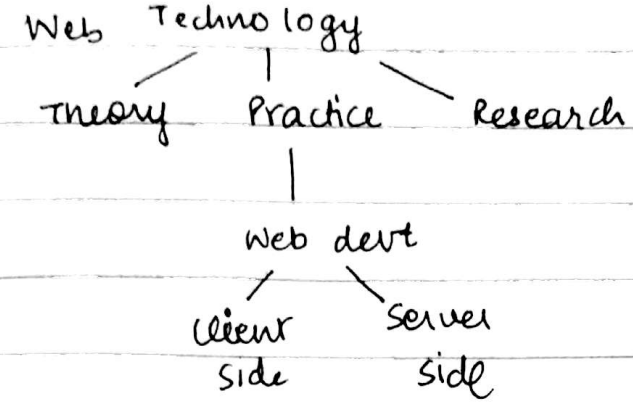
Blank Page

WEB TECHNOLOGY



Frameworks → Django, Ruby on rails, JSNode.

Data	Information	Knowledge
(Chunk, raw unstructured) - Details of every student	(specific, structured) - DOB of every student	(what we extract) - How many are adults vs minors.

INTERNETWORKING :

Acronym

Need of networking : (1) communication
(2) sharing resources
(3) sharing information.

Metadata : sth which tells us about data. (Data about data)

Meta: replicate of sth. eg. Data about data (meta-data)
Network of networks. (meta-n/w)

Protocols
HTTP
SMTP
FTP
IMP
VOIP

Internet:

(1) Network of networks (aka metanetwork)
(2) Collection of small LANs to form one big WAN.

components ~~of~~ ~~&~~ ~~netw~~ which help internet to work:

(1) Client - server (Architecture)
(2) Protocol
(3) IP address
(4) DNS
(5) URL

Internetworking
- Internet computing
- Internet tech

Every service of I/n is either a client (who seeks info) or server. (provides service or responds to request of client).

90% of services provided by internet have client - server interaction

eg. we access a website
↓ client ↓ ~~client~~ server

skype chat (client uses server to get address of other person)

Protocols

HTTP	Web
SMTP	Email
FTP	File transfer
IMP	Instant messaging
VOIP	Internet Technology

} SERVICES PROVIDED BY INTERNET

Protocols: Set of Rules / regulations that help client & server's interaction.

http:// www. .....

IP Address: Every machine on the internet has unique address. (address where sth is stored)

Microsoft: 207.46.92.254

we cant remember IP addresses, so we remember URLs.

~~Mesp~~

Mapping b/w IP address & URL is done by DNS.

DNS: Domain Name Server.

ANATOMY OF URL:

protocol:// host / path / filename.

URL: Uniform Resource Locator

Every host machine has unique address → IP address

Path of every file on that host → URL.

http://www.dtu.ac.in/academics/syllabus

Protocol | host (www.) | path | filename

Hypertext Transfer Protocol

More than text

(videos, audios, advertisements, copyrights are present APART from text.

Multimedia + text)

www.dtu.ac.in

2nd LEVEL | 1st LEVEL | TOP LEVEL DOMAIN NAME

ac → Academic Institution

in → country India

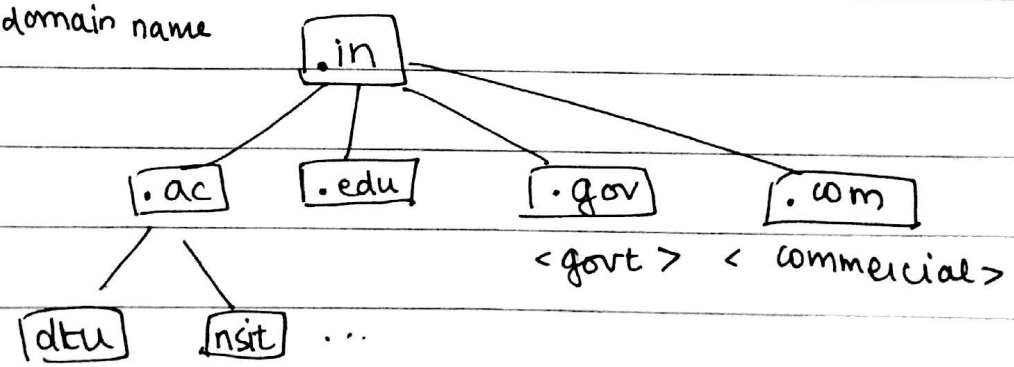
.in
.jp
.UK
:

