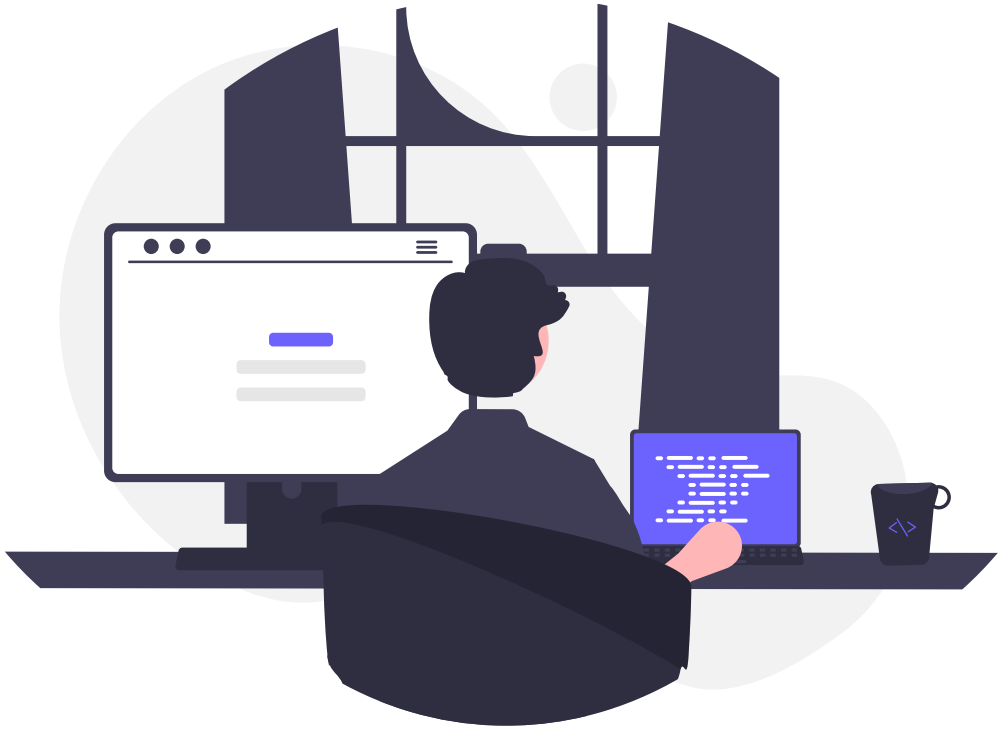
Web Fundamentals

CSCI 11052



Tharuka Vishwajith
M.Sc. in Big Data Analytics
B.Sc. (Hons) in Software Engineering

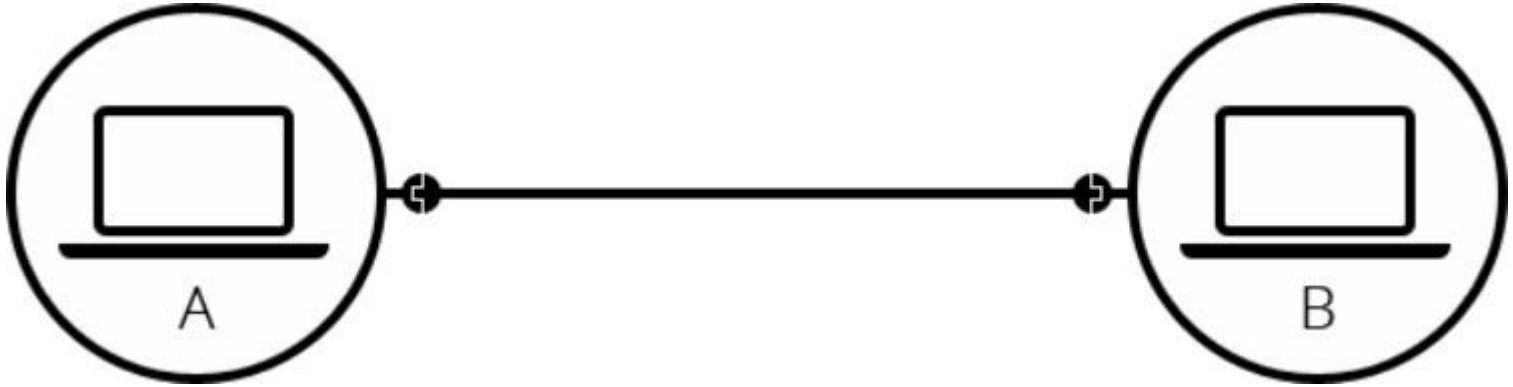
Course Overview

- Overview of the Internet
- Internet Technologies
- Overview of Web Services
- Introduction to web programming
- HTML
- CSS
- Bootstrap

Overview of the Internet

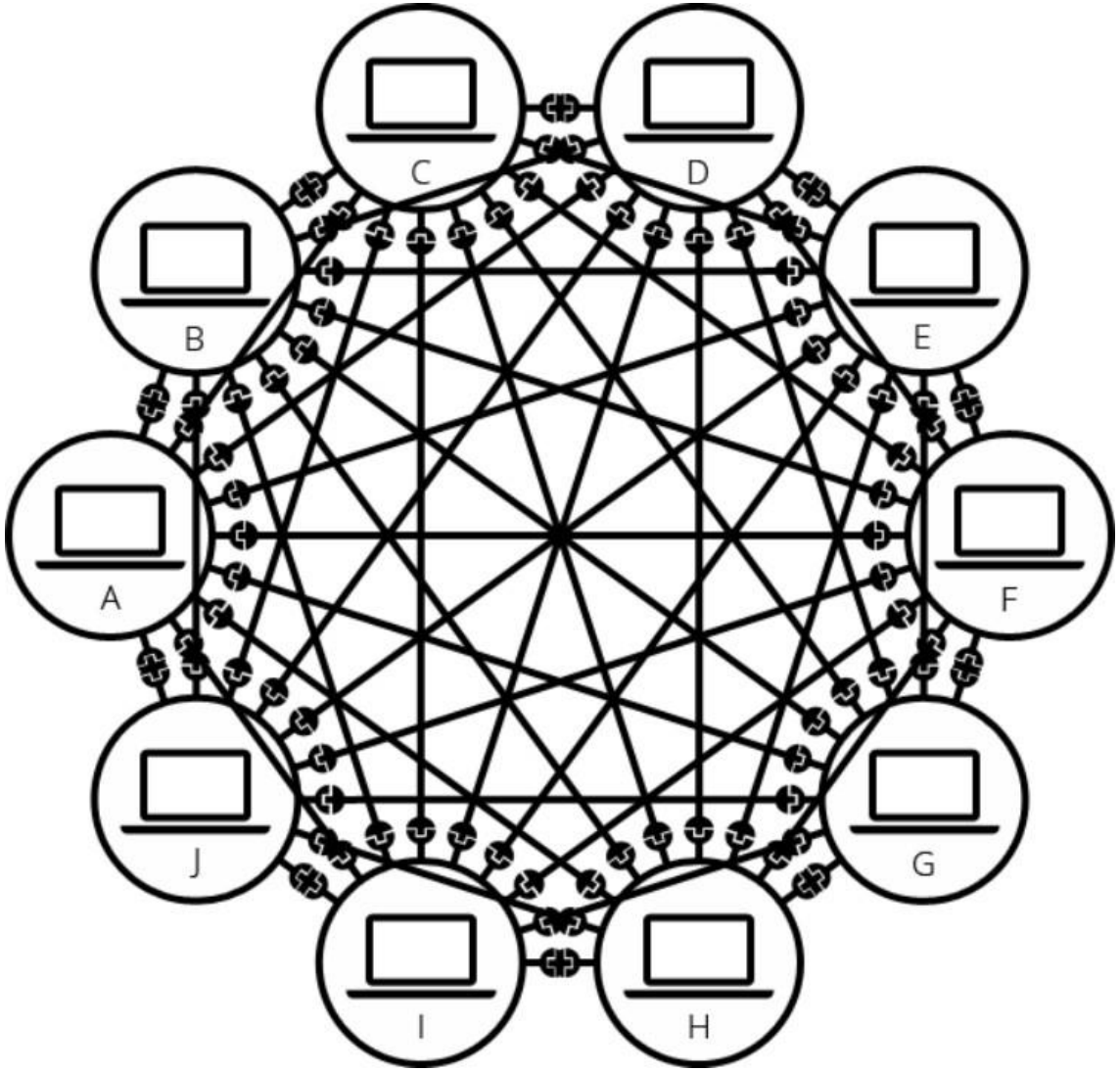
A simple network

- When two computers need to communicate, you have to link them, either physically (usually with an Ethernet cable) or wirelessly (for example with WiFi or Bluetooth systems).



