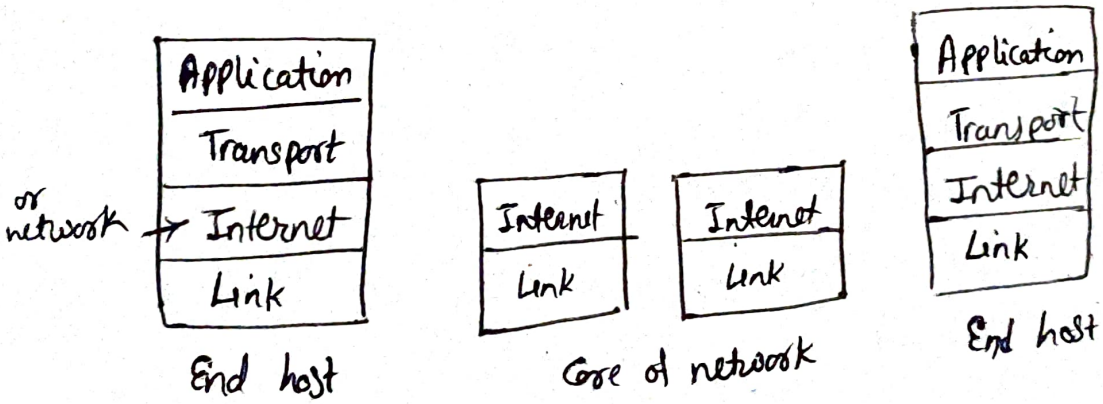
Web Programming
Class Notes

- ## Internet Architecture

1) Internet is a network. A network is a collection of computers that can communicate with each other. Each computer in a network is called host.

2) To promote unification of disparate networks, a layered architecture with a suite of protocols was invented to unify the networks.
A protocol is set of rules that partners in communication use when they communicate.

Layered Architecture of Internet



Link Layer

- Physical transmission of raw bits across network media.
- MAC (media access control) addresses are unique identifiers assigned to network hardware and which are used at physical networking level.

Network Layer

- Taking individual packets of information & forwarding them to their destination.
- It routes packets between communication partners across networks.
- The Internet uses the Internet Protocol (IP) address to identify destinations on the internet.
- Every device connected to Internet has an IP address, which is numeric code that is meant for unique identification.

Transport Layer

TCP uses UDP (User Diagram Protocol)

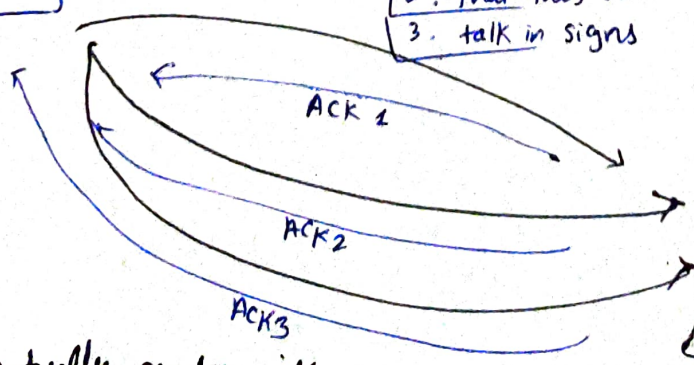
- The transport layer ensures transmission arrive in order & without error.
- Web communication in particular uses the one: Transmission control protocols (TCP).
- whereas IP is implemented on the routers in core of network and on end hosts, transport layer protocols only have to be implemented on end hosts.

Message = Thou map of woe, that thus dost talk in signs!

