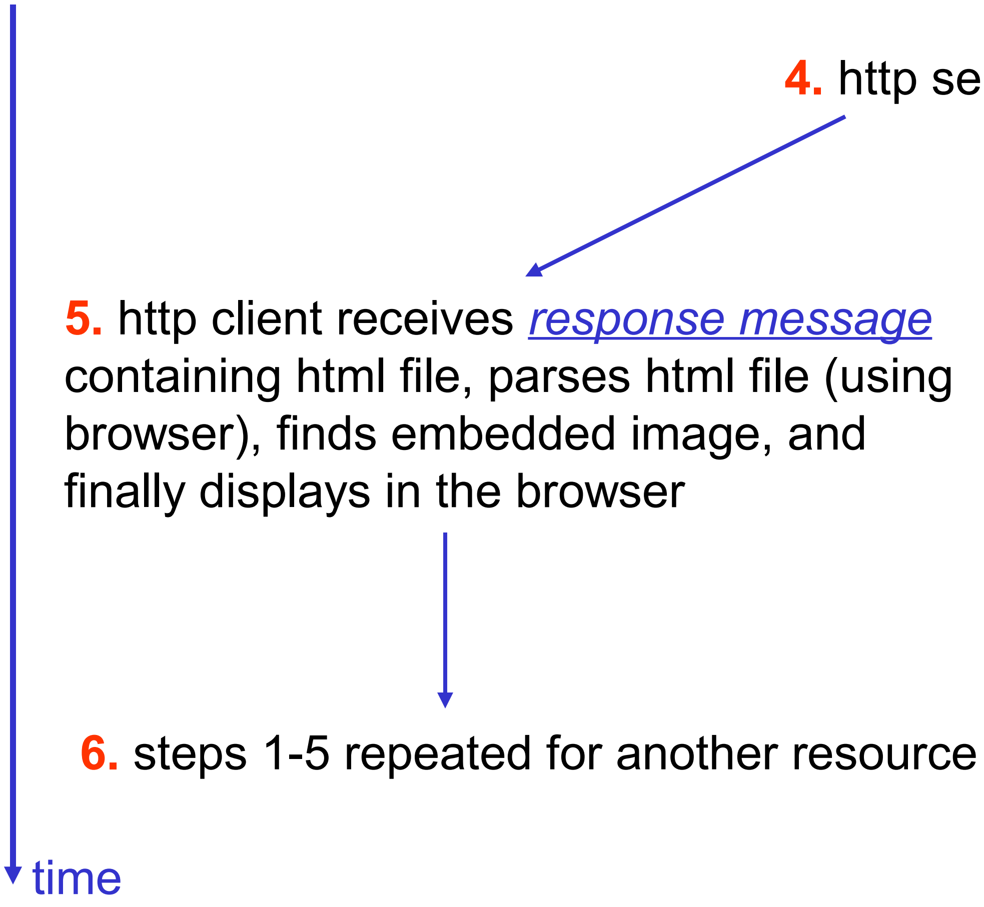
1a. http client initiates TCP connection to http server (process) at www.yahoo.com. Port 80 is the default for http server

1b. http server at host www.yahoo.com waiting for TCP connection at port 80 "accepts" connection, notifying client

2. http client sends http request message (containing URL) into TCP connection socket

3. http server receives request message, forms *response message* containing requested object (index.html), sends message into socket

HTTP – example (cnt'd)



4. http server closes TCP connection

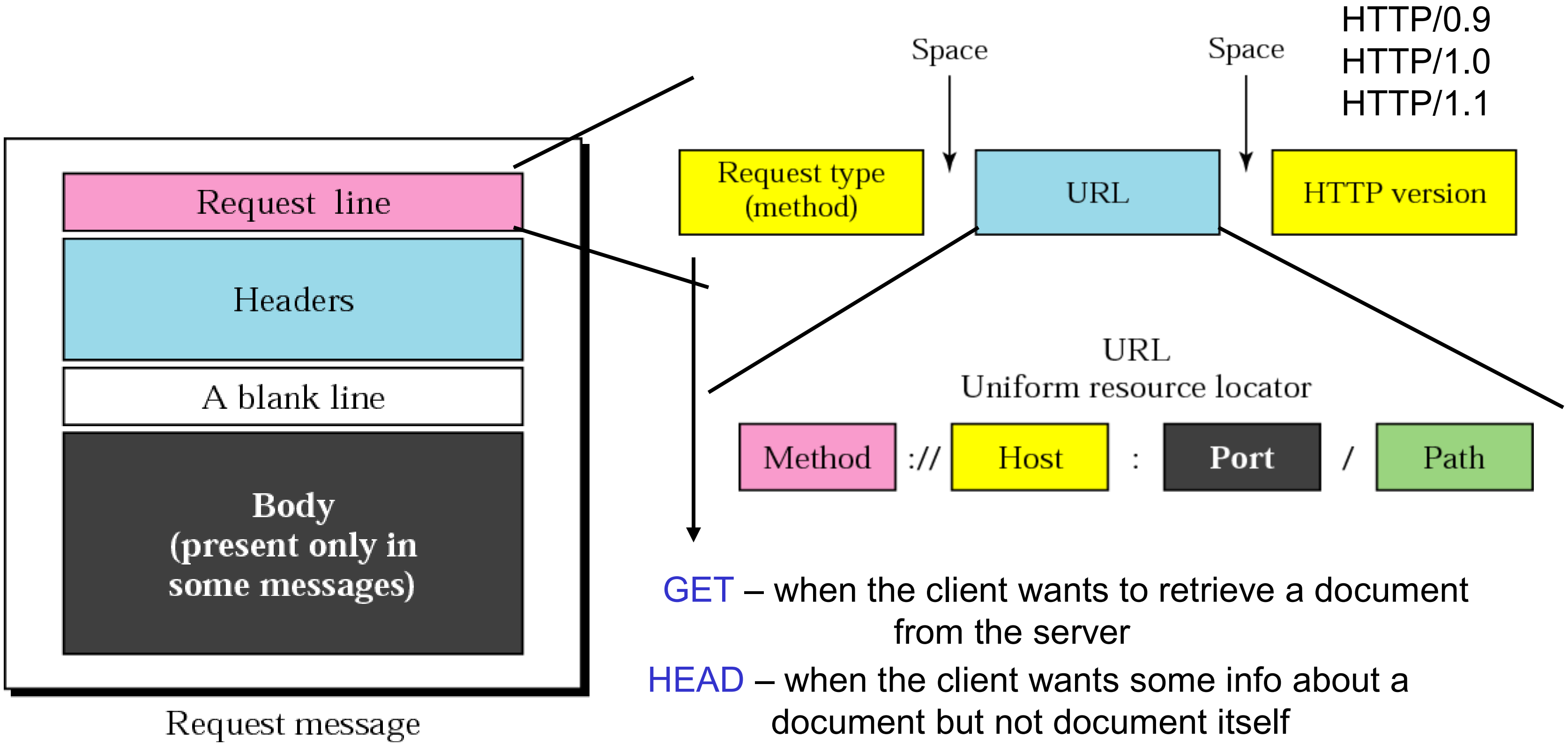
5. http client receives *response message* containing html file, parses html file (using browser), finds embedded image, and finally displays in the browser

6. steps 1-5 repeated for another resource

time

HTTP protocol – message format

- two types of messages: request & *response*
- HTTP request message



GET – when the client wants to retrieve a document from the server

HEAD – when the client wants some info about a document but not document itself

COPY – copies the file to another location

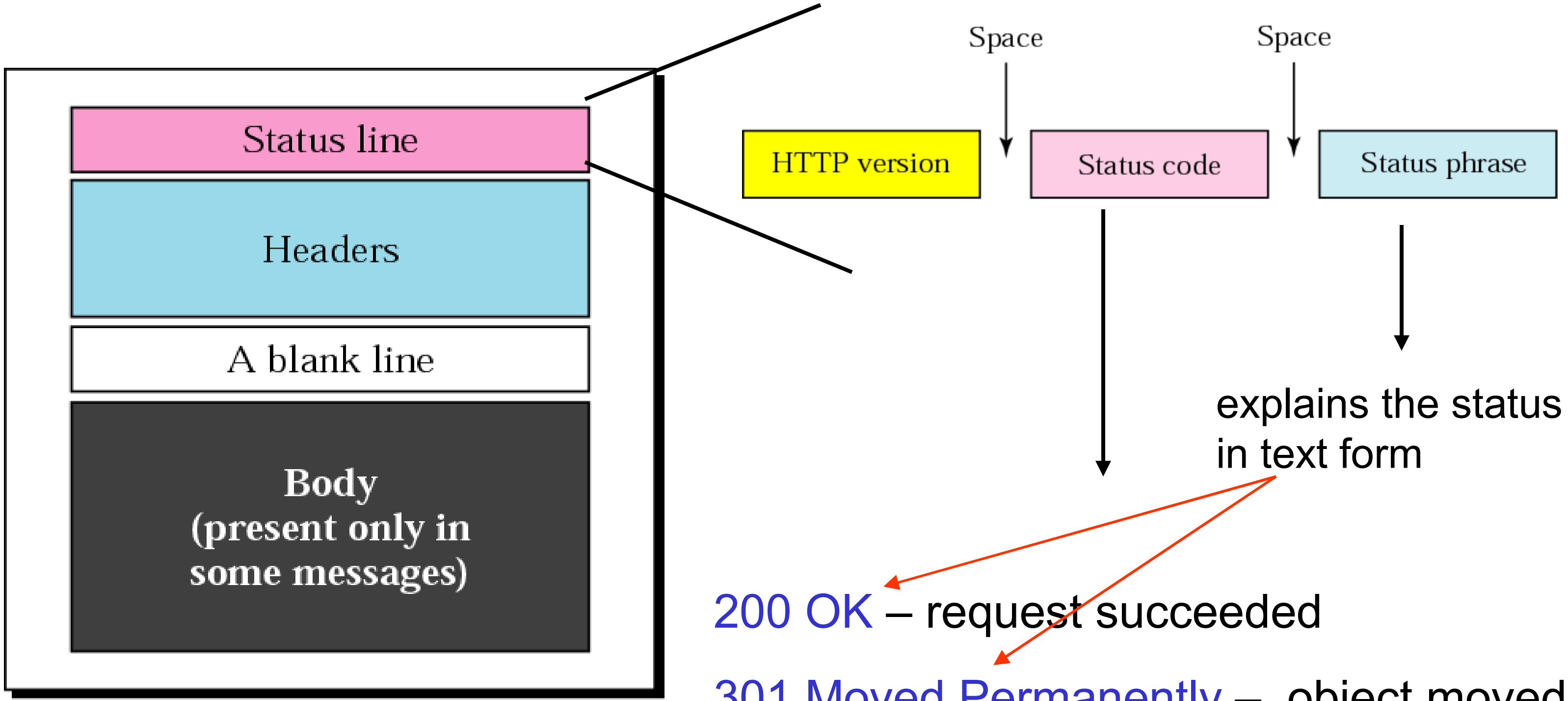
Other Request type (method)

Method	Description
POST	Used to provide information (e.g. input) to the server
PUT	Used to provide a new or replacement document to be stored on the server
PATCH	Similar to PUT except that the request contains only list of differences that should be implemented in the existing file
MOVE	Used to copy a file to another location
DELETE	Used to remove a document from the server
LINK	Used to create a link or links of a document to another location
UNLINK	Used to delete link created by LINK
OPTION	Used by the client to ask the server about abailable options

HTTP – message format

- HTTP response message

http://www.w3.org/Protocols/HTTP/HTRESP.html


Status line

Headers

A blank line

Body
(present only in
some messages)

Space

Space

HTTP version

Status code

Status phrase

Response message

explains the status code
in text form

200 OK – request succeeded

301 Moved Permanently – object moved

400 Bad Request – not understood by server

404 Not Found – req. document not found

HTTP – message format (Status code)

100 range	Informational
200 range	Successful request
300 range	Redirectional
400 range	Client Error
500 range	Server Error

HTTP – message format (Status code)

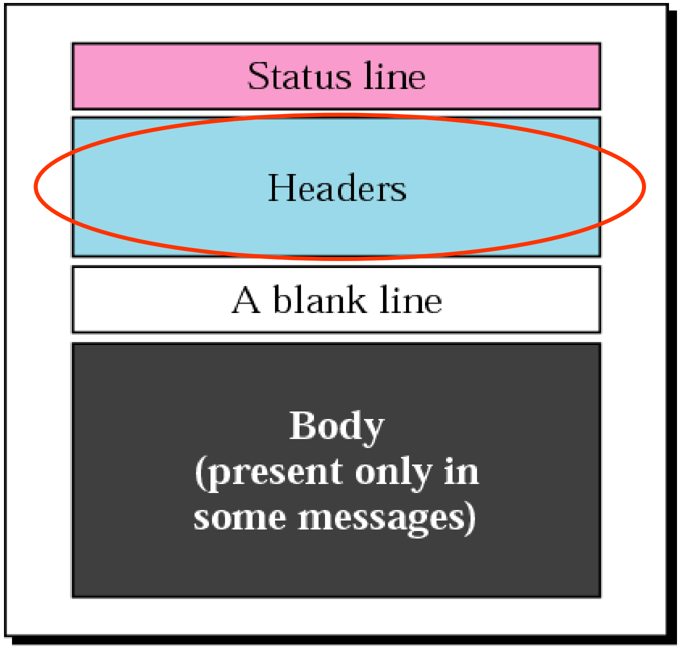
Code	Phrase	Description
Informational		
100	Continue	The initial part of the request has been received and the client may continue with its request
101	Switching	The server is complying with a client request to switch protocols defined in the upgrade header
Success		
200	OK	The request is successful
201	Created	A new URL is created
202	Accepted	The request is accepted, but it is not immediately acted upon
204	No content	There is no content in the body
Redirection		
301	Multiple choices	The requested URL refers to more than one request
302	Moved permanently	The requested URL is no longer used by the server
304	Moved temporarily	The requested URL has moved temporarily

HTTP – message format (Status code)

Code	Phrase	Description
Client Error		
400	Bad Request	There is a syntax error in the request
401	Unauthorized	The request lacks proper authorization
403	Forbidden	Service is denied
404	Not found	The document is not found
405	Method not allowed	The method is not supported in this URL
406	Not acceptable	The format request is not acceptable
Server Error		
500	Internal Server Error	There is an error, such as crash, the server side
501	Not Implemented	The action requested can not be performed
503	Service unavailable	The service is temporarily unavailable, but may be requested in the future

HTTP – message format

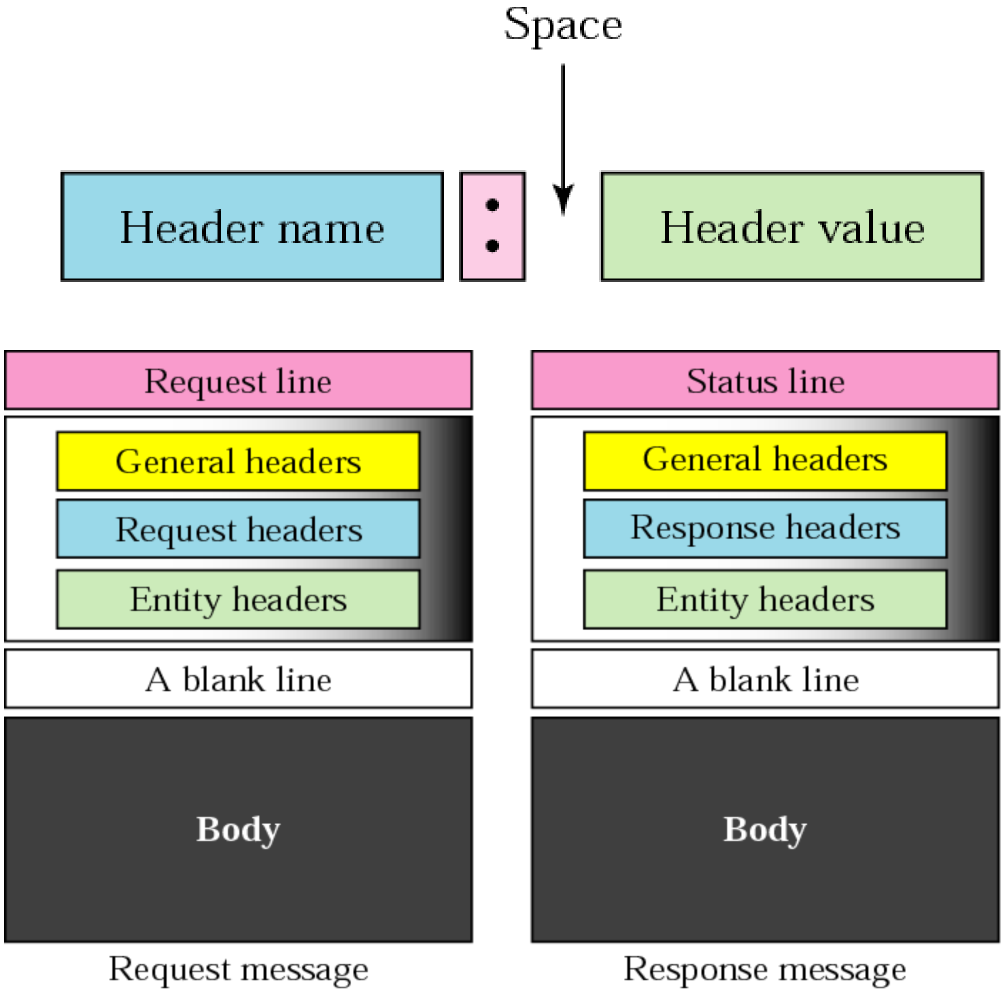
- HTTP response message



Response message

HTTP – message format

- Headers
  - exchange additional information between the client & the server
  - example
    - Date
    - Client’s email address
    - Document age
    - Content length



HTTP – message format

General Header

Header	Description
Cache-control	Specifies information about caching
Connection	Shows whether the connection should be closed or not
Date	Shows the current date
MIME-version	Shows the MIME version used
Upgrade	Specifies the preferred communication protocol

HTTP – message format (Request Header)

Header	Description
Accept	Shows media format the client can accept
Accept-charset	Shows the character set the client can handle
Accept-encoding	Shows the encoding scheme the client can handle
Accept-language	Shows the language the client can accept
Authorization	Shows the permission the client has
From	Shows the email address of the user
Host	Shows the host and port number of the client
If-modified-since	Send the document if newer than specified date
If-match	Send the document only if matches given tag
If-non-match	Send the document only if does not match given tag
If-range	Send only the portion of the document that is missing
If-unmodified-since	Send the document if not changed since specified date
Referrer	Specifies the URL of the linked document
User-agent	Identifies the client program

HTTP – message format (Response Header)