Hierarchical tree:

→ Top level domain name

→ 1st level

→ 2nd level



DIRECTORY

http:// www. dtu.ac.in /academics / syllabus

protocol Host Path Filename

Bookmark : URL stored by web browser.

Browser: used to access URLs.

services provided by internet can be used only because of an **ISP**.

eg. MSN, AOL

Intranet → Within an organisation. → Helps for training, leave access, tutorials, etc.

Extranet → Intranet + Access given to ppl outside organisation (extra clients < collaborating ppl >)

→ for client-side projects, ie, outsourcing.

eg. we access Bank of America's website's server.

MAJOR ISSUE : higher security required.

Website - Collection of web pages.

eg. Facebook (website), & our profiles are webpages.

IETF → Manages net

No one owns the internet → intangible entity (can't touch or feel)

Layers of Network

OSI Reference Model

Application: HTTP, FTP, VOIP are application layer protocol. client & server ke beech mein protocol hota hai

TCP/IP → connection protocol

Transport → Routes
Connection: TCP/IP

Network layer → which way/ path to choose

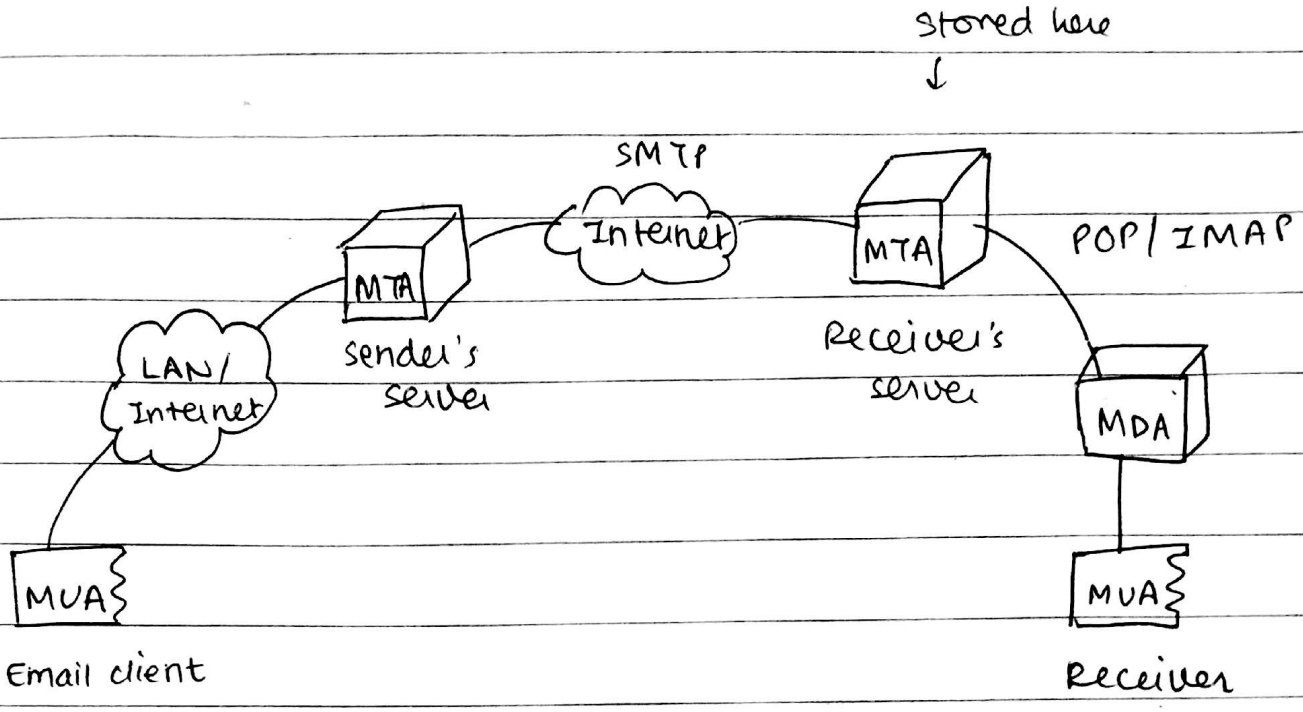
Physical → which cables/ fibres to use.

