Homework 2: Application Layer Continuation and Transport Layer

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 [Hint: To get IP addresses use networking tools, like ping, traceroute, nslookup, ip address, etc.]
A. First, the end machine contacts the local DNS server; Second, the root DNS server is contacted to get the server with info about '.com'; Third, the top-level domain server for '.com' is contacted to get information about 'nytimes.com'. Fourth, the authoritative server for 'nytimes.com' is contacted to resolve the IP address for www.nytimes.com.
B. For http, just add "http://" in front of the URL, same for "https". To add a port number, we need to add it explicitly in the form 'http://www.example.com:8080/path',
so, in our case it is: http://www.nytimes.com:8080/2019/09/26/technology/ai-computer-expense.html

Q2- Discuss how the following technologies (or their variations) help in improving the performance of content distribution networks (CDNs):
A. HTTP
B. DNS [Hint: Write a one paragraph of ~4-6 lines/sentences on each technology]
A. HTTP variant, 'DASH' for dynamic adaptive streaming over HTTP, adds a level of indirection, and provides a manifest file with pointers to different files (each containing an encoding for the streaming media, corresponding to different resolutions at different data rate). The client monitors the quality periodically and picks the file (from the manifest) that provides the best quality at that time. This allows CDNs to account for the Internet's heterogeneity, and scale better by not needing to transmit at higher rate than the clients (and their connections) can handle.
B. DNS is used to provide a level of indirection for services that are using a 3rd party host (or cloud service) for their streaming. Instead of the DNS authoritative server returning an IP address to the video server, instead it provides a URL to the 3rd party's website and path. This way the information about the 3rd party details can be hidden from the user until they access the video, providing flexibility to change the hosting arrangements in a way that is transparent to the user.

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
A. http uses data pull as it relies on the client initiating the request to transfer the data from the server; SMTP, uses data push from the email sender to its email server, then another push to the intended recipient's email server. The last hop pull is used to get the data to the email agent/reader.
B. In P2P hierarchies involving super-nodes (like Skype), group leaders or cluster heads, the goal is to

attempt to reduce the search overhead and delay.
C. Proxy and web caching: the first time an object is requested it is pulled from the data-origin server. If the object is 'cachable' then it is cached in the proxy and is locally pulled next time a similar request is sent. This saves network bandwidth and reduces user delay. In case of encrypted or dynamic data, customized or copyrighted content, then the object cannot be cached.
D. CDNs: The duplication/copying of content (e.g., videos) by the distributor/company on the distributed servers/cluster in a CDN can be costly. So, the CDN companies perform a pattern analysis, and attempt to replicate the content where and when it is popular, to increase the hit ratio, while keeping the cost of replication down. Pushing the popular data/movies to where it is likely to get accessed saves network bandwidth, reduces delays for the users, and utilizes storage efficiently. For nonpopular content the pull model is used where the request to send to the origin servers to complete the request.

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.
Advantages:
Removing reliance of any 3rd party server (including DNS servers), along with any vulnerability such servers can be subject to.
Reducing delays in name resolution as the name to address mapping is done at the local machine, avoiding DNS delays.
Control the machines accessed.
Disadvantages:
Any change to the name-to-IP address mapping would have to be reflected in the hosts.txt files, by changing the files on all the hosts.
Dynamic mapping needed for load balancing may not be easily implemented.

Q5- What is 'saw-tooth' behavior in TCP, and what is causing it?
TCP uses additive increase multiplicative decrease, increasing cwnd by '1' for every RTT when there is no packet loss, and decreasing the window to half with every packet loss. The slow increase, followed by fast decrease in the window size, results in this saw-tooth behavior in the window size, and subsequently in the TCP throughput.

Q6- A TCP flow and a UDP flow share a bottleneck link. Discuss the packet rate dynamics if the link gets congested, and comment on the eventual result. [Write a paragraph of ~4-6 sentences]
TCP performs congestion control and responds to packet loss by reducing its window size, effectively cutting down its throughput and share of the bottleneck link. UDP on the other hand does not slow down and can effectively squeeze-out the TCP flow by forcing more losses over the bottleneck link, which slows down the TCP flow even further. The result is that the UDP flow wins the major share of the bottleneck link, and the TCP flow loses by having close-tozero throughput. This is under the assumption that UDP rate is higher than the bottleneck link and that the UDP flow is not blocked or rate-limited by firewalls.

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. The new share you will be getting may not be 'fair'!
By using parallel TCP sessions, each sending a chunk of the file. If every session gets R/M rate, then by using N parallel flows instead of one, an end host can get NR/(N+M).

