CHAPTER SEVEN

Application Layer

CH-07:Application Layer

- The Application Layer:
  - receives services from the transport layer and provides services to users.
  - provides user interfaces and support services such as email, remote file access and transfer, access to the WWW.
  - Three general issues related to the application layer:the client-server paradigm, addressing, and services

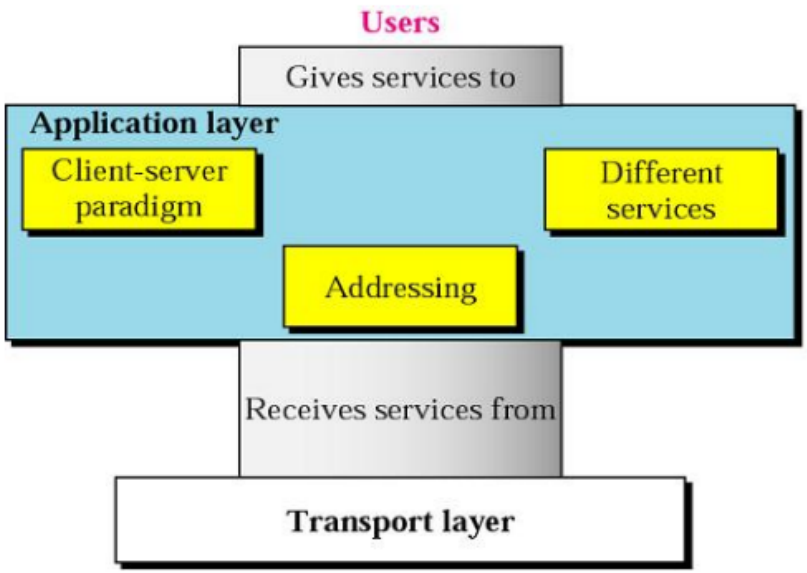
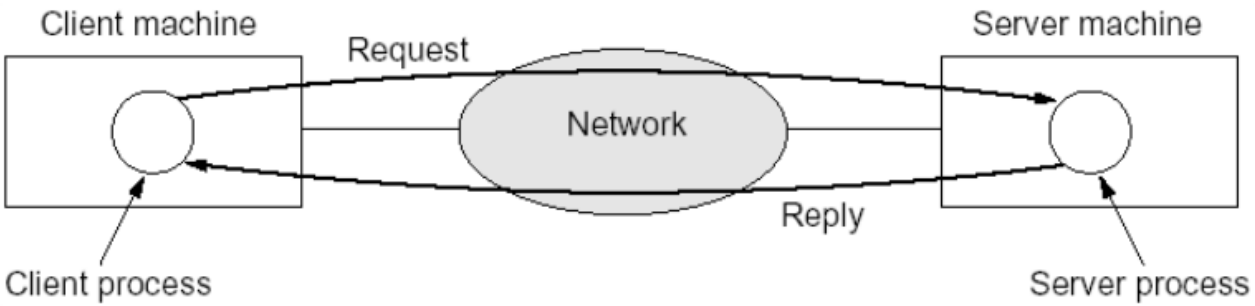


Figure: Application layer Services ...

CH-07:Application Layer

1. Client-Server Model

- To do a task, there must be a client and a server.
- A computer runs a program to either request a service from another computer (client) or to provide a service to another computer (server).
- Communication takes the form of the client process sending a message to the server process and then waiting for a reply a client program runs when needed, but the server program runs all the time
- Socket Interface: a set of system calls for communication between clients and servers



CH-07:Application Layer

2. Addressing

- Client and a server communicate with each other using addresses
- Addressing mechanism in the application layer is different from the ones in other layers

  email address: aastuavp@aastu.edu.et
  Web page (host name): http://www.aastu.edu.et
- these are aliases convenient for human beings; they must be mapped to IP addresses
- an application program needs the services of another program for this;

  ↓

  DNS - Domain Name System
- it uses port 53
- it is not directly used by the user; but by application programs to perform the mapping

CH-07:Application Layer

- Originally (when Internet was small), mapping was done using a host file with two columns stored on every host impossible nowadays
- The host file is too large
- Updating all the host files in the world every time there is a change is difficult
- There are different Protocol
  - Electronic mail (SMTP, MIME, POP3, IMAP, Webmail)
  - File transfer (FTP - File Transfer Protocol)
  - HTTP - Hypertext Transfer Protocol for accessing data on the WWW
  - Multimedia

CH-07:Application Layer

Web Protocol

- The Web began in 1989 at CERN
- The Web is an architectural framework for accessing linked content spread out over millions of machines all over the Internet.
- Architectural Overview $\mapsto$ From User Perspective
  - The Web consists of a vast, worldwide collection of content in the form of **Web pages**
  - The idea of having one page point to another, $\mapsto$ Hypertext
  - Pages are generally viewed with a program called a browsr

    $\downarrow$

    Displaying a Web page on the client machine.
  - How do we access a page ?

    $\downarrow$

    Each page is fetched by sending a request to one or more servers, which respond with the contents of the page.

    $\Downarrow$

    Request-response protocol

CH-07:Application Layer

- HTTP - Hypertext Transfer Protocol
  - is a protocol for fetching resources such as HTML documents.
  - It is the foundation of any data exchange on the Web and it is a client-server protocol,
  - A complete document is reconstructed from the different sub-documents fetched

    ↓

    For instance, text, layout description, images, videos, scripts, and more.
  - uses a port number 80;
  - Sending ⇆ receiving; in the form of message
  - two kinds of messages: request and response

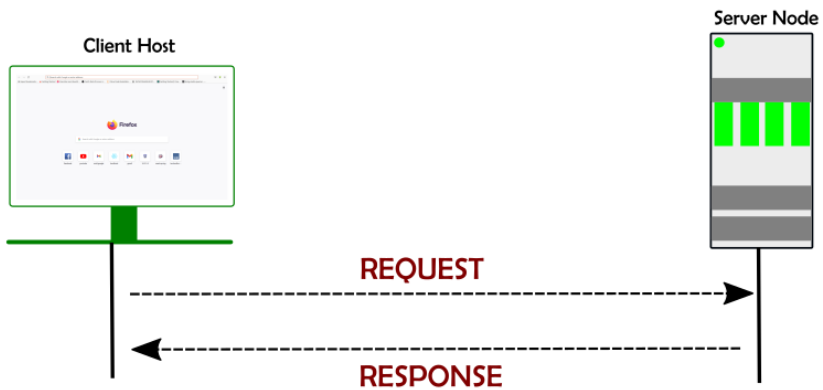


Figure: Http message

CH-07:Application Layer

1: Request Message

- consists of a request line, headers, and body
- The request line defines the request type, resource (URL), and HTTP version → currently 3.0
- A URL: Uniform Resource Locator
  - is a unique identifier used to locate a resource on the Internet.
  - It is also referred to as a web address.
  - URLs consist of four parts.
    - HTTP Method: the protocol used to retrieve a document
    - Host address: the computer where the information is located
    - Port : the port number of the server
    - Path: of the file where the information is located