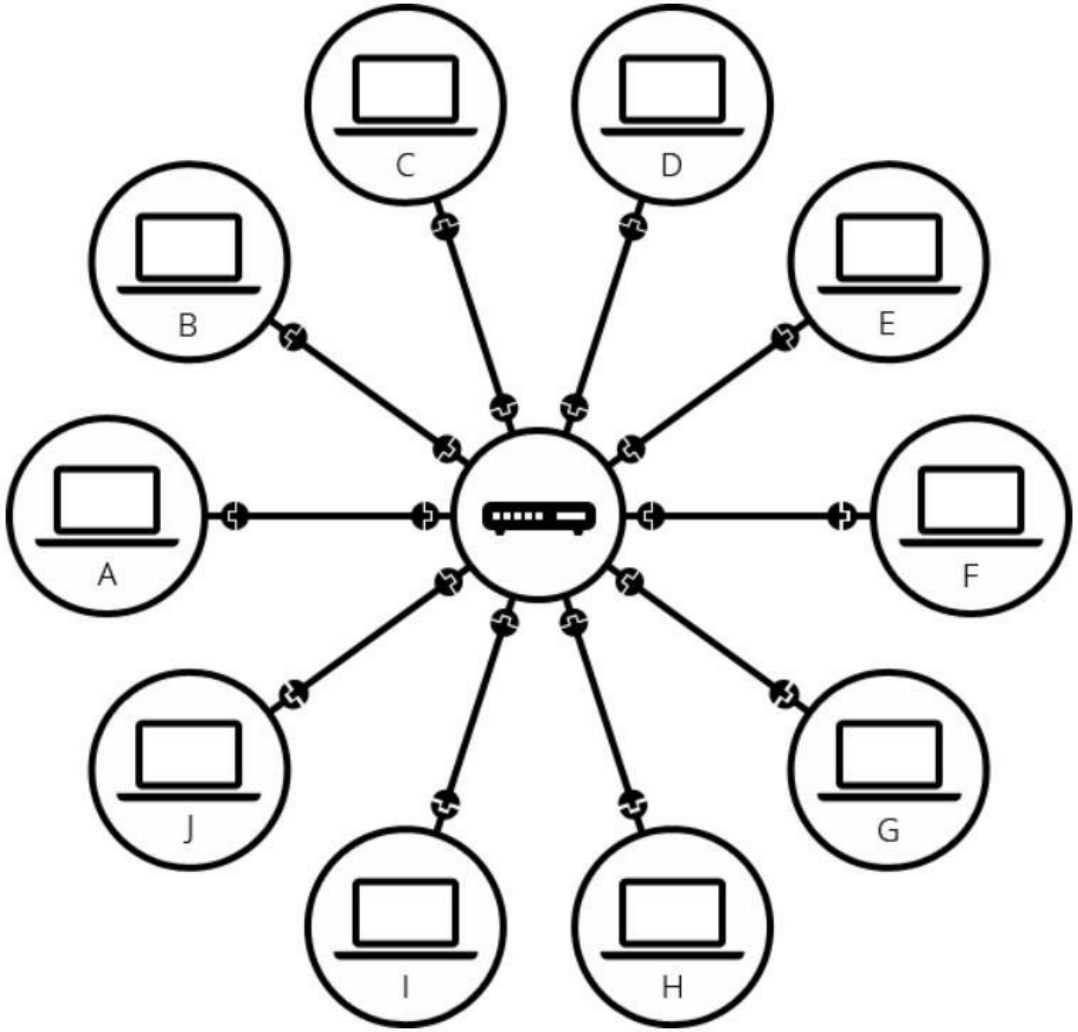
A simple network

- Such a network is not limited to two computers. You can connect as many computers as you wish. But it gets complicated quickly. If you're trying to connect, say, ten computers, you need 45 cables, with nine plugs per computer!



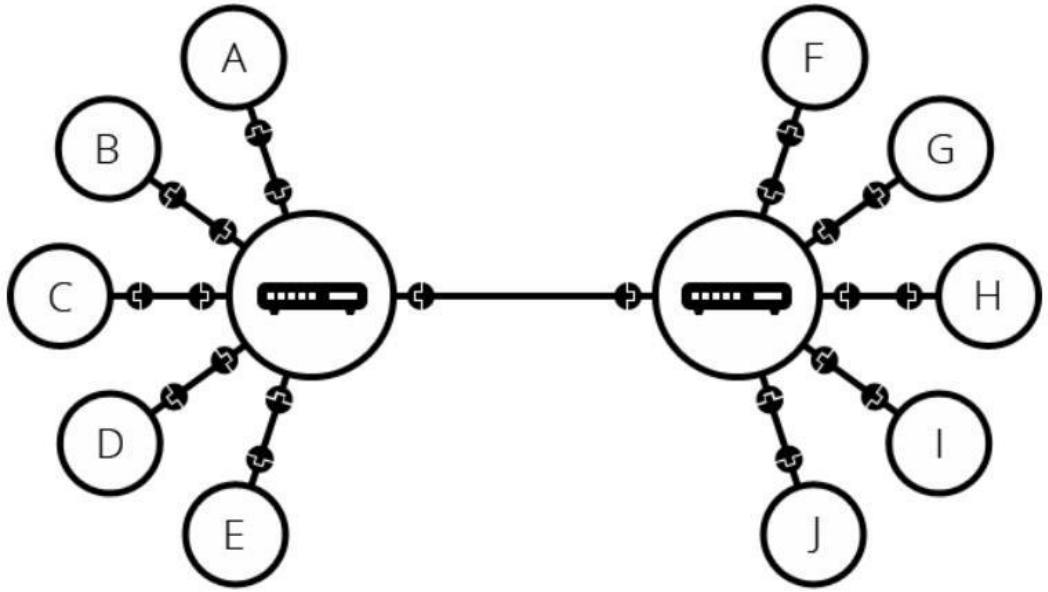
A simple network

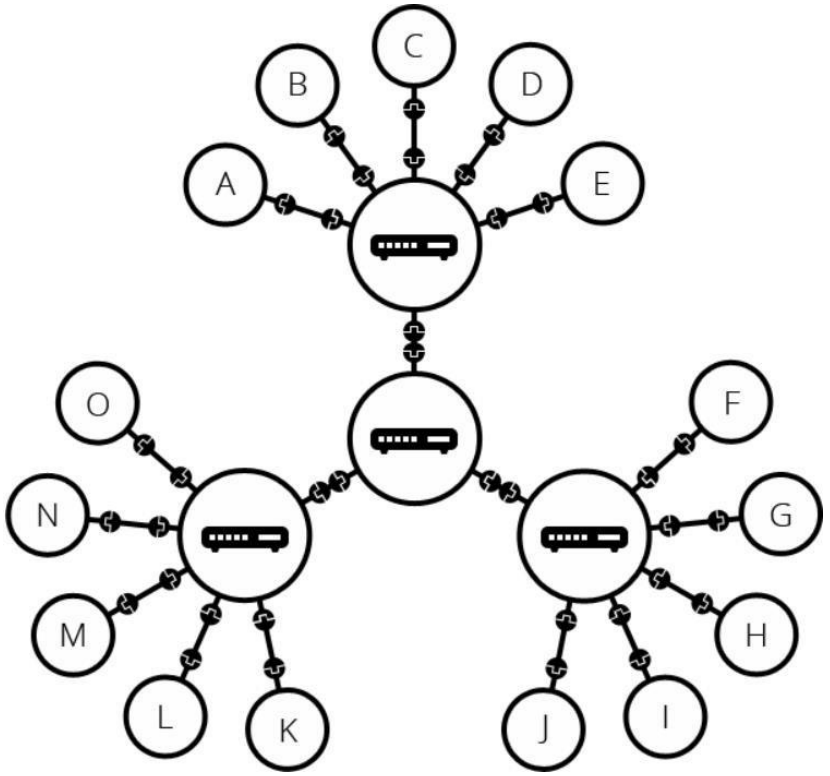
- Each computer on a network is connected to a special tiny computer called a *router*.
- This *router*: makes sure that a message sent from a given computer arrives at the right destination computer.
- To send a message to computer B, computer A must send the message to the router, which in turn forwards the message to computer B and makes sure the message is not delivered to computer C.
- Once we add a router to the system, our network of 10 computers only requires 10 cables: a single plug for each computer and a router with 10 plugs.



A network of networks

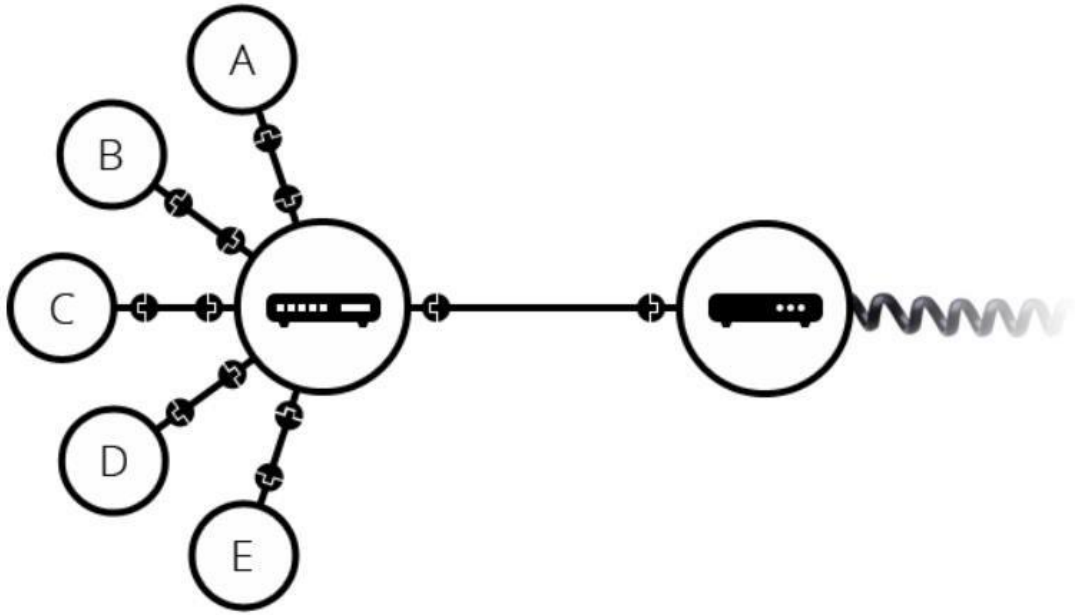
- By connecting computers to routers, then routers to routers, we are able to scale infinitely.