Data link → packet.

Services

(1) EMAIL :

Follows store-and-forward model



Mail User Agent w/ help of LAN/Internet sends to Mail Transfer Agent (sender's server)

SMTP used & POP/IMAP used → 2 protocols.

2nd MTA → Receiver's server

Mail Delivery Agent: jo deliver karega doosre ko.

POP : Post Office Protocol

IMAP : Instant Message Access Protocol → maintains a copy of mail on server using IMAP

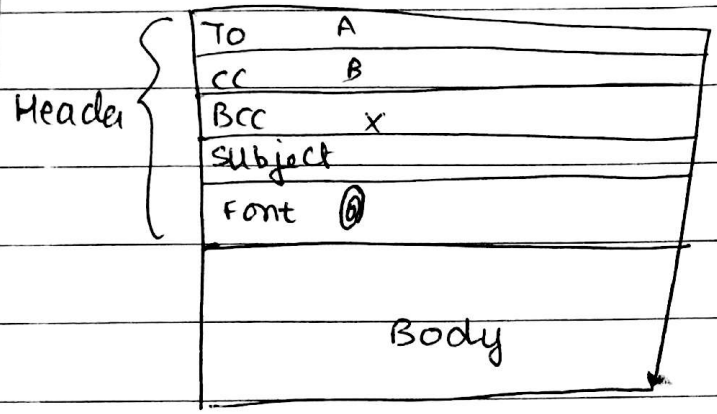
POP : Every msg when retrieved may not be stored.

(1) Email Client

(2) Web mail.

(1) → On Android/ iOS phones → the app

(2) → on a browser → using web address of email provider.



abc.def @ gmail.com

user @ domain

X can see A & B have been sent

A & B cant see X.

(2) FILE TRANSFER: Platform independent

Architecture → client - server, Protocol → FTP

Telnet / Usenet

Telnet: Permission to use someone else's identity or computer to access a resource. → Providing remote login instead of user logging in. <Protocol hai>
eg. We access library's catalogues.

Usenet: News groups & discussion forum. Global bulletin board.

Torrent → Peer to peer.

VOIP → Hybrid (Client-server AND peer to peer)
Client tells server I want to video chat. server agrees & connects 2 peers (2 client machines).

For a common man, internet would be useless w/o web. ∵ we use web for doing everything like web mail, etc. ∴ It is most important service provided by internet.

(3) HTTP (HyperText Transfer Protocol)

HTTP's characteristics to define it:

(a) Application layer protocol

(b) request - response protocol

(c) Stateless

Client - server architecture present.

NON-PERSISTENT HTML 1.0: ⇒ 2 response times + 1 transmission time.

STEP 1 (a) Client initiates a connection to the server

(b) Server accepts the connection.

STEP 2 Client initiates a request msg.