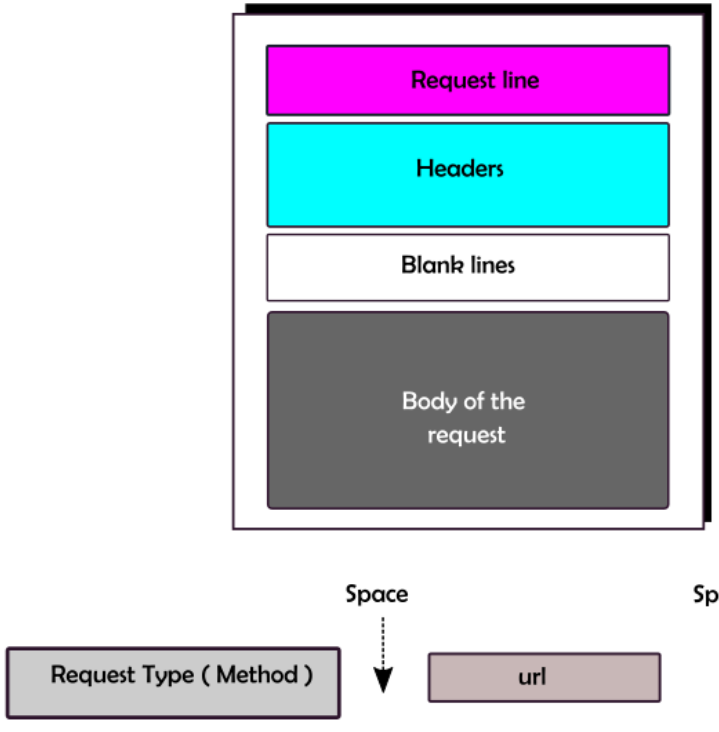


Figure: HTTP request parts

CH-07:Application Layer

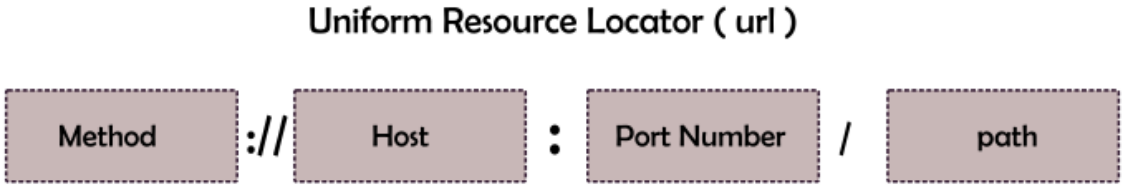


Figure: url format

Request type / Methods

- defines several kinds of messages; commands or requests a client issues to the server
  - GET: to retrieve a document.
  - HEAD: when a client wants some information about the document (its header) but not the document itself
  - POST: to provide some input to the server - appending a new data such as posting a message to a newsgroup or adding a file to a bulletin board
  - PUT: to provide a new or replacement document to the server - authentication required
  - PATCH: similar to PUT but the request contains a list of differences that should be implemented in the existing file
  - MOVE: to move a file to another location
  - DELETE: removes a page from a server - authentication required

CH-07:Application Layer

- LINK: create a link or links from a document to another location
- UNLINK: delete a link

2: Response Message

- consists of a STATUS-LINE, HEADER, and BODY
  - STATUS-LINE: consists of the protocol version followed by a numeric status code and its associated textual phrase.
    - HTTP version: same as above
    - Status code : a three digit number specifying the status of a command; the first digit is used to divide the responses into 5 major groups
    - Status phrase: status code in text form

CH-07:Application Layer

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

Figure: HTTP status code

- HEADER
  - Headers (for both types of messages) to exchange additional information
  - Each header line consists of a header name, and a header value

Figure: HTTP Header values

CH-07:Application Layer

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
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
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
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

Figure: Header properties

CH-07:Application Layer

- 4 different Categories of Headers
  1. General Header: gives general information about the message
  2. Request Header: specifies the client's configuration and preferred document format
  3. Response Header: specifies the server's configuration and special information about the request
  4. Entity Header: information about the body of the document

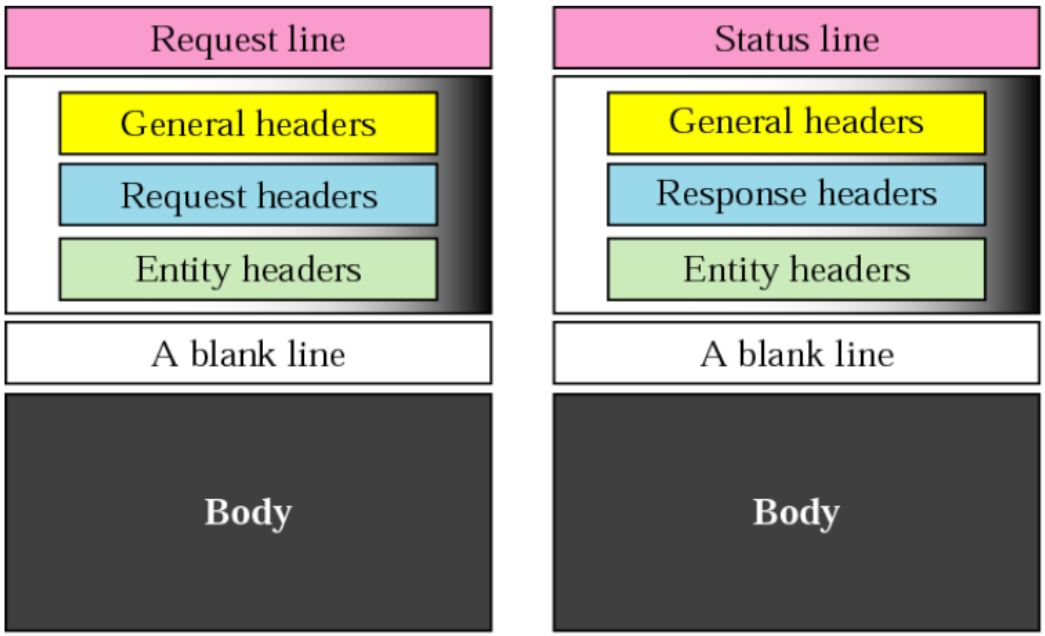


Figure: Types of Header

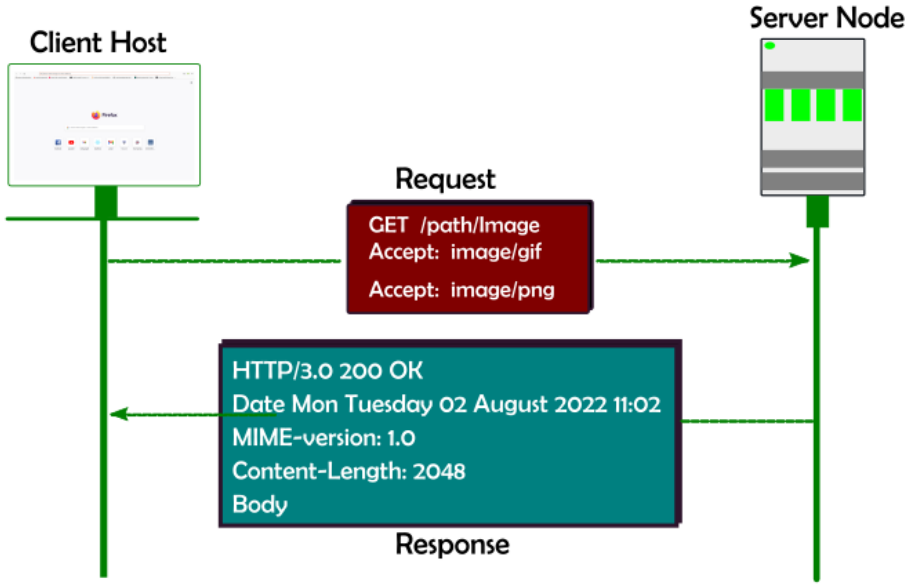
CH-07:Application Layer

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
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
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
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