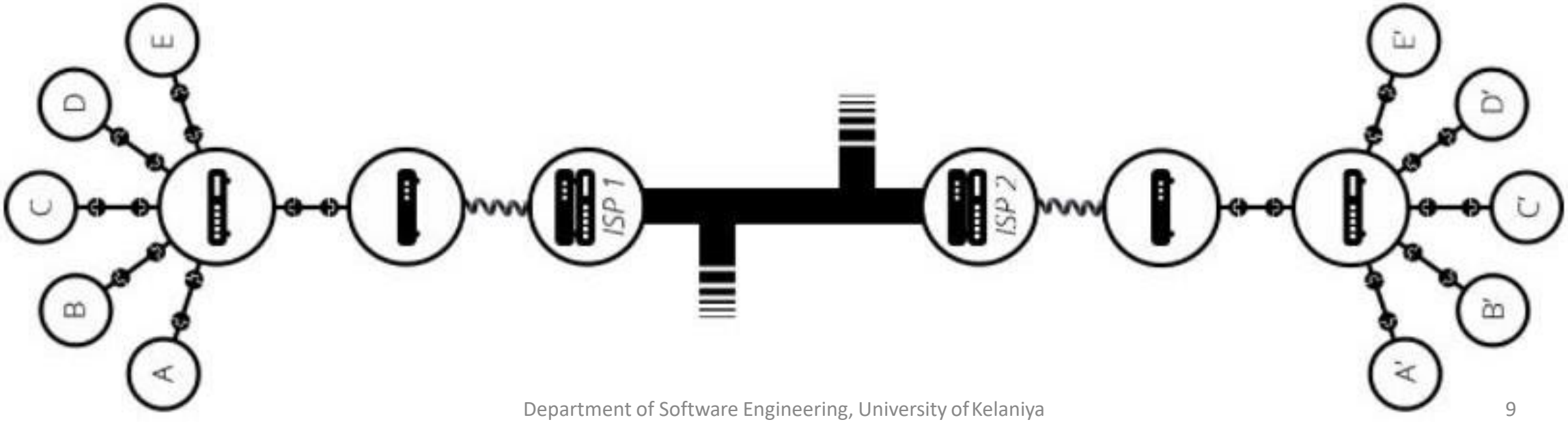
A network of networks

- The telephone infrastructure already connects your house with anyone in the world.
- To connect our network to the telephone infrastructure, we need a special piece of equipment called a *modem*.
- This *modem* turns the information from our network into information manageable by the telephone infrastructure and vice versa.



Internet Service Provider (ISP)

- So we are connected to the telephone infrastructure.
- The next step is to send the messages from our network to the network we want to reach.
- To do that, we will connect our network to an Internet Service Provider (ISP).
- An ISP is a company that manages some special *routers* that are all linked together and can also access other ISPs' routers. So the message from our network is carried through the network of ISP networks to the destination network.
- The Internet consists of this whole infrastructure of networks.



Finding computers

- Any computer linked to a network has a unique address that identifies it, called an "IP address" (where IP stands for Internet Protocol).
- It's an address made of a series of four numbers separated by dots, for example: 192.168.2.10.

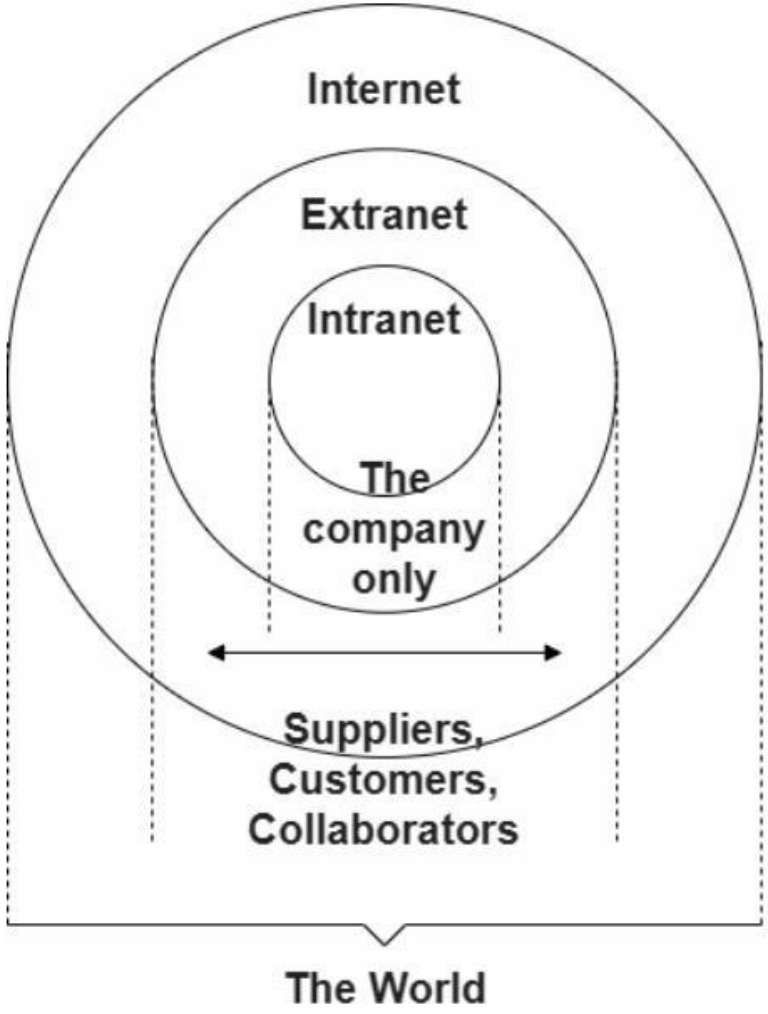
- To make things easier, we can alias an IP address with a human readable name called a domain name.
- Using the domain name is the easiest way for us to reach a computer over the Internet.

Internet and the web

- When we browse the Web with a Web browser, we usually use the domain name to reach a website.
- Does that mean the Internet and the Web are the same thing?
- It's not that simple.
- The Internet is a technical infrastructure which allows billions of computers to be connected all together.
- Among those computers, some computers (called *Web servers*) can send messages intelligible to web browsers.
- The *Internet* is an infrastructure, whereas the *Web* is a service built on top of the infrastructure.

Intranets and Extranets

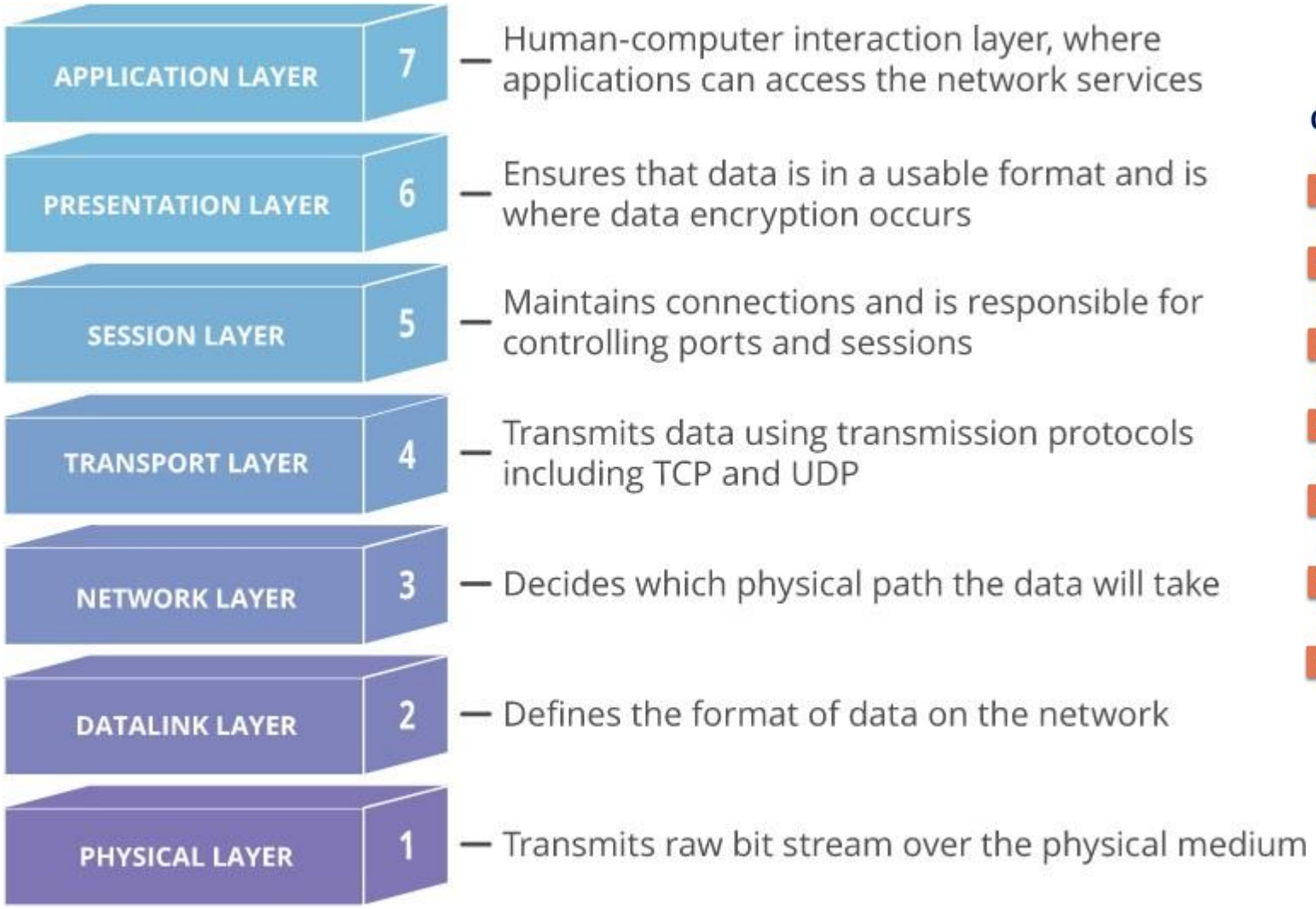
- Intranets are *private* networks that are restricted to members of a particular organization.
- Extranets are very similar to Intranets, except they open all or part of a private network to allow sharing and collaboration with other organizations.
- Both intranets and extranets run on the same kind of infrastructure as the Internet, and use the same protocols. T
- hey can therefore be accessed by authorized members from different physical locations.

