**UNIT-V**

**APPLICATION LAYER:**

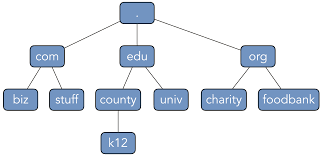
* DNS-Domain Name System
* Electronic mail
* World wide web(www)

**DNS:**

Domain Name System, the essence of DNS is the invention of a hierarchical domain based naming scheme and a distributed database system for implementing this scheme.

* DNS name space socket
* Resource records
* Name servers

**DNS Namespace socket:**



* DNS following the postal hierarchical addressing system, using the postal addressing system there is no confusing.
* The internet is divided into over 200 top level domain where each domain covers many hosts.
* Each domain is partitioned into sub domains and these are further partitioned.
* All these domains can be represented by a tree, they leafs of the tree represented domains that have no sub domains but it contain machine.
* They top level domain divided into two categories.
* General
* Countries

The original Generic domains

Com-[commercial]

Edu-[educational institutions]

Gov-[us federal government]

Int-[international organization]

mil-[us military]

net-[network provider]

org-[non profit organization]

The country domain include one entry for country.

* In nov2000 added four new General purposed domain namely,

.Biz[business]

.INFO[information]

.name[people name]

.pro[professional such as doctors & layers]

In addition 3 more specialized top level domains where introduced.

Aero-[aerospace]

Coop-[cooperative]

Museum-[museum]

Getting second level domain such as name of the company. Com is easy, it requires going to register for the corresponding top level domain.

Ex:

* cs.bdu.com
* They components are separated by period(.), domain names are case insensitive so edu,Edu,EDU,eDU all are same.
* Component name can be upto 63 characters long and full path name must not exceed 255 character.

Ex:

Cs.yale.edu

Cs.yale.ct.us

**Resource record:**

* Every domain whether it is a single host or a top level domain, can have a set of resource record associated with it.
* In single host they most common resource record is just it internet port address.
* The resource record is 5 tuples resource records are presented as ASCII text, one line per resource record.

**Format:**

Domain name Time -to –live class type value.

**Ex:**

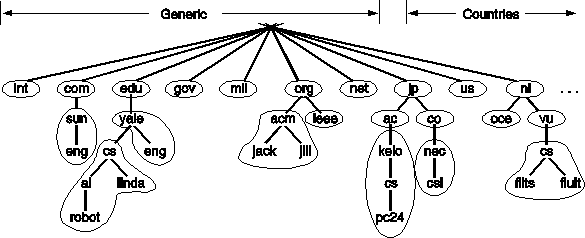
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cs.mail.edu | 86400 | In | CName | Cs.mil.edu |

* The domain name tells the domain to which this record Applies.
* The time-to-live fields gives on indication of how stable is record.
* The third field of every resource record is the class for internet information it is always IN.
* The type field tells what kinds of record or files.
* Value –this field can be a number, a domain name or ASCII string.

|  |  |  |
| --- | --- | --- |
| **Type** | **meaning** | **Value** |
| SOA | Start of authority | Parameter for this zone. |
| A | IP address of a host | 32 bit integer. |
| MX | Mail exchange | Domain willing to accept email. |
| CNAME | Canonical name | Domain name |
| PTR | Pointer | Alias for an IP address. |

**Name server:**

* A single name server contain the entire DNS database and response to all request about it.
* A DNS namespace is divided into non over lapping zones.



**Electronic mail:**

* Email, like most other forms of communicative has it’s own convention & style.
* It’s very informal and has a layer threshold of use.

**Architecture & service:**

Email system consists of two sub system.

* User agent(UA)
* Message transfer agent(MTA)

**User Agent:**

* It allows people to read & send emails.

**MTA:**

* It moves the message from the source to the destination.

**Basic function:**

* Composition
* Transfer
* Reporting
* Displaying
* Disposition

**Composition:**

* It refer to the process of creating message and answer.
* Any text editor can be used for the body of the message.

