Application Layer: DNS. (Lecture-8)

- Domain Name System maps hostnames to IP addresses.
  ↳ Internet Governance Part II.
  - Internet Engineering Task Force is a community for making Internet standards.
    ↳ (They are released as Request for Comment.)
- IP address allocation & DNS allocation is done by Internet corporation for Assigned Names & Numbers (ICANN).
- Top Level Domain (TLD) has the domain names ending with, .com, .org, .edu, etc, ...

  → TLD has rules on who can use it.
- Root name server is the root of the DNS hierarchy. 13 root name servers keep a registry of where to go to for every TLD.
- TLD servers have a registry of every domain name registered on their TLD.

  → They can tell you where to find NMSU.edu's DNS servers, NMSU.edu's subdomain space is called a DNS zone.
- Name Server (NS) Records.

  → what domain names to go for DNS info in this zone.

  Eg:) Sent along with A(IP) AAAA records

so you know how to reach them.

- Start of Authority (SOA) Records.

Contains:- 1) The NS, which created this SOA Record.
2) An e-mail contact address for DNS issues.
3) Minimum TTL.
4) Serial Number of latest address info.

- CName Records

→ A record that says, Look for me elsewhere.

- TXT Records

→ Can hold any arbitrary text. It is used to prove to a search engine (or) CA that we control the DNS for a Domain name.

- PTR Record.

→ A reverse-DNS record used to announce that an IP address is bound to this Domain name.

→ we can't look up an IP address to see Domain names unless they give a PTR record.

- CAA Record

→ Lists which certificate Authorities (CA) are allowed to issue certificates for a particular Domain.

HTTP: (Lecture 9)

→ It is a protocol to Request & Recieve Hyper Text & Hyper media. The standard port is "80".

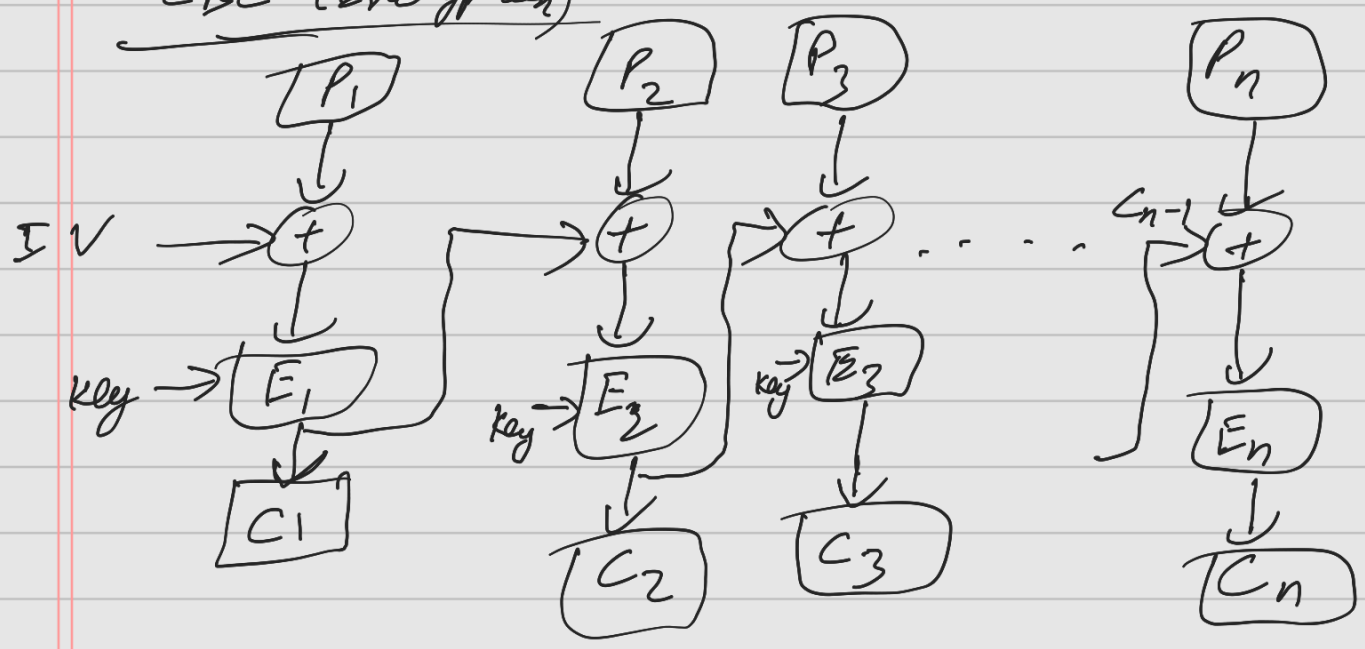
- **HTTP Request methods:-**
  - → **GET:** To get Data from a resource.
  - → **PUT:** To update Data at a resource.
  - → **POST:** To create Data at a resource.
  - → **DELETE:** To Delete Data at a resource.
  - → **Patch:** To partially update table at a resource.
  - → **Idempotent:** any of the above methods has no change on requests then, it is said to be idempotent.
- **HTTP Response Codes:**

