RESEARCH

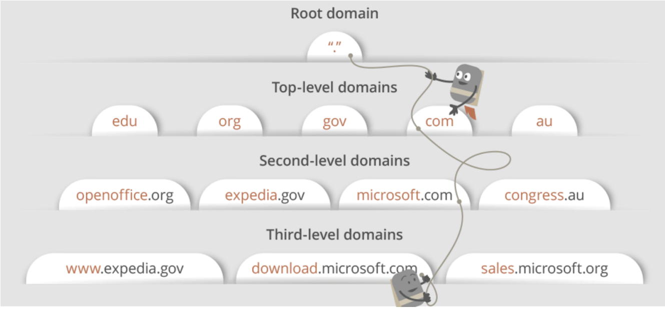
1. a. What is the entire cycle of events that follows when you type in the url of a webpage?

Inshort:

1. You enter a URL into a web browser
2. The browser looks up the IP address for the domain name via DNS
3. The browser sends a HTTP request to the server
4. The server sends back a HTTP response
5. The browser begins rendering the HTML
6. The browser sends requests for additional objects embedded in HTML (images, css, JavaScript) and repeats steps 3-5.
7. Once the page is loaded, the browser sends further async requests as needed.

In other words:

1. You type maps.google.com into the address bar of your browser.
2. The browser checks the cache for a DNS(Domain Name System) record to find the corresponding IP address of maps.google.com.
3. If the requested URL is not in the cache, ISP's DNS server initiates a DNS query to find the IP address of the server that hosts maps.google.com.



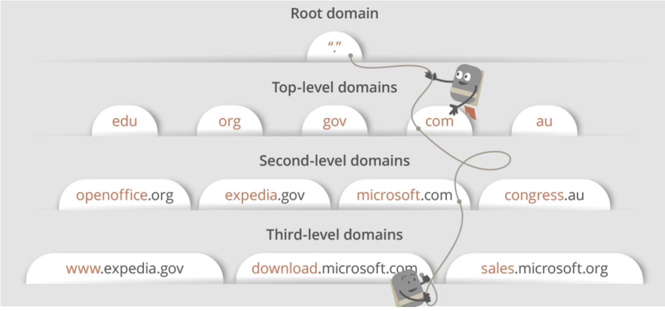
4. The browser initiates a TCP connection with the server.
5. The browser sends an HTTP request to the webserver.
6. The server handles the request and sends back a response.
7. The server sends out an HTTP response.
8. The browser displays the HTML content (for HTML responses, which is the most common).

1. b. How are urls mapped to websites hosted on a server in Bangalore?

I might have understood the question wrong.

For example maps.google.com, first, the DNS recursor will contact the root name server. The root name server will redirect it to the .com domain name server. .com name server will redirect it to the google.com name server. The google.com name server will find the matching IP address for maps.google.com in its DNS records and return it to your DNS recursor, which will send it back to your browser.

A URL consists of five parts -- the scheme, subdomain, top-level domain, second-level domain, and subdirectory.



1. c. Read up in brief about TCP and UDP, that enable your request to reach a particular server, hopping from one router to another.

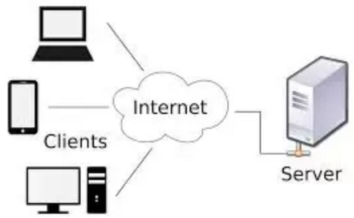
TCP is a connection-oriented protocol, which means a connection is established and maintained until the application programs at each end have finished exchanging messages. It determines how to break application data into packets that networks can deliver, sends packets to and accepts packets from the network layer, manages flow control and -- because it is meant to provide error-free data transmission -- handles retransmission of dropped or garbled packets as well as acknowledgement of all packets that arrive.

This process of error detection -- making retransmissions and reordering packets after they arrive -- can introduce latency in a TCP stream. Highly time-sensitive applications such as voice over IP (VoIP), streaming video and gaming generally rely on a transport process such as User Datagram Protocol (UDP) because it reduces latency and jitter -- variation in latency -- by not worrying about reordering packets or getting missing data retransmitted.

UDP is classified as a datagram protocol, or connectionless protocol, because it has no way of detecting whether or not both applications have finished their back-and-forth communication. Instead of correcting invalid data packets, as TCP does, UDP simply discards those packets and defers to the application layer for more detailed error detection.

1. d. What does setting up a server even mean?

A server is a computer that serves many kinds of information to user or client machines. Usually a server will only do a few things for many clients. Every type of thing a server does is called a service. Services are used by other computers that are called clients. The relationship between client and server is called a client-server relationship.



A web server is a type of server that is used to host websites. Examples of web server software include Apache or IIS. A web server can host one, or many, websites. The default port for a web server to listen to is port 80 (HTTP) or 443 (HTTPS).

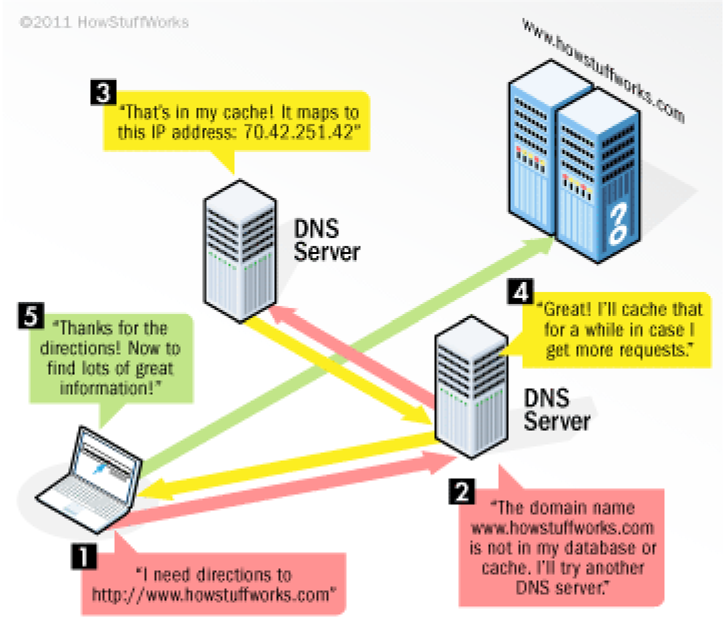
Some web servers do other things than just serving a web page. For example, they may have something called SSI that makes building a website easier.