Sender

① message broken into packets with sequence number

1. Thou map of woe
2. that thus dost
3. talk in signs



receiver

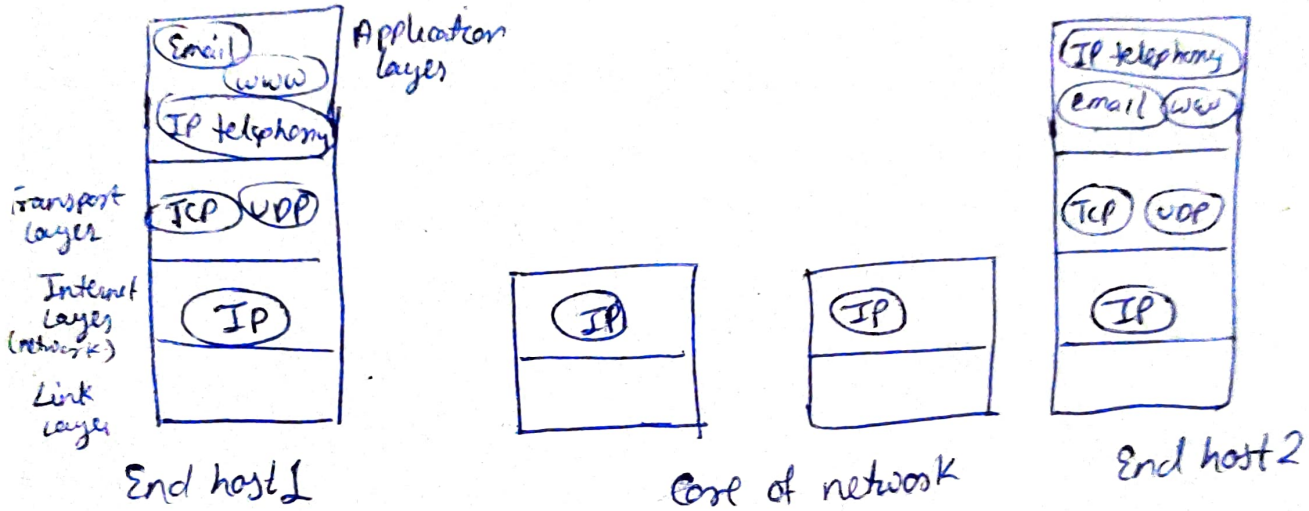
② for each TCP packet sent, an ACK (acknowledgment) must be received back

③ Eventually, sender will resend any packets that didn't send an ACK back

④ message is reassembled from packets & ordered according to sequence number.

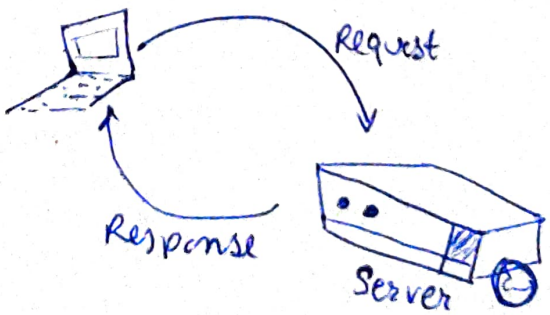
Application Layer

- Contains a range of protocols that let applications communicate with each other
- To communicate with an application on a different end host, both application programs exchange data according to format & conventions specified in application protocol.



Client - Server model

- 2 type of actors
  - clients & servers



WWW

- Tim Berners Lee first outlined the advantages of hypertext-based, linked information system.
- The term hypertext persists as label for technology that connects documents and information resources through links
- The World Wide Web (WWW), commonly known as Web, is an information system where documents and other web resources are identified by Uniform Resource Locator (URL) which may be interlinked by hypertext, and are accessible over the internet.
- The resources of Web are transferred via Hypertext transfer Protocol (HTTP) and may be accessed by a software application called a web browser and are published by a software application called web server.

Cookie

An HTTP cookie is a small piece of data sent from a website & stored on user's computer by web browser while user is searching. Cookies are used to remember stateful information (such as item added in cart) or to record user's browsing activity.

Authentication cookies are most common used to know whether the user is logged in or not, and which account they are logged in.

Lect 2 URL

1) URI (Uniform Resource Locator / identifier)

URI is a string of characters that identifies a particular resource. To guarentee uniformity, All URI's follows a predefined set of syntax rules.