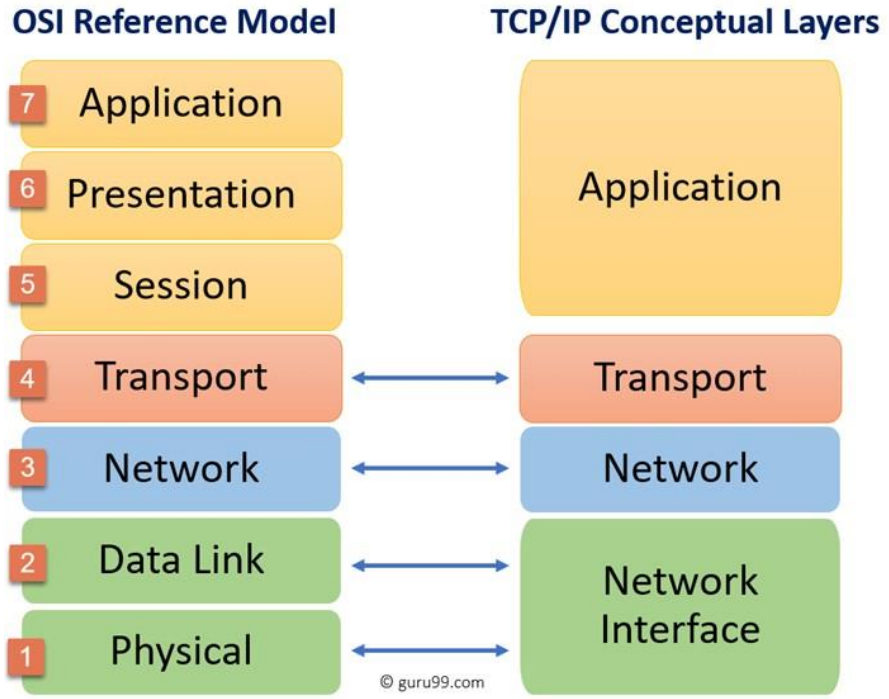


What is a Network Protocol

- Network protocols are a set of rules, conventions, and data structures that dictate how devices exchange data across networks.
- In other words, network protocols can be equated to languages that two devices must understand for seamless communication of information, regardless of their infrastructure and design disparities.
- A protocol provides a standardized way of doing certain actions and formatting data so that two or more devices are able to communicate with and understand each other.

OSI model and **Internet protocol stack**





Some protocols

- **TCP:** TCP is a transport layer protocol that ensures reliable data delivery. TCP is meant to be used with IP, and the two protocols are often referenced together as TCP/IP.
- **HTTP:** The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, the Internet that most users interact with. It is used for transferring data between devices. HTTP belongs to the application layer, because it puts data into a format that applications (e.g. a browser) can use directly, without further interpretation. The lower layers of the OSI model are handled by a computer's operating system, not applications.
- **HTTPS:** The problem with HTTP is that it is not encrypted — any attacker who intercepts an HTTP message can read it. HTTPS (HTTP Secure) corrects this by encrypting HTTP messages.
- **TLS/SSL:** Transport Layer Security (TLS) is the protocol HTTPS uses for encryption. TLS used to be called Secure Sockets Layer (SSL).
- **UDP:** The User Datagram Protocol (UDP) is a faster but less reliable alternative to TCP at the transport layer. It is often used in services like video streaming and gaming, where fast data delivery is paramount.

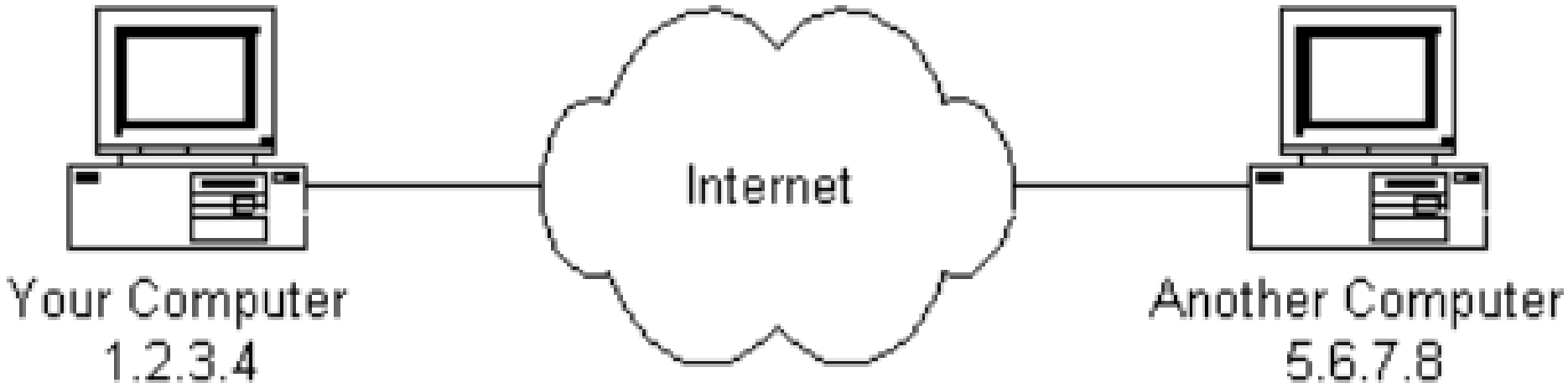
Transport Layer:

- The protocol governing the transport layer is Transmission Control Protocol (TCP).
- This layer controls the transmission of a file (say) between two hosts.
- At the client end TCP sends a request to a server for a connection.
- If the server responds affirmatively a socket connection is established.
- The TCP segments the file into packets, stamps each packet with a sequence number and a checksum.
- The checksum is the count of the bits in the packet.
- The packets are then sent to IP for internetwork routing.
- As the IP receives the packets they are sent to the TCP for checking.
- The client then sends an acknowledgment to the server as to the result.

Application Layer:

- Some examples:
- Telnet - log to a remote computer
- FTP (File Transfer Protocol) - log to a remote machine to transfer files
- SMTP (Simple Mail Transfer Protocol) - transport e-mail among mail servers.
  - POP (Post Office Protocol) - copies messages from mail server to client machines.
  - IMAP (Internet Mail Access Protocol) - copies only the message headers. When the user selects a message IMAP transfers the body of the message from the server.
- HTTP (Hyper Text Transfer Protocol)- transfers web pages from the server to the client.

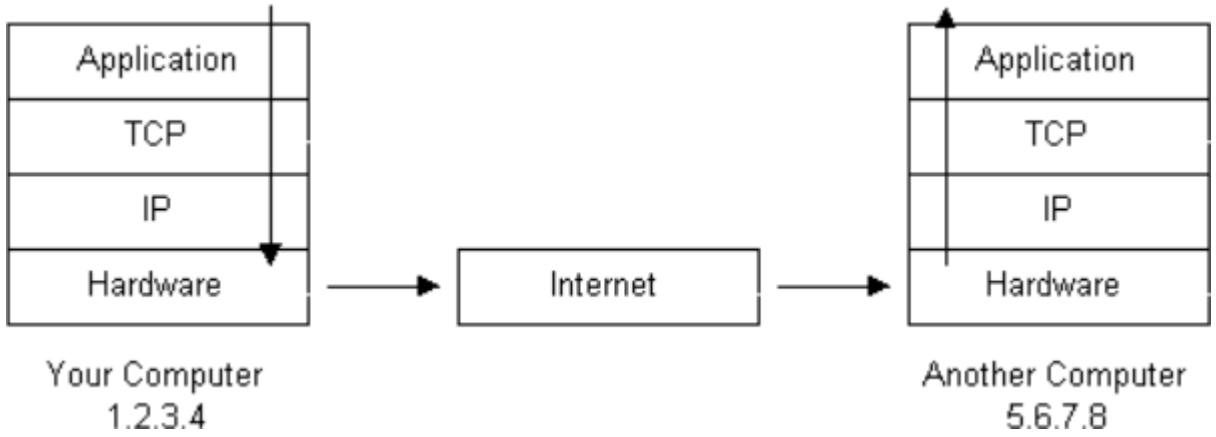
Communicate over the internet



- Internet is a global network of computers each computer connected to the Internet **must** have a unique address.
- Internet addresses are in the form **nnn.nnn.nnn.nnn** where nnn must be a number from 0 - 255.
- This address is known as an IP address. (IP stands for Internet Protocol