101 — Switching protocol
200 — Success
204 — Success, no content.
301 — Moved Permanently
304 — not modified
400 — Bad Request
404 — not Found
500 — Server Error.
503 — Service Unavailable.

• AES

→ It is a symmetric Encryption & Decryption.
→ Message encryptors and decryptors both know the key.
→ works on a fixed sized blocks, one at a time.
→ 128 bit block size, 3 key sizes.

• CBC (Encryption)



• Safe key Length (Size)

Symmetric Encryption – use AES 256.
Hash Functions – use SHA-256 (or) SHA 3.
Asymmetric RSA – use 2048 bits (or) more, 112 bits for Elliptic curves.

Ex's: Bittorrent, ToR, web-servers, etc.,

Properties of Distributed Systems:-

i) Multi-Agent:- Made up of independent machines (or) Programs working together.

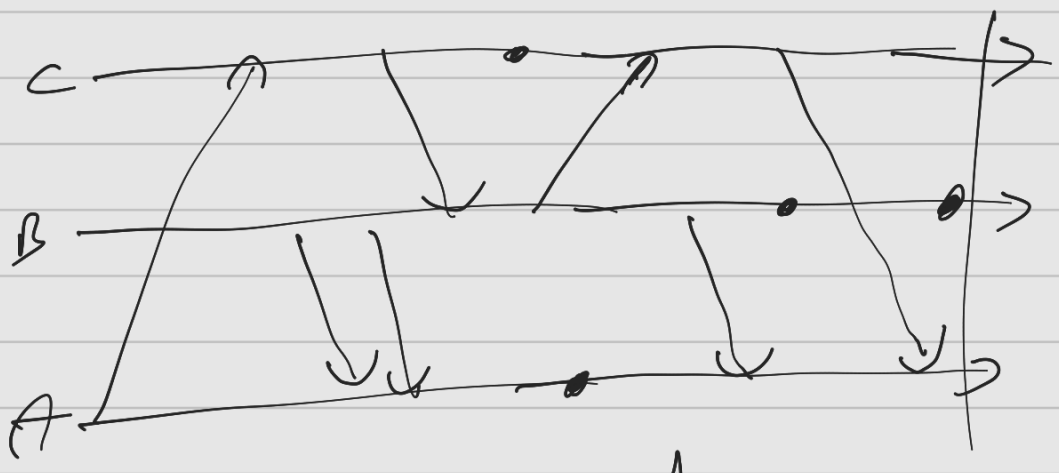
ii) Transparency:- It acts like a single system.

iii) Heterogenity:- It can be made up of different types of agents. (Symbols).

iv) Fault Tolerance:- Parts of it will Fail, but the whole system will still work.

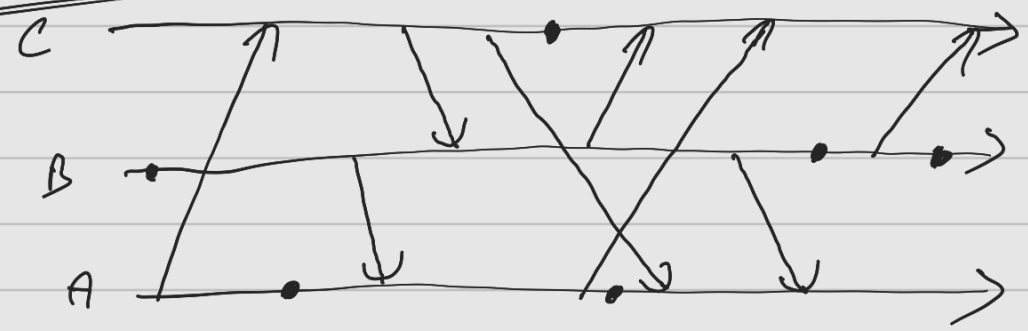
v) Concurrency:- Multiple agents are Running at the same time.