Figure: Header properties

CH-07:Application Layer

- For Instance :



- Request message
  - To retrieve an image using the GET command with the path path/image.gif
  - Request-line shows the method, the URL, and the HTTP version
  - Two header lines:
    ↓
    The client specifying that it can accept images in GIF and JPEG formats has no body

- Response message: Contain
  - The status line and four lines of header
  - The header lines define the date, server, MIME version, and length of the document
  - The body of the document follows the header

CH-07:Application Layer

- PROXY SERVER
  - HTTP supportsproxy servers
  - A computer that keeps copies of responses to recent requests
  - The HTTP client → sends a request to the proxy server

    ↓

    The proxy server checks its cache;

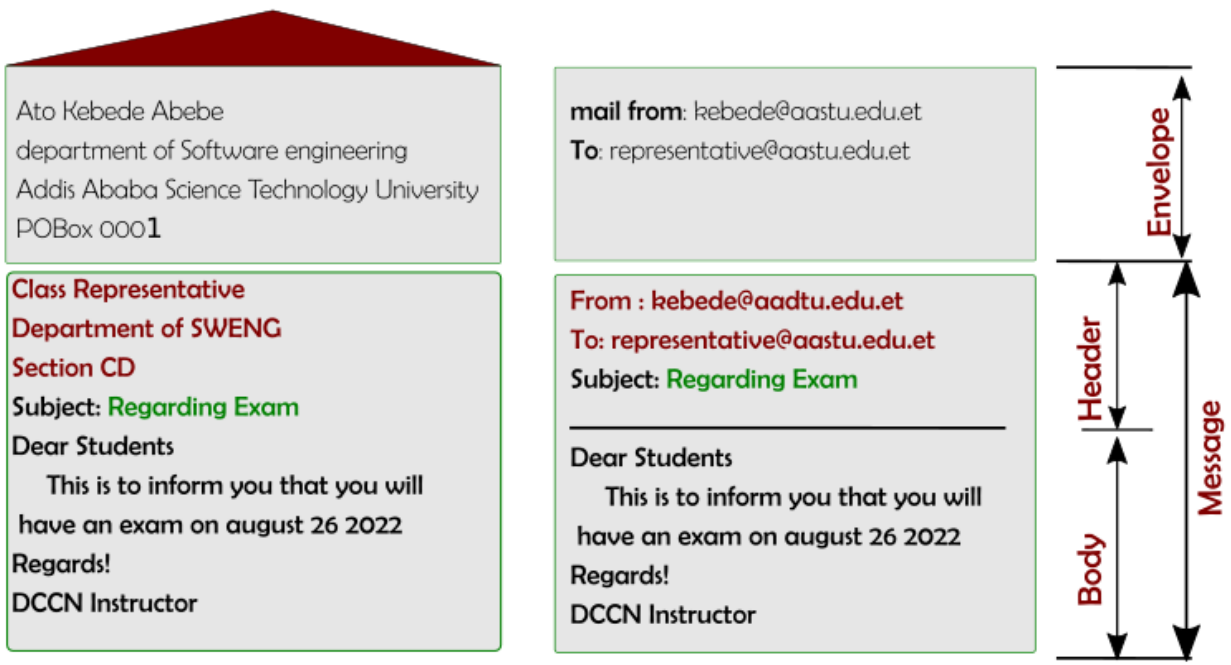
    ↓

    If the response is not stored in the cache, it sends the request to the corresponding server.
  - The proxy server reduces the load on the original server, decreases traffic, and improves latency
- WWW
  - a repository of information spread all over the world and linked together;
  - WWW is a client-server service distributed over many locations called websites
  - Hypertext and Hypermedia
    - Heypertext: information is stored in a set of documents that are linked using the concept of pointers; such documents contain only text
    - WHypermedia: documents can contain pictures, graphics, and sound

CH-07:Application Layer

Electronic Mail Service ...

- What is Email ?
  - A message that includes text, voice, video, or graphics and sent to one or more recipients an email has an envelope and a message
  - Envelope: information used for routing or transporting the email
  - Message : part consists of a header (information for the system) and a body (the actual information for the human recipient);



CH-07:Application Layer

↦ how email systems are organized and what they can do ?

- Two different subsystem defines the Email-system ↦ Architecture of Email
  1. User Agent
  2. Message transfer agents,

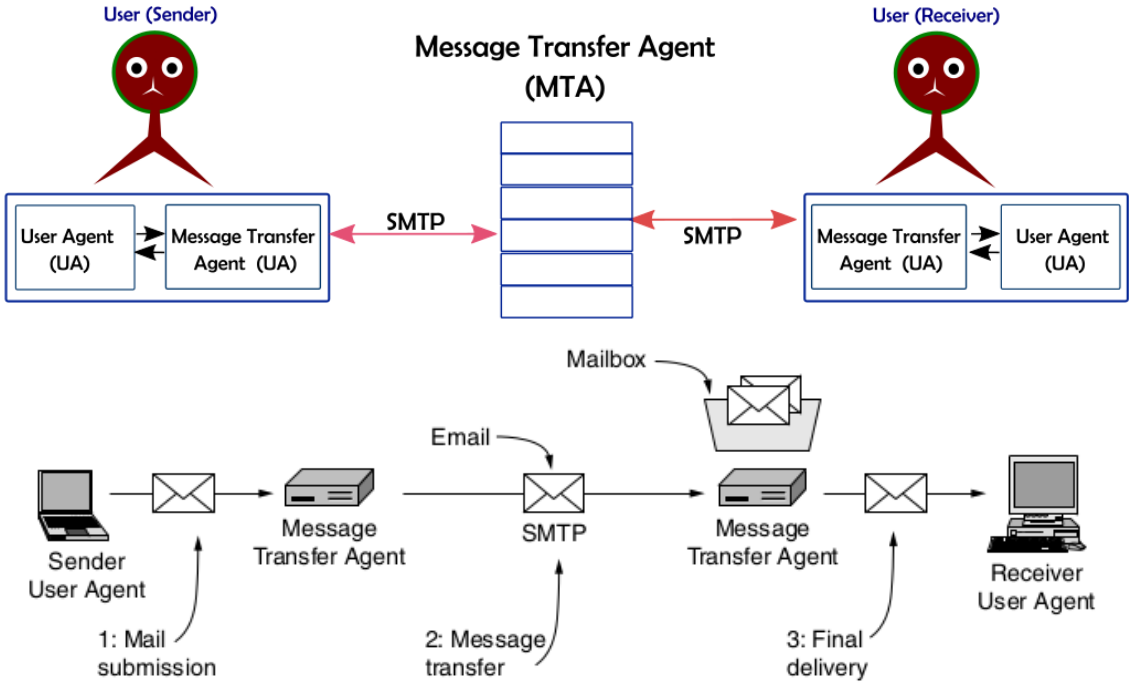


Figure: Architecture of the email system.