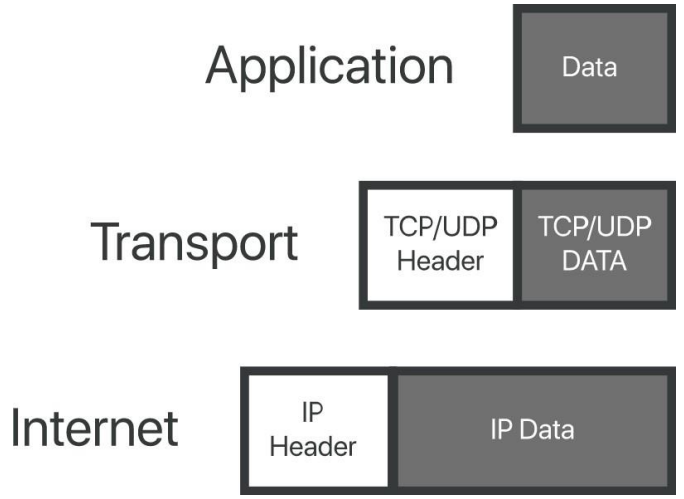
Communicate over the internet

Protocol Layer	Comments
Application Protocols Layer	Protocols specific to applications such as WWW, e-mail, FTP, etc.
Transmission Control Protocol Layer	TCP directs packets to a specific application on a computer using a port number.
Internet Protocol Layer	IP directs packets to a specific computer using an IP address.
Hardware Layer	Converts binary packet data to network signals and back. (E.g. ethernet network card, modem for phone lines, etc.)



IP - Internet Protocol

- The Internet Protocol (IP) is a protocol, or set of rules, for routing and addressing packets of data.
- Data traversing the Internet is divided into smaller pieces, called packets.
- IP information is attached to each packet, and this information helps routers to send packets to the right place.
- Every device or domain that connects to the Internet is assigned an IP address
- Protocols attach packet headers at different layers.



What is TCP/IP?

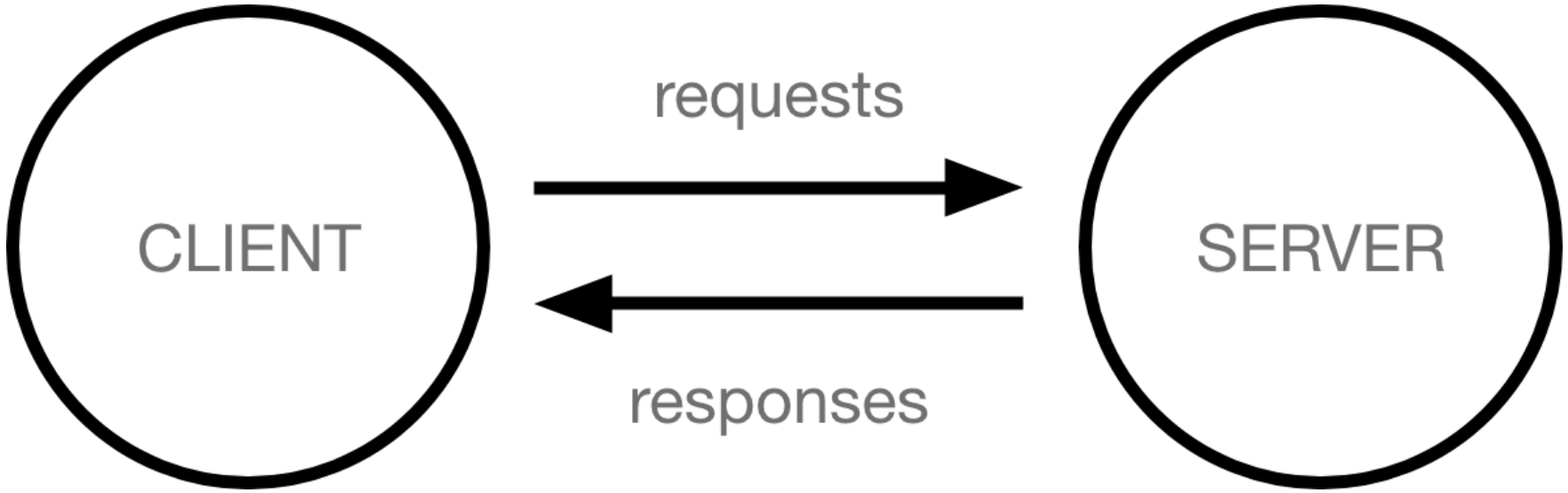
- The Transmission Control Protocol (TCP) is a transport protocol - it dictates the way data is sent and received.
- A TCP header is included in the data portion of each packet that uses TCP/IP.
- TCP ensures that all packets arrive in order once transmission begins.
- Via TCP, the recipient will acknowledge receiving each packet that arrives. Missing packets will be sent again if receipt is not acknowledged.
- Because TCP has to make sure all packets arrive in order, loading data via TCP/IP can take longer if some packets are missing.
- TCP and IP were originally designed to be used together, and these are often referred to as the TCP/IP suite.
- However, other transport protocols can be used with IP.

Client-server architecture

- Architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer).
- Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns.
- Servers wait for requests to arrive from clients and then respond to them.
- Server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service.
- Clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines.
- This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform.

How the Web works

Clients and servers : Computers connected to the web are called **clients** and **servers**. A simplified diagram of how they interact might look like this:



Source: https://developer.mozilla.org/

How the Web works, Cont.

•**Your internet connection**: Allows you to send and receive data on the web.

•**TCP/IP**: Transmission Control Protocol and Internet Protocol are communication protocols that define how data should travel across the internet.

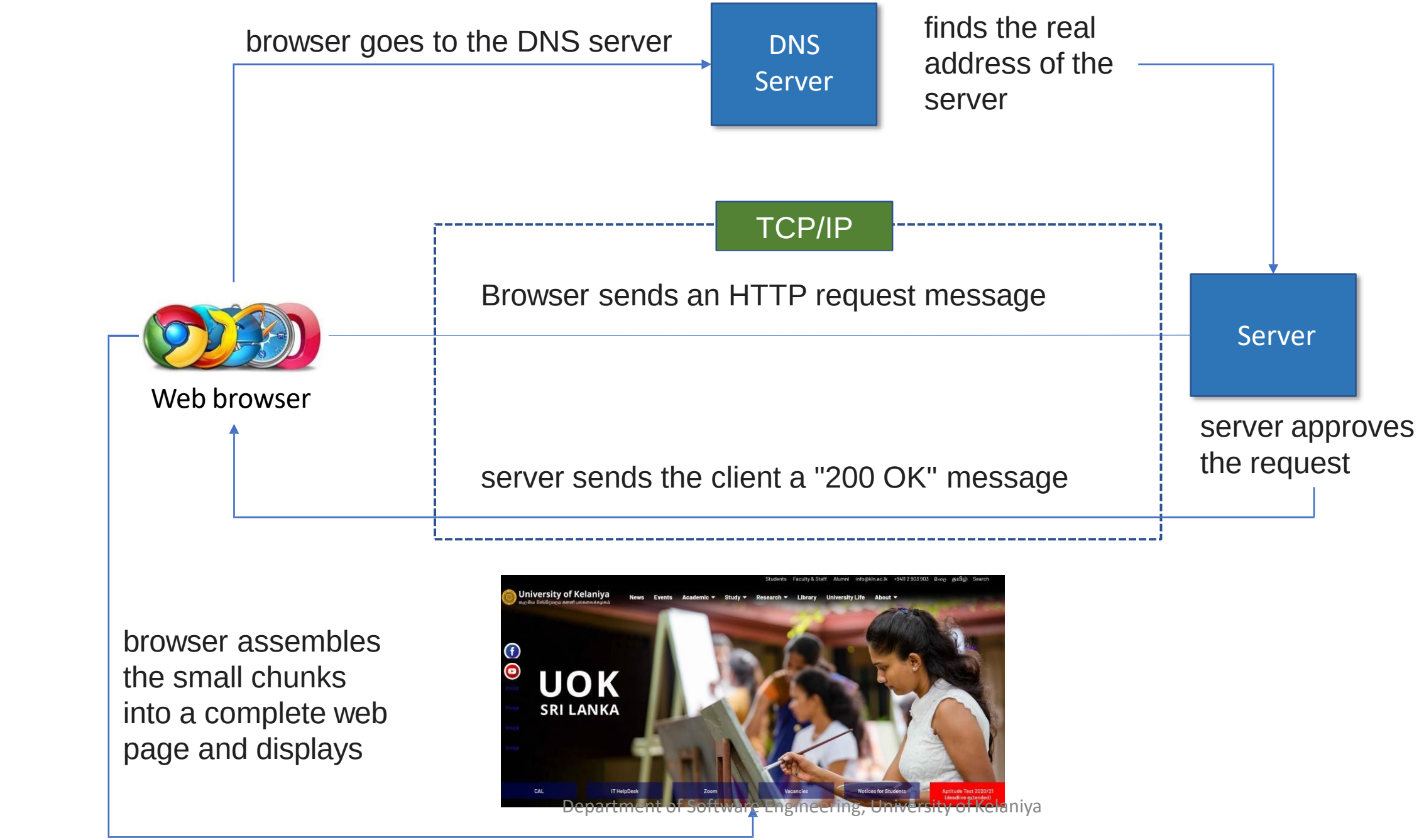
•**DNS**: Domain Name Servers are like an address book for websites. When you type a web address in your browser, the browser looks at the DNS to find the website's real address before it can retrieve the website. The browser needs to find out which server the website lives on, so it can send HTTP messages to the right place.

