Q1- DNS-related: Consider a scenario of a user browsing the web from the machine storm.cise.ufl.edu, accessing an article in a website at URL:

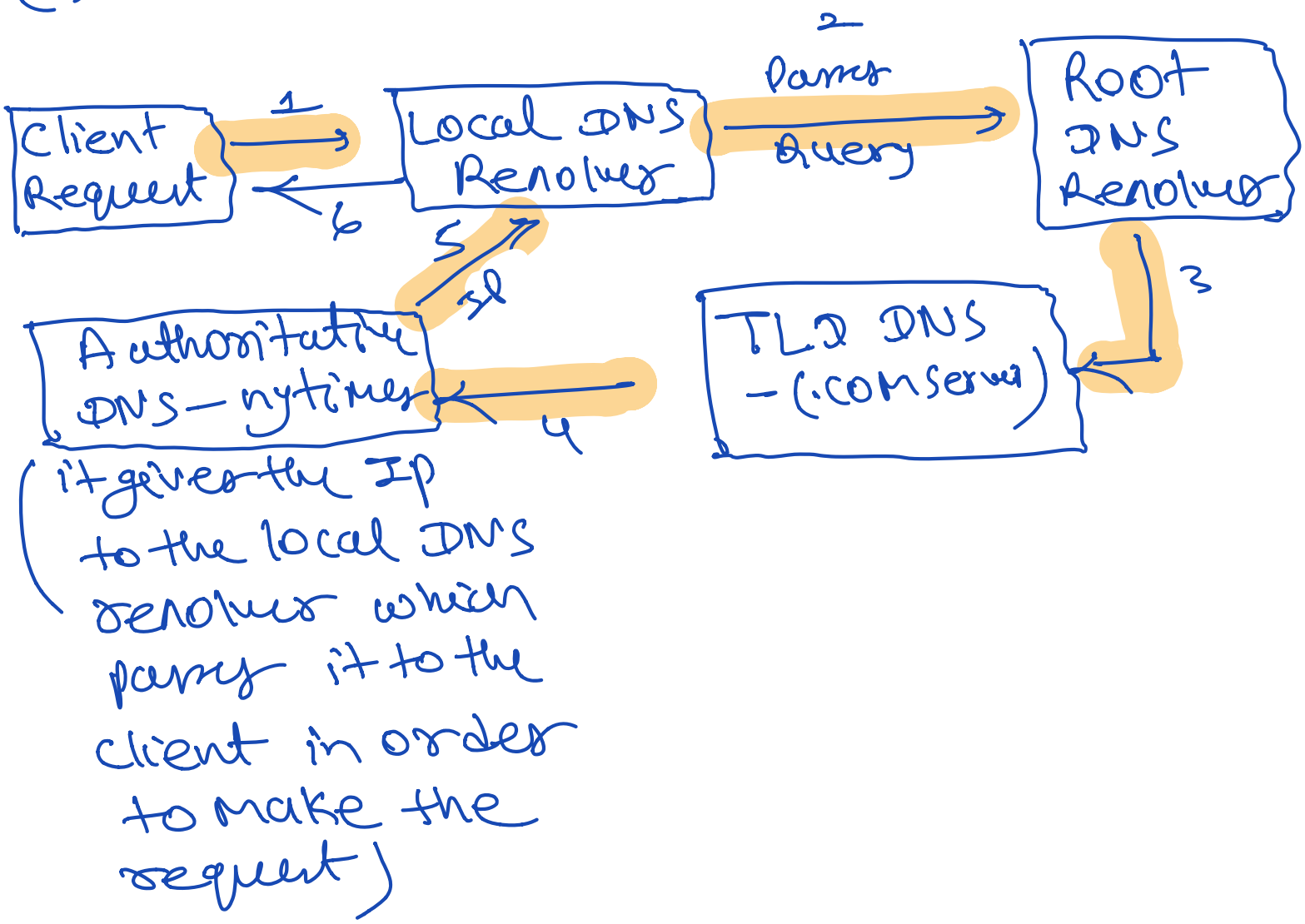
www.nytimes.com/2019/09/26/technology/ai-computer-expense.html

The user performs three accesses: - using http, - using https and - using port 8080.