→ 2 types
URL , URN

- Uniform Resource Name (URN) is a URI that identifies a resource by name in particular namespace. A URN may be used to talk about a resource without implying its location or how to access it.
- Uniform Resource locator (URL) also termed as web address is a reference to web resource that specifies its location on computer network and a mechanism of retrieving it. A URL is specific type of URI, although many people use this terms interchangebly. URL occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC) and many other application.

Syntax of URL

URL = scheme : [// authority] path [?query] [# fragment]

authority = [userinfo@] host [:port]

- **scheme** → identifies protocol to be used to access the resource on internet.
- **host** → web server name or IP address
- **port** → port number, web server listens to.
- **path** → refers to exact location of page, post, file or other asset. It is often analogous to underlying file structure of website.

  But not all URL display the path

eg

https : // example.com / media/upload/filename.jpg

scheme — authority — path

- **Parameters** (query) → Parameters are found at very end of URL or within the path. Parameters are commonly used for tracking & analytics - as well as encoding specific information for use within website & application, represented in key/value pairs, beginning with '?' and seperated by '&'.

  https : // www.example.com / solutions? user = 123 & color = blue

- **fragments** → HTML Anchors are used on websites to implement "bookmarks" & internal page navigation elements. These can be used to provide links to specific locations within a page.

DNS

- Every device connected to internet has an IP address, which is a numeric code that is meant to uniquely identify it.
- IP addresses are 32 bit addresses.
- DNS (Domain Name System) is like phone book of internet. Humans access information online through domain names like espn.com. Web browser interact through Internet Protocal (IP) address. DNS translates domain names to IP address so browsers can load internet resources.

- Domain names have hierarchical structure

most general

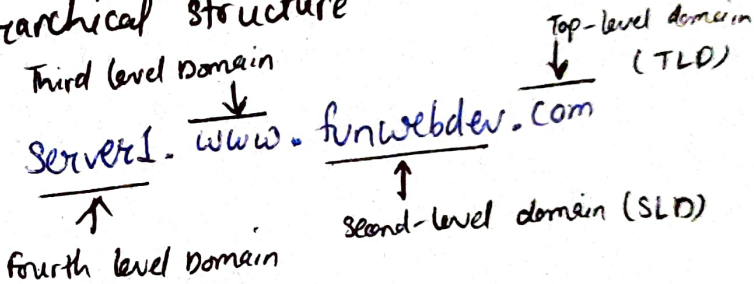
Top level domain (TLD) — Com

Second level Domain (SLD) — funwebdev

Third level Domain — www

Fourth level Domain — server1

host specific



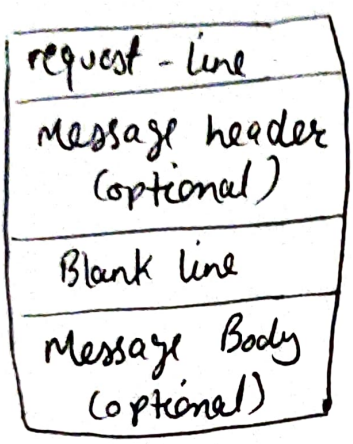
HTTP

Hyper text transfer protocol. The communications protocol used to connect to web servers on internet or on a local network (intranet). Its primary function is to establish a connection with the server and send HTML pages back to the user's browser. It is also used to download files from server either to the browser or to any other requesting application that uses HTTP.

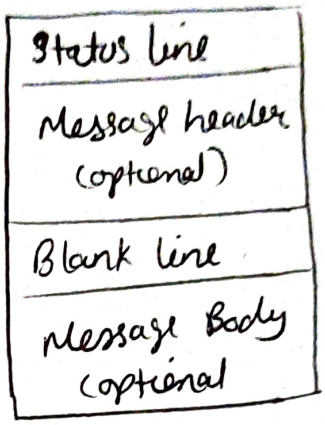
- HTTPS (HTTP secure) is encrypted version of HTTP.
- (HTTP) The client-server TCP/IP protocol used on World-wide web for exchange of HTML documents. It conventionally uses port 80.
- HTTP is a "stateless" request/response system.
- The structure of request & response messages contain message headers, followed by blank line, followed by message body.