- Specifies the server’s configuration and special information about the request

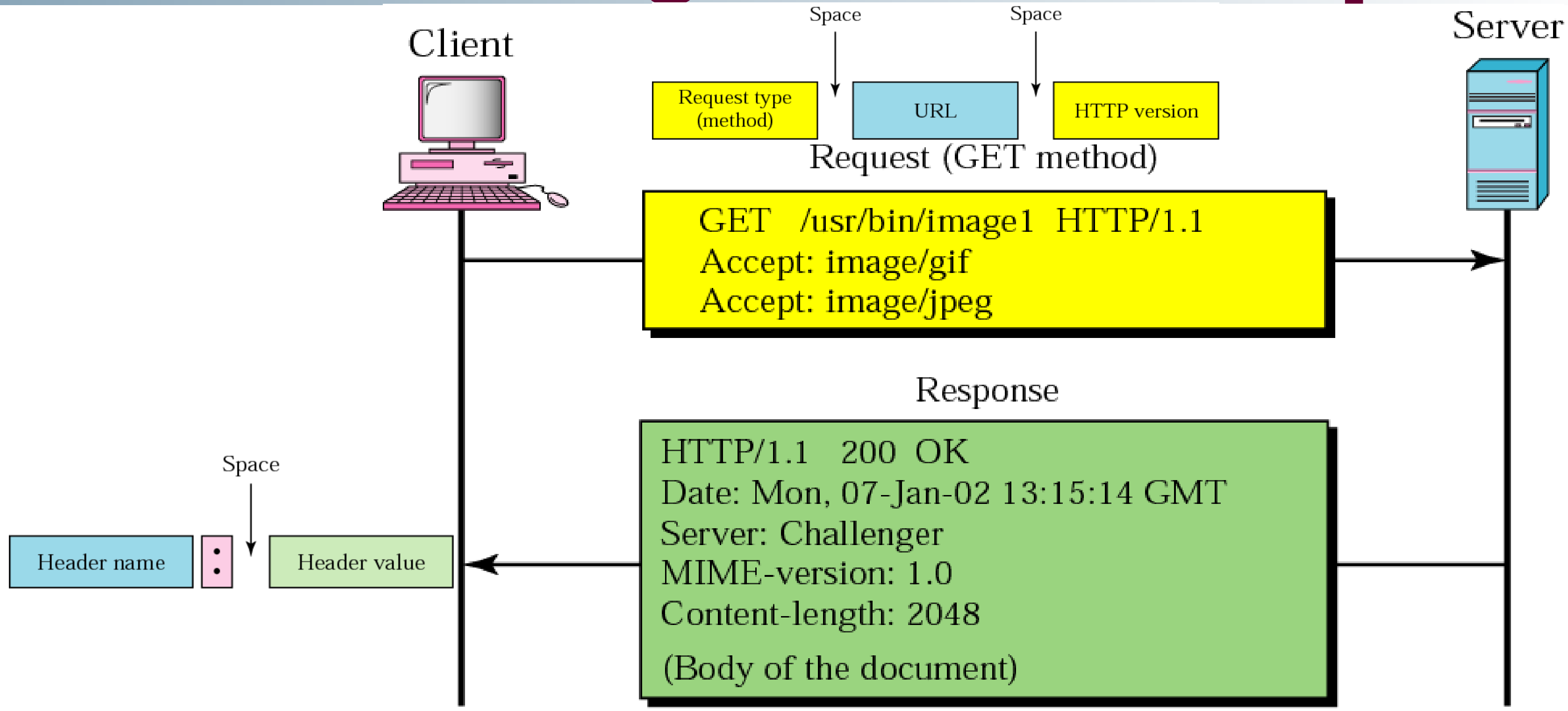
Header	Description
Accept-range	Shows if server accepts the range requested by client
Age	Shows the age of the document
Public	Shows the supported list of methods
Retry-after	Specifies the date after which the server will be available
Server	Shows the server name and version number

HTTP – message format (Entity Header)

- Specifies information about the body

Header	Description
Allow	List of valid methods that can be used with a URL
Content-encoding	Specifies the encoding scheme
Content-language	Specifies the language
Content-length	Shows the length of the document
Content-range	Specifies the range of the document
Content-type	Specifies the media type
Etag	Gives an entity tag
Expires	Gives the date and time when contents may change
Last-modified	Gives the date and time of the last change
Location	Specifies the location of the created of moved document

HTTP messages – an example



Client

Server

Request type (method) | Space | URL | Space | HTTP version

Request (GET method)

```
GET   /usr/bin/image1  HTTP/1.1
Accept: image/gif
Accept: image/jpeg
```

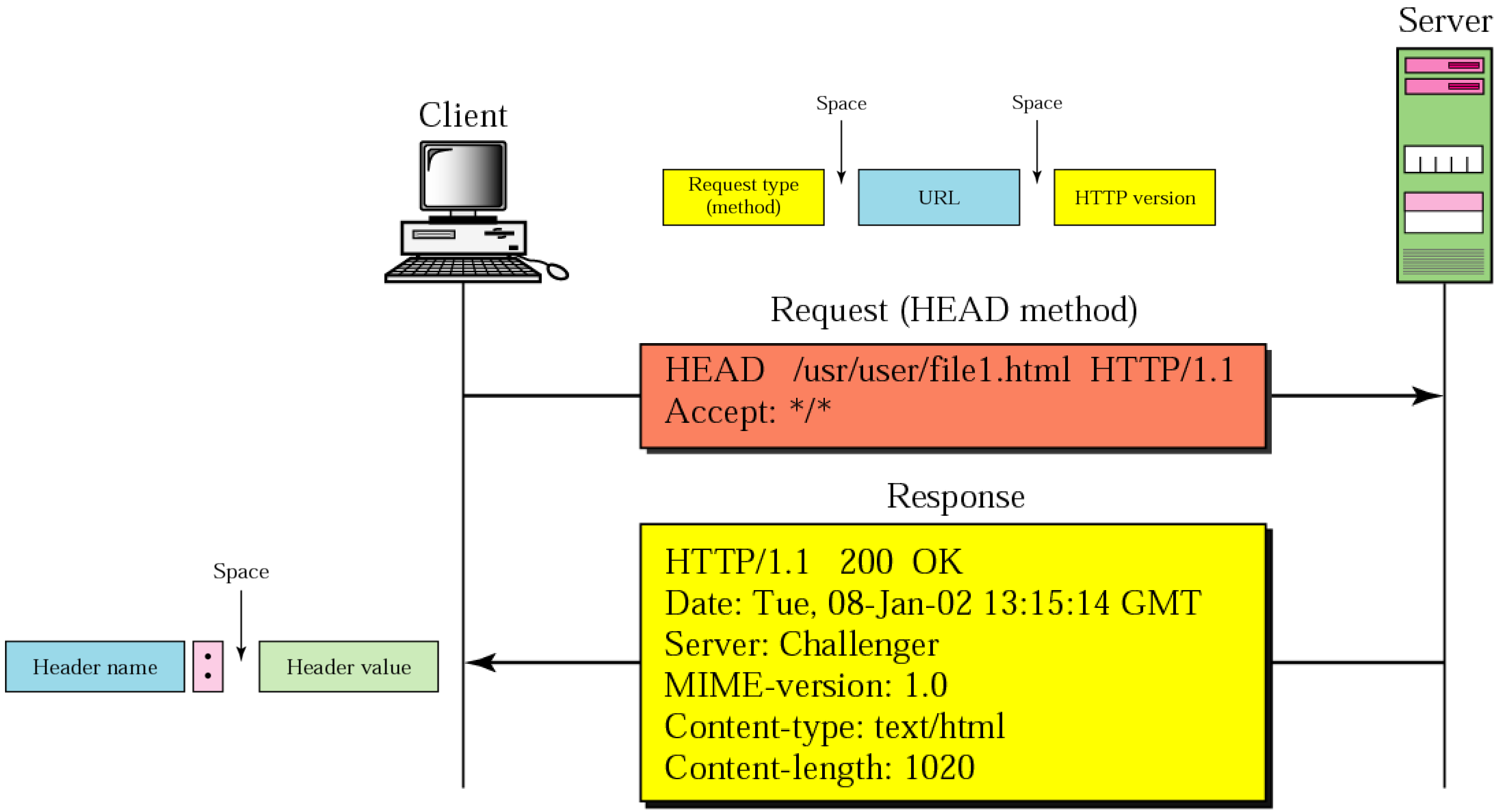
Response

```
HTTP/1.1   200  OK
Date: Mon, 07-Jan-02 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-length: 2048
(Body of the document)
```

Header name | : | Space | Header value

This example retrieves a document.
We use the GET method to retrieve an image with the path /usr/bin/image1.
The request line shows the method (GET), the URL, and the HTTP version (1.1).
The header has two lines that show that the client can accept images in GIF and JPEG format.

HTTP messages – an example



Client

Server

Space

Space

Request type (method)

URL

HTTP version

Request (HEAD method)

```
HEAD   /usr/user/file1.html  HTTP/1.1
Accept: */*
```

Response

```
HTTP/1.1   200  OK
Date: Tue, 08-Jan-02 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-type: text/html
Content-length: 1020
```

Space

Header name

:

Header value

This example retrieves information about a document. We use the HEAD method to retrieve information about an HTML document

Persistent and nonpersistent connections

- Nonpersistent
  - HTTP 1.0
  - one TCP connection for each request/response

  3. the client opens a TCP connection and sends a request
  4. the server sends the response and closes the connection
  5. the client reads data and closes the connection

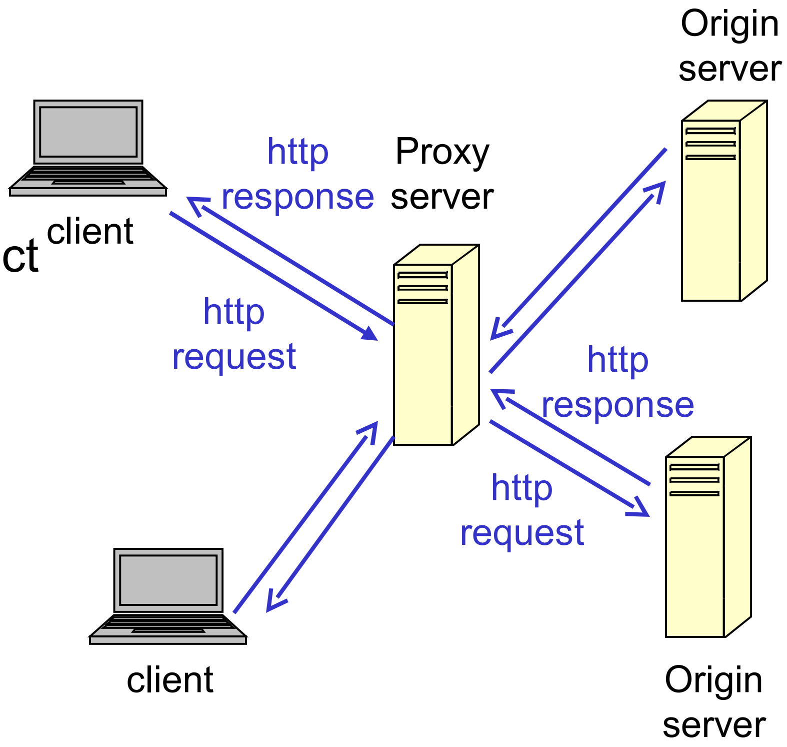
  - each object transfer is independent

- Persistent
  - default for HTTP 1.1
  - the server leaves the TCP connection open for more requests after sending a response
  - client sends requests for all referenced objects as soon as it receives base HTML
    - pipelining
  - fewer RTT

Web caches - Proxy

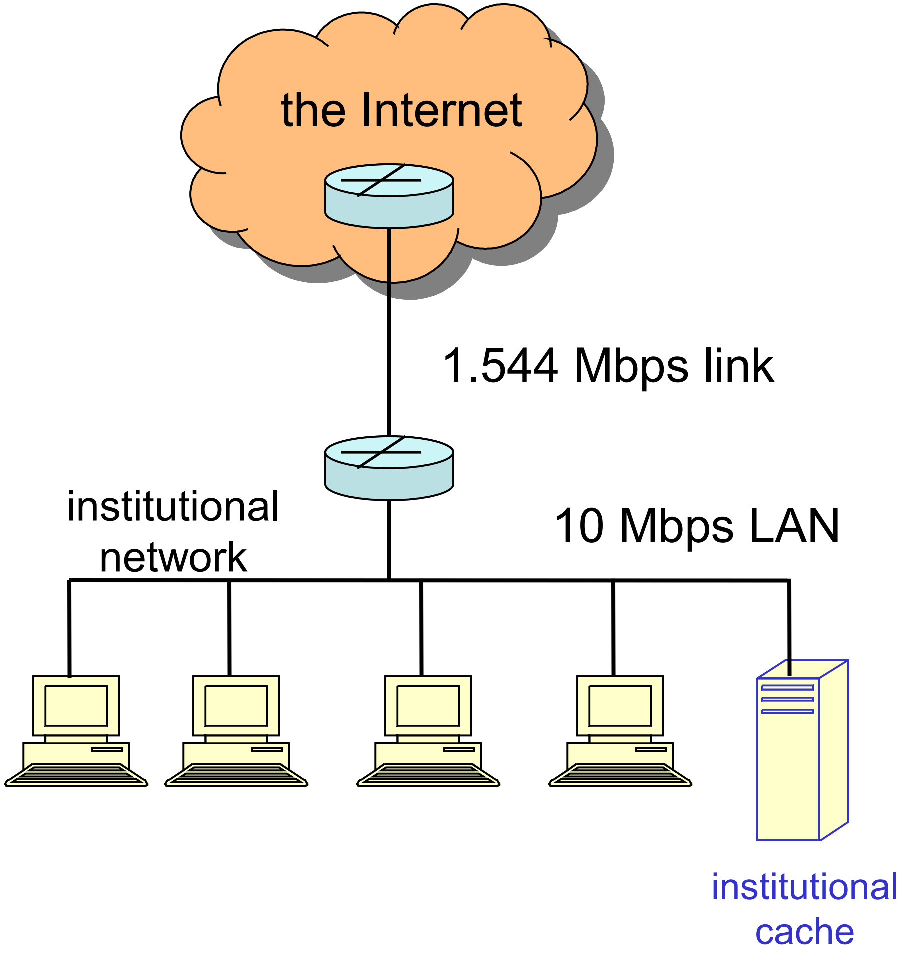
- HTTP supports Proxy servers
- Proxy server
  1. a computer that keeps copies of responses to recent requests
- Goal: satisfy client request without involving original server

- client sends all http requests to the proxy server
- if object at web cache sends the object in http response
- else request object from the origin server, then returns http response to client



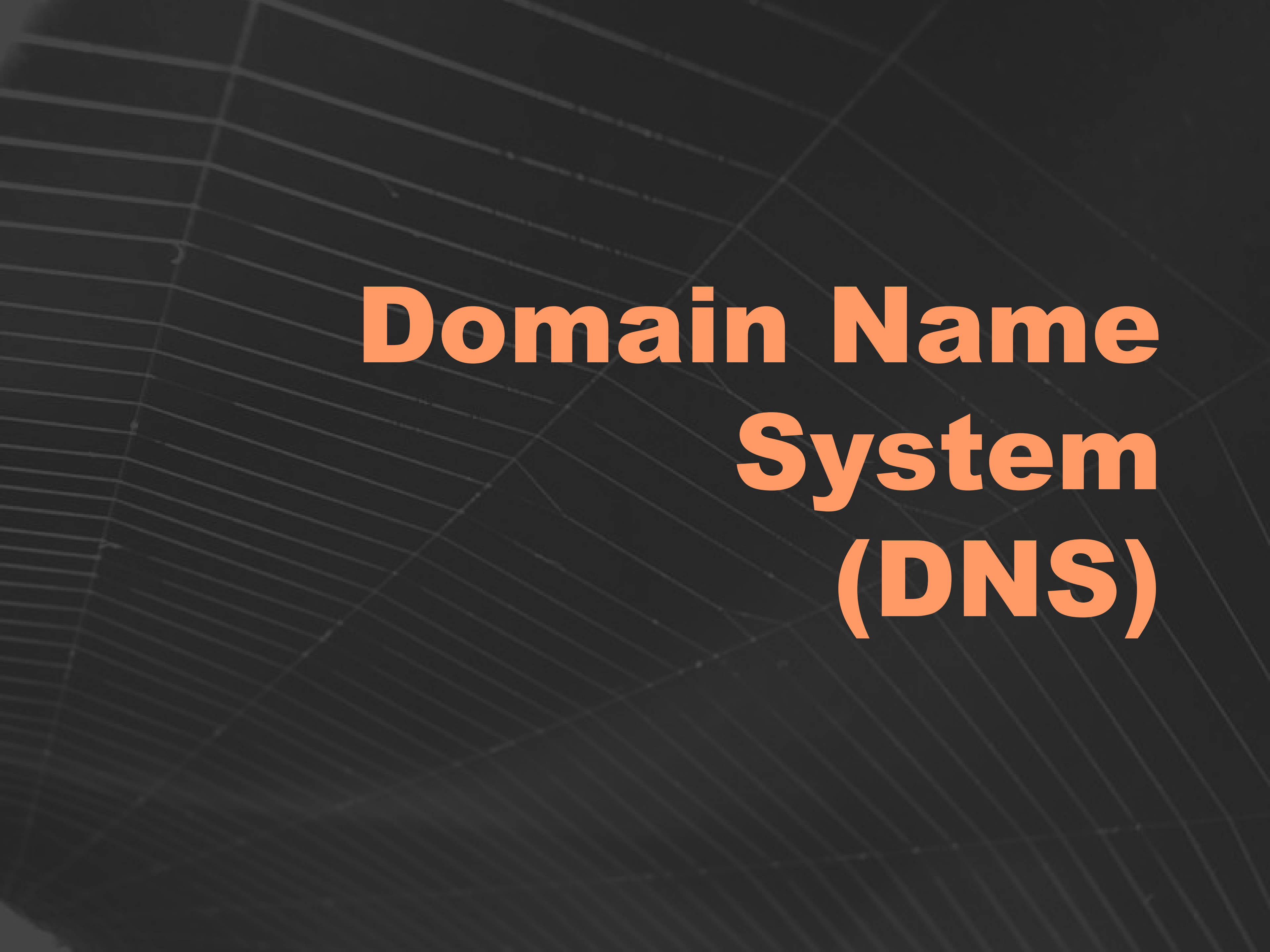
Why Web caching?

- Assume: cache is close to a client (in the same network)
  - smaller response time (improved latency)
  - decrease traffic to distance servers
    - link out of ISP network is often a bottleneck



Consistency of Web caching

- The major issue: How to maintain consistency?
- Two ways:
  - Pull
    - Web caches periodically pull the web server to see if a document is modified
  - Push
    - Whenever a server gives a copy of a web page to a web cache, they sign a lease with an expiration time;
      if the web page is modified before the lease, the server notifies the cache



Domain Name System (DNS)

Domain Name System (DNS)

- TCP/IP uses IP address—difficult to remember
- Solution: use names instead of IP addresses
- Used to map a name to an IP address & vice-versa
  - example:
    - www.itd.jusl.ac.in -> 203.197.107.107
    - www.yahoo.com -> 209.73.186.238
    - www.google.com -> 64.233.189.104

Domain Name System (DNS)

- Possible solution:
  - a host file, two columns: name & address
  - Every host stores this file
  - Update periodically from master file
- Problems:
  - Host file would be too large to store
  - Updation problem
  - Solution
    - Store this host file centrally
    - Problem: Huge amount of traffic

Domain Name System (DNS)

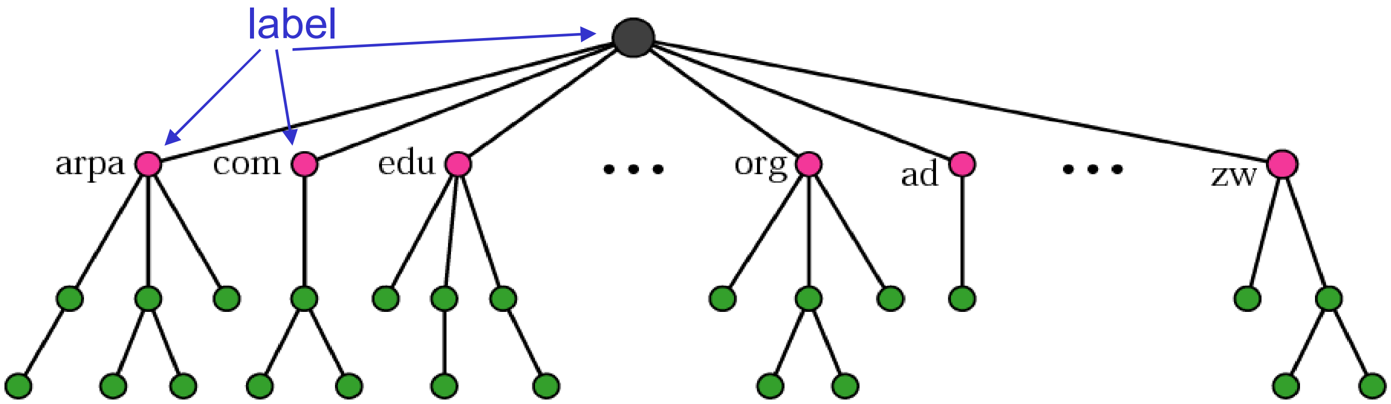
- Solution for huge amount of information:
  - divide it into smaller parts and store each part on different computer—called DNS Server
  - Host needs name resolution contacts nearest DNS Server
  - if one DNS server doesn't know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned.