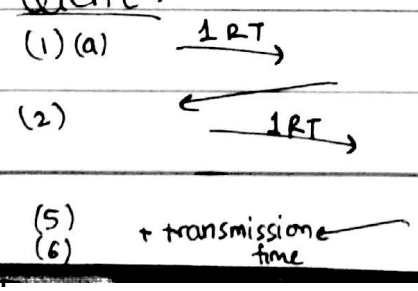
STEP 3 Server ~~reqtes~~ receives request. Forms a response msg.

STEP 4 ~~the~~ Server closes the connection.

STEP 5 Receives the response (html file)*

STEP 6 Steps 1-5 repeated as many times as no. of referenced objects.

* if html file has 10 imgs ⇒ 10 objects. ∴ steps 1-5 repeated 10x.



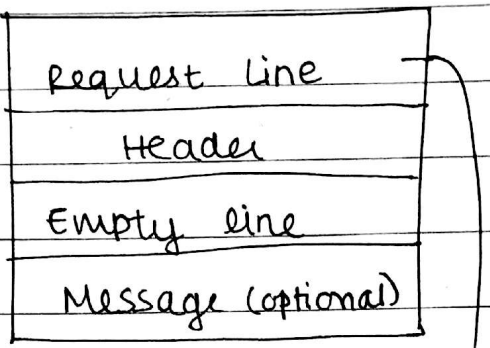
HTTP
- Non-persistent → HTTP 1.0 → 2rt + 1tt
- Persistent
  - without pipeline ① → 1rt + 1tt
  - with pipeline. → HTTP 1.1 ②

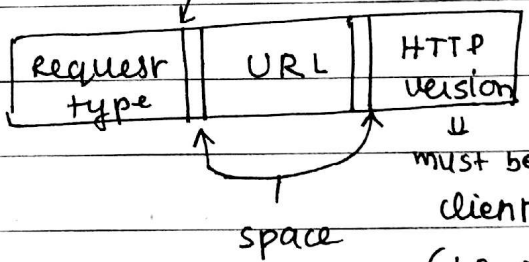
HTTP Messages
- Request
- response

① : sends request msgs continuously w/o checking if response is being sent ⇒ MIXING of msgs occurs. (issue).

② : Send next request only one prv response is received.

Request





must be same for client & server (1.0 or 1.1 ⇒ pers. or non-pers)

Methods / Commands:

GET, POST, DELETE, PUT.

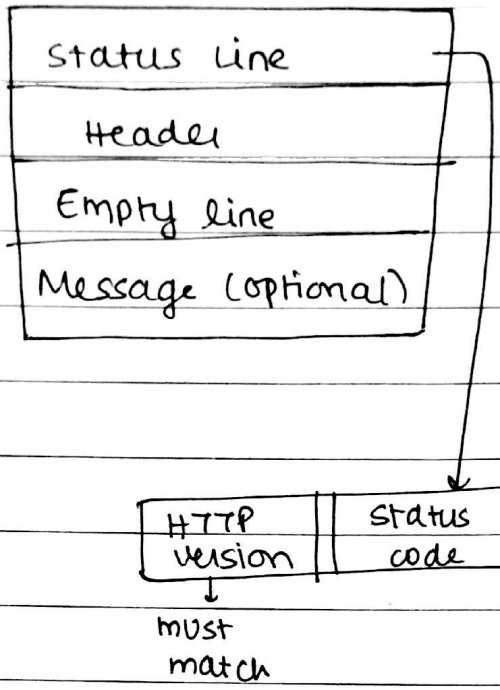
Example:

req. type | URL | HTTP version

GET / somedir / page.htm HTTP / 1.1

Header {
- Host: www.someschool.edu
- User-agent: Mozilla/4.0
- Connection: close

Response :



Status code: 1XX (One hundred series)

2XX (2 hundred series)

3XX

4XX → Bad request

5XX

1XX → Informational series (provisional responses)

100 → If half page loaded, tells it to CONTINUE

100 : CONTINUE

101 : SWITCHING

102 : PROCESSING

2XX → Success series

200 : OKAY → we dont see this ∵ it means website is displayed.