A. Show the sequence of DNS servers queried to resolve the URLs (assume no caching)
B. Write the complete URL for each of the three accesses [you may have to edit the URL]
C. Write the four tuple [src address, src port, dst address, dst port] for each of the accesses

Ans.1 =

(A) =



(it gives the IP to the local DNS resolver which parse it to the client in order to make the request)

(B) =

1) = Http =

http://www.nytimes.com/2009/09/26/technology/ai-computer-experiment.html

2) = Https =

https://www.nytimes.com/2009/09/26/technology/ai-computer-experiment.html

3) = Port 8080 =

www.nytimes.com:8080/2009/09/26/technology/ai-computer-experiment.html

(C) = Http =

[128.227.205.236, 59853, 151.101.49.164, 80]

2) = Https =

[128.227.205.236, 58263, 151.101.49.164, 443]

3) = port 8080

[128.227.205.236, 58273, 151.101.49.164, 8080]

Q2- Discuss how the following technologies (or their variations) help in improving the performance of content distribution networks (CDNs):

A. HTTP
B. DNS

Ans:2 =

HTTP = with the help of Http CDN's can cache the static web contents for example templates, css files, Javascript files, music, videos etc. caching web content helps in improving performence of web requests and reduce the response time. to cache the web content Http Headers are used.

DNS = Request routing is the critical issue in CDN's. it directs the end users to optimal surrogate servers. DNS based Request routing is most popular in CDN's. DNS which is a domain name server translate a domain name to an Ip address and can help CDN's to redirect user request to nearest surrogate server to the user to get response. it improves response time and hence helps in Performance Improvement.

Q3- Elaborate on the data '*push*' vs '*pull*' in the context of

A. http vs SMTP
B. peer-to-peer network hierarchy communication (e.g., super nodes, group leaders)
C. Proxy and web caching
D. CDNs

Ans-3 = Push Protocol = client opens a connection to the server and keeps it constantly active

Pull protocol = In pull protocols the client periodically connects to the server checks for and gets recent events then closes the connection & disconnects from the server.

(A) = Http Vs SMTP =

SMTP ⇒ A Push Protocol

HTTP ⇒ A pull Protocol

In SMTP the sender mail server pushes the data onto receiving mail server by initiating a TCP connection. This connection is maintained.

In HTTP client pulls information available on a server by initiating a TCP connection intermittently. i.e. it connects to server periodically for different requests and once request is served connection is closed.

(B) = Peer to Peer network Heirarchy communication (Super nodes, group leaders)

Group leaders pull from other group leaders.

Group members push to group leaders.

(C) = Pull based model is used in the Proxy Service. if the cache is maintained by Proxy Server.

Cache model is push based if the web Server initiates the cache at this level.

(d) = Push CDN = content is distributed proactively among the edge servers in chosen CDN locations, web content is populated in CDN POP closest to end users location. So when the end user sends a request for a file, CDN has it ready and its delivered seamlessly.

Pull CDN = when the end user sends the request for the web content it pulls it down from the nearest edge server. all the content is cached in one place and the CDN does the work to pull it down into the end users browser.

Q4- Someone suggested to use a local file called *hosts.txt* on each machine instead of DNS. Discuss the advantages (at least 2) and disadvantages (at least 2) of such suggestion.

Ans-4 = Advantages & Disadvantages of using hostn.txt instead of DNS.

Advantages =

(1) = Increased browsing speed because lookup time is less as the lookup table is available on the name computer locally and a separate server is not queried for domain name translation.

(2) = configurability / Block Spyware ⇒ By using hostn.txt file we can add a list of known ad networks and map it to 127.0.0.1, This will allows us to block these sites from being able to be accessed.

Disadvantages =

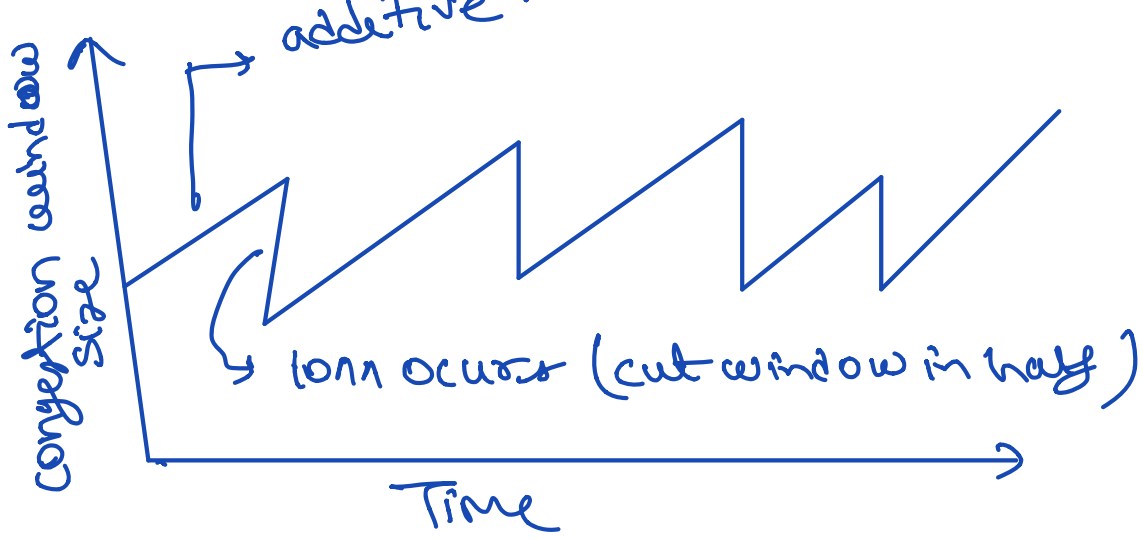
① = Difficulty in maintenance for ex. if a new server is added, the hosts.txt file has to be updated with the new server entry, any IP address change should be updated as well and this has to be done in all hosts.txt files of all connected machines. In absence of a central server this is difficult while in case of DNS it could be done easily, changes has to be made only at one place which are available for all.