How the Web works, Cont.

•**HTTP**: Hypertext Transfer Protocol is an application protocol that defines a language for clients and servers to speak to each other.

•**Component files**: A website is made up of many different files. These files come in two main types:

- **Code files**: Websites are built primarily from HTML, CSS, and JavaScript, etc.
- **Assets**: This is a collective name for all the other stuff that makes up a website, such as images, music, video, Word documents, and PDFs.


browser goes to the DNS server
DNS Server
finds the real address of the server
TCP/IP
Browser sends an HTTP request message
Server
Web browser
server approves the request
server sends the client a "200 OK" message
browser assembles the small chunks into a complete web page and displays
University of Kelaniya
UOK
SRI LANKA

DNS explained

- Real web addresses aren't the nice, memorable strings you type into your address bar to find your favorite websites. They are special numbers that look like this: 63.245.215.20.

- This is called an IP address, and it represents a unique location on the web. However, it's not very easy to remember, is it? That's why Domain Name Servers were invented. These are special servers that match up a web address you type into your browser (like "mozilla.org") to the website's real (IP) address.

- Websites can be reached directly via their IP addresses. You can find the IP address of a website by typing its domain into a tool like IP Checker.

Source: https://developer.mozilla.org/

Packets explained

- When data is sent across the web, it is sent in thousands of small chunks.
- There are multiple reasons why data is sent in small packets.
- They are sometimes dropped or corrupted, and it's easier to replace small chunks when this happens.
- Additionally, the packets can be routed along different paths, making the exchange faster and allowing many different users to download the same website at the same time.
- If each website was sent as a single big chunk, only one user could download it at a time, which obviously would make the web very inefficient and not much fun to use.

What is a Domain Name?

- Domain names are a key part of the Internet infrastructure.
- They provide a human-readable address for any web server available on the Internet.
- Any Internet-connected computer can be reached through a public IP address, either an IPv4 address (e.g. 173.194.121.32) or an IPv6 address (e.g., 2027:0da8:8b73:0000:0000:8a2e:0370:1337).
- Computers can handle such addresses easily, but people have a hard time finding out who's running the server or what service the website offers. IP addresses are hard to remember and might change over time.
- To solve all those problems we use human-readable addresses called domain names.

Structure of domain names

- TLD (Top-Level Domain).
  - TLDs tell users the general purpose of the service behind the domain name.
  - The most generic TLDs (.com, .org, .net) don't require web services to meet any particular criteria, but some TLDs enforce stricter policies so it is clearer what their purpose is.
    - example:Local TLDs such as .us, .fr, or .se can require the service to be provided in a given language or hosted in a certain country — they are supposed to indicate a resource in a particular language or country.
- TLDs containing .gov are only allowed to be used by government departments.
- The .edu TLD is only for use by educational and academic institutions.
- TLDs can contain special as well as latin characters. A TLD's maximum length is 63 characters, although most are around 2–3.

Structure of domain names

- The **labels** are what follow the TLD.
- A label is a case-insensitive character sequence anywhere from one to sixty-three characters in length, containing only the letters A through Z, digits 0 through 9, and the - character (which may not be the first or last character in the label).
  - E.g. a, 97, and hello-strange-person-16-how-are-you are all examples of valid labels.
- The label located right before the TLD is also called a Secondary Level Domain (SLD).
- A domain name can have many labels (or components).
- You can create "subdomains" with different content located at each, like developer.mozilla.org, iot.mozilla.org, or wiki.developer.mozilla.org.

Buying a domain name

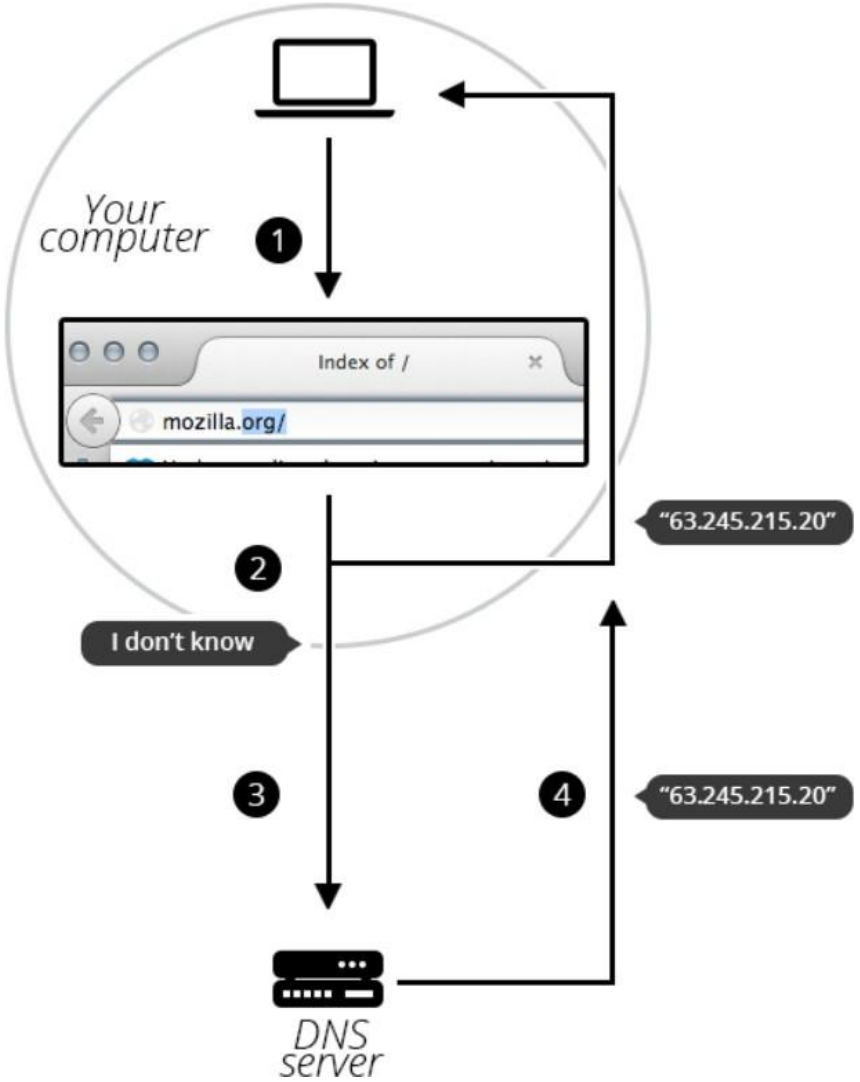
- Who owns a domain name?
- You cannot “buy a domain name”.
  - If every domain name was bought, the web would quickly fill up with unused domain names that were locked and couldn't be used by anyone.
- Instead, you pay for the right to use a domain name for one or more years. You can renew your right, and your renewal has priority over other people's applications.
- Companies called *registrars* use domain name registries to keep track of technical and administrative information connecting you to your domain name.

Getting a domain name

1. Go to a registrar's website.
2. Check the availability of your desired domain name.
3. If the domain name is available you can purchase it for a particular time period (It can be renewed when the time limit expires.).
4. There will be a place to “Get a domain name” - Click on it.
5. Fill out the form with all required details. Make sure not to misspelled. Once it's paid for, it's too late!
6. The registrar will let you know when the domain name is properly registered. Within a few hours, all DNS servers will have received your DNS information.

How does a DNS request work - Example

1. Type the domain name in your browser's location bar.
2. Your browser asks your computer if it already recognizes the IP address identified by this domain name (using a local DNS cache). If it does, the name is translated to the IP address and the browser negotiates contents with the web server. (End of story).
3. If your computer does not know which IP is behind the mozilla.org name, it goes on to ask a DNS server, whose job is precisely to tell your computer which IP address matches each registered domain name.
4. Now that the computer knows the requested IP address, your browser can negotiate contents with the web server.



HTTP and HTML

- HTML is a Language while HTTP is a Protocol

- **HTTP – Hypertext Transfer Protocol**
- HTTP is a protocol for transferring the hypertext pages from Web Server to Web Browser. For exchanging web pages between Server and Browser, an HTTP session is setup using protocol methods (e.g. GET, POST etc.). This would be explained in another post.

- **HTML** - Hypertext Markup Language
- HTML is a language for marking the normal text so that it gets converted into hypertext.
- HTML tags (e.g. “<head>”, “<body>” etc.) are used to *tag* or *mark* normal text so that it becomes hypertext and several hypertext pages can be interlinked with each other resulting in the Web.