201 : CREATED

202 : ACCEPTED

Response msgs from server { 404 → Page not found
301 →

3xx → Redirection series.

301 → Moved permanently

4xx → Client Error

400 → Bad request

401 → Unauthorised

403 → forbidden

404 → Not found

408 → Request timeout

5xx → Server error

500 → Internal server error

502 → Bad gateway

504 → Gateway ~~xxx~~ ~~xxxx~~. timeout.

505 → HTTP version not supported.

1 marker → status code & its use.

Example: HTTP/1.1 200 OK

Header { connection: close
Date:
Server: Apache
Last modified:
content-length:
content-type:

Then msg.

Stateless : No state info of client stored in server
(of any previous request → not saved by server)

Cookies : Not softwares, cannot be programmed or add
Not malware & cant contain virus → Not corrupting system
can be used as spyware to keep track of browsing activity/history. ∴ safer to remove them.

Cookies → to save info in a stateless systems.

User used net for these purposes:
(1) Informational
(2) Navigational
(3) Transactional

TYPES OF COOKIES: (Based on above 3 purposes)
(1) Session cookies
(2) Persistent cookies → still exist after session.
(3) Secure cookies → banking → when redirected to other pg.
(4) Zombie cookies → recreated after user is gone.

server overloading → server Bottleneck.

Proxy server → web cache.

Web caching.

Cache Memory → High speed memory which was employed b/w main memory & CPU to compensate for speed difference b/w these 2. CPU → fast
MM → slow.

CPU → Main memory
↓
cache memory (beech ki speed)

Web caching → Small servers used to save some user activity. Proxy servers → Name of these servers.
↓
False server. Tells you falsely that it is a server

Web 1.0

↓

Classical IR → Web IR (SE) (search engine)

represented by

→ Quadruple [Q, D, F, $R_{q,d}$]

[Query, Documents which will be retrieved, Framework, Ranking mechanism to sort materials]

Database records (SQL) ← Queries	Documents (Web pages, things)

Web IR

Framework (models)