② = hosts.txt file can easily be corrupted by malicious software or humans which can result in network problems.

Q5- What is '*saw-tooth*' behavior in TCP, and what is causing it?

Ans.5 =

TCP congestion control follows the additive increase multiplicative decrease (AIMD) mechanism. In this approach when the sender increases the rate of transmission i.e. the window size until there's loss. increase congestion control until loss is detected and cut in the half once the loss occurs. This is multiplicative decrease. if we plot the usage on the graph there would be steady increase and it suddenly falls straight. This behaviour repeats resulting in saw tooth behaviour:



Q6- A TCP flow and a UDP flow walk into a network, sharing a bottleneck link …. Complete the story, detailing the packet rate dynamics if the link gets congested, and comment on the end result.

Ans-6 = if A TCP flow and a UDP flow walk into a network sharing a bottleneck link. then UDP flow will continue transmitting data packets through the bottleneck link. Since UDP doesn't care about transmission bcz it will keep on transmitting data. UDP doesn't have any congestion control hence transmission is not regulated.

On the other hand TCP regulates the data transmission in case of congestion in the the shared link. TCP would send the packets effectively after UDP has completed the transmission at that point the shared link would be less congested and TCP will start sending data packets more effectively.

we will observe that for same transmission rate UDP will have more packets in network than TCP.

Q7- TCP is supposedly fair, dividing the bandwidth between competing TCP flows. You want to transfer a huge file fast, suggest a way of doing so using TCP to get over the fairness delays, and approximate your new bandwidth share.

Ans. 7= multiple simultaneous TCP connection are required to transfer large files across a network. Having more than one TCP connection means the application reserves the larger portion of bandwidth hence allowing for parallel transfer of file chunks. This technique is used by many browsers to transfer large files over a TCP connection.

for Ex. a link with Rate R and 4 existing connections if a new app requests for 1 TCP connection it gets R/5 on the bandwidth, if it is instead asks for 6 connections it gets R/2 (half of the total bandwidth)

Q8- Given that most data-link layers perform error checking (and correction to some extent) why do we need checksum in UDP (and TCP)?

Ans.8 = Although data link layers perform error checking recovery to some extent transport layers has checksum in UDP/TCP headers for the error control.

The error control in the data link layer works at the Packet or stream level while the error control in the transport layer works at the message level. The data link is not concerned with ordering of packets while the transport layer is.

Different networks are designed for different packet lengths so when a packet goes from one network to other it might be fragmented by gateway router These fragments are then reassembled at host machine.

It is possible to do reassembly at the network layer but this causes a lot of problems for Ex. buffering etc. also if this would be done then

it is necessary that all fragments of a packet pass through the same gateway this will compromise packet switching. nature of network. also packet fragmentation can also happen at last gateway

This forces host machine to do the reassembly, Hence in a packet switching network a host machine inevitably faces the task of error correction. which is why in TCP/IP Transport layers have checksums to do error corrections.

Q9- Comment on response to *packet-loss* vs *ack-loss* in

A. Go-back-N
B. selective repeat

Ans.9 =

(A)= Go-back-N =

1)= Packet loss = Sender has upto N non-ACK'd packets in the pipeline, receiver sends cumulative ACK, when the packet loss occurs receiver resends the ACK of received packets with highest inorder seq # when sender receives duplicate ACK it knows to resend the packets with seq # greater than that of duplicate ACK.

2)= ACK LOSS = Sender maintains a timer for oldest in flight packet(n) if timer runs out then sender retransmits packet n and all packets with seq # Higher than n.