References :

https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm

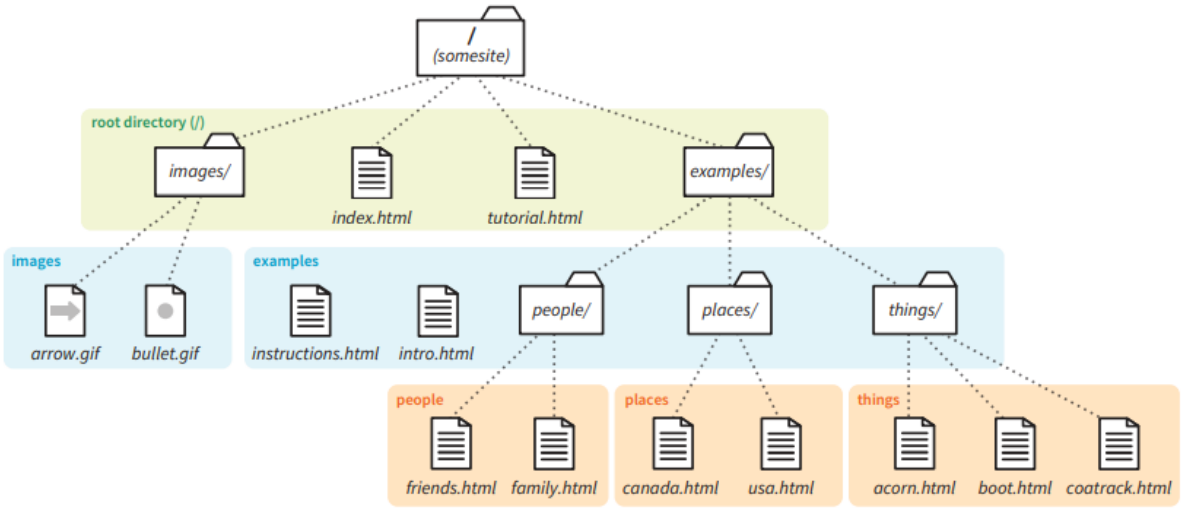
https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work

https://www.w3.org/People/Frystyk/thesis/Internet.html

Exercise 02

Reference : *Jennifer, N.R., (2018), Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, 5th edition – Chapter 06*

Test Yourself



This diagram should provide you with enough information to answer the questions. If you need hands-on work to figure them out, the directory structure is available in the test directory in the materials for this chapter. The documents are just dummy files and contain no content. The first one is filled for you as an example.

1. In index.html (the site's home page), write the markup for a link to the tutorial.html page.

   ...

2. In index.html, write the anchor element for a link to instructions.html.

3. Create a link to family.html from the page tutorial.html.

4. Create a link to boot.html from the family.html page, but this time, start with the root directory.

5. Create a link back to the home page (index.html) from instructions.html.

6. Create a link to the website for this book (learningwebdesign.com) in the file intro.html.

7. Create a link to instructions.html from the page usa.html.

8. Create a link back to the home page (index.html) from acorn.html.

We haven’t covered the image (img) element in detail yet, but you should be able to fill in the relative URLs after the src attribute to specify the location of the image files for these examples.

9. To place the graphic arrow.gif on the page index.html, use this URL:

   <img src=" " alt="">

10. To place the graphic arrow.gif on the page intro.html, use this URL:

    <img src=" " alt="">

11. To place the graphic bullet.gif on the friends.html page, use this URL:

    <img src=" " alt="">

ANSWERS

Test Yourself

1. ...

2. ...

3. ...

4. ...

5. ...

6. ...

7. ...

8. ...

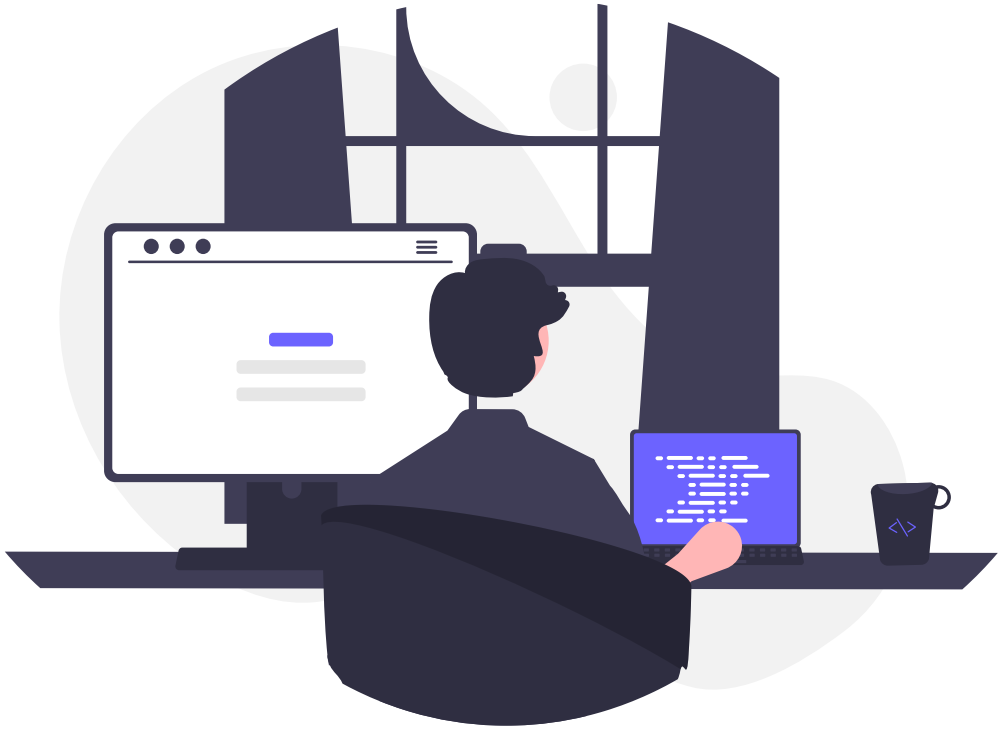
9.

10.

11.

Web Fundamentals

CSCI 11052



Tharuka Vishwajith
M.Sc. in Big Data Analytics
B.Sc. (Hons) in Software Engineering

Overview of Distributed Systems,

- A distributed system contains multiple nodes that are physically separate but linked together using the network.
- distributed system is the collection of autonomous computers that are connected using a communication network and they communicate with each other by passing messages.
- Examples of distributed systems / applications of distributed computing :
  - Intranets, Internet, WWW, email.
  - Telecommunication networks: Telephone networks and Cellular networks.
  - Network of branch office computers -Information system to handle automatic processing of orders,
  - Real-time process control: Aircraft control systems,
  - Electronic banking,
  - Airline reservation systems,
  - Sensor networks,
  - Mobile and Pervasive Computing systems.

Search Engines

- A search engine is a special kind of website that helps users find web pages from *other* websites.
- There are plenty out there: Google, Bing, Yandex, DuckDuckGo, and many more.
- Search engines search for billions of pieces of content and evaluate thousands of factors to determine which content is most likely to answer our query.
- Search engines do all of this by discovering and cataloguing all available content on the Internet (web pages, PDFs, images, videos, etc.) via a process known as “crawling and indexing,” and then ordering it by how well it matches the query in a process we refer to as “ranking.”

How do search engines work?

- Search engines work through three primary functions:

1. **Crawling:** Scour the Internet for content, looking over the code/content for each URL they find.
2. **Indexing:** Store and organize the content found during the crawling process. Once a page is in the index, it's in the running to be displayed as a result to relevant queries.
3. **Ranking:** Provide the pieces of content that will best answer a searcher's query, which means that results are ordered by most relevant to least relevant.

Reference : https://moz.com/beginners-guide-to-seo/how-search-engines-operate

What Is a Firewall?

- A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.
- Firewalls have been a first line of defense in network security for over 25 years. They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet.
- A firewall can be hardware, software, or both.

What's a Proxy Server?

- A proxy server is any machine that translates traffic between networks or protocols.
- It's an intermediary server separating end-user clients from the destinations that they browse.
- If you're using a proxy server, traffic flows through the proxy server on its way to the address you requested. The request then comes back through that same proxy server (there are exceptions to this rule), and then the proxy server forwards the data received from the website to you.

Why Should You Use a Proxy Server?

- To control internet usage of employees and children
- Bandwidth savings and improved speeds
- Privacy benefits
- Improved security

Proxy Server Risks

- Free proxy server risks
- Browsing history log
- No encryption

Cache and Cookies

- Cache and cookies are two different forms of temporary storage kept on client's machine to improve the users experience and performance of web pages.
- **Cache** is used to store online page resources during a browser for the long run purpose or to decrease the loading time.
  - A web cache (or HTTP cache) is an information technology for the temporary storage (caching) of web documents, such as HTML pages and images, to reduce bandwidth usage, server load, and perceived lag. Cache is just a collection of data downloaded to help display a web page.
- **Cookies** are employed to store user choices such as browsing session to trace the user pre
  - Cookies are small files that contain information useful to a web site — such as password, preferences, browser, IP Address, date and time of visit, etc. Every time the user loads the website, the browser sends the cookie back to the server to notify the website of the user's previous activity.
  - Cookies have a certain life span defined by their creators and it expires after the fixed time span.

eXtensible Markup Language (XML)

- XML stands for eXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to store and transport data
- XML was designed to be self-descriptive
- XML is a W3C Recommendation
- XML Does Not Use Predefined Tags
- XML is Extensible