Domain Name System (DNS)

- Name space
  - flat name space
    - Centrally controlled to avoid ambiguity and duplication
      - cannot be used in larger networks like the Internet
  - hierarchical name space
    - each name is made of several parts
    - central authority only partially control names (www.jadavpur.edu)
      - www.itd.jusl.ac.in
      - www.cse.iitk.ac.in

Domain Name Space

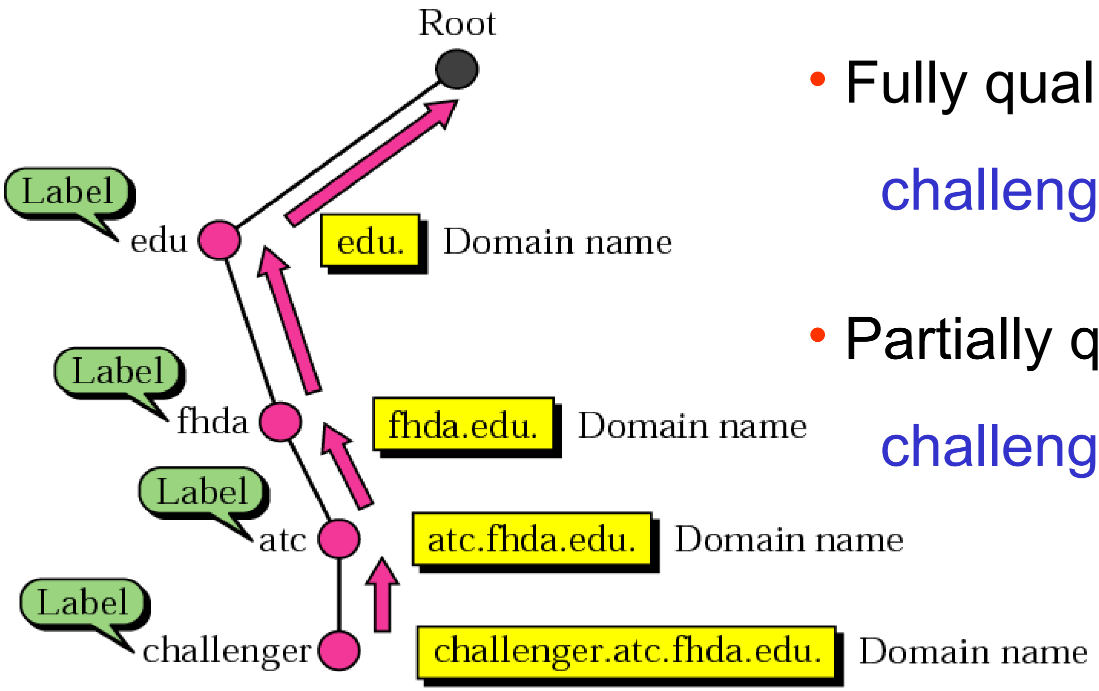
- designed to have a hierarchical name space
- tree structure (maximum 128 levels)



- all labels (maximum of 63 characters) have different names
    - uniqueness of the domain names
- root label - null

Domain name

- Domain name – a sequence of labels separated by dots
- read from the node up to the root
- full domain name ends with the null



- Fully qualified domain name

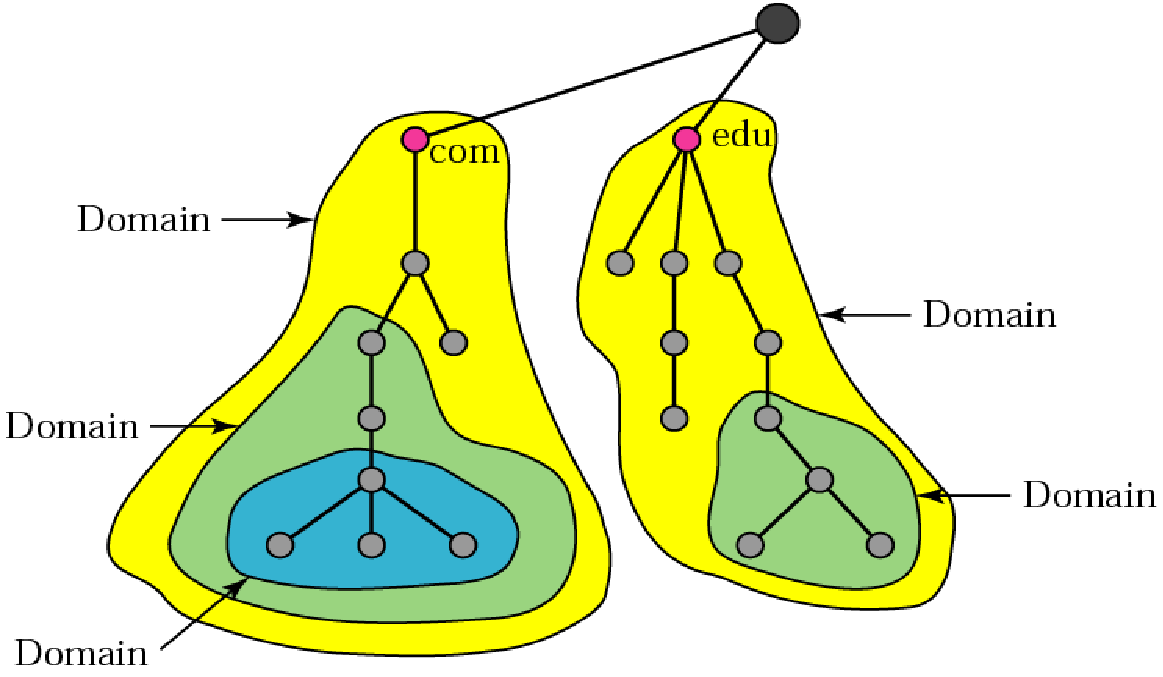
  challenger.atc.fhda.edu.

- Partially qualified domain name

  challenger

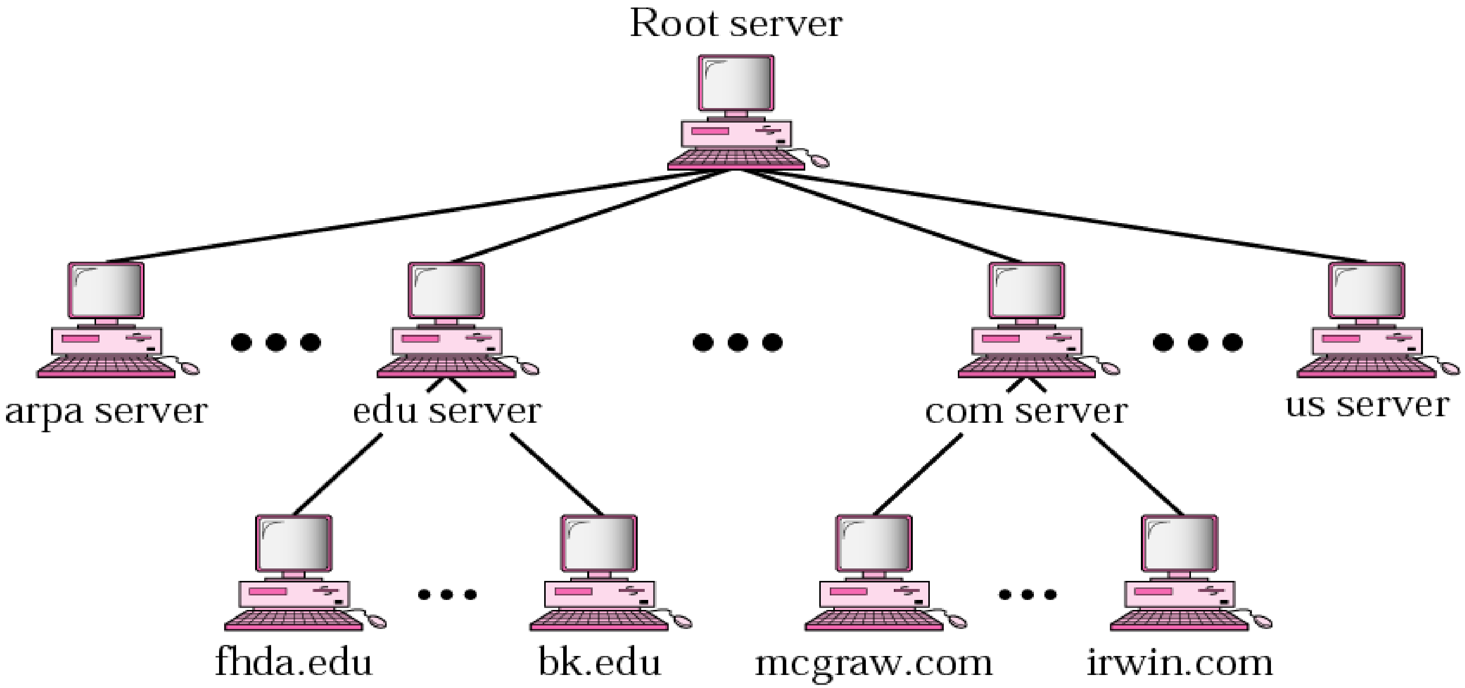
Domain

- A sub-tree of the Domain Name Space
- Name of a domain is the domain name of the node at the root of the subtree



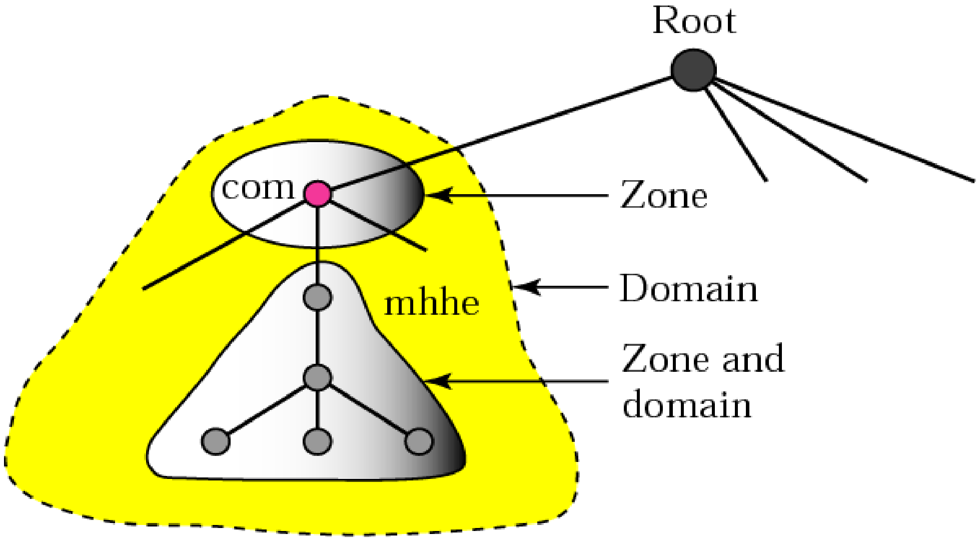
Distribution of name space

- Storing all naming information in one computer is
  - unreliable
  - inefficient
    - Responding to requests from all over the world places a heavy load on the system
- Hierarchy of Name Servers



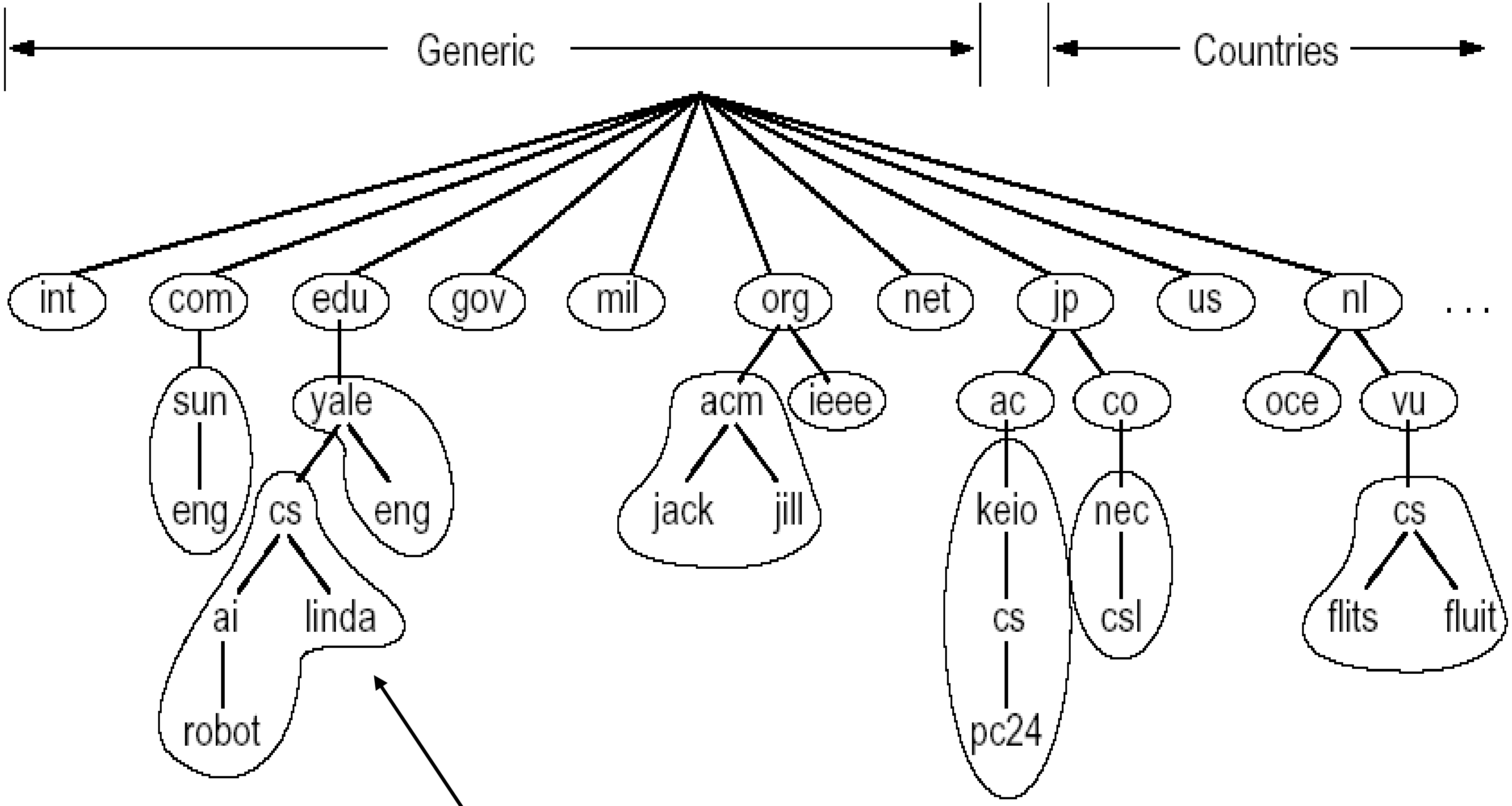
DNS zones, servers -

- original server keeps a sort of a reference to the lower-level servers



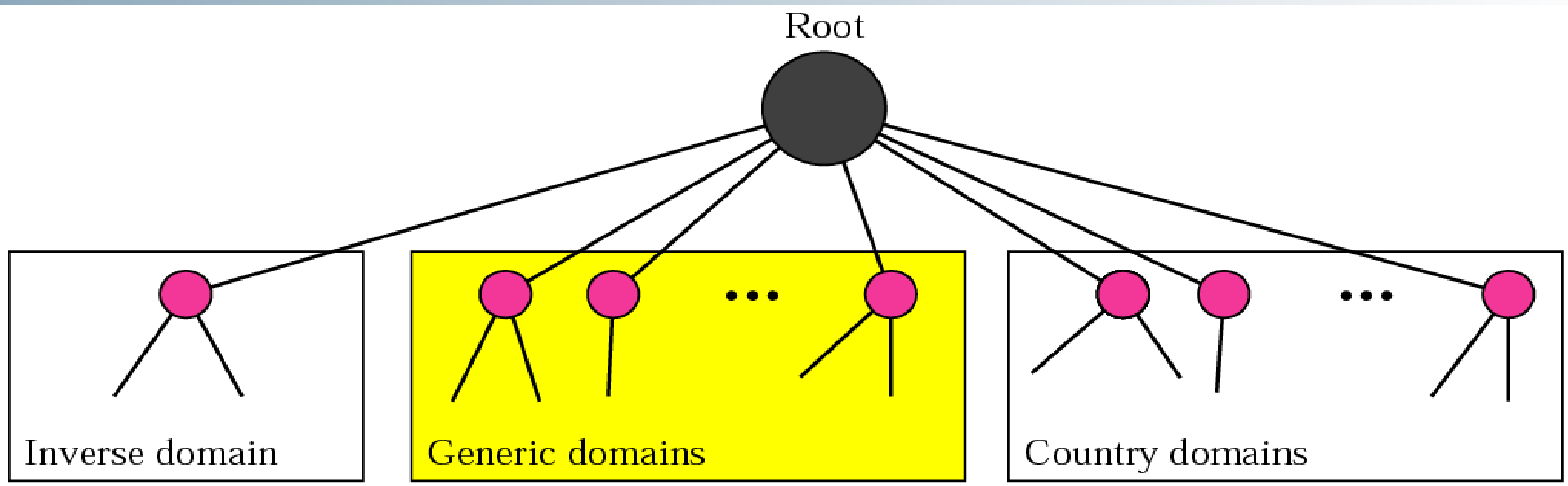
- Root servers
  - zone is a whole tree
  - 13 in the world
- Primary server
  - loads the information about the the zone from the disk
- Secondary server
  - loads the info from the primary server
    - redundancy against failure

Zones (cnt'd)



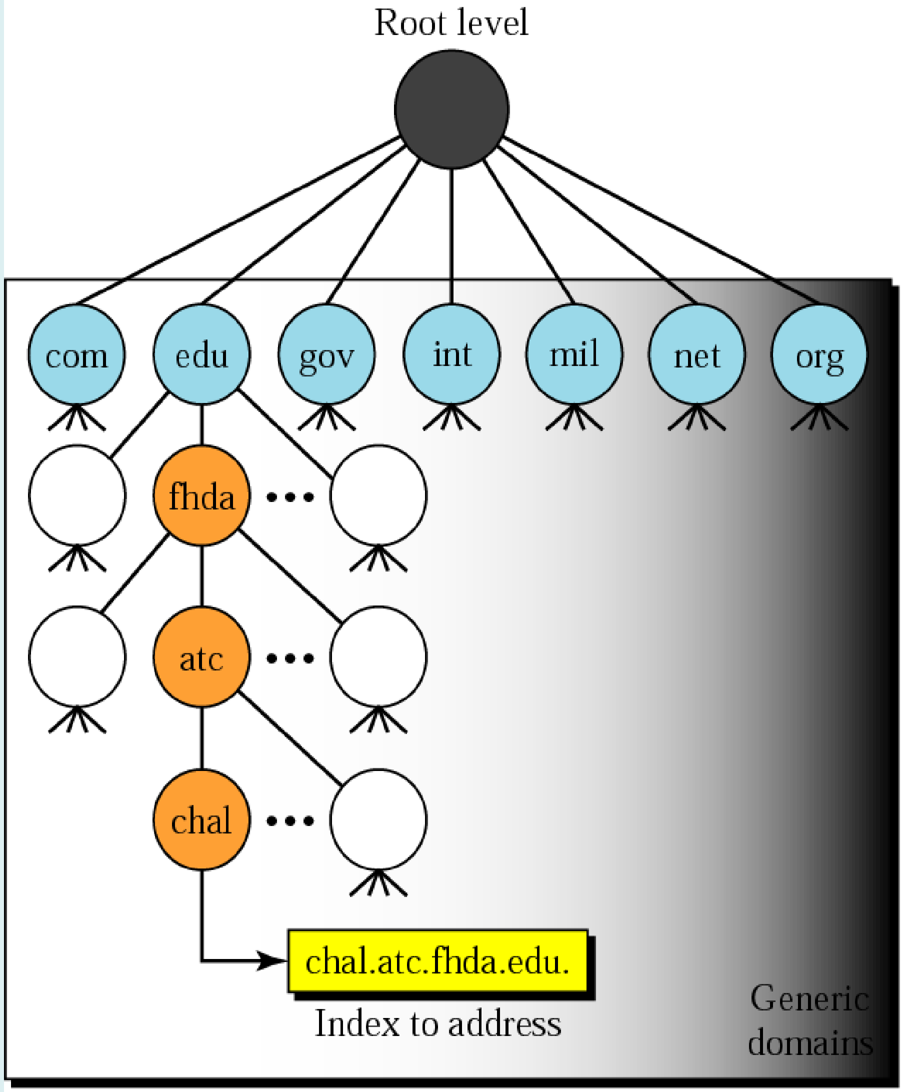
separate name
server

DNS in the Internet



- Generic domains
  - registered host according to their generic behavior
- Inverse domain
  - used to map an address to a name
- Country domains
  - the same format as in generic domain just 2 character format
    - us; nl; jp; fr; in