(B) = Selective Repeat =

1) = Packet loss = Sender retransmit only the packet which was lost. for ex. packets 1, 2, 3, 4 are in pipeline and packet 3 is lost but 1, 2, 4 are received then receiver sends the ACK for those and stores the packets in a buffer once the sender retransmits the packet and it is received by the receiver, the receiver sorts the packets in correct order and delivers it to the application layer.

2) = ACK loss = Sender maintains timers for each un-ACK'd packet and retransmits it if timer runs out.

Q10- Congestion Signaling:

A. What is meant by implicit congestion signaling and explicit congestion signaling? Give examples of congestion control protocols that use each type signaling.
B. Discuss the advantages and disadvantages of each of the above schemes.
C. What kind of signaling does TCP use to detect network congestion? Explain the different signals that TCP uses for that task.

Ans.10 =

(A)= In Implicit signalling there is no communication between the congested nodes and the source. The source makes the guess that there might be congestion in the network. for Ex. when a sender sends data to the receiver and receiver does not acknowledge the reception of data then assumption is made that there is congestion in the network.

Ex.= TCP incorporates implicit Signalling.

On the other hand in Explicit signal-ling congestion is informed by the node to the source explicitly by sending the data packet.

In Explicit Signalling the congestion information is sent using a separate data packet by the node to source.

Ex:- IP incorporates Explicit Signalling.

(B) =

Implicit Signalling =

(1) = Advantages =

a) = Implicit congestion signal does not require routers Support.

b) = Transport protocol always need to adapt to implicit congestion signals.

(2) = Disadvantages =

a) = Implicit congestion Signal require Higher latency rate to interpret packet loss.

(b) = Implicit congestion signals also result in the problems like zigzag source rates, large Buffer requirements and potential unfairness.

Explicit Signalling =

(1) = Advantages =

(a) = Explicit congestion signal requires the lowest latency to interpret. compared to implicit congestion signal.

(b) = An Explicit congestion signal can differentiate between congestion loss from other loss.

(2) = Disadvantages =

(1) = requires an extra data packet to be sent to inform congestion which can cause control traffic congestion to collapse.

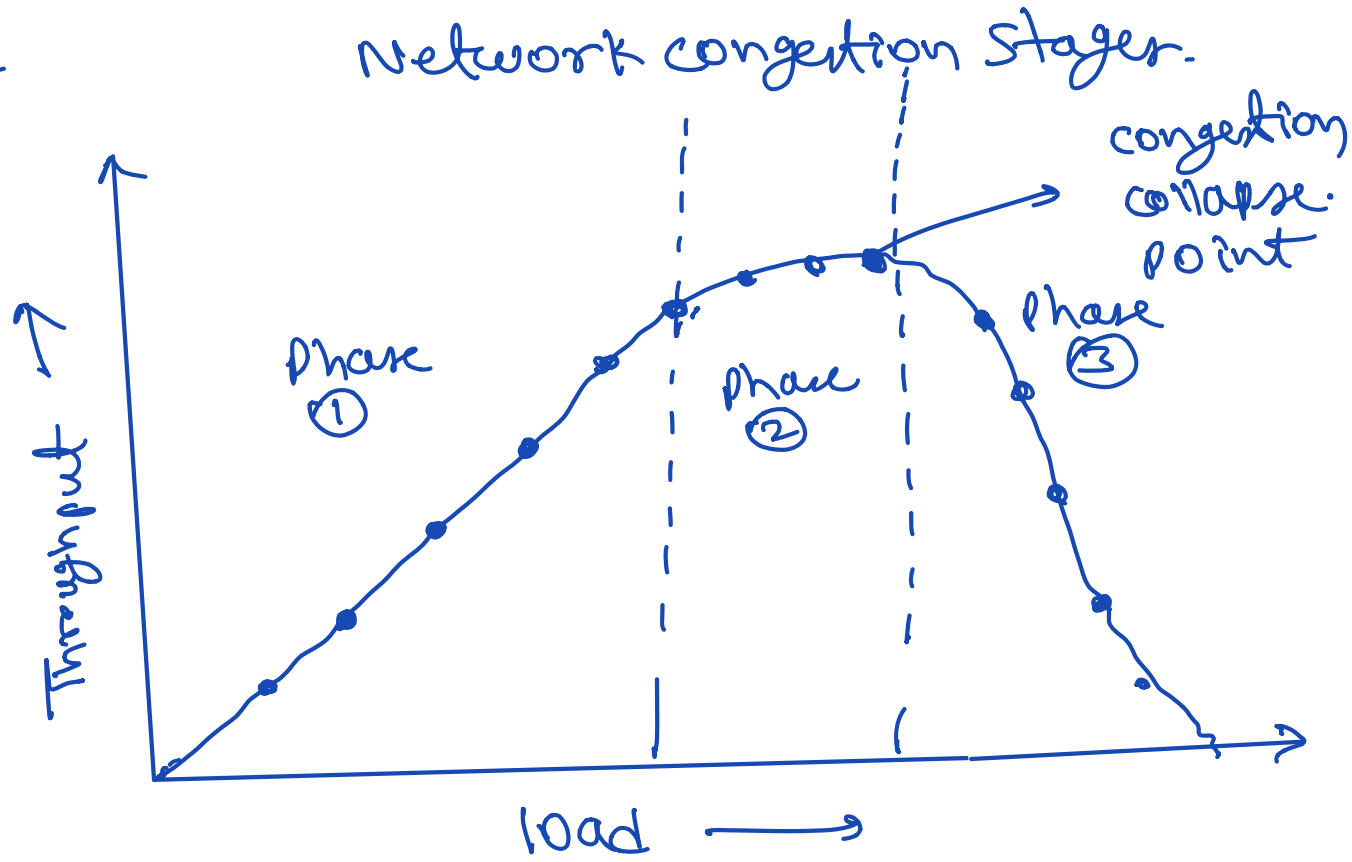
(C) = TCP Protocol uses Implicit Signalling for congestion control.

