Chapter 2

Application Layer

①

- Application layer provides services to users.
- Application layer assumes that there is a logical connection between sender and reciever
- Fig. 2.1 (slide no 2.5)
- Application layer protocols does not provide service to any other protocol in the suite.
- Application layer recieves services from transport layer.
- Fexibility of the application layer is that it allows new application protocol to be easily added.
- Application layer protocol can be standard protocol or non-standaud protocol.
- Standard protocol is a ~~pair~~ computer program that interact with the user and transport layer to provide a specific service to user.
  e.g - HTTP, DNS, FTP etc.
- A programmer can create a non-standard applicat^{n} layer protocol that provide service to user.

Application Layer Paradigm :-

- To use Internet two application program are needed for each party.
- Should both application program be able to request service and provide service.
- For this two paradigms has been developed-

1. Client server paradigm

2. Peer to peer

1. Client server paradigm:

- It is the traditional paradigm
- The service provider is an application program called server process
- Another application program is called client process who asks for service.
- Server run contineously.
- client process started when client needs service
- Fig. 2.2 (Slide no 2.10)
- Communicatⁿ load is more on server.
- Server should be a powerful computer.
- This type of paradigm is used by HTTP, FTP, email.
- An interface in programming is a set of instructⁿ betⁿ two entities.
- A set of instruction like above is called application programming interface (API)
- One such API is socket interface
- Socket is a data structure created and used by application program.
- Communication betⁿ client process and server process is communicatⁿ betⁿ sockets created at both ends.
- Fig. 2.6 (Slide no 2.17)
- Socket address is needed for socket to carry out communication.
- Socket address is the combination of IP address and port number.
- Application layer need to use the services provided by transport layer.
- Transport layer protocols used are UDP and TCP.

(2) Peer-to-peer Paradigm: (P2P)

- Its the new one.
- No need for server procn to run all time.
- All process here are called peers.
- Responsibility is shared between peer.
- Fig. 2.3 (Slide no. 2.11)
- Peer can provide and recieve service at the same time.
- Advantage of peer to peer n/w is that it is easily scalable and cost effective.
- Drawback of P2P n/w is security issues.
- e.g - Bit torrent.

Hyper text transfer Protocol : (HTTP)

- world wide web (www) or web
- HTTP is the most common client server application program used in relation to web.
- web is a repository of information in which the documents are called web pages.
- web pages are distributed and the related documents are linked.
- Linking allows one web page to refer to another stored in some other server.
- linking was achieved through hypertext.
- Hyper text is changed to hypermedia now ~~bcz~~ as web page can be text, image, audio or video.
- www is a distributed client-server service
- client uses a browser to access services.
- Service is distributed over many locations called sites which hold web pages.
- web page can be simple or composite (has link to other)

- Fig 2.8(slide no 2.26)
- web client in the browser.
- Browser consist of three parts:
  - A controller
  - Client protocols
  - Interpreters.
- Fig. 2.9(slide no 2.27)
- Web server stores the web pages.
- server uses cache memory to store requested file.
- e.g - Apache
- Uniform resource locator (URL)
  - web pages are distinguished by identifier called URL.
  - It has 4 parts
    * Protocol
    * Host
    * Port
    * Path.
  - e.g - http://www.google.com:90/powertransmission.

    Protocol Host Port path

- Web documents:
  - Static - fixed content documents
  - Dynamic - created when requested by client
  - Active - script to be run at client site.
- HTTP protocol is used to define how client server program can be written to retrive web pages.
- HTTP client sent request & HTTP server returns a response.
- Server uses port number80 for HTTP.
- client " temporary port number.
- HTTP uses TCP.
- Two types of connections are used in HTTP

(1) Non persistent

(2) persistent

- Non persistent connections :-
  - TCP connection is made for each request/response.
  - Fig. 2.10 (slide no 2.30
  - If a file contains links to N different files the connection must be open and close (N+1) times.
- Persistent Connections :
  - Server leaves the connection open for more requests after sending a response.
  - fig. 2.11(Slide no 2.32)
- Message Format of HTTP:

