15.1 The Internet & 15.2 The web

Notebook:	How Computers Work [CM1030]
Created:	2019-10-09 10:09 AM **Updated:** 2020-01-18 3:28 PM
Author:	SUKHJIT MANN

Cornell Notes	**Topic:** 15.1 The internet & 15.2 The web	Course: BSc Computer Science
		Class: How Computer Work [CM1030]-Lecture
		Date: January 16, 2020

Essential Question:
What are the various technologies/protocols that make up the World Wide Web and its Websites?

Questions/Cues:
- What is SMTP?   - What is VoIP?   - What is the WWW?   - What is a Domain Name?   - What is DNS?   - What is a Cluster?   - What is a Virtual Server?   - What are Virtual Machines?   - What are Containers?   - What is LAMP?   - What is MEAN?

Notes

RECAP: The link, network & transport layers are all about getting packets from one end to another. The application layer on the other hand, is more about the content of the communication, what data is sent and how that data is used by your applications.

- SMTP (Simple Mail Transfer Protocol) = When email servers send emails to each. This protocol used for comms between servers. Also used to send email from your comp to server
    - Comps & phones use different protocols to fetch emails from server, POP/POP3 designed to downloading email to your comp & storing it there. IMAP stores email on the sever, but lets you access it on your phone or comp.
        - Microsoft has own email protocol called Exchange
    - Nowadays, we use webmail- which is email access to on a webpage, not an app on your comp, protocol used here is HTTP for viewing your email, not downloading them

- VoIP (Internet Telephony) = telephone convo by sending sound of your voices over the Internet rather than traditional phone lines
  - phone convos nowadays are all sent using IP. In VoIP, when making your comp sends request to VoIP server for connection, once connection is established, peer-to-peer comms begins
- The World Wide Web(WWW) is a client/server protocol, involves client sending a request for webpage using an HTTP request to server, which creates page & sends it back.
  - We humans use URLs (Uniform Resource Locators) to locate and identify webpages, but internet protocols don't understand URLs, they use numerical addresses called IP addresses
  - The middle part of URL ie. **www.gold.ac.uk** is address of a web server, which the comp needs to contact the server.
- Domain Name = The readable part of the address
  - This is easy for us to understand, but not network protocols
- Domain Name Server (DNS) = converts Domain names in IP address. An DNS server knows how to convert domain names into IP addresses.
  - To find out IP address of the domain you want to use, your client sends a request to DNS server, which sends back an IP address, which your comp can use an HTTP request to the server
  - Most apps will DNS first to get a hold of an IP address before starting main comms
    - This means it doesn't need to use the full reliable channels setup by TCP. It only involves two msgs & if your client doesn't get a reply it know something is wrong. This is why it's an example of an app protocol that uses UDP in the transport layer
- Modern websites use a combination of a database, assets(audio, video, images), & a template to generate the page to your browser when requested.
  - Even more modern websites don't even use the file system anymore or even include a file in the URL. They use a piece of software called a routing engine to interpret URL & work out what web page to generate. All elements of URL after domain name essentially become arguments to a single routing scripts. Routing on a web server is working out which bit of the server program to use to generate a web page
- Any client-side interaction is done in your web browser & is implemented in a lang called JavaScript
- Templating, previously discussed as server-side activity, but is now client-side. Site template are sent directly to client, data is fetched from database separately & sent to client independently from the template. It is client that combines the two.
  - Advantage of this is, server doesn't have send whole webpages every time changes, just sends little bit of data and client updates the user's visual.
  - Also output doesn't have to be HTML and the client viewing doesn't have be web browsers. Mobile apps use a similar framework, in that they also receive webpages, but use it in a different way to display on a phone screen.
- Cluster = multiple comps networked to perform single function
- Virtual servers = server that are implemented in software, so that several virtual servers can run on a single comp
- Virtual machine = one comp implements machine lang of another comp in software, so each virtual machine seems like its own independent comp with its own OS & app software, but it's actually implemented in software in another comp.
- Containers = like a virtual machine except all of the containers share the same OS kernel, but otherwise act like different comps with different app software.
- Most websites are now implemented on cloud servers. These cloud server use both clusters & virtual servers. They consist of huge clusters of comps, and each website has one or more virtual servers implemented not on in a single comp but on a cluster.
- LAMP = is a "stack", collection of software that run websites: L for Linux (OS), Apache (Web Server), MySQL (Database management system) & P for PHP, Python, or in the early days Perl. In the early days, LAMP wasn't distributed as a single unit.

- MEAN= also a stack: M for MongoDB (different database management system), E for Express (web app framework that does a number of things, but particularly routing), A for Angular (HTML generation & templating engine) & N for Node (web server framework). MEAN doesn't include OS in its stack, Node has some features of Apache but all of them so its used in conjunction with other web server software. PHP is prog lang, but none of the MEAN apps are a lang; Node, Angular & Express are all based on JavaScript, a lang that originated as a client-side scripting lang, but Node make it possible to run it on the server.
  - Angular & Express are frameworks that do many of the things that would otherwise have to written by hand, by a PHP programmer, so they make it easier to write complex websites

Summary

In this week, we learned about the Internet app protocols like SMTP, IMAP, VoIP. Alongside this, we explored the make of website technology with Domain names, DNS, Databases, & Templates. Finally, we looked at the various ways a web server could and is implemented ie. clusters or containers.