when a data packet is sent from sender to receiver and sender fails to receive an acknowledgement receipt from the receiver within an estimated time limit In such scenario TCP decreases the congestion window to one maximum segment size (MSS). any other case the congestion window is increased by one MSS. when a retransmit happen the congestion threshold is set to half of the congestion window size.

Q11- Network congestion phases:

A. Describe (with the aid of a graph) the different phases of network load/overload outlining the degrees of congestion with increase of load. Indicate the point of congestion collapse and explain why it occurs.
B. Where does TCP operate on that graph? Explain for the various phases of TCP; slow start, congestion avoidance (due to timeout), fast retransmit-fast recovery triggered by duplicate ACKs.

Ans.11 =



Phase 1 is ideal as the performence is at its best for given load. In this plese no congestion occurs and network perform smoothly.

In phase2 as the load increases, the network starts experiencing moderate congestion, Hence the performence of

the network decreases and throughput tends to saturate and overall through-put decreases.

In the beginning of phase 3 load on the network increases and severe congestion occurs. at this point increased loss of packets and delays which lead to many timeouts. that further adds the retransmission of these lost packets which further increase the congestion. and a chain reaction is triggered and overall throughput decreases exponentially.

(B) = At the begining of phase-1 the load starts at cwnd=1, Then starts growing exponentially. until a loss is experienced. (In phase-2 and 3)

After the loss if a timeout occurs

TCP goes down to cwnd = 1 which is beginning of phase 1 then again ramps up to half of the load that led to the loss. (i.e. half way in phase-1)

In congestion avoidance cwnd increases linearly which means that the load increases slowly towards the end of phase-1 and into phase-2. until another loss occurs.

In fast retransmit fast recovery (due to duplicate Acks) the load is cut in half (half way in phase-1) then slow (linear) increase towards phase 2 as in congestion avoidance.

Q12- Argue for or against this statement (reason using examples as necessary): "Packets are lost only when network failures occur (e.g., a link goes down). But when the network heals (e.g., the failed link comes back up again), packets do not get lost."

Ans.12 = The given statement is not always true.

First of all reasons behind packet loss could be many other than just failure of transmission line —

for Eg.= weak signalling, congestion in network, damaged link, DOS attacks etc.

In case of UDP the statement is again false, because in UDP data packets can get lost even without any transmission link failure. the UDP protocol has no special measure to stop packet loss even after the recovery of damaged link lost data packets cant be recovered neither they are retransmitted.

Q13- Compare and contrast *AIMD* vs *MIMD*. Focus on network stability.

Ans.13 =

① = AIMD (Adaptive Increase Multiplicative Decrease)

AIMD is a feedback control algorithm which is widely used in TCP congestion control mechanism. It combines linear growth of congestion window followed by the multiplicative decrease when the congestion is detected.

a) CWD increment by 1 (additive increase)

b) = when loss occurs CWD size is halved.

AIMD does not reach to network stability AIMD is good for achieving network fairness while being efficient.

② = MIMD (multiplicative increase multiplicative decrease)

MIMD combines multiplicative increase of congestion window and multiplicative decrease when the congestion occurs. MIMD is also unstable & nonconverging MIMD is good and faster for congestion control.