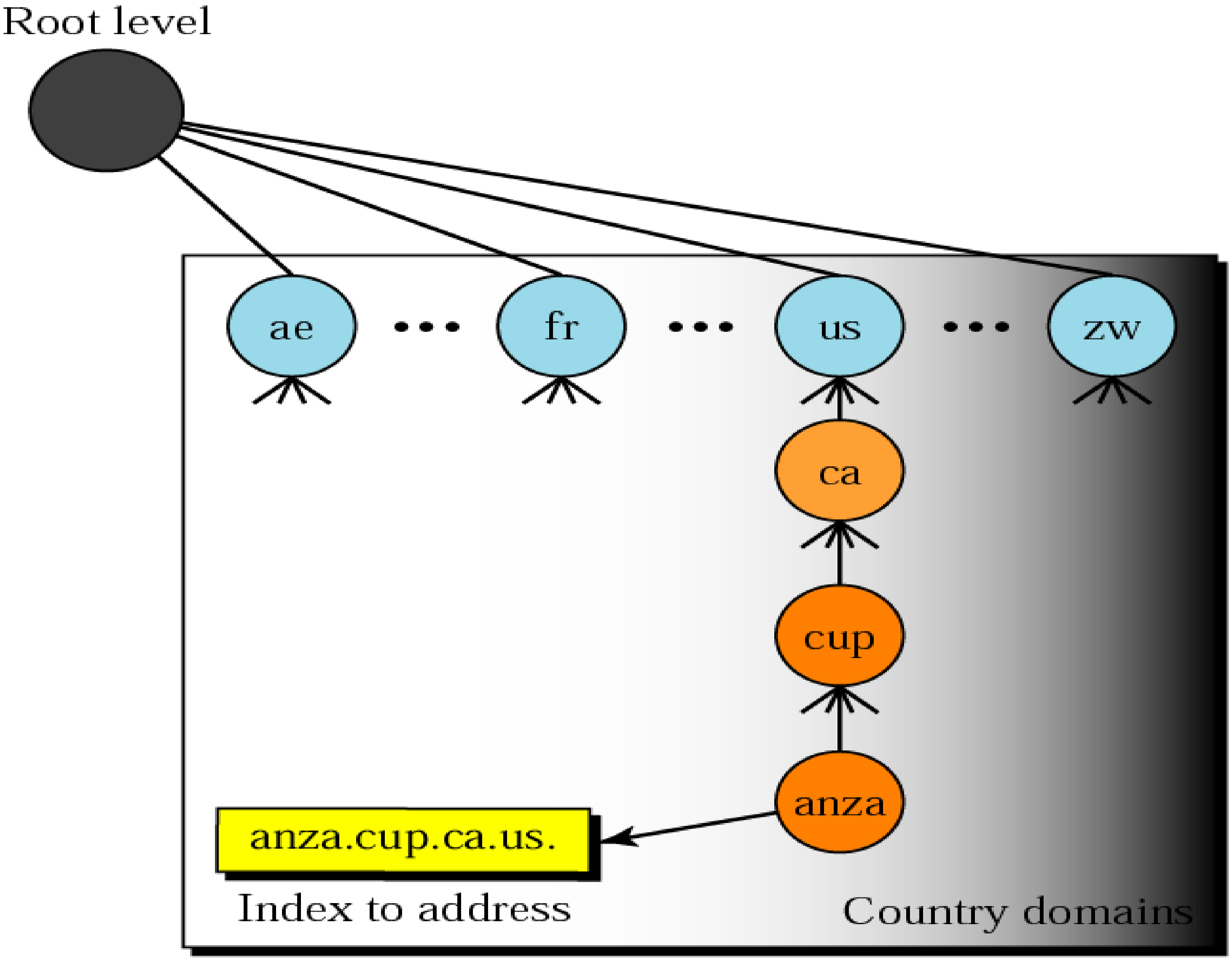
Generic domain



Label	Description
com	Commercial organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	Military groups
net	Network support centers
org	Nonprofit organizations

aero	Airlines and aerospace companies
biz	Businesses or firms (similar to ‘com’)
coop	Cooperative business organizations
info	Information service providers
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
pro	Professional individual organizations

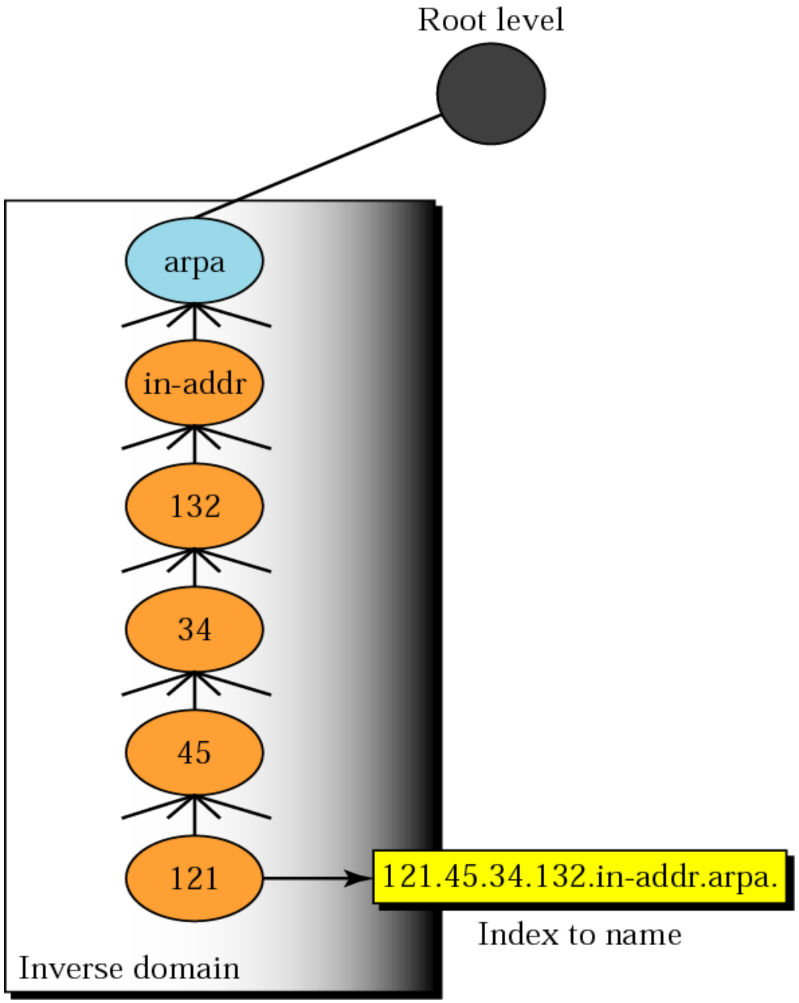
Country domains



Inverse domain

Example: a server wishes to determine whether the client is on the authorized list

- First-level node arpa for historical reasons
- The servers are also hierarchical
- Domain looks inverted compared to a generic or country domain

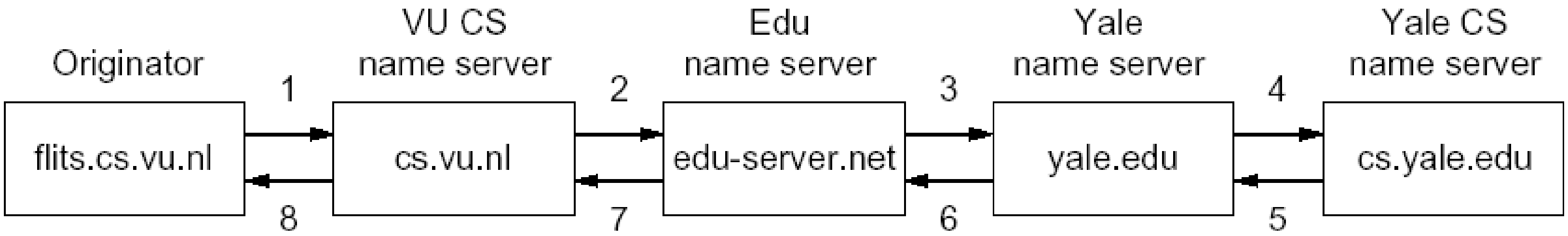


Resolution

- Mapping a name to an address or vice-versa
- Resolver
  - DNS client
  - When a host needs to map an address to a name it calls resolver that in turn access the nearest DNS server with a mapping request
  - A server either
    - responds directly with an info, or
    - refers the resolver to other servers
    - asks other servers to provide info
- Recursive resolution
- Iterative resolution

Recursive resolution

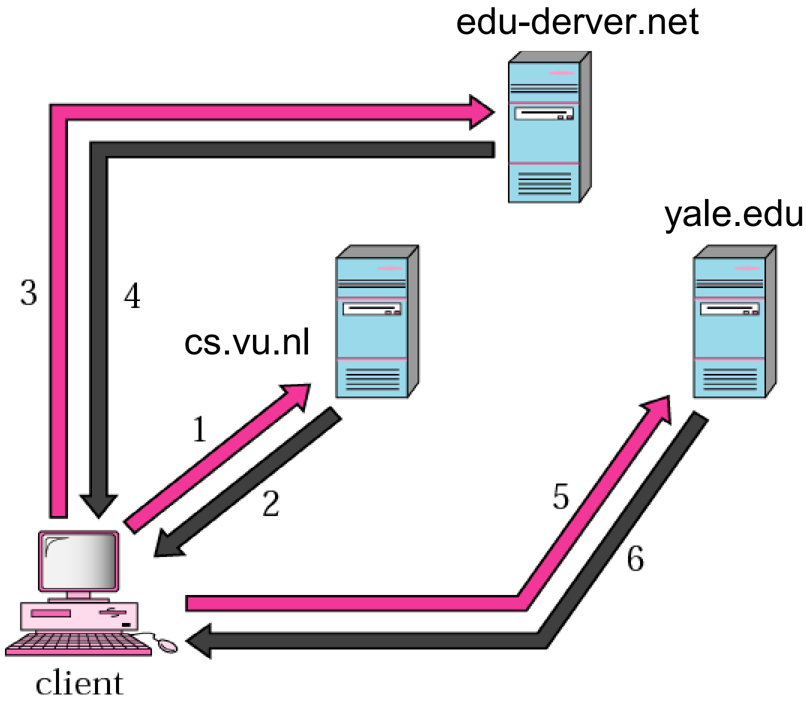
flits.cs.vu.nl -> linda.cs.yale.edu



- if the server is the authority for the domain name it checks its data base and responds, otherwise
- it sends a request to another server…

Iterative resolution

- The server returns either IP requested address or the IP address of the server it thinks can resolve the querry



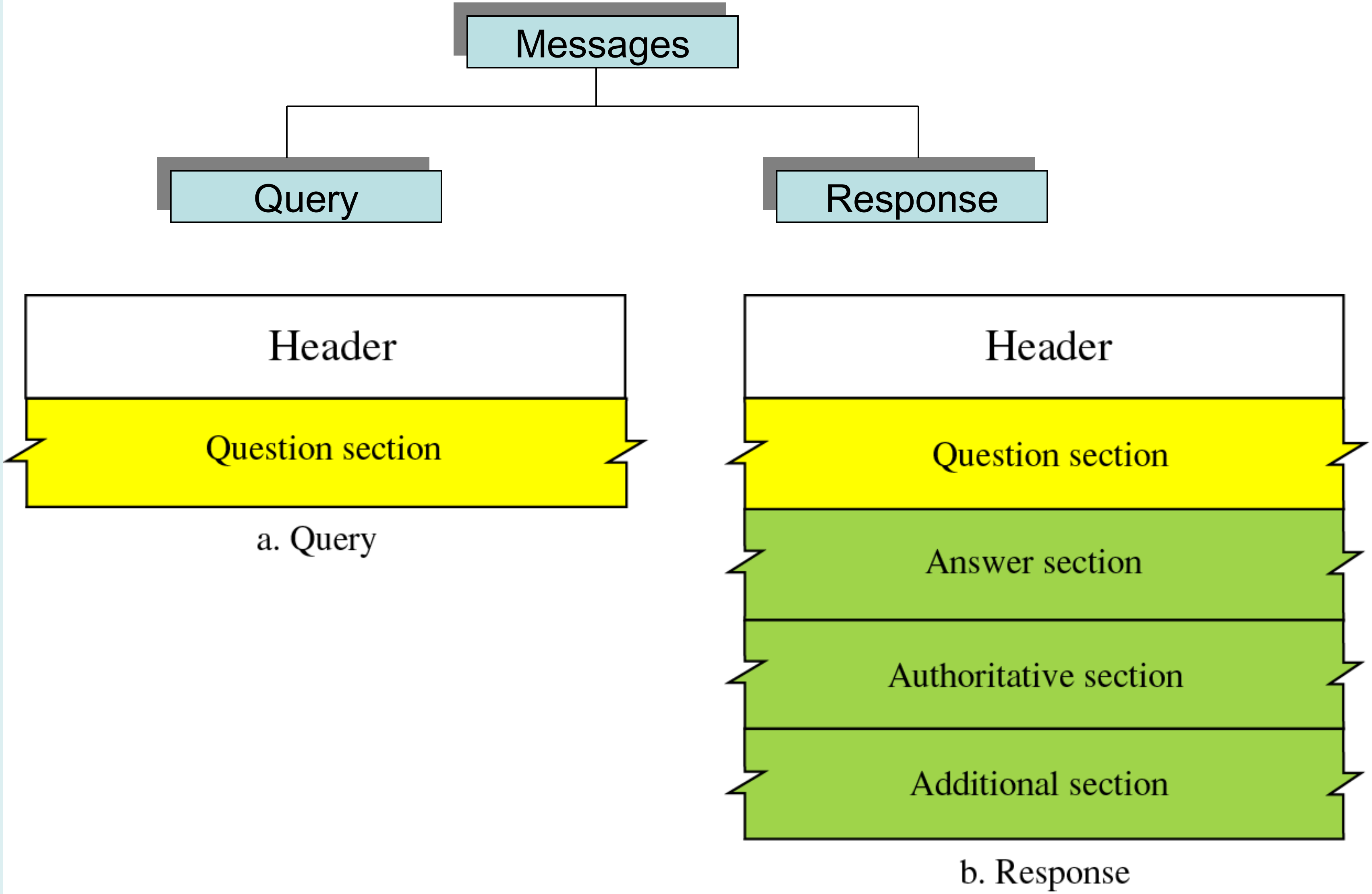
Dynamic DNS

- What if a new host joins the network or a host is removed or an IP address is changed?
- DNS master file also has to be changed
- Changes so dynamic – a problem!

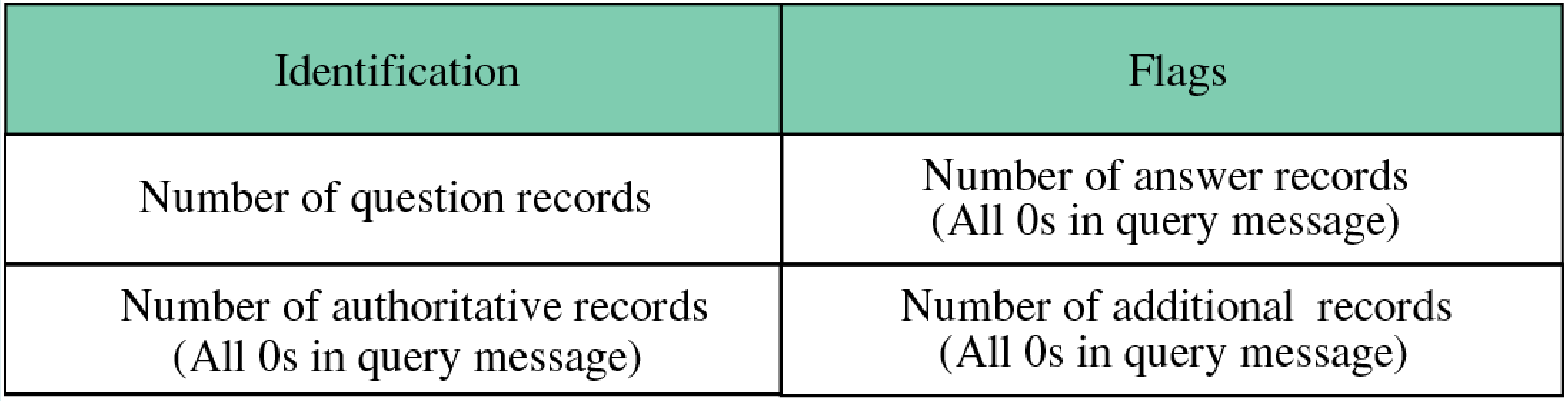
- Dynamic Domain Name System
- When a binding between IP address & host name is determined (usually) DHCP informs DNS server

- Encapsulation
  - DNS can use either UDP or TCP, using the well-known port 53

DNS Mesaages



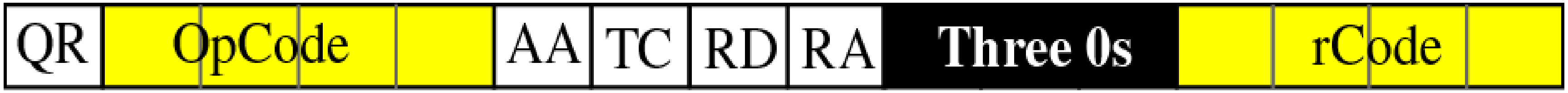
Header Format



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)

- Identification
  - 16-bit field used by the client to match response with the query

Flag Fields



QR	OpCode	AA	TC	RD	RA	Three 0s	rCode

- QR: Query/Response
- OpCode: 0 standard, 1 inverse, 2 server status
- AA: Authoritative
- TC: Truncated
- RD: Recursion Desired
- RA: Recursion Available
- rCode: Status of the error

Resource Records

- Five tupple in the form
  - Domain_name Time_to_live Type Class Value

Type	Meaning	Value
SOA	Start of Authority	Parameters for this Zone
A	IP address of a host	32-bit Integer
MX	Mail Exchange	Priority, domain willing to accept mail
NS	Name Server	Name of a Server for this domain
CNAME	Canonical Name	Domain Name
PTR	Pointer	Alias for an IP address
HINFO	Host Description	CPU and OS in ASCII
TXT	Text	Uninterrupted ASCII text

Resource Records

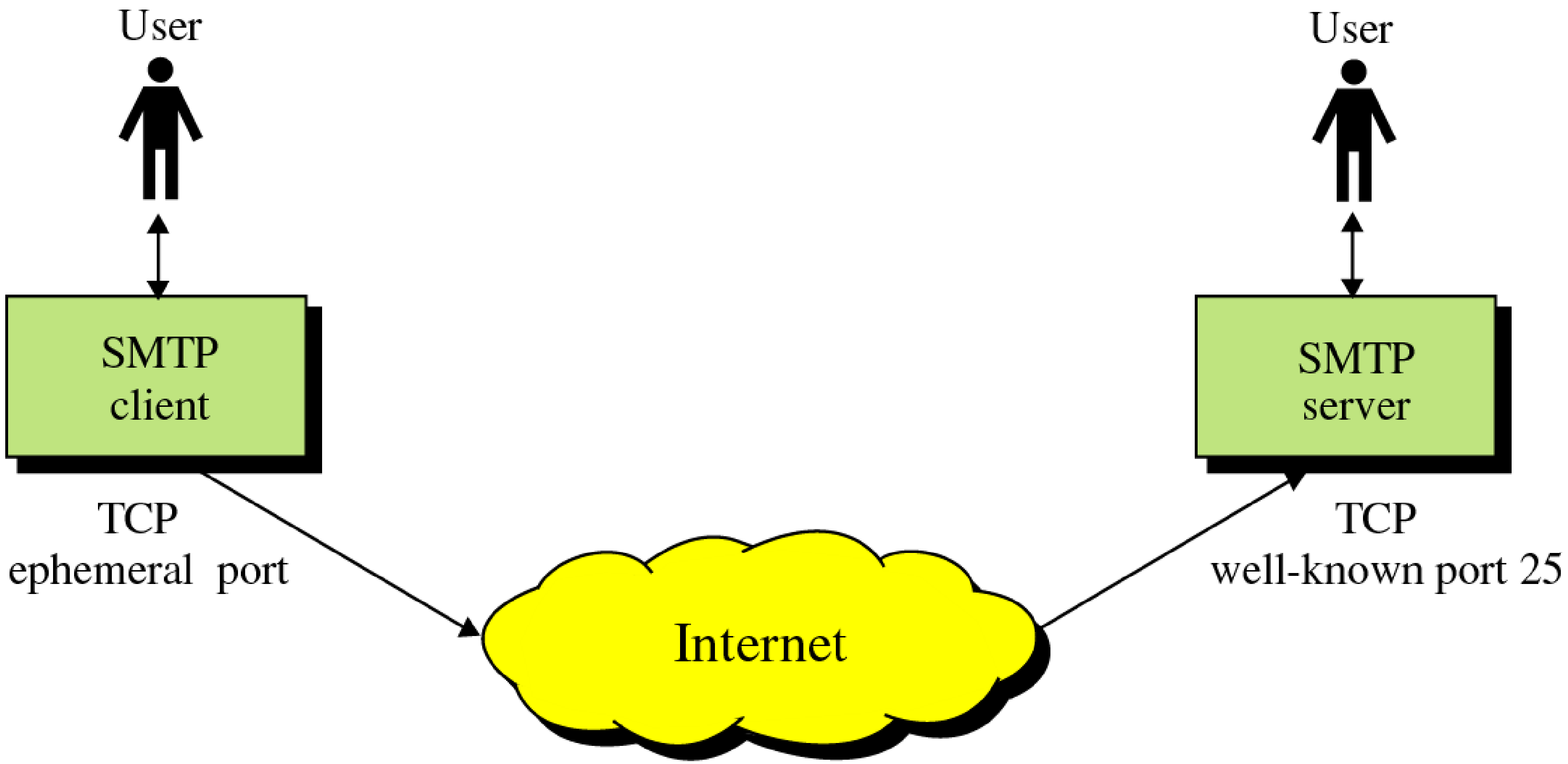
```
$TTL    86400
@       IN      SOA     rose.itd.jusl.ac.in. rose.itd.jusl.ac.in. (
                          2006062101
                          3H
                          15M
                          1W
                          1D )

                IN      NS              rose.itd.jusl.ac.in.
;               IN      NS              galaxy.itd.jusl.ac.in.
                IN      MX              1 rose.itd.jusl.ac.in.
rose            IN      A               203.197.107.107
www             IN      CNAME           rose.itd.jusl.ac.in.
mail            IN      CNAME           rose
dns             IN      CNAME           rose
gateway         IN      CNAME           rose
hporacle        IN      A               172.16.6.97
lotus           IN      A               172.16.6.107
galaxy          IN      A               172.16.6.108
nfs             IN      CNAME           lotus
dhcp            IN      CNAME           lotus
nis             IN      CNAME           lotus
```