**Transfer:**

* It refers to moving message from the originator to the recipient.
* The email system should to this automatically.

**Reporting:**

* It has to deal with telling the originator what happen to the message.

Ex:

Was it delivered?

Was it rejected?

Was it last loss?

**Displaying:**

Displaying incoming message is needed, so people can read the email.

**Disposition:**

* This is the final step the recipient does with to the message after receiving it.

**Extra function:**

* Mail box
* Mailing list

**Mail box:**

* To store incoming emails.

**Mailing list:**

* It contain a list of email address.

**User Agent:**

* The user agent is normally a program that accept a variety of commands for composing, receiving and replaying to message as well as manipulation mail boxes.
* Sending email
* Reading email

**Sending email:**

* To send on email messages, a user must provide the message, the destination address and some other possible parameters.
* The message can be produced with text editor, a word processing program build into the user agent.
* The destination address must be in a format that the user agent can deal with it.
* \c=US\ST=NEW YORK\<=cambridge/PA=36,south st/CN=Trichy/.
* They address specifies a country, states location, personal address and a common name.
* Most email system have aliases that allow users to enter or select a persons name and get a correct email address.
* Most e – mail system supporting mailing list, so that a user send a message to a list of People.

**Reading email:**

* The user mailbox for incoming email before displaying anything on the screen, then it may announce they number of message is the mailbox.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Flag | bytes | sender | Subject |
| 1  2  3  4 | K  KA  KF  K | 1.30  1500  350  850 | Anitha  Mercy  Santhi  punitha | Hai  Hello  Resume  welcome |

* First field is the message number, second field flag, k means that the message is not new but was read previously and kept in the mail box.
* Flag A means that the message has been already answered.
* The third field tells the how long the message.
* The fourth field tells who send the message, this field simplify extracted from the message this field may contain first name, last name initial.
* Finally the subject field give a brief summary of what the message is about.
* After the header has been displayed the user can perform any of several actions .

**Message format:**

* RFC-822: consist of some primitive envelope.
* To-email address of primary recipient multiple recipient is also allowed.
* cc-email address of secondary recipient.
* Bcc-email address of blind carbon copy.
* From-person to create a message.
* Sender-email address of the actual sender.
* Receiver-line address of the transfer agent along the root.
* Return path-can be used to identify a path back to the sender
* Date-the date & time they message was send.
* Replay to-email address to which reply as should be send
* Message ID-unique number for referencing this message later.
* References-other relevant msgid.
* Keyword – user chose the keywords.
* Subject-sort summary of the message.

**MIMF(Multipurpose internet mail extension):**

* In the early days of email consisted exclusively of text message written in English and expressed in ASCII string.
* It specify the headers only, now a days this approach is no longer, problem include sending and receiving.
* A solution was proposed in RFC 1341 and updated in RFC 2045-2049, this solution is called MIME.
* Now it is widely used, the basic idea of MIME is to use the RFC-822 format, but add structure to the message body and define encoding rules for non ASCII message.
* MIME defines 5 new message heads

|  |  |
| --- | --- |
| **Header** | **Meaning** |
| MIME version | Identities the MIME version |
| Content description | Human readable string telling what is in the message |
| Content-ID | Unique identifier |
| Content transfer encoding | How the message body is wrapped for transmission. |
| Content type | Type and format of the content. |

* MIME VERSION is used identify the MIME version and any message not contain a MIME version header is assumed by English blind text message.
* Content description header is on ASCII string what is in the message.
* Content id header identify the content, it uses the same format as the standard message id header.
* Content transfer encoding tells how the message be is wrapped for transmission through a network.
* Content type header specify the nature of the message body, each have one or more sub type separated by slash.