Web servers use services like CGI to let software on the server make web pages. Some of the programming languages that can use CGI are scripting languages like Perl, Python, PHP, or ASP.

1. e. Get an overall idea of how a server resolves a request? [Google Apache and NginX, and see what role they play].

https://computer.howstuffworks.com/dns.htm
https://flaviocopes.com/http-request/
https://vanseodesign.com/web-design/browser-requests/

Read these three pages in this particular order with the last one answering the asked question. These though don't have reference of NginX or Google Apache...


©2011 HowStuffWorks
www.howstuffworks.com
3
"That's in my cache! It maps to this IP address: 70.42.251.42"
DNS Server
4
"Great! I'll cache that for a while in case I get more requests."
5
"Thanks for the directions! Now to find lots of great information!"
DNS Server
2
"The domain name www.howstuffworks.com is not in my database or cache. I'll try another DNS server."
1
"I need directions to http://www.howstuffworks.com"

1. f. Suppose you built a html page for the first time [you should if you haven’t already :P], and you are very enthusiastic to show it to your friend on his device. How would you do this if you are not allowed to share your code with him or upload your website. [Every laptop is a server. There’s actually a place like 127.0.0.1)

If it is a static website

1. Open Terminal
2. cd into the root directory of the website.
3. Run , python -m simpleHTTPServer 8000

This will start a web server on localhost:8000

Now any device in the same network can access the website by using <ip-address>:8000 in their browser. Eg: 192.168.4.15:8000

Using MAMP should have worked on any computer on the same network using http://ipaddresofserver/Directory

Eg : http://192.168.4.15/mywebsite

OR

(Preferably)

On an internal network you need to set the system hosting the web server to a fixed ip address. So if your network was 192.168.0.1 to 192.168.0.254 with your router on 192.168.0.254 I would stick the web server on 192.168.0.10. Install MAMP or WAMP or Bitnami on your system. Can you open the homepage on that machine at 127.0.0.1?

Yes - good continue.

No - the webserver is not running properly on this system, fix it here first.

If yes, Can you open the homepage on another computer on your network using 192.168.0.10?

Yes - Job done.

No - Can you ping 192.168.0.10?

If you can ping but cannot open 192.168.0.10 in a browser, it could just be the firewall on the device blocking port 80 so check out the error message post any more clues up here.

No - You have network issues.

2 and 3 : got the basic idea/principle behind the Django framework.

THINK

1. Suppose you want to now design the database schema of Book-ed!. This should include the name of the book, author, isbn number, short_description, and any other field you deem necessary. Similarly you'd have to design the scheme for the users and define the relationship between a book(s) and a user(s). Assume all the data that you need is already available.

Database schema of Book-ed!

- Class Book:
    - Book ID: Number
    - Name: String
    - Author: String
    - ISBN number: Number
    - Short Description: String
    - Genre: String
    - Edition: String
    - Publication: String
    - Year of publishing: Number
    - Lent to: User ID
- Class User:
    - User ID: Number
    - Full Name: String
    - Nickname: String (used as username)
    - Address: String
    - Coordinates: 2-D map point (from GPS)
    - Interest: Array of string
    - Email: String
    - User rating: Number
    - Owned books: Array of Book ID
- Relationship:
    - Multiplicity relation between User and Book, i.e. User may be related to zero to many Book(s).

I am not good in Database Schema drawing. But my intention is to have User know the Book ID of the books he owns, and the book remembering, if lent, lent to whom...

I can explain better than writing database schema…

My vocabulary with respect to terms of Database and relationships isn't good.

2. You want to introduce the feature that the books are searchable by the author? What will be the best way to accommodate these. Keep in mind there is additional load on the server when you are querying the tables to get the information.

a. You want to minimize the number of queries. The simplest way is to fetch all the books and match the author for each of them. Can you do any better?

From what I have understood, making another table relating author to book and using this table for searching books by author.
Add author ID as a field in class Book, and whenever a new book object is created, associate the book to its author using author ID in the above described table.
So, every time someone searches for a book by the author, the server doesn't need to fetch all books...

References:
https://wsvincent.com/what-happens-when-url/
https://medium.com/@maneesha.wijesinghe1/what-happens-when-you-type-an-url-in-the-browser-and-press-enter-bb0aa2449c1a
https://simple.wikipedia.org/wiki/Server
https://www.wired.com/2010/02/set-up-a-home-server/
https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works
https://computer.howstuffworks.com/dns.htm
https://flaviocopes.com/http-request/
https://vanseodesign.com/web-design/browser-requests/
https://www.quora.com/How-do-you-host-a-website-locally-in-your-personal-network
https://creately.com/blog/diagrams/class-diagram-relationships/
https://en.wikipedia.org/wiki/Data_modeling
https://www.nginx.com/resources/glossary/nginx/
https://kinsta.com/knowledgebase/what-is-apache/

Devansh Jain
190100044