XML - example

```
<book category="cooking">
   <title lang="en">Everyday Italian</title>
   <author>Giada De Laurentiis</author>
   <year>2005</year>
   <price>30.00</price>
 </book>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

https://www.w3schools.com/xml/default.asp

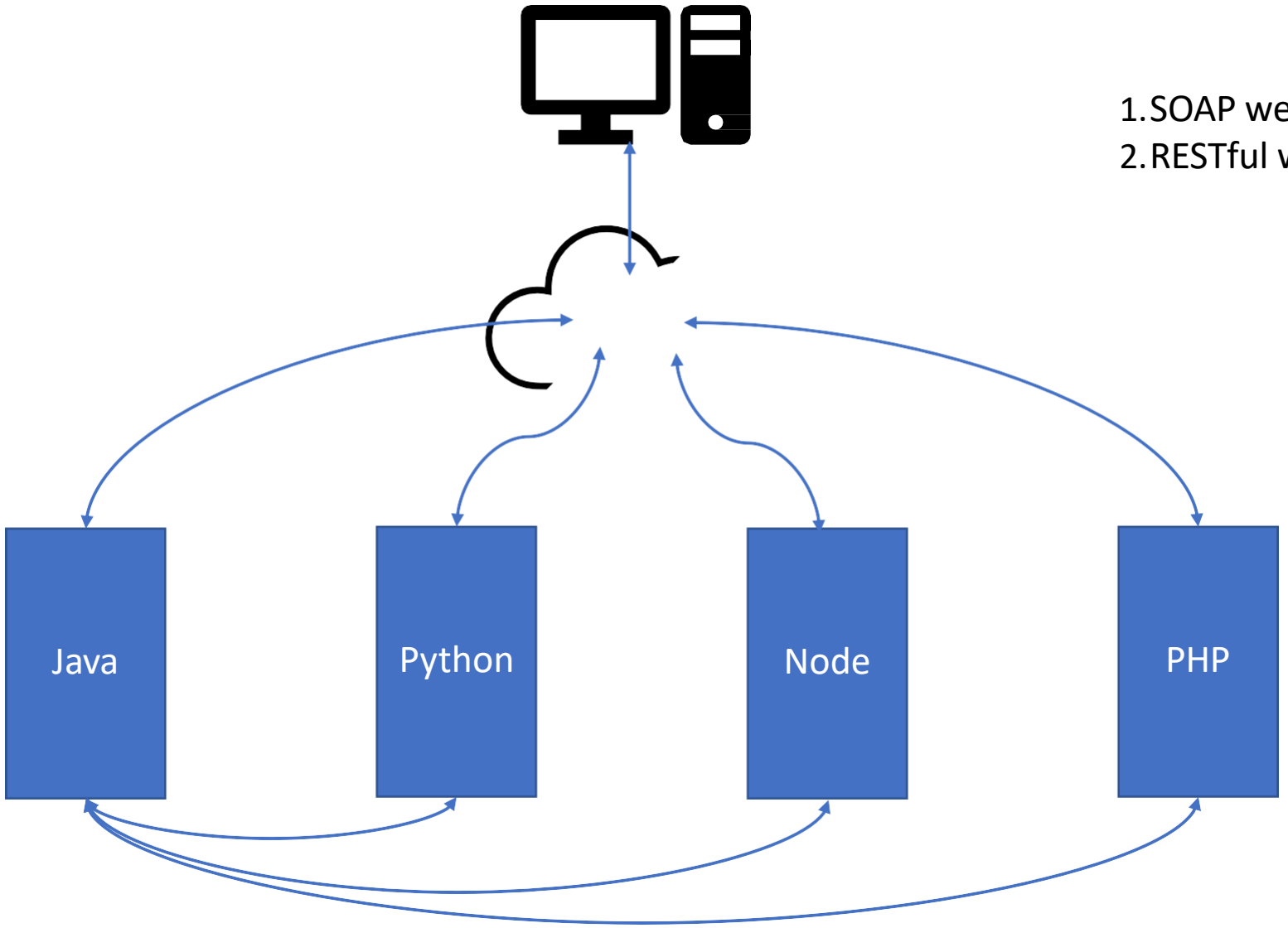
VoiceXML

- The Voice eXtensible Markup Language (VoiceXML) is an XML-based markup language for creating distributed voice applications that users can access from any telephone.
- VoiceXML supports dialogs that feature:
  - Spoken input
  - Telephone keypad input
  - Recording of spoken input
  - Synthesized speech output ("text-to-speech")
  - Recorded audio output
  - Telephony features such as call transfer and disconnect
  - Dialog flow control
  - Scoping of input

Web services

- A web service is a generic term for an interoperable machine-to-machine software function that is hosted at a network addressable location.
- A web service has an interface, which hides the implementation details so that it can be used independently of the hardware or software platform on which it is implemented, and independently of the programming language in which it is written.
- This independence encourages web service based applications to be loosely coupled, component-oriented, cross-technology implementations.
- Web services can be used alone or with other web services to carry out a complex aggregation or a business transaction.

Web services



1. SOAP web services.
2. RESTful web services.

A Brief History of the Web

- Started at CERN, a particle physics lab in Geneva, Switzerland
- 1989: Tim Berners-Lee proposed a system for sharing documents via “hyperlinks”
- 1990: Prototypes built, first by Tim B-L, then Robert Cailliau
- 1992: Approximately 25 servers worldwide
- 1993: Web opened up for commercial use

The World Wide Web Consortium - W3C

- The World Wide Web Consortium (called the W3C for short) is the organization that oversees the development of web technologies.
- The group was founded in 1994 by Tim Berners-Lee, the inventor of the Web, at the Massachusetts Institute of Technology (MIT).
- In the beginning, the W3C concerned itself mainly with the HTTP protocol and the development of the HTML.
- Now, the W3C is laying a foundation for the future of the Web by developing dozens of technologies and protocols that must work together in a solid infrastructure.
- For the definitive answer on any web technology question, the W3C site is the place to go.

www.w3.org
www.w3.org/Consortium/ (Includes what they do)

Web server

- A *web server* is a computer hosting one or more *websites*.
- "Hosting" means that all the *web pages* and their supporting files are available on that computer.
- The *web server* will send any *web page* from the *website* it is hosting to any user's browser, per user request.

Web server (cont'd)

IP address

A unique number assigned to a device connected to the internet (IP = Internet Protocol). Example: 199.27.145.64

Domain Name System (DNS)

A system that allows internet users to refer to servers by name rather than number

Domain name

A name assigned to a web server (easier to use than IP numbers). Example: oreilly.com

DNS server

A server that matches domain names to their respective IP addresses

The Browser

- The software that requests data or documents from the web server
- Also referred to as the client or user agent
- Could be on a desktop machine, smartphone, other connected device, or an assistive device such as a screen reader
- The program in the browser that interprets HTML/CSS/JavaScript is called the rendering engine

Internet tools

A variety of browsers - Because browsers render pages differently, you'll want to test your pages on as many browsers as possible, both on the desktop and on mobile devices.

Windows:

Internet Explorer
(the current version and at least two prior versions)

Chrome

Firefox

Safari

Opera

Macintosh OS X:

Safari

Chrome

Firefox

Opera

mobile browsers - Mobile Safari (iOS) , Android Browser (Android) , BlackBerry Browser (RIM) , Nokia Series 40 and Nokia Browser for Symbian , Opera Mobile and Mini (installed on any device) , Internet Explorer Mobile (Windows Phone) , Silk (Kindle Fire)

Internet tools

A file-transfer program (FTP)

- An FTP program enables you to upload and download files between your computer and the computer that will serve your pages to the web.
- The web authoring tools have FTP programs built right in. There are also dedicated FTP programs, as listed here:

Windows	Macintosh OS X:
WS_FTP	Transmit
CuteFTP	Cyberduck
AceFTP	Fetch
Filezilla	

Internet tools

Terminal application

- terminal (command-line) - allows you to type Unix commands on the server. (setting file permissions, moving or copying files and directories, or managing the server software)
- For Windows users - Linux emulator called Cygwin for commandline access. There is also PuTTY, a free Telnet/SSH client.
- Mac OS X includes an application called Terminal that is a full-fledged terminal application, giving you access to the underlying Unix system and the ability to use SSH to access other command-line systems over the Internet.

Server-side vs. Client-side

Indicates which machine is doing the processing:

- Client-side **applications run on the user's machine**
- Server-side **applications use the processing power of the server**

TERMINOLOGY

Server-side and Client-side

Often in web design, you'll hear reference to "client-side" or "server-side" applications. These terms are used to indicate which machine is doing the processing. Client-side applications run on the user's machine, while server-side applications and functions use the processing power of the server computer.

Technology stack

Web programming can be briefly categorized into client and server coding. Also, it needs database technology if needs to store data.

Client-side coding:

- HTML / HTML 5
- CSS / CSS3
- JavaScript
- Angular
- Vue
- Etc.

Server-side coding:

- PHP
- ASP
- Python
- Node
- Java
- Etc.

Database

- MySQL
- Oracle
- PostgreSQL
- MS SQL
- MongoDB
- CouchDB
- Etc.

Client-side coding – Code host in server, Execute in browser

Web browser:
Google Chrome
Firefox
Edge
Safari, etc.

Server-side coding – Web Server or Application Server

Web Server:
Apache
nginx
IIS
etc.

Application Server:
Tomcat
WebLogic
Glassfish
JBoss
etc.

Database– Relational databases / NoSQL database

Hosted in a server

Every **Web Developer** must have a basic understanding of **HTML**, **CSS**, and **JavaScript**.

HTML	CSS	JavaScript
Basic HTML	Basic CSS	Basic JavaScript
HTML 5	CSS3	ECMAScript 5

Step 1 : Competent with **HTML** and **CSS**,

Step 2 : Look at some **Frameworks**.

- On the **CSS** side you should choose a framework for responsive web design:
  Bootstrap / **Material Design** / **W3.CSS**
- On the **JavaScript** side you should learn at least one modern framework:
  React.js / **Angular.js** / **Vue.js** / **W3.JS**
- Maybe the popularity of **jQuery** has passed the top, but it is still the most used JavaScript framework.

AT A GLANCE

Web-related technologies:

- Hypertext Markup Language (HTML)
- Cascading Style Sheets (CSS)
- JavaScript and DOM scripting
- Server-side programming and database management