(1) Request Message:-

- Refer Fig. 2.12 (Slide no 2.33)
- It has 4 parts:
  - * Request line
  - * Header "
  - * Blank "
  - * Body
- Request line:
  - * It has three fields separated by one space and terminated by 2 charecters (Carriage return and line feed)
  - * Three fields are - method, URL, version
  - * Method field defines the request type (Table 2.1) (Slide no 2.34)
  - * Some methods are GET, HEAD, PUT, POST
  - * URL define the address and name of corresponding web page.
  - * Version gives version of protocol like 1.0, 1.1 etc
- Header lines:
  - * It sends additional information from client to server (Table 2.2) (Slide no 2.35)
  - * It has header names, colon, space and value

(2) Response message:

- Refer Fig. 2.12 (slide no 2.33)
- It also has 4 parts
  - * Status line
  - * Header line
  - * Bank line
  - * Body
- Status line
  - * It has 3 fields - version, status code, phrase
  - * Version is the version of HTTP
  - * Status code define status of requent. which consist of 3 digits

    e.g - 100.... - informationdl

    200.... - succenfnl requent

    300... - redirect client to another URL

    400.... - Error at client site

    500.... - " " server "
  - * Header lines (Table 2.3) (slide no 2.36)

- Proxy Servers:-

- HTTP supports proxy servers
- Proxy server Keeps copies of response to recent requests.
- Proxy server reduces loads on original server
- " " decreane the traffic
- " " improves latency.
- " " act as both server and client
- Refer Fig. 2.16 (slide no 2.45)

- HTTPS provides confidentiality, cliend & server authentication and data integrity.

Electronic mail:

- Electronic mail or email allows users to exchange message.
- Fig. 2.19 (Slide no 2.60)
- email uses 3 agents:
  - User Agent (UA)
  - Mail transfer Agent (MTA)
  - Message Access " (MAA)
- MTA client server program is a push program.
- MAA is a pull program.
- email needs two UA, two pair of MTA and a pair of MAA.
- UA provids service to user by sending and recievong messages.
- UA is a software package that composes, reads, replies to and forward messages.
- It handle local mailbox on user computers.
- UA are two types - command driven & GUI based
- To send mail, user uses UA.
- email has a envelope and message.
- Envelope contains sender address & reciever address.
- Message " header and body.
- Header define sender, reciever & subject of message.
- Body of the message contains the information.
- email address.

  local part @ domain name

- Message Transfer Agent:
  - email application needs 3 use of client - server paradigm.
  - fig 2.22 (slide no 2.63)
  - The protocol that defines the MTA client & server

is called simple mail transfer protocol (SMTP)

- SMTP used two times:
    * Bet^n sender and sender's mail server
    * " two mail servers.
- SMTP define how command & responses must be sent back and forth.
- Process of transferring mail occur in 3 phases -
    * Connection establishment
    * Mail Transfer
    * Connection Termination.

- Message access agent:
- At email receiving side, client pulls the messages from mail servers.
- Two protocols are used for this:
    * Post office protocol (POP)
    * Internet mail access protocol (IMAP)
- POP:
    * It is simple but has limited functionality.
    * User needs to download email from mailbox on mail server
    * Port no 110 is used
    * POP has 2 mode - delete & keep.
- IMAP:
    * IMAP has more features than POP
    * User can check mail header prior to downloading.
    * " " search content of email
    * User can create, delete or rename mailbox.

MIME :

- Email can send message only in NVT 7bit ASCII format.
- It cannot use language other than english.
- " " be used to send video or audio data.
- Multipurpose Internet Mail Extensions (MIME) is a protocol that allows non ASCII data to sent through email.
- MIME transforms Non-ASCII data at sender site to NVT ASCII data
- Message at reciever site transform back to original data
- fig 2.25 (slide no 2.70)

Domain Name System (DNS):

- IP address is used to uniquely identify a Host connectn to internet.
- But Host has name instead of numeric address.
- Internet have a directory to map names into a IP address.
- A central directory system cannot hold all mapping.
- If central computer fails, whole network will collapse.
- The mapping is distributed world wide in some servers.
- Fig 2.35 (slide no 2.91)
- DNS client and DNS server map name to IP address

- steps :

  1. User pass the hostname to file transfer client
  2. File transfer client pass the host name to DNS Client
  3. DNS client send message to DNS server.
  4. DNS server responds with a IP address to DNS Client
  5. DNS client pass IP address to file transfer server.