Ex:

Content type=video\MPEG

|  |  |
| --- | --- |
| Type | Subtype |
| Text  Image  Audio  Video | Main enriced  JPEG.,GIF  Basic  MPEG |

**MTA:**

* Message Transfer Agent
* The system is concerned with relaying message from originator to the recipient.
* The simplest way to also this to establish a transport connection form source machine to the destination machine and then just transfer message.

**SMTP: [SIMPLE MAIL TRANSFER PROTOCOL]**

* Within the internet email is delivered by having the source machine establish a TCP connection to port 25 of the destination machine, this port accept Incoming connection and copy message from them into the appropriate mail boxes.
* If a message cannot be delivered on failure report is return to the sender.
* SMTP is a simple ASCII protocol a sending machine operating as the client, wait for the receiving machine operating as the server to talk.
* All the emails has been exchanged in both direction.

Ex:

MAIL from:<elinor@abcd.com>

Sender of

RCPT TO:[carlyn@xyz.com](mailto:carlyn@xyz.com)

RECEIVER ok

Data end with”.”

From:[elinor@abcd.com](mailto:elinor@abcd.com)

TO: [carolya@xyz.com](mailto:carolya@xyz.com)

MIME VERSION:1.0

Message id:<0704AA014>

Content type:Text/plain

Subject: Hai friend

Content transfer encoding: base 64 quit.

**WWW Architectural overview:**

* Client side
* Server side

**Client side:**

URL Has three Parts.

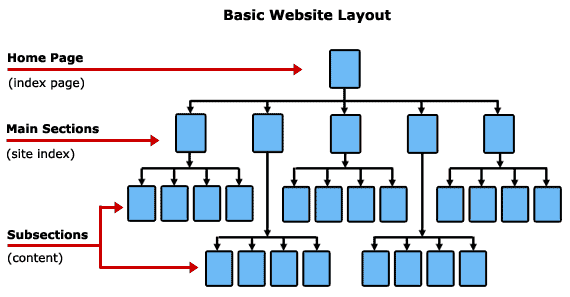
1. Determine URL by browser.
2. Browser asks DNS to want IP Address.
3. DNS replay.
4. The browser make TCP Connection of port 80.
5. If then send request.
6. Server send the file.
7. Connection released.
8. Display all text.

**Server side:**

1. Accept TCP Connection.
2. Get the name of the file.
3. Get the file from the disk.
4. Return to the client.
5. Release.

**World wide web:**

* World wide collection of documents is called web pages each page may contain link to other pages any where in the world.
* The idea of having one page point to another page is called hypertext.
* Pages are viewed with a program called browsers, internet explorer, Netscape navigator are one of the popular browsers.



* They browser display in a web page on the line text. That is linked to a page on the abcd.com.
* They browser follow the hyperlink by sending a message to the abcd.com server asking a page when page arrives it is displayed.
* It this page contains the hyper link to a page on the xyz.com.
* That is clicked on the browser then send a request to that machine for the page.

**Client side:**

* Client side on a page is selected, they browser follow the hyperlink and fetch the page selected.
* The embedded hyperlink way to name any other page on the web.
* Pages are named using uniform resource locator (URL).

A difficult URL is

* <http://www.abcd.com\fibo.html>

It have three parts:

* Name of the protocol(http)
* DNS name of the machine where the page is located ([www.abcd.com](http://www.abcd.com))
* Name of file containing thepage(fibo.html)

A browser click on the a hyperlink, the browser carry out a series of steps in order to fetch the page.

1. They browser determines the URL.
2. The browser ask to DNS for the IP address
3. DNS replays with 156.106.192.32
4. The browser makes a TCP connection to port 80 on 156.106.192.32.
5. If then send over a request asking for file.
6. The [www.idu.org](http://www.idu.org) server send the file
7. The TCP connection is released
8. The browser display all the text
9. The browser fetch and display all images in this file.

**Server side:**

* Accept a TCP connection from a client
* Get the name of the file requested.
* Get the file from disk.
* Return the file to the client.
* Release the TCP connection.