Simple Mail Transfer Protocol(SMTP)

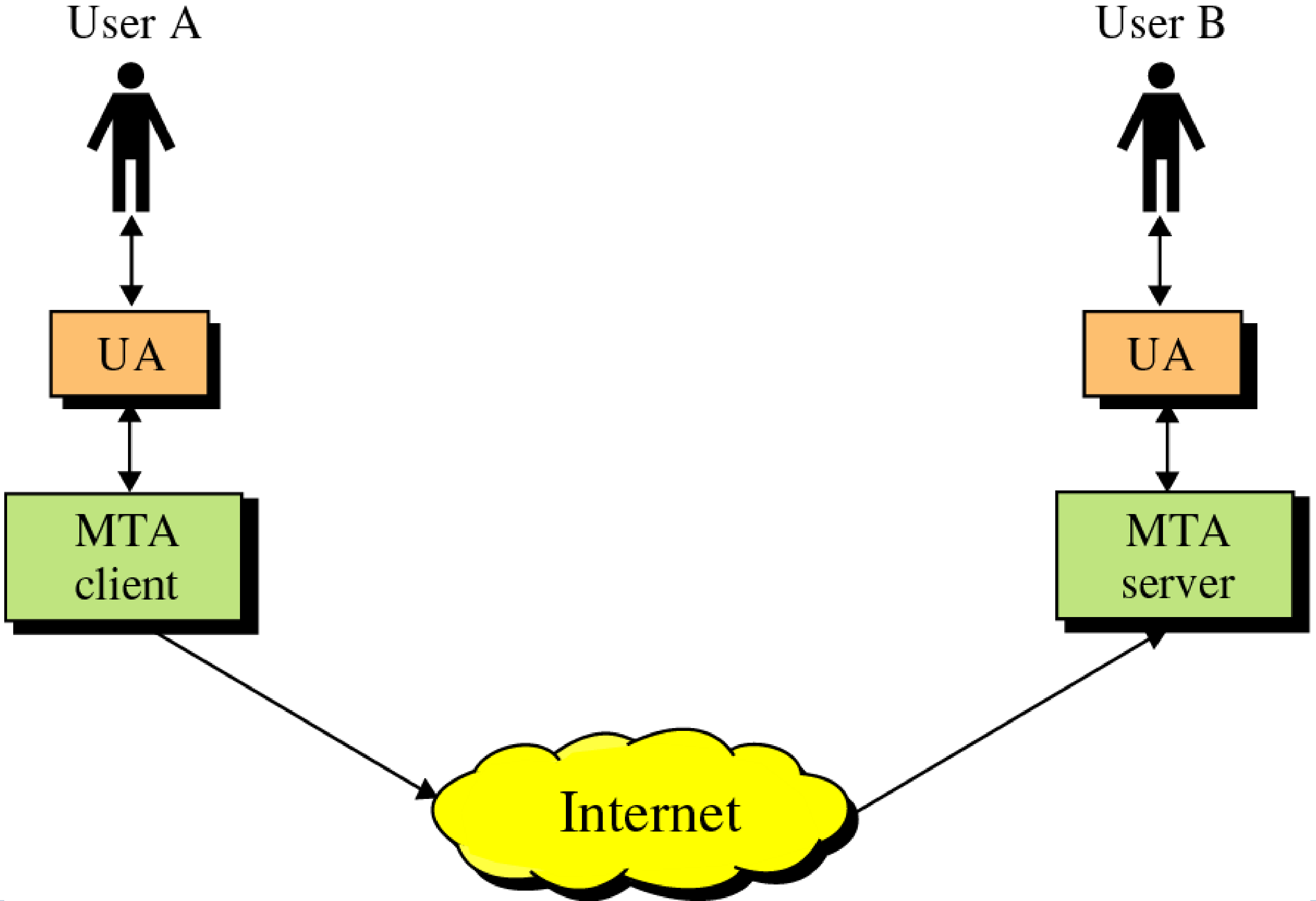
SMTP

- Provides electronic mail(email) services using email addresses
  - Sending a single message to one or more recipients
  - Sending messages that include text, graphics, voice and video
- Asynchronous service



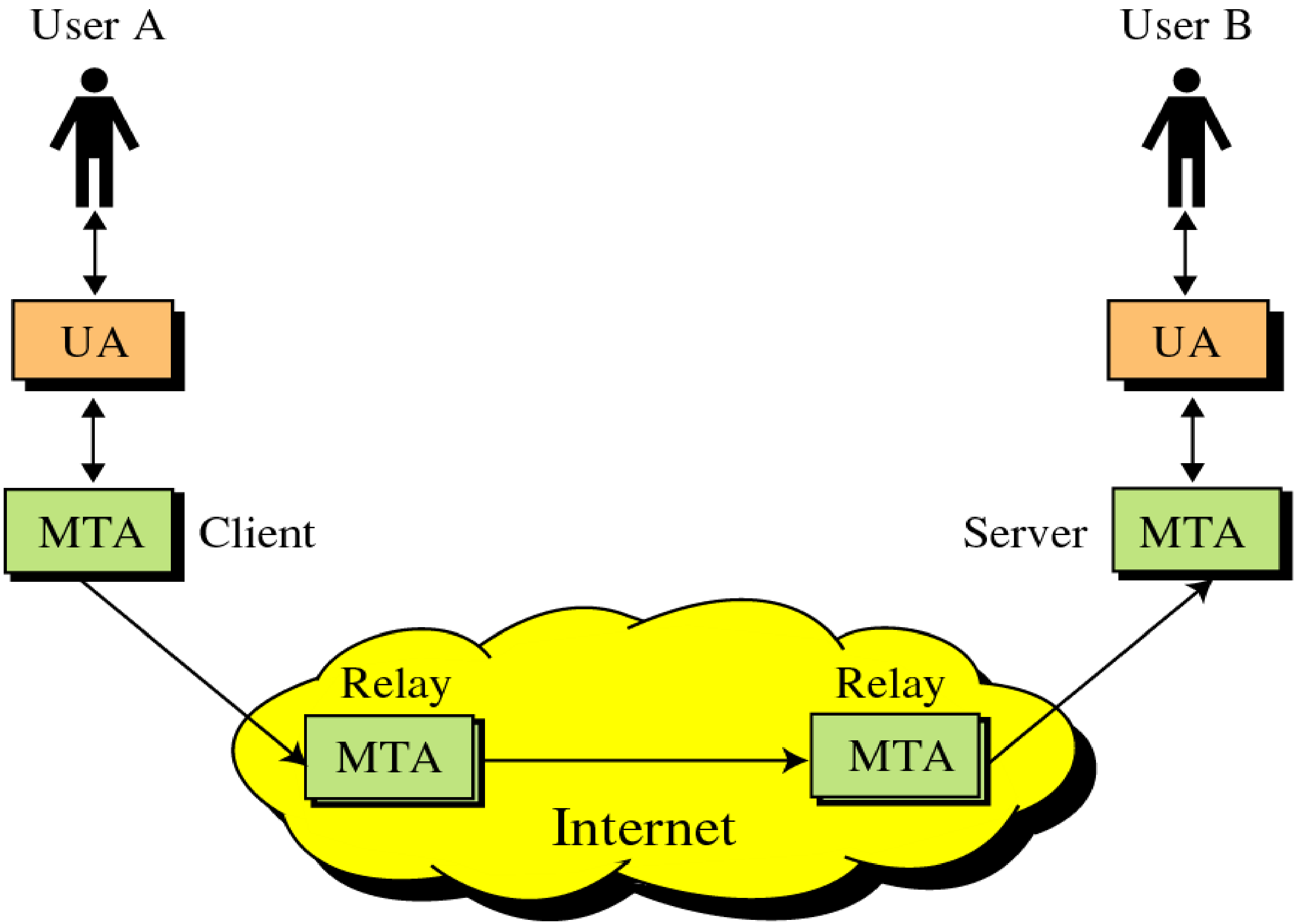
SMTP

- SMTP Client/Server
  - User Agent(UA)
  - Mail Transfer Agent(MTA)



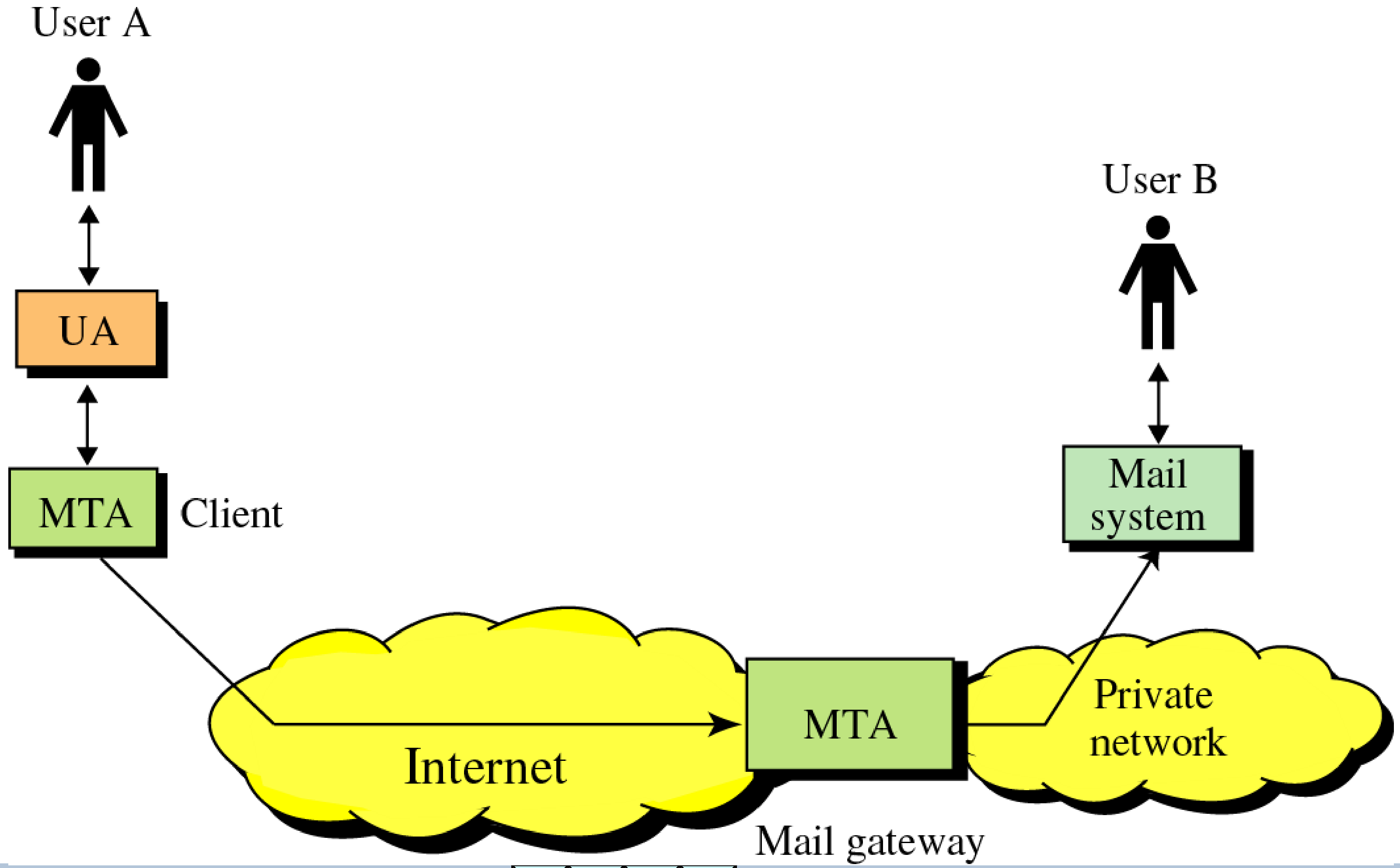
SMTP

- Relay MTA—used to store mail in an intermediate stage



SMTP

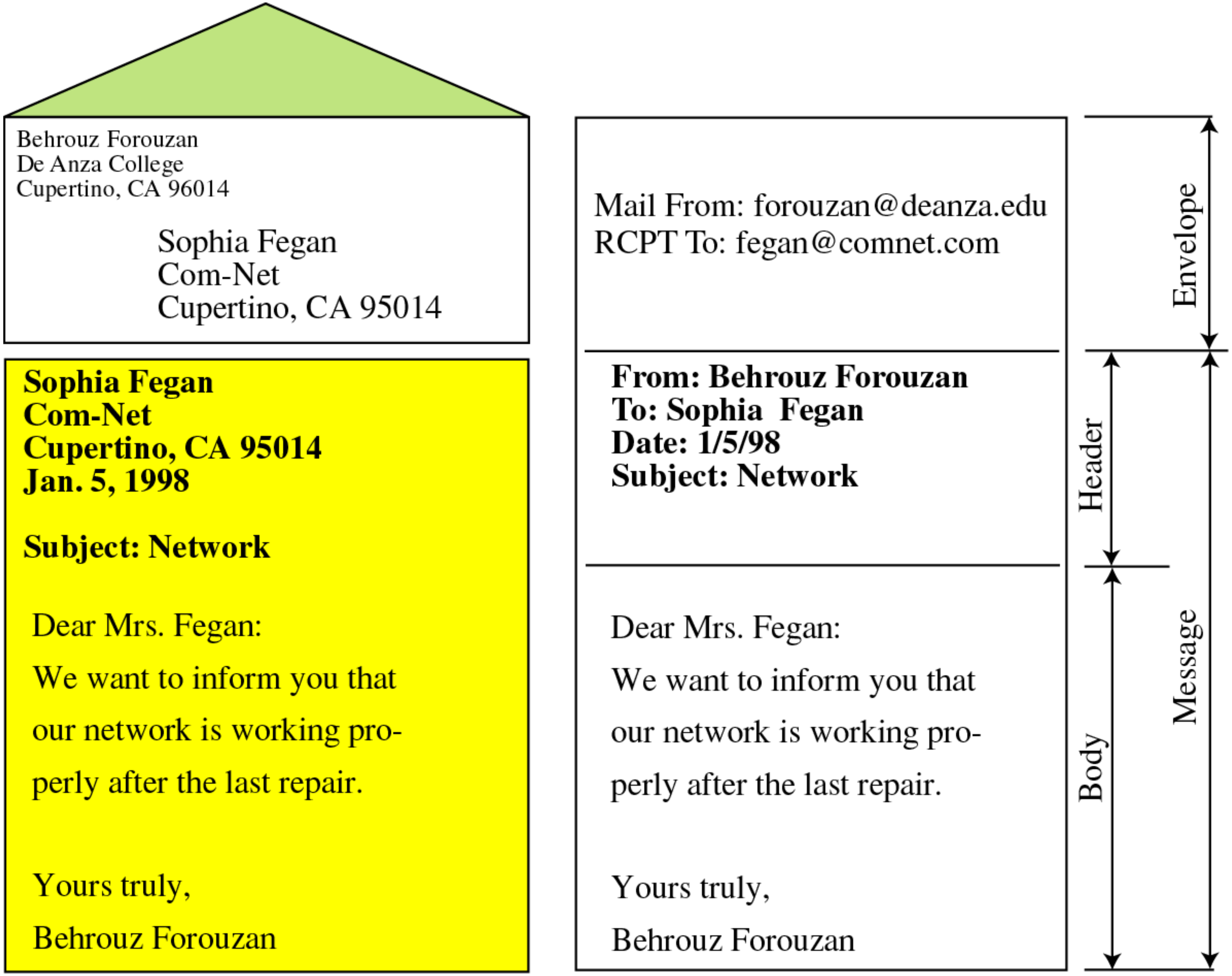
- Mail Gateway—used when either side does not use TCP/IP protocol



SMTP(User Agent)

- Defined in SMTP without any implementation details
- Normally a program that provides an interface to send and receive mails
- Example
  - Elm, Pine, MH, Berkley Mail, Zmail, Mush
  - Eudora, Webmail etc.
- Sending Mail
  - Envelop
  - Message
    - Header
    - Body
- Receiving Mail
  - UA checks mailbox periodically

email format



Behrouz Forouzan
De Anza College
Cupertino, CA 96014

Sophia Fegan
Com-Net
Cupertino, CA 95014

Sophia Fegan
Com-Net
Cupertino, CA 95014
Jan. 5, 1998

Subject: Network

Dear Mrs. Fegan:

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

Yours truly,

Behrouz Forouzan

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

From: Behrouz Forouzan
To: Sophia Fegan
Date: 1/5/98
Subject: Network

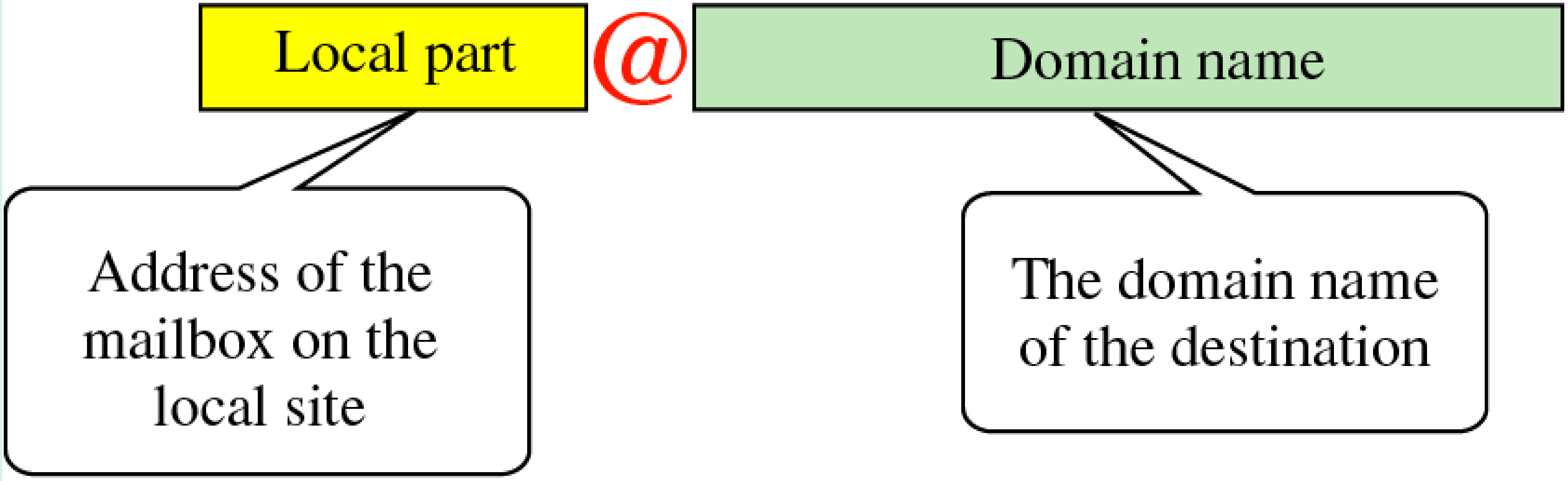
Dear Mrs. Fegan:

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

Yours truly,

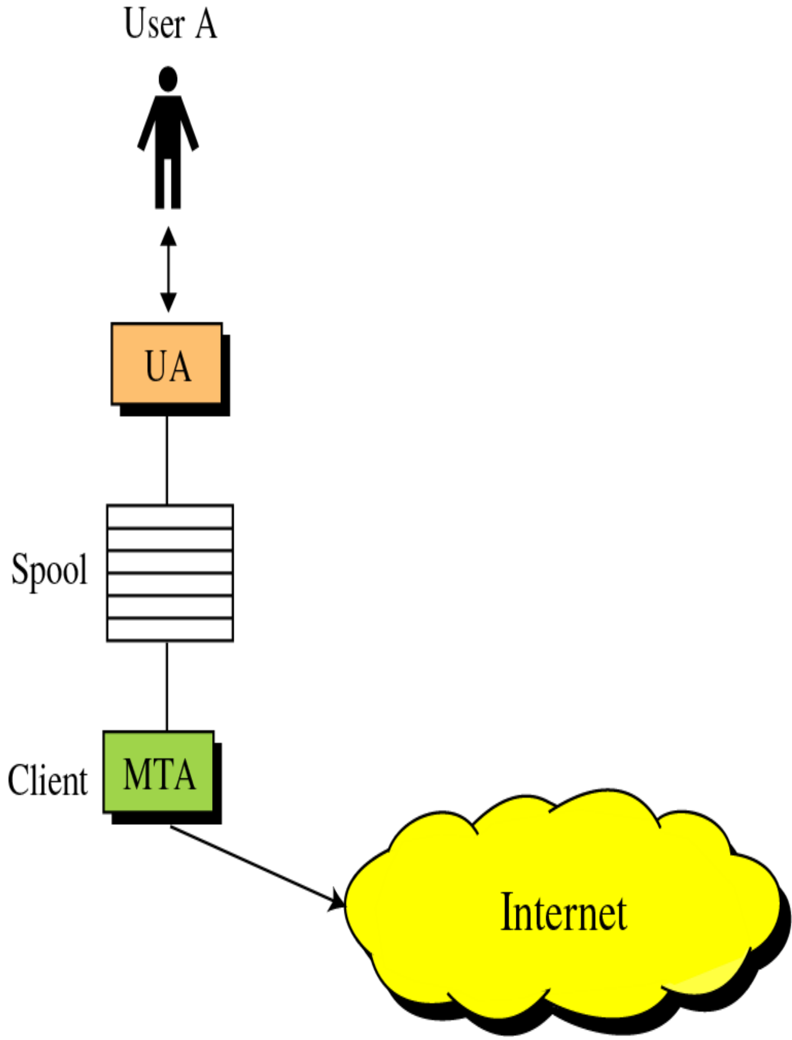
Behrouz Forouzan

Addresses


Local part
@
Domain name
Address of the mailbox on the local site
The domain name of the destination

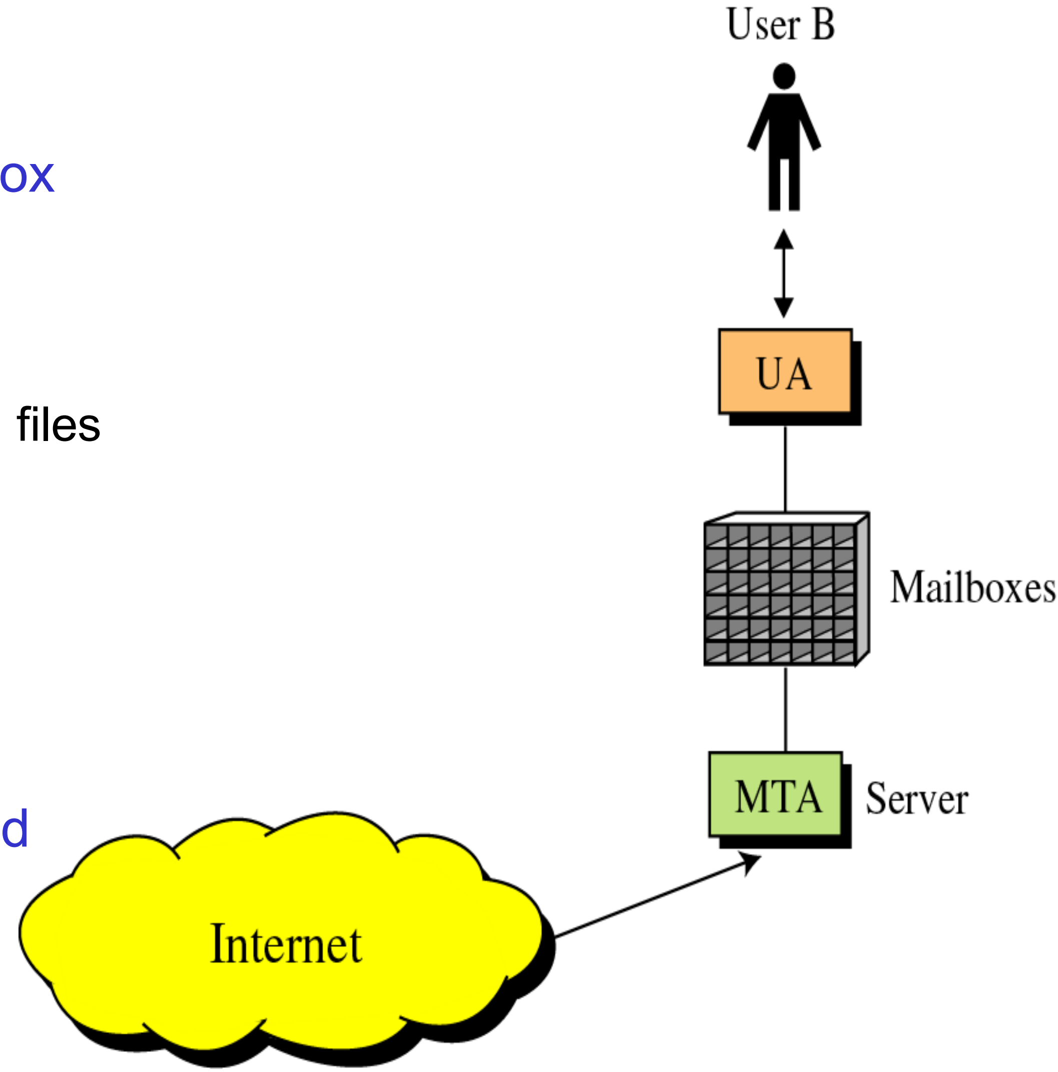
Delayed Delivery

- Sender-site Delay
  - Sender site stipulates a *spooling system*
  - UA creates message and forwards it to Spooling system to store
  - MTA checks spool periodically for new mail
  - Delay depends upon following conditions
    - IP address of the server is obtained through DNS
    - Receiver is ready or not
  - If the message can not be delivered, it is returned to the sender



Delayed Delivery

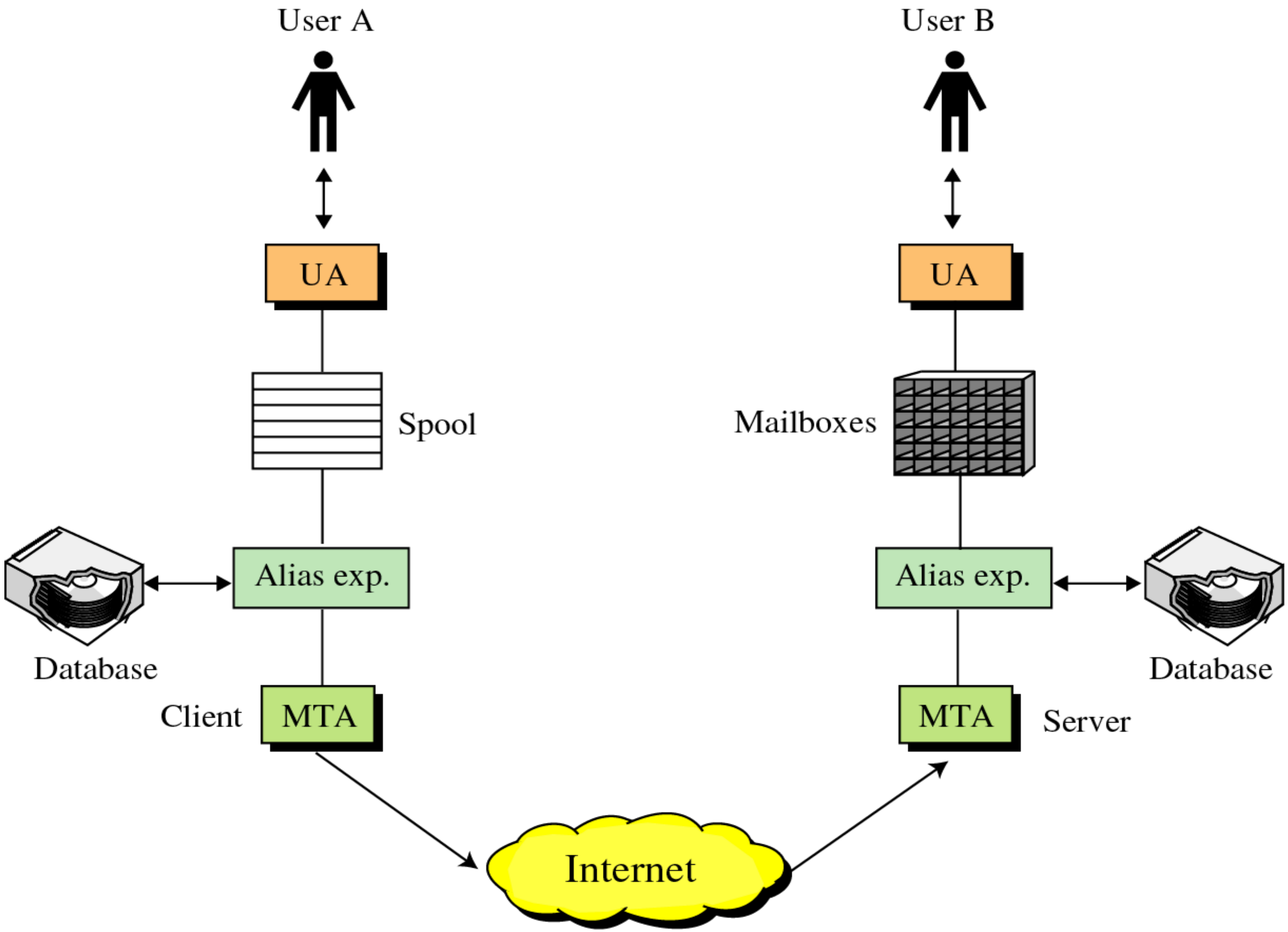
- Receiver-site Delay
  - After receiving mail, it is stored in respective mailbox for reading
  - Example
    - Sendmail uses individual files to store mails
- Intermediate Delay
  - Mails can be stored by intermediate MTAs to send them when appropriate



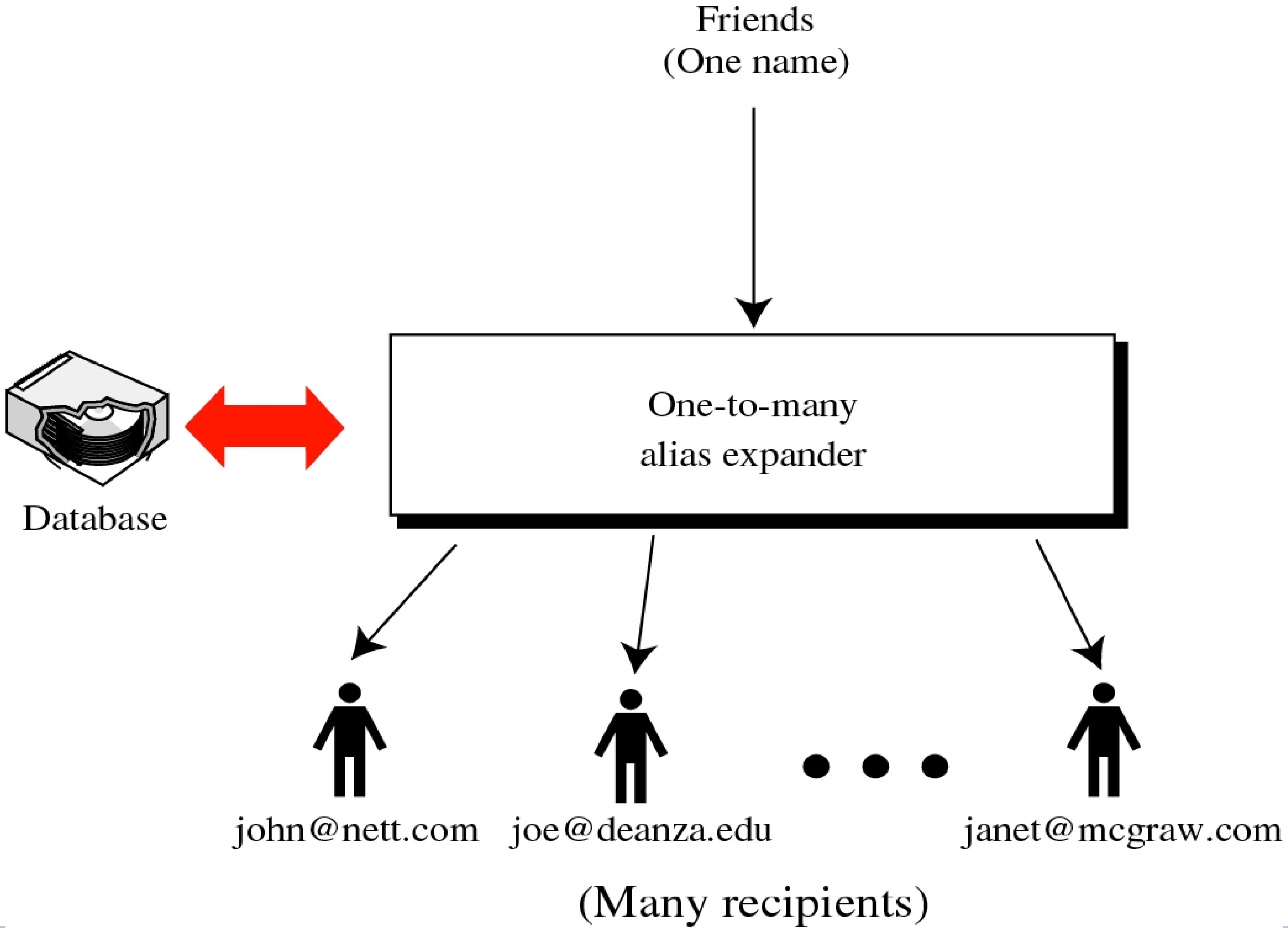
Aliases

- One-to-many Expansion
  - Allows one name, called alias to represent several different email addresses
  - A list of email addresses is associated with the alias using a database map
  - If an alias is defined, mail destined to that name is sent to every recipient's of the list
  - If not defined, mail is sent to the user only

Aliases


User A
User B
UA
UA
Spool
Mailboxes
Alias exp.
Alias exp.
Database
Database
Client
MTA
MTA
Server
Internet

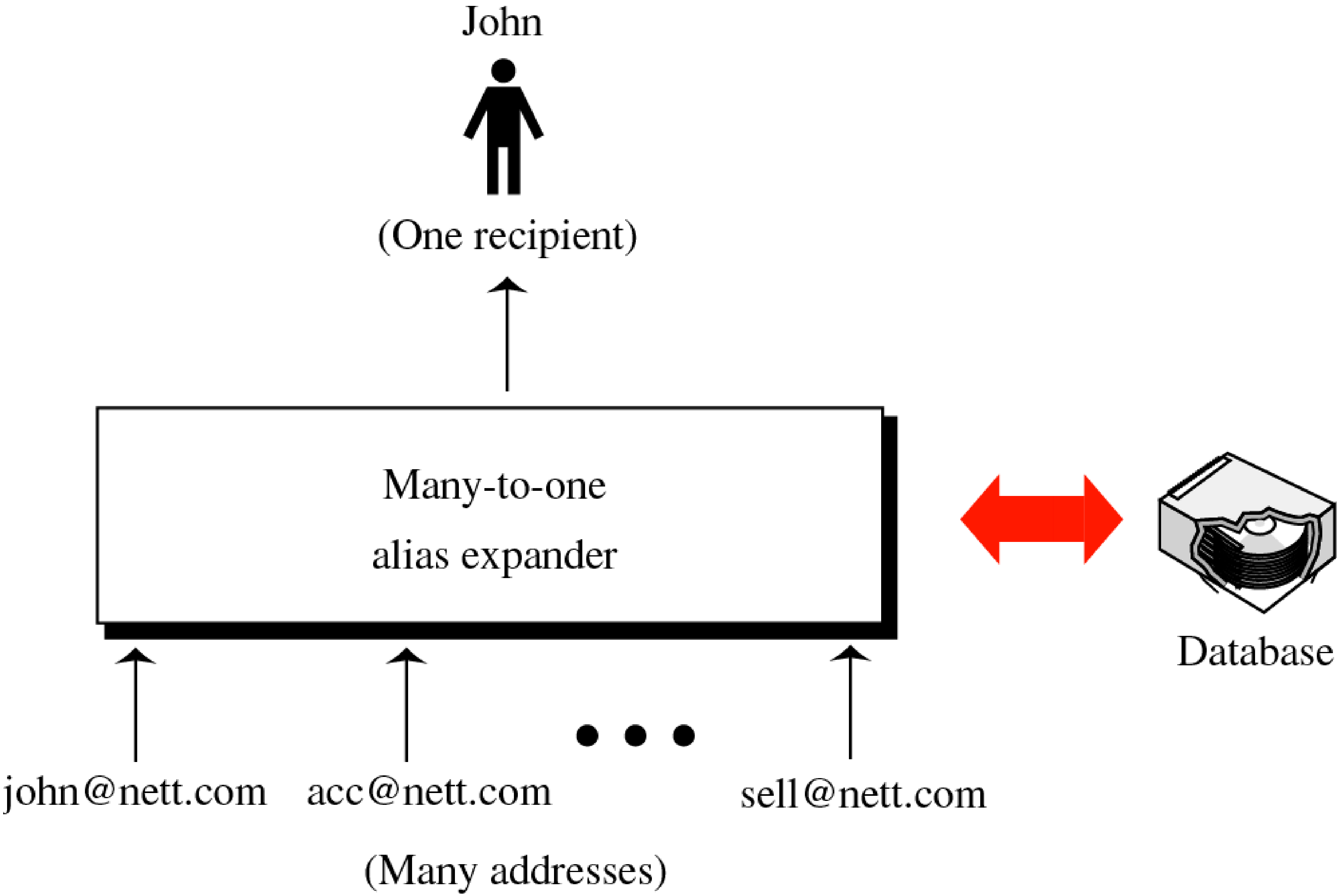
One-to-many expansion



Aliases

- Many-to-one Expansion
  - A user can have many different email addresses
  - An alias database is used for this map
  - Single mailbox is used
  - Mails destined to all theses email addresses are sent to single user