Lamport Clocks:-



Process A	Process B	Process C
A: 2	A: 0	A: 1
B: 4	B: 6	B: 3
C: 3	C: 1	C: 7

vector clock :-



Process A	Process B	Process : C
A : 4	A : 1	A : 3
B : 4	B : 7	B : 6
C : 2	C : 1	C : 3

Consensus : It means that everybody (or almost everybody agrees, so that we can move forward.

Fail-Stop Tolerance :-

→ Need to have majority of votes to outvote to move forward.

Solving Fail-Stop Tolerance :-

$(N > 2f) \Rightarrow \boxed{N = 2f+1}$

→ If we have 'f' fail-stop failures, we will have $(N-f)$ working nodes, who will all agree.

→ If $(N-f) = 0$, Then there's nothing we can do.

→ If $(N-f) > f$, then we have a majority, so we can move on even if 'f' nodes never participate.

Ex:- If, $N = 7$ then Max. Failures = 3.

majority nodes = 4. should be > Max. Failures.

Byzantine Fault Tolerance :-

$3\delta + 1 = N$ → Solving Byzantine Fault Tolerance.

$(N \geq 3\delta)$

Ex:- If $N = 25$, then

$$3\delta + 1 = 25$$
$$3\delta = 24$$
$$\delta = 8.$$

$N = 5000$, then

$$3\delta + 1 = 5000$$
$$3\delta = 4999$$
$$\delta \approx 2000.$$

→ what is the probability of one node failing [.001%] when the nodes are 5000.

$$p = .00001$$
$$\times \quad 5000$$
$$(.05)\%.$$

→ what is the chance of probability for two nodes failing for the above.

$$(.05)^2\%.$$

→ $(N-\phi)$ is the no. of honest, working nodes.
→ Once we hear from $(N-\phi)$ nodes, we have to make a decision to avoid the Byzantine nodes stalling forever.
→ But, the honest nodes might also be all the slow ones, so among our group of $(N-\phi)$ nodes, we make a decision on, we still have to consider that up to ϕ of them could be the Byzantine liars.
→ So, $(N-2\phi)$ is the no. of honest nodes we can be sure of.
→ Until we hear from $(2\phi+1)$ nodes, the honest nodes could still outvote the current favorite.

Deadlock prevention options:-

→ Don't use any mutex locks.
→ Never wait for a mutex while holding another mutex lock.
→ Allow processes to forcibly acquire mutex locks from another threads.
→ Set up the threads so that there is never a "circular wait".

Bit-Torrent (Lecture-12):- created by Bram Cohen in 2001.

↳ Originally built on HTTP & FTP and extended to use HTTPS.

→ "peers" on the n/w work together to split up large files.

→ Clients can request all the pieces at once from the nearest peers, more efficiently uses the n/w.

N/w benefits of Bit torrent:-

→ Original uploader (or) "seeder" needs less upload bandwidth.

→ Downloaders get their content faster.

→ multiple seeders can send the file. (Popular files are even faster).

→ Data sharing costs are shared democratically.

N/w Disadvantages of Bit torrent:-

→ We can get involved in criminal conspiracy.

→ we can't change the torrented file to correct errors, it has to be replaced.

Bittorrent Topology:- Trackers - keeps track of who is seeding.

Leechers - Downloaders who may (or) may not be also helping others.

Seeders - Have a copy of a file and share pieces.

Avoiding too many upload requests while downloading:-

Choking:

→ Answer Requests 4 at a time, choke all the others.

→ Open a 5th connection to see if you can find a more

efficient peer, if you do, replace one of your '4'.
→ If you transfer nothing for 60sec, replace all of your 4 as well.

Sharing Ratio:- → If your bytes downloading equals bytes uploaded, you are sharing fairly.
→ If you are downloading far more than you upload, you are a leech.