- Name space
  - Name space mapps each address to unique name.
  - Two types :
    - * flat
    - * Heirarchical
  - In flat name space, a name is assign to address.

    e.g - To name a computer of CS dept. lab1 → Computer12
  - Heirarchical - Name made up several part.

    e.g - Kiit. CS. Lab1. Computer12

- Domain name space
  - It is designed to have hierarchical name space.
  - fig = 2.36 (slide no 2.92)
  - Names are defined in an inverted tree structure with the root at the top.
  - Tree can have only 128 labels.
  - Each node in a tree has a label which is a string with maximum 63 charecter
  - children of a node should have different lables
  - Each node in a tree has a domain name.
  - Domain name is a sequence of labels separated by dots (.) .
  - Fig. 2.37 (slide no 2.93)

- Domain name 2 types.
  1. Fully qualified domain name (FQDN)
  2. Partially " " " (PQDN)
- FQDN - Lable terminated with null string or (.)
- PQDN - " not " " " " " "
- A domain is a subtree of domain name space
- Name of domain is name of node at top of tree.
- Fig. 2.38 (slide no 2.94)
- Its unreliable and inefficient to store all domain name space ~~in~~ one server.
- So it is distributed among many computers called DNS servers.
- Hierarchy of name servers:
  - * fig 2.39 (slide no 2.95)
  - * Three levels of DNS servers.
    1. Root Servers
    2. Top level Domain Servers (TLD)
    3. Authoritative servers
  - * Root servers can stand alone and create as many ~~at~~ domain.
  - * ~~Do~~ TLD can divided into small domains.
- What a server is responsible for or authority over is called zone.

- Root Servers:
  - It is a server whose zone consist of the whole tree.
  - It does not store information about a domain but delegates its authority to other servers.
  - Two types of Servers
    1. Primary servers: It is a server that stores a

file about the zone for which it is an authority.

* Its responsible for creating, maintening and updating the zone file.

* Stores zone file on local disk.

(2) Secondary Servers.

* It is a server that transfers the complete information about a zone from another server and stores the file on local disk.

- Initially domain name space divided into generic domain, Country domain and inverse domain.
- Generic domain defined according to behaviours.
- Country " " " " country.
- Fig. 2.41 and Fig 2.42. (Slide no 2.97 and 2.99)

DNS Resolution:

- Mapping name to an address is called name-address resolution.
- DNS designed as client server architecture.
- A host that map name to address or address to name is called resolver.
- Resolver access the closest DNS server with mapping request
- DNS resolution is done in 2 ways:

(1) Recursive Resolution:

- Fig. 2.43 (Slide no 2.100)
- Application program of source host calls the DNS resolver (client) to find the IP address of destination host.
- Resolver send the query to local DNS server
- If local DNS server does not know it will send a query to root DNS servers
- Root DNS server will send query to TLD servers.
- TLD server will send query to authoritative DNS server.
- Authoritative DNS server will send corresponding IP address to TLD server.

- TLD server will send IP address to root servers.
- Root " " " " " " local DNS servers.
- Local DNS server will send the IP address to source.

(2) Iterative Resolution

- fig. 2.44 (Slide no 2.101)
- Each server that does not know the mapping sends the IP address of the next server back to the one that requested it.

- DNS Caching:

- When server asks for a mapping from another server and recieves the response, it stores the information in its cache memory.
- Caching speed up the resolution
- If server caches a mapping for long time, it may send outdated mapping
- So authoritative server always add information to mapping called time to live (TTL).
- TTL define the time a recieving server can cache the information.

- DNS Resource Record:

- Zone information associated with a server is implemented as a set of resource record.
- A resource record is a 5-tuple structure.

  (Domain Name, Type, Class, TTL, value)

- Domain name identifies the resource record.
- Value field define information kept about the domain name.
- TTL define number of second it is valid.
- Class define the type of network. Here the class is internet (IN)

- Type debine how the value should be interpreted.
- List ob types (Table 2.13) (Slide no.102)

DNS Messages:

- To retrieve information about hos DNS uses 2 types of messages:
  1. query
  2. response
- Fig. 2.45 (Slide no 2.103)
- Identification field is used by client to match the response with the query.
- Flag field define whether the message is query or response.

- DDNS - Dynamic DNS → DNS files are updated dynamically using DDNS.