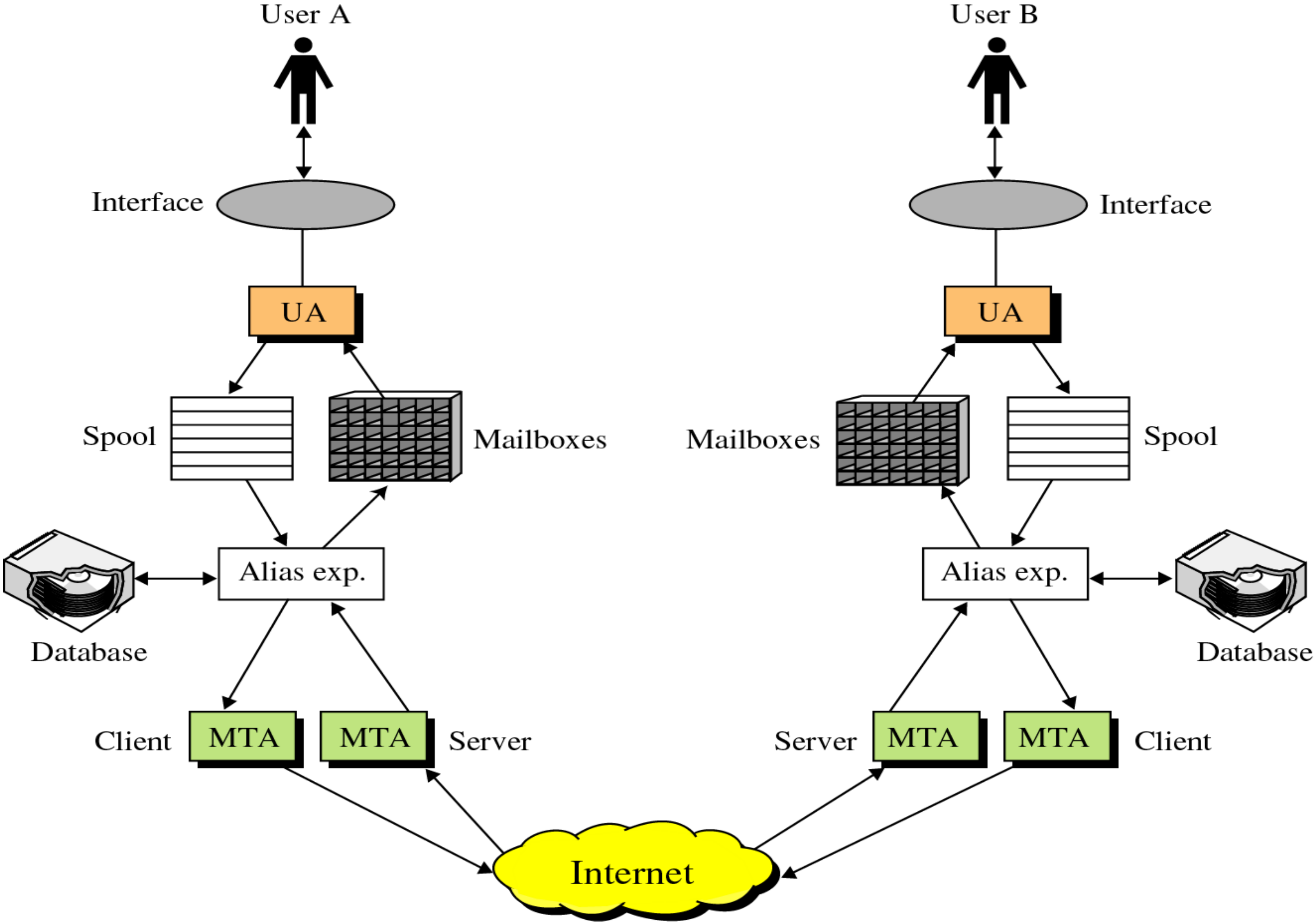
Many-to-one expansion



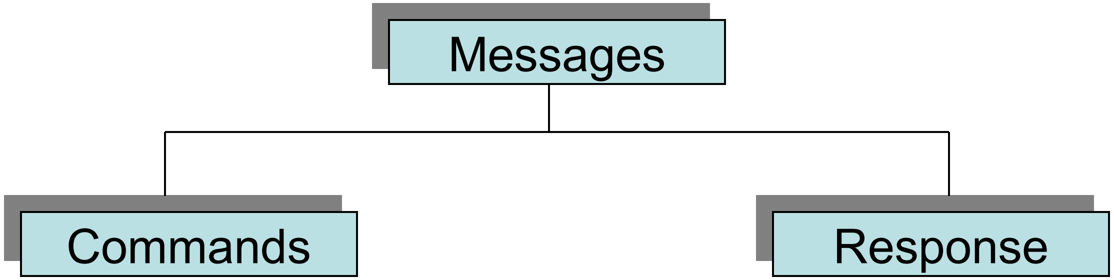
Mail Transfer Agent(MTA)

- Actual mail transfer is done through MTAs
  - Client MTA is required to send mail
  - Server MTA is required to receive mail
  - Example
    - Sendmail, squirlmail etc.

SMTP


User A
User B
Interface
Interface
UA
UA
Spool
Mailboxes
Mailboxes
Spool
Alias exp.
Alias exp.
Database
Database
Client
MTA
MTA
Server
Server
MTA
MTA
Client
Internet

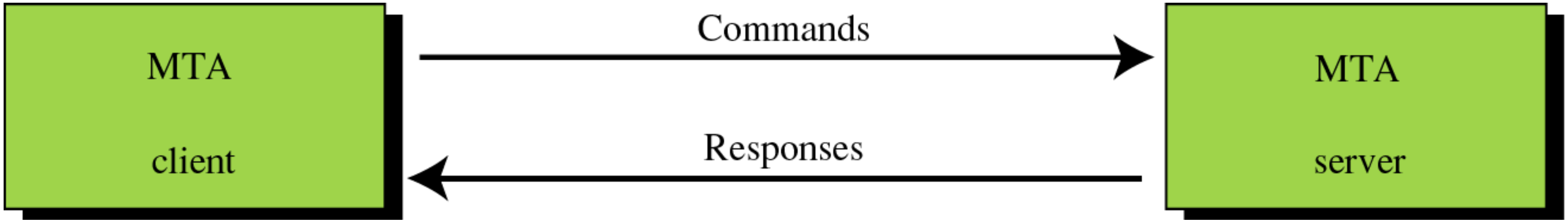
SMTP Messages



Messages

Commands

Response



MTA

client

Commands

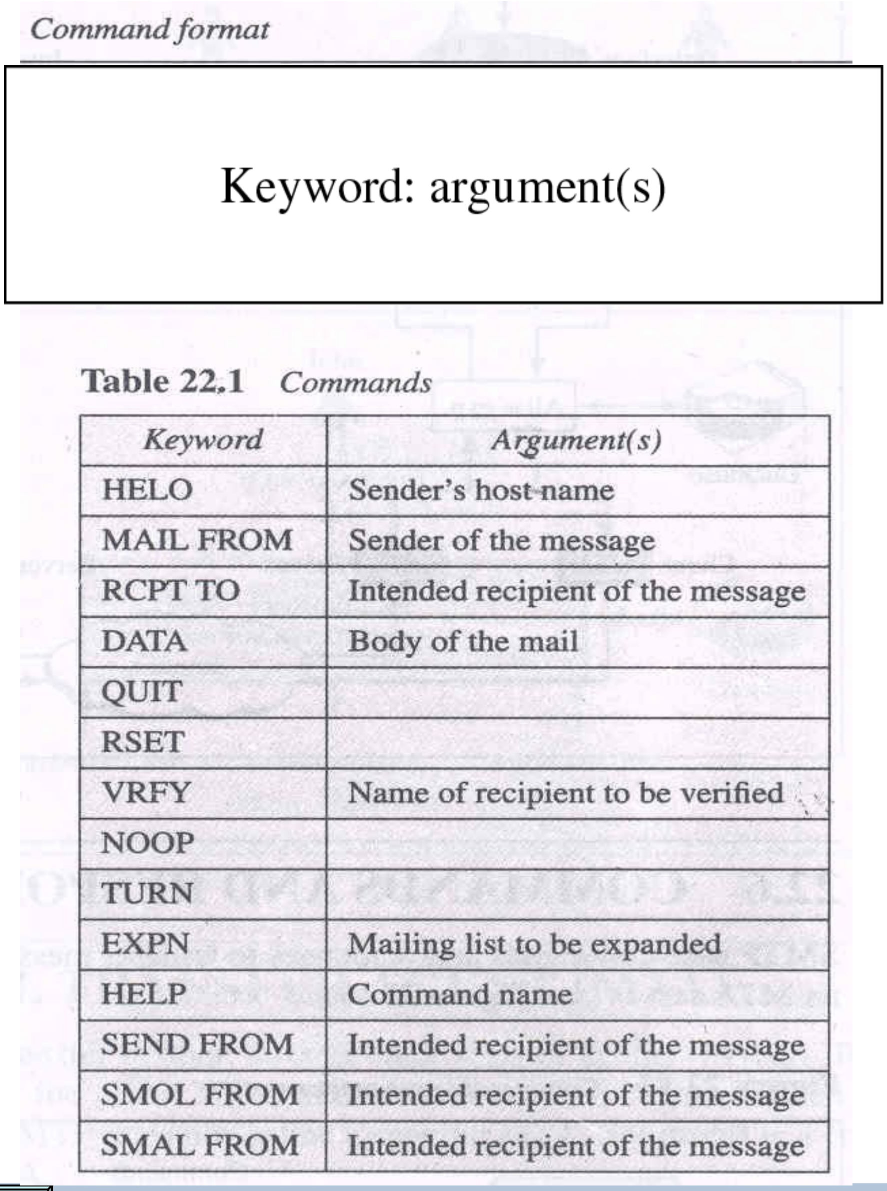
Responses

MTA

server

SMTP Messages(Commands)

- Commands
  - Commands are sent from client to server
  - First five are mandatory



Command format

Keyword: argument(s)

Table 22.1 *Commands*

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL FROM	Intended recipient of the message
SMAL FROM	Intended recipient of the message

SMTP Messages(Responses)

- Responses
  - Commands are sent from server to client
  - 3 digit code of the following form
    - 2yz(positive completion)
      - Requested command has been successfully completed and new commands can be started
    - 3yz(positive intermediate response)
      - Requested command has been accepted, but recipient needs more information for completion
    - 4yz(transient negative completion reply)
      - Command has been rejected, but error is temporary. The command can be sent again
    - 5yz(permanent negative completion reply)
      - Command has been rejected permanently. The command can not be sent again during this session

SMTP Messages(Responses)

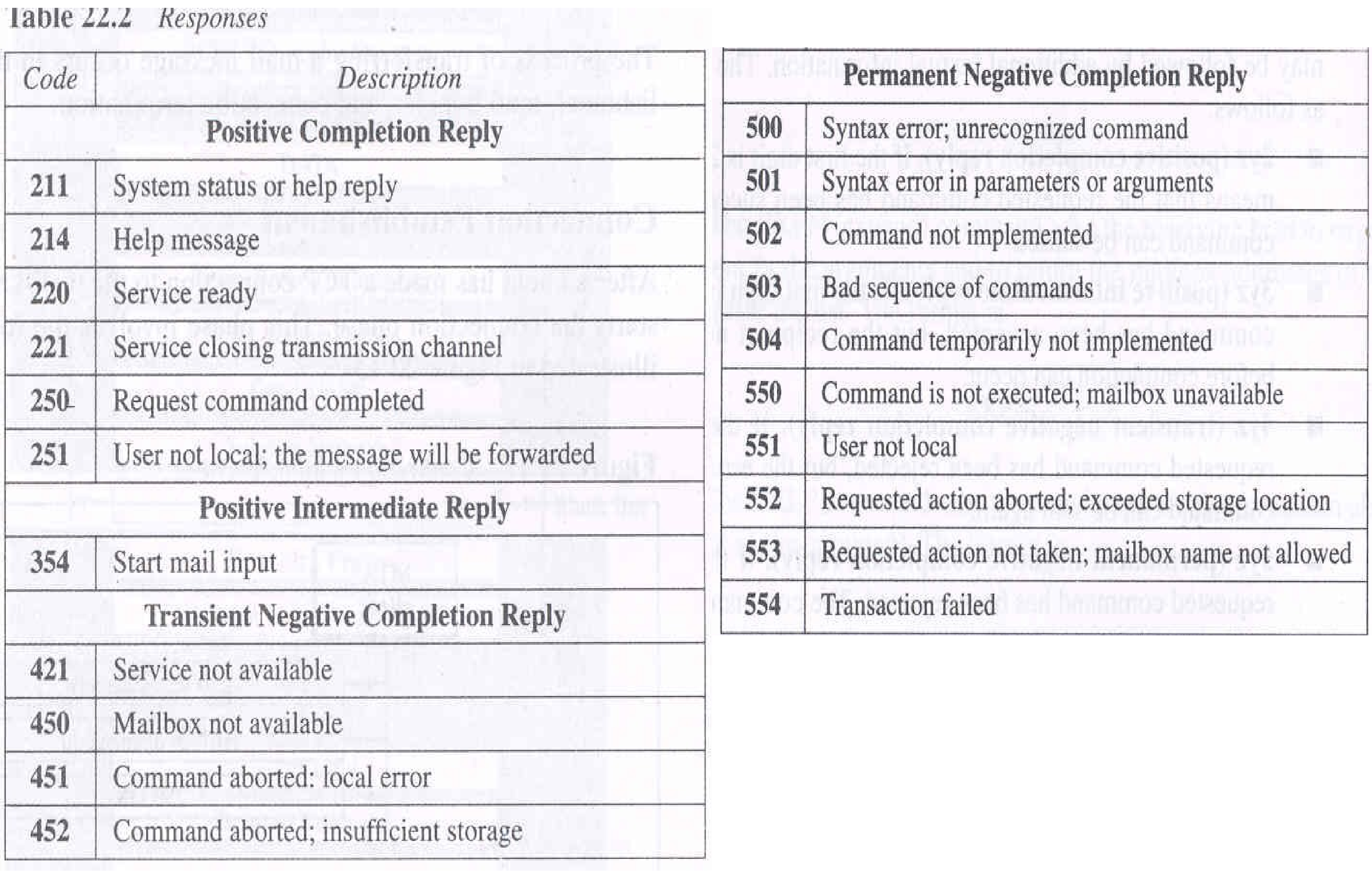
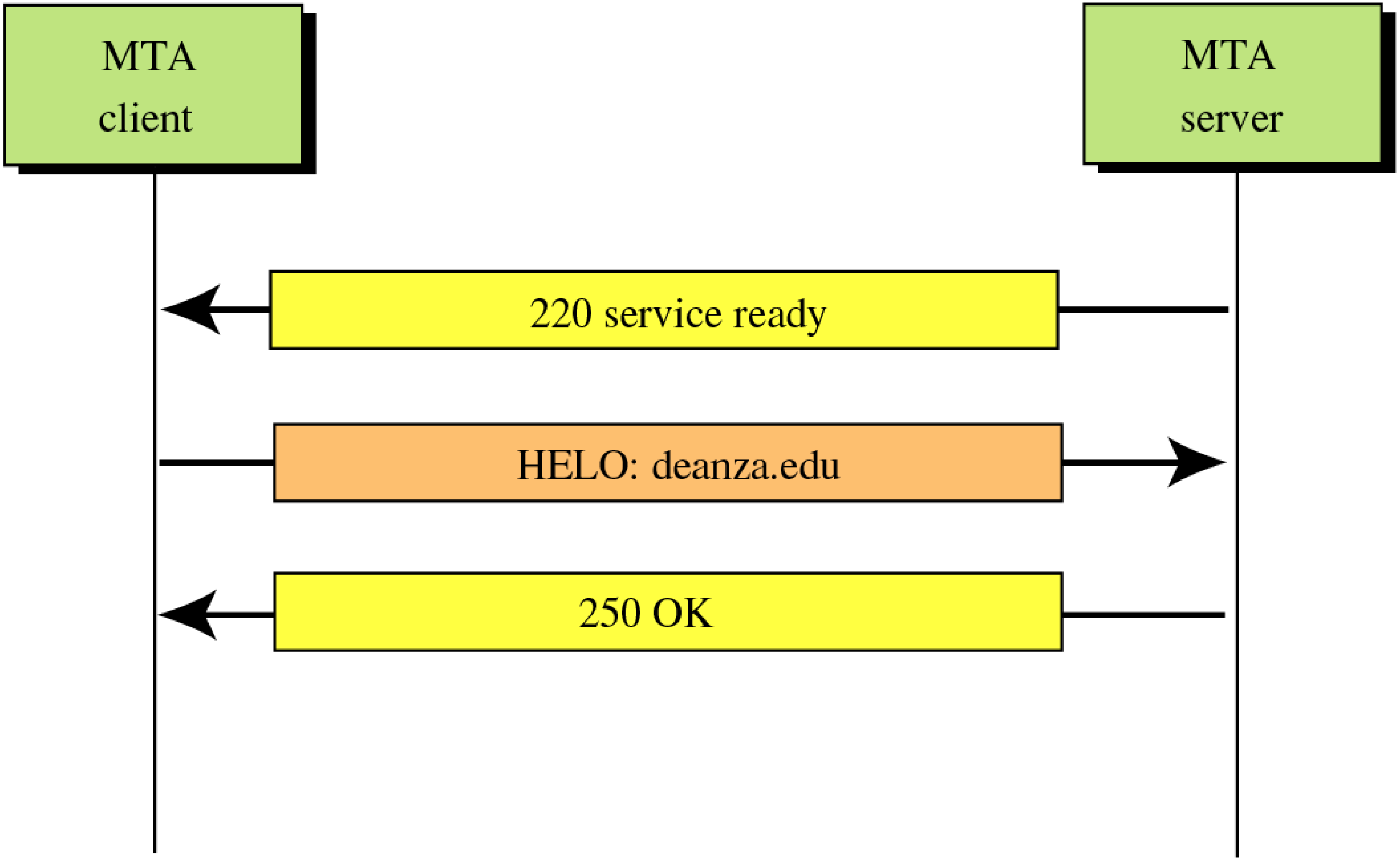


Table 22.2 *Responses*

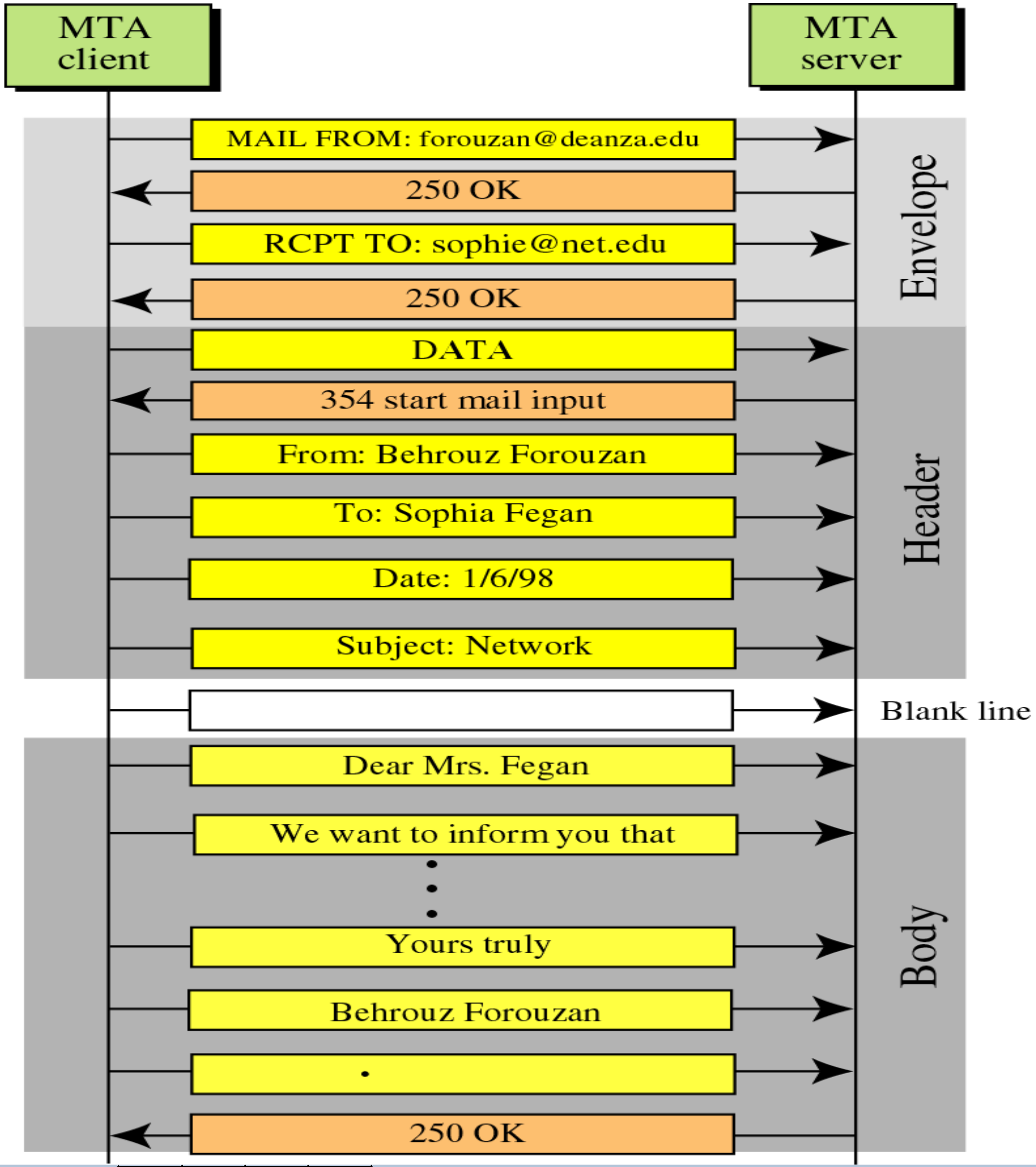
Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
	Positive Intermediate Reply
354	Start mail input
	Transient Negative Completion Reply
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
	Permanent Negative Completion Reply
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Connection establishment