- set-theoretic
- Boolean
- vector space
- Probabilities

search Engine ≡ WebIR

SEARCH ENGINE ARCHITECTURE (SEA) : 2 modules -

(1) Indexing : → Text acquisition
→ Text transformation
→ Index creation

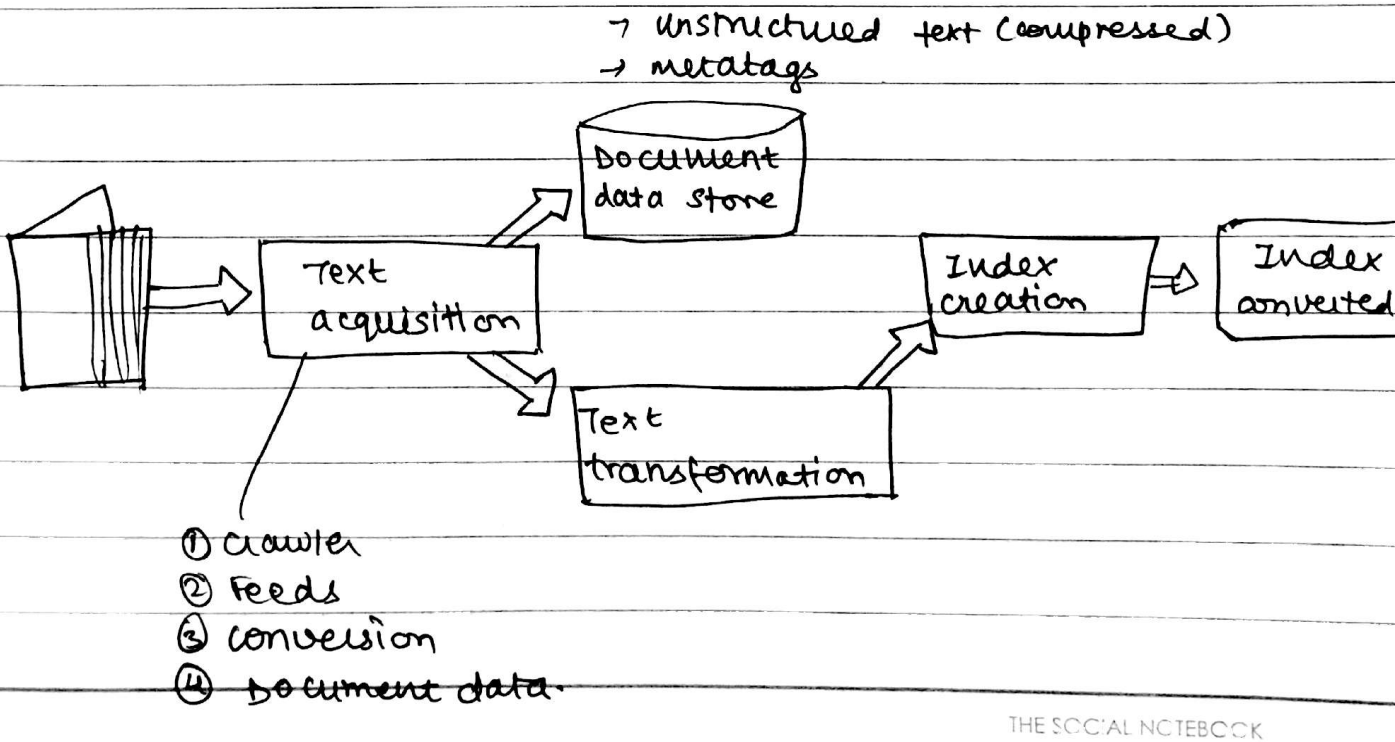
] OFFLINE PHASE

(2) **Querying:** → User Interaction
→ Ranking
→ Evaluation
] - ONLINE PHASE

term	Present in - DocID
abc	11, 3, 4, 41, 200, 5001
def	3, 2, 17, 400, ...
xyz	...

Each Doc has its own table

Doc No.	URL	Title	Snippet → which is seen.



Crawler - Program which helps to crawl the web graph to acquire the info.
Vertical crawler: topic related / wordical searches.

RSS feeds: Rich site summaries → realtime streaming of documents (like news / mail)

∴ Info retrieved through either crawlers / feeds.

Conversion - all formats have to be converted to consistent formats.

Text transformation: identifying imp. terms & features in query to help w/ indexing
eg. how to get admission in dtu ≡ dtu admissions
5 substeps in t.t → ① • Parsing
② • Stemming
③ • Stopping
④ • Link analysis
• Info extraction
⑤ • Classification. Classifier

Stemming: Natural language processing ~~processing~~ procedure (transforming text)

like fish — fishes
fisherman
stemmed

Stopping: removal of common words like punctuation or how, to, get, in, etc.

Info Extraction:

Name Entity Recognisers → identify nouns related to place, name, company, etc. To identify imp / indexed terms in query / documents.

Classifier: Pre-defined set of classes, to categorise the data.

Maintaining statistical data facilitate smooth ranking mechanism (online part)

Index creation: → Doc statistics → term freq (TF) calculated for keywords collected before.
→ Weighting → to store ranking using maintained statistics
→ Inversion
→ Index distribution.

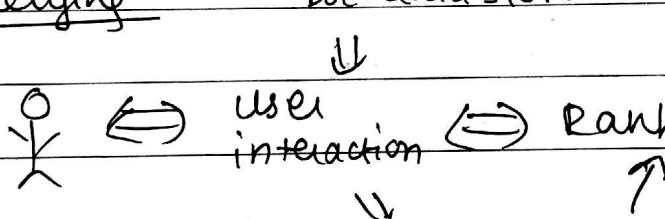
Inverse document frequency (IDF)
within the set of doc. what is freq of ~~term~~ term.