CH-07:Application Layer

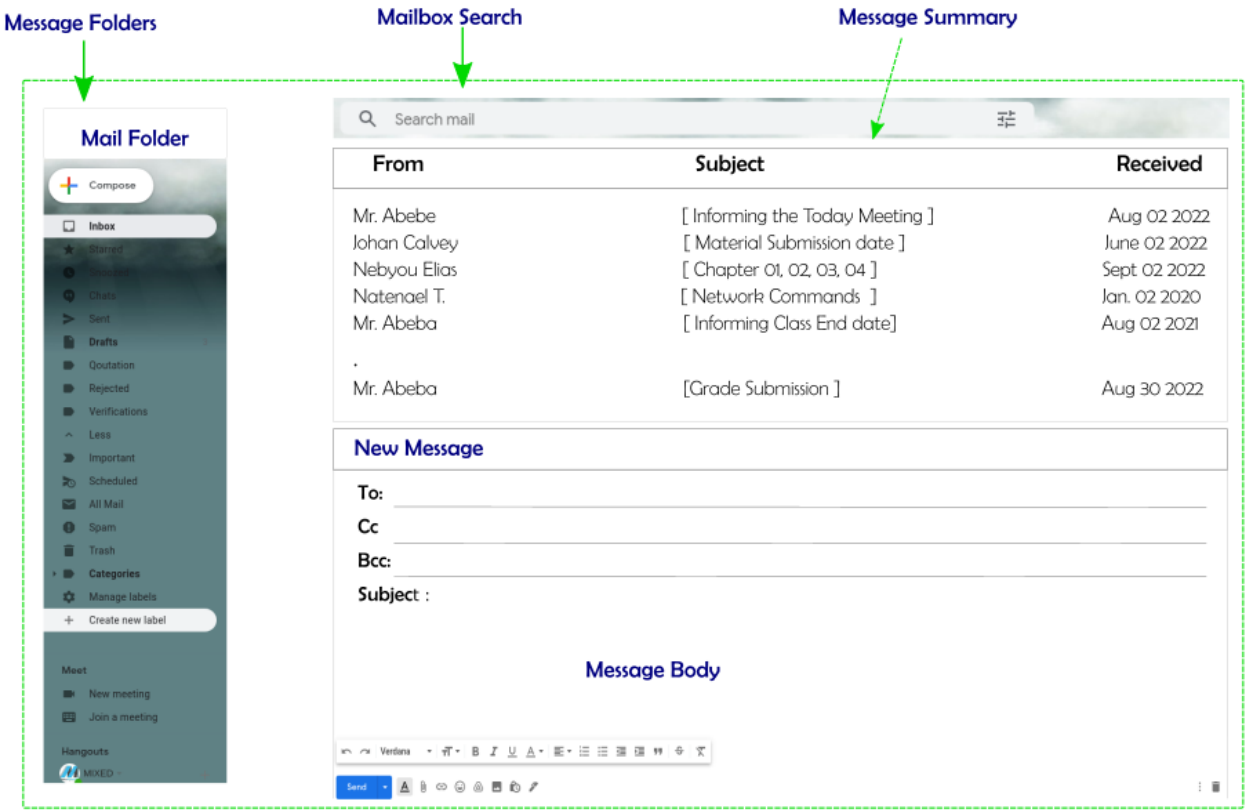
User Agent

- a user agent is a software that
  - Composes messages :
    ↓
    - By providing a template on the screen
    - some have built-in editors with more functionalities - spell and grammar checking, etc.
  - Reads incoming messages
    ↓
    - Checks incoming mails in the mailbox,
    - shows an on-line summary of each received mail - sender, size, date, flag fields, ...)
  - Replies to messages
    ↓
    - to the original sender or to all recipients, and
    - forwards messages (to a third party) handling mailboxes (inbox and outbox)
- a user agent can be command-driven or GUI-based

CH-07:Application Layer

- Message Submission → The act of sending new messages into the mail system for delivery.
- User agent processes may be done automatically, anticipating what the user wants

  ↦ For.Ex: incoming mail may be filtered to extract or deprioritize messages that are likely spam.



CH-07:Application Layer

Message Transfer Agent

- convey messages from the sender to the receiver.
- it is a daemon running in the background.
- to send mail, a system must have a client MTA; and to receive mail, a system must have a server MTA
- in the Internet, message transfer is done through a protocol (and software) named SMTP (Simple Mail Transfer Protocol);
- SMTP uses port 25
- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.
- Message transfer agents also implement **mailing lists**
- Moreover, has an advanced features
  - **Cc**: Carbon copies
  - **BCc**: Blind Carbon copies
  - **High-priority email**
  - **Security** $\mapsto$ Encryption

CH-07:Application Layer

- Originally Email support 7 bit ASCII text $\mapsto$ Problem

  $\downarrow$

  Language that are not supported by 7-bit ASCII characters (French, German, Hebrew, ...) can not be used to send binary files, video or audio
- MIME - Multipurpose Internet Mail Extensions
  - is a supplemental protocol that allows non-ASCII data to be sent using the existing mail programs and protocols;
  - only the sending and receiving programs need to be changed.
  - it transforms non-ASCII data at the sender site to ASCII and back to non-ASCII at the receiving site