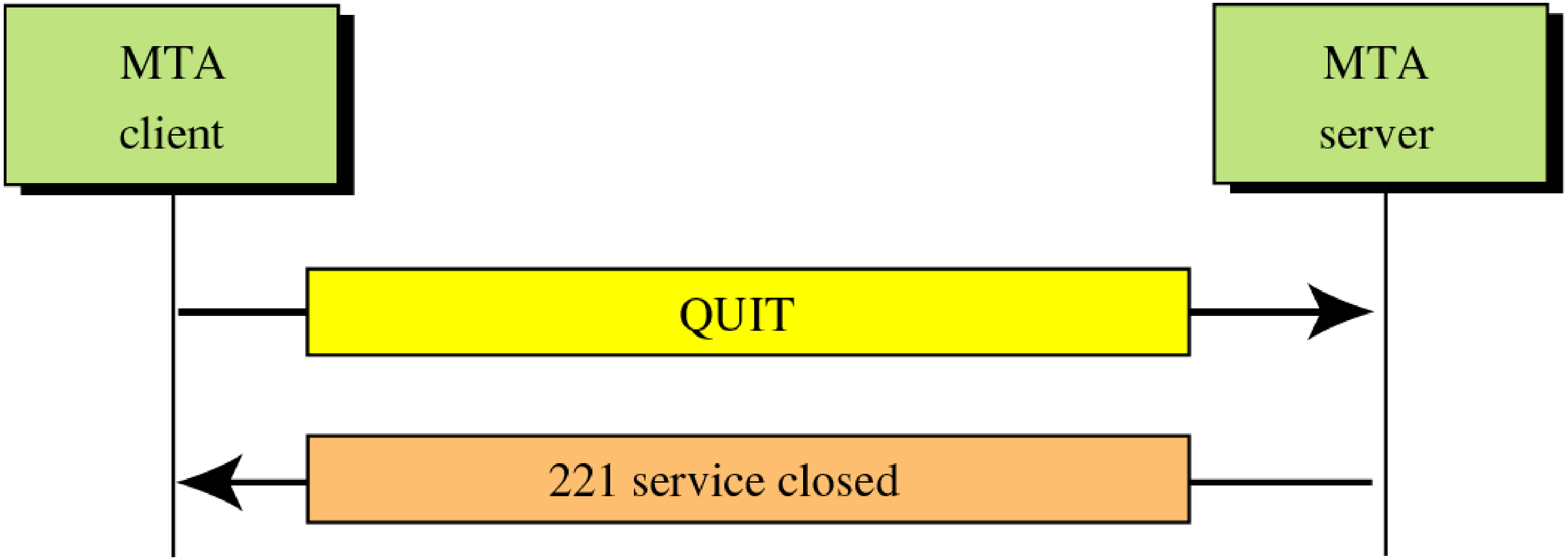


SMTP

- An Example



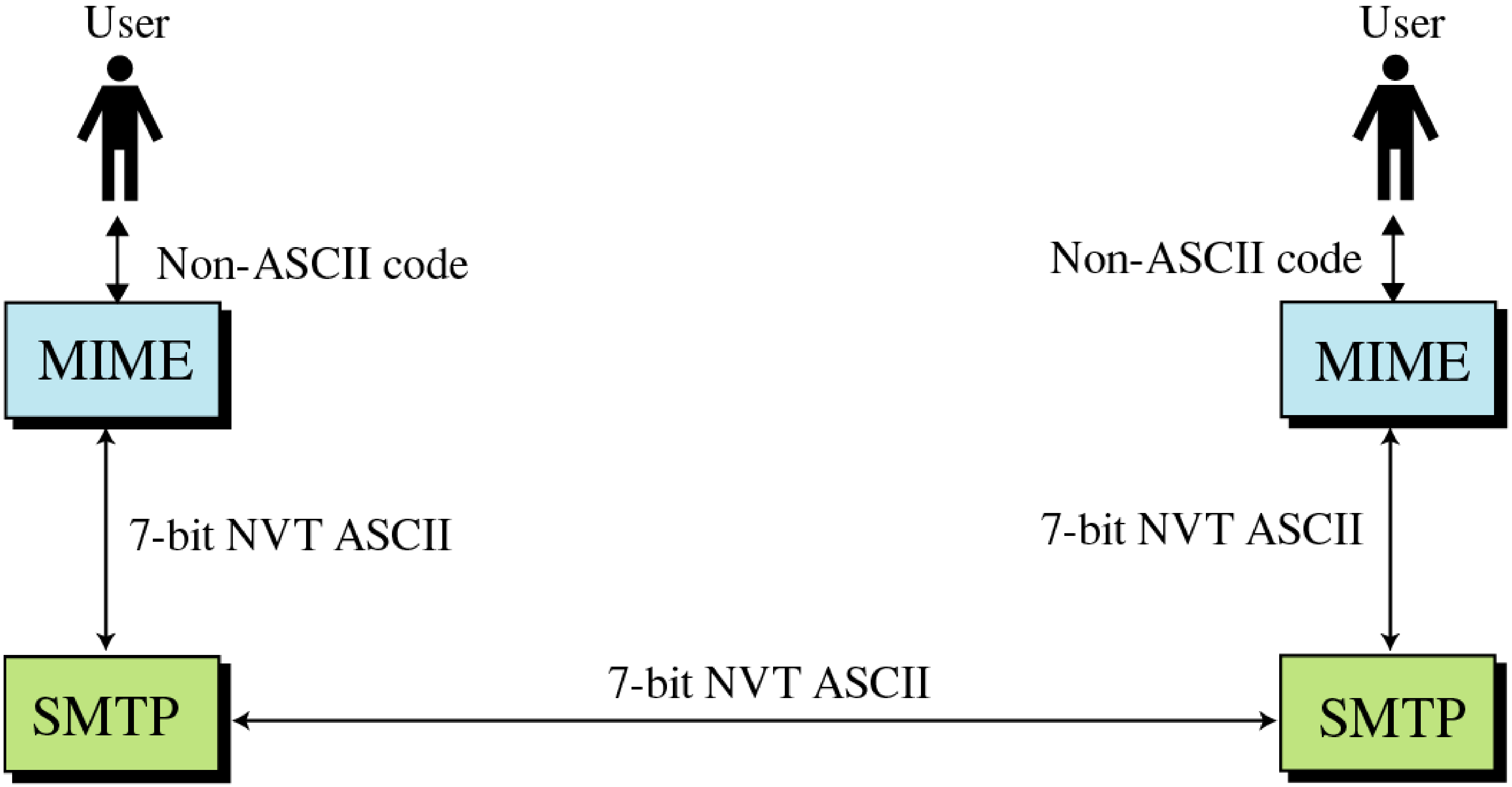
Connection Termination



MIME

- SMTP uses NVT 7-bit ASCII character set
  - Can not be used for languages that are not supported by 7-bit ASCII characters. E.g French, German, Hebrew, Russian, Chinese, Japanese etc.
  - Can not be used to send binary data or audio or video
- MIME(Multipurpose Internet Mail Extension)
  - A supplementary protocol that allows non-ASCII data to be sent SMTP
  - Can be thought of as software functions that transform non-ASCII to ASCII and vice versa

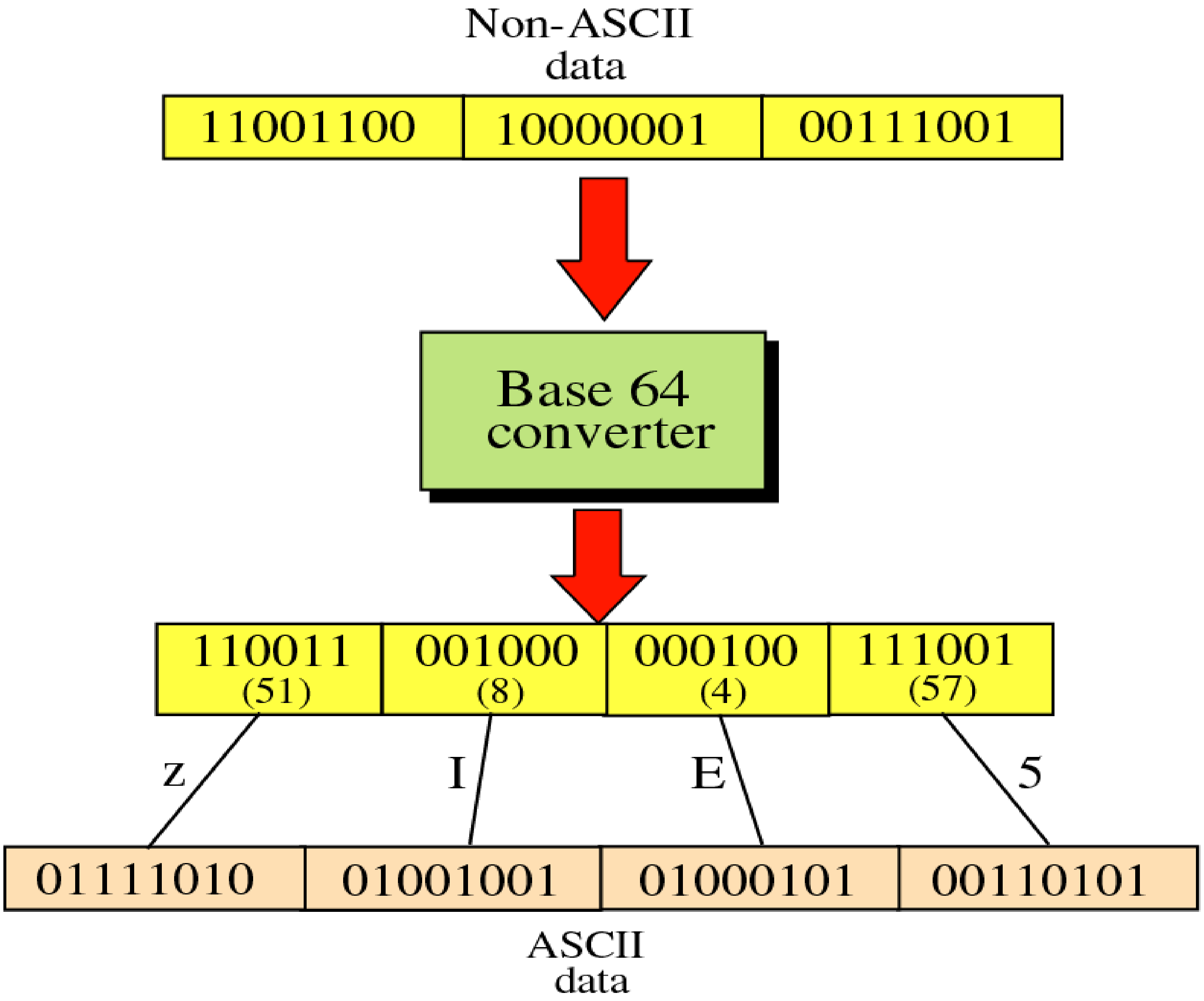
MIME


User
User
Non-ASCII code
Non-ASCII code
MIME
MIME
7-bit NVT ASCII
7-bit NVT ASCII
SMTP
7-bit NVT ASCII
SMTP

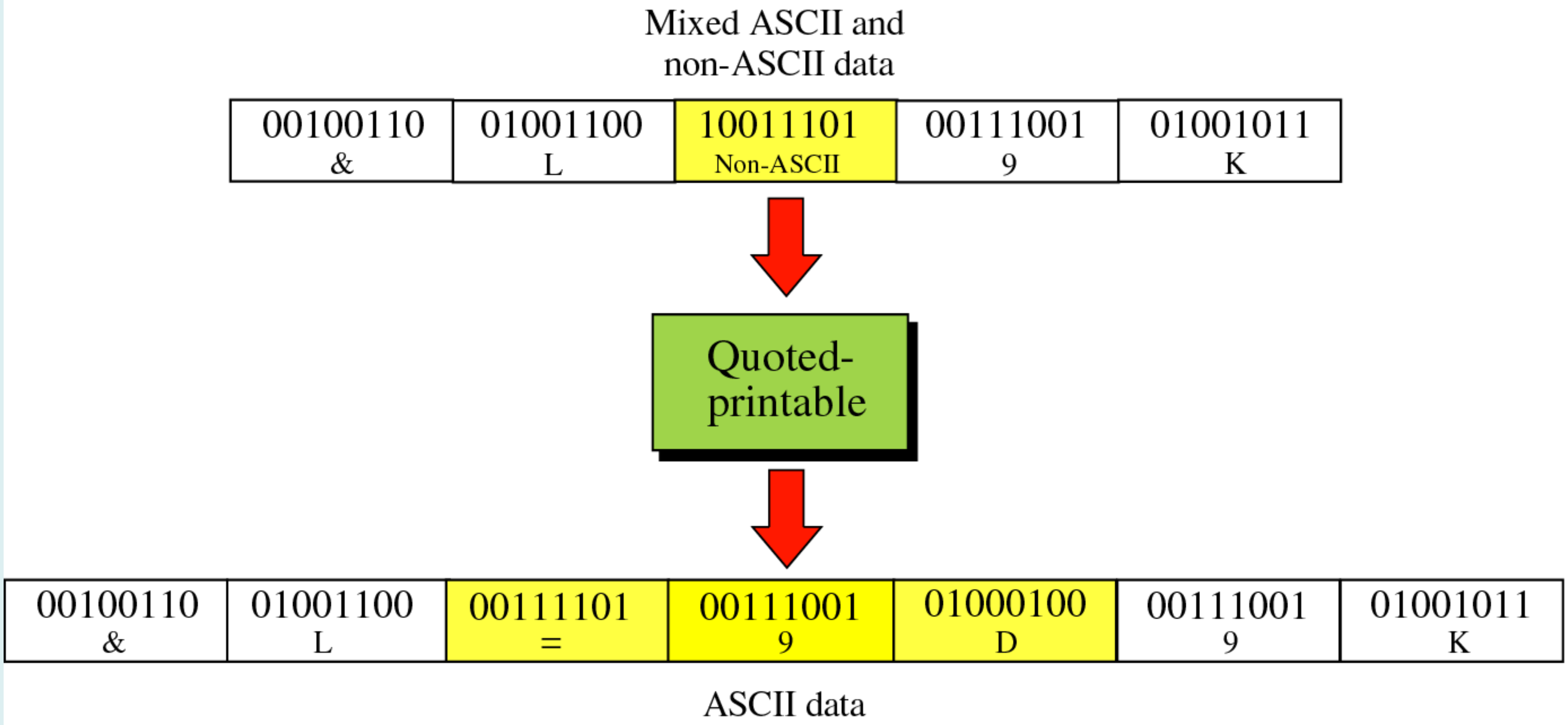
MIME

- Defines five additional headers
  - MIME-version
    - MIME-Version: 1.1
  - Content-Type
    - Type of the data used in the body
    - Content-Type: <type/subtype; parameters>
    - Subtype
      - Text, Message, Image, Video, Audio etc
  - Content-Transfer-Encoding
    - Encoding to be used
    - Content-Transfer-Encoding: <type>
    - Type
      - 7bit, 8bit, binary, Base64 etc.
  - Content-Id
  - Content-Description

Base64


Non-ASCII
data
11001100
10000001
00111001
Base 64
converter
110011
(51)
001000
(8)
000100
(4)
111001
(57)
z
I
E
5
01111010
01001001
01000101
00110101
ASCII
data

Quoted-printable



Mixed ASCII and non-ASCII data

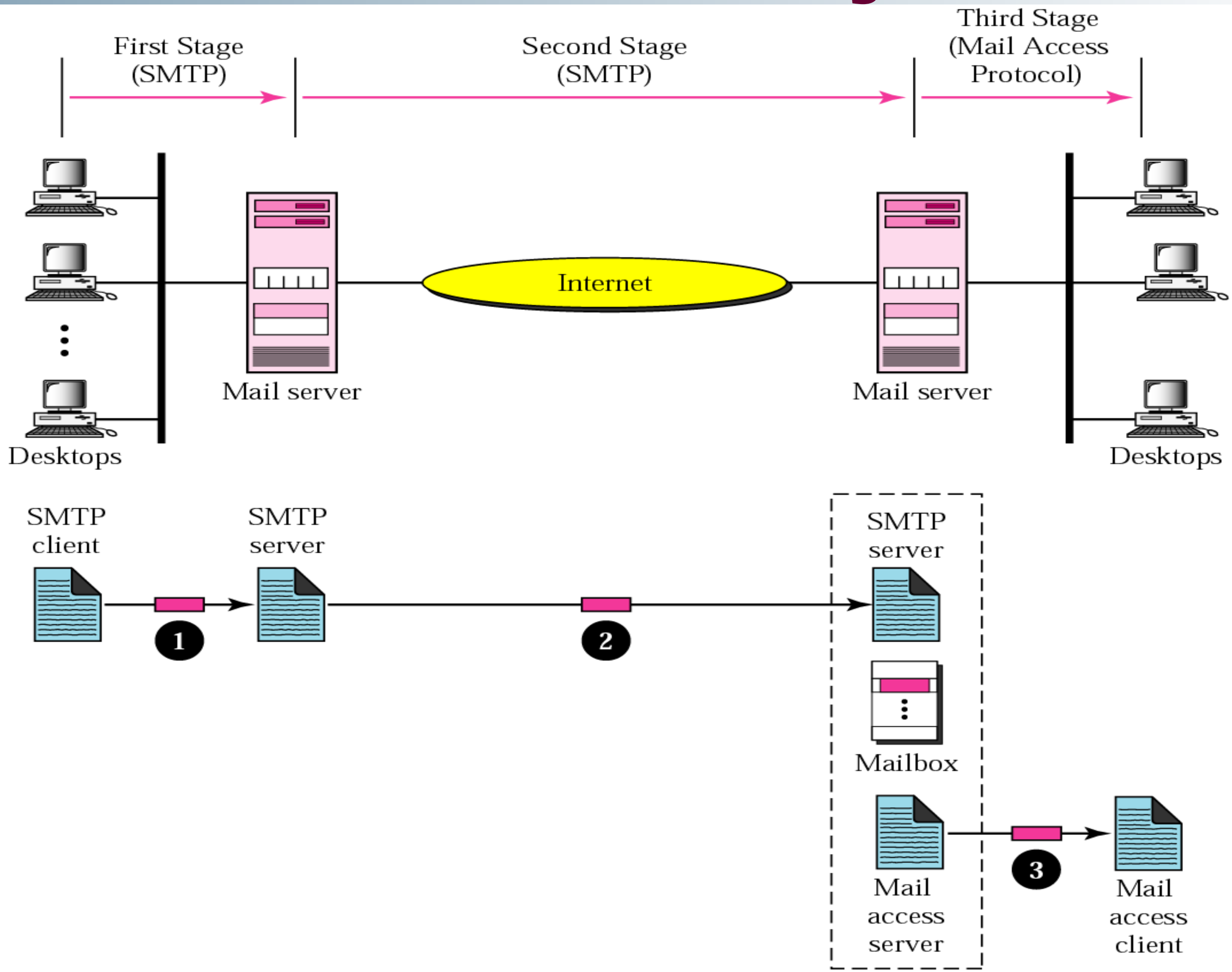
00100110	01001100	10011101	00111001	01001011
&	L	Non-ASCII	9	K

Quoted-printable

00100110	01001100	00111101	00111001	01000100	00111001	01001011
&	L	=	9	D	9	K

ASCII data

Mail Delivery


First Stage
(SMTP)
Second Stage
(SMTP)
Third Stage
(Mail Access
Protocol)
Desktops
Mail server
Internet
Mail server
Desktops
SMTP
client
SMTP
server
1
2
SMTP
server
Mailbox
Mail
access
server
3
Mail
access
client