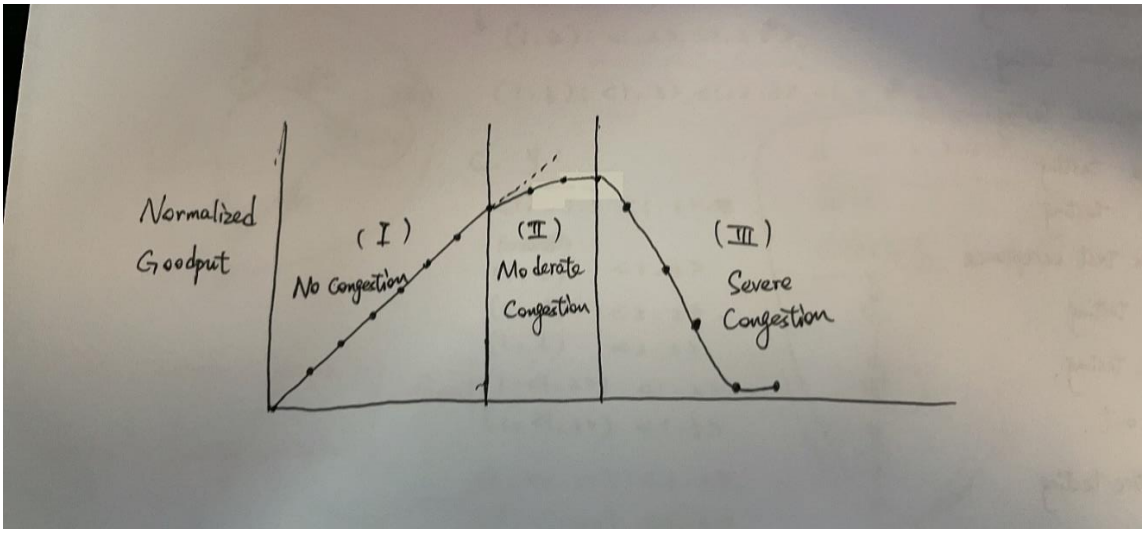
Q8- Comment on (and compare/contrast) response to packet-loss vs ack-loss in
A. Go-back-N
B. selective repeat [Hint: Write ~3-6 lines/sentences for each]
A. Go-back-N uses 'cumulative acknowledgements' so when an 'ack' is lost, the next packet (x+1) will trigger ack x+1 which acknowledges everything up to. So ack-loss in Go-back-N will not trigger any retransmissions from the sender. This is a very different behavior than packet loss in Go-back-N.
B. Selective repeat does not use cumulative acknowledgements. Hence, receiving ack x+1 means that packet x+1 was received, but does not indicate anything about packet x. In this case, the loss of packet x will produce the same behavior as does the loss of ack x.

Q9- Congestion Signaling:
A. What is meant by implicit congestion signaling and explicit congestion signaling? Give an example of a congestion control protocol 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.
A. implicit congestion signaling uses inference based on measurements from the edges/ends of the network, such as delays, round trip times, gaps in sequence numbers, or duplicate acks, to attempt to detect congestion without aid from the network. On the other hand, explicit congestion signaling requires a signal from the network nodes when congestion occurs. It can use congestion thresholds on queue average size to sense the onset of congestion and sends a signal to the end hosts indicating network congestion. Examples include TCP ECN, ICMP source quench, or ATM ABR service.
B. Implicit signaling:
Advantage: operates end-to-end without any change to the network, adhering to the end-to-end design principle of the Internet, allowing complex functionality to be performed at the edges, and facilitating evolution and innovation at the edges.
Disadvantage: since it attempts to estimate network congestion, there a higher chance of error. For example, packet loss could be due to channel bit error rate and not congestion, misleading protocols to cut down their rate when it's unnecessary.
Explicit signaling:
Advantage: there is no 'guessing' involved at the edges. Rather, the edges get a clear signal from the network, based on in-network measurements, that congestion is occurring. Sometimes the signal also indicates the level of congestion and suggests a rate for the senders.
Disadvantage: it requires deployment in the network. It also increases the processing complexity in the routers to estimate the onset of the congestion and to send signals to sources who may be the cause of such congestion.
C. TCP uses implicit congestion signaling at the edges. It uses sequence numbers for packets and ack's, along with the timer mechanisms based on estimated round trip times (RTT). The sender looks for gaps in sequence numbers in the acks to infer packet loss and perform retransmission and window adjustments.

Q10- Network congestion phases:

A. Using a graph, describe 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.

A. 

B. For TCP: in slow start, the load starts from cwnd=1, then ramps up quickly until a loss is experienced. After the loss, if a timeout occurs, TCP goes down to cwnd=1 then ramps up to roughly half the load the led to the loss. In congestion avoidance cwnd increases linearly, which means the load increases slowly towards the end of phase I and into phase II, until another loss occurs. In fast retransmit fast recovery, the load is cut in half, then slow increase towards phase II.

Q11- TCP interaction with routing: Argue for or against the following statement: "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." [Hint: Write ~4-6 lines/sentences]

When the network fails, a number of packets that cross that link may be lost. When the network heals, packets/flows may cross relatively shorter paths to get to the destination. Shorter paths have a "delay-bandwidth product" less than longer paths, and hence can hold fewer bytes in the pipe. Having a TCP connection with a high CongWin go over a shorter path may also cause packet loss, since the shorter paths will get congested, and buffers may overflow causing multiple packet losses.