    $\downarrow$

    non-ASCII data $\leftrightarrows$ ASCII

CH-07:Application Layer

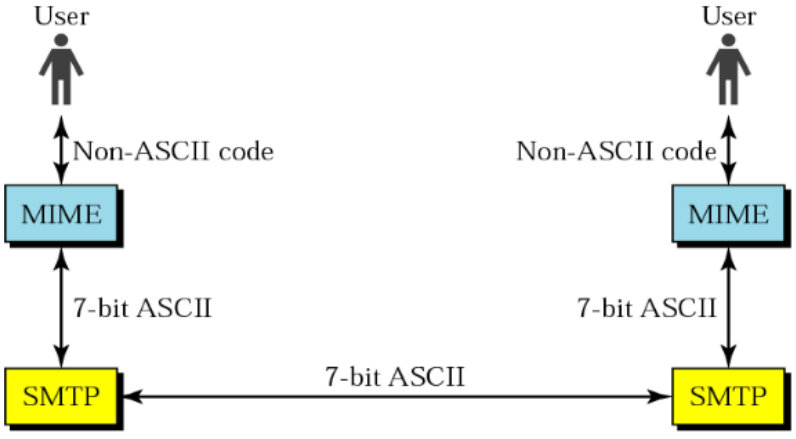


Figure: MIME ASII character

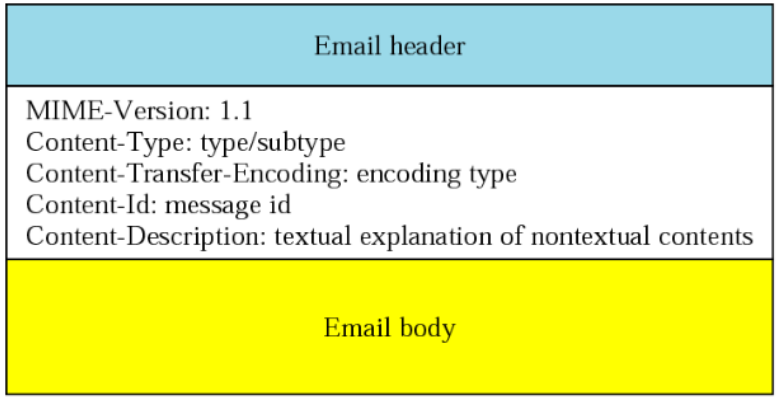
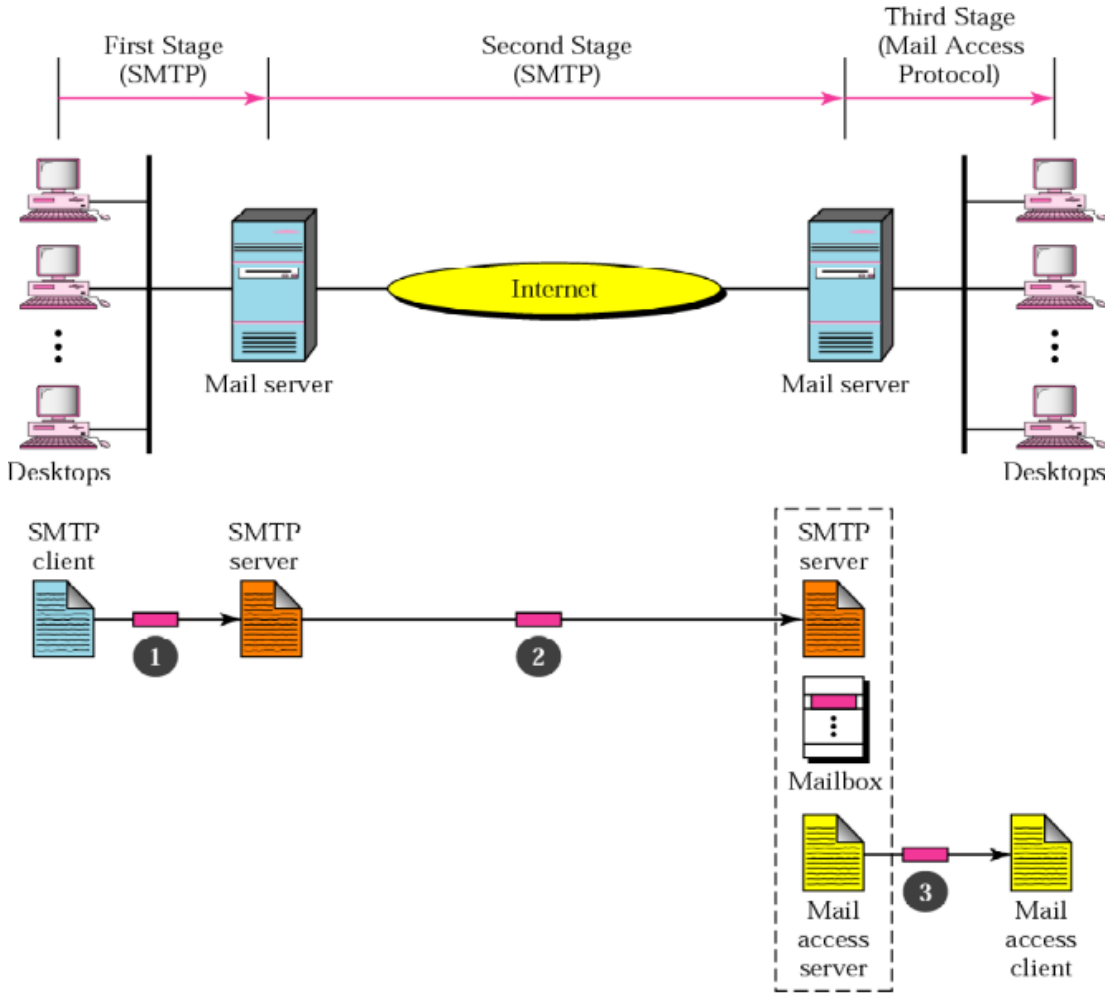


Figure: MIME Header

- MIME defines five headers $\mapsto$ added to the original SMTP header to define the transformation parameters
  - MIME-Version: the current version is 1.1
  - Content-Id: uniquely identifies the whole message in a multiple-message environment.
  - Content-Description: defines whether the body is image, audio, or video
  - Content-Type: the type of data used in the body of the message

CH-07:Application Layer

- Message Delivery : a way of delivering the message
- has three sequential steps :



CH-07:Application Layer

- Mail Access Protocols
  - The first and second stages of mail delivery use SMTP
  - SMTP is not involved in the third stage since SMTP is a push protocol;
  - it pushes the message from the sender to the receiver the operation of SMTP starts with the sender, not the receiver
  - The third stage needs a pull protocol; the operation must start with the recipient;
  - There are three common protocols used to deliver email over the Internet
    - The Simple Mail Transfer Protocol (SMTP)
    - The Post Office Protocol (POP) and
    - The Internet Message Access Protocol (IMAP)

CH-07:Application Layer

Simple Main Transfer Protocol(SMTP)

- It is responsible for sending email messages.
- This protocol is used by email clients and mail servers to exchange emails between computers.
- A mail client and the SMTP server communicate with each other over a connection established through a particular email port.
- Both entities are using SMTP commands and replies to process your outgoing emails.

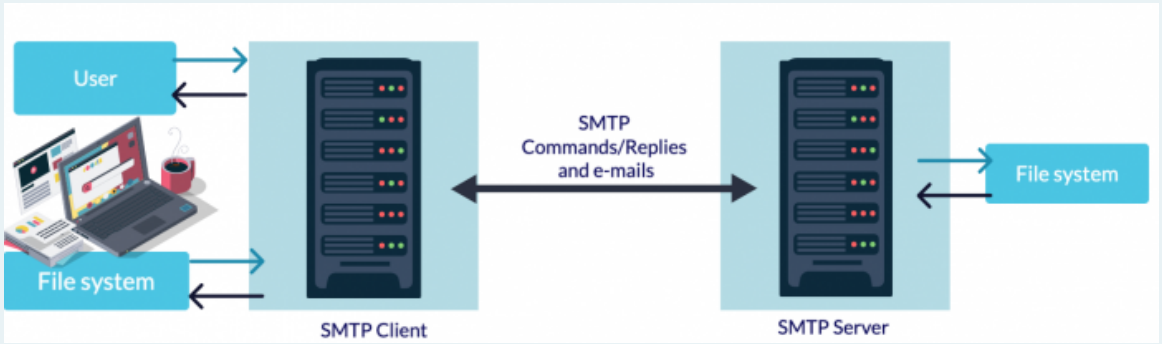


Figure: SMTP working principle

CH-07:Application Layer

Post Office Protocol (POP)

- Which provides access to an inbox stored in an email server.
- Allows us to download and store in local machine $\mapsto$ removed from the server.
- It executes the download and delete operations for messages.