HTTP message & its Structure

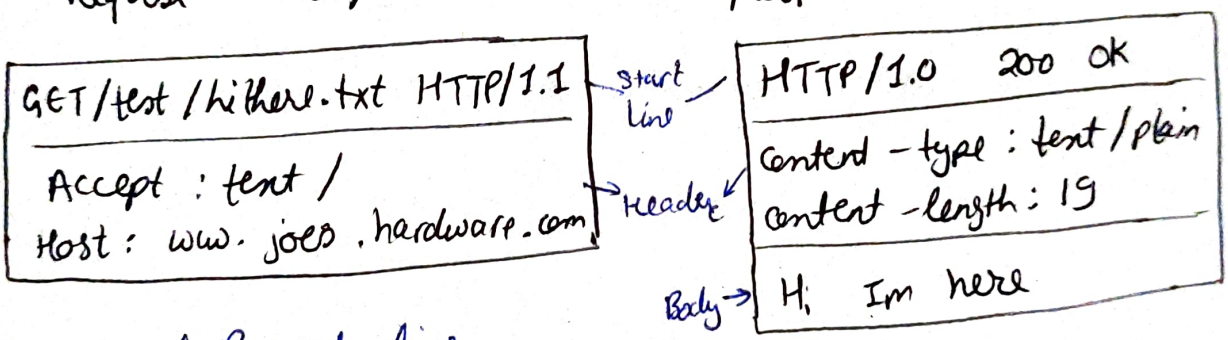
- HTTP messages are how data is exchanged between a server and client. There are 2 types of messages: requests sent by client to trigger an action to server, and responses, the answer from server.



Request Message



Response Message



- format of Request-line

<method> <request-url> <version>

GET/ test HTTP/1.1

- format of status-line

<version> <status> <response phrase>

eg HTTP/1.1 200 OK

Response Status code

100-101	Category
200-206	Informational
300-305	Redirectional
400-415	client error
500-505	Server error

200 - OK
301 - Moved permanently
304 - not modified
307 - Temporary redirect
400 - Bad request
401 - unauthorized
404 - not found
500 - Internal server error

Request methods

(Message Body)

GET → get document from server (No)
POST → Post sends data to server for processing (YES)
HEAD → Get just header for document from server (No)
PUT → Store body of request on server (YES)
Trace → Trace the message through proxy server to server (No)
DELETE → Remove Document from server (No)

Working

HTTP HEADERS

HTTP header fields are components of header section of request & response message in Hypertext Transfer Protocol (HTTP). They define the operating parameters of an HTTP transaction.

Some common headers are :-

- General headers
- Request headers
- Response headers
- Entity headers
- Extension headers.

→ header fields are transmitted after request line (in case of request HTTP message) or response line (in case of response HTTP message), which is first line of message. Header fields are colon-seperated key-value pairs in clear-text string format.

eg, browser may indicate that it accepts information in German or English, with German as preferred by setting the q value for de higher than that of en, as follows :

Accept-Language : de; q=1.0, en; q=0.5

1) General header :- These header fields have general applicability for both request and response message.

5- Date : Sun, 11 Feb 2001
- Connection : Close
- Transfer - encoding : gZIP

2) Request header :- Make sense only in request message
- Server can use information about client to try to give client a better response
• From : , • Host : , Referer : , User-Agent : , Accept : Accept-Charset : , Accept-Encoding :

3) Response header :-
Help server to pass additional information about response that can not be infered from status code alone.
• Location : Server : Age : Retry-After :

4) Entity header :-
These header field define meta information about entity-body or, if no body is present, about resource identified by request.