Google's pagerank works by counting how many links point to a particular webpage ⇒ rank it higher & show it on top.

TF: Term kitni baar aayi doc. mein.

IDF: Agar term zyada baar na aaye but page is overall more imp. so find an avg value.

Querying



input query } ∴ 2-way
output result

(1) input query
(2) query transformation — suggestions → DTU ⇒ (1) admis (2) Placements
— spell-check
— expands acronyms (DTU)
(3) Output result

Results < Paid results (ads on top)
Organic results

Recall, Precision } Evaluation's effectiveness measured in 2 ways

Recall → how many relevant docs retrieved from doc. set
Precision → how many are actually precisely/closely relevant from total relevant retrieved docs from doc set.

Efficiency measured in response time (google shows it too).

Pagerank → PageRank™

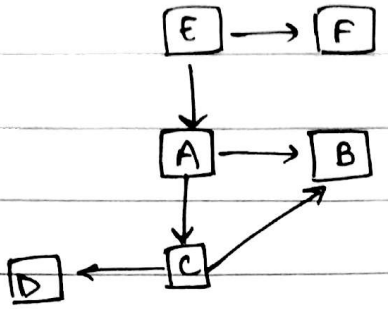
Made by Larry Page & Sergey Brin

Google sees web as a web graph & uses it as a RANDOM SURFER MODEL.

Pagerank → Based on probability distribution ⇒ probability of user clicking on a particular link.

* PageRank ranks webpages & not websites.
" works on in-links & out-links

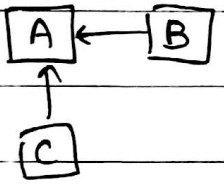
Each link that points to a page acts as a vote. More the pointing link, higher rank.



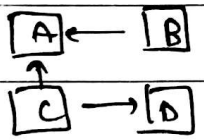
PageRank formula is :

$$PR(A) = (1-d) + d\left(PR\left(\frac{Ti}{Ci}\right) + \cdots + PR\left(\frac{Tn}{Cn}\right)\right)$$

d: damping factor , $d = 0.85$

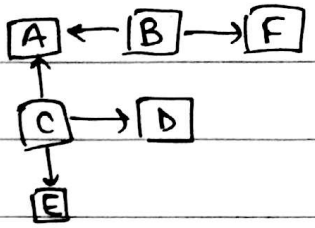


$\Rightarrow PR(A) = PR(B) + PR(C)$



$\Rightarrow PR'(A) = PR(B) + \frac{PR(C)}{2}$

$\because$ C points to 2 others $\Rightarrow$ vote share divided equally



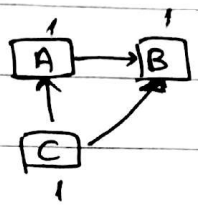
$\Rightarrow PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{3}$

damping factor was introduced $\because$ if no link points to a page, it should still be displayed

$PR = (1-0.85) + 0.85(0)$

$= 0.15$ ✓ $\Rightarrow$ seen.

Initial probability is taken as 1 for all.
∴ it keeps changing ∵ all are ~~co-dependent~~.



$$PR(A) = 1-0.85 + 0.85 \times \frac{PR(C)}{2}$$

$$PR(B) = 0.15 + 0.85\left(PR(A) + \frac{PR(C)}{2}\right)$$

$$PR(C) = 0.15$$

Google says it only uses link analysis & not content analysis, but it does actually happen ∵ keywords are shown ⇒ May be it occurs in backend.

Q1 Explain Web & Internet.

Q2 Explain Store & forward model of email.

Q3 Briefly describe search engine architecture.

Q4 ~~Differentiate b/w~~ Explain need of web caching.

Q5 Explain the following:

(a) URL (b) HTTP (c) HTTP 1.0