Detecting Bit Torrent:-
→ Trackers are available on Port 6969.
→ Trackers know IP addresses and what was downloaded
→ Regular Traffic available on Port Range 6881-6889. but you can choose to run on any Port.
→ Deep Packet Inspection.

Trackerless Torrents:-
→ Remove trust in torrent tracker.
→ Required Peers to keep track of each other & who has what called the 'Bit Torrent Peer Exchange Extension'.
→ Still rely on trackers to add and remove peers.
→ Any peer can give out a subset of other peers to get pieces of files from.

Block-Chain (Lecture-12(1)):-

→ It is an example of one such distributed database, where the data that stored in the block-chain is validated and is maintained by set of individuals across the block chain ecosystem, who are called as Miners.

→ Access protected writes to an authoritative database, & Transactions, timestamps, contracts, etc.

Block chains Replace: → Authoritative access control replaced with distributed consensus.

→ Database state dependent upon majority agreement of update validity. (Majority voting)

(2 million Bit-coins available to mine in the world)

How Hard is the Game:-

→ For 'N' zeros, have to try $2^{N/2}$ nonces.

★ → The Honest majority agreement on the blockchain ecosystem among the Miners is 51%.

★ → Transactions written to the blockchain are not mutable.

★ → The No. of 0's Preceding the Hash is the metric that identifies whether a Miner has done enough work to mine a block.

★ → 2 Miners can't mine a block Together.

★ → The blocks that are not valid in the block-chain ecosystem are called as uncle blocks.

Software-Defined-networking (Lecture-(3)):-

Control Plane:- decides and communicates routing rules and policies.
→ Low bandwidth, high CPU needs.

Data Plane:- carries out decisions from the control plane.
→ High bandwidth, low CPU needs.

SDN Topology:-

Applications: Flow Optimization, management, visualization

↕ northbound API's.

Control Plane (SDN controller)

↕ southbound API's.

Dataplane: Top of rack, switches, TOR switches, emulated switches, etc.,

★ **Open Flow:-** → It uses a TCP in a client-server model where the controller is the server and the switches are the clients.
→ The controller sends rules to the switches to follow, the rule can match combinations of things like SRC MAC, DST MAC, SRC IP, DST IP, etc. (anything from layer 2 packet header).
→ The rule points to an action, usually the layer-2 interface where this switch should forward the packet to.
→ It can also drop packets, NAT, send to multiple outputs.

Open Flow Rule Tables:- They also store,
→ statistics on how many packets & bytes have been recorded on this rule
→ Switches with powerful CPU's can execute simple Programs on data flows.
→ Rules can also state their level of Priority.

Open Flow Rule Example Activity:-
→ Implement a Firewall rule blocking incoming HTTP requests
→ Implement a Firewall rule blocking any outgoing connection to a BitTorrent tracker.
→ Implement a Rule for NAT.
→ Implement a Rule to only allow SMTP server to respond to the internal N/w.

★ What happens in OSI Layer when we enter a URL?

Ans) When we open a URL in a browser, many things happen, but in particular there are a no. of N/w requests that are made. Each one of those N/w requests involves the OSI model.

Now, Assuming we have the DNS info of google.com,
→ In the Application, Presentation, Session Layer the browser creates the GET request and tells the N/w stack to send it to google.com
→ In the Transport and Network Layer, the TCP/IP stack breaks the request up into packets, and sends them out over the datalink. It resends packets as necessary and manages the rate at which the packets are sent. This is essentially the O.S.

→ The data link layer wraps each packet into N/w frame, and sends it over the physical link to the next switch (or) router that the computer is connected to. This is essentially the N/w card.

→ The Physical link is the wire and electrical signals that actually transmit the data.

Lecture-8 (DNS):

→ A(or) AAAA Records: Older 'A' records should be set to a faster pace TTL as IP address lookups change frequently. A standard baseline would be around the 1-hr mark, but may need to be adjusted depending on how often changes need to be made on an emergency basis.

→ CName Records: - It is recommended to set the TTL for something longer than avg session time (which is btw 2-3 mins).

→ TXT Records: → TTL is anywhere within 1-12 hr timeframe.

→ MX Records: - → TTL can be set btw 12-24 hr range. In case of emergencies, it may be set to 1-4 hr range.