Ruby on Rails

- Simple ruby code

```
class Hello
    def initialize (name)
        @name = name.capitalize
    end
    def salute
        puts "Hello #{@name}!"
    end
end
h = Hello.new("Ruby")
h.salute
```

output → Hello Ruby!

Embedded Ruby

Ruby provided program called ERB (Embedded Ruby), it allows to put Ruby code inside HTML file.

→ if you want some Ruby code executed, enclose it betn <%. & %>

→ if you want result of code executed to be printed out, as a part of output, enclose code betn <%= and %>

ruby file saved as .rb

```
<%. page_title = "Demonstrate ERB" %>
<%. salutation = "Dear programmer." %>
<html>
    <head>
        <title><%.= page_title %.></title>
    </head>
    <body>
        <p><%.= salutation %.></p>
        <p> this is example of ERB </p>
    </body>
</html>
```

executed as

erb erbdemo.rb

Rails strengths

1) Metaprogramming →
2) Active record → which saves objects into database.
3) Convention over configuration → not needed much configuration
4) Scaffolding → early stage code generated itself
5) Built in testing →
6) Three environments → development, testing, production

Installation

1) install ruby.
2) on installation you also get Ruby gems, to install rails do
   gem install rails

Framework - Rails

A framework is a program, set of programs and/or code library that writes most of your application for you.

MVC

Model View controller.

1) Model (active record) It maintains relationship betn objects & database & handles validation, association, transactions & more.
2) View (action view) It is presentation of data in particular format, triggered by controller's decision to present the data. They are script-based technologies like JSP, ASP, PHP & easy to integrate with AJAX use ERB. Every web connection to a Rails application results in displaying of view.
3) Controller (active controller) It directs traffic on one hand, querying models for specific data on other hand, organizing that data (search, sort, messaging it) into a form that fits the needs of given view.

Action view & active controller are bundled together under action Pack. Active record provides a range of programming techniques & shortcuts for manipulating data from an SQL database. Action controller & action view provide facilities for manipulating & displaying that data. Rails ties it all together.

Directory structure

it has directories

- app → It organizes your application components. It's got subdirectory that hold view (view & helphers), controller (controllers), & backend business logic (models).
- app/controllers → here Rails looks to find controller classes. A controller handles a web request from user.
- app/helpers → holds helper classes used to assist model, view & controller classes. This helps to keep model, view & controller code small, focused & uncluttered.
- app/models → holds classes that model and wrap data stored in our applications database.
- app/view → holds display templates to fill in with data from our application, convert to HTML and return to user's browser.
- app/view/layouts → holds template files for layouts to be used with views. This models common header/footer methods. Stored as default.html.erb.
- Components → holds components, tiny self contained applications that bundle model, view & controller.
- Config → contains small amount of configuration code that your app will need, including database configuration (database.yml), rails environment structure (environment.rb), routing of incoming request (routes.rb)

- db → You can manage relational database with scripts you create & place in this directory.
- doc → Ruby has a framework, called RubyDoc, that can automatically generate documentation for code you create. You can assist RubyDoc with comments in your code. This directory holds all RubyDoc-generated Rails & application documentation.
- lib → You will put libraries here, unless they explicitly belong elsewhere
- log → Error logs go here.
- public → Like public directory for web server, this directory has web files that don't change, such as javascript files, graphics, HTML, css
- script → This directory holds script to launch & manage various tools that you'll use with rails, eg there are scripts to generate code (generate) & launch web server (server)
- test → tests you write & that Rails create for you are here.
- tmp → holds temporary files for intermediate processing
- vendor → Libraries provided by third-party vendors (such as security libraries or database utilities beyond basic rails distribution) go here.

Create app library

rails new library

→ Starting web server default 3000

Rails Server

Notes →

- The controller / action will be accessible for external web requests in Rails only if a corresponding route is mapped to it.
- Rails application framework is called → Action Pack
- Active record is rails object - relational mapping library (ORM)

Ertoa

1) Firewall is used to prevent unauthorized access of private networks through packet screening from outside used, especially internet.