  $\downarrow$

  Thus, when a POP3 client connects to the mail server, it retrieves all messages from the mailbox

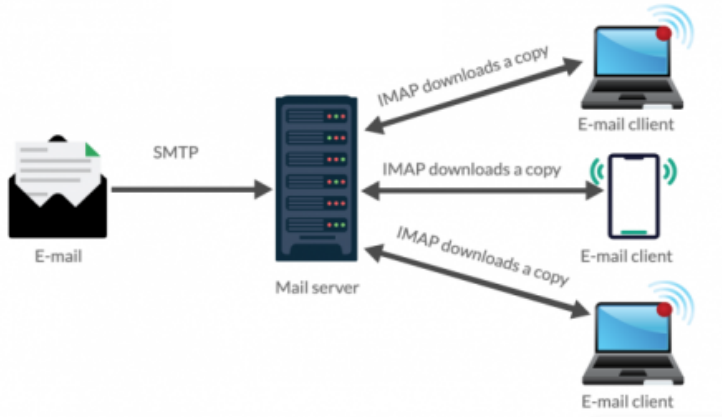
  $\downarrow$

  Then it stores them on your local computer and deletes them from the remote server.
- After establishing a TCP connection at port 110, the POP3 protocol goes through three stages
  - Authorization: $\mapsto$ for the user to login
  - Transactions: $\mapsto$ reading emails, marking them for deletion, ...
  - Update: $\mapsto$ committing deletions

CH-07:Application Layer

The Internet Message Access Protocol (IMAP)

- Allows you to access and manage your email messages on the email server.
- This protocol has more features.
  - a user can check the email header prior to downloading
  - a user can search the contents of the email for a specific string
  - a user can partially download an email; like without attachments
  - a user can create, delete, or rename mailboxes on the mail server (POP3 can do this only on the local machine)
  - a user can create a hierarchy of mailboxes in a folder for email storage
- it uses port 143



→ IMAP working Principle

CH-07:Application Layer

Feature	POP3	IMAP
Where is protocol defined	RFC 1939	RFC 2060
TCP port 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

Table: Comparision on POP3 and IMAP

CH-07:Application Layer

===== COMMON APPLICATION LAYER SERVICES ======

Domain Name Space (DNS)

- Obiviously, Mapping the crucial task Name $\leftrightharpoons$ Ip-address
- Mapping was done using a host file with two columns stored on every host.
- Is it possible to store in current Internet ? $\rightarrow$ NOP!

  $\downarrow$

  Domain Name Space (DNS)
- is a hierarchical, domain-based naming scheme and a distributed database system for implementing the naming scheme
- used for mapping Host names $\rightarrow$ IP addresses
- Name Space
  - Names assigned to machines must be carefully selected from a name space;
  - Names must be unique (as are addresses)
  - Organized in two ways!
    - Flat Name Space
    - Hierarchical Name Space

CH-07:Application Layer

Name Space

- Host names are divided into several pieces called domains
- Domains are designed in a hierarchical structure
- Flat-name Space
  - a name is a sequence of characters without structure; like human names? may be
  - It is Difficult to be used in a large system due to → duplication
  - Example: any names
- Hierarchical-name Space
  - each name is made of several parts;
    - The first may define the nature of the organization,
    - The second the name
    - The third departments,
  - There are authorities to assign and control the name spaces can be decentralized

↓

Central authority assigns only the first two parts

CH-07:Application Layer

- The DNS Name Space
  - The domain name space is also a hierarchical tree structure,
  - The tree can have only 128 levels

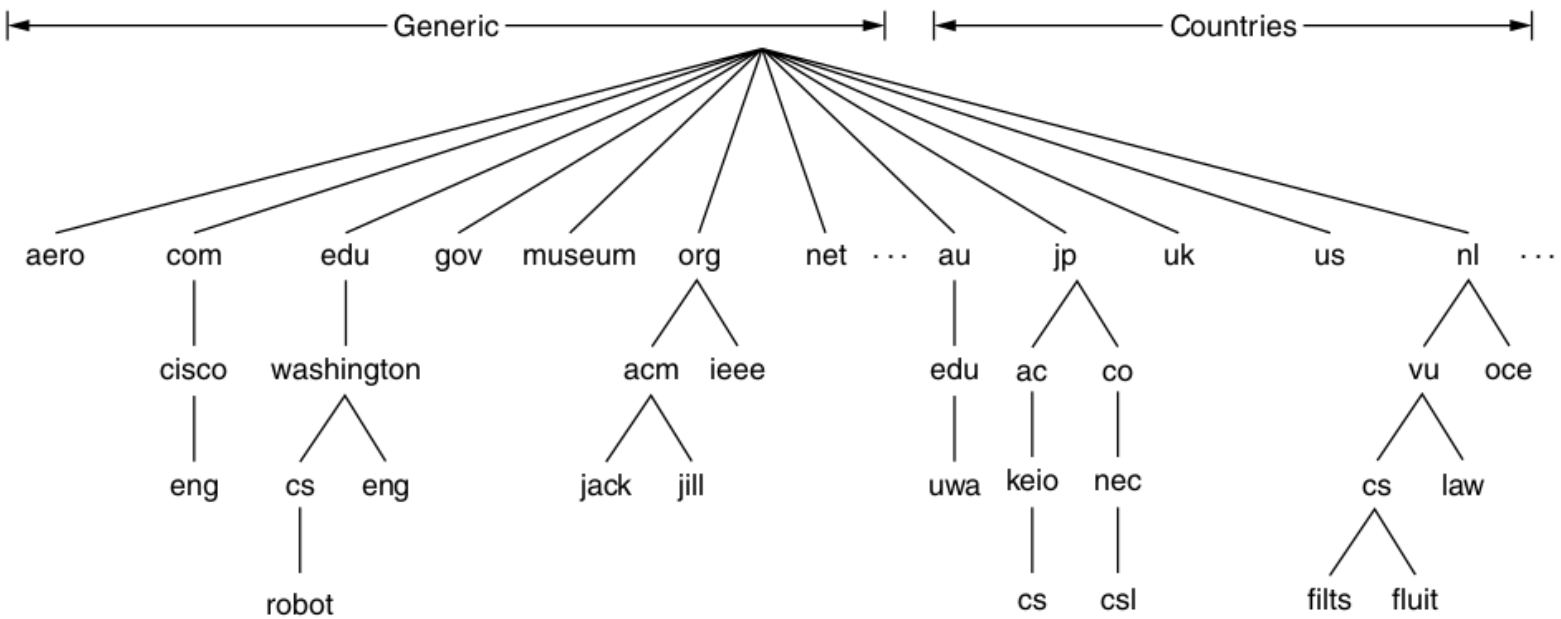


Figure: Domain Name space

CH-07:Application Layer

$\Rightarrow$ Label

- Each node has a label, string $\rightarrow$ max 63 charachter
- the root label is null
- Children of a node must have different names (to guarantee uniqueness)
- Each node has a domain name
- Fll domain name is a sequence of labels separated by dots (the last character is a dot; null string is nothing)
- case insensitive
- Domain names are read from the node up to the root

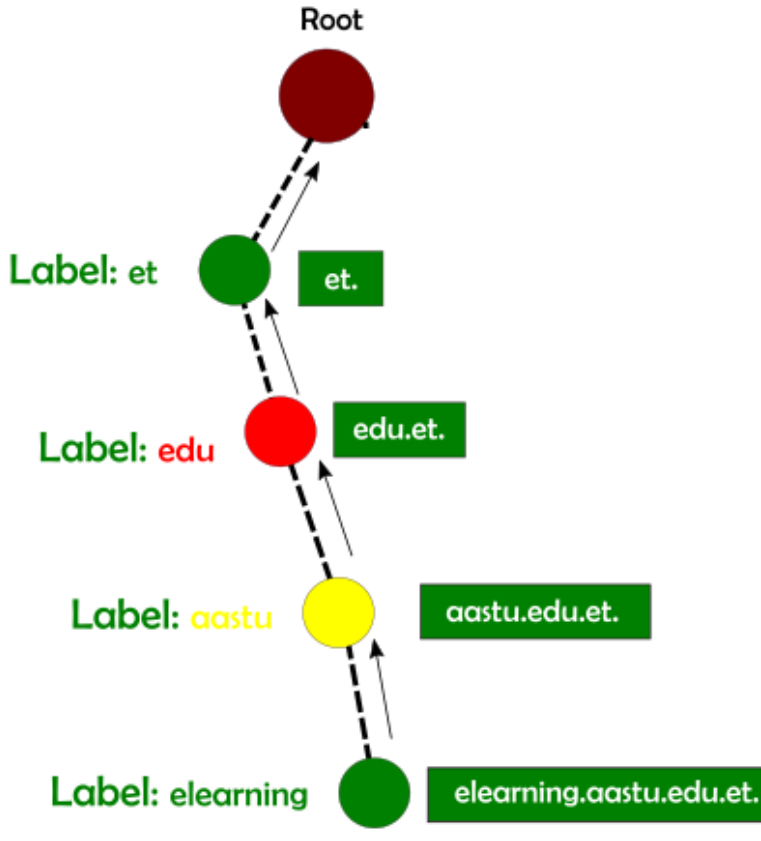


Figure: Label and Domain-name

CH-07:Application Layer

- Domain
  - A domain is a subtree of the domain name space.
  - The name of the domain is the domain name of the node at the top of the subtree
- Two types Domain-Name:
  1. Fully Qualified Domain Name (FQDN) or Absolute
     - Absolute domain name
     - terminated by a null string
     - contains the full name of a host, e.g., sweng.aastu.edu.et.
  2. Partially Qualified Domain Name (PQDN)
     - also Relative Domain-name
     - not terminated with a null string
     - it starts from a node but does not reach the root
     - used when the name to be resolved belongs to the same site as the client (the resolver supplies the missing part, called the suffix to create an FQDN)

CH-07:Application Layer

- A domain is a subtree of the domain name space.
- The name of the domain is the domain name of the node at the top of the subtree.
- The Internet is divided into over 200 top-level domains;
- Each partitioned into subdomains,
- A leaf domain may contain a single host or represent a company and contain thousands of hosts

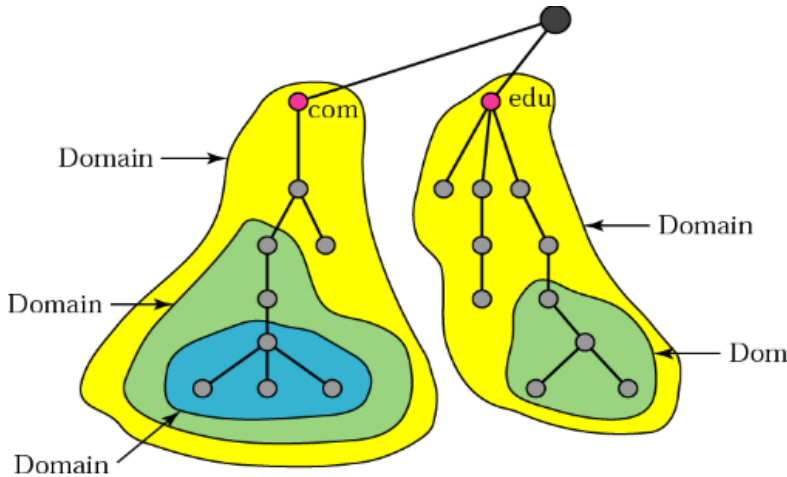


Figure: Domain and sub-domain

CH-07:Application Layer

- Storing the information contained in the domain name space in a single computer is inefficient and unreliable. → Failurity and Load : So?
  ↓
  Hierarchy of Name Servers
- distribute the information among many computers called DNS servers
- There is a hierarchy of name servers as we have a hierarchy of names

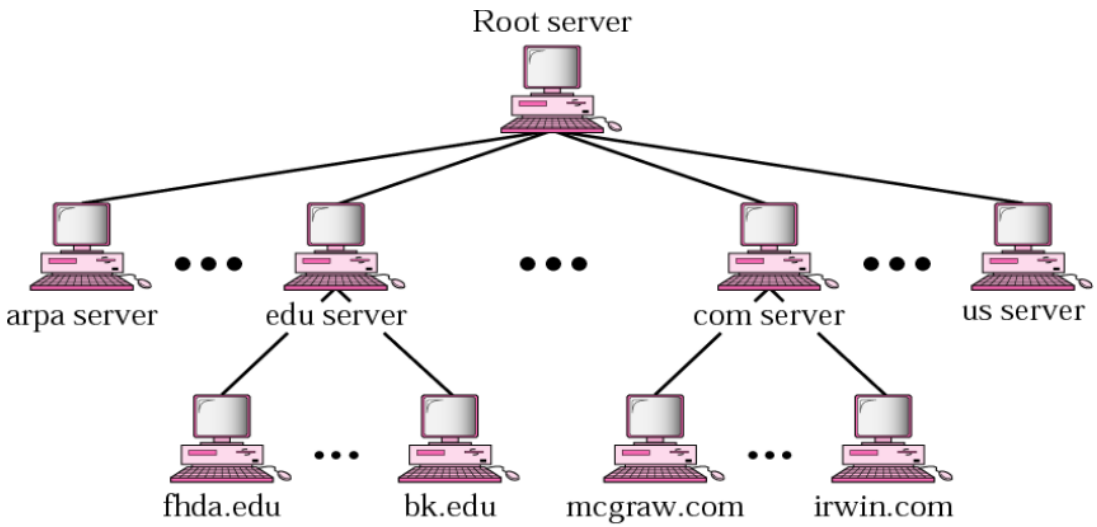


Figure: Hierarchical DNS

CH-07:Application Layer

Types of Top-Level Domains

- 2 types: Generic domains and country domains;
- there is a third one called Inverse Domain

1. Generic Domains
   - define registered hosts according to their generic behaviour

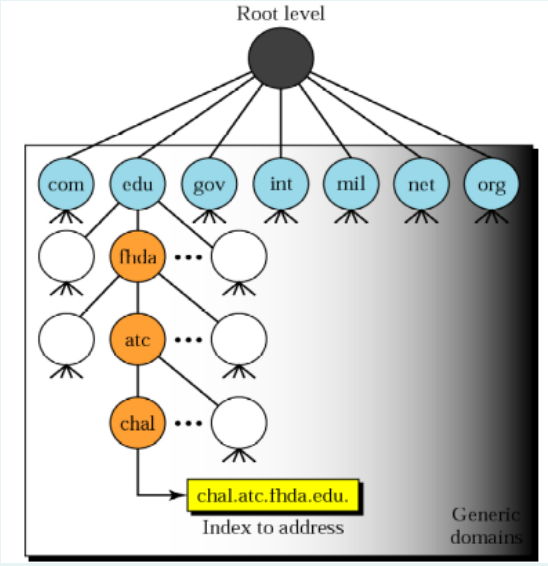


Figure: Some of TLD

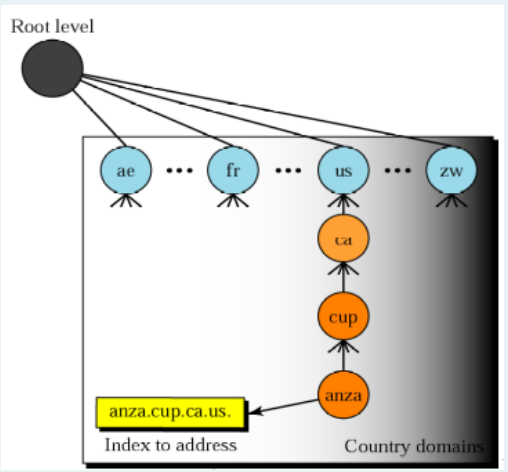
Label	Description
com	Commercial Organization
edu	Educational Institution
gov	Governmental Institution
int	International Institution
mil	Military Organization
net	Network Support Center
org	Non-Profit Organizations

CH-07:Application Layer

Types of Top-Level Domains ...

2 Country Level-domain

- include one entry for every country (as defined by ISO) - two character abbreviations.
- defines name of the countries



Label	Description
et	Ethiopia
fr	France
uk	United Kingdom
us	United State
jp	Japan
nl	Netherland
au	Australia

Figure: Some of TLD

CH-07:Application Layer

$\Rightarrow$ How DNS resolve the name to IP ?

- Resolution $\mapsto$ mapping a name to an address or an address to a name is called name-address resolution
- Resolver $\mapsto$
  - First $\rightarrow$ host that needs to map an address to a name or a name to an address calls a DNS client named a resolver
  - Then $\rightarrow$ the resolver accesses the closest DNS server with a mapping request
  - if the server has the information, $\rightarrow$ provides the resolver;
  - otherwise, it either refers the resolver to other servers $\rightarrow$ called Iterative Resolution

    $\downarrow$

    or asks other servers to provide the information called Recursive Resolution

CH-07:Application Layer

3. File-Transfer Protocol

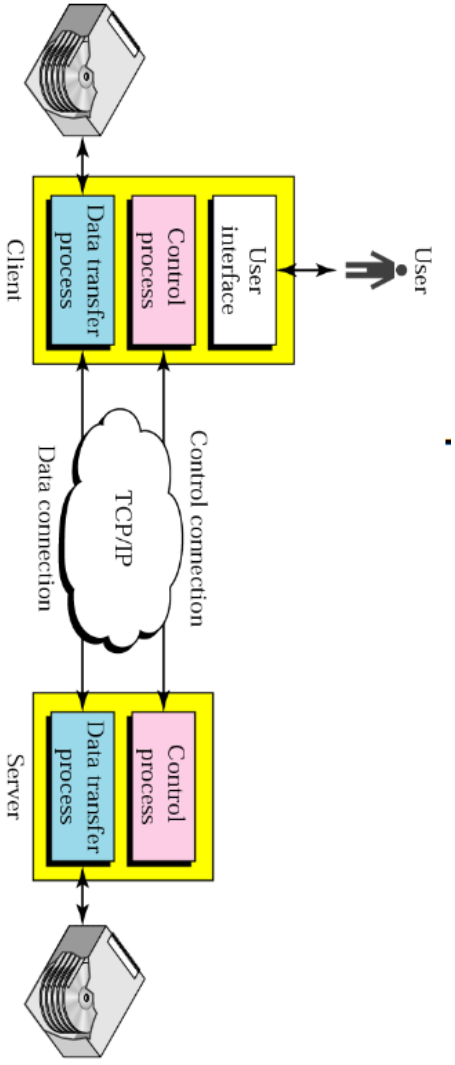
- a standard mechanism provided by the Internet for copying a file from one host to another
- some heterogeneity problems solved by FTP: two systems may
  - use different file name conventions
  - have different ways to represent text and data
  - have different directory structures
- FTP establishes two connections between the client and the server $\mapsto$ different from other client-server applications)

First $\mapsto$ is used for data transfer,
Second $\mapsto$ control information (commands and responses)

- FTP uses the services of TCP; $\mapsto$ it needs two TCP connections;
  - The well-known port 21 is used for the control connection, and
  - The well-known port 20 is used for the data connection

CH-07:Application Layer

- The client has three components and the server has two
  - The control connection is maintained during the entire interactive FTP session;
  - the data connection is opened and then closed for each file transferred
- two different approaches of communication; one each for control connection and for data transfer



CH-07:Application Layer

- Communication over Control Connection
  - communication is achieved through commands and responses
  - each command or response is one short line (no worry about file format or structure
- Communication over Data Connection
  - to solve the heterogeneity problem, the client must define
    1. The type of file to be transferred
       - ▶ Type of data format such as ASCII, EBCDIC file, Image file
    2. The structure of the data, and
       - ▶ File Structure: default mode; the file has no structure; it is a continuous stream of bytes
       - ▶ Record Structure: the file is divided into records (or structs in C); can be used only with text files
       - ▶ Page Structure:
         - the file is divided into pages, with each page having a page number and a page header;
         - the pages can be stored or accessed randomly or sequentially
    3. The transmission mode
       - ▶ Stream mode, Block mode, Compressed mode

CH-07:Application Layer

↦ How do nodes get an IP?

- Manually assigning IP for each node is cumbersome ? ↦ So w@?

  ↓

  Dynamic Host Configuration Protocol (DHCP)

- DHCP is a network management protocol used to dynamically assign an Internet Protocol (IP) address to any device.
- Implemented on small local networks, as well as large enterprise networks.
- Assigns :
  - IP address
  - subnet Mask
  - Default Gateway
  - DNS server address
- DHCP is a client-server protocol in which servers manage a pool of unique IP addresses, as well as information about client configuration parameters, and assign addresses out of those address pools
- DHCP-enabled clients send a request to the DHCP server whenever they connect to a network.

CH-07:Application Layer

DHCP Components

- DHCP has three Component
  - DHCP CLIENT:
    - client is a device – such as a computer or phone – that can connect to a network and communicate with a DHCP server.
  - DHCP SERVER:
    - is a networked device that runs on the DHCP service
  - DHCP RELAYS:
    - The DHCP relay will manage requests between DHCP clients and servers.
    - relays are used when an organization has to handle large or complex networks
  - IP address pool:
    - The range of IP addresses that are available to DHCP clients is the IP address.
    - Addresses are typically handed out sequentially from lowest to highest.
  - Subnet:
    - IP networks can be partitioned into segments known as subnets. Subnets help keep networks manageable.
  - Lease:
    - The length of time for which a DHCP client holds the IP address information is known as the lease.
    - When a lease expires, the client must renew it.

CH-07:Application Layer

Dynamic and Static DHCP leases

- Dynamic DHCP:
  - A Client does not own the IP address assigned to it but instead leases it for a period of time.
  - Each time a device with a dynamic IP address is powered up, it must communicate with the DHCP server to lease another IP address
  - Example: Wireless devices are examples of clients
- Static DHCP
  - static devices – such as web servers and switches – are assigned permanent IP addresses.
- DHCP provides a range of benefits to network administrators:
  - Reliable IP address configuration
  - Reduced network administration
  - Mobility
  - IP address optimization
  - Efficient change management

Read More the details ....

¿END OF CHAPTER SEVEN?!
Question!